

HOPEX Data Governance

Guide d'utilisation



Bizzdesign

Hopex Aquila

Les informations contenues dans ce document pourront faire l'objet de modifications sans préavis et ne sauraient en aucune manière constituer un engagement de la société Bizzdesign.

Aucune partie de la présente publication ne peut être reproduite, enregistrée, traduite ou transmise, sous quelque forme et par quelque moyen que ce soit, sans un accord préalable écrit de Bizzdesign.

© Bizzdesign, Paris, 2026

Tous droits réservés.

Hopex Data Governance est une marque réservée de Bizzdesign.

Windows est une marque réservée de Microsoft.

Les autres marques citées appartiennent à leurs propriétaires respectifs.

SOMMAIRE



Sommaire	3

Introduction à HOPEX Data Governance	23

Périmètre de la solution HOPEX Data Governance	24

Découverte de données	24
Glossaire métier	24
Architecture des données	24
<i>Trois niveaux de modélisation</i>	24
<i>Catégorie des données</i>	25
<i>Workflow de conception</i>	25
Utilisation des données	25
Qualité et conformité des données	26
<i>Définition des responsabilités</i>	26
<i>Définition des règles et standards</i>	26
<i>Traçabilité des données (Lignage de données)</i>	26
<i>Evaluation des données</i>	26
<i>Rapports d'analyse</i>	27
Data Governance vs Information Architecture	27

Se connecter à HOPEX Data Governance	29

Les profils utilisateurs de HOPEX Data Governance	29
Les rôles métier de HOPEX Data Governance	30

Le bureau HOPEX Data Governance	32

Page d'accueil de HOPEX Data Governance	32
<i>Indicateurs du périmètre</i>	33
Afficher l'environnement de travail d'une entreprise	34
<i>Créer une entreprise et son environnement de travail</i>	34

CATALOGUES DE DONNÉES

Description d'un catalogue de données	39
Contenu d'un catalogue de données	40
Sources de données	40
<i>Métadonnée physique</i>	41
<i>Champ de métadonnée physique</i>	41
Réglementations associées au catalogue	41
Lignage de données d'un catalogue de données	42
Carte de domaines de concepts d'un catalogue de données	42
Infrastructure des sources de données	44
Organisation des dépôts de données	44
<i>Hébergement</i>	44
<i>Utilisation</i>	45
Compléter un centre de données	45
Recherche dans le catalogue de données	46
Effectuer une recherche	46
<i>Condition préalable</i>	46
<i>Filtres de la recherche</i>	47
<i>Filtres des résultats</i>	47
Afficher le détail d'une métadonnée	48
Créer un catalogue de données	49
Workflow du catalogue de données	49
<i>Démarrer le workflow d'un catalogue de données</i>	49
Associer des mots-clés à des métadonnées	50
Définir des catégories de métadonnées	51
Rapports d'un catalogue de données	52
Vue d'ensemble du catalogue de données	52
Rapport de catalogue de données	52
Relier les métadonnées aux informations métier	53
Collecter des données métier	54
Suivi des formulaires de collecte de données	55
Qualité des données du catalogue	56
Relier une politique de qualité au catalogue de données	56
Relier une politique de qualité à des métadonnées physiques	56
Instantané d'un catalogue de données	57

Découverte de données	59
Le module HOPEX Data Discovery	60
Déploiement	60
Options de connexion aux sources de données	60

Lancer HOPEX Data Discovery	61
Conditions d'accès	61
Mode "en ligne"	61
Mode autonome	65
<i>Condition préalable</i>	65
<i>Lancer l'import en mode autonome</i>	65
Import incrémental	65
 Lignages de données	 67
Lignage de données fonctionnel	68
Créer un lignage de données fonctionnel et son diagramme	68
<i>Initialisation du diagramme</i>	68
<i>Workflow du lignage de données</i>	69
Objets du diagramme de lignage de données fonctionnel	69
<i>Participant</i>	69
<i>Donnée métier origine</i>	70
<i>Flux de données</i>	70
<i>Règles de propriété de concept</i>	70
<i>Mesure de qualité des données</i>	71
<i>Logiciels de traitement</i>	72
<i>Technologie</i>	72
Lignage de données technique	73
Présentation de l'import MANTA	73
Conditions préalables	74
<i>Autoriser l'url d'accès aux objets dans MANTA</i>	74
<i>Paramétrer les options CSP (Content Security Policies)</i>	74
<i>Paramétrer un nom de serveur unique lors de l'extraction des métadonnées</i>	75
<i>Gestion des révisions dans MANTA</i>	76
Importer des données de MANTA	76
Visualiser le lignage de données technique	77
<i>Accéder au lignage de données technique</i>	77

GLOSSAIRE MÉTIER

Introduction à la conception d'une ontologie métier	81
Le processus de gestion du vocabulaire	81
<i>L'analyse et l'organisation des concepts métier</i>	82
<i>La réalisation des concepts</i>	82

Consulter le glossaire métier	83
Rechercher des termes dans le glossaire métier	84
Condition préalable	84
Périmètre de recherche	84
Lancer la recherche	84
<i>Filtres de la recherche</i>	84
<i>Filtres des résultats</i>	85
Afficher le détail d'un terme	87
<i>Caractéristiques standard</i>	88
<i>Caractéristiques avancées</i>	88
Générer un glossaire	89
Lancer un rapport de glossaire	89
Utiliser le glossaire dans un contexte multilingue	90
Utiliser Data ID Card	91

Définir les informations métier	93
Synthèse des objets utilisés	94
Concept et terme	94
Les liens entre les concepts	94
<i>Les liens de définition</i>	95
<i>Les liens de dépendance</i>	95
Propriétés de concept	97
Instances de concept : les individus	97
Le cycle de vie d'un concept ou d'un individu	99
<i>Le cycle de vie d'un concept</i>	100
<i>Le cycle de vie d'un individu</i>	100
Les périodes	102
La classification des concepts et la notion de type de concept	103
La vue de concept	104
La réalisation des éléments de dictionnaire	104
Présentation des diagrammes de modélisation des concepts	105
Le diagramme du domaine de concepts	105
Le diagramme de structure de concept	106
Le diagramme de structure de concept type	106
Le diagramme de structure de concept d'état	106
Le diagramme de structure d'un individu	107
Le diagramme de structure de cycle de vie de concept	107
Dictionnaire métier	108
Les éléments d'un dictionnaire métier	108
<i>Accéder aux éléments d'un dictionnaire métier</i>	109
<i>Importer des données métier</i>	109
Dictionnaire métier de travail	110
Créer un dictionnaire métier	110
Initialiser un dictionnaire métier à partir de données logiques ou physiques	110
<i>Initialiser un dictionnaire métier à partir de données logiques</i>	110
<i>Initialiser un dictionnaire métier à partir de données physiques</i>	111

<i>Initialiser un dictionnaire métier à partir de métadonnées physiques</i>	111
<i>Initialiser un dictionnaire métier lors de la création de données logiques ou physiques.</i>	112
<i>Afficher le graphe de réalisation</i>	112
Carte des domaines de concepts	115
Créer une carte des domaines de concepts	115
Les composants d'une carte des domaines de concepts	115
Exemple de domaines de concepts	116
Rapports disponibles sur une carte des domaines de concepts	117
Domaine de concepts	118
Créer un domaine de concepts	118
Créer le diagramme de structure d'un domaine de concepts	118
Construire un diagramme de concepts	118
<i>Créer un diagramme de concepts d'un domaine de concept</i>	119
<i>Les composants d'un diagramme de concepts</i>	119
<i>Synthèse des liens entre les objets</i>	121
<i>Aspect graphique des objets du diagramme</i>	125
Définir les composants d'un domaine de concepts	126
<i>Les composants d'un domaine de concepts.</i>	126
<i>Définir le mode d'accès aux composants (CRUD)</i>	127
Concept.	128
Accéder à la liste des concepts	128
Créer un concept	128
Concepts et termes	128
<i>Relier un terme existant à un concept</i>	128
<i>Créer des termes en plusieurs langues à partir d'un concept.</i>	129
<i>Créer des synonymes en plusieurs langues.</i>	129
Renommer un concept	129
Les propriétés d'un concept	130
<i>Vue d'ensemble.</i>	130
<i>Caractéristiques.</i>	130
<i>Composants</i>	131
<i>Super types</i>	131
<i>Réalisations</i>	132
<i>Réglementations</i>	132
<i>Qualité des données.</i>	132
<i>Rapports.</i>	132
<i>Workflows.</i>	133
Composants d'un concept.	134
Accéder aux composants d'un concept	134
Créer un composant de concept à partir d'un diagramme	135
Décrire le super composant d'un concept	136
Décrire un composant de concept calculé	136
Propriétés de concept.	137
Créer une propriété de concept	138
<i>Créer une propriété de concept</i>	138
<i>Relier une propriété de concept à un concept</i>	138
<i>Relier deux propriétés de concept</i>	138
Créer une propriété de concept calculée	138
Héritages d'un concept.	139
Accéder aux héritages d'un concept	139
Définir l'héritage d'un concept à partir d'un diagramme de concept	139

Définir l'héritage d'un composant de concept	140
Créer une substitution d'un composant de concept	140
Le diagramme de structure de concept.	141
Individus	143
Accéder aux individus d'un concept	143
Créer un individu	143
Les propriétés d'un individu	143
Créer une classification d'individu	144
Créer un composant d'entité de dictionnaire	144
Le diagramme de structure d'individu	145
Etats d'un concept ou d'un individu	146
Décrire les concepts d'état	146
<i>Accéder à la liste des concepts d'état</i>	<i>147</i>
<i>Créer un concept d'état à partir d'un dictionnaire métier</i>	<i>147</i>
<i>Les propriétés d'un concept d'état</i>	<i>148</i>
Décrire les concepts événement	149
<i>Accéder à la liste des concepts événement</i>	<i>149</i>
<i>Créer un concept événement à partir d'un dictionnaire métier</i>	<i>149</i>
<i>Les propriétés d'un concept événement.</i>	<i>150</i>
<i>Relier un concept événement à son concept</i>	<i>150</i>
Le diagramme de structure de concept d'état	151
Décrire les états et les événements d'un individu	152
<i>Accéder à la liste des états et des événements d'individu</i>	<i>154</i>
<i>Créer un état d'individu à partir d'un domaine de concepts</i>	<i>154</i>
<i>Les propriétés d'un état d'individu</i>	<i>155</i>
<i>Créer un événement d'individu à partir d'un domaine de concepts</i>	<i>155</i>
<i>Relier un événement d'individu à un individu</i>	<i>155</i>
Le diagramme de structure de cycle de vie de concept	155
<i>Créer un cycle de vie de concept</i>	<i>157</i>
<i>Créer un diagramme de structure de cycle de vie de concept.</i>	<i>157</i>
<i>Ajouter un événement de cycle de vie de concept</i>	<i>157</i>
<i>Créer une Transition de cycle de vie de concept</i>	<i>158</i>
Utiliser les périodes.	158
Concept type	160
Accéder à la liste des concepts type	160
Créer un nouveau concept type	160
Les propriétés d'un concept type	161
Décrire les composants d'un concept type	161
<i>Accéder aux composants de concept type</i>	<i>162</i>
<i>Créer un composant de concept type à partir d'un domaine de concepts.</i>	<i>162</i>
Décrire les variations d'un concept type.	163
<i>Accéder aux variations d'un concept type</i>	<i>163</i>
<i>Créer une variation d'un concept type à partir d'un domaine de concepts</i>	<i>163</i>
Le diagramme de structure de concept type.	163
Vue de concept	165
Créer une vue de concept	165
Définir le contenu de la vue de concept	166
<i>Affichage des objets de la vue</i>	<i>166</i>
<i>Ajouter un objet source à la vue de concept</i>	<i>166</i>
<i>Ajouter un composant à la vue de concept</i>	<i>167</i>
Le rapport de vue	167

Règle de calcul sur un concept. 169
Associer une règle de calcul à une information métier 170

Règle de calcul sur une propriété de concept 170

Créer la règle de calcul. 171
Définir les objets en entrée et en sortie de la règle 171

Règle de calcul sur un concept 172

Relier les concepts métier à l'architecture logique et physique. 173
La réalisation de concept 174

Définir l'objet qui réalise un concept 174

Grapher la réalisation du concept. 175

Définir le concept réalisé par une classe 175

Utiliser les matrices de réalisation 177

Les différents niveaux de réalisation 177

Créer une matrice de réalisation 178

ARCHITECTURE ET OUTILS

Modéliser les dictionnaires de données 181
Options de modélisation des données logiques 182

Les formalismes 182

Les notations 182

Vue d'ensemble des données logiques. 183

Dictionnaire de données 183

Carte des domaines de données 183

Domaine de données logiques 183

Vue de données logiques 183

Modèle de données 184

Exemple 184
Dictionnaire de données. 185

Éléments d'un dictionnaire de données 185

Accéder aux éléments d'un dictionnaire de données. 186
Importer des données logiques 186
Carte des domaines de données. 187

Créer une carte des domaines de données 187

Les composants d'une carte des domaines de données 187

Domaines de données et domaines de données logiques 188

Créer un domaine de données 188

Le diagramme de domaine de données 188

<i>Exemple de diagramme</i>	189
<i>Créer le diagramme d'un domaine de données</i>	189
<i>Ajouter un objet au diagramme</i>	189
Ajouter un composant au domaine de données	190
<i>Définir le mode d'accès aux composants (CRUD)</i>	190
Vues de données logiques	192
Créer une vue de données logiques	192
<i>Créer une vue de données (à partir de la liste des vues)</i>	193
<i>Créer une vue de données à partir d'un objet</i>	193
Affichage des objets sources dans la vue de données	194
Définir les composants de la vue de données	195
<i>Composant embarqué</i>	195
<i>Composant référencé</i>	195
<i>Utiliser une vue dans une autre vue</i>	196
Diagramme de classes	197
Créer un paquetage	197
Créer un diagramme de classes	198
Types de données	199
Paquetage de types de données	199
Créer un nouveau paquetage de types de données	200
<i>Créer un type de données</i>	201
Référencer un paquetage de types de données	201
Affecter des types aux attributs	201
Modèle de données	203
Synthèse des concepts	203
<i>Modèle de données</i>	203
<i>Diagramme de données</i>	203
Construire un modèle de données	204
<i>Condition préalable</i>	204
<i>Créer le modèle de données</i>	204
<i>Créer un diagramme de données</i>	204
Les entités	205
<i>Créer une entité</i>	206
Les attributs	206
<i>Créer un attribut</i>	207
<i>Attributs hérités</i>	207
Les associations	207
<i>Créer une association</i>	209
<i>Définir le rôle des associations</i>	210
<i>Multiplicités</i>	211
<i>Agrégation</i>	213
<i>Composition</i>	213
Associations réflexives	214
Association "plus que binaire"	215
Les contraintes	216
Règles de normalisation	217
<i>Première forme normale</i>	217
<i>Deuxième forme normale</i>	218
<i>Troisième forme normale</i>	218
Généralisations	219
<i>Qu'est-ce qu'une généralisation ?</i>	219
<i>Cas de plusieurs sous-entités</i>	221

<i>Héritage multiple</i>	223
<i>Créer une généralisation</i>	223
<i>Discriminant</i>	223
Identifiant d'une entité	224
<i>Identification par un attribut</i>	225
Correspondance des modèles de données	225
<i>Objectifs fonctionnels</i>	225
<i>Lancer l'éditeur de correspondances</i>	226
<i>Créer une correspondance</i>	227
<i>Supprimer une correspondance</i>	227
<i>Détails des correspondances</i>	227
<i>Exemple de correspondance entre modèles de données</i>	228
La notation IDEF1X	230
A propos de la modélisation des données avec IDEF1X	230
Synthèse des concepts	231
Construire un modèle de données (IDEF1X)	231
<i>Condition préalable</i>	231
<i>Créer le modèle de données</i>	231
<i>Diagramme de données (IDEF1X)</i>	232
Entités (IDEF1X)	232
<i>Créer une entité</i>	233
<i>Attributs</i>	233
Associations (IDEF1X)	234
<i>Relation identifiante obligatoire</i>	235
<i>Relation non-identifiante obligatoire</i>	236
<i>Relation optionnelle non-identifiante</i>	237
<i>Relation non-spécifique</i>	237
<i>Entité associative</i>	238
<i>Définir les rôles des associations</i>	239
<i>Multiplicités</i>	240
Relations de catégorisation (généralisations) - (IDEF1X)	241
<i>Qu'est-ce qu'une Catégorisation (Généralisation) ?</i>	242
<i>Créer une catégorisation</i>	242
<i>Catégories multiples</i>	243
<i>Multiples groupes de catégories</i>	244
<i>Catégorisation complète</i>	244
<i>Discriminant</i>	245
La Notation I.E.	246
A propos de la modélisation des données avec I.E.	246
Synthèse des concepts	247
Créer un modèle de données (I.E)	247
<i>Condition préalable</i>	247
<i>Créer le modèle de données</i>	248
<i>Diagramme de données (I.E)</i>	248
Entités (I.E)	248
<i>Créer une entité</i>	249
<i>Attributs</i>	249
Associations (I.E)	250
<i>Présentation</i>	250
<i>Associations et multiplicités</i>	251
Sous-types (I.E)	253
<i>Qu'est-ce qu'un sous-type ?</i>	253

<i>Cas de plusieurs sous-types</i>	254
<i>Intérêt des sous-types</i>	255
<i>Héritage multiple</i>	256
<i>Créer un sous-type</i>	256
La Notation Merise	257
A propos de la modélisation des données	257
Synthèse des concepts	257
Créer un modèle de données (Merise)	257
<i>Condition préalable</i>	258
<i>Créer le modèle de données</i>	258
<i>Diagramme de données (Merise)</i>	258
Les entités (Merise)	258
<i>Créer une entité</i>	259
Les associations (Merise)	260
<i>Exemples d'associations</i>	260
<i>Relation réflexive</i>	261
<i>Relation plus que binaire</i>	261
<i>Les participations ou cardinalités</i>	262
<i>Créer une association (relation)</i>	263
Les attributs (informations) - Merise	264
<i>Les propriétés</i>	264
<i>L'identifiant</i>	265
<i>Créer des attributs</i>	265
Les règles de normalisation (Merise)	266
<i>Première forme normale</i>	267
<i>Deuxième forme normale</i>	267
<i>Troisième forme normale</i>	268
Compléter la spécification du modèle de données (Merise)	268
<i>Ordonner les attributs</i>	268
<i>Description des attributs</i>	269
<i>Participations ou Cardinalités</i>	269
Sous-typage (Merise)	271
<i>Qu'est-ce qu'un sous-type</i>	272
Modéliser les bases de données	275
Formalisme logique et synchronisation	276
Base de données	277
Créer une base de données	277
Propriétés d'une base de données	277
Associer un paquetage à une base de données	278
Importer une version de SGBD	278
Carte de domaines relationnels et domaines relationnels	279
Carte de domaines relationnels	279
<i>Créer une carte de domaines relationnels</i>	279
<i>Les composants d'une carte de domaines relationnels</i>	279
Domaine relationnel	280
<i>Créer un Domaine relationnel</i>	280
<i>Diagramme d'un domaine relationnel</i>	280

Diagramme relationnel	282
Construire le diagramme relationnel	282
<i>Créer des objets dans le diagramme</i>	283
<i>Paramétrer l'affichage des diagrammes relationnels</i>	283
Composants d'une base de données	285
Les tables d'une base de données	285
<i>Créer une table</i>	285
<i>Supprimer une table</i>	285
Les colonnes d'une table	286
<i>Visualiser les colonnes</i>	286
<i>Créer une colonne</i>	286
<i>Supprimer une colonne</i>	287
Modifier les clés et les index	287
Créer une clé	288
<i>Clé primaire</i>	288
<i>Clé étrangère</i>	289
Créer un index	289
Ajouter une colonne à une clé ou un index	290
Clés primaires et étrangères	291
Spécifier les clés primaires	291
Spécifier les clés étrangères	292
Colonne clé primaire de deux tables	293
Colonne clé primaire de trois tables	293
Types des données et Datatypes des colonnes	294
Types de données des attributs	294
Déduire les datatypes des colonnes à partir des types des attributs	294
<i>Types pivots</i>	294
<i>Relier un type de données à un type pivot</i>	296
<i>Relier un type de données à un type pivot dans la notation UML</i>	297
Correspondances entre types pivots et datatypes	299
<i>Exemple de correspondances entre les types pivots et les datatypes pour Oracle 8</i>	299
Créer de nouveaux datatypes	301
<i>Exemple pour Oracle 10</i>	302
<i>Exemple pour SQL Server 7</i>	305
Règles de modélisation d'une base de données	307
Catégories de données	309
Définir des catégories de données	310
Accéder aux catégories de données	310
Créer une catégorie de données	310
Associer les catégories aux données	311
Définir une catégorie sur une donnée métier	311
Associer une catégorie de données à une réglementation	311

Responsabilité des données.	313
Usage des responsabilités sur les données.	314
Rôles applicables aux données	314
Accéder à ses notifications	315
Définir les responsabilités sur les données.	316
Visualiser les responsables d'un objet	316
Assignation automatique d'un objet	316
Assignation explicite d'un objet	316
Consulter les rapports de gouvernance des données	317

Utilisation des données par le SI	319
Définir les données utilisées par les applications.	320
Créer des applications avec HOPEX Data Governance	320
Relier des données à une application	320
Créer un dépôt de données sur une application	321
<i>Créer une donnée à partir d'un dépôt de données</i>	<i>322</i>
<i>Définir le mode d'accès aux données</i>	<i>322</i>
Définir l'application comme source de vérité	323
<i>Définir une application Golden Copy</i>	<i>324</i>
Impact des données utilisées	324
Voir dans quelles applications est utilisée une donnée	325
Définir les données utilisées par les processus	326
Faire l'inventaire des processus	326
<i>Types de processus</i>	<i>326</i>
<i>Créer un processus.</i>	<i>327</i>
Relier des données à un processus	327
<i>Créer un dépôt de données</i>	<i>327</i>
<i>Définir le mode d'accès aux données</i>	<i>327</i>
Relier des données à un contenu	328
Définir les données utilisées par les capacités métier	329
Créer des capacités métier avec HOPEX Data Governance	329
Relier des données à une capacité métier	329

Synchroniser les modèles logiques et physiques.	331
Options d'affichage de la synchronisation	332
Règles de synchronisation "logique vers physique".	333
Synchronisation logique > physique : les Entités (ou Classes)	333
<i>Règle générale.</i>	<i>333</i>
<i>Sous-entité</i>	<i>333</i>
<i>Entité abstraite</i>	<i>334</i>
<i>Entité réalisée</i>	<i>334</i>
Synchronisation logique > physique : les Associations.	334

Association contrainte (multiplicités : 0,1 ou 1,1)	334
Association contrainte (multiplicités : 0,1 et 0,1)	336
Verrous croisés (dead locks)	336
Association non-contrainte	338
Classe associative	339
Synchronisation logique > physique : les Parties (Formalisme UML)	339
Exemple 1 : Non renseigné / *	340
Exemple 2 : Agrégation / *	340
Exemple 3 : Composition / 0..1	340
Du modèle logique au modèle physique.	341
Lancer la synchronisation	341
Etape 1 : sélectionner les objets sources à synchroniser.	341
Etape 2 : options de synchronisation	342
Etape 3 : protéger des objets	343
Etape 4 : valider les résultats	343
Cas d'emploi des options	344
Prendre en compte les optimisations	344
Prendre en compte les suppressions	344
Combinaisons des options possibles	345
Protéger des objets	346
Mode Figé	346
Mode Réalisé	346
Résultats de la synchronisation : les correspondances	347
Caractéristiques de la correspondance	347
Synchronisation réduite (mode logique vers physique).	349
Objets sources de la synchronisation réduite	349
Lancement sur un modèle de données	349
Lancement sur une entité du modèle de données	350
Lancement sur une entité hors contexte	350
Les stratégies de synchronisation réduite	350
Impact de l'objet synchronisé sur les autres objets	350
Impact des autres objets sur l'objet synchronisé	351
Tous les impacts	353
Lancer la synchronisation réduite	353
Options de la synchronisation réduite	354
Relancer la synchronisation après modifications.	355
Synchronisation après modification du diagramme de données	355
Création d'entités, d'associations et d'attributs dans le diagramme de données.	355
Suppression d'entités, d'associations, d'attributs dans le diagramme de données	355
Modification des caractéristiques d'un attribut.	356
Modification du nom d'un attribut, d'une entité, ou d'une association.	356
Modification de la multiplicité maximum d'une association	356
Modification des liens d'une association	356
Synchronisation après des modifications du diagramme physique	357
Suppression d'une table ou d'une colonne.	357
Créations d'objets	357
Modifications de caractéristiques d'objets créés par la synchronisation.	357
Modifications d'ordre	357
Du modèle physique au modèle logique.	358
Règles de synchronisation "Physique vers Logique"	358
Lancer la synchronisation	361
Etape 1 : sélectionner les objets à synchroniser	361

<i>Etape 2 : options de synchronisation</i>	361
<i>Etape 3 : protéger des objets.</i>	362
<i>Etape 4 : valider les résultats.</i>	362
<i>Synchronisation réduite.</i>	362
Résultat de la synchronisation "Physique vers Logique"	362
<i>Modèle de données détenteur</i>	362
<i>Diagrammes de données.</i>	363
<i>Les correspondances.</i>	363
Paramétrer la synchronisation	364
Préparer la synchronisation	364
Options de création	364
<i>Sur une base de données</i>	364
<i>Sur le SGBD.</i>	366
Paramétrer la génération des noms.	366
<i>Règles de construction des noms</i>	366
<i>Modifier une règle de construction</i>	368
<i>Saisie du masque SQL.</i>	368
<i>Paramétrer le nom des colonnes de PK (identifiant implicite)</i>	370
Synchronisation des diagrammes	373
Cas de mise à jour des diagrammes lors d'une synchronisation	373
<i>Après modification du diagramme source</i>	373
<i>Après modification du diagramme cible.</i>	373
<i>Après modification des deux diagrammes</i>	374
<i>Pas de modification détectée</i>	374
<i>Cas particulier : une entité en correspondance avec deux tables</i>	374
Correspondance des modèles	375
L'éditeur de base de données	376
Lancer l'éditeur sur une base de données	376
<i>Créer un arbre de correspondance logique/physique.</i>	376
Créer une correspondance	376
Supprimer une correspondance	378
Détails des correspondances	379
Propriétés d'une correspondance	379
Rapport des correspondances	380
Etat des objets	381
<i>Enregistrer l'affichage des indicateurs de l'éditeur</i>	381
Source de la correspondance	381
Dessin de la correspondance	383
Dénormaliser les modèles logiques et physiques	385
Principes de la dénormalisation	386
Dénormalisation : cohérence des modèles	386
<i>Report des correspondances</i>	386

<i>Supprimer les objets sources</i>	386
Synchronisation et dénormalisation	386
<i>Combinaison des options de dénormalisation et synchronisation</i>	387
Dénormalisation : cas d'emploi	387
Dénormalisation logique	390
Lancer une dénormalisation logique	390
<i>Exemple de dénormalisation logique</i>	390
Liste des assistants de dénormalisation logique	392
<i>Transformer une association en entité</i>	392
<i>Transformer une entité en association</i>	392
<i>Transformer une généralisation en association</i>	393
<i>Transformer une association en généralisation</i>	394
<i>Partition verticale d'une entité</i>	394
<i>Partition horizontale d'une entité</i>	395
<i>Fusion d'entités</i>	396
<i>Fusion d'entités ascendantes</i>	396
<i>Fusion d'entités descendantes</i>	397
<i>Copier/Coller d'attributs</i>	397
Dénormalisation physique	398
Lancer une dénormalisation physique	398
<i>Exemple de dénormalisation physique</i>	398
Liste des assistants de dénormalisation physique	399
<i>Partition verticale d'une table</i>	399
<i>Partition horizontale d'une table</i>	400
<i>Fusion de tables</i>	401
<i>Transformation d'une clé étrangère en table</i>	402
<i>Transformation d'une table en clé étrangère</i>	403
<i>Copier/Coller des colonnes</i>	403
Générer des scripts SQL	405
Lancer la génération SQL	406
Condition préalable	406
Objets de la génération SQL	407
Lancer l'assistant de génération	407
Génération incrémentale	409
Objets de la génération incrémentale	409
Lancer la génération incrémentale	409
<i>Options de la génération</i>	409
<i>Lancer l'assistant de génération</i>	410
Paramétrer la génération SQL	412
Paramétrer la version de SGBD	412
<i>SGBD et versions supportées</i>	412
<i>Modifier les propriétés de la version de SGBD</i>	412
Paramétrer la génération d'une base de données	413
Préfixer le nom des objets	415
<i>Héritage</i>	415
<i>SGBD concernés</i>	415

Syntaxe supportée	416
Instruction CREATE TABLE	416
<i>Gestion de NOT NULL</i>	416
<i>Clause PRIMARY KEY</i>	417
<i>Clause FOREIGN KEY</i>	417
<i>Clause UNIQUE</i>	419
Instruction CREATE INDEX (Oracle, Sybase, SQL Server)	419
<i>Définition d'un index</i>	419
<i>Traitement et génération des ordres SQL</i>	420
<i>Clause CREATE VIEW</i>	420
Définir les vues d'une base de données	421
Créer les vues d'une base de données	421
<i>Ajouter une table ou une colonne de vue</i>	422
Définition SQL	422
<i>Jointures de vue</i>	422
<i>Mode utilisateur</i>	422
<i>Champs</i>	423
Définir un groupement de données	423
Définir les triggers pour une base de données	425
Créer un trigger	425
<i>Déclenchement du trigger</i>	425
<i>Références</i>	425
<i>Définition SQL</i>	426
Intégrité référentielle	426
Utiliser des procédures stockées	427
Ajouter des propriétés physiques aux objets d'une base de données	429
SGBD cibles	429
Créer des propriétés physiques	429
<i>Objets contenant des paramètres physiques</i>	430
<i>Créer une nouvelle clause</i>	431
<i>Relier une clause</i>	431
<i>Nommage des clauses</i>	431
<i>Exemple de personnalisation d'un modèle physique</i>	433
Générer le fichier SQL	435
Rétro-générer des tables	437
Lancer la rétro-génération	438
Reconnaissance des Datatypes par ODBC	439
Problèmes de reconnaissance des datatypes	441
Rétro-génération des propriétés physiques	442
Valeurs par défaut	442
Élimination des valeurs redondantes et transverses	442
Cas spécifiques	443
<i>Propriétés physiques des tablespaces</i>	443
<i>Rétro-génération des clusters</i>	443
Extraire la description de schémas de bases de données à partir de sources de données	

444	
Configuration des sources de données requise	444
Télécharger HOPEX Data Source Extractor	444
Lancer l'extraction de données	444
Fichier de compte-rendu de l'extraction	450
Fichier résultat de l'extraction	451
Personnaliser l'extraction ODBC	451
<i>Utilisation du fichier Odwdbex.ini et des requêtes personnalisées</i>	452
<i>Utilisation des API standard ODBC</i>	452
Format des clauses SELECT	453
<i>Clés primaires</i>	453
<i>Clés étrangères</i>	454
<i>Index</i>	455
<i>Colonnes</i>	456

Tableaux de correspondances entre types pivots et datatypes	459

CONFORMITÉ

Réglementations.	493

Faire l'inventaire des textes de référence (réglementations externes).	494

Importer le module des textes de référence	494
Structure d'un texte de référence	494
Créer un texte de référence	495
<i>Définir les objets de votre organisation contraints par un texte de référence.</i>	495
<i>Voir les objets contraints par un texte de référence</i>	496
<i>Voir les catégories d'objet contraintes par un texte de référence</i>	496
Ajouter une section au texte de référence	496
<i>Définir les objets contraints par une section de texte de référence.</i>	497
Ajouter un article à une section	497
Ajouter une définition à un article	497
Faire l'inventaire des obligations	498
<i>Accéder à la liste des obligations</i>	498
<i>Obligations impactées et contributrices</i>	499
<i>Mise en application des obligations</i>	499
<i>Associer une obligation à un article de texte de référence.</i>	500
<i>Voir les articles associés à une obligation</i>	500

Faire l'inventaire des cadres de politiques d'entreprise (réglementations internes) .	501

Créer un cadre de politique d'entreprise	501
Ajouter une section au cadre de politique d'entreprise	501
Ajouter une politique d'entreprise	502
Analyser les informations contraintes par une réglementation	502

Importer des documents d'UCF.	504
Utiliser l'import UCF	504
<i>Conditions préalables à l'import UCF.</i>	<i>504</i>
<i>Paramétrer l'import UCF</i>	<i>505</i>
<i>Importer des données à partir du Common Controls Hub</i>	<i>505</i>
Spécifier le contenu réglementaire applicable	506
<i>Pertinence du contenu réglementaire</i>	<i>506</i>
<i>Procéder à la revue des textes de référence après import</i>	<i>506</i>
<i>Sélectionner le contenu réglementaire pertinent pour votre organisation.</i>	<i>507</i>
Assurer la conformité des données : créer des règles métier.	508
Définir une règle d'entreprise	508
Rapport de règle	509

Qualité des données. 511

Définir une politique de qualité des données	512
Créer une politique de qualité	512
Définir des indicateurs de qualité	513
<i>Créer un indicateur de qualité</i>	<i>513</i>
<i>Définir la dernière valeur d'un indicateur.</i>	<i>513</i>
<i>Rapport de l'indicateur</i>	<i>513</i>
Importer des politiques et indicateurs de qualité des données	514
<i>Télécharger le modèle d'import</i>	<i>514</i>
<i>Importer les objets dans HOPEX.</i>	<i>515</i>
Définir des contrôles de qualité des données	515
Identifier les problèmes de qualité des données	516
Remédier aux problèmes de qualité rencontrés : plans d'action	516
<i>Propriétés d'un plan d'action dans HOPEX Data Governance</i>	<i>518</i>
Évaluer la qualité des données	520
Dimensions de qualité	520
Objets de l'évaluation	522
Évaluation directe	523
Évaluation par campagne	523
<i>Conditions préalables à l'évaluation des données</i>	<i>523</i>
<i>Créer une campagne d'évaluation.</i>	<i>523</i>
Rapports de qualité des données	524

Rapports et outils d'analyse des données 525

Rapports d'analyse	526
Accéder aux rapports d'HOPEX Data Governance	526
Rapports de description	526
<i>Rapport de vue</i>	<i>526</i>
<i>Rapport de glossaire.</i>	<i>526</i>
<i>Carte de domaines de données</i>	<i>527</i>
<i>Graphe de réalisation</i>	<i>529</i>
<i>Dépendances entre domaines de données</i>	<i>529</i>

<i>Rapport de catalogue de données</i>	530
<i>Dendrogramme des catégories de données</i>	530
Rapports nuages de mot	530
<i>Nombre d'informations des domaines d'information</i>	530
<i>Etendue de la description de l'information</i>	531
<i>Utilisation des données dans les domaines de données</i>	531
Rapports d'utilisation des données	531
<i>Utilisation des informations détenues par un conteneur</i>	531
<i>Utilisation des informations d'un domaine</i>	532
<i>Utilisation des informations d'une carte d'informations</i>	532
<i>Utilisation de l'information</i>	532
<i>Utilisation des informations des domaines d'un conteneur</i>	533
Rapports réglementaires	533
<i>Rapport des cadres réglementaires</i>	533
<i>Conformité réglementaire et transformation des données</i>	534
<i>Rapport de description des règles</i>	534
Rapports de qualité des données	535
<i>Evolution de la qualité des données</i>	535
<i>Qualité des données par dimension</i>	535
Report DataSets	536
Créer un Report DataSet dans HOPEX Data Governance	536
Exemple de Report Dataset	536
<i>Définition des termes</i>	536
<i>Matrice Dictionnaire métier x Concept vérifiant changement de nom</i>	537
Tableau de bord des données	539
Tableaux de bord	539
Tableaux de bord BoldBI	539
<i>Exemple de tableau de bord construit avec Bold BI</i>	540
 Workflow de validation des données	 541
<i>Etapes du workflow de validation</i>	541
<i>Générer un rapport de workflow</i>	542
 Import et export de données	 543
Importer des données métier à partir d'un fichier Excel	544
Télécharger le modèle de fichier Excel	544
Contenu du modèle Excel	544
<i>Feuille Concepts</i>	544
<i>Feuille Properties</i>	546
Importer des données logiques à partir d'un fichier Excel	548
Télécharger le modèle de fichier Excel	548
Contenu du modèle Excel	548
<i>Feuille Data Dictionary</i>	548
<i>Feuille Data Type</i>	549
<i>Feuille Data Type Component</i>	550

<i>Feuille Class.</i>	550
<i>Feuille Attribute</i>	551
<i>Feuille Relationship.</i>	552
<i>Feuille Generalization</i>	552
Importer des évaluations de données	553
Exemple d'import	553
Contenu du modèle Excel	554
Télécharger le modèle Excel	554
Importer un fichier Excel d'évaluations de données	554

INTRODUCTION À HOPEX DATA GOVERNANCE



Hopex Data Governance vous permet d'améliorer la qualité des données qui circulent au sein de votre entreprise. Elle permet de construire l'architecture globale des données et de tracer l'utilisation de l'information à travers toutes les fonctions de votre organisation.

- ✓ [Périmètre de la solution Hopex Data Governance](#)
- ✓ [Le bureau Hopex Data Governance](#)
- ✓ [Les profils de HOPEX Data Governance](#)
- ✓ [Les rôles métier de Hopex Data Governance](#)

PÉRIMÈTRE DE LA SOLUTION HOPEX DATA GOVERNANCE

Hopex Data Governance s'inscrit dans une démarche de gouvernance des données. A ce titre, elle offre un ensemble de fonctionnalités qui couvre les dimensions suivantes :

- la découverte de données, à travers des connecteurs prêts à l'usage
- la définition d'un glossaire métier basé sur les termes et leurs définitions
- l'architecture des données
- la spécification de règles métier et la gestion de la conformité aux réglementations qui s'appliquent à l'entreprise
- la mise en place de lignage de données pour suivre le parcours de l'information
- l'évaluation de la qualité des données
- l'analyse des données, à travers des rapports fournis en standard

Découverte de données

A travers l'outil de découverte des données **HOPEX Data Discovery**, vous parcourez différentes sources de données et définissez les métadonnées à importer dans votre catalogue de données.

Voir [Découverte de données](#).

Glossaire métier

Hopex Data Governance permet de faire l'inventaire des termes de l'entreprise et de générer un glossaire métier à partir duquel vous pouvez consulter leur définition, ainsi que leur synonyme et composants.

Voir [Générer un glossaire](#).

Architecture des données

Trois niveaux de modélisation

La solution **Hopex Data Governance** couvre les trois niveaux de modélisation des données d'une organisation :

- Niveau métier (conceptuel): permet de définir les concepts de l'architecture métier et de générer des glossaires. Ces concepts peuvent être mis en œuvre par des objets du niveau logique et être décrits par

des modèles de données.

Voir [Introduction à la conception d'une ontologie métier](#).

- Niveau logique : s'adresse à la maîtrise d'ouvrage qui souhaite élaborer des modèles généraux orientés métier. Il s'agit ici de modéliser les données d'un domaine, d'une application ou encore d'un processus. Il représente ce que l'on veut faire, à quoi l'on veut arriver, indépendamment des questions techniques liées à la mise en oeuvre. La représentation des données peut se faire dans un modèle de données ou un diagramme de classes.

Voir : [Modéliser les dictionnaires de données](#).

- Niveau physique : consiste à définir les modèles destinés à persister dans un SGBD. Il s'agit de spécifications détaillées en vue de la réalisation du schéma physique de la base. Il est représenté par le diagramme relationnel.

Le niveau physique définit également la façon selon laquelle sont stockées les données et les méthodes pour y accéder. Il permet l'exploitation des données par les SGBD.

Voir [Modéliser les bases de données](#).

Catégorie des données

Vous pouvez classer les données du référentiel par catégorie. Un arbre dédié répertorie les différentes catégories et les données associées. Les données ainsi classées peuvent être exploitées dans la solution **Hopex Privacy Management** spécifique à la gestion des données sensibles et de la conformité au RGPD.

Voir [Catégories de données](#).

Workflow de conception

En tant que concepteur de données (responsable de l'actif informationnel) ou créateur (qui concerne tous les profils d'IA), vous pouvez lancer un workflow sur certains objets de l'architecture de données (tels qu'un lignage de données ou un domaine de données) afin de suivre leur conception, leur mise à jour et leur validation.

Des rapports de workflow vous permettent de voir le nombre d'objets qui se trouvent à chaque étape du workflow (nombre d'objets en cours de conception, en cours d'analyse, etc).

Voir [Workflow de validation des données](#).

Utilisation des données

Pour indiquer quelles informations métier sont manipulées au niveau de l'architecture du SI, **Hopex Data Governance** propose une fonctionnalité de "Réalisation" qui consiste à relier les informations métier aux objets du SI.

Voir [Relier les concepts métier à l'architecture logique et physique](#).

Vous pouvez également spécifier quels processus et applications utilisent quelles données, qu'il s'agisse de données métier, logiques ou physiques.

Voir [Utilisation des données par le SI](#).

Qualité et conformité des données

Afin de garantir des données fiables et complètes, **Hopex Data Governance** fournit les outils pour définir les responsabilités sur les données, leur parcours, les règles et standards auxquels l'entreprise doit se conformer, et évaluer dans quelle mesure les données répondent aux exigences de l'organisation et de ses parties prenantes.

Définition des responsabilités

Lors de la conception des données, des responsables sont définis. Ils sont notifiés des demandes de mise à jour ou de validation dans le cadre de workflow de conception ou d'évaluation sur les données en question.

Voir [Responsabilité des données](#).

Définition des règles et standards

Hopex Data Governance vous permet de faire l'inventaire des réglementations et des règles d'entreprise, et de définir précisément à quels articles ou sections doivent se conformer les différentes informations et entités d'une organisation. La solution fournit notamment les contenus de la réglementation bancaire BCBS 239 et de la réglementation Solvabilité II.

Voir [Réglementations](#).

Traçabilité des données (Lignage de données)

A travers le lignage de données vous pouvez représenter les différents processus de traitement de la donnée, afin de faciliter l'identification des erreurs et réduire les risques de non-conformité. Cela permet aux Data Steward et Data Owner de s'assurer de la qualité des données utilisées.

Voir [Lignages de données](#).

Evaluation des données

Hopex Data Governance permet d'assurer la qualité des données à travers des évaluations. La solution fournit pour cela un modèle d'évaluation qui évalue les données sur six dimensions : la complétude, l'unicité, la fraîcheur, l'exactitude, la cohérence et la validité.

L'évaluation porte sur les métadonnées. Lorsque des informations existent dans des outils tiers sur les instances de données, les experts tels que le scientifique de données ou le responsable de la qualité des données peuvent remonter ces informations sur les métadonnées dans **Hopex Data Governance**.

Par exemple, l'information "Account Manager" est utilisée dans différents outils de traitement de données. L'expert qui constate un manque dans certains enregistrements de

cette information, par exemple sur les coordonnées de la personne, pourra signaler ce défaut de complétude dans HOPEX au niveau de la métadonnée (la classe "Account Manager" par exemple) afin qu'un plan d'action puisse être créé pour y remédier (via des champs obligatoires par exemple).

L'évaluation peut se faire directement dans la fenêtre de propriétés des données ou via un questionnaire d'évaluation envoyé aux responsables des données en question.

Voir [Qualité des données](#).

Rapports d'analyse

Les rapports d'analyse sont des rapports dynamiques qui permettent d'analyser sous différents axes les données du référentiel : complétude des données, utilisation des données, responsabilités, etc. **Hopex Data Governance** fournit des rapports types par défaut qui vous permettent de vérifier la qualité, l'utilisation et la conformité de vos données.

Voir [Rapports et outils d'analyse des données](#).

Data Governance vs Information Architecture

Fonctionnalité	Data Governance	Information Architecture
Découverte de données , à l'aide de connecteurs prêts à l'emploi	X	
Gestion de catalogues de données pour les métadonnées découvertes, avec un moteur de recherche dédié	X	
Glossaire métier des termes et de leurs définitions, avec un moteur de recherche dédié	X	X
Architecture de données (couches métier, logique et physique)	X	X
Modélisation de bases de données	X	X
Génération de schéma de base de données	X	X
Gestion de la conformité des données , par le biais de réglementations et de règles métier	X	

Fonctionnalité	Data Governance	Information Architecture
Lignage de données (fonctionnel et technique)	X	
Politiques et évaluations de la qualité des données	X	X
Analyse des données via des rapports standard	X	X

SE CONNECTER À HOPEX DATA GOVERNANCE

Pour se connecter à **Hopex Data Governance**, voir Hopex Common Features, "Le bureau Hopex Web Front-End".

☛ Pour plus de détails sur l'utilisation de la plateforme Web des solutions Hopex, voir le guide **Hopex Common Features**.

Les menus et commandes disponibles dans **Hopex Data Governance** dépendent du profil avec lequel vous êtes connecté.

Les profils utilisateurs de Hopex Data Governance

Dans **Hopex Data Governance**, il existe, par défaut, des profils auxquels sont associés des droits et accès.

Profil	Description
Concepteur de données (Data Designer)	Il a pour responsabilité de modéliser les données (métier, logiques et physiques) ainsi que les domaines de données qui permettent d'utiliser ces données dans les cartographies de processus ou d'application.
Gestionnaire de données (Data Asset Manager)	Responsable de la gouvernance des données

Profil	Description
Administrateur fonctionnel de la gouvernance des données (Data Governance Functional Administrator)	Il a la charge de gérer toutes les tâches d'administration du produit. Il possède des droits sur tous les objets. - Il gère les utilisateurs et leur assignation aux profils. - Il prépare l'environnement de travail et crée les éléments nécessaires à la gestion de l'information. - Il peut intervenir sur : • les dictionnaires, • les domaines d'information, • les concepts, vues de concept • etc.
Contributeur des données (Data Contributor)	Le contributeur aux données intervient dans le workflow des données ; il participe à la définition et à la description des données.
Observateur de données (Data Viewer)	Ce profil peut être utilisé pour consulter des informations sur les données, telles que la définition des termes dans le glossaire métier, l'utilisation des données dans les domaines d'information métier, dans les applications, etc.

Les rôles métier de Hopex Data Governance

Dans **Hopex Data Governance**, les objets peuvent être assignés à des personnes avec les rôles suivants :

- **Directeur de données (DCO)** : le directeur de données est un gestionnaire de données, qui est responsable de l'ensemble de la stratégie des données de l'entreprise. C'est lui qui va optimiser la collecte des données, leur stockage, la fiabilité et la qualité de ces dernières et en optimiser la valorisation.
- **Propriétaire de données** : c'est l'autorité qui décide de l'accès aux données et de leur utilisation. Le propriétaire des données peut être le concepteur des données, un de ses utilisateurs ou une tierce partie. Les data stewards peuvent demander aux propriétaires de données de vérifier ou de compléter la valeur d'un champ, pour corriger par exemple un défaut de qualité de données.
- **Concepteur de données** : le concepteur de données est responsable de la définition, de la description et de la classification des données du système d'information. Il enrichit et maintient le dictionnaire et les modèles de données de référence. Il accompagne la définition des besoins de l'entreprise en matière de description et d'utilisation des

données. Il participe également au développement de lignages de données qui est une contribution à la fiabilité du traitement des données.

- **Data Engineer** : construit et maintient les outils et les infrastructures nécessaires à l'analyse des données par les data scientists. Il veille à créer des solutions capables de traiter de gros volumes de données tout en garantissant leur sécurité. Il représente le premier maillon de la chaîne informatique.
- **Scientifique de données** : il est en charge du rapprochement entre le concepteur des données (métier et logiques) et les responsables des processus qui utilisent ces données.
- **Responsable qualité des données** : doit s'assurer de la pertinence et de l'utilité des données de l'entreprise. Pour cela il doit mettre en place des procédures de contrôle des données.
- **Coordinateur de données** : le coordinateur de données (an anglais, Data Steward) est le chef de file d'un projet de gouvernance des données. Il joue un rôle clé dans sa réalisation. Il a la connaissance des données et de leurs métadonnées. Il se les approprie et travaille avec les équipes métier pour définir les objectifs du projet de gouvernance des données.
- **Responsable infrastructure** : responsable de panorama d'infrastructure d'entreprise et de centre de données.

Voir aussi [Responsabilité des données](#).

LE BUREAU HOPEX DATA GOVERNANCE

Page d'accueil de Hopex Data Governance

La page d'accueil de la solution **Hopex Data Governance** se compose des sections suivantes.

- L'en-tête présente quelques informations d'intérêt général.
 - ☛ *Celles-ci peuvent être définies dans le menu **Environnement** > **Domaine méthodologique** de l'administrateur.*
- **Nos objectifs** : indique les principaux thèmes stratégiques d'intérêt pour les utilisateurs de la solution.
- **Aide** : pointe vers la documentation utilisateur et la communauté d'utilisateurs
- la section **Mon périmètre** présente des indicateurs utiles sur le contenu du référentiel. Voir ci-dessous [Indicateurs du périmètre](#).
- La section **Accès rapide** présente des raccourcis utiles :
 - **Vus récemment** : derniers objets et diagrammes accédés par l'utilisateur.
 - **Favoris** : objets favoris de l'utilisateur et objets favoris partagés
 - **Actions** : accès rapide à la création d'éléments d'architecture
- **Mon rapport favori** : affiche le rapport défini par l'utilisateur ou prédéfini par l'administrateur, qui peut servir de point d'entrée dans le référentiel.

Indicateurs du périmètre

La section **Mon périmètre** présente des indicateurs utiles sur les éléments du parc applicatif. En cliquant sur l'indicateur vous accédez à l'ensemble des objets correspondants. Il existe trois groupes d'indicateurs :

- Gouvernance de données
- Conformité
- Inventaire



Gouvernance de données

Cette vignette recense les objets suivants :

- **Métadonnées sans propriétaire** : affiche les métadonnées sans conteneur détenteur.
*Le conteneur détenteur d'une métadonnée peut être un dépôt de données ou un paquetage. Il est visible dans la page **Caractéristiques** de ses propriétés.*
- **Métadonnées non reliées à des données conceptuelles** : indique le nombre de métadonnées non associées à un concept au niveau métier.
Vous pouvez initialiser un dictionnaire métier à partir de métadonnées. Cela consiste à créer dans un dictionnaire métier les concepts et les éléments d'information qui correspondent aux métadonnées physiques et aux champs de métadonnées, et à établir ainsi un lien de réalisation entre ces objets. Voir [Initialiser un dictionnaire métier à partir de métadonnées physiques](#).
- **Données conceptuelles sans définition**
*La définition d'un concept est visible dans la page **Caractéristiques** de ses propriétés.*
- **Données conceptuelles non reliées à un domaine de concepts**
Un domaine de concepts est un sous-ensemble d'éléments d'un dictionnaire métier qui permet de réduire le champ d'une étude. Il est décrit par un diagramme de concepts.
- **Domaines de données vides utilisés dans des Applications et Processus**
Pour définir quelles données sont utilisées par quels processus et applications, voir [Utilisation des données par le SI](#).

Conformité

Cette vignette recense les objets suivants :

- **Données critiques** : données classées dans la catégorie "Données critiques".
 Voir [Catégories de données](#).
- **Réglementations à respecter** : liste les contraintes réglementaires associées aux données.
 Voir [Réglementations](#).
- **Politiques de qualité des données**
 Les politiques de qualité des données sont des directives qui décrivent à quels critères doivent répondre les données pour être utiles et exploitables au sein d'une organisation.
Voir [Qualité des données](#).
- **Données conceptuelles avec Lignage de données**
 Voir [Lignages de données](#).
- **Données évaluées les 6 derniers mois**

Inventaire

La vignette **Inventaire** affiche le nombre d'objets suivants :

- Concepts
- Propriétés de concept
- Sources de données découvertes
- Métadonnées physiques
- Champs de métadonnées physiques

Afficher l'environnement de travail d'une entreprise

Un référentiel peut être partitionné en *Entreprises*.

Une entreprise est un projet d'entreprise, qui établit des buts à atteindre, ainsi que les plans d'action stratégiques visant à l'accomplissement de ces buts. Elle est constituée de phases de transformation dans lesquelles sont définis les capacités ou livrables à atteindre.

Lorsqu'elles sont associées à un environnement de travail, les entreprises constituent des points d'entrée dans le bureau Data Management ; l'environnement offre un accès privilégié aux objets détenus et utilisés par l'entreprise en question.

Créer une entreprise et son environnement de travail

La création des entreprises et des environnements de travail est réalisée par l'administrateur fonctionnel.

Pour créer une entreprise dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Environnement** > **Navigation standard**.

2. Passez le curseur de la souris sur le dossier **Entreprises** et cliquez sur le bouton **+** (Nouveau > Entreprise).
3. Dans l'assistant de création, indiquez le nom de l'entreprise.
4. Pour créer en même temps l'environnement de l'entreprise, sélectionnez le type d'environnement "Information Architecture".
5. Cliquez sur **OK**.

Si aucun environnement n'a été créé en même temps que l'entreprise, vous pouvez le créer ultérieurement.

Pour assigner un environnement de travail à une entreprise existante dans **Hopex Data Governance** :

1. Sélectionnez le projet ou l'entreprise en question pour afficher ses propriétés.
 Cliquez sur le bouton *Propriétés* de la zone d'édition si les propriétés ne s'affichent pas.
2. Sélectionnez la page **Assignation de l'environnement de travail**.
 La page peut être masquée par défaut.
3. Cliquez sur **Nouveau**.
4. Renommez si besoin le nouvel environnement et sélectionnez le type "Information Architecture".
5. Cliquez sur **OK**.
 Vous pouvez aussi créer l'environnement de travail d'une entreprise lors de sa création.

Voir aussi : [Entreprises et bibliothèques](#).



Catalogues de données



DESCRIPTION D'UN CATALOGUE DE DONNÉES



Un catalogue de données est un outil de gestion des métadonnées qui centralise les métadonnées physiques en un seul endroit. Il facilite l'accès à un périmètre de métadonnées et permet à l'organisation de gérer leur définition, leur utilisation et leur qualité.

Vous avez la possibilité de relier les métadonnées du catalogue aux objets métier de l'organisation, de référencer les réglementations qui s'appliquent aux métadonnées du catalogue et auxquelles sont assujetties les organisations impliquées. Vous pouvez également créer un lignage de données à partir du catalogue pour mieux suivre l'utilisation des métadonnées, leur qualité et leur conformité aux règles auxquelles elles sont soumises.

Pour définir le périmètre des métadonnées, **Hopex Data Governance** offre un outil de découverte de données. Il vous permet de vous connecter à des sources de données multiples (SQL, No SQL, fichiers, stockage, etc.) et d'extraire un périmètre de métadonnées. Voir [Découverte de données](#).

- ✓ [Contenu d'un catalogue de données](#)
- ✓ [Infrastructure des sources de données](#)
- ✓ [Recherche dans le catalogue de données](#)
- ✓ [Créer un catalogue de données](#)
- ✓ [Associer des mots-clés à des métadonnées](#)
- ✓ [Définir des catégories de métadonnées](#)
- ✓ [Rapports d'un catalogue de données](#)
- ✓ [Relier les métadonnées aux informations métier](#)
- ✓ [Collecter des données métier](#)
- ✓ [Qualité des données du catalogue](#)
- ✓ [Instantané d'un catalogue de données](#)

CONTENU D'UN CATALOGUE DE DONNÉES

Un catalogue de données est l'outil de gestion des métadonnées dans HOPEX. Il référence des sources de données qui contiennent les métadonnées physiques.

Les catalogues de données sont disponibles avec le profil "Gestionnaire de données".

Pour accéder aux catalogues de données :

- 1 Dans la barre de navigation cliquez sur **Catalogues** puis sur **Hiérarchie**.

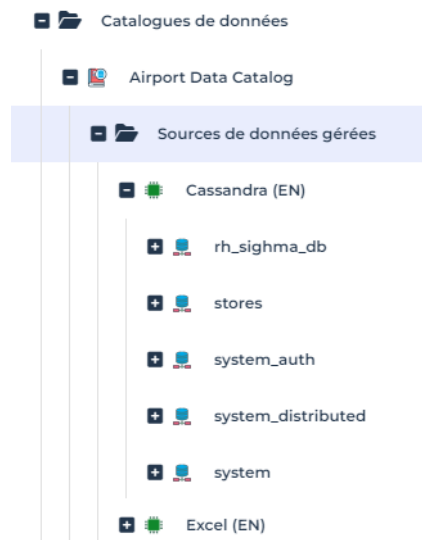
Dans un catalogue de données vous pouvez définir :

- les organisations impliquées, qui utilisent ou consomment les données du catalogue.
- les textes de référence et politiques d'entreprise qui s'appliquent aux métadonnées du catalogue, et auxquelles sont assujetties les organisations impliquées. Voir [Règlementations associées au catalogue](#).
- une carte de domaines de concepts
- des lignages de données

Sources de données

Les sources de données d'un catalogue sont des dépôts de données déployés.

Les dépôts de données sont regroupés par technologie au sein d'un catalogue.



Pour plus de détails sur l'infrastructure des sources de données, voir [Infrastructure des sources de données](#).

Un dépôt de données déployé regroupe un ensemble de métadonnées. Il peut contenir :

- Des schémas de métadonnées physiques, qui regroupent des métadonnées physiques
- Des métadonnées physiques (ex : des tables).

Métadonnée physique

Une métadonnée physique représente la structure d'une collection de données connexes détenues dans une source de données.

La structure de données représentée peut être relationnelle (une table) ou NoSQL (fichier plat, triplestore, etc).

Champ de métadonnée physique

Un champ de métadonnées est une propriété physique des données qui constitue la spécification du contenu d'une métadonnée physique. Il correspond à une colonne dans une table ou une colonne de feuille Excel.

Il peut être des types suivants :

- champ de métadonnée simple
- champ de métadonnée structuré : il s'agit d'un champ de donnée composé d'autres champs de donnée (exemple : adresse postale)
- type élémentaire d'un champ : type qui identifie le champ (exemple : "Adresse", "Code", "Date")

Pour relier un champ de métadonnée (simple ou structuré) à une métadonnée physique, vous allez d'abord indiquer si c'est un champ qui identifie la métadonnée, un champ référencé ou un champ qui n'est ni identifiant ni référencé.

Pour relier un champ de données à une métadonnée physique :

1. Ouvrez les propriétés de la métadonnée physique.
2. Dans la page **Champs de métadonnée physique**, cliquez sur **Nouveau**.
3. Dans la fenêtre qui apparaît, sélectionnez le type de champ voulu, qui indique si le champ à relier est identifiant ou référencé :
 - "Identifiant" : champ de métadonnée qui permet de distinguer l'objet. Ce peut être un champ de métadonnée physique simple ou structuré.
 - "Référence" : champ de donnée qui n'est pas détenu par la donnée physique. Ce peut être un champ de métadonnée physique simple ou structuré.
 - "Standard" : champ de métadonnée qui n'est ni identifiant ni référencé. Ce peut être un champ de métadonnée physique simple ou structuré.

Réglementations associées au catalogue

Vous pouvez associer au catalogue des textes de référence ou des cadres de politique d'entreprise.

Pour relier une réglementation au catalogue :

1. Sous le catalogue de données, passez la souris sur le dossier **Textes de référence et cadres de politiques d'entreprise**.
2. Cliquez sur le lien qui apparaît à droite du dossier.
La fenêtre de recherche apparaît.
3. Sélectionnez le type de réglementation et cliquez sur le bouton **Rechercher**.
La liste des réglementations correspondantes apparaît.
4. Sélectionnez la réglementation voulue et cliquez sur **Relier**.

Vous pouvez définir plus précisément à quelles sections d'une réglementation et/ou d'une règle d'entreprise une entité doit se conformer.

Exemple

Une organisation est contrainte par la réglementation sur l'accessibilité des bâtiments et l'adaptation des conditions de travail aux personnes handicapées.

Pour définir les éléments d'une réglementation d'un catalogue de données :

1. Cliquez sur l'icône du catalogue de données et sélectionnez **Éditer le catalogue de données**.

La fenêtre d'édition du catalogue de données apparaît.

A gauche de l'éditeur apparaissent les textes de référence ou les cadres de politique d'entreprise rattachés au catalogue de données. A droite apparaît le catalogue de données. Pour définir les éléments du catalogue de données, glissez/déplacez les sections de la réglementation de l'arbre de gauche dans l'arbre de droite

Lignage de données d'un catalogue de données

Vous pouvez associer au catalogue de données un lignage de données afin de visualiser l'implémentation et assurer le suivi de la réglementation qui contraint le catalogue des données au sein de l'organisation.

Pour relier un lignage de données au catalogue de données :

1. Ouvrez les propriétés du catalogue de données.
2. Cliquez sur la page **Activités**.
 Si la page n'apparaît pas par défaut, cliquez sur le bouton **Afficher/Masquer**, sélectionnez **Activités** et cliquez sur **Appliquer**.
3. Sous la section **Lignage de données**, cliquez sur **Nouveau** ou **Relier** selon que vous voulez créer un lignage de données ou relier un lignage de données existant.

Voir [Lignages de données](#).

Carte de domaines de concepts d'un catalogue de données

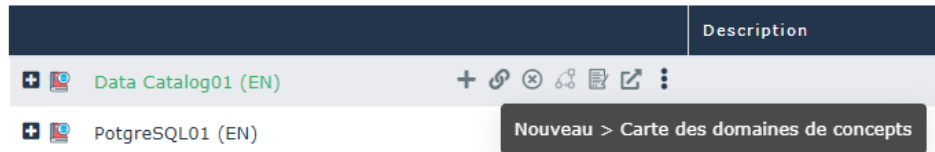
Une carte de domaines de concepts permet d'organiser par domaines l'usage des données métier qui sont implémentées par les métadonnées physiques et les

champs de métadonnées. Ainsi, le catalogue de données ne référence pas uniquement les sources de données physiques et techniques mais apporte également via cette carte un point d'entrée orienté métier.

☛ Pour plus d'informations sur les cartes de domaines de concepts, voir [Carte des domaines de concepts](#) dans le chapitre "Définir les informations métier".

Pour relier une carte de domaines de concepts au catalogue de données :

1. Passez la souris sur le catalogue de données.
2. Cliquez sur le bouton **+ Nouveau > Carte des domaines de concepts**.



Les données métier reliées au catalogue de données par le biais de la carte sont affichées dans la page Données métier des propriétés du catalogue.

Voir aussi : [Collecter des données métier](#).

INFRASTRUCTURE DES SOURCES DE DONNÉES

Organisation des dépôts de données

Hébergement

Les sources de données d'un catalogue sont des dépôts de données déployés issus principalement de la découverte de données effectuée avec l'outil dédié de HOPEX ou un outil intermédiaire tel que l'import Manta.

➡ Voir [Découverte de données](#) et [Lignages de données](#).

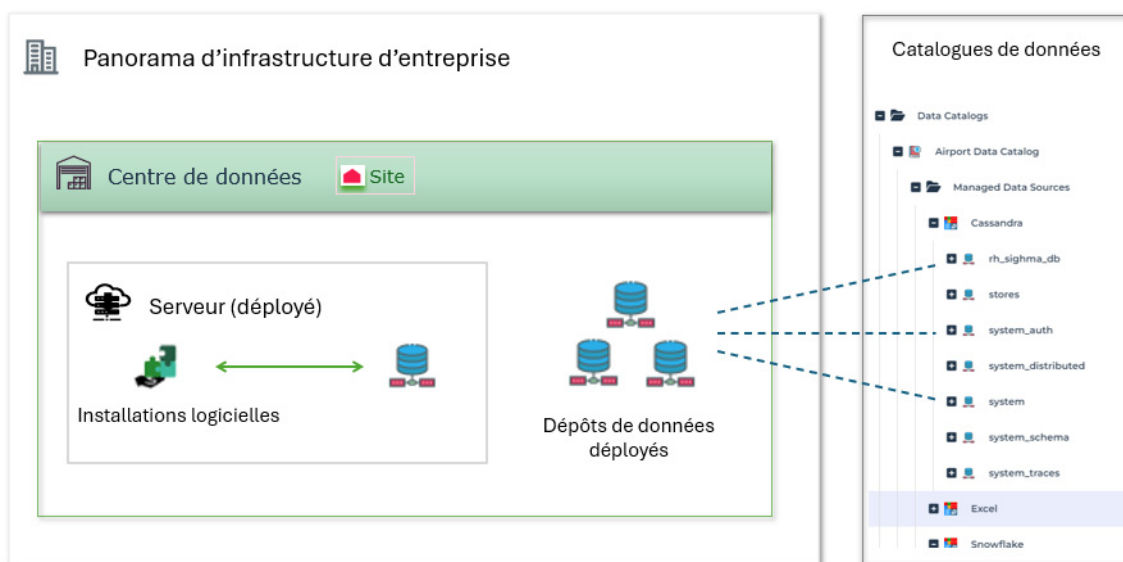
Un dépôt de données est détenu par un centre de données.

📖 *Un centre de données est un site physique qui regroupe des installations informatiques chargées de stocker et de distribuer des données à travers un réseau interne ou via un accès Internet.*

Lors de l'import de métadonnées dans le référentiel HOPEX, un assistant vous permet de définir le centre de données qui héberge les métadonnées et le catalogue qui va référencer les métadonnées, afin de faciliter leur gouvernance.

Dans un centre de données vous pouvez préciser les serveurs déployés dont sont issues les métadonnées importées.

➡ Lors de l'import d'un fichier Manta qui détient la description de métadonnées, HOPEX crée également les serveurs déployés dont sont issues les métadonnées lorsqu'ils n'existent pas dans le référentiel.



Utilisation

Dans un centre de données vous pouvez également préciser les installations logicielles qui utilisent les métadonnées.

Voir ci-dessous comment définir les éléments d'un centre de données.

☛ Voir aussi : [Créer des applications avec Hopex Data Governance](#).

Compléter un centre de données

Vous pouvez visualiser l'infrastructure des centres de données sous le menu de navigation **Inventaires > Paysages > Infrastructure**.

Un centre de données peut être contenu dans un panorama d'infrastructure d'entreprise et se compose des éléments suivants :

- Dépôts de données déployés
- Serveurs déployés
- Installations logicielles

Description	Site	Technologie logicielle	Catalogue de données
Airport Infrastructure Landscape...			
Airport Data Center (EN)	@Cloud		
airport_system		Excel (EN)	Airport Data Catalog
MEGA Bank Infrastructure Landscape (EN)			
Germany Datacenter (EN)	Allemagne		
MCS-SRII (EN)			
USA Datacenter (EN)	Etats-Unis		

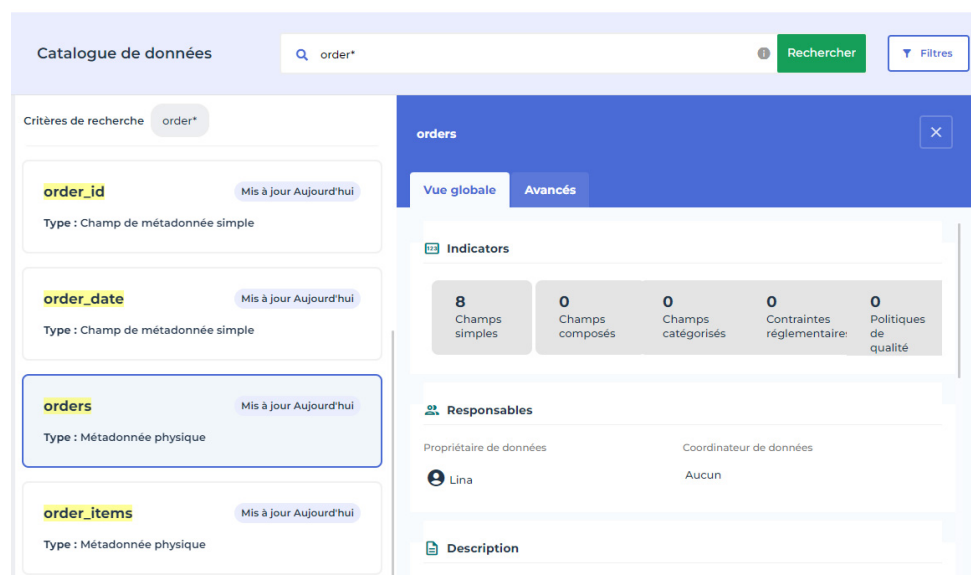
Pour définir les éléments d'un centre de données :

1. Cliquez sur l'onglet **Centres de données**.
2. Cliquez sur le centre de données concerné.
Les propriétés du centre de données apparaissent.
3. Cliquez sur l'onglet qui correspond au type d'objet à définir.
4. Cliquez sur **Nouveau** ou **Relier** selon que vous voulez créer un objet ou relier un objet existant.

RECHERCHE DANS LE CATALOGUE DE DONNÉES

L'outil de recherche **Hopex** permet de rechercher les métadonnées découvertes et importées dans un catalogue de données. Dans les résultats de la recherche vous pouvez :

- identifier l'origine et les responsables des métadonnées
- voir l'usage des métadonnées dans les processus et applications
- accéder au lignage des données techniques, pour visualiser le traitement des données et les flux de données
- voir les politiques de qualité et les réglementations associées



Effectuer une recherche

L'accès à la recherche s'effectue dans la barre d'outils de votre bureau **Hopex**.

➡ Pour plus d'information sur l'outil de recherche, voir [Recherche](#).

Condition préalable

L'utilisation du glossaire nécessite que le référentiel soit indexé. L'indexation du référentiel est effectuée par votre administrateur.

➡ Voir le guide **Hopex Administration - Supervisor**, chapitre "Gérer les référentiels > Utiliser les référentiels > Activer/Désactiver l'indexation d'un référentiel".

Filtres de la recherche

Les métadonnées concernées par la recherche sont les métadonnées physiques, les champs de métadonnées simples et structurés, les schémas, les vues de métadonnées et les dépôts de données déployés. Vous pouvez effectuer une recherche sur l'ensemble ou sur certains de ces objets.

L'outil recherche les chaînes de caractères dans le Nom et la Description des métadonnées.

Pour rechercher une métadonnée dans un catalogue de données :

1. Dans la page d'accueil, cliquez dans le cadre de l'outil de recherche.
2. Cliquez sur le menu déroulant qui apparaît. Il affiche par défaut les objets principaux.
➡ Ensuite, la dernière liste sélectionnée devient la liste par défaut.
3. Sélectionnez **Catalogue de données**.
4. Saisissez le nom ou les premières lettres du nom de la métadonnée recherchée.
➡ Utilisé à la fin d'un mot comme dans «Libéral», l'astérisque permet d'en trouver les différents suffixes comme «Libéralisme» ou «Libéralisation».*

La liste des objets correspondants apparaît.

5. Cliquez sur **Voir tous les résultats** pour afficher une vue détaillée des résultats obtenus et accéder à des filtres supplémentaires.

Filtres des résultats

Vous pouvez affiner les résultats à l'aide des filtres situés dans le panneau de gauche. Les filtres proposés sont fonction des résultats obtenus : par exemple,

lorsque la métadonnée recherchée apparaît dans plusieurs sources, le filtre permet de restreindre la recherche à l'une des sources affichées.

Afficher le détail d'une métadonnée

Lorsque vous sélectionnez une métadonnée dans la liste des résultats, des informations complémentaires apparaissent dans la partie droite de la zone d'édition.

L'onglet **Vue globale** présente :

- Les responsables de la métadonnée
- Des indicateurs sur les objets associés
- Une description
- La structure de la métadonnée

L'onglet **Avancés** présente :

- Les lignages de données
- Les applications et processus qui utilisent la métadonnée
- Les politiques de qualité qui portent sur la métadonnée
- Les réglementations associées

L'onglet **Traitement de données** affiche sous forme de dendrogramme les détails des traitements de données physiques scannés et importés de Manta dans **Hopex Data Governance**.

CRÉER UN CATALOGUE DE DONNÉES

Pour créer un catalogue de données :

1. Allez sous **Catalogues > Hiérarchie**.
2. Passez la souris sur le dossier **Catalogues de données** et cliquez sur le bouton **+** (Nouveau > Catalogue de données) qui apparaît.
3. Dans la fenêtre de création du catalogue vous pouvez définir :
 - le nom du catalogue
 - le détenteur
 - les organisations impliquées
 - les textes de référence et politiques d'entreprises qui s'appliquent aux métadonnées du catalogue

 Pour plus de détails sur les réglementations, voir [Réglementations](#).
4. Cliquez sur **OK**.

Une fois le catalogue créé, vous pouvez définir les dépôts de données déployés.

Workflow du catalogue de données

Un workflow est disponible pour suivre les différents états d'un catalogue de données :

- Préparation
- Validation
- Mise à jour
- Production

Il peut être instancié par le propriétaire du catalogue de données et tout responsable qui lui est assigné peut participer à sa progression.

Des notifications sont envoyées aux différents responsables à chaque fois que l'état du catalogue de données change.

Vous pouvez définir les responsables d'un catalogue de données dans les propriétés du catalogue, sous la page **Caractéristiques > Responsabilités**.

Démarrer le workflow d'un catalogue de données

Pour démarrer le workflow d'un catalogue de données :

1. Ouvrez les propriétés du catalogue de données.
2. Cliquez sur la page **Workflows**.
3. Cliquez sur **Démarrer le workflow**.

ASSOCIER DES MOTS-CLÉS À DES MÉTADONNÉES

Vous pouvez ajouter des tags aux métadonnées physiques, et plus précisément à leurs champs, et ainsi les relier au glossaire métier pour partager la définition.



Un tag est une description typologique qui permet de caractériser les objets. Les tags sont utilisés comme #tag (hashtag) dans l'outil de recherche rapide.

Pour ajouter un tag à une métadonnée physique :

1. Sous le menu **Catalogues > Métadonnées physiques**, cliquez sur la métadonnée pour afficher ses propriétés.
2. Dans la page **Caractéristiques > Identification**, sous le champ **#Tags**, sélectionnez le tag approprié.

DÉFINIR DES CATÉGORIES DE MÉTADONNÉES

Pour définir la catégorie d'une métadonnée physique :

1. Sous le menu **Catalogues** > **Métadonnées physiques**, cliquez sur la métadonnée pour afficher ses propriétés.
2. Dans la page **Caractéristiques** > **Identification**, sous le champ **Catégorie de données**, sélectionnez la catégorie voulue.

Voir aussi [Catégories de données](#).

RAPPORTS D'UN CATALOGUE DE DONNÉES

Dans les propriétés d'un catalogue de données, deux outils vous permettent d'analyser son contenu :

- La vue d'ensemble intégrée au catalogue
- Le rapport de catalogue de données

Vue d'ensemble du catalogue de données

La vue d'ensemble offre une synthèse des propriétés du catalogue.

Elle présente :

- L'identification du catalogue : description et tags associés
- Les organisations impliquées
- Les responsables
- Le nombre de sources de données, de dépôts de données déployés et de métadonnées
- Les métadonnées du catalogue qui participent à la description et à l'implémentation des objets métier.
- La catégorisation de données

Rapport de catalogue de données

Le rapport de catalogue de données permet d'analyser le contenu d'un ou de plusieurs catalogue(s) de données. Il est disponible pour tous les profils de **Hopex Data Governance**.

Pour générer un rapport de catalogue de données :

1. Ouvrez les propriétés du catalogue de données.
2. Cliquez sur la page **Rapport** > **Rapport de catalogues de données**.

Le rapport présente les chapitres suivants :

- Organisations impliquées : entités qui utilisent ou consomment les données du catalogue.
- Textes de référence et cadres de politiques d'entreprise référencés
- Catégories/Sections des réglementations
- Lignages de données détenus. Voir aussi [Lignages de données](#).

RELIER LES MÉTADONNÉES AUX INFORMATIONS MÉTIER

Vous pouvez initialiser un dictionnaire métier à partir de métadonnées. Cela consiste à créer dans un dictionnaire métier les concepts et les éléments d'information qui correspondent aux métadonnées physiques et aux champs de métadonnées, et à établir ainsi un lien de réalisation entre ces objets.

Pour plus de détails, voir [Initialiser un dictionnaire métier à partir de métadonnées physiques](#).

COLLECTER DES DONNÉES MÉTIER

Sous la page **Données métier** des propriétés d'un catalogue de données apparaissent les données métier mises en œuvre par les métadonnées et/ou définies dans les domaines de concepts du catalogue de données.

➤ Pour l'implémentation de données métier par des métadonnées, voir [Relier les métadonnées aux informations métier](#).

➤ Pour la définition de domaines de concepts dans un catalogue, voir [Carte de domaines de concepts d'un catalogue de données](#).

L'administrateur du catalogue de données peut demander aux responsables métier de compléter les propriétés de données métier associées au catalogue. Cette demande se fait au travers d'une session d'évaluation.

Les responsables contactés sont le Propriétaire, le Concepteur et le Coordinateur de données définis sur la donnée en question, ou, à défaut, ceux définis sur le dictionnaire métier ou le domaine de la donnée.

Pour collecter des données métier :

1. Ouvrez les propriétés du catalogue de données concerné.
2. Cliquez sur la page **Données métier**.
3. Dans la liste affichée, sélectionnez les données à compléter.
4. Cliquez sur le bouton **Plus...** puis sur **Collecter les données**.

Général ▾ Characteristics **Données métier** Métadonnées Politique de qualité des données Rapports Tableau de bord

Données métier mises en oeuvre par les métadonnées et/ou définies dans les domaines de concepts du catalogue de données.

Concepts Information Items Others Enlever Lignage de données Rapport instantané 3 sélectionnés

☐	Nom Local ↑	Description	Catégories
☒	Abonnement livre		Collecter les données Excel
☒	Abonnement médiathèque		
☒	Abonnement tout média		
☐	Demande de restitution		
☐	Prêt d'ouvrage		

L'outil d'évaluation apparaît.

5. Planifiez si besoin la session d'évaluation.
6. Cliquez sur **Suivant**.
7. Sélectionnez les éléments que vous souhaitez envoyer aux responsables de données :
 - une ou plusieurs pages de propriétés (par exemple la page Data Quality si vous souhaitez que les responsables évaluent la qualité des données)
 - des caractéristiques avancées (attributs particuliers, par exemple la date de modification, ou propriétés techniques)
8. Cliquez sur **Suivant**.

9. Lancez la session immédiatement ou à une date et heure spécifiques.
Une notification est envoyée aux responsables des données, elle contient un accès direct au formulaire.

Suivi des formulaires de collecte de données

Pour accéder aux formulaires de collecte reçus ou envoyés :

1. Cliquez sur le menu de navigation **Outils** > **Evaluation** > **Collecte de données**.
2. Dans la liste déroulante sélectionnez le type de formulaire à afficher :
 - Formulaires envoyés
 - Formulaires remplis
 - Formulaires non remplis
 - Formulaires en retard

QUALITÉ DES DONNÉES DU CATALOGUE

Pour assurer la qualité des données d'un catalogue, vous pouvez :

- relier une politique de qualité au catalogue de données
- relier plus précisément une politique de qualité des données à des métadonnées physiques du catalogue de données.

Relier une politique de qualité au catalogue de données

Pour relier une politique de qualité à un catalogue de données :

1. Cliquez sur le menu de navigation **Catalogues > Hiérarchie**.
2. Dans la zone d'édition, sélectionnez le catalogue voulu et affichez sa fenêtre de propriétés.
3. Dans la fenêtre de propriétés, cliquez sur la page **Politique de qualité des données**.
4. Cliquez sur **Nouveau** ou **Relier** selon que vous voulez créer ou relier une politique de qualité des données.

Relier une politique de qualité à des métadonnées physiques

Pour relier une politique de qualité à une métadonnée physique :

1. Cliquez sur le menu de navigation **Conformité > Qualité des données**.
2. Cliquez sur **Politiques**.
3. Ouvrez les propriétés de la politique de qualité voulue.
4. Cliquez sur la page **Caractéristiques**.
5. Dans le champ **Sujet de la politique de qualité**, cliquez sur **Nouveau** et sélectionnez l'objet contraint par la politique de qualité.

Voir aussi : [Définir une politique de qualité des données](#).

INSTANTANÉ D'UN CATALOGUE DE DONNÉES

L'objectif des instantanés (snapshots) est de collecter à intervalles réguliers des valeurs d'indicateurs de performance qui permettent de contrôler la qualité des données.

Vous pouvez créer l'instantané d'un ensemble de métadonnées lors de l'import de ces métadonnées dans **Hopex Data Governance**.

Pour visualiser l'instantané d'un ensemble de données physiques :

1. Cliquez sur le menu de navigation **Outils**.
2. Sous **Découverte de données**, cliquez sur **Instantanés**.
3. Sélectionnez :
 - le catalogue de données
 - le dépôt de données déployé
 - l'instantané créé lors de l'import des métadonnées. Voir [Lancer HOPEX Data Discovery](#).

Pour plus d'informations sur le contenu d'un catalogue de données, voir [Contenu d'un catalogue de données](#).



DÉCOUVERTE DE DONNÉES



L'outil **HOPEX Data Discovery** permet d'extraire des données de sources variées et de les importer dans le catalogue de données de votre référentiel.

HOPEX Data Discovery est disponible sous forme de deux modules qui correspondent à des modes d'extraction différents :

- ✓ Le mode Online qui peut être déployé pour les clients qui ont déployé **HOPEX** sur leur propre infrastructure (on-premise, par exemple) ;
- ✓ Le mode autonome (standalone) qui peut être installé dans une infrastructure indépendante de celle où **HOPEX** est installé.

Les points abordés ci-après concernent :

- ✓ [Le module HOPEX Data Discovery](#)
- ✓ [Lancer HOPEX Data Discovery](#)

LE MODULE HOPEX DATA DISCOVERY

Déploiement de HOPEX Data Discovery en mode Online

Le mode Online de l'outil **HOPEX Data Discovery** est disponible sous forme de module, téléchargeable dans la console HAS de HOPEX.

Pour mettre en place le module **HOPEX Data Discovery** :

- 1 A partir de votre version **HOPEX**, ouvrez la console d'administration HAS Console et téléchargez le module "HOPEX Data Discovery".

➡ Pour plus de détails sur l'import d'un module, voir [Importer un module dans HOPEX](#).

The screenshot displays the 'HOPEX Application Server - Console' interface. On the left, a sidebar contains navigation links: 'Installation', 'Modules' (selected), 'Module List', 'Module Settings', 'Authentication', 'Cluster', and 'Monitoring'. The main area is titled 'HOPEX Application Server - Console' and features a tabbed interface with 'Installed (3)', 'Update', 'Add new', and 'Remove' tabs. Below the tabs, a search bar is present with the text 'Find here all the installed modules with their versions.' To the left of the module list, there are filters for 'Type' (Content, Hopex, Redis, Site, WebService) and 'Tags' (Business process analysis, Data governance, which is checked). The module list itself contains three entries: 'HOPEX Data Discovery' (highlighted with an orange box), 'Privacy Management Content', and 'Sample Data Types'. Each entry includes an icon and a brief description.

Type	Module Name	Description
☐ Content	HOPEX Data Discovery	Metadata Discovery tool from Data Sources
☐ Hopex	Privacy Management Content	Add mandatory content to setup Privacy Management solution
☐ Redis	Sample Data Types	Package of Reference Data types
☐ Site		
☐ WebService		

Déploiement de HOPEX Data Discovery en mode autonome

Conditions préalables

L'utilisation de l'outil **HOPEX Data Discovery** est soumise aux conditions suivantes :

- Le poste sur lequel est installé le module en mode autonome doit disposer de **.Net Core 8** et de **Microsoft Edge WebView2**.
- Le module de l'outil **HOPEX Data Discovery** en mode Online doit être installé.

☛ Pour plus de détails sur l'installation du module Online, voir [Déploiement de HOPEX Data Discovery en mode Online](#).

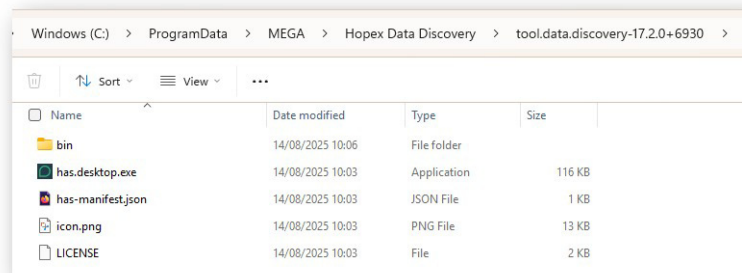
☛ Pour plus de détails sur l'utilisation des fonctionnalités du mode online en utilisation autonome, voir [Lancer l'import en mode autonome](#).

Installation de l'exécutable

Le mode autonome (standalone) de l'outil **HOPEX Data Discovery** est disponible sous la forme d'un exécutable qui vous est envoyé dans dossier compressé (.zip, par exemple).

La procédure à suivre pour déployer **HOPEX Data Discovery** en mode autonome est la suivante :

1. Prenez contact avec votre responsable de compte MEGA afin de définir les modalités d'envoi du dossier compressé.
2. Décompressez le dossier reçu.
Après extraction, le répertoire reçu doit contenir les éléments suivants.



☛ L'exécutable est "**has.desktop.exe**". Pour plus de détails sur son activation, voir [Déploiement de HOPEX Data Discovery en mode autonome](#).

Options de connexion aux sources de données

Les options de connexion pour chaque source de données sont détaillées en annexe du guide Hopex Data Governance. Voir "Appendix - Data Discovery Connectors".

LANCER HOPEX DATA DISCOVERY

L'outil de découverte des données **HOPEX Data Discovery** vous permet de parcourir différentes sources de données et de définir les métadonnées à importer dans votre catalogue de données.

Data Discovery est disponible en mode **Online** à partir du bureau **HOPEX**. Il permet de se connecter aux sources de données et d'importer des métadonnées directement dans **HOPEX**. Pour plus de détails, voir [Utiliser Data Discovery en mode Online](#).

Si vous ne pouvez pas vous connecter à la source de données à partir du bureau **Hopex Data Governance**, pour des raisons de sécurité par exemple, vous pouvez utiliser **Data Discovery** en mode **autonome**. Dans ce mode, la description des métadonnées est exportée dans un fichier Json. que vous pouvez ensuite importer dans HOPEX. Pour plus de détails, voir [Utiliser Data Discovery en mode autonome](#).

Conditions d'accès

L'outil de découverte des données est accessible aux utilisateurs de profil "Administrateur fonctionnel de la gouvernance des données" et "Gestionnaire de données."

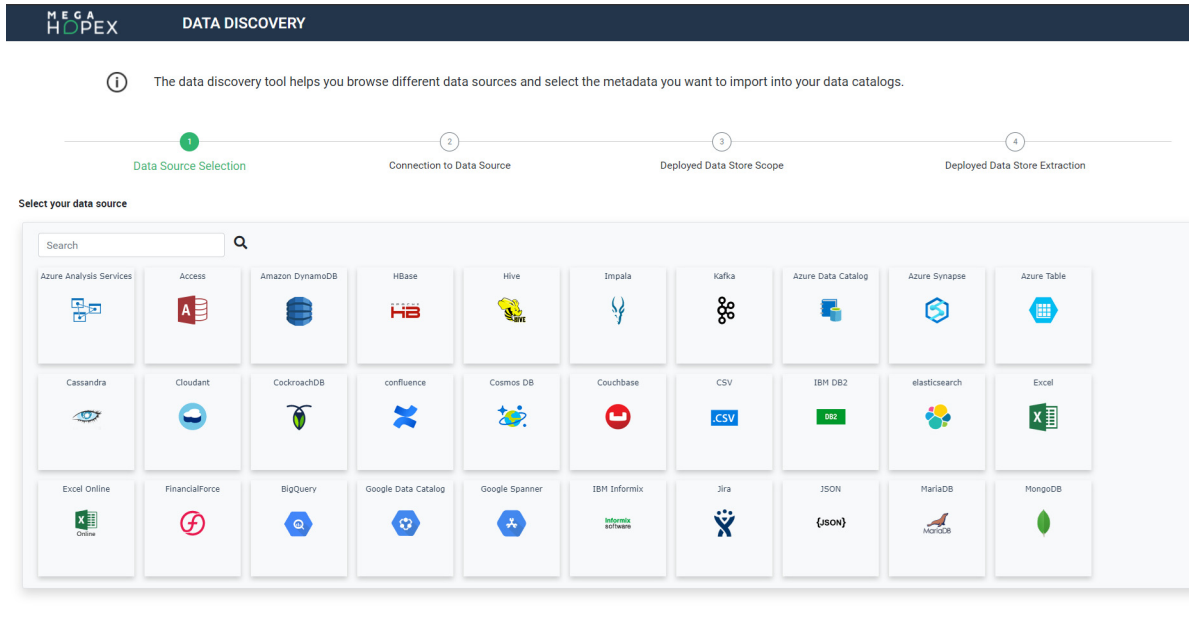
Utiliser Data Discovery en mode Online

 Vous devez avoir acquis la licence **DDISC** pour utiliser l'outil **HOPEX Data Discovery en mode Online**.

Pour lancer l'import de métadonnées en mode Online :

1. Dans la barre de navigation, cliquez sur **Outils**.

2. Cliquez sur **Discovery en ligne**.
L'outil HOPEX Data Discovery s'ouvre dans un nouvel onglet du navigateur.



3. Sélectionnez la source de données. Cliquez sur le bouton **Connect** de la source.
Les paramètres de connexion apparaissent. Ils sont fonction de la source de données choisie. Vous pouvez consulter les informations associées.

☛ En annexe du guide **Hopex Data Governance** sont détaillées les options de connexion du driver JDBC driver pour chaque source de données. Voir "Appendix - Data Discovery Connection Options".

Set your credential to connect to your Data Source.

The screenshot shows a configuration window titled "Cassandra" with an eye icon. It contains several input fields for connection details, each with an information icon (i) to its right:

- Database: Empty text field.
- User: Text field containing "vpl".
- Password: Text field with masked characters "....".
- Server: Empty text field.
- Port: Text field containing "9042".
- Options: Empty text field.

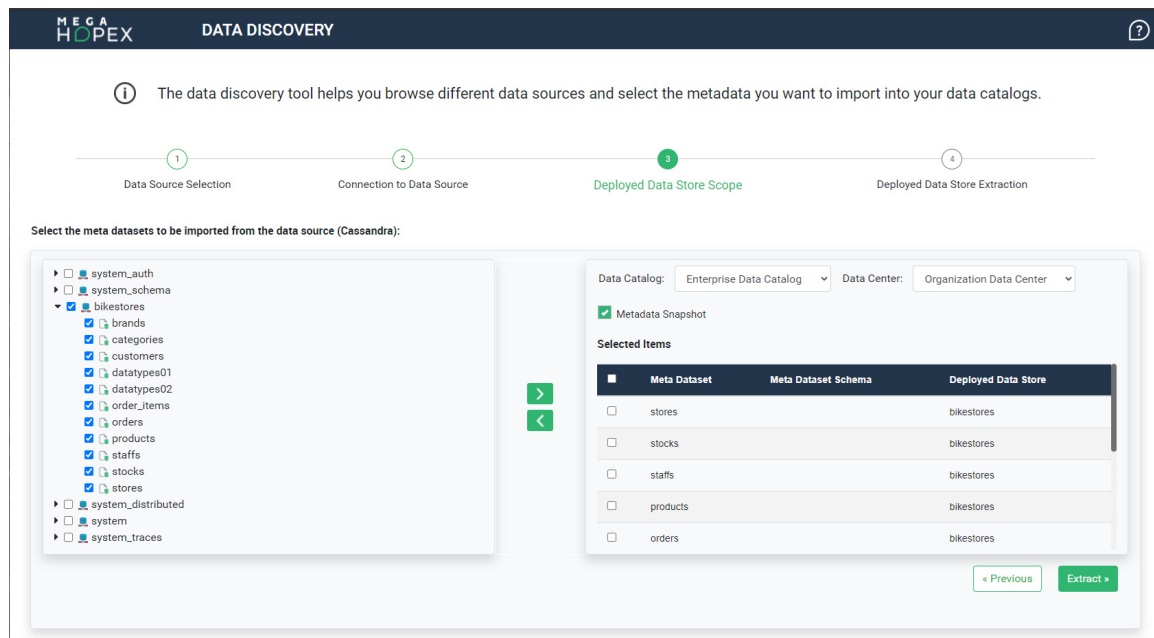
At the bottom right, there is a green "Advanced" button. Below the input fields are two buttons: "« Previous" and "Connect »". A tooltip is visible over the "Server" field, stating: "The host name or IP address of the server hosting the Cassandra".

Le bouton **Advanced** permet l'accès à des paramètres avancés.

4. Une fois connecté, dans l'arbre de gauche, sélectionnez les objets à importer dans HOPEX.
5. A droite, sélectionnez le centre de données du référentiel qui va détenir les métadonnées et le catalogue qui les référence.

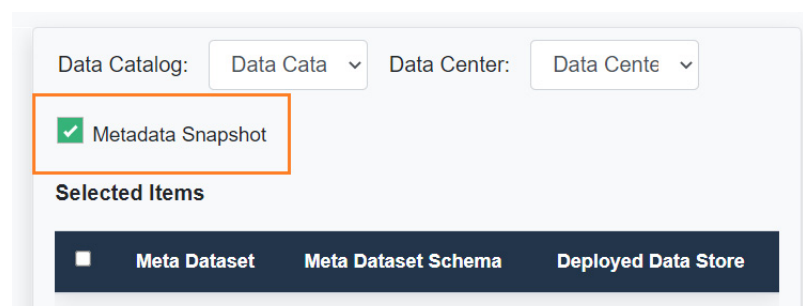
☛ Pour plus d'informations sur les centres de données, voir [Infrastructure des sources de données](#).

6. A l'aide de la première flèche, ajoutez les éléments voulus dans l'arbre du catalogue.



Vous pouvez créer un instantané de l'ensemble de métadonnées sélectionnées en cochant la case **Metadata Snapshot**. Voir [Instantané d'un catalogue de données](#).

Un instantané affiche des indicateurs statistiques relatifs à la qualité des données des métadonnées. Pour les obtenir ces informations, l'outil exécute uniquement des requêtes de **lecture** des données. Ces données des metadata ne sont ni modifiées, ni exportées.



7. Cliquez sur **Extract**.

Pour visualiser dans HOPEX les objets importés :

1. Dans le menu de navigation, cliquez sur **Catalogues**.
2. Affichez la vue hiérarchique.
3. Sélectionnez le catalogue.
4. Dépliez le dossier **Sources de données gérées**.

Voir aussi : [Contenu d'un catalogue de données](#).

Utiliser Data Discovery en mode autonome

L'utilisation de l'outil **HOPEX Data Discovery** en mode autonome est soumise à ses contraintes d'installation.

☛ Pour plus de détails sur les conditions d'utilisation du mode autonome, voir [Déploiement de HOPEX Data Discovery en mode autonome](#).

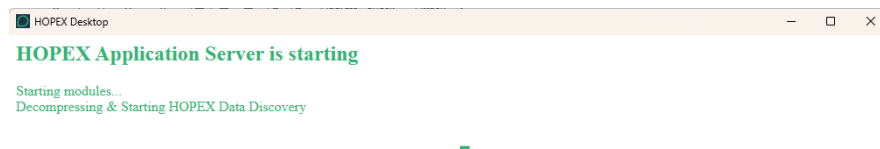
Extraire les schémas des sources de données

Après avoir installé le programme "has.desktop.exe", vous devez lancer son exécution.

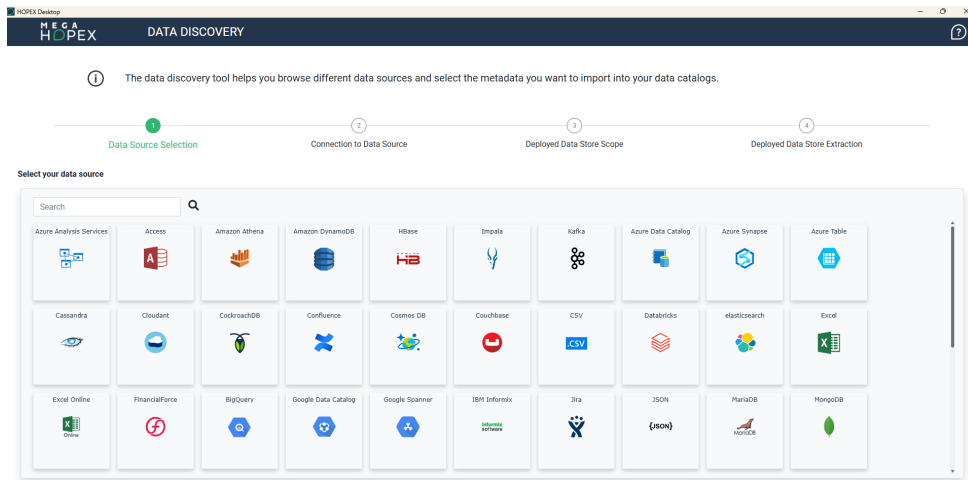
☛ Pour plus de détails sur l'installation du programme "has.desktop.exe", voir [Déploiement de HOPEX Data Discovery en mode autonome](#).

Pour extraire les schémas des sources de données :

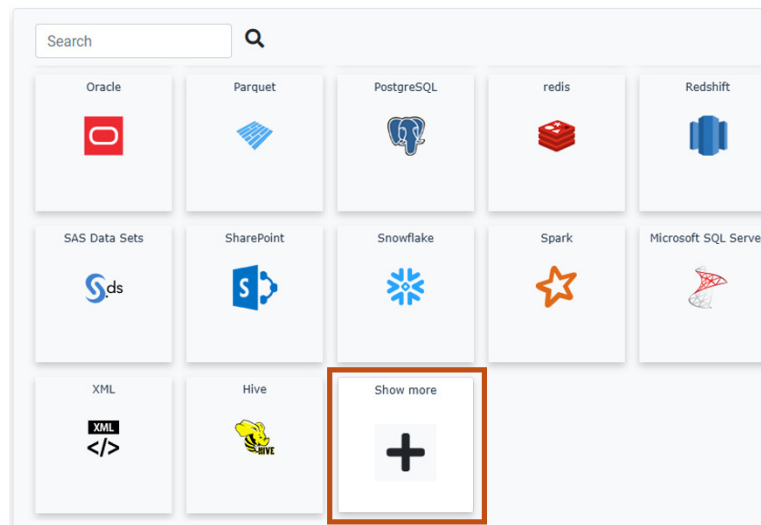
1. Lancez l'exécution de "has.desktop.exe".
L'écran de chargement des différents connecteurs s'affiche.



2. Après ouverture de l'onglet **HOPEX Data Discovery**, vérifiez que tous les connecteurs ont été chargés.



3. A partir du champ **Search**, recherchez le type de votre source de données.
Notez que la tuile **Show more** vous permet d'afficher au premier plan d'autres connecteurs disponibles dans l'outil.

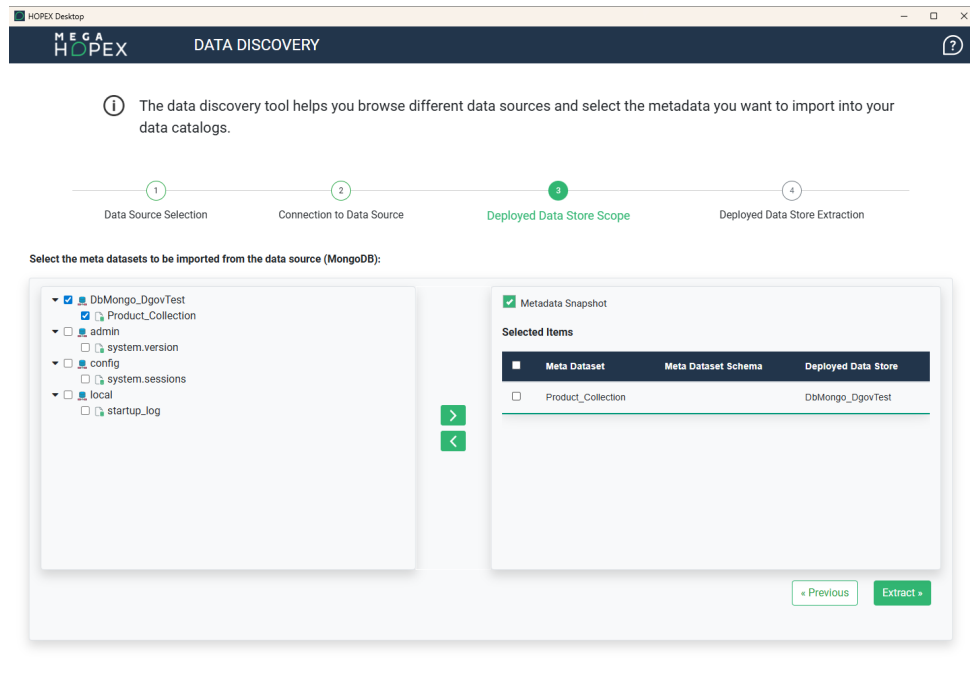


4. Cliquez sur **Connect**.
Une fenêtre de connexion s'ouvre.

The screenshot shows the HOPEX Desktop Data Discovery window. At the top, there's a header with the HOPEX logo and 'DATA DISCOVERY'. Below the header, an information icon is followed by the text: 'The data discovery tool helps you browse different data sources and select the metadata you want to import into your data catalogs.' A progress bar below this shows four steps: 1. Data Source Selection, 2. Connection to Data Source (highlighted in green), 3. Deployed Data Store Scope, and 4. Deployed Data Store Extraction. Below the progress bar, the text 'Set your credential to connect to your Data Source.' is displayed. The main area is a form titled 'MongoDB' with a green leaf icon. It contains input fields for 'Server' (localhost), 'Port' (27017), 'User', 'Password', 'Database', and 'Options'. Each field has an information icon to its right. At the bottom right of the form is an 'Advanced' button. At the bottom of the form are three buttons: '< Previous', 'Test Connect', and 'Connect >'. The 'Test Connect' button is highlighted in green.

5. Spécifiez les paramètres de connexion.
➡ Ces paramètres de connexion dépendent de chaque source de données, des modes de protections et d'accès mis en place dans votre organisation. Contactez votre DBA pour vous les communiquez. Voir également [Options de connexion aux sources de données](#).
6. (Optionnel) Cliquez sur **Test Connect** pour vérifier votre connexion.

7. Cliquez sur **Connect**.
Une nouvelle fenêtre s'ouvre. La connexion à la source de données est établie. Les métadonnées s'affichent dans l'arbre de gauche.



8. Sélectionnez, dans la partie gauche, les objets à importer dans votre référentiel **HOPEX**.
9. A droite, sélectionnez le centre de données du référentiel qui va détenir les métadonnées et le catalogue qui les référence.
➡ Pour plus d'informations sur les centres de données, voir [Infrastructure des sources de données](#).
10. (Optionnel) Pour créer un instantané de l'ensemble des métadonnées sélectionnées, cochez la case **Metadata Snapshot**. Pour plus de détails, voir [Infrastructure des sources de données](#).
11. Cliquez sur **Extract** pour lancer l'extraction des métadonnées.
12. Cliquez sur **Download JSON File** pour télécharger le fichier json du résultat de l'extraction.

Pour importer le fichier Json des résultats de l'extraction dans votre référentiel, voir [Lancer l'import en mode autonome](#).

Lancer l'import en mode autonome

L'outil **HOPEX Data Discovery** en mode Online est indispensable pour réaliser l'import du fichier Json résultant de l'exécution du mode autonome. Pour plus de détails sur le déploiement de ce module, voir [Déploiement de HOPEX Data Discovery en mode Online](#).

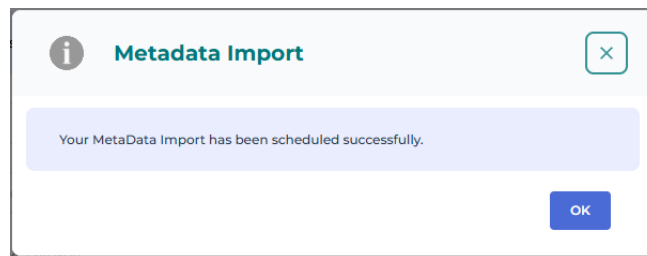
➡ Vous devez avoir acquis la licence **DDISC** pour utiliser l'import de de votre fichier Json avec **HOPEX Data Discovery**.

Pour importer les métadonnées à l'aide du fichier .json :

1. Dans la barre de navigation, cliquez sur **Outils**.
2. Cliquez sur **Import des métadonnées**.
Une fenêtre s'ouvre.

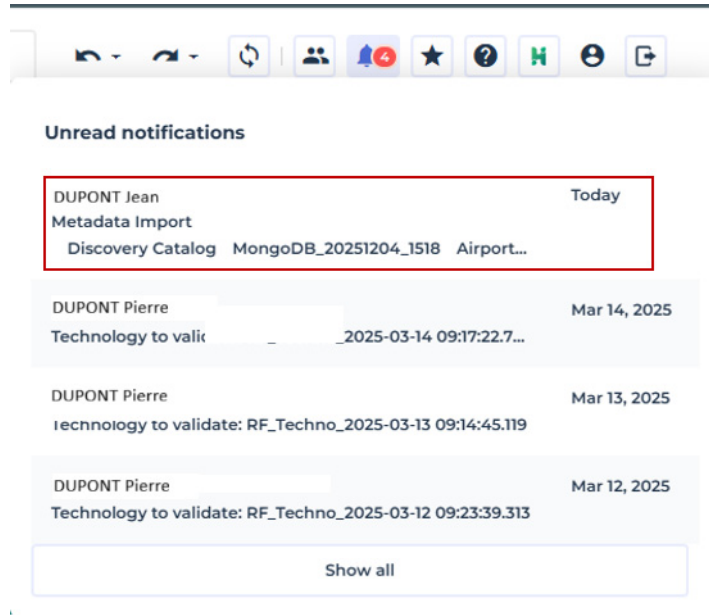
The screenshot shows the 'Data Catalog Import' window. On the left is a dark sidebar with navigation icons for Architecture, Compliance, Inventories, Reports, and Tools. The main panel has a header 'Data Catalog Import' and a sub-header 'This tool allows to import Meta Datasets into HOPEX. Only JSON Files (*.json) are accepted.' Below this are three input fields: 'Data Catalog*' with the value 'Discovery Catalog', 'Data Center*' with the value 'Airport Data Center', and 'File*' with the value 'MongoDB_20251204_1518.json'. At the bottom right, there is a blue button and a status indicator showing 'Data assessed in the last 6 months'.

3. Sélectionnez :
 - le catalogue de données dans lequel importer des données
 - le centre de données détenteur
 - le fichier .json
4. Cliquez sur **OK**.
L'assistant planifie l'import qui sera exécuté en arrière-plan.

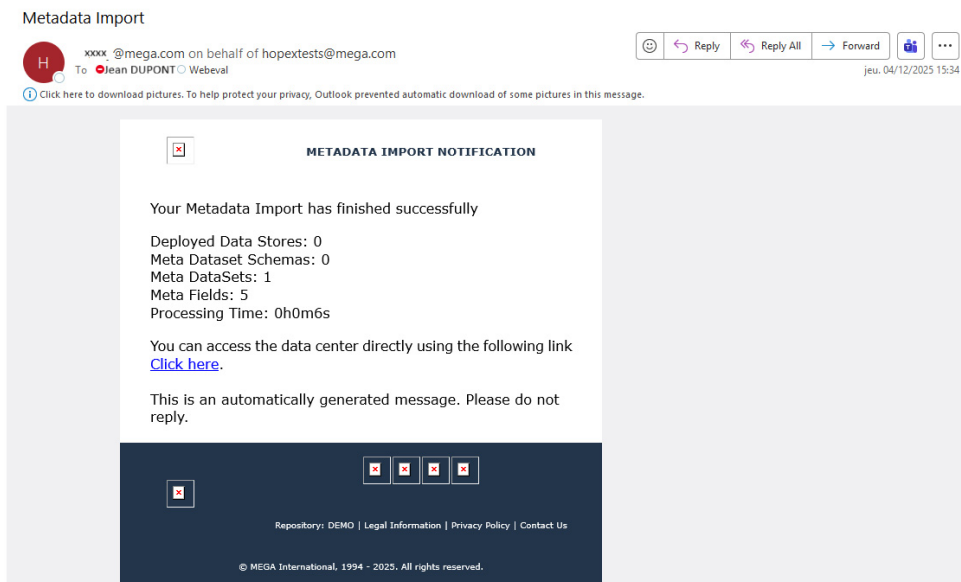


A la fin du traitement, l'utilisateur en charge de l'import est notifié du résultat du traitement par deux canaux :

- les notifications de **HOPEX**,



- il reçoit un email.



Voir aussi : [Options de connexion aux sources de données](#).

Import incrémental

Après une première initialisation de métadonnées dans HOPEX, il est possible de faire d'autres imports de la même source. Dans ce cas, seuls les nouveaux objets et les mises à jour des propriétés sont pris en compte (exemple : création d'une nouvelle métadonnée physique, nouveau champ de métadonnées).

Les métadonnées dans HOPEX sont identifiées par leur nom + le nom du dépôt de données déployé (issu de la base de données source) + le nom du centre de données. Dès qu'un composant du nom est modifié, il y a rupture de correspondance, et de nouveaux objets sont créés. Ainsi, par exemple, si vous modifiez le dépôt de données d'une table dans HOPEX, lors d'un nouvel import à partir de la même source, cette table est recréée dans le dépôt de données initial.

Voir aussi : [Contenu d'un catalogue de données](#).

LIGNAGES DE DONNÉES



Les réglementations financières et les nouveaux cadres tels que la protection des données à caractère personnel (GDPR) requièrent de disposer d'une cartographie et de contrôles sur les traitements des données.

Hopex Data Governance fournit un outil de lignage de données pour assurer la traçabilité des flux de données. Un lignage de données permet de décrire les processus de traitement de l'information et les règles de transformation de la donnée, de la source jusqu'à la restitution. Il facilite ainsi l'identification des erreurs dans les traitements et réduit le risque de non-conformité sur les données.

- ✓ [Lignage de données fonctionnel](#)
- ✓ [Lignage de données technique](#)

LIGNAGE DE DONNÉES FONCTIONNEL

Le lignage de données fonctionnel est un moyen de spécifier, définir ou consulter le lignage de données sans les informations techniques dont l'utilisateur métier n'a pas besoin.

Pour spécifier ou consulter un lignage de données fonctionnel, les gestionnaires de données disposent d'un diagramme dans lequel ils peuvent dessiner les différents traitements et transformations qui s'appliquent aux données indépendamment de leur implémentation. Ils peuvent ainsi spécifier :

- Où se trouvent les données et comment elles sont stockées dans un environnement, par exemple sur un site ou dans un dépôt de données.
- Par quels processus et systèmes les données sont utilisées, et avec quel droit d'accès.
- Qui est responsable de leur mise à jour.
- Etc.

Un rapport d'analyse montre les dépendances qui existent entre les données d'entreprise.

Créer un lignage de données fonctionnel et son diagramme

Vous pouvez créer un lignage de données fonctionnel sur un concept, une classe ou une table. La fonctionnalité est accessible avec le profil Gestionnaire de données.

Pour créer un lignage de données dans **Hopex Data Governance** :

1. Ouvrez les propriétés de l'objet concerné (concept, classe ou table).
2. Cliquez sur la page **Caractéristiques**.
3. A droite de la page, sous le champ **Lignage de données**, cliquez sur le bouton **Nouveau**.
4. Sélectionnez le type de diagramme.

Exemple : un diagramme de lignage de données physique pour une table.

L'option **Initialiser le diagramme** est cochée par défaut. Voir ci-dessous [Initialisation du diagramme](#).

5. Cliquez sur **OK**.
Le diagramme du lignage de données apparaît.

Initialisation du diagramme

Lorsque la donnée décrite par le lignage est utilisée dans des contenus existants ou associée à des règles de calcul, l'option **Initialiser le diagramme** s'affiche ; elle permet d'initialiser le diagramme avec les contenus et les règles en question.

Plus précisément, l'initialisation se base sur :

- les contenus qui portent la donnée dans des flux d'échanges entre logiciels (applications, Application Systems...), décrits dans des scénarios de flux.
- Les logiciels du scénario permettent d'initialiser les sources de données dans le lignage. L'envoi du flux, sa réception et son sens dans le scénario permettent d'initialiser les traitements de données, leurs transitions et leur sens.
- Lorsque les tâches effectuées par le logiciel d'un scénario sont détaillées dans un processus applicatif par exemple, elles apparaissent également dans les nœuds de traitement du lignage.
- les règles de calcul disponibles sur cette donnée. Les informations définies en paramètres d'entrée de ces règles initialisent les nœuds d'origine du lignage.

Workflow du lignage de données

Lors de la création d'un lignage de données (métier, logique ou physique), la première transition du Workflow "En cours de spécification" est créée automatiquement.

Pour plus de détails sur le workflow, voir [Workflow de validation des données](#).

Objets du diagramme de lignage de données fonctionnel

La barre d'outils située à gauche du diagramme indique les objets que vous pouvez insérer dans le diagramme. Certains, tels que les acteurs, peuvent être reliés. D'autres objets, tels que les applications ou services applicatifs, peuvent également être créés.



Participant

Un participant permet de définir où un traitement se produit. Il peut s'agir d'un acteur ou d'un poste.

Ajouter un participant

Vous pouvez ajouter dans le diagramme un acteur ou un poste existant.

Pour ajouter un acteur :

1. Dans la barre d'insertion d'objets, cliquez sur l'icône **Acteur** puis dans le diagramme.
2. Dans la fenêtre qui apparaît, indiquez le nom de l'acteur.
3. Sélectionnez l'acteur qui apparaît dans la liste.
4. Cliquez sur **OK**.

Donnée métier origine

Il s'agit de la ou des donnée(s) en entrée du lignage de données.

Flux de données

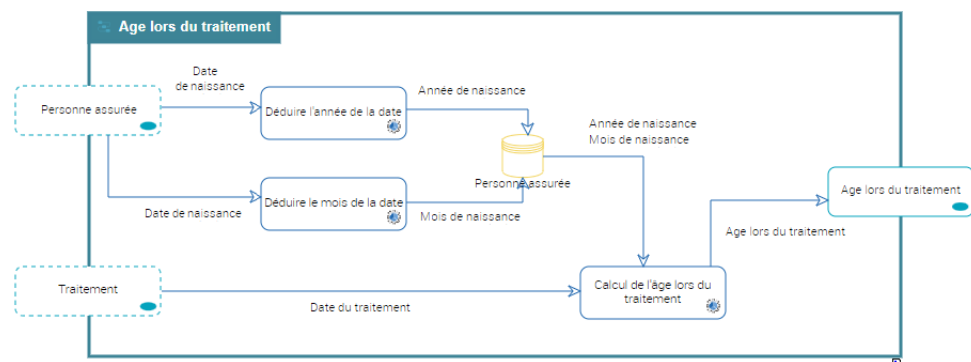
Les flux de données représentent les transitions entre les étapes du lignage de données, via lesquelles sont fournies les données en entrée et en sortie.

Règles de propriété de concept

Une règle de propriété de concept est la collecte et la manipulation d'éléments de données pour produire des informations significatives.

Dans l'exemple ci-dessous, la règle de propriété de concept "Calcul de l'âge lors du traitement" permet de calculer l'âge du patient au moment du traitement. La règle prend comme paramètres :

- le mois et l'année de naissance disponibles dans le dépôt de données "Personne assurée" et déduits des propriétés du concept "Personne assurée".
- la date du traitement, issue du concept "Traitement".



Créer une règle de propriété de concept

Dans l'exemple ci-dessus, pour définir la règle qui calcule l'âge de la personne assurée au moment du traitement, vous devez :

- Créer une règle de propriété de concept
- Créer les flux entre les sources d'information et la règle de calcul
- Indiquez sur les flux les informations transmises
- Indiquez sur la règle les informations en entrée et les informations en sortie

Pour cela :

1. Dans la barre d'insertion des objets, cliquez sur **Règle de propriété de concept**.
2. Cliquez dans le diagramme.
3. Indiquez le nom de la règle ("Calcul de l'âge lors du traitement") et cliquez sur **Suivant**.
La règle apparaît dans le diagramme.
4. Cliquez sur l'icône **Flux de données** et créez un flux de données entre le dépôt de données "Personne assurée" et la règle de propriété de concept.
5. Sélectionnez le flux de données et cliquez sur le bouton **Propriétés**.
6. Cliquez sur la page **Caractéristiques**.
7. Sous la section **Owned Data Property Lineage**, ajoutez les propriétés "Mois de naissance" et "Année de naissance".
8. Créez un second flux entre le Concept "Traitement" et la règle de propriété de concept.
9. Sur ce flux ajoutez la propriété "Date du traitement".
10. Sélectionnez la règle de propriété de concept pour afficher ses **Propriétés**.
11. Cliquez sur la page **Caractéristiques**.
12. Sous la section **Paramètres**, sélectionnez :
 - les **paramètres en entrée** de la règle : "Mois de naissance", "Année de naissance" et "Date du traitement".
 - le **paramètre en sortie** de la règle : "Age lors du traitement".

Mesure de qualité des données

Une mesure de qualité des données contrôle que les données issues d'un traitement sont conformes aux exigences de qualité. Elle permet ainsi d'éviter les risques (de données erronées ou manquantes par exemple) liés à un calcul ou un traitement sur les données.

Ajouter une mesure de qualité des données

Dans le diagramme de lignage vous pouvez relier des mesures existantes. Pour cela :

1. Dans la barre d'insertion d'objets, cliquez sur l'icône **Mesure de qualité des données**.
2. Cliquez dans le diagramme.
3. Dans la fenêtre qui apparaît, saisissez le nom ou les premières lettres du nom de la mesure de qualité des données.
La mesure de qualité correspondante apparaît dans la liste déroulante.

4. Sélectionnez-la.
5. Cliquez sur **OK**.

Pour accéder à l'ensemble des mesures de qualité des données, cliquez sur le menu de navigation **Efficacité opérationnelle > Lignages de données > Inventaire des mesures de qualité**.

Logiciels de traitement

Il s'agit des systèmes logiciels qui effectuent les traitements de données. Vous pouvez sélectionner comme source logicielle un des éléments suivants :

- Système applicatif
- Application
- Microservice
- Service IT

Vous pouvez ainsi visualiser la traçabilité de l'utilisation des données au niveau du logiciel. Le système logiciel peut être sous la responsabilité d'un acteur ou d'un type de poste.

Sur le système logiciel vous pouvez définir :

- les dépôts de données (domaines de données) qui contiennent les informations mémorisées (donnée origine, flux de données...), le CRUD défini.
- les composants logiciels qui accèdent au dépôt de données
- les processus applicatifs où les données sont traitées

Ajouter un logiciel de traitement

Vous pouvez ajouter dans le diagramme un logiciel existant ou en créer un nouveau.

Pour créer une application dans le diagramme par exemple :

1. Dans la barre d'insertion d'objets, cliquez sur l'icône **Application** puis dans le diagramme.
2. Dans la fenêtre qui apparaît, saisissez le nom de l'application.
3. Cliquez sur **Suivant**.
4. Indiquez éventuellement son détenteur.
5. Cliquez sur **OK**.
L'application apparaît dans le diagramme.

Technologie

Les technologies représentent les composants nécessaires au fonctionnement des logiciels (serveurs, systèmes d'exploitation, etc.) qui interviennent dans le cycle de vie des données.

LIGNAGE DE DONNÉES TECHNIQUE

Un lignage de données technique est la représentation technique et détaillée des transformations qui s'appliquent aux données. Cette représentation est destinée aux gestionnaires de données techniques et aux ingénieurs de données.

Depuis HOPEX vous pouvez accéder aux lignages de données techniques disponibles dans MANTA. MANTA est une plate-forme automatisée de lignage de données conçue pour extraire, à partir d'une sélection de systèmes sources (sources de données, systèmes de fichiers), les informations techniques des flux, transformations et dépendances de données.

Présentation de l'import MANTA

L'intégration MANTA et HOPEX consiste à importer dans HOPEX le dictionnaire des métadonnées scannées par MANTA avec leurs descriptions et leurs références (URL) afin de pouvoir consulter depuis HOPEX leur lignage technique dans MANTA.

Les objets importés sont des métadonnées issues de sources de données, comme les stores de données déployés, les schémas, les métadonnées physiques et les champs de métadonnées physiques.

CORRESPONDANCE DES OBJETS		
MANTA	HOPEX	Propriétés
Serveur	Serveur déployé	Identifiant MANTA, nom
Base de données	Dépôt de données déployé	Identifiant MANTA, nom
Schéma	Schéma de métadonnées physiques	Identifiant MANTA, nom, lien MANTA
Table	Métadonnée physique	Identifiant MANTA, nom, description, lien Manta
Colonne	Champ de métadonnée simple	Identifiant MANTA, description, type de donnée, longueur, décimale, lien MANTA, champ identifiant, champ référant
Vue	Vue de métadonnée	Identifiant MANTA, nom, lien MANTA
SGBD	Technologie logicielle	Nom

Les traitements (procédures stockées, fonctions, triggers, programmes, scripts, rapport, lignage technique) ne sont pas importés dans HOPEX. Ces informations sont stockées dans MANTA et référencées par HOPEX à travers les métadonnées.

Voir aussi [Infrastructure des sources de données](#).

Conditions préalables

Les fonctionnalités liées au lignage de données technique sont disponibles avec l'option de produit **HTDL**, protégée par une licence.

Autoriser l'url d'accès aux objets dans MANTA

Pour visualiser les lignages des données techniques de MANTA depuis HOPEX, l'accès doit être autorisé entre le serveur MANTA et celui de HOPEX qui peut être déployé sur site ou en Cloud.

Aussi, assurez-vous que les URL des objets extraits de MANTA et importés dans HOPEX sont publiques et accessibles pour la solution **Hopex Data Governance**. Ces URL sont générées dans le fichier "Dictionary.csv", dans la colonne "manta_hyperlink".

Rapprochez-vous de votre administrateur MANTA afin de spécifier une URL qui permette l'accès à MANTA depuis HOPEX.

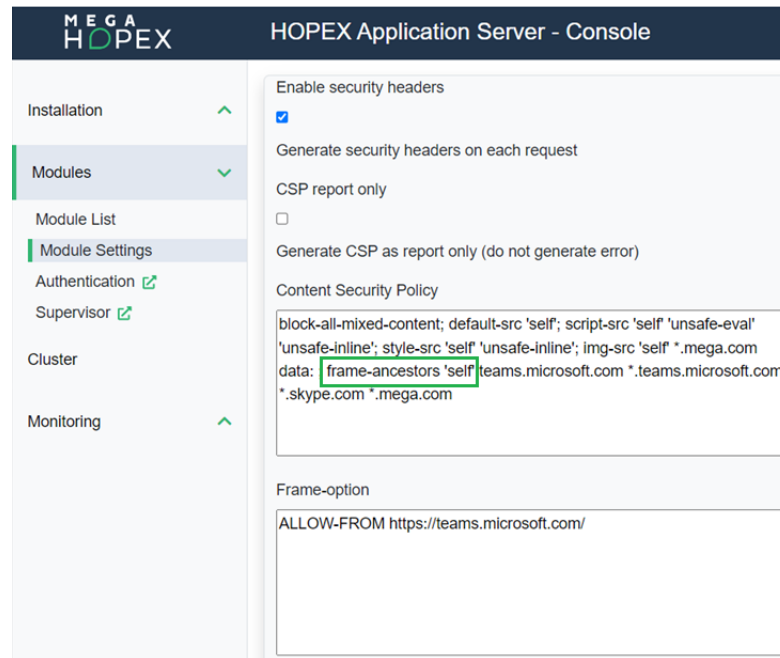
D		E	F	
name	object_name	object_description	manta_hyperlink	
	VP-SNY-MANTA\SQLSERVER2019		Visualization of data lineage in MANTA Visualization of indirect data lineage in MANTA	("COR CT_St
	Hotel_Booking		Visualization of data lineage in MANTA Visualization of indirect data lineage in MANTA	("COR CT_St
	tel_Book dbo		Visualization of data lineage in MANTA Visualization of indirect data lineage in MANTA	("COR CT_St

Paramétrer les options CSP (Content Security Policies)

Pour visualiser les URL importées de MANTA, vous devez également ajouter la directive frame-ancestors 'self' dans les CSP afin d'autoriser l'intégration de ressources externes :

1. Ouvrez la console HAS.
2. Cliquez sur le menu **Modules > Module Settings**.

3. Dans le champ **Content Security Policy**, ajouter frame-ancestors 'self'.



Paramétrer un nom de serveur unique lors de l'extraction des métadonnées

MANTA extrait généralement les métadonnées à partir d'un ensemble de connexions pour une technologie donnée (par exemple, toutes les bases de données Oracle).

Dans le cas où MANTA extrait des noms de serveurs différents à chaque connexion, il est important de préciser un nom unique pour le serveur. En effet, ce nom est utilisé pour identifier le namespace des métadonnées exportées de MANTA et importées dans HOPEX.

Pour paramétrer le nom du serveur :

1. Dans **Manta Admin UI**, ouvrez les paramètres de connexion de la source de données.

2. Dans le champ **User-set Server name**, saisissez le nom du serveur.

The screenshot shows the 'MSSQL Connection Hotel_Booking' configuration page in the Manta Admin UI. The page is divided into two main sections: '1. Name the connection' and '2. Setup the connection'. In the first section, the 'Dictionary ID' is set to 'Hotel_Booking'. In the second section, the 'Sub-dialect' is 'SQL Server (2008 and n...)', the 'Connection string' is 'jdbc:sqlserver://VP-SNY-MANTA:50040', the 'Authentication type' is 'SQL Server', the 'Username' is 'mega', and the 'Password' is masked. The 'User-set server name' field is highlighted with a tooltip that explains its purpose: 'User-set server name. Leave blank, to use the automatically extracted server name. Set to a custom unique server name to override this automatically extracted server name. For example, in Multi-AZ deployments for Amazon RDS for Microsoft SQL Server, the value should specify a "cluster name", because (1) any of the individual SQL Server instances could be used during extraction (and therefore the server name can change over time for MANTA) and (2) also the individual server names are re-generated after each reboot with failover by AWS. The physical identifier of this property is MSSQL_SERVERNAME.' Below this, the 'In' field is set to 'Hotel_Booking/dbo'.

Gestion des révisions dans MANTA

Pour scanner un système source, MANTA se connecte à ce système et analyse les métadonnées avant de les introduire dans son référentiel. Lors d'un scan, les nouvelles métadonnées sont chargées directement ; les métadonnées actualisées sont comparées aux actifs existants afin de détecter les changements dans le lignage de données.

Chaque scan d'un système source donne lieu à une révision dans MANTA. Une révision est une tranche de temps qui montre comment le système analysé se présentait au moment du scan.

Lorsque vous lancez un scan dans MANTA, vous pouvez créer une nouvelle révision ou enrichir une révision existante par des révisions mineures.

Il est important de noter que les métadonnées déjà importées dans HOPEX doivent toujours exister dans la dernière révision majeure ou dans ses révisions mineures afin que l'URL MANTA fournie dans HOPEX donne accès à leurs lignages de données. En effet, l'URL affichée dans HOPEX correspond à la dernière version majeure dans MANTA.

Importer des données de MANTA

Les données issues de MANTA sont exportées dans un fichier .csv, nommé Dictionary.csv. Ce fichier peut ensuite être importé dans HOPEX dans un fichier .zip, à destination d'un catalogue de données.

Pour importer le fichier Dictionary.csv dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Outils > Import Manta**.
2. L'assistant d'import d'un lignage de données technique apparaît.
3. Indiquez :
 - le catalogue de données qui va référencer les métadonnées importées
 - le centre de données qui va détenir les métadonnées
 - le fichier .zip contenant le lignage de données issu de MANTA.

☛ Pour que l'import dans HOPEX fonctionne, veillez à ce que les fichiers .csv soient à la racine du fichier .zip et non dans des dossiers.
4. Cliquez sur **OK**.
La fenêtre du planificateur apparaît. Elle indique que l'import du fichier va se faire en mode batch.
5. Cliquez sur **OK**.

A la fin de l'import, une notification et un email sont envoyés à l'utilisateur qui a lancé l'import et au propriétaire du catalogue de données.

Une fois les métadonnées importées dans HOPEX, il est possible de visualiser leur lignage de données technique en appelant l'outil Manta Viewer.

Visualiser le lignage de données technique

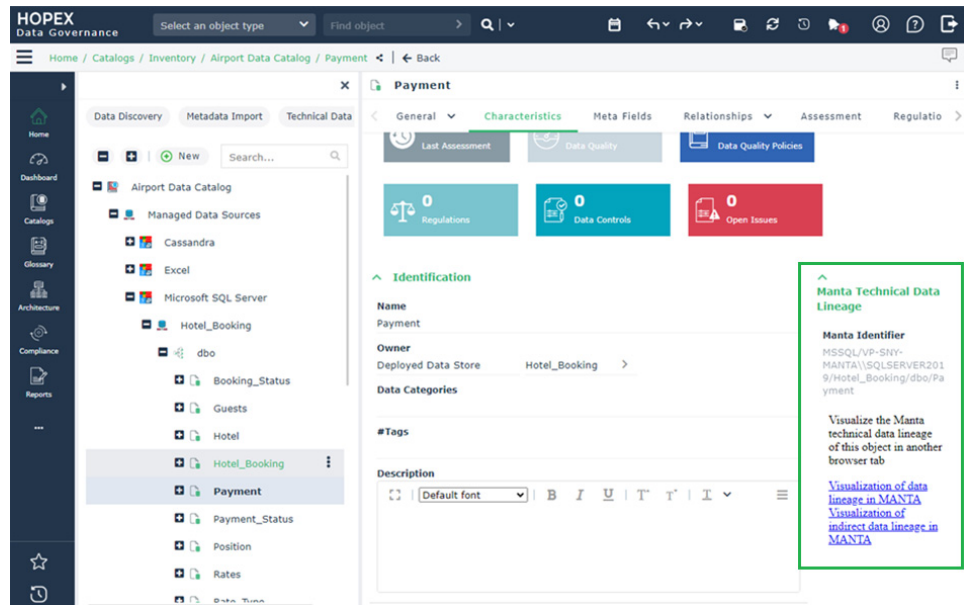
Accéder au lignage de données technique

Vous pouvez accéder depuis **Hopex Data Governance** au lignage de données MANTA d'une métadonnée physique. Pour cela :

1. Ouvrez les propriétés de la métadonnée physique.

☛ Il doit s'agir d'une métadonnée qui a été importée de MANTA.
2. Cliquez sur la page **Caractéristiques**.

3. A droite de la page, cliquez sur la visualisation MANTA.



4. Connectez-vous à votre compte MANTA et visualisez le lignage de données.

A stylized, light blue globe with a textured, watercolor-like appearance, centered on the page. The globe shows the outlines of continents and is slightly tilted.

Glossaire métier

INTRODUCTION À LA CONCEPTION D'UNE ONTOLOGIE MÉTIER



Les solutions **Hopex Data Governance** et **Hopex Data Architecture** offrent les outils de gestion et de partage du vocabulaire spécifique à votre entreprise. Vous pouvez recenser, définir, classer et organiser les concepts métier afin d'établir un lien pertinent avec les objets techniques mis en œuvre au niveau du système d'information.

Au niveau métier, ces solutions proposent aux intervenants métier un outillage pour décrire les concepts qu'ils manipulent ainsi que les liens qui régissent leur organisation. Pour ce faire, **MEGA** s'est appuyé sur les principes largement répandus du web sémantique, ainsi que des cadres ontologiques tels qu'IDEAS ou la norme ISO 15926 (type de haut niveau, cycle de vie et événements).

Au niveau de l'architecture des SI, **Hopex Data Governance** et **Hopex Data Architecture** offrent les outils pour établir des correspondances entre vos modèles de données, basés sur le formalisme UML, et les informations décrites au niveau métier.

☛ Pour plus de détails sur l'interface et les fonctionnalités **Hopex** en général, voir [Le bureau Hopex](#)

Le processus de gestion du vocabulaire

La définition des informations métier passe par la création d'un dictionnaire métier et des termes de ce dictionnaire.

Un terme est la désignation d'un concept dans une langue donnée.

Exemple : le concept "Pays" a pour termes "Pays" en français et "Country" en anglais.

Le concept porte la définition du terme. Un terme peut être associé à plusieurs concepts, donc plusieurs définitions.

Exemple : le terme "Ordre" peut signifier "Disposition d'éléments dans un ensemble" ou "Indication autoritaire à laquelle il faut obéir".

Afin qu'intervenants métier et intervenants SI partagent un vocabulaire commun, **Hopex Data Governance** et **Hopex Data Architecture** reposent sur deux fonctions majeures :

- L'analyse et l'organisation des concepts métier,
- La mise en relation des concepts métier avec les éléments de l'architecture du système d'information.

Voir aussi : [Consulter le glossaire métier](#)

L'analyse et l'organisation des concepts métier

Ce travail est réalisé par un utilisateur métier. Il consiste à décrire l'ensemble des concepts métier en s'appuyant sur un modèle sémantique simple articulé autour des notions de concept, d'événement et d'état.

- Un concept représente un objet métier, il est caractérisé par :
 - son périmètre, c'est-à-dire ses relations avec les autres concepts

Par exemple, un ouvrage est caractérisé par son auteur, son titre, sa date d'édition, etc.

- ses liens d'héritage avec d'autres concepts

Par exemple, un abonnement est un abonnement livre ou un abonnement média.

- ses occurrences,

Par exemple, Alexandre Dumas est une occurrence d'Auteur.

- Un Etat permet d'identifier une évolution temporelle d'un concept

Par exemple, un ouvrage est disponible ou emprunté.

- Un Événement, représente un fait marquant qui modifie l'état d'un ou de plusieurs concepts.

Par exemple, la parution d'un ouvrage.

Le profil standard "Architecte de données" est utilisé pour assurer le travail d'analyse et d'organisation des concepts métier.

Voir [Définir les informations métier](#)

La réalisation des concepts

Les concepts métier sont, en général, mis en œuvre dans le SI avec le support de la méthode et du formalisme UML.

Le travail de "réalisation des concepts" consiste à rapprocher les éléments des modèles de données avec les concepts métier afin de :

- définir de manière plus précise les objets manipulés au niveau de l'architecture du SI,
- assurer un meilleur partage du vocabulaire et une meilleure communication globale entre intervenants métier et intervenants SI.

Le profil standard "Architecte de données" assure le travail de "réalisation des concepts".

Voir [Relier les concepts métier à l'architecture logique et physique](#).

CONSULTER LE GLOSSAIRE MÉTIER



Hopex Data Governance et **Hopex Data Architecture** offrent un outil de consultation des termes, à partir desquels vous pouvez générer un glossaire métier.

Voir :

- ✓ [Rechercher des termes dans le glossaire métier](#)
- ✓ [Générer un glossaire](#)

Vous pouvez également initialiser un glossaire métier à partir de données existantes. Voir [Initialiser un dictionnaire métier à partir de données logiques ou physiques](#).

Les termes créés peuvent être classés dans des dictionnaires métier. La description de dictionnaires métier et de l'ensemble des éléments de construction de l'ontologie métier vient enrichir les glossaires. Pour plus d'informations, voir [Les éléments d'un dictionnaire métier](#).

RECHERCHER DES TERMES DANS LE GLOSSAIRE MÉTIER

Vous pouvez rechercher des termes dans le glossaire métier de votre référentiel.

Un terme est le nom d'un concept, d'une propriété de concept ou toute autre information métier, dans une langue donnée. Ces concepts ou propriétés de concept portent la définition du terme.

Exemple : le concept "Pays" a pour termes "Pays" en français et "Country" en anglais.

Voir aussi : [Concept et terme](#).

Condition préalable

L'utilisation du glossaire nécessite que le référentiel soit indexé.

Pour indexer le référentiel, voir "Activer et personnaliser l'indexation d'un référentiel" dans le guide de l'administrateur.

Périmètre de recherche

La recherche d'un terme dans le glossaire porte sur toutes les définitions du glossaire qui contiennent le terme en question ainsi que sur les éléments du glossaire qui sont associés au terme.

Par exemple, si "Réservation" est associé à "Client", lors d'une recherche sur "Client", "Reservation" apparaît aussi dans les résultats.

Lancer la recherche

L'accès à la recherche s'effectue dans la barre d'outils de votre bureau **Hopex**.

➡ Pour plus d'information sur l'outil de recherche, voir [Recherche](#).

Filtres de la recherche

Les objets concernés par la recherche sont les concepts, les concepts d'état, les concepts types, les concepts événements, les propriétés de concepts, les vues de

concepts. Vous pouvez effectuer une recherche sur l'ensemble ou sur certains de ces objets.

Pour rechercher la définition d'un terme :

1. Dans la page d'accueil, cliquez dans le cadre de l'outil de recherche.
2. Cliquez sur le menu déroulant qui apparaît. Il affiche par défaut les objets principaux.
3. Sélectionnez **Glossaire métier**.
4. Saisissez le terme ou les premières lettres du terme recherché pour afficher la liste des termes correspondants.

☛ *Utilisé à la fin d'un mot comme dans «Libéral*», l'astérisque permet d'en trouver les différents suffixes comme «Libéralisme» ou «Libéralisation».*

La liste des objets correspondants apparaît.

5. Cliquez sur **Voir tous les résultats** pour afficher une vue détaillée des résultats obtenus et accéder à des filtres supplémentaires.

Filtres des résultats

Vous pouvez affiner les résultats à l'aide des filtres situés dans le panneau de gauche. Les filtres proposés sont fonction des résultats obtenus : par exemple,

lorsque le terme recherché apparaît dans plusieurs dictionnaires métier, le filtre permet de restreindre la recherche à l'un des dictionnaires métier trouvés.

Filtres

Dictionnaire métier

Rechercher

Media Library Vocabulary

Publishing Secteur Vocabulary

TOGAF 9 Definitions

TOGAF 9 Supplementary Definitions

Common

work

Work Edition

Mise à jour : Hier

A **Work** Edition includes all Published **works** of a **Work**, edited "from substantially the same setting of type" including all minor variants (typographical, online quality, ...).

Dictionnaire métier : Publishing Secteur Vocabulary

Artistic Product

Mise à jour : Hier

The abstraction of Body of **work** and **Work**.

Dictionnaire métier : Publishing Secteur Vocabulary

Work

Mise à jour : Hier

An artistic creation, such as a painting, sculpture, or literary or musical composition; a **work** of art.

Dictionnaire métier : Publishing Secteur Vocabulary

Afficher le détail d'un terme

Pour afficher les caractéristiques d'un terme :

- 1 Dans les résultats de la recherche, cliquez sur le terme en question. Une fenêtre apparaît à droite de la recherche, avec deux onglets : **Vue globale** et **Avancés**.

The screenshot displays the 'Glossaire métier' (Job Glossary) interface. At the top, there is a search bar with the text 'client*' and a green 'Rechercher' button. To the right of the search bar is a 'Filtres' button. Below the search bar, the 'Critères de recherche' (Search criteria) section shows 'client*'. The main content area is divided into two columns. The left column lists search results for 'client*'. The first result is 'Client', which is highlighted with a yellow background. It includes a comment: 'Commentaire : Une personne ou une organisation qui achète...' and a dictionary entry: 'Dictionnaire métier : Réservation Vacances'. The second result is also 'Client', with a comment: 'Commentaire : Une personne ou une organisation qui achète...' and a dictionary entry: 'Dictionnaire métier : Reverse Finance Data'. The third result is 'Réservation', with a comment: 'Commentaire : Action de réserver un forfait voyage compre...'. The right column shows the detailed view of the 'Client' term. It has a blue header with the title 'Client' and a close button. Below the header are two tabs: 'Vue globale' (selected) and 'Avancés'. The 'Vue globale' tab contains a section titled 'Responsables' (Responsible) with a table showing the roles of 'Propriétaire de données' (Data owner) and 'Coordinateur de données' (Data coordinator). The 'Définition' (Definition) section provides a description: 'Une personne ou une organisation qui achète un bien ou un service de la société MEGA Travel.'

Responsables	
Propriétaire de données	Coordinateur de données
Aucun	Aucun

Définition

Une personne ou une organisation qui achète un bien ou un service de la société MEGA Travel.

