

# **HOPEX Data Architecture**

## **Guide d'utilisation**



## HOPEX Aquila 6.2

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 MEGA International.

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 MEGA International.

© MEGA International, Paris, 1996 - 2026

Tous droits réservés.

HOPEX Data Architecture et HOPEX sont des marques réservées de MEGA International.

Windows est une marque réservée de Microsoft.

Les autres marques citées appartiennent à leurs propriétaires respectifs.

# SOMMAIRE



---

<b>Sommaire</b> . . . . .	<b>3</b>
---------------------------	----------

---

<b>Introduction à HOPEX Data Architecture</b> . . . . .	<b>19</b>
---	-----------

---

<b>Périmètre de la solution HOPEX IA</b> . . . . .	<b>20</b>
--	-----------

Trois niveaux de modélisation . . . . .	20
---	----

Catégorie des données . . . . .	20
---------------------------------	----

Workflow de conception . . . . .	20
----------------------------------	----

Définition des responsabilités . . . . .	21
--	----

Rapports d'analyse . . . . .	21
------------------------------	----

<b>Se connecter à HOPEX Data Architecture</b> . . . . .	<b>22</b>
---	-----------

Les profils de HOPEX Data Architecture . . . . .	22
--	----

Les rôles métier de HOPEX Data Architecture . . . . .	23
---	----

<b>Le bureau de HOPEX Information Architecture</b> . . . . .	<b>25</b>
--	-----------

Page d'accueil de HOPEX Data Architecture . . . . .	25
---	----

<i>Indicateurs du périmètre</i> . . . . .	26
---	----

Afficher l'environnement de travail d'une entreprise . . . . .	27
--	----

<i>Créer une entreprise et son environnement de travail</i> . . . . .	27
---	----

## GLOSSAIRE MÉTIER

---

<b>Introduction à la conception d'une ontologie métier</b> . . . . .	<b>31</b>
--	-----------

Le processus de gestion du vocabulaire . . . . .	31
--	----

<i>L'analyse et l'organisation des concepts métier</i> . . . . .	32
--	----

<i>La réalisation des concepts</i> . . . . .	32
--	----

---

<b>Consulter le glossaire métier . . . . .</b>	<b>33</b>
<b>Rechercher des termes dans le glossaire métier . . . . .</b>	<b>34</b>
Condition préalable . . . . .	34
Périmètre de recherche . . . . .	34
Lancer la recherche . . . . .	34
<i>Filtres de la recherche</i> . . . . .	34
<i>Filtres des résultats</i> . . . . .	35
Afficher le détail d'un terme . . . . .	37
<i>Caractéristiques standard</i> . . . . .	38
<i>Caractéristiques avancées</i> . . . . .	38
<b>Générer un glossaire . . . . .</b>	<b>39</b>
Lancer un rapport de glossaire . . . . .	39
Utiliser le glossaire dans un contexte multilingue . . . . .	39

---

<b>Définir les informations métier . . . . .</b>	<b>41</b>
<b>Synthèse des objets utilisés . . . . .</b>	<b>42</b>
Concept et terme . . . . .	42
Les liens entre les concepts . . . . .	42
<i>Les liens de définition</i> . . . . .	43
<i>Les liens de dépendance</i> . . . . .	43
Propriétés des concepts . . . . .	45
Instances de concept : les individus . . . . .	45
Le cycle de vie d'un concept ou d'un individu . . . . .	47
<i>Le cycle de vie d'un concept</i> . . . . .	48
<i>Le cycle de vie d'un individu</i> . . . . .	48
Les périodes . . . . .	50
La classification des concepts et la notion de type de concept . . . . .	51
La vue de concept . . . . .	52
La réalisation des éléments de dictionnaire . . . . .	52
<b>Présentation des diagrammes de modélisation des concepts . . . . .</b>	<b>53</b>
Le diagramme du domaine de concepts . . . . .	53
Le diagramme de structure de concept . . . . .	54
Le diagramme de structure de concept type . . . . .	54
Le diagramme de structure de concept d'état . . . . .	54
Le diagramme de structure d'individu . . . . .	55
Le diagramme de structure de cycle de vie de concept . . . . .	55
<b>Dictionnaire métier . . . . .</b>	<b>56</b>
Les éléments d'un dictionnaire métier . . . . .	56
<i>Accéder aux éléments d'un dictionnaire métier</i> . . . . .	57
<i>Importer des données métier</i> . . . . .	57
Dictionnaire métier de travail . . . . .	58
Créer un dictionnaire métier . . . . .	58
Initialiser un dictionnaire métier à partir de données logiques ou physiques . . . . .	58
<i>Initialiser un dictionnaire métier à partir de données logiques</i> . . . . .	58
<i>Initialiser un dictionnaire métier à partir de données physiques</i> . . . . .	59
<i>Initialiser un dictionnaire métier à partir de métadonnées physiques</i> . . . . .	59