Caractéristiques standard

L'onglet **Vue globale** affiche :

- le propriétaire et le coordinateur de la donnée métier
- le type d'objet, par exemple, Concept, Concept type, etc.
- le statut dans le workflow d'approbation de la donnée. Voir [Workflow de validation des données](#).
- la définition
- les synonymes, la catégorie de données, le domaine métier, etc.
- les propriétés associées : par exemple, le concept "Client" est associé au concept "Réservation". C'est pourquoi le concept "Réservation" apparaît dans le résultat de la recherche portant sur le terme "Client".

Propriétés associées

Nom	Définition	Type	Présence	Cardinalité
Réservation		Concept	Toujours	*

- (dans **Hopex Data Governance**) les métadonnées associées, qui réalisent la donnée métier recherchée.
Pour chaque métadonnée sont affichés :
 - le catalogue de données
 - la source de données
 - le type d'objet : métadonnée physique ou champ de métadonnée
 - la qualité : la valeur affichée est la moyenne des différentes valeurs de la dernière évaluation effectuée sur la métadonnée.
 - les réglementations associées

➡ Pour plus d'informations sur les réalisations, voir [Relier les concepts métier à l'architecture logique et physique](#).

Caractéristiques avancées

L'onglet **Avancés** affiche :

- Les lignages qui contiennent la donnée. Voir [Lignages de données](#).
- Les applications et processus qui font usage de la donnée. Voir [Utilisation des données par le SI](#).
- les politiques de qualité associées à la donnée. Voir [Définir une politique de qualité des données](#).
- les réglementations qui portent sur la donnée. Voir [Réglementations](#).

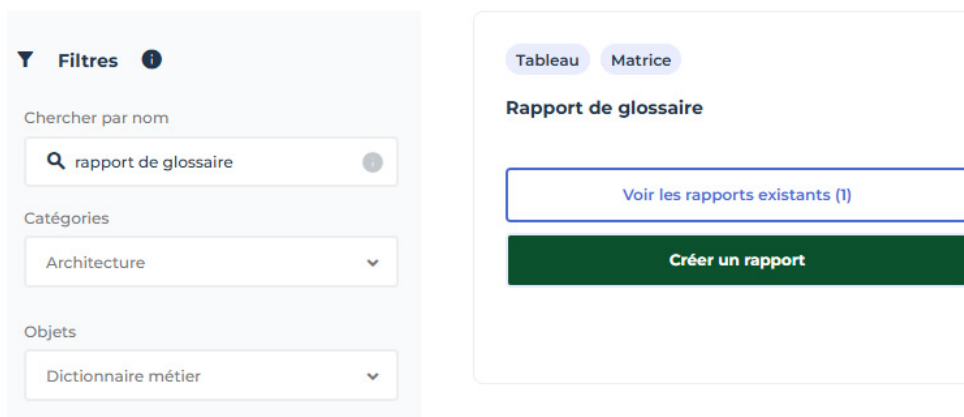
GÉNÉRER UN GLOSSAIRE

Hopex Data Governance et **Hopex Data Architecture** proposent un rapport de glossaire prêt à l'emploi pour construire automatiquement le glossaire métier de termes issus d'un ensemble de dictionnaires métier ou, le cas échéant, de bibliothèques. Pour chaque terme, le glossaire affiche une liste des définitions associées avec leur texte, synonyme et liste de composants.

Lancer un rapport de glossaire

Pour lancer un rapport de glossaire :

1. Cliquez sur le menu de navigation **Rapports**.
2. Dans la zone d'édition, cliquez sur le bouton **+ Créer un rapport**.
3. Recherchez le modèle de rapport "Rapport de glossaire".
Celui-ci apparaît dans la zone d'édition.
4. Passez la souris sur le modèle "Rapport de glossaire" et cliquez sur **Créer un rapport**.



L'assistant de création de rapport de glossaire apparaît.

5. Cliquez sur **Relier**.
6. Sélectionnez le dictionnaire métier source.
 Vous pouvez en sélectionner plusieurs.
7. Cliquez sur **Prévisualiser** puis sur **Continuer**.
8. Nommez le rapport et validez.
Le rapport créé apparaît.

Pour plus de détails voir [Rapport de glossaire](#).

Utiliser le glossaire dans un contexte multilingue

Pour plus de détails, voir "Utiliser HOPEX dans un contexte multilingue" dans le guide Hopex Common Features.

DÉFINIR LES INFORMATIONS MÉTIER



Hopex Data Governance et **Hopex Data Architecture** permettent de décrire l'architecture des informations métier de votre entreprise selon une démarche dont les différentes étapes sont décrites dans ce chapitre.

- ✓ Synthèse des objets utilisés
- ✓ Présentation des diagrammes de modélisation des concepts
- ✓ Dictionnaire métier
- ✓ Carte des domaines de concepts
- ✓ Domaine de concepts
- ✓ Concept
- ✓ Composants d'un concept
- ✓ Propriétés de concept
- ✓ Héritages d'un concept
- ✓ Le diagramme de structure de concept
- ✓ Individus
- ✓ Etats d'un concept ou d'un individu
- ✓ Concept type
- ✓ Vue de concept

SYNTHÈSE DES OBJETS UTILISÉS

Avec **Hopex Data Governance** et **Hopex Data Architecture**, vous pouvez construire un dictionnaire métier qui décrit et définit les éléments de votre vocabulaire métier.

Le composant de base d'un dictionnaire métier est le **Concept**.



Un concept représente la détermination de ce qu'est un être, une chose ou un mot, par ses propriétés et caractéristiques essentielles ou ses qualités propres.

Le mot qui est associé à un **Concept** et qui dépend de la langue est un **Terme**.



Un terme est un mot ou groupe de mots considéré dans sa valeur de désignation, en particulier dans un vocabulaire spécialisé.

Concept et terme

Un terme est spécifique à une langue et il n'est pas traduisible.

Dans des langues différentes, un même terme peut représenter des concepts différents.

Par exemple : le terme "car" en anglais représente une voiture (véhicule léger), alors que le même terme en français représente un véhicule de transport collectif.

Dans une même langue, un même terme peut représenter plusieurs concepts et le sens qui est donné à ce terme dépend de son contexte d'utilisation.

Par exemple, le mot "ring" représente en anglais une sonnerie de téléphone et une bague.

Donc, pour une même langue, un objet de type **Terme** peut être relié à plusieurs concepts. Chacun des concepts donne une définition spécifique de ce terme, dans son Dictionnaire métier.

Par conséquent, avec **Hopex**, un concept porte le nom du terme qui lui est associé dans la langue choisie par l'utilisateur. Pour changer le nom d'un concept, dans une langue donnée, il faut changer le nom du terme associé.

Pour plus de détails voir [Renommer un concept](#).

Voir aussi : [Rechercher des termes dans le glossaire métier](#)

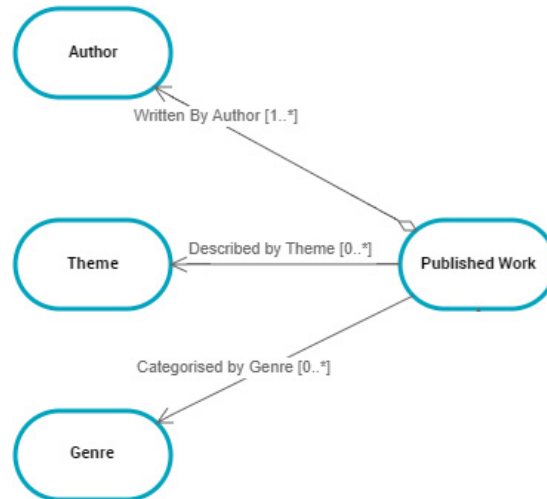
Les liens entre les concepts

Pour définir la sémantique d'un concept, vous pouvez tracer plusieurs types de liens entre les concepts : des liens de définition ou des liens dépendance.

Les liens de définition

Les liens de définition permettent de caractériser un concept.

Par exemple, un ouvrage ("Published Work") est défini par sa catégorie d'ouvrage (oeuvre littéraire ou musicale), son auteur, son thème.



Un lien de définition est décrit par un **Composant de concept** qui peut, éventuellement, être associé à un terme.

 Un composant de concept permet de représenter une relation de dépendance entre deux concepts. Cette relation est orientée.

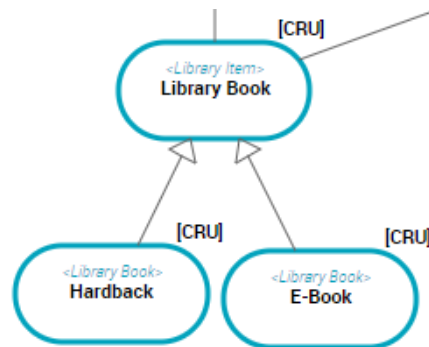
 Pour plus de détails, voir [Composants d'un concept](#).

Les liens de dépendance

Certains concepts métier sont des déclinaisons d'autres concepts : ils héritent des mêmes composants de concept.

Par exemple, le concept de Livre ("Library Book") est décliné en livre cartonné ou numérique. Ces deux types

d'ouvrage héritent des liens spécifiés au niveau du concept "Library Book".



Cette relation est décrite par une **Variation**.

 Une variation décrit comment un concept peut être varié sous une autre forme. La variante est un objet quasi-similaire à l'objet varié mais avec des propriétés ou des relations qui peuvent différer.

➡ Pour plus de détails sur les variations, voir le guide **Hopex Common Features**, chapitre "Manipuler les objets du référentiel", "Les variations d'objets".

Il est également possible de créer une **Variation** entre deux **Composants de concept**.

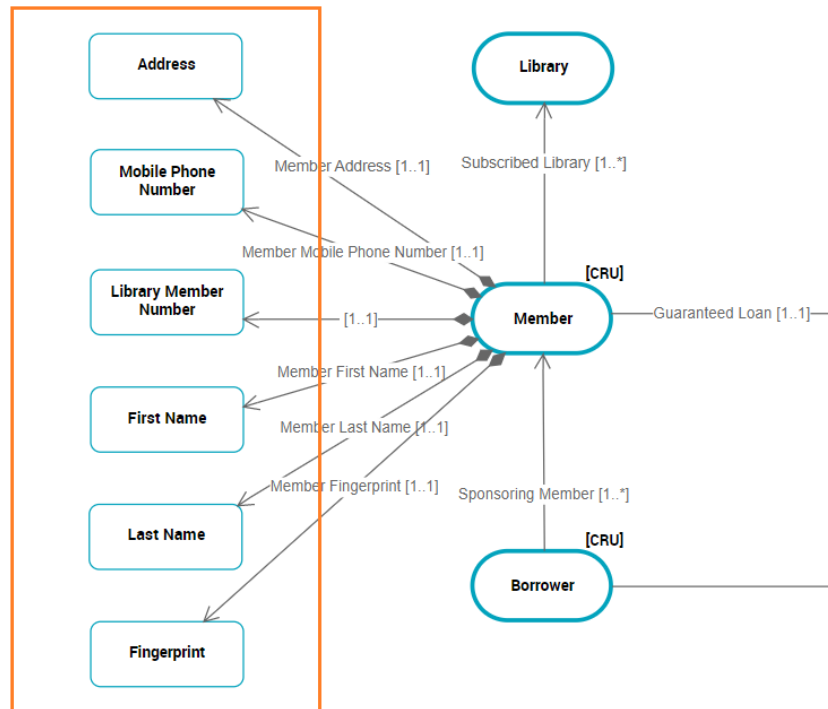
Par exemple, le "Souscripteur" est aussi un "Adhérent".

➡ Pour plus de détails, voir [Héritages d'un concept](#).

Propriétés de concept

Afin de décrire les caractéristiques attachées à un concept, vous pouvez lier au concept des Propriétés de concept.

Par exemple, une personne (ci-dessous, "Member") est associée à une adresse postale obligatoire et unique, un nom, prénom, numéro de téléphone, etc.



Le lien entre un concept et une propriété de concept est décrit par une **Sous-propriété** qui peut, éventuellement, être associée à un terme.

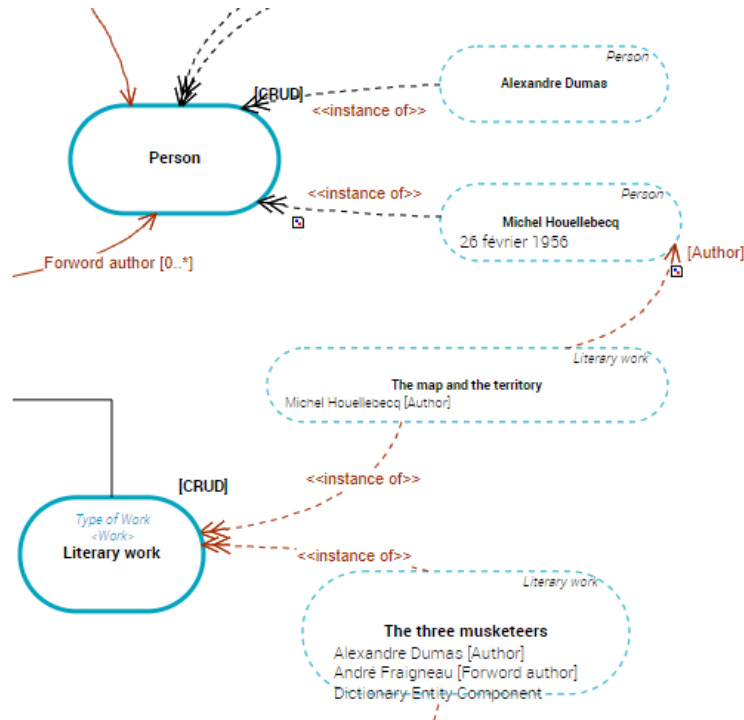
➡ Pour plus de détails, voir [Propriétés de concept](#).

Instances de concept : les individus

Afin de valider le modèle sémantique construit à partir des concepts, vous pouvez introduire des instances de concept, c'est à dire des objets réels.

Vous pouvez ainsi construire votre modèle sémantique selon deux approches : soit en partant des objets réels pour en déduire les concepts, soit en partant des concepts pour introduire ensuite les objets réels.

Par exemple, "Michel Houellebecq" est une instance de "Personne" et "La carte et le territoire" est une instance d'ouvrage.



Une instance de concept est un **individu**.

 Un individu représente l'instance d'un concept.

La relation entre un concept et ses instances est décrite par une **Classification d'individu**.

 Une classification d'individu permet de relier un individu au concept qui le caractérise.

Vous pouvez relier deux individus par une relation de type **Composant d'entité du dictionnaire**.

 Un composant d'entité permet de relier un individu à un élément de dictionnaire.

Il devient alors possible de préciser que "Asimov" est l'auteur de l'ouvrage "Les Robots".

 Il n'y a pas de possibilité de décrire des variations entre des individus ou entre des classification d'individus.

 Pour plus de détails, voir [Individus](#).

Le cycle de vie d'un concept ou d'un individu

Afin de prendre en compte l'évolution dans le temps des concepts métier, vous disposez de deux concepts particuliers :

- Le **Concept d'état** qui permet d'identifier une évolution temporelle d'un concept,



Un concept d'état est une situation au cours de la vie d'un concept durant laquelle il satisfait à certaines conditions, exerce une certaine activité ou attend un événement de concept. Un concept d'état représente un intervalle de temps dont les bornes sont deux événements de concept. Un état de concept est une phase par laquelle passe le concept au cours de son cycle de vie.

- Le **Concept événement** qui représente un fait marquant qui modifie l'état d'un ou de plusieurs concepts.



Un concept événement représente un fait se produisant durant la vie d'un concept, par exemple un changement de saison. Un concept événement permet de marquer l'impact sur un concept d'un phénomène interne ou externe au concept. On peut distinguer les événements de début de concept, les événements de fin de concept et les événements intermédiaires de concept.

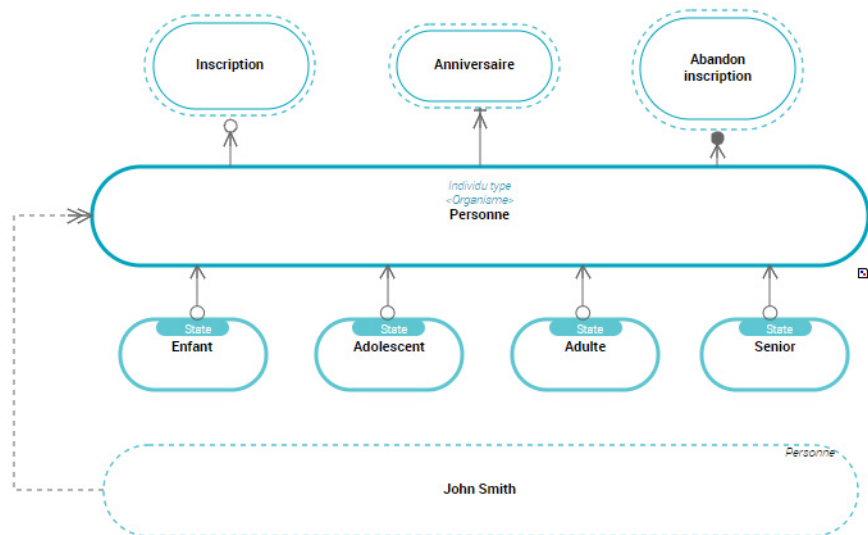
Les **Concepts d'état** et les **Concepts événement** sont des concepts à part entière qui peuvent être décrits comme n'importe quel concept.

Le cycle de vie d'un concept

Un même concept métier peut prendre plusieurs états.

Par exemple, un même Abonné peut passer d'un état de "Enfant" à l'état "Adolescent" puis à l'état "Adulte" et enfin "Senior".

Le passage d'un état à un autre peut être lié à un événement, un "Anniversaire", par exemple.



La relation entre un concept et son **Concept d'état** est décrite par un **Etat du dictionnaire de**.

 Un état du dictionnaire permet de relier un concept à un état de concept et de spécifier la nature de l'état.

La relation entre un concept et son **Concept événement** est décrite par :

- un **Événement de début**,
- un **Événement de fin**,
- ou un **Événement intermédiaire**.

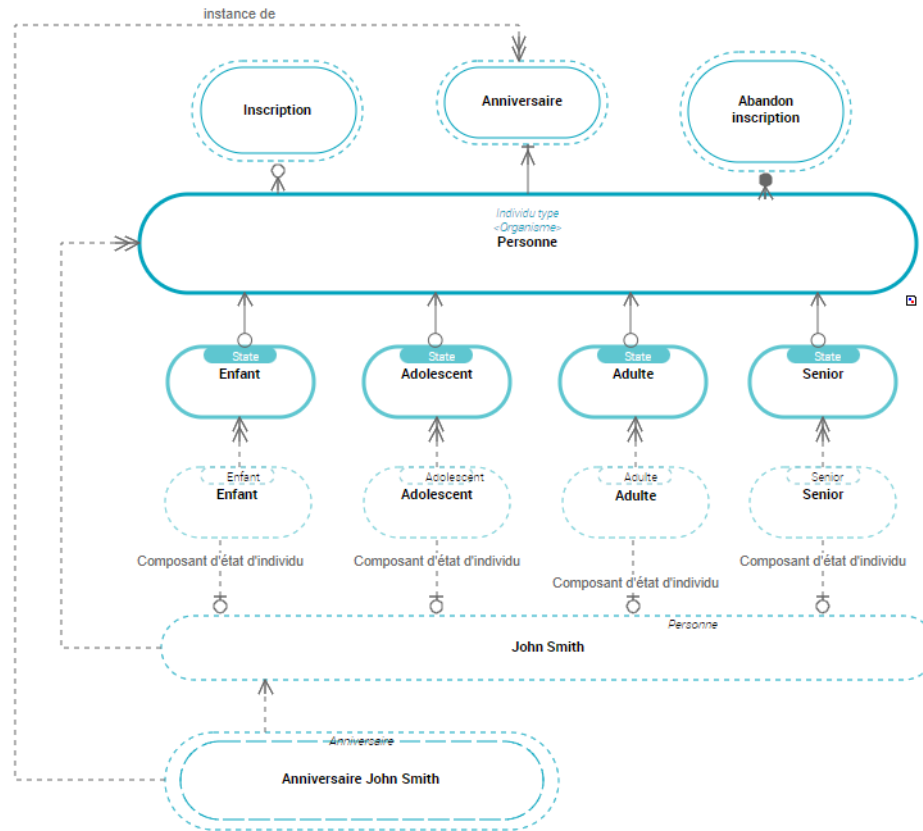
➡ Pour plus de détails, voir [Etats d'un concept ou d'un individu](#).

Le cycle de vie d'un individu

➡ Pour plus de détails, voir [Décrire les états et les événements d'un individu](#).

Si un concept est associé à des états et à des événements, les instances de ce concept peuvent également être associées à des événements et des états.

Par exemple, "John Smith" est une "Personne" qui peut passer d'un état à autre le jour de son anniversaire.



Pour représenter la notion d'état d'individu, **Hopex Data Architecture** propose l'**Etat d'individu**.

Un état d'individu est une instance d'un état du concept auquel l'individu est relié. Il représente un état de l'individu au cours de son cycle de vie.

La relation entre un individu et son **Etat d'individu** est décrite par un **Composant d'état d'individu**.

Un composant d'état d'individu permet de relier un individu à un état d'individu.

Par ailleurs, le passage d'un état d'individu à un autre peut être conditionné par un **Événement d'individu**.

Un événement d'individu représente un fait se produisant durant la vie de l'individu. C'est une instanciation d'un concept événement du concept auquel l'individu est relié.

La relation entre un individu et son **Événement d'individu** est décrite par **Composant d'entité**.

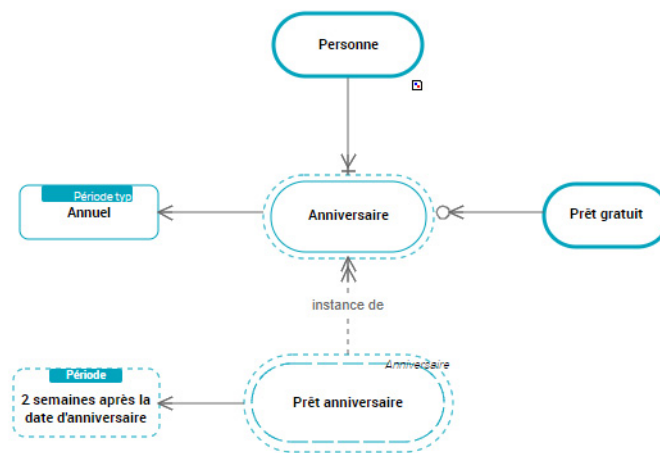


Un composant d'entité permet de relier un individu à un élément de dictionnaire.

Les périodes

Les **Périodes** permettent d'apporter des précisions temporelles sur les événements.

Par exemple, un prêt gratuit peut être proposé aux abonnés à chaque anniversaire. Ce prêt annuel est valide pendant une durée de deux semaines.



Une **période type** est reliée à un **Concept événement**.



Un concept événement représente un fait se produisant durant la vie d'un concept, par exemple un changement de saison. Un concept événement permet de marquer l'impact sur un concept d'un phénomène interne ou externe au concept. On peut distinguer les événements de début de concept, les événements de fin de concept et les événements intermédiaires de concept.

La **période** est reliée à un **Événement d'individu**.



Un événement d'individu représente un fait se produisant durant la vie de l'individu. C'est une instanciation d'un concept événement du concept auquel l'individu est relié.

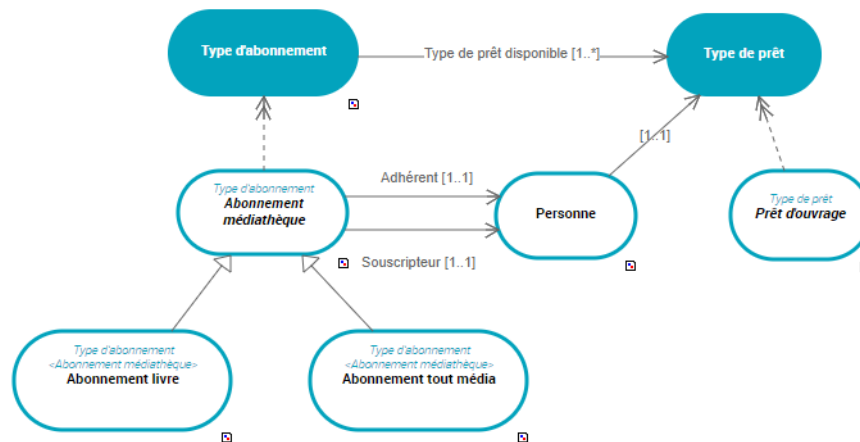


Pour plus de détails, voir [Utiliser les périodes](#).

La classification des concepts et la notion de type de concept

Un concept type permet de classer les concepts. Les relations entre les concepts type sont représentées par des composants de concept type.

Par exemple, les "Abonnement" peuvent être classés par "Type d'abonnement". Un "Type d'abonnement" étant caractérisé par un "Type de prêt".



Hopex Data Governance et **Hopex Data Architecture** fournit les outils pour établir les relations suivantes :

- la relation entre deux **Concept type** est décrite par un **Composant de concept type**.

Par exemple, un "Type d'abonnement" est caractérisé par un "Type de prêt disponible".

 *Un composant de concept type permet de spécifier la relation entre deux concepts type.*

- la relation d'appartenance d'un concept à un **Concept type** est décrite par une **Classification de concept**.

Par exemple, tous les "Abonnements" doivent correspondre à un "Type d'abonnement".

 *Une classification de concept permet de relier un concept au concept type qui le caractérise.*

- la relation entre un concept et un **Concept type** qui permet de caractériser le concept est décrite par un **Super-composant de concept**.

Par exemple, chaque "Personne" adhérentes pourrait être caractérisée par un "Type de prêt".

 *Un super-composant de concept permet de relier un concept à un concept type pour caractériser une propriété du concept.*

La vue de concept

Une vue de concept permet de représenter le périmètre sémantique couvert par un objet métier. Une vue de concept est construite à partir d'une sélection de plusieurs concepts reliés dans le contexte spécifique de la vue.

A partir d'un concept de départ, lié à l'objet métier que vous souhaitez décrire, vous parcourez les liens sémantiques qui le définissent. Vous identifiez ainsi plusieurs concepts qui définissent l'objet décrit dans un contexte précis.

☛ Vous pouvez construire différentes vues pour un même objet métier.

☛ Pour plus de détails, voir [Vue de concept](#).

La réalisation des éléments de dictionnaire

A travers les **Réalisations**, vous pouvez assurer la cohérence entre les objets qui constituent votre référentiel organisationnel et technique, d'une part, et les concepts métier qui constituent votre dictionnaire, d'autre part.

📖 Une réalisation de concept relie un objet technique ou organisationnel du référentiel à un élément de dictionnaire.

Pour plus de détails sur les réalisations, voir le chapitre [Relier les concepts métier à l'architecture logique et physique](#).

Pour plus de détails sur la génération du dictionnaire, voir [Rapport de glossaire](#).

PRÉSENTATION DES DIAGRAMMES DE MODÉLISATION DES CONCEPTS

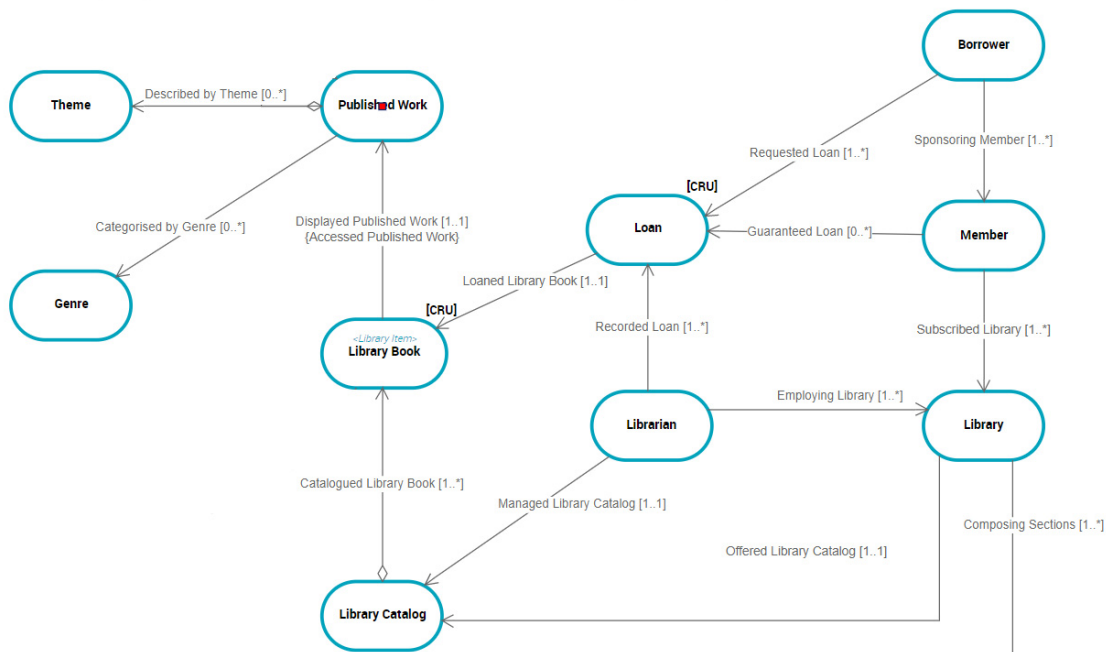
Pour la définition des données métier, **Hopex Data Governance** et **Hopex Data Architecture** fournissent différents types de diagramme.

Le diagramme du domaine de concepts

Un domaine de concepts fournit une vue partielle des modèles ontologiques de l'information métier. Il est décrit par un diagramme de concepts qui présente les concepts, leurs composants, les sur-types et leurs liens.

Le sens des liens fournit un mécanisme naturel de lecture et de déduction du périmètre de "l'objet métier".

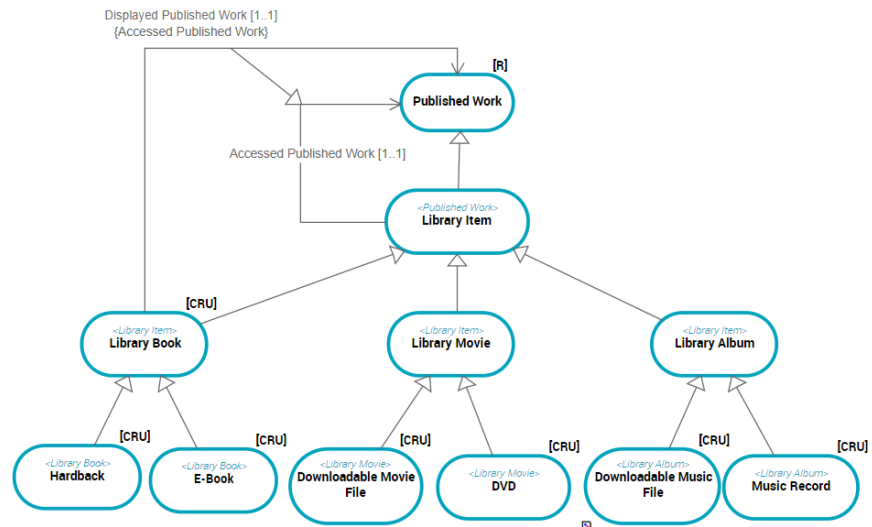
Le diagramme du domaine de concepts suivant présente une vue partielle du Dictionnaire métier "Bibliothèque".



➤ Pour plus de détails, voir [Construire un diagramme de concepts](#).

Le diagramme de structure de concept

Le contenu des objets métier peut être représenté dans un "Diagramme de structure de concept", qu'il est possible d'initialiser à partir des éléments du diagramme de concept.



➡ Pour plus de détails, voir [Le diagramme de structure de concept](#).

Le diagramme de structure de concept type

Les types de concept peuvent être représentés dans un "Diagramme de structure de type concept", qu'il est possible d'initialiser à partir des éléments du graphe de concept.

➡ Pour plus de détails, voir [Le diagramme de structure de concept type](#).

Le diagramme de structure de concept d'état

Les concepts d'état peuvent être représentés dans un "Diagramme de structure de concept d'état", qu'il est possible d'initialiser à partir des éléments du graphe de concept.

➡ Pour plus de détails, voir [Le diagramme de structure de concept d'état](#).

Le diagramme de structure d'un individu

Le diagramme de structure d'individu décrit la structure interne de l'instance de concept et les liens entre tous les composants. Il est possible d'initialiser ce diagramme depuis les éléments du graphe de concept.

➡ Pour plus de détails, voir [Le diagramme de structure d'individu](#).

Le diagramme de structure de cycle de vie de concept

Le diagramme de structure de cycle de vie de concept permet de décrire l'enchaînement des concepts d'état qui s'opèrent au cours du cycle de vie d'un concept. Chaque concept d'état, que l'on peut considérer comme un point dans le temps, est suivi d'autres concepts d'état.

Le passage d'un état à un autre est modélisé par une transition.

➡ Pour plus de détails, voir [Le diagramme de structure de cycle de vie de concept](#).

DICTIONNAIRE MÉTIER

Un dictionnaire métier rassemble et structure un ensemble de concepts qui expriment les connaissances d'un domaine particulier.

Exemple de dictionnaire : ontologie médicale

Vous pouvez décomposer un dictionnaire métier en domaines de concepts.

Exemples de domaine : psychologie, pédiatrie, etc.

Voir [Domaine de concepts](#).

Les dictionnaires métier peuvent être créés avec le profil **Concepteur de données**.

Les éléments d'un dictionnaire métier

Un dictionnaire métier vous permet de décrire tous les éléments qui définissent votre architecture d'information métier :

- Les concepts



Un concept représente la détermination de ce qu'est un être, une chose ou un mot, par ses propriétés et caractéristiques essentielles ou ses qualités propres.



Pour plus de détails, voir [Concept](#)

- Les termes



Un terme est un mot ou groupe de mots considéré dans sa valeur de désignation, en particulier dans un vocabulaire spécialisé.



Pour plus de détails, voir [Rechercher des termes dans le glossaire métier](#)

- Les variations de concept



Une variation décrit comment un concept peut être varié sous une autre forme. La variante est un objet quasi-similaire à l'objet varié mais avec des propriétés ou des relations qui peuvent différer.



Pour plus de détails, voir [Composants d'un concept](#)

- Les concepts type



Un concept type permet de classifier les concepts. Les relations entre les concepts type sont représentées par des composants de concept type.



Pour plus de détails, voir [Concept type](#)

- Les concepts d'état



Un concept d'état est une situation au cours de la vie d'un concept durant laquelle il satisfait à certaines conditions, exerce une certaine activité ou attend un événement de concept. Un concept d'état représente un intervalle de temps dont les bornes sont deux

événements de concept. Un état de concept est une phase par laquelle passe le concept au cours de son cycle de vie.

☛ Pour plus de détails, voir [Etats d'un concept ou d'un individu](#)

- Les concepts événement

📖 Un concept événement représente un fait se produisant durant la vie d'un concept, par exemple un changement de saison. Un concept événement permet de marquer l'impact sur un concept d'un phénomène interne ou externe au concept. On peut distinguer les événements de début de concept, les événements de fin de concept et les événements intermédiaires de concept.

☛ Pour plus de détails, voir [Décrire les concepts événement](#)

- Les individus

📖 Un individu représente l'instance d'un concept.

☛ Pour plus de détails, voir [Individus](#)

- Les états d'individu

📖 Un état d'individu est une instance d'un état du concept auquel l'individu est relié. Il représente un état de l'individu au cours de son cycle de vie.

☛ Pour plus de détails, voir [Etats d'un concept ou d'un individu](#)

Un dictionnaire métier peut être complètement, ou partiellement, décrit par un diagramme de concepts.

☛ Pour plus de détails sur les graphes de concepts, voir [Présentation des diagrammes de modélisation des concepts](#)

Accéder aux éléments d'un dictionnaire métier

Pour accéder aux éléments d'un dictionnaire métier dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Glossaire > Dictionnaires métier**.
2. Dans la zone d'édition, cliquez sur le dictionnaire métier qui vous intéresse.

Les propriétés du dictionnaire apparaissent. Les concepts et termes sont visibles dans les pages dédiées.

☛ Les concepts portent le nom des termes qui leur sont associés, dans la langue des données. Pour plus de détails, voir [Utiliser le glossaire dans un contexte multilingue](#).

De la même façon vous pouvez accéder aux propriétés des concepts.

Dans les propriétés d'un concept, les termes et synonymes sont accessibles dans toutes les langues disponibles pour votre environnement **Hopex**.

☛ Le nombre de langues proposées, à partir des dossiers, dépend de votre environnement **Hopex**. Pour configurer la liste des langues, voir le guide **Hopex Power Supervisor**, chapitre "Gérer les options", "Gérer les langues", "Installer des langues supplémentaires".

☛ La liste des éléments d'un dictionnaire métier est également accessible dans la fenêtre de propriétés du dictionnaire, sous la page **Caractéristiques**, section **Information métier**.

Importer des données métier

Vous pouvez importer dans votre référentiel des données métier existantes à l'aide d'un fichier Excel. Voir [Importer des données métier à partir d'un fichier Excel](#).

Dictionnaire métier de travail

Lors de l'initialisation de données métier, un dictionnaire métier de travail est créé par défaut pour y ranger les données créées et définir leur détenteur.

Voir [Initialiser un dictionnaire métier à partir de données logiques ou physiques](#).

Créer un dictionnaire métier

Pour créer un dictionnaire métier :

1. Cliquez sur le menu de navigation **Glossaire > Dictionnaires métier**.
 2. Passez la souris sur le dossier **Dictionnaires métier** et cliquez sur le bouton  **Nouveau > Dictionnaire métier**.
 3. Saisissez le nom du dictionnaire et cliquez sur **OK**.
-

Initialiser un dictionnaire métier à partir de données logiques ou physiques

Initialiser un dictionnaire métier consiste à créer dans un dictionnaire métier les concepts qui correspondent à des données logiques ou physiques, et à établir ainsi un lien de réalisation entre les concepts métier et les données logiques ou physiques en question.

Un rapport de réalisation permet de visualiser ces liens de réalisation.

Il est possible d'initialiser un dictionnaire métier de trois façons :

- à partir d'un dictionnaire de données logiques (paquetage de données)
- à partir d'un dictionnaire de données physiques (base de données)
- à partir d'un inventaire de métadonnées (catalogue de données)
- lors de la création de données logiques ou physiques : une option d'initialisation automatique vous permet, lorsque vous créez des données logiques ou physiques, de créer automatiquement les concepts correspondants dans le dictionnaire de travail.

Initialiser un dictionnaire métier à partir de données logiques

Pour Initialiser un dictionnaire à partir de données logiques :

1. Cliquez sur le menu de navigation **Outils > Initialiser un dictionnaire métier**.
2. Cliquez sur la tuile **Données logiques**.
Un assistant apparaît.
3. Sous **Options**, indiquez si les attributs deviennent des éléments d'information ou des composants d'élément d'information.
4. Sous **Sélection de la source**, sélectionnez le paquetage source.

5. Une fois le paquetage sélectionné, cliquez sur **Suivant**.
L'assistant affiche le nom du dictionnaire métier correspondant.
6. Cliquez sur **OK**.
Le dictionnaire métier est créé. Il apparaît dans la liste des dictionnaires métier de travail du référentiel.

Correspondances des données logiques et des objets métier

Données logiques	Objets métier
Classe	Concept
Partie	Composant de concept
Attribut	Élément d'information Composant d'élément d'information

Initialiser un dictionnaire métier à partir de données physiques

Pour Initialiser un dictionnaire à partir de données physiques :

1. Cliquez sur le menu de navigation **Outils > Initialiser un dictionnaire métier**.
2. Cliquez sur la tuile **Données physiques**.
3. Dans l'assistant qui apparaît, sous **Sélection de la source**, sélectionnez la base de données source.
4. Une fois la base de données sélectionnée, cliquez sur **Suivant**.
L'assistant affiche le nom du dictionnaire métier correspondant.
5. Cliquez sur **OK**.
Le dictionnaire métier est créé. Il apparaît dans la liste des dictionnaires métier de travail du référentiel.

Correspondances des données physiques et des objets métier

Données physiques	Objets métier
Table	Concept
Colonne	Élément d'information

Initialiser un dictionnaire métier à partir de métadonnées physiques

Pour Initialiser un dictionnaire à partir de métadonnées physiques :

1. Cliquez sur le menu de navigation **Outils > Initialiser un dictionnaire métier**.
2. Cliquez sur la tuile **Métadonnées**.
3. Dans l'assistant qui apparaît, sous **Sélection de la source**, sélectionnez le dépôt de données déployé source.
4. Cliquez sur **Suivant**.
L'assistant affiche le nom du dictionnaire métier correspondant.

5. Cliquez sur **OK**.

Le dictionnaire métier est créé. Il apparaît dans la liste des dictionnaires métier de travail du référentiel.

Correspondances des métadonnées physiques et des objets métier

Métadonnées physiques	Objets métier
Métadonnée physique	Concept
Champ de métadonnée	Élément d'information

Initialiser un dictionnaire métier lors de la création de données logiques ou physiques

Par défaut, lorsque vous créez des données logiques ou physiques, les concepts métier correspondants sont automatiquement créés et associés à ces données.

Ces concepts apparaissent dans un dictionnaire métier de travail qui porte le même nom que le dictionnaire de donnée détenteur des données logiques ou physiques dont sont issus les concepts.

Cette initialisation automatique est définie par une option. Vous pouvez modifier l'option pour ne créer aucun concept par défaut ou pour les créer/les réutiliser dans d'autres dictionnaires.

Modifier l'option d'initialisation

Pour modifier l'option d'initialisation :

1. Cliquez sur l'icône du profil utilisateur puis sur **Paramètres > Options**. La fenêtre des options apparaît.
2. Dans la partie gauche de la fenêtre, cliquez sur **HOPEX Solutions > Data Management > Data Governance**.
3. Dans la partie droite, sous le champ **Initialisation de dictionnaire métier**, sélectionnez la valeur d'initialisation automatique voulue :
 - **Jamais** : désactive l'initialisation automatique du dictionnaire métier.
 - **Dictionnaire métier de travail** : active l'initialisation automatique du dictionnaire métier. Les concepts sont créés automatiquement dans le dictionnaire métier de travail.
 - **Tous les dictionnaires métier** : active l'initialisation automatique du dictionnaire métier. Les concepts peuvent être créés ou réutilisés sur l'ensemble des dictionnaires métier.

Afficher le graphe de réalisation

Vous pouvez utiliser le graphe de réalisation pour visualiser par quels éléments d'architecture sont implémentés des éléments de dictionnaire.

Pour accéder au rapport :

1. Ouvrez les propriétés de l'objet concerné.
2. Cliquez sur **Rapports**.

3. Sélectionnez le rapport **Data Management** > **Graphe de réalisation**.

Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètre	Type du paramètre	Contrainte
Liste d'objets	Acteur, Application, Bibliothèque, Capacité, Classe, Concept, Concept d'état, Concept événement, Concept type, Contenu, Contrat d'échange, Cycle de vie de concept, Echange, Entité (MD) Fonctionnalité Métier Processus applicatif, Processus fonctionne, Processus métier, Processus organisationnel, Service applicatif Vue de données Vue de concept	Un objet obligatoire

Exemple de rapport

L'exemple ci-dessous permet de visualiser les objets qui implémentent le concept "Purchase".



CARTE DES DOMAINES DE CONCEPTS

Une carte des domaines de concepts est un outil d'urbanisation des informations métier. Elle permet de représenter les domaines de concepts d'un dictionnaire métier et leurs liens de dépendance.



Un domaine de concepts est un sous-ensemble d'éléments d'un dictionnaire métier qui permet de réduire le champ d'une étude.

Les liens de dépendance entre domaines sont automatiquement déduits à partir des objets utilisés dans chacun des domaines de la carte, vous n'avez pas à les créer.

Pour plus de détails sur les liens entre les informations métier, voir [Synthèse des liens entre les objets](#).

Les cartes de domaines de concepts peuvent être créées avec le profil **Concepteur de données**.

Créer une carte des domaines de concepts

Pour créer la carte des domaines de concepts d'un dictionnaire métier :

1. Faites un clic droit sur le dictionnaire métier et sélectionnez **Nouveau > Carte des domaines de concepts**.
La carte apparaît sous le dictionnaire métier.

Pour créer le diagramme de la carte des domaines de concepts :

1. Cliquez sur l'icône de la carte des domaines de concepts et sélectionnez **Nouveau > Diagramme**.
2. Sélectionnez **Carte des domaines de concepts** et cliquez sur **OK**.
Le diagramme apparaît dans la zone d'édition.

Les composants d'une carte des domaines de concepts

Dans une carte des domaines de concepts vous pouvez ajouter des composants internes et externes.

Les composants internes sont les domaines de concepts qui font partie du périmètre de la carte (qu'ils appartiennent ou non au dictionnaire métier détenteur).

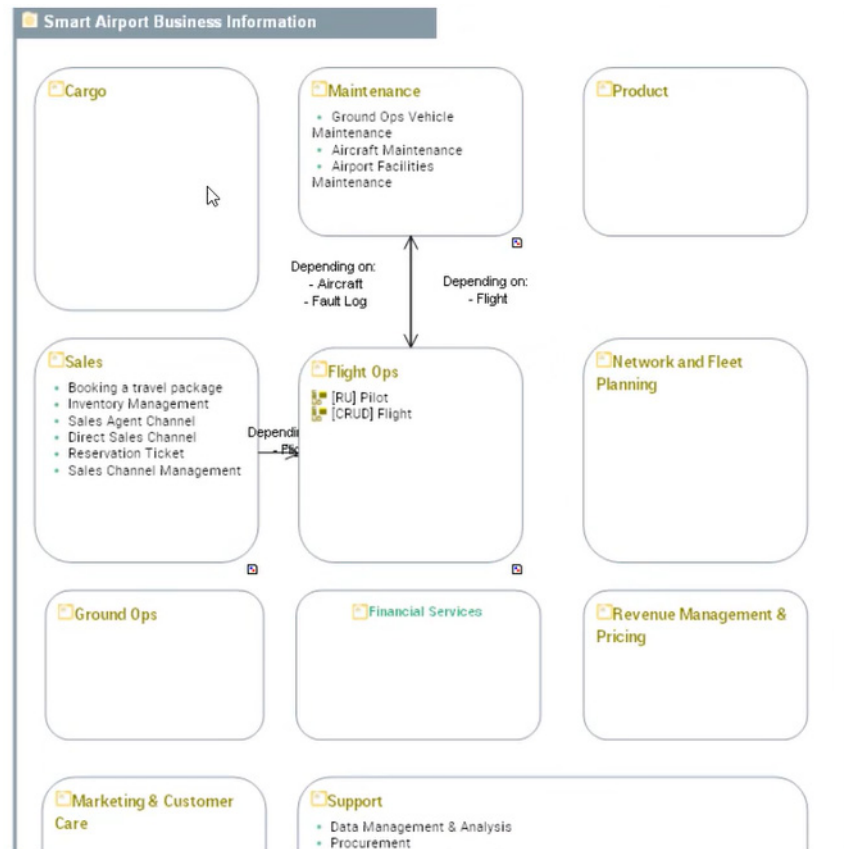
Les composants externes sont ceux qui sont utilisés dans la carte mais qui ne font pas partie du périmètre étudié.

Pour ajouter un composant à une carte des domaines de concepts :

1. Ouvrez les propriétés de la carte en question.
2. Cliquez sur la page **Domaines de données**.
3. Sélectionnez l'onglet **Domaines internes** ou **Domaines externes** selon le type de composant à ajouter et cliquez sur **Nouveau**.
Un assistant apparaît. Vous pouvez créer un domaine de concepts ou ajouter un domaine de concepts existant.

Exemple de domaines de concepts

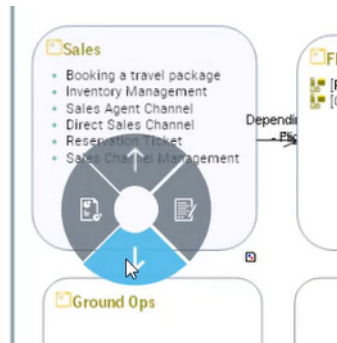
Ci-dessous la carte des domaines de concepts d'un aéroport. Des liens entre les domaines de concepts indiquent les dépendances qui existent entre des concepts de ces domaines.



Vous pouvez plonger sur un domaine de concepts pour accéder au diagramme qui décrit le domaine en question, le cas échéant.

Vous pouvez plonger sur un domaine :

- via le menu circulaire lorsque vous êtes en mode aperçu du diagramme.



- ou via le **Menu contextuel** du domaine lorsque vous êtes en mode édition.

Rapports disponibles sur une carte des domaines de concepts

Dans les propriétés d'une carte des domaines de concepts vous disposez de rapports qui vous permettent de visualiser :

- la hiérarchie des domaines d'une carte, et si ces domaines utilisent des données sensibles ou de référence. Pour plus de détails voir [Carte de domaines de données](#).
- les dépendances entre les domaines d'une carte. Voir [Dépendances entre domaines de données](#).
- les éléments d'architecture qui implémentent les informations de la carte. Voir [Afficher le graphe de réalisation](#).
- l'utilisation des informations d'une carte. Voir [Utilisation des informations d'une carte d'informations](#).

DOMAINE DE CONCEPTS

Un domaine de concepts est un sous-ensemble d'éléments d'un dictionnaire métier qui permet de réduire le champ d'une étude. Il est décrit par un diagramme de concepts.

Les domaines de concepts peuvent être créés avec le profil **Concepteur de données**.

Créer un domaine de concepts

Vous pouvez créer un domaine de concepts à partir d'une carte de domaines de concepts.

Pour créer un domaine de concepts :

1. Cliquez sur le menu de navigation **Glossaire > Domaines de concepts**.
2. Dans la zone d'édition, passez la souris sur la carte de domaine de concepts concernée et cliquez sur le bouton **Nouveau > Domaine de concepts**.

Créer le diagramme de structure d'un domaine de concepts

Le diagramme de structure définit les sous-domaines du domaine de concepts et leurs relations.

Pour créer le diagramme de structure d'un domaine de concepts :

1. Dans la zone d'édition, passez la souris sur le domaine de concepts et cliquez sur le bouton **Créer un diagramme**.
2. Sélectionnez le type de diagramme **Diagramme de structure de domaine de concepts**.
3. Cliquez sur **OK**.
Le diagramme de structure associé au domaine de concepts s'ouvre dans la fenêtre d'édition.

Construire un diagramme de concepts

Un diagramme de concepts est une représentation graphique des concepts qui sont utilisés dans le contexte d'un domaine de concepts, ainsi que des liens qui existent entre ces concepts.

Un domaine de concepts peut être décrit par plusieurs diagrammes de concepts.

Un objet conceptuel appartient au dictionnaire métier à partir duquel il a été créé mais il peut être utilisé/référencé par un domaine de concepts d'un dictionnaire métier différent.

Voir aussi [Domaine de concepts](#).

Créer un diagramme de concepts d'un domaine de concept

Pour créer le diagramme de concepts d'un domaine de concepts :

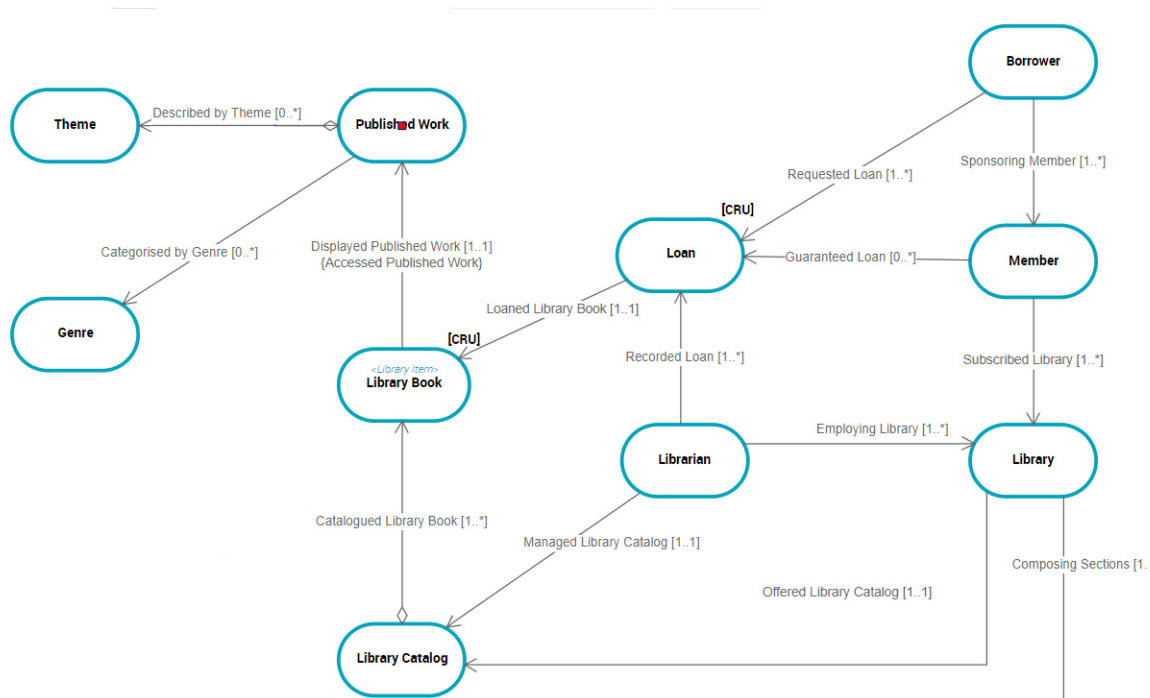
1. Dans la zone d'édition, passez la souris sur le domaine de concepts et cliquez sur le bouton **Créer un diagramme**.
2. Sélectionnez **Diagramme du domaine de concepts**.
3. Cliquez sur **OK**.

Le diagramme de concepts s'ouvre dans la fenêtre d'édition.

Les composants d'un diagramme de concepts

Un diagramme de concepts décrit l'architecture de l'information. Par défaut, vous ne visualisez dans le diagramme de concepts que des concepts, des variations et des individus.

Le diagramme de concepts suivant décrit partiellement le dictionnaire métier "Médiathèque".



Le diagramme est initialisé avec les objets qui appartiennent au domaine. Il peut également contenir des objets qui n'appartiennent pas au domaine.

Voir : [Les composants d'un domaine de concepts](#).

Activer la fenêtre de vue

La fenêtre **Vues et détails** présente une liste de vues (types d'objets à afficher) plus étendue.

Pour activer la fenêtre **Vues et détails** :

1. Dans un diagramme, cliquez sur  **Vues et détails**.
La liste des vues (types d'objets à afficher) apparaît.
2. Sélectionnez ou dé-sélectionnez les vues que vous voulez afficher ou non.

Les vues disponibles pour un domaine de concepts sont :

- Les concepts,
- Les concepts type,
- Les concepts d'état,
- Les concepts événement,
- les propriétés de concept
- Les individus,
- Les états d'individu,
- Les événements d'individu,
- Les vues de concept.



Une vue de concept permet de représenter le périmètre sémantique couvert par un objet métier. Une vue de concept est construite à partir d'une sélection de plusieurs concepts reliés dans le contexte spécifique de la vue.

➡ Pour plus de détails, voir [Vue de concept](#)

Ajouter un élément dans un diagramme de concepts

Pour ajouter, par exemple, un concept existant dans un domaine de concepts :

1. Dans la barre d'objets du diagramme de concept, cliquez sur le bouton  **Concept**.
2. Cliquez dans le diagramme.
Une fenêtre d'ajout d'un concept s'ouvre et vous demande de choisir le concept.
3. Sélectionnez le concept qui vous intéresse.
4. Cliquez sur **Ajouter**.
Le concept apparaît dans le diagramme.

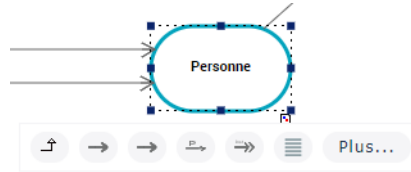
➡ Pour plus de détails sur la création de concept, voir [Concept](#)

Utiliser la barre d'insertion de l'objet

La barre d'insertion disponible sur chaque objet facilite la création d'objets en proposant une aide à la sélection des objets. Elle propose uniquement les objets que vous pouvez relier à l'objet courant.

Pour créer, par exemple, un concept à partir d'un concept du diagramme :

1. Cliquez sur le concept du diagramme qui vous intéresse.
Une barre qui contient les objets que vous pouvez insérer apparaît.



2. Sélectionnez le type d'objet désiré.

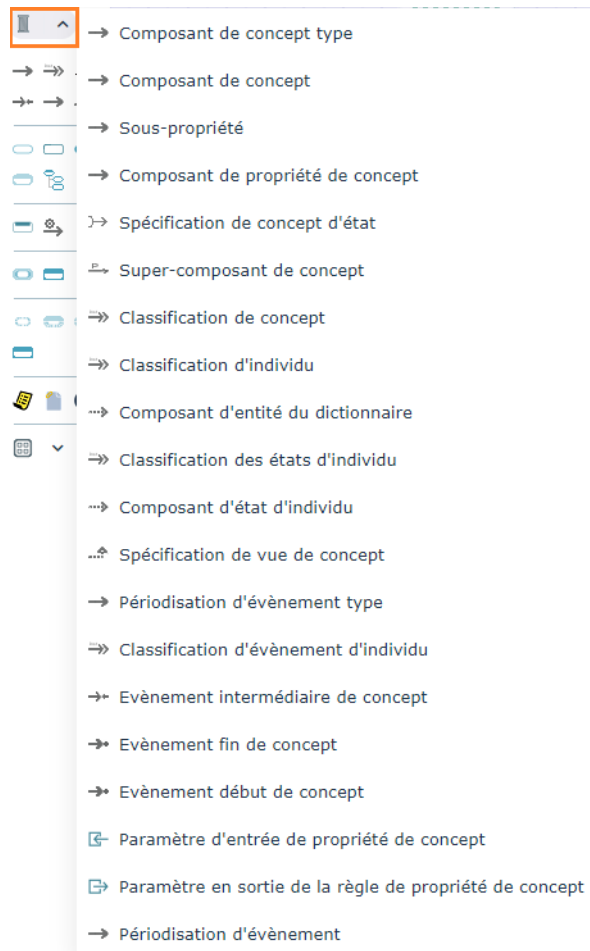
Par exemple : **Composant de concept**

3. Cliquez dans le graphe à l'endroit où vous souhaitez poser l'objet.
L'objet est créé, ainsi que son lien à l'objet de départ.

Synthèse des liens entre les objets

Dans chaque graphe de concept les relations entre les concepts, concepts type et individus de concept sont représentées par des liens.

Le sens des liens fournit un mécanisme naturel de lecture et de déduction du périmètre définissant "l'objet métier".



Détails des liens

Type de lien	Définition et Commentaire
Composant de concept type	Un composant de concept type permet de spécifier la relation entre deux concepts type.
Composant de concept	Un composant de concept permet de représenter une relation de dépendance entre deux concepts. Cette relation est orientée.
Etat du dictionnaire de	Un état du dictionnaire permet de relier un concept à un état de concept et de spécifier la nature de l'état. Avec la vue "Concept d'état»

Type de lien	Définition et Commentaire
Super-composant de concept	Un super-composant de concept permet de relier un concept à un concept type pour caractériser une propriété du concept.
Classification de concept	Une classification de concept permet de relier un concept au concept type qui le caractérise.
Classification d'individu	Une classification d'individu permet de relier un individu au concept qui le caractérise.
Composant d'entité du dictionnaire	Un composant d'entité permet de relier un individu à un élément de dictionnaire.
Classification d'état d'individu	Une classification d'état d'individu permet de relier un état d'individu au concept d'état qui le caractérise. Ce lien est proposé avec la vue "Etat d'individu".
Composant d'état d'individu	Un composant d'état d'individu permet de relier un individu à un état d'individu. Ce lien est proposé avec la vue "Etat d'individu".
Classification d'événement d'individu	Une classification d'événement d'individu permet de relier un individu au concept d'événement qui le caractérise. Ce lien est proposé avec la vue "Etat d'individu".
Événement intermédiaire de concept	Un concept événement représente un fait se produisant durant la vie d'un concept, par exemple un changement de saison. Un concept événement permet de marquer l'impact sur un concept d'un phénomène interne ou externe au concept. On peut distinguer les événements de début de concept, les événements de fin de concept et les événements intermédiaires de concept. Ces liens sont proposés avec la vue "Concept événement".
Événement fin de concept	
Événement début de concept	

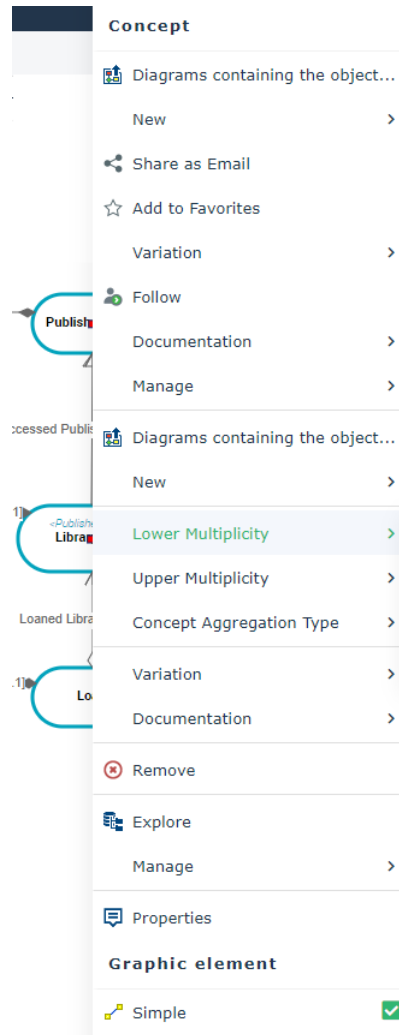
Accéder aux propriétés d'un lien dans un diagramme de concepts

Dans un diagramme de concepts, les liens sont orientés et donnent accès à la fois aux propriétés du lien et à l'objet cible du lien.

➡ Pour plus de détails sur la liste des liens disponibles dans un domaine de concepts, voir [Synthèse des liens entre les objets](#).

Le menu contextuel d'un lien de type **Composant de concept**, par exemple, présente :

- les commandes propres au type d'objet utilisé par le composant (le concept)
- les commandes relatives au composant lui-même
par exemple **la multiplicité**
- les commandes relatives au graphisme.



Pour accéder aux propriétés d'un lien de type "composant" :

Par exemple **Composant de concept**

1. Faites un clic droit que le lien pour ouvrir son menu contextuel.
2. Sélectionnez le lien puis cliquez sur **Propriétés**.
La fenêtre de propriétés du lien s'ouvre.

Dans la page **Caractéristiques** de la fenêtre de propriétés du lien, la dernière section **Composant** indique :

- Le **Nom** du lien qui correspond par défaut à l'élément de dictionnaire cible ou au terme associé au lien.
 Pour plus de détails sur l'association d'un terme à un lien, voir [Composants d'un concept](#).
- Le **Concept composé** ciblé par le lien.
- Le **Détenteur** qui est l'élément de dictionnaire à l'origine du lien.
- La **Multiplicité minimale** qui est le nombre d'éléments origine qui peuvent accéder au même éléments cible.
Par exemple, combien d' "Ouvrages" peuvent appartenir à une même "Catégorie d'ouvrage".
- La **Multiplicité maximale** qui est le nombre d'éléments cibles qui peuvent être liés à un même élément origine.
Par exemple, un "Ouvrages" ne peut appartenir qu'à une seule "Catégorie d'ouvrage".
- La case **Concept abstrait** qui précise le caractère concret ou abstrait d'un concept.
- Le **Type d'agrégation de concept** qui peut prendre les valeurs suivantes :
 - "Référencement" : pour signifier que le concept cible est seulement référencé par le lien,
 - "Embarqué" : pour signifier que le concept cible a son existence propre, mais fait partie intégrante du concept qui est à l'origine du lien
 - "Composite" : pour signifier que le concept cible est un composant du concept qui est à l'origine du lien, si le concept cible est détruit, le composite l'est également.
- La **Désignation** du lien et le champ **Texte de la définition** qui permettent d'associer un terme et une définition au lien.
 Pour plus de détails sur l'association d'un terme à un lien, voir [Composants d'un concept](#).

Pour plus de détails sur la définition des concepts, voir [Concept](#).

Aspect graphique des objets du diagramme

Pour modifier l'aspect graphique des objets du diagramme :

1. Sélectionnez l'objet en question dans le diagramme (ouvert en mode édition).
2. A droite du diagramme, cliquez sur la flèche **Ouvrir la page Aspect**.
Le panneau des propriétés apparaît. Sous l'onglet **Aspect**, des commandes vous permettent de personnaliser le texte et la forme de l'objet sélectionné.

Définir les composants d'un domaine de concepts

Vous pouvez mettre à jour vos dictionnaires métier à partir des *Domaines de concepts* et des éléments de dictionnaire qui existent déjà : *Terme*, *Concept*, *Concept d'état*, *Concept événement* ou *Vue de concept*.

Les composants d'un domaine de concepts

Un domaine de concepts inclut ou référence un ensemble de concepts ou de sous-domaines.

Vous pouvez visualiser les éléments qui appartiennent au domaine de concepts dans la fenêtre de propriétés du domaine en question, dans la page **Entités du domaine**.

Ajouter un concept à un domaine de concepts

Pour ajouter un concept à un domaine de concepts :

1. Ouvrez la fenêtre de propriétés du domaine de concepts.
2. Sélectionnez la page **Entités du domaine**.
3. Cliquez sur **Nouveau**.
L'assistant de création d'une information métier apparaît.
4. Sélectionnez le type d'objet "Concept" et indiquez son nom.
5. Cliquez sur **Ajouter**.
Le concept est ajouté à la liste des composants du domaine de concepts.

😊 Vous pouvez également glisser dans la section les concepts en question à partir de la vue hiérarchique du dictionnaire métier.

Relier ou supprimer un composant à partir d'un diagramme de concepts

Pour relier un élément de dictionnaire à la liste des composants d'un domaine de concepts :

1. Ouvrez le diagramme de concepts associé au domaine de concepts.
2. Ajoutez l'élément de dictionnaire qui vous intéresse dans le diagramme.
3. Faites un clic droit sur cet élément pour faire apparaître son menu contextuel.
4. Sélectionnez **Ajouter au domaine courant**.
L'élément est ajouté à la liste des éléments du domaine de concepts, dans la page **Entités du domaine** de la fenêtre de propriétés du domaine.

Pour supprimer un élément de dictionnaire d'un domaine de concepts :

1. Faites un clic droit sur l'élément de dictionnaire qui vous intéresse pour faire apparaître son menu contextuel.
2. Sélectionnez **Supprimer du domaine courant**.
L'élément est supprimé de la liste des composants du domaine de concepts.

Définir le mode d'accès aux composants (CRUD)

Il est possible de spécifier les droits d'accès à chacun des composants d'un domaine de concepts à travers le CRUD du composant en question (Create, Read, Update Delete).

Ouvrage		
Général Caractéristiques Entités du domaine Relations		
Nouveau Réordonner Hérité Exclu		
☐	Nom Local ↑	Accès aux données
☐	Body of work (EN)	CRUD
☐	Cinematographic work (EN)	CRUD
☐	Literary work (EN)	CRUD
☒	Musical work (EN)	CRUD
☐	Person (EN)	CRUD
☐	Work (EN)	CRUD

Pour définir le CRUD sur un composant du domaine de concepts :

1. Ouvrez la fenêtre de propriétés du domaine de concepts.
2. Cliquez sur la page **Entités du domaine**.
3. Sélectionnez la ligne du composant en question.
Des commandes s'ajoutent, dont le bouton **CRUD**.
4. Cliquez sur ce bouton.
5. Dans la fenêtre qui apparaît, cochez ou décochez les cases de chaque associées aux actions : Create, Read, Update Delete.

Le contenu de la colonne **Accès au données** est calculé automatiquement en fonction des actions cochées. Ce résultat apparaît sur la forme de l'objet dans le diagramme de concepts associé au domaine de concepts.

CONCEPT

Le concept est l'élément de base d'un dictionnaire métier.

Un concept représente la détermination de ce qu'est un être, une chose ou un mot, par ses propriétés et caractéristiques essentielles ou ses qualités propres.

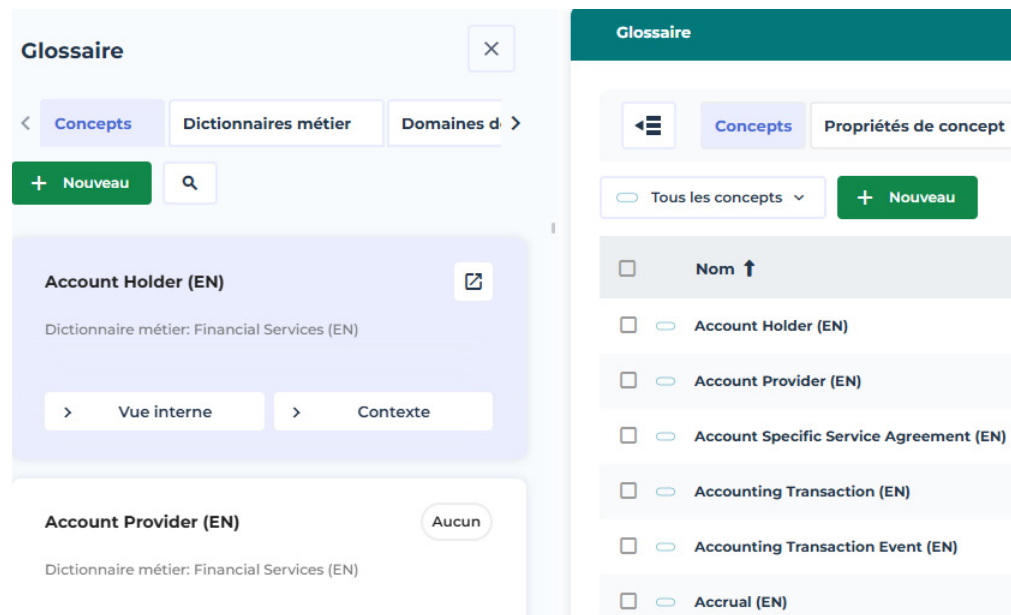
Un concept est associé à un ou des terme(s) qui désignent le concept dans une langue donnée. Voir [Consulter le glossaire métier](#).

Accéder à la liste des concepts

Pour accéder aux concepts dans **Hopex Data Governance** :

- 1 Cliquez sur le menu de navigation **Glossaire** > **Concepts**.
La liste des concepts apparaît.

Le bouton **Exploration**  présente la fiche de chaque concept, avec sa description et les contextes d'utilisation.



The screenshot displays the 'Glossaire' (Glossary) interface in Hopex Data Governance. On the left, a sidebar shows the navigation menu with 'Concepts' selected. The main area shows a list of concepts under the 'Concepts' tab. The first concept is 'Account Holder (EN)', which is linked to the 'Financial Services (EN)' dictionary. Below it, there are buttons for 'Vue interne' (Internal View) and 'Contexte' (Context). The second concept is 'Account Provider (EN)', also linked to 'Financial Services (EN)', with a button labeled 'Aucun' (None). On the right, a detailed view of the 'Concepts' list is shown, including a search bar, a '+ Nouveau' (New) button, and a list of concepts with checkboxes and their names: 'Account Holder (EN)', 'Account Provider (EN)', 'Account Specific Service Agreement (EN)', 'Accounting Transaction (EN)', 'Accounting Transaction Event (EN)', and 'Accrual (EN)'.

Créer un concept

Pour créer un **concept** :

1. Cliquez sur le bouton **Nouveau** associé à la liste des concepts.
2. Saisissez le **Nom** du concept et le **Dictionnaire métier détenteur**.
La section **Termes existants** dresse la liste des termes portant le même nom que le nouveau concept. Vous avez le choix entre utiliser un terme qui existe déjà ou créer un nouveau terme.

 *Un terme est un mot ou groupe de mots considéré dans sa valeur de désignation, en particulier dans un vocabulaire spécialisé.*

 *Si un terme a déjà été créé avec le même nom que le nouveau concept, ce terme est automatiquement relié et il apparaît dans la section **Terme**.*

3. Dans le champ **Description**, saisissez le texte de la définition du concept.
4. Cliquez sur **Suivant** pour associer une image au concept ou **OK** pour terminer.
Le nom du nouveau concept apparaît dans l'arborescence. Il apparaît également dans l'arborescence du dictionnaire métier détenteur. Un nouveau terme portant le même nom que le concept est également créé.

Concepts et termes

Relier un terme existant à un concept

Un même terme peut être lié à plusieurs concepts. Vous pouvez lier un terme lors de la création d'un concept ou ultérieurement.

Pour relier un terme à un concept :

1. Ouvrez la fenêtre de propriétés du concept qui vous intéresse et sélectionnez la page **Caractéristiques**.
2. Dépliez la section **Identification**.
3. Cliquez sur **Terme désigné** puis sur **Relier**.
4. Sélectionnez le terme voulu.

De même vous pouvez relier un concept à un terme dans les propriétés du terme :

1. Ouvrez les propriétés du terme.
2. Sélectionnez la page **Caractéristiques**.
La liste des objets reliés au terme apparaît dans le champ **Dictionnaire type identifié**.

 *Les objets dont le terme est déclaré comme synonyme ne figurent pas dans la fenêtre de propriétés.*

3. Cliquez sur le bouton **Relier**.
La fenêtre de recherche apparaît.
4. Sélectionnez le type d'objet "Concept" et cliquez sur le bouton **Rechercher**.
La liste des concepts existants apparaît.
5. Sélectionnez le concept voulu et cliquez sur **Relier**.

Créer des termes en plusieurs langues à partir d'un concept

Vous pouvez associer des termes à un concept pour chacune des langues de données de votre environnement.

Pour créer un terme à partir d'un concept :

1. Ouvrez la fenêtre de propriétés du concept qui vous intéresse et sélectionnez la page **Caractéristiques**.
2. Dépliez la section **Identification**.
3. Cliquez sur **Terme désigné** puis sur **Nouveau**. Une fenêtre de création de terme s'ouvre.
4. Spécifiez le **Nom local** du terme du terme.
5. Sélectionnez la **Langue** et cliquez sur **OK**.
Le nouveau terme apparaît dans la fenêtre de propriétés du concept.

Créer des synonymes en plusieurs langues



Un synonyme est un terme qui est interchangeable avec un autre dans le contexte d'un concept de ce terme avec le même sens, ou presque.

Il est possible d'ajouter des synonymes sur un concept dans plusieurs langues. Cette fonctionnalité sert à indiquer à l'utilisateur qu'un concept défini et utilisé dans un certain contexte correspond à d'autres synonymes dans une autre langue.

Renommer un concept

Le nom d'un concept est issu du terme associé. Pour renommer un concept, vous devez donc modifier le terme associé.

Pour modifier le terme associé à un concept :

1. Ouvrez les propriétés du concept.
2. Cliquez sur la page **Caractéristiques**.
3. A l'extrémité du champ **Nom**, cliquez sur la flèche droite puis sur **Renommer le terme existant**. Une fenêtre apparaît. Elle indique les objets touchés par le changement de nom.
4. Modifiez le nom du terme.
5. Cliquez sur **OK**.

Les propriétés d'un concept

Pour accéder aux propriétés d'un concept :

1. Sélectionnez le concept en question et cliquez sur le bouton **Propriétés**



de la zone d'édition.

 Certaines pages sont cachées par défaut. Pour les afficher, cliquez sur le bouton **Afficher/Masquer** puis sélectionnez la page voulue.

Vue d'ensemble

La vue d'ensemble d'un concept résume les informations principales du concept :

- Le dictionnaire métier auquel il appartient
- Les personnes responsables
- Son statut de workflow
- La catégorie de données (critique, privée, publiques, etc.)

Un dendrogramme illustre l'environnement du concept avec sa structure, les termes associés et le domaine dans lequel il est utilisé.

Caractéristiques

La page **Caractéristiques** de la fenêtre de propriétés d'un concept permet d'accéder aux principales caractéristiques du concept.

 Certaines caractéristiques sont dans des sections cachées par défaut. Pour les afficher, cliquez sur **Gérer les sections** et sélectionnez la section voulue.

Le concept est décrit par :

- la case **Concept abstrait** qui précise le caractère concret ou abstrait d'un concept
- sa **Définition**
- sa désignation qui est représentée par un ou plusieurs **termes**

 Pour modifier le nom d'un concept dans une langue donnée, vous devez accéder aux propriétés du concept et modifier le nom du terme qui lui est associé dans la langue concernée.

- ses **synonymes, hyperonymes** et **hyponymes**

Par exemple, dans le domaine financier, le terme "Avance" est reconnu comme synonyme du terme "Acompte".

 Un synonyme est un terme qui est interchangeable avec un autre dans le contexte d'un concept de ce terme avec le même sens, ou presque.

Composants

La page **Composant** présente :

- la liste des composants de concept détenus. Voir [Composants d'un concept](#).
- la liste des Super composants de concept. Voir [Décrire le super composant d'un concept](#).

 Les Concepts d'état reliés à un concept ne sont pas présentés dans la fenêtre de propriétés, pour plus de détails voir [Etats d'un concept ou d'un individu](#).

Le type de lien entre les concepts est défini par le **Type d'agrégation de concept** qui peut prendre les valeurs suivantes :

- "Référencement" : pour signifier que le concept cible est seulement référencé par le lien
- "Embarqué" : pour signifier que le concept cible a son existence propre, mais fait partie intégrante du concept qui est à l'origine du lien
- "Composite" : pour signifier que le concept cible est un composant du concept qui est à l'origine du lien, si le concept cible est détruit, le composite l'est également.

Relations

Super types

La page **Relations** > **Super types** présente les concepts dont les propriétés sont héritées par le concept décrit.

☛ Pour plus de détails sur les héritages, voir [Héritages d'un concept](#).

☛ Voir aussi [Concept type](#).

Réalisations

La page **Relations** > **Réalisations** permet de définir les objets que le concept met en œuvre. Il peut s'agir de deux types de réalisation :

- Une exigence réglementaire : un texte de référence, une politique d'entreprise, une dimension de qualité, etc.
- Une réalisation d'information métier.

☛ Pour plus de détails sur les réalisations, voir [Relier les concepts métier à l'architecture logique et physique](#).

Utilisation des données

La page **Utilisation des données** affiche les applications, processus et capacités métier qui utilisent la donnée en question.

☛ La page est cachée par défaut. Pour l'afficher, cliquez sur le bouton **Afficher/Masquer** puis sélectionnez la page voulue.

Voir [Utilisation des données par le SI](#).

Réglementations

La page **Réglementations** permet de définir quelles réglementations internes ou externes portent sur le concept en question.

Lorsqu'un concept appartient à une catégorie de données visée par une réglementation, le nom de cette réglementation apparaît dans cette page.

Pour relier une réglementation au concept :

1. Cliquez sur la page **Réglementations**.
2. Cliquez sur le bouton **Relier**.
Une fenêtre apparaît.
3. Indiquez le type de réglementation à relier (Texte de référence, politique d'entreprise, etc.).

4. Dans la liste qui s'affiche, sélectionnez la réglementation en question.
 Pour plus de détails sur les réglementations, voir [Réglementations](#).

Qualité des données

Sous la page **Qualité des données**, vous pouvez indiquer si une politique de qualité porte sur le concept en question.

 Pour plus de détails, voir [Qualité des données](#).

Rapports

Différents types de rapport sont disponibles sur le concept :

- **Dendrogramme des données** : illustre l'environnement du concept.
- **Graphe d'usage et de traitement de l'information**
- **Graphe de réalisation** : montre par quels éléments d'architecture est implémenté le concept ou les éléments d'information que le concept réalise.
- **Utilisation de l'information**. Le rapport affiche :
 - les domaines qui utilisent le concept, avec quels droits d'accès
 - dans quels systèmes (applications, systèmes applicatifs, service applicatif ou microservice) ces domaines sont utilisés et par quels composants de ces systèmes, en précisant le mode d'accès (lecture ou lecture/écriture).

Pour plus de détails sur les rapports de données, voir [Rapports et outils d'analyse des données](#).

Workflows

La page workflows vous permet de lancer sur le concept le workflow de conception propre aux données.

Voir aussi [Workflow de validation des données](#).

COMPOSANTS D'UN CONCEPT

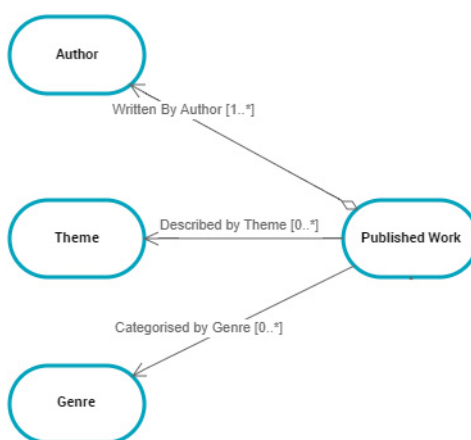
Il est possible de relier un concept à un autre concept pour le caractériser.

Par exemple, le concept d' "Ouvrage" est relié au concept de "Personne" pour caractériser l' "Auteur" d'un ouvrage.

Cette relation est décrite par un **Composant de concept** qui peut, éventuellement, être associé à un terme.



Un composant de concept permet de représenter une relation de dépendance entre deux concepts. Cette relation est orientée.



Accéder aux composants d'un concept

Pour accéder aux composants d'un concept :

1. Ouvrez la fenêtre de propriétés d'un concept.
 2. Cliquez sur la page **Caractéristiques**.
 3. Dépliez la section **Composants**.
- La liste des composants détenus apparaît.



Vous pouvez également consulter la liste des composants d'un concept à partir de son diagramme de structure de concept. Pour plus de détails, voir [Le diagramme de structure de concept](#).

Créer un composant de concept à partir d'un diagramme

La procédure de création de composant de concept "Auteur" entre les concepts "Ouvrage" et "Personne" est décrite à titre d'exemple.

Pour créer un composant de concept entre deux concepts d'un domaine de concepts :

1. Dans le diagramme de concept associé au domaine de concepts, cliquez sur le concept qui détient le lien, par exemple "Ouvrage".
Une barre qui contient les objets que vous pouvez insérer apparaît.
2. Cliquez sur l'icône qui représente le **Composant de concept**.



3. Glissez le curseur vers le concept cible, par exemple "Personne".
4. Lorsque le curseur prend la forme d'un double maillon, relâchez le bouton de la souris.
L'assistant de création d'un composant de concept apparaît.
5. Saisissez le **Nom**, par exemple "Auteur".
6. Etant donné que le terme "Auteur" doit être créé, cochez la case "Création avec terme".

La section **Terme** apparaît dans la fenêtre de création.

 *Un terme est un mot ou groupe de mots considéré dans sa valeur de désignation, en particulier dans un vocabulaire spécialisé.*

7. Dans le champ **Texte de la définition**, saisissez le texte de la définition du composant de concept et cliquez sur **OK**.
Le composant de concept apparaît dans le graphe.

 *Un nouveau terme portant le même nom que le composant de concept est également créé.*

Vous pouvez également créer un composant de concept dans un diagramme de structure de concept. Dans ce cas là, vous devez préciser le concept cible dans l'assistant de création d'un composant de concept.

 *Pour plus de détails, voir [Le diagramme de structure de concept](#).*

Décrire le super composant d'un concept

De même qu'un **Concept** peut être caractérisé par un lien vers un autre concept, un concept peut être caractérisé par un lien vers un **Concept type**.



Un concept type permet de classifier les concepts. Les relations entre les concepts type sont représentées par des composants de concept type.

Par exemple, chaque "Personne" adhérente pourrait être caractérisée par un "Type de prêt".

➡ Pour plus de détails, voir [Concept type](#).

La relation entre un **Concept** et un **Concept type** est décrite par un **Super-composant de concept**.



Un super-composant de concept permet de relier un concept à un concept type pour caractériser une propriété du concept.

Pour créer un **Super-composant de concept** entre un concept et un concept type dans un diagramme de domaine de concepts :

1. Dans la barre d'insertion, cliquez sur le bouton **Lien**.
2. Cliquez sur le concept qui détient le lien.

Par exemple, "Personne"

3. Cliquez sur le concept type cible.

Par exemple, "Type de prêt".

L'assistant de création d'un Super-composant de concept apparaît.

4. Renseignez le **Nom local**.
5. Si aucun terme n'est à créer, cochez la case "Création sans terme".
6. Cliquez sur **OK**.

Le Super-composant de concept apparaît dans le diagramme.

Décrire un composant de concept calculé

Voir [Règle de calcul sur un concept](#).

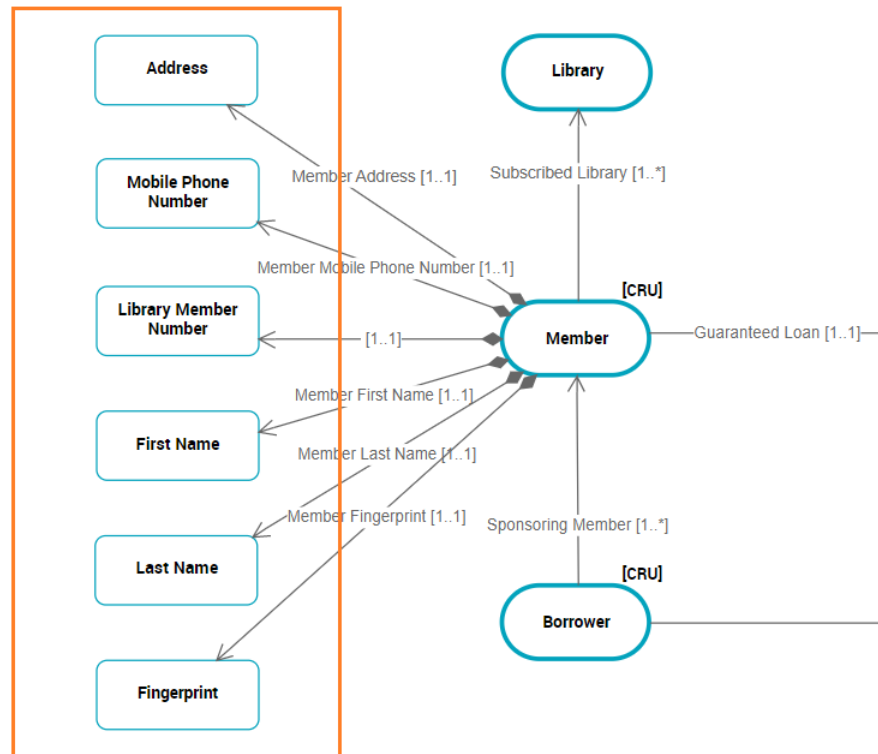
PROPRIÉTÉS DE CONCEPT

Les **propriétés de concept** servent à définir les caractéristiques attachées à un concept.

Par exemple, une personne est associée à une adresse postale obligatoire et unique, une adresse e_mail éventuellement et un ou plusieurs numéros de téléphone.

Une **propriété de concept** peut, elle-même, être reliée à d'autres propriétés de concept

Par exemple, l'adresse postale est définie à partir du nom de la rue et du nom de la ville.



Créer une propriété de concept

Créer une propriété de concept

Pour créer une propriété de concept :

1. Dans le diagramme de concept associé au domaine de concepts, cliquez sur l'icône **Propriété de concept** de la barre d'objets.
2. Cliquez dans le diagramme.
3. Saisissez le nom de la propriété de concept et cliquez sur **Ajouter**.

Relier une propriété de concept à un concept

Le lien entre un concept et une propriété de concept est décrit par une **sous-propriété** qui peut, éventuellement, être associée à un terme.



Une sous-propriété permet de spécifier la relation entre un concept et une propriété de concept.

Relier deux propriétés de concept

Le lien entre des propriétés de concept est décrit par un **composant de propriété de concept** qui peut, éventuellement, être associé à un terme.



Un composant de concept permet de représenter une relation de dépendance entre deux concepts. Cette relation est orientée.

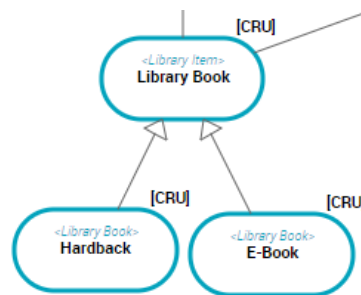
Créer une propriété de concept calculée

Voir [Règle de calcul sur un concept](#).

HÉRITAGES D'UN CONCEPT

Certains concepts métier sont des déclinaisons d'autres concepts : ils sont définis par les mêmes concepts.

Par exemple, le concept "Abonnement" est décliné en "Abonnement livre" et "Abonnement tout média". Ces deux types d'abonnement héritent des liens "Souscripteur" et "Adhérent" du spécifié au niveau du concept "Abonnement".



Accéder aux héritages d'un concept

Les héritages sont représentés par des variations.

Pour accéder aux variations d'un concept :

1. Ouvrez la fenêtre de propriétés d'un concept.
2. Sélectionnez la page **Relations** > **Super types**.
La liste des variations associées au concept apparaît.

Définir l'héritage d'un concept à partir d'un diagramme de concept

Vous pouvez spécifier qu'un concept hérite des caractéristiques définies pour un autre concept.

Par exemple, le concept "Abonnement livre" hérite du concept "Abonnement".

Pour spécifier dans un diagramme de concept qu'un concept hérite d'un autre :

1. Dans la barre d'insertion, cliquez sur le bouton **Variation**.
2. Cliquez sur le concept qui hérite et faites glisser la souris jusqu'au concept racine, avant de relâcher votre pression.
La variation est représentée par un lien mais il s'agit d'un objet **Hopex**.
3. Indiquez le **Nom** de la variation et cliquez sur **Ajouter**.
Un lien fléché du concept qui hérite vers le concept racine apparaît.

Définir l'héritage d'un composant de concept

Il est également possible de créer un héritage entre deux **Composants de concept**.

Par exemple, le "Souscripteur" est aussi un "Adhérent".

Pour définir un héritage entre deux composants de concept, il faut qu'ils soient reliés aux mêmes concepts, soit directement soit à travers des variations.

Pour créer une variation entre deux composants de concept :

1. Ouvrez la fenêtre de propriétés du composant de concept à varier.
2. Sélectionnez l'onglet **Variation**.
3. Cliquez sur le bouton **Nouveau**.
L'assistant de création d'une variation s'ouvre.
4. Sélectionnez les options :
 - "Initialisation des attributs"
 - "Initialisation des diagrammes" pour que la variation apparaisse dans les diagrammes.
5. Cliquez sur le bouton **OK**.
La variations est créée.

☛ Une variation entre deux **Composants de concept** est représentée graphiquement dans un diagramme de structure de concept. Pour plus de détails, voir [Le diagramme de structure de concept](#).

Créer une substitution d'un composant de concept

Si, au contraire d'une variation, un lien est une redéfinition d'un autre lien, vous allez créer une **substitution**.

📖 Une substitution détermine quel élément peut être utilisé à la place d'un autre ou est effectivement remplacé par un élément existant dans un contexte donné (par exemple dans le cadre d'une variation). Contrairement à la variation, une substitution n'induit pas d'héritage mais une équivalence fonctionnelle.

☛ Pour plus de détails sur les variations et les substitutions, voir le guide **Hopex Common Features**, chapitre "Manipuler les objets du référentiel", "Les variations d'objets".

Pour définir une substitution entre deux composants de concept, il faut qu'ils soient reliés aux mêmes concepts, soit directement soit à travers des variations.

Pour créer une substitution entre deux composants de concept à partir d'un diagramme de structure de concept :

1. Dans la barre d'insertion, cliquez sur le bouton **Substitution**.
2. Cliquez sur le composant à substituer, et faites glisser la souris jusqu'au composant substituant, avant de relâcher votre pression.
3. Renseignez le **Nom** et cliquez sur **Ajouter**.
Un lien en pointillé et fléché du composant à substituer vers le composant substituant apparaît.

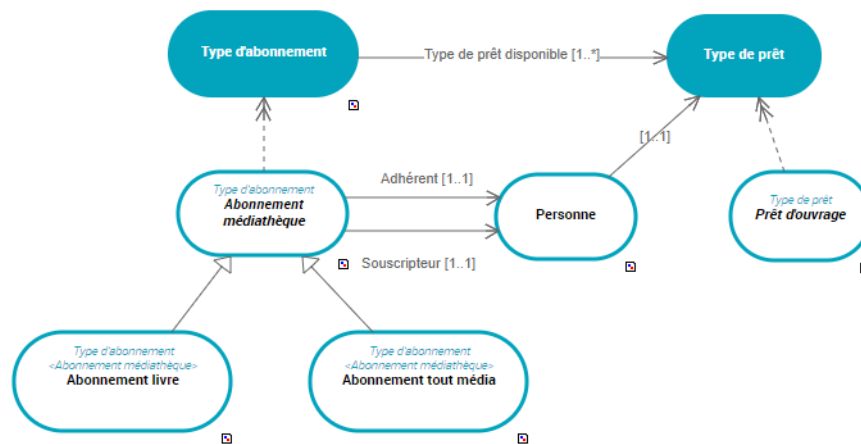
☛ La substitution est représentée par un lien mais il s'agit d'un objet **Hopex**.

LE DIAGRAMME DE STRUCTURE DE CONCEPT

Dans **Hopex Data Governance** et **Hopex Data Architecture**, un diagramme de structure de concept rassemble l'ensemble des informations relatives au concept. Ce diagramme est initialisé à partir de des éléments du diagramme de concept.

Par exemple, les "Abonnement" peuvent être classés par "Type d'abonnement".

Un "Type d'abonnement" est caractérisé par un "Type de prêt".



Ce diagramme est composé de :

- **variation** entre les composants,

Par exemple, les "Abonnement" peuvent être classés par "Type d'abonnement". Un "Type d'abonnement" étant caractérisé par un "Type de prêt".

 Une variation décrit comment un concept peut être varié sous une autre forme. La variante est un objet quasi-similaire à l'objet varié mais avec des propriétés ou des relations qui peuvent différer.

 Pour plus de détails, voir [Définir l'héritage d'un composant de concept](#).

- **substitution** entre les composants,

 Une substitution détermine quel élément peut être utilisé à la place d'un autre ou est effectivement remplacé par un élément existant dans un contexte donné (par exemple dans le cadre d'une variation).

Contrairement à la variation, une substitution n'induit pas d'héritage mais une équivalence fonctionnelle.

☛ Pour plus de détails, voir [Créer une substitution d'un composant de concept](#).

- **Composant de concept** qui décrit la relation entre deux **Concepts**,

Par exemple, un "Type d'abonnement" est caractérisé par un "Type de prêt disponible".



Un composant de concept permet de représenter une relation de dépendance entre deux concepts. Cette relation est orientée.

☛ Pour plus de détails, voir [Composants d'un concept](#).

- **Super-composant de concept** qui permet de caractériser le concept à partir d'un **Concepts type**,

Par exemple, chaque "Personne" adhérentes pourrait être caractérisée par un "Type de prêt".



Un super-composant de concept permet de relier un concept à un concept type pour caractériser une propriété du concept.

☛ Pour plus de détails, voir [Décrire les variations d'un concept type](#).

- **Événement de début**, **Événement intermédiaire** et **Événement de fin** qui permettent de définir les événements qui contribuent au changement d'état d'un concept,

Par exemple, le changement de état d'un adhérent peut être provoqué par son anniversaire.



Un concept événement représente un fait se produisant durant la vie d'un concept, par exemple un changement de saison. Un concept événement permet de marquer l'impact sur un concept d'un phénomène interne ou externe au concept. On peut distinguer les événements de début de concept, les événements de fin de concept et les événements intermédiaires de concept.

☛ Pour plus de détails, voir [Décrire les concepts d'état](#).

INDIVIDUS

Un individu représente l'instance d'un concept.

Accéder aux individus d'un concept

Vous pouvez accéder aux individus d'un concept dans les propriétés du concept :

1. Cliquez sur le menu de navigation **Glossaire > Dictionnaires métier**.
2. Cliquez sur l'onglet **Concepts**.
3. Cliquez sur le concept en question pour afficher ses propriétés.
4. Cliquez sur la page **Caractéristiques**.
5. En haut de la page, cliquez sur **Gérer les sections** et sélectionnez **Individus**.

La section correspondante apparaît avec la liste des occurrences du concept.

Créer un individu

Pour créer un individu à partir d'un concept :

1. Ouvrez les propriétés du concept.
2. Cliquez sur la page **Caractéristiques**.
3. Sous la section **Individus**, cliquez sur **Nouveau**.
L'assistant de création apparaît.
4. Sélectionnez l'option **Créer un objet**.
5. Saisissez le **Nom** de l'individu et cliquez sur **Créer**.

Les propriétés d'un individu

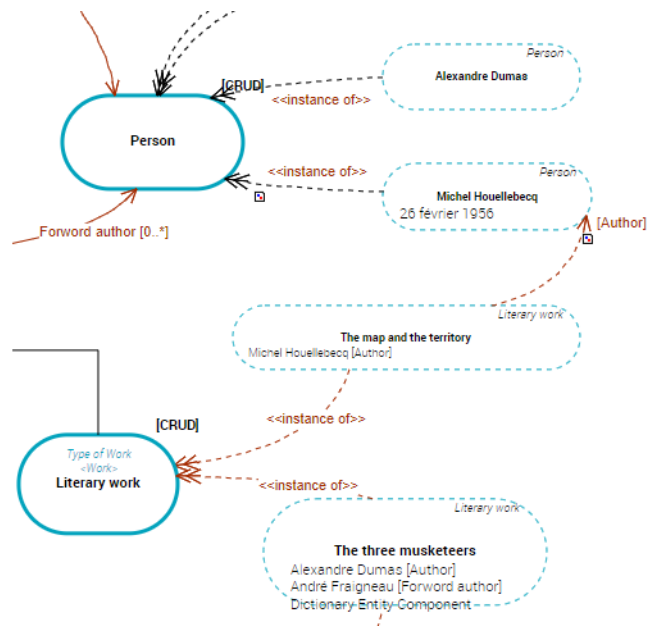
La fenêtre de propriétés d'un individu présente les éléments suivants :

- Ses **Caractéristiques** principales avec son **Nom Local** et le dictionnaire détenteur.
- Les **Classifications**, qui indiquent quel concept caractérise l'individu.
 Pour plus de détails, voir [Créer une classification d'individu](#).
- Les **Etats de l'individu** qui présentent les différents états d'un individu. Par exemple, l'individu "John Smith", relié au concept "Personne", passe de l'état d'enfant à "Adolescent", "Adulte", etc.
 Pour plus de détails, voir [Décrire les états et les événements d'un individu](#).
- Les **Composants**
 Pour plus de détails, voir [Créer un composant d'entité de dictionnaire](#).

Créer une classification d'individu

Une classification d'individu permet de relier un individu au concept qui le caractérise.

Par exemple, l'individu "Asimov" est une instance de "Personne" et "Les Robots" est une instance d'"Ouvrage".



Pour créer une classification d'individu :

1. Ouvrez la fenêtre de propriétés de l'individu qui porte la relation.

Par exemple, l'individu "Asimov".

2. Sélectionnez l'onglet **Caractéristique**.
3. Dans la section **Classification** cliquez sur le bouton **Nouveau**. L'assistant de création s'ouvre.
4. A gauche du champ **Élément caractérisant**, cliquez sur le bouton **Relier**. L'assistant de recherche s'ouvre s'ouvre.
5. Sélectionnez le concept que vous voulez relier.
Par exemple, le concept "Personne".
6. Cliquez sur le bouton **OK**.

Créer un composant d'entité de dictionnaire

Un composant d'entité permet de relier un individu à un élément de dictionnaire.

Vous pouvez également relier deux individus par une relation de type **Composant d'entité du dictionnaire**.

Par exemple, vous pouvez préciser que "Asimov" est l'auteur de l'ouvrage "Les Robots".

Pour créer un composant d'entité du dictionnaire entre deux individus :

1. Ouvrez la fenêtre de propriétés de l'individu qui porte la relation.

Par exemple, l'individu "Asimov".

2. Sélectionnez l'onglet **Composant**.

3. Cliquez sur le bouton **Nouveau**.

L'assistant de création s'ouvre.

4. A gauche du champ **Élément caractérisant**, cliquez sur le bouton **Relier**.

L'assistant de recherche s'ouvre s'ouvre.

5. Sélectionnez l'individu que vous voulez relier.

Par exemple, l'individu "Les Robots".

6. Cliquez sur le bouton **OK**.

Le composant d'entité est créé. Il apparaîtra dans le diagramme de structure d'individu de l'objet décrit.

➡ Pour plus de détails, voir [Le diagramme de structure d'individu](#).

Le diagramme de structure d'individu

Un diagramme de structure d'individu décrit la structure interne de l'instance de concept et les liens entre ses composants. Ce diagramme est initialisé à partir de des éléments du graphe de concept.

Ce diagramme est composé de *composants d'entité du dictionnaire* qui permettent de relier deux individus.

Il devient alors possible de préciser que "Asimov" est l'auteur de l'ouvrage "Les Robots".



Un composant d'entité permet de relier un individu à un élément de dictionnaire.

➡ Pour plus de détails, voir [Créer un composant d'entité de dictionnaire](#).

ETATS D'UN CONCEPT OU D'UN INDIVIDU

Un objet métier peut avoir un cycle de vie au cours duquel il prend des états différents en fonction d'événements. Si un concept est lié à un objet métier, d'autres concepts peuvent être liés aux différents états de l'objet métier ainsi qu'aux événements qui sont à l'origine des changements d'état. Avec **Hopex Data Governance** et **Hopex Data Architecture**, il est possible d'associer un cycle de vie à un concept ainsi que des concepts état et des concepts événement.

Enfin, les individus peuvent également être liés à des états et des événements d'individu qui sont des instanciations des concepts d'état et des concepts événement.

Décrire les concepts d'état

La notion d'état d'un concept est représentée par le **Concept d'état**.



Un concept d'état est une situation au cours de la vie d'un concept durant laquelle il satisfait à certaines conditions, exerce une certaine activité ou attend un événement de concept. Un concept d'état représente un intervalle de temps dont les bornes sont deux événements de concept. Un état de concept est une phase par laquelle passe le concept au cours de son cycle de vie.

Par exemple, un même Abonné peut passer d'un état de "Enfant" à l'état "Adolescent" puis à l'état "Adulte" et enfin "Senior".

Le passage d'un concept d'état à un autre peut être conditionné par un **Concept événement**.

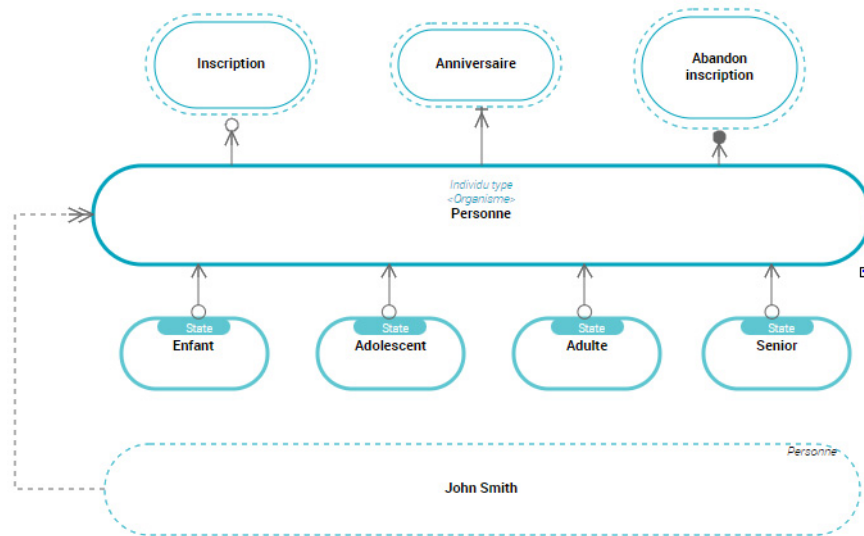


Un concept événement représente un fait se produisant durant la vie d'un concept, par exemple un changement de saison. Un concept événement permet de marquer l'impact sur un concept d'un phénomène interne ou externe au concept. On peut distinguer les événements de

début de concept, les événements de fin de concept et les événements intermédiaires de concept.

Par exemple, le passage d'un état à un autre peut être lié à un événement, un "Anniversaire", par exemple.

➡ Pour plus de détails, voir [Décrire les concepts événement](#).



Accéder à la liste des concepts d'état

Pour accéder aux concepts d'état d'un dictionnaire métier dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Glossaire > Dictionnaires métier**.
2. Cliquez sur dictionnaire métier qui vous intéresse.
Les propriétés du dictionnaire apparaissent.
3. Cliquez sur la page **Information métier**.
4. Dans la liste déroulante, sélectionnez **Concepts d'état**.
La liste des concepts d'état du dictionnaire métier apparaît.

Créer un concept d'état à partir d'un dictionnaire métier

Lors de la création d'un concept d'état, **HOPEX** construit également un **Etat du dictionnaire** qui représente la relation entre un concept d'état et son concept.

📖 Un état du dictionnaire permet de relier un concept à un état de concept et de spécifier la nature de l'état.

Pour créer un concept d'état à partir d'un dictionnaire métier :

1. Ouvrez les propriétés du dictionnaire métier.
2. Cliquez sur la page **Information métier**.
3. Dans la liste déroulante, sélectionnez **Concepts d'état**.
4. Cliquez sur **Nouveau**.

5. Indiquez le **Nom**.

Dans la section **Terme**, le tableau **Termes existants** dresse la liste des termes portant le même nom que le nouveau concept d'état.



Un terme est un mot ou groupe de mots considéré dans sa valeur de désignation, en particulier dans un vocabulaire spécialisé.



*Si un terme a déjà été créé avec le même nom que le nouveau concept d'état, ce terme est automatiquement relié et il apparaît dans la section **Terme**.*

6. Dans le champ **Entité métier phasée**, précisez à quel concept est relié le concept d'état que vous êtes en train de créer.



*Un **Etat du dictionnaire de** est automatiquement créé entre le concept et le concept d'état.*

7. Dans le champ **Texte de la définition**, saisissez le texte de la définition du concept d'état et cliquez sur **OK**.

Le nom du concept d'état apparaît dans l'arborescence sous le dictionnaire métier.



Vous pouvez également créer un concept d'état dans un domaine de concepts.

Les propriétés d'un concept d'état

Les caractéristiques d'un concept d'état

L'onglet **Caractéristiques** de la fenêtre de propriétés d'un concept d'état permet d'accéder à ses principales caractéristiques.

Un concept d'état est décrit par :

- sa désignation qui est représentée par un ou plusieurs **termes**



Pour modifier le nom d'un concept dans une langue donnée, vous devez accéder aux propriétés du concept et modifier le nom du terme qui lui est associé dans la langue concernée. Pour plus de détails, voir [Concept et terme](#).

- une **Définition**
- des **Synonymes**



Un synonyme est un terme qui est interchangeable avec un autre dans le contexte d'un concept de ce terme avec le même sens, ou presque.



Pour plus de détails, voir [Concept et terme](#).

Les liens entre un concept d'état et les autres éléments de dictionnaire

Outre les caractéristiques de terminologie, un concept d'état est caractérisé par ses relations avec les autres éléments de dictionnaire.

- L'onglet **Sur-type** présente les concepts dont les propriétés sont héritées par le concept décrit, pour plus de détails voir [Héritages d'un concept](#)
- L'onglet **Relations** > **Réalisation** permet d'associer au concept un élément de l'architecture applicative.
 - ☛ Pour plus de détails, voir [Relier les concepts métier à l'architecture logique et physique](#).
- L'onglet **Composant** présente :
 - la liste des composants de concept détenus, pour plus de détails voir [Composants d'un concept](#)
 - la liste des Super composants de concept, pour plus de détails voir [Décrire le super composant d'un concept](#).

☛ Les concepts reliés à un concept d'état ne sont pas présentés dans la fenêtre de propriétés.

Décrire les concepts événement

Un concept événement représente un fait se produisant durant la vie d'un concept, par exemple un changement de saison. Un concept événement permet de marquer l'impact sur un concept d'un phénomène interne ou externe au concept. On peut distinguer les événements de début de concept, les événements de fin de concept et les événements intermédiaires de concept.

Accéder à la liste des concepts événement

Pour accéder aux concepts événement d'un dictionnaire métier dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Glossaire** > **Dictionnaires métier**.
2. Cliquez sur dictionnaire métier qui vous intéresse.
Les propriétés du dictionnaire apparaissent.
3. Cliquez sur la page **Information métier**.
4. Dans la liste déroulante, sélectionnez **Concepts événement**.
La liste des concepts événement du dictionnaire métier apparaît.

Créer un concept événement à partir d'un dictionnaire métier

Pour créer un concept événement à partir d'un dictionnaire métier :

1. Ouvrez les propriétés du dictionnaire métier.
2. Cliquez sur la page **Information métier**.
3. Dans la liste déroulante, sélectionnez **Concepts événement**.
4. Cliquez sur **Nouveau**.

5. Indiquez son **Nom**.

La section **Termes existants** dresse la liste des termes portant le même nom que le nouveau concept événement.



Un terme est un mot ou groupe de mots considéré dans sa valeur de désignation, en particulier dans un vocabulaire spécialisé.



*Si un terme a déjà été créé avec le même nom que le nouveau concept événement, ce terme est automatiquement relié et il apparaît dans la section **Terme**.*

6. Dans le champ **Définition**, saisissez le texte de la définition du concept événement et cliquez sur **OK**.

Le nom du concept événement apparaît dans l'arborescence sous le dictionnaire métier.



Vous pouvez également créer un concept événement dans un domaine de concepts.

Les propriétés d'un concept événement

L'onglet **Caractéristiques** de la fenêtre de propriétés d'un concept événement permet d'accéder à ses principales caractéristiques.

Un concept événement est décrit par :

- une **Définition**
- sa désignation qui est représentée par un ou plusieurs **termes**,



Pour modifier le nom d'un concept dans une langue donnée, vous devez accéder aux propriétés du concept et modifier le nom du terme qui lui est associé dans la langue concernée. Pour plus de détails, voir [Concept et terme](#).

- des **Synonymes**



Un synonyme est un terme qui est interchangeable avec un autre dans le contexte d'un concept de ce terme avec le même sens, ou presque.



Pour plus de détails, voir [Concept et terme](#).

La section **Réalisation** permet d'associer au concept un élément de l'architecture applicative.



Pour plus de détails, voir [Relier les concepts métier à l'architecture logique et physique](#).

Relier un concept événement à son concept

La relation entre un concept et un concept événement est décrite par :

- un **Événement de début**,
- un **Événement de fin**,
- ou un **Événement intermédiaire**.

Pour relier un concept événement à son concept dans le diagramme associé à un domaine de concepts :

1. Dans la barre d'insertion, cliquez sur le bouton **Lien**.
2. Cliquez sur le concept auquel le concept événement est attaché.

Par exemple, "Personne"

3. Cliquez sur le concept événement à relier.
Par exemple, "Anniversaire".
Un assistant vous propose de choisir le type d'événement :
 - **Événement de début de concept,**
 - **Événement de fin de concept,**
 - **Événement intermédiaire de concept.**
4. Sélectionnez le type d'événement et cliquez sur **OK**.
L'assistant de création du type d'événement de concept sélectionné s'ouvre.
5. indiquez son **Nom**.
6. Etant donné qu'aucun terme n'est à créer, cochez la case "Création sans terme".
7. Cliquez sur **OK**.
Le lien entre le concept et le concept événement apparaît dans le diagramme avec une icône qui représente son type.

Le diagramme de structure de concept d'état

Un diagramme de structure de concept d'état rassemble l'ensemble des informations relatives au concept d'état décrit. Ce diagramme est initialisé à partir de des éléments du diagramme de concept.

Par exemple,

Ce diagramme est composé de :

- **variation** entre les composants,

Par exemple, les "Abonnement" peuvent être classés par "Type d'abonnement". Un "Type d'abonnement" étant caractérisé par un "Type de prêt".

 Une variation décrit comment un concept peut être varié sous une autre forme. La variante est un objet quasi-similaire à l'objet varié mais avec des propriétés ou des relations qui peuvent différer.

 Pour plus de détails, voir [Définir l'héritage d'un composant de concept](#).

- **substitution** entre les composants,

 Une substitution détermine quel élément peut être utilisé à la place d'un autre ou est effectivement remplacé par un élément existant dans un context donné (par exemple dans le cadre d'une variation).

Contrairement à la variation, une substitution n'induit pas d'héritage mais une équivalence fonctionnelle.

☛ Pour plus de détails, voir [Créer une substitution d'un composant de concept](#).

- **Elements d'information de concept**,

📖 Une sous-propriété permet de spécifier la relation entre un concept et une propriété de concept.

☛ Pour plus de détails, voir [Propriétés de concept](#).

- **Composant de concept** qui décrit la relation entre deux **Concepts**,

Par exemple, un "Type d'abonnement" est caractérisé par un "Type de prêt disponible".

📖 Un composant de concept permet de représenter une relation de dépendance entre deux concepts. Cette relation est orientée.

☛ Pour plus de détails, voir [Composants d'un concept](#).

- **Événement de début**, **Événement intermédiaire** et **Événement de fin** qui permettent de définir les événements qui contribuent au changement d'état d'un concept,

Par exemple, le changement de état d'un adhérent peut être provoqué par son anniversaire.

📖 Un concept événement représente un fait se produisant durant la vie d'un concept, par exemple un changement de saison. Un concept événement permet de marquer l'impact sur un concept d'un phénomène interne ou externe au concept. On peut distinguer les événements de début de concept, les événements de fin de concept et les événements intermédiaires de concept.

☛ Pour plus de détails, voir [Décrire les concepts d'état](#).

Décrire les états et les événements d'un individu

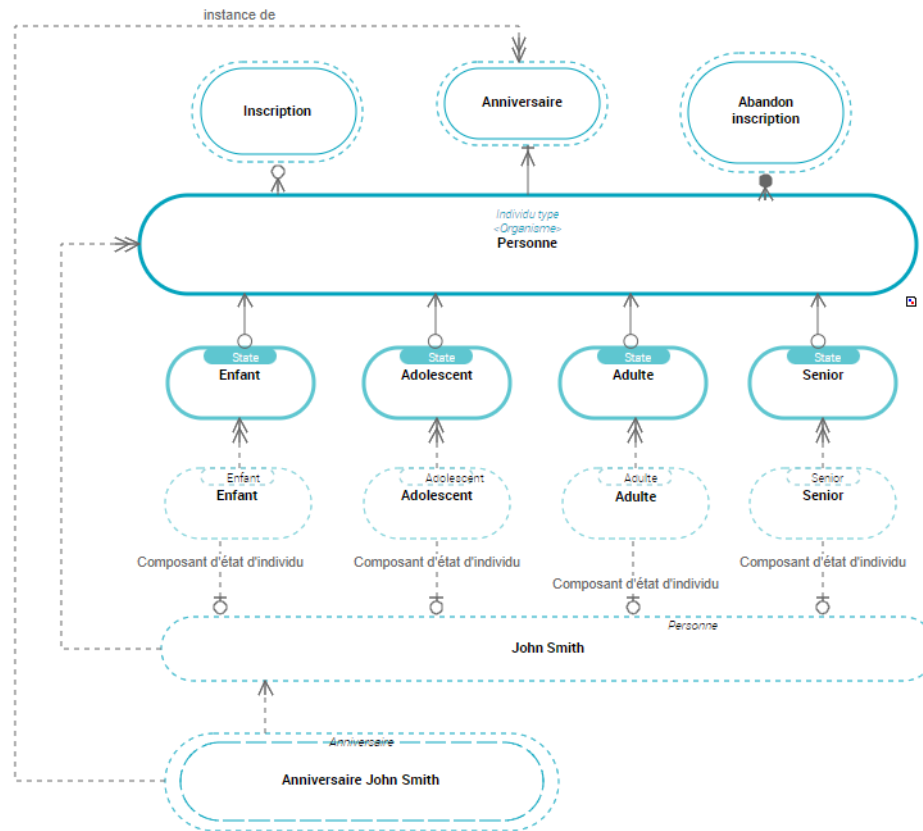
Si un concept est associé à des états, les occurrences de ce concept peuvent également être associées à des états. Ainsi, **Hopex Data Governance** et **Hopex Data Architecture** propose l'**Etat d'individu**.

📖 Un état d'individu est une instance d'un état du concept auquel l'individu est relié. Il représente un état de l'individu au cours de son cycle de vie.

Par ailleurs, le passage d'un état d'individu à un autre peut être conditionné par un **Événement d'individu**.

 Un événement d'individu représente un fait se produisant durant la vie de l'individu. C'est une instanciation d'un concept événement du concept auquel l'individu est relié.

Par exemple, "John Smith" est une "Personne" qui peut passer d'un état à autre le jour de son anniversaire.



La relation entre un individu et son **Etat d'individu** est décrite par un **Composant d'état d'individu**.

 Un composant d'état d'individu permet de relier un individu à un état d'individu.

La relation entre un individu et son **Événement d'individu** est décrite par un **Composant d'entité du dictionnaire**.

 Un composant d'entité permet de relier un individu à un élément de dictionnaire.

Avec **Hopex Data Architecture** :

- un état d'individu est une instance d'un concept d'état

 Un concept d'état est une situation au cours de la vie d'un concept durant laquelle il satisfait à certaines conditions, exerce une certaine

activité ou attend un événement de concept. Un concept d'état représente un intervalle de temps dont les bornes sont deux événements de concept. Un état de concept est une phase par laquelle passe le concept au cours de son cycle de vie.

- un événement d'individu est une instance concept événement.



Un concept événement représente un fait se produisant durant la vie d'un concept, par exemple un changement de saison. Un concept événement permet de marquer l'impact sur un concept d'un phénomène interne ou externe au concept. On peut distinguer les événements de début de concept, les événements de fin de concept et les événements intermédiaires de concept.

Accéder à la liste des états et des événements d'individu

Pour accéder aux états et événements d'individu d'un dictionnaire métier dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Glossaire > Dictionnaires métier**.
2. Affichez la vue hiérarchique.
3. A partir du dictionnaire métier qui vous intéresse, dépliez le dossier **Etat d'individu**.



Le dossier est visible uniquement si des états d'individu existent.

La liste des états d'individu du dictionnaire métier apparaît.

4. De même, dépliez le dossier **Evénement d'individu**.
La liste des événements d'individu du dictionnaire apparaît.

Créer un état d'individu à partir d'un domaine de concepts

La relation entre un individu et son **Etat d'individu** est décrite par un **Composant d'état d'individu**.



Un composant d'état d'individu permet de relier un individu à un état d'individu.

Si vous créez un état d'individu dans un diagramme, il est possible de créer automatiquement le composant d'état d'individu associé.

Pour créer un état d'individu à partir d'un diagramme de concept :

1. Dans le diagramme, cliquez sur l'individu qui détient l'état d'individu.
2. Sélectionnez **Etat d'individu**.
3. Cliquez dans le diagramme.
L'assistant de création apparaît.
4. Saisissez le **Nom local** et cliquez sur **Ajouter**.
Le nouvel état d'individu apparaît dans le diagramme.



Vous pouvez également créer un état d'individu à partir de son dictionnaire métier.

Les propriétés d'un état d'individu

La fenêtre de propriétés d'un état d'individu présente dans l'onglet **Caractéristiques** les éléments suivants :

- Son **Nom Local**.
- Les classifications d'individu qui apparaissent dans la section **Classification**.



Un composant d'état d'individu permet de relier un individu à un état d'individu.



Pour plus de détails, voir [Créer une classification d'individu](#).

- L'onglet **Composant** qui présente les individus qui définissent l'individu décrit.



Pour plus de détails, voir [Créer un composant d'entité de dictionnaire](#).

Créer un événement d'individu à partir d'un domaine de concepts

Pour créer un événement d'individu à partir d'un domaine de concepts :

1. Dans la barre d'insertion cliquez sur le bouton **Événement d'individu** et cliquez dans le diagramme.
L'assistant d'ajout d'un événement d'individu apparaît.
2. Saisissez le **Nom** et cliquez sur **Ajouter**.
L'événement d'individu apparaît dans le diagramme.

Relier un événement d'individu à un individu

La relation entre un individu et son **Événement d'individu** est décrite par un **Composant d'entité du dictionnaire**.



Un composant d'entité permet de relier un individu à un élément de dictionnaire.

Pour relier un concept événement à son concept dans le diagramme :

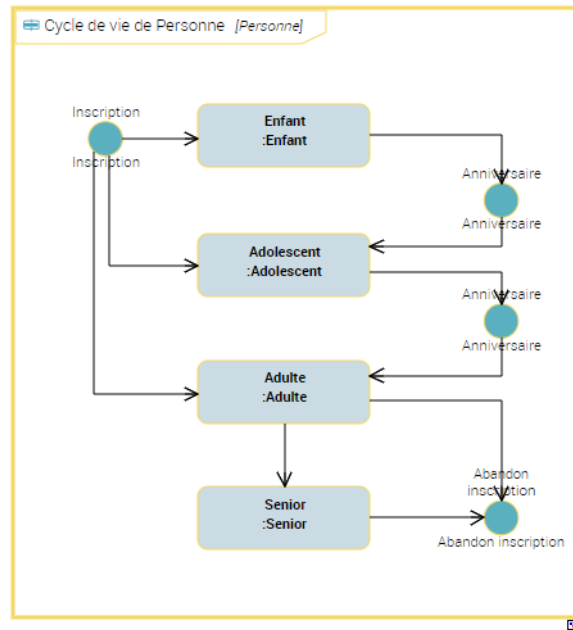
1. Dans la barre d'insertion, cliquez sur le bouton **Lien**.
2. Cliquez sur l'événement d'individu.
3. Cliquez sur l'individu.
Le lien apparaît dans le diagramme.

Le diagramme de structure de cycle de vie de concept

Le diagramme de structure de cycle de vie de concept permet de décrire le cycle de vie d'un concept.

Par exemple, une "Personne", devient visible d'une médiathèque après son "Inscription". Elle peut être inscrit avec l'état de "Enfant", "Adolescent", "Adulte" ou

"Senior". Le passage d'un état à un autre peut être lié à un événement, un "Anniversaire", par exemple.



Un diagramme de structure de cycle de vie de concept est composé des éléments suivants :

- Les **Phases de cycle de vie de concept**, qui sont reliées aux concepts d'état du concept "Personne",

 Un concept d'état est une situation au cours de la vie d'un concept durant laquelle il satisfait à certaines conditions, exerce une certaine activité ou attend un événement de concept. Un concept d'état représente un intervalle de temps dont les bornes sont deux événements de concept. Un état de concept est une phase par laquelle passe le concept au cours de son cycle de vie.

 Pour plus de détails sur les concepts d'état, voir [Décrire les concepts d'état](#)

- Les **Événements de cycle de vie de concept**, qui sont reliées aux concepts événements du concept "Personne",

 Un concept événement représente un fait se produisant durant la vie d'un concept, par exemple un changement de saison. Un concept événement permet de marquer l'impact sur un concept d'un phénomène interne ou externe au concept. On peut distinguer les événements de début de concept, les événements de fin de concept et les événements intermédiaires de concept.

 Pour plus de détails sur les concepts événements, voir [Décrire les concepts événement](#)

- Des **Transition de cycle de vie de concept** qui représente les enchaînements entre les états et les événements de concept.

Créer un cycle de vie de concept

Pour construire un diagramme de structure de cycle de vie de concept et décrire l'enchaînement des états qui définissent le cycle de vie d'un concept, il faut d'abord créer le **Cycle de vie du concept**.

Pour créer un cycle de vie de concept à partir d'un dictionnaire métier :

1. Faites un clic droit sur le dictionnaire métier qui vous intéresse et cliquez sur **Nouveau > Brique d'information métier**.
2. Sélectionnez le type d'objet **Cycle de vie de concept**.
3. Renseignez le **Nom local** et cliquez sur **Suivant**.
4. Dans le champ **Cycle de vie de**, précisez le concept sur lequel porte le cycle de vie.

Par exemple, "Personne".

5. La section **Termes existants** dresse la liste des termes portant le même nom que l'objet créé.

☛ Si un terme a déjà été créé avec le même nom que le cycle de vie, ce terme est automatiquement relié au concept et il apparaît automatiquement dans la section **Terme**.

6. Dans le champ **Texte de la définition**, saisissez le texte de la définition du concept d'état et cliquez sur **OK**.
Le nom du nouveau cycle de vie de concept apparaît dans l'arborescence sous le dictionnaire métier.

Créer un diagramme de structure de cycle de vie de concept

Pour créer un diagramme de structure de cycle de vie de concept à partir d'un cycle de vie de concept :

1. Faites un clic droit sur le cycle de vie de concept qui vous intéresse et cliquez sur **Nouveau > Diagramme de structure de cycle de vie de concept**.
Le diagramme s'ouvre dans la fenêtre d'édition. Les concepts d'état associés au concept décrit sont positionnés dans le diagramme à travers des objets de type **Phases de cycle de vie de concept**.

Ajouter un événement de cycle de vie de concept

Pour ajouter un événement de cycle de vie de concept dans le diagramme de structure de cycle de vie de concept :

Par exemple, l'événement de cycle de vie de concept qui correspond à l'"Inscription" de l'adhérent.

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Événement de cycle de vie de concept**.
2. Cliquez dans le cadre du cycle de vie de concept.
Une fenêtre de création d'un événement de cycle de vie de concept s'ouvre.
3. Dans le champ **Type composé**, précisez le nom du concept événement sur lequel porte l'objet créé.

Par exemple, "Inscription".

☛ Si une fenêtre de choix s'ouvre, sélectionnez l'objet qui vous intéresse.

4. Renseignez le **Nom local**.
5. Etant donné qu'aucun terme n'est à créer, cochez la case "Création sans terme".
6. Cliquez sur **OK**.
L'événement de cycle de vie de concept apparaît dans le diagramme.

Créer une Transition de cycle de vie de concept

Pour représenter l'enchaînement d'une phase vers un événement de cycle de vie de concept, vous devez créer une transition de cycle de vie de concept.

Pour créer une transition de cycle de vie de concept.

1. Dans la barre d'insertion, cliquez sur le bouton **Transition de cycle de vie de concept**.
2. Cliquez sur la phase (ou l'événement) de cycle de vie de concept déclenchante et, en maintenant le bouton gauche de la souris enfoncé, déplacez le curseur sur la phase (ou l'événement) déclenché.
3. Relâchez le bouton de la souris.
Le lien apparaît dans le diagramme.

Utiliser les périodes

Une **Période** permet d'apporter des précisions temporelles sur un **événement d'individu**.



Un événement d'individu représente un fait se produisant durant la vie de l'individu. C'est une instanciation d'un concept événement du concept auquel l'individu est relié.

Par exemple, un prêt gratuit peut être proposé aux abonnés à chaque anniversaire. Ce prêt est valide pendant une durée de deux semaines après la date d'anniversaire.

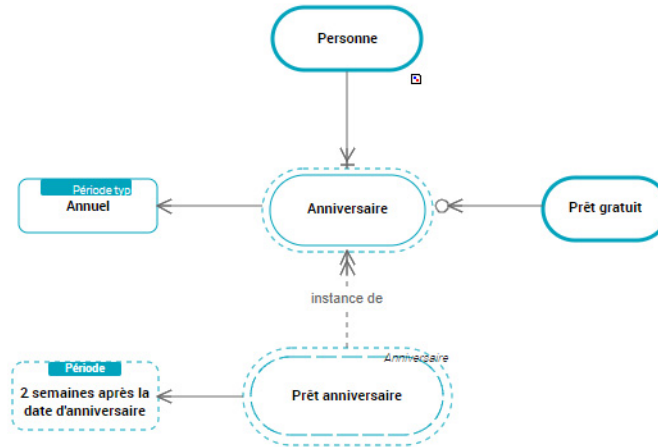
Une **Période type** permet de préciser un **concept événement**.



Un concept événement représente un fait se produisant durant la vie d'un concept, par exemple un changement de saison. Un concept événement permet de marquer l'impact sur un concept d'un phénomène interne ou externe au concept. On peut distinguer les événements de

début de concept, les événements de fin de concept et les événements intermédiaires de concept.

Par exemple, un prêt gratuit d'anniversaire est proposé tous les ans.



La relation entre une **période type** et un **événement d'individu** est décrite par une **Périodisation de Concept Événement**.

La relation entre une **période** et un **concept événement** est décrite par une **Périodisation d'événement**.

CONCEPT TYPE

Un concept type permet de classer les concepts. Les relations entre les concepts type sont représentées par des composants de concept type.

Accéder à la liste des concepts type

Pour accéder aux concepts type d'un dictionnaire métier dans **Hopex Data Governance**:

1. Cliquez sur le menu de navigation **Glossaire > Dictionnaires métier**.
2. Cliquez sur dictionnaire métier qui vous intéresse.
Les propriétés du dictionnaire apparaissent.
3. Cliquez sur la page **Information métier**.
4. Dans la liste déroulante, sélectionnez **Concept type**.
La liste des concept types du dictionnaire métier apparaît.

Créer un nouveau concept type

Pour créer un concept type à partir d'un dictionnaire métier :

1. Ouvrez les propriétés du dictionnaire métier.
2. Cliquez sur la page **Information métier**.
3. Dans la liste déroulante, sélectionnez **Concept type**.
4. Cliquez sur **Nouveau**.
5. Indiquez son **Nom**.

Dans la section **Terme**, le tableau **Termes existants** dresse la liste des termes portant le même nom que le nouveau concept type.



Un concept événement représente un fait se produisant durant la vie d'un concept, par exemple un changement de saison. Un concept événement permet de marquer l'impact sur un concept d'un phénomène interne ou externe au concept. On peut distinguer les événements de début de concept, les événements de fin de concept et les événements intermédiaires de concept.



*Si un terme a déjà été créé avec le même nom que le nouveau type de concept, ce terme est automatiquement relié et il apparaît dans la section **Terme**.*

6. Dans le champ **définition**, saisissez le texte de la définition du concept type et cliquez sur **OK**.



Un nouveau terme portant le même nom que le concept type est également créé.

Les propriétés d'un concept type

Les caractéristiques d'un concept type

L'onglet **Caractéristiques** de la fenêtre de propriétés d'un concept type permet d'accéder à ses principales caractéristiques.

Un concept type est décrit par :

- une **Définition**
- une **Désignation**, qui est représentée par un ou plusieurs termes,
 - ✎ Pour modifier le nom d'un concept dans une langue donnée, vous devez accéder aux propriétés du concept et modifier le nom du terme qui lui est associé dans la langue concernée. Pour plus de détails, voir [Concept et terme](#).
- des **Synonymes**
 - 📖 Un synonyme est un terme qui est interchangeable avec un autre dans le contexte d'un concept de ce terme avec le même sens, ou presque.
 - ✎ Pour plus de détails, voir [Concept et terme](#).

Les liens entre un concept type et les autres éléments de dictionnaire ou d'architecture

Outre les caractéristiques de terminologie, un concept est caractérisé par ses relations avec les autres éléments de dictionnaire.

- L'onglet **Composant** présente la liste des composants de concept type détenus, pour plus de détails voir [Composants d'un concept](#).
- L'onglet **Relations** > **Réalisation** permet d'associer au concept un élément de l'architecture applicative.
 - ✎ Pour plus de détails, voir [Relier les concepts métier à l'architecture logique et physique](#).
- L'onglet **Relations** > **Sur-types** présente les concepts type dont les propriétés sont héritées par le concept type décrit, pour plus de détails voir [Décrire les variations d'un concept type](#).

Décrire les composants d'un concept type

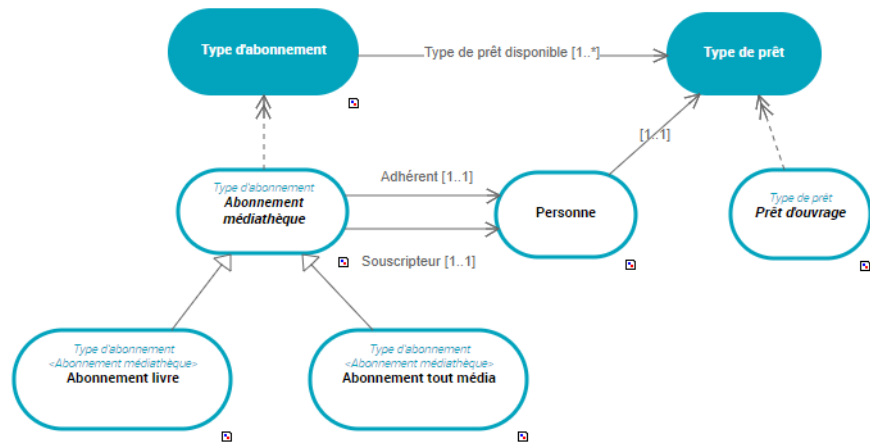
Il est possible de relier un concept type à un autre concept type pour le caractériser.

Par exemple, un "Type d'abonnement" est caractérisé par un "Type de prêt".

Cette relation est décrite par un **Composant de concept type** qui peut, éventuellement, être associé à un terme.



Un composant de concept type permet de spécifier la relation entre deux concepts type.



Accéder aux composants de concept type

Pour accéder aux composants de concept type d'un concept type :

1. Ouvrez la fenêtre de propriétés du concept type.
2. Sélectionnez l'onglet **Composants**.

La liste des composants de concept type associés au concept type apparaît.



Vous pouvez également consulter la liste des composants d'un concept type à partir de son diagramme de structure. Pour plus de détails, voir [Le diagramme de structure de concept type](#).

Créer un composant de concept type à partir d'un domaine de concepts

Pour créer un composant de concept type entre deux concepts type dans un diagramme de domaine de concepts :

1. Dans la barre d'insertion, cliquez sur le bouton **Lien**.
2. Cliquez sur le concept type qui détient le lien.

Par exemple, "Type d'abonnement"

3. Cliquez sur le concept type cible.

Par exemple, "Type de prêt".

L'assistant de création d'un composant de concept type apparaît.

4. Indiquez son **Nom**.
5. Etant donné qu'aucun terme n'est à créer, cochez la case "Création sans terme".
6. Cliquez sur **OK**.

Le composant concept type apparaît dans le diagramme.

Vous pouvez également créer un composant de concept type dans un diagramme de structure de concept type. Dans ce cas là, vous devez préciser le concept type cible dans l'assistant de création d'un composant de concept type.

✎ Pour plus de détails, voir [Le diagramme de structure de concept type](#).

Décrire les variations d'un concept type

Certains concepts type sont des déclinaisons d'autres concepts type : ils sont caractérisés par les mêmes composants de concept type.

Cette relation est décrite par une **Variation**.

📖 Une variation décrit comment un concept peut être varié sous une autre forme. La variante est un objet quasi-similaire à l'objet varié mais avec des propriétés ou des relations qui peuvent différer.

✎ Pour plus de détails sur les variations et les substitutions, voir le guide **Hopex Common Features**, chapitre "Manipuler les objets du référentiel", "Les variations d'objets".

Accéder aux variations d'un concept type

Pour accéder aux variations d'un concept type :

1. Ouvrez la fenêtre de propriétés du concept type.
2. Sélectionnez l'onglet **Relations** > **Sur-type**.
La liste des variations associées au concept type apparaît.

Créer une variation d'un concept type à partir d'un domaine de concepts

Pour spécifier qu'un concept type hérite des caractéristiques définies pour un autre concept type à partir d'un diagramme de domaine de concepts :

1. Dans la barre d'insertion, cliquez sur le bouton **Variation**.
2. Cliquez sur le concept type à varier, et faites glisser la souris jusqu'au nouveau concept type, avant de relâcher votre pression.
3. Indiquez son **Nom** et cliquez sur **Ajouter**.
Un lien fléché du concept type à varier vers le concept type racine apparaît.

✎ La variation est représentée par un lien mais il s'agit d'un objet **Hopex**.

Le diagramme de structure de concept type

Un diagramme de structure de concept type décrit la structure interne de l'instance de concept type par relations définies vers les autres concepts type qui le caractérisent.

Ce diagramme est composé des *composants de concept type* qui permettent de caractériser le concept type en le reliant à d'autres concepts type.

Par exemple, un "Type d'abonnement" est caractérisé par un "Type de prêt".



Un composant de concept type permet de spécifier la relation entre deux concepts type.



Pour plus de détails, voir [Décrire les composants d'un concept type](#).

VUE DE CONCEPT

Une vue de concept permet de représenter le périmètre sémantique couvert par un objet métier. Une vue de concept est construite à partir d'une sélection de plusieurs concepts reliés dans le contexte spécifique de la vue.

Un éditeur vous permet de créer et de visualiser des vues métier et leurs composants.

☛ Selon le même principe, la vue de données peut être utilisée pour naviguer à partir des Classes ou Entités. Pour plus de détails, voir [Vue de données logiques](#).

Créer une vue de concept

Pour créer une vue de concept :

1. Faites un clic droit sur le nom du concept et sélectionnez **Nouveau > Vue de concept**.
L'assistant de création d'une vue de concept apparaît.
2. Saisissez son **Nom**.
3. Le champ **Termes existants** dresse la liste des termes portant déjà le nom de la vue.
 *Un terme est un mot ou groupe de mots considéré dans sa valeur de désignation, en particulier dans un vocabulaire spécialisé.*
4. Dans le champ **Définition**, saisissez le texte de la définition de la vue et cliquez sur **Suivant**.
5. Pour spécifier le concept source de la vue de concept, cliquez sur **Nouveau**.
6. Dans la fenêtre qui apparaît indiquez :
 - la **MetaClasse** sur laquelle porte la vue (concept, concept d'état ou concept événement)
 - Le concept de référence de la vue de données
7. Cliquez sur **Ajouter**.
8. Cliquez sur **OK** pour fermer l'assistant de création d'une vue de concept.
La nouvelle vue de concept apparaît dans le dossier **Vues de concept** du dictionnaire métier concerné.

Définir le contenu de la vue de concept

Affichage des objets de la vue

L'éditeur de vue se décompose en plusieurs parties :

- la partie gauche présente l'ensemble des composants du concept source de la vue, tels que définis dans le dictionnaire de données
- la partie droite présente les composants du concept qui sont conservés dans la vue de concept créée
- les boutons de la colonne **Action** permettent d'ajouter les composants à la vue de concept.

Booking (Flight) (EN) x

Chemin de composant de vue:

View Specification Source:

	Action
Booking (Flight) (EN)	+
Réservation	
Annulation	+
Client	+
Création	+
Forfait voyage	+
Numéro de réservat...	+
Palement	+
Prise en charge	+
Prix	+
Validation	+
Réservation [Client]	+

View Component:

	Action
Booking (Flight) (EN)	
Booking End Date	✗ 🔍
Booking Number	✗ 🔍
Booking Start Date	✗ 🔍
Customer	✗ 🔍
Travel Package	✗ 🔍

Ajouter un objet source à la vue de concept

Pour ajouter un objet source à une vue de concept :

1. Ouvrez la vue de concept.
2. Du côté des objets sources, sous la colonne **Action**, cliquez sur le bouton 
3. Dans la fenêtre qui apparaît, indiquez :
 - la **MetaClasse** sur laquelle porte la vue (concept, concept d'état ou concept événement)
 - Le concept de référence de la vue de données
4. Cliquez sur **Ajouter**.

Une fois le concept source défini, vous pouvez sélectionner les composants de ce concept - ou le concept lui-même - à ajouter à la vue de concept.

Ajouter un composant à la vue de concept

A partir des objets sources de la vue, vous pouvez définir des composants embarqués et des composants référencés.

Un composant embarqué permet de prendre dans la vue toutes les informations qui composent l'objet. Un composant référence ne fait que référencer l'objet dans la vue.

Pour ajouter un composant embarqué à la vue de concept :

1. Dans l'arbre de la partie gauche, sélectionnez le composant que vous souhaitez ajouter à la vue.
2. Cliquez sur le bouton **Ajouter un composant d'inclusion de vue**.
Le composant ajouté apparaît dans l'arbre de droite.

☛ De la même manière, vous pouvez **Ajouter un composant référencé**.

Les objets embarqués dans la vue sont précédés d'une coche, contrairement aux objets référencés.

Les vues sont ensuite accessibles dans un rapport. Pour plus de détails voir [Report DataSets](#).

Le rapport de vue

Le rapport de vue offre un compte-rendu sur une vue de concept et ses composants.

Pour générer un rapport de vue :

1. Cliquez sur le menu de navigation **Rapports**.
2. Dans la zone d'édition cliquez sur **Créer un rapport**.
3. Recherchez le rapport type "Rapport de vue".
4. Sous le rapport type, cliquez sur **Créer un rapport**.
5. Sélectionnez la vue en question et rafraîchissez le rapport.

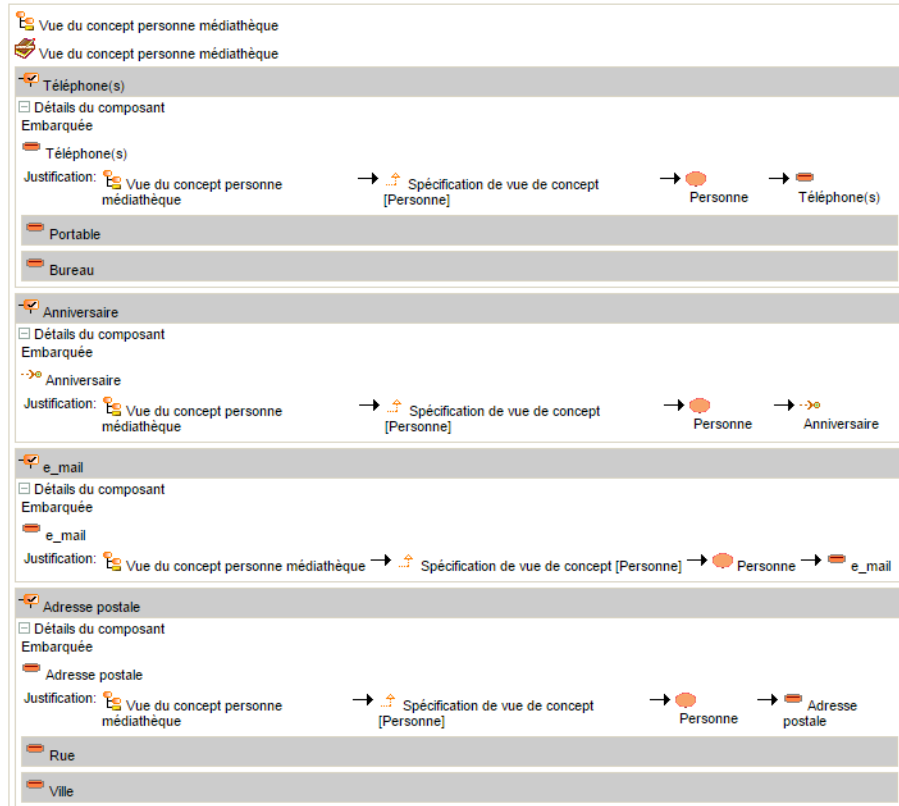
Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Vue	Vue	Obligatoire.
Sous-vue	oui ou non	Oui par défaut
Justification	oui ou non	Oui par défaut
Niveau de profondeur	Entier	

Exemple de rapport

L'exemple ci-dessous permet de visualiser les éléments de la vue basée sur le concept "Personne".



RÈGLE DE CALCUL SUR UN CONCEPT



Hopex Data Governance et **Hopex Data Architecture** vous permettent de définir des informations métier calculées, dont la valeur se calcule à partir des valeurs d'une ou de plusieurs autres informations métier.

ASSOCIER UNE RÈGLE DE CALCUL À UNE INFORMATION MÉTIER

Vous pouvez spécifier des règles de calcul sur les concepts et les propriétés de concept.

Règle de calcul sur une propriété de concept

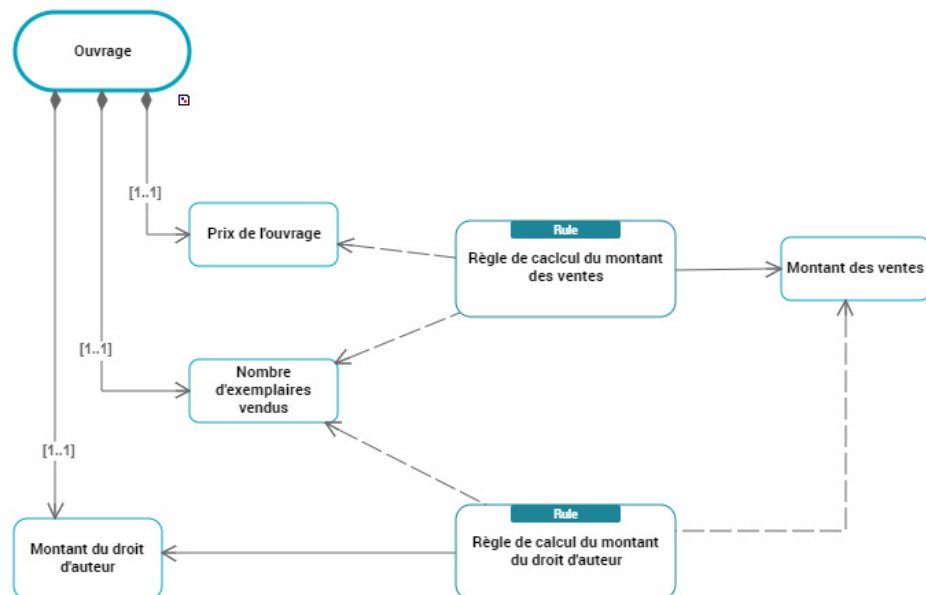
La règle de calcul prend en entrée des concepts ou propriétés de concept et en sortie la propriété de concept calculée.

Il convient ensuite de définir l'expression de la règle de calcul.

Exemple

Le concept "Ouvrage" est décrit par les propriétés de concept suivantes :

- "Prix de l'ouvrage"
- "Nombre d'exemplaires vendus"
- "Montant des ventes"
- "Montant du droit d'auteur"



Le montant des ventes est calculé selon :

- le prix de l'ouvrage
- le nombre d'exemplaires vendus

La définition de la règle de calcul du montant des ventes est la suivante :

Montant des ventes = Nombre d'exemplaires vendus x Prix de l'ouvrage

Le montant du droit d'auteur est calculé selon le nombre d'exemplaires vendus. La règle de calcul est la suivante :

- 8% du montant de la vente si le nombre d'exemplaires vendus est inférieur à 10 000.
- 10% du montant de la vente si le nombre d'exemplaires vendus est inférieur à 20 000 et supérieur à 10 000.
- 12% du montant de la vente si le nombre d'exemplaires vendus est supérieur à 20 000.

Créer la règle de calcul

Pour créer une règle de calcul dans un diagramme de concepts (par exemple, la "Règle de calcul du montant des ventes") :

1. Ouvrez le diagramme du domaine de concepts concerné.
2. Dans la barre d'insertion des objets, cliquez sur le bouton  **Règle de propriété de concept** et cliquez dans le diagramme. La fenêtre de création d'une règle apparaît.
3. Saisissez le nom de la règle et cliquez sur **Créer**.

Définir les objets en entrée et en sortie de la règle

Il s'agit ici de définir les objets sources de la règle, à partir desquels est calculée la valeur de l'objet cible.

Pour définir la propriété de concept en entrée de la règle :

1. Dans la barre d'insertion des objets, cliquez sur le bouton  **Lien** et tirez un lien de la propriété de concept source vers la règle de calcul.

Dans notre exemple, tirez un lien de "Prix de l'ouvrage" vers "Règle de calcul du montant des ventes" et un lien de "Nombre d'exemplaires vendus" vers la règle de calcul.

Une flèche en pointillé apparaît entre la règle et la propriété de concept source.



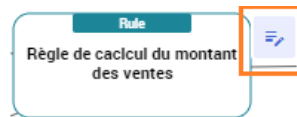
Pour définir la propriété de concept cible :

1. Cliquez sur le bouton  **Lien** et tirez un lien de la règle de calcul vers la propriété de concept calculée.
Une ligne fléchée apparaît entre les deux objets.



Pour accéder à la définition de la règle de calcul :

1. Passez le curseur sur la règle de calcul et cliquez sur le bouton des **Propriétés**.



La fenêtre de propriétés de la règle apparaît à droite du diagramme.

2. Sous la page **Caractéristiques**, section **Description de la règle**, saisissez la définition de la règle.

Dans notre exemple : "Montant des ventes = Nombre d'exemplaires vendus x Prix de l'ouvrage"

☛ Sous la section **Paramètres**, vous pouvez voir les objets que vous avez relié en entrée et en sortie de la règle.

Règle de calcul sur un concept

De même que pour une propriété de concept, vous pouvez indiquer qu'un concept est calculé, en lui associant une règle de calcul qui définit les paramètres en entrée et en sortie, ainsi que l'expression de la règle.

Pour un concept calculé, les objets en entrée de la règle peuvent être des concepts ou des propriétés de concept. L'objet en sortie est le concept calculé.

RELIER LES CONCEPTS MÉTIER À L'ARCHITECTURE LOGIQUE ET PHYSIQUE



Vous pouvez indiquer comment les concepts métier sont mis en œuvre dans le SI en les reliant aux objets de la couche logique ou physique.

Le travail de "réalisation des concepts" consiste à rapprocher les éléments de modèles de données ou de base de données avec les concepts métier afin de :

- définir de manière précise les objets manipulés au niveau de l'architecture du SI,
- assurer un meilleur partage du vocabulaire et une meilleure communication globale entre intervenants métier et intervenants SI.

Les points abordés dans ce chapitre sont :

- ✓ [La réalisation de concept](#)
- ✓ [Utiliser les matrices de réalisation](#)

LA RÉALISATION DE CONCEPT

A travers la fonctionnalité de "réalisation", vous pouvez relier des objets de la vue logique ou physique à des éléments de dictionnaire.

La réalisation peut être définie sur l'objet réalisé (le concept) ou sur l'objet qui réalise (donnée logique ou physique), et étendue aux composants des objets réalisés et réalisateurs.

Notez qu'il est également possible de relier des informations métier à d'autres informations métier.

Définir l'objet qui réalise un concept

Pour définir l'objet qui réalise un concept dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Glossaire > Dictionnaires métier**.
2. Dans la zone d'édition, cliquez sur l'onglet **Concepts**.
3. Sélectionnez le concept en question et ouvrez ses propriétés.
4. Dans la page de propriétés, cliquez sur la page **Relations > Réalisations**.
5. Sous l'onglet **Objets réalisateurs**, cliquez sur **Nouveau**. L'assistant de création d'une réalisation métier apparaît.
6. Indiquez :
 - le type d'objet qui réalise (donnée logique ou physique)
 - le nom de l'objet en question
7. Cliquez sur **Ajouter**.

L'objet réalisateur apparaît dans les propriétés du concept. Lorsque vous sélectionnez cet objet, une matrice apparaît sous la section qui suit. Elle affiche en ligne les composants de l'objet réalisateur (la classe) et en colonne les composants de l'objet réalisé (le concept). A partir de cette matrice vous pouvez définir quels composants de la classe (ex: attributs) réalisent quels composants de concept.

(Class of Block Component...	Catalog_Endin...	Catalog_Id	Catalog_Startin...
Catalog_Ending_date	✓		
Catalog_Id		✓	
Catalog_Starting_date			✓

Graphe de réalisation du concept

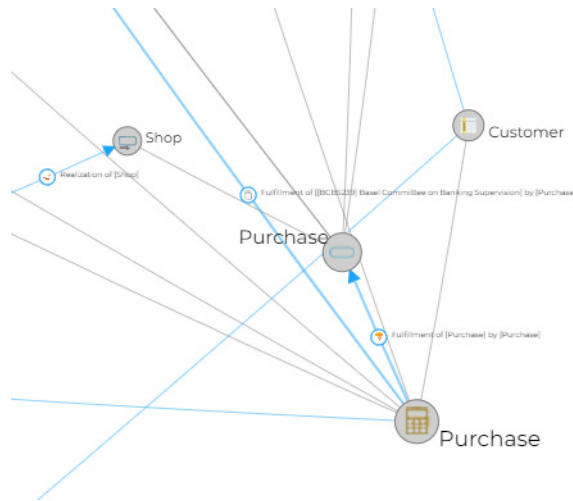
Le graphe de réalisation vous permet de visualiser par quels éléments d'architecture est implémenté un concept.

Pour accéder à ce rapport :

1. Ouvrez la fenêtre des propriétés du concept.
2. Cliquez sur la page **Rapports**.
3. Dans la liste des rapports sélectionnez le **Graphe de réalisation**.

L'exemple ci-dessous permet de visualiser les objets qui implémentent le concept "Purchase".

Notez que les réalisations des composants structurels des concepts passés en paramètre sont également affichées.



Définir le concept réalisé par une classe

Pour spécifier le concept réalisé par une classe :

1. Cliquez sur le menu de navigation **Architecture** puis sur **Dictionnaires de données**.
2. Dans la zone d'édition, cliquez sur l'onglet **Classes**.
3. Sélectionnez la classe que vous voulez relier à un concept et ouvrez sa fenêtre de propriétés.
4. Cliquez sur la page **Réalisations**.
5. Sous l'onglet **Objets réalisés**, cliquez sur **Nouveau**. La fenêtre **Ajout d'une réalisation détenue** apparaît.
6. Dans le champ **Type d'objet**, sélectionnez "Réalisation d'information métier" et cliquez sur **Suivant**.
7. Dans le champ **MetaClasse** sélectionnez "Concept".
8. Dans le champ **Information métier réalisée**, sélectionnez le concept qui vous intéresse.

9. Cliquez sur **Ajouter**.

Le concept réalisé apparaît dans les propriétés de la classe. Lorsque vous sélectionnez ce concept une matrice apparaît sous la section qui suit. Elle affiche en ligne les composants de l'objet réalisateur (la classe) et en colonne les composants de l'objet réalisé (le concept). A partir de cette matrice vous pouvez définir quels composants de la classe (ex : attributs) réalisent quels composants de concept.

UTILISER LES MATRICES DE RÉALISATION

Les matrices de réalisation permettent de définir et de visualiser les liens de réalisation entre des objets du référentiel.

Exemple

Données métier X données logiques

À travers cette matrice vous pouvez indiquer que des données logiques (classes, vue de données, etc.) réalisent des données métier (concepts, types de concept, etc.).

Les différents niveaux de réalisation

Informations métier X Actifs AE

Cette matrice met en correspondance des données métier (concepts, types de concept, etc.) avec des objets de l'architecture d'entreprise (applications, processus, etc.).

Données métier X données logiques

Réalisation de données métier (concepts, types de concept, etc.) par des données logiques (classes).

Informations métier X métadonnées

Réalisation de données métier (concepts, types de concept, etc.) par des métadonnées.

Informations métier X données physiques

Réalisation de données métier (concepts, types de concept, etc.) par des données physiques (tables, domaines relationnels, etc.).

Données logiques X données logiques

Réalisation de données logiques par d'autres données logiques.

Données logiques X métadonnées

Réalisation de données logiques par des métadonnées.

Données physiques X métadonnées

Réalisation de données physiques par des métadonnées.

Cartes de domaines de concepts X cartes de domaines de données

Réalisation de cartes de domaines de concepts par des cartes de domaines de données

Domaines de concepts X domaines de données logiques

Réalisation de domaines de concepts par des domaines de données logiques

Domaines de données logiques X domaines de données

Réalisation de domaines de données logiques par des domaines de données

☛ La réalisation de données logiques (classes, vue de données, etc.) par des données physiques (tables, vue de tables, etc.) se fait via l'outil de synchronisation. Voir [Synchroniser les modèles logiques et physiques](#).

Créer une matrice de réalisation

Pour créer une matrice de réalisation dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation puis sur **Outils > Réalisation de données**.
2. Dans la zone d'édition, cliquez sur la liste déroulante et sélectionnez le type de matrice à créer.
3. Cliquez sur **Nouveau**.
La matrice apparaît dans la zone d'édition.
4. Ajoutez en ligne les données qui réalisent et en colonne les données réalisées.
5. Pour indiquer qu'un objet en réalise un autre, cliquez sur la cellule de la matrice qui relie les deux objets en question.

Logical Data Realizatio...		Business Information R...						
Ajouter une ligne		Ajouter une colonne		Excel				
(Élément de données E/R / Class o...	Ada (EN)	ADM (EN)	agence d'envoi	agence de réception	agence de retour	agence location véhicule	ANSI (EN)	API (EN)
Activités								
Agence								
agence								
Agence de voyage								
Agent Call-Center								

Voir aussi : [Initialiser un dictionnaire métier à partir de données logiques ou physiques](#)

Architecture et outils



MODÉLISER LES DICTIONNAIRES DE DONNÉES



Les solutions **Hopex Data Governance** et **Hopex Data Architecture** permettent aux organisateurs et aux architectes d'entreprise de décrire le fonctionnement de l'entreprise par la modélisation des données utilisées lors de la mise en œuvre des processus et des applications. A cette fin, **Hopex Data Governance** et **Hopex Data Architecture** mettent à leur disposition plusieurs outils et notations.

A partir des modèles de données logiques, vous pouvez construire les modèles physiques correspondants, autrement dit créer les tables d'une base de données, avec ses colonnes, index et clés ainsi que les dessins du diagramme relationnel. Voir [Synchroniser les modèles logiques et physiques](#).

Vous pouvez également faire l'inventaire des applications qui utilisent les données logiques modélisées. Voir [Utilisation des données par le SI](#).

- ✓ [Options de modélisation des données logiques](#)
- ✓ [Vue d'ensemble des données logiques](#)
- ✓ [Dictionnaire de données](#)
- ✓ [Carte des domaines de données](#)
- ✓ [Domaines de données et domaines de données logiques](#)
- ✓ [Vues de données logiques](#)
- ✓ [Types de données](#)
- ✓ [Diagramme de classes](#)
- ✓ [Modèle de données](#)
- ✓ [La notation IDEF1X](#)
- ✓ [La Notation I.E.](#)
- ✓ [La Notation Merise](#)

OPTIONS DE MODÉLISATION DES DONNÉES LOGIQUES

Les formalismes

Vous pouvez modéliser les données logiques à partir de deux formalismes :

- le paquetage de données, pour construire les diagrammes de classes (notation UML)
- le modèle de données, pour les diagrammes de données (notations standard, IDEF1X, I.E, Merise)

Pour afficher un des formalismes :

1. Dans le bureau, cliquez sur le menu **Menu principal > Paramètres > Options.**
2. Dans l'arbre de navigation des options, dépliez les dossiers **HOPEX Solutions > Fonctionnalités communes liées aux données.**
3. Cliquez sur **Formalisme de données.**
4. Dans la partie droite de la fenêtre cochez le(s) formalisme(s) que vous voulez afficher.
5. Cliquez sur **OK.**
Les dossiers correspondant aux paquetages et aux modèles de données apparaissent dans le volet de navigation **Architecture.**

Voir aussi : [Formalisme logique et synchronisation.](#)

Les notations

Vous disposez d'une notation standard de modèle de données, cochée par défaut. Pour afficher une autre notation (DEF1X, I.E ou Merise) :

1. Dans le bureau, cliquez sur le menu **Menu principal > Paramètres > Options.**
2. Dans l'arbre de navigation des options, dépliez les dossiers **HOPEX Solutions > Fonctionnalités communes liées aux données.**
3. Cliquez sur **Notation des données.**
4. Dans la partie droite de la fenêtre, cochez les notations que vous voulez utiliser.
5. Cliquez sur **OK.**

VUE D'ENSEMBLE DES DONNÉES LOGIQUES

Les solutions **Hopex Data Governance** et **Hopex Data Architecture** offrent chacune un profil dédié à la définition des données logiques : le **Gestionnaire de données** et **l'Architecte de données**.

Dictionnaire de données

Un dictionnaire de données rassemble et détient un ensemble de données logiques, et leur fournit un espace de nommage.

Le dictionnaire de données participe donc à l'organisation des données dans le référentiel Hopex.

Voir [Dictionnaire de données](#).

Carte des domaines de données

Une carte des domaines de données permet de représenter les domaines de données logiques d'un dictionnaire de données et leurs liens de dépendance.

Voir [Carte des domaines de données](#).

Domaine de données logiques

Un domaine de données logique représente une structure de données restreinte, dédiée à la description d'un stockage de données logicielles. Il est constitué de classes et/ou de vue de classes et peut être décrit par un diagramme de domaine de données.

Pour plus de détails, voir [Domaines de données et domaines de données logiques](#).

Pour répondre à des cas d'utilisation précis, vous pouvez créer des Vues de données dans lesquelles vous pouvez visualiser et modifier le périmètre couvert par des classes.

Vue de données logiques

A partir du périmètre d'un objet d'un dictionnaire de données ou d'un domaine de données, une vue de données logiques permet de définir un ensemble d'informations pour un usage spécifique. Voir [Vues de données logiques](#).

Modèle de données

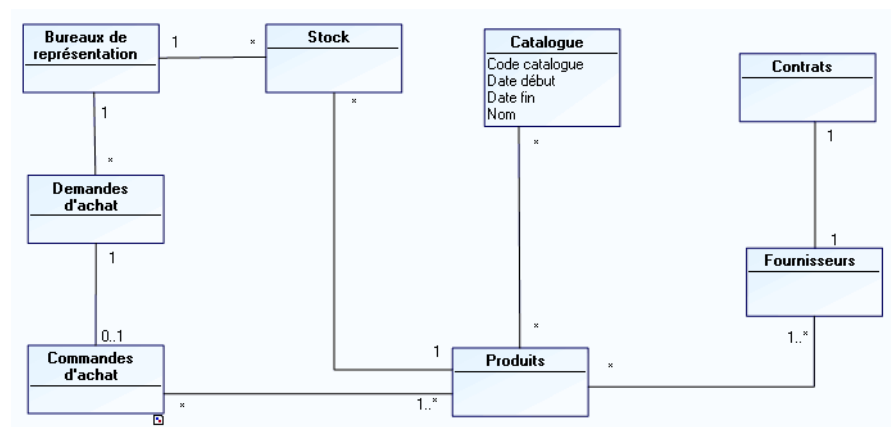
Lorsque vous faites le choix de travailler avec le formalisme "Modèle de données", les dictionnaires métier sont représentés par des modèles de données (et non des dictionnaires de données).

Voir [Formalisme logique et synchronisation](#).

Pour plus de détails sur la création et la mise à jour d'un modèle de données voir [Modèle de données](#).

Exemple

Le modèle de données du projet "Automatisation des demandes d'achat" est présenté ci-dessous.



L'application gère les demandes d'achat, les commandes et les stocks de produits dans chacun des bureaux de représentation.

Un catalogue centralisé des produits et des fournisseurs est mis en place.

Les contrats avec les fournisseurs référencés sont également accessibles depuis l'application.

DICTIONNAIRE DE DONNÉES

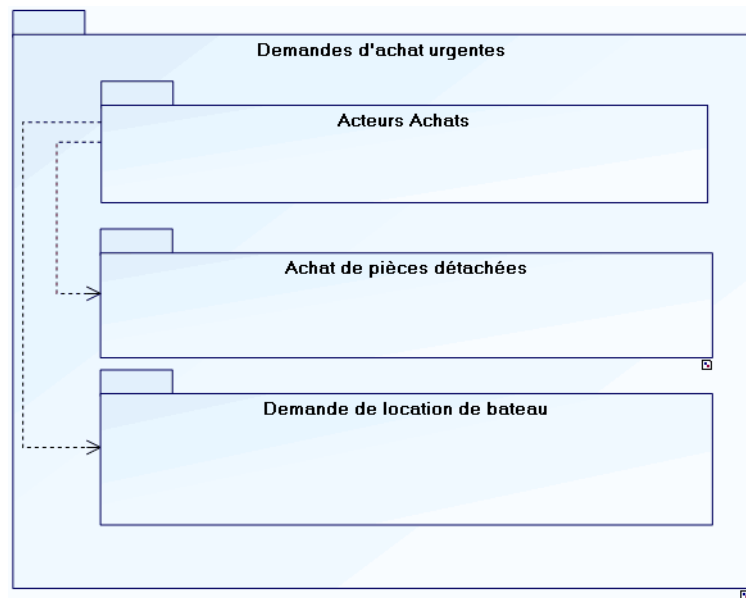
Un dictionnaire de données rassemble et détient un ensemble de données logiques. Il peut être découpé en domaines de données logiques. Voir [Domaine de données logiques](#).

Éléments d'un dictionnaire de données

Un dictionnaire de données vous permet de décrire tous les éléments qui définissent votre architecture de données logiques :

- les domaines de données
- les classes
- les attributs
- les parties
- Etc.

Un dictionnaire de données est implémenté par un **Paquetage** qui rassemble les données. Il est possible de créer des sous-paquetages.



Les demandes d'achat urgentes sont prévues pour traiter les achats de pièces détachées et les demandes de location de bateau. Dans ces deux cas, les utilisateurs sont des acteurs du domaine des achats.

☛ Pour plus de détails sur l'utilisation des paquetages, voir le guide **Hopex IT Architecture**.

Accéder aux éléments d'un dictionnaire de données

Pour accéder aux éléments d'un dictionnaire de données dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture > Dictionnaires de données**.

 Dans **Hopex Data Architecture**, cliquez sur le menu de navigation **Dictionnaires > Vue hiérarchique**.

2. Dans la zone d'édition, déployez le dossier du dictionnaire de données qui vous intéresse.
Les éléments qui composent le dictionnaire apparaissent.

Importer des données logiques

Vous pouvez importer dans votre référentiel des données logiques existantes à l'aide d'un fichier Excel. Voir [Importer des données logiques à partir d'un fichier Excel](#).

CARTE DES DOMAINES DE DONNÉES

Un dictionnaire de données peut être décomposé en domaines de données logiques. Une carte des domaines de données permet de visualiser les liens de dépendance entre des domaines de données logiques.

Les liens de dépendance entre domaines sont automatiquement déduits à partir des objets décrits préalablement dans chaque domaine de la carte.

☛ Pour plus de détails sur le dictionnaire de données voir [Dictionnaire de données](#).

Créer une carte des domaines de données

Vous pouvez créer une carte des domaines de données à partir du dictionnaire de données qu'elle décrit.

Pour créer une carte des domaines de données (logique) dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture > Dictionnaires de données**.

☛ Dans **Hopex Data Architecture**, cliquez sur le menu de navigation **Dictionnaires > Vue hiérarchique**.

2. Dans la zone d'édition, sous la vue hiérarchique, déployez le dossier **Dictionnaires de données**.
3. Faites un clic droit sur le nom du dictionnaire concerné et cliquez sur **Nouveau > Carte des domaines de données**.
La carte créée apparaît dans l'arbre.

Pour créer le diagramme de la carte des domaines de données :

1. Faites un clic droit sur la carte et sélectionnez **Nouveau > Diagramme**.
2. Sélectionnez le type de diagramme **Carte des domaines de données**.
3. Cliquez sur **OK**.
Le diagramme apparaît dans la zone d'édition.

Les composants d'une carte des domaines de données

Dans une carte des domaines de données vous pouvez ajouter des composants internes et externes.

Les composants internes sont les domaines de données logiques qui font partie du périmètre de la carte (qu'ils appartiennent ou non au paquetage détenteur).

Les composants externes sont ceux qui sont utilisés dans la carte mais qui ne font pas partie du périmètre étudié.

DOMAINES DE DONNÉES ET DOMAINES DE DONNÉES LOGIQUES

Les domaines de données et domaines de données logiques permettent de définir une structure de données logiques constituée de classes et de vues de classes.

- Le domaine de données est utilisé pour décrire les dépôts de données de logiciels (Système applicatif, Application, Service applicatif ou Micro Service).
- Le domaine de données logique est utilisé pour décrire les dépôts de données (internes ou externes) de systèmes applicatifs logiques.

➡ Pour plus de détails sur l'usage des domaines de données dans une architecture applicative, voir dans la documentation **Hopex IT Architecture** "Modéliser les architectures fonctionnelles et techniques".

Les deux sont détenus par un paquetage et peuvent référencer des objets détenus dans d'autres paquetages.

Il est possible de définir le mode d'accès (CRUD) aux objets référencés par un domaine de données en les intégrant comme composants du domaine de données.

➡ Une structure physique correspondante peut être définie via un domaine de données physiques. Celui-ci est constitué de tables et de vues de tables. Voir [Modéliser les bases de données](#)

Créer un domaine de données

Vous pouvez créer un domaine de données / domaine de données logique à partir du dictionnaire de données qu'il décrit.

Pour créer un domaine de données logique dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture > Dictionnaires de données**.

➡ Dans **Hopex Data Architecture**, cliquez sur le menu de navigation **Dictionnaires > Vue hiérarchique**.

2. Dans la zone d'édition, déployez le dossier **Dictionnaires de données**.
3. Faites un clic droit sur le nom du dictionnaire concerné et cliquez sur **Nouveau > Domaine de données** ou **Domaine de données logique**. Le domaine de données apparaît dans l'arbre.

Le diagramme de domaine de données

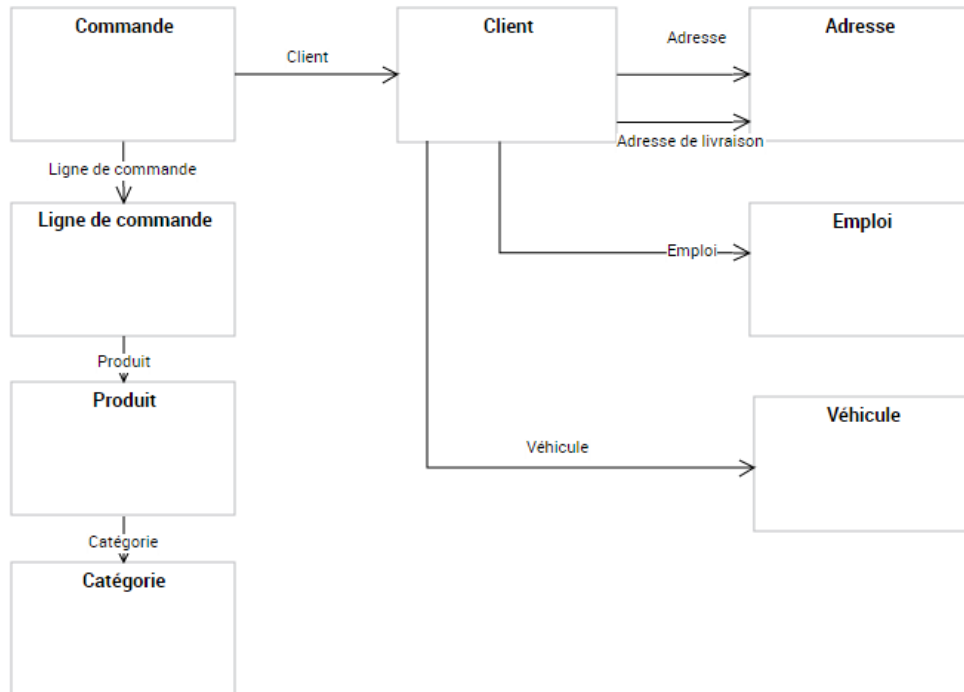
Les domaines de données / domaines de données logiques peuvent être décrits par un diagramme.

Un diagramme de domaine de données est un diagramme de structure qui définit des classes et leurs relations suivant le formalisme Tout/Partie, en rapport avec le sujet du domaine de données décrit.

Vous pouvez relier plusieurs diagrammes de domaine de données à un domaine de données, suivant ce que vous voulez décrire.

Exemple de diagramme

Le diagramme de domaine de données suivant représente une structure de données relative aux Commandes ; il décrit sous forme de tout/partie les classes et leurs relations.



Créer le diagramme d'un domaine de données

Pour créer un diagramme de domaine de données à partir du domaine de données :

1. Faites un clic droit sur le domaine de données et sélectionnez **Nouveau > Diagramme**.
2. Sélectionnez le type de diagramme **Diagramme de domaine de données**.
3. Cliquez sur **OK**.

Ajouter un objet au diagramme

Dans le diagramme du domaine de données, vous pouvez ajouter un nouvel objet ou relier un objet existant.

Les objets visibles dans un diagramme de données ne sont pas automatiquement rattachés au domaine de données. Une commande vous permet de définir les objets

comme composants du domaine. Voir [Ajouter un composant au domaine de données](#).

Ajouter une classe

Pour ajouter une nouvelle classe dans le diagramme :

1. Dans la barre d'insertion du diagramme, cliquez sur le bouton **Classe** puis cliquez dans le diagramme.
La fenêtre **Ajout d'une classe** apparaît.
2. Saisissez le nom de la classe et cliquez sur **Ajouter**.

Ajouter une vue de données

Pour ajouter une nouvelle vue de données dans le diagramme :

1. Dans la barre d'insertion du diagramme, cliquez sur le bouton **Vue de données** puis cliquez dans le diagramme.
La fenêtre **Ajout d'une vue de données** apparaît.
2. Saisissez le nom de la vue de données et cliquez sur **ajouter**.
3. L'éditeur de vue apparaît. Il vous permet de définir les composants de la vue. Voir [Vues de données logiques](#).

Ajouter un composant au domaine de données

Il est possible d'attacher des objets au domaine de données à travers des composants. Un composant référence un objet (classe ou vue de classe) et définit le type d'accès à l'objet en question (lecture seule, modification, suppression, etc.).

Le domaine de données est rattaché à un paquetage ; les objets créés directement par le biais de composants sont reliés automatiquement au paquetage du domaine de données.

Vous pouvez créer un composant à partir d'un objet qui figure dans le diagramme du domaine de données ou à partir des propriétés du domaine de données.

Pour créer un composant à partir d'un objet du diagramme de domaine de données :

1. Dans le diagramme, faites un clic droit sur l'objet en question et sélectionnez **Ajouter à (nom du domaine de données)**.
Le nom du composant créé apparaît dans les propriétés du domaine de données. Par défaut il porte le nom de l'objet qu'il référence.

Définir le mode d'accès aux composants (CRUD)

Vous pouvez spécifier les droits d'accès à chacun des composants d'un domaine de données en définissant le CRUD du composant en question (Create, Read, Update Delete).

Pour définir le CRUD sur un composant du domaine de données :

1. Ouvrez la fenêtre de propriétés du domaine de données.
2. Cliquez sur la page **Composants**.
3. Sélectionnez la ligne du composant en question.
Des commandes s'ajoutent, dont le bouton **CRUD**.

4. Cliquez sur ce bouton.
5. Dans la fenêtre qui apparaît, cochez ou décochez les cases de chaque associées aux actions : Create, Read, Update Delete.

Le contenu de la colonne **Accès au données** est calculé automatiquement en fonction des actions cochées. Ce résultat apparaît sur la forme de l'objet dans le diagramme associé au domaine de données.

VUES DE DONNÉES LOGIQUES

Une vue de données est une sélection de classes associées dans le contexte spécifique de la vue, pour un usage précis. Elle est construite à partir du périmètre d'un objet d'un dictionnaire de données ou d'un domaine de données.

➤ Selon le même principe, la vue de concept est utilisée pour visualiser le périmètre sémantique d'un objet métier. Pour plus de détails, voir [Vue de concept](#).

Créer une vue de données logiques

Créer une vue de données logiques consiste à :

- définir les objets sources, sur lesquels porte la vue (une classe ou une vue de données)
- définir plus précisément les propriétés des objets sources à prendre en compte dans la vue (les attributs, les parties)

Par exemple, pour la gestion de commandes, vous avez besoin de récupérer l'adresse de livraison disponible pour chaque client. Pour prendre en compte uniquement cette information, vous allez créer une vue sur la classe Client qui prend uniquement l'attribut "Adresse", sans tenir compte des autres attributs que peut contenir la classe Client.

Vue de données Custo...

Chemin de composant de vue:

View Specification Source:

	Action
- [icon] Vue de données Customer	+
+ [icon] Customer (EN)	
+ [icon] Customer Id (EN)	[checkmark]
+ [icon] Customer Name (EN)	[checkmark]
+ [icon] Address (EN)	[checkmark]
+ [icon] Client [Account (EN)]	[checkmark] [refresh]
+ [icon] Customer (EN) [Order (EN)]	[checkmark] [refresh]

View Component:

	Action
- [icon] Vue de données Customer	
+ [icon] Address (EN)	[X] [magnifying glass]

A partir des objets sources (arbre de gauche), vous pouvez définir dans la vue des composants embarqués et des composants référencés.

Un composant embarqué indique que toutes les informations qui composent l'objet source sont à prendre en compte lors de l'exploitation de la vue (par exemple les parties et les attributs associés à une classe). Un composant référencé ne fait que référencer l'objet dans la vue.

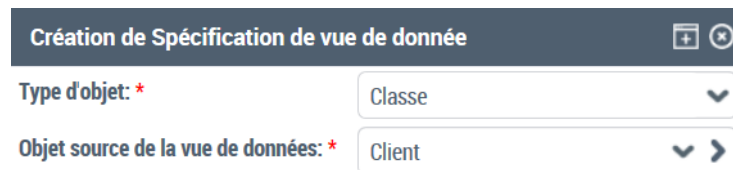
Créer une vue de données (à partir de la liste des vues)

Pour créer une vue de données dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture** > **Dictionnaires de données**.

 Dans **Hopex Data Architecture**, cliquez sur le menu de navigation **Dictionnaires** > **Vue hiérarchique**.

2. Dans la zone d'édition, dépliez le dossier **Dictionnaires de données**.
3. Faites un clic droit sur le nom du dictionnaire concerné et cliquez sur **Nouveau** > **Vue de données**.
La fenêtre de création d'une vue de données apparaît.
4. Pour spécifier l'objet source de la vue de données, cliquez sur **Nouveau**.
5. Dans la fenêtre qui apparaît indiquez :
 - le **Type d'objet** sur lequel porte la vue.
 - L'**Objet source de la vue de données**.



6. Cliquez sur **Ajouter**.
7. Répétez la procédure pour ajouter éventuellement d'autres objets sources.
8. Cliquez sur **OK**.
La nouvelle vue apparaît dans la liste des vues de données.

Créer une vue de données à partir d'un objet

Vous pouvez définir l'objet source d'une vue en créant la vue directement sur l'objet en question.

 Il est possible par la suite d'ajouter un autre objet à la vue.

Pour créer une vue de données sur un objet :

1. Faites un clic droit sur l'objet en question et sélectionnez **Nouveau** > **Vue de données**.
L'assistant de création d'une vue de donnée apparaît.
2. Saisissez le nom de la vue.
3. Indiquez éventuellement le détenteur.
4. Cliquez sur **OK**.
L'éditeur de vue apparaît.

Affichage des objets sources dans la vue de données



Classe



Attribut



Partie



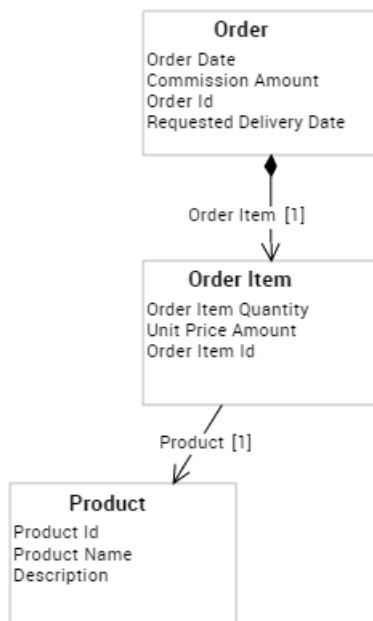
**Partie (classe
composante)**



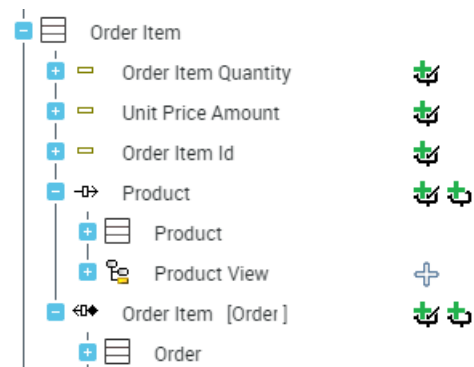
**Partie (classe
composée)**

Exemple

Modèle logique



Vue de données logiques



Définir les composants de la vue de données

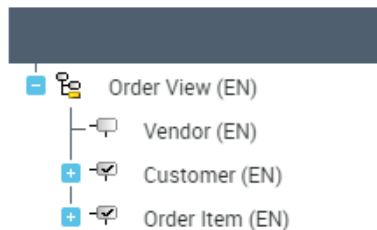
Composant embarqué

Un composant embarqué permet de prendre dans la vue toutes les informations qui composent l'objet (par exemple les parties et les attributs associés à une classe).

Pour ajouter un composant embarqué à la vue :

1. Ouvrez la vue de données.
2. Du côté des objets sources, sélectionnez l'élément à ajouter à la vue de données.
3. Sous la colonne **Action**, cliquez sur le bouton  **Ajouter un composant d'inclusion de vue.**
L'objet apparaît dans la partie droite de l'éditeur de vue.

View Component:



Composant référencé

Référencer un composant dans la vue permet d'afficher l'objet dans la vue, sans embarquer toutes ses propriétés.

Vous pouvez référencer dans la vue les objets qui contiennent un certain nombre d'informations comme les classes. Pour les attributs, seul le bouton d'inclusion est proposé.

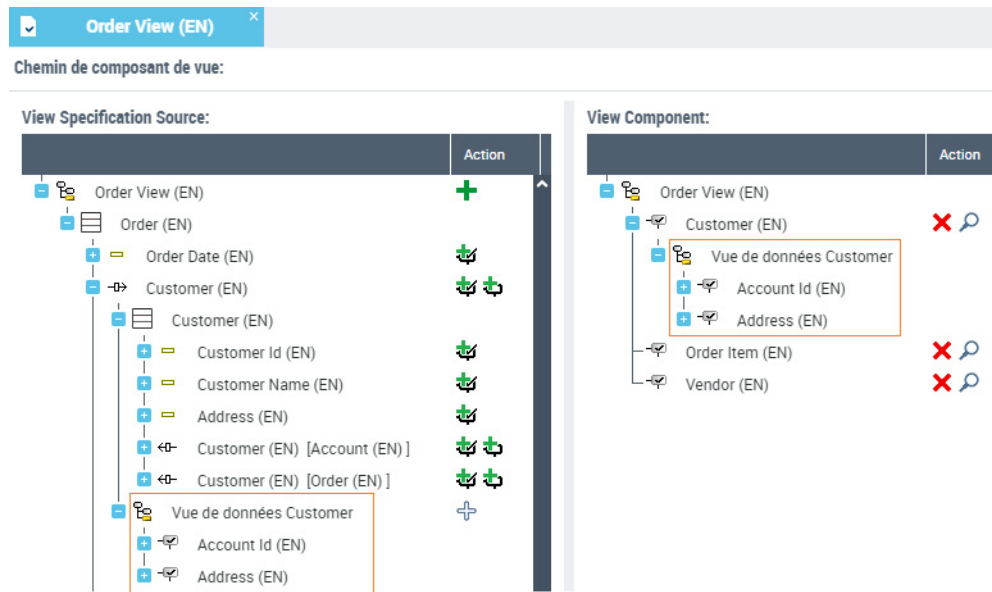
Pour référencer un objet dans la vue :

1. Ouvrez la vue de données.
2. Du côté des objets sources, sélectionnez l'élément à ajouter à la vue de données.
3. Sous la colonne **Action**, cliquez sur le bouton  **Ajouter un composant de référencement de vue.**
L'objet apparaît dans la partie droite de l'éditeur de vue.

Utiliser une vue dans une autre vue

Lorsque vous embarquez une classe dans une vue de données, tous les attributs de la classe sont ajoutés par défaut dans la vue. Vous pouvez limiter la liste des attributs à ceux déjà définis dans une vue.

Ci-dessous, seuls les attributs définis dans la vue "Customer" (Account Id et Address) sont ajoutés à la vue "Order".



Pour ajouter une vue de données (source) à une vue de données (cible) :

1. Ouvrez la vue de données cible.
2. Sous la partie gauche, déployez la classe sur laquelle porte la vue source à ajouter.

👉 La classe source a été préalablement embarquée dans la vue cible.

3. Sélectionnez la vue source et sous la colonne **Action**, cliquez sur  **Ajouter une vue.**

La vue associée à la classe apparaît dans la partie droite de l'éditeur de vue, sous le nom de la classe en question.

DIAGRAMME DE CLASSES

Hopex Data Governance et **Hopex Data Architecture** mettent à votre disposition deux formalismes pour décrire les données logiques :

- le paquetage de données (représenté par un dictionnaire de données), pour construire les diagrammes de classes (notation UML)
- le modèle de données, pour les diagrammes de données (notations standard, IDEF1X, I.E, Merise). Voir [Modèle de données](#).

La description des données en notation UML s'effectue dans un diagramme de classes.

Créer un paquetage

Un paquetage (ou dictionnaire de données) partitionne le domaine d'étude et les travaux associés. Il est destiné à contenir les éléments modélisés. La représentation graphique de tout ou partie de ces éléments se fait dans un diagramme de classes.

Une base de données peut être reliée dès sa création à un paquetage. C'est sur cette base de données que pourront se lancer par la suite les différents outils de traitements des données (génération, synchronisation etc.). Le paquetage de la base de données est le détenteur par défaut des classes et associations représentées dans le diagramme de classes.

Voir aussi : [Dictionnaire de données](#).

Créer un paquetage (dictionnaire de données)

Pour créer un paquetage dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture** > **Dictionnaires de données**.

 Dans **Hopex Data Architecture**, cliquez sur le menu de navigation **Dictionnaires** > **Vue hiérarchique**.

2. Dans la zone d'édition, faites un clic droit sur le dossier **Dictionnaires de données** et cliquez sur **Nouveau** > **Paquetage**.
3. Saisissez le nom du paquetage.
4. Cliquez sur **OK**.

Relier un paquetage à une base de données

Pour créer un paquetage à partir d'une base de données :

1. Cliquez sur le menu de navigation **Architecture** > **Bases de données**.

 Dans **Hopex Data Architecture**, cliquez sur le menu de navigation **Bases de données** > **Vue hiérarchique**.

2. Dans la zone d'édition, sous le dossier **Bases de données**, sélectionnez la base de données concernée.
3. Cliquez sur le bouton **Propriétés** associé.

4. Dans les propriétés de la base de données, cliquez sur la page **Caractéristiques**.
5. Dépliez la section **Paquetage de données**.
6. Cliquez sur **Nouveau**.
7. Saisissez le nom du paquetage et cliquez sur **OK**.

Pour relier un paquetage existant à une base de données :

1. Dans les propriétés de la base de données, cliquez sur la page **Caractéristiques**.
2. Dépliez la section **Paquetage de données**.
3. Cliquez sur **Relier**.
La fenêtre de recherche apparaît.
4. Cliquez sur le bouton **Rechercher**.
5. Sélectionnez le paquetage et cliquez sur **OK**.

Créer un diagramme de classes

Un diagramme de classes est une représentation graphique permettant de représenter la structure statique d'un système, en particulier les types d'objets manipulés dans le système, leur structure interne et les relations qui existent entre eux.

Un diagramme de classes inclut :

- Des classes, qui représentent les concepts de base (client, compte, produit, etc.).
- Des parties, qui définissent les relations entre les différentes classes.
- Des attributs, qui décrivent les caractéristiques des classes.
- Des opérations, qui peuvent être effectuées sur les objets de la classe.

☛ *Les opérations ne sont pas prises en compte par les outils de **Hopex Data Architecture** (synchronisation, génération etc.).*

Pour créer le diagramme de classes d'un paquetage dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture > Dictionnaires de données**.

☛ *Dans **Hopex Data Architecture**, cliquez sur le menu de navigation **Dictionnaires > Vue hiérarchique**.*

2. Dans la zone d'édition, dépliez le dossier **Dictionnaire de données**.
3. Déplacez la souris sur le dictionnaire et cliquez sur le bouton **Créer un diagramme**  qui apparaît à droite.
4. Sélectionnez le type de diagramme **Diagramme de classes**.
5. Cliquez sur **OK**.

Le nouveau diagramme de classes s'ouvre.

Notez que lorsque vous créez un paquetage à partir d'une base de données, un diagramme de classes est créé automatiquement en même temps.

Plus de détails sur la construction d'un diagramme de classes, voir [Le diagramme de classes](#).

TYPES DE DONNÉES

Un type de données permet de mettre en commun des caractéristiques communes à plusieurs attributs. Les types de données sont implémentés sous forme de classe.

Un paquetage de types de données est un paquetage de référence détenant tout ou partie des types de données utilisés dans l'entreprise. Chacun des autres paquetages sera déclaré client du paquetage de référence des types de données.

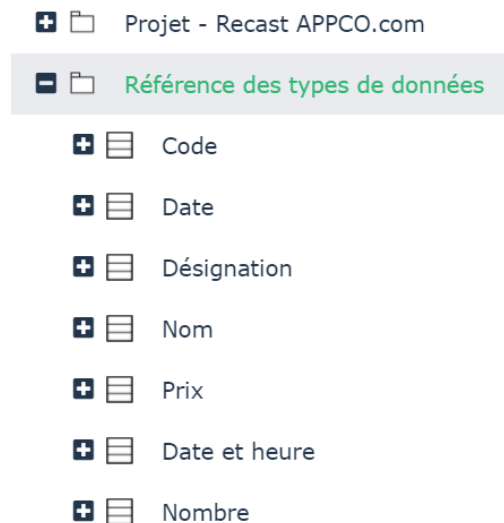
Paquetage de types de données

Un type de données définit la nature des valeurs que peut prendre une donnée. Il peut être simple (entier, caractère, text, booléen, date...) ou plus élaboré et composé.

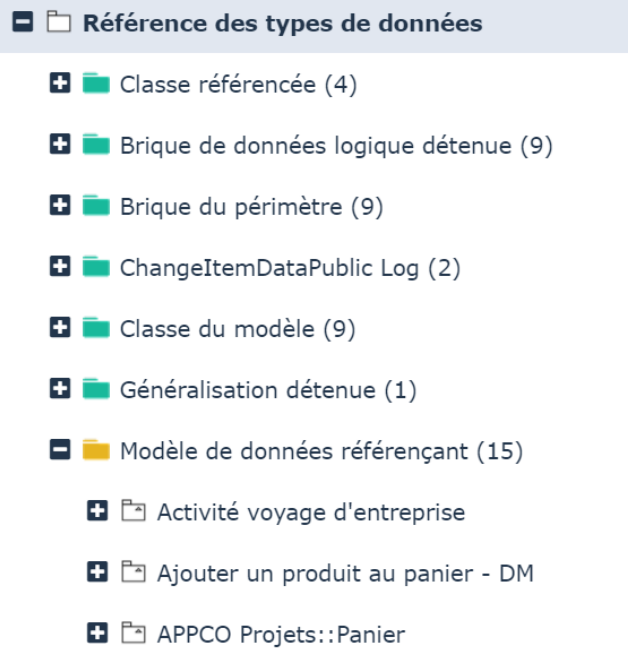
Pour typer les attributs d'une entité, ne sont proposés que les types de données définis pour le modèle de données qui contient cette entité.

Lorsque vous créez un modèle de données, le paquetage de types de données "Référence des types de données" lui est automatiquement associé par défaut.

Ce paquetage "Référence des types de données" détient les types de données "Adresse", "Code", "Date", etc.



En ouvrant l'explorateur sur ce paquetage de type de données, vous pouvez voir qu'il est référencé par plusieurs modèles de données.



Les attributs des entités de ces modèles peuvent donc être typés à l'aide des types de données "Adresse", "Code", "Date", etc.

Créer un nouveau paquetage de types de données

Vous pouvez définir un nouveau paquetage de types de données de référence détenant les types de données utilisés dans l'entreprise.

Pour créer votre propre paquetage de types de données dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture > Dictionnaires de données**.

➡ Dans **Hopex Data Architecture**, cliquez sur le menu de navigation **Dictionnaires > Vue hiérarchique**.

2. Dans la zone d'édition, faites un clic droit sur le dossier **Dictionnaires de données** et cliquez sur **Nouveau**.
3. Tapez le nom du paquetage et cliquez sur **OK**.
4. Ouvrez les propriétés du paquetage.
5. Cliquez sur la page **Caractéristiques**.
6. Sélectionnez le stéréotype "Datatype Package".

Vous pouvez ensuite ajouter des types à ce paquetage.

Créer un type de données

Pour créer un type de données :

1. Ouvrez les **Propriétés** du paquetage.
2. Cliquez sur la page **Type de données**.
3. Cliquez sur **Nouveau**.
La fenêtre de création d'un type de données apparaît.
4. Saisissez le nom du type et cliquez sur **OK**.

Type de données composé

Vous pouvez créer des types de données composés en leur ajoutant une liste d'attributs, par exemple un type "Adresse" composé du numéro, de la rue, du code postal, de la ville et du pays.

Valeur littérale

Vous pouvez affecter des valeurs littérales à un type de données qui définissent les valeurs qu'il peut prendre. Les attributs basés sur un tel type de données ne peuvent prendre que les valeurs définies par le type de données.

Une fois le nouveau paquetage de types de données créé, il convient de le référencer sur le modèle de données client.

Référencer un paquetage de types de données

Pour relier un paquetage de types de données à un dictionnaire de données ou à un modèle de données :

1. Ouvrez les propriétés du dictionnaire ou du modèle de données.
2. Cliquez sur la page **Caractéristiques**.
3. Sous la section **Référence**, sélectionnez à l'aide de la première flèche **Paquetage des types de données**.
4. Avec la deuxième flèche, cliquez sur **Relier paquetage des types de données**.
La fenêtre de recherche apparaît.
5. Cliquez sur le bouton **Chercher**.
6. Sélectionnez le paquetage voulu et cliquez sur **OK**.

Affecter des types aux attributs

Une fois le paquetage de types de données référencé pour un dictionnaire ou un modèle de données, la liste des types qu'il contient est disponible sur chaque attribut des classes du dictionnaire ou entités du modèle. Il vous reste à sélectionner celui qui convient.

Pour définir le type d'un attribut :

1. Ouvrez les **Propriétés** de la classe ou de l'entité concernée.
2. Cliquez sur la page **Attributs**.

3. Dans la colonne **Type de données (MD)** correspondant à l'attribut, sélectionnez le type voulu dans la liste.

Voir aussi [Types des données et Datatypes des colonnes](#).

MODÈLE DE DONNÉES

Modéliser les données consiste à identifier les entités considérées d'intérêt pour représenter l'activité de l'entreprise, et définir les associations qui existent entre elles. Il faut que les entités et les associations qui constituent le diagramme de données associé à un secteur de l'entreprise suffisent à le décrire complètement sur le plan sémantique. En d'autres termes, on doit pouvoir décrire l'activité de l'entreprise en utilisant seulement ces entités et associations.

Ceci n'implique pas que, pour chaque mot ou verbe utilisé pour cette explication, il y ait un correspondant direct dans le diagramme de données. Il s'agit de pouvoir traduire ce que l'on veut exprimer, au travers des entités et des associations.

La spécification des modèles de données est souvent considérée comme la partie la plus importante dans la modélisation d'un système d'information.

Pour vous aider à décrire les données logiques, **Hopex Data Governance** et **Hopex Data Architecture** propose une notation simple, basée sur le modèle de données.

Synthèse des concepts

Modèle de données

Le modèle de données permet de représenter la structure statique d'un système, en particulier les types d'objets manipulés dans le système, leur structure interne et les relations qui existent entre eux.

Un modèle de données regroupe un ensemble d'entités avec leurs attributs, les associations qui existent entre ces entités, des contraintes qui portent sur ces entités et associations, etc.

Diagramme de données

Un diagramme de données est une représentation graphique du modèle ou d'une partie du modèle.

Un diagramme de données est représenté par :

- des **entités**, qui représentent les concepts de base (client, compte, produit, etc.).
- des **associations**, qui définissent les relations entre les différentes entités.
- des **attributs**, qui décrivent les caractéristiques des entités et, dans certains cas, des associations.

L'attribut ou l'ensemble d'attributs qui permet d'identifier de façon unique l'entité est appelé identifiant.

Le diagramme de données est complété avec la définition des multiplicités.

Construire un modèle de données

Les modèles de données sont disponibles avec les profils suivants :

- **Hopex Data Governance** : Concepteur de données et Administrateur fonctionnel de la gouvernance des données
- **Hopex Data Architecture** : Architecte de données et Administrateur fonctionnel de données

Condition préalable

Pour utiliser les modèles de données, vous devez cocher l'option correspondante :

1. Dans le bureau, cliquez sur le menu **Menu principal > Paramètres > Options.**
2. Dans l'arbre de navigation des options, dépliez le dossier **HOPEX Solutions > Fonctionnalités communes liées aux données.**
3. Cliquez sur **Formalisme des données.**
4. Dans la partie droite de la fenêtre cochez la notation Modèles de données.
5. Cliquez sur **OK.**

Créer le modèle de données

Pour créer un modèle de données dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture > Modèles de données.**

➡ Dans **Hopex Data Architecture**, cliquez sur le menu de navigation **Modèles de données.**

2. Dans la zone d'édition, cliquez sur l'onglet **Vue hiérarchique.**
3. Cliquez sur le bouton **Nouveau.**
4. Saisissez le nom du modèle de données, et éventuellement un détenteur.
5. Cliquez sur **OK.**

Le modèle de données créé apparaît dans la liste des modèles de données.

Créer un diagramme de données

Un diagramme de données est une représentation graphique du modèle ou d'une partie du modèle.

Un diagramme de données est représenté par :

- des **entités**, qui représentent les concepts de base (client, compte, produit, etc.).
- des **associations**, qui définissent les relations entre les différentes entités.
- des **attributs**, qui décrivent les caractéristiques des entités et, dans certains cas, des associations.

L'attribut ou l'ensemble d'attributs qui permet d'identifier de façon unique l'entité est appelé identifiant.

Le diagramme de données est complété avec la définition des multiplicités.

Pour créer un diagramme de données :

1. Déplacez la souris sur le modèle de données et cliquez sur le bouton **Créer un diagramme**  qui apparaît à droite.
2. Sélectionnez le type de diagramme **Structure de données**.
Le diagramme apparaît.

Les type de données

Un type de données permet de mettre en commun des caractéristiques communes à plusieurs attributs.

Lorsque vous créez un modèle de données, le paquetage de types de données "Référence de types de données" lui est automatiquement relié par défaut. Ainsi, la liste des *types de données* qu'il contient est disponible sur chaque attribut des entités du modèle. Mais vous pouvez lui attribuer un autre *paquetage de types de données*.

Le paquetage de données de référence d'un modèle de données est visible dans la fenêtre de propriétés du modèle, sous la page **Caractéristiques**.

Pour plus d'informations, voir [Paquetage de types de données](#).

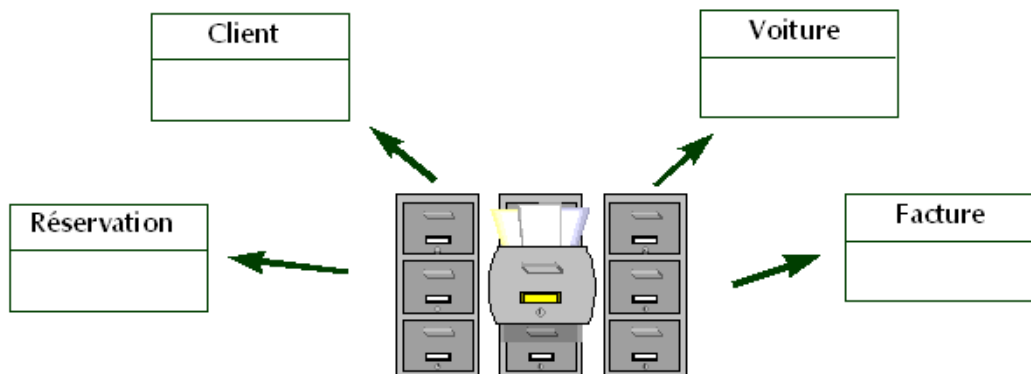
Les entités

 Une entité est un regroupement d'objets possédant des caractéristiques communes et un comportement semblable. Les entités sont des éléments de gestion considérés d'intérêt pour représenter l'activité de l'entreprise et sont donc généralement conservés à cet effet. Elles pourront, par exemple, donner lieu à des tables dans une base de données.

Une *entité* est décrite par une liste d'attributs.

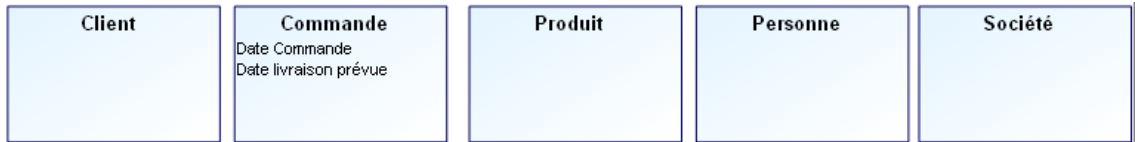
Une entité est reliée à d'autres entités via des associations. L'ensemble des entités et des associations constitue le noyau d'un diagramme de données.

Nous pouvons illustrer la notion d'entité en comparant les entités à des fiches dans des tiroirs.



Les entités peuvent représenter des objets de gestion.

Exemples : Client, Commande, Produit, Personne, Société, etc.



Les entités peuvent représenter des objets techniques utilisés dans l'industrie.

Exemples : Alarme, Capteur, Zone

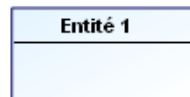
Créer une entité

Pour créer une entité :

1. Dans la barre d'insertion du diagramme de données, cliquez sur le bouton **Entité** .
2. Cliquez sur le plan de travail du diagramme. La fenêtre **Ajout d'une entité** s'ouvre.
3. Saisissez le **Nom** de l'entité.

☛ Quand le bouton **OK** ou **Ajouter** est grisé, c'est que la fenêtre où il apparaît n'a pas été complètement renseignée.

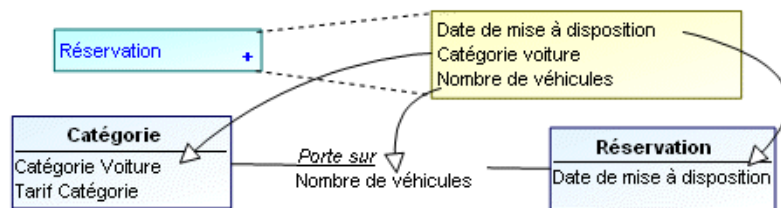
4. Cliquez sur **Ajouter**.
L'entité apparaît dans le diagramme.



😊 Vous pouvez créer plusieurs entités à la suite sans revenir à la barre d'outil en double-cliquant sur le bouton **Entité**. Pour revenir ensuite au mode normal, utilisez la touche <Echap>, ou cliquez sur la flèche de la barre d'outils.

Les attributs

Les entités et les associations peuvent être caractérisées par des *attributs*.



Ces attributs ont pu, par exemple, être révélés par l'étude du contenu des messages qui circulent à l'intérieur de l'entreprise.

 *Un attribut est une propriété nommée d'une classe.*

Exemples :

- "Nom du client" (attribut de l'entité client).
- "N° client" (identifiant de l'entité client).
- "Solde du compte" (attribut de l'entité compte).

Un attribut caractérise une association quand l'attribut dépend de l'ensemble des entités participant à l'association.

Dans l'exemple présenté ci-après, le rôle qu'un "Consultant" a joué sur un "Contrat" dépend du consultant et du contrat, donc de l'association "Intervenir".



Créer un attributs

Pour définir un attribut sur une entité :

1. Cliquez avec le bouton droit sur l'entité et sélectionnez **Propriétés**.
La fenêtre des propriétés de l'entité s'ouvre.
2. Cliquez sur la liste déroulante puis sur **Attributs**.
La page des attributs apparaît.
3. Cliquez sur le bouton **Nouveau**.
Un nom vous est automatiquement proposé pour ce nouvel attribut. Vous pouvez le modifier.
4. Cliquez sur **OK**.

Vous pouvez préciser son **Type de données**.

Exemple : Numérique.

 Voir [Types des données et Datatypes des colonnes](#) pour plus de détails sur les **types de données** qui peuvent être affectés à un attribut.

Attributs hérités

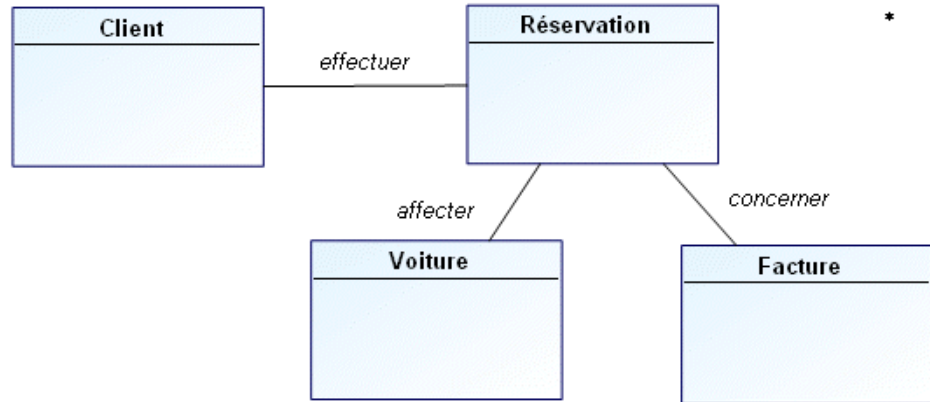
Lorsqu'une généralisation existe entre une entité générale et une entité particulière, l'entité particulière hérite des attributs de l'entité générale.

Voir [Généralisations](#).

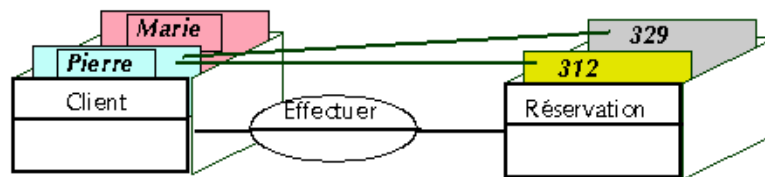
Les associations

 *Une association est une relation existant entre deux classes.*

Les *associations* peuvent être comparées à des liens entre des fiches.



Le dessin suivant permet de visualiser "en trois dimensions" les situations qu'un diagramme de données permet de mémoriser.



Pierre et Marie sont des clients. Pierre a effectué les réservations numéros 312 et 329.

Un diagramme de données doit permettre de mémoriser toutes les situations du contexte de l'entreprise, mais rien que celles-là.

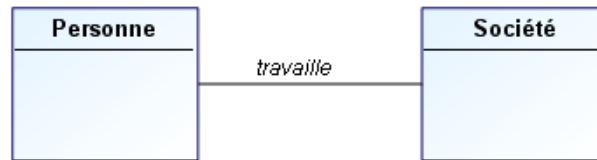
☛ *Le diagramme ne doit pas permettre de représenter des situations irréalistes ou aberrantes.*

Exemples d'association :

- Un client passe une commande.
- Une commande comprend plusieurs produits.



- Une personne travaille pour une société.



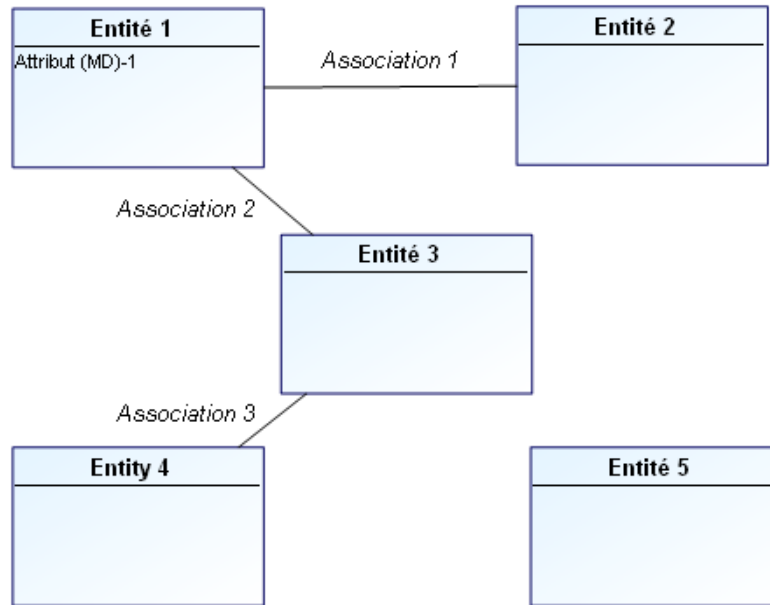
- Une alarme est déclenchée par un capteur.
- Un capteur couvre une zone.
- Une fenêtre affiche une chaîne de caractères.

Créer une association

Pour créer une association :

1. Dans la barre d'insertion du diagramme de données, cliquez sur le bouton **Association** .
2. Cliquez sur une des entités concernées et, en gardant le bouton de la souris enfoncé, déplacez le pointeur jusqu'à la deuxième entité, avant de relâcher votre pression. L'association apparaît sous forme d'un trait.
3. Pour préciser le nom de l'association, cliquez sur le bouton droit sur l'association et sélectionnez **Propriétés**.
⚠ Veillez à bien cliquer sur le trait qui matérialise l'association, et non sur un des rôles situés aux extrémités de l'association.
4. Dans la page **Caractéristiques**, dans le champ **Nom Local**, saisissez le nom de l'association.
5. Cliquez sur **OK**.

Exemple



En cas d'erreur, vous pouvez supprimer un élément ou un lien en sélectionnant la commande **Supprimer** du menu contextuel de l'élément ou du lien.

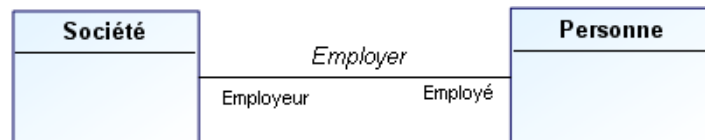
Définir le rôle des associations



Un rôle permet d'indiquer une des entités concernées par l'association. L'indication des rôles est particulièrement importante dans le cas d'une association entre une entité et elle-même.

Chaque extrémité d'une association permet de préciser le **rôle** joué par chaque entité dans l'association.

Visuellement, le nom du rôle se distingue du nom d'une association, car il est placé près de son extrémité. De plus, il apparaît en caractères droits, alors que le nom de l'association est en italique.



La barre d'état (située au bas de la fenêtre) permet aussi de distinguer les différentes zones : lorsque vous déplacez la souris le long de l'association, elle indique si vous vous trouvez sur l'association ou sur un rôle.

Lorsque deux entités sont reliées par une seule association, le nom des entités suffit souvent à caractériser le rôle ; nommer les rôles prend tout son intérêt lorsque plusieurs associations relient deux entités.

Multiplicités

Chaque rôle d'une association porte une indication de multiplicité qui montre combien d'objets de l'entité considérée peuvent être liés à un objet de l'autre entité. La multiplicité est une information portée par le rôle, sous la forme d'une expression entière bornée. On l'indique en particulier pour chacun des rôles que jouent les entités dans une association.

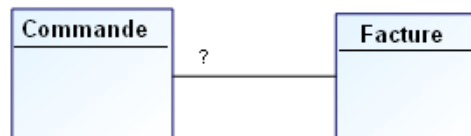
La multiplicité exprime le nombre de participations minimum et maximum d'un objet donné d'une entité à une association.

Les multiplicités usuelles sont "1", "0..1", "*" ou "0..*", "1..*", et "M..N" où "M" et "N" sont des entiers :

- La multiplicité "1" indique que chaque objet de l'entité est relié par cette association une fois et une seule.
- La multiplicité "0..1" indique qu'un objet de l'entité ne peut être relié par cette association qu'une fois au plus.
- La multiplicité "*" ou "0..*" indique qu'un objet de l'entité peut être relié par l'association une ou plusieurs fois ou pas du tout.
- La multiplicité "1..*" indique que chaque objet de l'entité est obligatoirement relié par l'association et qu'il peut l'être plusieurs fois.
- La multiplicité "M..N" indique que chaque objet de l'entité est obligatoirement relié par l'association au moins "M" fois et qu'il peut l'être au maximum "N" fois.

1	Un et un seul
0..1	Zéro ou un
M..N	De M à N (entiers naturels)
*	De zéro à plusieurs
0..*	De zéro à plusieurs
1..*	De un à plusieurs

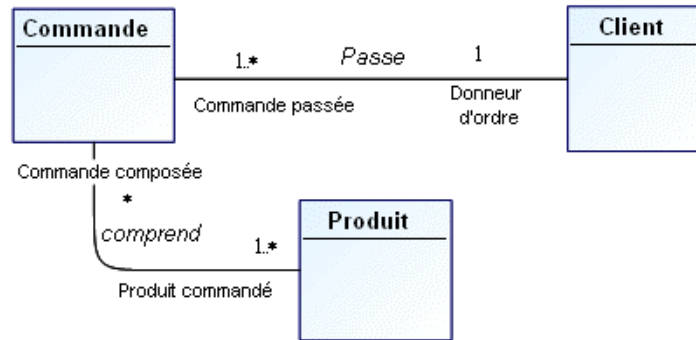
Exemple :



0..1 A une commande correspond une facture au maximum ou aucune.

- * Aucune restriction n'est imposée sur le nombre de factures correspondant à une commande.
- 1 A chaque commande correspond une facture et une seule.
- 1..* A chaque commande correspond une ou plusieurs factures.

Autres exemples de multiplicité :

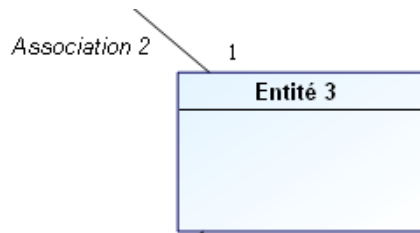


- 1..* Un client peut passer une ou plusieurs commandes.
- 1 Une commande est passée par un et un seul client.
- 1..* Une commande comprend un ou plusieurs produits.
- * Un produit peut faire partie de plusieurs commandes, ou d'aucune.
- 0..1 Une personne travaille pour une société.
- 1..* Une alarme est déclenchée par un ou plusieurs capteurs.
- 1 Un capteur couvre une et une seule zone.
- 1..* Une fenêtre affiche une ou plusieurs chaînes de caractères.

Pour préciser la multiplicité d'un rôle :

1. Dans le diagramme de données, cliquez avec le bouton droit sur le trait qui se trouve entre l'association et l'entité, afin d'ouvrir le menu contextuel du rôle.
2. Cliquez sur **Propriétés**.
La fenêtre de propriétés du rôle s'ouvre.
3. Cliquez sur liste déroulante puis sur **Caractéristiques**.
4. Dans le champ **Multiplicité**, sélectionnez la multiplicité voulue.

La représentation de l'association change en fonction de la nouvelle valeur de ses multiplicités.

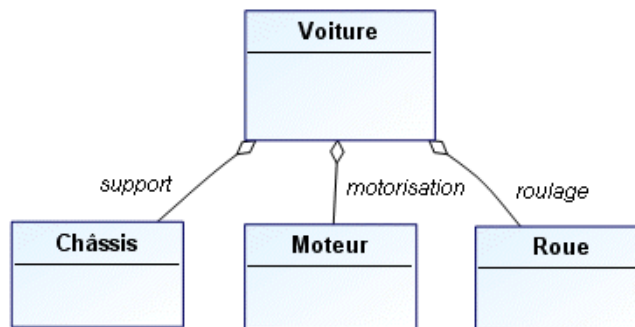


Agrégation

L'agrégation est une forme particulière d'association qui indique que l'une des entités contient l'autre.

Exemple d'*agrégation* :

Une voiture comprend un châssis, un moteur et des roues.



Pour définir l'agrégation entre les entités "Voiture" et "Moteur" :

1. Cliquez avec le bouton droit sur le rôle joué par l'entité "Voiture" dans son association avec l'entité "Moteur" et sélectionnez **Propriétés**.
Les propriétés du rôle apparaissent.
2. Cliquez sur **Caractéristiques**.
3. Dans le champ **Tout/partie**, sélectionnez "Agrégat".
Un losange représentant l'agrégation apparaît alors sur le rôle.

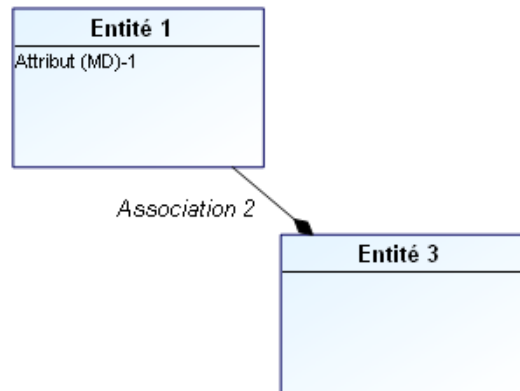
Composition

La composition est une agrégation forte pour laquelle la durée de vie des composants coïncide avec celle du composé. Une composition est une agrégation immuable avec une multiplicité 1.

Exemple de *composition* :

Une commande est composée de plusieurs lignes de commande qui n'existent plus si la commande est supprimée.

La composition est matérialisée par un losange noir.

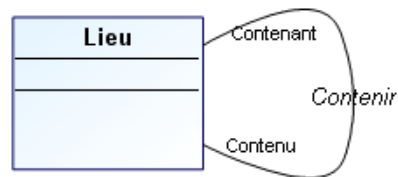


Pour préciser la composition d'un rôle :

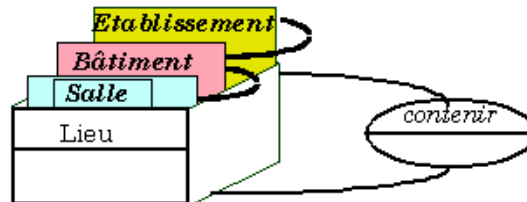
1. Cliquez avec le bouton droit sur le rôle et sélectionnez **Propriétés**. Les propriétés du rôle apparaissent.
2. Cliquez sur la liste déroulante puis sur **Caractéristiques**.
3. Dans le champ **Tout/partie**, sélectionnez "Composé".

Associations réflexives

Certaines associations mettent en jeu plusieurs fois la même entité.



Une salle de classe, un bâtiment, un établissement scolaire sont tous des lieux.



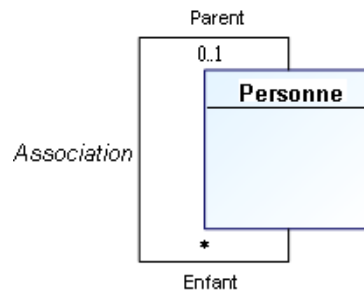
Une salle de classe est contenue dans un bâtiment, lui-même contenu dans un établissement scolaire.

Une association réflexive porte deux fois sur la même entité.

Pour créer une association réflexive :

1. Dans la barre d'insertion du diagramme de données, cliquez sur le bouton **Association** 
2. Cliquez dans l'entité concernée et faites glisser la souris en dehors de cette entité, puis revenez-y ; relâchez ensuite le bouton de la souris. L'association réflexive apparaît sous forme d'un demi-cercle en ligne brisée.

☛ Dans le cas d'une association entre une entité et elle-même, il est indispensable de préciser les rôles afin de distinguer les liens correspondants dans le dessin.



Ci-dessus "Parent" et "Enfant" sont les deux *rôles* joués par l'entité "Personne" dans l'association.

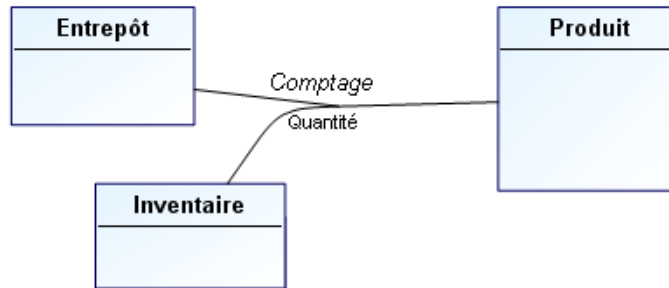
📖 Un rôle permet d'indiquer une des entités concernées par l'association. L'indication des rôles est particulièrement importante dans le cas d'une association entre une entité et elle-même.

Vous pouvez segmenter une ligne en y ajoutant des sommets pour modifier le tracé des traits. Vous pouvez en particulier segmenter un rôle, pour lui faire contourner un obstacle par exemple. Pour cela, cliquez sur la ligne en tenant la touche <Ctrl> enfoncée et déplacez cette ligne.

Association "plus que binaire"

Certaines associations associent non pas deux, mais davantage d'entités. Ces associations sont, en principe, rares.

Exemple : Lors d'un inventaire, une certaine quantité de produit a été comptée dans chaque entrepôt.



Pour créer une association ternaire :

1. Dans le diagramme de données, créez l'association entre deux des entités.
2. Cliquez sur le bouton **Rôle de l'association**  et reliez la troisième entité à l'association.

Les contraintes



Une contrainte représente un contrôle ou une règle de gestion qui doit être appliquée lors de l'exécution d'un traitement.

La plupart des **contraintes** impliquent les associations entre les entités.

Exemples de contraintes :

La personne responsable d'un service doit appartenir à ce service.

Toute commande facturée doit avoir été préalablement livrée.

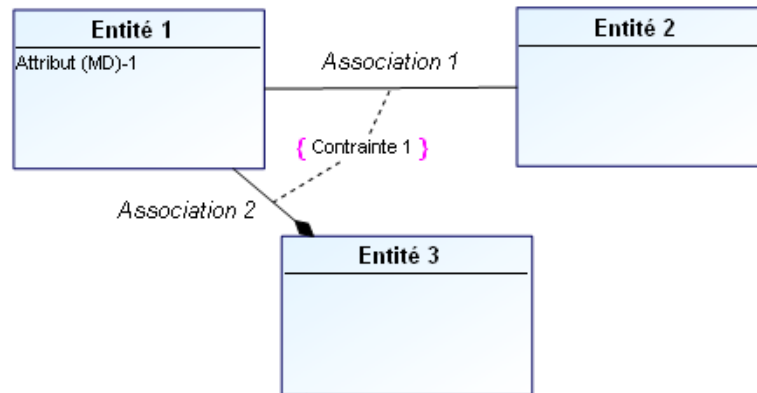
La date de livraison doit être postérieure à la date de commande.

Un capteur couvrant une zone ne peut déclencher qu'une alarme protégeant cette même zone.

Pour créer une contrainte :

1. Dans la barre d'insertion du diagramme, cliquez sur le bouton **Contrainte** .
2. Cliquez sur une des associations concernées par la contrainte et faites glisser la souris jusqu'à la deuxième association, avant de relâcher votre pression.
La fenêtre **Ajout d'une contrainte** s'ouvre.

3. Saisissez le nom de la contrainte puis cliquez sur **Ajouter**.
La contrainte apparaît dans le diagramme.



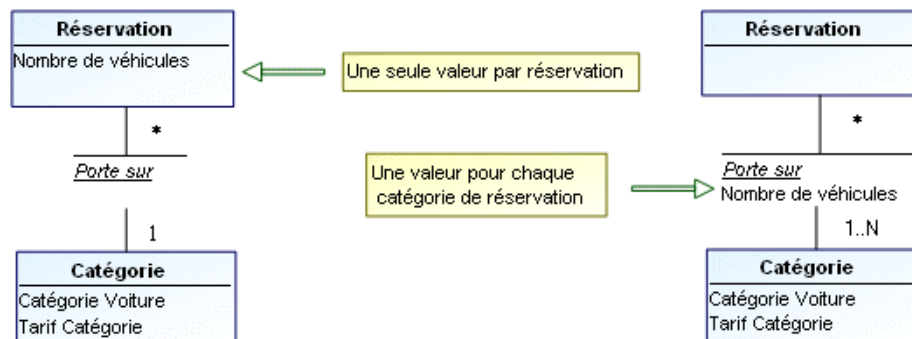
😊 Enregistrez régulièrement votre dessin à l'aide du bouton **Enregistrer**

Règles de normalisation

Les formes normales sont des règles qui visent à éviter des erreurs de modélisation. A ce jour, il existe de six à sept formes normales. Nous allons voir les trois premières.

Première forme normale

Règle : La valeur d'un attribut est fixée de manière unique dès que l'on connaît le ou les objets concernés.

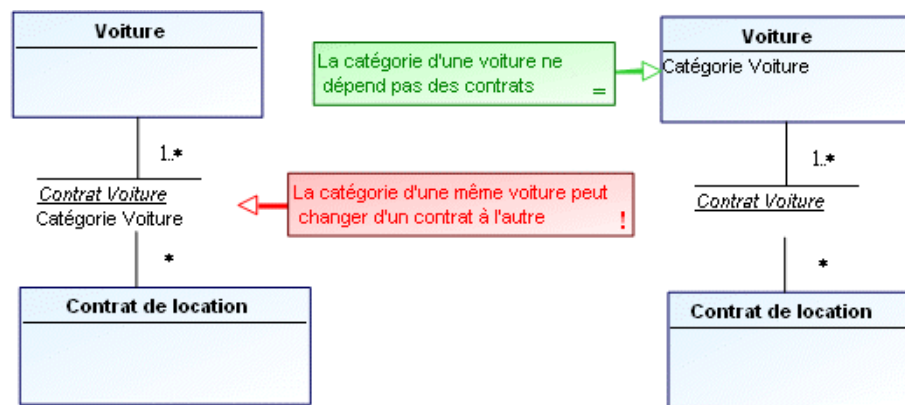


Si le nombre de véhicules est porté par l'entité "Réservation", on ne peut indiquer que le nombre total de véhicules pour une réservation. On doit donc faire une réservation par catégorie de véhicule loué (multiplicité 1).

Si le nombre de véhicules est porté par l'association, on peut préciser le nombre de véhicules réservés pour chaque catégorie sur l'association. On peut donc faire une seule réservation pour plusieurs catégories de véhicules (multiplicité 1..N).

Deuxième forme normale

Règle : La valeur d'un attribut d'association n'est fixée que lorsque l'on connaît toutes les entités concernées.

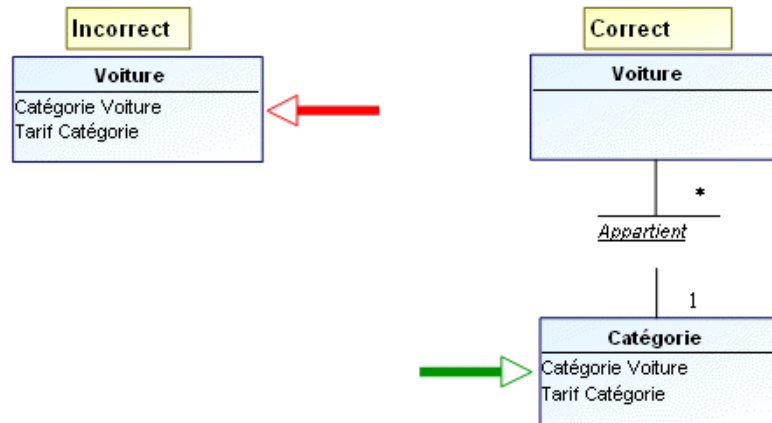


Le fait que la catégorie de voiture porte sur l'association "Catégorie Voiture" suppose que la catégorie de la voiture puisse changer d'un contrat à l'autre, ce qui ne serait pas très honnête.

Pour que la catégorie de la voiture ne dépende pas du contrat, il faut qu'elle soit portée par l'entité "Voiture".

Troisième forme normale

Règle : Un attribut doit dépendre directement et uniquement de l'entité qu'il décrit.



Si le "Tarif Catégorie" est porté par l'entité "Voiture", cela signifie que deux voitures de la même catégorie peuvent avoir un "Tarif Catégorie" différent.

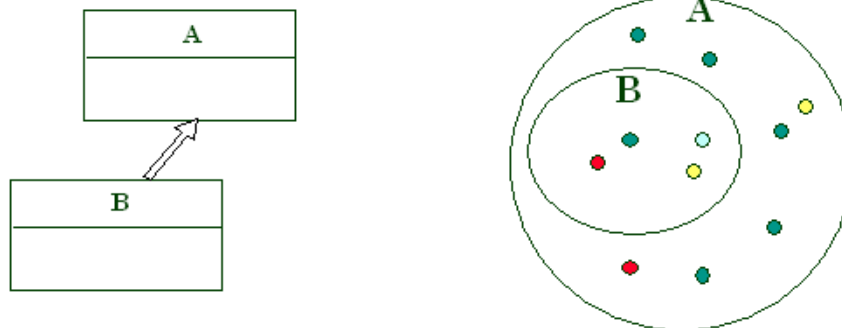
Pour éviter cela, il faut créer une entité "Catégorie" qui portera le tarif.

☺ Cette règle permet de faire émerger des concepts qui n'apparaissent pas dans la première ébauche de diagramme de données.

Généralisations

Qu'est-ce qu'une généralisation ?

📖 Une généralisation représente une relation d'héritage entre une entité générale et une entité plus spécifique. L'entité spécifique est cohérente avec l'entité plus générale et hérite de ses caractéristiques et de son comportement. Elle peut cependant comporter des attributs ou des associations supplémentaires. Tout objet de l'entité spécifique est aussi un objet de l'entité générale.



L'entité A est une *généralisation* de l'entité B. Cela suppose que tous les objets de l'entité B sont aussi des objets de l'entité A. Autrement dit, B est un sous-ensemble de A. B est alors la sous-entité, A la super-entité.

Exemple :

A : Personne, B : Parisien.

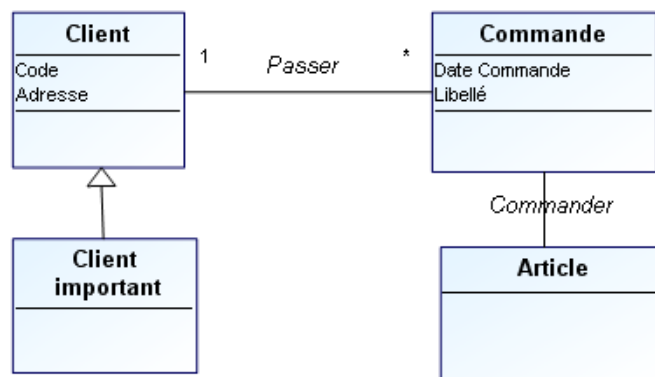
B étant un sous-ensemble de A, les objets de l'entité B "héritent" des caractéristiques de ceux de l'entité A.

Il n'est donc pas nécessaire de décrire de nouveau pour l'entité B :

- Ses attributs
- Ses associations

Exemple :

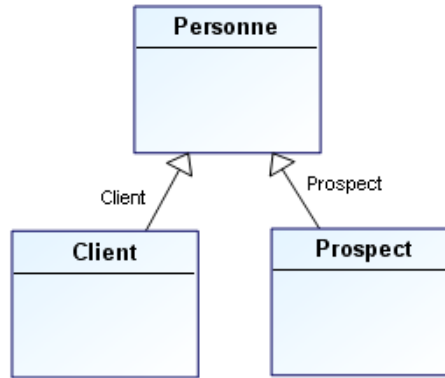
L'entité "Client important" qui représente les clients dont le "C.A. sur les 12 derniers mois" dépasse 1 million d'euros, peut être une spécialisation de l'entité "Client".



Dans l'exemple qui précède, les associations et les attributs spécifiés pour "Client" sont aussi valables pour "Client important".

Autres exemples de généralisations :

"prospect" et "client" sont deux sous-entités de "personne".



"commande export" est une sous-entité de l'entité "commande".

"personne physique" et "personne morale" sont deux sous-entités de l'entité "personne".

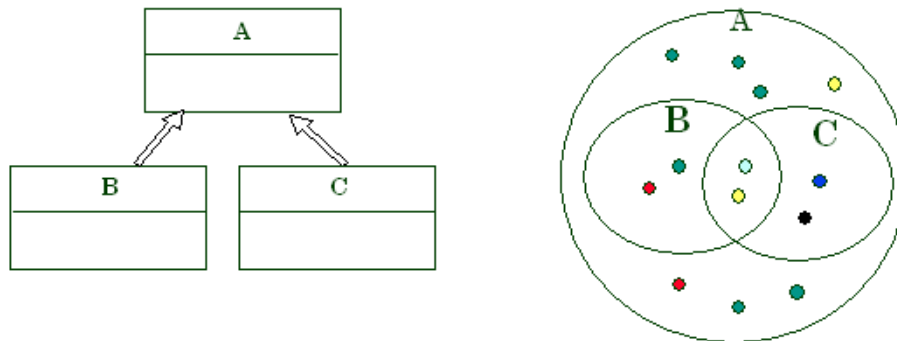
"polygone", "ellipse" et "cercle" sont des sous-entités de l'entité "forme".

"chêne", "orme", et "bouleau" sont des sous-entités de l'entité "arbre".

"véhicule à moteur", "véhicule tout-terrain" et "véhicule amphibie" sont des sous-entités de l'entité "véhicule".

"camion" est une sous-entité de l'entité "véhicule à moteur".

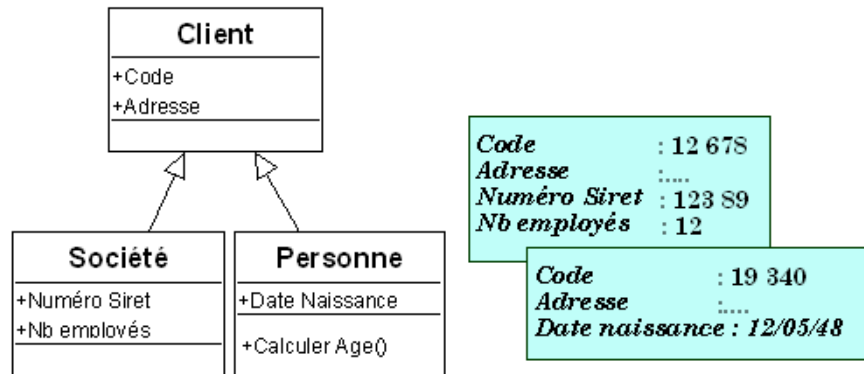
Cas de plusieurs sous-entités



Plusieurs sous-entités d'une même entité :

- Ne sont pas forcément exclusives.
- Ne forment pas nécessairement une partition.

Intérêt des sous-entités

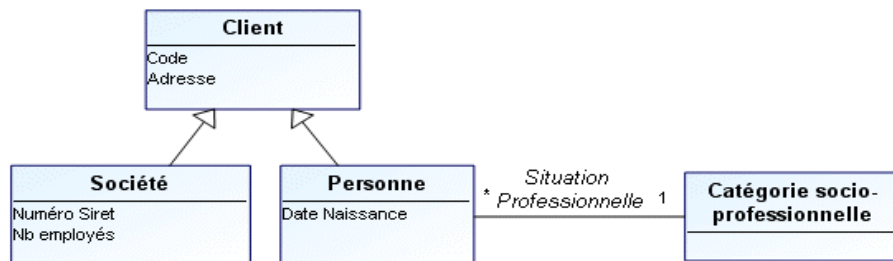


Une sous-entité hérite de tous les attributs et associations de sa super-entité, mais elle peut avoir des attributs ou des associations que ne possède pas sa super-entité.

Une sous-entité peut ainsi avoir des attributs spécifiques. Ceux-ci n'ont de sens que pour une sous-entité particulière. Dans l'exemple ci-dessus :

- Le "numéro de Siret" et le "nombre d'employés" n'ont de sens que pour une "société".
- La "date de naissance" est caractéristique d'une "personne", pas d'une "société".

Une sous-entité peut également avoir des associations spécifiques.



- Une "personne" entre dans une "catégorie socio-professionnelle" : "cadre", "employé", "commerçant", "agriculteur", etc. Cette classification n'a pas de sens pour une "société" (il existe également une classification pour les entreprises, mais ce n'est pas la même que pour les personnes).

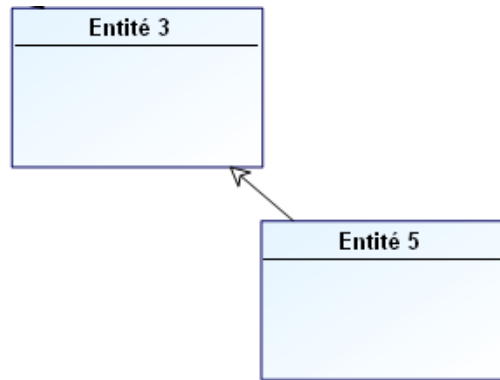
Héritage multiple

Il est parfois utile de spécifier une entité ayant plusieurs super-entités. La sous-entité hérite alors de toutes les caractéristiques des deux super-entités. Cette possibilité doit être utilisée avec précaution.

Créer une généralisation

Pour créer une *généralisation* :

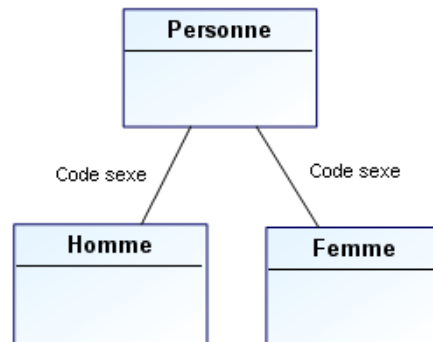
1. Dans la barre d'insertion du diagramme de données, cliquez sur le bouton **Généralisation** 
2. Cliquez dans la sous-entité, par exemple "Entité 5", et faites glisser la souris jusqu'à l'entité générale, par exemple "Entité 3", avant de relâcher votre pression.
La généralisation apparaît dans le diagramme, matérialisée par une flèche.



Discriminant

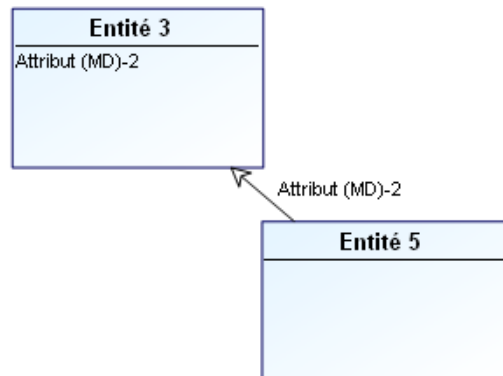
Le discriminant est l'attribut de l'entité générale dont la valeur permet de répartir les objets entre les sous-entités associées à la généralisation.

Par exemple, l'attribut code-sexe permet de répartir les objets de l'entité personne entre la sous-entité homme ou femme.



Pour définir un discriminant sur une généralisation :

1. Ouvrez les propriétés de la généralisation.
 2. Cliquez la liste déroulante puis sur **Caractéristiques**.
 3. Dans le champ **Discriminant**, cliquez sur la flèche puis sur **Relier Attribut (MD)**.
 4. Recherchez et sélectionnez le discriminant parmi les attributs de la super-entité.
- Une fois sélectionné, le discriminant s'affiche sur la généralisation.



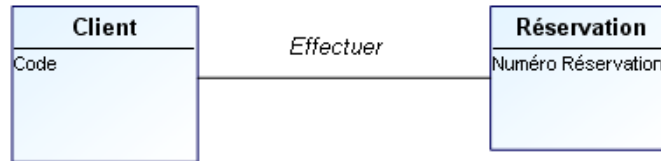
☞ Il est également possible de préciser si une généralisation est **Complète** : dans ce cas toutes les instances de la super-entité appartiennent à au moins une des sous-entités de cette généralisation.

Identifiant d'une entité

Tout objet possède une identité qui caractérise son existence propre. L'*identifiant* permet de distinguer tout objet d'une entité de façon non ambiguë. Cela permet,

entre autres, de distinguer deux objets dont toutes les valeurs d'attribut sont identiques.

 *Un identifiant est constitué d'un ou plusieurs attributs ou rôles obligatoires qui permettent d'identifier de façon unique une entité.*



Le client dont le code est 2718 effectue la réservation numéro 314159.

Chaque entité possède un identifiant unique dont la valeur permet de retrouver sans ambiguïté chacun de ses exemplaires.

Par défaut, l'identifiant est implicite. Dans ce cas, une clé primaire sera générée automatiquement à partir du nom de l'entité.

Identification par un attribut

Il est possible de choisir un des attributs de l'entité comme identifiant. Pour cela :

1. Ouvrez les propriétés de l'entité.
2. Cliquez sur la page **Attributs**.
La liste des attributs apparaît.
3. Pour l'attribut choisi, sélectionnez la valeur "Oui" dans la colonne **Identifiant**.

Correspondance des modèles de données

La modélisation de données reflète l'activité d'une entreprise et s'appuie sur l'historique du métier. Les différences constatées entre les modèles sont généralement d'ordre culturel ou liées à des conventions qui varient d'une personne à l'autre et dans le temps. Aussi, dans l'expression d'un besoin métier, le modélisateur doit tenir compte de l'existant et réussir à concilier différentes visions d'une même réalité.

La correspondance des modèles de données facilite l'alignement de ce patrimoine hétérogène sur un socle sémantique commun.

Objectifs fonctionnels

Distinguer les définitions d'entreprise et les données métiers

Pour assurer la cohérence des données métiers, les modélisateurs peuvent se référer à des définitions d'entreprise servant de cadre de référence.

La correspondance de modèles de données établit une frontière entre les définitions de niveau entreprise et les données métier tout en assurant une traçabilité. L'outil

Dictionnaire complète cette approche en permettant de constituer un vocabulaire métier sur la base d'un dictionnaire.

Intégrer les modèles existants

Les modèles existants qui décrivent un patrimoine applicatif doivent être pris en compte lors de l'élaboration de nouveaux modèles ou dans un projet de refonte. Les besoins varient selon les cas d'emplois :

- Démarche de type "As-is to-be" : l'évolution d'un modèle de données est progressive et s'appuie sur un état de référence stable, qui correspond généralement aux données du système en production.
- Mise en place progiciel : chaque progiciel (PGI, CRM, etc.) impose son modèle de données, ce qui accroît la tendance au morcellement et au cloisonnement du SI. D'où le besoin de disposer d'un modèle indépendant, lié aux différents modèles imposés.

La correspondance de modèles de données est un moyen de rapprocher les modèles de données de différentes sources.

Cas d'emploi

Un cas typique de mise en correspondance de données intervient dans le cadre d'échanges entre des applications disposant de leurs propres modèles de données. Lorsque le nombre d'applications devient trop important, vous pouvez mettre en place un modèle pivot de référence qui va servir d'intermédiaire entre les applications et éviter ainsi la multiplication des correspondances.

Lancer l'éditeur de correspondances

L'éditeur de correspondances est l'outil qui permet d'aligner deux modèles de données ou de mettre en correspondance les vues logiques et physiques d'une base de données. Il se compose d'un arbre de correspondances qui juxtapose les vues des deux modèles.

Vous pouvez lancer l'éditeur de correspondances depuis :

- Le menu de navigation d'Hopex
- Un modèle de données
- Un paquetage de données
- Une base de données

Pour lancer l'éditeur de correspondance à partir du menu de navigation :

1. Cliquez sur le menu de navigation **Outils > Éditeur de correspondances**. Une fenêtre apparaît.
2. Laissez l'option cochée par défaut **Créer un arbre de correspondances**, et cliquez sur **Suivant**.
3. Indiquez le nom du nouvel arbre de correspondance.
4. Dans le champ **Nature**, précisez le type de correspondance effectué.
5. Dans les cadres **Objet gauche** et **Objet droit**, à partir des types d'objet concernés, sélectionnez les modèles que vous souhaitez aligner.
6. Cliquez sur **OK**.
L'éditeur affiche l'arbre de correspondance qui juxtapose les deux modèles.

Une fois l'arbre de correspondances créé, vous pouvez le retrouver ultérieurement dans l'éditeur de correspondances.

Créer une correspondance

Pour créer une correspondance entre deux objets :

1. Dans l'éditeur de correspondances, sélectionnez successivement les deux objets concernés.
2. Cliquez sur le bouton **Créer un élément de correspondance**.

La correspondance se crée à partir du dernier objet sélectionné.

Supprimer une correspondance

Pour supprimer une correspondance sur un objet :

1. Cliquez sur l'objet en question et cliquez sur le bouton **Suppression d'un élément de correspondance**.

Détails des correspondances

Les objets qui ont une correspondance sont cochés en vert. Lorsque vous sélectionnez l'un de ces objets dans l'arbre de navigation, sa correspondance apparaît dans la fenêtre de détails située par défaut en bas de l'éditeur de correspondances. Elle regroupe le nom des objets reliés, le type des objets et éventuellement un commentaire.

Catalog DB (Logique/Physique)

	Validité	Figé
Catalogue		
Diagramme de données	✓	
Catalogue	✓	
Contact	✓	
Personne	○	

	Validité	Figé
Diagramme relationnel	✓	✗
Catalog	✓	
Contact	✓	
Insurance_Product	✓	
Promotion	✓	

Liste des éléments de correspondance

Validité	Objet logique	Objet physique	Type
✓ Valide	Catalogue	Catalog	Entité - Table

Propriétés d'une correspondance

Pour visualiser les propriétés d'une correspondance :

- 1 Dans la fenêtre de détails de l'éditeur, sélectionnez la correspondance et cliquez sur le bouton **Propriétés**.

Etat des objets

Des indicateurs permettent d'indiquer l'état des objets mis en correspondance.

Les objets peuvent être caractérisés par les états suivants :



Valide



Invalide (Lorsqu'un objet a conservé une correspondance vers un objet qui n'existe plus)



Sans correspondance

Source de la correspondance

En sélectionnant un objet dans l'arbre d'un des modèles présentés dans l'éditeur, vous pouvez retrouver sa correspondance dans l'autre modèle.

Pour afficher la correspondance d'un objet :

1. Sélectionnez l'objet en question.
S'il a une correspondance, celle-ci s'affiche en bas de l'éditeur.
2. Sélectionnez la correspondance et cliquez sur le bouton **Chercher** 
Les deux objets en correspondances apparaissent en gras dans l'éditeur.

Exemple de correspondance entre modèles de données

Différents niveaux de modélisation peuvent couvrir des besoins distincts. Prenons l'exemple de deux modèles de données. Un modèle de données métier "Gestion des commandes (MD)" se situe au niveau conceptuel. Il décrit au niveau métier comment doivent être gérées les commandes.

Au niveau logique, le modèle de données "Gestion des commandes (Agence)" présente une vision opérationnelle des données du système d'informations propre à chaque agence.

On retrouve des concepts identiques dans chacun des modèles. Il s'agit cependant d'objets distincts.

Vous pouvez mettre en correspondance ces deux modèles de données afin de favoriser la cohésion entre les besoins métiers et les systèmes qui les supportent.

Pour cela :

1. Ouvrez l'**Editeur de correspondance**.
2. Créez un arbre de correspondance.
3. Sélectionnez les deux modèles à aligner.

4. Cliquez sur **OK**.
L'éditeur affiche l'arbre de correspondance qui juxtapose les deux modèles.
5. Créez les correspondances entre les objets similaires puis enregistrez.

Une fois les modèles mis en correspondance, vous pouvez savoir quel objet logique est rattaché à un objet métier. Vous pouvez de même analyser l'impact des changements effectués au niveau métier sur le niveau opérationnel et inversement.

LA NOTATION IDEF1X

A propos de la modélisation des données avec IDEF1X

Modéliser les données consiste à identifier les objets de gestion (entités) et les associations ou relations entre ces objets, considérés d'intérêt pour représenter l'activité de l'entreprise.

IDEF1X est utilisé pour produire un modèle graphique qui représente la sémantique et la structure de l'information manipulée à l'intérieur d'une entreprise ou d'un système. L'utilisation de ce standard permet la construction d'un modèle sémantique qui peut servir de support à la gestion des données en tant que ressource, à l'intégration des systèmes d'information, ainsi qu'à la construction de bases de données informatisées.

Un des principaux objectifs est de servir de support à l'intégration des systèmes. L'approche de l'intégration avec IDEF1X est focalisée sur la saisie, la gestion et l'utilisation d'une seule définition sémantique des données appelée "schéma conceptuel". Le "schéma conceptuel" fournit une définition des données manipulées à l'intérieur de l'entreprise qui est unique, n'est pas biaisée par une seule application de ces données et qui est indépendante de la façon dont les données sont stockées physiquement ou de la façon dont on y accède. Le principal objectif de ce schéma conceptuel est de fournir une définition cohérente de la signification des données et de leurs interrelations qui puisse être utilisée pour intégrer ces données, les partager, et gérer leur intégrité. Un schéma conceptuel doit posséder trois caractéristiques essentielles :

- Il doit être cohérent avec l'infrastructure de l'entreprise et doit être valide dans tous ses domaines d'application.
- Il doit être extensible, de façon à ce que chaque nouvelle donnée puisse être définie sans avoir à modifier les données précédemment définies.
- Il doit être adaptable aussi bien aux différentes visions des utilisateurs qu'à une large variété de structures de stockage et d'accès aux données.

Les éléments de base d'un modèle IDEF1X sont :

- Les objets au sujet desquels les données sont conservées, c-à-d les personnes, lieux, idées, événements, etc., qui sont représentés par des boîtes ;
- Les relations entre ces objets, représentées par des lignes qui relient ces boîtes ; et
- Les caractéristiques de ces objets représentées par les noms des attributs à l'intérieur des boîtes.

Synthèse des concepts

Dans **Hopex Data Governance** et **Hopex Data Architecture**, un modèle de données (IDEF1X) est représenté par :

- Des entités, qui représentent les concepts de base (client, compte, produit, etc.).
- Des associations, qui définissent les associations entre les différentes entités.
- Des attributs qui décrivent les caractéristiques des entités.

L'attribut qui permet d'identifier de façon unique l'entité est appelé identifiant.

Le modèle de données est complété par la définition des multiplicités (ou cardinalités).

Construire un modèle de données (IDEF1X)

La modélisation des données avec IDEF1X est disponible avec les profils suivants :

- **Hopex Data Governance** : Concepteur de données et Administrateur fonctionnel de la gouvernance des données
- **Hopex Data Architecture** : Architecte de données et Administrateur fonctionnel de données

Condition préalable

Pour utiliser la notation IDEF1X, vous devez cocher l'option correspondante :

1. Dans le bureau, cliquez sur le menu **Menu principal** > **Paramètres** > **Options**.
2. Dans l'arbre de navigation des options, dépliez le dossier **HOPEX Solutions** > **Fonctionnalités communes**.
3. Cliquez sur **Notation des données**.
4. Dans la partie droite de la fenêtre cochez la notation IDEF1X.
5. Cliquez sur **OK**.

Créer le modèle de données

Pour créer un modèle de données dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture** > **Modèle de données**.
2. Dans la fenêtre d'édition, cliquez sur **Vue hiérarchique**.

☛ Dans **Hopex Data Architecture**, cliquez sur le menu de navigation **Modèles de données**.

3. A droite du dossier **Hiérarchie des modèles de données**, cliquez sur le bouton **+ Nouveau**.
La fenêtre de création d'un modèle de données apparaît.
4. Saisissez le nom du modèle.
5. Cliquez sur **OK**.
Le modèle de données apparaît dans la liste des modèles de données.

Diagramme de données (IDEF1X)

Un diagramme de données est une représentation graphique du modèle ou d'une partie du modèle.

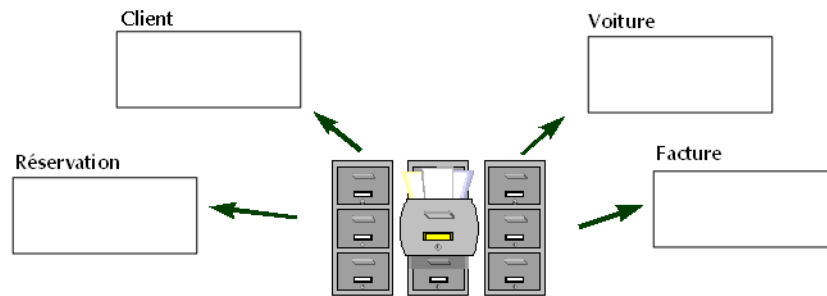
Pour créer un diagramme de données :

1. Déplacez la souris sur le modèle de données et cliquez sur le bouton **Créer un diagramme**  qui apparaît à droite.
2. Sélectionnez le type de diagramme **Diagramme de données (IDEF1X)**.
Le diagramme apparaît.

Entités (IDEF1X)

 Une entité est un regroupement d'objets possédant des caractéristiques communes et un comportement semblable. Les entités sont des éléments de gestion considérés d'intérêt pour représenter l'activité de l'entreprise et sont donc généralement conservés à cet effet. Elles pourront, par exemple, donner lieu à des tables dans une base de données.

Nous pouvons illustrer la notion d'*entité* par une comparaison, par exemple, des fiches dans des tiroirs.



Une entité représente une classe particulière d'objets, dont tous les exemplaires peuvent être décrits de la même manière.

Une entité est "indépendante" si chaque instance de l'entité peut être définie de manière unique sans connaître ses relations avec d'autres entités. Une entité est "dépendante" si l'identification de manière unique d'une de ses instances dépend de ses relations avec d'autres entités.

Une entité est représentée par une boîte. Si l'entité est "dépendante", les angles de la boîte sont arrondis.

Créer une entité

Pour créer une entité :

1. Cliquez sur le bouton **Entité**  de la barre d'objets du diagramme.
2. Cliquez sur le plan de travail du diagramme.
La fenêtre **Ajout d'une entité** s'ouvre.
3. Saisissez le nom de l'entité.
4. Cliquez sur **Créer** (Windows Front-End) ou **Ajouter** (Web Front-End).
L'entité apparaît dans le diagramme.

Attributs

 *Un attribut représente un type de caractéristique ou une propriété associée à un ensemble d'objets abstraits ou concrets. Une instance d'entité a généralement une valeur spécifique pour chacun des attributs. Une combinaison d'un ou de plusieurs attribut(s) peut être utilisée comme identifiant quand elle permet d'identifier de manière unique chacune des instances d'une entité.*

Exemples d'*attribut* :

- "Nom du client" (attribut de l'entité client).
- "N° client" (identifiant de l'entité client).
- "Solde du compte" (attribut de l'entité compte).

Définir les attributs

Pour créer un attribut :

1. Cliquez avec le bouton droit sur l'entité et sélectionnez **Propriétés**.
La fenêtre des propriétés de l'entité s'ouvre.
2. Cliquez sur l'onglet **Attributs**.
3. Pour ajouter un nouvel attribut à l'entité, cliquez sur le bouton  .
Un nom vous est automatiquement proposé pour ce nouvel attribut. Vous pouvez le modifier.

Vous pouvez préciser son **Type de données**.

Exemple : Numérique.

 *Un type de données permet de mettre en commun des caractéristiques communes à plusieurs attributs. Les types de données sont implémentés sous forme de classe.*

 Voir [Types des données et Datatypes des colonnes](#) pour plus de détails sur les **types de données** qui peuvent être affectés à un attribut.

Attributs hérités

Lorsqu'une relation de catégorisation (généralisation) existe entre une entité générale et une entité particulière, l'entité particulière hérite des attributs de l'entité générale.

Voir [Généralisations](#).

Préciser l'identifiant d'une entité

Pour préciser l'identifiant d'une entité :

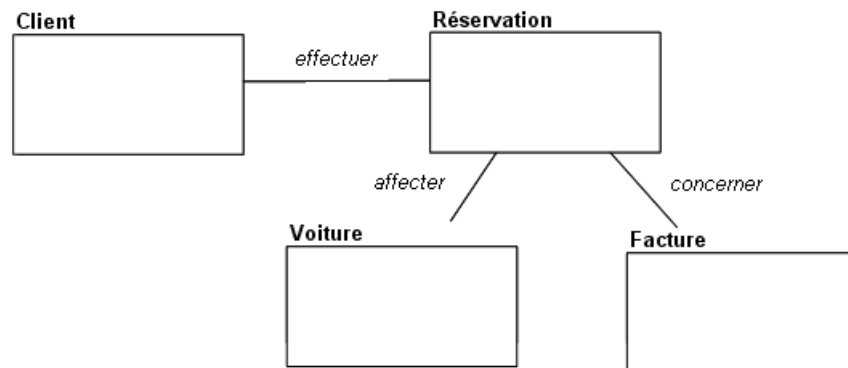
1. Ouvrez la fenêtre de propriétés de l'entité.
2. Cliquez sur l'onglet **Attributs**.
3. Pour l'attribut choisi sélectionnez la valeur "Oui" dans la colonne **Identifiant**.

➡ Pour plus de détails, voir [Identifiant d'une entité](#).

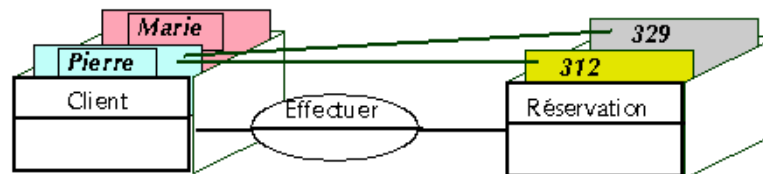
Associations (IDEF1X)

📖 Une association est une relation existant entre deux classes.

Les **associations** peuvent être comparées à des liens entre des fiches.



Le dessin suivant permet de visualiser "en trois dimensions" les situations qu'un diagramme de données permet de mémoriser.



Pierre et Marie sont des clients. Pierre a effectué les réservations numéros 312 et 329.

Un diagramme de données doit permettre de mémoriser toutes les situations du contexte de l'entreprise, mais rien que celles-là.

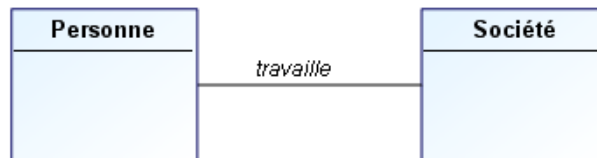
➡ Le diagramme ne doit pas permettre de représenter des situations irréalistes ou aberrantes.

Exemples d'association :

- Un client passe une commande.
- Une commande comprend plusieurs produits.



- Une personne travaille pour une société.



- Une alarme est déclenchée par un capteur.
- Un capteur couvre une zone.
- Une fenêtre affiche une chaîne de caractères.

Relation identifiante obligatoire

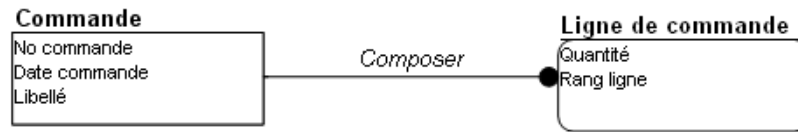
 Une relation identifiante obligatoire est une association entre des entités dans laquelle chaque instance de la première entité est associée à zéro, une ou plusieurs instances de la seconde entité et chaque instance de la seconde entité est associée à une instance de la première entité et est identifiée par cette association. La seconde entité est toujours une entité dépendante représentée par une boîte aux angles arrondis. Une relation identifiante est représentée par une ligne pleine avec un point du côté de l'entité dépendante.

Si une instance d'entité est identifiée par son association avec une autre entité, la relation est dite "identifiante", et chaque instance de cette entité doit être associée avec exactement une instance de l'autre entité. Par exemple, si une ou plusieurs tâches sont associées à chaque projet et que les tâches sont identifiées de manière unique à l'intérieur d'un projet, une relation identifiante existe entre les entités "Projet" et "Tâche". C'est-à-dire que le projet associé doit être connu pour pouvoir identifier de manière unique une tâche des autres. L'existence d'un fils dans une relation identifiante est toujours dépendante de celle de son parent, c-à-d qu'une instance de l'entité fils ne peut exister que si elle est reliée à une instance de l'entité parente.

Pour créer une *relation identifiante* :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Relation identifiante obligatoire** .
2. Cliquez sur l'entité parente, et en gardant le bouton de la souris enfoncé, déplacez le pointeur jusqu'à l'entité fille, avant de relâcher votre pression.

L'association apparaît dans le diagramme. Elle est représentée par un trait plein avec un point à l'extrémité du trait du côté de l'entité dépendante. La forme de l'entité dépendante est automatiquement changée en une boîte aux angles arrondis.



Relation obligatoire identifiante

Dans l'exemple ci-dessus, une commande est composée de lignes de commandes, et chaque ligne de commande est identifiée par son association avec la commande. La ligne de commande est une entité dépendante représentée par une boîte aux angles arrondis.

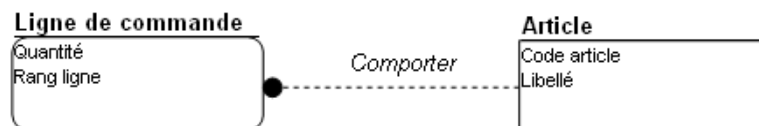
Relation non-identifiante obligatoire

 Une relation non-identifiante obligatoire est une association entre des entités dans laquelle chaque instance de la première entité est associée à zéro, une ou plusieurs instances de la seconde entité et chaque instance de la seconde entité est associée à une instance de la première entité mais n'est pas identifiée par cette association. Elle est représentée par une ligne pointillée avec un point du côté de l'entité dépendante.

Si chaque instance d'une entité peut être identifiée de manière unique sans connaître l'instance de l'autre entité associée, la relation est "non-identifiante". Par exemple, bien qu'il puisse exister une relation de dépendance entre les entités "Acheteur" et "Ordre d'achat", les ordres d'achat peuvent être identifiés de manière unique par un numéro d'ordre d'achat sans nécessairement connaître l'acheteur concerné.

Pour créer une **relation non-identifiante** :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Relation non-identifiante obligatoire** .
2. Cliquez dans l'entité parente, et en gardant le bouton de la souris enfoncé, déplacez le pointeur jusqu'à l'entité fille, avant de relâcher votre pression.
L'association apparaît dans le diagramme.



Relation obligatoire non-identifiante

Dans l'exemple ci-dessus, une ligne de commande inclut un article, mais elle n'est pas identifiée par son association avec cet article.

Relation optionnelle non-identifiante

 Une relation optionnelle est une association entre entités dans laquelle chaque instance de la première entité est associée à zéro, une ou plusieurs instances de la seconde entité et chaque instance de la seconde entité est associée à zéro, une ou plusieurs instances de la première entité. Elle est représentée par une ligne pointillée avec un point du côté de la seconde entité et un petit losange à l'autre extrémité.

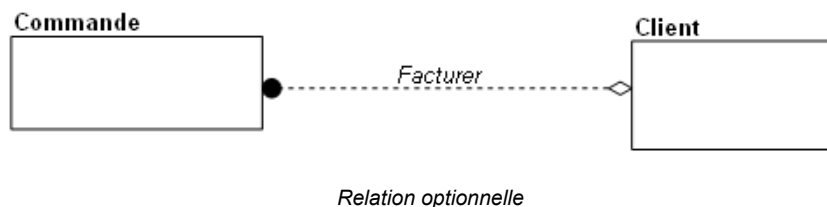
Dans une **relation optionnelle** non-identifiante, chaque instance de l'entité fille est reliée à zéro ou une instance de l'entité parente.

Une relation optionnelle non-identifiante représente une dépendance conditionnelle. Une relation optionnelle non-identifiante est représentée par un trait pointillé entre l'entité parente et l'entité fille, avec un petit losange du côté de l'entité parente.

Une instance de l'entité fille pour laquelle chaque attribut de la clé étrangère qui correspond à la relation possède une valeur, doit être associée à une instance de l'association parente dans laquelle les valeurs des attributs de la clé primaire de l'entité parente doit être égale à la valeur des attributs de la clé étrangère de l'entité fille.

Pour créer une relation optionnelle non-identifiante :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Relation optionnelle** .
2. Cliquez sur l'entité parente, et en gardant le bouton de la souris enfoncé, déplacez le pointeur jusqu'à l'entité fille, avant de relâcher votre pression.
L'association apparaît dans le diagramme.



Dans l'exemple ci-dessus, une commande devrait être facturée à un client, mais ce n'est pas obligatoire (problèmes de livraison, etc.).

Relation non-spécifique

 Une relation non-spécifique est une association entre entités dans laquelle chaque instance de la première entité est associée à zéro, une ou plusieurs instances de la seconde entité et chaque instance de la seconde entité est associée à zéro, une ou plusieurs instances de la première entité. Elle est représentée par une ligne tirée entre les deux entités avec un point à chaque extrémité.

Des **relations non-spécifiques** sont utilisées dans des vues Entité-Relation de haut niveau pour représenter des relations plusieurs-à-plusieurs entre des entités.

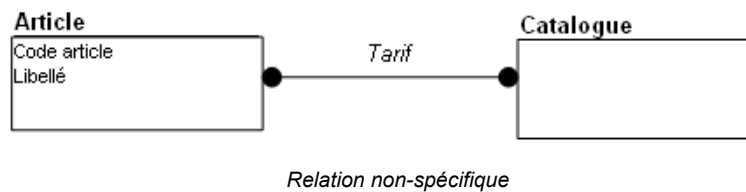
Dans le développement initial d'un modèle, il est souvent pratique d'identifier des "relations non-spécifiques" entre les entités. Ces relations non-spécifiques sont raffinées dans les phases de développement ultérieures du modèle.

Une relation non-spécifique, également appelée "relation plusieurs-à-plusieurs", est une association entre deux entités dans laquelle chaque instance de la première entité est associée avec zéro, une ou plusieurs instances de la deuxième entité et chaque instance de la deuxième entité est associée avec zéro, une ou plusieurs instances de la première entité. Par exemple, si un employé peut être affecté à plusieurs projets et si plusieurs employés peuvent être affectés au même projet, le lien entre les entités "Employé" et "Projet" peut être exprimé par une relation non-spécifique. Cette relation non-spécifique pourra être remplacée par des relations spécifiques plus tard dans le développement du modèle en introduisant une troisième entité, telle que "Affectation Projet", qui est une entité fille commune dans les relations spécifiques avec les entités "Employé" et "Projet". Ces nouvelles relations spécifient que chaque employé est affecté à zéro, un, ou plusieurs projets. Chaque assignement de projet est pour exactement un employé et un projet. Les entités introduites pour résoudre les relations non-spécifiques sont parfois appelées "entités associatives" ou "entités d'intersection".

La définition d'une relation non-spécifique peut ensuite être complétée en spécifiant les cardinalités pour chaque sens de la relation.

Pour créer une relation non-spécifique :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Relation non-spécifique** .
2. Cliquez sur la première entité, et en gardant le bouton de la souris enfoncé, déplacez le pointeur jusqu'à la deuxième entité, avant de relâcher votre pression.
L'association apparaît dans le diagramme.



Dans l'exemple ci-dessus, un article peut apparaître dans zéro, un ou plusieurs catalogues et un catalogue peut contenir zéro, un ou plusieurs articles.

Entité associative



Une entité associative est une entité qui est introduite pour résoudre une relation non-spécifique ou pour afficher des attributs en tant que propriétés d'une association.

Des relations non-spécifiques sont utilisées dans des vues Entité-Relation de haut niveau pour représenter des relations plusieurs-à-plusieurs entre des entités. Dans une vue basée sur les clés ou définie complètement avec ses attributs, toutes les associations entre les entités doivent être exprimées sous forme de relations spécifiques. Cependant, dans le développement initial d'un modèle, il a été souvent

pratique d'identifier des "relations non-spécifiques" entre les entités. Ces relations non-spécifiques sont raffinées dans les phases de développement ultérieures du modèle.

Les entités introduites pour résoudre les relations non-spécifiques sont parfois appelées "entités associatives" ou "entités d'intersection".

Pour créer une *entité associative* :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Entité** .
2. Cliquez sur le plan de travail du diagramme.
La fenêtre **Ajout d'une entité** s'ouvre.
3. Saisissez le Nom de l'entité associative.
4. Cliquez sur **Créer** (Windows Front-End) ou **Ajouter** (Web Front-End).
L'entité apparaît dans le diagramme.
5. Cliquez sur le bouton **Relation identifiante obligatoire** .
6. Cliquez sur la première entité, et en gardant le bouton de la souris enfoncé, déplacez le pointeur jusqu'à l'entité associative, avant de relâcher votre pression.
L'association apparaît dans le diagramme. La forme de l'entité associative est changée en une boîte aux angles arrondis pour indiquer qu'il s'agit d'une entité dépendante.
7. Créez de la même manière la deuxième association en cliquant sur la deuxième entité et, en gardant le bouton de la souris enfoncé, déplacez le pointeur jusqu'à l'entité associative, avant de relâcher votre pression.

☛ Vous pouvez ajouter des attributs à une entité associative.



Entité associative

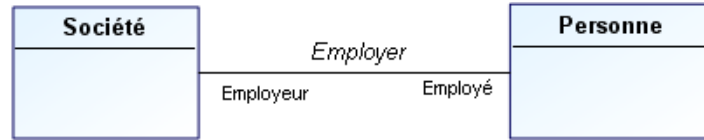
Dans l'exemple ci-dessus, un article peut faire l'objet d'une remise pour zéro, un ou plusieurs clients, et un client peut bénéficier de remises sur zéro, un ou plusieurs articles. Dans chaque cas, le taux de remise est indiqué sur l'entité associative.

Définir les rôles des associations

📖 Un rôle permet d'indiquer une des entités concernées par l'association. L'indication des rôles est particulièrement importante dans le cas d'une association entre une entité et elle-même.

Chaque extrémité d'une association permet de préciser le *rôle* joué par chaque entité dans l'association.

Visuellement, le nom du rôle se distingue du nom d'une association, car il est placé près de son extrémité. De plus, il apparaît en caractères droits, alors que le nom de l'association est en italique.



☺ La barre d'état (située au bas de la fenêtre) permet aussi de distinguer les différentes zones : lorsque vous déplacez la souris le long de l'association, elle indique si vous vous trouvez sur l'association ou sur un rôle.

Lorsque deux entités sont reliées par une seule association, le nom des entités suffit souvent à caractériser le rôle ; nommer les rôles prend tout son intérêt lorsque plusieurs associations relient deux entités.

Certaines associations peuvent mettre en œuvre plus de deux entités. Ces associations sont généralement rares.

Pour ajouter un rôle à une association :

1. Cliquez sur le bouton **Rôle de l'association**  et reliez l'association à l'entité.

Multiplicités

Chaque rôle d'une association porte une indication de multiplicité qui montre combien d'objets de l'entité considérée peuvent être liés à un objet de l'autre entité. La multiplicité est une information portée par le rôle, sous la forme d'une expression entière bornée. On l'indique en particulier pour chacun des rôles que jouent les entités dans une association.

La multiplicité exprime le nombre de participations minimum et maximum d'un objet donné d'une entité à une association.

Les multiplicités usuelles sont "1", "0..1", "*" ou "0..*", "1..*", et "M..N" où "M" et "N" sont des entiers :

- La multiplicité "1" indique que chaque objet de l'entité est relié par cette association une fois et une seule.
Elle est représentée par une relation obligatoire avec un point sur ce rôle et pas de point sur le rôle opposé.
- La multiplicité "0..1" indique qu'un objet de l'entité ne peut être relié par cette association qu'une fois au plus.
Elle est représentée par un "Z" à côté du rôle.
- La multiplicité "*" ou "0..*" indique qu'un objet de l'entité peut être relié par l'association une ou plusieurs fois ou pas du tout.
C'est la valeur par défaut.
- La multiplicité "1..*" indique que chaque objet de l'entité est obligatoirement relié par l'association et qu'il peut l'être plusieurs fois.
Elle est représentée par un "P" (pour Positif) à côté du rôle.
- La multiplicité "M..N" indique que chaque objet de l'entité est obligatoirement relié par l'association au moins "M" fois et qu'il peut l'être au maximum "N" fois.

1	Un et un seul
0..1	Zéro ou un (Z)
M..N	De M à N (entiers naturels)
*	De zéro à plusieurs
0..*	De zéro à plusieurs
1..*	De un à plusieurs (P)

Pour préciser la multiplicité d'un rôle :

1. Cliquez avec le bouton droit sur le trait qui se trouve entre l'association et l'entité, afin d'ouvrir le menu contextuel du rôle.
2. Cliquez sur **Propriétés**.
La page de propriétés du rôle s'ouvre.
3. Cliquez sur l'onglet **Caractéristiques**.
4. Dans le champ **Multiplicité**, sélectionnez la multiplicité voulue.

La représentation de l'association change en fonction de la nouvelle valeur de ses multiplicités.

Relations de catégorisation (généralisations) - (IDEF1X)



Une généralisation représente une relation d'héritage entre une entité générale et une entité plus spécifique. L'entité spécifique est cohérente avec l'entité plus générale et hérite de ses caractéristiques et de son comportement. Elle peut cependant comporter des attributs ou des associations supplémentaires. Tout objet de l'entité spécifique est aussi un objet de l'entité générale.

Qu'est-ce qu'une Catégorisation (Généralisation) ?

Les relations de catégorisation sont utilisées pour représenter des structures dans lesquelles une entité est un "type" (catégorie) d'une autre entité.

Les entités sont utilisées pour représenter des "objets au sujet desquels nous avons besoin d'information". Comme certains objets du monde réel sont des catégories d'autres objets, certaines entités doivent, dans un certain sens, être des catégories d'autres entités. Par exemple, supposons que nous avons besoin d'informations à propos des employés.

Bien que nous disposions d'informations à propos des employés en général, nous pouvons avoir besoin d'informations complémentaires à propos des employés salariés qui soient différentes de celles à propos des employés payés à l'heure. Dans ce cas, les entités "Employé salarié" et "Employé payé à l'heure" sont des catégories de l'entité "Employé". Dans la notation IDEF1X, elles sont reliées par des relations de catégorisation (*généralisation*).

Dans un autre cas, une catégorie d'entité peut être nécessaire pour exprimer une relation qui n'est valide que pour une catégorie particulière, ou pour documenter les différences entre les relations autour des diverses catégories de l'entité. Par exemple, un "Employé à plein temps" peut profiter de la participation aux bénéfices, tandis qu'un "Employé à temps partiel" ne peut pas.

Une "relation de catégorisation" ou "généralisation" est une relation entre une entité, appelée "entité générale" et une autre entité, appelée "catégorie" ou "entité spécialisée". La cardinalité n'est pas spécifiée pour les catégories, car elle est toujours zéro ou un.

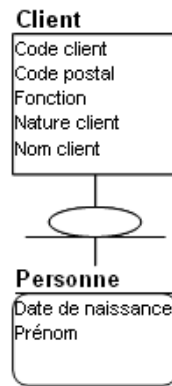
Les catégories sont aussi toujours des entités dépendantes.

Créer une catégorisation

Pour créer une relation de catégorisation :

1. Cliquez sur le bouton **Généralisation**  de la barre d'objets.

2. Cliquez dans l'entité catégorie, et faites glisser la souris jusqu'à l'entité générale, avant de relâcher votre pression.
La généralisation est représentée dans le diagramme par un cercle souligné, relié par un trait à l'entité générale et par un autre à l'entité spécialisée.



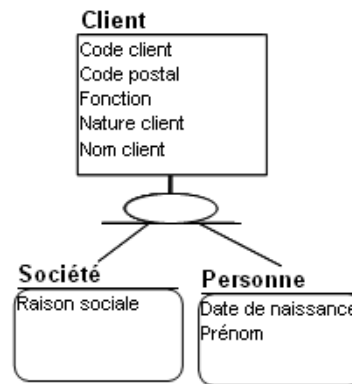
Relation de catégorisation

Dans l'exemple ci-dessus, certains attributs ont un intérêt pour les personnes et n'ont pas de sens pour d'autres catégories de clients. Personne est une entité dépendante représentée par une boîte aux angles arrondis.

Catégories multiples

Un "groupe de catégories" est un ensemble constitué d'une ou de plusieurs relations de catégorisation. Une instance de l'entité générale peut être associée à une instance d'une seule des catégories du groupe, et chaque instance d'une catégorie est associée avec exactement une instance de l'entité générale. Chaque instance de la catégorie représente le même objet du monde réel que l'instance associée de l'entité générale. Dans l'exemple ci-dessus, "Employé" est l'entité générale et "Employé salarié" et "Employé payé à l'heure" sont les catégories. Il y a deux

relations de catégorisation dans ce groupe, une entre "Employé" et "Employé salarié" et une entre "Employé" et "Employé payé à l'heure".



Catégories multiples

Dans l'exemple ci-dessus, les sociétés et les personnes représentent deux catégories de clients.

Multiples groupes de catégories

Comme une instance de l'entité générale ne peut pas être associée à plus d'une instance des catégories du groupe, ces catégories sont mutuellement exclusives. Dans l'exemple précédent, cela implique qu'un employé ne peut pas être à la fois "salarié" et "payé à l'heure". Cependant, une même entité peut être l'entité générale de plus d'un groupe de catégories, et les catégories d'un groupe ne sont pas exclusives des catégories d'autres groupes. Par exemple, "Employé" peut être l'entité générale dans un deuxième groupe de catégories comprenant les catégories "Employé masculin" et "Employé féminin". Une instance d'employé peut être une instance d'"employé salarié" ou d'employé payé à l'heure" et en même temps d'"employé masculin" ou d'"employé féminin".

Catégorisation complète

Dans un groupe de catégorie "complet", chaque instance de l'entité générale est une instance d'une catégorie du groupe, c'est-à-dire que toutes les catégories possibles sont présentes. Par exemple, chaque employé est soit masculin, soit féminin, ainsi le deuxième groupe de catégories est complet. Dans un groupe de catégorie "incomplet", une instance de l'entité générale peut exister sans être associée à une instance d'une des catégories, c'est-à-dire que certaines catégories sont omises. Par exemple, si certains employés sont payés à la commission plutôt que payés à l'heure ou salariés, le premier groupe de catégorie serait incomplet.

Il est possible de spécifier si une relation de catégorisation est complète ou non dans l'onglet **Caractéristiques** de la fenêtre des propriétés d'une généralisation. Si la valeur de la caractéristique **Complète** est "Oui", toutes les instances de l'entité générale appartiennent à au moins une des catégories de la généralisation.

Discriminant

Un attribut de l'entité générale, ou de l'un de ses ancêtres, peut être désigné comme discriminant pour un groupe de catégorie particulier. La valeur prise par le discriminant détermine la catégorie à laquelle appartient l'instance d'entité. Dans l'exemple précédent, le discriminant pour le groupe incluant les catégories "salariés" et "payés à l'heure" pourrait être "Type d'employé". Si un groupe a un discriminant, il doit être différent de tous les autres discriminants.

Pour définir un identifiant sur une généralisation :

1. Ouvrez les propriétés de la généralisation.
2. Cliquez sur **Caractéristiques**.
3. Dans le champ **Discriminant**, sélectionnez le discriminant parmi les attributs de la super-entité.
Une fois sélectionné, le discriminant s'affiche sur la généralisation.

LA NOTATION I.E.

A propos de la modélisation des données avec I.E.

"Information Engineering" fut à l'origine développé par Clive Finkelstein en Australie à la fin des années 1970. Il a ensuite collaboré avec James Martin pour diffuser sa méthode aux Etats-Unis et en Europe.

Information Engineering est un ensemble intégré et évolutif de tâches et de techniques utilisées pour la planification de l'entreprise, la modélisation des données, la modélisation des processus, la conception des systèmes et leur mise en oeuvre. Il permet à une entreprise de maximiser les ressources financières, humaines ou informationnelles disponibles pour supporter la réalisation de ses objectifs métiers.

Tirée par les métiers, Information Engineering est une des méthodes de développement de systèmes dominantes dans le monde, à une époque où les entreprises tentent de se positionner dans la compétition exacerbée qui règne depuis les années 1990.

Elle se focalise sur les données avant les processus, ce qui assure que les entreprises identifient ce qui est requis par leur métier avant d'analyser la façon de le produire. IE fournit un ensemble important de techniques pour l'analyse stratégique que l'on ne retrouve pas dans les méthodologies basées sur les processus.

Information Engineering guide l'entreprise à travers une série d'étapes prédéfinies qui permettent d'identifier l'information importante pour l'entreprise et établit les relations entre les éléments d'information. Ainsi, les besoins en information sont clairement définis à partir des données de gestion et peuvent être traduits directement dans les systèmes qui supportent les plans stratégiques.

La plupart des systèmes d'information développés durant les vingt-cinq dernières années ont été réalisés depuis la vue spécifique de chaque application, ce qui a beaucoup contribué à l'effet tunnel. Le résultat est que beaucoup d'entreprises ont des systèmes séparés qui sont incapables de partager des données. Dans cette situation, les systèmes ne peuvent pas atteindre leur potentiel initial, et peuvent en fait devenir des fardeaux pour l'entreprise. IE identifie clairement les besoins en partage des données à travers l'entreprise de façon à ce que les systèmes puissent être intégrés en conséquence.

En utilisant IE, les entreprises disposent d'un cadre stable mais flexible sur lequel baser les activités de développement. Cela élimine les redondances et permet la réutilisation de modules de programmation et le partage des données requises par l'entreprise, ce qui aide à alléger le poids de la maintenance.

Modéliser les données consiste à identifier les objets de gestion (entités) et les associations ou relations entre ces objets, considérés d'intérêt pour représenter l'activité de l'entreprise.

I.E. est utilisé pour produire un modèle graphique qui représente la sémantique et la structure de l'information manipulée à l'intérieur d'une entreprise ou d'un

système. L'utilisation de ce standard permet la construction d'un modèle sémantique qui peut servir de support à la gestion des données en tant que ressource, à l'intégration des systèmes d'information, ainsi qu'à la construction de bases de données informatisées.