<i>Initialiser un dictionnaire métier lors de la création de données logiques ou physiques.</i>	60
<i>Afficher le graphe de réalisation</i>	60
<b>Carte des domaines de concepts</b>	<b>63</b>
Créer une carte des domaines de concepts	63
Les composants d'une carte des domaines de concepts	63
Exemple de domaines de concepts	64
Rapports disponibles sur une carte des domaines de concepts	65
<b>Domaine de concepts</b>	<b>66</b>
Créer un domaine de concepts	66
Créer le diagramme de structure d'un domaine de concepts	66
Construire un diagramme de concepts	66
<i>Créer un diagramme de concepts d'un domaine de concept</i>	67
<i>Les composants d'un diagramme de concepts</i>	67
<i>Activer la fenêtre de vue.</i>	67
<i>Ajouter un élément dans un diagramme de concepts</i>	68
<i>Utiliser la barre d'insertion de l'objet</i>	68
<i>Synthèse des liens entre les objets</i>	69
<i>Accéder aux propriétés d'un lien dans un diagramme de concepts</i>	71
Définir les composants d'un domaine de concepts	73
<i>Les composants d'un domaine de concepts.</i>	73
<i>Définir le mode d'accès aux composants (CRUD).</i>	75
<b>Concept.</b>	<b>76</b>
Accéder à la liste des concepts	76
Créer un concept	76
Concepts et termes	77
<i>Relier un terme existant à un concept</i>	77
<i>Créer des termes en plusieurs langues à partir d'un concept.</i>	77
<i>Créer des synonymes en plusieurs langues.</i>	77
Renommer un concept	78
Les propriétés d'un concept	78
<i>Caractéristiques.</i>	78
<i>Composants</i>	79
<i>Super types</i>	79
<i>Réalisations</i>	79
<i>Réglementations</i>	79
<i>Qualité des données.</i>	79
<i>Rapports.</i>	80
<i>Workflows.</i>	80
<b>Composants d'un concept.</b>	<b>81</b>
Accéder aux composants d'un concept	81
Créer un composant de concept à partir d'un diagramme	82
Décrire le super composant d'un concept	83
Décrire un composant de concept calculé	83
<b>Propriétés d'un concept</b>	<b>84</b>
Créer une propriété de concept	85
<i>Créer une propriété de concept</i>	85
<i>Relier une propriété de concept à un concept</i>	85
<i>Relier deux propriétés de concept</i>	85
Créer une propriété de concept calculée	85
<b>Héritages d'un concept.</b>	<b>86</b>
Accéder aux héritages d'un concept	86