Les éléments de base d'un modèle de données Information Engineering sont :

- Les objets au sujet desquels les données sont conservées, c-à-d les personnes, lieux, idées, événements, etc., qui sont représentés par des boîtes ;
- Les relations entre ces objets, représentées par des lignes qui relient ces boîtes ; et
- Les caractéristiques de ces objets représentées par les noms des attributs à l'intérieur des boîtes.

Synthèse des concepts

Dans **Hopex Logical Data**, un modèle de données (I.E.) est représenté par :

- Des entités, qui représentent les concepts de base (client, compte, produit, etc.).
- Des associations, qui définissent les associations entre les différentes entités.
- Des attributs qui décrivent les caractéristiques des entités.

L'attribut qui permet d'identifier de façon unique l'entité est appelé identifiant.

Le modèle de données est complété par la définition des multiplicités (ou cardinalités).

Créer un modèle de données (I.E)

Un modèle de données I.E. présente les entités comme des boîtes rectangulaires (une entité est une personne ou une chose à propos de laquelle des données sont stockées). Les entités sont reliées les unes aux autres, par exemple, une entité "Produit" est achetée par une entité "Client". Des traits reliant les boîtes montrent ces associations. Des indicateurs de cardinalité (multiplicité) sont affichés sur ces traits.

La modélisation des données avec I.E est disponible avec les profils suivants :

- **Hopex Data Governance** : Concepteur de données et Administrateur fonctionnel de la gouvernance des données
- **Hopex Data Architecture** : Architecte de données et Administrateur fonctionnel de données

Condition préalable

Pour utiliser la notation I.E, vous devez cocher l'option correspondante :

1. Dans le bureau, cliquez sur le menu **Menu principal > Paramètres > Options.**

2. Dans l'arbre de navigation des options, déployez le dossier **HOPEX Solutions > Fonctionnalités communes**.
3. Cliquez sur **Notation des données**.
4. Dans la partie droite de la fenêtre cochez la notation I.E.
5. Cliquez sur **OK**.

Voir aussi [Options de modélisation des données logiques](#).

Créer le modèle de données

Pour créer un modèle de données dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture > Modèle de données**.
2. Dans la fenêtre d'édition, cliquez sur **Vue hiérarchique**.

 Dans **Hopex Data Architecture**, cliquez sur le menu de navigation **Modèles de données**.

3. A droite du dossier **Hiérarchie des modèles de données**, cliquez sur le bouton **+ Nouveau**.

La fenêtre de création d'un modèle de données apparaît.

4. Saisissez le nom du modèle.

5. Cliquez sur **OK**.

Le modèle de données apparaît dans la liste des modèles de données.

Diagramme de données (I.E)

Un diagramme de données est une représentation graphique du modèle ou d'une partie du modèle.

Pour créer un diagramme de données :

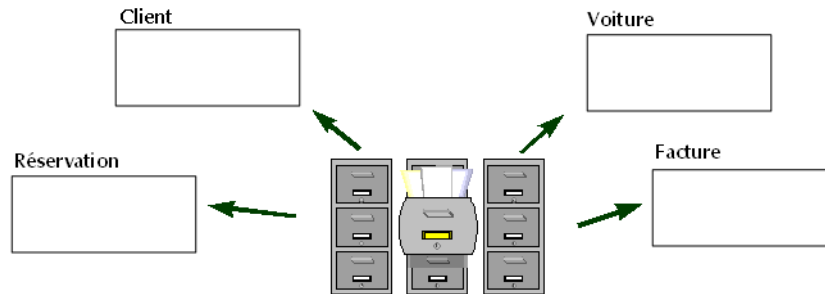
1. Déplacez la souris sur le modèle de données et cliquez sur le bouton **Créer un diagramme**  qui apparaît à droite.
2. Sélectionnez le type de diagramme **Diagramme de données (IE)**.
Le diagramme de données s'ouvre.

Entités (I.E)



Une entité est un regroupement d'objets possédant des caractéristiques communes et un comportement semblable. Les entités sont des éléments de gestion considérés d'intérêt pour représenter l'activité de l'entreprise et sont donc généralement conservés à cet effet. Elles pourront, par exemple, donner lieu à des tables dans une base de données.

Nous pouvons illustrer la notion d'**entité** par une comparaison, par exemple, des fiches dans des tiroirs.



Une entité représente une classe particulière d'objets, dont tous les exemplaires peuvent être décrits de la même manière. Une entité est représentée par une boîte rectangulaire.

Créer une entité

Pour créer une entité :

1. Sélectionnez le bouton **Entité**  dans la barre d'objets en cliquant dessus avec le bouton gauche de la souris.
2. Cliquez sur le plan de travail du diagramme.
La fenêtre **Ajout d'une entité** s'ouvre.
3. Saisissez le nom de l'entité.
4. Cliquez sur **Créer** (Windows Front-End) ou **Ajouter** (Web Front-End).
L'entité apparaît dans le diagramme.

Attributs

Exemples d'**attributs** :

- "Nom du client" (attribut de l'entité client).
- "N° client" (identifiant de l'entité client).
- "Solde du compte" (attribut de l'entité compte).

 *Un attribut représente un type de caractéristique ou une propriété associée à un ensemble d'objets abstraits ou concrets. Une instance d'entité a généralement une valeur spécifique pour chacun des attributs. Une combinaison d'un ou de plusieurs attribut(s) peut être utilisée comme identifiant quand elle permet d'identifier de manière unique chacune des instances d'une entité.*

Définir les attributs

Pour créer un attribut :

1. Cliquez avec le bouton droit sur l'entité et sélectionnez **Propriétés**.
La fenêtre des propriétés de l'entité s'ouvre.
2. Cliquez sur l'onglet **Attributs**.

3. Pour ajouter un nouvel attribut à l'entité, cliquez sur le bouton  .
Un nom vous est automatiquement proposé pour ce nouvel attribut. Vous pouvez le modifier.

Vous pouvez préciser son **Type de données**.

Exemple : Numérique.



Un type de données permet de mettre en commun des caractéristiques communes à plusieurs attributs. Les types de données sont implémentés sous forme de classe.



Voir [Affecter des types aux attributs](#) pour plus de détails sur les **types de données** qui peuvent être affectés à un attribut.

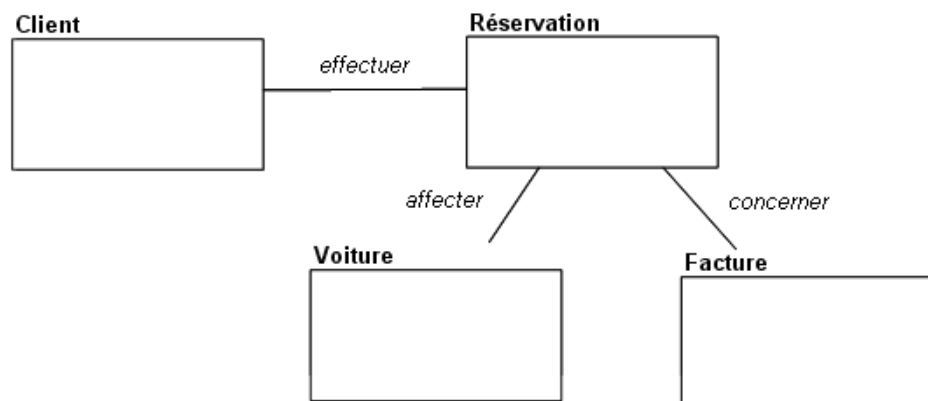
Associations (I.E)



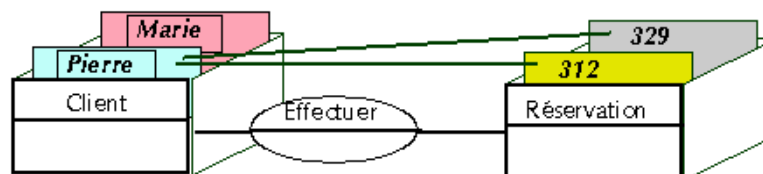
Une association représente un lien significatif entre deux objets. Les associations sont utilisées pour saisir des données au sujet de la relation qui existe entre les deux objets.

Présentation

Les **associations** peuvent être comparées à des liens entre des fiches.



Le dessin suivant permet de visualiser "en trois dimensions" les situations qu'un diagramme de données permet de mémoriser.



Pierre et Marie sont des clients. Pierre a effectué les réservations numéros 312 et 329.

Associations et multiplicités

Chaque rôle d'une association porte une indication de multiplicité qui montre combien d'objets de l'entité considérée peuvent être liés à un objet de l'autre entité. La multiplicité est une information portée par le rôle, sous la forme d'une expression entière bornée. On l'indique en particulier pour chacun des rôles que jouent les entités dans une association.

Pour indiquer qu'un rôle est optionnel, un cercle "O" est placé à l'autre extrémité de du trait, ce qui représente une multiplicité minimum de 0.

Pour indiquer qu'un rôle est obligatoire, un trait "|" est placé à l'autre extrémité de du trait, ce qui représente une multiplicité minimum de 1.

Une patte d'oie est utilisée pour représenter une multiplicité maximum égale à plusieurs.

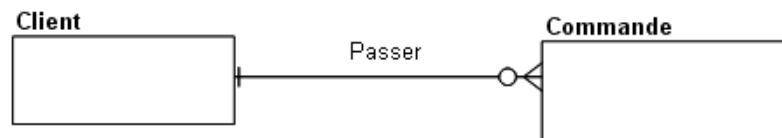
Combinée avec une multiplicité de 0 ou 1, un trait "|" est souvent utilisé pour représenter une multiplicité maximum de 1.

De cette façon, la combinaison "O|" signifie "au plus un" et la combinaison "|"|" ou simplement "|" signifie exactement 1.

Relation obligatoire



Une relation obligatoire signifie que chaque instance de la première entité est associée avec exactement une instance de la deuxième entité et que la deuxième entité peut être associée à zéro, une ou plusieurs instances de la première entité.

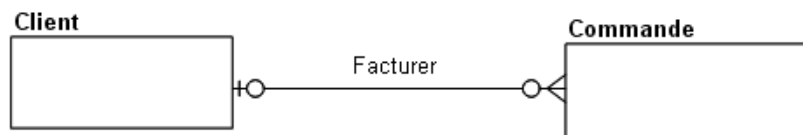


Dans l'exemple ci-dessus, un client peut passer zéro, une ou plusieurs commandes, mais une commande est toujours passée par un et un seul client.

Relation optionnelle



Une relation optionnelle signifie que chaque instance de la première entité est associée avec zéro ou une instance de la deuxième entité et que la deuxième entité peut être associée à zéro, une ou plusieurs instances de la première entité.



Dans l'exemple ci-dessus, un client peut se voir facturer zéro, une ou plusieurs commandes, et une commande devrait être facturée à un client, mais ce n'est pas obligatoire (problèmes de livraison, etc.).

Relation non-spécifique

 Une relation non spécifique signifie que chaque instance de la première entité est associée avec zéro, une ou plusieurs instances de la deuxième entité et que la deuxième entité peut être associée à zéro, une ou plusieurs instances de la première entité.



Dans l'exemple ci-dessus, un article peut apparaître dans zéro, un ou plusieurs catalogues et un catalogue peut contenir zéro, un ou plusieurs articles.

Créer une association

Pour créer une association :

1. Choisissez le type d'association en cliquant sur le bouton correspondant

 ,  ou  dans la barre d'objets.

2. Cliquez dans une des entités concernées, et en gardant le bouton de la souris enfoncé, déplacez le pointeur jusqu'à l'autre entité, avant de relâcher votre pression.

La fenêtre **Ajout d'une association** s'ouvre.

3. Saisissez le nom de l'association, puis cliquez sur **Créer**.

L'association apparaît dans le diagramme.

Pour modifier la multiplicité d'un rôle :

1. Cliquez avec le bouton droit sur le trait qui se trouve entre l'association et l'entité, afin d'ouvrir le menu contextuel du rôle.
2. Cliquez sur **Propriétés**.
La page de propriétés du rôle s'ouvre.
3. Cliquez sur l'onglet **Caractéristiques**.
4. Dans le champ **Multiplicité**, sélectionnez la multiplicité voulue.

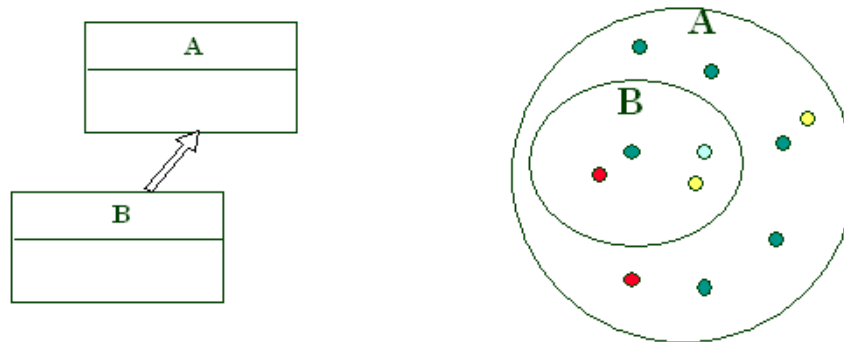
La représentation de l'association change en fonction de la nouvelle valeur de ses multiplicités.

➡ Dans Hopex Windows Front-End, la multiplicité est également affichée dans le menu contextuel du rôle. Si le menu affiché ne propose pas les multiplicités, vérifiez que vous avez bien cliqué sur le trait qui matérialise le rôle, et non sur l'association.

Sous-types (I.E)

📖 Une généralisation représente une relation d'héritage entre une entité générale et une entité plus spécifique. L'entité spécifique est cohérente avec l'entité plus générale et hérite de ses caractéristiques et de son comportement. Elle peut cependant comporter des attributs ou des associations supplémentaires. Tout objet de l'entité spécifique est aussi un objet de l'entité générale.

Qu'est-ce qu'un sous-type ?



L'entité B est *sous-type* de l'entité A. Cela suppose que tous les exemplaires de l'entité B sont aussi des exemplaires de l'entité A. Autrement dit, B est un sous-ensemble de A. B est alors la sous-type, A le sur-type.

Exemple :

A : Personne, B : Parisien.

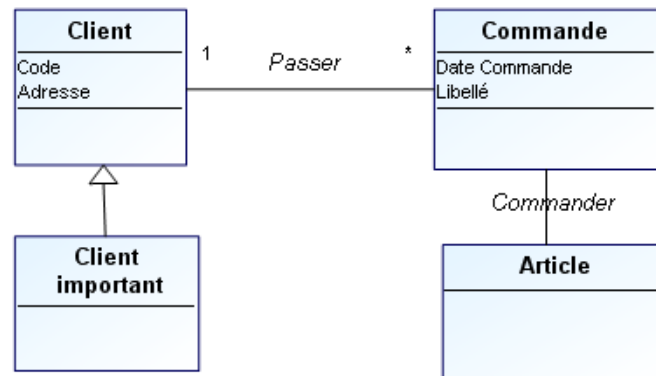
B étant un sous-ensemble de A, les instances de l'entité B "héritent" des caractéristiques de celles de l'entité A.

Il n'est donc pas nécessaire de décrire de nouveau pour l'entité B :

- ses attributs
- ses associations

Exemple :

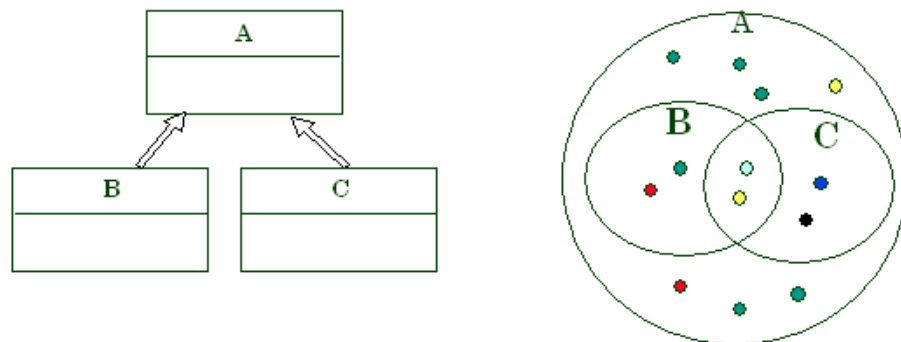
L'entité "Client important" qui représente les clients dont le "C.A. sur les 12 derniers mois" dépasse 1 million d'euros, peut être un sous-type de l'entité client.



Un sous-type hérite de tous les attributs, associations, rôles et contraintes de son sur-type mais il peut avoir des attributs, associations, rôles ou contraintes que n'a pas son sur-type.

Dans l'exemple ci-dessus, les attributs, les associations, les rôles et les contraintes spécifiés pour "Client" sont aussi valables pour "Client important".

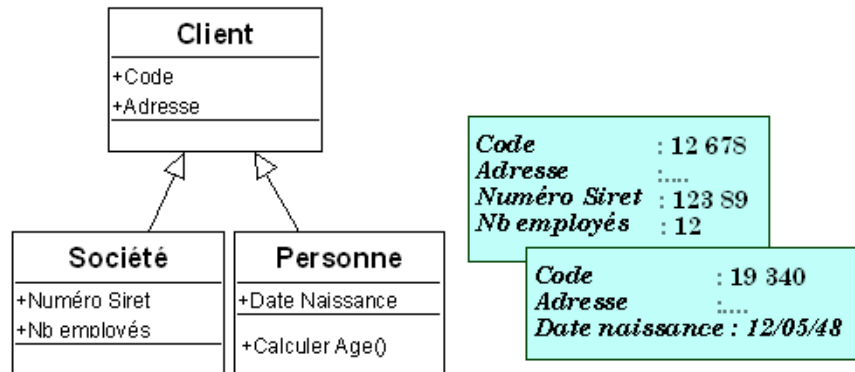
Cas de plusieurs sous-types



Plusieurs sous-types d'une même entité :

- ne sont pas forcément exclusifs.
- ne forment pas nécessairement une partition du type.

Intérêt des sous-types

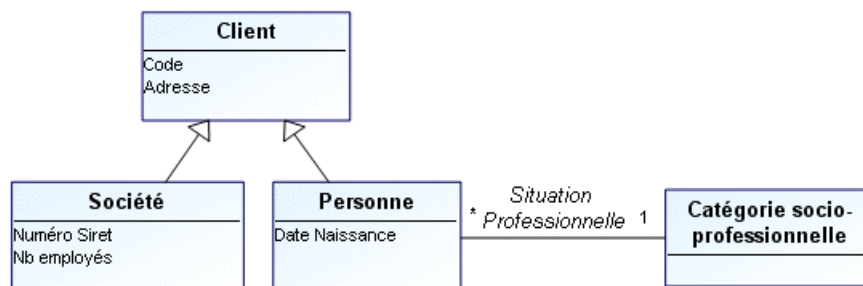


Une entité sous-type hérite de tous les attributs et associations de son sur-type, mais elle peut avoir des attributs ou des associations que ne possède pas son sur-type.

Une entité sous-type peut ainsi avoir des attributs spécifiques. Ceux-ci n'ont de sens que pour une entité sous-type particulière. Dans l'exemple ci-dessus :

- Le numéro de Siret et le nombre d'employés n'ont de sens que pour une société.
- La date de naissance est caractéristique d'une personne, pas d'une société.

Une entité sous-type peut également avoir des associations spécifiques.



- Une personne entre dans une catégorie socio-professionnelle : cadre, employé, commerçant, agriculteur, etc. Cette classification n'a pas de sens pour une entreprise. Il existe également une classification pour les entreprises, mais ce n'est pas la même que pour les personnes.

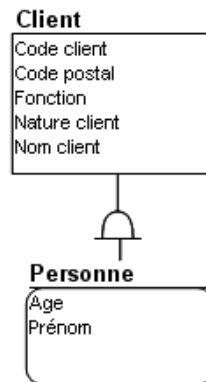
Héritage multiple

Il est parfois utile de spécifier une entité ayant plusieurs entités sur-types. Le sous-type hérite alors de toutes les caractéristiques des deux sur-types. Cette possibilité doit être utilisée avec précaution.

Créer un sous-type

Pour créer un sous-type :

1. Cliquez sur le bouton **Généralisation**  de la barre d'objets.
2. Cliquez sur l'entité sous-type, et faites glisser la souris jusqu'à l'entité sur-type, avant de relâcher votre pression.
La généralisation est représentée dans le diagramme par un demi-cercle souligné, relié par un trait à l'entité sur-type et par un autre à l'entité sous-type.



Dans l'exemple ci-dessus, certains attributs ont un intérêt pour les personnes et n'ont pas de sens pour d'autres catégories de clients. L'entité sous-type est représentée par une boîte aux angles arrondis.

LA NOTATION MERISE

A propos de la modélisation des données

Modéliser les données consiste à identifier les objets de gestion (entités) et les associations ou relations entre ces objets, considérés d'intérêt pour représenter l'activité de l'entreprise.

Il faut que les entités, les associations et les propriétés qui constituent le modèle de données associé à un secteur de l'entreprise suffisent à le décrire complètement sur le plan sémantique.

En d'autres termes, on doit pouvoir décrire l'activité de l'entreprise en utilisant seulement les entités, les associations et les propriétés choisies.

Ceci n'implique pas que, pour chaque mot ou verbe utilisé pour cette explication, il y ait un correspondant direct dans le modèle de données. Il s'agit de pouvoir traduire ce que l'on veut exprimer, au travers des entités, des associations et des propriétés.

Synthèse des concepts

Dans **Hopex Data Architecture**, un modèle de données (Merise) est représenté par :

- Des entités, qui représentent les concepts de base (client, compte, produit, etc.).
- Des associations, qui définissent les associations entre les différentes entités.
- Des attributs (informations ou propriétés), qui décrivent les caractéristiques des entités et, dans certains cas, des associations.

L'attribut qui permet d'identifier de façon unique l'entité est appelé identifiant.

Le modèle de données est complété par la définition des cardinalités.

Créer un modèle de données (Merise)

La modélisation des données avec Merise est disponible avec les profils suivants :

- **Hopex Data Governance** : Concepteur de données et Administrateur fonctionnel de la gouvernance des données
- **Hopex Data Architecture** : Architecte de données et Administrateur fonctionnel de données

Condition préalable

Pour utiliser la notation Merise, vous devez cocher l'option correspondante :

1. Dans le bureau, cliquez sur le menu **Menu principal > Paramètres > Options.**
2. Dans l'arbre de navigation des options, déployez le dossier **HOPEX Solutions > Fonctionnalités communes.**
3. Cliquez sur **Notation des données.**
4. Dans la partie droite de la fenêtre cochez la notation Merise.
5. Cliquez sur **OK.**

Voir aussi [Options de modélisation des données logiques.](#)

Créer le modèle de données

Pour créer un modèle de données dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture > Modèle de données.**
2. Dans la fenêtre d'édition, cliquez sur **Vue hiérarchique.**

☛ Dans **Hopex Data Architecture**, cliquez sur le menu de navigation **Modèles de données.**

3. A droite du dossier **Hiérarchie des modèles de données**, cliquez sur le bouton **+ Nouveau.**
La fenêtre de création d'un modèle de données apparaît.
4. Saisissez le nom du modèle.
5. Cliquez sur **OK.**
Le modèle de données apparaît dans la liste des modèles de données.

Diagramme de données (Merise)

Un diagramme de données est une représentation graphique du modèle ou d'une partie du modèle. La création d'un diagramme varie légèrement suivant que vous êtes en Windows Front-End ou Web Front-End.

Pour créer un diagramme de données :

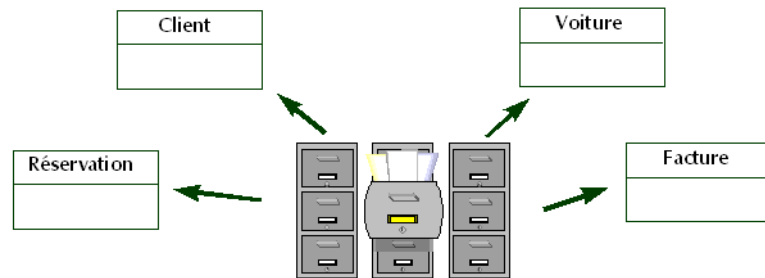
1. Déplacez la souris sur le modèle de données et cliquez sur le bouton **Créer un diagramme**  qui apparaît à droite.
2. Sélectionnez le type de diagramme **Diagramme de données (Merise).**
Le diagramme apparaît.

Les entités (Merise)



Une entité est un regroupement d'objets possédant des caractéristiques communes et un comportement semblable. Les entités sont des éléments de gestion considérés d'intérêt pour représenter l'activité de l'entreprise et sont donc généralement conservés à cet effet. Elles pourront, par exemple, donner lieu à des tables dans une base de données.

Nous pouvons illustrer la notion d'*entité* par une comparaison, par exemple, des fiches dans des tiroirs.

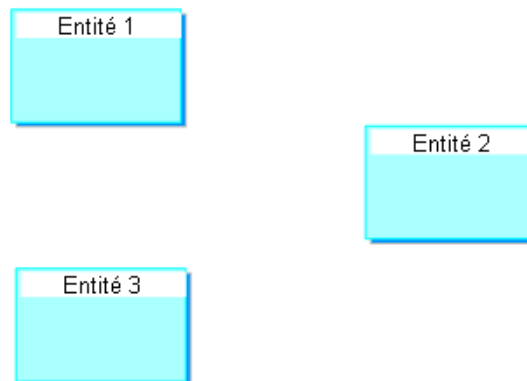


Une entité représente une classe particulière d'objets, dont tous les exemplaires peuvent être décrits de la même manière.

Créer une entité

Pour créer une entité :

1. Sélectionnez le bouton **Entité**  dans la barre d'objets.
2. Cliquez sur le plan de travail du diagramme.
La fenêtre **Ajout d'une entité** s'ouvre.
3. Saisissez le nom de l'entité.
4. Cliquez sur **Créer** (Windows Front-End) ou **Ajouter** (Web Front-End).
L'entité apparaît dans le diagramme.



☛ Vous pouvez créer plusieurs entités à la suite sans revenir à la barre d'objets, avec un double-clic sur le bouton  . Pour revenir ensuite au mode normal, utilisez la touche <Echap>, ou cliquez sur un autre bouton de la barre d'objets, par exemple sur la flèche  .

☛ Les objets que vous créez, ainsi que leurs caractéristiques et leurs liens sont enregistrés automatiquement à chaque fois que la pointe de

la souris prend la forme . Le dessin du diagramme n'est enregistré que lorsque vous le demandez explicitement à l'aide du bouton **Enregistrer** .

Préciser l'identifiant d'une entité

Pour préciser l'identifiant d'une entité :

1. Ouvrez la fenêtre de propriétés de l'entité.
2. Cliquez sur l'onglet **Attributs**.
3. Pour l'attribut choisi sélectionnez la valeur "Oui" dans la colonne **Identifiant**.

➡ Pour plus de détails, voir [Identifiant d'une entité](#).

Les associations (Merise)

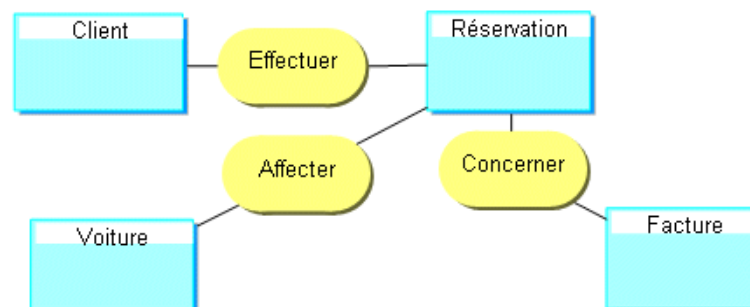
Une association représente un lien significatif entre deux objets. Les associations sont utilisées pour saisir des données au sujet de la relation qui existe entre les deux objets.

Exemples d'associations

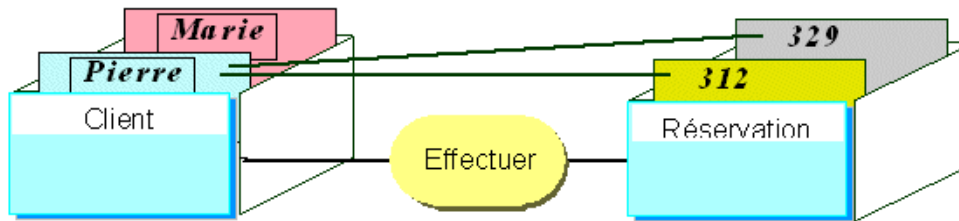
Pour modéliser qu'un "salarié" est responsable d'un "service" et préciser la "date début" de ses fonctions, est créé le modèle de données suivant, où "date début" est une propriété de l'association.



Autre comparaison : des liens entre les fiches.



Le dessin suivant permet de visualiser "en trois dimensions" les situations qu'un modèle de données permet de mémoriser.



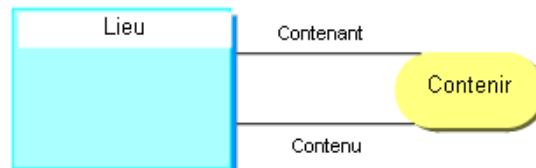
Pierre et Marie sont des clients. Pierre a effectué les réservations numéros 312 et 329.

Un modèle de données doit permettre de mémoriser toutes les situations du contexte de l'entreprise, mais rien que celles-là.

☛ Le modèle ne doit pas permettre de représenter des situations irréalistes ou aberrantes.

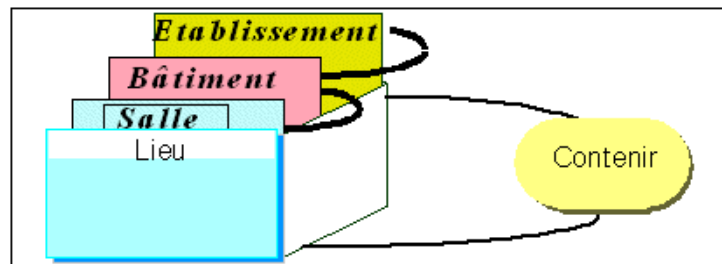
Relation réflexive

Certaines *associations* mettent en jeu la même entité.



Exemple

Une salle de classe, un bâtiment, un établissement scolaire sont tous des lieux.



Une salle de classe est contenue dans un bâtiment, lui-même contenu dans un établissement scolaire.

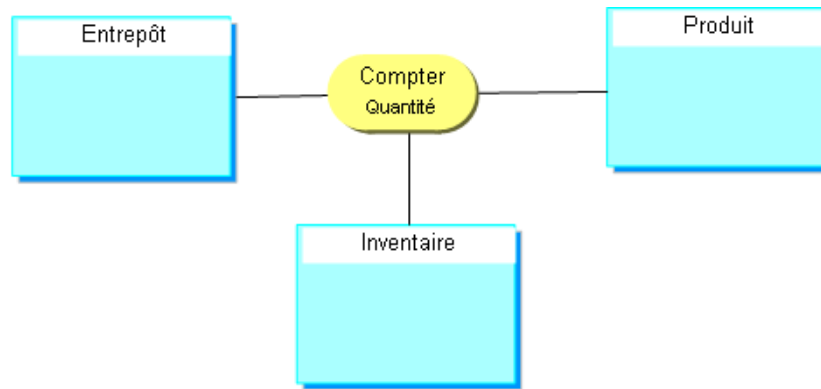
Relation plus que binaire

Certaines associations associent non pas deux, mais davantage d'entités.

Ces associations sont, en principe, rares.

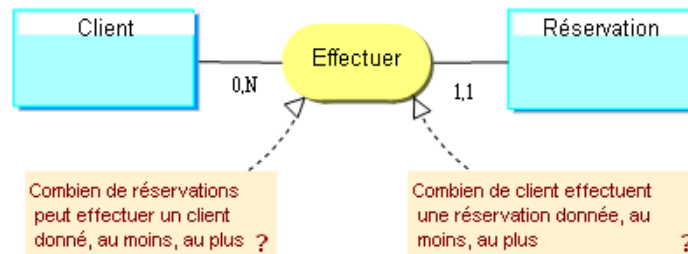
Exemple

Lors d'un inventaire, une certaine quantité de produit a été comptée dans chaque entrepôt.



Les participations ou cardinalités

Cardinalités minimum et maximum expriment le nombre de participations minimum et maximum d'un exemplaire d'une entité à une association.

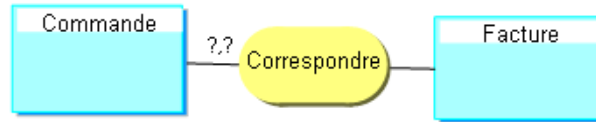


Les participations ou cardinalités usuelles sont 0,1 1,1 0,N 1,N.

- Participation optionnelle : la cardinalité minimum à 0 indique que l'association n'est pas obligatoirement renseignée.
- Participation obligatoire : La cardinalité minimum à 1 indique que l'association est obligatoirement renseignée.
- Participation unique : La cardinalité maximum à 1 indique que l'entité ne peut être reliée par l'association qu'une fois au plus.
- Participation non unique : La cardinalité maximum à N indique que l'entité peut être reliée par l'association plusieurs fois.

Exemple

L'exemple suivant nous permet d'illustrer la signification de chacune de ces cardinalités ou participations:



- 0,1** A une commande correspond aucune facture ou une facture au maximum.
- 0,N** Aucune restriction n'est imposée sur le nombre de factures correspondant à une commande. C'est la valeur par défaut.
- 1,1** A chaque commande correspond une facture et une seule.
- 1,N** A chaque commande correspond une ou plusieurs factures.

Créer une association (relation)

Pour créer une association :

1. Cliquez sur le bouton **Association**  de la barre d'objets.
2. Cliquez sur une des entités concernées et faites glisser la souris jusqu'à la deuxième entité, avant de relâcher votre pression.
La fenêtre **Ajout d'une association** apparaît.
La flèche placée à l'extrémité du champ **Nom** ouvre un menu qui permet :
 - De **Rechercher** les associations existantes, par l'intermédiaire de la fenêtre **Choix d'une sélection**.
 - De **Lister** les associations de la base.
 - De **Créer** une association.
3. Saisissez le nom de l'association puis cliquez sur **Créer** (Windows Front-End) ou **Ajouter** (Web Front-End)
L'association apparaît dans le diagramme.



☛ En cas d'erreur, vous pouvez supprimer un objet en cliquant avec le bouton droit sur cet élément, et en sélectionnant la commande **Supprimer** dans le menu contextuel de l'objet.

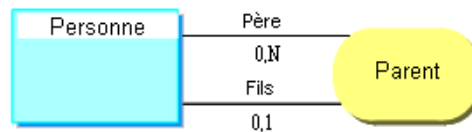
Relation réflexive

Si la demande de création est effectuée sur une entité sans déplacement du pointeur, une association réflexive (aussi appelée "nomenclature") est automatiquement créée sur l'entité.

Dans le cas d'une association entre une entité et elle-même, il est indispensable de préciser les rôles afin de distinguer les liens correspondants dans le dessin.

Exemple :

"Père" et "fils" sont les deux rôles joués par l'entité "personne" dans l'association "parent".



Préciser les participations

Dans l'onglet **Caractéristiques** de la fenêtre de propriétés des rôles, vous pouvez indiquer le nombre de participations minimum et maximum de chaque entité à la relation (cardinalités).

Fenêtre de propriétés des rôles pour "Facture". L'onglet "Caractéristiques" est sélectionné. Les champs suivants sont visibles :

- Nom Local: Facture
- Détenteur: Association (MD) Facturer
- Rôle identifiant: (chambre à sélectionner)
- Tout/Partie: (chambre à sélectionner)
- Multiplicité: (chambre à sélectionner)
- Participation Min: Optionnel
- Participation Max: Non Unique

Les attributs (informations) - Merise

Les propriétés

Les entités et les associations peuvent être caractérisées par des attributs : leurs propriétés.

Ces attributs ont pu être révélés par l'étude du contenu des messages qui circulent à l'intérieur de l'entreprise.

☛ *Un attribut est la donnée élémentaire mémorisée dans le système d'information de l'entreprise. Un attribut est une propriété quand il décrit une entité ou une association, un identifiant quand il est choisi comme moyen d'identification de chaque exemplaire d'une entité.*

Une propriété caractérise une association quand la propriété dépend de l'ensemble des entités participant à l'association.

Dans le diagramme présenté ci-après, le "rôle" qu'un "consultant" a joué sur un "contrat" dépend du consultant et du contrat, donc de l'association "intervenir".

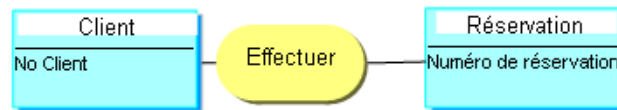
Exemples d'attributs

"Nom du client" (Propriété de l'entité client).

"No client" (identifiant de l'entité client).

"Solde du compte" (Propriété de l'entité compte).

L'identifiant



Le client qui a le numéro 2718 effectue la réservation 314159

Chaque entité possède un **identifiant** unique qui permet de retrouver sans ambiguïté chacun de ses exemplaires.

☛ *Le métamodèle définit la structure du langage utilisé dans les modèles.*

Les associations n'ont pas d'identifiants propres : une association est identifiée par les identifiants des entités reliées.

Créer des attributs

La création des **Attributs** se fait dans la fenêtre de propriétés des associations et des entités.

L'onglet **Attributs** de cette fenêtre présente les attributs déjà reliés à l'entité ou à l'association.

CRUD	Objectifs et exigences	Identifiants	Compléments	Textes	
Général	Caractéristiques	Attributs	Attributs hérités	Modèles de données utilisateurs	Propriétés
					
Nom Local	Type de donnée...	Unicité	Identifiant	Exigé	Remplaçable
Code Client			Oui		
Nom Client			Non		
Fonction			Non		
Nature Client			Non		
Code Postal			Non		

Pour créer un attribut :

- 1 Cliquez sur le bouton et saisissez le nom de l'attribut.

Vous pouvez préciser ses caractéristiques (Voir [Description des attributs](#) pour plus de détails).

Vous pouvez préciser sa **Longueur**, éventuellement complétée par le nombre de **Décimales** ; il faut noter que le nombre de décimales ne s'ajoute pas à la longueur ; une information de longueur 5 avec deux décimales se présentera sous la forme " 999,99 ".

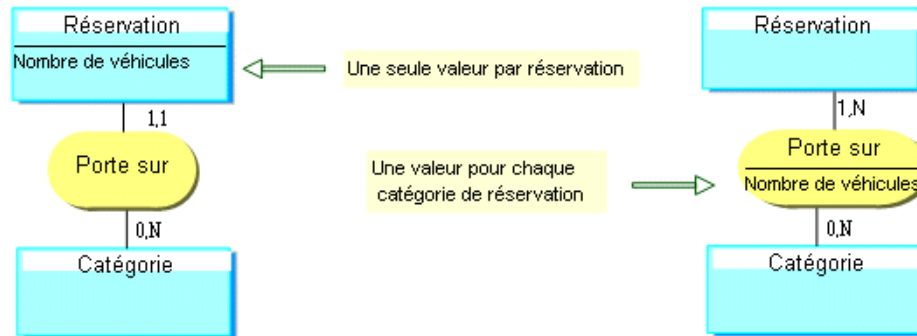
Quand vous avez terminé, fermez la fenêtre de propriétés.

Les règles de normalisation (Merise)

Les formes normales sont des règles qui visent à éviter des erreurs de modélisation. A ce jour, il existe six ou sept formes normales. Nous allons voir les trois premières.

Première forme normale

La valeur d'une Propriété d'entité (ou d'association) est fixée de manière unique dès que l'on connaît l'entité concernée (les entités concernées).

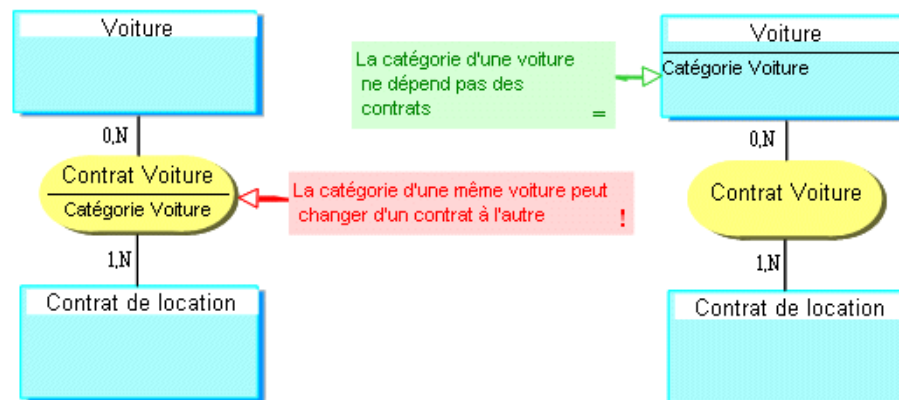


Si le nombre de véhicules est porté par l'entité "Réservation", on ne peut indiquer que le nombre total de véhicules pour une réservation. On doit donc faire une réservation par catégorie de véhicule loué (cardinalités 1,1).

Si le nombre de véhicules est porté par l'association, on peut préciser le nombre de véhicules réservés pour chaque catégorie sur l'association. On peut donc faire une seule réservation pour plusieurs catégories de véhicules (cardinalités 1,N)

Deuxième forme normale

La valeur d'une Propriété d'association n'est fixée que lorsque l'on connaît toutes les entités concernées.

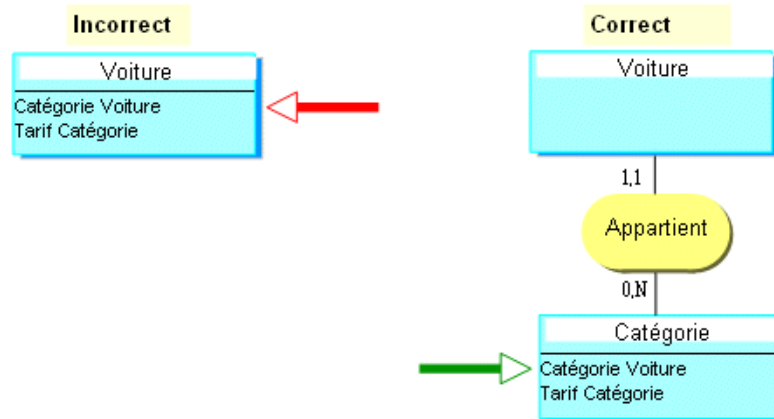


Le fait que la catégorie de voiture porte sur l'association "Catégorie Voiture" suppose que la catégorie de la voiture puisse changer d'un contrat à l'autre, ce qui ne serait pas très honnête.

Pour que la catégorie de la voiture ne dépende pas du contrat, il faut qu'elle soit portée par l'entité "Voiture".

Troisième forme normale

Une Propriété doit dépendre directement et uniquement de l'entité qu'elle décrit.



Si le "Tarif Catégorie" est porté par l'entité "Voiture", cela signifie que deux voitures de la même catégorie peuvent avoir un "Tarif Catégorie" différent. Pour éviter cela, il faut créer une entité "Catégorie" qui portera le tarif.

☛ Cette règle permet de faire émerger des concepts qui n'apparaissent pas dans la première ébauche de modèle de données.

Compléter la spécification du modèle de données (Merise)

En cours de spécification, il est souvent nécessaire de compléter le modèle de données.

Les compléments de spécification consistent à :

- Renseigner les caractéristiques Longueur et Décimale et documenter les attributs.

Dans le modèle de données, il est également possible de spécifier :

- Les entités sous-types.
- Les contraintes que les données doivent respecter, à titre documentaire. Ces contraintes se traduiront par des contrôles à effectuer lors des traitements de mise à jour des données.

Ordonner les attributs

L'ordre initial des attributs est l'ordre de création (ou du création du lien avec l'entité ou l'association).

Pour modifier cet ordre :

1. Dans l'onglet **Attributs** de la fenêtre de propriétés de l'objet, cliquez sur le bouton  **Réordonner**.
La fenêtre **Modification de l'ordre** est présentée.

Pour réordonner les attributs :

1. Sélectionnez l'attribut à déplacer en cliquant sur son nom.
2. Déplacez le curseur jusqu'à l'emplacement souhaité ; il prend la forme suivante : 
L'attribut est placé à l'endroit indiqué, et l'ordre du lien avec l'entité est modifié.

Cet ordre sera utilisé pour générer l'ordre des colonnes dans les tables. Il sera également exploité dans le document associé au modèle de données.

Description des attributs

La description des attributs peut se faire de deux façons :

- Dans la fenêtre de propriétés de l'entité qui les détient : en saisissant cette description dans les différentes cellules de la liste présentée dans l'onglet **Attributs**.
- Dans la fenêtre de propriétés de chaque attribut. Cette fenêtre est ouverte par la commande **Propriétés** du menu contextuel d'un attribut.

Vous pouvez saisir la valeur des caractéristiques des attributs dans les champs correspondants.

- Le **Type de données** qui est la classe utilisée pour préciser le type de l'attribut.
- Le champ **Identifiant** indique si l'attribut fait partie de l'identifiant de l'entité.
- Le champ **Exigé** qui permet de préciser s'il est obligatoire de saisir une valeur pour cet attribut.
- Le champ **Unicité** qui permet d'indiquer que deux instances de cette entité ne peuvent avoir la même valeur pour cet attribut.
- Le champ **Remplaçable** qui permet d'indiquer que la valeur de cet attribut n'est plus modifiable une fois qu'elle a été saisie.

Participations ou Cardinalités

Pour modifier les participations ou *cardinalités* d'une association :

1. Ouvrez la fenêtre de propriétés de l'association.
2. Cliquez sur l'onglet **Caractéristiques**.

3. Saisissez les valeurs de participations (cardinalités)

Nom Local	Entité (MD)	Tout/Pa...	Multiplicité	Participation Min	Participation Max	Rôle identifiant
Article	Article-6			Optionnel	Non Unique	
Client de la remise	Client-10			Optionnel	Non Unique	

☛ Une cardinalité est le nombre de fois minimum (respectivement maximum) où une entité "participe" à une association (voir aussi multiplicité).

Les cardinalités ou participations les plus communément utilisées sont :

- 0 ou 1 pour la cardinalité minimum (participation minimum optionnelle ou obligatoire).
 - 1 ou N pour la cardinalité maximum (participation maximale unique ou non unique).
- Des valeurs différentes sont admises.

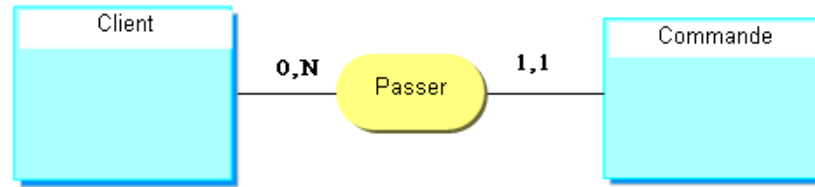
Quand plusieurs rôles, c'est-à-dire plusieurs liaisons, existent entre une entité et une association, les cardinalités sont définies pour chaque rôle.

La cardinalité d'une entité dans une association peut aussi être définie comme suit :

- Pour une association binaire, c'est le nombre minimum (ou maximum) d'exemplaires de l'autre entité intervenant dans l'association, qui peuvent être liés à l'entité de départ.
- Pour une association ternaire, c'est le nombre de couples des deux autres entités intervenant dans l'association, qui peuvent être liés à l'entité de départ.
- Pour une association quaternaire, c'est le nombre de triplets, etc.

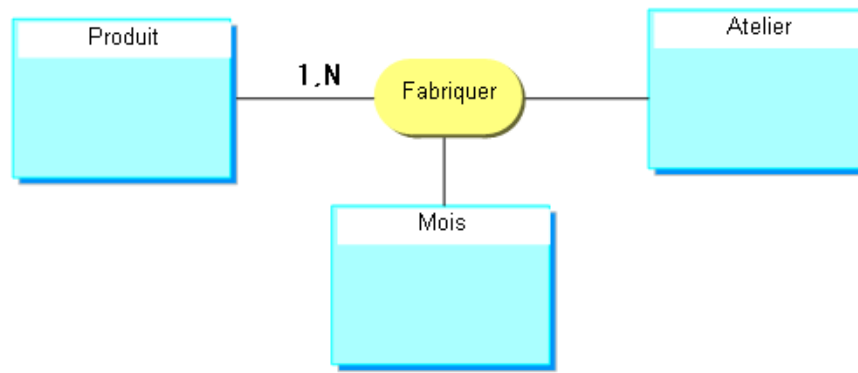
☛ Si l'expression des cardinalités n'est pas suffisante pour décrire la liaison qui existe entre une entité et une association, par exemple quand une cardinalité dépend d'un contexte d'organisation, il est possible d'utiliser les contraintes de cardinalité, qui permettent une description plus précise.

Exemples



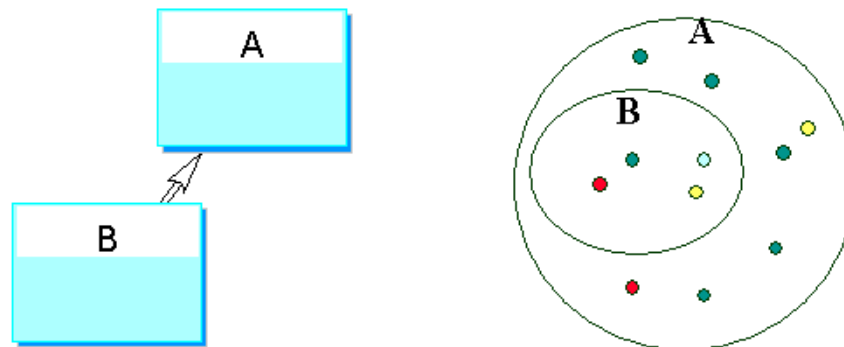
0,N : Le client peut ne pas passer de commande, il peut passer un maximum de N commandes (N indéterminé).

1,1 : La commande doit être passée par un client et un seul.



1,N : Un produit doit être fabriqué au minimum dans 1 atelier pendant un mois. Il peut être fabriqué dans plusieurs ateliers et/ou pendant plusieurs mois (plusieurs couples atelier-mois).

Sous-typage (Merise)



Qu'est-ce qu'un sous-type

L'entité B est sous-type de l'entité A. Cela suppose que tous les exemplaires de l'entité B sont aussi des exemplaires de l'entité A. Autrement dit, B est un sous-ensemble de A.

Exemple A : Personne, B : Parisien.

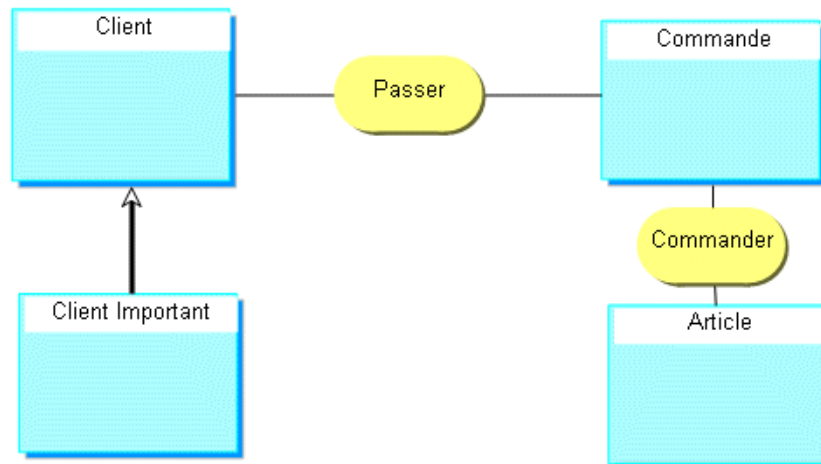
B étant un sous-ensemble de A, les exemplaires de l'entité B " héritent " des caractéristiques de ceux de l'entité A.

Il n'est donc pas nécessaire de décrire de nouveau pour l'entité B :

- ses propriétés,
- ses associations.

Exemple

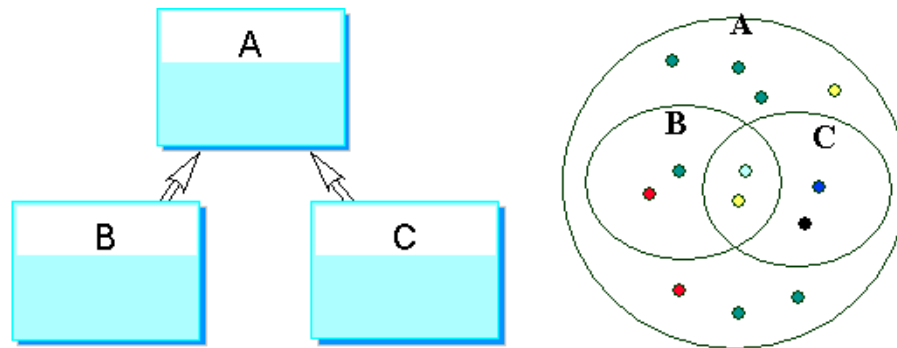
L'entité "Client important" qui représente les clients dont le "C.A. sur les 12 derniers mois" dépasse 1 million d'euros, peut être un sous-type de l'entité client (origine).



Un sous-type hérite de toutes les propriétés, les associations, les rôles et les contraintes de son sur-type mais il peut avoir des propriétés, associations, rôles ou contraintes que n'a pas son sur-type.

Dans l'exemple ci-dessus, les propriétés, les associations, les rôles et les contraintes spécifiés pour "Client" sont aussi valables pour "Client important".

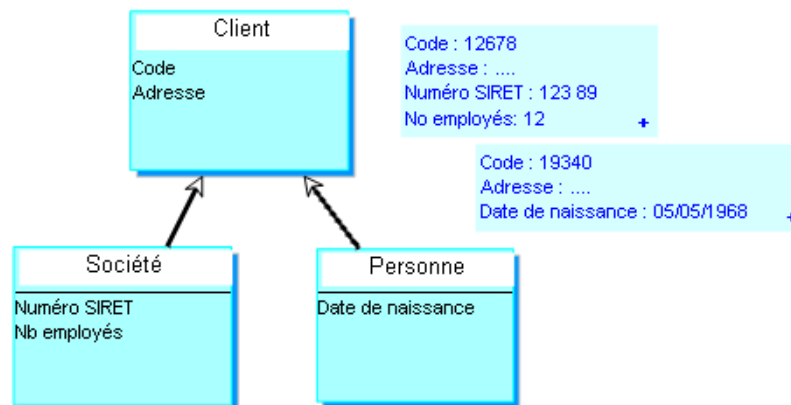
Cas de plusieurs sous-types



Plusieurs sous-types d'une même entité

- ne sont pas forcément exclusifs.
- ne forment pas nécessairement une partition du type.

Intérêt des sous-types



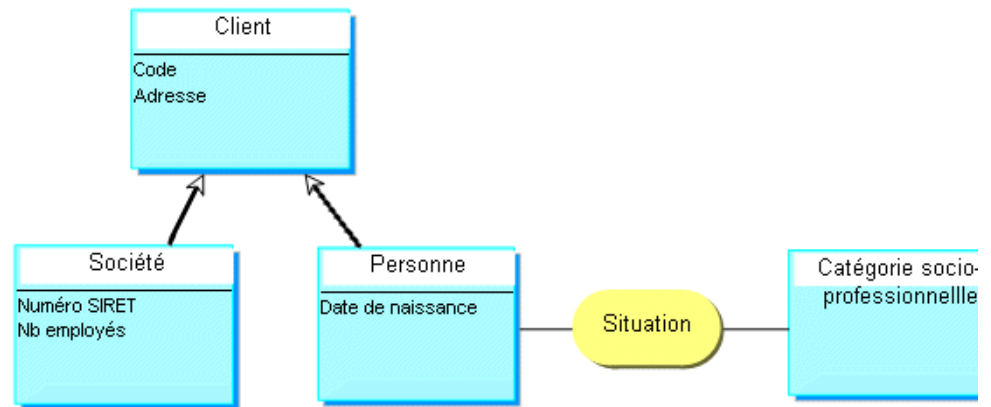
Une entité sous-type peut avoir des propriétés spécifiques. Celles-ci n'ont de sens que pour un sous-type particulier. Dans l'exemple ci-dessus :

- Le numéro de Siret et le nombre d'employés n'ont de sens que pour une société.
- La date de naissance est caractéristique d'une personne, pas d'une société.

Une entité B est sous-type d'une entité A, si B représente un sous-ensemble de A et que les exemplaires de l'entité B héritent de la description de ceux de l'entité A et si ils ont des éléments descriptifs spécifiques.

Le lien Sous-type est représenté graphiquement par une flèche double.

Une entité sous-type peut également avoir des associations spécifiques.



Une personne entre dans une catégorie socio-professionnelle : cadre, employé, commerçant, agriculteur, etc. Cette classification n'a pas de sens pour une entreprise (il existe également une classification pour les entreprises, mais ce n'est pas la même que pour les personnes.)

MODÉLISER LES BASES DE DONNÉES

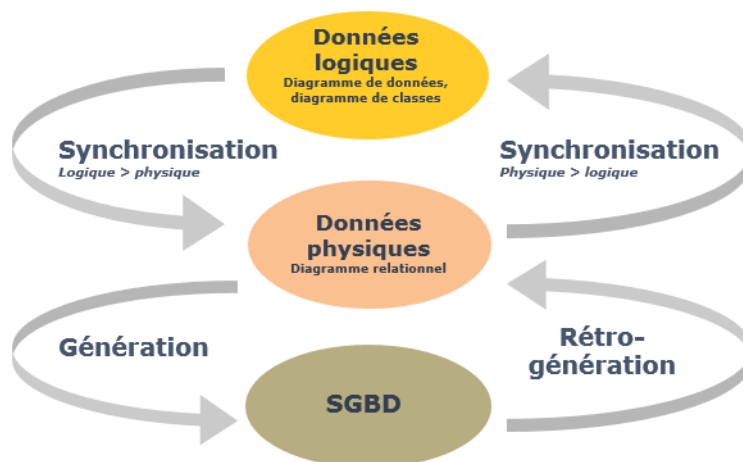


La base de données est l'objet physique qui permet de stocker et d'organiser les données logiques en vue de leur utilisation par des programmes correspondant à des applications distinctes, et de manière à faciliter l'évolution indépendante des données et des programmes.

Les solutions **Hopex Data Governance** et **Hopex Data Architecture** intègrent les niveaux de modélisation logique et physique et permettent le passage d'un modèle à un autre. Vous pouvez ainsi :

- Construire un diagramme de données ou diagramme de classes. Voir [Modéliser les dictionnaires de données](#).
- Créer, à partir de ce diagramme, les tables d'une base de données, avec ses colonnes, index et clés ainsi que les dessins du diagramme relationnel correspondant. Voir [Synchroniser les modèles logiques et physiques](#).
- Optimiser le diagramme relationnel obtenu et générer les ordres SQL de définition des tables. **Hopex Data Governance** et **Hopex Data Architecture** permettent en particulier de prendre en compte des évolutions du modèle conceptuel sans perdre les optimisations effectuées sur le diagramme relationnel. Voir [Dénormaliser les modèles logiques et physiques](#).
- Rétro-générer la définition d'une base de données à l'aide du protocole ODBC, pour créer les tables et colonnes correspondantes dans **Hopex**

Data Architecture, et obtenir le diagramme de données ou diagramme de classes correspondant. Voir [Rétro-générer des tables](#).



Formalisme logique et synchronisation

Le formalisme logique appliqué par défaut dans la synchronisation est la notation UML avec la prise en compte des parties, et non des associations. Si vous synchronisez un modèle de données en modèle physique, les associations du modèle ne sont pas traitées par la synchronisation.

Il est possible de prendre en compte la notation UML et les modèles de données avec le traitement des associations et non des parties. Le changement se fait dans l'application **HOPEX Administration**.

L'option définie dans **HOPEX Administration** s'applique par défaut à toutes les bases de données du référentiel, mais vous pouvez modifier le formalisme uniquement sur une base de données, dans ses options de synchronisation.

Pour accéder à l'option dans **HOPEX Administration** :

1. Ouvrez l'outil d'administration.
2. Ouvrez les options de l'environnement concerné.
3. Dans la fenêtre des options, dans l'arbre de gauche, déployez le dossier **HOPEX Solutions > Data Management**.
4. Cliquez sur **Synchronisation de base de données**.
5. Dans la partie droite de la fenêtre, dans **Nature de correspondance par défaut**, sélectionnez la valeur voulue :
 - UML - Physical : option par défaut (prise en compte des relations de type "Partie")
 - Datamodel - Physical : ancienne option (prise en compte des associations).

Voir aussi : [Options de modélisation des données logiques](#).

BASE DE DONNÉES

Sur une base de données, et en fonction du SGBD cible, vont être définis les paramètres de pilotage des différents outils de traitement des données (synchronisation, génération, rétro-génération etc.).

Créer une base de données

Une base de données permet de spécifier la structure de stockage physique des données.

Pour créer une *base de données* dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture** > **Bases de données**.
 Dans **Hopex Data Architecture**, cliquez sur le menu de navigation **Bases de données**.
2. Dans la zone d'édition, faites un clic droit sur le dossier des bases de données et cliquez sur **Nouveau** > **Base de données**.
3. Saisissez le nom de la base de données.
4. Cliquez sur **OK**.
 La base de données créée apparaît dans la liste des bases.

 Lors de la création d'une base de données, le SGBD **SQL ANSI/ISO 9075:1992** lui est associé par défaut.

Propriétés d'une base de données

Pour accéder aux propriétés d'une base de données :

1. Sélectionnez la base de données voulue et cliquez sur le bouton

Propriétés .

La fenêtre de propriétés de la base de données apparaît.

2. Cliquez sur la liste déroulante pour accéder aux différentes pages de propriétés.

Les pages de propriétés permettent :

- D'accéder aux **Composants** de la base de données (tables, vues physiques, groupements de données, etc.).
- De modifier les **Caractéristiques** de la base (nom, SGBD cible, etc.).
- De définir les **Responsabilités**.
- De définir les **Risques** associés, les **Standards** utilisés, les **Objectifs et exigences**.
- De définir les **Options** liées à :
 - la génération des tables. Voir [Paramétrer la génération d'une base de données](#).
 - la synchronisation. Voir [Paramétrer la synchronisation](#).

Associer un paquetage à une base de données

Vous pouvez créer un paquetage de données à partir de la base de données ou lui relier un paquetage existant. Le paquetage permet de représenter la structure de la base de données, les classes et les parts qu'elles contient.

Le paquetage de la base de données est le détenteur par défaut des objets représentés dans le diagramme de classes. Néanmoins, il est possible d'utiliser des objets détenus dans d'autres paquetages.

☛ *De la même façon, vous pouvez relier un modèle de données à une base de données, lorsque vous avez sélectionné le formalisme correspondant. Voir [Les formalismes](#).*

Pour créer un paquetage à partir d'une base de données :

1. Faites un clic droit sur la base de données et cliquez sur **Nouveau > Paquetage**.
2. Indiquez le nom du paquetage.
3. Cliquez sur **OK**.

Pour relier un paquetage à une base de données :

1. Faites un clic droit sur la base de données et cliquez sur **Relier > Paquetage**.
La fenêtre de recherche apparaît.
2. Cliquez sur **Chercher**.
3. Sélectionnez le paquetage voulu et cliquez sur **Relier**.

Vous pouvez voir le nom des paquetages associés à une base de données dans les propriétés de la base de données, sous la page **Caractéristiques**.

Importer une version de SGBD

Lors de la création d'un nouveau référentiel, vous disposez par défaut de différents SGBD et versions, excepté les dernières versions, telles que "Oracle 18c" et "Oracle 19c". Vous pouvez télécharger ces versions avec leurs datatypes dans HOPEX Store.

Lors de la création d'une base de données, le SGBD SQL ANSI/ISO 9075:1992 lui est associé par défaut. Vous pouvez choisir un autre SGBD cible dans ses pages de propriétés.

Les datatypes associés aux SGBD ne sont pas livrés par défaut. Pour voir dans le référentiel HOPEX les datatypes d'un SGBD, vous devez importer le module correspondant, dont le nom a pour forme "Database Design XXX" avec le nom du SGBD. Les modules sont disponibles dans HOPEX Store.

☛ *Pour plus d'informations sur l'import d'un module, voir [Importer un module dans HOPEX](#).*

Voir aussi [Paramétrer la synchronisation](#).

CARTE DE DOMAINES RELATIONNELS ET DOMAINES RELATIONNELS

Une base de données peut être décomposée en domaines relationnels.

Un domaine relationnel permet de définir une structure de données physiques.

Une carte de domaines relationnels permet de visualiser les liens de dépendance entre des domaines relationnels.

Carte de domaines relationnels

Une carte de domaines relationnels est un outil d'urbanisation des informations physiques. Elle permet de représenter un ensemble de domaines relationnels dans un contexte particulier.

Créer une carte de domaines relationnels

Pour créer une carte de domaines relationnels dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture** > **Bases de données**.

➡ Dans **Hopex Data Architecture**, cliquez sur le menu de navigation **Bases de données**.

2. Dans la zone d'édition, faites un clic droit sur la base de données concernée et cliquez sur **Nouveau** > **Carte de domaines relationnels**. La carte créée apparaît.

Pour créer le diagramme de la carte de domaines relationnels :

1. Passez le curseur de la souris sur la carte et cliquez sur le bouton **Créer un Diagramme**.
2. Sélectionnez **Carte de domaines relationnels**.
3. Cliquez sur **OK**.

Le diagramme apparaît dans la zone d'édition.

Les composants d'une carte de domaines relationnels

Dans une carte de domaines relationnels vous pouvez ajouter des composants internes et externes.

Les composants internes sont les domaines relationnels qui font partie du périmètre de la carte (qu'ils appartiennent ou non au même élément détenteur).

Les composants externes sont ceux qui sont utilisés dans la carte mais qui ne font pas partie du périmètre étudié.

Pour ajouter un domaine relationnel à la carte :

1. Dans la barre d'insertion du diagramme, cliquez sur le bouton **Composant interne** puis cliquez à l'intérieur de la carte.

2. Indiquez le nom du domaine relationnel et cliquez sur **Ajouter** ou **Créer** selon qu'il s'agit d'un domaine relationnel existant ou d'un nouveau domaine relationnel.

Domaine relationnel

Un domaine relationnel représente une structure de données relationnelles restreinte.

Créer un Domaine relationnel

Pour créer un domaine relationnel dans une base de données :

1. Cliquez sur le menu de navigation puis sur **Architecture > Vue hiérarchique**.
2. Dans la zone d'édition, dépliez le dossier **Base de données**.
3. Cliquez sur la base de données concernée pour ouvrir ses propriétés.
4. Cliquez sur la page **Caractéristiques**.
5. Sous la section **Groupe physique**, cliquez sur l'onglet **Domaine relationnel**.
6. Cliquez sur **Nouveau**.

Le nouveau domaine relationnel est créé. Vous pouvez ouvrir ses propriétés pour modifier ou compléter ses caractéristiques.

Diagramme d'un domaine relationnel

Un domaine relationnel est constitué de tables et/ou de vue physiques et peut être décrit par deux types de diagramme :

- le diagramme des tables qui permet de visualiser un ensemble de tables et leurs relations (FK).
- le diagramme de structure qui permet de décomposer un domaine relationnel en sous-domaines.

Vous pouvez relier plusieurs diagrammes à un domaine relationnel, suivant ce que vous voulez décrire.

Pour créer un diagramme à partir du domaine relationnel :

1. Passez la souris sur le domaine relationnel et cliquez sur le bouton **Créer un diagramme**, suivi du type de diagramme (tables ou structure).

Ajouter un composant à un domaine relationnel

Il est possible d'attacher des objets au domaine relationnel à travers des composants.

Un domaine relationnel peut être composé de :

- Sous-domaines, visibles dans le diagramme de structure du domaine relationnel
- Données métier, auxquelles on définit le type d'accès (lecture seule, modification, suppression, etc.)
- Tables ou vues physiques, auxquelles on définit le type d'accès (lecture seule, modification, suppression, etc.), et visibles dans le diagramme des tables du domaine relationnel.

Pour ajouter un composant au domaine relationnel :

1. Ouvrez les propriétés du domaine relationnel en question.
2. Cliquez sur la page **Composants**.
La première section vous permet d'ajouter des domaines.
La deuxième section permet d'ajouter des objets de type table ou vue physique.
3. Cliquez sur **Nouveau** pour ajouter un composant.

Définir le mode d'accès à l'objet référencé

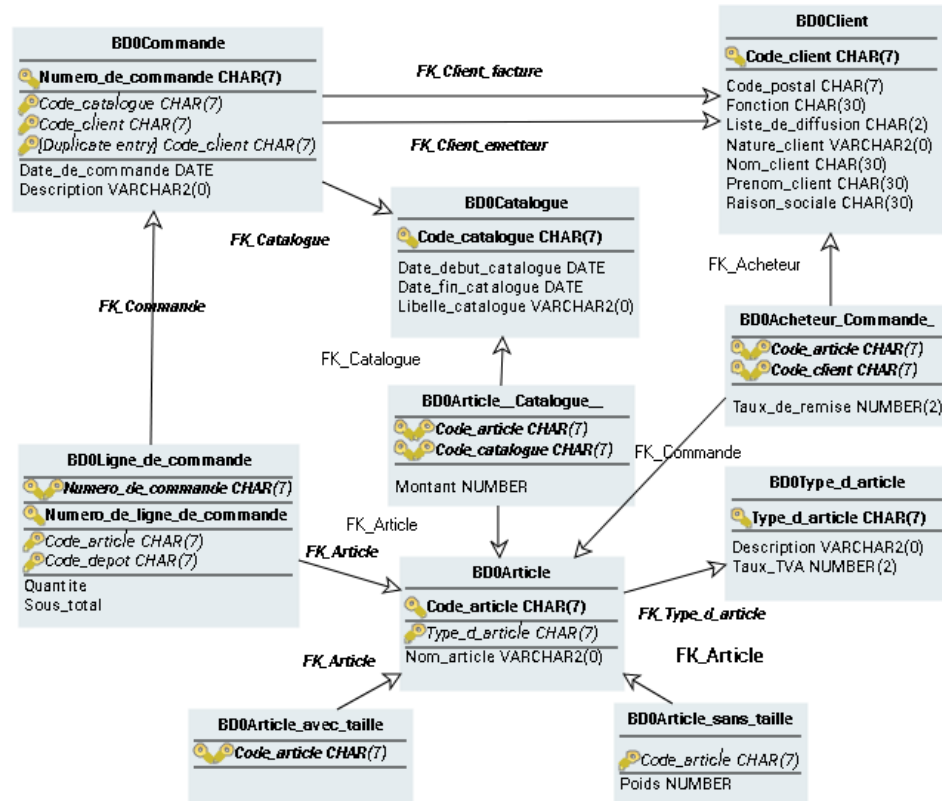
Sur les composants de type table ou vue physique vous pouvez définir le mode d'accès à l'objet référencé (création, lecture, suppression, etc.).

Pour définir le mode d'accès à l'objet dans le domaine relationnel :

1. Sur la page **Composants**, sélectionnez la ligne du composant en question.
Des commandes s'ajoutent, dont le bouton **CRUD**.
2. Cliquez sur ce bouton.
3. Dans la fenêtre qui apparaît, cochez ou décochez les cases de chaque associées aux actions : Create, Read, Update Delete.

DIAGRAMME RELATIONNEL

Un diagramme relationnel (DR) décrit une base de données : il représente les structures des données physiques exploitées par les programmes.



La description dans **Hopex Data Architecture** des diagrammes relationnels permet d'envisager l'utilisation d'une interface avec le SGBD retenu, garantissant ainsi la cohérence sémantique des données de conception avec celles qui sont exploitées.

Construire le diagramme relationnel

Généralement, le diagramme relationnel est construit en deux phases :

1. La synchronisation automatisée du ou des diagrammes de données permet d'obtenir le diagramme "brut".
Voir [Synchroniser les modèles logiques et physiques](#).

2. L'optimisation du diagramme, aussi appelée dénormalisation, permet de prendre en compte les besoins d'accès aux données de l'applicatif et d'optimiser les performances de la base de données.
Voir [Dénormaliser les modèles logiques et physiques](#).

Le concept central d'un diagramme relationnel est la table, issue d'une entité ou d'une association.

 *La table est une structure logique de données, utilisée comme référence pour le passage en réalisation ; c'est l'élément central de la base de données. Une table est accessible par une clé primaire, et éventuellement des clés étrangères ; elle est décrite par une séquence ordonnée de colonnes. Une table est généralement issue d'une entité ou d'une association.*

Une **table** est accessible par une ou plusieurs clés, dont le type indique si elles sont primaires ("primary key") ou étrangères ("foreign key"). Il est possible de définir les index d'une table, en précisant s'ils sont uniques et leur sens de tri (ascendant ou descendant). Les clés et les index sont reliés aux colonnes qui les constituent.

Créer des objets dans le diagramme

Pour créer une clé ou un index dans le diagramme relationnel :

1. Cliquez sur la table concernée.
La liste des commandes associées à la table apparaît.
2. Cliquez sur le bouton **Clé**  ou le bouton **Index** .

 *Assurez-vous que les colonnes sur lesquelles doivent porter cette clé ou cet index existent déjà dans la table.*

Vous pouvez également utiliser la page **Composants** de la fenêtre de propriétés de la base de données pour créer ces objets. Voir [Créer une clé](#) et [Créer un index](#).

Pour créer une clé étrangère :

1. Cliquez sur le bouton **Clé** , placez le pointeur sur la première table puis, en maintenant le bouton enfoncé, faites glisser le pointeur vers la seconde table.
Une fenêtre de création apparaît.
2. Indiquez le nom de la clé et cliquez sur **Ajouter**.
Une seconde fenêtre demande si vous voulez créer automatiquement les colonnes de clé étrangères à partir de celles de la clé primaire.
3. Cliquez sur **Oui** pour valider ou **Non** pour créer

Paramétrer l'affichage des diagrammes relationnels

De même que pour les diagrammes de données, il est possible de préciser les éléments qui doivent apparaître dans le diagramme :

- Soit par le bouton **Vues et détails** qui indique globalement les types d'objets présentés dans le diagramme.
- Soit par les options d'affichage des éléments, qui permettent de définir quelles caractéristiques des objets doivent être présentées.

Pour paramétrer l'affichage d'un seul objet :

- 1 Faites un clic droit sur l'objet et sélectionnez **Formes et détails**.

☛ Lorsque l'affichage concerne un objet, la fenêtre **Affichage** présente d'abord les formes qu'il est possible d'utiliser pour la présentation de l'objet. La sélection d'un élément dans l'arborescence provoque l'affichage de son contenu.

COMPOSANTS D'UNE BASE DE DONNÉES

Une base de données est un ensemble de données organisé en vue de son utilisation par des programmes correspondant à des applications distinctes et de manière à faciliter l'évolution indépendante des données et des programmes.

Une base de données est constituée de tables, de colonnes, de clés et d'index :

- La **table** est l'unité logique de stockage des colonnes.
- Une **colonne** est contenue dans une table.
- De même qu'un identifiant "estampille" une entité, la **clé primaire** de la table "estampille" une ligne de table.
- Une **clé étrangère** permet l'accès à une autre table, et le contrôle de cohérence des colonnes entre les tables concernées.
- Un **index** permet d'accélérer l'accès aux données ; il peut être unique ou non, ascendant ou descendant.

Les tables d'une base de données

Il est possible de consulter et de mettre à jour les tables d'une base de données de deux façons :

- Dans son diagramme relationnel, c'est-à-dire le diagramme des tables de la base de données.
- Dans ses propriétés, sous la page **Tables**.

Créer une table

Voir préalablement : [Propriétés d'une base de données](#).

Pour créer une table à partir des propriétés de la base de données :

1. Ouvrez la fenêtre de propriétés de la base de données.
2. Cliquez sur la page **Tables**.
3. Cliquez sur **Nouveau**.
4. Saisissez le nom de la table et cliquez sur **OK**.

Supprimer une table

Pour supprimer une table, avec ses colonnes, clés et index :

1. Sélectionnez la table et cliquez sur **Enlever** 
Un message demande si vous souhaitez enlever la table de la base ou la supprimer du référentiel.
2. Sélectionnez **Supprimer** et cliquez sur le bouton **Enlever**.

 Une table supprimée ne sera pas recréée automatiquement lors d'une nouvelle synchronisation. Pour recréer la table, lors de l'étape de validation des résultats de la synchronisation, validez l'action de

création proposée pour cette table (cochez la case correspondante).
Voir [Etape 4 : valider les résultats](#).

Les colonnes d'une table

Visualiser les colonnes

Voir préalablement : [Propriétés d'une base de données](#).

Pour consulter les colonnes d'une table :

1. Ouvrez la fenêtre de propriétés de la table.
2. Cliquez sur la page **Colonnes**.

Pour chaque colonne sont présentés :

- Son **Nom local**
- Son type (**Datatype**)

☛ L'administrateur peut compléter la liste des datatypes (voir [Créer de nouveaux datatypes](#)).

- Éventuellement sa longueur (**Lng**) et son nombre de décimales (**Déc**)
- La valeur de l'attribut **Not null**
- Sa **valeur par défaut** : lors de la génération de la table, la valeur prise par défaut est celle de l'attribut dont elle découle. Si aucune valeur initiale n'est renseignée au niveau de l'attribut ou si vous voulez modifier la valeur d'une colonne, entrez une valeur dans ce champ.
- Le fait que la colonne soit reliée ou non à une clé primaire (PK) ou étrangère (FK). Cela est indiqué par **Y** ("Oui") ou **N** ("Non").

Il est possible de modifier le **Nom Local** d'une colonne en cliquant sur son nom et en saisissant le nouveau nom. Ce nom local sera utilisé dans le script généré pour la table.

Il est possible d'indiquer un **Nom SQL** directement dans la page **SQL** de la fenêtre de propriétés d'un attribut dans le diagramme de données. De cette façon, toutes les colonnes créées à partir de cet attribut auront pour base le même nom local. De plus ce nom sera réutilisé lors des synchronisations successives, y compris lors d'une réinitialisation totale ou partielle.

Il est également possible de modifier la valeur des autres caractéristiques de la colonne.

☛ Ces modifications seront conservées lors des synchronisations ultérieures.

Il est possible de créer des colonnes qui ne proviennent pas d'attributs du diagramme de données, que ce soit dans une table générée ou créée par l'utilisateur.

Créer une colonne

Voir préalablement : [Propriétés d'une base de données](#).

Pour créer une colonne :

1. Ouvrez les propriétés de la table concernée.

2. Sous la page **Colonnes**, cliquez sur le bouton **Nouveau**.

 Lorsque la création d'une colonne n'est pas effectuée à partir des **Propriétés** d'une table, par exemple avec l'explorateur, il est nécessaire de sélectionner au préalable la table qui la contiendra, sinon un message indique que la création est impossible.

Supprimer une colonne

Pour supprimer une colonne :

1. Faites un clic droit sur la colonne et sélectionnez **Enlever**.
Un message vous demande confirmation de la suppression définitive.
2. Cliquez sur **Supprimer**.

 Une colonne supprimée ne sera pas recréée automatiquement lors d'une nouvelle synchronisation. Pour recréer la colonne, lors de l'étape de validation des résultats de la synchronisation, validez l'action de création proposée pour cette colonne (cochez la case correspondante). Voir [Etape 4 : valider les résultats](#).

Le bouton **Réordonner**  permet d'accéder à la fenêtre de **Modification de l'ordre**.

Modifier les clés et les index

La création automatique des clés primaires et étrangères, et celle d'index sur ces clés, est indiquée dans le paramétrage de la synchronisation.

Lorsque ces créations sont demandées :

- Les clés primaires portent sur les colonnes correspondant aux identifiants.
- Les clés étrangères portent sur les colonnes qui migrent dans les tables à cause d'une association contrainte.

Un index est créé sur chaque clé.

Il est possible de compléter, modifier ou supprimer les clés et index proposés lors de la génération. Pour cela :

1. Ouvrez les propriétés de la table concernée.
2. Cliquez sur la page **Caractéristiques**.
3. Affichez les sections **Clés** et **Index** qui peuvent être cachés par défaut.
La page présente les **Clés** et **Index** de la table.

Dans la section **Clés** sont indiqués :

- Le type de la clé (**Type-Clé**) : "Etrangère" ou "Primaire".
- Dans le cas d'une clé étrangère :
 - La table référencée.
 - La gestion de l'intégrité référentielle en mise à jour (**On Update**) et en suppression (**On Delete**) ; référez-vous à la documentation du SGBD cible pour le type d'ordres gérés.

 Lorsqu'une colonne dite "migrante" est créée dans une table pour prendre en compte une association contrainte, il est possible de demander que le SGBD contrôle la valeur mise à jour dans cette colonne. Le SGBD vérifie alors que cette valeur existe toujours dans la table d'origine (intégrité référentielle).

Lors d'une mise à jour (**On Update**) ou d'une suppression (**On Delete**) dans la table d'origine, le SGBD peut :

- Mettre à jour les valeurs dans les tables concernées, avec l'option **Cascade**.
- Ne rien faire, avec l'option **No Action**.
- Interdire la mise à jour ou la suppression, avec l'option **Restrict**.
- Remettre la valeur par défaut dans les tables concernées, avec l'option **Set Default**.
- Remettre la valeur à **Null** dans les tables concernées, avec l'option **Set Null**.

Dans la section **Index** sont indiqués :

- Son **Type** : Bitmap, Standard, Unique, Unique where not null.
- Son **Sens de Tri** (**Ascendant** ou **Descendant**).
- S'il s'agit d'un index groupé (**Clustered**).

☛ La création d'une colonne à partir d'une clé ou d'un index n'est pas possible. Il faut d'abord créer la colonne dans la table, puis la relier à la clé ou à l'index.

Créer une clé

Voir préalablement : [Les tables d'une base de données](#).

Pour créer une clé :

1. Ouvrez les propriétés de la table concernée.
2. Cliquez sur la page **Caractéristiques**.
3. Sous la section **Clés**, cliquez sur le bouton **Nouveau**. La fenêtre de création d'une clé apparaît.
4. Sélectionnez le type de clé à créer ; "étrangère" ou "primaire". La création d'une clé varie selon le type indiqué.

Clé primaire

Lorsque vous sélectionnez le type "Primaire", la clé apparaît dans les propriétés de la table.

Pour définir les propriétés de la clé :

1. Passez le curseur de la souris sur la clé et cliquez sur le bouton **Propriétés**.
2. Dans la page **Colonnes** vous pouvez spécifier les colonnes sur lesquelles porte la clé.

Il est également possible de spécifier la clé primaire d'une table dans la page **Identifiants** de la fenêtre de propriétés de l'entité dont elle est issue. Voir [Identifiant d'une entité](#).

Il est également possible de spécifier manuellement la clé primaire en la faisant porter sur des éléments qui peuvent être des attributs de l'entité ou la clé primaire d'une autre table reliée par une association contrainte (multiplicité 1).

Dans tous les cas, la clé ainsi spécifiée sera créée dans la table lors de la synchronisation.

Clé étrangère

Lorsque la clé créée est une clé étrangère, une liste des tables de la base de données est présentée.

1. Sélectionnez la table de référence sur laquelle porte la clé étrangère.
Si la table que vous choisissez comporte une clé primaire, un message apparaît.



2. Sélectionnez **Oui**.
la clé apparaît dans les propriétés de la table.

Vous pouvez modifier le **Nom local** de la clé (le nom complet d'une clé est composé du nom de la base de données à laquelle elle appartient, suivi du nom de la table, puis de son nom local : dans l'exemple qui précède, "BD Bourse::Concerner::Cle1").

Pour une clé étrangère, comme lors de l'édition d'une clé, il est possible de préciser la gestion de l'intégrité référentielle en mise à jour (**On Update**) et en suppression (**On Delete**).

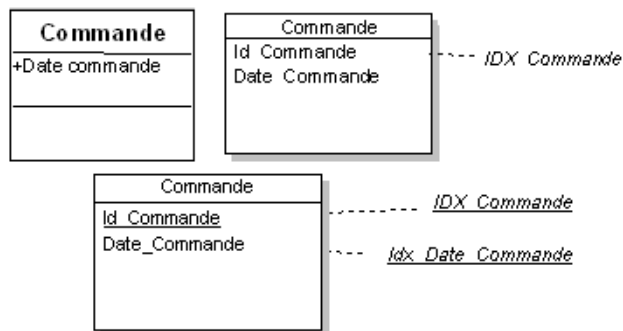
Créer un index

Voir préalablement : [Les tables d'une base de données](#).

Des index sont créés automatiquement sur les clés primaires et étrangères. Il est possible d'ajouter, aux index générés, des colonnes utilisées fréquemment comme critères de recherche.

☺ *Il est également possible de spécifier un index dans la page **Identifiants** de la fenêtre de propriétés de l'entité dont la table est issue. L'index ainsi spécifié sera créé dans la table lors de la synchronisation.*

Exemples d'index :



Pour créer un index :

1. Ouvrez les propriétés de la table concernée.
2. Cliquez sur la page **Caractéristiques**.
3. Sous la section **Index**, cliquez sur le bouton **Nouveau**. La fenêtre **Création d'un index** est présentée.
4. Saisissez le **Nom local** et cliquez sur **OK**.

En fonction des possibilités offertes par le SGBD utilisé, il est possible de préciser le **Type** de l'index, le **Sens de tri** de l'index ("Ascendant" ou "Descendant"), et s'il s'agit d'un index groupé (**Clustered**).

Il est alors possible de sélectionner les colonnes de la clé (de l'index) dans la page **Colonnes** de la fenêtre de propriétés de l'index.

Ajouter une colonne à une clé ou un index

Voir préalablement : [Les tables d'une base de données](#).

Pour ajouter une colonne à une clé (ou à un index) :

1. Ouvrez les propriétés de la table concernée.
2. Cliquez sur la page **Caractéristiques**.
3. Sous la section **Index**, passez la souris sur l'index et sélectionnez **Propriétés**. La fenêtre de propriétés de l'index apparaît.
4. Cliquez sur la page **Colonnes**.
5. Cliquez sur le bouton **Relier**. La fenêtre de recherche apparaît.
6. Recherchez et sélectionnez la colonne à ajouter à la clé (ou à l'index).

Il est possible d'indiquer le sens de tri de la clé ou de l'index, qui peut être "Ascendant" ou "Descendant".

CLÉS PRIMAIRES ET ÉTRANGÈRES

Lorsque les clés d'une base de données ne sont pas complètement spécifiées, il est nécessaire de les **Compléter**.

Spécifier les clés primaires

Pour spécifier les clés primaires d'une base de données :

- 1 Faites un clic droit sur la base de données et sélectionnez **Compléter les clés**.

La fenêtre **Compléter les clés** apparaît.

Lorsque la base est complètement spécifiée, la fenêtre présente une liste vide : aucun complément de spécification n'est nécessaire.



La liste **Proposition** permet d'indiquer le critère utilisé pour compléter les clés primaires :

- **A partir des index uniques** : les colonnes qui appartiennent à un index unique sont proposées comme composantes de la clé primaire.
- **A partir des colonnes obligatoires** : ces colonnes sont proposées comme composantes d'une clé.
- **Par rapprochement des noms** : si le même nom de colonne est retrouvé dans plusieurs tables, la colonne est proposée comme clé primaire.

Chaque clé est proposée sous la table à laquelle elle appartient.

Pour valider une clé primaire :

1. Cochez la case de la colonne **Périmètre** qui correspond à la clé.
Les colonnes associées sont automatiquement cochées par défaut. Vous pouvez éliminer celles qui répondent aux critères de recherche mais ne sont pas des constituants de la clé.
2. Cliquez sur le bouton **Appliquer**.
Le bouton **Appliquer** retire de la liste des propositions les clés explicitement acceptées ou refusées.

Pour les clés étrangères, deux clés incluant une même colonne sur une même table sont incompatibles : l'acceptation de l'une provoque automatiquement le rejet de l'autre.

Il n'est pas possible de sélectionner plusieurs clés primaires sur une même table : l'acceptation d'une clé provoque le rejet des autres.

☺ *Il vous est possible de compléter la spécification des clés en plusieurs passages. Cela vous permet de consulter le contenu de la base pendant que vous faites vos choix. Pour cela :*

- Cliquez sur le bouton **Appliquer** pour enregistrer vos modifications.
- Cliquez sur le bouton **Annuler** pour quitter cette fenêtre sans lancer de traitement.

Spécifier les clés étrangères

Pour spécifier les clés étrangères d'une base de données :

1. Faites un clic droit sur la base de données et sélectionnez **Compléter les clés**.

La fenêtre de **Compléter les clés** apparaît.

☛ *Lorsque la base est complètement spécifiée, la fenêtre présente une liste vide : aucun complément de spécification n'est nécessaire.*

La liste **Proposition** permet d'indiquer le critère utilisé pour compléter les clés étrangères :

- **A partir des index**
- **Par rapprochement des noms**

Si la proposition est faite à partir des index, la proposition se base sur les index non uniques de la table. La table de référence est indiquée après le nom de la clé.

Pour valider une clé étrangère :

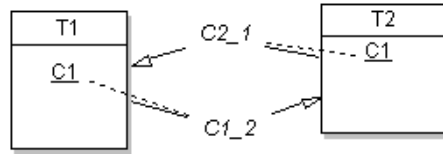
1. Cochez la case de la colonne **Périmètre** qui correspond à la clé.
2. Cliquez sur le bouton **Appliquer**.
Le bouton **Appliquer** retire de la liste des propositions les clés explicitement acceptées ou refusées.

Lorsqu'aucune table de référence n'est définie, l'assistant en propose automatiquement. Les clés qui n'ont pas de table de référence ne peuvent être acceptées.

Lors de la proposition de clés, on peut trouver plusieurs tables possédant une clé primaire identique. Ce pourrait être le cas par exemple, pour les tables correspondant aux différents sous-types d'une même entité.

Colonne clé primaire de deux tables

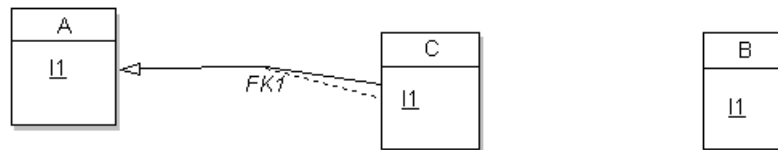
Quand la même colonne est clé primaire de deux tables, la proposition de clé étrangères propose la création de chacune des deux clés.



Il convient alors de choisir laquelle des deux clés est effectivement à prendre en compte.

Colonne clé primaire de trois tables

Quand la même colonne est clé primaire de trois tables, la proposition de clés permet de créer une seule clé étrangère à partir d'une colonne d'une table.



La proposition de clé permet de créer une seule clé étrangère à partir de la table C. Il faut aller rajouter l'autre dans la saisie des tables de la base de données.

TYPES DES DONNÉES ET DATATYPES DES COLONNES

Les données n'ont pas toutes le même type de valeur. Déterminer le type des données permet d'indiquer leur format et ainsi de favoriser leur manipulation par les différents outils de traitement de données.

Hopex gère les types de données aux différents niveaux de modélisation, en assurant la correspondance entre les types de données du niveau logique et les types de données pris en charge par les différents SGBD supportés.

Types de données des attributs

Un type de données permet de mettre en commun des caractéristiques communes à plusieurs attributs.

Pour typer les attributs d'une entité, ne sont proposés que les types de données définis pour le *modèle de données* qui contient cette entité.



Un modèle de données permet de représenter la structure statique d'un système, en particulier les types d'objets manipulés dans le système, leur structure interne et les relations qui existent entre eux. Un modèle de données regroupe un ensemble d'entités avec leurs attributs, les associations qui existent entre ces entités, des contraintes qui portent sur ces entités et associations, etc.

Pour plus de détails sur les types de données et les paquetages de types de données de référence, voir [Types de données](#).

Déduire les datatypes des colonnes à partir des types des attributs

Les types de données définis au niveau logique ne sont pas toujours compréhensibles pour le SGBD cible. Ils nécessitent dans ce cas d'être convertis en types de données correspondant au SGBD visé.

Cette conversion intervient notamment lors de la synchronisation. Les types de données des attributs définis dans le modèle logique sont traduits en datatypes pour les colonnes générées.

La conversion est assurée par un lien d'équivalence avec des types pivots. Les types pivots constituent un intermédiaire entre les types de données logiques et les datatypes générés.

Types pivots

Les types pivots sont des types de données définis indépendamment du SGBD cible et que vous pouvez utiliser quand vous ne connaissez pas encore le système dans lequel sera hébergée la base de données, ou quand plusieurs systèmes sont susceptibles d'être utilisés.

Les types pivots ont un datatype équivalent dans chaque SGBD supporté. Ils permettent ainsi de définir une seule fois les types des attributs et de les réinterpréter ensuite en fonction du SGBD cible.

Pour disposer des datatypes d'un SGBD, vous devez importer le module correspondant. Voir [Importer une version de SGBD](#).

Liste des types pivots

Une fois importés, les types pivots sont disponibles dans l'onglet de navigation **Données logiques**, sous le paquetage de types de données "Pivot".

Types alphanumériques

		Compléments
P-String	Chaîne de caractères alphanumériques	
P-Text	Chaîne de caractères alphanumériques	
P-Character	Chaîne de caractères alphanumériques de taille fixe	Longueur
P-Varchar	Chaîne de caractères alphanumériques de taille variable	

Types numériques

P-Decimal	Décimal	
P-Double		
P-Float		
P-Integer	Entier	
P-Long Integer		
P-Long Real		
P-Real		
P-Smallint		
P-Tinyint		
P-Numeric	Numérique	Longueur, Décimale
P-Currency	Montant exprimé en monnaie	Longueur, Décimale

Types dates

P-Date	Date
P-Time	Heure
P-DateTime	Date et heure

Types binaires

P-Binary	Chaîne binaire
P-Byte	Chaîne binaire
P-Timestamp	Identification générée automatiquement à partir de la date et de l'heure exprimée en millièmes de secondes après le 01 Janvier 1970
P-Boolean	Booléen valant 0 ou 1
P-Multimedia	Chaîne binaire
P-Varbinary	Chaîne binaire

Relier un type de données à un type pivot

Les types de données contenus dans le paquetage "Référence des types de données" et associés par défaut à tout nouveau modèle de données sont reliés à ces types pivots. Aussi, lorsque vous créez de nouveaux types de données, il est nécessaire de les relier aux types pivots correspondants afin qu'ils puissent être exploités par la suite au niveau physique.

Pour relier un type de données à un type pivot :

1. Ouvrez les propriétés du type de données.
2. Cliquez sur la page **Caractéristiques**.
3. Dans le champ **SQL Datatype**, sélectionnez le type pivot.

Prenons le type de données "Code". Ouvrez sa fenêtre de propriétés et cliquez sur la page **Caractéristiques**. Dans le champ **SQL Datatype**, vous pouvez voir qu'il est relié au type pivot "P-Character".

Caractéristiques

Nom Local: Code

Défendeur: Paquetage des types de données [Référence des types de données](#)

Longueur:

Décimale:

SQL Datatype: P-Character

Lors de la synchronisation d'un modèle logique vers un modèle physique, ce type pivot "P-Character" donnera un datatype CHAR, VARCHAR, LONG ou TEXT suivant le SGBD concerné par la synchronisation. Vous pouvez modifier le SGBD cible sans avoir à modifier le type de données, **Hopex** assure la conversion automatique. Voir [Correspondances entre types pivots et datatypes](#).

Relier un type de données à un type pivot dans la notation UML

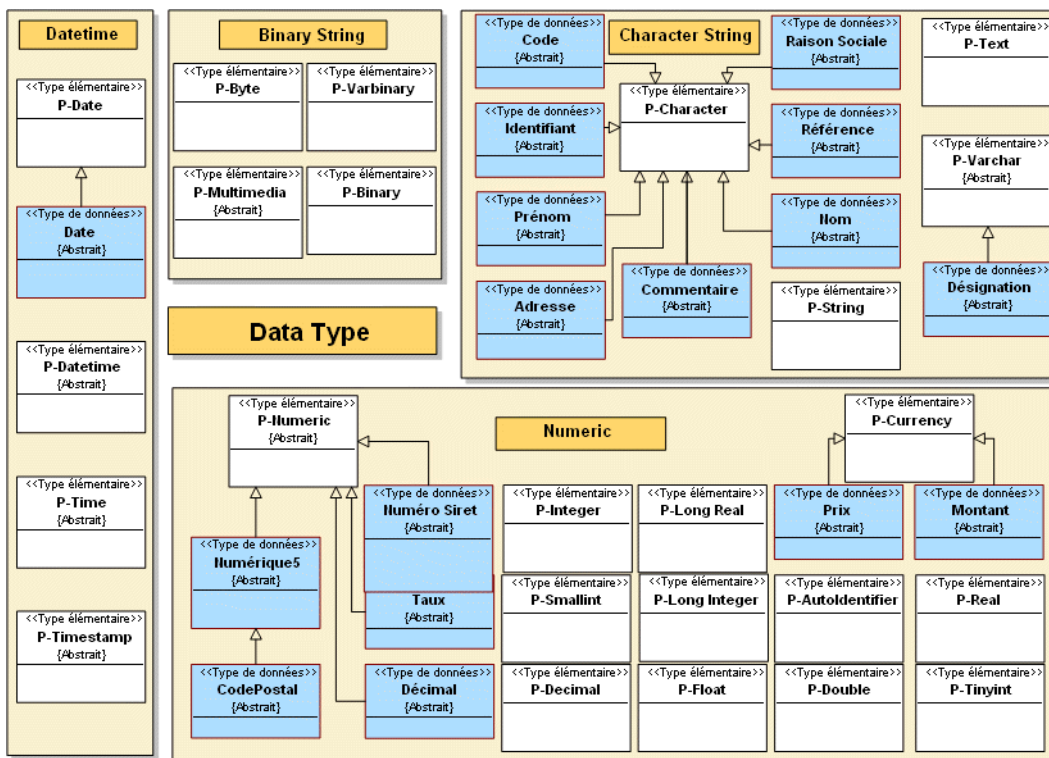
Si vous utilisez la notation UML et les diagrammes de classes pour modéliser vos données - et à titre de compatibilité avec les versions antérieures de **Hopex Database** - d'autres méthodes de référencement des types pivots sont possibles.

Vous pouvez créer de nouveaux types de données et les relier à des types pivots :

- Par héritage
- Par un lien de correspondance
- Par un lien d'équivalence
- En créant un type de données composé

Par héritage

Vous pouvez définir vos propres types de données en les déclarant sous-classes des types pivots proposés en standard comme dans l'exemple ci-dessous.



Les types de données définis comme sous-classes vont hériter automatiquement des caractéristiques de leur super-classe. En particulier, la règle de transformation en datatype de la super-classe est appliquée à la sous-classe.

Il est possible de préciser sur la sous-classe une longueur et un nombre de décimales. Ceux-ci seront pris en compte pour la génération des datatypes s'ils n'ont pas déjà été définis pour la super-classe.

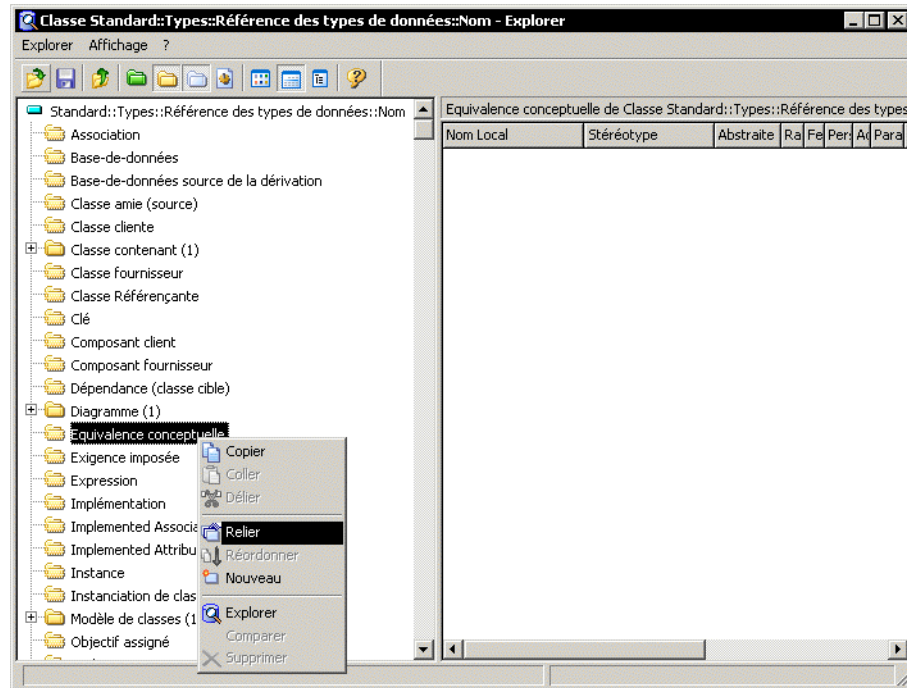
Par un lien de correspondance

Pour effectuer ce lien :

1. Ouvrez la fenêtre de propriétés de la classe.
2. Cliquez sur la page **Génération** > **SQL**.
3. Indiquez le **Type SQL** associé à la classe.

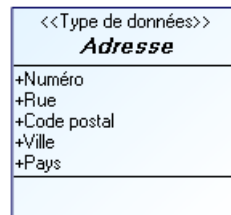
☛ *Seuls les types pivots du paquetage Standard::Types::Pivot sont proposés dans la liste.*

4. Précisez si nécessaire la longueur et le nombre de décimales à appliquer.



En créant un type de données composé

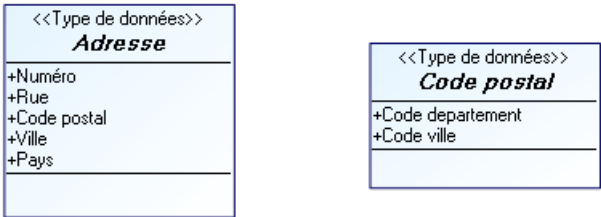
On peut définir un type de données composé en lui précisant une liste d'attributs.



Ici le type "Adresse" est composé du numéro, de la rue, du code postal, de la ville et du pays.

Un attribut de type "Adresse" donnera lieu lors de la dérivation à ces cinq colonnes.

Il est possible de décomposer un type à plusieurs niveaux en affectant un type décomposé à l'un de ses attributs.
Par exemple, on peut décomposer le code postal en code ville et code département.



Correspondances entre types pivots et datatypes

Les types pivots établissent une correspondance entre les types de données logiques auxquels ils sont reliés et les datatypes pour lesquels ils ont un équivalent dans chaque SGBD cible.

Les liens d'équivalence comportent des conditions qui permettent de les distinguer les uns des autres.

Pour visualiser les correspondances entre les types pivots et les datatypes des différents SGBD supportés, voir [Tableaux de correspondances entre types pivots et datatypes](#).

Exemple de correspondances entre les types pivots et les datatypes pour Oracle 8

Pivot vers Datatype

Pivot	Condition	Datatype
P-AutoIdentifier		NUMBER
P-Binary		RAW(@L)
P-Boolean	L<2 or L ∅	RAW(1)
	L>1	RAW(@L)
P-Byte		RAW(1)
P-Character	L<256 or L ∅	CHAR(@L)
	L>2000	LONG
	255<L<2001	VARCHAR2(@L)
P-Currency		NUMBER(@L,@D)

Pivot	Condition	Datatype
P-Date		DATE
P-Datetime		DATE
P-Decimal		NUMBER(@L,@D)
P-Double		NUMBER(@L,@D)
P-Float		NUMBER(@L,@D)
P-Integer		NUMBER(@L)
P-Long Integer		NUMBER(@L)
P-Long Real		NUMBER(@L,@D)
P-Multimedia		LONG RAW
P-Numeric	L=0 or L ø	NUMBER
	L>0 et D ø	NUMBER(@L)
	L>0 and D not ø	NUMBER(@L,@D)
P-Real		NUMBER(@L,@D)
P-Smallint		NUMBER(@L)
P-String		LONG
P-Text		VARCHAR2(@L)
P-Time		DATE
P-Timestamp		ROWID
P-Tinyint		NUMBER(@L)
P-Varbinary		LONG RAW
P-Varchar	L>2000 or L=0 or L ø	LONG
	0<L<2001	VARCHAR2(@L)

Datatype vers Pivot

Datatype	Condition	Pivot
CHAR(L)		P-Character
DATE		P-Date
LONG		P-String
LONG RAW		P-Multimedia
NUMBER		P-Numeric

Datatype	Condition	Pivot
NUMBER(L)		P-Numeric
NUMBER(L,D)		P-Numeric
RAW(1)		P-Boolean
RAW(L)		P-Boolean
ROWID		P-Timestamp
VARCHAR2(L)		P-Varchar

Dans ce tableau, on peut voir que le type "P-Numeric" a trois correspondances pour les classes types, grâce à trois conditions différentes sur les liens d'équivalence.

Exemple : si P_Numeric est affecté à un attribut et que la longueur de cet attribut est de 10, alors la colonne justifiée par cet attribut via la synchronisation donnera Number(10).

La condition est écrite en langage VB Script. Les principaux éléments de la condition sont les suivants :

- **Sub ConditionInvoke (Colonne, ByRef bValid)** : la première ligne constitue la signature de la fonction.
- **Colonne** : la colonne est donnée en paramètre d'entrée.
- **bValid** : c'est le paramètre de retour. Sa valeur est "True" si la condition est vérifiée, "False" en cas contraire.

Exemple:

```
Sub ConditionInvoke (Colonne, ByRef bValid)
    bValid = False
    If (IsNumeric(Colonne.Length)) Then bValid = True
End Sub
```

Dans la condition, il est possible de préciser les éléments suivants :

- Présence d'un nombre
- Présence d'une décimale
- Plage concernée (exemple : compris entre 0 et 150)

Créer de nouveaux datatypes

Chaque datatype est implémenté sous forme de classe ; il est propre à une version de SGBD. Il est possible d'utiliser des masques avec les datatypes.

Exemple pour Oracle 10

Objectif

Faire apparaître un datatype Data8 (numérique avec une longueur et une décimale) dans les scripts ORACLE.

Étapes

Les étapes sont les suivantes :

1. Créez un nouveau datatype dans **Hopex**.
2. Reliez le datatype à la version du SGBD cible (en l'occurrence Oracle 10).
3. Reliez le datatype au type correspondant dans le paquetage "Pivot".
4. Paramétrez les conditions sur chaque lien, et ce dans les deux sens (du datatype aux types pivots et du type pivot aux datatypes).

☛ Pour plus d'informations sur les liens d'équivalence et les conditions, voir [Déduire les datatypes des colonnes à partir des types des attributs](#).

Conditions préalables

Pour voir apparaître les paquetages contenant les datatypes des SGBD, vous devez importer le module correspondant. Voir [Importer une version de SGBD](#).

💡 **Il est recommandé de ne définir un datatype que dans une seule version de SGBD.**

Pour importer le module des datatypes :

1. A partir de votre version HOPEX, ouvrez la console d'administration HAS Console.
2. Dans le menu de navigation, cliquez sur **Modules**.
La console HAS affiche :
 - les modules installés
 - le store dans lequel vous pouvez télécharger des modules
 - les modules à mettre à jour
3. Dans le store, recherchez le module à installer puis téléchargez-le.

De plus, certaines données sont protégées dans **Hopex**. Afin de pouvoir modifier les objets contenus dans les paquetages des SGBD :

1. Ouvrez le bureau d'administration avec le profil Administrateur HOPEX.
2. Dans la partie supérieure droite du bureau, cliquez sur le menu associé au compte de l'administrateur puis cliquez sur **Options**.
3. Dans les options, déployez le dossier **Installation**.
4. Cliquez sur le dossier **Personnalisation**.
La liste des options liées à la personnalisation apparaît dans la partie droite de la fenêtre.
5. Dans le champ **Autoriser la modification des données HOPEX**, sélectionnez "Autoriser".
6. Cliquez sur **OK**.

Créer un nouveau datatype

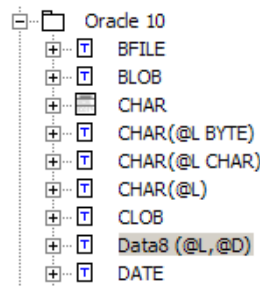
Pour créer un nouveau datatype dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture > Dictionnaires de données**.

➡ Dans **Hopex Data Architecture**, cliquez sur le menu de navigation **Dictionnaires > Vue hiérarchique**.

2. Dépliez le dossier des dictionnaires de données.
3. Cliquez sur l'icône du paquetage "Oracle 10" et sélectionnez **Nouveau > Classe**.
La fenêtre de **Création d'une classe** s'ouvre.
4. Nommez votre classe "Data8 (@L,@D)".
5. Ouvrez la fenêtre de propriétés de cette nouvelle classe.
6. Dans la page **Caractéristiques**, sélectionnez dans le champ **Stereotype** la valeur "Expression".
7. Dans le champ **Type expression** qui apparaît, un peu plus bas, sélectionnez la valeur "Data8 (@L,@D)".

Vous pouvez constater dans le navigateur qu'une nouvelle classe "Data8" a été créée.



➡ Cette nouvelle classe est créée automatiquement pour les besoins de fonctionnement d'UML.

Relier le datatype au type pivot

Si vous voulez obtenir ce datatype après synchronisation, vous devez lui donner un équivalent au niveau logique :

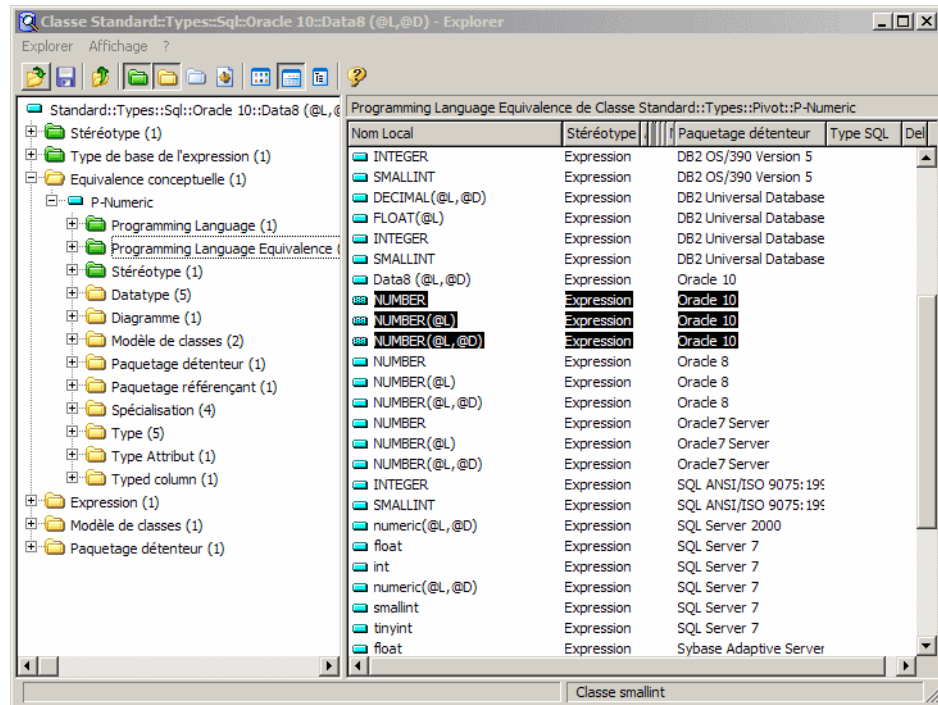
1. Ouvrez la fenêtre de propriétés du datatype "Data8 (@L,@D)".
2. Cliquez sur la page **Compléments**.
3. Faites un clic droit sur le dossier "Equivalence conceptuelle" et sélectionnez **Relier**.
4. Dans la fenêtre de recherche, sélectionnez la classe "P-Numeric".

Paramétrer les conditions sur les liens

Pour paramétrer la condition sur les liens :

1. Faites un clic droit sur la classe "Data8" et sélectionnez **Explorer**.
2. Dépliez le dossier "Equivalence conceptuelle".

3. Sélectionnez le dossier vert "Programming Language Equivalence".
Vous constatez qu'il existe trois autres correspondances pour Oracle 10.



Il s'agit donc de modifier les conditions sur ces correspondances pour qu'elles soient cohérentes avec les conditions posées sur le nouveau datatype.

4. Faites un clic droit sur le datatype "NUMBER(@L,@D)" et sélectionnez **Propriétés**.
5. Dans la page **Textes**, sélectionnez "Condition d'équivalence de langage", et modifiez le texte pour qu'il corresponde à ceci :

```
Sub ConditionInvoke (Colonne, ByRef bValid)
    bValid = False
    Dim IsNumericLength
    IsNumericLength = IsNumeric(Colonne.Length)
    Dim IsNumericDecimal
    IsNumericDecimal = IsNumeric(Colonne.Decimal)
    If (IsNumericLength and IsNumericDecimal) Then
        If (Colonne.Length <> 8) Then
            bValid = True
        End If
    End If
End Sub
```


- De la même façon, ajoutez le texte suivant dans la fenêtre de propriétés du nouveau datatype "Data8".

```
Sub ConditionInvoke (Colonne, ByRef bValid)
    bValid = False
    Dim IsNumericLength
    IsNumericLength = IsNumeric(Colonne.Length)
    Dim IsNumericDecimal
    IsNumericDecimal = IsNumeric(Colonne.Decimal)
    If (IsNumericLength and IsNumericDecimal) Then
        If (Colonne.Length = 8) Then
            bValid = True
        End If
    End If
End Sub
```

Vérifier les datatypes

Pour vérifier les datatypes :

- Dans les propriétés de la base de données, cliquez sur la page **Tables**.
- Ouvrez les propriétés de la table.
- Cliquez sur la page **Colonnes**.
- Affichez les propriétés d'une colonne concernée par les conditions posées sur le datatype.
- Vérifiez que le masque affiché dans la colonne **Datatype** est bien "Data8 (%l, %d)" pour cette colonne.

Exemple pour SQL Server 7

Objectif

Relire des colonnes SQL Server 7 comportant un datatype non standard.

Les manipulations sont les mêmes que pour Oracle (voir [Exemple pour Oracle 10](#)). Cette fois-ci, nous ne créerons pas de masque.

Créer un nouveau datatype

Pour créer un nouveau datatype :

- Dans le bureau, cliquez sur le menu de navigation puis sur **Architecture des données > Données logiques**.
- Cliquez sur la tuile **Dictionnaires de données**.
- Cliquez sur l'icône du paquetage "Oracle 10" et sélectionnez **Nouveau > Classe**.
La fenêtre de **Création d'une classe** s'ouvre.
- Nommez votre classe "TLibelleLong".
- Ouvrez la fenêtre de propriétés de cette nouvelle classe.
- Dans la page **Caractéristiques**, sélectionnez dans le champ **Stéréotype** la valeur "Expression", puis cliquez sur **OK**.

Relier le datatype au type pivot

Pour relier le datatype au type pivot :

1. Ouvrez la fenêtre de propriétés du datatype "TLibelleLong".
2. Sélectionnez la page **Compléments**.
3. Faites un clic droit sur le dossier "Equivalence contextuelle" et sélectionnez **Relier**.
4. Dans la fenêtre de recherche, sélectionnez la classe "P-Text".

Paramétrer les conditions sur les liens

Pour paramétrer la condition sur les liens :

1. Faites un clic droit sur la classe "TLibelleLong" et sélectionnez **Explorer**.
2. Sélectionnez le dossier vert "Programming Language Equivalence". Vous constatez qu'il existe une autre correspondance pour SQL Server 7.
3. Ouvrez la fenêtre de propriétés du datatype "text".
4. Dans la page **Textes**, sélectionnez "Condition d'équivalence de langage", et modifiez le texte pour qu'il corresponde à ceci :

```
Sub ConditionInvoke (Colonne, ByRef bValid)
    bValid = False
    Dim IsNumericLength
    IsNumericLength = IsNumeric(Colonne.Length)
    If (IsNumericLength) Then
        If (Colonne.Length > 255) Then
            bValid = True
        End If
    End If
End Sub
```

5. De la même façon, ajoutez le texte suivant dans la fenêtre de propriétés du nouveau datatype "TLibelleLong".

```
Sub ConditionInvoke (Colonne, ByRef bValid)
    bValid = False
    Dim IsNumericLength
    IsNumericLength = IsNumeric(Colonne.Length)
    If (IsNumericLength) Then
        If (Colonne.Length <= 255) Then
            bValid = True
        End If
    End If
End Sub
```

RÈGLES DE MODÉLISATION D'UNE BASE DE DONNÉES

Hopex Data Architecture fournit un ensemble de règles qui permettent de contrôler la modélisation des bases de données. Le règlement physique contient les règles relatives au domaine relationnel d'une base de données. Il est utilisé pour vérifier le domaine relationnel correspondant dans un SGBD.

Le règlement physique contient les règles relatives aux spécifications techniques propres au SGBD de la base de données. Il est utilisé pour vérifier la cohérence des paramètres physiques du domaine relationnel spécifique au SGBD.

Contrôler une base de donnée

Vous pouvez lancer un contrôle sur la base de données ou sur un objet de la base de données.

Pour contrôler une base de données :

1. Faites un clic droit sur le nom de la base de données.
2. Sélectionnez **Administrer > Contrôler > Règlement avec Propagation**.

Lorsque plusieurs règlements peuvent s'appliquer à l'objet contrôlé, une fenêtre vous demande de sélectionner le règlement voulu.

Le contrôle s'applique à la base de données ainsi qu'aux objets qu'elle détient.

Les résultats apparaissent dans un rapport HTML.

Pour plus de détails sur les contrôles, voir le guide **Hopex Common Features**, "Explorer le référentiel", "Les outils de contrôle des objets".



CATÉGORIES DE DONNÉES



Vous pouvez classer les données du référentiel par catégorie. Un arbre dédié répertorie les différentes catégories et les données associées. Les données classées peuvent être exploitées dans la solution **Hopex Privacy Management** spécifique à la gestion des données sensibles et de la conformité au RGPD.

DÉFINIR DES CATÉGORIES DE DONNÉES

Les solutions **Hopex Data Governance** et **Hopex Data Architecture** fournissent une liste de catégories qui permettent de classifier les données. Vous pouvez compléter cette liste en important le module **Sample Privacy Management Content**.

➡ Pour plus d'informations sur l'import d'un module, voir [Importer un module dans HOPEX](#).

Vous pouvez également créer des catégories.

Exemples de classification des données :

- Données sensibles
- Données de référence
- Données confidentielles
- Etc.

Accéder aux catégories de données

Pour accéder aux catégories de données dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture**.
2. Cliquez sur le bouton de la **Vue hiérarchique**  associé aux **Dictionnaires de données**.
3. Dans la zone d'édition, déployez le dossier **Catégories de données**.

Créer une catégorie de données

Pour créer une catégorie de données :

1. Dans la zone d'édition, faites un clic droit sur le dossier **Catégories de données**.
2. Cliquez sur **Nouveau > Catégorie de données**.
3. Dans l'assistant de création saisissez le nom de la catégorie et éventuellement un détenteur.
4. Cliquez sur **OK**.
La catégorie apparaît dans la liste. Vous pouvez saisir une description.

ASSOCIER LES CATÉGORIES AUX DONNÉES

Définir une catégorie sur une donnée métier

Une donnée peut appartenir à plusieurs catégories de données.

Pour indiquer à quelle catégorie appartient une donnée :

1. Ouvrez les propriétés de la donnée en question.
2. Cliquez sur la page **Caractéristiques**.
3. Sous la section **Catégories de données**, sélectionnez la catégorie en question.

Associer une catégorie de données à une réglementation

Vous pouvez indiquer qu'une réglementation porte sur une catégorie de données, par exemple, les données "sensibles". Toutes les données qui entrent dans cette catégorie sont soumises à la réglementation.

Pour définir une catégorie de données sur une réglementation :

1. Cliquez sur le menu de navigation puis sur **Conformité > Textes de référence**.
2. Ouvrez les propriétés de la réglementation.
3. Cliquez sur la page **Catégories de données contraintes**.
4. Cliquez sur **Nouveau** pour créer une catégorie ou **Relier** pour relier une catégorie existante.

Vous pouvez également définir des catégories dans **HOPEX** à partir de données importées via un fichier Excel. Voir [Import et export de données](#).

Voir aussi : [Dendrogramme des catégories de données](#).



RESPONSABILITÉ DES DONNÉES



Hopex Data Governance et **Hopex Data Architecture** vous permettent de désigner des responsables (de conception, de qualité, etc.) sur les données du référentiel.

- ✓ [Usage des responsabilités sur les données](#)
- ✓ [Définir les responsabilités sur les données](#)

USAGE DES RESPONSABILITÉS SUR LES DONNÉES

Assigner des responsables sur les données qui circulent au sein d'une organisation offre plusieurs avantages :

- Suivre la progression de la conception des données. Voir [Workflow de validation des données](#).
- Communiquer et collaborer sur les données : des demandes de mise à jour ou de validation sur des données peuvent être envoyées aux responsables désignés. Ces derniers reçoivent alors une notification et peuvent suivre leurs tâches dans **Hopex**. Voir [Accéder à ses notifications](#).
- Assurer la traçabilité des données. Voir [Lignages de données](#).
- Évaluer la donnée, directement à titre d'expert, ou par une campagne d'évaluation, suivi de plans d'action si nécessaire. Voir [Qualité des données](#).

A cet effet, **Hopex Data Governance** et **Hopex Data Architecture** proposent différents rôles par défaut, tels que le Propriétaire de données ou le Responsable de la qualité des données.

Rôles applicables aux données

- **Directeur de données (DCO)** : responsable des données et de leur gouvernance.
- **Propriétaire de données** : c'est l'autorité qui décide de l'accès aux données et de leur utilisation. Le propriétaire des données peut être le concepteur des données, un de ses utilisateurs ou une tierce partie. Les data stewards peuvent demander aux propriétaires de données de vérifier ou de compléter la valeur d'un champ, pour corriger par exemple un défaut de qualité de données.
- **Concepteur de données** (Data Designer) : personne responsable de la conception d'un objet (tel qu'un paquetage, un domaine de données, une bases de données, etc.).
- **Coordinateur de données** (Data Steward) : le Coordinateur de données est le chef de file d'un projet de gouvernance des données. Il joue un rôle clé dans sa réalisation. Il a la connaissance des données et de leurs métadonnées. Il se les approprie et travaille avec les équipes métier pour définir les objectifs du projet de gouvernance des données.
- **Scientifique de données** (Data Scientist) : il est en charge du rapprochement entre le concepteur des données (métier et logiques) et les responsables des processus qui utilisent ces données.
- **Responsable de la qualité des données** (Data Quality Manager) : doit s'assurer de la pertinence et de l'utilité des données de l'entreprise. Pour cela il doit mettre en place des procédures de contrôle des données.

Accéder à ses notifications

Pour visualiser vos notifications dans Hopex Data Architecture :

- Dans la partie supérieure droite du bureau, cliquez sur l'icône **Notifications**.

Pour plus d'information sur les outils de collaboration de la plateforme voir [Communiquer dans HOPEX](#).

DÉFINIR LES RESPONSABILITÉS SUR LES DONNÉES

Lors de leur création, certaines données (métier, logique et physique) sont automatiquement assignées à un **Concepteur** et un **Propriétaire**. Par défaut il s'agit de la personne qui crée la donnée, mais il est possible de lui associer d'autres personnes.

☛ Sur certains types d'objet de **Hopex Data Architecture**, tels que les éléments d'information sur les concepts par exemple, les assignations ne sont pas automatiques et doivent être réalisées si besoin de manière explicite.

Visualiser les responsables d'un objet

Pour accéder aux informations qui concernent l'assignation d'un objet :

1. Ouvrez les propriétés de l'objet en question.
2. Dans la page **Caractéristiques**, affichez la section **Responsabilités**.

☛ Seuls les objets assignables ont une page **Responsabilités**.

Assignation automatique d'un objet

Lors de la création de certains objets, une assignation est automatiquement créée. L'objet est assigné à la personne qui l'a créé avec le rôle **Concepteur de données** et **Propriétaire de données**.

Un responsable est automatiquement assigné aux objets suivants :

- Carte d'information métier / carte de données logiques ou physiques
- domaine de concepts / domaine de données logiques ou physiques
- Lignage de données
- Les conteneurs tels que les dictionnaires de données

Ces objets sont visibles dans les listes d'objets de l'utilisateur en question (par exemple **Mes domaines d'information métier**).

Assignation explicite d'un objet

Vous pouvez définir explicitement l'assignation d'un objet à une personne existante.

Pour assigner un objet à une personne :

1. Ouvrez la fenêtre de propriétés de l'objet assignable.
2. Sélectionnez la page **Caractéristiques > Responsabilités**.
3. Cliquez sur **Nouveau**.
Une fenêtre **Création d'une nouvelle responsabilité** s'ouvre.

4. A partir du champ **Personne ou Groupe de personnes**, cliquez sur **Relier**.
Une fenêtre de **Connexion** s'ouvre.
5. Recherchez et sélectionnez la personne qui vous intéresse et cliquez sur **Relier**.
6. Dans la fenêtre de **Création d'une nouvelle responsabilité**, sélectionnez le **Rôle métier** de la personne que vous avez sélectionnée.
 Pour plus de détails sur les rôles métier utilisés pour les assignations, voir [Les rôles métier de HOPEX Information Architecture](#).
7. Cliquez sur **OK**.
Une nouvelle assignation est ajoutée à la liste des assignations associées à l'objet.

Consulter les rapports de gouvernance des données

Les rapports de gouvernance des données sont un ensemble de rapports qui listent pour chaque rôle métier le nombre et le détail des données concernées.

Les rôles métier sont les suivants :

- Chief Data Officer (DCO)
- Propriétaire de données
- Concepteur de données (Data Designer)
- Scientifique de données (Data Scientist)
- Responsable de la qualité des données (Data Quality Manager)
- Coordinateur de données (Data Steward)

Chaque rapport énumère dans un graphique en barres les personnes qui détiennent le rôle en question, par exemple les personnes qui sont propriétaires de données. Il affiche pour chacune d'elles le détail des informations dont elles sont responsables.

Sous le rapport, des détails supplémentaires sont affichés sur le responsable des données :

- nom
- email
- numéro de téléphone
- nombre d'informations assignées

Lorsque vous cliquez sur une barre du graphique, la liste des informations s'affiche avec les détails suivants :

- nom des données
- commentaire
- état d'avancement dans la conception
- propriétaire



UTILISATION DES DONNÉES PAR LE SI



Hopex Data Governance vous permet de définir et de visualiser quelles données sont utilisées par quels processus et applications.

DÉFINIR LES DONNÉES UTILISÉES PAR LES APPLICATIONS

Hopex Data Governance vous permet de faire un inventaire des applications qui utilisent les données de votre référentiel.

Grâce à la plateforme HOPEX intégrée vous pouvez exploiter cet inventaire dans les solutions HOPEX spécifiques à la description de l'architecture applicative, telles que **Hopex IT Architecture** ou **Hopex IT Portfolio Management**.

Grâce à des rapports dédiés vous pouvez visualiser dans quelles applications sont utilisées certaines données. Voir [Rapports d'utilisation des données](#).

Créer des applications avec Hopex Data Governance

L'inventaire du patrimoine applicatif est réalisé par l'Administrateur fonctionnel des données.

Pour créer une application dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Inventaires > Applications**.
2. Cliquez sur l'onglet **Applications** puis sur le bouton **Nouveau**.
3. Indiquez le nom de l'application et éventuellement son détenteur.
4. Cliquez sur **OK**.

De la même façon vous pouvez créer des systèmes applicatifs, des services applicatifs, des microservices et des scénarios de flux applicatifs.

Pour plus de détails sur ces objets de l'architecture applicative, voir [Modéliser les applications et les architectures système](#).

Relier des données à une application

Le Gestionnaire de données a la possibilité de définir les données qui sont utilisées par les applications.

Les données que vous associez à une application doivent être définies dans un domaine de données qui est relié à l'application par un dépôt de données.

☛ *L'application peut référencer plusieurs domaines de données. Chaque domaine de données est associé à un dépôt de données spécifique pour cette application.*

Le dépôt de données offre un mécanisme de stockage et de consultation des données par l'application.

Le domaine de données détermine le périmètre des données utilisées par l'application.

Hopex offre une façon simplifiée de relier des données à une applications : lorsque vous reliez une donnée à une application, cette donnée est intégrée automatiquement au domaine de données associé par défaut à l'application. Si aucun domaine n'a été défini sur l'application, Hopex en crée un automatiquement.

Le dépôt de données sous-jacent est également créé automatiquement. Par la suite, toutes les données associées à l'application sont intégrées dans le domaine par défaut.

Si vous souhaitez définir des données dans un nouveau domaine, vous devez créer un nouveau dépôt de données sur lequel stocker les données. Voir [Créer un dépôt de données sur une application](#).

Pour relier des données à une application :

1. Ouvrez les pages de propriétés de l'application.
2. Cliquez sur la page **Caractéristiques**.
3. Dépliez la section **Données**.

 Cette page peut être cachée par défaut. Pour l'afficher, cliquez sur le bouton **Gérer les sections** et sélectionnez **Données**.

4. Sélectionnez le type d'objet à créer : métier, logique ou physique.
5. Cliquez sur **Nouveau**.
6. Dans l'assistant qui apparaît, sélectionnez le type d'objet qui constitue la donnée (par exemple, Classe, Concept, etc.) et l'objet en question.
7. Cliquez sur **Ajout**.

Une fois les données définies, vous pouvez spécifier :

- l'accès aux données : en création (**Create**), lecture (**Read**), modification (**Update**), Suppression (**Delete**).

 Le contenu de la colonne **Accès aux données** est calculé automatiquement en fonction des actions cochées ("CRUD" est la valeur par défaut).

- si l'application est "Source unique de vérité" ou "Golden copy" de ces données.

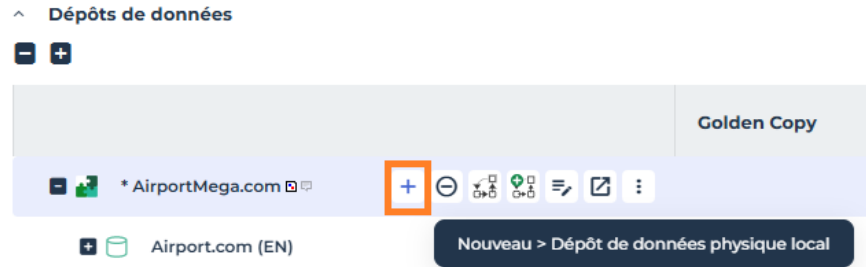
Créer un dépôt de données sur une application

La création d'un dépôt de données est associée automatiquement à celle d'un domaine de données du même nom.

Pour créer un dépôt de données sur une application :

1. Ouvrez les propriétés de l'application en question.
2. Cliquez sur la page **Composants**.
3. Allez sous la section **Dépôts de données**.
4. Passez la souris sur le nom de l'application.
Des commandes apparaissent à droite de l'application.

5. Cliquez sur le bouton **Nouveau > Dépôt de données physique local**.

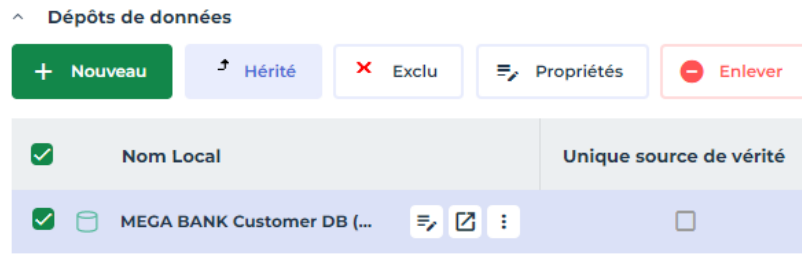


6. Dans la fenêtre qui apparaît, indiquez le domaine de données référencé par le dépôt de données et cliquez sur **OK**.
Le dépôt de données apparaît dans les propriétés de l'application. Un domaine de données du même nom lui est automatiquement associé.

Créer une donnée à partir d'un dépôt de données

Pour créer une donnée à partir du dépôt de données :

1. Sous la section **Dépôts de données**, sélectionnez le domaine de données référencé par le dépôt de données.



2. Les sections **Données métier** et **Données logiques** apparaissent.
3. Dans la section qui correspond au type de donnée à créer, cliquez sur **Nouveau**.

😊 Pour un mode de création simplifié, vous pouvez créer des données sans passer par la définition d'un dépôt de données, dans la page **Caractéristiques** de l'application > sections **Données métier** et **Données logiques**.

Définir le mode d'accès aux données

Pour définir le mode d'accès aux données (création, lecture, suppression, etc.) :

1. Dans les propriétés de l'application, sélectionnez le dépôt de données. Les données du domaine de données correspondant apparaissent.

2. Sélectionnez la donnée et cliquez sur le bouton **CRUD**.

The screenshot shows the 'Composants' tab with the following structure:

- Top navigation: Vue d'ensemble, Caractéristiques, **Composants**, Rapports, Impact de l'application, Diagrammes, and a settings gear icon.
- Section: Gérer les sections
- Composants internes
 - Dépôts de données
 - Buttons: + Nouveau, Hérité, Exclu, Propriétés, Enlever, Rapport instantané, and a menu icon.
 - Table:

☒	Nom Local ↑	Unique source de vérité	Golden Copy	Description
☒	Airport.com (EN)	☒	☐	
 - Données métier
 - Buttons: + Nouveau, Hérité, Exclu, Enlever, **CRUD** (highlighted), Rapport instantané, and a menu icon.
 - Table:

☒	Nom Local ↑	Responsable fonctionnel	Accès aux données
☒	Compte	☒	CRUD

3. Cochez les cases qui correspondent aux types d'accès voulus (Création, Lecture, etc.).

Définir l'application comme source de vérité

Une donnée contenue dans un domaine peut être utilisée par différents processus ou applications.

Une application définie comme "Unique source de vérité" est une application au sein du système d'information qui garantit la validité des données qu'elle détient.

Pour définir quelle est l'application considérée comme source de vérité pour la donnée, vous devez indiquer que :

- la donnée est sous la responsabilité fonctionnelle du domaine qui l'utilise (1)
- ce domaine est référencé par un dépôt de données tagué "Unique source de vérité"(2)

Dans l'exemple ci-dessous, l'application est "Unique source de vérité" pour la donnée "Aéroport de départ".

Données

Données métier Données logiques Données physiques

+ Nouveau Rapport instantané

	Nom Local ↑	Unique source de vérité	Golden Copy	Responsable fonctionnel	Accès aux données	Contexte
☐	Account Holder (EN)	☐	☐	☐	CRUD	Airport.com (EN)
☐	Aéroport de départ	☒	☐	☒	CRUD	Airport.com (EN)
☐	Aéroport de...	☐	☐	☐	CRUD	Airport.com (EN)

Définir une application Golden Copy

De la même façon que pour une application "Golden Source", vous pouvez indiquer qu'une application est "Golden Copy". Une application "Golden Copy" constitue une copie conforme de la source de vérité.

Impact des données utilisées

La page de propriétés **Impact des données** permet de visualiser les données utilisées par une application et de mesurer :

- l'impact de l'application sur les données qu'elle utilise (par exemple, l'impact de la suppression d'une application lorsqu'elle est source de vérité sur des données)
- et inversement, l'impact des données sur l'application (par exemple, l'impact de données qui ne sont pas fiables).

Pour visualiser l'impact des données sur une application :

1. Ouvrez les propriétés de l'application.
2. Cliquez sur la page **Impact des données**.

La page présente les étiquettes suivantes :

- **Données** : données utilisées par l'application
- **Données échangées** : données contenues dans les flux échangés par l'application.
- **Unique source de vérité** : données pour lesquelles l'application est déclarée "Unique source de vérité".
- **Données stratégique** : données de l'application classées dans la catégorie "Donnée stratégique".
- **Données sensibles** : données de l'application classées dans la catégorie "Données sensibles".
- **Qualité des données** : niveau de qualité des données utilisées par l'application. Le rapport fournit une moyenne de la qualité des données des dernières évaluations effectuées sur l'ensemble des

données utilisées par l'application, au niveau du dépôt de données de celle-ci ou à travers les échanges de flux avec d'autres applications.

3. Cliquez sur une étiquette pour afficher le détail.



Voir dans quelles applications est utilisée une donnée

Des rapports vous permettent de visualiser où sont utilisées les données dans votre parc applicatif.

Voir : [Rapports d'utilisation des données](#).

DÉFINIR LES DONNÉES UTILISÉES PAR LES PROCESSUS

Hopex Data Governance vous permet de faire un inventaire des processus qui utilisent les données de votre référentiel.

Grâce à la plateforme HOPEX intégrée vous pouvez exploiter cet inventaire dans les solutions HOPEX spécifiques à la description des processus, telles que **Hopex Business Architecture & Strategic Planning**.

Faire l'inventaire des processus

L'inventaire des processus est réalisé par l'administrateur fonctionnel des données.

Types de processus

Chaîne de valeur

Une chaîne de valeur est un ensemble d'étapes de création de valeur qui crée un résultat pour le client, qui peut être un client final ou un client interne dans la chaîne de valeur.

Les chaînes de valeur référencent uniquement des domaines de données métier.

Processus organisationnel

Un processus organisationnel est un ensemble d'opérations réalisées par des acteurs d'une entreprise ou d'une organisation en vue de produire un résultat. Il est décrit comme une séquence d'opérations, contrôlée par des événements et des conditions.

Les processus organisationnels peuvent référencer des domaines de données métier ou logiques.

Processus applicatifs

Un processus applicatif est la représentation exécutable d'un processus. Il définit un enchaînement des tâches.

Vous pouvez lui relier les domaines de données suivants :

- Domaine de données applicatif
- Domaine relationnel
- Domaine de données NoSQL
- Structure de fichier

Voir aussi :

[Domaines de données et domaines de données logiques](#)

[Carte de domaines relationnels et domaines relationnels.](#)

Créer un processus

Pour créer un processus dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Inventaires > Processus**.
2. Sélectionnez le type de processus à créer, par exemple Processus applicatif.
3. Cliquez sur le bouton **Nouveau**.
4. Indiquez le nom du processus et éventuellement son détenteur.
5. Cliquez sur **OK**.

Relier des données à un processus

Le Gestionnaire de données a la possibilité de définir les données qui sont utilisées dans les processus.

Pour relier des données logiques à un processus vous devez créer un dépôt de données sur le processus. Le dépôt de données offre un mécanisme de stockage et de consultation des données par le processus.

Le dépôt de données référence un domaine de données qui représente la structure des données dont le processus a besoin.

Créer un dépôt de données

Pour créer un dépôt de données sur un processus :

1. Ouvrez les propriétés du processus en question.
2. Cliquez sur la page **Dépôt de données**.
3. Cliquez sur **Nouveau**.
4. Dans la fenêtre qui apparaît indiquez le domaine de données référencé par le dépôt de données et cliquez sur **Ajouter**.
Le dépôt de données (représenté par le domaine de données) apparaît dans les propriétés du processus.

Définir le mode d'accès aux données

Dans les propriétés du domaine de données vous pouvez définir le mode d'accès aux données (création, lecture, suppression, etc.).

Pour accéder aux propriétés du domaine de données :

1. Dans les propriétés du processus, cliquez sur l'icône du dépôt de données puis sur **Propriétés**.
Les propriétés du domaine de données correspondant apparaissent.
2. Cliquez sur la page **Composants**.
3. Sélectionnez le composant et cochez les cases qui correspondent aux types d'accès voulus (Création, Lecture, etc.).

Voir aussi : [Managing Data](#).

Relier des données à un contenu

Les contenus correspondent aux informations échangées entre un processus et son environnement, via des flux.

Vous pouvez relier des données métier ou logiques à un flux ; elles sont définies comme composants du contenu.

DÉFINIR LES DONNÉES UTILISÉES PAR LES CAPACITÉS MÉTIER

Hopex Data Governance vous permet de faire un inventaire des capacités métier qui utilisent les données de votre référentiel.

Grâce à la plateforme HOPEX intégrée vous pouvez exploiter cet inventaire dans les solutions HOPEX spécifiques à la description des capacités métier, telles que **Hopex IT Business Management**.

Créer des capacités métier avec Hopex Data Governance

Pour créer une capacité métier dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Inventaires > Capacités métier**.
La liste des capacités métier apparaît.
2. Cliquez sur **Nouveau**.
3. Indiquez le nom de la capacité métier et éventuellement son détenteur.
4. Cliquez sur **OK**.