Définir l'héritage d'un concept à partir d'un diagramme de concept	86
Définir l'héritage d'un composant de concept	87
Créer une substitution d'un composant de concept	87
<b>Le diagramme de structure de concept</b>	<b>88</b>
<b>Individus</b>	<b>90</b>
Accéder à la liste des individus	90
Créer un individu à partir d'un dictionnaire métier	90
Les propriétés d'un individu	91
Créer une classification d'individu	91
Créer un composant d'entité de dictionnaire	92
Le diagramme de structure d'individu	92
<b>Etats d'un concept ou d'un individu</b>	<b>94</b>
Décrire les concepts d'état	94
<i>Accéder à la liste des concepts d'état</i>	95
<i>Créer un concept d'état à partir d'un dictionnaire métier</i>	95
<i>Les propriétés d'un concept d'état</i>	96
Décrire les concepts événement	97
<i>Accéder à la liste des concepts événement</i>	97
<i>Créer un concept événement à partir d'un dictionnaire métier</i>	97
<i>Les propriétés d'un concept événement</i>	98
<i>Relier un concept événement à son concept</i>	98
Le diagramme de structure de concept d'état	99
Décrire les états et les événements d'un individu	100
<i>Accéder à la liste des états et des événements d'individu</i>	102
<i>Créer un état d'individu à partir d'un domaine de concepts</i>	102
<i>Les propriétés d'un état d'individu</i>	103
<i>Créer un événement d'individu à partir d'un domaine de concepts</i>	103
<i>Relier un événement d'individu à un individu</i>	103
Le diagramme de structure de cycle de vie de concept	103
<i>Créer un cycle de vie de concept</i>	105
<i>Créer un diagramme de structure de cycle de vie de concept</i>	105
<i>Ajouter un événement de cycle de vie de concept</i>	105
<i>Créer une Transition de cycle de vie de concept</i>	106
Utiliser les périodes	106
<b>Concept type</b>	<b>108</b>
Accéder à la liste des concepts type	108
Créer un nouveau concept type	108
Les propriétés d'un concept type	108
Décrire les composants d'un concept type	109
<i>Accéder aux composants de concept type</i>	110
<i>Créer un composant de concept type à partir d'un domaine de concepts</i>	110
Décrire les variations d'un concept type	111
<i>Accéder aux variations d'un concept type</i>	111
<i>Créer une variation d'un concept type à partir d'un domaine de concepts</i>	111
Le diagramme de structure de concept type	111
<b>Vue de concept</b>	<b>113</b>
Créer une vue de concept	113
Définir le contenu de la vue de concept	114
<i>Affichage des objets de la vue</i>	114
<i>Ajouter un objet source à la vue de concept</i>	114
<i>Ajouter un composant à la vue de concept</i>	115
Le rapport de vue	115

---

**Règle de calcul sur un concept. . . . . 117**
**Associer une règle de calcul à une information métier . . . . . 118**

Règle de calcul sur une propriété de concept . . . . . 118

*Créer la règle de calcul. . . . . 119*
*Définir les objets en entrée et en sortie de la règle . . . . . 119*

Règle de calcul sur un concept . . . . . 120

---

**Relier les concepts métier à l'architecture logique et physique. . . . . 121**
**La réalisation de concept . . . . . 122**

Définir l'objet qui réalise un concept . . . . . 122

*Grapher la réalisation du concept. . . . . 123*

Définir le concept réalisé par une classe . . . . . 124

**Utiliser les matrices de réalisation . . . . . 126**

Les différents niveaux de réalisation . . . . . 126

Créer une matrice de réalisation . . . . . 127

## CONCEPTION DES DONNÉES ET DES BASES DE DONNÉES

---

**Modéliser les dictionnaires de données . . . . . 131**
**Options de modélisation des données logiques . . . . . 132**

Les formalismes . . . . . 132

Les notations . . . . . 132

**Vue d'ensemble des données logiques. . . . . 133**

Dictionnaire de données . . . . . 133

Carte des domaines de données . . . . . 133

Domaine de données logiques . . . . . 133

Vue de données logiques . . . . . 133

Modèle de données . . . . . 134

*Exemple . . . . . 134*
**Dictionnaire de données. . . . . 135**

Éléments d'un dictionnaire de données . . . . . 135

*Accéder aux éléments d'un dictionnaire de données. . . . . 136*
*Importer des données logiques . . . . . 136*
**Carte des domaines de données. . . . . 137**