Relier des données à une capacité métier

Pour relier des données à une capacité métier, vous devez créer un domaine de concepts sur la capacité métier. Les données peuvent être modélisées dans le diagramme correspondant.

Une fois les données modélisées dans le diagramme, vous devez les rattacher au domaine décrit.

Pour créer un diagramme de domaine de concepts sur une capacité métier :

1. Cliquez sur la capacité métier.
Sa fenêtre de propriétés apparaît.
2. Cliquez sur la page **Diagrammes**.
3. Cliquez sur **Créer un diagramme**.
4. Sélectionnez le type de diagramme **Domaine de concepts**.
A partir de la barre d'objets, vous pouvez créer les éléments de votre diagramme. Pour plus de détails, voir [Construire un diagramme de concepts](#).

Pour indiquer qu'un objet, par exemple un concept, appartient au domaine décrit :

1. Faites un clic droit sur l'objet en question pour faire apparaître son menu contextuel.
2. Sélectionnez **Ajouter au domaine courant**.
L'élément est relié au domaine de concepts et par là-même à la capacité métier sur laquelle il porte.

Pour visualiser les données métier reliées à la capacité métier :

1. Ouvrez les propriétés de la capacité métier.
2. Cliquez sur la page **Caractéristiques**.
3. Dépliez la section **Données métier**.

SYNCHRONISER LES MODÈLES LOGIQUES ET PHYSIQUES



La synchronisation est le traitement qui permet de traduire un diagramme de classes exprimé avec le formalisme classes/parties en un modèle physique exprimé avec le formalisme relationnel, et inversement. Elle met ainsi en correspondance les objets des deux modèles.



Une option de compatibilité vous permet également de mettre en correspondance un modèle de données (entités/associations) et un modèle physique. Voir [Formalisme logique et synchronisation](#).

Ce processus est à réaliser à certaines périodes. En effet, tout au long du projet de modélisation, ces modèles subissent des changements chacun de leur côté. La synchronisation intervient lorsque vous voulez confronter les deux modèles et rétablir de façon automatique les correspondances canoniques qui les unissent.

La fonctionnalité de synchronisation est disponible avec l'accès au référentiel **Hopex** en mode "Avancé".



La synchronisation de modèles d'une base de données peut s'effectuer dans un sens ou dans l'autre - soit dans le sens physique > logique soit dans le sens logique > physique - mais pas les deux à la fois. Une fois le sens de la synchronisation déterminé, il convient de ne pas inverser la synchronisation. La justification des correspondances entre les niveaux logique et physique n'est pas garantie si cette règle n'est pas respectée.

Voir aussi [Correspondance des modèles](#).

- ✓ Règles de synchronisation "logique vers physique"
- ✓ Du modèle logique au modèle physique
- ✓ Synchronisation réduite (mode logique vers physique)
- ✓ Relancer la synchronisation après modifications
- ✓ Du modèle physique au modèle logique
- ✓ Paramétrer la synchronisation
- ✓ Synchronisation des diagrammes

Options d'affichage de la synchronisation

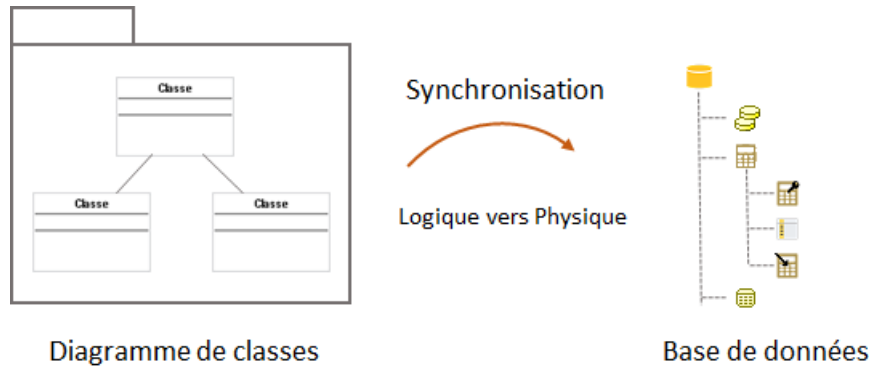
Certaines options de synchronisation sont filtrées par défaut. Pour les afficher :

1. Cliquez sur **Menu principal** > **Paramètres** > **Options**.
2. Dans l'arborescence **Options** du volet de gauche, déployez le dossier **HOPEX Solutions** > **Data Management**.
3. Cliquez sur le sous-dossier **Synchronisation de base de données**.
4. Dans le volet de droite, activez les options de synchronisation voulues :
 - synchronisation logique > physique
 - synchronisation physique > logique
 - synchronisation réduite logique > physique
 - synchronisation réduite physique > logique

Vous pouvez voir que le formalisme logique appliqué par défaut dans la synchronisation est la notation UML. Voir [Les formalismes](#).

RÈGLES DE SYNCHRONISATION "LOGIQUE VERS PHYSIQUE"

Les règles qui suivent sont utilisées pour la transformation des diagrammes de classes et modèles de données en formalisme relationnel.



☛ Voir également [Paramétrer la génération des noms](#) et [Types des données et Datatypes des colonnes](#).

Synchronisation logique > physique : les Entités (ou Classes)

En mode logique > physique, les classes et les entités sont traitées de la même façon dans l'outil de synchronisation.

🔗 **Par défaut la synchronisation applique le formalisme logique du diagramme de classes. Voir [Formalisme logique et synchronisation](#).**

Règle générale

- Toute entité non abstraite du modèle devient une table.
- L'identifiant de l'entité devient la clé primaire de la table. Si l'identifiant est implicite, une colonne est automatiquement créée. Voir [Paramétrer la génération des noms](#).
- Les attributs de l'entité deviennent des colonnes de la table.
- Des règles de correspondance permettent de déduire les datatypes des colonnes à partir du type de données (MD) de chaque attribut. Les paramètres proposés dépendent du SGBD.

☛ Pour plus d'informations à ce sujet, voir [Types des données et Datatypes des colonnes](#).

Sous-entité

- La clé étrangère issue de la dépendance entre la sous-entité et sa super-entité est créée.

Entité abstraite

Une entité abstraite ne donne lieu à aucune table lors de la synchronisation.

Si des associations contraintes pointent vers une entité abstraite, les clés étrangères correspondantes ne sont pas créées, mais les colonnes correspondant aux clés étrangères sont créées pour respecter l'intégrité de la table.

Lorsqu'une sous-entité est abstraite, toutes les colonnes et clés étrangères de la table correspondante sont prises en charge par la table correspondant à la super-entité.

Inversement, lorsqu'une super-entité est abstraite, toutes les colonnes et clés étrangères de la table correspondante sont prises en charge par la table correspondant à la sous-entité.

Pour définir une entité abstraite :

1. Ouvrez la fenêtre de propriétés de l'entité.
2. Cliquez sur l'onglet **Caractéristiques**.
3. Dans le champ **Abstraite**, sélectionnez "Oui".

Entité réalisée

On dit d'une entité qu'elle est réalisée si elle donne lieu à la création d'une table lors d'une synchronisation.

Une entité "non réalisée" est traitée comme une entité abstraite.

Contrairement à la propriété "abstraite" qui caractérise l'entité dans tous ses cas d'emploi, le concept "Réalisé" s'applique uniquement dans le cadre de la synchronisation d'une base de données. Voir [Mode Réalisé](#).

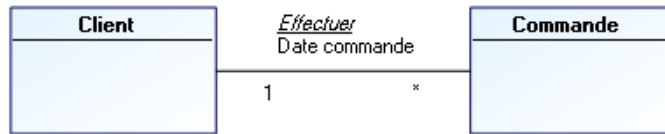
Synchronisation logique > physique : les Associations

La synchronisation des associations est disponible avec l'ancien formalisme UML qui prend en compte les associations et non les parties. Voir [Formalisme logique et synchronisation](#).

Association contrainte (multiplicités : 0,1 ou 1,1)

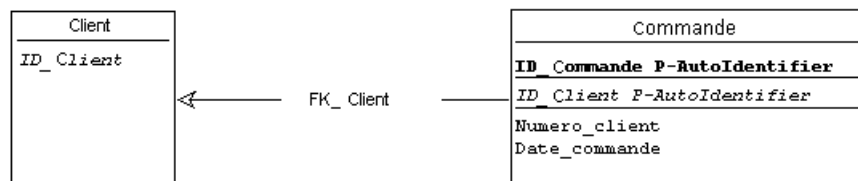
Une association contrainte est une association binaire dont la multiplicité maximum d'un de ses rôles est 1. Dans ce cas, il n'est pas nécessaire de créer une table correspondant à cette association. Une colonne rajoutée dans la table correspondant à l'entité suffit.

Une association contrainte (une de ses multiplicités maximum est à 1) ne donne pas lieu à une table. Dans l'exemple suivant, une commande comporte un client et un seul.



La synchronisation de ce diagramme de données donne l'un des deux résultats suivants :

L'association ne donne pas lieu à une table



- Une colonne correspondant à la clé de l'entité "Client" est créée dans la table "Commande".
- Une colonne est également créée pour chaque attribut de l'association.
- Une clé étrangère "FK_Client" est ajoutée pour assurer le contrôle de la colonne "ID_Client" de la table "Commande". Elle indique que les valeurs possibles pour la colonne "Numéro Client" de la table "Commande" sont celles qui existent dans la colonne "Numéro Client" de la table "Client".

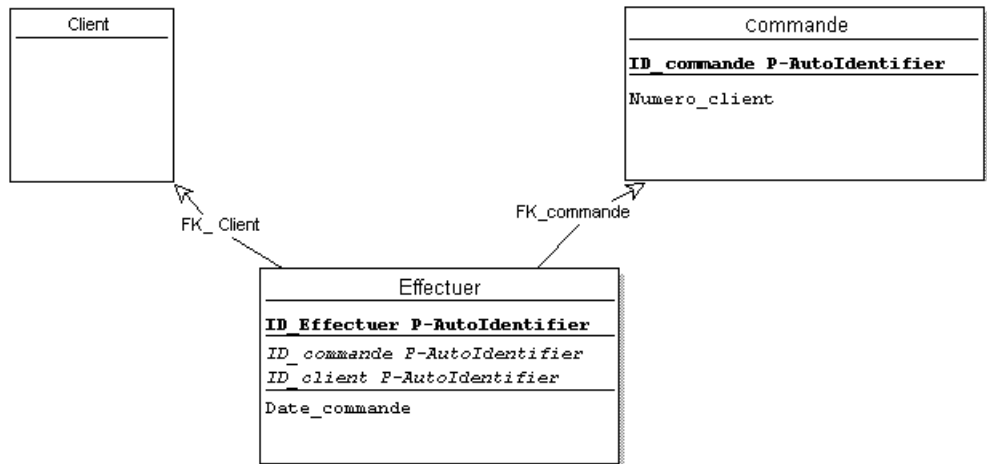
La clé étrangère ("foreign key") est créée à partir de l'identifiant de l'entité.

L'association est transformée en table

Pour que l'association se transforme en table :

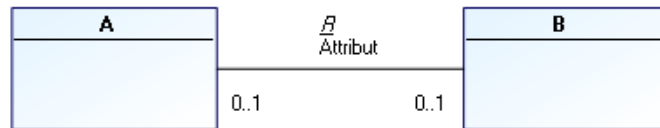
1. Ouvrez la fenêtre de propriétés de l'association.
2. Cliquez sur l'onglet **Caractéristiques**.

3. Dans le champ **Correspondance potentielle**, sélectionnez la valeur "Table".



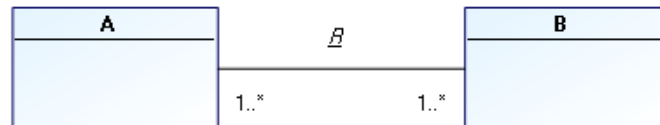
Association contrainte (multiplicités : 0,1 et 0,1)

Dans ce cas particulier, la combinaison des multiplicités est ambiguë. Il n'y a pas d'élément permettant de choisir dans quelle table on va créer la colonne correspondant à l'attribut.



La synchronisation propose une colonne dans chaque table.

Verrous croisés (dead locks)



Les multiplicités 1..x, 1..x signifient que chacun des deux objets doit être relié à au moins un objet de l'autre type pour pouvoir exister.

Ceci pose un problème pour créer les premiers objets de chaque type. En effet :

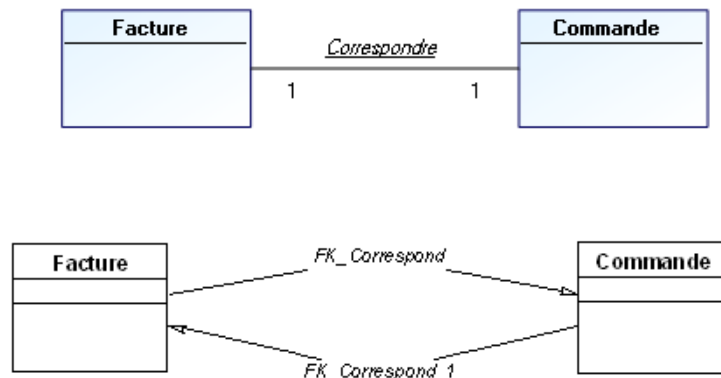
- Un objet de type A doit exister pour pouvoir créer un objet de type B et le lui relier.
- Réciproquement, un objet de type B doit exister pour pouvoir créer un objet de type A et le lui relier.

C'est le cas en particulier pour les couples de multiplicités suivants :

- Multiplicités 1..*, 1..*
- Multiplicités 1, 1..*

Ces cas n'amènent cependant pas d'impossibilité physique car le problème se cantonne à la création des premiers objets de chaque type. De plus, aucune clé étrangère pour effectuer le contrôle d'intégrité n'est générée dans le premier cas, et une seule dans le deuxième cas, ce qui n'amène pas à une situation de verrou croisé ("dead lock").

Les multiplicités 1, 1 génèrent plusieurs clés étrangères obligatoires croisées :



On est alors dans une situation de blocage total, car on devrait, pour respecter les contraintes, créer plusieurs tables à la fois.

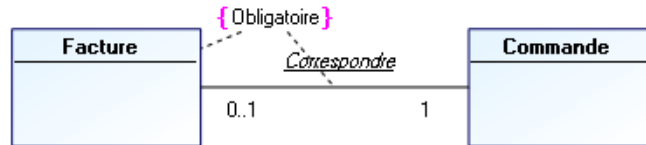
Certains SGBD interdisent complètement la création de tables de ce type.

Pour permettre une synchronisation correcte, il vous faut éviter les situations de ce genre.

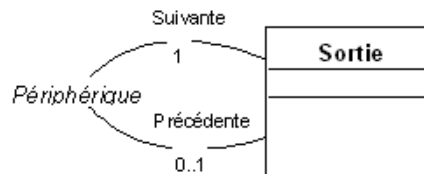
☛ Choisissez une des clés étrangères pour laquelle vous mettez la multiplicité 1 et sur l'autre, mettez 0..1. Vous pouvez également retirer la clé étrangère de la table après la synchronisation, mais c'est moins pratique.

Pour ne pas perdre l'information concernant la multiplicité minimum à 1, vous pouvez ajouter une contrainte comme celle présentée dans le diagramme qui suit. Cette contrainte ne sera pas prise en compte dans la synchronisation.

☛ Voir [Les contraintes](#) pour plus de détails.

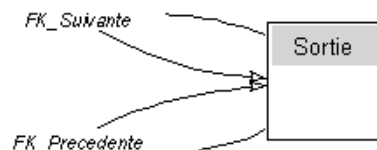


Voici un autre exemple dans le cas d'une nomenclature : Nomenclature 1, 1 ou 0..1, 1.



On veut exprimer que chaque sortie du boulevard périphérique précède une sortie et en suit une autre.

Cette modélisation oblige à créer toutes les sorties à la fois, puisque pour chaque sortie, on doit avoir créé la précédente.



Pour éviter cela, il faut mettre les deux multiplicités à 0..1 et 0..1.

Association non-contrainte

Une association dont les multiplicités maximum sont différentes de 1 donne lieu à une table :

- Une colonne est créée pour chacun des attributs des identifiants des entités reliées.
- La clé primaire de la table porte sur l'ensemble de ces colonnes.
- Une clé étrangère est également constituée pour chaque entité reliée.
- Une colonne supplémentaire est créée pour chaque attribut de l'association.

Avant de lancer la synchronisation, il est souhaitable de contrôler la validité du

diagramme de données, et de contrôler que le paramétrage de la synchronisation est correct. Voir [Préparer la synchronisation](#).

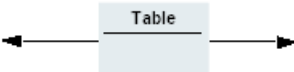
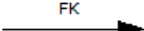
Classe associative

Les associations reliant des classes associatives ne sont pas prises en compte dans la synchronisation.

Synchronisation logique > physique : les Parties (Formalisme UML)

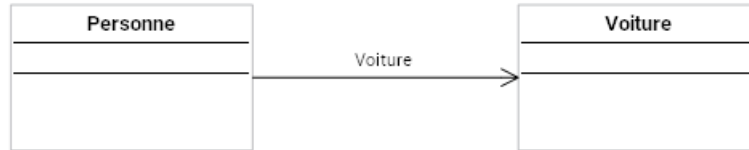
La synchronisation des parties est disponible par défaut avec le nouveau formalisme UML.

Le résultat de la synchronisation est fonction de la combinaison du lien **Tout/partie** (Non renseigné, Agrégation, Composition) et de la **Multiplicité** définis sur la partie.

Multiplicité	Lien Tout/Partie	
	Agrégation  Composition 	Non renseigné
Non renseigné (*) 2..6 1..*	La partie donne lieu à une clé étrangère vers la classe détentrice 	La partie donne lieu à une table entre les deux classes 
1 0..1	La partie donne lieu à une clé étrangère vers la classe référencée  et donne lieu à une clé étrangère vers la classe détentrice 	La partie donne lieu à une clé étrangère vers la classe référencée 

Exemple 1 : Non renseigné / *

Dans l'exemple suivant, la classe "Personne" référence la classe "Voiture", sans contrainte de multiplicité.

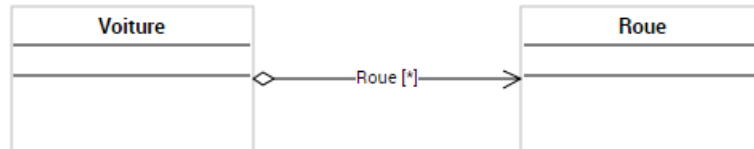


Après synchronisation, la partie "Voiture" donne lieu à une table :

- Une colonne est créée pour chacun des attributs des identifiants des entités reliées.
- La clé primaire de la table porte sur l'ensemble de ces colonnes.
- Une clé étrangère est également constituée pour chaque entité reliée.

Exemple 2 : Agrégation / *

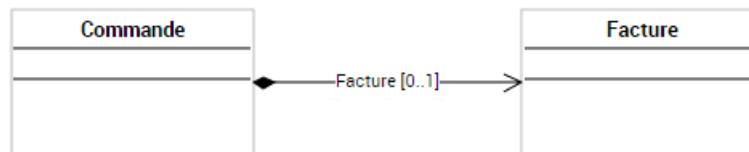
Une voiture peut avoir une ou plusieurs roues.



Après synchronisation, la partie "Roue" donne lieu à une clé étrangère vers la table "Voiture".

Exemple 3 : Composition / 0..1

Une commande détient une facture.



Après synchronisation :

- une clé étrangère référence la table "Facture" dans la table "Commande"
- une clé étrangère référence la table "Commande" dans la table "Facture".

DU MODÈLE LOGIQUE AU MODÈLE PHYSIQUE

Cette partie présente comment synchroniser le modèle logique d'une base de données (représenté par un diagramme de données) avec le modèle physique (relationnel) correspondant.

Bien que la synchronisation du modèle relationnel à partir du diagramme de données concerne essentiellement des entités, elle peut également se faire à partir des classes d'un diagramme de classes.

La synchronisation dans l'autre sens, d'un modèle physique vers un modèle logique est également possible mais pas sur une même base de données. Une fois le sens de la synchronisation choisie pour un objet, la synchronisation dans l'autre sens n'est plus possible.

Voir [Du modèle physique au modèle logique](#).

Les points qui suivent présentent la synchronisation d'une base de données. Vous pouvez également lancer une synchronisation réduite, autrement dit sur un objet particulier de la base de données. Voir [Synchronisation réduite \(mode logique vers physique\)](#).

Lancer la synchronisation

La synchronisation Logique vers Physique consiste à construire le modèle physique à partir du modèle logique, autrement dit à créer les tables et colonnes correspondant aux entités et attributs du diagramme de données.

L'outil de synchronisation est disponible dans le volet de navigation **Outils** > **Synchronisation de données**. Vous pouvez également ouvrir l'outil de synchronisation directement à partir de la base de données concernée.

Pour lancer une synchronisation Logique vers Physique sur une base de données :

1. Cliquez sur l'icône de la base de données et sélectionnez **Synchroniser**. L'assistant de synchronisation apparaît.
2. Sélectionnez le type de synchronisation "logique vers physique".
3. Cliquez sur **Suivant**.

Etape 1 : sélectionner les objets sources à synchroniser

Pour définir le périmètre de la synchronisation :

1. Dans l'arbre de la vue logique, déroulez la liste des objets contenus dans la base de données.

2. Par défaut, tous les objets sont cochés et donc inclus dans la synchronisation. Pour exclure un objet de la synchronisation, décochez-le dans la colonne **Périmètre**. Lorsqu'un objet est exclu, sa correspondance l'est aussi.

DataBase Selection

Définissez votre périmètre de synchronisation :

	Périmètre	Non réalisé	Nom	Type expression
Commandes	☒		Commandes	
Contrat d'achat	☒		Achat::Contrat d'ac...	
Ordre d'achat	☒		Achat::Ordre d'achat	
Diagramme de classes	☒		Achat::Ordre d'acha...	
Diagramme de données	☒		Achat::Ordre d'acha...	
Article Commandé	☒		Achat::Article Com...	
Contrat	☒		Achat::Contrat	
Employé	☒		Employé::Employé	
Fournisseur	☒		Achat::Fournisseur	
Ordre d'achat	☒		Achat::Ordre d'achat	
Produit	☒		Achat::Produit	
Site (EN)	☒		Achat::Site (EN)	
Fournisseur , Site fourni...	☒		Fournisseur , Site fo...	

Précédent Suivant OK Annuler

3. Par défaut, tous les objets sont "réalisés", autrement dit donnent lieu à la création d'un objet lors de la synchronisation. Pour indiquer qu'un objet est "non réalisé", cochez-le dans la colonne **Non réalisé**. Pour plus d'informations voir [Mode Réalisé](#).
4. Une fois la liste des objets définie, cliquez sur le lien **Suivant** de l'assistant.

Etape 2 : options de synchronisation

Parmi les options de synchronisation, vous pouvez :

- Réinitialiser les objets cibles : dans le cas où les objets de la vue logique ont déjà été synchronisés, la synchronisation repart de zéro et supprime les objets cibles existants.



Lorsque les modèles ont déjà été synchronisés, pour que les correspondances établies soient prises en compte lors d'une

**nouvelle synchronisation, veillez à ce que l'option
"Réinitialisation des objets cibles" soit décochée.**

- Recalculer le nom des objets cibles : les noms des objets physiques sont recalculés en fonction de ceux des objets sources. Cela signifie que toute modification manuelle du nom des objets physiques est annulée.
- Prendre en compte les optimisations : toutes les optimisations - dont celles qui ne sont pas cochées dans l'étape de validation (voir étape 4) - sont proposées.
- Prendre en compte les suppressions : les entités, associations et diagrammes qui ont été supprimés font partie du périmètre. En conséquence, la suppression des objets ou des liens cibles correspondants est proposée.
Voir [Cas d'emploi des options](#)

Les autres options concernent la mise à jour des propriétés des objets cibles. Par défaut, la synchronisation met à jour l'ensemble des propriétés de chaque objet concerné.

Scheduling

Vous pouvez exécuter la synchronisation :

- Immédiatement
- dès que possible (après publication des mises à jour)
- A une date et heure prédéfinies

】 Une fois les options définies, cliquez sur **Suivant**.

Etape 3 : protéger des objets

La synchronisation peut avoir un impact sur tous les objets d'une base de données existante.

- 】 Pour garder un objet intact, cochez-le dans la colonne **Figé**.
- 】 Cliquez sur **Suivant** pour passer à la suite.

Voir [Protéger des objets](#).

Etape 4 : valider les résultats

L'assistant affiche les résultats qu'entraînera la validation de la synchronisation.

Les objets devant subir une modification automatique sont signalés d'une coche.

Des icônes précédant le nom des objets indiquent les actions qui vont être effectuées sur ces objets. Il peut s'agir d'une création , d'une suppression  ou d'une mise à jour 

Une flèche  devant un objet indique que la synchronisation a un impact sur les sous-objets de l'objet en question.

- 】 Dépliez l'objet pour visualiser les modifications concernées.

Validation des optimisations

Les optimisations sont des personnalisations sur des objets ainsi soustraits au traitement automatique de la synchronisation.

Exemples d'optimisation :

Une coche désigne les objets qui vont être modifiés. Si vous ne souhaitez pas valider les modifications portant sur certains objets, vous devez décocher les cases correspondantes. Cette optimisation est conservée lors des prochaines synchronisations.

Par ailleurs, **Hopex** déduit des optimisations suite aux actions que vous avez pu effectuer manuellement. Si vous avez ajouté une table dans la vue physique sans avoir créé d'objet correspondant dans la vue logique, la synchronisation ne coche

pas la suppression de cette table :   Table1 

Pour que l'objet soit supprimé, vous devez cocher la case correspondante.

- Une fois les actions sur les objets cibles définies, vous pouvez cliquer sur **Suivant**.

Un compte-rendu vous indique les actions effectuées.

Vous pouvez fermer l'assistant et visualiser les résultats dans l'éditeur.

Cas d'emploi des options

La combinaison des options "Prendre en compte les optimisations" et "Prendre en compte les suppressions" varie en fonction du périmètre des objets que vous souhaitez mettre à jour.

Prendre en compte les optimisations

Lorsque cette option est cochée, la synchronisation propose toutes les créations, suppressions et modifications, y compris les optimisations non cochées par défaut à l'étape de validation.

Lorsque l'option est décochée, seules les modifications cochées par défaut sont proposées à l'étape de validation.

Cette option permet de filtrer le résultat d'une synchronisation pour présenter uniquement les modifications qui ont un impact réel sur les données cibles.

Prendre en compte les suppressions

Lorsque cette option est cochée, la synchronisation prend en compte les entités, associations et diagrammes qui ont été supprimés. En conséquence, la suppression des objets ou liens cibles correspondants est proposée.

Lorsque l'option est décochée, les entités, associations et diagrammes qui ont été supprimés ne sont pas pris en compte. En conséquence, les objets cibles correspondants ne sont pas modifiés.

☛ **Cette option ne s'applique qu'aux entités, associations et diagrammes. Pour les autres types d'objets supprimés (attributs, identifiants, etc.), l'impact sur les objets ou liens cibles est conditionné non pas par cette option mais par l'objet qui les contient.**

Cette option permet de limiter l'impact d'une synchronisation au strict périmètre source défini par l'utilisateur, en excluant tout objet qui n'est pas explicitement déclaré dans le périmètre. Cette option peut être associée au périmètre de synchronisation pour des cas d'emplois types.

Combinaisons des options possibles

1. Option "Prendre en compte les suppressions" cochée et périmètre de synchronisation complet

Il s'agit d'un cas d'emploi qui privilégie une synchronisation complète entre la source et la cible. Dans ce cas, tous les objets ont une correspondance valide à l'issue de chaque lancement de l'assistant de synchronisation. Ce mode est à utiliser lorsque la source et la cible doivent être mis en cohérence en totalité.

2. Option "Prendre en compte les suppressions" cochée et périmètre de synchronisation partiel

Ce cas d'emploi permet de travailler sur le périmètre choisi, tout en incluant l'impact des objets supprimés. Ce mode permet notamment de s'assurer des suppressions d'objets cibles suite à la suppression d'objets sources de type entité, association ou diagramme : ces objets sources ayant été supprimés, il n'est théoriquement pas possible de les inclure dans le périmètre de synchronisation. Cocher cette option rend ce choix possible. Il s'agit du mode à privilégier lorsque le périmètre est large et que le peu d'objets exclus du périmètre le sont à titre provisoire (ce sont par exemple de nouveaux objets dont on veut différer l'impact sur la cible).

3. Option "Prendre en compte les suppressions" cochée et périmètre de synchronisation vide

Il s'agit d'un mode particulier permettant de "nettoyer" les objets cibles dont la correspondance n'est plus valide, sans autre impact.

4. Option "Prendre en compte les suppressions" non cochée et périmètre de synchronisation partiel

Ce cas d'emploi permet de travailler strictement sur le périmètre choisi, en excluant tout impact en dehors de ce périmètre. Ce mode permet d'éviter des suppressions d'objets cibles suite à la suppression d'objets sources de type entité, association ou diagramme : ces objets sources ayant été supprimés, il n'est théoriquement pas possible de les exclure du périmètre de synchronisation. L'option décochée rend ce choix possible. Il s'agit du mode à privilégier lors d'une synchronisation ponctuelle sur un périmètre restreint, qui ne prend pas en compte la cohérence totale de la source et de la cible. C'est par ailleurs le mode le plus rapide.

5. Option "Prendre en compte les suppressions" non cochée et périmètre de synchronisation complet

Cette combinaison présente un cas d'emploi en principe peu fréquent. Elle correspond à un mode de travail dans lequel la suppression d'objets sources est systématiquement sans effet sur la cible; autrement dit, tout objet cible créé n'est plus supprimé tant que ce mode est activé.

6. Option "Prendre en compte les suppressions" non cochée et périmètre de synchronisation vide

Cette combinaison est sans effet.

Protéger des objets

Vous pouvez protéger un objet afin que la synchronisation n'ait pas d'impact sur lui. Cela exclut l'objet de la synchronisation sans le faire disparaître.

Il existe deux modes de protection des projets ; l'un en amont de la synchronisation et l'autre en aval.

Mode Figé

Le mode "Figé" concerne l'objet cible, celui qui résulte de la synchronisation.

Lorsque vous figez une table du modèle relationnel, vous figez également tous les objets fils de cette table : aucun objet fils n'est créé, modifié ou supprimé suite à la synchronisation.

Vous pouvez figer des objets :

- Avant de lancer la synchronisation, dans l'éditeur de base de données.
- Lors du lancement de la synchronisation, dans les options présentées dans l'assistant. Voir [Etape 3 : protéger des objets](#).

Mode Réalisé

Le mode "Réalisé" concerne l'objet source de la synchronisation.

Un objet est dit réalisé s'il donne lieu à la création d'un objet lors d'une synchronisation.

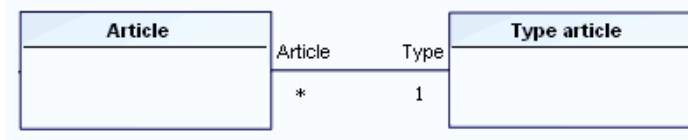
Un objet non réalisé ne donne pas lieu à la création d'un objet lors de la synchronisation mais il est traité comme un objet abstrait. Voir [Entité abstraite](#).

Par défaut, tous les objets sont réalisés.

Vous pouvez exclure un objet source de la synchronisation en cochant la colonne "Non réalisé" sur l'objet en question dans l'assistant de synchronisation. Cette action est disponible sur les entités, les attributs et les associations. L'action "Réalisé" ou "Non Réalisé" se propage aux objets fils.

Exemple d'entité non réalisée

L'entité "Article" possède une association vers l'entité "Type article".



L'entité "Type article" est dite "Non réalisée".

Lors de la synchronisation, l'entité "Type article" ne donne lieu à la création d'aucune table. Dans la table "Article", la clé étrangère vers "Type article" n'est pas créée; en revanche, la colonne "Code_Type_Article" est créée.

Résultats de la synchronisation : les correspondances

Quand la synchronisation est terminée, les tables, colonnes, clés et index du diagramme physique ont été mis en phase avec le diagramme de données. Ils peuvent être consultés et modifiés pour prendre en compte les optimisations éventuelles.

Caractéristiques de la correspondance

Pour plus de détails sur la correspondance :

- 1 Dans l'éditeur de correspondances, sélectionnez l'élément de correspondance et cliquez sur **Propriétés**.
- 2 Dans la fenêtre qui apparaît, cliquez sur la liste déroulante puis sur



Caractéristiques.

Périmètre de la synchronisation

Par défaut, tous les objets faisant partie des modèles synchronisés sont inclus dans la synchronisation. Vous avez cependant la possibilité d'exclure un objet de la synchronisation.

Voir [Etape 1 : sélectionner les objets sources à synchroniser](#).

Etat de la synchronisation

Vous pouvez protéger un objet afin qu'il ne subisse aucune modification lors de la synchronisation, vous préciserez alors qu'il est "Figé".

Voir aussi [Protéger des objets](#).

Sens de la synchronisation

Le sens de la synchronisation sur une correspondance permet de savoir quel objet est mis à jour par rapport à l'autre.

Dans certains cas, la synchronisation est possible dans les deux sens (par exemple, lorsque deux objets pouvant être synchronisés n'ont pas encore de correspondance). Dans d'autres cas, elle n'est possible que dans un sens (par exemple, si l'un des deux objets est déjà synchronisé) ou bien impossible dans les deux (parce que chaque objet a déjà une correspondance ou parce que les types des objets concernés ne peuvent faire l'objet d'une synchronisation). Cette indication est fournie par le champ **Synchronisation**.

Pour résumer :

- **Bidirectionnelle** : la synchronisation se fait dans les deux sens.
- **De gauche à droite** : la synchronisation se fait de gauche à droite (l'objet de droite est synchronisé par rapport à l'objet de gauche).
- **De droite à gauche** : la synchronisation se fait de droite à gauche.
- **Jamais** : pas de correspondance possible entre les deux objets.

Pour plus d'informations sur les correspondances, voir [L'éditeur de base de données](#).

SYNCHRONISATION RÉDUITE (MODE LOGIQUE VERS PHYSIQUE)

La fonctionnalité de synchronisation permet de synchroniser un modèle logique et un modèle physique d'une base de données. En phase de conception, il est souvent utile de synchroniser une partie du modèle courant sans avoir besoin de prendre en charge la totalité de la base de données qui peut être très volumineuse. **Hopex Data Architecture** permet de limiter le périmètre d'une synchronisation à un ensemble d'objets, réduisant ainsi le temps de traitement de la synchronisation.

Les points qui suivent détaillent le mode "Synchronisation réduite" dans le sens Logique > Physique mais il est également disponible dans le sens Physique > Logique.

Objets sources de la synchronisation réduite

La synchronisation réduite est une synchronisation appliquée à un objet autre que la base de données. La synchronisation réduite s'applique uniquement à un objet dont la base de données a déjà été synchronisée.

Les objets sur lesquels vous pouvez lancer une synchronisation réduite sont :

- Classe
- Association
- Paquetage
- Table
- Dossier de Table
- Entité (DM)
- Association (DM)
- Modèle de données

Le périmètre de la synchronisation réduite est déterminé par l'objet sur lequel vous lancez la synchronisation réduite.

Les cas suivants illustrent des cas de synchronisation réduite dans le sens logique vers physique.

Lancement sur un modèle de données

Lorsque vous lancez une synchronisation sur un modèle de données, tous les objets du modèle sont sélectionnés par défaut dans le périmètre de la synchronisation ; ils sont tous cochés dans l'éditeur.

Lancement sur une entité du modèle de données

Lorsque vous lancez une synchronisation à partir d'une entité (ou un autre objet) appartenant à un modèle de données, seuls les objets en lien avec cette entité au sein du même modèle sont sélectionnés par défaut.

Les objets en lien avec l'entité mais qui appartiennent à un autre modèle sont affichés dans l'éditeur (dès lors qu'ils sont reliés à la base de données cible) mais non sélectionnés par défaut. Vous devez cocher les cases associées pour les prendre en compte dans la synchronisation.

Le modèle de données de l'entité synchronisée est pris comme périmètre uniquement lorsque le lien de détention entre le modèle de données et l'entité est clairement identifié : c'est le cas lorsque vous sélectionnez l'entité dans la fenêtre de navigation, ce n'est pas le cas lorsque vous la sélectionnez dans un diagramme.

Lancement sur une entité hors contexte

Lorsque vous lancez la synchronisation sur une entité hors contexte, par exemple dans une fenêtre d'explorateur, tous les objets qui dépendent de l'entité modifiée, compris ou non dans des modèles différents (à condition que ces modèles soient reliés à la base de données cible) sont cochés par défaut dans le périmètre de la synchronisation, car aucun contexte de modèle en particulier n'est identifié.

Les stratégies de synchronisation réduite

Lors de la synchronisation à partir d'un objet, il est possible d'appliquer une des trois stratégies ci-dessous.

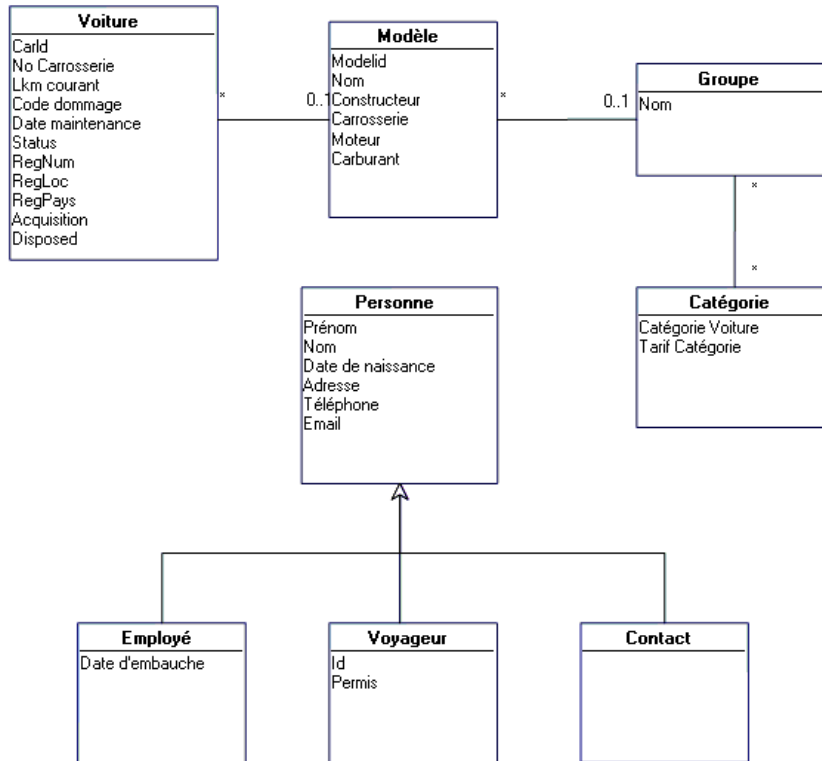
Impact de l'objet synchronisé sur les autres objets

Cette stratégie permet de définir le périmètre de la synchronisation depuis l'objet source et de l'étendre à tous les objets qui dépendent de lui et qui sont susceptibles d'être influencés par sa modification.

Exemple

Avec cette stratégie, dans le modèle ci-dessous, la synchronisation réduite de l'entité "Modèle" permet d'inclure également l'entité "Voiture" qui a une association contrainte vers l'entité "Modèle".

Modèle logique



Résultats de la synchronisation réduite

Indiquer les objets à rejeter :

	Nom	Type expression		Périmètre
	Location voiture	Location voiture		
	Location véhicule	Location véhicule		
	Modèle	Modèle de véhicule:...		
	Voiture	Modèle de véhicule:...		
	Modèle de véhicule	Modèle de véhicule		

Impact des autres objets sur l'objet synchronisé

Cette stratégie permet d'intégrer dans le périmètre de la synchronisation réduite les objets dont dépend directement l'objet source, par exemple, tous les objets

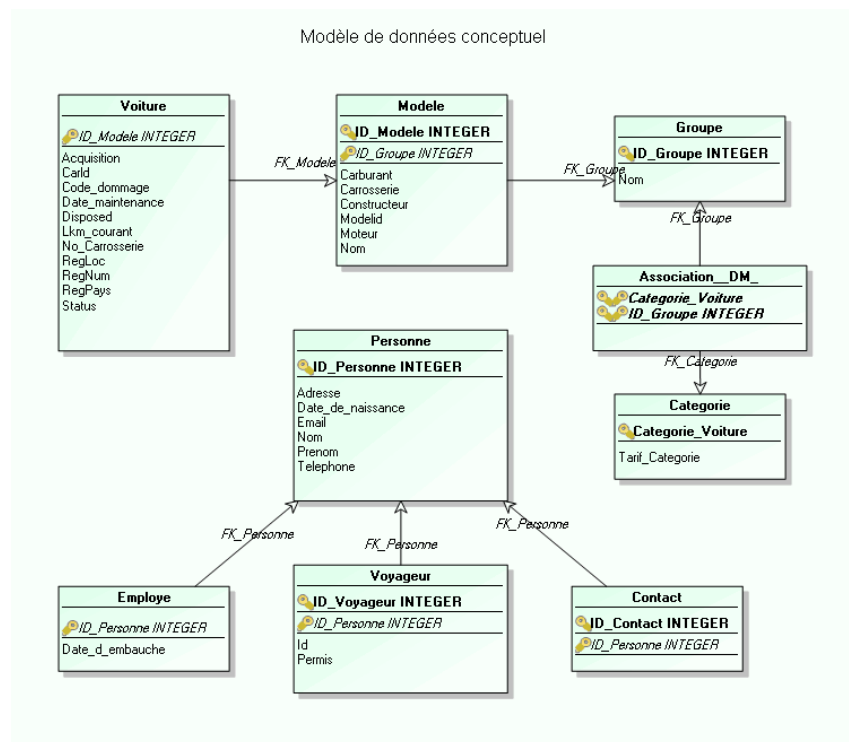
associés à l'entité source et qui sont nécessaires à la mise à jour de la table correspondante.

Exemple

Dans le même exemple que précédemment, en prenant l'entité "Modèle" comme source d'une synchronisation réduite, le périmètre s'étend à l'entité "Groupe" car les tables correspondant à ces deux entités sont liées par une clé étrangère : l'entité "Groupe" peut modifier la table "modèle" associée à l'entité "Modèle" par l'intermédiaire de la clé étrangère "FK_Group" (voir schéma ci-dessous).

L'entité "Voiture" quant à elle n'est pas prise en compte dans le périmètre car elle ne peut pas agir sur la table "Modèle".

Modèle physique



Résultats de la synchronisation réduite

Indiquer les objets à rejeter :

	Nom		Périmètre	Nom
	Location voiture			Location voiture
	Location véhicule			Groupe
	Groupe			Modèle
	Modèle			
	Modèle de véhicule			

Tous les impacts

Cette stratégie permet de combiner les deux stratégies citées ci-dessus. Ainsi le périmètre de la synchronisation réduite est étendu aux objets dont l'objet source a besoin et à tous les objets susceptibles d'être influencés.

Exemple

Résultat de la synchronisation réduite

Indiquer les objets à rejeter :

	Nom	Type exp		Périmètre	Nom
	Location voiture				Location voiture
	Location véhicule				Groupe
	Groupe				Modèle
	Modèle				Voiture
	Voiture				
	Modèle de véhicule				

Lancer la synchronisation réduite

Avant de commencer, vérifiez que l'option «Synchronisation réduite (logique -> physique) » est active :

1. Dans le bureau, cliquez sur le menu **Menu principal** > **Paramètres** > **Options**.
La fenêtre des options apparaît.
2. Dans la partie gauche de la fenêtre, déployez le dossier **HOPEX Solutions** > **Data Management**.
3. Cliquez sur **Synchronisation de base de données**.

4. Dans la partie droite de la fenêtre, cochez la case **Activer la synchronisation réduite (logique > physique)**.

Pour lancer la synchronisation réduite :

1. Faites un clic droit sur l'objet à synchroniser.
2. Sélectionnez **Synchroniser**.
L'assistant de synchronisation apparaît.
3. Sélectionnez le type de synchronisation "logique vers physique".
4. Cliquez sur **Suivant**.
Une entité peut être utilisée par plusieurs modèles de données, donc par plusieurs bases de données. Quand c'est le cas, vous devez sélectionner la base concernée.
5. Sélectionnez la **Stratégie**.
6. Cliquez sur **Suivant**.
7. Sélectionnez les objets à synchroniser.
Le périmètre coché par défaut est fonction du contexte dans lequel vous sélectionnez l'objet à synchroniser : si la synchronisation réduite est initialisée à partir d'une entité dans un diagramme, le modèle du diagramme en question est sélectionné. Si l'entité est sélectionnée hors contexte, l'ensemble des modèles dans lesquels elle apparaît s'affiche dans l'éditeur.
 *Les objets sélectionnés ne sont pas mémorisés ; lors d'une nouvelle synchronisation, le périmètre par défaut s'affiche à nouveau.*
8. Cliquez sur **Suivant**.
9. Définissez les **Options de synchronisation**. Toutes les options de la synchronisation standard sont disponibles à l'exception de l'option "Réinitialiser les objets cibles".
10. Cliquez sur **Suivant**.
L'option de protection des objets cibles s'affiche ; vous pouvez visualiser les objets figés. Il n'est pas possible de modifier la protection des objets.
11. Validez les résultats en cliquant sur **Suivant**.
Le rapport de la synchronisation apparaît.
12. Cliquez sur **OK** pour fermer l'assistant de synchronisation.
L'éditeur de correspondances apparaît (sauf si l'option correspondante a été décochée de l'assistant de synchronisation).

Options de la synchronisation réduite

La synchronisation réduite présente les mêmes options que la synchronisation totale à l'exception de :

- Réinitialisation les objets cibles
- Ordre

En effet, le périmètre réduit de la synchronisation réduite ne permet pas de rendre un résultat valide pour ces deux options.

RELANCER LA SYNCHRONISATION APRÈS MODIFICATIONS

Lorsqu'une base de données a été synchronisée puis modifiée manuellement, les spécifications complémentaires effectuées directement dans la base de données sont conservées, sauf si :

- Une réinitialisation est demandée.
- Des changements intervenus dans les diagrammes de données s'y opposent (ajout ou suppression d'objets ou de liens).

Parmi ces changements peuvent être cités :

- La création d'entités, d'associations, d'attributs dans le diagramme de données
- La suppression d'entités, d'associations, d'attributs dans le diagramme de données
- La modification des caractéristiques d'un attribut
- La modification du nom d'un attribut, d'une entité, ou d'une association
- La modification de la multiplicité maximum d'une association
- La modification des liens d'une association

Les spécifications complémentaires dans le diagramme relationnel peuvent être :

- Des suppressions d'objets créés par la synchronisation
- Des créations d'objets
- Des modifications des caractéristiques d'objets créés par la synchronisation
- Des modifications de l'ordre des colonnes dans les tables, les clés, ou les index.

Synchronisation après modification du diagramme de données

Création d'entités, d'associations et d'attributs dans le diagramme de données

Les éléments correspondants sont créés dans le diagramme relationnel, en suivant les règles utilisées lors de la première synchronisation.

Suppression d'entités, d'associations, d'attributs dans le diagramme de données

Les éléments correspondants sont supprimés dans la base de données. Par exemple, lorsqu'un attribut est supprimé dans le diagramme de données, la ou les colonnes correspondantes sont également supprimées.

Modification des caractéristiques d'un attribut

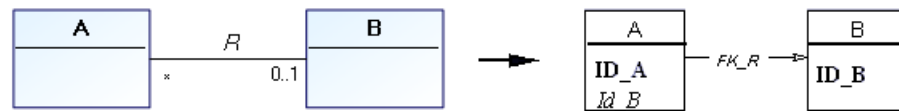
Les modifications des caractéristiques d'un attribut (type, longueur, décimales,...) sont reportées sur la colonne correspondante du diagramme relationnel.

Si la valeur d'une caractéristique de la colonne a été modifiée directement dans le diagramme relationnel, cette valeur est conservée.

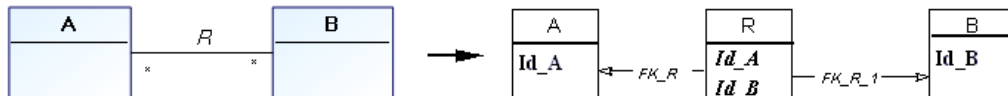
Modification du nom d'un attribut, d'une entité, ou d'une association

La modification du nom d'un attribut, d'une entité, ou d'une association n'est pas reportée sur l'objet correspondant du diagramme relationnel.

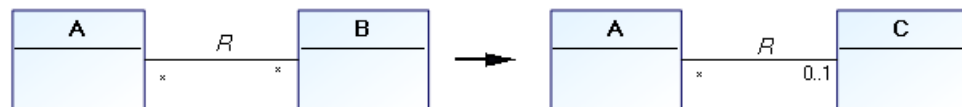
Modification de la multiplicité maximum d'une association



Si la multiplicité maximum d'une association qui était à 1, entraînant ainsi la création d'une colonne migrante, passe à N, la colonne migrante est supprimée, et la table justifiée par la multiplicité maximum à N est créée.



Modification des liens d'une association



L'association R ne porte plus sur l'entité B, mais sur l'entité C.

Dans ce cas, la colonne migrante de B dans A n'est plus justifiée.

- Elle est supprimée.
- Une colonne migrante provenant de C est créée.

Synchronisation après des modifications du diagramme physique

Suppression d'une table ou d'une colonne

Les suppressions dans la base de données d'objets qui ont été créés par la synchronisation (table, colonne, clé, index) sont mémorisées et conservées.

Tant que l'entité, l'association ou l'attribut qui justifie la table ou la colonne existe, la table ou la colonne n'est pas recréée.

Pour rétablir une colonne créée par la synchronisation puis supprimée :

1. Lancez la synchronisation de la base de données
2. Lors de l'étape de validation des résultats, confirmez l'action de création (cochez la case correspondante) proposée pour cette colonne.

Créations d'objets

Les créations d'objets (table, colonne, clé, index) effectuées dans le diagramme relationnel sont conservées.

Cependant, la suppression d'objets dont ils dépendent peut entraîner leur suppression.

Par exemple, on crée une colonne dans une table justifiée par une entité. Si on paramètre l'entité comme "non table" ou qu'on délie l'entité du modèle de données, la table correspondante va disparaître, et avec elle la colonne.

☛ *Les objets créés dans le diagramme relationnel peuvent être mis en correspondance manuellement. Les objets créés lors de la synchronisation sont mis en correspondance automatiquement.*

Modifications de caractéristiques d'objets créés par la synchronisation

Les modifications de caractéristiques (nom SQL, longueur, not null, datatypes) d'objets créés par la synchronisation sont conservées.

Modifications d'ordre

Concernant les modifications d'ordre, le traitement dépend des options définies dans l'assistant de synchronisation (voir [Etape 2 : options de synchronisation](#)).

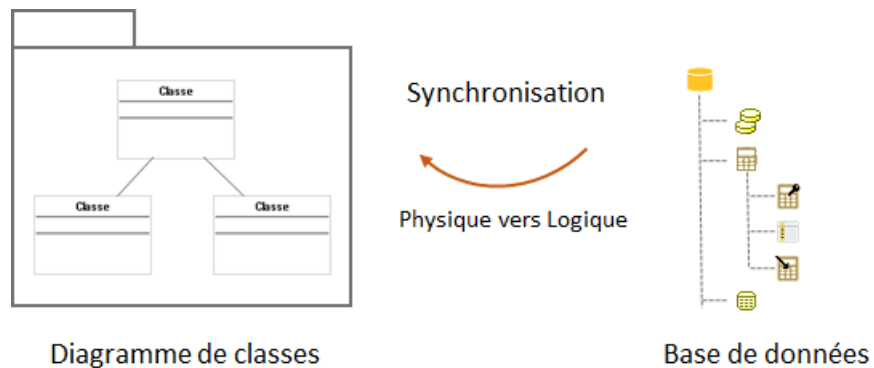
DU MODÈLE PHYSIQUE AU MODÈLE LOGIQUE

Cette partie présente comment synchroniser le modèle physique d'une base de données avec le modèle logique correspondant.

Règles de synchronisation "Physique vers Logique"

Synchroniser le modèle logique à partir du modèle physique permet de créer le diagramme de données d'une base à partir de ses tables.

Les règles utilisées pour cette transformation sont :



- Une table dont la clé primaire est constituée d'une clé étrangère qui porte sur les mêmes colonnes devient une entité. Une généralisation est créée entre cette entité et l'entité correspondant à la table vers laquelle pointe la clé étrangère.

Exemple niveau physique :

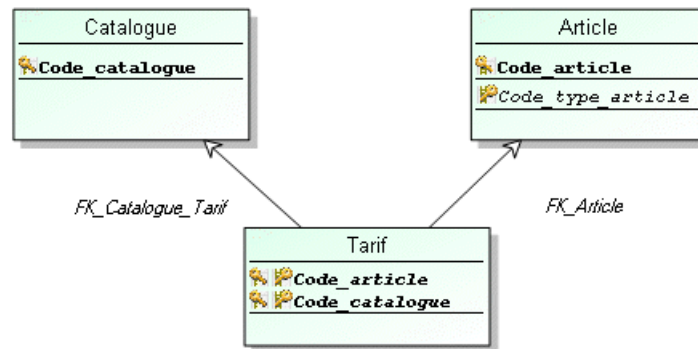


Résultat au niveau logique :

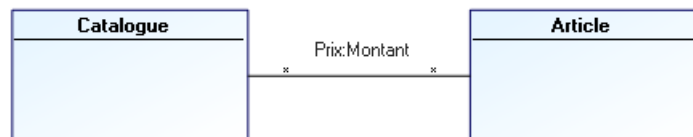


- Une table dont la clé primaire est constituée uniquement de clés étrangères devient une association de multiplicités (*..*). S'il existe des colonnes de la table qui ne font pas partie de la clé primaire, chacune de ces colonnes donne lieu à la création d'un attribut qui est rattaché à l'association.

Exemple niveau physique :



Résultat au niveau logique :



- Une table dont la clé primaire contient des clés étrangères et au moins une colonne qui n'est pas clé étrangère devient une entité. Une association agrégée est créée entre cette entité et l'entité correspondant à la table vers laquelle pointe chacune des clés étrangères.

La clé candidate de l'entité est composée des rôles des associations agrégées et des attributs correspondant aux autres colonnes de la clé primaire de la table.

Exemple niveau physique :

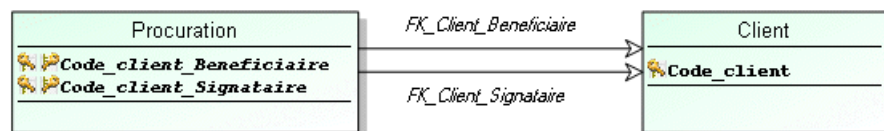


Résultat niveau logique :

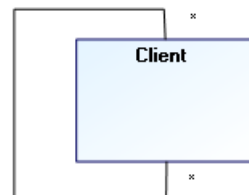


- Une table dont la clé primaire est constituée uniquement de deux clés étrangères pointant sur une même table devient une association réflexive de multiplicités (*..*).

Exemple niveau physique :



Résultat niveau logique :



- Dans les autres cas, chaque table devient une entité et ses colonnes les attributs de l'entité.

- Une clé étrangère devient une association (0..1, *). Si toutes les colonnes de la clé sont obligatoires, ses cardinalités deviennent (1, *).

- Les types des attributs sont recalculés à l'aide du tableau de conversion spécifique au SGBD cible (voir [Types des données et Datatypes des colonnes](#)).

Lancer la synchronisation

Pour lancer la synchronisation :

1. Sélectionnez la base de données concernée (dans la liste des bases de données ou dans un diagramme par exemple).
2. Cliquez sur l'icône de la base de données et sélectionnez **Synchroniser**. L'assistant de synchronisation apparaît.
3. Sélectionnez le type de synchronisation "physique vers logique".
4. Cliquez sur **Suivant**.

Etape 1 : sélectionner les objets à synchroniser

Pour définir le périmètre de la synchronisation :

Déroulez la liste des objets contenus dans la base de données.

1. Par défaut, tous les objets sont cochés et donc inclus dans la synchronisation. Pour exclure un objet de la synchronisation, décochez-le dans la colonne **Périmètre**. Lorsqu'un objet est exclu, sa correspondance l'est aussi.
2. Par défaut, tous les objets sont "réalisés", autrement dit donnent lieu à la création d'un objet lors de la synchronisation. Pour indiquer qu'un objet est "non réalisé", cochez-le dans la colonne **Non réalisé**. Pour plus d'informations voir [Mode Réalisé](#).
3. Une fois la liste des objets définie, cliquez sur le lien **Suivant** de l'assistant.

Etape 2 : options de synchronisation

Parmi les options de synchronisation, vous pouvez :

- Réinitialiser les objets cibles : la synchronisation repart de zéro et supprime les objets cibles existants.
- Recalculer le nom des objets cibles : les noms des objets du diagramme de données sont recalculés en fonction de ceux des objets du diagramme relationnel. Cela signifie que toute modification manuelle du diagramme de données est annulée.
- Prendre en compte les optimisations : toutes les optimisations - dont celles qui ne sont pas cochées dans l'étape de validation (voir [Validation des optimisations](#)) sont proposées.
- Prendre en compte les suppressions : les entités, associations et diagrammes qui ont été supprimés font partie du périmètre. En conséquence, la suppression des objets ou des liens cibles correspondants est proposée.
Voir [Cas d'emploi des options](#)

Vous devez également indiquer le modèle de données qui va détenir l'ensemble des objets créés par la synchronisation.

Les autres options concernent la mise à jour des propriétés des objets cibles. Par défaut, la synchronisation met à jour l'ensemble des propriétés de chaque objet concerné.

Scheduling

Vous pouvez exécuter la synchronisation :

- Immédiatement
- dès que possible (après publication des mises à jour)
- A une date et heure prédéfinies

】 Une fois les options définies, cliquez sur **Suivant**.

Etape 3 : protéger des objets

La synchronisation peut avoir un impact sur tous les objets cibles.

- 】 Pour garder un objet intact, cochez-le dans la colonne **Figé**.
- 】 Cliquez sur **Suivant** pour passer à la suite.

Voir [Protéger des objets](#).

Etape 4 : valider les résultats

L'assistant affiche les résultats qu'entraînera la validation de la synchronisation.

- 】 Pour valider ces résultats, cliquez sur **Suivant**.

Un compte-rendu présente la liste des traitements effectués.

Synchronisation réduite

La synchronisation ci-dessus s'applique à une base de données mais vous pouvez également lancer une synchronisation Physique->Logique sur un objet particulier d'une base de données afin de réduire le périmètre et le temps de traitement de la synchronisation. Voir [Synchronisation réduite \(mode logique vers physique\)](#).

Résultat de la synchronisation "Physique vers Logique"

Modèle de données détenteur

Un modèle de données détenteur par défaut des entités est créé lors de la synchronisation Physique vers Logique. Les entités et associations synchronisées lui sont automatiquement rattachées. Vous pouvez par la suite répartir ces entités et associations entre les différents modèles de données de votre étude.

Diagrammes de données

Un diagramme de données est créé pour chacun des diagrammes relationnels de la base de données. Les entités et associations issues des tables du diagramme relationnel correspondant y sont reliées.

Les correspondances

Voir [Résultats de la synchronisation : les correspondances](#).

PARAMÉTRER LA SYNCHRONISATION

Cette partie présente les options et les paramètres par défaut pris en compte lors de la synchronisation.

Préparer la synchronisation

Pour préparer la synchronisation :

1. Sélectionnez la base de données.
Une barre de commandes apparaît à droite de la base de données.
2. Cliquez sur **Propriétés**.
La fenêtre de propriétés apparaît.
3. Cliquez sur la page **Options > Standard**.
4. Si vous voulez que le nom de la base physique utilisé lors de la génération de scripts SQL soit différent de celui de la base de données, indiquez le nom de la base cible dans le champ **NomSql**.
5. Indiquez éventuellement le **Préfixe** des tables SQL. Ce préfixe sera ajouté devant le nom de chacune des tables générées.
*☛ Des paramètres supplémentaires pour le paramétrage des ordres de synchronisation et de génération peuvent être indiqués sous les **Options**. Ces paramètres varient en fonction du SGBD sélectionné.*
6. Cliquez à nouveau sur la liste déroulante de la fenêtre de propriétés puis sur **Caractéristiques**.
7. Sélectionnez le **SGBD cible** ainsi que sa version.
☛ Le type de SGBD détermine :
 - Pour la synchronisation, la génération des datatypes des colonnes en fonction du type et de la longueur des attributs (voir [Déduire les datatypes des colonnes à partir des types des attributs](#)).
 - Pour la génération, la syntaxe des ordres SQL générés.
8. Cliquez sur **OK** pour fermer la fenêtre en prenant en compte les modifications.

Options de création

Sur une base de données

Il est possible de paramétrer la synchronisation pour chaque base de données, afin de modifier :

- Ses options de création
- Le traitement de l'intégrité référentielle (clés en suppression et en mise à jour, en fonction des possibilités offertes par le SGBD cible)

Cette configuration concerne également le traitement des colonnes Not Null et la création automatique d'index sur les clés primaires.

Pour configurer les options de création de la base de données :

1. Faites un clic droit sur la base de données et sélectionnez **Propriétés**.
La fenêtre de propriétés apparaît.
2. Cliquez sur la page **Options** > **Synchronisation**.
Les options correspondantes s'affichent.

Vous pouvez renseigner les paramètres suivants

- **Colonnes Not Null** : active/désactive l'option WITH DEFAULT pour les colonnes Not Null.
- **OnDelete** : stratégie par défaut pour la suppression des clés. Les valeurs possibles sont :
 - **Restrict** : la suppression est refusée.
 - **Cascade** : la suppression d'une colonne est répercutée dans les tables dépendantes.
 - **Set Null** : indique "Null" à la place.
 - **Set Default** : donne la valeur par défaut lorsque celle-ci est renseignée. Si aucune valeur par défaut n'est renseignée, il ne se passe rien ("No action").
 - **No Action** : il ne se passe rien.
- **OnUpdate** : stratégie par défaut pour la mise à jour des clés.
- **Noms cols PK auto** : colonnes issues de l'identifiant implicite. Voir [Règle générale](#).

Les paramètres dont le nom commence par "**Noms de**" permettent d'indiquer les règles de génération des noms (voir [Paramétrer la génération des noms](#)).

Sur le SGBD

Les valeurs par défaut des paramètres de synchronisation et de génération des bases de données sont accessibles dans la fenêtre de propriétés du SGBD utilisé.

Pour afficher les paramétrages de synchronisation d'un SGBD :

1. Dans la zone d'édition, cliquez sur le bouton **Menu principal** puis sur **Recherche avancée**.
2. Sélectionnez le type d'objet "Version SGBD" et cliquez sur **Chercher**.
3. Faites un clic droit sur le nom du SGBD cible et ouvrez sa fenêtre de **Propriétés**.
4. Cliquez sur la liste déroulante puis sur **Options > Synchronisation**.

Par défaut, les paramètres spécifiés au niveau du SGBD sont valables pour toutes les nouvelles bases de données. Lorsque vous modifiez les paramètres de synchronisation au niveau d'une base, cette base ne tient plus compte des paramètres du SGBD.

 Pour plus d'informations sur le paramétrage de la synchronisation, voir également [Lancer la synchronisation](#).

Paramétrer la génération des noms

Règles de construction des noms

Les noms des objets physiques créés lors de la synchronisation "logique vers physique" sont déduits du **Nom Local** des objets logiques dont ils sont issus.

Les noms des objets logiques (classe, association, partie, attribut, rôle) n'étant soumis à aucune restriction particulière, des règles de transformation s'appliquent par défaut lors de leur synchronisation. Ces règles sont accessibles localement dans la page **Options > Standard** des propriétés de la base de données synchronisée, ou globalement dans les propriétés des SGBD cibles :

- **Taille identifiant** : taille maximale d'un identificateur SQL pour ce SGBD cible.
- **1er caractère** : jeu de caractères autorisé pour le 1er caractère d'un identificateur SQL.
- **Caractères autorisés** : jeu de caractères autorisé pour les caractères d'un identificateur SQL.
- **Caractère de remplacement** : caractère de remplacement des caractères non autorisés.
- **Caractères convertis** : jeu de caractères à convertir d'un identificateur SQL.
- **Caractères de conversion** : jeu de caractères correspondant aux caractères à convertir.
- **Conversion majuscule** : conversion en majuscules des identificateurs SQL.

Il est possible d'indiquer un autre nom pour chaque objet synchronisé à l'aide de son **Nom SQL**. Le **Nom SQL** se substitue au **Nom Local** lors de la synchronisation, tout en tenant compte des règles de transformation par défaut.

Le **Nom SQL** des objets logiques est accessible dans la page **Génération > SQL** (ou page **SQL**) de leur fenêtre de propriétés.

Vous pouvez donner un nom différent selon la base de données ou le SGBD.

/ Vue hiérarchique / Location / PostgreSQL9.3 | ← Retour

Rechercher...

Location véhicule

Eléments Détenus

Club fidélité

Compte fidélité

Enregistrement Local

Agence

Catégorie durée

Conducteur

Location

Taux location

Logistique

Médiathèque - Emprunts

Location

généralisations

Politiques de qualité de données

Génération SQL

Rapports

NomSql

BodyNumber

Longueur

Décimale

Type SQL

Base-de-données

Relier

Propriétés

Rapport instantané

Nom Local ↑

Nom SQL (Base de données)

Location voiture

Rental_BodyNum

<< < | Page 1 sur 1 | > >> | Montrer 50 éléments Pa

Version SGBD

Relier

Propriétés

Rapport instantané

Nom Local ↑

Nom SQL (SGBD)

PostgreSQL9.3

POSBodNum

Lorsque les champs **Nom SQL (Base de données)** et **Nom SQL (SGBD)** indiquent des noms différents, c'est le nom défini au niveau de la base de données qui prévaut lors de la synchronisation.

Par défaut, les noms des objets relationnels sont générés selon les masques suivants :

Table	Préfixe de la base de données + nom* de l'entité ou de l'association
Colonne	Nom* de l'attribut
Clé primaire	"PK_" + nom* de l'entité ou de l'association
Clé étrangère	"FK_" + nom* de l'entité cible ou du rôle si renseigné
Index de clé primaire	"IDX_" + nom* de l'entité ou de l'association
Index de clé étrangère	"IDX_" + nom* de l'entité cible ou du rôle si renseigné

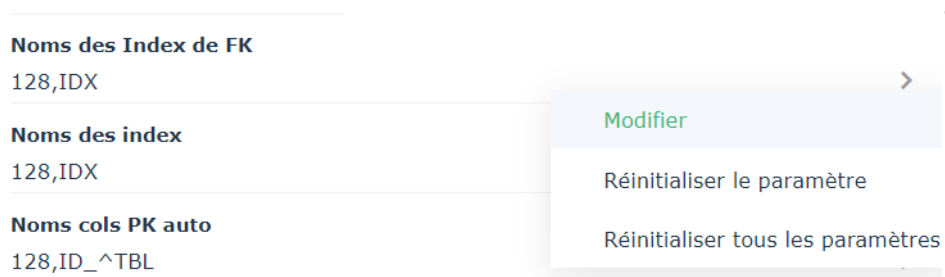
**Nom calculé selon les règles de construction expliquées précédemment.*

Ces masques peuvent être modifiés localement dans chaque base de données, ou globalement pour un SGBD cible donné.

Modifier une règle de construction

Pour modifier le masque d'une règle de construction de nom :

1. Faites un clic droit sur la base de données et sélectionnez **Propriétés**.
2. Cliquez sur la liste déroulante puis sur **Options** > **Synchronisation**.
3. Dans le champ de la règle en question, cliquez sur la flèche.



4. Cliquez sur **Modifier**.
La fenêtre de **Saisie du masque SQL** est présentée.

Saisie du masque SQL

Les masques SQL définissent la règle de construction des noms des objets relationnels lors de la synchronisation.

Exemple : dans la base BASE_EMPLOYES qui a pour préfixe EMP, le masque ^DB_^ROOT génère pour la table issue de l'entité Client : EMP_CLIENT.

Dans la fenêtre de saisie du masque SQL, vous pouvez saisir directement le **Masque**, en utilisant la syntaxe indiquée ci-après, mais vous pouvez également utiliser l'aide à la saisie proposée par le cadre **Composant**.

- » Dans la liste du cadre **Composant**, indiquez les éléments qui doivent préfixer les noms. Ces éléments sont les suivants :

^ROOT	• Pour une table : nom* de la classe, de l'association ou de la partie dont elle provient. • Pour une colonne : nom* de l'attribut ou nom de l'identifiant. • Pour une clé primaire : nom* de la classe, de l'association ou partie dont elle provient. • Pour une clé étrangère : nom* de la classe cible ou du rôle si renseigné. • Pour une colonne de FK : nom* de l'attribut. • Pour une colonne de PK auto : nom* de l'identifiant de la classe. • Pour un index sur clé primaire : nom* de la classe, de l'association ou de la partie. • Pour un index sur clé étrangère : nom* de la classe cible ou du rôle si renseigné. • Pour un index : nom* de l'attribut, du rôle ou de la classe.
^DB	Préfixe de la base de données
^EXT	• Pour une clé étrangère : nom* de l'association, de la partie ou de la généralisation • Pour un index sur clé étrangère : nom* de l'association, de la partie ou de la généralisation
^TBL	Nom local de la table ou nom local de la table de référence
^TBO	• Pour une clé étrangère : nom* de la classe, de l'association ou de la partie dont elle provient • Pour un index sur clé étrangère : nom* de la classe, de l'association ou de la partie d'origine
^TBR	Nom de la table de référence
^KEY	Nom de la clé étrangère
^CPT	Composteur

**Nom calculé selon les règles de construction expliquées précédemment.*

La **Taille** indique la longueur limite totale du nom généré. Elle est également disponible pour chacun des éléments utilisés, qui seront tronqués au nombre de caractères indiqué entre parenthèses après l'élément concerné.

La définition d'un **Composteur** ("^CPT") permet de générer automatiquement un numéro d'ordre, et d'en indiquer la longueur (par exemple, ^CPT[^1^] générera "1", "2", "3"; ^CPT[^3^] générera "001", "002", "003").

L'option **Toujours** indique que le compostage commence dès la première occurrence (CLI00, CLI01, .., au lieu de CLI, CLI01,...).

Vous pouvez également préciser les caractères utilisés comme préfixe et suffixe de ce composteur.

L'option **Unicité du nom** permet de garantir l'unicité du nom d'un objet au niveau de la base de données, du référentiel ou de la table dans lesquels se trouve cet objet. Ainsi, lorsque vous définissez l'unicité de nom au niveau de la table, il n'est pas possible d'attribuer un même nom à des objets différents au sein d'une même table.

Paramétrer le nom des colonnes de PK (identifiant implicite)

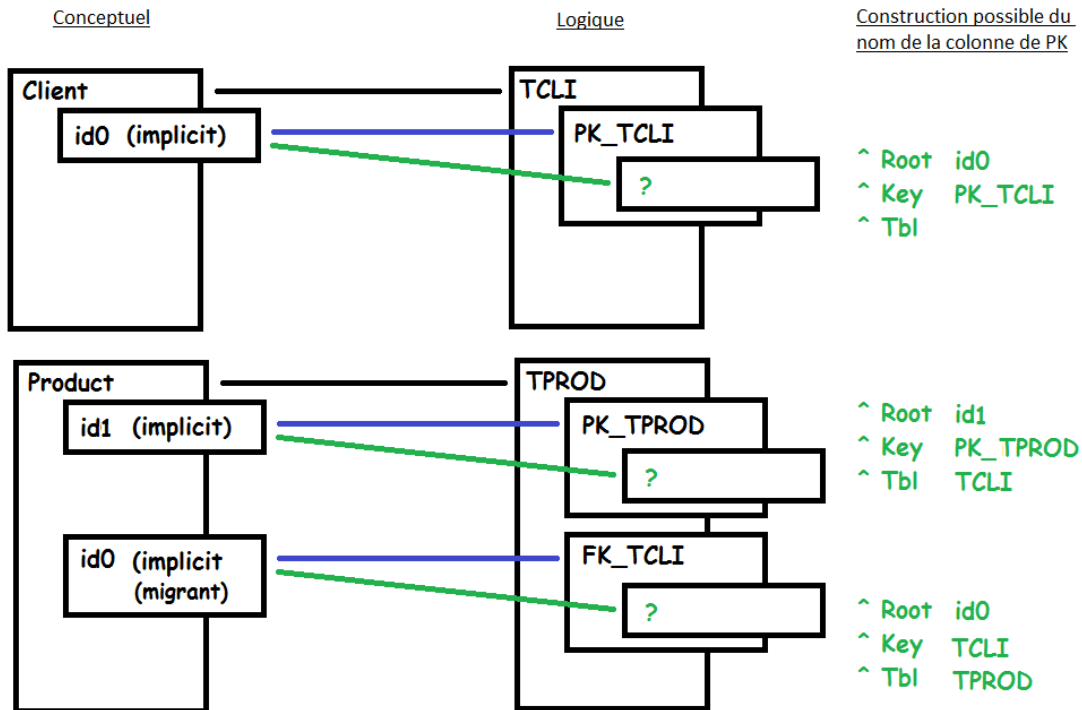
Lors de la synchronisation en mode Logique > Physique, l'identifiant de l'entité devient la clé primaire de la table. Si l'identifiant est implicite, une colonne est automatiquement créée. Pour plus de détails, voir [Règles de synchronisation "logique vers physique"](#).

Par défaut le nom d'une colonne issue d'un identifiant implicite est construit à l'aide du mot clé ^TBL qui correspond :

- au nom de la table migrante (autrement dit issue d'une clé étrangère) si l'identifiant est migrant
- au nom de la table si l'identifiant n'est pas migrant

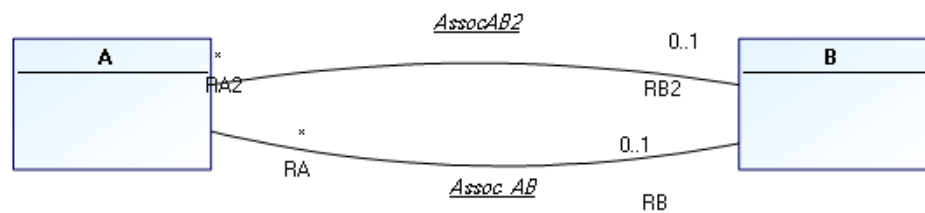
Vous pouvez modifier les règles de construction et construire le nom de ces colonnes avec le mot clé ^KEY qui correspond au nom de la clé étrangère (sans " FK_ ") si

l'identifiant est migrant, ainsi qu'avec le mot clé ^ROOT qui correspond au nom de l'identifiant (ID). Voir [Modifier une règle de construction](#).

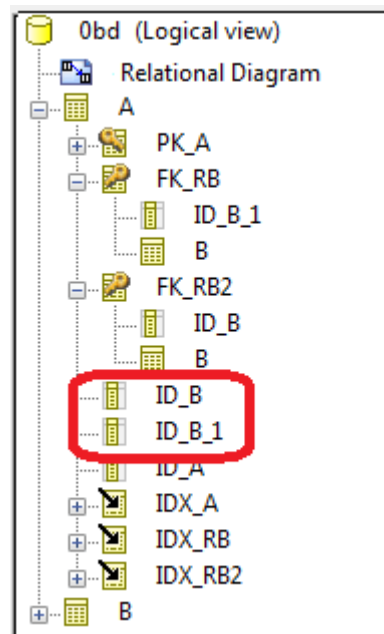


Exemple

Lorsqu'il existe deux associations contraintes entre deux entités comme ci-dessous :



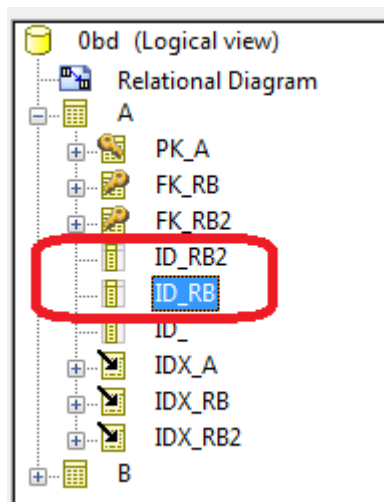
Par défaut, après synchronisation, vous obtenez deux colonnes avec des noms identiques seulement différenciés par le préfixe "1".



Vous pouvez modifier la règle de nommage et construire le nom de ces colonnes avec le mot clé ^KEY qui correspond au nom de la clé étrangère (sans " FK_ ").

Le nom de la clé étrangère étant calculé sur le nom du Rôle lorsqu'il est renseigné, alors le nom obtenu pour les deux colonnes sera différent.

Dans notre exemple, si vous remplacez "ID^TBL" par "ID^KEY" vous obtenez après synchronisation :



SYNCHRONISATION DES DIAGRAMMES

La synchronisation permet de traduire un modèle de données logique en un modèle physique, et inversement. Avant de lancer une synchronisation, les diagrammes sources (diagrammes de données ou diagrammes physiques) doivent être enregistrés et fermés.

Une première synchronisation crée de façon automatique le diagramme du modèle cible. Une nouvelle synchronisation sur un modèle préalablement synchronisé n'entraîne pas une mise à jour automatique du diagramme, autrement dit de la représentation graphique du modèle. Suivant les modifications que vous avez apportées, l'assistant de synchronisation propose ou non de mettre à jour le diagramme cible.

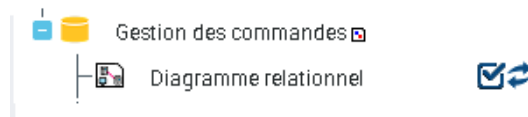
Cas de mise à jour des diagrammes lors d'une synchronisation

L'assistant de synchronisation propose une mise à jour des diagrammes selon les cas suivants.

Après modification du diagramme source

Lorsque vous lancez la synchronisation après modification du diagramme source, l'assistant active par défaut la mise à jour du diagramme cible. La mise à jour est signalée par l'affichage des deux petites flèches bleues.

Ci-dessous, lors d'une synchronisation en mode logique vers physique, la modification du diagramme logique entraîne automatiquement la mise à jour du diagramme physique.



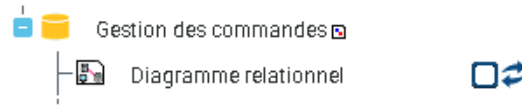
Après modification du diagramme cible

Si vous avez modifié le diagramme cible, par exemple le diagramme physique, par défaut la synchronisation ne propose pas de mettre à jour ce diagramme cible.

Pour afficher la case de mise à jour du diagramme cible, vous devez cocher l'option de synchronisation "Prendre en compte les optimisations". Et pour que la mise à jour soit activée, vous devez cocher la case en question.

Après modification des deux diagrammes

Si les deux diagrammes - logique et physique - ont été modifiés, la mise à jour du diagramme cible est proposée mais non cochée par défaut.



Pas de modification détectée

Si aucun des deux diagrammes n'a été modifié, l'assistant ne propose aucune mise à jour. Il est à noter que toute modification d'un diagramme doit faire l'objet d'une sauvegarde avant fermeture du diagramme pour être prise en compte par l'assistant de synchronisation.

Cas particulier : une entité en correspondance avec deux tables

Lorsqu'une entité du modèle logique est associée à deux tables dans le modèle physique (suite à une fusion des entités par exemple), le diagramme physique mis à jour n'affiche qu'une des deux tables.

CORRESPONDANCE DES MODÈLES



Dans beaucoup de projets de modélisation se pose le problème de la communication entre les équipes d'analystes, d'architectes et les équipes de développement des bases de données.

Il existe dans **Hopex Data Architecture** deux niveaux de modélisation :

- ✓ Le niveau logique, qui décrit la modélisation des données en termes d'entités et de relations, et qui s'adresse aux analystes et utilisateurs.
- ✓ Le niveau physique (ou relationnel) qui décrit la base de données en termes de tables et qui fait l'interface avec le SGBD. Ce niveau s'adresse au concepteur et à l'administrateur de base de données.

En permettant le passage d'un modèle de données à un autre, l'éditeur de base de données favorise la cohésion entre l'architecture des données et les systèmes qui les supportent.

Les points abordés ci-après sont :

- ✓ [L'éditeur de base de données](#)
- ✓ [Détails des correspondances](#)

L'ÉDITEUR DE BASE DE DONNÉES

L'éditeur d'une base de données permet de mettre en correspondance de façon manuelle les différentes vues d'une base de données, qui sont le modèle logique et le modèle physique (ou relationnel).

L'assistant de synchronisation met en correspondance de façon automatique les deux vues. Voir [Synchroniser les modèles logiques et physiques](#).

Après synchronisation vous pouvez créer ou modifier manuellement des correspondances dans l'éditeur, mais cette méthode ne garantit plus la cohérence entre les deux modèles. L'assistant de dénormalisation permet de maintenir cette cohérence. Voir [Dénormaliser les modèles logiques et physiques](#).

Lancer l'éditeur sur une base de données

Pour ouvrir l'éditeur sur une base de données:

- 1 Cliquez sur l'icône de la base de données et sélectionnez **Éditeur de correspondances**.

L'éditeur de correspondances juxtapose la vue logique et la vue physique de la base. Lorsqu'un arbre de correspondance existe, il l'affiche automatiquement. Lorsqu'aucun arbre n'a été créé pour la base de données, une fenêtre vous propose de le créer.

Créer un arbre de correspondance logique/physique

Pour créer un arbre de correspondance :

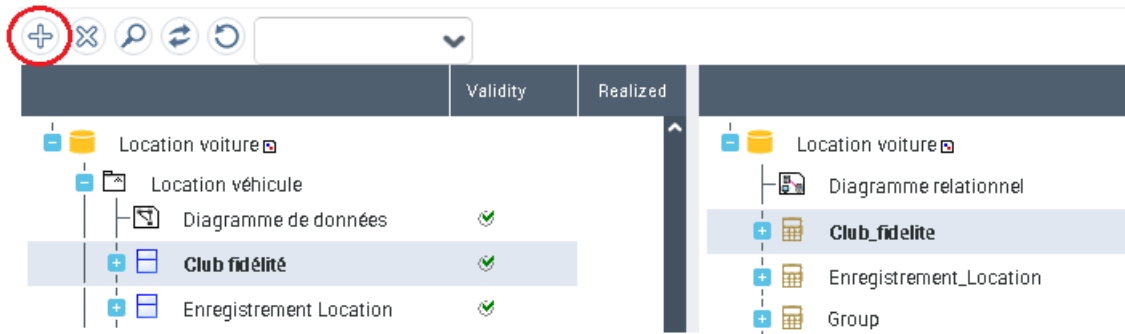
1. Dans la fenêtre de création, indiquez le nom du nouvel arbre de correspondance.
2. Dans le champ **Nature**, précisez le type de correspondance effectué : "Logique/Physique".
3. Dans les cadres **Objet gauche** et **Objet droit**, sélectionnez les modèles logique et physique que vous souhaitez aligner.
4. Cliquez sur **OK**.
L'éditeur affiche l'arbre de correspondance qui juxtapose les deux modèles.

Créer une correspondance

Pour créer une correspondance entre une entité et une table :

1. Dans l'éditeur de la base de données, sélectionnez l'entité puis la table.

2. Cliquez sur le bouton **Etablir une correspondance**.

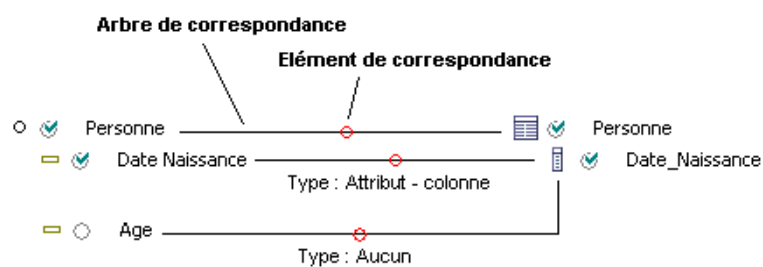


La correspondance se crée à partir du dernier objet sélectionné. Ainsi, pour créer une correspondance du modèle logique vers physique, autrement dit pour définir un objet du modèle physique à partir d'un objet du modèle logique, vous devez sélectionner l'objet du modèle logique puis celui du modèle physique, et créer la correspondance depuis ce dernier. Si une correspondance ne peut être créée, un message d'erreur apparaît (Voir [Sens de la synchronisation](#)).

Exemple de nouvelle correspondance

Prenons l'entité "Personne" qui contient l'attribut "Date Naissance". Elle a comme correspondance dans le formalisme physique la table "Personne" qui contient la colonne "Date_Naissance".

Supposons que l'on ajoute l'attribut "Age" sur l'entité. Celui-ci peut être calculé depuis la date de naissance. Pour ne pas créer de colonne correspondant à ce nouvel attribut lors de la synchronisation, vous pouvez le relier directement à la colonne "Date_Naissance".



Pour créer une correspondance entre l'attribut "Age" et la colonne "Date_Naissance" :

1. Dans l'éditeur, sélectionnez d'un côté la colonne "Date_Naissance" et de l'autre l'attribut "Age" .
2. Cliquez sur le bouton **Etablir une correspondance**.

Une fois la correspondance créée, une coche apparaît devant l'attribut "Age".

Supprimer une correspondance

Pour supprimer une correspondance sur un objet :

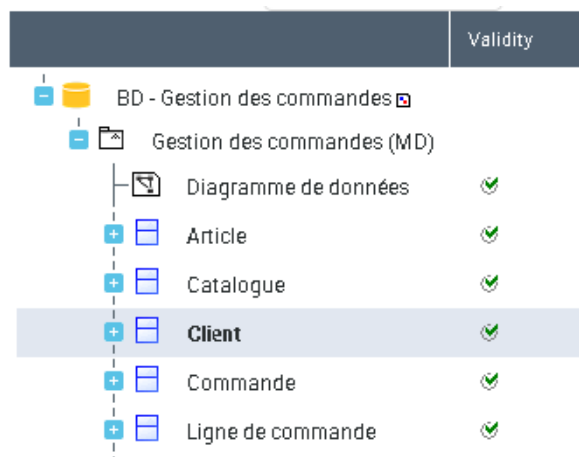
- Sélectionnez l'objet en question et cliquez sur le bouton **Enlever la correspondance**.

DÉTAILS DES CORRESPONDANCES

Les objets qui ont une correspondance sont cochés en vert. Lorsque vous sélectionnez l'un de ces objets, sa correspondance apparaît dans la fenêtre de propriétés de la correspondance située par défaut en bas de l'éditeur de base de données. Elle regroupe le nom des objets reliés dans les deux formalismes, le type des objets et éventuellement un commentaire.

Exemple de correspondance

On sélectionne la table "Client" dans l'arbre de la vue logique :



La correspondance affiche les objets suivants :

Validité	Objet logique	Objet physique	Type
✓ Valide	Client	Client	Entité - Table

Cela signifie que la table "Client" est issue de l'entité du même nom.

Propriétés d'une correspondance

Pour visualiser les propriétés d'une correspondance :

1. Dans l'éditeur de correspondances, sélectionnez la correspondance et cliquez sur **Propriétés**.
2. Dans la fenêtre qui apparaît, cliquez sur la liste déroulante puis sur **Caractéristiques**.

Voir aussi [Caractéristiques de la correspondance](#).

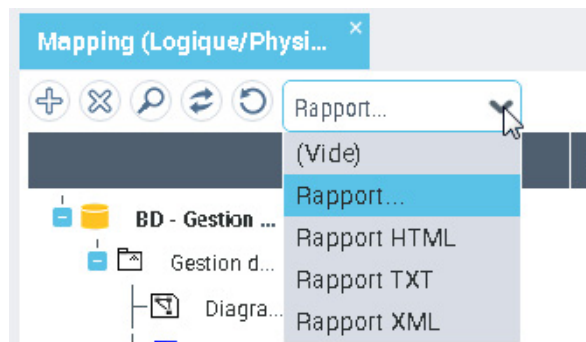
Rapport des correspondances

Vous pouvez générer dans un document le détail des correspondances entre les deux modèles de la base de données. Il peut s'agir d'un fichier HTML, texte ou XML.

Générer un rapport HTML

Pour générer le rapport HTML d'un arbre de correspondance :

- 1 Dans la barre d'outils de l'éditeur de la base de données, cliquez sur la liste déroulante puis sur **Rapport HTML**.



Le fichier correspondant s'ouvre. Il présente sous forme de tableau le détail des correspondances de la vue physique d'une part et de la vue logique d'autre part.

Chaque ligne du tableau présente un objet du modèle et son équivalence dans l'autre modèle. Au milieu de chaque ligne apparaît l'état de la correspondance entre les deux objets.

Description of Mapping View "Vue logique"	
Object	Mapping
BD - Gestion des commandes	
Modèle relationnel	
Article	Article
FK_Type_article	ID Article type3
Code_type_article	Code type article
	Code type article
	Code type article

A la fin du document, vous trouvez également la liste des correspondances invalides.

Etat des objets

Des indicateurs permettent d'indiquer l'état des objets mis en correspondance. Une barre de filtrage vous permet d'afficher l'ensemble ou uniquement certains de ces indicateurs. Cette barre est disponible sous le menu **Affichage > Barre d'outils** de l'éditeur.

Les objets peuvent être caractérisés par les états suivants :



Valide



Invalide (Lorsqu'un objet a conservé une correspondance vers un objet qui n'existe plus)



Sans correspondance



Figé (Protégé)



Standard

Enregistrer l'affichage des indicateurs de l'éditeur

Une option vous permet d'enregistrer l'état des indicateurs dans l'éditeur de correspondance. Elle est propre à l'utilisateur et à l'arbre de correspondance courants. Ainsi, un utilisateur qui a coché l'option d'affichage des indicateurs retrouvera automatiquement l'état des objets dans l'arbre de correspondance qu'il a précédemment créé.

L'option d'affichage des indicateurs de l'éditeur est décochée par défaut. Pour l'activer :

1. Dans le bureau, cliquez sur **Menu principal > Paramètres > Options**.
2. Dans la partie gauche de la fenêtre des options, cliquez sur **Éditeur de correspondance**.
3. Dans la partie droite, cochez l'option **Enregistrer l'affichage des indicateurs de l'éditeur**.
4. Cliquez sur **OK**.

Source de la correspondance

En sélectionnant un objet dans un des formalismes présentés dans l'éditeur, vous pouvez afficher sa correspondance dans l'autre formalisme.

Pour afficher la correspondance d'un objet :

1. Dans l'éditeur de correspondances, sélectionnez l'objet en question et cliquez sur le bouton **Chercher**.

L'éditeur affiche l'objet source.

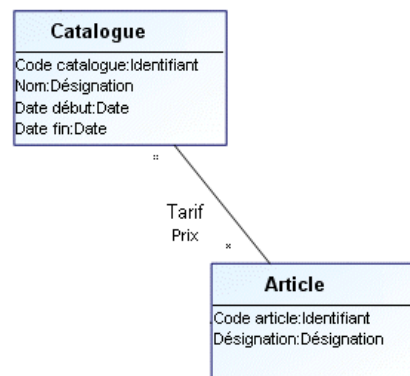
Exemple de correspondance

Prenons la table "Tarif" créée dans le modèle logique. Dans le menu contextuel de la table, cliquez sur **Chercher**. Vous constatez qu'elle correspond à une association.

Mapping (Logique/Physi... x

	Validity		Validity
BD - Gestion des commandes		BD - Gestion des commandes	
Gestion des commandes (MD)		Diagramme relationnel	✓
Diagramme de données	✓	Modèle relationnel	○
Article	✓	Article	✓
Catalogue	✓	Catalogue	✓
Client	✓	Client	✓
Commande	✓	Commande	✓
Ligne de commande	✓	Ligne_de_commande	✓
Personne	✓	Personne	✓
Société	✓	Remise	✓
Type article	○	Societe	✓
Remise	✓	Tarif	✓
Tarif	✓		

En ouvrant le diagramme du modèle logique, vous pouvez voir qu'il s'agit de l'association "Tarif" qui relie les entités "Catalogue" et "Article".



Lors de la synchronisation, cette association non contrainte donne lieu à une table dans laquelle :

- Une colonne est créée pour l'identifiant de chaque entité reliée.
- La clé primaire de la table porte sur l'ensemble de ces colonnes.
- Une clé étrangère est constituée pour chaque entité reliée.

Pour plus d'informations sur la synchronisation des associations, voir [Synchronisation logique > physique : les Associations](#).

Dessin de la correspondance

Pour visualiser le dessin d'une correspondance :

- 】 Sélectionnez dans l'arbre de correspondance les objets concernés.

Une fenêtre apparaît en bas de l'éditeur et présente le dessin des objets et les liens de correspondance qui les unissent.



DÉNORMALISER LES MODÈLES LOGIQUES ET PHYSIQUES



Après avoir synchronisé un modèle de données et un modèle physique, vous pouvez les faire évoluer en apportant des modifications directement dans leur diagramme. Mais cette méthode ne garantit pas la cohérence entre les deux modèles.

Les assistants de dénormalisation ont été créés pour maintenir cette cohérence. Il vous permettent de toucher à la définition d'un modèle tout en maintenant les correspondances avec l'autre.

Les assistants vous permettent également d'effectuer rapidement des opérations de type duplication ou fusion d'entités sans report des correspondances et/ou sans suppression des objets sources lorsque vous souhaitez reporter les modifications dans le modèle relationnel.

 **Dans Hopex Data Architecture V2R1, la fonctionnalité de dénormalisation est disponible uniquement pour le modèle de données, elle ne s'applique pas aux paquetages (notation UML).**

Les points suivants sont abordés ici :

- ✓ [Principes de la dénormalisation](#)
- ✓ [Dénormalisation logique](#)
- ✓ [Dénormalisation physique](#)

PRINCIPES DE LA DÉNORMALISATION

La dénormalisation permet de transformer ou d'affiner les modèles en fonction de besoins spécifiques : choix de modélisation, optimisation des performances, choix de mise en œuvre physique, redondances etc.

L'outil de dénormalisation est présenté sous forme d'assistants permettant de réaliser ces transformations.

Un assistant s'applique à un objet ou à un groupe d'objets. Selon le type d'optimisation demandé, la dénormalisation crée dans un modèle de nouveaux objets (objets cibles) à partir des objets de départ (objets sources).

Dénormalisation : cohérence des modèles

La dénormalisation effectue un changement sur l'objet d'un modèle. Lorsque ce modèle a été mis en correspondance avec un autre (voir la [Synchroniser les modèles logiques et physiques](#)), vous devez gérer l'impact du changement sur l'autre modèle.

Lors de la dénormalisation, vous pouvez ainsi :

- Reporter ou non les correspondances avec les objets synchronisés.
- Supprimer ou conserver les objets sources.

Report des correspondances

Le report des correspondances permet d'assurer la stabilité et la cohérence entre deux modèles. Lorsqu'une dénormalisation crée un nouvel objet au niveau logique, la correspondance avec l'objet physique est reportée sur le nouvel objet logique. Lorsque vous décochez cette option, les correspondances vers les nouveaux objets ne sont pas créées, les deux modèles doivent donc être re-synchronisés.

 **Afin que la synchronisation valide les changements occasionnés par la dénormalisation, veillez à ce que l'option "Réinitialisation des objets cibles" soit décochée.**

Supprimer les objets sources

Par défaut, les objets sources sont supprimés. La non suppression des objets sources vous permet de conserver les objets de départ après la dénormalisation.

Synchronisation et dénormalisation

Dans l'activité de modélisation de données, synchronisation et dénormalisation doivent souvent être combinées pour répondre à tel ou tel cas d'emploi.

Exemple

Soit une entité PAIEMENT que vous voulez représenter dans la base de données par trois tables VIREMENT, CHEQUE et AUTRE. Pour réaliser cette modélisation :

1. Créez l'entité PAIEMENT.
2. Lancez la synchronisation (logique vers physique) pour obtenir la table PAIEMENT.
3. Lancez l'assistant physique pour partitionner horizontalement la table PAIEMENT.
4. Renommez les trois duplicatas en VIREMENT, CHEQUE et AUTRE.

Les trois tables ainsi obtenues sont dès lors liées à l'entité PAIEMENT et suivent ses évolutions lors des futures synchronisations.

Combinaison des options de dénormalisation et synchronisation

L'impact d'une dénormalisation sur deux modèles synchronisés varie en fonction des options choisies.

Prenons l'exemple d'une dénormalisation logique. Les combinaisons possibles sont :

- Suppression des objets sources + report des correspondances : les objets sources du modèle logique sont supprimés. Le niveau physique reste inchangé ; la correspondance avec les objets logiques résultant de la dénormalisation est assurée.
- Suppression des objets sources + non report des correspondances : les objets physiques correspondant aux objets logiques résultant de la dénormalisation sont créés. Les objets physiques correspondant aux objets logiques supprimés sont supprimés.
- Non suppression des objets sources + report des correspondances : les objets sources du modèle logique sont conservés. Le niveau physique reste inchangé ; la correspondance avec les objets résultant de la dénormalisation est assurée.
- Non suppression des objets sources + non report des correspondances : les objets physiques correspondant aux objets résultant de la dénormalisation sont créés. Les objets physiques existants sont conservés.



Cas particuliers : fusions ascendante et descendante :

Dans le cas d'une fusion ascendante, l'entité sur-type a un rôle particulier. La dénormalisation conserve sa correspondance dans tous les cas.

De même, dans le cas d'une fusion descendante, les entités sous-types jouent un rôle particulier ; leur correspondance est conservée dans tous les cas.

Dénormalisation : cas d'emploi

La combinaison des options de dénormalisation varie en fonction du mode de conception de vos modèles.

1. Préserver la stabilité du niveau physique lorsqu'une modification est appliquée au niveau logique.

Contexte : une synchronisation a déjà été établie entre les niveaux logique et physique. Le niveau physique est en production. Une modification doit être appliquée au niveau logique, sans impact sur le niveau physique.

Options préconisées pour la dénormalisation : report des correspondances, suppression des objets sources.

Ce cas d'emploi correspond à un mode de travail orienté maintenance anticipée : des modifications sont effectuées par anticipation sur le niveau logique, sachant que le niveau physique ne doit pas être modifié jusqu'à nouvel ordre.

Résultat : après la dénormalisation, les correspondances sont rétablies entre les objets logiques cibles et les objets physiques. Lors de la synchronisation, rien ne change au niveau physique.

2. Faire évoluer le niveau physique lorsqu'une dénormalisation est appliquée au niveau logique.

Contexte : une synchronisation a déjà été établie entre les niveaux logique et physique. Le niveau physique n'est pas figé et doit évoluer en fonction du niveau logique.

Options préconisées pour la dénormalisation : non report des correspondances, suppression des objets sources.

Ce cas d'emploi permet de privilégier les évolutions du niveau logique, sans se préoccuper de l'impact au niveau physique.

Résultat : après la dénormalisation, les objets logiques cibles sont sans correspondance; les objets physiques correspondant aux objets logiques sources ne sont plus synchronisés. Après synchronisation, le niveau physique est mis à jour : les objets physiques correspondant aux objets logiques sources (objets existants avant la dénormalisation) sont supprimés ; de nouveaux objets physiques correspondant aux objets logiques cibles (objets créés par la dénormalisation) sont créés.

3. Faciliter le développement du niveau logique, en phase de développement.

Contexte : une synchronisation a déjà été établie entre les niveaux logique et physique, ou bien le niveau physique n'est pas encore implémenté.

Options préconisées pour la dénormalisation : non report des correspondances, non suppression des objets sources.

Ce cas d'emploi correspond à un mode de travail "incrémental" : les objets sources du niveau logique restent inchangés. Le modèle est enrichi des objets cibles issus de la dénormalisation. Ces objets cibles donnent lieu à une partie nouvelle dans le niveau physique, la partie physique existante restant stable.

Résultat : après la dénormalisation, les correspondances restent inchangées. Après synchronisation, de nouveaux objets physiques correspondant aux nouveaux objets logiques sont créés ; les objets physiques correspondant aux objets logiques sources sont inchangés.

4. Favoriser la mise en place de scénarios multiples, en phase de développement.

Contexte : une synchronisation a déjà été établie entre les niveaux logique et physique, plusieurs options de modélisation coexistent temporairement, pour un seul niveau physique.

Options préconisées pour la dénormalisation : report des correspondances, non suppression des objets sources.

Ce cas d'emploi, à utiliser avec précaution, permet de conserver deux options de modélisation au niveau logique, donnant lieu à un résultat commun au niveau physique.

Résultat : après la dénormalisation, les objets physiques restent reliés aux objets logiques sources, et sont également reliés aux objets logiques cibles. Après synchronisation, les objets du niveau physique restent inchangés.

DÉNORMALISATION LOGIQUE

La dénormalisation logique s'applique aux entités (ou classes) et attributs du modèle de données.

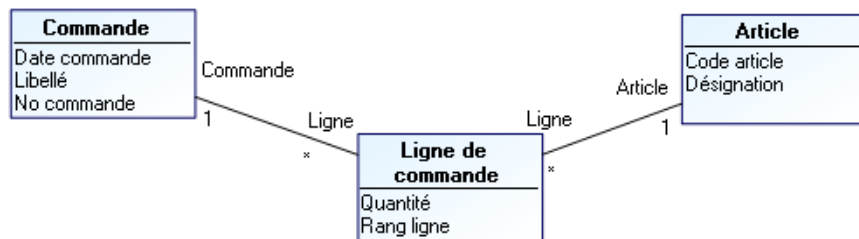
Lancer une dénormalisation logique

Pour dénormaliser le formalisme logique :

1. Faites un clic droit sur la base de données à laquelle est associé le modèle de données et sélectionnez **Dénormaliser**.
Un assistant apparaît.
2. Sélectionnez le type de dénormalisation (logique) et suivez les étapes de l'assistant. Voir [Liste des assistants de dénormalisation logique](#).

Exemple de dénormalisation logique

Supposons que vous vouliez transformer l'entité "Ligne de commande" en une association.



Pour transformer cette entité en association :

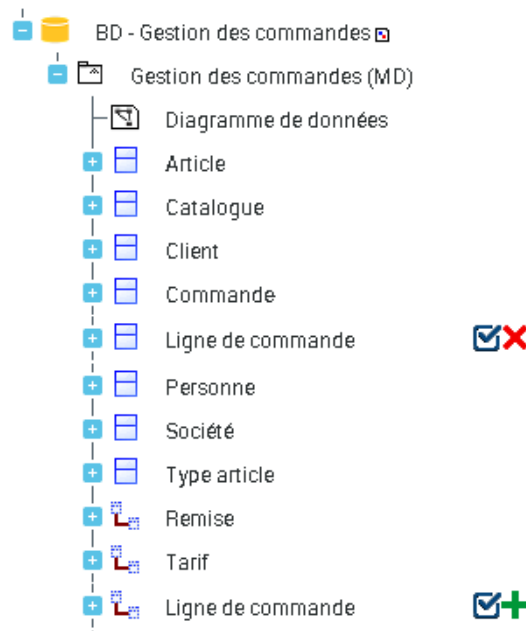
1. Faites un clic droit sur la base de données "Gestion des commandes" qui contient cette entité et sélectionnez **Dénormaliser**.
Un assistant apparaît.
2. Cliquez sur la dénormalisation **Logique**.
3. Cliquez sur **Suivant**.
4. Dans le champ **Sélectionnez le type de dénormalisation**, sélectionnez "Transformer une entité en association".
5. Cliquez sur **Suivant**.

6. Dans l'arbre de l'éditeur, cochez la colonne **Périmètre** face à l'entité "Ligne de commande" afin de la sélectionner.



Vous avez la possibilité de sélectionner plusieurs entités lors d'une dénormalisation.

7. Cliquez sur **Suivant**.
Les options de dénormalisation apparaissent. Par défaut, le report des correspondances et la suppression des objets sources sont activés. Cela signifie que l'entité "Ligne de commande" va être supprimée et que le lien de correspondance avec la table "Ligne de commande" va être reporté sur l'association qui la remplace.
8. Cliquez sur **Suivant**.
L'éditeur affiche les changements occasionnés par cette dénormalisation. Vous pouvez voir que l'entité "Ligne de commande" va être supprimée et que l'association "Ligne de commande" va être créée.



Lorsqu'une entité sélectionnée ne peut être transformée, l'éditeur vous en explique la raison.

Vous pouvez refuser une modification en décochant la case correspondante.

9. Validez les résultats en cliquant sur **Suivant**.

Cette transformation est définitive et sera prise en compte par la prochaine synchronisation.

A la fin de la dénormalisation, vous pouvez voir que l'association "Ligne de commande" qui remplace l'entité est désormais en correspondance avec la table "Ligne de commande".

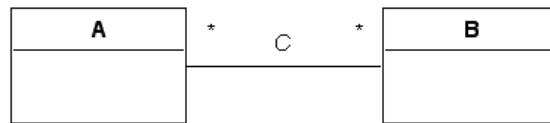
☛ Si un objet est protégé, il n'est pas possible de le sélectionner pendant la dénormalisation.

Liste des assistants de dénormalisation logique

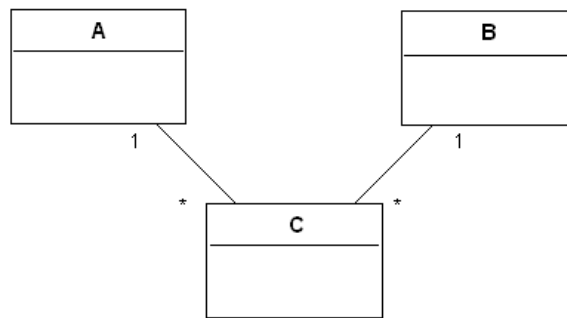
Transformer une association en entité

Cette dénormalisation permet de transformer une association n-aire, quelles que soient ses multiplicités, en entité. Une association de multiplicité '*',1' est créée entre cette entité et les entités de l'association.

Avant



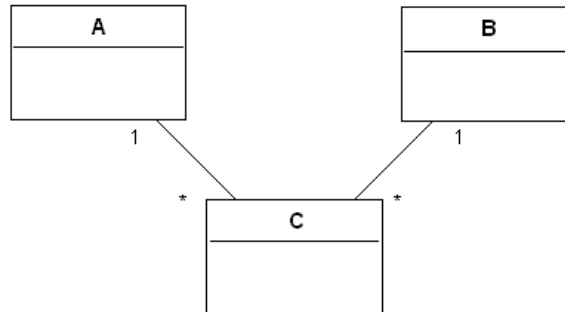
Après



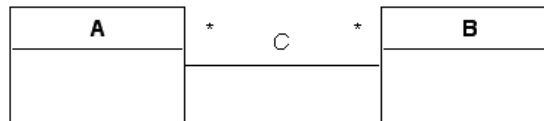
Transformer une entité en association

Cette dénormalisation permet de transformer une entité qui possède n associations binaires dont les rôles opposés sont de multiplicité '1' en association. Une nouvelle association n-aire de multiplicité '*' est créée entre ces entités.

Avant



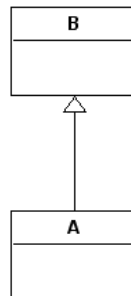
Après



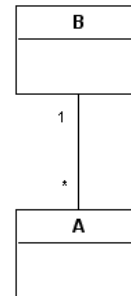
Transformer une généralisation en association

Cette dénormalisation permet de transformer une généralisation entre deux entités en association. Une association de multiplicité *,1 est créée entre les deux entités, et la généralisation est supprimée.

Avant



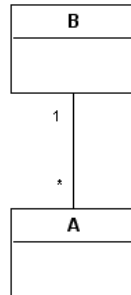
Après



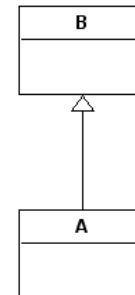
Transformer une association en généralisation

Cette dénormalisation permet de transformer une association 1,* en généralisation. Une généralisation est créée entre les deux entités, et l'association est supprimée.

Avant



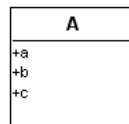
Après



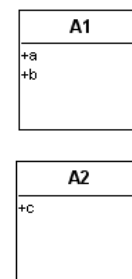
Partition verticale d'une entité

Cette dénormalisation permet de diviser une entité en plusieurs entités. Les entités obtenues se répartissent les attributs, les associations et les généralisations.

Avant



Après



Partition horizontale d'une entité

Cette dénormalisation permet de dupliquer une entité.

Avant

A
+a
+b
+c

Après

A1
+a
+b
+c

A2
+a
+b
+c

Partition horizontale et synchronisation en mode Logique > Physique

Prenons une entité "Catalogue". Après partition horizontale, cette entité donne deux entités "Catalogue-1" et "Catalogue-2".

Après synchronisation du mode Logique > Physique, les deux entités ont pour correspondance une table.

Si vous affichez les propriétés des correspondances, vous remarquez que les deux sont bidirectionnelles, ce qui signifie que entités et table se mettent à jour dans les deux sens de la synchronisation.

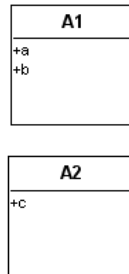
Si vous effectuez des modifications logiques sur ces entités puis relancez la synchronisation, l'éditeur affiche un signal sur la table cible ; en effet il ne sait quelle entité prendre pour effectuer les mises à jour.

Lorsque vous choisissez une entité, celle-ci est conservée comme entité de référence (par exemple Catalogue-1). L'autre entité sera conservée dans un seul sens, autrement dit Catalogue-2 pourra être mise à jour dans le modèle logique mais n'aura pas d'impact sur la table.

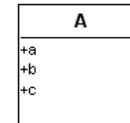
Fusion d'entités

Cette dénormalisation permet de fusionner des entités.

Avant



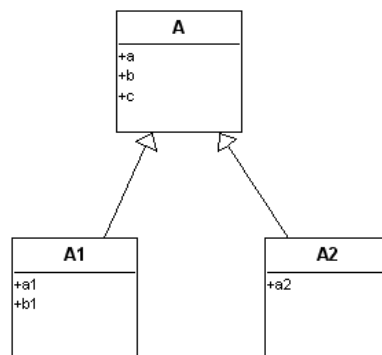
Après



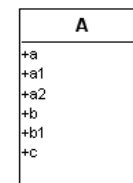
Fusion d'entités ascendantes

Cette dénormalisation permet de fusionner une entité avec son entité mère : tous les attributs et les liens sont reportés sur l'entité parente et l'entité fille est supprimée.

Avant



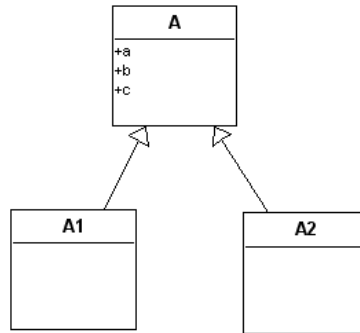
Après



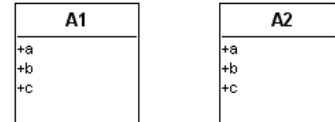
Fusion d'entités descendantes

Cette dénormalisation permet de fusionner une entité avec son entité fille : tous les attributs et les liens sont reportés sur l'entité fille et l'entité parente est supprimée.

Avant



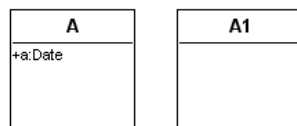
Après



Copier/Coller d'attributs

Cette dénormalisation permet de reporter des attributs d'une entité ou d'une association vers d'autres entités ou d'autres associations.

Avant



Après



DÉNORMALISATION PHYSIQUE

La dénormalisation physique s'applique aux objets de la base de données représentés par les tables, les colonnes, les clés et les index.

Lancer une dénormalisation physique

Pour dénormaliser le formalisme physique :

1. Faites un clic droit sur la base de données concernée et sélectionnez **Dénormaliser**.

Un assistant présente l'ensemble des personnalisations possibles sur les objets de la base de données.

Exemple de dénormalisation physique

Il est possible de faire en sorte qu'une table se divise en deux tables distinctes, soit pour séparer les colonnes dans deux tables (partition verticale), soit pour dupliquer les informations d'une table dans deux autres (partition horizontale).

Prenons l'exemple d'une prise de commande. Cette prise de commande est représentée par une entité au niveau logique, elle donne une table au niveau physique. Supposons que cette prise de commande soit amenée à être gérée différemment au niveau technique selon qu'il s'agisse d'une commande par téléphone ou d'une commande par Internet. On peut intégrer ce changement dans le modèle physique sans être obligé de modifier en parallèle le modèle logique. Pour cela, on va faire en sorte que l'entité représentant la prise d'une commande se divise en deux tables; une table pour les commandes par téléphone, une autre pour les commandes Internet. En intégrant cette modification depuis l'assistant, la partition devient automatique lors de chaque synchronisation, et les deux modèles restent cohérents.

Pour créer une partition horizontale telle que celle décrite ci-dessus :

1. Faites un clic droit sur la base de données "Gestion des commandes" qui contient la table "Commande" et sélectionnez **Dénormaliser**.
Un assistant apparaît.
2. Sélectionnez la dénormalisation **Physique**.
3. Cliquez sur **Suivant**.
4. Dans le champ **Sélectionnez le type de dénormalisation**, sélectionnez «Partition horizontale d'une table».
5. Cliquez sur **Suivant**.
6. Dans l'arbre de l'éditeur, sélectionnez la table que vous voulez dupliquer, ici "Commande".
7. Cliquez sur **Suivant**.
Les options de dénormalisation apparaissent. Par défaut, le report des correspondances et la suppression des objets sources sont activés. Cela signifie que la table "Commande" va être supprimée et que le lien de

correspondance avec l'entité "Commande" va être reporté sur les deux tables.

Précisez le nombre de partitions, autrement dit le nombre de tables créées. Par défaut, l'outil en crée deux.

8. Cliquez sur **Suivant**.

L'éditeur affiche les changements occasionnés par cette dénormalisation. Vous pouvez voir que la table "Commande" va être supprimée et que deux nouvelles tables vont être créées.

	Périmètre	Nom
BD - Gestion des commandes		Groupe HBC::Comm...
Diagramme relationnel		BD - Gestion des co...
Modèle relationnel		BD - Gestion des c...
Article		Groupe HBC::Comm...
Catalogue		Groupe HBC::Comm...
Client		Groupe HBC::Comm...
Commande	☒ ☒	Groupe HBC::Comm...
Commande_1	☒ ☒	Groupe HBC::Comm...
Commande_2	☒ ☒	Groupe HBC::Comm...
Ligne_de_commande		Groupe HBC::Comm...

9. Validez les résultats en cliquant sur **Suivant**.

Cette transformation est définitive et sera prise en compte par la prochaine synchronisation.

A la fin de la dénormalisation, vous pouvez voir que les deux nouvelles tables sont désormais en correspondance avec l'entité "Commande".

La dénormalisation s'applique ici au modèle physique. La synchronisation qui prendra en compte cette personnalisation devra donc se faire dans le même sens, autrement dit du modèle logique vers le modèle physique. Elle n'est pas valable dans l'autre sens.

Liste des assistants de dénormalisation physique

Partition verticale d'une table

Cette dénormalisation permet de diviser une table en plusieurs tables. Les tables obtenues se répartissent les colonnes.

Seules les colonnes ne faisant pas partie d'une clé peuvent être réparties entre les tables.

Avant

A
a
b
c

Après

A1
a
b

A2
c

Partition horizontale d'une table

Cette dénormalisation permet de dupliquer une table. Les deux tables obtenues contiennent l'ensemble des colonnes de la table d'origine.

Avant

A
a
b
c

Après

A1
a
b
c

A2
a
b
c

Fusion de tables

Cette dénormalisation permet de fusionner des tables.

Avant

A1
a
b

A2
c

Après

A
a
b
c

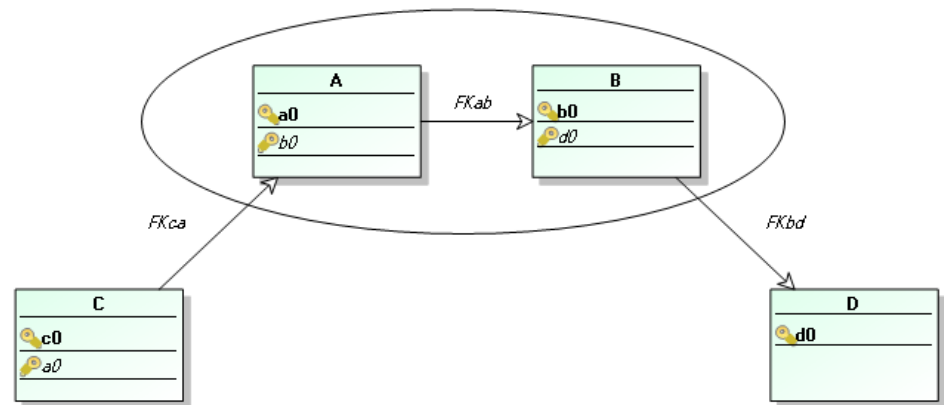
Option des clés primaires

Lorsque vous lancez une fusion de tables, une option vous permet de déterminer la clé primaire de la table fusion : vous pouvez sélectionner une des clés primaires des tables sources ou fusionner l'ensemble des clés primaires des tables sources.

Lorsque vous sélectionnez une clé primaire pour la table fusion, seules les clés étrangères qui font référence à cette clé primaire sont reportées. Les clés étrangères faisant référence aux clés primaires qui ne sont pas reportées ne sont pas prises en compte.

Ainsi, dans l'exemple suivant, si vous fusionnez les tables A et B et conservez la clé primaire de la table A, la clé primaire de B disparaît lors de la fusion. La clé étrangère FKab n'est pas non plus reportée puisqu'elle fait référence à la clé primaire de B. Les

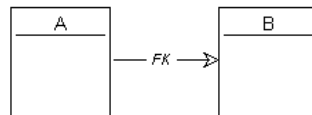
autres clés étrangères, FKca et FKbd, sont bien reportées dans le diagramme relationnel.



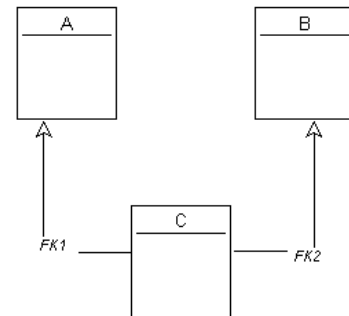
Transformation d'une clé étrangère en table

Cette dénormalisation permet de transformer une clé étrangère en table. Une nouvelle table est créée, ainsi que deux nouvelles clés étrangères. La clé étrangère d'origine est supprimée.

Avant



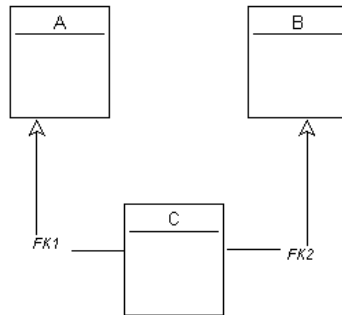
Après



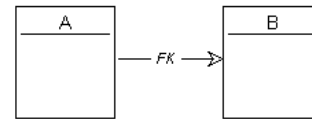
Transformation d'une table en clé étrangère

Cette dénormalisation permet de transformer une table en clé étrangère. La table, ainsi que ses deux clés étrangères sont supprimées, et une nouvelle clé étrangère est créée.

Avant



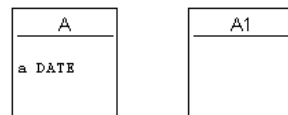
Après



Copier/Coller des colonnes

Cette dénormalisation permet de reporter des colonnes d'une table vers une autre.

Avant



Après





GÉNÉRER DES SCRIPTS SQL



La fonctionnalité de génération SQL produit des fichiers de scripts SQL, qui, à partir des objets logiques de votre référentiel **Hopex** (BD, table, colonne, etc.), vous permettent de créer, modifier ou de mettre à jour les objets correspondants dans le SGBD cible de votre choix.

La génération prend en compte les paramètres hérités du SGBD cible (spécifié pour la base de données), paramètres que vous pouvez personnaliser à un niveau global (voir [Paramétrer la génération d'une base de données](#)), ou à un niveau de détail plus fin, sur une colonne ou sur une clé primaire, par exemple.

Pour les principaux SGBD cibles du marché, l'éditeur de base de données rend accessible une "vue physique" qui vous permet d'optimiser la grammaire SQL des scripts générés, pour y intégrer des options techniques spécifiques au SGBD choisi, tel le partitionnement. Voir [Ajouter des propriétés physiques aux objets d'une base de données](#).

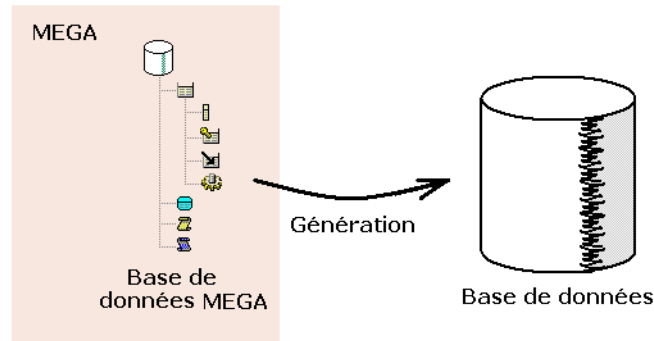
Enfin, le niveau logique peut être complété par la génération d'objets physiques propres à chaque base de données pour un SGBD, comme les vues logiques, les procédures stockées et les triggers.

Les différents modes de génération présentés ci-après autorisent la prise en compte de contraintes liées à l'administration des bases de données sous différents systèmes, avec un maximum de souplesse.

Les points abordés ici sont :

- ✓ [Lancer la génération SQL](#)
- ✓ [Génération incrémentale](#)
- ✓ [Paramétrer la génération SQL](#)
- ✓ [Syntaxe supportée](#)
- ✓ [Définir les vues d'une base de données](#)
- ✓ [Définir les triggers pour une base de données](#)
- ✓ [Utiliser des procédures stockées](#)
- ✓ [Ajouter des propriétés physiques aux objets d'une base de données](#)

LANCER LA GÉNÉRATION SQL



Condition préalable

L'affichage de certaines cibles de génération peut être filtré. Avant de lancer la génération, vérifiez que la cible de génération choisie est bien activée :

1. Dans le bureau, cliquez sur **Menu principal > Paramètres > Options**.
2. Dans la partie gauche de la fenêtre, déployez les dossiers **HOPEX Solutions > Data Management**.
3. Cliquez sur le dossier **Génération SQL**.
Ce dossier contient l'ensemble des générateurs SQL supportés.
4. Dans la partie droite de la fenêtre, cochez ceux que vous voulez afficher dans **Hopex Data Governance** ou **Hopex Data Architecture**.

Pour les cibles de génération, référez-vous aux [SGBD et versions supportées](#).

Objets de la génération SQL

Les objets pris en compte par la génération sont :

- Table
- Colonne
- Clé primaire
- Clé étrangère
- Index
- Groupe de données
- Vue logique
- Vue matérialisée
- Trigger
- Procédure stockée
- Fonction
- Synonyme
- Séquence
- Cluster
- Partition

Les scripts générés par **Hopex** ne gèrent que la structure des objets relationnels, leur contenu n'est pas pris en charge.

Lancer l'assistant de génération

Voir [Condition préalable](#).

Pour lancer une génération SQL dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Outils > Génération de code SQL**.
Un assistant apparaît.
2. Définissez le **Périmètre de génération** :
 - base de données pour une génération complète
 - autre objet SQL pour une génération partielle.
3. Pour une génération complète, sélectionnez la base de données concernée et cliquez sur **Suivant**.
4. Sélectionnez le **Mode de génération**.
Quatre modes de génération sont disponibles :
 - "Création" : génère les ordres de création de l'ensemble des objets.
 - "Suppression" : génère uniquement les ordres de suppression d'objets.
 - "Remplacement" : lance la suppression d'objets, puis les re-crée (afin d'éviter la création de doublons, par exemple). Suppose que le SGBD cible supporte ce mode de génération.
 - "Modification" : prend en compte les modifications uniquement.
Contrairement aux autres modes qui agissent sans tenir compte d'un éventuel existant, ce mode permet de se connecter au serveur de SGBD et d'obtenir un accès en lecture à la base de données déjà créée. L'assistant compare le fichier des données **Hopex** et les

informations de la base de données. Après analyse des deux structures, l'assistant génère les ordres SQL de modification correspondants. Voir [Génération incrémentale](#).

☛ *Ce mode n'est disponible que pour les principaux SGBD. Voir [SGBD et versions supportées](#).*

5. Cliquez sur **Suivant**.
Une fenêtre présente les objets générés.
6. Cliquez sur **Suivant**.
La génération est lancée. Une fenêtre présente l'avancement de l'opération et précise les fichiers contenant le résultat.
Le bouton **Ouvrir** permet de visualiser les fichiers résultats de la génération.

Lorsque l'extension utilisée pour les fichiers générés est reconnue par Windows, il suffit d'effectuer un double-clic sur le nom du fichier pour le consulter avec l'éditeur associé à l'extension.

GÉNÉRATION INCRÉMENTALE

Lorsqu'une base de données a déjà été générée, vous pouvez répercuter par la suite uniquement les changements apportés à la base de données grâce au mode "Modification" de la génération SQL.

La génération incrémentale permet, pour une base de données :

- de consulter dans un rapport .html les différences entre la base de données et sa représentation dans **Hopex**.
- de produire les scripts SQL permettant de mettre à jour la base de données cible à partir de sa description dans **Hopex**.

Objets de la génération incrémentale

Les objets gérés par la génération incrémentale sont les mêmes que ceux de la génération en mode "Création" : table, colonne, clé primaire, clé étrangère, index, groupe de données, vue logique, vue matérialisée, trigger, procédure stockée, fonction, synonyme, séquence, cluster, partition.

Les scripts générés par **Hopex** ne gère que la structure des objets relationnels, leur contenu n'est pas pris en charge. Des options de génération incrémentale permettent d'isoler les ordres SQL qui nécessitent des précautions particulières ou des traitements complémentaires.

Lancer la génération incrémentale

Options de la génération

La génération incrémentale se fait dans un fichier global; elle s'effectue à partir de la base de données et non sur un objet modifié en particulier.

Avant de lancer la génération :

1. Faites un clic droit sur la base de données et sélectionnez **Propriétés**.
La fenêtre de propriétés de la base de données apparaît.
2. Cliquez sur la liste déroulante puis sur **Options > Génération**.
3. Dans le champ **Ventilation Script**, sélectionnez "Un fichier global".
4. Cliquez sur **OK**.

Sous les options, vous devez également indiquer le mode de génération incrémentale, qui autorise ou non la suppression d'objets.

Pour plus de détails sur les options de génération, voir [Paramétrer la génération d'une base de données](#).

Lancer l'assistant de génération

Pour lancer la génération incrémentale :

1. Faites un clic droit sur la base de données et sélectionnez **Générer le code**.
2. Dans le champ **Mode de génération**, sélectionnez "Modification".
La commande "Modification" n'est disponible que pour les principaux SGBD. Voir [SGBD et versions supportées](#).
3. Sélectionnez la **Source de données**. La génération incrémentale peut s'effectuer :
 - A partir d'une connexion ODBC.
 - A partir d'un fichier d'extraction. Voir [Extraire la description de schémas de bases de données à partir de sources de données](#).
4. Cliquez sur **Suivant**.
5. Une fois connecté au SGBD cible, saisissez le nom du propriétaire. Cela vous permet de filtrer les tables à prendre en compte dans la génération.
6. Cliquez sur **Suivant**.

La fenêtre de résultat présente deux fichiers, le fichier .sql ainsi qu'un fichier "Report.htm". Ce dernier fichier est un compte-rendu de génération. C'est un fichier dynamique qui présente le contenu de la base initiale et les modifications apportées.

Source (MEGA)		Cible (Oracle 10)	Action de mise à jour
- TBSTABLE - FORCE LOGGING	↕ ⚠	- TBSTABLE - NO FORCE LOGGING	Aperçu
	✗ ✗	- TBSPARTITION - TBSOVERFLOW	
- DEPARTEMENT_TEST1 - DEPTNO - DOM - NAME - PAYS - REGION - SEGMENT - LOGGING - COMPRESS	↕ ↕	- DEPARTEMENT_TEST1 - DEPTNO - DOM - NAME - PAYS - REGION - SEGMENT - NOLOGGING	Aperçu

Chaque ligne de la liste décrit :

- L'objet **Hopex**. Celui-ci peut être vide si il a été supprimé du référentiel **Hopex**.
- Le symbole de mise à jour du SGBD par rapport à **Hopex**. Les différentes actions possibles sur les objets sont :
 - la création ✖
 - la modification ↕
 - la suppression ✗
 - le remplacement 🔄
- Un symbole d'avertissement ⚠ lorsque la mise à jour d'un objet du SGBD n'est pas complète ou que celle-ci doit être opérée avec prudence.

Lorsque cette icône est présente, un bloc dans le script généré détaille ce qui ne peut être mis à jour.

- L'objet du côté SGBD. Celui-ci peut être vide dans le cadre d'une création du côté **Hopex**.
- Le lien vers le script de mise à jour de l'objet.

A partir de chaque objet, vous pouvez accéder à l'ensemble de ses sous-objets en dépliant l'arborescence correspondante. Vous pouvez également visualiser les paramètres physiques. Dans le cadre d'une modification d'un objet, seuls les paramètres physiques modifiés sont affichés.

Il existe également un compte-rendu de génération (.txt) dans la fenêtre de propriétés de la base générée.

PARAMÉTRER LA GÉNÉRATION SQL

Paramétrer la version de SGBD

SGBD et versions supportées

La liste suivante est donnée à titre indicatif. Elle ne prétend pas être exhaustive et elle est susceptible d'évoluer en fonction de la sortie de nouvelles versions des SGBD.

Produit	Éditeur	Versions supportées
SQL ANSI	ISO 9075	1992
DB2	IBM	OS 390 V5 / OS 390 V7 / OS 390 V8 UDB V5 / UDB V7 / UDB V8
Ingres II	Computer Associates	2.0
Dynamic Server	Informix	7.3
Oracle	Oracle	8 / 9i / 10 / 11
SQL Server	Microsoft	7 / 2000 / 2005 / 2008
Adaptative Server	Sybase	11 / 12.5
Teradata Database	Teradata	14
PostgreSQL	PostgreSQL Global Development Group	9.3

Modifier les propriétés de la version de SGBD

Le paramétrage de la génération est effectué pour le contrôle, entre autres, de la taille des identificateurs générés, le contrôle des caractères autorisés, etc.

Pour paramétrer les options de génération d'une version de SGBD :

1. Recherchez le SGBD en question via l'outil de recherche.
2. Ouvrez les propriétés du SGBD.
3. Dans la fenêtre des propriétés du SGBD, cliquez sur la liste déroulante puis sur **Options > Génération**.
4. Modifiez les paramètres que vous souhaitez changer.

☛ Les paramètres disponibles varient en fonction du SGBD cible. Si le sous-onglet **Génération** n'apparaît pas, vérifiez que l'option de génération liée au SGBD en question est bien cochée.

Paramétrer la génération d'une base de données

Pour paramétrer la génération d'une base de données :

1. Ouvrez la fenêtre de propriétés de la base de données.
2. Cliquez sur la liste déroulante puis sur **Options > Génération**.

De même que pour le paramétrage de la cible, les paramètres proposés varient en fonction du SGBD, et un message explicatif indique l'utilisation de chaque paramètre.

Vous pouvez notamment renseigner les paramètres suivants :

- Les paramètres **Nom Trigger** définissent les noms des trois types de trigger.
- **Base numéro d'erreur** : numéro d'erreur utilisateur pour le SGBD courant.
- **Val. par défaut** : active/désactive la génération des ordres DEFAULT pour les colonnes.
- **Quoted Identifier** : active/désactive la génération des guillemets autour des identifiants SQL (Nom SQL).
- **Qualifier** : permet de préfixer le nom des objets. Voir [Préfixer le nom des objets](#).
- **Mode génération inc** : ce paramètre s'applique à la génération incrémentale et peut prendre les valeurs "Alter" et "Drop/Create".
 - "Alter" n'autorise pas la suppression d'objets (tables, indexes, etc.) au niveau des scripts générés. Seules les instructions pouvant être réalisées au travers de la commande ALTER sont générées. Pour les paramètres physiques, la suppression reste autorisée (c'est le cas notamment pour les partitions).
 - "Drop/Create" autorise la suppression d'objets. Si une mise à jour ne peut s'effectuer au travers de la commande ALTER, l'objet est supprimé puis recréé.
- Par défaut, le paramètre prend la valeur "Alter".
- **Ventilation script** : indique si le résultat de la génération doit être créé dans un fichier unique, ou réparti dans un fichier distinct pour chaque type d'objet ou pour chaque objet.
- **Script SQL** : nom du fichier généré lorsqu'il s'agit d'un fichier global. Par défaut, il s'appelle MEGASQL.SQL. Vous pouvez le personnaliser au niveau du SGBD. La flèche située à l'extrémité du champ vous permet de réinitialiser le paramètre.

 **Vous pouvez également réinitialiser tous les paramètres de l'objet en question. Cette action est à utiliser avec précaution.**

- **Répertoire Script** : répertoire de génération relatif.
- Les différents paramètres **Ext.** permettent de préciser les extensions de chaque fichier généré, pour les tables, les groupements de données, les vues etc.
- **Conversion** : format des fichiers générés (ANSI Windows ou ASCII MS-DOS).
- **CREATE CLUSTER** : active/désactive la génération des ordres CREATE CLUSTER.

- **CREATE TABLE** : active/désactive la génération des ordres CREATE TABLE.
 - **CREATE TABLESPACE** : active/désactive la génération des ordres CREATE TABLESPACE.
 - **PRIMARY KEY** : active/désactive la génération des ordres PRIMARY KEY.
 - **FOREIGN KEY** : active/désactive la génération des ordres FOREIGN KEY.
 - **CREATE INDEX** : active/désactive la génération des ordres CREATE INDEX.
 - **CREATE PROCEDURE** : active/désactive la génération des procédures stockées.
 - **CREATE INDEX PK** : active/désactive la génération des ordres CREATE INDEX pour les index de clés primaires.
 - **CREATE INDEX[UNIQUE]** : active/désactive la génération des ordres CREATE INDEX pour les index uniques.
 - **CREATE VIEW** : active/désactive la génération des vues logiques.
 - **CREATE SEQUENCE** : active/désactive la génération des ordres CREATE SEQUENCE.
 - **CREATE SYNONYM** : active/désactive la génération des ordres CREATE SYNONYM.
 - **CREATE TRIGGER** : active/désactive la génération des triggers.
 - **Commentaires** : active/désactive la génération des commentaires **Hopex** dans le script SQL.
 - **UNIQUE** : active/désactive la génération des ordres UNIQUE.
 - **UNIQUE[PK]** : active/désactive la génération des ordres UNIQUE portant sur les clés primaires.
 - **Syntaxe PRIMARY KEY** : les ordres PRIMARY KEY sont générés dans l'ordre CREATE TABLE ou dans un ordre ALTER TABLE.
 - **Position FOREIGN KEY** : génération des ordres FOREIGN KEY après chaque CREATE TABLE ou groupés en fin de script.
 - **COMMENT ON TABLE** : commentaires sur les tables (0 : pas de commentaire, 1 : une ligne, Total : texte complet).
 - **COMMENT ON COLUMN** : commentaires sur les colonnes (0 : pas de commentaire, 1 : une ligne, Total : texte complet).
- ☛ La génération des commentaires n'est possible que pour les systèmes cibles qui les acceptent (Oracle, DB2,...).
- Les différents paramètres **Complément** activent/désactivent la génération des compléments sur les tables, les groupements de données etc.
 - **Tbpace des tables** : par défaut, les tables sont générées dans le tablespace SYSTEM.
 - **Tbpace des index** : par défaut, les index sont générés dans le tablespace SYSTEM.

Préfixer le nom des objets

Pour la plupart des SGBD, il existe une notion de schéma, qui permet de définir un regroupement logique pour les objets.

Ainsi, à la création d'une table, par exemple, celle-ci peut être automatiquement rangée dans un schéma, et si celui-ci n'est pas renseigné, un schéma par défaut peut lui être automatiquement affecté.

Dans **Hopex** il n'existe pas de notion de schéma mais un concept qui lui est propre - le Qualifier - qui permet de préfixer les noms des objets d'une base de données lors de la génération. Si vous souhaitez par exemple que les objets d'une base aient leur nom préfixé par "MEGA", vous devez saisir cette valeur dans le champ Qualifier des objets en question.

Héritage

Le Qualifier peut être défini au niveau de la base de données et également sur tous les autres types d'objet. Il existe un système d'héritage : si le Qualifier n'est pas spécifié au niveau d'une table, par défaut c'est la valeur saisie sur la base de données qui est prise en compte.

Pour préfixer le nom d'un objet lors de la génération :

1. Ouvrez la fenêtre de propriétés de l'objet en question.
2. Cliquez sur la liste déroulante puis sur **Options > Génération**.
3. Dans le champ **Qualifier**, saisissez la valeur qui viendra préfixer le nom de l'objet.

SGBD concernés

Le Qualifier est disponible pour les SGBD suivants :

- Oracle
- SQL Server
- DB2
- MySQL

SYNTAXE SUPPORTÉE

Instruction CREATE TABLE

L'instruction CREATE TABLE définit une table. La définition inclut :

- Le nom de la table
- Les noms et attributs de ses colonnes
- Les attributs de la table tels que ses clés primaire et étrangères

La syntaxe est la suivante :

```
CREATE TABLE nom-table (nom-coll type-coll [NOT NULL]
...
nom-coln type-coln [NOT NULL])
```

Pour DB2, la syntaxe est la suivante :

```
CREATE TABLE nom-table (nom-coll type-coll [NOT NULL]
...
nom-coln type-coln [NOT NULL])
[in Tablespace <Nom>]
```

Pour Oracle, la syntaxe est la suivante :

```
CREATE TABLE nom-table (nom-coll type-coll [NOT NULL]
...
nom-coln type-coln [NOT NULL])
[Tablespace <Nom>]
```

- **nom-table** : Valeur " SQL " de la table, ou, à défaut, nom de la table ; les caractères non reconnus sont remplacés par "_"
- **nom-col** : Valeur de l'attribut SQL Name de la colonne ou, à défaut, nom de la colonne ; les caractères non reconnus sont remplacés par "_"
- **type-col**
- **NOT NULL** : Voir [Gestion de NOT NULL](#)
- **Tablespace** : DB2 et Oracle : Nom du tablespace cible pour les tables

La clause PRIMARY KEY est précisée dans l'ordre "CREATE TABLE" (voir [Clause PRIMARY KEY](#)).

Gestion de NOT NULL

Les clauses NULL, NOT NULL et NOT NULL WITH DEFAULT sont générées automatiquement sur les colonnes de clés primaires et sur les colonnes issues d'attributs obligatoires lors de la synchronisation.

Ces valeurs peuvent être initialisées à "Null", "Not Null" ou "Not Null with Default" en fonction du paramétrage défini dans la fenêtre de propriétés de la base de

données pour **Colonnes Not Null**, dans le sous-onglet **Synchronisation** de l'onglet **Options**.

Les valeurs ainsi proposées peuvent ensuite être modifiées sur chaque colonne.

Clause PRIMARY KEY

Définition d'une clé primaire

Une ou plusieurs colonnes d'une table permettent d'identifier chaque ligne de cette table. Les valeurs dans ces colonnes doivent être renseignées. Elles constituent la clé primaire ("primary key") de la table.

Une table doit avoir une seule clé primaire ou aucune.

Chaque nom de colonne doit identifier une colonne de la table et la colonne ne doit pas être identifiée plus d'une fois.

Traitements et génération des ordres SQL

Après la déclaration des noms de colonnes de la table, si l'option **PRIMARY KEY** est active, le ou les noms de colonnes de la clé primaire sont déclarés de la manière suivante :

```
PRIMARY KEY (liste des colonnes de la clé primaire)
```

La clause PRIMARY KEY est générée dans l'ordre **CREATE TABLE**.