Créer une carte des domaines de données . . . . . 137

Les composants d'une carte des domaines de données . . . . . 137

**Domaines de données et domaines de données logiques . . . . . 138**

Créer un domaine de données . . . . . 138

Le diagramme de domaine de données . . . . . 138

<i>Exemple de diagramme</i> . . . . .	139
<i>Créer le diagramme d'un domaine de données</i> . . . . .	139
<i>Ajouter un objet au diagramme</i> . . . . .	139
Ajouter un composant au domaine de données . . . . .	140
<i>Définir le mode d'accès aux composants (CRUD)</i> . . . . .	140
<b>Vues de données logiques</b> . . . . .	<b>142</b>
Créer une vue de données logiques . . . . .	142
<i>Créer une vue de données (à partir de la liste des vues)</i> . . . . .	143
<i>Créer une vue de données à partir d'un objet</i> . . . . .	143
Affichage des objets sources dans la vue de données . . . . .	144
Définir les composants de la vue de données . . . . .	145
<i>Composant embarqué</i> . . . . .	145
<i>Composant référencé</i> . . . . .	145
<i>Utiliser une vue dans une autre vue</i> . . . . .	146
<b>Diagramme de classes</b> . . . . .	<b>147</b>
Créer un paquetage . . . . .	147
Créer un diagramme de classes . . . . .	148
<b>Types de données</b> . . . . .	<b>149</b>
Paquetage de types de données . . . . .	149
Créer un nouveau paquetage de types de données . . . . .	150
<i>Créer un type de données</i> . . . . .	151
Référencer un paquetage de types de données . . . . .	151
Affecter des types aux attributs . . . . .	151
<b>Modèle de données</b> . . . . .	<b>153</b>
Synthèse des concepts . . . . .	153
<i>Modèle de données</i> . . . . .	153
<i>Diagramme de données</i> . . . . .	153
Construire un modèle de données . . . . .	154
<i>Condition préalable</i> . . . . .	154
<i>Créer le modèle de données</i> . . . . .	154
<i>Créer un diagramme de données</i> . . . . .	154
Les entités . . . . .	155
<i>Créer une entité</i> . . . . .	156
Les attributs . . . . .	157
<i>Créer un attribut</i> . . . . .	158
<i>Attributs hérités</i> . . . . .	158
Les associations . . . . .	158
<i>Créer une association</i> . . . . .	159
<i>Définir le rôle des associations</i> . . . . .	160
<i>Multiplicités</i> . . . . .	161
<i>Agrégation</i> . . . . .	163
<i>Composition</i> . . . . .	164
Associations réflexives . . . . .	165
Association "plus que binaire" . . . . .	166
Les contraintes . . . . .	167
Règles de normalisation . . . . .	168
<i>Première forme normale</i> . . . . .	168
<i>Deuxième forme normale</i> . . . . .	169
<i>Troisième forme normale</i> . . . . .	169
Généralisations . . . . .	170
<i>Qu'est-ce qu'une généralisation ?</i> . . . . .	170
<i>Cas de plusieurs sous-entités</i> . . . . .	172

<i>Héritage multiple</i> . . . . .	174
<i>Créer une généralisation</i> . . . . .	174
<i>Discriminant</i> . . . . .	174
Identifiant d'une entité . . . . .	175
<i>Identification par un attribut</i> . . . . .	176
Correspondance des modèles de données . . . . .	176
<i>Objectifs fonctionnels</i> . . . . .	176
<i>Lancer l'éditeur de correspondances</i> . . . . .	177
<i>Créer une correspondance</i> . . . . .	178
<i>Supprimer une correspondance</i> . . . . .	178
<i>Détails des correspondances</i> . . . . .	178
<i>Exemple de correspondance entre modèles de données</i> . . . . .	179
<b>La notation IDEF1X</b> . . . . .	<b>181</b>
A propos de la modélisation des données avec IDEF1X . . . . .	181
Synthèse des concepts . . . . .	182
Construire un modèle de données (IDEF1X) . . . . .	182
<i>Condition préalable</i> . . . . .	182
<i>Créer le modèle de données</i> . . . . .	182
<i>Diagramme de données (IDEF1X)</i> . . . . .	183
Entités (IDEF1X) . . . . .	183
<i>Créer une entité</i> . . . . .	183
<i>Attributs</i> . . . . .	184
Associations (IDEF1X) . . . . .	185
<i>Relation identifiante obligatoire</i> . . . . .	186
<i>Relation non-identifiante obligatoire</i> . . . . .	187
<i>Relation optionnelle non-identifiante</i> . . . . .	188
<i>Relation non-spécifique</i> . . . . .	188
<i>Entité associative</i> . . . . .	189
<i>Définir les rôles des associations</i> . . . . .	190
<i>Multiplicités</i> . . . . .	191
Relations de catégorisation (généralisations) - (IDEF1X) . . . . .	192
<i>Qu'est-ce qu'une Catégorisation (Généralisation) ?</i> . . . . .	193
<i>Créer une catégorisation</i> . . . . .	193
<i>Catégories multiples</i> . . . . .	194
<i>Multiples groupes de catégories</i> . . . . .	195
<i>Catégorisation complète</i> . . . . .	195
<i>Discriminant</i> . . . . .	196
<b>La Notation I.E.</b> . . . . .	<b>197</b>
A propos de la modélisation des données avec I.E. . . . .	197
Synthèse des concepts . . . . .	198
Créer un modèle de données (I.E) . . . . .	198
<i>Condition préalable</i> . . . . .	199
<i>Créer le modèle de données</i> . . . . .	199
<i>Diagramme de données (I.E)</i> . . . . .	199
Entités (I.E) . . . . .	200
<i>Créer une entité</i> . . . . .	200
<i>Attributs</i> . . . . .	200
Associations (I.E) . . . . .	201
<i>Présentation</i> . . . . .	201
<i>Associations et multiplicités</i> . . . . .	202
Sous-types (I.E) . . . . .	204
<i>Qu'est-ce qu'un sous-type ?</i> . . . . .	204

<i>Cas de plusieurs sous-types</i> . . . . .	205
<i>Intérêt des sous-types</i> . . . . .	206
<i>Héritage multiple</i> . . . . .	207
<i>Créer un sous-type</i> . . . . .	207
<b>La Notation Merise</b> . . . . .	<b>208</b>
A propos de la modélisation des données . . . . .	208
Synthèse des concepts . . . . .	208
Créer un modèle de données (Merise) . . . . .	208
<i>Condition préalable</i> . . . . .	209
<i>Créer le modèle de données</i> . . . . .	209
<i>Diagramme de données (Merise)</i> . . . . .	209
Les entités (Merise) . . . . .	209
<i>Créer une entité</i> . . . . .	210
Les associations (Merise) . . . . .	211
<i>Exemples d'associations</i> . . . . .	211
<i>Relation réflexive</i> . . . . .	212
<i>Relation plus que binaire</i> . . . . .	212
<i>Les participations ou cardinalités</i> . . . . .	213
<i>Créer une association (relation)</i> . . . . .	214
Les attributs (informations) - Merise . . . . .	215
<i>Les propriétés</i> . . . . .	215
<i>L'identifiant</i> . . . . .	216
<i>Créer des attributs</i> . . . . .	216
Les règles de normalisation (Merise) . . . . .	217
<i>Première forme normale</i> . . . . .	218
<i>Deuxième forme normale</i> . . . . .	218
<i>Troisième forme normale</i> . . . . .	219
Compléter la spécification du modèle de données (Merise) . . . . .	219
<i>Ordonner les attributs</i> . . . . .	219
<i>Description des attributs</i> . . . . .	220
<i>Participations ou Cardinalités</i> . . . . .	220
Sous-typage (Merise) . . . . .	222
<i>Qu'est-ce qu'un sous-type</i> . . . . .	223

---

## **Modéliser les bases de données. . . . . 227**

Formalisme logique et synchronisation . . . . .	228
<b>Base de données</b> . . . . .	<b>229</b>
Créer une base de données . . . . .	229
Propriétés d'une base de données . . . . .	229
Associer un paquetage à une base de données . . . . .	230
Importer une version de SGBD . . . . .	230
<b>Carte de schémas relationnels et schémas relationnels</b> . . . . .	<b>231</b>
Carte de schémas relationnels . . . . .	231
<i>Créer une carte de schémas relationnels</i> . . . . .	231
<i>Les composants d'une carte de schémas relationnels</i> . . . . .	231
Schéma relationnel . . . . .	232
<i>Créer un schéma relationnel</i> . . . . .	232
<i>Diagramme d'un schéma relationnel</i> . . . . .	232