Exemple 1	Exemple 2
La clé primaire "CP" n'a qu'une colonne, "Col-cp". CREATE TABLE nom-table (Col-cp CHAR(9) NOT NULL, info1 CHAR(7), info2 CHAR(7), PRIMARY KEY (Col-cp)	La clé primaire "CP1" a pour colonnes "CP11" et "CP12". CREATE TABLE nom-table (CP11 CHAR(9) NOT NULL, CP12 CHAR(9) NOT NULL, info1 CHAR(7), info2 CHAR(7), PRIMARY KEY (CP11, CP12)

Pour Oracle, la clause PRIMARY KEY complète est la suivante :

```
CONSTRAINT <nom de la clé> (liste des colonnes de la clé  
primaire)
```

Clause FOREIGN KEY

L'intégrité référentielle peut être exprimée, selon le système cible, soit par des clauses FOREIGN KEY, soit par la génération de triggers. Pour Oracle, elle est exprimée soit par des triggers, soit par des clauses FOREIGN KEY, en fonction du paramétrage de la base de données.

Une ou plusieurs colonnes d'une table peuvent faire référence à une clé primaire dans cette table ou dans une autre. Ces colonnes constituent une clé étrangère. Ces colonnes peuvent ne pas avoir de valeur sur chaque ligne.

La table qui contient la clé primaire référencée est une table parente. La table qui contient la clé étrangère est une table dépendante.

Chaque nom de colonne doit identifier une seule colonne de la table et une même colonne ne doit pas être identifiée plus d'une fois. Si la même liste de noms de colonne est spécifiée dans plus d'une clause FOREIGN KEY, ces clauses ne doivent pas identifier la même table.

Le nom de la table spécifié dans la clause FOREIGN KEY doit identifier une table parente. Une clé étrangère de table dépendante doit avoir le même nombre de colonnes que la clé primaire de la table parente.

Le nombre de clés étrangères n'est pas limité.

Traitement et génération des ordres SQL

Après la déclaration des clés primaires (PRIMARY KEY), le(s) nom(s) de colonnes de la (des) clé(s) étrangère(s) de la table sont déclarés avec l'indication FOREIGN KEY :

```
FOREIGN KEY (liste des colonnes de la clé étrangère)
REFERENCES <Nom table parente> [ON DELETE <Action>] [ON
UPDATE <Action>]
```

ou :

```
ALTER TABLE nomtable [ADD] FOREIGN KEY (liste des colonnes
de la clé étrangère) REFERENCES <Nom table parente> [ON
DELETE <Action>] [ON UPDATE <Action>]
```

Pour Oracle, la syntaxe est la suivante :

```
CONSTRAINT <nom de la clé étrangère> (liste des colonnes de
la clé étrangère) REFERENCES <Nom table parente> [ON DELETE
<Action>] [ON UPDATE <Action>]
```

ou :

```
ALTER TABLE...
```

Exemples

La table "nom-table1" a deux clés étrangères. Ces clés n'ont pas de composante.

```
CREATE TABLE nom-table1
(cp1 CHAR(9) NOT NULL,
cp2-rel12 CHAR(7) NOT NULL,
cp3-rel13 CHAR(7) NOT NULL,
info1 CHAR(7),
info2 CHAR(7),
PRIMARY KEY (cp1))
ALTER TABLE nom-table1 ADD FOREIGN KEY(cp2-rel12)
REFERENCES nom-table2
ALTER TABLE nom-table2 ADD FOREIGN KEY(cp3-rel13)
REFERENCES nom-table3
```

La table "nom-table1" possède une clé étrangère "ce2" qui a deux composantes, "ce21" et "ce22". La clé étrangère "ce2" n'a pas de référence (elle est donc composante de la clé primaire d'une autre table).

```
CREATE TABLE nom-table1
(cp1 CHAR(9) NOT NULL,
ce21 CHAR(7) NOT NULL,
ce22 CHAR(7) NOT NULL,
info1 CHAR(7),
PRIMARY KEY (cp1))
ALTER TABLE nom-table 1 ADD FOREIGN KEY (ce21, ce22)
REFERENCES nom-table 2
```

La table "nom-table1" a une clé étrangère, "ce2". La clé étrangère "ce2" est équivalente à la clé primaire "cp2" qui a deux composantes, "cp21" et "cp22". Les colonnes identifiées par "cp21" et "cp22" sont "NOT NULL".

```
CREATE TABLE nom-table1
(cp1 CHAR(9) NOT NULL,
cp21 CHAR(7) NOT NULL,
cp22 CHAR(7) NOT NULL,
info1 CHAR(7),
info2 (CHAR7),
PRIMARY KEY (cp1))
ALTER TABLE nom-table1 ADD FOREIGN KEY (cp21, cp22)
REFERENCES nom-table2
```

Clause UNIQUE

Une clause UNIQUE est générée pour chaque index unique de la table, sauf si cet index correspond à la clé primaire.

Traitement et génération des ordres SQL

Pour chaque index unique, la clause suivante est générée :

```
UNIQUE (col1,...,coln)
```

(col1,...n,coln) représente les colonnes de l'index.

Instruction CREATE INDEX (Oracle, Sybase, SQL Server)

Définition d'un index

Un index est un ensemble de colonnes d'une table sur lesquelles un accès direct est défini.

Pour Sybase et SQL Server, la valeur de l'attribut type-index de l'index détermine le type de l'index généré : UNIQUE, CLUSTERED ou UNIQUE CLUSTERED.

Traitement et génération des ordres SQL

Pour chaque index, une clause est générée, en fonction du SGBD cible.

Pour Oracle :

```
CREATE INDEX (colonne1,..., colonneN) [TABLESPACE  
(NomTbSpace)
```

(colonne1,..., colonneN) représente les colonnes de l'index ; NomTbSpace est le nom du tablespace pour les index (voir [Paramétrer la génération SQL](#)).

Pour Sybase et SQL Server :

```
CREATE [UNIQUE] [CLUSTERED] INDEX (NomIndex) (NomTable)  
(colonne1,..., colonneN)
```

(colonne1,..., colonneN) représente les colonnes de l'index ; NomTbSpace est le nom du tablespace pour les index (voir [Paramétrer la génération SQL](#)).

Clause CREATE VIEW

Une vue est définie sur une base de données. Elle peut utiliser une ou plusieurs tables.

```
CREATE VIEW nom-vue  
AS  
SELECT  
(nom-colonne, nom-colonne,...)
```

Il est possible de compléter cette définition dans la spécification de la vue (voir [Définir les vues d'une base de données](#)).

DÉFINIR LES VUES D'UNE BASE DE DONNÉES

Une vue physique est une table virtuelle dont la structure et le contenu sont déduits d'une ou plusieurs autres tables par une requête SQL.

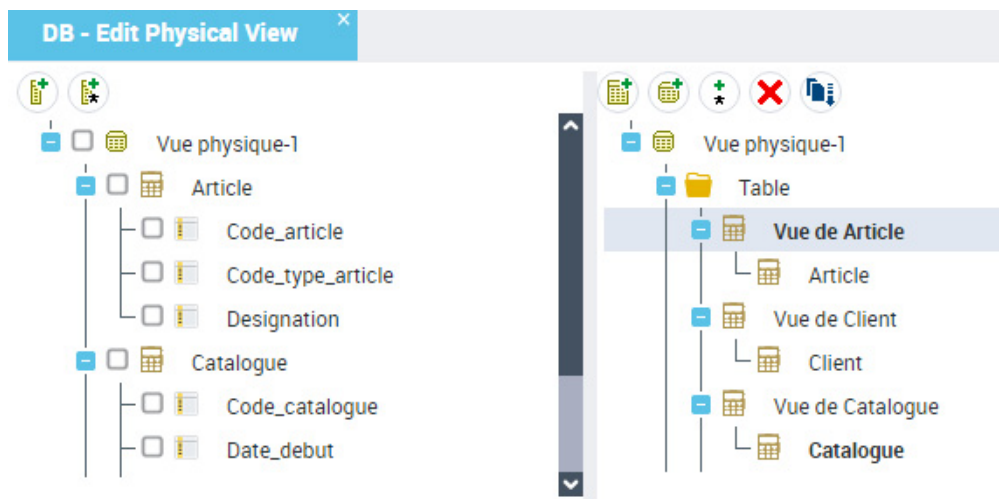
La création des *vues* d'une base de données se fait par l'intermédiaire d'un arbre, qui permet de générer automatiquement une partie de la définition de la vue, et peut être complétée par l'utilisateur.

Créer les vues d'une base de données

Pour créer une vue physique d'une base de données dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Architecture > Base de données**.
2. Dans la zone d'édition, cliquez sur **Hiérarchie**.
3. Dépliez le dossier de la base de données concernée.
4. Faites un clic droit sur le dossier **Vue physique** et sélectionnez **Nouveau > Vue physique**.
L'assistant de création des vues apparaît.
5. Dans le champ **Détenteur**, sélectionnez la base de données concernée.
6. Sous le champ **Table**, cliquez sur **Nouveau**.
7. Sélectionnez les tables sur lesquelles porte la vue.
8. Cliquez sur **OK**.
L'éditeur de la vue physique apparaît.

Le navigateur de gauche affiche les tables de la base de données sur lesquelles porte la vue physique, avec leurs colonnes. L'arbre de droite affiche les tables et les colonnes de vue, qui constituent la vue. Par défaut, celles-ci portent le nom des tables et colonnes sources ; vous pouvez les renommer.



Ajouter une table ou une colonne de vue

Pour ajouter une table à une vue :

1. Dans l'arbre droit de l'éditeur, faites un clic droit sur le dossier **Table** et sélectionnez **Table de vue**.

Sélectionnez la table voulue et cliquez sur **OK**.

Pour ajouter une colonne à une vue :

1. Dans la partie droite de l'éditeur, faites un clic droit sur le dossier **Colonne** et sélectionnez **Colonne de vue**.
2. Sélectionnez la colonne voulue et cliquez sur **OK**.

Définition SQL

La fenêtre d'édition des vues présente, dans le cadre **Définition SQL**, le code de définition de la vue tel qu'il sera généré. Le code est, au départ, calculé en fonction de la définition indiquée dans l'arborescence.

Vous pouvez modifier ce code, notamment à l'aide des jointures. Vous pouvez également saisir directement les modifications dans le cadre SQL.

Jointures de vue

Par défaut, la fenêtre d'édition des vues propose les clés étrangères des tables sélectionnées, lorsqu'elles existent.

Il est ainsi possible de compléter la spécification d'une vue en lui associant des clés étrangères, sources potentielles de jointures.

Pour associer une clé étrangère à la vue :

- 】 Sélectionnez dans arborescence la clé étrangère et glissez-la dans le champ de définition SQL.

Mode utilisateur

Vous pouvez modifier le code de la vue en tapant directement dans le cadre de définition SQL :

- 】 Cliquez sur le bouton **Enregistrer** pour que la **Définition SQL** soit conservée dans le référentiel en tant que telle.

Après modification, il est possible de revenir à la définition telle qu'elle est déduite de l'arborescence :

- 】 Cliquez sur le bouton **Initialiser la définition SQL**  .
Un message vous avertit que la définition préalablement sauvegardée va être réinitialisée. Autrement dit, tous les compléments éventuellement apportés à la définition de la vue sont perdus.
- 】 Cliquez sur **OK** pour confirmer.

Champs

Les catégories de champ correspondent aux types d'objets utilisés dans l'arbre déclaratif : table, vue, colonne et colonne de clé étrangère. Les champs affichés dans la définition SQL correspondent aux éléments déclarés dans l'arbre.

Le type clé étrangère ne donne pas lieu à une catégorie de champ : les champs utilisables proviennent des colonnes de clé et non pas des clés elles-mêmes.

Le bouton **Propriétés du champ**  vous permet d'afficher les propriétés de l'objet correspondant au champ sélectionné.

Si un objet est ajouté dans l'arbre, un champ correspondant devient disponible pour insertion.

Si un objet est renommé dans l'arbre ou dans le référentiel, ses références restent valides et les champs sont affichés avec le nouveau nom dans le texte.

Si un objet est supprimé dans l'arbre ou dans le référentiel, ses références deviennent invalides et sont signalées comme telles dans les champs.

Définir un groupement de données

Un groupement de données - ou tablespace - est un regroupement dans une même page ou des pages physiques voisines de la base de données des lignes de plusieurs tables à des fins d'optimisation, en particulier pour les jointures. Ex : Tablespace dans DB2, Cluster dans Oracle etc.

Pour définir des *groupements de données* dans la base de données :

1. Ouvrez la fenêtre de propriétés de la base de données.
2. Cliquez sur la liste déroulante puis sur **Composants**.
3. La section **Groupements de données** affiche la liste des groupements de données
4. Cliquez sur le bouton **Nouveau**.
5. Dans la fenêtre qui apparaît, indiquez le **Nom** du groupement de données.
6. Cliquez sur **OK**.

Vous allez ensuite ouvrir la fenêtre de propriétés du groupement de données pour définir les tables et les index qu'il inclut.

Pour spécifier les tables et les index inclus dans le groupement de données :

1. Sélectionnez le groupement de données et cliquez sur le bouton **Propriétés**.
La fenêtre des propriétés du groupement de données apparaît.
2. Cliquez sur la liste déroulante puis sur **Tables**.
3. Cliquez sur le bouton **Relier**.
4. Dans la liste des tables de la base de données qui est présentée, sélectionnez les tables de la base de données qu'il devra inclure.

 Le bouton **Délier** permet d'enlever une table de la liste en cas d'erreur.

5. Effectuez les mêmes opérations pour les index du groupement de données.

6. Cliquez sur **OK**.

DÉFINIR LES TRIGGERS POUR UNE BASE DE DONNÉES

Un trigger est un traitement enregistré dans une base de données et qui est déclenché automatiquement lors de la mise à jour d'une table.

Créer un trigger

Les triggers se définissent au niveau des tables d'une base de données.

☛ Il faut noter que les triggers sont définis en fonction du SGBD cible; c'est pourquoi il est important de contrôler, avant de créer les triggers, que le SGBD cible est correct.

Si le SGBD cible est modifié a posteriori, les triggers créés pour ce SGBD ne sont pas supprimés, mais ils sont inactivés.

Pour créer un trigger dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation puis sur **Architecture > Vue hiérarchique**.

☛ Dans **Hopex Data Architecture**, cliquez sur le menu de navigation puis sur **Architecture de données > Données physiques**.

2. Dépliez le dossier de la base de données puis de la table concernées.
3. Faites un clic droit sur le dossier **Trigger** et sélectionnez **Nouveau > Trigger**.

La fenêtre de création d'un trigger apparaît.

4. Indiquez le nom du trigger et les actions déclenchées. Voir [Déclenchement du trigger](#).

Déclenchement du trigger

Le trigger peut se déclencher suite à l'une des trois actions suivantes :

- Lors de la **Création** d'une ligne dans la table
- Lors de la **Suppression** d'une ligne
- Lors d'une **Modification** de la table ou d'une colonne en particulier

De plus, vous pouvez choisir de le lancer **Avant** ou **Après** ces actions, sur l'ensemble de la table ou sur chaque ligne concernée.

Références

Les champs «Référence des anciens tuples» et «Référence des nouveaux tuples» créent dans le code du trigger des références aux lignes insérées, supprimées ou mises à jour.

Le nom indiqué dans le champ "Référence des anciens tuples" correspond à la ligne qui existait avant la mise à jour.

Le nom du champ «Référence des nouveaux tuples» fait référence à la ligne après la mise à jour.

En cas d'insertion, seule la nouvelle ligne est valide.

En cas de suppression, seule l'ancienne est valide.

Définition SQL

L'option **Définition SQL**, dans les propriétés du trigger, présente le code du trigger.

Pour afficher le code du trigger :

1. Faites un clic droit sur le trigger et sélectionnez **Propriétés**.
La fenêtre des propriétés du trigger apparaît.
2. Cliquez sur la liste déroulante puis sur **Textes**.
3. Cliquez sur l'onglet **Définition SQL**.

Intégrité référentielle

L'intégrité référentielle est gérée par la création des clés étrangères sur une base de données.

Elle regroupe l'ensemble des contraintes permettant de contrôler l'incidence d'une modification d'une table dans les tables qui lui sont liées.

Il se peut que l'existence de clés dans certains SGBD n'implique pas un contrôle systématique. Il se peut également que vous vouliez personnaliser les contraintes à appliquer sur une table en particulier.

C'est pourquoi vous pouvez générer dans des triggers le code qui correspond à la gestion de l'intégrité référentielle.

Pour générer l'intégrité référentielle d'une table :

1. Faites un clic droit sur la base de données concernée et sélectionnez **Initialisation de trigger**.
La fenêtre de génération des triggers apparaît.
2. Sélectionnez une des options de génération :
 - Générer la cible par type
 - Générer la cible par intégrité du référentiel
3. Sélectionnez les tables de la base de données.
4. Cliquez sur **Suivant**.
Les triggers sont automatiquement créés pour les tables sélectionnées.
Une fois la génération terminée, les triggers apparaissent sous le dossier **Trigger** disponible sous chaque table. Il existe trois type de trigger :
 - Un trigger de mise à jour (U_ suivi du nom de la table), qui permet de spécifier l'action à réaliser en cas de modification d'une ligne de la table faisant partie de la clé étrangère.
 - Un trigger de suppression (D_), qui spécifie l'action à réaliser en cas de suppression.
 - Un trigger d'insertion (I_), qui spécifie l'action à réaliser en cas d'insertion.

Ces triggers ne sont valables que pour un SGBD cible. Lorsque vous changez de SGBD, vous devez régénérer les triggers.

UTILISER DES PROCÉDURES STOCKÉES

Hopex Data Architecture vous permet de créer des *procédures stockées*.

Une procédure stockée combine un langage procédural et des requêtes SQL au sein d'un programme. Elle permet d'exécuter une tâche particulière sur une base de données. Elle est enregistrée dans une base de données et peut être appelée depuis un programme extérieur à la base de données ou depuis un trigger.

Une procédure stockée peut être implémentée de deux façons; sous la forme d'une procédure ou d'une fonction.

 Une procédure est un ensemble d'instructions exécutant un sous-programme.

 Une fonction est une procédure retournant une valeur à la fin de l'exécution.

Pour créer une procédure stockée sur une base de données :

1. Ouvrez la fenêtre de propriétés de la base de données.
2. Cliquez sur la liste déroulante puis sur **Composants**.
La section **Procédures stockées** affiche la liste des procédures stockées.
3. Cliquez sur le bouton **Nouveau**.
4. Dans la fenêtre qui apparaît, indiquez le nom de la procédure et sa nature (Procédure ou Fonction).
5. Cliquez sur **OK**.
La procédure stockée apparaît. Ouvrez ses propriétés pour définir son code.

Exemple de procédure stockée pour Oracle

Voici l'exemple d'une procédure stockée mettant à jour le prix unitaire d'une pièce en fonction de l'identifiant de la pièce :

```
CREATE PROCEDURE update_part_unitprice (part_id IN INTEGER,
new_price IN NUMBER)
IS
  Invalid_part EXCEPTION;
BEGIN
  -- HERE'S AN UPDATE STATEMENT TO UPDATE A DATABASE RECORD
  UPDATE sales.parts
    SET unit_price = new_price
    WHERE id = part-id;
  -- HERE'S AN ERROR-CHECKING STATEMENT
  If SQL%NOTFOUND THEN
    RAISE invalid_part;
  END IF;
EXCEPTION
  -- HERE'S AN ERROR-HANDLING ROUTINE
  WHEN invalid_part THEN
```

```
        raise_application_error(-20000, 'Invalid Part ID');  
END update_part_unitprice;
```

AJOUTER DES PROPRIÉTÉS PHYSIQUES AUX OBJETS D'UNE BASE DE DONNÉES

Une fois votre base de données définie dans un diagramme relationnel, vous pouvez générer les scripts SQL correspondants à destination de différents SGBD.

Le volet de navigation des données physiques vous permet de compléter la modélisation physique d'une base de données en spécifiant les paramètres propres à chaque SGBD et de produire ainsi des scripts SQL complets.

Vous pouvez également importer dans **Hopex** les paramètres physiques définis sur des objets rétro-générés. Voir la [Rétro-génération des propriétés physiques](#).

Vous pouvez adapter un même modèle logique à plusieurs SGBD. Il n'est pas nécessaire de dupliquer les objets.

SGBD cibles

Pour définir un SGBD cible sur une base de données :

1. Ouvrez la fenêtre de propriétés de la base de données concernée.
2. Cliquez sur la page **Caractéristiques**.
3. Indiquez le champ **SGBD cible** dans le champ correspondant.

Voir aussi [Importer une version de SGBD](#).

Créer des propriétés physiques

Pour créer des propriétés physiques sur les objets d'une base de données dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation puis sur **Actifs de données > Vue hiérarchique**.

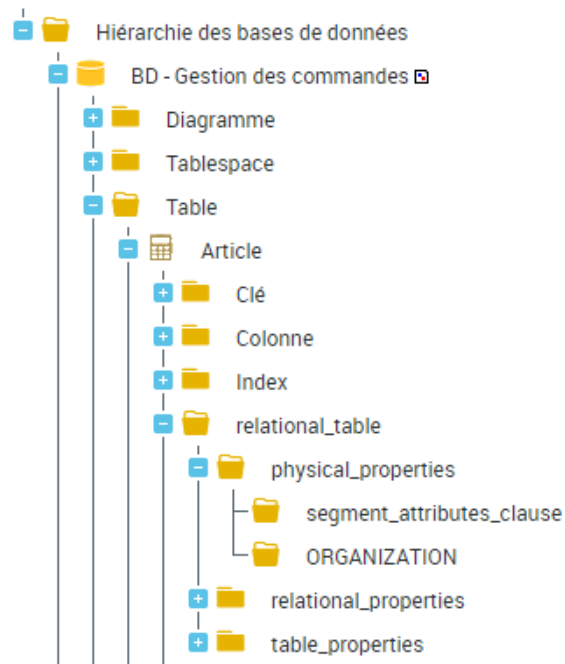
☛ Dans **Hopex Data Architecture**, cliquez sur le menu de navigation puis sur **Architecture de données > Données physiques**.

2. Dans la zone d'édition, déployez le dossier et les sous-dossiers de la base de données concernée.

Les paramètres sont présentés sous forme arborescente, conformément à la grammaire SQL du SGBD considéré (référez-vous à la documentation SQL du SGBD).

Deux types de dossier sont présentés dans l'arbre :

- Les dossiers de navigation.
- Les groupes de paramètres que vous devez instancier.



Chaque groupe de paramètres, représenté par un objet "clause SQL", dispose d'une page de propriétés permettant de définir leur valeur.

Les clauses SQL ainsi définies sont accessibles comme des objets standard du référentiel. Il est par exemple possible de rechercher des objets SQL ayant une valeur donnée pour un paramètre.

Par défaut, les clauses ne sont pas réutilisables d'un objet à l'autre. Il est néanmoins possible de définir une clause pour un objet et de la relier à d'autres objets. Dans ce cas, toute modification de la clause affecte les objets qui l'utilisent.

Objets contenant des paramètres physiques

Tous les objets dans **Hopex** ne supportent pas les paramètres physiques. Ceux-ci concernent uniquement :

- Les groupes de données
- Les tables
- Les index
- Les clusters

Créer une nouvelle clause

Pour définir les paramètres d'un objet :

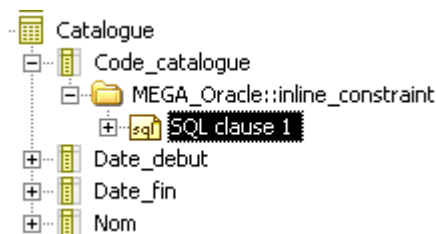
1. Cliquez avec le bouton droit sur le groupe de paramètres correspondant et sélectionnez **Nouveau > SQL clause**.
2. Ouvrez la fenêtre de propriétés de la clause et saisissez la valeur du paramètre à définir.

Relier une clause

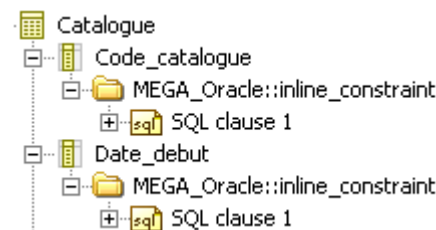
Vous pouvez affecter une même clause à plusieurs objets, à condition de relier le bon type de clause.

Prenons la base de données "Gestion des commandes" ayant pour SGBD Oracle 9i.

Sur la colonne "Code_catalogue", créez la "Clause 1" de type "inline_constraint".



Vous pouvez relier la "Clause 1" à une autre colonne. S'agissant du même type de clause, celle-ci est copiée sans problème sur la nouvelle colonne.



En revanche, si vous reliez la "Clause 1" à un objet dont le type est différent de celui défini initialement sur la "Clause 1" - par exemple "Storage_clause" - alors la "Clause 1" change de type pour prendre celui du dernier élément relié. Autrement dit, la "Clause 1" qui était de type "inline_constraint" prend le type "storage_clause". Ce changement se répercute sur les colonnes de départ auxquelles était reliée la "Clause 1".

Nommage des clauses

Cas standard

Par défaut, le nom de la clause prend le nom de la clause type à laquelle elle est rattachée. Lorsque vous rattachez une clause à un autre type, le nom s'adapte automatiquement.

Nommage spécifique

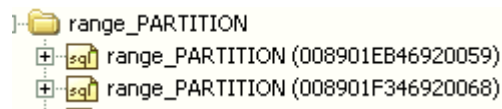
Vous avez la possibilité de donner un nom spécifique à une clause. Dans ce cas, le nom de la clause devient statique; le changement de clause type ne modifie pas le nom de la clause.

☛ Vous pouvez revenir en mode dynamique en forçant le nom à vide.

Un nommage spécifique est indispensable lorsqu'une clause est utilisée dans différents contextes (clause générique).

Clauses multiples

Pour un niveau donné, plusieurs clauses peuvent être rattachées à une même clause type. Afin de distinguer les différentes clauses, le nom de la clause est constitué du nom de la clause type puis de son hexaIdAbs.



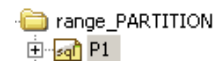
Nommage à partir d'une propriété

Il est possible de modifier le comportement standard en définissant une règle de nommage automatique pour un type de clauses SQL. Ce paramétrage s'effectue au niveau de la propriété `_settings` de la clause type. Dans l'exemple ci-dessous, le paramétrage effectué sur la clause type "range_PARTITION" pour Oracle 9i indique que le nom des clauses SQL de ce type se construit à partir de la valeur de la propriété PARTITION.

```
[NameIdentification]
AttrForNameIdentification=A3F2DE8C417E06C9
```

Une fois ce paramètre effectué, les noms des clauses SQL est automatiquement calculé à partir des valeurs de la propriété PARTITION.

Property	Value
Key compression	
PARTITION	P1
Value list	

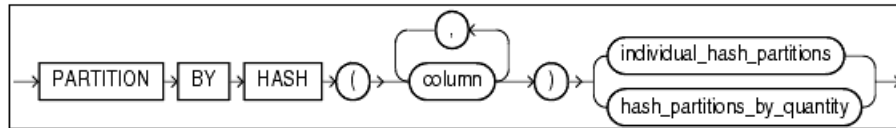


Le nom des clauses SQL n'est pas pris en compte dans la génération SQL. Dans l'exemple fourni, c'est bien la valeur de la propriété PARTITION qui alimente les scripts SQL générés.

Exemple de personnalisation d'un modèle physique

Vous pouvez partitionner une table afin de faciliter l'accès aux données ou de gérer différemment des blocs d'informations.

Supposons que vous vouliez partitionner la table "Ligne de commande" de la base de données "Gestion des commandes" selon la méthode *by hash* d'Oracle. Cette méthode permet de calculer dynamiquement la partition d'une table.



Syntaxe de l'instruction Hash partitioning

Vérifiez que la base de données a pour SGBD cible Oracle 9i.

- 1. Ouvrez sa fenêtre de propriétés et cliquez sur la page **Caractéristiques**. Le nom du SGBD est indiqué dans le champ **SGBD cible**.

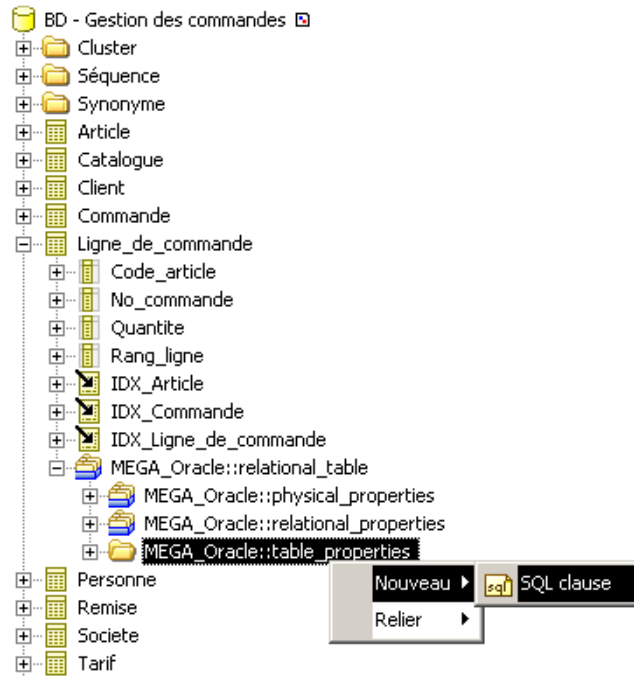
Affichez les propriétés physiques de la base de données "Gestion des commandes" :

1. Cliquez sur le menu de navigation puis sur **Actifs de données > Vue hiérarchique** (pour **Hopex Data Governance**).
 Dans **Hopex Data Architecture**, cliquez sur le menu de navigation puis sur **Architecture de données > Données physiques**.
2. Dans la zone d'édition, déployez le dossier et les sous-dossiers de la base de données "Gestion des commandes" pour afficher les paramètres liés à la grammaire Oracle.

Pour partitionner la table "Ligne de commande" :

1. Déployez la table "Ligne_de_commande".
2. Déployez le groupe de paramètres "MEGA_Oracle::relational_table".

3. Cliquez avec le bouton droit sur le type de clause "MEGA_Oracle::table_properties" et sélectionnez **Nouveau > SQL clause**.



4. Nommez la clause "Ligne_de_commande/Table_properties". Elle apparaît dans l'arbre de navigation.
5. Sous cette clause, déployez le groupe de paramètres "MEGA_Oracle::table_partitioning_clauses". Il contient les différents types de partitionnement qu'il est possible de réaliser dans Oracle.
6. Sur le dossier "MEGA_Oracle::hash_partitioning", créez la clause "Ligne_de_commande/hash_partitioning".
7. Ouvrez sa fenêtre de propriétés.
8. Dans la page **PARTITION BY HASH**, indiquez les colonnes sur lesquelles s'applique le découpage. Pour cela, reliez les colonnes concernées par la partition.
9. Fermez la fenêtre de propriétés.
10. Sous la clause "Ligne_de_commande/hash_partitioning" sont disponibles deux types de clauses :
 - **individual_hash_partitions** : permet de nommer chaque partition.
 - **hash_partitions_by_quantity** : permet de définir le nombre de partitions que vous voulez créer.
11. Créez la clause "Ligne_de_commande/Hash_partition_by_quantity".
12. Ouvrez sa fenêtre de propriétés.
13. Cliquez sur la page **PARTITIONS**.
14. Dans le champ **Hash partition quantity**, indiquez le nombre de partitions. Ces partitions sont représentées par des groupements de données.
15. Sous le champ **STORE IN**, reliez les groupements de données.

Pour obtenir le script correspondant à ce partitionnement, cliquez avec le bouton droit sur la table "Ligne de commande" et sélectionnez **Générer le code**.

```

SPOOL \_MEGASQL.LST;
PROMPT ----- ;
PROMPT Compte-Rendu de génération ;
PROMPT ----- ;
/* -----
/* Begin of generation Oracle 9i for database BD__Gestion_des_commandes the 5 Juin 2006 at 17:38:54 */
/* -----
/* -----
/* Table Ligne_de_commande */
/* -----
CREATE TABLE "Ligne_de_commande"
(
  "Code_article" CHAR(6),
  "No_commande" CHAR(5),
  "Quantite" NUMBER(7),
  "Rang_ligne" NUMBER(7),
  CONSTRAINT "PK_Ligne_de_commande" PRIMARY KEY
  (
    "No_commande",
    "Rang_ligne"
  )
)
PARTITION BY HASH ("No_commande","Rang_ligne")
PARTITIONS 5 STORE IN ("Tbs1","Tbs2","Tbs3","Tbs4") TABLESPACE "SYSTEM";
/* -----
/* Foreign Key FK_Commande */
/* -----
ALTER TABLE "Ligne_de_commande"
ADD CONSTRAINT "FK_Commande" FOREIGN KEY
(
  "No_commande"
) REFERENCES "Commande";
/* -----
/* Foreign Key FK_Article */
/* -----
ALTER TABLE "Ligne_de_commande"
ADD CONSTRAINT "FK_Article" FOREIGN KEY
(
  "Code_article"
) REFERENCES "Article";
SPOOL OFF;
DEFINE _EDITOR = "notepad.exe";
EDIT \_MEGASQL.LST
EXIT;
/* -----
/* End of generation Oracle 9i for database BD__Gestion_des_commandes the 5 Juin 2006 at 17:38:55 */
/* -----

```

Générer le fichier SQL

Lorsque vous avez personnalisé un objet, vous pouvez générer le fichier script correspondant afin d'observer les résultats, sans pour autant re-générer toute la base de données.

Par exemple, pour générer le fichier SQL d'un index :

- Faites un clic droit sur l'index et sélectionnez **Générer le code**.

Voir aussi [Générer des scripts SQL](#).



RÉTRO-GÉNÉRER DES TABLES



La rétro-génération permet, à partir de bases de données existantes, de créer dans le référentiel **Hopex**, les tables et colonnes correspondantes.

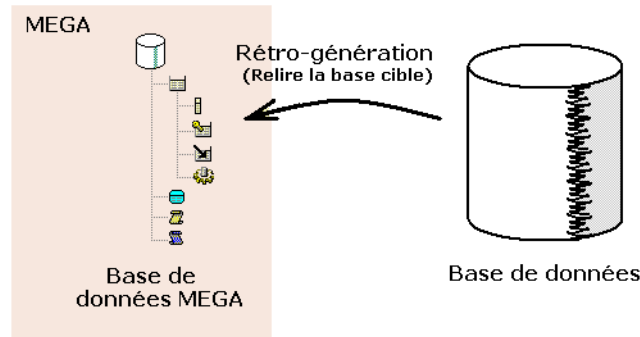
Cette rétro-génération est possible par extraction préalable. Voir [Extraire la description de schémas de bases de données à partir de sources de données](#).

Les tables et colonnes sont intégrées dans une base de données où elles peuvent être facilement maintenues et documentées.

Les points suivants sont abordés ici :

- ✓ [Lancer la rétro-génération](#)
- ✓ [Reconnaissance des Datatypes par ODBC](#)
- ✓ [Rétro-génération des propriétés physiques](#)
- ✓ [Extraire la description de schémas de bases de données à partir de sources de données](#)

LANCER LA RÉTRO-GÉNÉRATION



Pour lancer la rétro-génération :

1. Dans le menu de navigation cliquez sur **Outils > Relire la base cible**. Une fenêtre apparaît.
2. Sélectionnez la base de données source.
3. Cliquez sur **Suivant**.
4. Sélectionnez le **Type** de données à prendre en compte : "Fichier d'extraction" obtenu avec l'utilitaire d'extraction de données (voir [Extraire la description de schémas de bases de données à partir de sources de données](#)).
5. Indiquez l'emplacement du fichier.
6. Précisez les **Options** :
Si la base a déjà été rétro-générée, vous pouvez spécifier l'étendue de la **Réinitialisation**, qui peut concerner uniquement les **Ajouts**, les **Suppressions** ou les **Modifications**.
L'option **Simulation** permet de simuler l'opération, et de générer un compte-rendu indiquant les impacts sur le référentiel.
7. Cliquez sur le bouton **Suivant** pour lancer la rétro-génération. Des messages rendent compte du déroulement de l'exécution.
8. A la fin du traitement, un rapport affiche le détail de l'exécution. A la fin de la rétro-génération, les tables et les colonnes de la base sont créées ; en revanche, le diagramme relationnel doit être créé pour pouvoir consulter la base de données sur un modèle graphique.

☺ Si certains datatypes non standard ont été créés dans le SGBD, il faut les rajouter dans le paramétrage pour qu'ils soient reconnus par **Hopex**. Voir [Reconnaissance des Datatypes par ODBC](#).

RECONNAISSANCE DES DATATYPES PAR ODBC

Lors de la relecture d'une base de donnée, HOPEX demande à la base (Oracle dans notre exemple) quels sont ses datatypes, par l'intermédiaire d'ODBC.

Par exemple, si Oracle contient un datatype "SQL_LONGVARCHAR", HOPEX sait interpréter qu'il s'agit d'un datatype "P-Text".

Pour cela, HOPEX utilise le fichier "xmdb.xdr", situé dans le dossier "Mega\Mega_Std".

La partie du fichier "xmdb.xdr" qui concerne la correspondance des datatypes commence par la ligne suivante :

```
<!-- Mapping of standard Datatypes -->
```

Chaque ligne se présente ainsi :

```
<!-- #DataType(5B845EAE3BAF0020) = "SQL_LONGVARCHAR"  
Standard::Types::Pivot::P-Text -->
```

- Le chiffre entre parenthèses (ici 5B845EAE3BAF0020) est l'identifiant absolu du datatype.
- Le texte entre guillemets (ici SQL_LONGVARCHAR) est le nom du datatype ODBC.
- Le chemin qui suit (ici Standard::Types::Pivot::) précise dans quel paquetage HOPEX se trouve le datatype correspondant.
- Enfin, le datatype HOPEX (ici P-Text) est précisé.

Pour reconnaître les datatypes, l'outil utilise, dans l'ordre, les listes suivantes :

- **La liste des datatypes ajoutés par l'utilisateur.**
Lorsque vous créez manuellement un datatype, il faut ajouter une règle à la liste pour que celui-ci soit reconnu.

- **La liste des datatypes du driver ODBC utilisé.**

Exemple: ici il existe une liste de datatypes pour Ingres et deux listes pour Oracle, pour deux drivers différents (driver Oracle et driver Microsoft)

```
<!-- Declaration Rule Parser Datatypes for Ingres -->
<!-- #DataType(13ED147938B5008F) = "SQL_CHAR,SQL_INGRESC"
Standard::Types::Pivot::P-Character -->
<!-- #DataType(5B845EAE3BAF0020) =
"SQL_VARCHAR,SQL_INGRESTEXT" Standard::Types::Pivot::P-Text
-->
<!-- #DataType(13ED1A3D38B501E0) =
"SQL_DECIMAL,SQL_INGRESMONEY" Standard::Types::Pivot::P-
Currency -->
<!-- #DataType(13ED1A3038B501D5) =
"SQL_TIMESTAMP,SQL_INGRESDATE" Standard::Types::Pivot::P-
Date -->
<!-- Declaration Rule Parser Datatypes for Oracle with
Oracle Driver -->
<!-- #DataType(B55EB70C3471008F) =
"SQL_TIMESTAMP,SQL_Oracle8DATE" Standard::Types::Pivot::P-
Datetime -->
<!-- #DataType(45F44A3B3BAF003D) =
"SQL_VARBINARY,SQL_Oracle8RAW" Standard::Types::Pivot::P-
Binary -->
<!-- #DataType(B55EB71E347100BC) =
"SQL_VARCHAR,SQL_Oracle8ROWID" Standard::Types::Pivot::P-
Timestamp -->
<!-- #DataType(B55EB6F134710044) =
"SQL_DECIMAL,SQL_Oracle8DECIMAL" Standard::Types::Pivot::P-
Numeric -->
<!-- #DataType(B55EB6F134710044) =
"SQL_TINYINT,SQL_Oracle8NUMBER(3,0)"
Standard::Types::Pivot::P-Numeric -->
<!-- #DataType(B55EB6F134710044) =
"SQL_INTEGER,SQL_Oracle8DECIMAL" Standard::Types::Pivot::P-
Numeric -->
<!-- Declaration Rule Parser Datatypes for Oracle with
Microsoft Driver -->
<!-- #DataType(B55EB70C3471008F) =
"SQL_TIMESTAMP,SQL_OracleDATE" Standard::Types::Pivot::P-
Datetime -->
<!-- #DataType(45F44A3B3BAF003D) =
"SQL_VARBINARY,SQL_OracleRAW" Standard::Types::Pivot::P-
Binary -->
<!-- #DataType(B55EB71E347100BC) =
"SQL_CHAR,SQL_OracleROWID" Standard::Types::Pivot::P-
```



```
Timestamp -->
<!-- #DataType(B55EB6F134710044) =
"SQL_DECIMAL,SQL_OracleNUMBER" Standard::Types::Pivot::P-
Numeric -->
```

- **La liste des datatypes standard.**

Problèmes de reconnaissance des datatypes

Il est nécessaire d'ajouter une correspondance entre les datatypes dans le fichier "xmdb.xdr" dans les deux cas suivants :

- **Le datatype est mal reconnu.**
Vous pouvez vous en rendre compte en constatant par exemple qu'un datatype "Integer" dans Oracle devient un datatype "P-Number" dans HOPEX, au lieu de "P-Integer".
- **Le datatype n'est pas reconnu.**
Lorsque le datatype n'est pas reconnu, un message de ce type apparaît en bas de la fenêtre lors de la relecture de la base cible :
"Le DataType 'SQL_Data8' n'a pas été reconnu. Voir l'aide pour configurer la correspondance des Datatypes".

Pour ajouter une correspondance de datatypes :

1. Ouvrez le fichier résultat de l'extraction.
 Voir [Extraire la description de schémas de bases de données à partir de sources de données.](#)
2. Repérez dans ce fichier la correspondance entre les datatypes.
Par exemple, dans l'expression:

```
<COLDATATYPE
Name="COLDATATYPE.JAVA036HTTP036DEPLOYMENT036DIGEST036.IS09
5DOC095ROOT" DataType="SQL_DECIMAL"
```

le datatype ODBC est "SQL_DECIMAL" et le datatype HOPEX est "SQL_Oracle8DECIMAL".
3. Ajoutez à n'importe quel endroit dans le fichier "xmdb.xdr" une ligne construite comme ci-dessus indiquant la correspondance entre les datatypes.

Exemple :

```
<!-- #DataType(B55EB6F134710044) = "SQL_Data8"
Standard::Types::Pivot::P-Text -->
```

Ici, nous avons défini le datatype "SQL_Data8" comme un datatype "P-Text".

RÉTRO-GÉNÉRATION DES PROPRIÉTÉS PHYSIQUES

La rétro-génération permet également de créer dans **Hopex** les propriétés physiques sur les objets d'une base de données.

Les propriétés physiques sont des paramètres permettant d'exprimer, pour un objet relationnel (table, index, etc.), la manière dont les informations vont être stockées au sein d'une base de données. Ces paramètres sont spécifiques à chaque SGBD et peuvent évoluer selon les versions d'un même SGBD.

Voir aussi [Ajouter des propriétés physiques aux objets d'une base de données](#).

Valeurs par défaut

Certaines propriétés du SGBD sont automatiquement rétro-générées dans **Hopex** même lorsqu'elles n'ont pas été explicitement spécifiées.

Afin de ne pas récupérer ces valeurs par défaut, **Hopex** fournit pour chaque SGBD et pour chaque version de SGBD une clause générique qui contient la liste de ces valeurs par défaut et les traite de façon spécifique.

Lors d'une rétro-génération, les propriétés des SGBD dont les valeurs sont égales aux valeurs définies dans la clause générique ne sont pas importées.

Vous pouvez activer la clause générique en important dans **Hopex** le fichier .mol associé à chaque SGBD dans le dossier Mega_Std de votre installation.

Élimination des valeurs redondantes et transverses

Lors de la rétro-génération d'objets dans **Hopex**, certaines propriétés physiques qui n'ont pas été clairement déterminées sont automatiquement restituées par le SGBD au travers d'un mécanisme d'héritage.

Avec Oracle par exemple, la valeur de la propriété PCTFREE dans une table, si elle n'a pas été spécifiée, est directement héritée de celle du tablespace qui lui est rattaché. Une telle valeur est dite transverse car elle est issue d'un héritage entre deux types d'objets distincts. Une valeur est dite redondante si l'héritage est issu d'objets de même type.

Lors de la rétro-génération, **Hopex** ne récupère pas les valeurs transverses et redondantes.

Seule la gestion des valeurs redondantes peut être personnalisée .

Cas spécifiques

Propriétés physiques des tablespaces

Dans certains cas, la rétro-génération de propriétés physiques des objets requiert une connexion ODBC bénéficiant des droits Administrateur du SGBD. Par exemple, avec Oracle, la rétro-génération des propriétés physiques des Tablespaces nécessite que vous utilisiez un compte "System".

Rétro-génération des clusters

La rétro-génération d'un cluster dans Oracle s'effectue correctement si l'utilisateur de connexion vérifie l'une des deux conditions suivantes :

- L'utilisateur est le propriétaire du cluster
- L'utilisateur à un profil administrateur

Lorsque le cluster n'est pas accessible, il n'est pas rétro-généré.

Lorsque l'utilisateur voit le cluster mais n'est ni propriétaire, ni administrateur, le cluster est rétro-généré mais la liaison entre les colonnes du cluster et les colonnes des tables qui lui sont rattachées n'est pas rétro-générée dans **Hopex**.

☛ *D'un point de vue technique, pour un utilisateur non administrateur, la rétro-génération s'appuie sur la vue oracle `sys.all_clu_columns` (relation entre les colonnes de cluster et les colonnes de tables). Cette vue permet de lire uniquement les informations relatives aux objets pour lesquels l'utilisateur est propriétaire.*

EXTRAIRE LA DESCRIPTION DE SCHÉMAS DE BASES DE DONNÉES À PARTIR DE SOURCES DE DONNÉES

HOPEX Data Source Extractor est une application qui utilise les API ODBC pour extraire la définition du schéma d'une base de données. Cette description, obtenue en format structuré, peut être ensuite utilisée pour la rétro-génération dans **Hopex** ou pour la génération en mode modification.

L'outil d'extraction est disponible en version 64 bits.

Il peut être déployé séparément de HOPEX.

Configuration des sources de données requise

Pour utiliser **HOPEX Data Source Extractor**, vous devez disposer de l'outil **ODBC Data Sources Administrator (64 bits)**. Cet outil Microsoft est installé avec Windows et est accessible dans le menu **Démarrer**.

Télécharger HOPEX Data Source Extractor

Pour installer l'outil, assurez-vous de disposer des droits d'installation sur le poste.

HOPEX Data Source Extractor est disponible sur l'HOPEX Store de MEGA. Pour le télécharger :

1. Rendez-vous sur l'HOPEX Store, à l'adresse suivante : <https://store.mega.com/modules>.
2. Sélectionnez le module **HOPEX Data Source Extractor**.
3. Décompressez le fichier sur un poste qui a accès à la base de données concernée.

Lancer l'extraction de données

Pour l'extraction de données, il est nécessaire de définir une source de données ODBC avec l'outil **ODBC Data Source Administrator** :

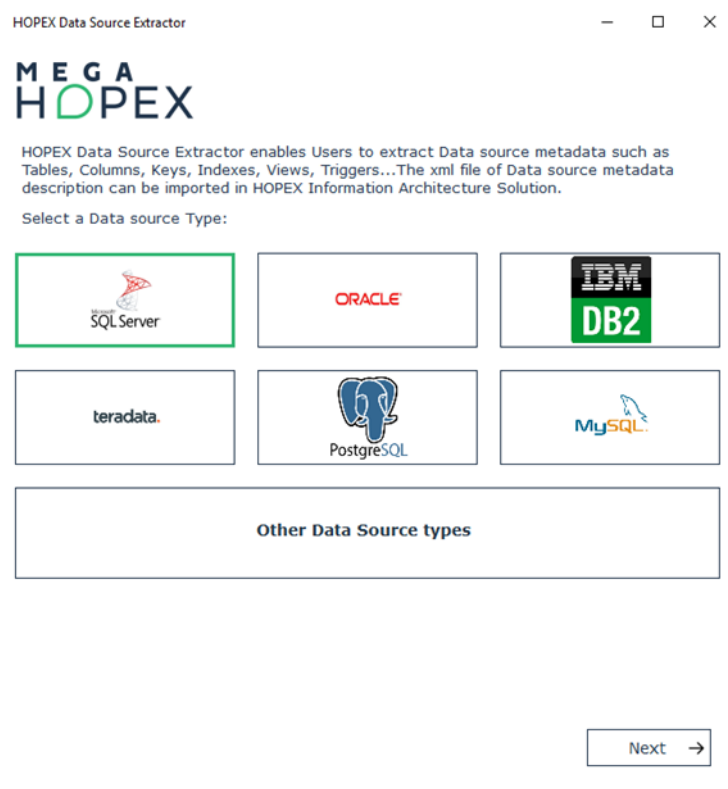
1. Lancez l'outil **ODBC Data Source Administrator**.
2. Cliquez sur l'onglet **Drivers**.
3. Sélectionnez le pilote et cliquez sur **OK**.

➡ Pour plus de détails sur **ODBC Data Source Administrator**, voir la documentation de l'outil dans MSDN : <http://go.microsoft.com/fwlink/?linkid=282669>.

Le pilote de la base de données doit avoir un niveau de conformité ("conformance level") supérieur ou égal à 1. Le champ d'extraction des objets dépend du pilote utilisé dans la définition de la source de données ODBC.

Pour extraire la description d'une base de données :

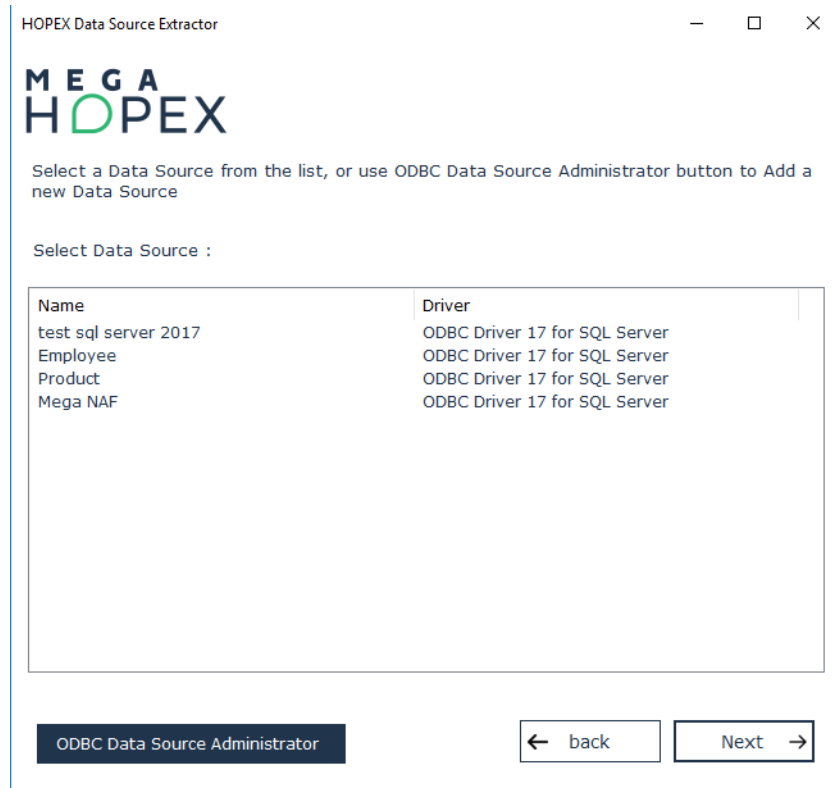
1. Lancez l'utilitaire **HOPEX Data Source Extractor** (dans le répertoire indiqué lors du téléchargement du module).
Un assistant apparaît.
2. Sélectionnez le type de source de données dont vous souhaitez extraire la description du schéma.
Les principaux SGBD supportés sont présentés, vous pouvez afficher les autres types de source de données en cliquant sur **Other Data Source Types**.



3. Cliquez sur **Next**.

La liste des sources de données correspondantes apparaît.

☛ Cette liste est vide si ces connexions ne sont pas définies ou n'ont pu être établies.



4. Sélectionnez une source de données et cliquez sur **Next**.

La fenêtre de connexion s'affiche.

5. S'il ne sont pas déjà définis au niveau de la source de données, précisez un code utilisateur (**User ID**), un mot de passe (**Password**) et un nom de serveur (**Server name**). Si d'autres paramètres sont requis par le pilote ODBC, ils sont demandés lors de la connexion.

HOPEX Data Source Extractor

MEGAHOPEX

SQL Server

Selected Data Source

test sql server 2017

User ID

Password

Server Name

← back

Next →

6. Cliquez sur **Next** pour valider la connexion.
Une fois la connexion établie, vous pouvez choisir les options d'extraction.
- ☛ *Si certaines des options restent grisées, c'est parce que le pilote ne les supporte pas.*
- 😊 *Pour obtenir des informations sur le protocole ODBC utilisé, cliquez sur le bouton **ODBC informations**.*

HOPEX Data Source Extractor

MEGA
HOPEX

Please select options for metadata description and define location of extraction file. The preview helps you to select the scope of objects to extract otherwise all objects of Data source will be extracted by default.

Selected Data Source: Developer
Driver Name: ODBC Driver 17 for SQL Server

Options

- Tables ☒
- Primary Key ☒
- Foreign Key ☒
- Indexes ☒
- Primary Indexes ☒
- [Show more](#)
- Synonyms ☐
- Tablespace ☐
- Physical Views ☐
- Procedures ☐
- Triggers ☐
- Sequences ☐
- Clusters ☐
- Physical Parameters ☐

ODBC Information

Owner: ALL
Location File: Browse
Preview

Name	Comment

back Extract

7. Sélectionnez les éléments à extraire, en plus des tables et des colonnes.
Par défaut, ces éléments sont tous sélectionnés.
Toutes les tables auxquelles l'utilisateur a accès, qu'elles lui appartiennent ou non, sont présentées. Les tables **synonymes** peuvent également apparaître si la case correspondante a été cochée.

📖 *Un synonyme est un nom alternatif donné à un objet (table, vue, procédure stockée, synonyme et séquence). On peut définir un synonyme pour désigner un objet d'une autre base de données.*

Il est possible de filtrer les tables par propriétaire (**Owner**), en le sélectionnant dans la zone correspondante. L'affichage de la liste des propriétaires, ainsi que celle de leurs tables peut prendre plusieurs secondes.

Les éléments suivants sont inclus dans l'extraction :

- Clés primaires (**Primary keys**).
- Clés étrangères (**Foreign keys**).
- Index (**Index**) : il s'agit des index qui ne portent pas sur des clés primaires.
- Index primaires (**Primary index**) : il s'agit des index qui portent sur des clés primaires.

⚠ Les primitives ODBC permettant d'extraire ces éléments ne sont pas supportés par tous les pilotes des sources de données ; un message en rend compte dans le fichier compte-rendu. De plus, certains SGBD ne gèrent pas les concepts correspondants, qui sont alors ignorés.

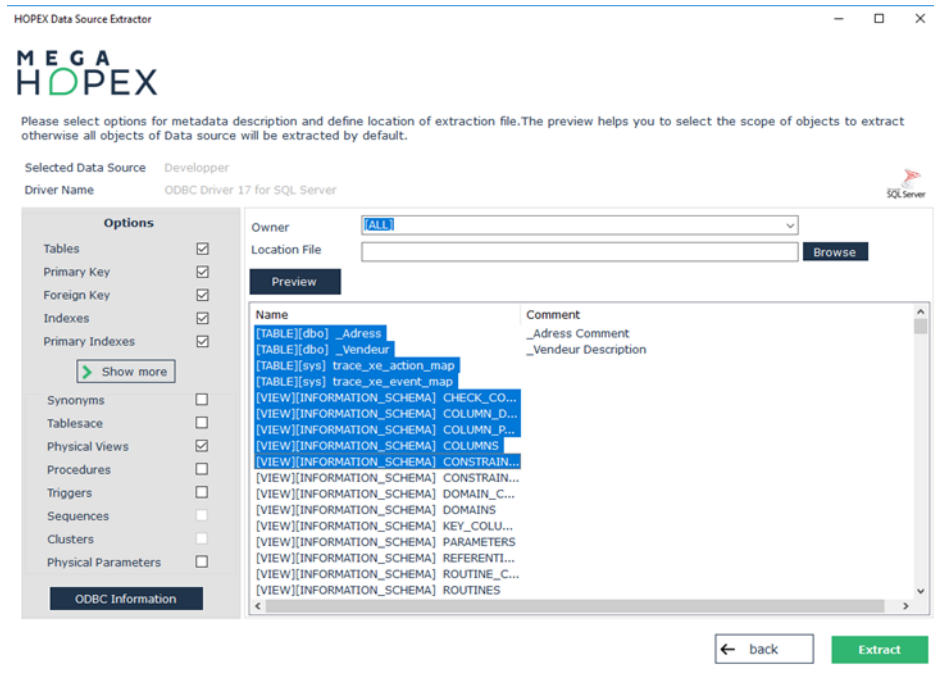
Le champ **Destination file** permet de préciser le chemin et le nom du fichier d'extraction ; le bouton **Browse** permet de parcourir les dossiers.

8. Après sélection des options d'extraction, appuyez sur **Extract** pour démarrer le traitement.

Un message rend compte du nombre de tables extraites. L'activation du bouton **Warnings** permet de consulter simultanément le compte-rendu.

Il est possible de visualiser la liste des tables accessibles en activant le bouton **List Tables**, et d'extraire uniquement les tables sélectionnées par

l'utilisateur dans la liste alors obtenue (toutes les tables sont sélectionnées par défaut).



A la fin de l'extraction, le bouton **Open file** permet de consulter le résultat. Le fichier de compte-rendu est disponible à l'emplacement {AppData d'utilisateur courant}/Local/Mega.

Si l'extraction est incomplète, il est conseillé d'utiliser un autre pilote dont le niveau de conformité est supérieur à 1.

Le fichier résultat peut être exploité par la rétro-génération (voir [Rétro-générer des tables](#)). Il contient la description de la base sous forme d'objets **Hopex**.

Une fois l'opération d'extraction terminée :

- » Cliquez sur **Close** pour vous déconnecter de la source de données.
- » Cliquez sur **Back** pour faire une nouvelle extraction.

Fichier de compte-rendu de l'extraction

Le fichier de compte-rendu de l'extraction des tables par l'utilitaire d'extraction ODBC s'appelle <FIC>_CRD.TXT où <FIC> représente les trois premiers caractères du nom du fichier résultat.

Il contient la liste des tables relues.

Example :

```
=====
Data Source Extracting : DATASOURCE
=====
Table      : OWNER.NOMTABLE1
Table      : OWNER.NOMTABLE2
(suite)
=====
End of extraction
=====
```

Fichier résultat de l'extraction

Le fichier résultat de l'extraction contient la description des tables et des colonnes, résultat de la relecture. Ce fichier porte l'extension ".xml".

Exemple de fichier d'extraction :

```
← → ↺ ↻ 🏠 📁 File C:/Temp/tmp.xml

This XML file does not appear to have any style information associated with it. The document tree is shown below.

<?xml NameId="Xmdb" Version="2.0">
  <DATABASE NameId="Database" DatabaseName="Airport Services">
    <QUALIFIER NameId="QUALIFIER.SQLPrmSpecific032Qualifier" QualifierValue="" />
    <TABLE NameId="TABLE.Account" DBBName="Account" Order="9999" Comment="">
      <TBLCOL NameId="TBLCOL.Account.Account095Number" Order="1" DBBName="Account Number" Comment="" Decimale="NULL" Length="10" NotNull="M" DefaultValue="">
        <COLDATATYPE NameId="COLDATATYPE.Account.Account095Number" DataType="SQL_CHAR" TypeTech="SQL_SQL Server 2008nchar"/>
      </TBLCOL>
      <TBLCOL NameId="TBLCOL.Account.Status" Order="2" DBBName="Status" Comment="" Decimale="NULL" Length="10" NotNull="N" DefaultValue="">
        <COLDATATYPE NameId="COLDATATYPE.Account.Status" DataType="SQL_CHAR" TypeTech="SQL_SQL Server 2008nchar"/>
      </TBLCOL>
      <TBLCOL NameId="TBLCOL.Account.Customer" Order="3" DBBName="Customer" Comment="" Decimale="NULL" Length="10" NotNull="N" DefaultValue="">
        <COLDATATYPE NameId="COLDATATYPE.Account.Customer" DataType="SQL_CHAR" TypeTech="SQL_SQL Server 2008nchar"/>
      </TBLCOL>
      <PK NameId="PK.Account.PK095Account" DBBName="PK_Account" KeyType="P" Order="1">
        <PKCOL NameId="PKCOL.Account.PK095Account.Account095Number" Order="1" Reference="TBLCOL.Account.Account095Number"/>
      </PK>
      <INDEX NameId="INDEX.Account.PK095Account" DBBName="PK_Account" Type="U" Clustered="Y" Sort="NULL">
        <IDXCOL NameId="IDXCOL.Account.PK095Account.Account095Number" Order="1" Sort="A" Reference="TBLCOL.Account.Account095Number"/>
      </INDEX>
    </TABLE>
    >TABLE NameId="TABLE.Customer" DBBName="Customer" Order="9999" Comment="">
      ...
    </TABLE>
    >TABLE NameId="TABLE.Purchase" DBBName="Purchase" Order="9999" Comment="">
      ...
    </TABLE>
  </DATABASE>
</XMDB>
```

Personnaliser l'extraction ODBC

Lorsque l'extraction est incomplète ou ne correspond pas à vos besoins, vous pouvez personnaliser l'extraction avec le fichier Odwdbex.ini. Ce paramétrage dépend du driver ODBC que vous utilisez.

Vous pouvez personnaliser l'extraction de différentes façons, en utilisant :

- les API standard ODBC, disponibles pour les concepts principaux (Table, Colonne, Clé, Index, etc.)
- des requêtes **Hopex**, fournies en remplacement des API standard ODBC
- des requêtes personnalisées.

Par défaut, on utilise les API standard ODBC pour les concepts principaux et des requêtes **Hopex** pour les autres concepts. Pour certains drivers ODBC, on utilise des requêtes **Hopex** pour les concepts principaux, le résultat obtenu par les API standard ODBC étant incomplet.

Utilisation du fichier Odwdbex.ini et des requêtes personnalisées

Pour personnaliser l'extraction :

1. Renseignez-vous auprès de votre administrateur de base de données pour obtenir les requêtes personnalisées correspondant à votre driver ODBC qui permettent de sélectionner les objets (Ex : clés primaires, clés étrangères, séquences, etc).
2. Dans le dossier "All users" de Windows, créez un fichier nommé "Odwdbex.ini" (exemple: C:\Documents and Settings\All Users\ApplicationData\Mega\Odwdbex.ini).
3. Éditez le fichier et ajoutez les requêtes pour les concepts dont vous voulez gérer le comportement. Les concepts qui ne sont pas cités ici restent inchangés.
[<DBMS Name>]
PRIMARY KEYS="Requête personnalisée"
FOREIGN KEYS="Requête personnalisée"
TBLCOLUMNS="Requête personnalisée"
...

La valeur de <DBMS Name> dépend de l'utilitaire ODBC. Pour avoir la valeur appropriée :

1. Lancez l'outil d'extraction **Hopex**.
2. Dans le menu **Data source**, sélectionnez la source de données.
3. Cliquez ensuite sur **System > ODBC Informations**.
4. Lisez le "DBMS Name".

Vous pouvez éditer le fichier Odwdbex.ini en cliquant sur le menu **System > Edit Odwdbex.ini** dans l'outil d'extraction **Hopex**. Assurez-vous que ce fichier est archivé.

Pour plus d'informations sur le format des requêtes, voir [Format des clauses SELECT](#)

Utilisation des API standard ODBC

Pour forcer l'utilisation des API ODBC :

1. Éditez le fichier Odwdbex.ini.

2. Au niveau de chaque concept concerné, modifiez la stratégie d'extraction en utilisant le mot clé : USE_DRIVER_ODBC.

Exemple dans le fichier ODWDBEX.INI :

```
[<Dbms Name>]
```

```
INDEXES=USE_DRIVER_ODBC
```

Format des clauses SELECT

☛ Il est important de respecter la syntaxe indiquée ; en particulier, les "1" ne doivent pas être omis. De plus, il faut noter que les clauses doivent être écrites sur une seule ligne dans le fichier ODWDBEX.INI.

Clés primaires

```
SELECT  
1,  
TABLE_OWNER,  
TABLE_NAME,  
COLUMN_NAME,  
KEY_SEQUENCE,  
PK_NAME  
FROM ...  
WHERE ...
```

- TABLE_OWNER : propriétaire de la table de la clé primaire
- TABLE_NAME : nom de la table de la clé primaire
- COLUMN_NAME : nom de la colonne de la clé primaire
- KEY_SEQUENCE : numéro de la colonne dans la clé (à partir de 1)
- PK_NAME : nom de la clé primaire ; "1" si ce nom n'est pas supporté par le SGBD.

Clés étrangères

```
SELECT
1,
PKTABLE_OWNER,
PKTABLE_NAME,
1,
1,
FKTABLE_OWNER,
FKTABLE_NAME,
FKCOLUMN_NAME,
KEY_SEQ,
UPDATE_RULE,
DELETE_RULE,
FK_NAME
FROM ...
WHERE...
```

- PKTABLE_OWNER : nom du propriétaire de la table de la clé primaire (table de référence)
- PKTABLE_NAME : nom de la table de la clé primaire
- FKTABLE_OWNER : nom du propriétaire de la table de la clé étrangère
- FKTABLE_NAME : nom de la table de la clé étrangère
- FKCOLUMN_NAME : nom de la colonne de la clé étrangère
- KEY_SEQ : numéro de la colonne dans la clé (à partir de 1)
- UPDATE_RULE : "R" : Restrict, "C" : Cascade
- DELETE_RULE : "R" : Restrict, "C" : Cascade
- FK_NAME : nom de la clé étrangère ; "1" si ce nom n'est pas supporté par le SGBD.

Index

```
SELECT
  1,
  TABLE_OWNER,
  TABLE_NAME,
  NON_UNIQUE,
  1,
  INDEX_NAME,
  TYPE,
  SEQ_IN_INDEX,
  COLUMN_NAME,
  COLLATION
FROM ...
WHERE...
```

- TABLE_OWNER : nom du propriétaire de la table de la concernée par la statistique ou par l'index
- TABLE_NAME : nom de la table de l'index
- NON_UNIQUE : les index doivent avoir une valeur unique
- INDEX_NAME : nom de l'index
- TYPE : Type de l'index
- SEQ_IN_INDEX : numéro de la colonne dans la clé (à partir de 1)
- COLUMN_NAME : nom de la colonne
- COLLATION : tri de la colonne ; "A" ascendant, "D" descendant

Colonnes

```
SELECT
1,
COLUMN_OWNER,
TABLE_NAME,
COLUMN_NAME,
DataType ODBC,
DataType Name,
Precision,
Lentgh,
Scale,
1,
NULLABLE,
COMMENT,
DEFAULT_VALUE,
1,
1,
1,
Order
WHERE [Jointure sur <MEGA:OWNER><MEGA:OBJECT_NAME>]
```

- <MEGA:OWNER> est remplacé par l'utilisateur, le Schéma ou "".
- <MEGA:OBJECT_NAME>] est remplacé par le nom de la table.
- COLUMN_OWNER : nom de la colonne, chaîne de 128 caractères.
- TABLE_NAME : nom de la table, chaîne de 128 caractères.
- DataType ODBC : type de données sous la forme d'un nombre entier. Cette valeur est la valeur des types de données ODBC donc voici un rappel :

```
# -1 (SQL_LONGVARCHAR)
# -2 (SQL_BINARY
# -3 (SQL_VARBINARY)
# -4 (SQL_LONGVARBINARY)
# -5 (SQL_BIGINT)
# -6 (SQL_TINYINT)
# -7 (SQL_BIT)
# 0 (SQL_UNKNOWN_TYPE)
# 1 (SQL_CHAR)
# 2 (SQL_NUMERIC)
# 3 (SQL_DECIMAL)
# 4 (SQL_INTEGER)
# 5 (SQL_SMALLINT)
```



```
# 6 (SQL_FLOAT)
# 7 (SQL_REAL)
# 8 (SQL_DOUBLE)
# 9 (SQL_DATE)
# 10 (SQL_TIME)
# 11 (SQL_TIMESTAMP)
# 12 (SQL_VARCHAR)
```

- **DataType Name** : nom du type de données, chaîne de 128 caractères. Il est construit comme ceci : "SQL_<DbmsName><String>"
- **Precision** : longueur dans MEGA si "Length" est vide.
- **Lentgh** : longueur dans MEGA si plus grand que 0.
- **Scale** : nombre entier
- **NULLABLE** : nombre entier spécifiant si la colonne peut être NULL . Valeurs ODBC possible : 0 (SQL_NO_NULLS), 1 (SQL_NULLABLE) ou 3 (SQL_NULL_WITH_DEFAULT).
- **COMMENT** : commentaire de la colonne, chaîne de 1257 caractères.
- **DEFAULT_VALUE** : valeur par défaut de la colonne, chaîne de 1257 caractères.



TABLEAUX DE CORRESPONDANCES ENTRE TYPES PIVOTS ET DATATYPES



Les tableaux suivants donnent les correspondances entre les types pivots et les datatypes des différents SGBD supportés et leurs versions.

- ✓ [DB2 Version 9 For OS](#)
- ✓ [MySQL 5.0](#)
- ✓ [Oracle 11](#)
- ✓ [PostgreSQL9.3](#)
- ✓ [SQL ANSI/ISO 9075:1992](#)
- ✓ [SQL Server 2008](#)
- ✓ [Teradata Database 14](#)

DB2 VERSION 9 FOR OS

Pivot --> Datatype (DB2 Version 9 For OS)

Pivot	Condition	Datatype
P-AutoIdentifier		INTEGER
P-Binary		CHAR(@L) FOR BIT DATA
P-Boolean		CHAR(@L) FOR BIT DATA
P-Byte		CHAR (1) FOR BIT DATA
P-Character	Not Unicode and (L=0 or L ø)	CHAR
	Not Unicode and 0<L<256	CHAR(@L)
	Unicode and (L<256 or L ø)	GRAPHIC(@L)
	Not Unicode and L>255	VARCHAR(@L)
	Unicode and L>255	VARGRAPHIC(@L)
P-Currency		DECIMAL(@L, @D)
P-Date		DATE
P-Datetime		TIMESTAMP
P-Decimal	L=0 or L ø	DECIMAL
	L>0 and D ø	DECIMAL(@L)
	L>0 and D not ø	DECIMAL(@L, @D)
P-Double		DOUBLE
P-Float	L=0 or L ø	FLOAT
	L<>0	FLOAT(@L)
P-Integer		INTEGER

Pivot --> Datatype (DB2 Version 9 For OS)

Pivot	Condition	Datatype
P-Long Integer		INTEGER
P-Long Real		REAL
P-Multimedia		BLOB
		BLOB(@L)
		CLOB
		CLOB FOR MIXED DATA
		CLOB(@L)
		CLOB(@L) FOR MIXED DATA
		LONG VARCHAR FOR BIT DATA
P-Numeric	L not ø and D not ø	DECIMAL(@L, @D)
	L>9 and D ø	FLOAT(@L)
	4<L<10 and D ø	INTEGER
	(L<5 and D ø) or L ø	SMALLINT
P-Real		REAL
P-Smallint		SMALLINT
P-String		LONG VARCHAR
P-Text		LONG VARCHAR
P-Time		TIME
P-Timestamp		TIMESTAMP
P-Tinyint		SMALLINT
P-Varbinary	L <=1024	VARCHAR(@L) FOR BIT DATA
	L>1024	XML

Pivot --> Datatype (DB2 Version 9 For OS)

Pivot	Condition	Datatype
P-Varchar	Not Unicode and (L=0 or L ø)	VARCHAR
	Not Unicode and L<>0	VARCHAR(@L)
	Unicode	VARGRAPHIC(@L)

Datatype --> Pivot (DB2 Version 9 For OS)

Datatype	Condition	Pivot
BLOB		P-Multimedia
BLOB(L)		P-Multimedia
CHAR		P-Character
CHAR (1) FOR BIT DATA		P-Byte
CHAR(L)		P-Character
CHAR(L) FOR BIT DATA		P-Boolean
CLOB		P-Multimedia
CLOB FOR MIXED DATA		P-Multimedia
CLOB(L)		P-Multimedia
CLOB(L) FOR MIXED DATA		P-Multimedia
DATE		P-Date
DECIMAL		P-Decimal
DECIMAL(L)		P-Decimal
DECIMAL(L, D)		P-Decimal
DOUBLE		P-Double
FLOAT		P-Float

Datatype --> Pivot (DB2 Version 9 For OS)

Datatype	Condition	Pivot
FLOAT(L)		P-Float
INTEGER		P-Integer
LONG VARCHAR		P-Text
LONG VARCHAR FOR BIT DATA		P-Multimedia
REAL		P-Real
SMALLINT		P-Smallint
TIME		P-Time
TIMESTAMP		P-Datetime
VARCHAR(L)		P-Varchar
VARCHAR(L) FOR BIT DATA		P-Varbinary
XML		P-Varbinary

MySQL 5.0

Pivot --> Datatype (MySQL 5.0)

Pivot	Condition	Datatype
	L>0 and D not ø	REAL (@L,@D) UNSIGNED
	L=0 or L ø	REAL UNSIGNED
	L>0 and D not ø	REAL (@L,@D) UNSIGNED ZEROFILL
	L=0 or L ø	REAL UNSIGNED ZEROFILL
P-AutoIdentifier		INTEGER
P-Binary	L ø or L<=0	BINARY
	L is numeric and L<>0	BINARY (@L)
P-Boolean		BOOLEAN
P-Byte	L ø or L<=0	BIT
	L is numeric and L<>0	BIT (@L)
P-Character	Not Unicode and L=0	CHAR
	Unicode and L<>0	CHAR (@L) UNICODE
	Unicode and L=0	CHAR UNICODE
	Not Unicode and L<>0	CHAR(@L)
P-Character Ascii	L ø or L<=0	CHAR ASCII
	L is numeric and L<>0	CHAR(@L) ASCII

Pivot --> Datatype (MySQL 5.0)

Pivot	Condition	Datatype
P-Character Binary	Not Unicode and L<>0	CHAR (@L) BINARY
	Unicode and L<>0	CHAR (@L) UNICODE BINARY
	Not Unicode and L=0	CHAR BINARY
	Unicode and L=0	CHAR UNICODE BINARY
P-Character Unicode	L is numeric and L<>0	CHAR (@L) UNICODE
	L ø or L<=0	CHAR UNICODE
P-Character Unicode Binary	L is numeric and L<>0	CHAR (@L) UNICODE BINARY
	L ø or L<=0	CHAR UNICODE BINARY
P-Currency		DECIMAL(@L,@D)
P-Date		DATE
P-Datetime		DATETIME
P-Decimal	L=0 or L ø	DECIMAL
	L>0 and D ø	DECIMAL (@L)
	L>0 and D not ø	DECIMAL(@L,@D)
P-Decimal Unsigned	L>0 and D ø	DECIMAL (@L) UNSIGNED
	L>0 and D not ø	DECIMAL (@L,@D) UNSIGNED
	L=0 or L ø	DECIMAL UNSIGNED

Pivot --> Datatype (MySQL 5.0)

Pivot	Condition	Datatype
P-Decimal Unsigned Zero-fill	L>0 and D $\emptyset$	DECIMAL (@L) UNSIGNED ZEROFILL
	L>0 and D not $\emptyset$	DECIMAL (@L,@D) UNSIGNED ZEROFILL
	L=0 or L $\emptyset$	DECIMAL UNSIGNED ZEROFILL
P-Double	L=0 or L $\emptyset$	DOUBLE PRECISION
	L>0 and D not $\emptyset$	DOUBLE PRECISION (@L,@D)
P-Double Unsigned	L>0 and D not $\emptyset$	DOUBLE PRECISION (@L,@D) UNSIGNED
	L=0 or L $\emptyset$	DOUBLE PRECISION UNSIGNED
P-Double Unsigned Zero-fill	L>0 and D not $\emptyset$	DOUBLE PRECISION (@L,@D) UNSIGNED ZEROFILL
	L=0 or L $\emptyset$	DOUBLE PRECISION UNSIGNED ZEROFILL
P-Float	L=0 or L $\emptyset$	FLOAT
	L>0 and D $\emptyset$	FLOAT (@L)
	L>0 and D not $\emptyset$	FLOAT (@L,@D)
P-Float Unsigned	L>0 and D $\emptyset$	FLOAT (@L) UNSIGNED
	L>0 and D not $\emptyset$	FLOAT (@L,@D) UNSIGNED
	L=0 or L $\emptyset$	FLOAT UNSIGNED

Pivot --> Datatype (MySQL 5.0)

Pivot	Condition	Datatype
P-Float Unsigned Zerofill	$L > 0$ and $D \emptyset$	FLOAT (@L) UNSIGNED ZEROFILL
	$L > 0$ and $D \text{ not } \emptyset$	FLOAT (@L,@D) UNSIGNED ZEROFILL
	$L = 0$ or $L \emptyset$	FLOAT UNSIGNED ZEROFILL
P-Integer	$L \emptyset$ or $L \leq 0$	INTEGER
	L is numeric and $L \neq 0$	INTEGER (@L)
P-Integer Unsigned	L is numeric and $L \neq 0$	INTEGER (@L) UNSIGNED
	$L \emptyset$ or $L \leq 0$	INTEGER UNSIGNED
P-Integer Unsigned Zerofill	L is numeric and $L \neq 0$	INTEGER (@L) UNSIGNED ZEROFILL
	$L \emptyset$ or $L \leq 0$	INTEGER UNSIGNED ZEROFILL
P-Long Integer	$L \emptyset$ or $L \leq 0$	BIGINT
	L is numeric and $L \neq 0$	BIGINT (@L)
P-Long Integer Unsigned	L is numeric and $L \neq 0$	BIGINT (@L) UNSIGNED
	$L \emptyset$ or $L \leq 0$	BIGINT UNSIGNED
P-Long Integer Unsigned Zerofill	L is numeric and $L \neq 0$	BIGINT (@L) UNSIGNED ZEROFILL
	$L \emptyset$ or $L \leq 0$	BIGINT UNSIGNED ZEROFILL
P-Longblob		LONGBLOB
P-Longtext		LONGTEXT
P-Mediumblob		MEDIUMBLOB

Pivot --> Datatype (MySQL 5.0)

Pivot	Condition	Datatype
P-Mediumint	L $\emptyset$ or L $\leq$ 0	MEDIUMINT
	L is numeric and L $\geq$ 0	MEDIUMINT (@L)
P-Mediumint Unsigned	L is numeric and L $\geq$ 0	MEDIUMINT (@L) UNSIGNED
	L $\emptyset$ or L $\leq$ 0	MEDIUMINT UNSIGNED
P-Mediumint Unsigned Zerofill	L is numeric and L $\geq$ 0	MEDIUMINT (@L) UNSIGNED ZEROFILL
	L $\emptyset$ or L $\leq$ 0	MEDIUMINT UNSIGNED ZEROFILL
P-Mediumtext		MEDIUMTEXT
P-Multimedia		BLOB
P-National Varchar	L $\emptyset$ or L $\leq$ 0	NATIONAL VARCHAR
	L is numeric and L $\geq$ 0	NATIONAL VARCHAR (@L)
P-National Varchar Binary	L is numeric and L $\geq$ 0	NATIONAL VARCHAR (@L) BINARY
	L $\emptyset$ or L $\leq$ 0	NATIONAL VARCHAR BINARY
P-Numeric	L=0 or L $\emptyset$	NUMERIC
	L $\geq$ 0 and D $\emptyset$	NUMERIC (@L)
	L $\geq$ 0 and D not $\emptyset$	NUMERIC (@L,@D)
P-Real	L=0 or L $\emptyset$	REAL
	L $\geq$ 0 and D not $\emptyset$	REAL (@L,@D)
P-Smallint	L $\emptyset$ or L $\leq$ 0	SMALLINT
	L is numeric and L $\geq$ 0	SMALLINT (@L)

Pivot --> Datatype (MySQL 5.0)

Pivot	Condition	Datatype
P-Smallint Unsigned	L is numeric and $L > 0$	SMALLINT (@L) UNSIGNED
	$L \leq 0$	SMALLINT UNSIGNED
P-Smallint Unsigned Zero-fill	L is numeric and $L > 0$	SMALLINT (@L) UNSIGNED ZEROFILL
	$L \leq 0$	SMALLINT UNSIGNED ZEROFILL
P-String		VARCHAR(@L)
P-Text		TEXT
P-Time		TIME
P-Timestamp		TIMESTAMP
P-Tinyblob		TINYBLOB
P-Tinyint	$L \leq 0$	TINYINT
	L is numeric and $L > 0$	TINYINT (@L)
P-Tinyint Unsigned	L is numeric and $L > 0$	TINYINT (@L) UNSIGNED
	$L \leq 0$	TINYINT UNSIGNED
P-Tinyint Unsigned Zero-fill	L is numeric and $L > 0$	TINYINT (@L) UNSIGNED ZEROFILL
	$L \leq 0$	TINYINT UNSIGNED ZEROFILL
P-Tinytext		TINYTEXT
P-Varbinary	$L \leq 0$	VARBINARY
	L is numeric and $L > 0$	VARBINARY (@L)
P-Varchar	$L \leq 0$	VARCHAR
	L is numeric and $L = 0$	VARCHAR(@L)

Pivot --> Datatype (MySQL 5.0)

Pivot	Condition	Datatype
P-Varchar Binary	L is numeric and $L > 0$	VARCHAR (@L) BINARY
	$L \leq 0$ or $L < 0$	VARCHAR BINARY
P-Wide Character	$L \leq 0$ or $L < 0$	NATIONAL CHAR
	L is numeric and $L > 0$	NATIONAL CHAR (@L)
P-Wide Character Binary	L is numeric and $L > 0$	NATIONAL CHAR (@L) BINARY
	$L \leq 0$ or $L < 0$	NATIONAL CHAR BINARY
P-Year	$L \leq 0$ or $L < 0$	YEAR
	L is numeric and $L > 0$	YEAR(@L)

Datatype --> Pivot (MySQL 5.0)

Datatype	Condition	Pivot
BIGINT		P-Long Integer
BIGINT (L)		P-Long Integer
BIGINT (L) UNSIGNED		P-Long Integer Unsigned
BIGINT (L) UNSIGNED ZEROFILL		P-Long Integer Unsigned Zerofill
BIGINT UNSIGNED		P-Long Integer Unsigned
BIGINT UNSIGNED ZEROFILL		P-Long Integer Unsigned Zerofill
BINARY		P-Binary
BINARY (L)		P-Binary
BIT		P-Byte
BIT (L)		P-Byte

Datatype --> Pivot (MySQL 5.0)

Datatype	Condition	Pivot
BLOB		P-Multimedia
BOOLEAN		P-Boolean
CHAR		P-Character
CHAR (L) BINARY		P-Character Binary
CHAR (L) UNICODE		P-Character Unicode
CHAR (L) UNICODE BINARY		P-Character Unicode Binary
CHAR ASCII		P-Character Ascii
CHAR BINARY		P-Character Binary
CHAR UNICODE		P-Character Unicode
CHAR UNICODE BINARY		P-Character Unicode Binary
CHAR(L)		P-Character
CHAR(L) ASCII		P-Character Ascii
DATE		P-Date
DATETIME		P-Datetime
DECIMAL		P-Decimal
DECIMAL (L)		P-Decimal
DECIMAL (L) UNSIGNED		P-Decimal Unsigned
DECIMAL (L) UNSIGNED ZEROFILL		P-Decimal Unsigned Zero-fill
DECIMAL (L,D) UNSIGNED		P-Decimal Unsigned
DECIMAL (L,D) UNSIGNED ZEROFILL		P-Decimal Unsigned Zero-fill

Datatype --> Pivot (MySQL 5.0)

Datatype	Condition	Pivot
DECIMAL UNSIGNED		P-Decimal Unsigned
DECIMAL UNSIGNED ZEROFILL		P-Decimal Unsigned Zero-fill
DECIMAL(L,D)		P-Decimal
DOUBLE PRECISION		P-Double
DOUBLE PRECISION (L,D)		P-Double
DOUBLE PRECISION (L,D) UNSIGNED		P-Double Unsigned
DOUBLE PRECISION (L,D) UNSIGNED ZEROFILL		P-Double Unsigned Zero-fill
DOUBLE PRECISION UNSIGNED		P-Double Unsigned
DOUBLE PRECISION UNSIGNED ZEROFILL		P-Double Unsigned Zero-fill
FLOAT		P-Float
FLOAT (L)		P-Float
FLOAT (L) UNSIGNED		P-Float Unsigned
FLOAT (L) UNSIGNED ZEROFILL		P-Float Unsigned Zerofill
FLOAT (L,D)		P-Float
FLOAT (L,D) UNSIGNED		P-Float Unsigned
FLOAT (L,D) UNSIGNED ZEROFILL		P-Float Unsigned Zerofill
FLOAT UNSIGNED		P-Float Unsigned
FLOAT UNSIGNED ZEROFILL		P-Float Unsigned Zerofill

Datatype --> Pivot (MySQL 5.0)

Datatype	Condition	Pivot
INTEGER		P-Integer
INTEGER (L)		P-Integer
INTEGER (L) UNSIGNED		P-Integer Unsigned
INTEGER (L) UNSIGNED ZEROFILL		P-Integer Unsigned Zerofill
INTEGER UNSIGNED		P-Integer Unsigned
INTEGER UNSIGNED ZEROFILL		P-Integer Unsigned Zerofill
LOBLOB		P-Longblob
LONGTEXT		P-Longtext
MEDIUMBLOB		P-Mediumblob
MEDIUMINT		P-Mediumint
MEDIUMINT (L)		P-Mediumint
MEDIUMINT (L) UNSIGNED		P-Mediumint Unsigned
MEDIUMINT (L) UNSIGNED ZEROFILL		P-Mediumint Unsigned Zerofill
MEDIUMINT UNSIGNED		P-Mediumint Unsigned
MEDIUMINT UNSIGNED ZEROFILL		P-Mediumint Unsigned Zerofill
MEDIUMTEXT		P-Mediumtext
NATIONAL CHAR		P-Wide Character
NATIONAL CHAR (L)		P-Wide Character
NATIONAL CHAR (L) BINARY		P-Wide Character Binary

Datatype --> Pivot (MySQL 5.0)

Datatype	Condition	Pivot
NATIONAL CHAR BINARY		P-Wide Character Binary
NATIONAL VARCHAR		P-National Varchar
NATIONAL VARCHAR (L)		P-National Varchar
NATIONAL VARCHAR (L) BINARY		P-National Varchar Binary
NATIONAL VARCHAR BINARY		P-National Varchar Binary
NUMERIC		P-Numeric
NUMERIC (L)		P-Numeric
NUMERIC (L,D)		P-Numeric
REAL		P-Real
REAL (L,D)		P-Real
REAL (L,D) UNSIGNED		
REAL (L,D) UNSIGNED ZEROFILL		
REAL UNSIGNED		
REAL UNSIGNED ZEROFILL		
SMALLINT		P-Smallint
SMALLINT (L)		P-Smallint
SMALLINT (L) UNSIGNED		P-Smallint Unsigned
SMALLINT (L) UNSIGNED ZEROFILL		P-Smallint Unsigned Zero-fill
SMALLINT UNSIGNED		P-Smallint Unsigned

Datatype --> Pivot (MySQL 5.0)

Datatype	Condition	Pivot
SMALLINT UNSIGNED ZEROFILL		P-Smallint Unsigned Zero-fill
TEXT		P-Text
TIME		P-Time
TIMESTAMP		P-Timestamp
TINYBLOB		P-Tinyblob
TINYINT		P-Tinyint
TINYINT (L)		P-Tinyint
TINYINT (L) UNSIGNED		P-Tinyint Unsigned
TINYINT (L) UNSIGNED ZEROFILL		P-Tinyint Unsigned Zero-fill
TINYINT UNSIGNED		P-Tinyint Unsigned
TINYINT UNSIGNED ZEROFILL		P-Tinyint Unsigned Zero-fill
TINYTEXT		P-Tinytext
VARBINARY		P-Varbinary
VARBINARY (L)		P-Varbinary
VARCHAR		P-Varchar
VARCHAR (L) BINARY		P-Varchar Binary
VARCHAR BINARY		P-Varchar Binary
VARCHAR(L)		P-Varchar
YEAR		P-Year
YEAR(L)		P-Year

ORACLE 11

Pivot --> Datatype (Oracle 11)

Pivot	Condition	Datatype
P-AutoIdentifier		NUMBER
P-Binary		RAW(@L)
P-Boolean	L<2 or L ø	RAW(1)
	L>1	RAW(@L)
P-Byte		RAW(1)
P-Character	Not Unicode and (L<2001 or L ø)	CHAR(@L)
	L>4000	LONG
	Unicode and (L<2001 or L ø)	NCHAR(@L)
	Unicode and 2000<L<4001	NVARCHAR2(@L)
	Not Unicode and 2000<L<4001	VARCHAR2(@L)
P-Currency		NUMBER(@L,@D)
P-Date		DATE
P-Datetime		DATE
P-Decimal		NUMBER(@L,@D)
P-Double		NUMBER(@L,@D)
P-Float	0<L<127	FLOAT(@L)
	L=0 or L>126 or L ø	FLOAT
P-Integer		NUMBER(@L)

Pivot --> Datatype (Oracle 11)

Pivot	Condition	Datatype
P-Long Integer		NUMBER(@L)
P-Long Real		NUMBER(@L,@D)
P-Numeric	L=0 or L ø	NUMBER
	L>0 and D ø	NUMBER(@L)
	L>0 and D not ø	NUMBER(@L,@D)
P-Real		NUMBER(@L,@D)
P-Smallint		NUMBER(@L)
P-String		LONG
P-Text	Unicode	NVARCHAR2(@L)
	Not Unicode	VARCHAR2(@L)
P-Time		DATE
P-Timestamp	L<10	TIMESTAMP(@L)
	L>9 or L ø	TIMESTAMP
P-Tinyint		NUMBER(@L)
P-Varchar	L>4000 or L=0 or L ø	LONG
	Unicode and 0<L<4001	NVARCHAR2(@L)
	Not Unicode and 0<L<4001	VARCHAR2(@L)

Datatype --> Pivot (Oracle 11)

Datatype	Condition	Pivot
CHAR(L)		P-Character
DATE		P-Date
FLOAT		P-Float

Datatype --> Pivot (Oracle 11)

Datatype	Condition	Pivot
FLOAT(L)		P-Float
LONG		P-String
NUMBER		P-Numeric
NUMBER(L)		P-Numeric
NUMBER(L,D)		P-Numeric
RAW(1)		P-Boolean
RAW(L)		P-Boolean
TIMESTAMP		P-Timestamp
TIMESTAMP(L)		P-Timestamp
VARCHAR2(L)		P-Varchar

POSTGRESQL9.3

Pivot --> Datatype (PostgreSQL9.3)

Pivot	Condition	Datatype
P-Boolean		boolean
P-Byte	L=0 or L ø	bit
	Valid	bit(@L)
P-Character	L=0 or L ø	char
	Valid	char(@L)
P-Currency		money
P-Date		date
P-Decimal	L=0 or L ø	decimal
	L>=1 and D ø	decimal(@L)
	L>=1	decimal(@L,@D)
P-Double		double precision
P-Integer		integer
P-Long Integer		bigint
P-Numeric	L=0 or L ø	numeric
	L>=1 and D ø	numeric(@L)
	L>=1	numeric(@L,@D)
P-Real		real
P-Smallint		smallint
P-Text		text
P-Time	L=0 or L ø	time
	Valid	time(@L)

Pivot --> Datatype (PostgreSQL9.3)

Pivot	Condition	Datatype
P-Timestamp	L=0 or L ø	timestamp
	L <> 0	timestamp(@L)
P-Varchar	L=0 or L ø	varchar
	Valid	varchar(@L)

Datatype --> Pivot (PostgreSQL9.3)

Datatype	Condition	Pivot
bigint		P-Long Integer
bit		P-Byte
bit(L)		P-Byte
boolean		P-Boolean
char		P-Character
char(L)		P-Character
date		P-Date
decimal		P-Decimal
decimal(L)		P-Decimal
decimal(L,D)		P-Decimal
double precision		P-Double
integer		P-Integer
money		P-Currency
numeric		P-Numeric
numeric(L)		P-Numeric
numeric(L,D)		P-Numeric
real		P-Real

Datatype --> Pivot (PostgreSQL9.3)

Datatype	Condition	Pivot
smallint		P-Smallint
text		P-Text
time		P-Time
time(L)		P-Time
timestamp		P-Timestamp
timestamp(L)		P-Timestamp
varchar		P-Varchar
varchar(L)		P-Varchar

SQL ANSI/ISO 9075:1992

Pivot --> Datatype (SQL ANSI/ISO 9075:1992)

Pivot	Condition	Datatype
P-AutoIdentifier		INTEGER
P-Binary		BIT VARYING(@L)
P-Boolean		BIT(@L)
P-Byte		BIT(@L)
P-Character		CHAR(@L)
P-Currency		DECIMAL(@L,@D)
P-Date		DATE
P-Datetime		DATETIME
P-Decimal		DECIMAL(@L,@D)
P-Double		DOUBLE PRECISION
P-Float		FLOAT
P-Integer		INTEGER
P-Long Integer		INTEGER
P-Long Real		REAL
P-Multimedia		BIT VARYING(@L)
P-Numeric	L>4	INTEGER
	L<5 or L ø	SMALLINT
P-Real		REAL
P-Smallint		SMALLINT
P-String		VARCHAR(@L)
P-Text		VARCHAR(@L)

Pivot --> Datatype (SQL ANSI/ISO 9075:1992)

Pivot	Condition	Datatype
P-Time		TIME
P-Timestamp		DATETIME
P-Tinyint		SMALLINT
P-Varbinary		BIT VARYING(@L)
P-Varchar		VARCHAR(@L)

Datatype --> Pivot (SQL ANSI/ISO 9075:1992)

Datatype	Condition	Pivot
BIT VARYING(L)		P-Multimedia
BIT(L)		P-Boolean
CHAR(L)		P-Character
DATE		P-Date
DATETIME		P-Datetime
DECIMAL(L,D)		P-Currency
DOUBLE PRECISION		P-Double
FLOAT		P-Float
INTEGER		P-Integer
REAL		P-Real
SMALLINT		P-Smallint
TIME		P-Time
VARCHAR(L)		P-Varchar

SQL SERVER 2008

Pivot --> Datatype (SQL Server 2008)

Pivot	Condition	Datatype
P-AutoIdentifier		uniqueidentifier
P-Binary		binary(@L)
P-Boolean		bit
P-Byte		bit
P-Character	Not Unicode and (L<8001 or L ø)	char(@L)
	Unicode and (L<8001 or L ø)	nchar(@L)
	Unicode and L>8000	ntext
	Not Unicode and L>8000	text
P-Currency	L not empty or L > 10	money
	L empty or L < 11	smallmoney
P-Date		smalldatetime
P-Datetime		datetime
P-Decimal		decimal(@L,@D)
P-Double		numeric(@L,@D)
P-Float	L empty	float
	L not empty	float(@L)
P-Integer		int
P-Long Integer		bigint
P-Long Real		real

Pivot --> Datatype (SQL Server 2008)

Pivot	Condition	Datatype
P-Multimedia		image
P-Numeric		numeric(@L,@D)
P-Real		real
P-Smallint		smallint
P-String	Unicode	ntext
	Not Unicode	text
P-Text	Unicode	ntext
	Not Unicode	text
P-Time		time
P-Timestamp		timestamp
P-Tinyint		tinyint
P-Varbinary		varbinary(@L)
P-Varchar	Unicode and L>8000	ntext
	Unicode and (L<8001 or L ø)	nvarchar(@L)
	Not Unicode and L>8000	text
	Not Unicode and (L<8001 or L ø)	varchar(@L)
P-Wide Character		nchar(@L)
P-Wide String		nvarchar(@L)

Datatype --> Pivot (SQL Server 2008)

Datatype	Condition	Pivot
bigint		P-Long Integer
binary(L)		P-Binary

Datatype --> Pivot (SQL Server 2008)

Datatype	Condition	Pivot
bit		P-Boolean
char(L)		P-Character
datetime		P-Datetime
decimal(L,D)		P-Decimal
float		P-Float
float(L)		P-Float
image		P-Multimedia
int		P-Integer
money		P-Currency
nchar(L)		P-Wide Character
numeric(L,D)		P-Numeric
nvarchar(L)		P-Wide String
real		P-Real
smalldatetime		P-Date
smallint		P-Smallint
smallmoney		P-Currency
text		P-Text
time		P-Time
timestamp		P-Timestamp
tinyint		P-Tinyint
uniqueidentifier		P-AutoIdentifier
varbinary(L)		P-Varbinary
varchar(L)		P-Varchar

TERADATA DATABASE 14

Pivot --> Datatype (Teradata Database 14)

Pivot	Condition	Datatype
P-AutoIdentifier		NUMBER
P-Boolean		BYTEINT
P-Byte	L=0 or L ø	BYTE
	L>0	BYTE(@L)
P-Character	L=0 or L ø	CHAR
	L>0	CHAR(@L)
P-Date		DATE
P-Datetime		DATE
P-Decimal	L=0 or L ø and D=0 Or D ø	DECIMAL
	L>0 and D=0 Or D ø	DECIMAL(@L)
P-Double		NUMBER(@L,@D)
P-Float		FLOAT
P-Integer		INTEGER
P-Long Integer		BIGINT
P-Long Real		FLOAT
P-Multimedia	L=0 or L ø	BLOB
	L>0	BLOB(@L)
P-Numeric	L=0 or L ø and D=0 Or D ø	NUMBER
	L ø and D > 0	NUMBER(*,@D)
	L>0 and D=0 Or D ø	NUMBER(@L)
	L>0 and D>0	NUMBER(@L,@D)

Pivot --> Datatype (Teradata Database 14)

Pivot	Condition	Datatype
P-Real		FLOAT
P-Smallint		SMALLINT
P-String		VARCHAR(@L)
P-Text		VARCHAR(@L)
P-Time		TIME
	D > 0	TIME(@D)
P-Timestamp		TIMESTAMP
	D > 0	TIMESTAMP(@D)
P-Tinyint		SMALLINT
P-Varbinary		VARBYTE(@L)
P-Varchar		VARCHAR(@L)

Datatype --> Pivot (Teradata Database 14)

Datatype	Condition	Pivot
BIGINT		P-Long Integer
BLOB		P-Multimedia
BLOB(L)		P-Multimedia
BYTE		P-Byte
BYTE(L)		P-Byte
BYTEINT		P-Boolean
CHAR		P-Character
CHAR(L)		P-Character
DATE		P-Date
DECIMAL		P-Decimal

Datatype --> Pivot (Teradata Database 14)

Datatype	Condition	Pivot
DECIMAL(L)		P-Decimal
FLOAT		P-Real
INTEGER		P-Integer
NUMBER		P-Numeric
NUMBER(*,D)		P-Numeric
NUMBER(L)		P-Numeric
NUMBER(L,D)		P-Numeric
SMALLINT		P-Smallint
TIME		P-Time
TIME(D)		P-Time
TIMESTAMP		P-Timestamp
TIMESTAMP(D)		P-Timestamp
VARBYTE(L)		P-Varbinary
VARCHAR(L)		P-Varchar





Conformité

RÉGLEMENTATIONS



Des réglementations incitent les organisations à appliquer une politique de gouvernance des données afin d'être conformes aux attentes des autorités, des partenaires et des clients.

L'administrateur fonctionnel peut définir dans **Hopex Data Governance** les textes de référence et les politiques d'entreprise auxquelles doit se conformer l'organisation.

La solution **Hopex Data Governance** offre les outils pour définir les textes qui contraignent l'organisation. Elle met notamment à disposition les contenus des réglementations BCBS 239 et Solvabilité II, que vous pouvez importer dans le référentiel. Une fois les réglementations définies vous pouvez évaluer le niveau de conformité de vos données à ces directives.

FAIRE L'INVENTAIRE DES TEXTES DE RÉFÉRENCE

(RÉGLEMENTATIONS EXTERNES)

Un texte de référence est un texte qui entre dans l'une des catégories suivantes : réglementations (textes de lois qui peuvent entraîner des pénalités s'ils ne sont pas respectés), lignes directrices, standards et normes, bonnes pratiques.

Hopex Data Governance fournit par défaut la réglementation bancaire BCBS 239 ainsi que de la réglementation Solvabilité II, sous forme de modules. Vous pouvez définir d'autres cadres réglementaires dans votre référentiel.

Importer le module des textes de référence

Les bibliothèques des réglementations à importer sont livrées sous forme de modules, téléchargeables dans la console HAS de HOPEX.

Pour télécharger le module d'une réglementation :

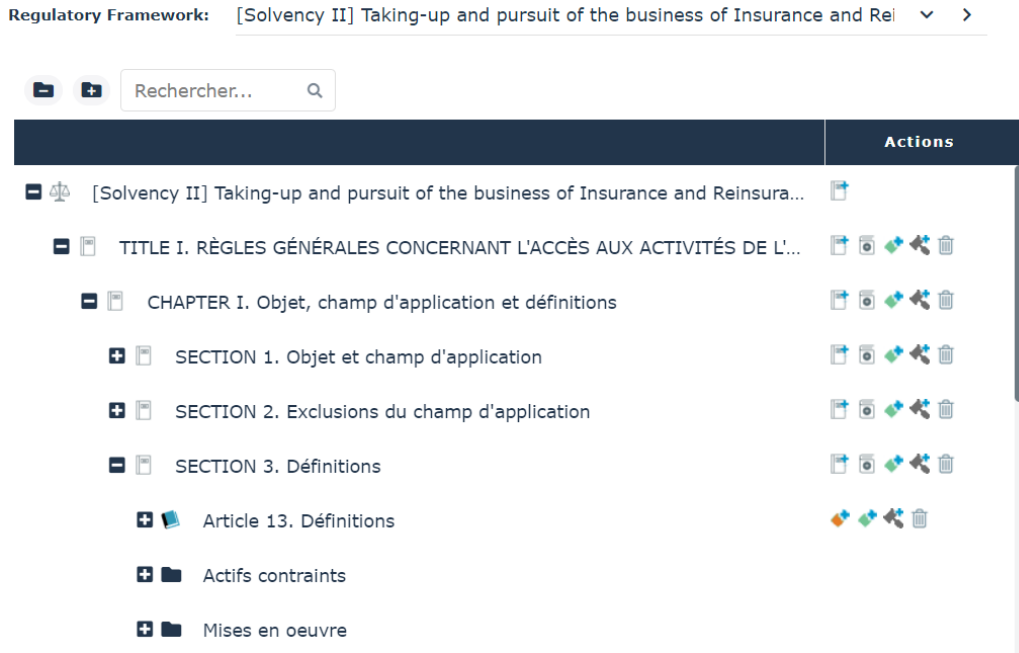
- A partir de votre version HOPEX, ouvrez la console d'administration HAS Console et téléchargez le module. Voir [Importer un module dans HOPEX](#).

Structure d'un texte de référence

Un texte de référence définit le contenu d'une réglementation. Il comprend un ensemble de sections, et sous chaque section des sous-sections et articles.

Vous pouvez définir les objets de votre organisation contraints par la réglementation ainsi que les règles métier qui assurent la mise en œuvre de la réglementation au sein de l'entreprise.

➡ Pour plus de détails sur les règles métier, voir [Assurer la conformité des données : créer des règles métier](#).



Créer un texte de référence

Pour définir un texte de référence dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Conformité > Textes de référence.**
2. Dans la zone d'édition, cliquez sur l'onglet **Textes de référence.**
3. Cliquez sur **Nouveau.**
4. Dans la fenêtre qui apparaît, indiquez :
 - le nom du texte de référence
 - sa description
5. Cliquez sur **OK.**
Le texte de référence apparaît dans la liste.

Définir les objets de votre organisation contraints par un texte de référence

Vous pouvez définir les objets qu'une réglementation contraint dans les propriétés de l'objet en question. Par exemple, pour définir la réglementation associée à un concept, voir [Réglementations](#).

Voir aussi : Analyser les informations contraintes par une réglementation.

Voir les objets contraints par un texte de référence

Pour voir les objets contraints par un texte de référence :

1. Passez le curseur de la souris sur le texte de référence.
Des boutons apparaissent à sa droite.
2. Cliquez sur **Propriétés**.



3. Dans les propriétés du texte, cliquez sur la page **Éléments sujets**.

Voir les catégories d'objet contraintes par un texte de référence

Pour voir les catégories d'objet contraintes par un texte de référence :

1. Sélectionnez le texte de référence.
Des boutons apparaissent à sa droite.
2. Cliquez sur **Propriétés**.
3. Dans les propriétés du texte, cliquez sur la page **Catégories de données contraintes**.

Ajouter une section au texte de référence

Pour ajouter une section au texte de référence :

1. Cliquez sur l'icône du texte de référence puis sur **Nouveau > Section de texte de référence**.
La fenêtre de création d'une section apparaît.
2. Indiquez :
 - (optionnel) le code de la section
 - l'intitulé
 - 👉 Le nom de la section est calculé automatiquement à partir du code et de l'intitulé.
 - la description, qui correspond au texte de la section.
 - 👉 Lorsque la section se compose d'autres sections et articles, les textes peuvent apparaître sous ces derniers.
 - (optionnel) les éléments sujets : il s'agit des objets contraints par la section. Il peut s'agir de données métier, physique, d'applications, etc.
 - (optionnel) la mise en œuvre : il s'agit des règles métier qui assurent l'implémentation effective de la réglementation au sein de l'entreprise.
 - 👉 Voir [Assurer la conformité des données : créer des règles métier](#).

3. Cliquez sur **OK**.
La section apparaît dans l'arbre de navigation.

Définir les objets contraints par une section de texte de référence

Pour définir les objets contraints par une section de texte de référence :

1. Sélectionnez la section du texte de référence.
Des boutons apparaissent à sa droite.
2. Cliquez sur **Propriétés**.
3. Dans les propriétés de la section, cliquez sur la page **Éléments sujets**.

Ajouter un article à une section

Pour ajouter un article à une section :

1. Cliquez sur l'icône de la section puis sur **Nouveau > Article de texte de référence**.
La fenêtre de création d'un article apparaît. Vous pouvez définir les informations suivantes :
 - le code de l'article
 - l'intitulé

 *Le nom de l'article est calculé automatiquement à partir du code et de l'intitulé.*

 - la description, qui correspond au texte de l'article.
 - (optionnel) les définitions de concepts qui peuvent être associées à des mots clés du texte.
 - (optionnel) les éléments sujets : il s'agit des objets contraints par l'article. Il peut s'agir de données métier, physique, d'applications, etc.
 - (optionnel) les mises en œuvre : il s'agit des règles métier qui assurent l'implémentation effective de la réglementation au sein de l'entreprise.

 *Voir [Assurer la conformité des données : créer des règles métier](#).*
2. Cliquez sur **OK**.
L'article apparaît dans l'arbre de navigation, sous la section.

Ajouter une définition à un article

Vous pouvez relier à l'article des définitions de concepts qui existent dans votre référentiel et qui favorisent la compréhension de l'article.

Pour créer la définition d'un article :

1. Dans l'arbre du texte de référence, passez le curseur de la souris sur l'article.
Des boutons apparaissent à sa droite.
2. Cliquez sur **Propriétés**.
3. Dans la fenêtre des propriétés, cliquez sur la page **Caractéristiques**.

4. Sous la section **Définitions**, cliquez sur **Nouveau**. La fenêtre de création d'une définition apparaît.
5. Sélectionnez le **Type d'objet** qui porte la définition, par exemple, un concept, et sélectionnez dans la liste **Nom court** l'objet en question.
6. Cliquez sur **OK**.

Voir aussi :

A travers un catalogue de données vous pouvez indiquer quels sections et articles d'un texte de référence s'appliquent à une organisation. Voir [Réglementations associées au catalogue](#).

Faire l'inventaire des obligations

Les obligations constituent une interprétation de la loi et contribuent à la mise en application de tout article auquel votre organisation doit se conformer.

Dans **Hopex Data Governance** vous pouvez visualiser les obligations de différentes façons :

- au sein des cadres de contrôle qui les contiennent
- dans une arborescence qui recense la liste des obligations
- au sein des textes de référence dans lesquels elles apparaissent
- dans les documents UCF. Voir [Importer des documents d'UCF](#).

Des contrôles de mise en œuvre peuvent être associés aux obligations. Voir : [Définir des contrôles de qualité des données](#).

Accéder à la liste des obligations

Pour accéder aux obligations contenues dans votre référentiel :

1. Cliquez sur le menu de navigation **Conformité > Textes de référence**.
2. Dans la zone d'édition, cliquez sur l'onglet **Obligations**.

 *Les obligations connexes de premier niveau apparaissent dans cette liste seulement si l'un de ses enfants revêt un caractère légal.*

Des colonnes vous indiquent :

- si l'obligation sélectionnée contraint l'organisation
- le nombre de contrôles de mis en oeuvre associés

Obligations impactées et contributrices

Toutes les obligations sont présentées dans une arborescence qui permet de visualiser leurs inter-relations.

^ Obligations

Obligation impactée

Privacy protection for information and data

Obligation contributrice

+ Nouveau Relier Réordonner Rapport instantané

Nom	Qualification de l'obligation	Nature
Establish, implement, and maintain a customer due dili...	Préconisé	Prévention
Define and assign the data controller's data quality role...	Préconisé	Prévention
Establish and maintain customer data authentication p...	Connexe	Prévention

Il est utile d'avoir plusieurs obligations dans l'onglet **Obligations impactées** d'une obligation. En effet, cela signifie que les efforts déployés pour mettre en œuvre une obligation légale contribuent à la mise en œuvre des obligations qu'elle impacte.

Mise en application des obligations

il existe trois niveaux de mise en application pour chaque obligation :

- **légal**

 Toute obligation légale est associée directement à un article. Elle permet de mettre en œuvre un texte de référence.

- **connexe**

 Une obligation connexe est une obligation qui ne revêt pas de caractère légal et qui est parent d'une obligation légale. Elle permet d'indiquer que l'une des obligations présente dans sa hiérarchie revêt un caractère légal.

- **préconisé**

 Une obligation préconisée est une obligation qui ne revêt pas de caractère légal et qui est enfant d'une obligation légale. Elle donne des

détails concernant la façon de mettre en œuvre l'obligation légale et facilite sa mise en œuvre.

Niveau d'application des obligations	
Obligation connexe	- N'a pas de caractère légal - Contient au moins une obligation légale dans sa hiérarchie - Permet d'afficher les obligations légales présentes dans la hiérarchie UCF - Est impactée par une obligation légale
Obligation préconisée	- N'a pas de caractère légal - Apparaît sous une obligation légale (contribue à une obligation légale)
Obligation légale	- Contribue à une obligation connexe - Peut être impactée par des obligations préconisées - Peut contribuer ou être impactée par d'autres obligations légales

Associer une obligation à un article de texte de référence

A partir d'un article, vous pouvez créer une obligation ou lui relier une obligation existante.

Pour créer une obligation sur un article :

3. Dans l'arbre du texte de référence, passez le curseur sur l'article. Des boutons apparaissent à sa droite.
4. Cliquez sur le bouton **Nouveau > Obligation**. La fenêtre de création d'une obligation apparaît.
5. Saisissez le nom de l'obligation.
6. Cliquez sur **OK**.

Pour relier une obligation existante à un article :

1. Dans l'arbre du texte de référence, sélectionnez l'article. Des boutons apparaissent à sa droite.
2. Cliquez sur le bouton **Relier > Obligation**. La fenêtre de recherche apparaît.
3. Saisissez le nom de l'obligation ou cliquez directement sur le bouton **Rechercher**. La liste des obligations apparaît.
4. Sélectionnez l'obligation en question.
5. Cliquez sur **Relier**.

Voir les articles associés à une obligation

Pour visualiser les articles associés à une obligation :

1. Ouvrez les propriétés de l'obligation.
2. Cliquez sur la page **Caractéristiques**.
3. Dépliez la section **Articles correspondants**.

FAIRE L'INVENTAIRE DES CADRES DE POLITIQUES D'ENTREPRISE (RÉGLEMENTATIONS INTERNES)

Une politique d'entreprise est une directive dont le but est de gouverner ou de guider l'entreprise.

Dans **Hopex Data Governance**, un *cadre de politique d'entreprise* définit un ensemble de politiques d'entreprise.

Il est composé de sections et de sous-sections qui représentent des catégories de politique entreprise. Sous ces sections vous pouvez définir les politiques d'entreprise, les actifs contraints par les politiques en question ainsi que leur mise en œuvre.

Exemple

Le règlement interne d'une entreprise est un cadre de politique d'entreprise.

Ce cadre a différentes sections qui représentent des catégories de politique : la sécurité, l'accueil, l'usage de la cafétéria, etc.

Chaque section a des politiques d'entreprise : fermeture des casiers, nettoyage des tables, etc.

Créer un cadre de politique d'entreprise

Pour créer un cadre de politique d'entreprise :

1. Cliquez sur le menu de navigation **Conformité > Textes de référenc.**
2. Dans la zone d'édition, cliquez sur l'onglet **Cadres de politique d'entreprise.**
3. Cliquez sur **Nouveau.**
4. Dans la fenêtre qui apparaît, indiquez le nom du cadre de politique d'entreprise et sa description.
5. Cliquez sur **OK.**
Le cadre apparaît dans la liste.

Ajouter une section au cadre de politique d'entreprise

Pour ajouter une section au cadre de politique d'entreprise :

1. Dans l'arbre de navigation, cliquez sur l'icône du cadre en question puis sur **Nouveau > Section de cadre de politique d'entreprise.**

2. Indiquez :
 - l'intitulé
 - la description
 - (optionnel) les objets contraints par la section. Il peut s'agir de données métier, physique, d'applications, etc.
 - (optionnel) la mise en œuvre : il s'agit des règles métier qui assurent l'implémentation des politiques de la section. Voir [Assurer la conformité des données : créer des règles métier](#).
3. Cliquez sur **OK**.
La section apparaît dans l'arbre de navigation.

Ajouter une politique d'entreprise

Une politique d'entreprise est une directive dont le but est de gouverner ou guider l'entreprise. Une politique d'entreprise sert de base à la définition des règles métier et gouvernent les processus de l'entreprise. Une politique d'entreprise est toujours sous le contrôle de l'entreprise. Elle permet de contrôler, guider et formaliser les stratégies et les tactiques de l'entreprise.

Pour ajouter une politique d'entreprise à une section :

1. Dans l'arbre de navigation, cliquez sur l'icône de la section concernée puis sur **Nouveau > Politique d'entreprise**.
2. Indiquez :
 - le nom de la section
 - l'intitulé
 - le commentaire
 - (optionnel) les définitions de concepts qui peuvent être associées à des mots clés du texte
 - (optionnel) les objets contraints par la politique d'entreprise
 - (optionnel) la mise en œuvre : il s'agit des règles métier qui assurent l'implémentation effective de la politique d'entreprise. Voir [Assurer la conformité des données : créer des règles métier](#).

Analyser les informations contraintes par une réglementation

Le rapport "Conformité réglementaire et transformation des données" vous permet de visualiser les données contraintes par une réglementation spécifique.

Pour afficher les données contraintes par des cadres réglementaires ou des cadres de politiques d'entreprise :

1. Cliquez sur le menu de navigation **Rapports**.
2. Dans la fenêtre d'édition, cliquez sur **Créer un rapport**.
3. Recherchez le rapport type "Conformité réglementaire et transformation des données".
4. Sous le rapport type, cliquez sur **Créer un rapport**.
L'assistant de création apparaît.
5. Cliquez sur **Relier**.
6. Recherchez et sélectionnez les cadres réglementaires à analyser.

7. Validez la création du rapport.

Voir aussi : Définir les objets de votre organisation contraints par un texte de référence.

IMPORTER DES DOCUMENTS D'UCF

UCF (Unified Compliance Framework) représente la plus grande bibliothèque de contenu réglementaire. UCF contient des :

- "Authority Documents"
- "Citations"
- "UCF Controls"

Hopex Data Governance offre une intégration qui vous permet d'accéder à la bibliothèque de documents et de contrôles de conformité de l'UCF - Unified Compliance Framework.

La combinaison de la solution **Hopex Data Architecture** et de l'UCF automatise les démarches de mise en conformité informatique et permet de gérer plus simplement un large éventail de réglementations et de normes IT. L'UCF aide à mettre en œuvre les contrôles appropriés pour chaque réglementation, tout en identifiant les contrôles communs à plusieurs documents réglementaires. Les entreprises peuvent ainsi éviter les duplications d'efforts et réduire ainsi les délais et les coûts liés à la conformité.

HOPEX dresse un inventaire des réglementations pertinentes en important les politiques, exigences et contrôles réglementaires d'UCF et les met à jour. La solution fournit également un ensemble de rapports et de tableaux de bord prêts à l'emploi pour suivre le niveau de conformité et les plans d'actions correctives, ainsi que leur impact sur l'écosystème informatique.

Pour plus de détails sur UCF, voir [A propos de "Unified Compliance Framework"](#).

Utiliser l'import UCF

Conditions préalables à l'import UCF

L'administrateur fonctionnel peut télécharger du contenu UCF ("Authority Documents", "Citations" et "Common Controls" et le mettre à jour.

Pour pouvoir importer ce contenu dans **HOPEX UCF**, vous devez avoir :

- **Hopex Data Governance ET HOPEX UCF**
- un compte UCF et une clé API UCF
- une "Shared List" (liste partagée) avec les (seuls) "Authority Documents" que vous voulez importer.
 - Pour plus de détails, voir [Unified Compliance Framework](#).
- paramétré les options UCF dans **HOPEX UCF**
 - Dans le Common Controls Framework d'UCF, les informations sont généralement disponibles en anglais.
 - Si vous souhaitez utiliser **HOPEX UCF** alors que vous utilisez **Hopex** dans une langue de données utilisateur autre que l'anglais, vous devez :
 - paramétrer la langue de données souhaitée (par exemple, le français, si vous souhaitez utiliser **Hopex** avec des données en français)
 - procéder à l'import
 - répéter l'opération (changer de langue de données + procéder à l'import) autant de fois que de langues souhaitées

Paramétrer l'import UCF

Pour paramétrer l'import UCF :

1. Dans **Menu principal**, sélectionnez **Paramètres > Options**.
2. Dans la fenêtre d'options, déployez **Outils > Echange de données > Import > Intégration UCF Common Controls Hub**.
3. Sélectionnez la case **Activer l'import UCF**.
4. Saisissez l'URL correspondant à l'API UCF.
`https://api.unifiedcompliance.com/`
5. Saisissez votre **Clé d'authentification** pour l'API UCF.
 - Pour récupérer votre clé d'authentification dans votre espace de travail Unified Compliance Framework :
 - Cliquez sur **Settings > API Manager > API Keys**.
 - Cliquez sur "Create Credentials" et copiez-collez votre clé API.
6. Cliquez sur **OK**.

Importer des données à partir du Common Controls Hub

Dans **Hopex Data Governance**, l'administrateur fonctionnel peut importer les données pertinentes à partir du Common Controls Hub d'UCF (Authority Documents, Citations et Controls).

Pour importer les données UCF :

1. Cliquez sur le menu de navigation **Conformité > Textes de référence**.
2. Dans la zone d'édition, cliquez sur **Importer du contenu UCF**.
3. Cliquez sur **Suivant**.
4. Sélectionnez la "Shared List" de votre Common Controls Hub.
5. Cliquez sur **Suivant**.

6. Sélectionnez le(s) "Authority Document(s)" que vous voulez importer dans **Hopex**.

UCF Import - Textes de référence				
Les textes de référence contenus dans la liste partagée préalablement sélectionnée apparaissent ci-dessous. Sélectionnez les textes de référence que vous souhaitez importer dans HOPEX.				
☐	Nom ↑	Présent dans HOPEX?	Dernier import...	Dernière mise à jour UCF ...
☐	3 CCR 702-6	Non		2/24/2019
☐	AICPA Incident Response	Non		9/9/2019
☒	AICPA Privacy	Non		9/9/2019

☛ Si vous mettez à jour un "Authority Document" déjà importé, il peut être utile de comparer les colonnes **Dernier import de mise à jour UCF** et **Dernière mise à jour UCF disponible**.

7. Cliquez sur **Suivant**.

Spécifier le contenu réglementaire applicable

Pertinence du contenu réglementaire

Tous les articles/sections d'un texte de référence importé ne sont pas applicables à votre organisation.

Les responsables de la conformité doivent étudier les textes de référence importés, identifier le contenu applicable et le déclarer comme tel.

Seul le contenu considéré applicable est visible des parties prenantes dans les bibliothèques **Hopex**.

☛ Le contenu réglementaire que vous créez directement dans **Hopex** est automatiquement considéré comme étant applicable.

Procéder à la revue des textes de référence après import

Une fois que les données UCF ont été importées, l'arborescence apparaît sous **Conformité > Par obligations > Import UCF**.

Cette arborescence affiche les textes de référence (Authority Documents), les articles (Citations) ainsi que les obligations légales rattachées (Common Controls). Elle se base sur les relations entre obligations impactées / contributrices (supported/supporting) telles que définies dans UCF.

Vous pouvez, à partir de cette arborescence :

- procéder à la revue des textes de référence importés.
- spécifier le contenu pertinent pour votre organisation.

Sélectionner le contenu réglementaire pertinent pour votre organisation

Pour spécifier le contenu pertinent :

- » Dépliez l'arborescence et sélectionnez la case correspondant aux textes de référence / articles / sections auxquels vous devez vous conformer.

	Applicable	Sous-éléments
 EU General Data Protection Regulation (GDPR)	☒	7
 Chapter II. Principles	☒	7
 Chapter III. Rights of the data subject	☐	4
 Chapter IV. Controller and processor	☐	5
 Chapter V. Transfers of personal data to third countries or international or...	☐	6

 Le cadre grisé  indique que seuls certains éléments du contenu ont été sélectionnés dans l'arborescence.

Les données correspondant aux articles sélectionnés sont visibles par les contrôleurs internes dans la bibliothèque d'obligations.

ASSURER LA CONFORMITÉ DES DONNÉES : CRÉER DES RÈGLES MÉTIER

Une règle métier constitue une action pour s'assurer qu'une réglementation interne ou externe est bien respectée au sein de l'organisation.

Par exemple, la réglementation bancaire en matière de prévention du blanchiment impose à chaque établissement bancaire une connaissance actualisée de ses clients. Pour appliquer cette obligation la banque crée une règle métier qui consiste à analyser systématiquement les revenus et le patrimoine de ses clients via l'envoi d'un questionnaire détaillé.

Hopex Data Governance distingue trois types de règle d'entreprise :

- règle métier : la définition référence une information métier (ex. concept)
- règle opérationnelle : la définition référence une donnée logique (classe, entité, vue de données)
- règle système : référence une donnée logique ou physique (classe, entité, table, vue de données, vue physique)

Définir une règle d'entreprise

Pour créer une règle :

1. Cliquez sur le menu de navigation **Conformité > Efficacité opérationnelle > Règles métier**.
2. Sélectionnez le type de règle à créer : métier, opérationnelle ou système.
3. Cliquez sur le bouton **Nouveau**.
4. Dans la fenêtre qui apparaît, indiquez le nom de la règle et éventuellement le détenteur.
5. Cliquez sur **OK**.

Pour définir le contenu de la règle :

1. Ouvrez les propriétés de la règle.
2. Sous la page **Caractéristiques** indiquez le texte de la règle et d'éventuelles **définitions**.
3. Sous la page **Application** de la règle indiquez le moyen qui permet d'être conforme à la réglementation sélectionnée.

Par exemple, l'application " APIX " applique la règle système " RS #1 " qui assure une mise en œuvre de la section 3 de BCBS 230.

Rapport de règle

Un rapport permet de visualiser le contenu d'une règle.

Pour générer ce rapport :

1. Cliquez sur le menu de navigation **Rapports**.
2. Dans la fenêtre d'édition, cliquez sur **Créer un rapport**.
3. Recherchez le rapport type "Description des règles".
4. Sous le rapport type, cliquez sur **Créer un rapport**.
L'assistant de création apparaît dans la zone d'édition.
5. Cliquez sur **Relier**.
6. Sélectionnez le type de règle (métier, opérationnelle ou système) et la règle à étudier.
7. Cliquez sur **Relier**.
8. Rafraîchissez le rapport.

Pour chaque règle sélectionnée en entrée, le rapport présente :

- sa description (commentaire)
- la réglementation ou schéma de réglementation qu'elle exécute
- l'objet qui l'applique (processus pour une règle métier, application pour une règle système...)



QUALITÉ DES DONNÉES



La qualité des données implique de préparer les données afin qu'elles répondent aux réglementations auxquelles est assujettie l'entreprise ainsi qu'aux besoins de ses utilisateurs métier.

Pour cela, l'entreprise doit définir des règles qui fixent les critères de qualité et les objectifs à atteindre pour chacun des ces critères.

Hopex Data Governance vous permet de mettre en place cette politique de qualité des données mais aussi de construire un référentiel d'évaluation et de contrôle pour faire en sorte que les données atteignent le niveau de qualité désiré.

- ✓ Définir une politique de qualité des données
- ✓ Évaluer la qualité des données

DÉFINIR UNE POLITIQUE DE QUALITÉ DES DONNÉES

Créer une politique de qualité

Les politiques de qualité des données sont des directives qui décrivent à quels critères doivent répondre les données pour être utiles et exploitables au sein d'une organisation.

Une politique de qualité des données s'inscrit dans un cadre de politique d'entreprise. Elle décrit :

- les objets auxquels elle s'applique
- les dimensions qui constituent les critères de qualité à respecter



Un cadre de politique d'entreprise est un document interne publié par l'organisation. Il définit un ensemble de politiques d'entreprise. Voir [Faire l'inventaire des cadres de politiques d'entreprise \(réglementations internes\)](#).

Exemple

Un règlement au sein de l'école invite les parents de nouveaux élèves à remplir une fiche qui comprend un questionnaire médical et les coordonnées des personnes à contacter en cas d'urgence, ainsi que la confirmation du nom, de l'adresse et de la date de naissance de l'élève.

Une politique de qualité des données est définie : elle doit respecter la dimension "Complétude", avec un seuil dont la valeur doit être égale à 100%.

Pour créer une politique de qualité des données dans **Hopex Data Governance** :

1. Cliquez sur le menu de navigation **Conformité > Qualité des données > Politiques**.
2. Dans la zone d'édition, cliquez sur l'onglet **Politiques**.
3. Cliquez sur **Nouveau**.
4. Indiquez :
 - le nom de la politique de qualité
 - une description
 - le sujet, qui représente l'actif contraint
 - la dimension



Une dimension de la qualité des données est une caractéristique mesurable des données. Pour plus de détails, voir [Dimensions de qualité](#).

5. Cliquez sur **OK**.
La politique de qualité apparaît dans la zone d'édition.

Définir des indicateurs de qualité

Vous pouvez créer des indicateurs afin de suivre la qualité des données de votre référentiel. Vous pouvez les créer directement sur les données et/ou les associer à une politique de qualité.

Exemple

Dans le cadre du règlement de l'école, la fiche de renseignements des élèves est associée à une politique de qualité qui spécifie que la fiche doit être complète. Vous pouvez créer un indicateur dont la valeur indique si la fiche est conforme à la politique de qualité.

Créer un indicateur de qualité

Pour créer un indicateur de qualité sur une donnée :

1. Ouvrez les propriétés de la donnée en question (par exemple un Concept).
2. Cliquez sur **Qualité des données > Indicateurs**.
3. Cliquez sur **Nouveau**.
4. Dans l'assistant qui apparaît, indiquez le nom de l'indicateur et une description.
5. Cliquez sur **Suivant**.
6. Indiquez :
 - la politique de qualité de données concernée, le cas échéant
 - les limites haute et basse, ainsi que l'unité de mesure de l'indicateur.

Définir la dernière valeur d'un indicateur

Les valeurs d'un indicateur sont les mesures prises aux dates indiquées. Elles peuvent provenir d'outils techniques spécifiques effectuant les tests ou des vérifications relatives aux politiques de qualité des sources de données cibles.

Pour définir la valeur d'un indicateur :

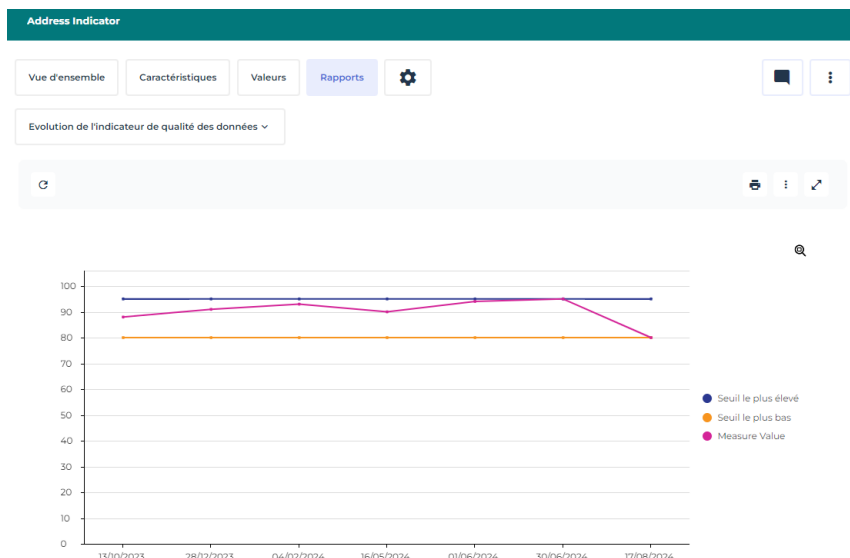
1. Ouvrez les propriétés de l'indicateur.
2. Cliquez sur la page **Valeurs**.
3. Indiquez la valeur de l'indicateur à la date donnée.

Rapport de l'indicateur

Un rapport permet de visualiser l'évolution des valeurs de l'indicateur.

Pour générer ce rapport :

- 1 Dans les propriétés de l'indicateur, cliquez sur **Rapports > Evolution de l'indicateur de qualité des données**.
Le rapport présente les courbes des seuils haut et bas de l'indicateur ainsi que la courbe de ses évaluations dans le temps.



Importer des politiques et indicateurs de qualité des données

A partir d'un fichier Excel vous pouvez créer dans le référentiel HOPEX un ensemble de politiques de qualité avec les indicateurs associés, leurs seuils et valeurs, ainsi que les actifs de données concernés.

Lorsque vous importez une politique de qualité dans HOPEX, vous pouvez définir un conteneur (Policy Framework) dans lequel créer cette politique de qualité. Si une politique de qualité du même nom existe dans le fichier Excel mais sans conteneur, celle-ci est donnée lieu à la création d'un autre objet.

Lorsque vous importez des actifs de données, vous pouvez préciser pour chacun :

- le conteneur (par exemple, un dictionnaire métier)
- le type d'objet (par exemple, un concept)
- et le nom de l'objet.

Dans la feuille **_Description** du fichier Excel vous pouvez consulter la liste des colonnes qu'il est obligatoire de définir, telles que le nom de l'objet.

Télécharger le modèle d'import

Pour télécharger le modèle que vous allez utiliser pour l'import de vos données :

1. Ouvrez le **Menu principal** d'Hopex.

2. Sélectionnez **Import** > **Téléchargement de modèle Excel**.
3. Passez le curseur sur la ligne du modèle qui vous intéresse.
4. Cliquez sur **Télécharger**.

Importer les objets dans HOPEX

Pour importer le fichier Excel qui contient les objets :

1. Cliquez sur **Menu principal** puis sur **Import** > **Excel**.
2. Dans l'assistant d'import, sélectionnez le fichier Excel et le mode d'import :
 - maintenant : l'import immédiat lance le processus d'import dès que vous cliquez sur **Importer**.
 - en arrière plan : l'import en arrière-plan vous permet de planifier l'exécution du processus d'import de façon à ce que vous puissiez continuer à utiliser HOPEX indépendamment de cette opération. Vous pouvez sauvegarder l'historique des objets.
3. Cliquez sur **Importer**.

De la même façon, vous pouvez exporter, via ce modèle Excel, les politiques de qualité des données contenues dans votre référentiel.

Exemple de fichier d'import

Le fichier Excel suivant contient une liste d'indicateurs.

B	C	D	E	F	G	H	I
Name	Description	Data Asset Container Type	Asset Container I	Data Asset Type	Data Asset Name	Lower Threshold	Higher Threshold
Customer_Id Indicator		Business Dictionary	B.Dictionary	Concept Property	Customer_Id	90	100
Customer First Name Indicator		Business Dictionary	B.Dictionary	Concept Property	Customer First Name	95	100
Email Indicator		Business Dictionary	B.Dictionary	Concept Property	Email	98	100
Address Indicator		Business Dictionary	B.Dictionary	Concept Property	Address	80	95
Phone Number Indicator		Business Dictionary	B.Dictionary	Concept Property	Phone Number	95	100
Birth Date Indicator		Business Dictionary	B.Dictionary	Concept Property	Birth Date	85	98
Customer Last Name Indicator		Business Dictionary	B.Dictionary	Concept Property	Customer Last Name	70	85

Une fois importés dans HOPEX, ils sont visibles sous le menu **Conformité** > **Qualité des données** > **Indicateurs**.

La politique de qualité et la donnée associée à un indicateur sont affichées dans la page **Caractéristiques** de ses propriétés.

Définir des contrôles de qualité des données

Des contrôles de mise en œuvre peuvent être associés aux obligations.



Les obligations constituent une interprétation de la loi et contribuent à la mise en application de tout article auquel votre organisation doit se conformer.

Pour créer un contrôle de qualité des données :

1. Cliquez sur le menu de navigation **Conformité** > **Efficacité opérationnelle** > **Contrôles de données**.
2. Sous l'onglet **Contrôle des données**, cliquez sur **Nouveau**.

3. Indiquez le nom et éventuellement le détenteur.
4. Cliquez sur **OK**.
Le contrôle apparaît dans la zone d'édition.
5. Passez le curseur sur le contrôle et cliquez sur le bouton **Propriétés**  pour définir ses caractéristiques :
 - **Code** : permet d'identifier le contrôle de manière unique.
 - **Nature** : correction, détection, prévention
 - **Fréquence** : quotidienne, hebdomadaire, mensuelle, à chaque transaction, bimensuelle
 - **Obligation** associée au contrôle

 *Les obligations constituent une interprétation de la loi et contribuent à la mise en application de tout article auquel votre organisation doit se conformer.*

 - **Articles** du texte de référence concernés
 - **Mesure de qualité des données**
 - **Contrôle d'exécution**

Identifier les problèmes de qualité des données

Lorsque les contrôles de qualité des données sont jugés insatisfaisants, vous pouvez spécifier des problèmes de qualité et mettre en place des plans d'action pour y remédier.

Pour définir un problème de qualité :

1. Cliquez sur le menu de navigation **Conformité > Efficacité opérationnelle > Remédiation des données**.
2. Dans la zone d'édition, cliquez sur l'onglet **Problèmes**.
La liste des problèmes apparaît.
3. Cliquez sur **Nouveau**.
4. Nommez le problème et cliquez sur **OK**.
Le problème apparaît dans la zone d'édition.

Remédier aux problèmes de qualité rencontrés : plans d'action

Vous pouvez mettre en place des plans d'action pour améliorer un contrôle de qualité qui n'a pas été jugé satisfaisant.

Pour créer un plan d'action :

1. Cliquez sur le menu de navigation **Conformité > Efficacité opérationnelle > Remédiation des données**.
2. Dans la zone d'édition, cliquez sur l'onglet **Plans d'action**.
La zone d'édition affiche :
 - Tous les plans d'action
 - Plans d'action en retard
3. Sous l'onglet **Tous les plans d'action**, cliquez sur **Nouveau**.
4. Indiquez le nom du plan d'action et éventuellement un commentaire ainsi que les dates de début et de fin.

5. Cliquez sur **OK**.
Le plan d'action apparaît dans la liste.
6. Passez le curseur sur le plan d'action et cliquez sur le bouton **Propriétés**
 pour définir ses caractéristiques.

Propriétés d'un plan d'action dans Hopex Data Governance

Caractéristiques principales

Vous pouvez spécifier les informations suivantes :

- **Nom** : nom du plan d'action
- **Propriétaire** : ce champ est renseigné par défaut par l'utilisateur qui crée le plan d'action.
- **Entité propriétaire** : entité responsable de la mise en œuvre du plan d'action.
- **Approbateur** : utilisateur responsable de la validation du plan d'action quand toutes les actions sont terminées.
- **Moyens** : description textuelle des moyens nécessaires /souhaités pour l'exécution du plan d'action.
- **Priorité du plan d'action**: permet d'indiquer un niveau. La priorité peut être :
 - "Basse"
 - "Moyenne"
 - "Elevée"
 - "Critique".
- **Origine** : permet de définir le contexte de réalisation du plan d'action :
 - "Audit"
 - "Conformité"
 - "Événement"
 - "Risque"
 - "RFC"
 - "Autre".
- **Catégorie** : le plan d'action peut par exemple être lié à :
 - la réduction de l'impact des risques
 - la gestion de projet
 - l'amélioration des processus
 - l'amélioration de la performance des contrôles
 - etc.
- *De nombreuses autres valeurs sont disponibles.*
- **Nature** : permet de définir s'il s'agit d'un plan d'action :
 - Correctif
 - Préventif.
- **Commentaire** : permet d'apporter un complément d'information sur le plan d'action et ses caractéristiques.
- **Calendrier de pilotage** : permet d'envoyer des rappels à la personne responsable d'un plan d'action afin qu'elle renseigne le taux d'avancement de ce plan d'action.
 -  *Un calendrier de pilotage pour un rappel mensuel d'avancement est fourni par défaut.*
- **Statut de plan d'action** : indique à quelle étape du workflow se situe le plan d'action.

Analyse financière

- **Coût prévu** : estimation du coût du plan d'action
- **Coût prévu (Jour-Homme)** : estimation exprimée en jours.homme de la charge nécessaire à la mise en œuvre du plan d'action

Responsabilités

L'utilisateur défini comme **Réalisateur** du plan d'action est responsable de la définition des actions à réaliser ainsi que de leur réalisation.

Cette section est définie par l'utilisateur qui crée le plan d'action ou par l'approbateur du plan d'action.

Facteurs de succès

Dans la section **Facteurs clés de succès**, vous pouvez renseigner de manière textuelle des indicateurs de succès qui permettront de juger de la réussite du plan d'action.

Périmètre

Pour positionner un plan d'action dans son environnement, vous pouvez associer des objets à ce plan d'action dans la section **Périmètre**.

Vous pouvez relier des objets des types suivants :

- contrôles
- applications
- risques
- entités
- processus
- problèmes

Jalons

Les jalons sont des dates clés du plan d'action.

🔒 La date de fin planifiée est obligatoire.

Actions

Le propriétaire du plan d'action doit définir les actions permettant au plan d'action d'aboutir. Il a la possibilité de créer des actions et les affecter.



Une action est incluse dans un plan d'action et représente une transformation ou un traitement dans une organisation ou un système.

7. Dans la section **Actions**, cliquez sur **Nouveau**.
8. Remplissez les colonnes associées à l'action.

🔒 Pour plus de détails sur les actions, voir [Gérer les actions](#).

Pièces jointes

Vous pouvez attacher des documents métier à un plan d'action.

🔒 Pour plus de détails sur l'utilisation des documents métier, voir le guide **Hopex Common Features**.

ÉVALUER LA QUALITÉ DES DONNÉES

L'évaluation peut être effectuée directement sur les données ou à distance via des questionnaires.

Hopex Data Governance fournit par défaut un modèle d'évaluation des données qui porte sur les dimensions suivantes :

- complétude
- exactitude
- cohérence
- validité
- unicité
- fraîcheur

Dimensions de qualité

Une dimension de la qualité des données est une caractéristique mesurable des données. Les dimensions de la qualité des données permettent d'établir les exigences en matière de qualité des données.

Hopex Data Governance livre un certain nombre de dimensions prédéfinies. Pour afficher la liste des dimensions prédéfinies :

1. Cliquez sur le menu de navigation **Conformité > Qualité des données**.
2. Dans la zone d'édition, cliquez sur l'onglet **Dimensions**.

Dimension	Description																
Accessibilité	Traduit la facilité d'accès et d'utilisation des données, à un niveau pratique (rapide, sans intervention de personnes extérieures).																
Actualité	Mesure dans laquelle les données représentent la réalité à partir du moment requis. L'actualité des données implique que les données aient été mises à jour si besoin pour rester pertinentes.																
Cohérence	Identifie le niveau de cohérence dans les données, l'absence de différence lorsqu'on compare deux ou plusieurs représentations d'une chose à une définition. Exemple Ci-dessous une incohérence dans le format des données. <table><tr><th>Order Number</th><th>Client Id</th><th>ShipDate</th><th>Total</th></tr><tr><td>1000</td><td>1</td><td>1/12/2018</td><td>100\$</td></tr><tr><td>1001</td><td>2</td><td>1/12/2018</td><td>200€</td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	Order Number	Client Id	ShipDate	Total	1000	1	1/12/2018	100\$	1001	2	1/12/2018	200€				
Order Number	Client Id	ShipDate	Total														
1000	1	1/12/2018	100\$														
1001	2	1/12/2018	200€														

Dimension	Description																				
Complétude	Identifie le niveau d'exhaustivité des données et les propriétés manquantes. Exemple : Ci-dessous certaines colonnes n'ont pas de valeur (en rouge) et d'autres sont tronquées (Dupont@Samp.gm) <table><tr><th>First Name</th><th>Last Name</th><th>Billing Address</th><th>Shipping Address</th><th>Email</th></tr><tr><td>Dupont</td><td></td><td>9 rue Rene Coty Paris 75002</td><td>NULL</td><td>Dupont@Sample.gm</td></tr><tr><td>Durand</td><td>Robin</td><td>344 rue de Rivoli 75001</td><td>NULL</td><td>Durand@Sample.com</td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>	First Name	Last Name	Billing Address	Shipping Address	Email	Dupont		9 rue Rene Coty Paris 75002	NULL	Dupont@Sample.gm	Durand	Robin	344 rue de Rivoli 75001	NULL	Durand@Sample.com					
First Name	Last Name	Billing Address	Shipping Address	Email																	
Dupont		9 rue Rene Coty Paris 75002	NULL	Dupont@Sample.gm																	
Durand	Robin	344 rue de Rivoli 75001	NULL	Durand@Sample.com																	
Confiance	La gouvernance des données, la protection des données et la sécurité des données sont-elles en place ? Quelle est la réputation des données, et sont-elles vérifiées et vérifiables ?																				
Exactitude	Identifie le niveau de données exactes, fiables. Exemple : Ci-dessous, pour Dupont, le poste et le département sont inversés. Pour Durand, le poste affiche une erreur typographique Pour René, le département affiche une valeur erronée. <table><tr><th>First Name</th><th>Position</th><th>Department</th><th>Email</th></tr><tr><td>Dupont</td><td>Product Management</td><td>Business Analyst</td><td>Dupont@Sample.gmail</td></tr><tr><td>Durand</td><td>Sftware Engineer</td><td>Product Development</td><td>Durand@Sample.com</td></tr><tr><td>René</td><td>Test Analyst</td><td>xxùpoi*£</td><td>Rene@Sample.com</td></tr></table>	First Name	Position	Department	Email	Dupont	Product Management	Business Analyst	Dupont@Sample.gmail	Durand	Sftware Engineer	Product Development	Durand@Sample.com	René	Test Analyst	xxùpoi*£	Rene@Sample.com				
First Name	Position	Department	Email																		
Dupont	Product Management	Business Analyst	Dupont@Sample.gmail																		
Durand	Sftware Engineer	Product Development	Durand@Sample.com																		
René	Test Analyst	xxùpoi*£	Rene@Sample.com																		
Fraîcheur	Ce critère évalue si l'information est disponible au moment requis. La fraîcheur des données est essentielle pour avoir une bonne vision d'une situation à un moment donné et pour prendre les décisions relatives aux données. La fraîcheur est importante à deux égards : un court délai entre la collecte des données et leur analyse et un court délai entre la notification et l'action d'optimisation ou de correction qui en résulte.																				
Pertinence	La pertinence des données indique dans quelle mesure les données répondent aux besoins des utilisateurs. Les besoins d'information peuvent changer et il est important que des évaluations aient lieu pour s'assurer que les données collectées sont toujours pertinentes pour les décideurs.																				
Sécurité des données	La sécurité des données couvre la notion d'empowerment (autorisation d'accès à des données sensibles), les mesures prises contre la perte d'informations, le contrôle du risque de fuite d'informations sensibles.																				
Traçabilité	La traçabilité permet de suivre l'évolution des informations depuis leur collecte jusqu'à leur restitution, y compris leur traitement. Très souvent, elle est associée à l'historique d'un processus ou d'un produit.																				

Dimension	Description																
Unicité	Ce critère évalue le niveau d'unicité des données. Exemple : On ne doit pas trouver dans la table "Client" deux fois la même occurrence, chaque enregistrement doit être unique.																
Utilisabilité	Les données sont-elles compréhensibles, simples, pertinentes, accessibles, mainte- nables et au bon niveau de précision ?																
Valeur	La valeur des données traduit leur intérêt : existe-t-il un bon rapport coût / bénéfice pour les données ? Sont-elles utilisées de manière optimale ? Mettent-elles en danger la sécurité ou la vie privée des personnes ou les responsabilités légales de l'entreprise ? Soutiennent-elles ou contredisent-elle l'image de marque ou le message de l'entre- prise ?																
Validité	Identifie le niveau de données valides. Les données sont valides lorsqu'elles sont conformes à la syntaxe (format, type) de leur définition. Exemple : La valeur du champ "Unités disponibles" sur Prod1 ne devrait pas être négative. Une date de retrait est définie sur Prod2 mais le champ "Unités disponibles" n'affiche pas une valeur nulle. <table><tr><th>Product Code</th><th>Name</th><th>Units Available</th><th>Retire Date</th></tr><tr><td>1000</td><td>Prod1</td><td>-10</td><td>12/4/2020</td></tr><tr><td>1001</td><td>Prod2</td><td>100</td><td>31/12/2017</td></tr><tr><td></td><td></td><td></td><td>-</td></tr></table>	Product Code	Name	Units Available	Retire Date	1000	Prod1	-10	12/4/2020	1001	Prod2	100	31/12/2017				-
Product Code	Name	Units Available	Retire Date														
1000	Prod1	-10	12/4/2020														
1001	Prod2	100	31/12/2017														
			-														
Vraisemblance	La vraisemblance examine si les données se situent dans une fourchette acceptée, par exemple, entre le maximum et le minimum, la distribution et les valeurs aber- rantes. Hopex fournit un modèle Excel qui vous permet d'évaluer les critères de données dans un fichier et de les importer dans votre référentiel.																

Objets de l'évaluation

L'évaluation peut porter sur les objets suivants :

- concept, vue de concept
- classe, vue de données
- table, vue physique
- tous les domaines de données (métier, logique et physique)

Évaluation directe

Pour évaluer de façon directe une donnée :

1. Ouvrez les propriétés de la donnée en question.
2. Sélectionnez la page **Evaluation**.
3. Cliquez sur **Evaluer**.
4. Dans la page qui apparaît, sélectionnez une valeur pour chaque question.
Voir [Dimensions de qualité](#).
5. Cliquez sur **OK**.

Évaluation par campagne

L'administrateur fonctionnel peut créer des campagnes ou des sessions d'évaluation des données.

Lors de la création d'une campagne, des questionnaires sont envoyés aux répondants désignés afin d'obtenir des estimations qualitatives sur des objets dont ils sont responsables.

Pour plus de détails sur les campagnes et sessions, voir [Campagnes d'évaluation](#).

Conditions préalables à l'évaluation des données

Avant de lancer une campagne d'évaluation de données vous devez avoir préparé l'environnement de travail. Assurez-vous d'avoir défini des répondants pour les données, et spécifié pour chacun l'entité à laquelle il est rattaché ainsi qu'un email.

Créer une campagne d'évaluation

Pour créer une campagne d'évaluation avec le modèle fourni en standard :

1. Cliquez sur le menu de navigation **Outils > Campagnes d'évaluation**.
2. Dans la fenêtre d'édition, sous l'onglet **Campagnes**, cliquez sur **Nouveau**.
La page de création d'une campagne apparaît.
3. Indiquez le nom de la campagne.
4. Sélectionnez le **Modèle d'évaluation** "Data Quality Assessment".
5. Modifiez éventuellement le **Calendrier**.
 *Le calendrier sert à initialiser les dates de début et de fin de la campagne d'évaluation.*
6. Indiquez la **Date de Début** et la **Date de fin**.
7. Cliquez sur **Suivant**.
8. Dans la fenêtre **Sélection du périmètre**, sélectionnez les objets qui définissent le contexte de l'évaluation.
Le contexte englobe les éléments de la branche qui remonte de l'objet en question jusqu'à la racine.
 *Si vous dé-sélectionnez un nœud d'une branche, seuls les enfants de cette branche sont dé-sélectionnés.*
9. Cliquez sur **Suivant**.

10. Dans la fenêtre de prévisualisation, cliquez sur **Rafraîchir le rapport**. Les éléments qui vont être évalués apparaissent.

Vous pouvez visualiser notamment :

- les caractéristiques évaluées (définies dans le modèle d'évaluation)
- les objets évalués
- les objets contextes
- les nœuds d'évaluation, qui correspondent aux objets placés dans les différents objets contextes, associés aux répondants.
- les répondants

11. Cliquez sur **OK**.

Pour plus d'informations sur les campagnes, voir [Campagnes d'évaluation](#).

Voir aussi :

- ✓ [Importer des évaluations de données](#)

Rapports de qualité des données

Des rapports de qualité offrent une synthèse de la qualité des données à partir des résultats de leur évaluation :

- Evolution de l'indicateur de qualité
- Evolution de la qualité des données
- Qualité des données par dimension

Pour générer un rapport de qualité des données :

1. Cliquez sur le menu de navigation **Rapports**.
2. Dans la zone d'édition cliquez sur **Créer un rapport**.
3. Rechercher le rapport type voulu.
4. Passez la souris sur le rapport type et cliquez sur **Créer un rapport**.
5. Définissez les objets à analyser.

Voir aussi : [Dimensions de qualité](#).

RAPPORTS ET OUTILS D'ANALYSE DES DONNÉES



Hopex Data Governance offre différents types de rapport qui visent à analyser les données métier définies dans le référentiel.

➤ Pour plus de détails sur le fonctionnement des rapports, voir le chapitre "Générer des rapports" dans le guide **Hopex Common Features**.

➤ Les rapports sur les diagrammes disponibles en standard avec **Hopex** sont aussi accessibles avec **Hopex Data Governance**.

RAPPORTS D'ANALYSE

Les rapports d'analyse sont des rapports dynamiques qui permettent d'analyser sous différents axes les données du référentiel.

Accéder aux rapports d'Hopex Data Governance

Pour accéder à l'ensemble des rapports d'**HOPEX Data Governance** :

- 1 Cliquez sur le menu de navigation **Rapports**.

Vous accédez aux rapports sauvegardés et pouvez créer de nouveaux rapports à partir des rapports types fournis dans la solution.

Certains rapports d'analyse sont également embarqués sur les objets du référentiel. Ces rapports sont disponibles dans les propriétés de ces objets, sous la page **Rapports**.

Par exemple, les rapports visibles sous le menu **Rapports** qui sont propres à l'objet **Carte de domaines de données** sont également accessibles dans les propriétés d'une carte de domaines de données.

Ces rapports embarqués sont dynamiques ; ils se mettent à jour de façon automatique avec les caractéristiques de l'objet.

Rapports de description

Rapport de vue

Voir [Le rapport de vue](#).

Rapport de glossaire

Ce rapport prêt à l'emploi construit automatiquement le glossaire métier de termes provenant d'un ensemble de dictionnaires métier. Pour chaque terme, le glossaire affiche une liste des définitions associées avec leur texte, synonyme et liste de composants.

Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Commentaire
Liste des bibliothèques	Bibliothèque	Non obligatoire
Liste des dictionnaires métier	dictionnaire métier	Obligatoire si pas de Bibliothèque
Option exemple	oui ou non	Permet d'afficher les exemples d'informations métier
Montrer l'icône du type de composant	oui ou non	

Exemple de rapport

L'exemple ci-dessous permet de visualiser les termes issu du dictionnaire "Domaine médiathèque".

Domaine médiathèque	
Abonné médiathèque	Élément de 1. (Synonyms) Media library member (Hyponymes) Souscripteur Etats 1. (Synonyms) Media library member, Media library subscriber (Produits d'architecture définis) ☐ Abonné
Achat d'ouvrage	Événement 1. (Synonyms) Publication purchase
Adhérent médiathèque	Etats 1. (Synonyms) Media library member, Media library subscriber (Produits d'architecture définis) ☐ Abonné

Carte de domaines de données

Sur une carte de données, deux rapports types permettent de visualiser la hiérarchie des domaines qui la composent.

Carte proportionnelle de domaines de données

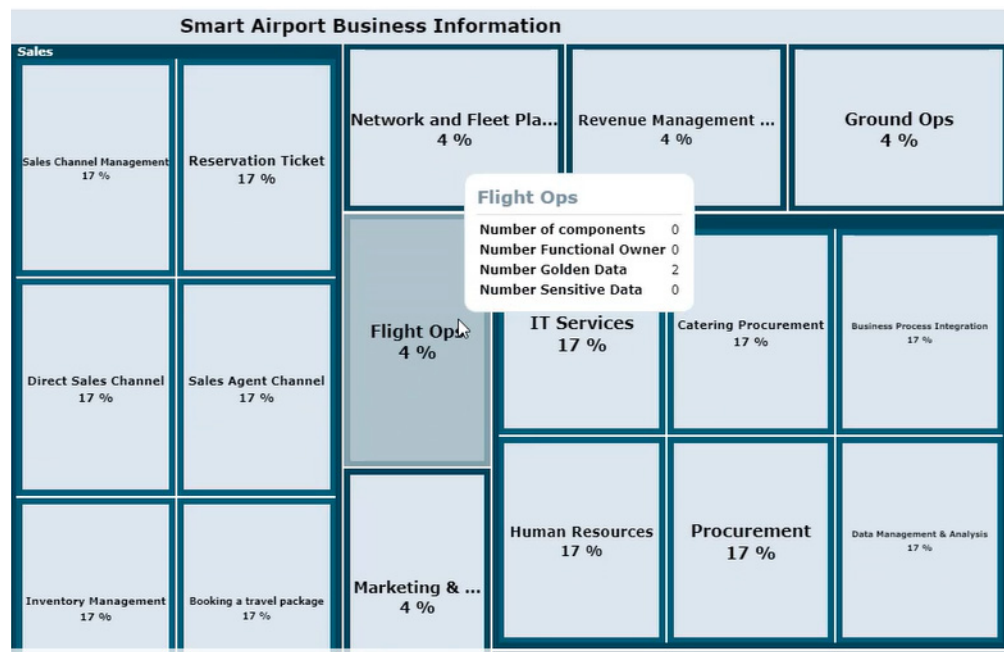
Dans ce rapport des filtres permettent de visualiser :

- le nombre de composants de chaque domaine
- le nombre de détenteurs fonctionnels



Le détenteur fonctionnel a une responsabilité sur l'usage d'une donnée dans un domaine métier. Vous pouvez spécifier les détenteurs fonctionnels des composants d'un domaine dans les propriétés du domaine en question, sous la section **Composants**. Voir [Définir les composants d'un domaine de concepts](#).

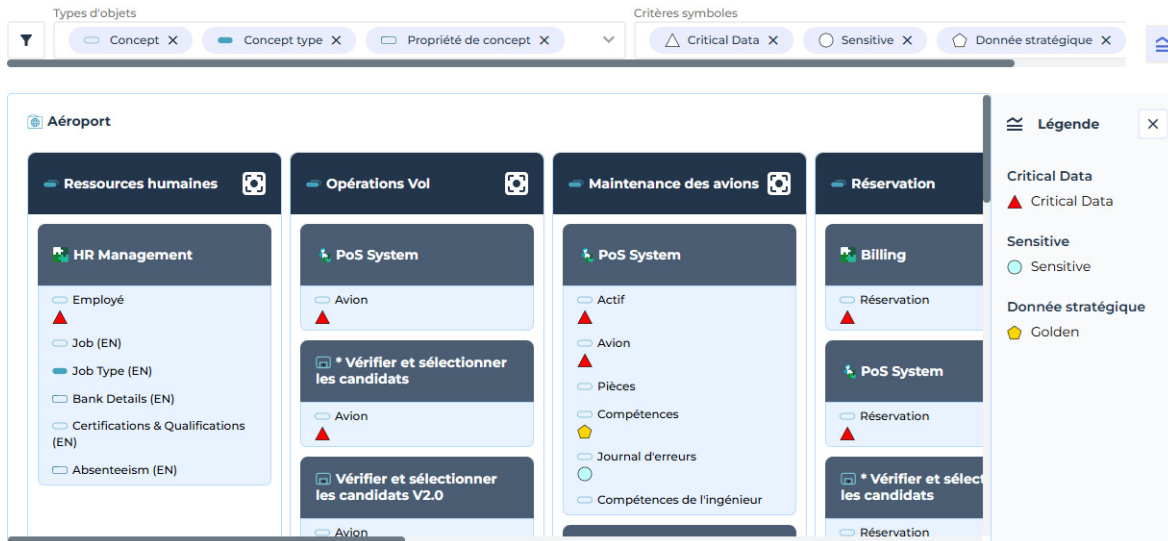
- le nombre de données de référence
- le nombre de données déclarées sensibles.



Rapport de décomposition d'une carte

Ce rapport affiche les domaines qui composent la carte, et dans chaque domaine :

- les applications et processus utilisateurs
- les données critiques
- les données sensibles
- les données stratégiques



Graphe de réalisation

Voir [Afficher le graphe de réalisation](#).

Dépendances entre domaines de données

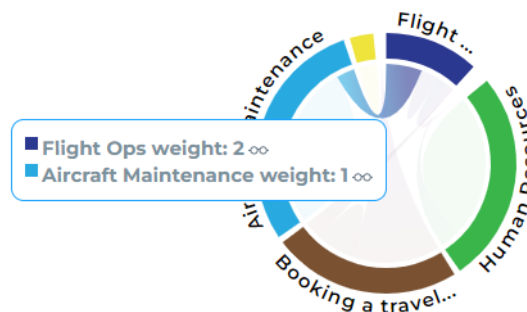
A partir d'une carte de données, ce rapport présente sous forme de graphique à cordes les dépendances structurales des données utilisées entre les domaines de données.

Il est possible de plonger sur un domaine de données et de visualiser les dépendances entre les sous-domaines lorsque celles-ci existent.



En sélectionnant un domaine, ses liens avec les autres domaines sont mis en évidence.

Ci-dessous vous pouvez voir les liens entre le domaine "Maintenance" et le domaine "Flight Ops".



Rapport de catalogue de données

Le rapport de catalogue de données permet d'analyser le contenu d'un ou de plusieurs catalogue(s) de données. Il présente les chapitres suivants :

- Organisation impliquée : entités qui utilisent ou consomment les données du catalogue.
- Textes de référence et cadres de politiques d'entreprise référencés
- Catégories/Sections
- Lignage de données. Voir aussi [Lignages de données](#).

Rapports nuages de mot

Nombre d'informations des domaines d'information

Ce rapport type porte sur les domaines d'un conteneur d'informations.

Dans le rapport généré, la taille des noms des domaines est proportionnelle au nombre d'informations qui les composent.

Etendue de la description de l'information

Ce rapport type porte sur les conteneurs d'informations (dictionnaire métier, dictionnaire de données logiques, base de données) et affiche les éléments correspondants : concepts, classes, tables, etc.

Dans le rapport, la taille des noms des éléments est proportionnelle au nombre d'informations qui les caractérisent (par exemple attributs et relations qui caractérisent une classe).

Utilisation des données dans les domaines de données

Ce rapport type porte sur les conteneurs d'informations (dictionnaire métier, dictionnaire de données logiques, base de données) et affiche les éléments correspondants : concepts, classes, tables, etc.

Dans le rapport, la taille des noms des éléments est proportionnelle à leur utilisation dans les domaines de données.

Rapports d'utilisation des données

Utilisation des informations détenues par un conteneur

Pour l'objet sélectionné en entrée (ex : un paquetage), le rapport affiche :

- les informations qu'il détient (ex : classes ou vues de données)
- les domaines qui utilisent ces informations, avec quels droits d'accès
- les applications qui utilisent les domaines (via les dépôts de données qui permettent de les déclarer sur les applications, systèmes applicatifs, service applicatif ou microservice), et à travers quels composants (en lecture seule ou lecture/écriture).

Un rapport type présente ces informations sous forme de dendrogramme, un autre rapport type sous forme de tableau.

Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Types d'objet
Sujet	Dictionnaire métier Base de données Paquetage Modèle de données Catalogue de brique de données NoSQL Paquetage de types de données

Utilisation des informations d'un domaine

Pour l'objet sélectionné en entrée (un domaine), le rapport affiche :

- les informations utilisées (ex : classes ou vues de données dans le cas d'un domaine de données applicatif), avec quels droits d'accès.
- dans quelles applications est utilisé le domaine sélectionné (via les dépôts de données qui permettent de les déclarer sur les applications, systèmes applicatifs, service applicatif ou microservice), et par quels composants (en lecture seule ou lecture/écriture).

Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Types d'objet
Sujet	Domaine de données applicatif Domaine de données logiques domaine de concepts Structure de fichier Domaine de données relationnelles Domaine de données NoSql

Un rapport type présente ces informations sous forme de dendrogramme, un autre rapport type sous forme de tableau.

Utilisation des informations d'une carte d'informations

Le paramètre d'entrée est une carte d'informations qui peut regrouper un ou plusieurs domaine(s).

Le rapport affiche

- les domaines et les données qu'ils utilisent, avec les droits d'accès à ces données.
- dans quels systèmes (applications, systèmes applicatifs, service applicatif ou microservice) les domaines sont utilisés et par quels composants de ces systèmes, en précisant le mode d'accès (lecture ou lecture/écriture).

Un rapport type présente ces informations sous forme de dendrogramme, un autre rapport type sous forme de tableau.

Utilisation de l'information

L'objet racine du rapport est une information (concept, classe, vue de données, table, etc.). Pour cet objet racine, le rapport affiche :

- les domaines qui utilisent l'information, avec quels droits d'accès
- dans quels systèmes (applications, systèmes applicatifs, service applicatif ou microservice) ces domaines sont utilisés et par quels composants de ces systèmes, en précisant le mode d'accès (lecture ou lecture/écriture).

Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Types d'objet
Sujet	Classe Concept Concept d'état Concept évènement Concept type Entité Période type Représentation type Table Type de données Vue de concept Vue de données Vue physique

Utilisation des informations des domaines d'un conteneur

Pour l'objet sélectionné en entrée (un conteneur), le rapport affiche :

- les domaines détenus
- les informations utilisées par les domaines, avec quels droits d'accès
- les systèmes qui utilisent ces domaines et le mode d'accès des composants de ces systèmes (lecture ou lecture/écriture)

Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètre	Types d'objet
Sujet	Dictionnaire métier Base de données Paquetage Modèle de données Catalogue de brique de données NoSQL Paquetage de types de données

Rapports réglementaires

Rapport des cadres réglementaires

Ce rapport affiche la liste des informations contraintes par un texte de référence (réglementation externe).

Les objets contraints sont ceux pour lesquels un texte de référence a été associé dans la page **Réglementation** de leurs propriétés (par exemple, un concept).

Vous pouvez voir les objets contraints par un texte de référence dans les propriétés du texte en question, dans la page **Éléments sujets**.

Le rapport affiche les articles et les sections de la réglementation et met en évidence ceux qui constituent des contraintes pour les données.

Parameters

Regulatory Framework

Basel II

Refresh the report

Regulation	Regulation Explanation	Constrained Data	Definition of Constrained Data	Is
¶ 115. The organization should implement robust procedures and policies to control residual risks.	Residual risk is the amount of risk that remains in the process after all the risks have been calculated, accounted, and hedged. During an investment or a business process, there are a lot of risks involved, and the entity takes into consideration all such risks	Potential loss	gradual decline in amount or activity	Data
¶ 163. The organization should have implemented processes to ensure it complies with all internal policies, procedures, and controls.	Internal policy means a Department statement that implements, interprets, or prescribes procedure or practice requirements for the internal operations of the Department	Customer Management		Data
¶ 165, ¶ 443, ¶ 620(7), ¶ 744. The internal auditing process should include an independent review of the risk measurement system. Internal audit must review the organization's rating system and operations at least annually. The internal audit should review the organization's adherence to applicable minimum requirements and document all of its findings. Internal or external auditors should regularly review the internal assessment process and the validity of the assessments.				Data

Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètre	Type d'objet
Liste de cadres réglementaires	Texte de référence

Conformité réglementaire et transformation des données

Ce rapport affiche sous forme de dendrogramme les données contraintes par des cadres réglementaires ou des cadres de politiques d'entreprise.

Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètre	Type d'objet
Policy Schema	Cadre de politique d'entreprise Texte de référence

Voir aussi : [Analyser les informations contraintes par une réglementation.](#)

Rapport de description des règles

Ce type de rapport permet de sélectionner un ensemble de règles (métier, opérationnelles ou système) et de visualiser sous forme de tableau les réglementations qu'elles mettent en œuvre.

Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètre	Type d'objet
Liste de règles	Règle métier Règle opérationnelle Règle système

Voir aussi : [Assurer la conformité des données : créer des règles métier.](#)

Rapports de qualité des données

Evolution de la qualité des données

Ce rapport affiche l'évolution de la qualité de données. Il s'appuie sur les évaluations effectuées sur les données en question.

Qualité des données par dimension

Ce rapport présente la qualité des données de domaines de données en fonction de dimensions telles que la cohérence, l'exactitude, etc.

Voir aussi : [Dimensions de qualité.](#)

REPORT DATASETS

Un Report DataSet est un tableau de données créé à partir d'objets du référentiel, sur lequel peuvent être générés des rapports instantanés.

Hopex Data Governance fournit différents types de Report DataSets.

Créer un Report DataSet dans Hopex Data Governance

Pour créer un Report DataSet :

1. Cliquez sur le menu de navigation puis sur **Rapports**.
2. Dans le volet de navigation cliquez sur **Autres rapports**.
3. Dans la zone d'édition cliquez sur **Mes Report DataSets**.
4. Cliquez sur **Nouveau**.
5. Indiquez :
 - le nom du rapport
 - le détenteur (optionnel)
 - la définition du Report DataSet, sur laquelle repose le rapport
6. Cliquez sur **OK**.

Exemple de Report Dataset

Définition des termes

Ce type de Report DataSet a comme paramètre d'entrée une liste de termes.

A partir des termes sélectionnés, vous pouvez par exemple créer un rapport instantané de type "matrice" qui présente la liste des concepts qui utilisent les termes en question.

The screenshot displays the 'Propriétés de Report DataSet-3' window. On the left, the 'Matrice' tab is active, showing a matrix with rows for terms like 'Book subscription (EN)', 'Work (EN)', 'compte cible', and 'Virement Planifié', and columns for concepts like 'Ville', 'Anniversaire', 'Abandonment subscription', 'Work', 'Abandon inscription', 'compte cible', 'Available Type of Loan', and 'Virement Planifié'. The matrix shows counts for each intersection. On the right, the 'Données' tab is active, showing a list of terms and a table of report data.

Term	Concept
Abandon inscription	
Abandonment subscription	
Available Type of Loan	
Birthday	
Book subscription	

Pour créer un rapport instantané sur ce type de Report DataSet :

1. Créez un Report DataSet de type "Définition de terme".
Voir plus haut [Créer un Report DataSet dans Hopex Data Governance](#).
2. Ouvrez les propriétés du Report DataSet.
3. Affichez la page **Données**.
4. (Si besoin) Dans la section **Paramètres**, cliquez sur **Ajouter** et sélectionnez les paramètres d'entrée, ici des termes.
5. (Si besoin) Dans la section **Report DataSet**, cliquez sur **Rafraîchir**.
6. Dans la section **Report DataSet**, cliquez sur **Rapport instantané**.
7. Sélectionnez le type de rapport instantané voulu, ici une matrice.
8. Cliquez sur **OK**.

Matrice Dictionnaire métier x Concept

Un concept peut être référencé par un ou plusieurs dictionnaires métier.

Ce Report DataSet a pour entrée une liste de dictionnaires métier. Il permet de créer un rapport instantané de type "Matrice" qui dresse la liste des concepts référencés dans les dictionnaires métier sélectionnés.

Pour créer un rapport instantané sur ce type de Report DataSet :

1. Créez un Report DataSet de type "Matrice Dictionnaire métier x Concepts".
Voir plus haut [Créer un Report DataSet dans Hopex Data Governance](#).

2. Ouvrez les propriétés du Report DataSet.
3. Affichez la page **Données**.
4. (Si besoin) Dans la section **Paramètres**, cliquez sur **Ajouter** et sélectionnez les paramètres d'entrée, ici des dictionnaires métier.
5. (Si besoin) Dans la section **Report DataSet**, cliquez sur **Rafraîchir**.
6. Dans la section **Report DataSet**, cliquez sur **Rapport instantané**.
7. Sélectionnez le type de rapport instantané voulu, ici une matrice.
8. Cliquez sur **OK**.

WORKFLOW DE VALIDATION DES DONNÉES

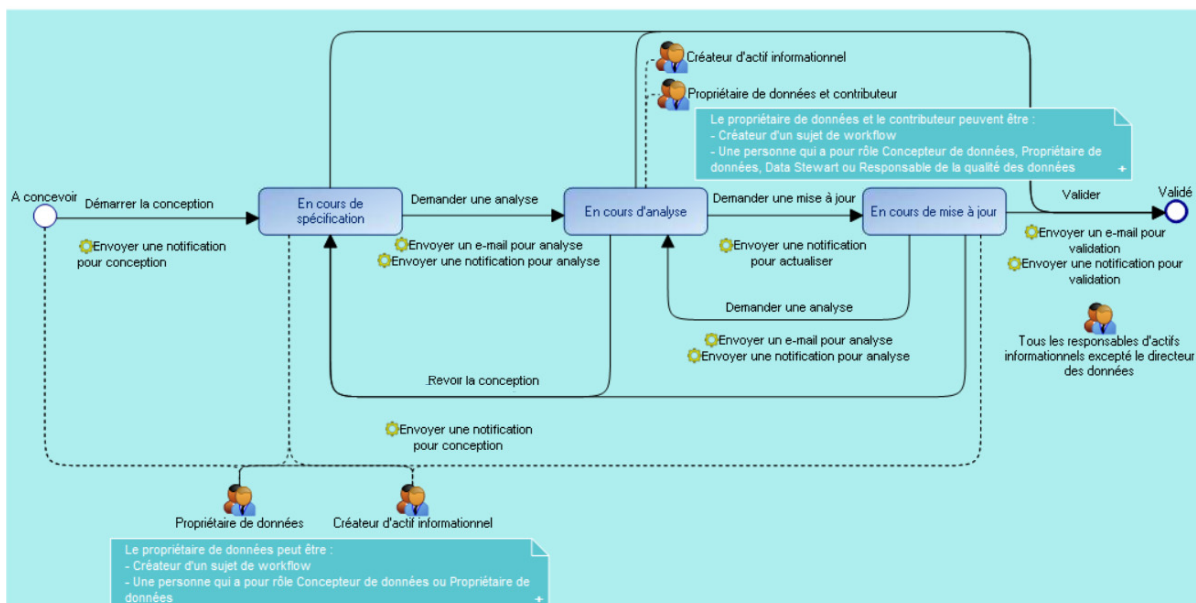
Hopex Data Governance et **Hopex Data Architecture** comprennent un workflow standard pour gérer la progression de la conception d'actifs informationnels, de leur création jusqu'à la fin de leur validation.

Une définition du workflow est fournie par défaut pour le lignage de données, le domaine de données et la carte de données, des couches métier, logique et physique.

Le workflow peut être étendu à d'autres éléments d'information.

Par défaut, l'instanciation de ce workflow est automatique lors de la création d'un lignage de données. Elle est optionnelle et non automatique pour les domaines et les cartes de données. Vous pouvez sélectionner et décider pour quel objet vous souhaitez créer une instance de workflow.

Etapes du workflow de validation



L'instanciation du workflow peut être effectuée par le concepteur de données ou le propriétaire des données de l'objet. De plus, toutes les autres personnes assignées en tant que Data Asset Manager, Data Steward, Data Quality Manager et Data Chef Officer peuvent utiliser le profil "Contributeur de données" pour se connecter aux données d'HOPEX et déclencher les étapes de validation.

Pour chaque transition déclenchée, une notification ou un e-mail est envoyé à la personne assignée sur le sujet du workflow.

Pour plus de détails sur les workflows voir [Utiliser les workflows](#).

Pour plus de détails sur les rôles métier voir [Les rôles métier de HOPEX Information Architecture](#).

Générer un rapport de workflow

Des rapports de workflow sont disponibles sur les différents types d'objets (lignage de données, domaines de données, etc.); ils affichent le nombre d'objets situés à chaque étape du workflow (nombre d'objets en cours de conception, en cours d'analyse, etc).

Pour générer un rapport de workflow :

1. Cliquez sur le menu de navigation puis sur **Tableau de bord**.
2. Cliquez sur le + pour ajouter un rapport.
3. Dépliez le dossier **Architecture d'information** puis le sous-dossier **Statuts de conception**.
4. Sélectionnez le rapport voulu.
Le rapport apparaît dans le tableau de bord.

IMPORT ET EXPORT DE DONNÉES



Hopex Data Governance et **Hopex Data Architecture** fournissent des modèles de fichier Excel pour importer dans le référentiel Hopex des données métier et logiques existantes, à partir desquelles vous pouvez construire des dictionnaires de données de façon automatique. Ces modèles vous permettent également d'exporter des données du référentiel Hopex.

Un modèle dédié à la qualité des données permet de collecter des mesures d'évaluation des critères de la qualité des données (Complétude, Unicité, Fraîcheur, etc.).

- ✓ [Importer des données métier à partir d'un fichier Excel](#)
- ✓ [Importer des données logiques à partir d'un fichier Excel](#)
- ✓ [Importer des évaluations de données](#)

IMPORTER DES DONNÉES MÉTIER À PARTIR D'UN FICHIER EXCEL

Un modèle de fichier Excel vous permet d'importer dans le référentiel Hopex un ensemble d'informations métier existantes.

Vous pouvez utiliser le modèle pour importer simplement une liste de termes et leur définition, en vue de générer un glossaire, ou pour une description d'informations métier plus détaillée, avec la possibilité de définir les relations entre les concepts, leurs synonymes, etc.

Vous pouvez aussi utiliser ce modèle pour exporter des informations métier du référentiel.

Télécharger le modèle de fichier Excel

Pour télécharger le modèle Excel associé aux informations métier :

1. Dans le bureau, cliquez sur **Menu principal > Export > Excel**.
2. Dans l'assistant qui apparaît, cochez l'option "A partir d'un modèle".
3. Cliquez sur **Suivant**.
4. Dans le champ **Fichier template prédéfini**, sélectionnez "Concept Template".
5. Cliquez deux fois sur **Suivant** et sauvegardez le fichier créé. Il contient la structure fournie par le modèle.

Contenu du modèle Excel

Le modèle contient les feuilles suivantes qui interagissent entre elles :

- Concepts
- Properties

Feuille Concepts

La feuille **Concepts** permet de créer dans HOPEX les concepts qui définissent les termes du glossaire.

Dans la feuille, le nom du concept est obligatoire. Les autres colonnes sont facultatives, elles peuvent être utilisées pour organiser les concepts dans les dictionnaires métier et pour relier les concepts à des propriétés de concepts, des domaines utilisateurs et des états de concept.

Pour saisir plusieurs objets dans une cellule, mettez un objet par ligne (séparez-les par un retour chariot).

Certaines colonnes sont multiples : Propriétés, Catégories de données, Étiquette, Domaines du concept, Synonymes, Hypernymes, Concepts liés, Concepts d'état.

Business Dictionary

Dans cette colonne vous pouvez indiquer le nom du dictionnaire métier dans lequel vont être créés les Concepts importés. Cette colonne est optionnelle.

Definition

Cette propriété contient la définition du terme dont est issu le concept.

Exemple de concepts et définitions :

Business Dictionary	Concept Name	Definition	Properties
D81D0FEC64553EDF	D81D10BF64553F7F	D81D10D464553FB0	D81D10F164553FE1
Order Management	Purchase Order	a commercial document used to request someone to supply something in return for payment and providing specifications and quantities	Order Number, Order Date, Order Type, Delivery Address
Order Management	Order Item	An order item refers to a specific product or service that is included as part of an order placed by a customer with a business or supplier. When a customer makes a purchase or requests a service, they typically provide the details of the items they want to acquire or receive, and these individual items are referred to as order items.	Product Quantity
Order Management	Customer	A customer is an individual, organization, or entity that purchases or consumes goods, products, or services from a business or supplier. In the context of commerce and business, customers play a crucial role as they are the ultimate consumers or clients who drive revenue and sustain a company's operations.	Name, Email Address, Delivery Address 1, Delivery Address 2,

Properties

Cette colonne énumère les propriétés du concept créé. Chaque propriété est créée en tant que "propriété du concept" du concept créé.

Data Categories

Cette propriété indique la catégorie de la donnée (exemple : donnée sensible). Il est possible d'indiquer plusieurs catégories. Si la catégorie n'existe pas, elle est créée.

Tags

Vous pouvez saisir un ou plusieurs tags pour les concepts créés.

Concept Domains

Pour chaque concept, vous pouvez indiquer le ou les domaines de concepts qui l'utiliseront.

Synonyms

Saisissez un ou plusieurs noms de termes métier qui seront spécifiés comme synonymes du terme correspondant au concept créé.

Hypernyms

Saisissez le nom d'un concept qui sera spécifié comme l'hyperonyme du concept créé.

Linked Concepts

Saisissez la liste d'un ou de plusieurs concepts à connecter au concept créé.

State Concepts

Vous pouvez définir ou plusieurs concepts d'état pour le concept créé.

Feuille Properties

La feuille **Properties** contient les propriétés des concepts associés aux termes.

Business Dictionary

Dans cette colonne vous devez indiquer le nom du dictionnaire métier dans lequel vont être créés les propriétés importées.

Concept Property Name

Ce nom est obligatoire. Il est utilisé pour créer la propriété de concept. Il correspond à un terme métier dans le glossaire. La propriété de concept créée désigne le terme et porte sa définition.

Definition

Définition de la propriété de concept créée et du terme correspondant.

Sub Property

Cette colonne énumère les propriétés de la propriété de concept créée. Chaque "sous-propriété" est créée en tant que "propriété de concept" de la propriété de concept créée.

Data Categories

Cette propriété indique la catégorie de donnée de la propriété de concept.

Tags

Vous pouvez saisir un ou plusieurs tags pour les propriétés de concepts créées.

Concept Domains

Pour chaque propriété de concept, vous pouvez indiquer le ou les domaines de concepts qui l'utiliseront.

Synonyms

Saisissez un ou plusieurs noms de termes métier qui seront spécifiés comme synonymes du terme correspondant à la propriété de concept créée.

Hypernyms

Saisissez le nom de la propriété de concept qui sera spécifiée comme l'hyperonyme de la propriété de concept créée.

Voir aussi : [Importer des données logiques à partir d'un fichier Excel](#).

IMPORTER DES DONNÉES LOGIQUES À PARTIR D'UN FICHIER EXCEL

HOPEX Data Governance et **Hopex Data Architecture** fournissent un modèle de fichier Excel pour importer dans le référentiel Hopex un ensemble de données logiques existantes. Vous pouvez aussi utiliser ce modèle pour exporter des données du référentiel.

Télécharger le modèle de fichier Excel

Pour télécharger le modèle Excel associé aux données logiques :

1. Dans le bureau, cliquez sur **Menu principal** > **Export** > **Excel**.
2. Dans l'assistant qui apparaît, cochez l'option "A partir d'un modèle".
3. Cliquez sur **Suivant**.
4. Dans le champ **Fichier template prédéfini**, sélectionnez "Data Excel Template".
5. Cliquez deux fois sur **Suivant** et sauvegardez le fichier créé. Il contient la structure fournie par le modèle.

Contenu du modèle Excel

Le modèle contient les feuilles suivantes.

Feuille Data Dictionary

La feuille **Data Dictionary** permet d'importer un ensemble de dictionnaires de données avec leur nom et leur propriétaire.

Data Dictionary Short Name

Nom du dictionnaire de données. Cette propriété est obligatoire.

Data Dictionary ID

L'ID permet d'identifier le dictionnaire de données, dans le cas où plusieurs dictionnaires auraient le même nom. Cette propriété est optionnelle.

Data Dictionary Owner Name

Nom du dictionnaire de données détenteur. Cette propriété est optionnelle.

Data Dictionary Owner ID

L'ID permet d'identifier le dictionnaire de données détenteur, dans le cas où plusieurs auraient le même nom. Cette propriété est optionnelle.

Comment

Commentaire du dictionnaire de données. Cette propriété est optionnelle.

Feuille Data Type

La feuille **Data Type** définit les types de données à importer.

Data Type Short Name

Nom du dictionnaire de données. Cette propriété est obligatoire.

Data Type ID

L'ID permet d'identifier le type de données, dans le cas où plusieurs auraient le même nom. Cette propriété est optionnelle.

Data Type Package Name

Nom du paquetage détenteur du type de données. Cette propriété est optionnelle.

En cas de hiérarchie dans la détention, la syntaxe du nom est la suivante : <Nom du paquetage détenteur 1>::<Nom du paquetage détenteur 2>

Exemple :

```
Standard::Types::Data Types Reference
```

Data Type Package ID

L'ID permet d'identifier le paquetage du type de données, dans le cas où plusieurs auraient le même nom.

Length

Longueur du type de données.

Decimal

Valeur "Decimal".

SQL Data Type

Vous pouvez associer un type standard au type de données à importer, par exemple "P-Character".

Comment

Commentaire du type de données.

Feuille Data Type Component

La feuille **Data Type Component** définit les attributs des types de données définis dans la feuille **Data Type**.

Attribute Short Name

Nom de l'attribut du type de données. Cette propriété est obligatoire.

Exemple : "Number".

Owner Data Type Name

Nom du type de données détenteur. Cette propriété est obligatoire si l'ID du type de données détenteur n'est pas spécifié.

Exemple : "Address".

Owner Data Type ID

ID du type de données détenteur. Cette propriété est obligatoire si le nom du type de données n'est pas spécifié.

Data Type Name

Nom du type de données de l'attribut.

Exemple : "Number".

Data Type ID

ID du type de données de l'attribut.

Length

Longueur du type de données de l'attribut.

Decimal

Valeur "Decimal".

Comment

Commentaire de l'attribut du type de données.

Feuille Class

La feuille **Class** permet de définir les classes à importer. Saisissez le nom de la classe. Saisissez l'ID des classes lorsque différentes classes portent le même nom.

Class Short Name

Nom de la classe. Cette propriété est obligatoire.

Class ID

L'ID permet d'identifier la classe, dans le cas où plusieurs auraient le même nom. Cette propriété est optionnelle.

Data Dictionary Name

Nom du dictionnaire de données détenteur de la classe. Cette propriété est optionnelle.

Data Category

Vous pouvez donner une classification à la classe, par exemple "donnée sensible", "donnée de référence", etc.

Comment

Commentaire de la classe.

Feuille Attribute

La feuille **Attribute** définit les attributs des classes définies dans la feuille **Class**.

Laissez la première colonne vide si vous voulez créer de nouvelles occurrences, ou remplissez-la avec l'Identifiant Absolu Hopex si vous voulez mettre à jour des données existantes.

Short Name

Nom de l'attribut.

Class Name

Nom de la classe détentrice. Le nom ou l'ID de la classe est obligatoire.

Class ID

ID de la classe détentrice. Le nom ou l'ID de la classe est obligatoire.

Length

Longueur de l'attribut.

Decimal

Valeur "Decimal".

Comment

Commentaire de l'attribut.

Feuille Relationship

Part Name

Nom de la partie.

Owner Class Name

Nom de la classe détentrice. Le nom ou l'ID de la classe est obligatoire.

Owner Classe ID

ID de la classe détentrice. Le nom ou l'ID de la classe est obligatoire.

Referenced Class Name

Nom de la classe référencée. Le nom ou l'ID de la classe référencée est obligatoire.

Referenced Class ID

ID de la classe référencée. Le nom ou l'ID de la classe est obligatoire.

Multiplicity

Multiplicité de la partie.

Feuille Generalization

Generalization Name

Nom de la partie.

Super Class Name

Nom de la classe générale. Le nom ou l'ID de la classe est obligatoire.

Super Classe ID

ID de la classe générale. Le nom ou l'ID de la classe est obligatoire.

Sub Class Name

Nom de la sous-classe. Le nom ou l'ID de la classe est obligatoire.

Sub Class ID

ID de la sous-classe. Le nom ou l'ID de la classe est obligatoire.

IMPORTER DES ÉVALUATIONS DE DONNÉES

Hopex fournit un modèle Excel pour importer dans votre référentiel des valeurs de critères de qualité de données.

Ce modèle Excel est mis en correspondance avec le modèle d'évaluation de la qualité des données implanté dans l'outil d'évaluation directe de **Hopex Data Architecture**.

Il peut être utilisé pour différents types d'objets qui ont une évaluation directe, par exemple les concepts, les classes, les tables, etc.

Exemple d'import

Ci-dessous l'exemple d'import des différentes valeurs de qualité d'un concept.

Valeurs définies dans le fichier Excel

	A	B	C	D	E	F	G	H	I	J
1	Hopex Identifier	Object Type	Object name	Completeness	Uniqueness	Timeliness	Validity	Accuracy	Consistency	Assessor Name
2		Concept	Artifact	Very Low	Low	Medium	Very High	High	Medium	NOURY Sam
3										
4										

Résultat dans Hopex Data Architecture

Artifact (EN)

| Evaluation

Evaluation directe

+ Evaluer

✖ Afficher les réponses

X

☐	Nœud d'évaluation	Date d'évaluat...	Exhaustivité	Unicité	Fraicheur	Validité	Exactit...	Consistance
☐	 Artifact (EN) (Concept)	21/08/2020	Very Low	Low	Medium	Very High	High	Medium

Contenu du modèle Excel

Colonnes	Description
HOPEX Identifier	_IdAbs d'un objet évalué. Lorsque cette propriété est spécifiée, les colonnes Object Type et Nom ne sont pas obligatoires.
Object Type	Cette propriété indique le type d'objet évalué : "Concept", "Vue du concept", "Classe", "Vue de données", "Table". Cette propriété doit être définie lorsque l'identifiant HOPEX n'est pas spécifié.
Object Name	Cette propriété indique le nom long de l'objet évalué. Elle doit être définie lorsque l'identifiant HOPEX n'est pas spécifié.
Completeness	Correspond au critère d'évaluation "Complétude"
Uniqueness	Correspond au critère d'évaluation "Unicité"
Timeliness	Correspond au critère d'évaluation "Fraîcheur"
Validity	Correspond au critère d'évaluation "Validité"
Accuracy	Correspond au critère d'évaluation "Exactitude"
Consistency	Correspond au critère d'évaluation "Cohérence"
Assessor Name	Nom de l'évaluateur. Si aucun évaluateur n'est indiqué, le nom de l'utilisateur courant est pris par défaut dans HOPEX.
Evaluation Date	Date d'évaluation de l'objet. Pour un même objet, des mesures peuvent être effectuées à différents moments.

Télécharger le modèle Excel

Le modèle **Data Quality Excel Template** est disponible dans l'HOPEX Store de MEGA. Pour le télécharger :

1. Rendez-vous sur l'HOPEX Store, à l'adresse suivante : <https://community.mega.com/t5/HOPEX-Store/bd-p/hopex-store>.
2. Cliquez sur **Data Quality Excel Template**.
3. Téléchargez le fichier.

Importer un fichier Excel d'évaluations de données

Pour importer des évaluations de données à partir d'un fichier Excel :

1. Cliquez sur le **Menu principal** puis sur **Import > Excel (*.xls; *.xlsx)**.
2. A droite du champ **Fichier d'import Excel**, cliquez sur **Parcourir**.

3. Sélectionnez le fichier qui contient les évaluations.
4. Cliquez sur **Import**.



Appendix – Data Discovery Connectors

Contents

INTRODUCTION	18
1.1. Prerequisite for using HOPEX Data Discovery	18
1.2. Connection issues	19
2. ACCESS 	20
2.1. Access Version Support	20
2.2. Connection Options	20
2.2.1. Database	20
2.2.2. Logging	20
2.2.3. Schema	20
2.2.4. Miscellaneous	21
3. AMAZON ATHENA 	22
3.1. Connection Options	22
3.1.1. Authentication	22
3.1.2. Connection	22
3.1.3. AWS Authentication	23
3.1.4. SSO	24
3.1.5. OAuth	24
3.1.6. SSL	25
3.1.7. Firewall	25
3.1.8. Proxy	25
3.1.9. Logging	26
3.1.10. Schema	26
3.1.11. Caching	27
3.1.12. Miscellaneous	27
4. AMAZON DYNAMODB 	30
4.1. Amazon DynamoDB Version Support	30
4.2. Connection Options	30
4.2.1. AWS Authentication	30
4.2.2. SSO	31
4.2.3. SSL	31
4.2.4. Firewall	32
4.2.5. Proxy	32
4.2.6. Logging	33
4.2.7. Schema	33
4.2.8. Caching	34
4.2.9. Miscellaneous	34
5. AZURE ANALYSIS SERVICES 	38
5.1. Connection Options	38
5.1.1. Authentication	38
5.1.2. Azure Authentication	38
5.1.3. OAuth	39
5.1.4. SSL	40

5.1.5.	Firewall	40
5.1.6.	Proxy	41
5.1.7.	Logging.....	41
5.1.8.	Schema	42
5.1.9.	Caching	43
5.1.10.	Miscellaneous	43

6. AZURE DATA CATALOG

6.1.	Connection Options	46
6.1.1.	Authentication	46
6.1.2.	Azure Authentication.....	46
6.1.3.	OAuth.....	46
6.1.4.	SSL.....	47
6.1.5.	Firewall	48
6.1.6.	Proxy	48
6.1.7.	Logging.....	49
6.1.8.	Schema	49
6.1.9.	Caching	50
6.1.10.	Miscellaneous	50

7. AZURE SYNAPSE

7.1.	Connection Options	52
7.1.1.	Authentication	52
7.1.2.	Bulk	52
7.1.3.	Azure Authentication.....	53
7.1.4.	OAuth.....	53
7.1.5.	SSL.....	54
7.1.6.	Firewall	54
7.1.7.	Logging.....	54
7.1.8.	Schema	55
7.1.9.	Miscellaneous	55

8. AZURE TABLE

8.1.	Connection Options	57
8.1.1.	Authentication	57
8.1.2.	SSL.....	57
8.1.3.	Firewall	57
8.1.4.	Proxy	58
8.1.5.	Logging.....	59
8.1.6.	Schema	59
8.1.7.	Caching	60
8.1.8.	Miscellaneous	60

9. BIGQUERY

9.1.	Connection Options	62
9.1.1.	Authentication	62
9.1.2.	BigQuery	62
9.1.3.	Storage API	63

9.1.4.	Uploading	63
9.1.5.	OAuth.....	64
9.1.6.	JWT OAuth	64
9.1.7.	SSL.....	65
9.1.8.	Firewall	65
9.1.9.	Proxy	66
9.1.10.	Logging.....	66
9.1.11.	Schema	67
9.1.12.	Caching	67
9.1.13.	Miscellaneous	68

10. CASSANDRA 71

10.1.	Cassandra Version Support	71
10.2.	Connection Options	71
10.2.1.	Authentication	71
10.2.2.	Kerberos	72
10.2.3.	SSL.....	72
10.2.4.	SSH.....	73
10.2.5.	Firewall	74
10.2.6.	Logging.....	74
10.2.7.	Schema	75
10.2.8.	Caching	75
10.2.9.	Miscellaneous	76

11. CLOUDANT 79

11.1.	Connection Options	79
11.1.1.	Authentication	79
11.1.2.	OAuth.....	79
11.1.3.	SSL.....	80
11.1.4.	Firewall	80
11.1.5.	Proxy	80
11.1.6.	Logging.....	81
11.1.7.	Schema	81
11.1.8.	Caching	82
11.1.9.	Miscellaneous	83

12. COCKROACHDB 85

12.1.	CockroachDB Version Support	85
12.2.	Connection Options	85
12.2.1.	Authentication	85
12.2.2.	SSL.....	85
12.2.3.	Firewall	86
12.2.4.	Logging.....	86
12.2.5.	Schema	87
12.2.6.	Miscellaneous	87

13. CONFLUENCE 89

13.1.	Connection Options	89

13.1.1.	Authentication	89
13.1.2.	SSO	89
13.1.3.	OAuth	90
13.1.4.	SSL	91
13.1.5.	Firewall	91
13.1.6.	Proxy	91
13.1.7.	Logging	92
13.1.8.	Schema	93
13.1.9.	Caching	93
13.1.10.	Miscellaneous	94

14. COSMOS DB 96

14.1.	Connection Options	96
14.1.1.	Authentication	96
14.1.2.	Azure Authentication	96
14.1.3.	OAuth	97
14.1.4.	SSL	98
14.1.5.	Firewall	98
14.1.6.	Proxy	99
14.1.7.	Logging	99
14.1.8.	Schema	100
14.1.9.	Caching	100
14.1.10.	Miscellaneous	101

15. COUCHBASE 104

15.1.	Connection Options	104
15.1.1.	Authentication	104
15.1.2.	SSL	105
15.1.3.	Firewall	105
15.1.4.	Proxy	106
15.1.5.	Logging	106
15.1.6.	Schema	107
15.1.7.	Caching	108
15.1.8.	Miscellaneous	109

16. CSV 112

16.1.	Connection Options	112
16.1.1.	Authentication	112
16.1.2.	Connection	112
16.1.3.	AWS Authentication	113
16.1.4.	Azure Authentication	114
16.1.5.	SSO	114
16.1.6.	OAuth	114
16.1.7.	JWT OAuth	116
16.1.8.	Kerberos	116
16.1.9.	SSL	117
16.1.10.	SSH	118
16.1.11.	Firewall	118
16.1.12.	Proxy	119

16.1.13. Logging.....	119
16.1.14. Schema	120
16.1.15. Caching	121
16.1.16. Data Formatting.....	121
16.1.17. Miscellaneous	123

17. DATABRICKS 126

17.1. Connection Options	126
17.1.1. Authentication	126
17.1.2. AWS Authentication	126
17.1.3. Azure Authentication.....	127
17.1.4. AzureServicePrincipal Authentication	127
17.1.5. SSL.....	128
17.1.6. Firewall	128
17.1.7. Proxy	129
17.1.8. Logging.....	129
17.1.9. Schema	130
17.1.10. Caching	130
17.1.11. Databricks	131
17.1.12. Miscellaneous	132

18. DYNAMICS 365 134

18.1. Microsoft Dynamics 365 Version Support	134
18.2. Connection Options	134
18.2.1. Authentication	134
18.2.2. Azure Authentication.....	134
18.2.3. OAuth.....	135
18.2.4. JWT OAuth	136
18.2.5. SSL.....	136
18.2.6. Firewall	136
18.2.7. Proxy	137
18.2.8. Logging.....	138
18.2.9. Schema	138
18.2.10. Caching	139
18.2.11. Miscellaneous	139

19. ELASTICSEARCH 143

19.1. Elasticsearch Version Support	143
19.2. Connection Options	143
19.2.1. Authentication	143
19.2.2. Connection.....	143
19.2.3. AWS Authentication	144
19.2.4. Kerberos	144
19.2.5. SSL.....	145
19.2.6. Firewall	145
19.2.7. Proxy	146
19.2.8. Logging.....	146
19.2.9. Schema	147

19.2.10. Caching	147
19.2.11. Miscellaneous	148

20. EXCEL 150

20.1. Excel Version Support	150
20.2. Connection Options	150
20.2.1. Authentication	150
20.2.2. Connection	150
20.2.3. AWS Authentication	151
20.2.4. Azure Authentication	152
20.2.5. SSO	152
20.2.6. Data	153
20.2.7. OAuth	153
20.2.8. JWT OAuth	155
20.2.9. Kerberos	155
20.2.10. SSL	156
20.2.11. SSH	156
20.2.12. Firewall	157
20.2.13. Proxy	157
20.2.14. Logging	158
20.2.15. Schema	158
20.2.16. Caching	159
20.2.17. Miscellaneous	159

21. FINANCIALFORCE 162

21.1. Connection Options	162
21.1.1. Authentication	162
21.1.2. Connection	162
21.1.3. SSO	162
21.1.4. BulkAPI	163
21.1.5. OAuth	163
21.1.6. SSL	164
21.1.7. Firewall	165
21.1.8. Proxy	165
21.1.9. Logging	166
21.1.10. Schema	166
21.1.11. Caching	167
21.1.12. Miscellaneous	167

22. EXCEL ONLINE 170

22.1. Connection Options	170
22.1.1. Authentication	170
22.1.2. Connection	170
22.1.3. Azure Authentication	171
22.1.4. OAuth	171
22.1.5. SSL	172
22.1.6. Firewall	172
22.1.7. Proxy	173
22.1.8. Logging	173

22.1.9.	Schema	174
22.1.10.	Caching	174
22.1.11.	Miscellaneous	175

23. **GOOGLE DATA CATALOG** 178

23.1.	Connection Options	178
23.1.1.	Authentication	178
23.1.2.	OAuth.....	178
23.1.3.	JWT OAuth	179
23.1.4.	SSL.....	180
23.1.5.	Firewall	180
23.1.6.	Proxy	180
23.1.7.	Logging.....	181
23.1.8.	Schema	182
23.1.9.	Caching	182
23.1.10.	Miscellaneous	183

24. **GOOGLE SPANNER** 185

24.1.	Connection Options	185
24.1.1.	Authentication	185
24.1.2.	OAuth.....	185
24.1.3.	JWT OAuth	186
24.1.4.	SSL.....	187
24.1.5.	Firewall	187
24.1.6.	Proxy	187
24.1.7.	Logging.....	188
24.1.8.	Schema	189
24.1.9.	Caching	189
24.1.10.	Miscellaneous	190

25. **HBASE** 192

25.1.	Apache HBase Version Support	192
25.2.	Connection Options	192
25.2.1.	Authentication	192
25.2.2.	Kerberos	192
25.2.3.	SSL.....	193
25.2.4.	Firewall	193
25.2.5.	Proxy	194
25.2.6.	Logging.....	194
25.2.7.	Schema	195
25.2.8.	Caching	196
25.2.9.	Miscellaneous	196

26. **HIVE** 199

26.1.	Apache Hive Version Support	199
26.2.	Connection Options	199
26.2.1.	Authentication	199
26.2.2.	Kerberos	199
26.2.3.	SSL.....	200

26.2.4.	SSH	201
26.2.5.	Firewall	201
26.2.6.	Proxy	202
26.2.7.	Logging.....	203
26.2.8.	Schema	203
26.2.9.	Caching	204
26.2.10.	Miscellaneous	204

27. IBM DB2 207

27.1.	Connection Options	207
27.1.1.	Authentication	207
27.1.2.	SSL.....	207
27.1.3.	SSH.....	208
27.1.4.	Firewall	208
27.1.5.	Logging.....	209
27.1.6.	Schema	209
27.1.7.	Miscellaneous	210

28. IBM DB2 FOR Z/OS 212

28.1.	Prerequisite	212
28.2.	Connection Options	213

29. IBM INFORMIX 216

29.1.	Informix Version Support	216
29.2.	Connection Options	216
29.2.1.	Authentication	216
29.2.2.	SSL.....	217
29.2.3.	SSH.....	217
29.2.4.	Firewall	218
29.2.5.	Logging.....	218
29.2.6.	Schema	219
29.2.7.	Miscellaneous	219

30. IMPALA 221

30.1.	Connection Options	221
30.1.1.	Authentication	221
30.1.2.	Kerberos	221
30.1.3.	SSL.....	222
30.1.4.	Firewall	223
30.1.5.	Proxy	223
30.1.6.	Logging.....	224
30.1.7.	Schema	224
30.1.8.	Caching	225
30.1.9.	Miscellaneous	225

31. JIRA 228

31.1.	Connection Options	228
31.1.1.	Authentication	228
31.1.2.	SSO.....	228

31.1.3.	OAuth.....	229
31.1.4.	SSL.....	230
31.1.5.	Firewall	230
31.1.6.	Proxy	231
31.1.7.	Logging.....	231
31.1.8.	Schema	232
31.1.9.	Caching	232
31.1.10.	Miscellaneous.....	233

32. JSON 236

32.1.	Connection Options	236
32.1.1.	Authentication	236
32.1.2.	Connection.....	236
32.1.3.	AWS Authentication	237
32.1.4.	Azure Authentication.....	238
32.1.5.	SSO.....	238
32.1.6.	OAuth.....	239
32.1.7.	JWT OAuth	240
32.1.8.	Kerberos	241
32.1.9.	SSL.....	242
32.1.10.	SSH.....	242
32.1.11.	Firewall	243
32.1.12.	Proxy	243
32.1.13.	Logging.....	244
32.1.14.	Schema	244
32.1.15.	Caching	245
32.1.16.	Miscellaneous.....	246

33. KAFKA 249

33.1.	Connection Options	249
33.1.1.	Authentication	249
33.1.2.	Connection.....	249
33.1.3.	Kerberos	250
33.1.4.	SSL.....	250
33.1.5.	SchemaRegistry	251
33.1.6.	Firewall	252
33.1.7.	Proxy	252
33.1.8.	Logging.....	253
33.1.9.	Schema	254
33.1.10.	Caching	254
33.1.11.	Miscellaneous.....	255

34. MARIADB 258

34.1.	MariaDB Version Support.....	258
34.2.	Connection Options	258
34.2.1.	Authentication	258
34.2.2.	SSL.....	258
34.2.3.	SSH.....	259
34.2.4.	Firewall	260

34.2.5.	Logging.....	260
34.2.6.	Schema	261
34.2.7.	Miscellaneous	261

35. MONGODB 263

35.1.	MongoDB Version Support.....	263
35.2.	Connection Options	263
35.2.1.	Authentication	263
35.2.2.	Kerberos	264
35.2.3.	SSL.....	264
35.2.4.	SSH.....	265
35.2.5.	Firewall	265
35.2.6.	Logging.....	266
35.2.7.	Schema	266
35.2.8.	Caching	267
35.2.9.	Miscellaneous	267

36. MYSQL 270

36.1.	MySQL Version Support.....	270
36.2.	Connection Options	270
36.2.1.	Authentication	270
36.2.2.	Azure Authentication.....	271
36.2.3.	OAuth.....	271
36.2.4.	SSL.....	272
36.2.5.	SSH.....	272
36.2.6.	Firewall	273
36.2.7.	Proxy	273
36.2.8.	Logging.....	274
36.2.9.	Schema	274
36.2.10.	Miscellaneous	275

37. SAS DATA SETS 277

37.1.	Connection Options	277
37.1.1.	Authentication	277
37.1.2.	Connection.....	277
37.1.3.	AWS Authentication	278
37.1.4.	Azure Authentication.....	279
37.1.5.	SSO.....	279
37.1.6.	OAuth.....	280
37.1.7.	JWT OAuth	281
37.1.8.	Kerberos	281
37.1.9.	SSL.....	282
37.1.10.	SSH.....	282
37.1.11.	Firewall	283
37.1.12.	Proxy	283
37.1.13.	Logging.....	284
37.1.14.	Schema	284
37.1.15.	Caching	285
37.1.16.	Miscellaneous	285

38.	NEO4J 	288
38.1.	Neo4j Version Support	288
38.2.	Connection Options	288
38.2.1.	Authentication	288
38.2.2.	Connection.....	288
38.2.3.	SSL.....	288
38.2.4.	Firewall	289
38.2.5.	Proxy	289
38.2.6.	Logging.....	290
38.2.7.	Schema	290
38.2.8.	Caching	291
38.2.9.	Miscellaneous	291
39.	ORACLE 	293
39.1.	Prerequisite	293
39.2.	Oracle OCI Version Support.....	294
39.3.	Connection Options	294
39.3.1.	Authentication	294
39.3.2.	Logging.....	294
39.3.3.	Schema	295
39.3.4.	Miscellaneous	295
40.	PARQUET 	298
40.1.	Connection Options	298
40.1.1.	Authentication	298
40.1.2.	Connection.....	298
40.1.3.	AWS Authentication	299
40.1.4.	Azure Authentication.....	299
40.1.5.	SSO.....	300
40.1.6.	OAuth.....	300
40.1.7.	JWT OAuth	302
40.1.8.	SSL.....	302
40.1.9.	SSH.....	303
40.1.10.	Firewall	303
40.1.11.	Proxy	303
40.1.12.	Logging.....	304
40.1.13.	Schema	305
40.1.14.	Caching	305
40.1.15.	Miscellaneous	306
41.	POSTGRESQL 	309
41.1.	PostgreSQL Version Support.....	309
41.2.	Connection Options	309
41.2.1.	Authentication	309
41.2.2.	AWS Authentication	310
41.2.3.	Azure Authentication.....	310
41.2.4.	OAuth.....	310
41.2.5.	Kerberos	311

41.2.6.	SSL.....	312
41.2.7.	SSH.....	312
41.2.8.	Firewall	313
41.2.9.	Logging.....	313
41.2.10.	Schema	314
41.2.11.	Miscellaneous.....	314

42. **REDIS** 316

42.1.	Redis Version Support	316
42.2.	Connection Options	316
42.2.1.	Authentication	316
42.2.2.	Connection.....	316
42.2.3.	SSL.....	317
42.2.4.	SSH.....	317
42.2.5.	Firewall	318
42.2.6.	Logging.....	318
42.2.7.	Schema	319
42.2.8.	Caching	319
42.2.9.	Miscellaneous.....	320

43. **REDSHIFT** 322

43.1.	Amazon Redshift Version Support.....	322
43.2.	Connection Options	322
43.2.1.	Authentication	322
43.2.2.	AWS Authentication	323
43.2.3.	SSO.....	323
43.2.4.	SSL.....	324
43.2.5.	SSH.....	324
43.2.6.	Firewall	325
43.2.7.	Proxy	325
43.2.8.	Logging.....	326
43.2.9.	Schema	326
43.2.10.	Miscellaneous.....	327

44. **SALESFORCE** 329

44.1.	Connection Options	329
44.1.1.	Authentication	329
44.1.2.	Connection.....	329
44.1.3.	SSO.....	330
44.1.4.	BulkAPI.....	330
44.1.5.	OAuth.....	331
44.1.6.	JWT OAuth	332
44.1.7.	SSL.....	332
44.1.8.	Firewall	332
44.1.9.	Proxy	333
44.1.10.	Logging.....	333
44.1.11.	Schema	334
44.1.12.	Caching	334
44.1.13.	Miscellaneous.....	335

45.	SAP 	338
45.1.	SAP ERP Version Support.....	338
45.2.	Connection Options.....	338
45.2.1.	Authentication.....	338
45.2.2.	Security.....	339
45.2.3.	SSL.....	339
45.2.4.	Firewall.....	339
45.2.5.	Proxy.....	340
45.2.6.	Logging.....	340
45.2.7.	Schema.....	341
45.2.8.	Caching.....	341
45.2.9.	Miscellaneous.....	342
46.	SAP CLOUD FOR CUSTOMER 	345
46.1.	SAP Cloud for Customer Version Support.....	345
46.2.	Connection Options.....	345
46.2.1.	Authentication.....	345
46.2.2.	SSO.....	345
46.2.3.	OAuth.....	346
46.2.4.	SSL.....	347
46.2.5.	Firewall.....	347
46.2.6.	Proxy.....	347
46.2.7.	Logging.....	348
46.2.8.	Schema.....	349
46.2.9.	Caching.....	349
46.2.10.	Miscellaneous.....	350
47.	SAP HANA 	352
47.1.	Connection Options.....	352
47.1.1.	Authentication.....	352
47.1.2.	SSL.....	352
47.1.3.	SSH.....	353
47.1.4.	Firewall.....	354
47.1.5.	Logging.....	354
47.1.6.	Schema.....	355
47.1.7.	Miscellaneous.....	355
48.	SERVICE NOW 	357
48.1.	ServiceNow Version Support.....	357
48.2.	Connection Options.....	357
48.2.1.	Authentication.....	357
48.2.2.	SSO.....	357
48.2.3.	OAuth.....	358
48.2.4.	SSL.....	359
48.2.5.	Firewall.....	359
48.2.6.	Proxy.....	359
48.2.7.	Logging.....	360
48.2.8.	Schema.....	361

48.2.9.	Caching	361
48.2.10.	Data Formatting.....	362
48.2.11.	Miscellaneous	362

49. **SHAREPOINT** 365

49.1.	SharePoint Version Support	365
49.2.	Connection Options	365
49.2.1.	Authentication	365
49.2.2.	Azure Authentication.....	365
49.2.3.	SSO.....	366
49.2.4.	OAuth.....	366
49.2.5.	JWT OAuth	367
49.2.6.	Kerberos	367
49.2.7.	SSL.....	368
49.2.8.	Firewall	368
49.2.9.	Proxy	369
49.2.10.	Logging.....	369
49.2.11.	Schema	370
49.2.12.	Caching	370
49.2.13.	Miscellaneous	371

50. **SNOWFLAKE** 374

50.1.	Snowflake Version Support	374
50.2.	Connection Options	374
50.2.1.	Authentication	374
50.2.2.	Connection.....	375
50.2.3.	Azure Authentication.....	375
50.2.4.	SSO.....	375
50.2.5.	KeyPairAuth	376
50.2.6.	OAuth.....	376
50.2.7.	SSL.....	377
50.2.8.	Firewall	377
50.2.9.	Proxy	378
50.2.10.	Logging.....	378
50.2.11.	Schema	379
50.2.12.	Caching	379
50.2.13.	Miscellaneous	380

51. **SPARK** 383

51.1.	Spark SQL Version Support	383
51.2.	Connection Options	383
51.2.1.	Authentication	383
51.2.2.	Kerberos	384
51.2.3.	SSL.....	384
51.2.4.	Firewall	385
51.2.5.	Proxy	385
51.2.6.	Logging.....	386
51.2.7.	Schema	386
51.2.8.	Caching	387

51.2.9. Miscellaneous	387

52. SQL SERVER 390

52.1. SQL Server Version Support.....	390
52.2. Connection Options	390
52.2.1. Authentication	390
52.2.2. Azure Authentication.....	391
52.2.3. OAuth.....	391
52.2.4. Kerberos	391
52.2.5. SSL.....	392
52.2.6. SSH.....	392
52.2.7. Firewall	393
52.2.8. Proxy	394
52.2.9. Logging.....	394
52.2.10. Schema	395
52.2.11. Miscellaneous.....	395
52.3. Description Extraction	397

53. SQL SERVER ANALYSIS SERVICES 398

53.1. Connection Options	398
53.1.1. Authentication	398
53.1.2. Kerberos	398
53.1.3. SSL.....	399
53.1.4. Firewall	399
53.1.5. Proxy	400
53.1.6. Logging.....	400
53.1.7. Schema	401
53.1.8. Caching	401
53.1.9. Miscellaneous.....	402

54. SYBASE 404

54.1. Sybase Version Support.....	404
54.2. Connection Options	404
54.2.1. Authentication	404
54.2.2. Kerberos	405
54.2.3. SSL.....	405
54.2.4. Firewall	406
54.2.5. Logging.....	406
54.2.6. Schema	407
54.2.7. Miscellaneous.....	407

55. TERADATA 409

55.1. Prerequisite	409
55.2. Connection Options	410
55.2.1. Authentication	410
55.2.2. Connection.....	410
55.2.3. Firewall	411

55.2.4.	Logging.....	411
55.2.5.	Schema	412
55.2.6.	Miscellaneous	412

56. **XML** 417

56.1.	Connection Options.....	417
56.1.1.	Authentication	417
56.1.2.	Connection.....	417
56.1.3.	AWS Authentication	418
56.1.4.	Azure Authentication.....	419
56.1.5.	SSO.....	419
56.1.6.	OAuth.....	420
56.1.7.	JWT OAuth	421
56.1.8.	Kerberos	422
56.1.9.	SSL.....	422
56.1.10.	SSH.....	423
56.1.11.	Firewall	423
56.1.12.	Proxy	424
56.1.13.	Logging.....	424
56.1.14.	Schema	425
56.1.15.	Caching	426
56.1.16.	Miscellaneous	426

INTRODUCTION

The HOPEX Data Discovery tool allows you to browse different data sources and define the metadata to import into your data catalogs.

Each data source has connection parameters. This document presents the connection string options of JDBC driver for each data source.

1.1. Prerequisite for using HOPEX Data Discovery

The HOPEX Data Discovery tool is available as a module, downloadable from the HOPEX HAS console.

For more details see the documentation [HOPEX Data Governance > Data Catalogs > Data Discovery](#).

In stand-alone mode, the workstation on which the module is installed must have .Net Core 6 and Microsoft Edge WebView2 installed.

After deploying the tool, you can connect to a data source: depending on the connector, see the list of supported versions and the description of the connection options.

After selecting the connector, to ensure the connection of HOPEX Data Discovery with the data source, follow these steps.

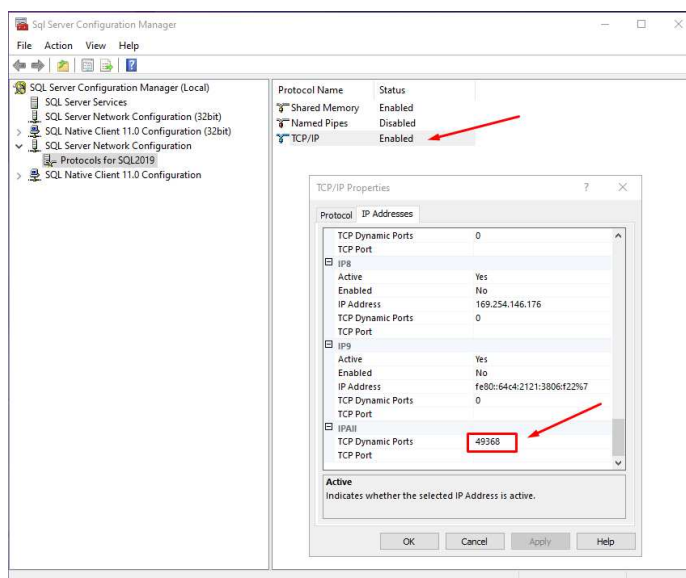
Connection to the data source

1. Make sure that the workstation where HOPEX Data Discovery is deployed is allowed to access the server where the data source is deployed (firewall, etc.).
2. Indicate the port/server/user names and the password.
3. Depending on the authentication system (for example Kerberos), check with the IT manager that the connection ticket is up to date.
4. For some connectors it is necessary to enter the name of the database you want to extract.

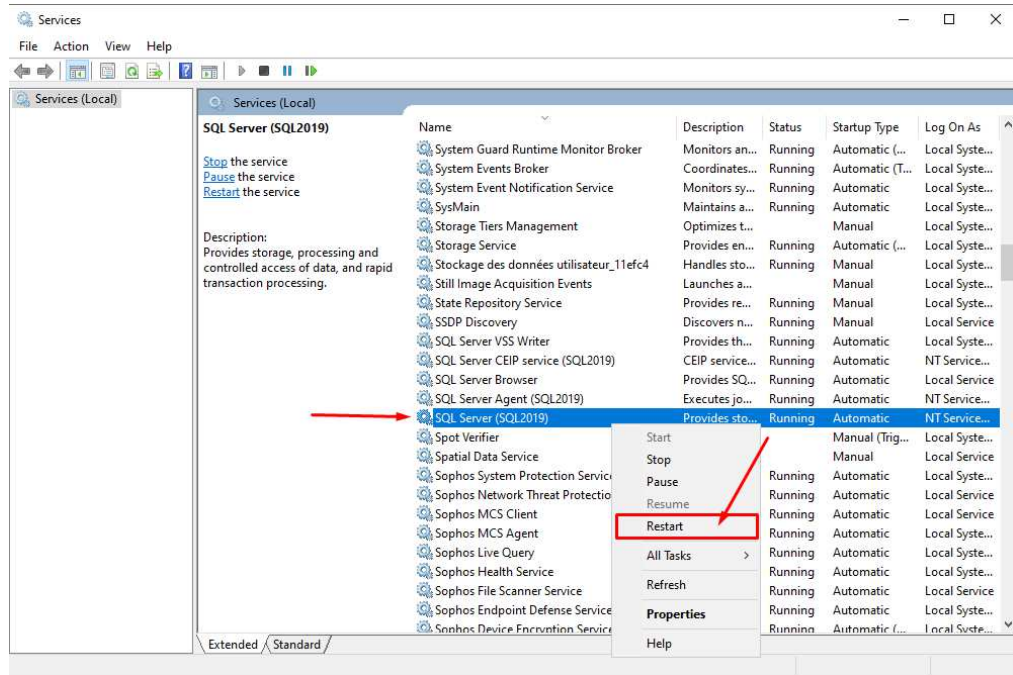
Network configuration

1. Check that the port used to connect to the data source is the correct one and that TCP/IP is enabled.

Example for SQL Server



2. After modifications restart the services.



1.2. Connection issues

If the connection is not established, provide MEGA Support with the Log file of connection errors.

To get the log file of connection errors:

1. Create a "DataDiscoveryLog.Txt" file and save it, in the "c:\temp" folder for example.
2. In the connection property "Logfile", specify the path to the file.
3. Add the property "Verbosity", with the value "5".

The syntax is as follows:

logfile="c:\tempDataDiscoveryLog.txt";verbosity="5".

Make sure there are no spaces in the path or in the file name.

In addition to the connection log file:

- Online mode: get all the logs from HOPEX.
- Stand-alone mode: get the log from HAS Light

2. ACCESS

2.1. Access Version Support

The driver may work with Access 1997 (.mdb, r/o), Access 2000 (.mdb), Access 2003 (.mdb), Access 2007 (.accdb), Access 2010 (.accdb), Access 2013 (.accdb) database files.

2.2. Connection Options

2.2.1. Database

Property	Description
DataSource	The full path and name of the MS Access database file.

2.2.2. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

2.2.3. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.

BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

2.2.4. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Access when the connection is opened.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
Readonly	You can use this property to enforce read-only access to Access from the provider.
RTK	The runtime key used for licensing.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/DCG/jdbc/>.

3. AMAZON ATHENA

3.1. Connection Options

3.1.1. [Authentication](#)

Property	Description
AthenaVPCEndpoint	Your Athena VPC Endpoint. It will override the default athena.{region}.amazonaws.com endpoint.
S3StagingDirectory	Specifies the location Amazon Athena will use to store the results of a query.
EncryptionType	Specifies the encryption option for query results in Athena.
EncryptionKey	Specifies the encryption key when encrypting results stored in S3.
DataSource	The name of the Athena Data Source.
Database	The name of the Athena database.

3.1.2. [Connection](#)

Property	Description
CrossAccountId	The Account Id to use when retrieving metadata from Glue. Default is current account id.
SimpleUploadLimit	This setting specifies the threshold, in bytes, above which the provider will choose to perform a multipart upload rather than uploading everything in one request.
UseLakeFormation	When this property is set to true, AWSLakeFormation service will be used to retrieve temporary credentials, which enforce access policies against the user based on the configured IAM role. The service can be used when authenticating through OKTA, ADFS, AzureAD, PingFederate, while providing a SAML assertion.

3.1.3. AWS Authentication

Property	Description
AuthScheme	The scheme used for authentication. Accepted entries are: , AwsRootKeys , AwsIAMRoles , AwsEC2Roles , ADFS, Okta, PingFederate , AwsCredentialsFile , AwsCognitoBasic , AwsCognitoSrp.
AWSAccessKey	Your AWS account access key. This value is accessible from your AWS security credentials page.
AWSSecretKey	Your AWS account secret key. This value is accessible from your AWS security credentials page.
AWSRoleARN	The Amazon Resource Name of the role to use when authenticating.
AWSPrincipalARN	The ARN of the SAML Identity provider in your AWS account.
AWSRegion	The hosting region for your Amazon Web Services.
AWSCredentialsFile	The path to the AWS Credentials File to be used for authentication.
AWSCredentialsFileProfile	The name of the profile to be used from the supplied AWSCredentialsFile.
AWSSessionToken	Your AWS session token.
AWSExternalId	A unique identifier that might be required when you assume a role in another account.
MFASerialNumber	The serial number of the MFA device if one is being used.
MFAToken	The temporary token available from your MFA device.
CredentialsLocation	The location of the settings file where MFA credentials are saved.
TemporaryTokenDuration	The amount of time (in seconds) a temporary token will last.
AWSCognitoRegion	The hosting region for AWS Cognito.
AWSUserPoolId	The User Pool Id.
AWSUserPoolClientId	The User Pool Client App Id.
AWSUserPoolClientAppSecret	Optional. The User Pool Client App Secret.
AWSIdentityPoolId	The Identity Pool Id.

[AWSWebIdentityToken](#)

The OAuth 2.0 access token or OpenID Connect ID token that is provided by an identity provider.

3.1.4. [SSO](#)

Property	Description

[User](#)

The IDP user used to authenticate the IDP via SSO.

[Password](#)

The password used to authenticate the IDP user via SSO.

[SSOLoginURL](#)

The identity provider's login URL.

[SSOProperties](#)

Additional properties required to connect to the identity provider in a semicolon-separated list.

[SSOExchangeUrl](#)

The URL used for consuming the SAML response and exchanging it for service specific credentials.

3.1.5. [OAuth](#)

Property	Description

[InitiateOAuth](#)

Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.

[OAuthClientId](#)

The client Id assigned when you register your application with an OAuth authorization server.

[OAuthClientSecret](#)

The client secret assigned when you register your application with an OAuth authorization server.

[OAuthAccessToken](#)

The access token for connecting using OAuth.

[OAuthSettingsLocation](#)

The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH . Alternatively, you can hold this location in memory by specifying a value starting with 'memory://'.

[CallbackURL](#)

The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.

[OAuthVerifier](#)

The verifier code returned from the OAuth authorization URL.

[OAuthRefreshToken](#)

The OAuth refresh token for the corresponding OAuth access token.

[OAuthExpiresIn](#)

The lifetime in seconds of the OAuth AccessToken.

[OAuthTokenTimestamp](#)

The Unix epoch timestamp in milliseconds when the current Access Token was created.

3.1.6. [SSL](#)

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

3.1.7. [Firewall](#)

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

3.1.8. [Proxy](#)

Property	Description
ProxyAutoDetect	When this connection property is set to True, the provider checks your system proxy settings for existing proxy server configurations (no need to manually supply proxy server details). Set to False if you want to manually configure the provider to connect to a specific proxy server.
ProxyServer	The hostname or IP address of the proxy server that you want to route HTTP traffic through.
ProxyPort	The TCP port that the proxy server (specified in the ProxyServer connection property) is running on.
ProxyAuthScheme	The authentication method the provider uses when authenticating to the proxy server specified in the ProxyServer connection property.

ProxyUser	The username of a user account registered with the proxy server specified in the ProxyServer connection property.
ProxyPassword	The password associated with the user specified in the ProxyUser connection property.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

3.1.9. [Logging](#)

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

3.1.10. [Schema](#)

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
FlattenArrays	By default, arrays are returned as strings of JSON. The FlattenArrays property can be used to flatten the elements of arrays into columns of their own. Set FlattenArrays to the number of elements you want to return from arrays.

[FlattenObjects](#)

Set FlattenObjects to true to flatten object properties into columns of their own. Otherwise, objects nested in arrays are returned as strings of JSON.

3.1.11. [Caching](#)

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

3.1.12. [Miscellaneous](#)

Property	Description
BatchSize	The maximum size of each batch operation to submit.
CleanQueryResults	Amazon Athena produces cache files with every query, in the folder specified in S3StagingDirectory . CleanQueryResults specifies whether these files should be deleted once the connection is closed.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property specifies whether to connect to the Amazon Athena when the connection is opened.
EnableFIPSMODE	A boolean indicating if FIPS URLs should be enabled.

<u>IncludeTableTypes</u>	If set to true, the provider will query for the types of individual tables.
<u>MaximumColumnSize</u>	The maximum column size.
<u>MaxRows</u>	Limits the number of rows returned when no aggregation or GROUP BY is used in the query. This takes precedence over LIMIT clauses.
<u>MetadataDiscoveryMethod</u>	API to use to retrieve the metadata for Amazon Athena.
<u>Other</u>	These hidden properties are used only in specific use cases.
<u>PageSize</u>	The number of results to return per page of data retrieved from Amazon Athena.
<u>PollingInterval</u>	This determines the polling interval in milliseconds to check whether the result is ready to be retrieved.
<u>PoolIdleTimeout</u>	The allowed idle time for a connection before it is closed.
<u>PoolMaxSize</u>	The maximum connections in the pool.
<u>PoolMinSize</u>	The minimum number of connections in the pool.
<u>PoolWaitTime</u>	The max seconds to wait for an available connection.
<u>PseudoColumns</u>	Specify a set of pseudocolumns to expose as columns.
<u>PushEmptyValuesAsNull</u>	Indicates whether to read the empty values as empty or as null.
<u>QueryCachingLevel</u>	Modifies the usage of query results stored in S3StagingDirectory . Cached queries will not be scanned again by Amazon Athena until the amount of seconds specified in QueryTolerance has passed.
<u>QueryPassthrough</u>	This option passes the query to the Amazon Athena server as is.
<u>QueryTimeout</u>	The timeout in seconds for requests issued by the provider to download large result sets.
<u>QueryTolerance</u>	Specifies the amount of time in seconds queries cache for if QueryCachingLevel is not set to None.
<u>ResultFetcher</u>	The fetcher that will be used to download query results.
<u>RTK</u>	The runtime key used for licensing.
<u>ShowS3Filepath</u>	Whether or not to expose the _S3Path column for the Amazon Athena tables.
<u>SkipHeaderLineCount</u>	Specifies the number of header rows to skip for SELECT queries.
<u>Timeout</u>	The value in seconds until the timeout error is thrown, canceling the operation.

[UseConnectionPooling](#)

This property enables connection pooling.

[UserDefinedViews](#)

A filepath pointing to the JSON configuration file containing your custom views.

[Workgroup](#)

Work group of the executed queries.

For more details, see <https://cdn.cdata.com/help/FKK/jdbc/>.

4. AMAZON DYNAMODB

4.1. Amazon DynamoDB Version Support

The driver uses the current version of the Amazon DynamoDB REST API, version 2012-08-10, to enable read/write access to DynamoDB instances.

4.2. Connection Options

4.2.1. AWS Authentication

Property	Description
AuthScheme	The scheme used for authentication. Accepted entries are: Auto, , AwsRootKeys , AwsIAMRoles , AwsEC2Roles , AwsMFA , ADFS, Okta, PingFederate , AwsCredentialsFile.
Domain	Your AWS domain name. You can optionally choose to associate your domain name with AWS.
AWSAccessKey	Your AWS account access key. This value is accessible from your AWS security credentials page.
AWSSecretKey	Your AWS account secret key. This value is accessible from your AWS security credentials page.
AWSRoleARN	The Amazon Resource Name of the role to use when authenticating.
AWSRegion	The hosting region for your Amazon Web Services.
AWSCredentialsFile	The path to the AWS Credentials File to be used for authentication.
AWSCredentialsFileProfile	The name of the profile to be used from the supplied AWSCredentialsFile.
AWSSessionToken	Your AWS session token.

AWSExternalId	A unique identifier that might be required when you assume a role in another account.
MFASerialNumber	The serial number of the MFA device if one is being used.
MFAToken	The temporary token available from your MFA device.
CredentialsLocation	The location of the settings file where MFA credentials are saved.
TemporaryTokenDuration	The amount of time (in seconds) a temporary token will last.

4.2.2. SSO

Property	Description
User	The IDP user used to authenticate the IDP via SSO.
Password	The password used to authenticate the IDP user via SSO.
SSOLoginURL	The identity provider's login URL.
SSOProperties	Additional properties required to connect to the identity provider in a semicolon-separated list.
SSOExchangeUrl	The url used for consuming the SAML response and exchanging it with Amazon DynamoDB specific credentials.

4.2.3. SSL

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

4.2.4. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

4.2.5. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

4.2.6. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

4.2.7. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

4.2.8. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

4.2.9. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Amazon DynamoDB when the connection is opened.

FlattenArrays	By default, nested arrays are returned as strings of JSON. The FlattenArrays property can be used to flatten the elements of nested arrays into columns of their own. Set FlattenArrays to the number of elements you want to return from nested arrays.
FlattenObjects	Set FlattenObjects to true to flatten object properties into columns of their own. Otherwise, objects nested in arrays are returned as strings of JSON.
FlexibleSchema	Set FlexibleSchema to true to scan for additional metadata on the query result set. Otherwise, the metadata will remain the same.
GenerateSchemaFiles	Indicates the user preference as to when schemas should be generated and saved.
IgnoreTypes	Removes support for the specified types. For example, Time. These types will then be reported as strings instead.
MaximumRequestRetries	The maximum number of times to retry a request.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from Amazon DynamoDB.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.

Readonly	You can use this property to enforce read-only access to Amazon DynamoDB from the provider.
RetryWaitTime	The minimum number of milliseconds the provider will wait to retry a request.
RowScanDepth	The maximum number of rows to scan to look for the columns available in a table.
RTK	The runtime key used for licensing.
SeparatorCharacter	The character or characters used to denote hierarchy.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
ThreadCount	The number of threads to use when selecting data via a parallel scan. Setting ThreadCount to 1 will disable parallel scans.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
TypeDetectionScheme	Determines how to determine the data type of columns.
UseBatchWriteItemOperation	When enabled the provider will use BatchWriteItem operation for handling updates and inserts. By default, the provider uses ExecuteStatement/BatchExecuteStatement operation. You need to enable BatchWriteItem only when inserting/updating binary/binary-set data. ExecuteStatement/BatchExecuteStatement doesn't support manipulating binary fields.
UseConnectionPooling	This property enables connection pooling.
UseConsistentReads	Whether to always use Consistent Reads or not when querying DynamoDb.
UseSimpleNames	Boolean determining if simple names should be used for tables and columns.

For more details, see <https://cdn.cdata.com/help/DDG/jdbc/>.

5. AZURE ANALYSIS SERVICES

5.1. Connection Options

5.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to Azure Analysis Services.
URL	The URL used to connect to the Azure Analysis Services.
User	The Azure Analysis Services user account used to authenticate.
Password	The password used to authenticate the user.

5.1.2. Azure Authentication

Property	Description
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.
AzureEnvironment	The Azure Environment to use when establishing a connection.

5.1.3. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

5.1.4. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

5.1.5. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

5.1.6. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

5.1.7. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.

LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

5.1.8. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
Catalog	The Analysis Services catalog to use. This may also be known as a Database from within Analysis Services.
IncludeJoinColumns	Set this to true to include extra join columns on each table.

5.1.9. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

5.1.10. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.

ConnectOnOpen	This property species whether to connect to the Azure Analysis Services when the connection is opened.
CustomHeaders	Other headers as determined by the user (optional).
ExtraProperties	Additional properties to submit on each MDX request to Azure Analysis Services.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
ResponseRowLimit	The number of response rows to allow before erroring. Set to 0 for now limit.
RTK	The runtime key used for licensing.
SplitMeasures	Set this to true to split Measures table into individual tables.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.

UseMDX	Set this to true to pass MDX queries to Azure Analysis Services as-is.

For more details, see <https://cdn.cdata.com/help/OAG/jdbc/>.

6. AZURE DATA CATALOG

6.1. Connection Options

6.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to Azure Data Catalog.
CatalogName	The name of the catalog to connect to.

6.1.2. Azure Authentication

Property	Description
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.
AzureEnvironment	The Azure Environment to use when establishing a connection.

6.1.3. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.

OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

6.1.4. SSL

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

6.1.5. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

6.1.6. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.

ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

6.1.7. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

6.1.8. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.

Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

6.1.9. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

6.1.10. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.

ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Azure Data Catalog when the connection is opened.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from Azure Data Catalog.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
RTK	The runtime key used for licensing.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/GNG/jdbc/>.

7. AZURE SYNAPSE

7.1. Connection Options

7.1.1. Authentication

Property	Description
AuthScheme	The scheme used for authentication. Accepted entries are Password, AzureAD, AzureMSI.
Server	The name of the server running Synapse.
Port	The port of the Synapse.
User	The Azure Synapse user account used to authenticate.
Database	The name of the Synapse database.
Password	The password used to authenticate the user.
Encrypt	This field sets whether SSL is enabled.

7.1.2. Bulk

Property	Description
BatchMode	The Batch Mode of Azure Synapse bulkInsert.
StorageAccountLocation	The Storage Account Location for staging your data.
AzureSASToken	The string value of the Azure Blob shared access signature.

7.1.3. Azure Authentication

Property	Description
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.

7.1.4. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
OAuthGrantType	The grant type for the OAuth flow.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

7.1.5. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

7.1.6. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

7.1.7. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.

Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

7.1.8. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

7.1.9. Miscellaneous

Property	Description
CustomizeDateFormat	Configure the Date format.
ApplicationIntent	Expresses the client application's request to be directed either to a read-write or read-only version of an availability group database.
BatchSize	The maximum size of each batch operation to submit.

ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Azure Synapse when the connection is opened.
EnableTransaction	Determines whether transactions are enabled.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
QueryPassthrough	This option passes the query to the Azure Synapse server as is.
Readonly	You can use this property to enforce read-only access to Azure Synapse from the provider.
RTK	The runtime key used for licensing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/HEG/jdbc/>.

8. AZURE TABLE

8.1. Connection Options

8.1.1. Authentication

Property	Description
Account	The Windows Azure Storage account name.
UseSSL	This field sets whether SSL is enabled. The default is true.
AccessKey	The key for the storage account.
AuthScheme	The scheme used for authentication. Accepted entries are AccessToken and SharedAccessSignature.
Backend	The backend where data is stored.
SharedAccessSignature	A shared access key signature that may be used for authentication.

8.1.2. SSL

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

8.1.3. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.

FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

8.1.4. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

8.1.5. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

8.1.6. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.

8.1.7. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

8.1.8. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Azure Table Storage when the connection is opened.

GenerateSchemaFiles	Indicates the user preference as to when schemas should be generated and saved.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
Readonly	You can use this property to enforce read-only access to Azure Table Storage from the provider.
RowScanDepth	The maximum number of rows to scan to look for the columns available in a table.
RTK	The runtime key used for licensing.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
TypeDetectionScheme	Determines how to determine the data type of columns.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/CAG/jdbc/>.

9. BIGQUERY

9.1. Connection Options

9.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to Google BigQuery.
ProjectId	The ProjectId used to resolve unqualified tables.
DatasetId	The DatasetId of the dataset you wish to connect to and view tables of.
BillingProjectId	The ProjectId of the billing project for executing jobs.

9.1.2. BigQuery

Property	Description
AllowLargeResultSets	Whether or not to allow large datasets to be stored in temporary tables for large datasets.
DestinationTable	This property determines where query results are stored in Google BigQuery.
UseQueryCache	Specifies whether to use Google BigQuery's built-in query cache.
PageSize	The number of results to return per page from Google BigQuery.
PollingInterval	This determines how long to wait in seconds, between checks to see if a job has completed.
AllowUpdatesWithoutKey	Whether or not to allow update without primary keys.

FilterColumns	Please set `AllowUpdatesWithoutKey` to true before you could use this property.
UseLegacySQL	Specifies whether to use BigQuery's legacy SQL dialect for this query. By default, Standard SQL will be used.

9.1.3. Storage API

Property	Description
UseStorageAPI	Specifies whether to use BigQuery's Storage API for bulk data reads.
UseArrowFormat	Specifies whether to use the Arrow format with BigQuery's Storage API.
StorageThreshold	The minimum number of rows a query must return to invoke the Storage API.
StoragePageSize	Specifies the page size to use for Storage API queries.

9.1.4. Uploading

Property	Description
InsertMode	Specifies what kind of method to use when inserting data. By default streaming inserts are used.
WaitForBatchResults	Whether to wait for the job to complete when using the bulk upload API. Only active when InsertMode is set to Upload.
GCSBucket	Specifies the name of a GCS bucket to upload bulk data for staging.
GCSBucketFolder	Specifies the name of the folder in GCSBucket to upload bulk data for staging.
TempTableDataset	The prefix of the dataset that will contain temporary tables when performing bulk UPDATE or DELETE operations.

9.1.5. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
Scope	Identifies the Google API access that your application is requesting. Scopes are space-delimited.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

9.1.6. JWT OAuth

Property	Description
OAuthJWTCert	The JWT Certificate store.

OAuthJWTCertType	The type of key store containing the JWT Certificate.
OAuthJWTCertPassword	The password for the OAuth JWT certificate.
OAuthJWTCertSubject	The subject of the OAuth JWT certificate.
OAuthJWTIssuer	The issuer of the Java Web Token.
OAuthJWTSubject	The user subject for which the application is requesting delegated access.

9.1.7. SSL

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

9.1.8. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

9.1.9. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

9.1.10. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).

MaxLogFileCount	A string specifying the maximum file count of log files.

9.1.11. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
RefreshViewSchemas	Allows the provider to determine up-to-date view schemas automatically.
ShowTableDescriptions	Controls whether table descriptions are returned via the platform metadata APIs and sys_tables / sys_views.
PrimaryKeyIdentifiers	Set this property to define primary keys.
AllowedTableTypes	Specifies what kinds of tables will be visible.

9.1.12. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.

CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

9.1.13. Miscellaneous

Property	Description
StorageTimeout	How long a Storage API connection must remain idle before the provider reconnects.
AllowAggregateParameters	Allows raw aggregates to be used in parameters when QueryPassthrough is enabled.
AuditLimit	The maximum number of rows which will be stored within an audit table.
AuditMode	What provider actions should be recorded to audit tables.
BatchSize	The maximum size of each batch operation to submit.
BigQueryOptions	A comma separated list of Google BigQuery options.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Google BigQuery when the connection is opened.

GenerateSchemaFiles	Indicates the user preference as to when schemas should be generated and saved.
MaximumBillingTier	The MaximumBillingTier is a positive integer that serves as a multiplier of the basic price per TB. For example, if you set MaximumBillingTier to 2, the maximum cost for that query will be 2x basic price per TB.
MaximumBytesBilled	Limits how many bytes BigQuery will allow a job to consume before it is cancelled.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
QueryPassthrough	This option passes the query to the Google BigQuery server as is.
Readonly	You can use this property to enforce read-only access to Google BigQuery from the provider.
RTK	The runtime key used for licensing.
TableSamplePercent	This determines what percent of a table is sampled with the TABLESAMPLE operator.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/DBG/jdbc/>.

10. CASSANDRA

10.1. Cassandra Version Support

The driver supports CQL versions 2.0, 3.0 and 4.0.

10.2. Connection Options

10.2.1. Authentication

Property	Description
AuthScheme	The scheme used for authentication. Accepted entries are Basic, DSE, Kerberos, and LDAP.
Server	The host name or IP address of the server hosting the Cassandra database.
Port	The port for the Cassandra database.
LDAPServer	The host name or IP address of the LDAP server.
User	The Cassandra user account used to authenticate.
Password	The password used to authenticate the user.
LDAPPort	The port for the LDAP server.
Database	The name of the Cassandra keyspace.
DefaultLDAPUser	The default LDAP user used to connect to and communicate with the server, it must be set if the LDAP server do not allow anonymous bind.
LDAPPassword	The password of the default LDAP user. It must be set if the LDAP server do not allow anonymous bind.
SearchBase	The search base for your LDAPServer, used to look up users.

SearchFilter	The search filter for looking up usernames in LDAP. The default setting is (uid=), When using Active Directory set the filter to (sAMAccountName=).
UseSSL	This field sets whether SSL is enabled.

10.2.2. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
KerberosRealm	The Kerberos Realm used to authenticate the user with.
KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosServiceRealm	The Kerberos realm of the service.
KerberosServiceKDC	The Kerberos KDC of the service.
KerberosTicketCache	The full file path to an MIT Kerberos credential cache file.

10.2.3. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.

SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

10.2.4. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertSubject	The subject of the SSH client certificate.
SSHClientCertType	The type of SSHClientCert certificate.
SSHServer	The SSH server.
SSHPort	The SSH port.
SSHUser	The SSH user.
SSHPassword	The SSH password.
SSHServerFingerprint	The SSH server fingerprint.
UseSSH	Whether to tunnel the Cassandra connection over SSH. Use SSH.

10.2.5. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

10.2.6. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

10.2.7. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

10.2.8. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.

CacheMetadata

This property determines whether or not to cache the table metadata to a file store.

10.2.9. Miscellaneous

Property	Description
AggregationsSupported	Whether or not to support aggregations in the Cassandra server. Note that in queries to the provider, you must use single quotes to define strings.
AllowFiltering	When true, slow-performing queries are processed on the server.
BatchSize	The maximum size of each batch operation to submit.
CaseSensitivity	Enable case sensitivity to the CQL sending to the server, if set to True, the identifiers in the CQL will be enclosed in double quotation marks.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Cassandra when the connection is opened.
ConsistencyLevel	The consistency level determines how many of the replicas of the data you are interacting with need to respond for the query to be considered a success.
FlattenArrays	By default, nested arrays are returned as strings of JSON. The FlattenArrays property can be used to flatten the elements of nested arrays into columns of their own. Set FlattenArrays to the number of elements you want to return from nested arrays.
FlattenObjects	Set FlattenObjects to true to flatten object properties into columns of their own. Otherwise, objects nested in arrays are returned as strings of JSON.

MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
NullToUnset	Use unset instead of NULL in CQL query when performing INSERT operations.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from Cassandra.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
QueryPassthrough	This option passes the query to the Cassandra server as is.
Readonly	You can use this property to enforce read-only access to Cassandra from the provider.
RowScanDepth	The maximum number of rows to scan to look for the columns available in a table.
RTK	The runtime key used for licensing.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.

UseJsonFormat	Whether to submit and return the JSON encoding for CQL data types.
VarintToString	Map Cassandra VARINT to String value.

For more details, see <https://cdn.cdata.com/help/RCG/jdbc/>.

11. CLOUDANT

11.1. Connection Options

11.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to Cloudant.
URL	The URL used to connect to the Cloudant.
User	The Cloudant user account used to authenticate.
Password	The password used to authenticate the user.
ApiKey	The API Key used to identify the user to IBM Cloud.

11.1.2. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.

OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

11.1.3. SSL

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

11.1.4. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

11.1.5. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.

ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

11.1.6. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

11.1.7. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.

BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
ListViews	Whether to list views from Cloudant or not.

11.1.8. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

11.1.9. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Cloudant when the connection is opened.
FlattenArrays	Set FlattenArrays to the number of array elements to flatten into columns. Otherwise, arrays are returned as JSON strings.
FlattenObjects	Set FlattenObjects to true to flatten object properties into columns of their own. Otherwise, embedded objects as raw JSON strings.
FlexibleSchema	Set FlexibleSchema to true to scan for additional metadata on the query result set. Otherwise, the metadata will remain the same.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from Cloudant.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.

PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
Readonly	You can use this property to enforce read-only access to Cloudant from the provider.
RowScanDepth	The maximum number of rows to scan to look for the columns available in a table.
RTK	The runtime key used for licensing.
SeparatorCharacter	The character or characters used to denote hierarchy.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
TypeDetectionScheme	Option for how the provider will scan the data to determine the fields and data types in each document collection.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/EWG/jdbc/>.

12. COCKROACHDB

12.1. CockroachDB Version Support

The driver enables standards-based access to Cockroach DB version 2.0 and later.

12.2. Connection Options

12.2.1. Authentication

Property	Description
Server	The host name or IP address of the server.
Port	The port number of the CockroachDB server.
Database	The name of the Cockroach database.
User	The CockroachDB user account used to authenticate.
Password	The password used to authenticate the user.
UseSSL	Whether SSL is enabled.

12.2.2. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.

SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

12.2.3. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

12.2.4. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

12.2.5. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

12.2.6. Miscellaneous

Property	Description
AllowPreparedStatement	Allow the preparation of a query statement before its execution.
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the CockroachDB when the connection is opened.
FetchResultSetMetadata	This field sets whether the provider is getting detailed information about resultset columns from the server.

MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
QueryPassthrough	This option passes the query to the CockroachDB server as is.
Readonly	You can use this property to enforce read-only access to CockroachDB from the provider.
RTK	The runtime key used for licensing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
TimeZone	The time zone for the current session.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/HJG/jdbc/>.

13. CONFLUENCE

13.1. Connection Options

13.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to Confluence.
URL	The URL to your JIRA endpoint.
CloudId	The Cloud Id for the Atlassian site that was authorized.
User	The Confluence user account used to authenticate.
Password	The password used to authenticate the user.
APIToken	APIToken of the currently authenticated user.
Timezone	Specify the timezone of the Confluence instance in order to use the datetime filters accordingly and retrieve the results according to your timezone. An example of a timezone would be America/New_York.

13.1.2. SSO

Property	Description
SSOLoginURL	The identity provider's login URL.
SSOProperties	Additional properties required to connect to the identity provider in a semicolon-separated list.
SSOExchangeUrl	The url used for consuming the SAML response and exchanging it with Confluence specific credentials.
SSOAppName	App Name used with SSO for IdPs that require it.

SSOAppPassword

App Password used with SSO for IdPs that require it.

13.1.3. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthVersion	The version of OAuth being used.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
CertificateStore	The certificate store used for Confluence authentication.
CertificateStorePassword	The password of the certificate store used with Confluence authentication.
CertificateSubject	The subject of the certificate used with Confluence Private Application authentication.
CertificateStoreType	The type of certificate store used with Confluence Private Application authentication.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.

OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

13.1.4. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

13.1.5. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

13.1.6. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

13.1.7. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).

MaxLogFileCount	A string specifying the maximum file count of log files.

13.1.8. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

13.1.9. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.

CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

13.1.10. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Confluence when the connection is opened.
IncludeArchivedSpaces	Whether to include content from archived spaces in the result. This defaults to false.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from Confluence.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.

PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
RTK	The runtime key used for licensing.
SpaceKey	By specifying the SpaceKey, the search results will only display contents from this specific space.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/UGG/jdbc/>.

14. COSMOS DB

14.1. Connection Options

14.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to Cosmos DB.
AccountEndpoint	The value should be the Cosmos DB account URL from the Keys blade of the Cosmos DB account.
AccountKey	A master key token or a resource token for connecting to the Cosmos DB REST API.
TokenType	Denotes the type of token: master or resource.

14.1.2. Azure Authentication

Property	Description
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.
AzureEnvironment	The Azure Environment to use when establishing a connection.

14.1.3. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
OAuthGrantType	The grant type for the OAuth flow.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

14.1.4. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

14.1.5. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

14.1.6. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

14.1.7. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.

MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

14.1.8. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
Schema	Specify the Cosmos DB database you want to work with.

14.1.9. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.

CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

14.1.10. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
CalculateAggregates	Specifies whether will return the calculated value of the aggregates or grouped by partiton range.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Cosmos DB when the connection is opened.
ConsistencyLevel	Denotes the type of token: master or resource.
FlattenArrays	By default, nested arrays are returned as strings of JSON. The FlattenArrays property can be used to flatten the elements of nested arrays into columns of their own. Set FlattenArrays to the number of elements you want to return from nested arrays.
FlattenObjects	Set FlattenObjects to true to flatten object properties into columns of their own. Otherwise, objects nested in arrays are returned as strings of JSON.
GenerateSchemaFiles	Indicates the user preference as to when schemas should be generated and saved.

MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
MaxThreads	Specifies the maximum number of concurrent requests for Batch CUD (Create, Update, Delete) operations.
MultiThreadCount	Aggregate queries in partitioned collections will require parallel requests for different partition ranges. Set this to the number of parallel request to be issued in the same time.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from Cosmos DB.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
Readonly	You can use this property to enforce read-only access to Cosmos DB from the provider.
RowScanDepth	The maximum number of rows to scan to look for the columns available in a table.
RTK	The runtime key used for licensing.
SeparatorCharacter	The character or characters used to denote hierarchy.
SetPartitionKeyAsPK	Whether or not to use the collection's Partition Key field as part of composite Primary Key for the corresponding exposed table.

SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
TypeDetectionScheme	Comma-separated options for how the provider will scan the data to determine the fields and datatypes in each document collection.
UseConnectionPooling	This property enables connection pooling.
UseRidAsPk	Set this property to false to switch using the id column as primary key instead the default _rid.
WriteThroughputBudget	Defines the Requests Units (RU) budget per Second that the Batch CUD (Create, Update, Delete) operations should not exceed.

For more details, see <https://cdn.cdata.com/help/EHG/jdbc/>.

15. COUCHBASE

15.1. Connection Options

15.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to Couchbase.
User	The Couchbase user account used to authenticate.
Password	The password used to authenticate the user.
CredentialsFile	Use this property if you need to provide credentials for multiple users or buckets. This file takes priority over other forms of authentication.
Server	The address of the Couchbase server or servers to which you are connecting.
CouchbaseService	Determines the Couchbase service to connect to. Default is N1QL. Available options are N1QL and Analytics.
ConnectionMode	Determines how to connect to the Couchbase server. Must be either Direct or Cloud.
DNSServer	Determines what DNS server to use when retrieving Couchbase Cloud information.
N1QLPort	The port for connecting to the Couchbase N1QL Endpoint.
AnalyticsPort	The port for connecting to the Couchbase Analytics Endpoint.
WebConsolePort	The port for connecting to the Couchbase Web Console.

15.1.2. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
UseSSL	Whether to negotiate TLS/SSL when connecting to the Couchbase server.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

15.1.3. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

15.1.4. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

15.1.5. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.

MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

15.1.6. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
Dataverse	Which Analytics dataverse to scan when discovering tables.
TypeDetectionScheme	Determines how the provider builds tables and columns from the buckets found in Couchbase.
InferNumSampleValues	The maximum number of values for every field to scan before determining its data type. Applies to Automatic Schema Discovery when TypeDetectionScheme is set to INFER.
InferSampleSize	The maximum number of documents to scan for the columns available in the bucket. Applies to Automatic Schema Discovery when TypeDetectionScheme is set to INFER.
InferSimilarityMetric	Specifies the similarity degree where different schemas will be considered to be the same flavor. Applies to Automatic

	Schema Discovery when TypeDetectionScheme is set to INFER.
FlexibleSchemas	Whether the provider allows queries to use columns that it has not discovered.
ExposeTTL	Specifies whether document TTL information should be exposed.
NumericStrings	Whether to allow string values to be treated as numbers.
IgnoreChildAggregates	Whether the provider exposes aggregate columns that are also available as child tables. Ignored if TableSupport is not set to Full.
TableSupport	How much effort the provider will put into discovering tables on the Couchbase server.
NewChildJoinsMode	Determines the kind of child table model the provider exposes.

15.1.7. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .

Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

15.1.8. Miscellaneous

Property	Description
AllowJSONParameters	Allows raw JSON to be used in parameters when QueryPassthrough is enabled.
BatchSize	The maximum size of each batch operation to submit.
ChildSeparator	The character or characters used to denote child tables.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Couchbase when the connection is opened.
CreateTableRamQuota	The default RAM quota, in megabytes, to use when inserting buckets via the CREATE TABLE syntax.
DataverseSeparator	The character or characters used to denote Analytics dataverses and scopes/collections.
FlattenArrays	The number of elements to expose as columns from nested arrays. Ignored if IgnoreChildAggregates is enabled.
FlattenObjects	Set FlattenObjects to true to flatten object properties into columns of their own. Otherwise, objects nested in arrays are returned as strings of JSON.
FlavorSeparator	The character or characters used to denote flavors.
GenerateSchemaFiles	Indicates the user preference as to when schemas should be generated and saved.

InsertNullValues	Determines whether an INSERT should include fields that have NULL values.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from Couchbase.
PeriodsSeparator	The character or characters used to denote hierarchy.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
QueryExecutionTimeout	This sets the server-side timeout for the query, which governs how long Couchbase will execute the query before returning a timeout error.
QueryPassthrough	This option passes the query to the Couchbase server as is.
Readonly	You can use this property to enforce read-only access to Couchbase from the provider.
RowScanDepth	The maximum number of rows to scan to look for the columns available in a table.
RTK	The runtime key used for licensing.
StrictComparison	Adjusts how precisely to translate filters on SQL input queries into Couchbase queries. This can be set to a

	comma-separated list of values, where each value can be one of: date, number, boolean, or string.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
TransactionDurability	Specifies how a document must be stored for a transaction to succeed. Specifies whether to use N1QL transactions when executing queries.
TransactionTimeout	This sets the amount of time a transaction may execute before it is timed out by Couchbase.
UpdateNullValues	Determines whether an UPDATE writes NULL values as NULL, or removes them.
UseCollectionsForDDL	Whether to assume that CREATE TABLE statements use collections instead of flavors. Only takes effect when connecting to Couchbase v7+ and GenerateSchemaFiles is set to OnCreate.
UseConnectionPooling	This property enables connection pooling.
UseTransactions	Specifies whether to use N1QL transactions when executing queries.
ValidateJSONParameters	Allows the provider to validate that string parameters are valid JSON before sending the query to Couchbase.

For more details, see <https://cdn.cdata.com/help/CKG/jdbc/>.

16. CSV

16.1. Connection Options

16.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to remote services.
AccessKey	Your account access key. This value is accessible from your security credentials page.
SecretKey	Your account secret key. This value is accessible from your security credentials page.
ApiKey	The API Key used to identify the user to IBM Cloud.
User	The CSV user account used to authenticate.
Password	The password used to authenticate the user.
SharePointEdition	The edition of SharePoint being used. Set either SharePointOnline or SharePointOnPremise.

16.1.2. Connection

Property	Description
URI	The Uniform Resource Identifier (URI) for the CSV resource location.
Region	The hosting region for your S3-like Web Services.
ProjectId	The id of the project where your Google Cloud Storage instance resides.

OracleNamespace	The Oracle Cloud Object Storage namespace to use.
StorageBaseURL	The URL of a cloud storage service provider.
SimpleUploadLimit	This setting specifies the threshold, in bytes, above which the provider will choose to perform a multipart upload rather than uploading everything in one request.
UseVirtualHosting	If true (default), buckets will be referenced in the request using the <code>hosted-style</code> request: <code>http://yourbucket.s3.amazonaws.com/yourobject</code> . If set to false, the bean will use the <code>path-style</code> request: <code>http://s3.amazonaws.com/yourbucket/yourobject</code> . Note that this property will be set to false, in case of an S3 based custom service when the CustomURL is specified.

16.1.3. AWS Authentication

Property	Description
AWSAccessKey	Your AWS account access key. This value is accessible from your AWS security credentials page.
AWSSecretKey	Your AWS account secret key. This value is accessible from your AWS security credentials page.
AWSRoleARN	The Amazon Resource Name of the role to use when authenticating.
AWSPrincipalArn	The ARN of the SAML Identity provider in your AWS account.
AWSRegion	The hosting region for your Amazon Web Services.
AWSSessionToken	Your AWS session token.
MFASerialNumber	The serial number of the MFA device if one is being used.
MFAToken	The temporary token available from your MFA device.

16.1.4. Azure Authentication

Property	Description
AzureStorageAccount	The name of your Azure storage account.
AzureAccessKey	The storage key associated with your CSV account.
AzureSharedAccessSignature	A shared access key signature that may be used for authentication.
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.
AzureEnvironment	The Azure Environment to use when establishing a connection.

16.1.5. SSO

Property	Description
SSOLoginURL	The identity provider's login URL.
SSOProperties	Additional properties required to connect to the identity provider in a semicolon-separated list.

16.1.6. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthVersion	The version of OAuth being used.

OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthAccessTokenSecret	The OAuth access token secret for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
OAuthGrantType	The grant type for the OAuth flow.
OAuthPasswordGrantMode	How to pass Client ID and Secret with OAuthGrantType is set to Password.
OAuthIncludeCallbackURL	Whether to include the callback URL in an access token request.
OAuthAuthorizationURL	The authorization URL for the OAuth service.
OAuthAccessTokenURL	The URL to retrieve the OAuth access token from.
OAuthRefreshTokenURL	The URL to refresh the OAuth token from.
OAuthRequestTokenURL	The URL the service provides to retrieve request tokens from. This is required in OAuth 1.0.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
AuthToken	The authentication token used to request and obtain the OAuth Access Token.

AuthKey	The authentication secret used to request and obtain the OAuth Access Token.
OAuthParams	A comma-separated list of other parameters to submit in the request for the OAuth access token in the format paramname=value.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

16.1.7. JWT OAuth

Property	Description
OAuthJWTCert	The JWT Certificate store.
OAuthJWTCertType	The type of key store containing the JWT Certificate.
OAuthJWTCertPassword	The password for the OAuth JWT certificate.
OAuthJWTCertSubject	The subject of the OAuth JWT certificate.
OAuthJWTIssuer	The issuer of the Java Web Token.
OAuthJWTSubject	The user subject for which the application is requesting delegated access.

16.1.8. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.

KerberosRealm	The Kerberos Realm used to authenticate the user with.
KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosServiceRealm	The Kerberos realm of the service.
KerberosServiceKDC	The Kerberos KDC of the service.
KerberosTicketCache	The full file path to an MIT Kerberos credential cache file.

16.1.9. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLMode	The authentication mechanism to be used when connecting to the FTP or FTPS server.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

16.1.10. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertType	The type of SSHClientCert certificate.

16.1.11. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

16.1.12. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

16.1.13. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.

MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

16.1.14. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
SchemaIniLocation	A path to the directory that contains the schema.ini file.
AggregateFiles	When set to true, the provider will aggregate all of the files located in the URI directory into a single table called AggregatedFiles .
MetadataDiscoveryURI	Used together with AggregateFiles , this property specifies a specific file to read the schema of the AggregatedFiles result set.
TypeDetectionScheme	Determines how to determine the data types of columns.
ColumnCount	The number of columns to detect when dynamically determining columns for the table.
RowScanDepth	The number of rows to scan when dynamically determining columns for the table.

16.1.15. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

16.1.16. Data Formatting

Property	Description
IncludeColumnHeaders	Whether to get column names from the first line of the specified files.
FMT	The format to be used to parse all text files.
ExtendedProperties	The Microsoft Jet OLE DB 4.0-compatible extended properties for text files.

RowDelimiter	The character which will be used to detect the end of a CSV row.
SkipTop	Skips the amount of rows specified starting from the top.
IgnoreBlankRows	Indicates whether to skip the empty rows.
IncludeEmptyHeaders	Whether to include empty value for a column name within column header.
SkipHeaderComments	If set to true, skips rows at the top of the file beginning with #.
Charset	Specifies the session character set for encoding and decoding character data transferred to and from the CSV file. The default value is UTF-8.
QuoteEscapeCharacter	Determines the character which will be used to escape quotes.
QuoteCharacter	Determines the character which will be used to quote values.
TrimSpaces	Set to True if you want the provider to trim preceeding and trailing spaces in a cell containing a quoted value.
PushEmptyValuesAsNull	Indicates whether to read the empty values as empty or as null.
NullValues	A comma separated list which will be replaced with nulls if there are found in the CSV file.
PathSeparator	Determines the character which will be used to replace the file separator.
IgnoreIncompleteRows	Indicates whether to ignore incomplete rows.
MaxCellLength	Max length a cell can hold, after passing this number the cell content will get truncated. If value is -1 then we don't limit the cell content length.

16.1.17. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ClientCulture	This property can be used to specify the format of data (e.g., currency values) that is accepted by the client application. This property can be used when the client application does not support the machine's culture settings. For example, Microsoft Access requires 'en-US'.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the CSV when the connection is opened.
Culture	This setting can be used to specify culture settings that determine how the provider interprets certain data types that are passed into the provider. For example, setting Culture='de-DE' will output German formats even on an American machine.
CustomHeaders	Other headers as determined by the user (optional).
CustomUrlParams	The custom query string to be included in the request.
DirectoryRetrievalDepth	Limit the subfolders recursively scanned when IncludeSubdirectories is enabled.
ExcludeFileExtensions	Set to true if file extensions should be excluded from table names.
ExcludeFiles	Comma-separated list of file extensions to exclude from the set of the files modeled as tables.

GenerateSchemaFiles	Indicates the user preference as to when schemas should be generated and saved.
IncludeDropboxTeamResources	Indicates if you want to include Dropbox team files and folders.
IncludeFiles	Comma-separated list of file extensions to include into the set of the files modeled as tables.
IncludeSubdirectories	Whether to read files from nested folders. In the case of a name collision, table names are prefixed by the underscore-separated folder names.
InsertMode	Specifies the mode for inserting data into CSV files.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from CSV.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
Readonly	You can use this property to enforce read-only access to CSV from the provider.
RTK	The runtime key used for licensing.

SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
TruncateOnInserts	Set to True if you want the provider to truncate on every (batch) insert.
UseConnectionPooling	This property enables connection pooling.
UseRowNumbers	Set this to true if you are deleting or updating in CSV and you do not want to specify a custom schema. This will create a new column with the name RowNumber which will be used as key for that table.

For more details, see <https://cdn.cdata.com/help/RVG/jdbc/>.

17. DATABRICKS

17.1. Connection Options

17.1.1. [Authentication](#)

Property	Description
AuthScheme	The authentication scheme used. Accepted entries are PersonalAccessToken, AzureServicePrincipal.
Server	The host name or IP address of the server hosting the Databricks database.
ProtocolVersion	The Protocol Version used to authenticate with Databricks.
Database	The name of the Databricks database.
HTTPPath	The path component of the URL endpoint.
Token	The token used to access the Databricks server.

17.1.2. [AWS Authentication](#)

Property	Description
AWSAccessKey	Your AWS account access key. This value is accessible from your AWS security credentials page.
AWSSecretKey	Your AWS account secret key. This value is accessible from your AWS security credentials page.
AWSRegion	The hosting region for your Amazon Web Services.
AWSS3Bucket	The name of your AWS S3 bucket.

17.1.3. [Azure Authentication](#)

Property	Description
<u>AzureStorageAccount</u>	The name of your Azure storage account.
<u>AzureAccessKey</u>	The storage key associated with your Databricks account.
<u>AzureBlobContainer</u>	The name of your Azure Blob storage container.

17.1.4. [AzureServicePrincipal Authentication](#)

Property	Description
<u>AzureTenantId</u>	The Tenant id of your Microsoft Azure Active Directory.
<u>AzureClientId</u>	The application(client) id of your Microsoft Azure Active Directory application.
<u>AzureClientSecret</u>	The application(client) secret of your Microsoft Azure Active Directory application.
<u>AzureSubscriptionId</u>	The Subscription id of your Azure Databricks Service Workspace.
<u>AzureResourceGroup</u>	The Resource Group name of your Azure Databricks Service Workspace.
<u>AzureWorkspace</u>	The name of your Azure Databricks Service Workspace.

17.1.5. [SSL](#)

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

17.1.6. [Firewall](#)

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

17.1.7. [Proxy](#)

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

17.1.8. [Logging](#)

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.

LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

17.1.9. [Schema](#)

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
PrimaryKeyIdentifiers	Set this property to define primary keys.

17.1.10. [Caching](#)

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver used to cache data.

CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

17.1.11. [Databricks](#)

Property	Description
CloudStorageType	Determine which cloud storage service will be used.
StoreTableInCloud	This option specifies whether Databricks server will create and save tables in cloud storage.
QueryTableDetails	Specifies whether to use DESCRIBE FORMATTED ... to query detailed table information, the query will take a long time if set it to True.
UseUploadApi	This option specifies whether the Databricks Upload API will be used when executing Bulk Insert operations.
UseCloudFetch	This option specifies whether to use CloudWatch to improve query efficiency when the data volume of the table is large.
UseLegacyDataModel	This option specifies whether to support Unity Catalog.

17.1.12. Miscellaneous

Property	Description
ApplicationName	The application name connection string property expresses the HTTP User-Agent.
AsyncQueryTimeout	The timeout for asynchronous requests issued by the provider to download large result sets.
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property specifies whether to connect to the Databricks when the connection is opened.
DescribeCommand	The describe command used to communicate with the Hive server. Accepted entries are DESCRIBE and DESC.
DetectView	Specifies whether to use DESCRIBE FORMATTED ... to detect the specified table is view or not.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.

QueryPassthrough	This option passes the query to the Databricks server as is.
Readonly	You can use this property to enforce read-only access to Databricks from the provider.
RTK	The runtime key used for licensing.
ServerConfigurations	A name-value list of server configuration variables to override the server defaults.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.
UseDescTableQuery	This option specifies whether the columns will be retrieved using a DESC TABLE query or the GetColumns Thrift API. The GetColumns Thrift API works for the Apache Spark 3.0.0 or later.
UseInsertSelectSyntax	Specifies whether to use an INSERT INTO SELECT statement.
UserDefinedViews	A filepath pointing to the JSON configuration file containing your custom views.

For more details, see [CData JDBC Driver for Databricks - CData JDBC Driver for Databricks](#).

18. DYNAMICS 365

18.1. Microsoft Dynamics 365 Version Support

The driver supports read/write access to Microsoft Dynamics 365 through OData Web services.

18.2. Connection Options

18.2.1. [Authentication](#)

Property	Description
AuthScheme	Specifies the type of authentication to use when connecting to Microsoft Dynamics 365. If this property is left blank, the default authentication is used.
OrganizationURL	The URL to your Dynamics 365 organization, sometimes referred to as the resource. To find the correct URL for your Microsoft Dynamics 365 instance, search the Web Services page.
Edition	Specifies the Microsoft Dynamics 365 edition in use. If not specified, the default (FinOpsOnline) is assumed.
ADFSServer	Identifies the FinOpsOnPremise ADFS Server to be used for authentication. To find this server name, enter echo %logonserver% at the command prompt.
ApiVersion	Identifies the API version being used. Keep this setting up to date to ensure that the full range of available API features are accessed during authentication.

18.2.2. [Azure Authentication](#)

Property	Description
AzureTenant	Identifies the Microsoft Dynamics 365 tenant being used to access data, either by name (for example, contoso.omnicrosoft.com) or ID. (Conditional).
AzureEnvironment	Specifies the Azure network environment to which you will connect. Must be the same network to which your Azure account was added.

18.2.3. [OAuth](#)

Property	Description
InitiateOAuth	Specifies the process for obtaining or refreshing the OAuth access token, which maintains user access while an authenticated, authorized user is working.
OAuthClientId	Specifies the client Id that was assigned the custom OAuth application was created. (Also known as the consumer key.) This ID registers the custom application with the OAuth authorization server.
OAuthClientSecret	Specifies the client secret that was assigned when the custom OAuth application was created. (Also known as the consumer secret). This secret registers the custom application with the OAuth authorization server.
OAuthAccessToken	A token received after authentication to the OAuth network, granting the user access. The access token is used in place of the user's login ID and password, which stay on the server.
OAuthSettingsLocation	Specifies the location of the settings file where OAuth values are saved. Storing OAuth settings in a central location avoids the need for users to enter OAuth connection properties manually each time they log in. It also enables credentials to be shared across connections or processes.
CallbackURL	The URL users return to after authenticating to Microsoft Dynamics 365 via OAuth.
OAuthGrantType	Specifies the grant type for the chosen OAuth flow. This value should be the same as the grant_type that was set during OAuth custom application creation.
OAuthVerifier	Specifies a verifier code returned from the OAuthAuthorizationURL . Used when authenticating to OAuth on a headless server, where a browser can't be launched. Requires both OAuthSettingsLocation and OAuthVerifier to be set.
OAuthRefreshToken	Gets and refreshes the currently-active OAuth Access Token.
OAuthExpiresIn	Specifies the duration in seconds, of an OAuth Access Token's lifetime. The token can be reissued to keep access alive as long as the user keeps working.

[OAuthTokenTimestamp](#)

Displays a Unix epoch timestamp in milliseconds that shows how long ago the current Access Token was created.

18.2.4. [JWT OAuth](#)

Property	Description
OAuthJWTCert	The JWT Certificate store.
OAuthJWTCertType	The type of key store containing the JWT Certificate.
OAuthJWTCertPassword	The password for the OAuth JWT certificate used to access a certificate store that requires a password. If the certificate store does not require a password, leave this property blank.
OAuthJWTCertSubject	The subject of the OAuth JWT certificate used to locate a matching certificate in the store. Supports partial matches and the wildcard '*' to select the first certificate.
OAuthJWTIssuer	The issuer of the Java Web Token.
OAuthJWTSubject	The user subject for which the application is requesting delegated access.

18.2.5. [SSL](#)

Property	Description
SSLServerCert	Specifies the certificate to be accepted from the server when connecting using TLS/SSL.

18.2.6. [Firewall](#)

Property	Description

FirewallType	Specifies the protocol the provider uses to tunnel traffic through a proxy-based firewall.
FirewallServer	Identifies the IP address, DNS name, or host name of a proxy used to traverse a firewall and relay user queries to network resources.
FirewallPort	Specifies the TCP port to be used for a proxy-based firewall.
FirewallUser	Identifies the user ID of the account authenticating to a proxy-based firewall.
FirewallPassword	Specifies the password of the user account authenticating to a proxy-based firewall.

18.2.7. [Proxy](#)

Property	Description
ProxyAutoDetect	Specifies whether the provider checks your system proxy settings for existing proxy server configurations, rather than using a manually specified proxy server.
ProxyServer	The hostname or IP address of the proxy server that you want to route HTTP traffic through.
ProxyPort	The TCP port on your specified proxy server (set in the ProxyServer connection property) that has been reserved for routing HTTP traffic to and from the client.
ProxyAuthScheme	Specifies the authentication method the provider uses when authenticating to the proxy server specified in the ProxyServer connection property.
ProxyUser	The username of a user account registered with the proxy server specified in the ProxyServer connection property.
ProxyPassword	The password associated with the user specified in the ProxyUser connection property.

ProxySSLType	The SSL type to use when connecting to the proxy server specified in the ProxyServer connection property.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the proxy server set in the ProxyServer connection property.

18.2.8. [Logging](#)

Property	Description
Logfile	Specifies the file path to the log file where the provider records its activities, such as authentication, query execution, and connection details.
Verbosity	Specifies the verbosity level of the log file, which controls the amount of detail logged. Supported values range from 1 to 5.
LogModules	Specifies the core modules to include in the log file. Use a semicolon-separated list of module names. By default, all modules are logged.
MaxLogFileSize	Specifies the maximum size of a single log file in bytes. For example, '10 MB'. When the file reaches the limit, the provider creates a new log file with the date and time appended to the name.
MaxLogFileCount	Specifies the maximum number of log files the provider retains. When the limit is reached, the oldest log file is deleted to make space for a new one.

18.2.9. [Schema](#)

Property	Description
Location	Specifies the location of a directory containing schema files that define tables, views, and stored procedures. Depending on your service's requirements, this may be expressed as either an absolute path or a relative path.

BrowsableSchemas	Optional setting that restricts the schemas reported to a subset of all available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC .
Tables	Optional setting that restricts the tables reported to a subset of all available tables. For example, Tables=TableA,TableB,TableC .
Views	Optional setting that restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC .

18.2.10. [Caching](#)

Property	Description
AutoCache	Specifies whether the content of tables targeted by SELECT queries is automatically cached to the specified cache database.
CacheDriver	The driver class of a JDBC driver. The specified driver is used to connect to the target database for all caching operations.
CacheConnection	The connection string for the specified cache database.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance, in seconds, for stale data in the specified cache database.
Offline	Get the data from the specified cache database instead of live Microsoft Dynamics 365 data.
CacheMetadata	Determines whether the provider caches table metadata to a file-based cache database.

18.2.11. [Miscellaneous](#)

Property	Description
AllowDuplicateRecords	Notes whether or not the driver will permit the adding of duplicate records during INSERT operations. To create the record (or update it

if a duplicate is found), set this to true. To avoid adding duplicate records, set this to false.

[BatchSize](#)

Specifies the maximum number of rows included in each batch operation. Set to 0 to submit the entire batch as a single request.

[ConnectionLifeTime](#)

Specifies the maximum lifetime of a connection in seconds. When the specified time elapses, the provider closes the connection. Set to 0 or leave empty for no time limit.

[ConnectOnOpen](#)

Specifies whether the provider establishes a connection to Microsoft Dynamics 365 immediately upon opening the connection. The default value is false. Enable this property only if immediate connectivity verification is necessary.

[ContinueOnError](#)

Specifies whether batch requests should continue after encountering an error. (OData 4.0 and higher only.).

[CrossCompany](#)

Specifies whether data should be pulled only the user's default company, or for all companies. (FinOpsOnline, FinOpsOnPremise, or HumanResources only.).

[CustomUrlParams](#)

A string of custom URL parameters to be included with the HTTP request, in the form field1=value1&field2=value2&field3=value3.

[EnableAtomicBatchOperations](#)

Specifies whether or not batch Create, Update, and Delete (CUD) operations execute independently (if one request fails, the others in the batch continue to execute) or atomically (if one request fails, the whole batch operation fails).

[IncludeFormattedValues](#)

Specifies whether columns that hold formatted values should be exposed.

[IncludeLookUpLogicalName](#)

Indicates if the referenced entity name for a LookUp field should be exposed.

[IncludeNavigationKeys](#)

Specifies whether the column listing should include the primary keys of navigation properties. (JSON responses only).

[IncludeNavigationProperties](#)

Specifies whether the column listing should include navigation properties.

[MaxRows](#)

Specifies the maximum rows returned for queries without aggregation or GROUP BY.

MinimumColumnSize	Specifies the minimum size of an Edm.String column, expressed as a number of characters. If the column size reported by Microsoft Dynamics 365 has a value lower than this value, the CData ADO.NET Provider for Microsoft Dynamics 365 overrides the service value with the one defined in the connection string.
Other	Specifies additional hidden properties for specific use cases. These are not required for typical provider functionality. Use a semicolon-separated list to define multiple properties.
Pagesize	Specifies the maximum number of results to return from Microsoft Dynamics 365, per page. This setting overrides the default page size set by the datasource, which is optimized for most use cases.
PoolIdleTimeout	Specifies the maximum idle time, in seconds, that a connection can remain in the pool before being closed.
PoolMaxSize	Specifies the maximum number of connections allowed in the connection pool. The default value is 100. Set to 0 or a negative value to disable pooling.
PoolMinSize	Specifies the minimum number of connections maintained in the connection pool. The default value is 1.
PoolWaitTime	Specifies the maximum number of seconds a connection request waits for an available connection in the pool. If the wait exceeds this time, an error is returned. The default is 60 seconds.
PseudoColumns	Specifies the pseudocolumns to expose as table columns. Use the format 'TableName=ColumnName;TableName=ColumnName'. The default is an empty string, which disables this property.
Readonly	Toggles read-only access to Microsoft Dynamics 365 from the provider.
RTK	Specifies the runtime key for licensing the provider. If unset or invalid, the provider defaults to the standard licensing method. Use this property only in environments where the standard licensing method is unsupported or requires a runtime key.
Timeout	Specifies the maximum time, in seconds, that the provider waits for a server response before throwing a timeout error. The default is 60 seconds. Set to 0 to disable the timeout.

UseClientSidePaging	Specifies whether or not Microsoft Dynamics 365 should return data using client-side paging, if your source supports it. If set to false, Microsoft Dynamics 365 uses server-side paging.
UseConnectionPooling	Enables the connection pooling feature, which allows the provider to reuse existing connections instead of creating new ones for each request.
UseDynamicStoredProcedures	Indicates whether or not stored procedures should be listed dynamically.
UseEntityDefinitions	Specifies that queries should use entity definition metadata endpoints, rather than the standard \$metadata endpoint.
UserDefinedViews	Specifies a filepath to a JSON configuration file defining custom views. The provider automatically detects and uses the views specified in this file.

For more details, see [CData JDBC Driver for Microsoft Dynamics 365](#).

19. ELASTICSEARCH

19.1. Elasticsearch Version Support

The driver models Elasticsearch data as a read/write, relational database. The driver can connect to Elasticsearch v2.2.0 and above via the REST API.

19.2. Connection Options

19.2.1. Authentication

Property	Description
AuthScheme	The scheme used for authentication. Accepted entries are None, Basic, Negotiate (Kerberos), AwsRootKeys, AwsIAMRoles, and APIKey. None is the default.
User	The user who is authenticating to Elasticsearch.
Password	The password used to authenticate to Elasticsearch.
Server	The host name or IP address of the Elasticsearch REST server.
Port	The port for the Elasticsearch REST server.
APIKey	The APIKey used to authenticate to Elasticsearch.
APIKeyId	The APIKey Id to authenticate to Elasticsearch.

19.2.2. Connection

Property	Description
DataModel	Specifies the data model to use when parsing Elasticsearch documents and generating the database metadata.

19.2.3. AWS Authentication

Property	Description
AWSAccessKey	Your AWS account access key. This value is accessible from your AWS security credentials page.
AWSSecretKey	Your AWS account secret key. This value is accessible from your AWS security credentials page.
AWSRoleARN	The Amazon Resource Name of the role to use when authenticating.
AWSRegion	The hosting region for your Amazon Web Services.
AWSSessionToken	Your AWS session token.
TemporaryTokenDuration	The amount of time (in seconds) an AWS temporary token will last.

19.2.4. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
KerberosRealm	The Kerberos Realm used to authenticate the user with.
KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosServiceRealm	The Kerberos realm of the service.
KerberosServiceKDC	The Kerberos KDC of the service.

KerberosTicketCache	The full file path to an MIT Kerberos credential cache file.

19.2.5. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

19.2.6. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

19.2.7. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

19.2.8. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).

MaxLogFileCount	A string specifying the maximum file count of log files.

19.2.9. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
FlattenObjects	Set FlattenObjects to true to flatten object properties into columns of their own. Otherwise, objects nested in arrays are returned as strings of JSON.
FlattenArrays	Set FlattenArrays to the number of nested array elements you want to return as table columns. By default, nested arrays are returned as strings of JSON.

19.2.10. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.

CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

19.2.11. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ClientSideEvaluation	Set ClientSideEvaluation to true to perform Evaluation client side on nested objects.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Elasticsearch when the connection is opened.
GenerateSchemaFiles	Indicates the user preference as to when schemas should be generated and saved.
MaxResults	The maximum number of total results to return from Elasticsearch when using the default Search API.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.

Other	These hidden properties are used only in specific use cases.
PageSize	The number of results to return per request from Elasticsearch.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
QueryPassthrough	This option allows you to pass exact queries to Elasticsearch.
Readonly	You can use this property to enforce read-only access to Elasticsearch from the provider.
RowScanDepth	The maximum number of rows to scan when generating table metadata. Set this property to gain more control over how the provider detects arrays.
RTK	The runtime key used for licensing.
ScrollDuration	Specifies the time unit to use when retrieving results via the Scroll API.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/EEG/jdbc/>.

20. EXCEL

20.1. Excel Version Support

The driver supports the Office Open XML spreadsheets used by Excel 2007 and later.

20.2. Connection Options

20.2.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to remote services.
User	The Excel user account used to authenticate.
Password	The password used to authenticate the user.
SharePointEdition	The edition of SharePoint being used. Set either SharePointOnline or SharePointOnPremise.

20.2.2. Connection

Property	Description
URI	The Uniform Resource Identifier (URI) for the Excel resource location.
DefineTables	Map Excel ranges to table names.
Header	Indicates whether the first row should be used as a column header.
Orientation	Indicates whether the data in Excel is laid out horizontally or vertically.

ExcelFile	The location of an Excel file.
BufferChanges	Indicates whether to hold changes to the data in memory until the connection is closed.
Region	The hosting region for your S3-like Web Services.
ProjectId	The id of the project where your Google Cloud Storage instance resides.
OracleNamespace	The Oracle Cloud Object Storage namespace to use.
StorageBaseURL	The URL of a cloud storage service provider.
UseVirtualHosting	If true (default), buckets will be referenced in the request using the <code>hosted-style</code> request: <code>http://yourbucket.s3.amazonaws.com/yourobject</code> . If set to false, the bean will use the <code>path-style</code> request: <code>http://s3.amazonaws.com/yourbucket/yourobject</code> . Note that this property will be set to false, in case of an S3 based custom service when the CustomURL is specified.

20.2.3. AWS Authentication

Property	Description
AWSAccessKey	Your AWS account access key. This value is accessible from your AWS security credentials page.
AWSSecretKey	Your AWS account secret key. This value is accessible from your AWS security credentials page.
AWSRoleARN	The Amazon Resource Name of the role to use when authenticating.
AWSPrincipalArn	The ARN of the SAML Identity provider in your AWS account.
AWSRegion	The hosting region for your Amazon Web Services.
AWSSessionToken	Your AWS session token.