<b>Diagramme relationnel</b>	<b>234</b>
Construire le diagramme relationnel	234
<i>Créer des objets dans le diagramme</i>	235
<i>Paramétrer l'affichage des diagrammes relationnels</i>	235
<b>Composants d'une base de données</b>	<b>237</b>
Les tables d'une base de données	237
<i>Créer une table</i>	237
<i>Supprimer une table</i>	237
Les colonnes d'une table	238
<i>Visualiser les colonnes</i>	238
<i>Créer une colonne</i>	238
<i>Supprimer une colonne</i>	239
Modifier les clés et les index	239
Créer une clé	240
<i>Clé primaire</i>	240
<i>Clé étrangère</i>	241
Créer un index	241
Ajouter une colonne à une clé ou un index	242
<b>Clés primaires et étrangères</b>	<b>243</b>
Spécifier les clés primaires	243
Spécifier les clés étrangères	244
Colonne clé primaire de deux tables	245
Colonne clé primaire de trois tables	245
<b>Types des données et Datatypes des colonnes</b>	<b>246</b>
Types de données des attributs	246
Déduire les datatypes des colonnes à partir des types des attributs	246
<i>Types pivots</i>	246
<i>Relier un type de données à un type pivot</i>	248
<i>Relier un type de données à un type pivot dans la notation UML</i>	249
Correspondances entre types pivots et datatypes	251
<i>Exemple de correspondances entre les types pivots et les datatypes pour Oracle 8</i>	251
Créer de nouveaux datatypes	253
<i>Exemple pour Oracle 10</i>	254
<i>Exemple pour SQL Server 7</i>	257
<b>Règles de modélisation d'une base de données</b>	<b>259</b>

---

## **Synchroniser les modèles logiques et physiques** . . . . . 261

Options d'affichage de la synchronisation	262
<b>Règles de synchronisation "logique vers physique"</b>	<b>263</b>
Synchronisation logique > physique : les Entités (ou Classes)	263
<i>Règle générale</i>	263
<i>Sous-entité</i>	263
<i>Entité abstraite</i>	264
<i>Entité réalisée</i>	264
Synchronisation logique > physique : les Associations	264
<i>Association contrainte (multiplicités : 0,1 ou 1,1)</i>	264
<i>Association contrainte (multiplicités : 0,1 et 0,1)</i>	266
<i>Verrous croisés (dead locks)</i>	266