MFASerialNumber	The serial number of the MFA device if one is being used.
MFAToken	The temporary token available from your MFA device.

20.2.4. Azure Authentication

Property	Description
AzureStorageAccount	The name of your Azure storage account.
AzureAccessKey	The storage key associated with your Excel account.
AzureSharedAccessSignature	A shared access key signature that may be used for authentication.
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.
AzureEnvironment	The Azure Environment to use when establishing a connection.

20.2.5. SSO

Property	Description
SSOLoginURL	The identity provider's login URL.
SSOProperties	Additional properties required to connect to the identity provider in a semicolon-separated list.

20.2.6. Data

Property	Description
AllowFormula	Whether or not to treat values starting with an equals (=) sign as formulas during inserts and updates.
EmptyValueMode	Indicates whether to read the empty values as empty or as null.
NullValues	A comma separated list which will be replaced with nulls if there are found in the CSV file.
IgnoreCalcError	Indicates whether to ignore errors that occurred during the calculation.
NullValueMode	Indicates whether to read empty cells as null or as empty.
HasCrossSheetReferences	Indicates how cross sheet references are handled.
Recalculate	Indicates whether to recalculate all formulas when data is read.
ShowEmptyRows	Indicates whether or not the empty rows should be pushed.
TypeDetectionScheme	Determines how the provider detects the data types of columns.
UseFastInsert	Indicates whether to use the fastInsert when executing Insert Statement.

20.2.7. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.

OAuthVersion	The version of OAuth being used.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthAccessTokenSecret	The OAuth access token secret for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
OAuthGrantType	The grant type for the OAuth flow.
OAuthPasswordGrantMode	How to pass Client ID and Secret with OAuthGrantType is set to Password.
OAuthIncludeCallbackURL	Whether to include the callback URL in an access token request.
OAuthAuthorizationURL	The authorization URL for the OAuth service.
OAuthAccessTokenURL	The URL to retrieve the OAuth access token from.
OAuthRefreshTokenURL	The URL to refresh the OAuth token from.
OAuthRequestTokenURL	The URL the service provides to retrieve request tokens from. This is required in OAuth 1.0.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
AuthToken	The authentication token used to request and obtain the OAuth Access Token.

AuthKey	The authentication secret used to request and obtain the OAuth Access Token.
OAuthParams	A comma-separated list of other parameters to submit in the request for the OAuth access token in the format paramname=value.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

20.2.8. JWT OAuth

Property	Description
OAuthJWTCert	The JWT Certificate store.
OAuthJWTCertType	The type of key store containing the JWT Certificate.
OAuthJWTCertPassword	The password for the OAuth JWT certificate.
OAuthJWTCertSubject	The subject of the OAuth JWT certificate.
OAuthJWTIssuer	The issuer of the Java Web Token.
OAuthJWTSubject	The user subject for which the application is requesting delegated access.

20.2.9. Kerberos

Property	Description
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.

20.2.10. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLMode	The authentication mechanism to be used when connecting to the FTP or FTPS server.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

20.2.11. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertType	The type of SSHClientCert certificate.

20.2.12. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

20.2.13. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.

ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

20.2.14. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

20.2.15. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

20.2.16. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

20.2.17. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
Charset	Specifies the session character set for encoding and decoding character data transferred to and from the Excel file. The default value is UTF-8.
ClientCulture	This property can be used to specify the format of data (e.g., currency values) that is accepted by the

	client application. This property can be used when the client application does not support the machine's culture settings. For example, Microsoft Access requires 'en-US'.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Excel when the connection is opened.
Culture	This setting can be used to specify culture settings that determine how the provider interprets certain data types that are passed into the provider. For example, setting Culture='de-DE' will output German formats even on an American machine.
CustomHeaders	Other headers as determined by the user (optional).
CustomUrlParams	The custom query string to be included in the request.
IncludeDropboxTeamResources	Indicates if you want to include Dropbox team files and folders.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.

PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
Readonly	You can use this property to enforce read-only access to Excel from the provider.
RowScanDepth	The maximum number of rows to scan to look for the columns available in a table.
RTK	The runtime key used for licensing.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/RXG/jdbc/>.

21. FINANCIALFORCE

21.1. Connection Options

21.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to FinancialForce.
User	The FinancialForce user account used to authenticate.
Password	The password used to authenticate the user.
SecurityToken	The security token used to authenticate access to the FinancialForce account.
UseSandbox	A boolean determining if the connection should be made to a FinancialForce sandbox account.

21.1.2. Connection

Property	Description
APIVersion	The version of the FinancialForce API used.
LoginURL	URL to the FinancialForce server used for logging in.

21.1.3. SSO

Property	Description
SSOLoginURL	The identity provider's login URL.

SSOProperties	Additional properties required to connect to the identity provider in a semicolon-separated list.
SSOExchangeUrl	The url used for consuming the SAML response and exchanging it with FinancialForce specific credentials.

21.1.4. BulkAPI

Property	Description
UseBulkAPI	Whether to use the synchronous SOAP API or the asynchronous Bulk API.
BulkAPIVersion	The version of the bulk API to use for processing queries.
BulkAPIConcurrencyMode	The concurrency mode for processing bulk rows with BULK API v1.
BulkPollingInterval	The time interval in milliseconds between requests that check the availability of the bulk query response. The default value is 500 ms.
BulkQueryTimeout	The timeout in minutes for which the provider will wait for a bulk query response. The default value is 25 minutes.
WaitForBulkResults	Whether to wait for bulk results when using the asynchronous API. Only active when UseBulkAPI is true.

21.1.5. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.

OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
OAuthServerURL	The server URL to use if authenticating with OAuth.
OAuthGrantType	The grant type for the OAuth flow.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

21.1.6. SSL

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

21.1.7. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

21.1.8. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.

ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

21.1.9. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

21.1.10. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.

Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

21.1.11. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

21.1.12. Miscellaneous

Property	Description
AllOrNone	A boolean indicating if you would like all inserts, updates, or deletes to fail in a request if even a single record fails.

ArchiveMode	Boolean indicating whether to include deleted and archived records with a standard SELECT query.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the FinancialForce when the connection is opened.
ContinueOnAlterException	Whether you want to continue after a ALTER statement has failed.
FilterScope	Optional scope to limit the records returned from queries.
IncludeMetadataDescription	Set this property to a value other than NONE if you want to retrieve the descriptions for columns, tables or both of them from the Metadata API.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.

PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
Readonly	You can use this property to enforce read-only access to FinancialForce from the provider.
RTK	The runtime key used for licensing.
ServerSideAggregation	A boolean determining if server side aggregation should be used.
SessionTimeout	The time in minutes for which a FinancialForce login session is reused.
SkipFormulaFields	Set to true if formula fields should be skipped when listing columns.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.
UseDisplayNames	Boolean determining if the display names for the columns should be used instead of the API names.

For more details, see <https://cdn.cdata.com/help/HFG/jdbc/>.

22. EXCEL ONLINE



22.1. Connection Options

22.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to Excel Online.
UseSandbox	A boolean indicating if you are using a sandbox account. The provider makes requests to the production environment by default.

22.1.2. Connection

Property	Description
DefineTables	Assign table names to ranges.
Workbook	The name or Id of the workbook.
Drive	The Id of the drive.
IncludeSharePointSites	Whether to retrieve drives for all SharePoint sites when querying Drives view. If 'true' the provider will retrieve all Site IDs recursively and for each of them issue a separate call to get their drives. Therefore, be aware that setting this property to 'true' may decrease performance for Drives view.

22.1.3. Azure Authentication

Property	Description
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.
AzureEnvironment	The Azure Environment to use when establishing a connection.

22.1.4. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
OAuthGrantType	The grant type for the OAuth flow.

OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

22.1.5. SSL

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

22.1.6. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

22.1.7. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

22.1.8. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.

LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

22.1.9. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

22.1.10. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with

	CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

22.1.11. Miscellaneous

Property	Description
AllowFormula	Whether or not to treat values starting with an equals (=) sign as formulas during inserts and updates.
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Excel Online when the connection is opened.
Header	Indicates whether or not the provider should detect column names from the first row.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.

Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from Excel Online when UsePagination is True.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
Readonly	You can use this property to enforce read-only access to Excel Online from the provider.
RowScanDepth	The maximum number of rows to scan to look for the columns available in a table.
RTK	The runtime key used for licensing.
SharepointURL	The base URL of your Sharepoint Server.
ShowSharedDocuments	Whether or not to show shared documents.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
TypeDetectionScheme	Determines how to determine the data types of columns.
UseConnectionPooling	This property enables connection pooling.

UsePagination	Whether or not the CData ADO.NET Provider for Excel Online should use client side paging.
UseSimpleNames	Boolean determining if simple names should be used for tables and columns.
ValueInputOption	Determines how the provider parses values the user submits to Excel Online.
ValueRenderOption	Determines how the provider renders values in the output.

For more details, see <https://cdn.cdata.com/help/FXG/jdbc/>.

23. GOOGLE DATA CATALOG

23.1. Connection Options

23.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to Google Data Catalog.
OrganizationId	The ID associated with the Google Cloud Platform organization resource to connect to.
ProjectId	The ID associated with the Google Cloud Platform project resource you would like to connect to.

23.1.2. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.

CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

23.1.3. JWT OAuth

Property	Description
OAuthJWTCert	The JWT Certificate store.
OAuthJWTCertType	The type of key store containing the JWT Certificate.
OAuthJWTCertPassword	The password for the OAuth JWT certificate.
OAuthJWTCertSubject	The subject of the OAuth JWT certificate.
OAuthJWTIssuer	The issuer of the Java Web Token.
OAuthJWTSubject	The user subject for which the application is requesting delegated access.

23.1.4. SSL

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

23.1.5. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

23.1.6. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.

ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

23.1.7. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

23.1.8. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

23.1.9. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .

Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

23.1.10. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Google Data Catalog when the connection is opened.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from Google Data Catalog.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.

PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
RTK	The runtime key used for licensing.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/HGG/jdbc/>.

24. GOOGLE SPANNER

24.1. Connection Options

24.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to Google Spanner.
InstancelId	The id of the Google Spanner instance to which you are connecting.
Database	The name of the Google Spanner database to connect to.
ProjectId	The id of the project where your Google Spanner instance resides.
UseGrpcEndpoint	Whether to use the Grpc endpoint or not.

24.1.2. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.

OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

24.1.3. JWT OAuth

Property	Description
OAuthJWTCert	The JWT Certificate store.
OAuthJWTCertType	The type of key store containing the JWT Certificate.
OAuthJWTCertPassword	The password for the OAuth JWT certificate.
OAuthJWTCertSubject	The subject of the OAuth JWT certificate.
OAuthJWTIssuer	The issuer of the Java Web Token.
OAuthJWTSubject	The user subject for which the application is requesting delegated access.

24.1.4. SSL

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

24.1.5. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

24.1.6. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.

ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

24.1.7. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

24.1.8. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
Schema	The schema which will be used by default.

24.1.9. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.

CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

24.1.10. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Google Spanner when the connection is opened.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.

PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
QueryPassthrough	This option passes the query to the Google Spanner server as is.
Readonly	You can use this property to enforce read-only access to Google Spanner from the provider.
RTK	The runtime key used for licensing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/OGG/jdbc/>.

25. HBASE

25.1. Apache HBase Version Support

The driver models HBase data as a read/write, relational database. The driver can connect to HBase REST Server 0.0.3, available in HBase version 0.98 and above.

25.2. Connection Options

25.2.1. Authentication

Property	Description
AuthScheme	The scheme used for authentication. Accepted entries are None, Basic, and Negotiate (Kerberos). None is the default.
Server	The host name or IP address of the Apache HBase REST server.
Port	The port for the Apache HBase REST server.
User	The user who is authenticating to Apache HBase.
Password	The password used to authenticate to Apache HBase.

25.2.2. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
KerberosRealm	The Kerberos Realm used to authenticate the user with.
KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.

KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosServiceRealm	The Kerberos realm of the service.
KerberosServiceKDC	The Kerberos KDC of the service.
KerberosTicketCache	The full file path to an MIT Kerberos credential cache file.

25.2.3. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

25.2.4. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.

FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

25.2.5. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

25.2.6. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

25.2.7. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

25.2.8. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

25.2.9. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ColumnEncoding	The Encoding used for the column values. Accepted entries are None, UTF8, and BASE64. None is the default.

ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Apache HBase when the connection is opened.
DatetimeFormat	The format used when inserting datetime values into the database.
IncludeDisabledTables	Specifies whether to retrieve the disabled table.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from Apache HBase.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
Readonly	You can use this property to enforce read-only access to Apache HBase from the provider.

ResetTableMetadataOnDelete	Specifies whether to reset the catch table after executing the Delete Statement.
RetrieveSelectedColumnsOnly	Specifies whether to retrieve selected columns only when executing a SELECT statement.
RowScanDepth	The maximum number of rows to scan to look for the columns available in a table.
RTK	The runtime key used for licensing.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
TypeDetectionScheme	Determines how to determine the data type of columns.
UseConnectionPooling	This property enables connection pooling.
UseSQLFiltering	Specifies whether to use the ScannerFilter when executing the Select Statement.

For more details, see <https://cdn.cdata.com/help/RHG/jdbc/>.

26. HIVE

26.1. Apache Hive Version Support

The driver models Apache Hive instances as relational databases. Hive versions 0.11.0 and above are supported.

26.2. Connection Options

26.2.1. Authentication

Property	Description
AuthScheme	The authentication scheme used. Accepted entries are Plain, LDAP, NoSasl, and Kerberos.
Server	The host name or IP address of the server hosting HiveServer2.
Port	The port for the connection to the HiveServer2 instance.
User	The username used to authenticate with Hive.
Password	The password used to authenticate with Hive.
ProtocolVersion	The Protocol Version used to authenticate with Hive.
TransportMode	The transport mode to use to communicate with the Hive server. Accepted entries are BINARY and HTTP.
Database	The name of the Hive database to use by default.
ImpersonationProxyUser	The proxy user of the Hive user impersonation.

26.2.2. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
KerberosRealm	The Kerberos Realm used to authenticate the user with.
SaslQop	Quality of protection for the SASL framework. The level of quality is negotiated between the client and server during authentication. Used by Kerberos authentication with TCP transport.
KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosServiceRealm	The Kerberos realm of the service.
KerberosServiceKDC	The Kerberos KDC of the service.
KerberosTicketCache	The full file path to an MIT Kerberos credential cache file.

26.2.3. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.

SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

26.2.4. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertSubject	The subject of the SSH client certificate.
SSHClientCertType	The type of SSHClientCert certificate.
SSHServer	The SSH server.
SSHPort	The SSH port.
SSHUser	The SSH user.
SSHPassword	The SSH password.
SSHServerFingerprint	The SSH server fingerprint.
UseSSH	Whether to tunnel the Apache Hive connection over SSH. Use SSH.

26.2.5. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.

FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

26.2.6. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

26.2.7. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

26.2.8. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

26.2.9. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

26.2.10. Miscellaneous

Property	Description
AsyncQueryTimeout	The timeout for asynchronous requests issued by the provider to download large result sets.
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.

ConnectOnOpen	This property species whether to connect to the Apache Hive when the connection is opened.
FailoverHosts	This property allows you to specify a list of failover hosts in addition to the one configured in Server and Port .
HTTPPath	The path component of the URL endpoint when using HTTP TransportMode.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from Apache Hive.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
QueryPassthrough	This option passes the query to the Apache Hive server as is.
Readonly	You can use this property to enforce read-only access to Apache Hive from the provider.
RTK	The runtime key used for licensing.

ServerConfigurations	A list of server configuration variables to override the server defaults.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.
UseDescTableQuery	This option specifies whether the columns will be retrieved using a DESC TABLE query or the GetColumns Thrift API.
UseShowDatabasesQuery	This option specifies whether the schemas will be retrieved using a SHOW DATABASES query or the GetSchemas Thrift API.
UseShowTablesQuery	This option specifies whether the tables will be retrieved using a SHOW TABLES query or the GetTables Thrift API.
UseSSL	Specifies whether to use SSL Encryption when connecting to Hive.
UseZookeeperDiscovery	Specifies whether to use ZooKeeper Service Discovery.
ZookeeperNamespace	The namespace configured on ZooKeeper for the Hive Server 2 znodes.

For more details, see <https://cdn.cdata.com/help/FIG/jdbc/>.

27. IBM DB2 DB2

Note: this driver does not support DB2 for z/OS. See [IBM DB2 for z/OS](#).

27.1. Connection Options

27.1.1. Authentication

Property	Description
AuthScheme	The authentication mechanism that the provider will use to authenticate with DB2.
Server	The name of the DB2 server.
Port	The port used to connect to the server hosting the DB2 database.
Database	The name of the DB2 database.
User	A database user.
Password	The user's password.
UseSSL	This field sets whether SSL is enabled.
AlternateServers	This property allows you to specify multiple servers in addition to the one configured in <code>Server</code> and <code>Port</code> . Specify both a server name and port; separate servers with a comma.

27.1.2. SSL

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

27.1.3. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertSubject	The subject of the SSH client certificate.
SSHClientCertType	The type of SSHClientCert certificate.
SSHServer	The SSH server.
SSHPort	The SSH port.
SSHUser	The SSH user.
SSHPassword	The SSH password.
SSHServerFingerprint	The SSH server fingerprint.
UseSSH	Whether to tunnel the DB2 connection over SSH. Use SSH.

27.1.4. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.

FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

27.1.5. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

27.1.6. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.

Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

27.1.7. Miscellaneous

Property	Description
AllowPreparedStatement	Prepare a query statement before its execution.
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the DB2 when the connection is opened.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.

PoolWaitTime	The max seconds to wait for an available connection.
QueryPassthrough	This option passes the query to the DB2 server as is.
Readonly	You can use this property to enforce read-only access to DB2 from the provider.
RTK	The runtime key used for licensing.
SwitchMode	This property allows you to specify a switching mode to select a server from AlternateServers as the active server.
SwitchStrategy	This property allows you to specify a switching strategy to select a server from AlternateServers as the active server.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/EDG/jdbc/>.

28. IBM DB2 FOR Z/OS

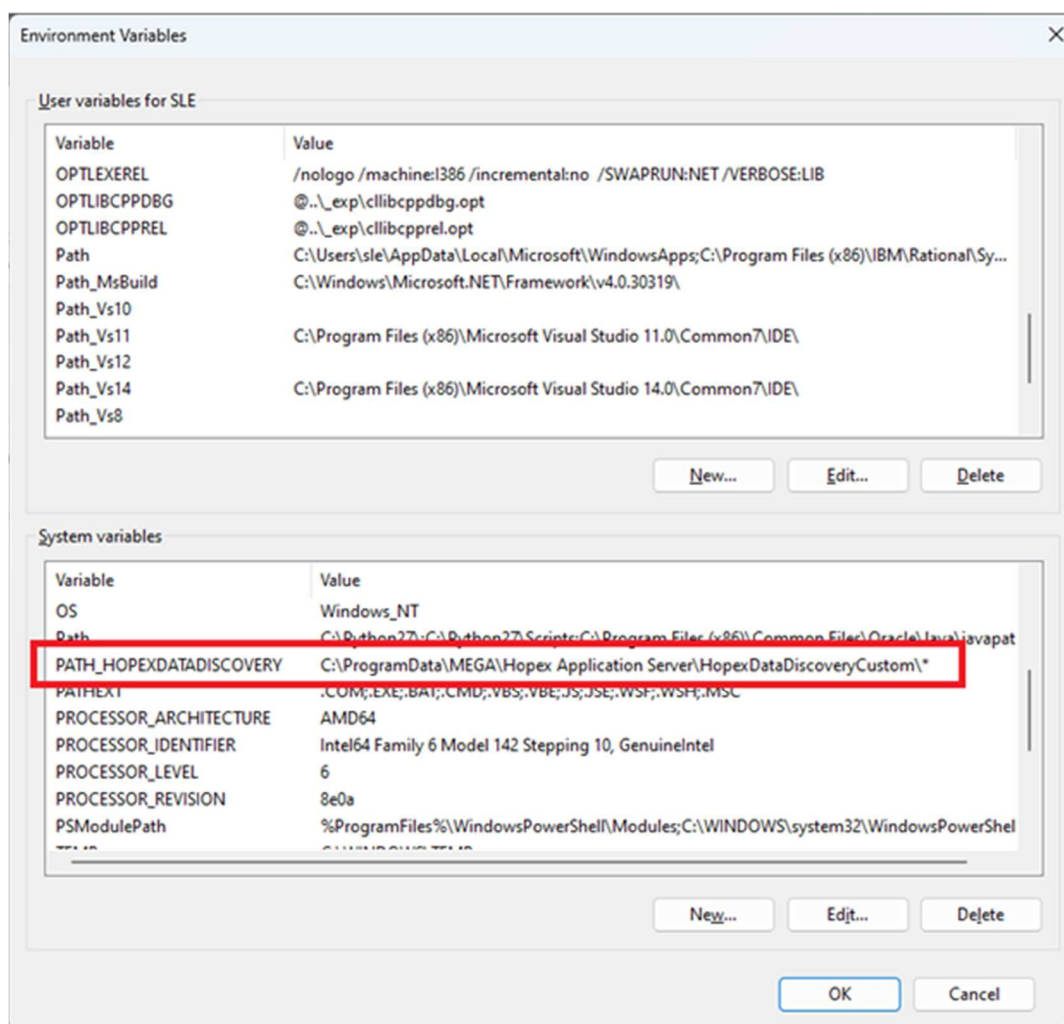
28.1. Prerequisite

1. Download the Db2 for z/OS driver: <https://www.ibm.com/support/pages/db2-jdbc-driver-versions-and-downloads>.

2. Place the Db2 for z/OS driver in a directory accessible from the HOPEX Data Discovery Module, but not in the module directory.

Example: "C:\ProgramData\MEGA\Hopex Application Server\HopexDataDiscoveryCustom"

3. Create the environment variable "PATH_HOPEXDATADISCOVERY" if it doesn't already exist and set it to the directory mentioned in step 2.



28.2. Connection Options

Property	Description
Database	A name for the data source. It's the Db2 location name that is defined during installation. All characters in the Db2 location name must be uppercase characters.
Server	The domain name or IP address of the data source
Port	The TCP/IP server port number that is assigned to the data source. This is an integer between 0 and 65535.
User	The user name for the connection
Password	The user's password for the connection.
DBNAME	For a table, or a view of tables, the name of the database that contains the table space that is named in TSNAME. For a created temporary table, an alias, or a view of a view, the value is DSNDB06.
specialRegisters	A list of special register settings for the JDBC connection. You can specify one or more special register name and value pairs. Special register name and value pairs must be delimited by commas (,).
globalSessionVariables	A list of session variable settings for the JDBC connection. You can specify one or more session variable name and value pairs. Session variable settings apply only to Db2 for z/OS Version 11 or later data servers. Session variable name and value pairs must be delimited by commas (,). The last pair must end with a semicolon (;).

queryAcceleration

Changes the value of the CURRENT QUERY ACCELERATION special register.

Possible values are:

NONE : Specifies that no query acceleration is done.

ENABLE : Specifies that queries are accelerated only if Db2 determines that it is advantageous to do so. If there is an accelerator failure while a query is running, or the accelerator returns an error, Db2 returns a negative SQLCODE to the application.

ENABLE WITH FAILBACK : Specifies that queries are accelerated only if Db2 determines that it is advantageous to do so. If the accelerator returns an error during the PREPARE or first OPEN for the query, Db2 executes the query without the accelerator. If the accelerator returns an error during a FETCH or a subsequent OPEN, Db2 returns the error to the user, and does not execute the query.

ELIGIBLE : Specifies that queries are accelerated if they are eligible for acceleration. Db2 does not use cost information to determine whether to accelerate the queries. Queries that are not eligible for acceleration are executed by Db2. If there is an accelerator failure while a query is running, or the accelerator returns an error, Db2 returns a negative SQLCODE to the application.

ALL : Specifies that queries are accelerated if they are eligible for acceleration. Db2 does not use cost information to determine whether to accelerate the queries. Queries that are not eligible for acceleration are not executed by Db2, and an SQL error is returned. If there is an accelerator failure while a query is running, or the accelerator returns an error, Db2 returns a negative SQLCODE to the application.

retrieveMessagesFromServerOnGetMessage

During a connection to a data server, set the retrieveMessagesFromServerOnGetMessage property to true if you want full message text from an SQLException.getMessage call. Default is false.

`emulateParameterMetaDataForZCalls`

This parameter enables the use of the method `PreparedStatement.getParameterMetaData()` against a DB2 for z/OS data source. Default is 0.

29. IBM INFORMIX

29.1. Informix Version Support

Compatible with IBM Informix 9.70 and newer. The driver supports industry-standard Distributed Relational Database Architecture (DRDA) connectivity only.

29.2. Connection Options

29.2.1. Authentication

Property	Description
AuthScheme	The authentication mechanism that the provider will use to authenticate with Informix.
Server	The name of the DB2 server.
Port	The port used to connect to the server hosting the database.
User	A database user.
Password	The user's password.
Database	The name of the DB2 database.
UseSSL	This field sets whether SSL is enabled.
AlternateServers	This property allows you to specify multiple servers in addition to the one configured in Server and Port . Specify both a server name and port; separate servers with a comma.

29.2.2. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

29.2.3. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertSubject	The subject of the SSH client certificate.
SSHClientCertType	The type of SSHClientCert certificate.
SSHServer	The SSH server.
SSHPort	The SSH port.

SSHUser	The SSH user.
SSHPassword	The SSH password.
SSHServerFingerprint	The SSH server fingerprint.
UseSSH	Whether to tunnel the Informix connection over SSH. Use SSH.

29.2.4. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

29.2.5. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.

MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

29.2.6. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

29.2.7. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Informix when the connection is opened.

MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
QueryPassthrough	This option passes the query to the Informix server as is.
Readonly	You can use this property to enforce read-only access to Informix from the provider.
RTK	The runtime key used for licensing.
SessionVariables	A semicolon-separated list of SET ENVIRONMENT commands to set on the current connection.
SwitchMode	This property allows you to specify a switching mode to select a server from AlternateServers as the active server.
SwitchStrategy	This property allows you to specify a switching strategy to select a server from AlternateServers as the active server.
Timeout	A timeout for connections in seconds.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/JIG/jdbc/default.htm>.

30. IMPALA

30.1. Connection Options

30.1.1. Authentication

Property	Description
AuthScheme	The authentication scheme used. Accepted entries are NoSasl, LDAP and Kerberos.
Server	The name of the server running SQL Server.
Port	The port for the connection to the Impala Server instance.
User	The username used to authenticate with Impala.
Password	The password used to authenticate with Impala.
ProtocolVersion	The Thrift protocol version to use when connecting to the Impala server.
Database	The name of the Impala database to use by default.
TransportMode	The transport mode to use to communicate with the Impala server. Accepted entries are BINARY and HTTP.

30.1.2. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
KerberosRealm	The Kerberos Realm used to authenticate the user with.

KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosServiceRealm	The Kerberos realm of the service.
KerberosServiceKDC	The Kerberos KDC of the service.
KerberosTicketCache	The full file path to an MIT Kerberos credential cache file.

30.1.3. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

30.1.4. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

30.1.5. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.

ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

30.1.6. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

30.1.7. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.

[Views](#)

Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

30.1.8. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

30.1.9. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.

ConnectOnOpen	This property specifies whether to connect to the Apache Impala when the connection is opened.
HTTPPath	The path component of the URL endpoint when using HTTP TransportMode.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from Apache Impala.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
QueryPassthrough	This option passes the query to the Apache Impala server as is.
Readonly	You can use this property to enforce read-only access to Apache Impala from the provider.
RTK	The runtime key used for licensing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.
UserDefinedViews	A filepath pointing to the JSON configuration file containing your custom views.

UseSSL

Specifies whether to use SSL Encryption when connecting to Impala.

For more details, see <https://cdn.cdata.com/help/GIH/jdbc/default.htm>.

31. JIRA

31.1. Connection Options

31.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to Jira Service Desk.
URL	The URL to your JIRA Service Desk endpoint.
User	The Jira Service Desk user account used to authenticate.
Password	The password used to authenticate the user.
APIToken	APIToken of the currently authenticated user.

31.1.2. SSO

Property	Description
SSOLoginURL	The identity provider's login URL.
SSOProperties	Additional properties required to connect to the identity provider in a semicolon-separated list.
SSOExchangeUrl	The url used for consuming the SAML response and exchanging it with Jira Service Desk specific credentials.
SSOAppName	App Name used with SSO for IdPs that require it.
SSOAppPassword	App Password used with SSO for IdPs that require it.

31.1.3. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthVersion	The version of OAuth being used.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
AuthToken	The authentication token used to request and obtain the OAuth Access Token.
AuthKey	The authentication secret used to request and obtain the OAuth Access Token.
CertificateStoreType	The type of certificate store used with Jira Service Desk Private Application authentication.
CertificateStore	The certificate store used for JIRA Service Desk authentication.
CertificateStorePassword	The password of the certificate store used with Jira Service Desk authentication.
CertificateSubject	The subject of the certificate used with Jira Service Desk Private Application authentication.

31.1.4. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

31.1.5. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

31.1.6. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

31.1.7. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.

Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

31.1.8. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

31.1.9. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .

CacheProvider	The name of the provider to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

31.1.10. Miscellaneous

Property	Description
DefaultDomain	This property is used for the Oracle Database Gateway for ODBC.
EnableForeignKeyDetection	Whether to detect the foreign keys in ODBC.
IncludeCustomFields	A boolean indicating if you would like to include custom fields in the column listing.
IncludeDualTable	Set this property to mock the Oracle DUAL table for better compatibility with Oracle database.
LimitKeySize	The maximum length of a primary key column.
MapBigintToVarchar	This property controls whether or not the bigint type maps to SQL_VARCHAR instead

	of SQL_BIGINT. This property is false by default.
MapToInt	This property controls whether or not the long type maps to SQL_INTEGER instead of SQL_BIGINT. This property is false by default.
MapToLongVarchar	This property controls whether or not a column is returned as SQL_LONGVARCHAR.
MapToWVarchar	This property controls whether or not string types map to SQL_WVARCHAR instead of SQL_VARCHAR. This property is set by default.
MaximumColumnSize	The maximum column size.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
Readonly	You can use this property to enforce read-only access to Jira Service Desk from the provider.
RequestLanguage	Use the requestLanguage to have column names translated in a specific language.
RTK	The runtime key used for licensing.
ServiceDeskID	Service Desk ID of the currently authenticated user.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through

	the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UpperCaseIdentifiers	This property reports all identifiers in uppercase. This is the default for Oracle databases and thus allows better integration with Oracle tools such as the Oracle Database Gateway.

For more details, see <https://cdn.cdata.com/help/GKG/odbc/>.

32. JSON {JSON}

32.1. Connection Options

32.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to remote services.
AccessKey	Your account access key. This value is accessible from your security credentials page.
SecretKey	Your account secret key. This value is accessible from your security credentials page.
ApiKey	The API Key used to identify the user to IBM Cloud.
User	The JSON user account used to authenticate.
Password	The password used to authenticate the user.
SharePointEdition	The edition of SharePoint being used. Set either SharePointOnline or SharePointOnPremise.

32.1.2. Connection

Property	Description
URI	The Uniform Resource Identifier (URI) for the JSON resource location.
JSONPath	The JSONPath of an array element that defines the separation of rows.
DataModel	Specifies the data model to use when parsing JSON documents and generating the database metadata.

JSONFormat	Specifies the format of the JSON document.
Region	The hosting region for your S3-like Web Services.
ProjectId	The id of the project where your Google Cloud Storage instance resides.
OracleNamespace	The Oracle Cloud Object Storage namespace to use.
StorageBaseURL	The URL of a cloud storage service provider.
SimpleUploadLimit	This setting specifies the threshold, in bytes, above which the provider will choose to perform a multipart upload rather than uploading everything in one request.
UseVirtualHosting	If true (default), buckets will be referenced in the request using the hosted-style request: <code>http://yourbucket.s3.amazonaws.com/yourobject</code> . If set to false, the bean will use the path-style request: <code>http://s3.amazonaws.com/yourbucket/yourobject</code> . Note that this property will be set to false, in case of an S3 based custom service when the CustomURL is specified.

32.1.3. AWS Authentication

Property	Description
AWSAccessKey	Your AWS account access key. This value is accessible from your AWS security credentials page.
AWSSecretKey	Your AWS account secret key. This value is accessible from your AWS security credentials page.
AWSRoleARN	The Amazon Resource Name of the role to use when authenticating.
AWSPrincipalArn	The ARN of the SAML Identity provider in your AWS account.

AWSRegion	The hosting region for your Amazon Web Services.
AWSSessionToken	Your AWS session token.
MFASerialNumber	The serial number of the MFA device if one is being used.
MFAToken	The temporary token available from your MFA device.

32.1.4. Azure Authentication

Property	Description
AzureStorageAccount	The name of your Azure storage account.
AzureAccessKey	The storage key associated with your JSON account.
AzureSharedAccessSignature	A shared access key signature that may be used for authentication.
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.
AzureEnvironment	The Azure Environment to use when establishing a connection.

32.1.5. SSO

Property	Description
SSOLoginURL	The identity provider's login URL.
SSOProperties	Additional properties required to connect to the identity provider in a semicolon-separated list.

32.1.6. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthVersion	The version of OAuth being used.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthAccessTokenSecret	The OAuth access token secret for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
OAuthGrantType	The grant type for the OAuth flow.
OAuthPasswordGrantMode	How to pass Client ID and Secret with OAuthGrantType is set to Password.
OAuthIncludeCallbackURL	Whether to include the callback URL in an access token request.

OAuthAuthorizationURL	The authorization URL for the OAuth service.
OAuthAccessTokenURL	The URL to retrieve the OAuth access token from.
OAuthRefreshTokenURL	The URL to refresh the OAuth token from.
OAuthRequestTokenURL	The URL the service provides to retrieve request tokens from. This is required in OAuth 1.0.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
AuthToken	The authentication token used to request and obtain the OAuth Access Token.
AuthKey	The authentication secret used to request and obtain the OAuth Access Token.
OAuthParams	A comma-separated list of other parameters to submit in the request for the OAuth access token in the format paramname=value.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

32.1.7. JWT OAuth

Property	Description
OAuthJWTCert	The JWT Certificate store.

OAuthJWTCertType	The type of key store containing the JWT Certificate.
OAuthJWTCertPassword	The password for the OAuth JWT certificate.
OAuthJWTCertSubject	The subject of the OAuth JWT certificate.
OAuthJWTIssuer	The issuer of the Java Web Token.
OAuthJWTSubject	The user subject for which the application is requesting delegated access.

32.1.8. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
KerberosRealm	The Kerberos Realm used to authenticate the user with.
KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosServiceRealm	The Kerberos realm of the service.
KerberosServiceKDC	The Kerberos KDC of the service.
KerberosTicketCache	The full file path to an MIT Kerberos credential cache file.

32.1.9. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLMode	The authentication mechanism to be used when connecting to the FTP or FTPS server.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

32.1.10. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertType	The type of SSHClientCert certificate.

32.1.11. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

32.1.12. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.

ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

32.1.13. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

32.1.14. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.

Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
FlattenArrays	By default, nested arrays are returned as strings of JSON. The FlattenArrays property can be used to flatten the elements of nested arrays into columns of their own. Set FlattenArrays to the number of elements you want to return from nested arrays.
FlattenObjects	Set FlattenObjects to true to flatten object properties into columns of their own. Otherwise, objects nested in arrays are returned as strings of JSON.
QualifyColumns	Controls whether the provider will use relative column names.

32.1.15. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.

CacheMetadata

This property determines whether or not to cache the table metadata to a file store.

32.1.16. Miscellaneous

Property	Description
BackwardsCompatibilityMode	Set BackwardsCompatibilityMode to true to use the JSON functionality and features available in the 2017 version.
BatchSize	The maximum size of each batch operation to submit.
Charset	Specifies the session character set for encoding and decoding character data transferred to and from the JSON file. The default value is UTF-8.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the JSON when the connection is opened.
Culture	This setting can be used to specify culture settings that determine how the provider interprets certain data types that are passed into the provider. For example, setting Culture='de-DE' will output German formats even on an American machine.
CustomHeaders	Other headers as determined by the user (optional).
CustomUrlParams	The custom query string to be included in the request.

ExcludeFiles	Comma-separated list of file extensions to exclude from the set of the files modeled as tables.
FlattenRowLimit	The maximum number of rows that can result from a single flattened element.
GenerateSchemaFiles	Indicates the user preference as to when schemas should be generated and saved.
IncludeDropboxTeamResources	Indicates if you want to include Dropbox team files and folders.
IncludeFiles	Comma-separated list of file extensions to include into the set of the files modeled as tables.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
MetadataDiscoveryURI	Used when aggregating multiple files into one table, this property specifies a specific file to read to determined the aggregated table schema.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from JSON.
PathSeparator	Determines the character which will be used to replace the file separator.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.

PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
Readonly	You can use this property to enforce read-only access to JSON from the provider.
RowScanDepth	The number of rows to scan when dynamically determining columns for the table.
RTK	The runtime key used for licensing.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
TypeDetectionScheme	Determines how to determine the data types of columns.
URISeparator	A delimiter used to separate different values in the URI property.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/DJG/jdbc/>.

33. KAFKA

33.1. Connection Options

33.1.1. Authentication

Property	Description
AuthScheme	The scheme used for authentication. Accepted entries are Auto,Plain,Scram,Kerberos.
User	The user who is authenticating to Apache Kafka.
Password	The password used to authenticate to Apache Kafka.
BootstrapServers	The address of the Apache Kafka BootstrapServers to which you are connecting to.
Topic	The name of the topic to produce into.
UseSSL	This field sets whether SSL is enabled.

33.1.2. Connection

Property	Description
ConsumerGroupld	Specifies which group the consumers created by the driver should belong to.
AutoCommit	Specifies if the Apache Kafka consumer should autocommit after each poll.

33.1.3. Kerberos

Property	Description
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosServiceName	The Kerberos service name you will authenticate.
UseKerberosTicketCache	In case that you don't want to use a keytab file but instead a ticket cache with the logged in user.

33.1.4. SSL

Property	Description
SSLCert	Path to the Keystore location or to the client's certificate location.
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLCertPassword	The password of the Keystore or of the client's private key.
SSLClientCertPassword	The password for the TLS/SSL client certificate.

SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.
SSLIdentificationAlgorithm	The endpoint identification algorithm used by the Apache Kafka data provider client app to validate server host name.
SSLKey	Set either the password to decrypt the client private key or the path of the client's private key depending on the data provider edition.
TrustStorePassword	The password of the Truststore, 'ssl.truststore.password'.
TrustStorePath	Path to the Truststore or Certification Authorities (CA) location.

33.1.5. SchemaRegistry

Property	Description
RegistryVersion	Version of the schema read from RegistryUrl for the specified topic.
RegistryUrl	The server for the schema registry. When this property is specified, the driver will read Apache Avro schema from the server.
RegistryUser	Username to authorize with the server specified in RegistryUrl .
RegistryPassword	Password to authorize with the server specified in RegistryUrl .

RegistryType	Type of the schema specified for the a specific topic.
RegistryService	The Schema Registry service used for working with topic schemas.

33.1.6. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

33.1.7. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.

ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

33.1.8. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

33.1.9. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

33.1.10. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.

CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

33.1.11. Miscellaneous

Property	Description
AggregateMessages	Specifies whether or not to return the message as a whole string.
BatchSize	The maximum size of each batch operation to submit.
CompressionType	Data compression type. Batches of data will be compressed together.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Apache Kafka when the connection is opened.
EnableIdempotence	If set to true, the Apache Kafka will ensure messages are delivered in the correct, and without duplicates.
FlattenArrays	By default, nested arrays won't show up if TypeDetectionScheme is set to SchemaRegistry. The FlattenArrays property can be used to flatten the elements of nested arrays into columns of their own. Set

	FlattenArrays to the number of elements you want to return from nested arrays.
GenerateSchemaFiles	Indicates the user preference as to when schemas should be generated and saved.
MaximumBatchSize	Specifies maximum batch size to gather before sending a request.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
MessageKeyColumn	If specified, the message key sent to Apache Kafka will be read from this column.
MessageKeyType	If MessageKeyColumn is specified, this property must be set to the expected type for the pertinent column.
OffsetResetStrategy	Specifies an offset for the consumer group.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.

ProduceMeta	Specifies whether or not to send a meta message while producing the outgoing message.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
ReadDuration	The duration which additional messages are allowed.
Readonly	You can use this property to enforce read-only access to Apache Kafka from the provider.
RowScanDepth	The maximum number of messages to scan for the columns available in the topic.
RTK	The runtime key used for licensing.
SerializationFormat	Specifies how to serialize/deserialize the incoming or outgoing message.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
TypeDetectionScheme	Comma-separated list of options specifying how the provider will scan the data to determine the fields and datatypes for the bucket.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/SKG/jdbc/>.

34. MARIADB

34.1. MariaDB Version Support

The driver enables connectivity to MariaDB Server through 5.0 to 10.5+.

34.2. Connection Options

34.2.1. Authentication

Property	Description
Server	The host name or IP address of the server.
Port	The port of the MariaDB server.
User	The MariaDB user account used to authenticate.
Password	The password used to authenticate the user.
Database	The name of the MariaDB database.
UseSSL	This field sets whether SSL is enabled.

34.2.2. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.

SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

34.2.3. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertSubject	The subject of the SSH client certificate.
SSHClientCertType	The type of SSHClientCert certificate.
SSHServer	The SSH server.
SSHPort	The SSH port.
SSHUser	The SSH user.
SSHPassword	The SSH password.
SSHServerFingerprint	The SSH server fingerprint.
UseSSH	Whether to tunnel the MariaDB connection over SSH. Use SSH.

34.2.4. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

34.2.5. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

34.2.6. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

34.2.7. Miscellaneous

Property	Description
AllowUserVariables	When set to True, user variables (prefixed by an @) can be used in SQL queries.
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the MariaDB when the connection is opened.

MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
QueryPassthrough	This option passes the query to the MariaDB server as is.
Readonly	You can use this property to enforce read-only access to MariaDB from the provider.
RTK	The runtime key used for licensing.
Timeout	The value in seconds until the connection timeout error is thrown.
UseConnectionPooling	This property enables connection pooling.
ZeroDatesToNull	Whether or not to return Date and DateTime values consisting of all zeros as NULL.

For more details, see <https://cdn.cdata.com/help/SRG/jdbc/>.

35. MONGODB

35.1. MongoDB Version Support

The driver models MongoDB instances as relational databases and supports MongoDB versions 2.6 through 5.0.

35.2. Connection Options

35.2.1. Authentication

Property	Description
AuthScheme	The authentication mechanism that MongoDB will use to authenticate the connection.
Server	The host name or IP address of the server hosting the MongoDB database.
Port	The port for the MongoDB database.
User	The MongoDB user account used to authenticate.
Password	The password used to authenticate the user.
Database	The name of the MongoDB database.
UseSSL	This field sets whether SSL is enabled.
AuthDatabase	The name of the MongoDB database for authentication.
ReplicaSet	This property allows you to specify multiple servers in addition to the one configured in Server and Port . Specify both a server name and port; separate servers with a comma.
DNSServer	Specify the DNS server when resolving MongoDB seed list.

35.2.2. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
KerberosRealm	The Kerberos Realm used to authenticate the user with.
KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosServiceRealm	The Kerberos realm of the service.
KerberosServiceKDC	The Kerberos KDC of the service.
KerberosTicketCache	The full file path to an MIT Kerberos credential cache file.

35.2.3. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

35.2.4. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertSubject	The subject of the SSH client certificate.
SSHClientCertType	The type of SSHClientCert certificate.
SSHServer	The SSH server.
SSHPort	The SSH port.
SSHUser	The SSH user.
SSHPassword	The SSH password.
SSHServerFingerprint	The SSH server fingerprint.
UseSSH	Whether to tunnel the MongoDB connection over SSH. Use SSH.

35.2.5. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.

FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

35.2.6. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

35.2.7. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

35.2.8. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

35.2.9. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the MongoDB when the connection is opened.

DataModel	By default, the provider will not automatically discover the metadata for a child table as its own distinct table. To enable this functionality, set DataModel to Relational .
FlattenArrays	By default, nested arrays are returned as strings of JSON. The FlattenArrays property can be used to flatten the elements of nested arrays into columns of their own. Set FlattenArrays to the number of elements you want to return from nested arrays.
FlattenObjects	Set FlattenObjects to true to flatten object properties into columns of their own. Otherwise, objects nested in arrays are returned as strings of JSON.
GenerateSchemaFiles	Indicates the user preference as to when schemas should be generated and saved.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
NoCursorTimeout	The server normally times out idle cursors after an inactivity period (10 minutes) to prevent excess memory use. Set this option to prevent that.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from MongoDB.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
QueryPassthrough	This option passes the query to MongoDB as-is.
Readonly	You can use this property to enforce read-only access to MongoDB from the provider.

ReadPreference	Set this to a strategy for reading from a replica set. Accepted values are primary, primaryPreferred, secondary, secondaryPreferred, and nearest.
RowScanDepth	The maximum number of rows to scan to look for the columns available in a table.
RTK	The runtime key used for licensing.
SlaveOK	This property sets whether the provider is allowed to read from secondary (slave) servers.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
TypeDetectionScheme	Comma-separated options for how the provider will scan the data to determine the fields and datatypes in each document collection.
UseConnectionPooling	This property enables connection pooling.
UseFindAPI	Execute MongoDB queries using <code>db.collection.find()</code> .
WriteConcern	Requests acknowledgment that the write operation has propagated to the specified number of mongod instances.
WriteConcernJournaled	It requires acknowledgment that the mongod instances, as specified in the WriteConcern , have written to the on-disk journal.
WriteConcernTimeout	This option specifies a time limit, in milliseconds, for the write concern.
WriteScheme	Sets whether the object type for inserted or updated objects is determined from the existing column metadata or the input value type.

For more details, see <https://cdn.cdata.com/help/DGG/jdbc/>.

36. **MYSQL**

36.1. MySQL Version Support

The driver enables connectivity to MySQL Server through 5.0 to 8.0.

36.2. Connection Options

36.2.1. Authentication

Property	Description
AuthScheme	The scheme used for authentication. Accepted entries are Password, AzureAD, AzurePassword, AzureMSI.
Server	The host name or IP address of the server. Supports cluster servers, for example '192.168.0.1,192.168.0.2'.
Port	The port of the MySQL server. Supports cluster servers, for example: '3306, 3307', the number of the port should match with Servers.
User	The MySQL user account used to authenticate.
Password	The password used to authenticate the user.
Database	The name of the MySQL database.
Domain	The name of the domain for a Windows (NTLM) security login.
NTLMVersion	The NTLM version.
IntegratedUser	The user that is authenticating to the Windows.
UseSSL	This field sets whether SSL is enabled.

36.2.2. Azure Authentication

Property	Description
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.

36.2.3. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

36.2.4. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

36.2.5. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertSubject	The subject of the SSH client certificate.
SSHClientCertType	The type of SSHClientCert certificate.
SSHServer	The SSH server.
SSHPort	The SSH port.
SSHUser	The SSH user.
SSHPassword	The SSH password.

SSHServerFingerprint	The SSH server fingerprint.
UseSSH	Whether to tunnel the MySQL connection over SSH. Use SSH.

36.2.6. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

36.2.7. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.

ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

36.2.8. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

36.2.9. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

36.2.10. Miscellaneous

Property	Description
AllowUserVariables	When set to True, user variables (prefixed by an @) can be used in SQL queries.
BatchSize	The maximum size of each batch operation to submit.
Characterset	The default client character set used by the provider. For example, 'utf8'.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the MySQL when the connection is opened.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
QueryPassthrough	This option passes the query to the MySQL server as is.
Readonly	You can use this property to enforce read-only access to MySQL from the provider.
RTK	The runtime key used for licensing.
ServerTimeZone	Specify a specific server time zone id of current platform(.Net or Java) by user.

Timeout	The value in seconds until the connection timeout error is thrown.
UseConnectionPooling	This property enables connection pooling.
ZeroDatesToNull	Whether or not to return Date and DateTime values consisting of all zeros as NULL.

For more details, see <https://cdn.cdata.com/help/DMG/jdbc/>.

37. SAS DATA SETS

37.1. Connection Options

37.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to remote services.
AccessKey	Your account access key. This value is accessible from your security credentials page.
SecretKey	Your account secret key. This value is accessible from your security credentials page.
ApiKey	The API Key used to identify the user to IBM Cloud.
User	The user account used to authenticate.
Password	The password used to authenticate the user.
SharePointEdition	The edition of SharePoint being used. Set either SharePointOnline or SharePointOnPremise.

37.1.2. Connection

Property	Description
ConnectionType	The type of connection to use.
URI	The Uniform Resource Identifier (URI) for the SAS resource location.

Region	The hosting region for your S3-like Web Services.
ProjectId	The Id of the project where your Google Cloud Storage instance resides.
OracleNamespace	The Oracle Cloud Object Storage namespace to use.
StorageBaseURL	The URL of a cloud storage service provider.
UseVirtualHosting	If true (default), buckets will be referenced in the request using the hosted-style request: <code>http://yourbucket.s3.amazonaws.com/yourobject</code> . If set to false, the bean will use the path-style request: <code>http://s3.amazonaws.com/yourbucket/yourobject</code> . Note that this property will be set to false, in case of an S3 based custom service when the CustomURL is specified.

37.1.3. AWS Authentication

Property	Description
AWSAccessKey	Your AWS account access key. This value is accessible from your AWS security credentials page.
AWSSecretKey	Your AWS account secret key. This value is accessible from your AWS security credentials page.
AWSRoleARN	The Amazon Resource Name of the role to use when authenticating.
AWSPrincipalARN	The ARN of the SAML Identity provider in your AWS account.
AWSRegion	The hosting region for your Amazon Web Services.
AWSCredentialsFile	The path to the AWS Credentials File to be used for authentication.

AWSCredentialsFileProfile	The name of the profile to be used from the supplied AWSCredentialsFile.
AWSSessionToken	Your AWS session token.
MFASerialNumber	The serial number of the MFA device if one is being used.
MFAToken	The temporary token available from your MFA device.
ServerSideEncryption	When activated, file uploads into Amazon S3 buckets will be server-side encrypted.

37.1.4. Azure Authentication

Property	Description
AzureStorageAccount	The name of your Azure storage account.
AzureAccessKey	The storage key associated with your SAS Data Sets account.
AzureSharedAccessSignature	A shared access key signature that may be used for authentication.
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.
AzureEnvironment	The Azure Environment to use when establishing a connection.

37.1.5. SSO

Property	Description
SSOLoginURL	The identity provider's login URL.
SSOProperties	Additional properties required to connect to the identity provider in a semicolon-separated list.

37.1.6. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client Id assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthAccessTokenSecret	The OAuth access token secret for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
Scope	Specify scope to obtain the initial access and refresh token.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

37.1.7. JWT OAuth

Property	Description
OAuthJWTCert	The JWT Certificate store.
OAuthJWTCertType	The type of key store containing the JWT Certificate.
OAuthJWTCertPassword	The password for the OAuth JWT certificate.
OAuthJWTCertSubject	The subject of the OAuth JWT certificate.
OAuthJWTIssuer	The issuer of the Java Web Token.
OAuthJWTSubject	The user subject for which the application is requesting delegated access.

37.1.8. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
KerberosRealm	The Kerberos Realm used to authenticate the user.
KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosServiceRealm	The Kerberos realm of the service.
KerberosServiceKDC	The Kerberos KDC of the service.
KerberosTicketCache	The full file path to an MIT Kerberos credential cache file.

37.1.9. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLMode	The authentication mechanism to be used when connecting to the FTP or FTPS server.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

37.1.10. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A private key to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert key if it has one.
SSHClientCertSubject	The subject of the SSH client certificate.
SSHClientCertType	The type of SSHClientCert private key.
SSHUser	The SSH user.
SSHPassword	The SSH password.

37.1.11. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

37.1.12. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

37.1.13. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

37.1.14. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

37.1.15. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache.
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

37.1.16. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
Charset	Specifies the session character set for encoding and decoding character data transferred to and from the SAS Data Sets file. The default value is UTF-8.

<u>ClientCulture</u>	This property can be used to specify the format of data (e.g., currency values) that is accepted by the client application. This property can be used when the client application does not support the machine's culture settings. For example, Microsoft Access requires 'en-US'.
<u>ConnectionLifeTime</u>	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
<u>ConnectOnOpen</u>	This property specifies whether to connect to the SAS Data Sets when the connection is opened.
<u>Culture</u>	This setting can be used to specify culture settings that determine how the provider interprets certain data types that are passed into the provider. For example, setting Culture='de-DE' will output German formats even on an American machine.
<u>DeleteDownloadedFiles</u>	When set to true, the provider will delete parsed .sas7bdat files downloaded from cloud sources.
<u>DirectoryRetrievalDepth</u>	Limit the subfolders recursively scanned when IncludeSubdirectories is enabled.
<u>ExcludeFiles</u>	Comma-separated list of file extensions to exclude from the set of the files modeled as tables.
<u>IncludeDropboxTeamResources</u>	Indicates if you want to include Dropbox team files and folders.
<u>IncludeFiles</u>	Comma-separated list of file extensions to include into the set of the files modeled as tables.
<u>IncludeSubdirectories</u>	Whether to read files from nested folders. In the case of a name collision, table names are prefixed by the underscore-separated folder names.
<u>MaxRows</u>	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
<u>Other</u>	These hidden properties are used only in specific use cases.
<u>PoolIdleTimeout</u>	The allowed idle time for a connection before it is closed.
<u>PoolMaxSize</u>	The maximum connections in the pool.

[PoolMinSize](#)

The minimum number of connections in the pool.

[PoolWaitTime](#)

The max seconds to wait for an available connection.

[PseudoColumns](#)

This property indicates whether or not to include pseudo columns as columns to the table.

[Readonly](#)

You can use this property to enforce read-only access to SAS Data Sets from the provider.

[RTK](#)

The runtime key used for licensing.

[TemporaryLocalFolder](#)

The path, or URI, to the folder that is used to temporarily download SAS file(s).

[Timeout](#)

The value in seconds until the timeout error is thrown, canceling the operation.

[UseConnectionPooling](#)

This property enables connection pooling.

[UserDefinedViews](#)

A filepath pointing to the JSON configuration file containing your custom views.

38. NEO4J

38.1. Neo4j Version Support

The driver leverages the Neo4j API. The driver supports Neo4j versions 4.2 and above.

38.2. Connection Options

38.2.1. Authentication

Property	Description
User	The Neo4j user account used to authenticate.
Password	The password used to authenticate the user.

38.2.2. Connection

Property	Description
Server	The server hosting the Neo4j instance.
Port	The port on which the Neo4j service is running.

38.2.3. SSL

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

38.2.4. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

38.2.5. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

38.2.6. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

38.2.7. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

38.2.8. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

38.2.9. Miscellaneous

Property	Description
Database	The default database queried for your Neo4j instance.
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property specifies whether to connect to the Neo4j when the connection is opened.

<u>MaxRows</u>	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
<u>Other</u>	These hidden properties are used only in specific use cases.
<u>Pagesize</u>	The maximum number of results to return per page from Neo4j.
<u>PoolIdleTimeout</u>	The allowed idle time for a connection before it is closed.
<u>PoolMaxSize</u>	The maximum connections in the pool.
<u>PoolMinSize</u>	The minimum number of connections in the pool.
<u>PoolWaitTime</u>	The max seconds to wait for an available connection.
<u>PseudoColumns</u>	This property indicates whether or not to include pseudo columns as columns to the table.
<u>QueryPassthrough</u>	This option passes the query to the Neo4j server as is.
<u>RowScanDepth</u>	The maximum number of rows to scan to look for the columns available in a table.
<u>RTK</u>	The runtime key used for licensing.
<u>Timeout</u>	The value in seconds until the timeout error is thrown, canceling the operation.
<u>UseConnectionPooling</u>	This property enables connection pooling.
<u>UserDefinedViews</u>	A filepath pointing to the JSON configuration file containing your custom views.

For more details, see [CData JDBC Driver for Neo4j - CData JDBC Driver for Neo4j](#).

39. ORACLE

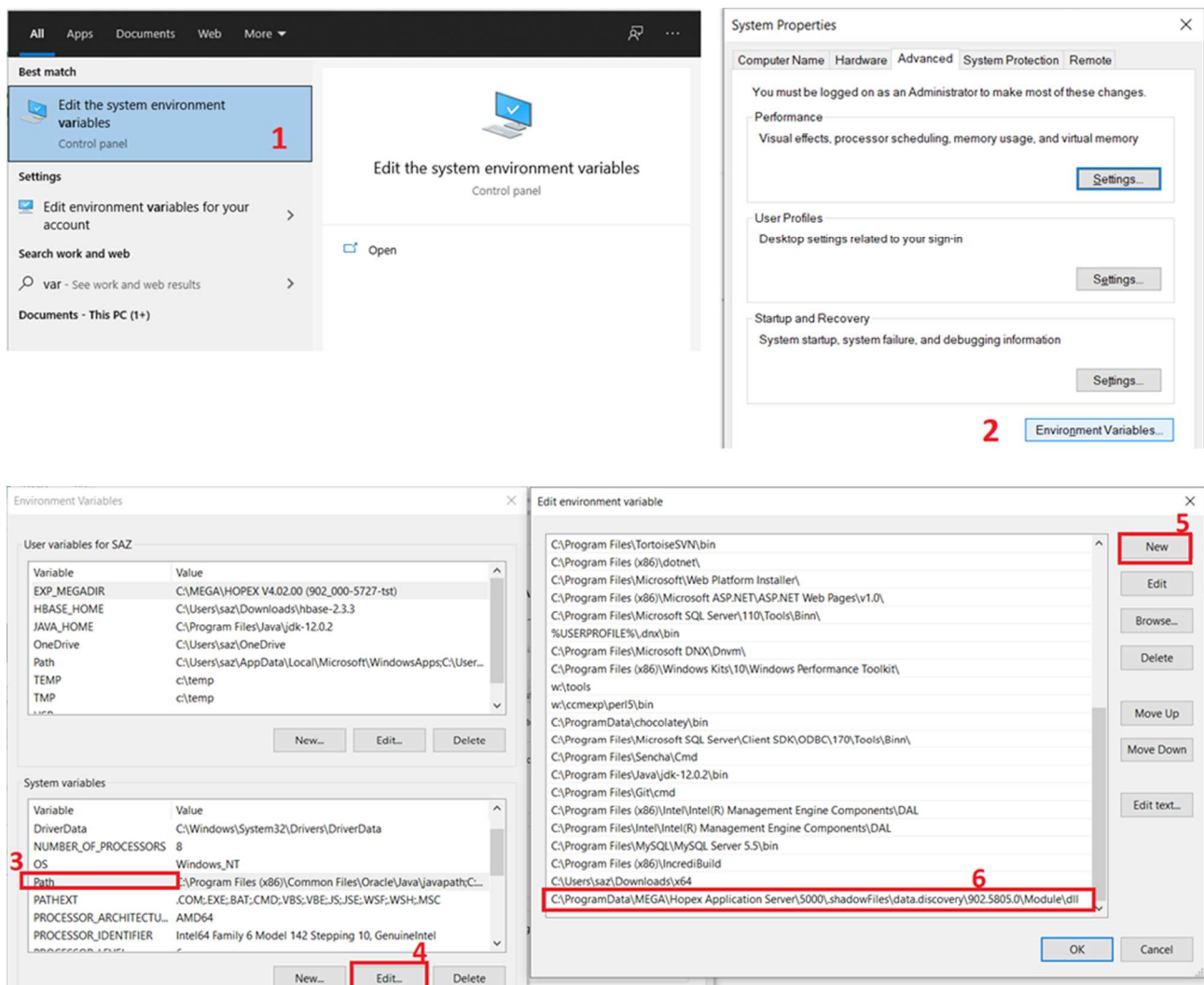
39.1. Prerequisite

After installing the HOPEX Discovery module, you must indicate the system where the supplied .dll files that allow connection to the Oracle server are located.

The path to these files must be added to the "PATH" variable in system variables of Windows. The path depends on system version:

32 bit: C:\ProgramData\MEGA\Hopex Application
Server\<ClusterName>\.shadowFiles\data.discovery\<Version Number>
\Module\Connectors\oracle\32

64 bit: C:\ProgramData\MEGA\Hopex Application
Server\<ClusterName>\.shadowFiles\data.discovery\<Version
Number>\Module\Connectors\oracle\64



You must restart your computer so that the addition of the .dll directory is taken into account by Windows.

39.2. Oracle OCI Version Support

The driver can connect Oracle Database 11.2 or later.

39.3. Connection Options

39.3.1. Authentication

Property	Description
Server	The host name or IP of the server hosting the Oracle database.
Port	The port used to connect to the server hosting the Oracle database.
ServiceName	The service name of the Oracle database.
User	The Oracle OCI user account used to authenticate.
Password	The password used to authenticate the user.
DataSource	Oracle Net Services Name, Connect Descriptor (known also as TNS Connect String), or an easy connect naming that identifies the database to which to connect.

39.3.2. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).

MaxLogFileCount	A string specifying the maximum file count of log files.

39.3.3. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
IncludeSynonyms	Query meta data for synonyms as they are being the regular tables.
ShowMetadataDescriptions	Controls whether table and column descriptions are returned via the platform metadata APIs and sys_tables / sys_views / sys_tablecolumns.
UseDBAMetadataViews	Query meta data from DBA_.. system views instead of ALL_.. system views.

39.3.4. Miscellaneous

Property	Description
AllowPreparedStatement	Prepare a query statement before its execution.
BatchSize	The maximum size of each batch operation to submit.

ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Oracle OCI when the connection is opened.
IncludeSynonyms	Query meta data for synonyms as they are being the regular tables.
MaxLobSize	The volume in numbers of bytes or UTF-8 chars which is allowed to query by non-parameterized SELECT query.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
QueryPassthrough	This option passes the query to the Oracle OCI server as is.
Readonly	You can use this property to enforce read-only access to Oracle OCI from the provider.
ReconnectTimeout	The sleep time, in seconds, before retrying to reconnect to the server on a maximum idle time exceeded error.
ReconnectTries	The number of retry to connect server when a maximum idle time exceeded error is reported by server.
RTK	The runtime key used for licensing.
UseConnectionPooling	This property enables connection pooling.
UseDBAMetadataViews	Query meta data from DBA_.. system views instead of ALL_.. system views.

For more details, see <https://cdn.cdata.com/help/SOG/jdbc/>.

40. PARQUET

40.1. Connection Options

40.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to remote services.
AccessKey	Your account access key. This value is accessible from your security credentials page.
SecretKey	Your account secret key. This value is accessible from your security credentials page.
ApiKey	The API Key used to identify the user to IBM Cloud.
SharePointEdition	The edition of SharePoint being used. Set either SharePointOnline or SharePointOnPremise.

40.1.2. Connection

Property	Description
URI	The Uniform Resource Identifier (URI) for the Parquet resource location.
DataModel	Specifies the data model to use when parsing Parquet documents and generating the database metadata.
Region	The hosting region for your S3-like Web Services.
ProjectId	The id of the project where your Google Cloud Storage instance resides.
OracleNamespace	The Oracle Cloud Object Storage namespace to use.

StorageBaseURL	The URL of a cloud storage service provider.
UseVirtualHosting	If true (default), buckets will be referenced in the request using the <code>hosted-style</code> request: <code>http://yourbucket.s3.amazonaws.com/yourobject</code> . If set to false, the bean will use the <code>path-style</code> request: <code>http://s3.amazonaws.com/yourbucket/yourobject</code> . Note that this property will be set to false, in case of an S3 based custom service when the CustomURL is specified.

40.1.3. AWS Authentication

Property	Description
AWSAccessKey	Your AWS account access key. This value is accessible from your AWS security credentials page.
AWSSecretKey	Your AWS account secret key. This value is accessible from your AWS security credentials page.
AWSRoleARN	The Amazon Resource Name of the role to use when authenticating.
AWSPrincipalArn	The ARN of the SAML Identity provider in your AWS account.
AWSRegion	The hosting region for your Amazon Web Services.
AWSSessionToken	Your AWS session token.
MFASerialNumber	The serial number of the MFA device if one is being used.
MFAToken	The temporary token available from your MFA device.

40.1.4. Azure Authentication

Property	Description
AzureStorageAccount	The name of your Azure storage account.

AzureAccessKey	The storage key associated with your Parquet account.
AzureSharedAccessSignature	A shared access key signature that may be used for authentication.
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.
AzureEnvironment	The Azure Environment to use when establishing a connection.

40.1.5. SSO

Property	Description
SSOLoginURL	The identity provider's login URL.
SSOProperties	Additional properties required to connect to the identity provider in a semicolon-separated list.

40.1.6. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthVersion	The version of OAuth being used.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.

OAuthAccessTokenSecret	The OAuth access token secret for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
OAuthGrantType	The grant type for the OAuth flow.
OAuthPasswordGrantMode	How to pass Client ID and Secret with OAuthGrantType is set to Password.
OAuthIncludeCallbackURL	Whether to include the callback URL in an access token request.
OAuthAuthorizationURL	The authorization URL for the OAuth service.
OAuthAccessTokenURL	The URL to retrieve the OAuth access token from.
OAuthRefreshTokenURL	The URL to refresh the OAuth token from.
OAuthRequestTokenURL	The URL the service provides to retrieve request tokens from. This is required in OAuth 1.0.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
AuthToken	The authentication token used to request and obtain the OAuth Access Token.
AuthKey	The authentication secret used to request and obtain the OAuth Access Token.
OAuthParams	A comma-separated list of other parameters to submit in the request for the OAuth access token in the format paramname=value.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.

OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

40.1.7. JWT OAuth

Property	Description
OAuthJWTCert	The JWT Certificate store.
OAuthJWTCertType	The type of key store containing the JWT Certificate.
OAuthJWTCertPassword	The password for the OAuth JWT certificate.
OAuthJWTCertSubject	The subject of the OAuth JWT certificate.
OAuthJWTIssuer	The issuer of the Java Web Token.
OAuthJWTSubject	The user subject for which the application is requesting delegated access.

40.1.8. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLMode	The authentication mechanism to be used when connecting to the FTP or FTPS server.

SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

40.1.9. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertType	The type of SSHClientCert certificate.

40.1.10. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

40.1.11. Proxy

Property	Description

ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

40.1.12. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

40.1.13. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
FlattenObjects	Set FlattenObjects to true to flatten object properties into columns of their own. Otherwise, objects nested in arrays are returned as strings of JSON.
FlattenArrays	By default, nested arrays are returned as strings. The FlattenArrays property can be used to flatten the elements of nested arrays into columns of their own. Set FlattenArrays to the number of elements you want to return from nested arrays.

40.1.14. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both

	properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

40.1.15. Miscellaneous

Property	Description
AggregateFiles	When set to true, the provider will aggregate all the files in URI directory into a single result. With this option enabled, the AggregatedFiles will be exposed which can be used to query the dataset.
BatchSize	The maximum size of each batch operation to submit.
Charset	Specifies the session character set for encoding and decoding character data transferred to and from the Parquet file. The default value is UTF-8.
ClientCulture	This property can be used to specify the format of data (e.g., currency values) that is accepted by the client application. This property can be used when the client application does not support the machine's culture settings. For example, Microsoft Access requires 'en-US'.
Compression	Specifies which compression encoding to be used when creating .parquet files using Create Table Statement and Bulk Inserts.

ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Parquet when the connection is opened.
Culture	This setting can be used to specify culture settings that determine how the provider interprets certain data types that are passed into the provider. For example, setting Culture='de-DE' will output German formats even on an American machine.
DeleteDownloadedFiles	When set to true, the provider will delete parsed .parquet files downloaded from cloud sources.
DirectoryRetrievalDepth	Limit the subfolders recursively scanned when IncludeSubdirectories is enabled.
EnableDictionary	When set to true, the provider will enable dictionary encoding when creating .parquet files using Create Table Statement and Bulk Inserts.
ExcludeFiles	Comma-separated list of file extensions to exclude from the set of the files modeled as tables.
IncludeDropboxTeamResources	Indicates if you want to include Dropbox team files and folders.
IncludeFiles	Comma-separated list of file extensions to include into the set of the files modeled as tables.
IncludeSubdirectories	Whether to read files from nested folders. In the case of a name collision, table names are prefixed by the underscore-separated folder names.
InsertMode	The behavior when using bulk inserts to create Parquet files.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.

MetadataDiscoveryURI	Used when aggregating multiple files into one table, this property specifies a specific file to read to determined the aggregated table schema.
Other	These hidden properties are used only in specific use cases.
PageSize	(Optional) PageSize value.
ParallelPagingSizeLimit	Parquet file size limit (MegaBytes) for which to use parallel paging.
PathSeparator	Determines the character which will be used to replace the file separator.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
RTK	The runtime key used for licensing.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
TemporaryLocalFolder	The path, or URI, to the folder that is used to temporarily download parquet file(s).
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/HIG/jdbc/>.

41. POSTGRESQL

41.1. PostgreSQL Version Support

The driver enables standards-based access to PostgreSQL databases version 7.4 and later.

41.2. Connection Options

41.2.1. Authentication

Property	Description
AuthScheme	The scheme used for authentication. Accepted entries are Password, AzureAD, AzurePassword, AzureMSI, AwsIAMRoles.
Server	The host name or IP address of the server.
Database	The name of the PostgreSQL database.
User	The PostgreSQL user account used to authenticate.
Password	The password used to authenticate the user.
Port	The port number of the PostgreSQL server.
UseSSL	This field sets whether SSL is enabled.
Visibility	Visibility restrictions used to filter exposed metadata for tables with privileges granted to them for current user. For example 'SELECT,INSERT' filter is restricting metatdata visibility only for those tables which may be accessed by current user for SELECT and INSERT operations. Supported privilege values are SELECT, INSERT, UPDATE, DELETE, REFERENCES.

41.2.2. AWS Authentication

Property	Description
AWSAccessKey	Your AWS account access key. This value is accessible from your AWS security credentials page.
AWSSecretKey	Your AWS account secret key. This value is accessible from your AWS security credentials page.
AWSRoleARN	The Amazon Resource Name of the role to use when authenticating.

41.2.3. Azure Authentication

Property	Description
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.

41.2.4. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.

OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
OAuthGrantType	The grant type for the OAuth flow.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

41.2.5. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
KerberosRealm	The Kerberos Realm used to authenticate the user with.
KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosServiceRealm	The Kerberos realm of the service.
KerberosServiceKDC	The Kerberos KDC of the service.

KerberosTicketCache

The full file path to an MIT Kerberos credential cache file.

41.2.6. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

41.2.7. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertSubject	The subject of the SSH client certificate.
SSHClientCertType	The type of SSHClientCert certificate.
SSHServer	The SSH server.
SSHPort	The SSH port.

SSHUser	The SSH user.
SSHPassword	The SSH password.
SSHServerFingerprint	The SSH server fingerprint.
UseSSH	Whether to tunnel the PostgreSQL connection over SSH. Use SSH.

41.2.8. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

41.2.9. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).

MaxLogFileCount	A string specifying the maximum file count of log files.

41.2.10. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
IgnoredSchemas	Visibility restriction filter which is used to hide the list of schemas by metadata quering. For example, 'information_schema, pg_catalog'. Schema names are case sensitive.

41.2.11. Miscellaneous

Property	Description
AllowPreparedStatement	Prepare a query statement before its execution.
BatchSize	The maximum size of each batch operation to submit.
BrowsePartitions	By default the provider exposes super table and its partitions by metadata. You may hide sub partition by setting this property into false.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.

ConnectOnOpen	This property specifies whether to connect to the PostgreSQL when the connection is opened.
FetchResultSetMetadata	This field sets whether the provider is getting detailed information about resultset columns from the server.
IncludeTableTypes	If set to true, the provider will query for the types of individual tables and views.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
QueryPassthrough	This option passes the query to the PostgreSQL server as is.
Readonly	You can use this property to enforce read-only access to PostgreSQL from the provider.
RTK	The runtime key used for licensing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
TimeZone	Notifies the server about the time zone on the client with standard SET TIMEZONE query when connection is being opened.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/FPG/jdbc/>.

42. REDIS

42.1. Redis Version Support

The driver models Redis instances as relational databases. The driver leverages the Redis commands to enable bidirectional access to Redis and Redis Enterprise data through SQL. Redis 2.8.0 or above is required.

42.2. Connection Options

42.2.1. Authentication

Property	Description
AuthScheme	The authentication mechanism that the provider will use to authenticate with Redis.
Server	The host name or IP address of the server hosting the Redis instance.
Port	The port for the Redis database.
LogicalDatabase	The index of the Redis Logical Database.
Password	The password used to authenticate with Redis.
EnableCluster	This field sets whether the Redis Cluster Mode is enabled.
UseSSL	This field sets whether SSL is enabled.

42.2.2. Connection

Property	Description
DefineTables	Define the tables exposed by the provider using table names and Redis key patterns.

PatternSeparator	Define the table pattern's separator.
TablePattern	Define the tables exposed by the provider using Redis key patterns.

42.2.3. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLStartMode	This property determines how the provider starts the SSL negotiation.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

42.2.4. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertSubject	The subject of the SSH client certificate.

SSHClientCertType	The type of SSHClientCert certificate.
SSHServer	The SSH server.
SSHPort	The SSH port.
SSHUser	The SSH user.
SSHPassword	The SSH password.
SSHServerFingerprint	The SSH server fingerprint.
UseSSH	Whether to tunnel the Redis connection over SSH. Use SSH.

42.2.5. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

42.2.6. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.

LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

42.2.7. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

42.2.8. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.

CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

42.2.9. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Redis when the connection is opened.
IgnoreTypeErrors	Removes support for the specified data types and ignores casting exceptions for those types.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
ParallelMode	This option sets whether the provider should use multiple connections when connecting to Redis.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.

PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
QueryTimeout	The timeout in seconds for which the provider will wait for the query response. The default value is -1, which indicates the provider should never time out.
Readonly	You can use this property to enforce read-only access to Redis from the provider.
RowScanDepth	The maximum number of rows to scan to look for the columns available in a table.
RTK	The runtime key used for licensing.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
TableScanDepth	The maximum number of keys to scan when looking for tables available in your Redis database.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/ElG/jdbc/>.

43. REDSHIFT

43.1. Amazon Redshift Version Support

The driver enables standards-based access to Amazon Redshift based on PostgreSQL 8.0.2 with Amazon Redshift SQL.

43.2. Connection Options

43.2.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to Amazon Redshift.
Server	The host name or IP address of the Amazon Redshift cluster.
Port	The port number of the Amazon Redshift server.
Database	The name of the Amazon Redshift database.
User	The Amazon Redshift user account used to authenticate.
Password	The password used to authenticate the user.
UseSSL	This field sets whether SSL is enabled.
Visibility	Visibility restrictions used to filter exposed metadata for tables with privileges granted to them for current user. For example 'SELECT,INSERT' filter is restricting metatdata visibility only for those tables which may be accessed by current user for SELECT and INSERT operations. Supported privilege values are SELECT, INSERT, UPDATE, DELETE, REFERENCES.

43.2.2. AWS Authentication

Property	Description
AwsAccessKey	Your AWS account access key or the access key for an authorized IAM user.
AwsSecretKey	Your AWS account secret key or the secret key for an authorized IAM user.
AutoCreate	Specify true to create a database user with the name specified for User if one does not exist while connecting with IAM credentials. See AuthScheme .
DbGroups	A comma-delimited list of the names of one or more existing database groups the database user joins for the current session when connecting with IAM credentials. See AuthScheme .
AWSRoleARN	The Amazon Resource Name of the role to use when authenticating.
AWSPrincipalArn	The ARN of the SAML Identity provider in your AWS account.
AWSRegion	The hosting region for your Amazon Web Services.
AWSSessionToken	Your AWS session token.

43.2.3. SSO

Property	Description
SSOLoginURL	The identity provider's login URL.
SSOProperties	Additional properties required to connect to the identity provider in a semicolon-separated list.
SSOExchangeUrl	The url used for consuming the SAML response and exchanging it with Amazon Redshift specific credentials.

43.2.4. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

43.2.5. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertSubject	The subject of the SSH client certificate.
SSHClientCertType	The type of SSHClientCert certificate.
SSHServer	The SSH server.
SSHPort	The SSH port.
SSHUser	The SSH user.

SSHPassword	The SSH password.
SSHServerFingerprint	The SSH server fingerprint.
UseSSH	Whether to tunnel the Amazon Redshift connection over SSH. Use SSH.

43.2.6. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

43.2.7. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.

ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

43.2.8. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

43.2.9. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.

Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
IgnoredSchemas	Visibility restriction filter which is used to hide the list of schemas by metadata quering. For example, 'information_schema, pg_catalog'. Schema names are case sensitive.

43.2.10. Miscellaneous

Property	Description
AllowPreparedStatement	Prepare a query statement before its execution.
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Amazon Redshift when the connection is opened.
FetchResultSetMetadata	This field sets whether the provider is getting detailed information about resultset columns from the server.
IncludeTableTypes	If set to true, the provider will query for the types of individual tables and views.
InsertMode	Specifies what method to use when inserting bulk data. By default DML mode is used.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.

PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
QueryPassthrough	This option passes the query to the Amazon Redshift server as is.
Readonly	You can use this property to enforce read-only access to Amazon Redshift from the provider.
RTK	The runtime key used for licensing.
S3Bucket	Specifies the name of AWS S3 bucket to upload bulk data for staging.
S3BucketFolder	Specifies the name of the folder in AWS S3 bucket to upload bulk data for staging. By default bulk data are staged in the root folder.
StripOutNulls	When set the null characters are stripped out from character values in bulk operations.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
TimeZone	Set time zone for the current session.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/FRG/jdbc/>.

44. SALESFORCE

44.1. Connection Options

44.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to Salesforce.
User	The Salesforce user account used to authenticate.
Password	The password used to authenticate the user.
SecurityToken	The security token used to authenticate access to the Salesforce account.
UseSandbox	A boolean determining if the connection should be made to a Salesforce sandbox account.
CredentialsLocation	The location of the settings file where token retrieved with OKTA MFA is saved.

44.1.2. Connection

Property	Description
APIVersion	The version of the Salesforce API used.
LoginURL	URL to the Salesforce server used for logging in.

44.1.3. SSO

Property	Description
SSOLoginURL	The identity provider's login URL.
SSOProperties	Additional properties required to connect to the identity provider in a semicolon-separated list.
SSOExchangeUrl	The url used for consuming the SAML response and exchanging it with Salesforce specific credentials.

44.1.4. BulkAPI

Property	Description
UseBulkAPI	Whether to use the synchronous SOAP API or the asynchronous Bulk API.
BulkAPIConcurrencyMode	The concurrency mode for processing bulk rows with BULK API v1.
BulkPageSize	The number of records to retrieve before returning results to the user when UseBulkAPI=true.
BulkPollingInterval	The time interval in milliseconds between requests that check the availability of the bulk query response. The default value is 500 ms.
BulkQueryTimeout	The timeout in minutes for which the provider will wait for a bulk query response. The default value is 25 minutes.
WaitForBulkResults	Whether to wait for bulk results when using the asynchronous API. Only active when UseBulkAPI is true.

44.1.5. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
OAuthServerURL	The server URL to use if authenticating with OAuth.
OAuthGrantType	The grant type for the OAuth flow.
OAuthAuthorizationURL	The authorization URL for the OAuth service.
OAuthAccessTokenURL	The URL to retrieve the OAuth access token from.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.

OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

44.1.6. JWT OAuth

Property	Description
OAuthJWTCert	The JWT Certificate store.
OAuthJWTCertType	The type of key store containing the JWT Certificate.
OAuthJWTCertPassword	The password for the OAuth JWT certificate.
OAuthJWTCertSubject	The subject of the OAuth JWT certificate.
OAuthJWTIssuer	The issuer of the Java Web Token.
OAuthJWTSubject	The user subject for which the application is requesting delegated access.

44.1.7. SSL

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

44.1.8. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.

FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

44.1.9. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

44.1.10. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.

Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

44.1.11. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

44.1.12. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.

CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

44.1.13. Miscellaneous

Property	Description
AllOrNone	A boolean indicating if you would like all inserts, updates, or deletes to fail in a request if even a single record fails.
ArchiveMode	Boolean indicating whether to include deleted and archived records with a standard SELECT query.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Salesforce when the connection is opened.
ContinueOnAlterException	Whether you want to continue after a ALTER statement has failed.
FilterScope	Optional scope to limit the records returned from queries.

IncludeMetadataDescription	Set this property to a value other than NONE if you want to retrieve the descriptions for columns, tables or both of them from the Metadata API.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
Readonly	You can use this property to enforce read-only access to Salesforce from the provider.
RTK	The runtime key used for licensing.
ServerSideAggregation	A boolean determining if server side aggregation should be used.
SessionTimeout	The time in minutes for which a Salesforce login session is reused.
SkipFormulaFields	Set to true if formula fields should be skipped when listing columns.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.

UseConnectionPooling	This property enables connection pooling.
UseDisplayNames	Boolean determining if the display names for the columns should be used instead of the API names.

For more details, see <https://cdn.cdata.com/help/RFG/jdbc/>.

45. SAP

45.1. SAP ERP Version Support

The driver enables relational access to tables, queries, and function modules in your SAP R/3, NetWeaver, or ERP/ECC 6.0+ system.

45.2. Connection Options

45.2.1. Authentication

Property	Description
ConnectionType	The type of connection you are making.
Host	Host name of the target system.
SystemNumber	The number by which the target system is defined. Used when setting the Host connection property.
User	The user that is authenticating to the SAP system.
Password	The password used to authenticate to the SAP system.
Client	The client authenticating to the SAP system.
X509Certificate	The X509 certificate used for login as an alternative to User , and Password .
MessageServer	The message server must be specified when connecting to an SAP system that uses load balancing.
Group	The Logon Group being used. This typically only needs to be specified when connecting to an SAP system that uses load balancing.
SystemId	The System Id or R3Name of the SAP System is a string with a maximum of three characters. It is often used in load balancing connections.

RFCURL	The URL of the SOAP interface to connect with SAP.
MessageServerService	The message server service you wish to connect to.

45.2.2. Security

Property	Description
SNCMode	A boolean determining if you are using SNC. Set this to true to use SNC.
SNCName	An optional input with the name of your SNC connection.
SNCQop	The quality of protection for your SNC connection.
SNCPartnerName	The application server's SNC name. This is a required input when using SNC.
SNCLibPath	An optional input detailing the path and file name of the external library.

45.2.3. SSL

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

45.2.4. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.

FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

45.2.5. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

45.2.6. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.

Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

45.2.7. Schema

Property	Description
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

45.2.8. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.

CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

45.2.9. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
Charset	The system code page used for Unicode to multibyte translations.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the SAP ERP when the connection is opened.
Destination	Reference to an existing destination that is specified in a local saprfc.ini file.
EndianType	The endian type for the SAP server. Enter either Big or Little.
GatewayHost	The gateway host you wish to connect to.
GatewayService	The gateway service you wish to connect to.
GenerateSchemaFiles	Determines how schema files should be generated.
InitialValueMode	How to treat initial values in SAP.
Language	The language value to be used when connecting to the SAP system.

Location	A path to the directory that contains the schema files defining tables, views, and stored procedures to work with your chosen data source. This must be set in order to add new tables to provider.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
Pagesize	The number of results to return per page from SAP. Only used for SAP tables.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
QueryMode	Determines which SAP queries will be displayed as views, if any.
ReadTableFunction	The function to use for reading table data.
RTK	The runtime key used for licensing.
StoredProcedureFilter	A filter indicating which function modules to report as stored procedures.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
TableMode	Determines which SAP tables will be displayed as views if any.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.

UseConnectionPooling	This property enables connection pooling.
UseLabels	Boolean determining if labels should be used for table and column names.
UseSimpleNames	Boolean determining if simple names should be used for tables and columns.
UseUnicodeRFC	A boolean indicating if you want to use RFC_GET_UNICODE_STRUCTURE to get structure information.

For more details, see <https://cdn.cdata.com/help/RYG/jdbc/>.

46. SAP CLOUD FOR CUSTOMER

46.1. SAP Cloud for Customer Version Support

The driver supports read/write access to SAP Cloud for Customer through the OData API.

46.2. Connection Options

46.2.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to SAP Cloud for Customer.
URL	The SAP Cloud for Customer url to connect to. Use this to override the default service of c4codata.
Tenant	The SAP Cloud for Customer tenant a connection should be established to.
UseCustomService	Indicates if the custom SAP Cloud for Customer OData service should be used.
User	Specifies the user ID of the authenticating SAP Cloud for Customer user account.
Password	Specifies the password of the authenticating user account.

46.2.2. SSO

Property	Description
SSOProperties	Additional properties required to connect to the identity provider, formatted as a semicolon-separated list.

46.2.3. OAuth

Property	Description
InitiateOAuth	Specifies the process for obtaining or refreshing the OAuth access token, which maintains user access while an authenticated, authorized user is working.
OAuthClientId	Specifies the client Id that was assigned the custom OAuth application was created. (Also known as the consumer key.) This ID registers the custom application with the OAuth authorization server.
OAuthClientSecret	Specifies the client secret that was assigned when the custom OAuth application was created. (Also known as the consumer secret). This secret registers the custom application with the OAuth authorization server.
OAuthAccessToken	A token received after authentication to the OAuth network, granting the user access. The access token is used in place of the user's login ID and password, which stay on the server.
OAuthSettingsLocation	Specifies the location of the settings file where OAuth values are saved. Storing OAuth settings in a central location avoids the need for users to enter OAuth connection properties manually each time they log in. It also enables credentials to be shared across connections or processes.
CallbackURL	The URL users return to after authenticating to SAP Cloud for Customer via OAuth.
OAuthVerifier	Specifies a verifier code returned from the OAuthAuthorizationURL . Used when authenticating to OAuth on a headless server, where a browser can't be launched. Requires both OAuthSettingsLocation and OAuthVerifier to be set.
OAuthExpiresIn	Specifies the duration in seconds, of an OAuth Access Token's lifetime. The token can be reissued to keep access alive as long as the user keeps working.
OAuthTokenTimestamp	Displays a Unix epoch timestamp in milliseconds that shows how long ago the current Access Token was created.

46.2.4. SSL

Property	Description
SSLServerCert	Specifies the certificate to be accepted from the server when connecting using TLS/SSL.

46.2.5. Firewall

Property	Description
FirewallType	Specifies the protocol the provider uses to tunnel traffic through a proxy-based firewall.
FirewallServer	Identifies the IP address, DNS name, or host name of a proxy used to traverse a firewall and relay user queries to network resources.
FirewallPort	Specifies the TCP port to be used for a proxy-based firewall.
FirewallUser	Identifies the user ID of the account authenticating to a proxy-based firewall.
FirewallPassword	Specifies the password of the user account authenticating to a proxy-based firewall.

46.2.6. Proxy

Property	Description
ProxyAutoDetect	Specifies whether the provider checks your system proxy settings for existing proxy server configurations, rather than using a manually specified proxy server.
ProxyServer	The hostname or IP address of the proxy server that you want to route HTTP traffic through.

ProxyPort	The TCP port on your specified proxy server (set in the ProxyServer connection property) that has been reserved for routing HTTP traffic to and from the client.
ProxyAuthScheme	Specifies the authentication method the provider uses when authenticating to the proxy server specified in the ProxyServer connection property.
ProxyUser	The username of a user account registered with the proxy server specified in the ProxyServer connection property.
ProxyPassword	The password associated with the user specified in the ProxyUser connection property.
ProxySSLType	The SSL type to use when connecting to the proxy server specified in the ProxyServer connection property.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the proxy server set in the ProxyServer connection property.

46.2.7. Logging

Property	Description
Logfile	Specifies the file path to the log file where the provider records its activities, such as authentication, query execution, and connection details.
Verbosity	Specifies the verbosity level of the log file, which controls the amount of detail logged. Supported values range from 1 to 5.
LogModules	Specifies the core modules to include in the log file. Use a semicolon-separated list of module names. By default, all modules are logged.
MaxLogFileSize	Specifies the maximum size of a single log file in bytes. For example, '10 MB'. When the file reaches the limit, the provider creates a new log file with the date and time appended to the name.
MaxLogFileCount	Specifies the maximum number of log files the provider retains. When the limit is reached, the oldest log file is deleted to make space for a new one.

46.2.8. Schema

Property	Description
Location	Specifies the location of a directory containing schema files that define tables, views, and stored procedures. Depending on your service's requirements, this may be expressed as either an absolute path or a relative path.
BrowsableSchemas	Optional setting that restricts the schemas reported to a subset of all available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC .
Tables	Optional setting that restricts the tables reported to a subset of all available tables. For example, Tables=TableA,TableB,TableC .
Views	Optional setting that restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC .

46.2.9. Caching

Property	Description
AutoCache	Specifies whether the content of tables targeted by SELECT queries is automatically cached to the specified cache database.
CacheDriver	The driver class of a JDBC driver. The specified driver is used to connect to the target database for all caching operations.
CacheConnection	The connection string for the specified cache database.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance, in seconds, for stale data in the specified cache database.
Offline	Get the data from the specified cache database instead of live SAP Cloud for Customer data.
CacheMetadata	Determines whether the provider caches table metadata to a file-based cache database.

46.2.10. Miscellaneous

Property	Description
BatchSize	Specifies the maximum number of rows included in each batch operation. Set to 0 to submit the entire batch as a single request.
ConnectionLifeTime	Specifies the maximum lifetime of a connection in seconds. When the specified time elapses, the provider closes the connection. Set to 0 or leave empty for no time limit.
ConnectOnOpen	Specifies whether the provider establishes a connection to SAP Cloud for Customer immediately upon opening the connection. The default value is false. Enable this property only if immediate connectivity verification is necessary.
MaxRows	Specifies the maximum rows returned for queries without aggregation or GROUP BY.
Other	Specifies additional hidden properties for specific use cases. These are not required for typical provider functionality. Use a semicolon-separated list to define multiple properties.
Pagesize	Specifies the maximum number of results to return from SAP Cloud for Customer, per page. This setting overrides the default page size set by the datasource, which is optimized for most use cases.
PoolIdleTimeout	Specifies the maximum idle time, in seconds, that a connection can remain in the pool before being closed.
PoolMaxSize	Specifies the maximum number of connections allowed in the connection pool. The default value is 100. Set to 0 or a negative value to disable pooling.
PoolMinSize	Specifies the minimum number of connections maintained in the connection pool. The default value is 1.
PoolWaitTime	Specifies the maximum number of seconds a connection request waits for an available connection in the pool. If the wait exceeds this time, an error is returned. The default is 60 seconds.

[PseudoColumns](#)

Specifies the pseudocolumns to expose as table columns. Use the format 'TableName=ColumnName;TableName=ColumnName'. The default is an empty string, which disables this property.

[Readonly](#)

Toggles read-only access to SAP Cloud for Customer from the provider.

[RTK](#)

Specifies the runtime key for licensing the provider. If unset or invalid, the provider defaults to the standard licensing method. Use this property only in environments where the standard licensing method is unsupported or requires a runtime key.

[Timeout](#)

Specifies the maximum time, in seconds, that the provider waits for a server response before throwing a timeout error. The default is 60 seconds. Set to 0 to disable the timeout.

[UseClientSidePaging](#)

Whether or not the CData ADO.NET Provider for SAP Cloud for Customer should use client side paging.

[UseConnectionPooling](#)

Enables the connection pooling feature, which allows the provider to reuse existing connections instead of creating new ones for each request.

[UserDefinedViews](#)

Specifies a filepath to a JSON configuration file defining custom views. The provider automatically detects and uses the views specified in this file.

For more details, see [CData JDBC Driver for SAP Cloud for Customer](#).

47. SAP HANA

47.1. Connection Options

47.1.1. Authentication

Property	Description
AuthScheme	The scheme used for authentication. Accepted entries are Password.
Server	The name of the server running SAP HANA database.
Port	The port of the SAP HANA database.
Database	The name of the SAP HANA database.
User	The SAP HANA user account used to authenticate.
Password	The password used to authenticate the user.
IncludeSystemObjects	Set IncludeSystemObjects to True to fetch Hana System schema and tables.
UseSSL	This field sets whether SSL is enabled.

47.1.2. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.

SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

47.1.3. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertSubject	The subject of the SSH client certificate.
SSHClientCertType	The type of SSHClientCert certificate.
SSHServer	The SSH server.
SSHPort	The SSH port.
SSHUser	The SSH user.
SSHPassword	The SSH password.
SSHServerFingerprint	The SSH server fingerprint.
UseSSH	Whether to tunnel the SAP HANA connection over SSH. Use SSH.

47.1.4. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

47.1.5. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

47.1.6. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

47.1.7. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the SAP HANA when the connection is opened.
IncludeTableTypes	If set to true, the provider will report the types of individual tables and views.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.

PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
QueryPassthrough	This option passes the query to the SAP HANA server as is.
Readonly	You can use this property to enforce read-only access to SAP HANA from the provider.
RTK	The runtime key used for licensing.
SessionVariables	A comma-separated list of session variables to set on the current connection.
Timeout	A timeout for the provider.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/SNG/jdbc/>.

48. SERVICE NOW

48.1. ServiceNow Version Support

The driver models ServiceNow APIs as bidirectional tables. The SOAP API is used to dynamically resolve the metadata from ServiceNow, regardless of the platform version. The REST API is then used to work with data.

48.2. Connection Options

48.2.1. Authentication

Property	Description
AuthScheme	The authorization scheme to be used when server authorization is to be performed.
URL	The base URL of your ServiceNow instance.
User	Specifies the user ID of the authenticating ServiceNow user account.
Password	Specifies the password of the authenticating user account.

48.2.2. SSO

Property	Description
SSOLoginURL	The identity provider's login URL.
SSOProperties	Additional properties required to connect to the identity provider, formatted as a semicolon-separated list.
CredentialsLocation	The location of the settings file where MFA credentials are saved.

48.2.3. OAuth

Property	Description
InitiateOAuth	Specifies the process for obtaining or refreshing the OAuth access token, which maintains user access while an authenticated, authorized user is working.
OAuthClientId	Specifies the client Id that was assigned the custom OAuth application was created. (Also known as the consumer key.) This ID registers the custom application with the OAuth authorization server.
OAuthClientSecret	Specifies the client secret that was assigned when the custom OAuth application was created. (Also known as the consumer secret). This secret registers the custom application with the OAuth authorization server.
OAuthAccessToken	A token received after authentication to the OAuth network, granting the user access. The access token is used in place of the user's login ID and password, which stay on the server.
OAuthSettingsLocation	Specifies the location of the settings file where OAuth values are saved. Storing OAuth settings in a central location avoids the need for users to enter OAuth connection properties manually each time they log in. It also enables credentials to be shared across connections or processes.
CallbackURL	The URL users return to after authenticating to ServiceNow via OAuth.
OAuthGrantType	Specifies the grant type for the chosen OAuth flow. This value should be the same as the grant_type that was set during OAuth custom application creation.
OAuthVerifier	Specifies a verifier code returned from the OAuthAuthorizationURL . Used when authenticating to OAuth on a headless server, where a browser can't be launched. Requires both OAuthSettingsLocation and OAuthVerifier to be set.
OAuthRefreshToken	Gets and refreshes the currently-active OAuth Access Token.
OAuthExpiresIn	Specifies the duration in seconds, of an OAuth Access Token's lifetime. The token can be reissued to keep access alive as long as the user keeps working.

[OAuthTokenTimestamp](#)

Displays a Unix epoch timestamp in milliseconds that shows how long ago the current Access Token was created.

48.2.4. SSL

Property	Description
SSLServerCert	Specifies the certificate to be accepted from the server when connecting using TLS/SSL.

48.2.5. Firewall

Property	Description
FirewallType	Specifies the protocol the provider uses to tunnel traffic through a proxy-based firewall.
FirewallServer	Identifies the IP address, DNS name, or host name of a proxy used to traverse a firewall and relay user queries to network resources.
FirewallPort	Specifies the TCP port to be used for a proxy-based firewall.
FirewallUser	Identifies the user ID of the account authenticating to a proxy-based firewall.
FirewallPassword	Specifies the password of the user account authenticating to a proxy-based firewall.

48.2.6. Proxy

Property	Description
ProxyAutoDetect	Specifies whether the provider checks your system proxy settings for existing proxy server configurations, rather than using a manually specified proxy server.

ProxyServer	The hostname or IP address of the proxy server that you want to route HTTP traffic through.
ProxyPort	The TCP port on your specified proxy server (set in the ProxyServer connection property) that has been reserved for routing HTTP traffic to and from the client.
ProxyAuthScheme	Specifies the authentication method the provider uses when authenticating to the proxy server specified in the ProxyServer connection property.
ProxyUser	The username of a user account registered with the proxy server specified in the ProxyServer connection property.
ProxyPassword	The password associated with the user specified in the ProxyUser connection property.
ProxySSLType	The SSL type to use when connecting to the proxy server specified in the ProxyServer connection property.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the proxy server set in the ProxyServer connection property.

48.2.7. Logging

Property	Description
Logfile	Specifies the file path to the log file where the provider records its activities, such as authentication, query execution, and connection details.
Verbosity	Specifies the verbosity level of the log file, which controls the amount of detail logged. Supported values range from 1 to 5.
LogModules	Specifies the core modules to include in the log file. Use a semicolon-separated list of module names. By default, all modules are logged.
MaxLogFileSize	Specifies the maximum size of a single log file in bytes. For example, '10 MB'. When the file reaches the limit, the provider creates a new log file with the date and time appended to the name.

[MaxLogFileCount](#)

Specifies the maximum number of log files the provider retains. When the limit is reached, the oldest log file is deleted to make space for a new one.

48.2.8. Schema

Property	Description
Location	Specifies the location of a directory containing schema files that define tables, views, and stored procedures. Depending on your service's requirements, this may be expressed as either an absolute path or a relative path.
BrowsableSchemas	Optional setting that restricts the schemas reported to a subset of all available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC .
Tables	Optional setting that restricts the tables reported to a subset of all available tables. For example, Tables=TableA,TableB,TableC .
Views	Optional setting that restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC .
IncludeDatabaseViews	Controls whether to expose the Database Views or not.
IncludeSystemTables	Controls whether to expose the system tables or not.

48.2.9. Caching

Property	Description
AutoCache	Specifies whether the content of tables targeted by SELECT queries is automatically cached to the specified cache database.
CacheDriver	The driver class of a JDBC driver. The specified driver is used to connect to the target database for all caching operations.
CacheConnection	The connection string for the specified cache database.

CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance, in seconds, for stale data in the specified cache database.
Offline	Get the data from the specified cache database instead of live ServiceNow data.
CacheMetadata	Determines whether the provider caches table metadata to a file-based cache database.

48.2.10. Data Formatting

Property	Description
PushEmptyValuesAsNull	Indicates whether to read the empty values as empty or as null.

48.2.11. Miscellaneous

Property	Description
BatchSize	Specifies the maximum number of rows included in each batch operation. Set to 0 to submit the entire batch as a single request.
ConnectionLifeTime	Specifies the maximum lifetime of a connection in seconds. When the specified time elapses, the provider closes the connection. Set to 0 or leave empty for no time limit.
ConnectOnOpen	Specifies whether the provider establishes a connection to ServiceNow immediately upon opening the connection. The default value is false. Enable this property only if immediate connectivity verification is necessary.
DisplayValue	Based on this value, the provider retrieves the display value or the actual value from the database.
DisplayValueColumnName	Pattern for displaying column names when DisplayValue property is set to ALL.

[EnforceColumnValues](#)

Controls whether available values for an element are retrieved and checked when executing INSERT or UPDATE statements.

[ExcludeReferenceLink](#)

Based on this value, the additional information provided for reference fields will be suppressed or not.

[GetColumnsMetadata](#)

Indicates the user preference as to when and how the provider should get the columns metadata.

[MaxRows](#)

Specifies the maximum rows returned for queries without aggregation or GROUP BY.

[Other](#)

Specifies additional hidden properties for specific use cases. These are not required for typical provider functionality. Use a semicolon-separated list to define multiple properties.

[Pagesize](#)

Specifies the maximum number of results to return from ServiceNow, per page. This setting overrides the default page size set by the datasource, which is optimized for most use cases.

[PoolIdleTimeout](#)

Specifies the maximum idle time, in seconds, that a connection can remain in the pool before being closed.

[PoolMaxSize](#)

Specifies the maximum number of connections allowed in the connection pool. The default value is 100. Set to 0 or a negative value to disable pooling.

[PoolMinSize](#)

Specifies the minimum number of connections maintained in the connection pool. The default value is 1.

[PoolWaitTime](#)

Specifies the maximum number of seconds a connection request waits for an available connection in the pool. If the wait exceeds this time, an error is returned. The default is 60 seconds.

[PseudoColumns](#)

Specifies the pseudocolumns to expose as table columns. Use the format 'TableName=ColumnName;TableName=ColumnName'. The default is an empty string, which disables this property.

[Readonly](#)

Toggles read-only access to ServiceNow from the provider.

[RTK](#)

Specifies the runtime key for licensing the provider. If unset or invalid, the provider defaults to the standard licensing method. Use this property only in environments where the standard licensing method is unsupported or requires a runtime key.

<u>ShowDescriptions</u>	Controls whether column and table descriptions are returned via the sys_documentation and sys_db_object API endpoints.
<u>Timeout</u>	Specifies the maximum time, in seconds, that the provider waits for a server response before throwing a timeout error. The default is 60 seconds. Set to 0 to disable the timeout.
<u>UseConnectionPooling</u>	Enables the connection pooling feature, which allows the provider to reuse existing connections instead of creating new ones for each request.
<u>UseDisplayColumnNames</u>	If True, the provider uses the display (GUI) names of columns instead of the API names.
<u>UseDisplayTableNames</u>	Boolean determining if the display names for the table should be used instead of the API names.
<u>UserDefinedViews</u>	Specifies a filepath to a JSON configuration file defining custom views. The provider automatically detects and uses the views specified in this file.

For more details, see [CData JDBC Driver for ServiceNow](#).

49. SHAREPOINT

49.1. SharePoint Version Support

The driver supports all versions of SharePoint that support the SOAP API. This includes: Windows SharePoint Services 3.0, SharePoint Server 2007+ (2010, 2013, etc.), and SharePoint Online.

49.2. Connection Options

49.2.1. Authentication

Property	Description
AuthScheme	The scheme used for authenticating to SharePoint. For On-Premise instances, accepted entries are NTLM, Basic, Digest, Forms, None, and Negotiate. NTLM is the default. For Sharepoint Online, the accepted entries are OAuth and OAuthJWT.
URL	The base URL for the site.
SharePointEdition	The edition of SharePoint being used. Set either SharePoint Online or SharePoint On-Premise.
User	The SharePoint user account used to authenticate.
Password	The password used to authenticate the user.

49.2.2. Azure Authentication

Property	Description
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tentant will be used.
AzureEnvironment	The Azure Environment to use when establishing a connection.

49.2.3. SSO

Property	Description
SSOLoginURL	The identity provider's login URL.
SSODomain	The domain of the user when using single sign-on (SSO).
SSOProperties	Additional properties required to connect to the identity provider in a semicolon-separated list.

49.2.4. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your Add-In settings.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
Scope	The scope used for the OAuth flow to access data from the Application.
OAuthGrantType	The grant type for the OAuth flow.

OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

49.2.5. JWT OAuth

Property	Description
OAuthJWTCert	The JWT Certificate store.
OAuthJWTCertType	The type of key store containing the JWT Certificate.
OAuthJWTCertPassword	The password for the OAuth JWT certificate.
OAuthJWTCertSubject	The subject of the OAuth JWT certificate.
OAuthJWTIssuer	The issuer of the Java Web Token.
OAuthJWTSubject	The user subject for which the application is requesting delegated access.

49.2.6. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
KerberosRealm	The Kerberos Realm used to authenticate the user with.

KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosServiceRealm	The Kerberos realm of the service.
KerberosServiceKDC	The Kerberos KDC of the service.
KerberosTicketCache	The full file path to an MIT Kerberos credential cache file.

49.2.7. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

49.2.8. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.

FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

49.2.9. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

49.2.10. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.

Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

49.2.11. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
Schema	The type of schema to use.

49.2.12. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.

CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

49.2.13. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
CalculatedDataType	The data type to be used for calculated fields.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the SharePoint when the connection is opened.
ContinueOnError	Indicates whether or not to continue updating items in a batch after an error.
CreateIDColumns	Indicates whether or not to create supplemental ID columns for SharePoint columns that use values from information stored in other Lists.
FolderOption	An option to determine how to display folders in results. Enter either FilesOnly, FilesAndFolders, Recursive, or RecursiveAll.

IncludeLookupColumns	This option controls whether the driver returns the lookup columns defined on a table.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from SharePoint.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
Readonly	You can use this property to enforce read-only access to SharePoint from the provider.
RTK	The runtime key used for licensing.
ShowHiddenColumns	Boolean determining if hidden columns should be shown or not. If false, all hidden columns will be removed from the column listing.
ShowPredefinedColumns	Boolean determining if predefined columns should be shown or not. If false, all columns derived from a base type will be removed from the column listing.
ShowVersionViews	Indicate whether to display the view of list versions. Such as ListA_Versions.
STSURL	The URL of the security token service (STS) when using single sign-on (SSO).

SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.
UseDisplayNames	Boolean determining if the display names for the columns should be used instead of the API names.
UseNTLMV1	Determines whether the driver will attempt to connect with NTLMv1 or NTLMv2 (default).
UseSimpleNames	Boolean determining if simple names should be used for tables and columns.

For more details, see <https://cdn.cdata.com/help/RSG/jdbc/>.

50. SNOWFLAKE

50.1. Snowflake Version Support

The driver enables standards-based access to all Snowflake editions.

50.2. Connection Options

50.2.1. Authentication

Property	Description
AuthScheme	The authentication scheme used. Accepted entries are Password, OKTA, PrivateKey, AzureAD, OAuth, or PingFederate.
Account	The Account provided for authentication with Snowflake database. This is usually derived from the URL automatically.
Warehouse	The name of the Snowflake warehouse.
User	The username provided for authentication with the Snowflake database.
Password	The user's password.
URL	The URL of Snowflake database.
MFAPasscode	Specifies the passcode to use for multi-factor authentication.
RoleName	The role of the Snowflake user: PUBLIC, SYSADMIN, or ACCOUNTADMIN.
CredentialsLocation	The location of the settings file where credentials are saved.

50.2.2. Connection

Property	Description
UseVirtualHosting	If true (default), buckets will be referenced in the request using the hosted-style request: <code>http://yourbucket.s3.amazonaws.com/yourobject</code> . If set to false, the bean will use the path-style request: <code>http://s3.amazonaws.com/yourbucket/yourobject</code> . Note that this property will be set to false, in case of an S3 based custom service when the CustomURL is specified.

50.2.3. Azure Authentication

Property	Description
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.

50.2.4. SSO

Property	Description
ProofKey	The ProofKey for authentication with Snowflake database. This is usually derived from GetSSOAuthorizationURL call.
ExternalToken	The External Token for authentication with the Snowflake database. This is usually derived from the external handler. For example, handle the callback URL from procedure GetSSOAuthorizationURL will get this token.
SSOProperties	Additional properties required to connect to the identity provider in a semicolon-separated list.

50.2.5. KeyPairAuth

Property	Description
PrivateKey	The private key provided for key pair authentication with Snowflake.
PrivateKeyPassword	The password for the private key specified in the PrivateKey property, if required.
PrivateKeyType	The type of key store containing the private key to use with key pair authentication.

50.2.6. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your Add-In settings.
State	An optional value that has meaning for your OAuth App.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.

OAuthAuthenticator	This determines the authenticator that the OAuth application requests from Snowflake.
Scope	This determines the scopes that the OAuth application requests from Snowflake.
OAuthAuthorizationURL	The authorization URL for the OAuth service.
OAuthAccessTokenURL	The URL to retrieve the OAuth access token from.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
PKCEVerifier	A random value used as input for calling GetOAuthAccessToken in the PKCE flow.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

50.2.7. SSL

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

50.2.8. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.

FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

50.2.9. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

50.2.10. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.

LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

50.2.11. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
Database	The name of the Snowflake database.
Schema	The schema of the Snowflake database.

50.2.12. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.

CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache.
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

50.2.13. Miscellaneous

Property	Description
AllowPreparedStatement	Prepare a query statement before its execution.
ApplicationName	The application name connection string property expresses the HTTP User-Agent.
AsyncQueryTimeout	The timeout for asynchronous requests issued by the provider to download large result sets.
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Snowflake when the connection is opened.
CustomStage	The name of a custom stage to use during bulk write operations.
EnableArrow	Whether to support Apache Arrow.

ExternalStageAWSAccessKey	Your AWS account access key. Only used when defining a CustomStage for bulk write operations.
ExternalStageAWSSecretKey	Your AWS account secret key. Only used when defining a CustomStage for bulk write operations.
ExternalStageAzureSASToken	The string value of the Azure Blob shared access signature.
IgnoreCase	Whether to ignore case in identifiers. Default: false.
IncludeTableTypes	If set to true, the provider will report the types of individual tables and views.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
MaxThreads	Specifies the number of concurrent requests.
MergeDelete	A boolean indicating whether batch DELETE statements should be converted to MERGE statements automatically. Only used when the DELETE statement's where clause contains a table's primary key field only and they are combined with AND logical operator.
MergeInsert	A boolean indicating whether INSERT statements should be converted to MERGE statements automatically. Only used when the INSERT contains a table's primary key field.
MergeUpdate	A boolean indicating whether batch UPDATE statements should be converted to MERGE statements automatically. Only used when the UPDATE statement's where clause contains a table's primary key field only and they are combined with AND logical operator.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from Snowflake.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.

PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
QueryPassthrough	This option passes the query to the Snowflake server as is.
Readonly	You can use this property to enforce read-only access to Snowflake from the provider.
ReplaceInvalidUTF8Characters	Specifies whether to replace invalid UTF8 characters with a '?'. s
RetryOnS3Timeout	Whether or not to retry when network issues occur at during chunk downloading.
RTK	The runtime key used for licensing.
SessionParameters	The session parameters for Snowflake. For example: SessionParameters='QUERY_TAG=MyTag;QUOTED_IDENTIFIERS_IGNORE_CASE=True;';.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseAsyncQuery	This field sets whether async query is enabled.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/OWG/jdbc/>.

51. SPARK

51.1. Spark SQL Version Support

The driver leverages Spark Thrift to enable bidirectional SQL access to SparkSQL data. Spark version 1.6 and above are supported.

51.2. Connection Options

51.2.1. Authentication

Property	Description
AuthScheme	The authentication scheme used. Accepted entries are Plain, LDAP, NOSASL, and Kerberos.
Server	The host name or IP address of the server hosting the SparkSQL database.
Port	The port for the SparkSQL database.
User	The username used to authenticate with SparkSQL.
Password	The password used to authenticate with SparkSQL.
Database	The name of the SparkSQL database.
ProtocolVersion	The Protocol Version used to authenticate with SparkSQL.
ImpersonationProxyUser	The proxy user of the Hive user impersonation.
SaslQop	Quality of protection for the SASL framework. The level of quality is negotiated between the client and server during authentication. Used by Kerberos authentication with TCP transport.
TransportMode	The transport mode to use to communicate with the Hive server. Accepted entries are BINARY and HTTP.

51.2.2. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
KerberosRealm	The Kerberos Realm used to authenticate the user with.
KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosServiceRealm	The Kerberos realm of the service.
KerberosServiceKDC	The Kerberos KDC of the service.
KerberosTicketCache	The full file path to an MIT Kerberos credential cache file.

51.2.3. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

51.2.4. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

51.2.5. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

51.2.6. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

51.2.7. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

51.2.8. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

51.2.9. Miscellaneous

Property	Description
AsyncQueryTimeout	The timeout for asynchronous requests issued by the provider to download large result sets.
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Spark SQL when the connection is opened.

DescribeCommand	The describe command to determine which describe command will use to communicate with the Hive server. Accepted entries are DESCRIBE and DESC.
DetectView	Specifies whether to use DESCRIBE FORMATTED ... to detect the specified table is view or not.
HTTPPath	The path component of the URL endpoint when using HTTP TransportMode.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
QueryPassthrough	This option passes the query to the Spark SQL server as is.
Readonly	You can use this property to enforce read-only access to Spark SQL from the provider.
RTK	The runtime key used for licensing.
ServerConfigurations	A name-value list of server configuration variables to override the server defaults.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.

UseConnectionPooling	This property enables connection pooling.
UseDatabricksUploadApi	This option specifies whether the Databricks Upload API will be use when executing batch insert.
UseInsertSelectSyntax	Specifies whether to use an INSERT INTO SELECT statement.
UseSSL	Specifies whether to use SSL Encryption when connecting to Hive.

For more details, see <https://cdn.cdata.com/help/ESG/jdbc/>.

52. SQL SERVER

52.1. SQL Server Version Support

The driver enables connectivity to SQL Server through the TDS protocol. SQL Server versions 2008, 2012, 2014, 2016, and 2019 are supported.

Connections to Azure SQL Server and Azure Data Warehouse instances are supported as well.

52.2. Connection Options

52.2.1. Authentication

Property	Description
AuthScheme	The scheme used for authentication. Accepted entries are Password, NTLM, Kerberos, AzurePassword, AzureAD, AzureMSI, AzureServicePrincipal.
Server	The name of the server running SQL Server.
Port	The port of the MS SQL Server.
Database	The name of the SQL Server database.
User	The SQL Server user account used to authenticate.
Password	The password used to authenticate the user.
Domain	The name of the domain for a Windows (NTLM) security login.
IntegratedSecurity	Whether or not to authenticate with Windows Integrated Security.
NTLMVersion	The NTLM version.
Encrypt	This field sets whether SSL is enabled.

52.2.2. Azure Authentication

Property	Description
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.

52.2.3. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
OAuthGrantType	The grant type for the OAuth flow.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.

52.2.4. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
KerberosRealm	The Kerberos Realm used to authenticate the user with.

KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosServiceRealm	The Kerberos realm of the service.
KerberosServiceKDC	The Kerberos KDC of the service.
KerberosTicketCache	The full file path to an MIT Kerberos credential cache file.

52.2.5. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

52.2.6. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.

SSHClientCertSubject	The subject of the SSH client certificate.
SSHClientCertType	The type of SSHClientCert certificate.
SSHServer	The SSH server.
SSHPort	The SSH port.
SSHUser	The SSH user.
SSHPassword	The SSH password.
SSHServerFingerprint	The SSH server fingerprint.
UseSSH	Whether to tunnel the SQL Server connection over SSH. Use SSH.

52.2.7. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

52.2.8. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

52.2.9. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).

MaxLogFileCount	A string specifying the maximum file count of log files.

52.2.10. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

52.2.11. Miscellaneous

Property	Description
ApplicationIntent	The application intent connection string property expresses the client application's request to be directed either to a read-write or read-only version of an availability group database. To use read-only routing, a client must use an application intent of read-only in the connection string when connecting to the availability group listener. Without the read-only application intent, connections to the availability group listener are directed to the database on the primary replica.
ApplicationName	The application name connection string property expresses the HTTP User-Agent.
DefaultDomain	This property is used for the Oracle Database Gateway for ODBC.

EnableForeignKeyDetection	Whether to detect the foreign keys in ODBC.
IncludeDualTable	Set this property to mock the Oracle DUAL table for better compatibility with Oracle database.
LimitKeySize	The maximum length of a primary key column.
MapBigintToVarchar	This property controls whether or not the bigint type maps to SQL_VARCHAR instead of SQL_BIGINT. This property is false by default.
MapToInt	This property controls whether or not the long type maps to SQL_INTEGER instead of SQL_BIGINT. This property is false by default.
MapToLongVarchar	This property controls whether or not a column is returned as SQL_LONGVARCHAR.
MapToWVarchar	This property controls whether or not string types map to SQL_WVARCHAR instead of SQL_VARCHAR. This property is set by default.
MaximumColumnSize	The maximum column size.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
QueryPassthrough	This option passes the query to the SQL Server server as is.
Readonly	You can use this property to enforce read-only access to SQL Server from the provider.
RTK	The runtime key used for licensing.
Timeout	A timeout for the provider.
UpperCaseIdentifiers	This property reports all identifiers in uppercase. This is the default for Oracle databases and thus allows better

integration with Oracle tools such as the Oracle Database Gateway.

For more details, see <https://cdn.cdata.com/help/RUG/odbc/>.

52.3. Description Extraction

To extract table and column descriptions for SQL Server, you must add the following parameters in the Options field of HOPEX Data Discovery:

ColumnExtendedProperty=MS_Description;

TableExtendedProperty=MS_Description;

Microsoft SQL Server

Server	vp-hadoop	i
Port	61299	i
User	sa	i
Password	*****	i
Azure Tenant		i
Options	ColumnExtendedProperty=MS_Description;TableExtendedProperty=MS_Description;	i

« Previous Test Connect Connect » Advanced

53. SQL SERVER ANALYSIS SERVICES

53.1. Connection Options

53.1.1. Authentication

Property	Description
AuthScheme	The scheme used for authentication. Accepted entries are NTLM, Basic, Digest, None, and Negotiate.
URL	The URL used to connect to the SQL Server Analysis Services.
User	The SQL Server Analysis Services user account used to authenticate.
Password	The password used to authenticate the user.

53.1.2. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
KerberosRealm	The Kerberos Realm used to authenticate the user with.
KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.

53.1.3. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

53.1.4. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

53.1.5. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

53.1.6. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).

MaxLogFileCount	A string specifying the maximum file count of log files.

53.1.7. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

53.1.8. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .

Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

53.1.9. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
Catalog	The Analysis Services catalog to use. This may also be known as a Database from within Analysis Services.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the SQL Server Analysis Services when the connection is opened.
CustomHeaders	Other headers as determined by the user (optional).
ExtraProperties	Additional properties to submit on each MDX request to SQL Server Analysis Services.
IncludeJoinColumns	Set this to true to include extra join columns on each table.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.

ResponseRowLimit	The number of response rows to allow before erroring. Set to 0 for no limit.
RTK	The runtime key used for licensing.
SplitMeasures	Set this to true to split Measures table into individual tables.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseConnectionPooling	This property enables connection pooling.
UseMDX	Set this to true to pass MDX queries to SQL Server Analysis Services as-is.

For more details, see <https://cdn.cdata.com/help/FYG/jdbc/>.

54. SYBASE

54.1. Sybase Version Support

The driver enables connectivity to Sybase Database (SAP Adaptive Server) through the TDS protocol. ASE Versions 12, 15, and 16 are supported. Connecting to Sybase SQL Anywhere or Sybase IQ are not supported in this driver.

54.2. Connection Options

54.2.1. Authentication

Property	Description
AuthScheme	The scheme used for authentication. Accepted entries are Password, Kerberos.
Server	The name of the server running Sybase Database.
Port	The port of the Sybase database.
Database	The name of the Sybase database.
User	The Sybase user account used to authenticate.
Password	The password used to authenticate the user.
Charset	Charset name to communicate with server.
UseSSL	This field sets whether SSL is enabled.
AlternateServers	This property allows you to specify multiple servers in addition to the one configured in Server and Port . Specify both a server name and port; separate servers with a comma.

54.2.2. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
KerberosRealm	The Kerberos Realm used to authenticate the user with.
KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosServiceRealm	The Kerberos realm of the service.
KerberosServiceKDC	The Kerberos KDC of the service.
KerberosTicketCache	The full file path to an MIT Kerberos credential cache file.

54.2.3. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

54.2.4. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

54.2.5. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

54.2.6. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

54.2.7. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Sybase when the connection is opened.
EncryptPassword	This field sets whether password encryption is enabled.
IgnoreGroupNumber	Whether to return group number (like 'num') of procedure when list procedures.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.

MaxTDSPacketSize	The protocol sending and receiving buffer size. For the lower version, it's from 512 to 512. For the newer version, it's from 512 to 2048.
Other	These hidden properties are used only in specific use cases.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
QueryPassthrough	This option passes the query to the Sybase server as is.
Readonly	You can use this property to enforce read-only access to Sybase from the provider.
RTK	The runtime key used for licensing.
SwitchMode	This property allows you to specify a switching mode to select a server from AlternateServers as the active server.
SwitchStrategy	This property allows you to specify a switching strategy to select a server from AlternateServers as the active server.
Timeout	A timeout for the provider.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/OYG/jdbc/>.

55. TERADATA

55.1. Prerequisite

The Teradata connector delivered by HOPEX is based on the native Teradata driver.

The native Teradata driver may already be installed with Teradata. If it is not the case, download it:

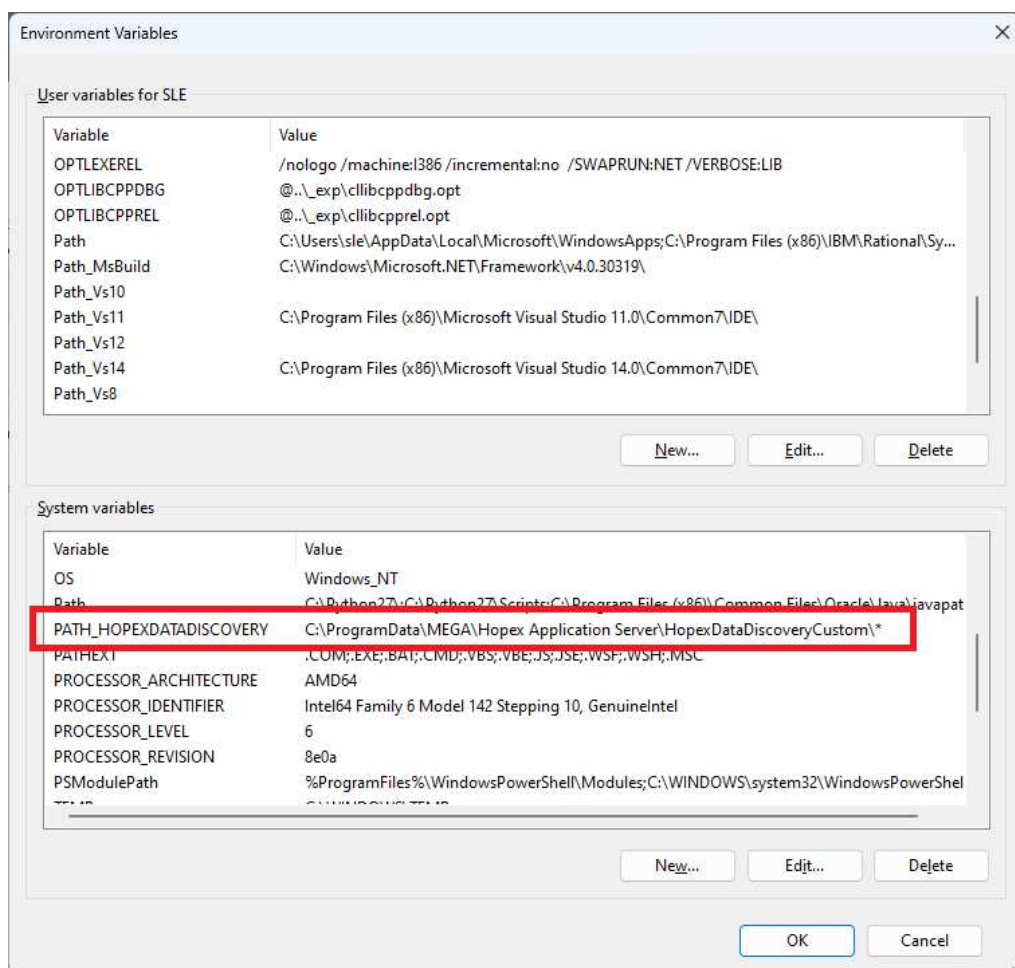
1> Download the terajdbc4.jar file from

<https://downloads.teradata.com/download/connectivity/jdbc-driver>

2> Put this file in a directory accessible from the HOPEX Data Discovery Module, but not the module directory.

For example: "C:\ProgramData\MEGA\Hopex Application Server\HopexDataDiscoveryCustom"

3> This directory must be filled in the environment variable "PATH_HOPEXDATADISCOVERY" which must be created if it does not exist:



55.2. Connection Options

55.2.1. Authentication

Property	Description
Port	Connects to the Teradata Database on the TCP/IP port specified. The default Teradata Database port is 1025.
User	The Teradata user account used to authenticate.
Password	The password used to authenticate the user.
Database	The database selected as the default database when a Teradata connection is opened.
DataSource	The Teradata server name or, equivalently, the DBC Name or TDPID.
Account	Specifies an account string to override the default account string defined for the Teradata Database user.
NewPassword	This connection parameter enables an application to change an expired password automatically.

55.2.2. Connection

Property	Description
ConnectFailureTTL	This option enables the CData ADO.NET Provider for Teradata remember the time of the last connection failure for each IP address/port combination. Also, the CData ADO.NET Provider for Teradata skips connection attempts to that IP address/port during subsequent logins for the number of seconds specified by the Connect Failure time-to-live (CONNECTFAILURETTL) value.
ConnectFunction	Specifies whether the Teradata Database should allocate a Logon Sequence Number (LSN) for this session or associate this session with an existing LSN.

55.2.3. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

55.2.4. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Log	Specifies the logging level (verbosity) for a connection. Logging is always enabled. The logging levels are listed in order from terse to verbose.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).
MaxLogFileCount	A string specifying the maximum file count of log files.

55.2.5. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.

55.2.6. Miscellaneous

Property	Description
BatchSize	The maximum size of each batch operation to submit.
Charset	Specifies the session character set for encoding and decoding character data transferred to and from the Teradata Database. The default value is ASCII.
ClientCharset	Specifies the Java character set for encoding and decoding character data transferred to and from the Teradata Database.
ColumnName	Controls the behavior of the ResultSetMetaData getColumnName and getColumnLabel methods.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the Teradata when the connection is opened.

Cop	Specifies whether COP Discovery is performed.
CopLast	Specifies how COP Discovery determines the last COP hostname.
Ddstats	Specify the value for DDSTATS.
DisableAutoCommitInBatch	Specifies whether or not disable the autocommit when executing the batch operation.
EncryptData	Specify the EncryptData value, ON or OFF.
ErrorQueryCount	Specifies the maximum number of times that JDBC FastLoad will attempt to query FastLoad Error Table 1 after a JDBC FastLoad operation.
ErrorQueryInterval	Specifies the number of milliseconds that JDBC FastLoad will wait in between attempts to query FastLoad Error Table 1 after a JDBC FastLoad operation.
ErrorTable1Suffix	Specifies the suffix for the name of FastLoad Error Table 1 created by JDBC FastLoad and JDBC FastLoad CSV.
ErrorTable2Suffix	Specifies the suffix for the name of FastLoad Error Table 2 created by JDBC FastLoad and JDBC FastLoad CSV.
ErrorTableDatabase	Specifies the database name for the FastLoad error tables created by JDBC FastLoad and JDBC FastLoad CSV.
FieldSep	Specifies a field separator for use with JDBC FastLoad CSV only. The default separator is ',' (comma).
FinalizeAutoClose	Specify the value for FinalizeAutoClose, ON or OFF.
GeturlCredentials	Specify the value for GeturlCredentials, ON or OFF.
Govern	Specify the value for GOVERN, ON or OFF.
LiteralUnderscore	Automatically escape LIKE-predicate patterns in DatabaseMetaData calls, such as schemPattern and tableNamePattern.
LobSupport	Specify the value for LobSupport, ON or OFF.

LobTempTable	Specifies the name of a table with the following columns: id integer, bval blob, cval clob.
LogData	Specifies additional data needed by a logon mechanism, such as a secure token, Distinguished Name, or a domain/realm name.
LogMech	Specifies the Logon Mechanism, which determines the connection's authentication and encryption capabilities.
LogonSequenceNumber	Specifies an existing Logon Sequence Number (LSN) to associate this session with.
MaxMessageBody	Specifies the maximum Response Message size in bytes.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
MaybeNull	Controls the behavior of the ResultSetMetaData.isNullable method.
Other	These hidden properties are used only in specific use cases.
Partition	Specifies the Teradata Database partition for the Connection.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PrepSupport	Specifies whether the Teradata Database performs a prepare operation when a PreparedStatement or CallableStatement is created.
QueryPassthrough	This option passes the query to the Teradata server as is.
Readonly	You can use this property to enforce read-only access to Teradata from the provider.

ReconnectCount	Enables Teradata Session Reconnect. Specifies the maximum number of times that the Teradata JDBC Driver will attempt to reconnect the session.
ReconnectInterval	Enables Teradata Session Reconnect. Specifies the number of seconds that the Teradata JDBC Driver will wait in between attempts to reconnect the session.
Redrive	Enables Teradata Session Reconnect, and also enables automatic redriving of SQL requests interrupted by database restart.
RTK	The runtime key used for licensing.
RunStartup	Specify the value for RunStartup, ON or OFF.
Sessions	Specifies the number of FastLoad or FastExport connections to be created, where $1 \leq \text{number of FastLoad or FastExport connections} \leq \text{number of AMPs}$.
SipSupport	Controls whether the Teradata Database and Teradata JDBC Driver use StatementInfo Parcel (SIP) to convey metadata.
SlobReceiveThreshold	Controls how small LOB values are received from the Teradata Database. Small LOB values are pre-fetched from the Teradata Database before the application explicitly reads data from Blob/Clob objects.
SlobTransmitThreshold	Controls how small LOB values are transmitted to the Teradata Database.
SpSpl	Specifies behavior for creating or replacing Teradata stored procedures.
StrictEncode	Specifies behavior for encoding character data to transmit to the Teradata Database.
Tcp	Specifies one or more TCP socket settings, separated by plus signs (" + ").
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.

TMode	Specifies the transaction mode for the connection.
TNano	Specifies the fractional seconds precision for all <code>java.sql.Time</code> values bound to a <code>PreparedStatement</code> or <code>CallableStatement</code> and transmitted to the Teradata Database as <code>TIME</code> or <code>TIME WITH TIME ZONE</code> values.
TrustedSql	Specify the value for <code>TrustedSql</code> .
TSNano	Specifies the fractional seconds precision for all <code>java.sql.Timestamp</code> values bound to a <code>PreparedStatement</code> or <code>CallableStatement</code> and transmitted to the Teradata Database as <code>TIMESTAMP</code> or <code>TIMESTAMP WITH TIME ZONE</code> values.
Type	Specifies the type of protocol to be used with the Teradata Database for SQL statements.
UpperCaseIdentifiers	This property reports all identifiers in uppercase. This is the default for Oracle databases and thus allows better integration with Oracle tools such as the Oracle Database Gateway.
UseConnectionPooling	This property enables connection pooling.
UseXViews	Specifies which Data Dictionary views should be queried to return result sets from <code>DatabaseMetaData</code> methods.

For more details, see <https://cdn.cdata.com/help/OTG/jdbc/>.

56. XML

56.1. Connection Options

56.1.1. Authentication

Property	Description
AuthScheme	The type of authentication to use when connecting to remote services.
AccessKey	Your account access key. This value is accessible from your security credentials page.
SecretKey	Your account secret key. This value is accessible from your security credentials page.
ApiKey	The API Key used to identify the user to IBM Cloud.
User	The XML user account used to authenticate.
Password	The password used to authenticate the user.
SharePointEdition	The edition of SharePoint being used. Set either SharePointOnline or SharePointOnPremise.

56.1.2. Connection

Property	Description
URI	The Uniform Resource Identifier (URI) for the XML resource location.
XPath	The XPath of an element that repeats at the same height within the XML document (used to split the document into multiple rows).

DataModel	Specifies the data model to use when parsing XML documents and generating the database metadata.
XMLFormat	Specifies the format of the XML document.
Region	The hosting region for your S3-like Web Services.
ProjectId	The id of the project where your Google Cloud Storage instance resides.
OracleNamespace	The Oracle Cloud Object Storage namespace to use.
StorageBaseURL	The URL of a cloud storage service provider.
SimpleUploadLimit	This setting specifies the threshold, in bytes, above which the provider will choose to perform a multipart upload rather than uploading everything in one request.
UseVirtualHosting	If true (default), buckets will be referenced in the request using the <code>hosted-style</code> request: <code>http://yourbucket.s3.amazonaws.com/yourobject</code> . If set to false, the bean will use the <code>path-style</code> request: <code>http://s3.amazonaws.com/yourbucket/yourobject</code> . Note that this property will be set to false, in case of an S3 based custom service when the CustomURL is specified.

56.1.3. AWS Authentication

Property	Description
AWSAccessKey	Your AWS account access key. This value is accessible from your AWS security credentials page.
AWSSecretKey	Your AWS account secret key. This value is accessible from your AWS security credentials page.
AWSRoleARN	The Amazon Resource Name of the role to use when authenticating.
AWSPrincipalArn	The ARN of the SAML Identity provider in your AWS account.

AWSRegion	The hosting region for your Amazon Web Services.
AWSSessionToken	Your AWS session token.
MFASerialNumber	The serial number of the MFA device if one is being used.
MFAToken	The temporary token available from your MFA device.

56.1.4. Azure Authentication

Property	Description
AzureStorageAccount	The name of your Azure storage account.
AzureAccessKey	The storage key associated with your XML account.
AzureSharedAccessSignature	A shared access key signature that may be used for authentication.
AzureTenant	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.
AzureEnvironment	The Azure Environment to use when establishing a connection.

56.1.5. SSO

Property	Description
SSOLoginURL	The identity provider's login URL.
SSOProperties	Additional properties required to connect to the identity provider in a semicolon-separated list.

56.1.6. OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthVersion	The version of OAuth being used.
OAuthClientId	The client ID assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthAccessTokenSecret	The OAuth access token secret for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
OAuthGrantType	The grant type for the OAuth flow.
OAuthPasswordGrantMode	How to pass Client ID and Secret with OAuthGrantType is set to Password.
OAuthIncludeCallbackURL	Whether to include the callback URL in an access token request.
OAuthAuthorizationURL	The authorization URL for the OAuth service.
OAuthAccessTokenURL	The URL to retrieve the OAuth access token from.

OAuthRefreshTokenURL	The URL to refresh the OAuth token from.
OAuthRequestTokenURL	The URL the service provides to retrieve request tokens from. This is required in OAuth 1.0.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
AuthToken	The authentication token used to request and obtain the OAuth Access Token.
AuthKey	The authentication secret used to request and obtain the OAuth Access Token.
OAuthParams	A comma-separated list of other parameters to submit in the request for the OAuth access token in the format paramname=value.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

56.1.7. JWT OAuth

Property	Description
OAuthJWTCert	The JWT Certificate store.
OAuthJWTCertType	The type of key store containing the JWT Certificate.
OAuthJWTCertPassword	The password for the OAuth JWT certificate.
OAuthJWTCertSubject	The subject of the OAuth JWT certificate.
OAuthJWTIssuer	The issuer of the Java Web Token.

OAuthJWTSubject	The user subject for which the application is requesting delegated access.

56.1.8. Kerberos

Property	Description
KerberosKDC	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
KerberosRealm	The Kerberos Realm used to authenticate the user with.
KerberosSPN	The service principal name (SPN) for the Kerberos Domain Controller.
KerberosKeytabFile	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
KerberosServiceRealm	The Kerberos realm of the service.
KerberosServiceKDC	The Kerberos KDC of the service.
KerberosTicketCache	The full file path to an MIT Kerberos credential cache file.

56.1.9. SSL

Property	Description
SSLClientCert	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
SSLClientCertType	The type of key store containing the TLS/SSL client certificate.
SSLClientCertPassword	The password for the TLS/SSL client certificate.
SSLClientCertSubject	The subject of the TLS/SSL client certificate.

SSLMode	The authentication mechanism to be used when connecting to the FTP or FTPS server.
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

56.1.10. SSH

Property	Description
SSHAuthMode	The authentication method to be used to log on to an SFTP server.
SSHClientCert	A certificate to be used for authenticating the user.
SSHClientCertPassword	The password of the SSHClientCert certificate if it has one.
SSHClientCertType	The type of SSHClientCert certificate.

56.1.11. Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

56.1.12. Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

56.1.13. Logging

Property	Description
Logfile	A filepath which designates the name and location of the log file.
Verbosity	The verbosity level that determines the amount of detail included in the log file.
LogModules	Core modules to be included in the log file.
MaxLogFileSize	A string specifying the maximum size in bytes for a log file (for example, 10 MB).

MaxLogFileCount	A string specifying the maximum file count of log files.

56.1.14. Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
BrowsableSchemas	This property restricts the schemas reported to a subset of the available schemas. For example, BrowsableSchemas=SchemaA,SchemaB,SchemaC.
Tables	This property restricts the tables reported to a subset of the available tables. For example, Tables=TableA,TableB,TableC.
Views	Restricts the views reported to a subset of the available tables. For example, Views=ViewA,ViewB,ViewC.
PushAttributes	Set PushAttributes to true to push any identified attributes as columns.
FlattenArrays	By default, nested arrays are returned as strings of XML. The FlattenArrays property can be used to flatten the elements of nested arrays into columns of their own. Set FlattenArrays to the number of elements you want to return from nested arrays.
FlattenObjects	Set FlattenObjects to true to flatten object properties into columns of their own. Otherwise, objects nested in arrays are returned as strings of XML.
QualifyColumns	Controls whether the provider will use relative column names.

56.1.15. Caching

Property	Description
AutoCache	Automatically caches the results of SELECT queries into a cache database specified by either CacheLocation or both of CacheConnection and CacheProvider .
CacheDriver	The database driver to be used to cache data.
CacheConnection	The connection string for the cache database. This property is always used in conjunction with CacheProvider . Setting both properties will override the value set for CacheLocation for caching data.
CacheLocation	Specifies the path to the cache when caching to a file.
CacheTolerance	The tolerance for stale data in the cache specified in seconds when using AutoCache .
Offline	Use offline mode to get the data from the cache instead of the live source.
CacheMetadata	This property determines whether or not to cache the table metadata to a file store.

56.1.16. Miscellaneous

Property	Description
BackwardsCompatibilityMode	Set BackwardsCompatibilityMode to true to use the XML functionality and features available in the 2017 version.
BatchSize	The maximum size of each batch operation to submit.

Charset	Specifies the session character set for encoding and decoding character data transferred to and from the XML file. The default value is UTF-8.
ClientCulture	This property can be used to specify the format of data (e.g., currency values) that is accepted by the client application. This property can be used when the client application does not support the machine's culture settings. For example, Microsoft Access requires 'en-US'.
ConnectionLifeTime	The maximum lifetime of a connection in seconds. Once the time has elapsed, the connection object is disposed.
ConnectOnOpen	This property species whether to connect to the XML when the connection is opened.
Culture	This setting can be used to specify culture settings that determine how the provider interprets certain data types that are passed into the provider. For example, setting Culture='de-DE' will output German formats even on an American machine.
CustomHeaders	Other headers as determined by the user (optional).
CustomUrlParams	The custom query string to be included in the request.
ExcludeFiles	Comma-separated list of file extensions to exclude from the set of the files modeled as tables.
FlattenRowLimit	The maximum number of rows that can result from a single flattened element.
GenerateSchemaFiles	Indicates the user preference as to when schemas should be generated and saved.
IncludeDropboxTeamResources	Indicates if you want to include Dropbox team files and folders.
IncludeFiles	Comma-separated list of file extensions to include into the set of the files modeled as tables.

MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
MetadataDiscoveryURI	Used when aggregating multiple files into one table, this property specifies a specific file to read to determined the aggregated table schema.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from XML.
PathSeparator	Determines the character which will be used to replace the file separator.
PoolIdleTimeout	The allowed idle time for a connection before it is closed.
PoolMaxSize	The maximum connections in the pool.
PoolMinSize	The minimum number of connections in the pool.
PoolWaitTime	The max seconds to wait for an available connection.
PseudoColumns	This property indicates whether or not to include pseudo columns as columns to the table.
Readonly	You can use this property to enforce read-only access to XML from the provider.
RowScanDepth	The number of rows to scan when dynamically determining columns for the table.
RTK	The runtime key used for licensing.
SupportEnhancedSQL	This property enhances SQL functionality beyond what can be supported through the API directly, by enabling in-memory client-side processing.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.

TypeDetectionScheme	Determines how to determine the data types of columns.
URISeparator	A delimiter used to separate different values in the URI property.
UseConnectionPooling	This property enables connection pooling.

For more details, see <https://cdn.cdata.com/help/DVG/jdbc/>.