<i>Association non-contrainte.</i> . . . . .	268
<i>Classe associative.</i> . . . . .	269
Synchronisation logique > physique : les Parties (Formalisme UML) . . . . .	269
<i>Exemple 1 : Non renseigné / *.</i> . . . . .	270
<i>Exemple 2 : Agrégation / *.</i> . . . . .	270
<i>Exemple 3 : Composition / 0..1.</i> . . . . .	270
<b>Du modèle logique au modèle physique</b> . . . . .	<b>271</b>
Lancer la synchronisation . . . . .	271
<i>Etape 1 : sélectionner les objets sources à synchroniser.</i> . . . . .	271
<i>Etape 2 : options de synchronisation.</i> . . . . .	272
<i>Etape 3 : protéger des objets.</i> . . . . .	273
<i>Etape 4 : valider les résultats.</i> . . . . .	273
Cas d'emploi des options . . . . .	274
<i>Prendre en compte les optimisations.</i> . . . . .	274
<i>Prendre en compte les suppressions.</i> . . . . .	274
<i>Combinaisons des options possibles.</i> . . . . .	275
Protéger des objets . . . . .	276
<i>Mode Figé.</i> . . . . .	276
<i>Mode Réalisé.</i> . . . . .	276
Résultats de la synchronisation : les correspondances . . . . .	277
<i>Caractéristiques de la correspondance.</i> . . . . .	277
<b>Synchronisation réduite (mode logique vers physique)</b> . . . . .	<b>279</b>
Objets sources de la synchronisation réduite . . . . .	279
<i>Lancement sur un modèle de données.</i> . . . . .	279
<i>Lancement sur une entité du modèle de données.</i> . . . . .	280
<i>Lancement sur une entité hors contexte.</i> . . . . .	280
Les stratégies de synchronisation réduite . . . . .	280
<i>Impact de l'objet synchronisé sur les autres objets.</i> . . . . .	280
<i>Impact des autres objets sur l'objet synchronisé.</i> . . . . .	281
<i>Tous les impacts.</i> . . . . .	283
Lancer la synchronisation réduite . . . . .	283
<i>Options de la synchronisation réduite.</i> . . . . .	284
<b>Relancer la synchronisation après modifications</b> . . . . .	<b>285</b>
Synchronisation après modification du diagramme de données . . . . .	285
<i>Création d'entités, d'associations et d'attributs dans le diagramme de données.</i> . . . . .	285
<i>Suppression d'entités, d'associations, d'attributs dans le diagramme de données.</i> . . . . .	285
<i>Modification des caractéristiques d'un attribut.</i> . . . . .	286
<i>Modification du nom d'un attribut, d'une entité, ou d'une association.</i> . . . . .	286
<i>Modification de la multiplicité maximum d'une association.</i> . . . . .	286
<i>Modification des liens d'une association.</i> . . . . .	286
Synchronisation après des modifications du diagramme physique . . . . .	287
<i>Suppression d'une table ou d'une colonne.</i> . . . . .	287
<i>Créations d'objets.</i> . . . . .	287
<i>Modifications de caractéristiques d'objets créés par la synchronisation.</i> . . . . .	287
<i>Modifications d'ordre.</i> . . . . .	287
<b>Du modèle physique au modèle logique</b> . . . . .	<b>288</b>
Règles de synchronisation "Physique vers Logique" . . . . .	288
Lancer la synchronisation . . . . .	291
<i>Etape 1 : sélectionner les objets à synchroniser.</i> . . . . .	291
<i>Etape 2 : options de synchronisation.</i> . . . . .	291
<i>Etape 3 : protéger des objets.</i> . . . . .	292
<i>Etape 4 : valider les résultats.</i> . . . . .	292

<i>Synchronisation réduite</i> . . . . .	292
Résultat de la synchronisation "Physique vers Logique" . . . . .	292
<i>Modèle de données détenteur</i> . . . . .	292
<i>Diagrammes de données</i> . . . . .	293
<i>Les correspondances</i> . . . . .	293
<b>Paramétrer la synchronisation</b> . . . . .	<b>294</b>
Préparer la synchronisation . . . . .	294
Options de création . . . . .	294
<i>Sur une base de données</i> . . . . .	294
<i>Sur le SGBD</i> . . . . .	296
Paramétrer la génération des noms . . . . .	296
<i>Règles de construction des noms</i> . . . . .	296
<i>Modifier une règle de construction</i> . . . . .	298
<i>Saisie du masque SQL</i> . . . . .	298
<i>Paramétrer le nom des colonnes de PK (identifiant implicite)</i> . . . . .	300
<b>Synchronisation des diagrammes.</b> . . . .	<b>303</b>
Cas de mise à jour des diagrammes lors d'une synchronisation . . . . .	303
<i>Après modification du diagramme source</i> . . . . .	303
<i>Après modification du diagramme cible</i> . . . . .	303
<i>Après modification des deux diagrammes</i> . . . . .	304
<i>Pas de modification détectée.</i> . . . .	304
<i>Cas particulier : une entité en correspondance avec deux tables</i> . . . . .	304
 <b>Correspondance des modèles</b> . . . . .	 <b>305</b>
 <b>L'éditeur de base de données.</b> . . . .	 <b>306</b>
Lancer l'éditeur sur une base de données . . . . .	306
<i>Créer un arbre de correspondance logique/physique</i> . . . . .	306
Créer une correspondance . . . . .	306
Supprimer une correspondance . . . . .	308
<b>Détails des correspondances</b> . . . . .	<b>309</b>
Propriétés d'une correspondance . . . . .	309
Rapport des correspondances . . . . .	310
Etat des objets . . . . .	311
<i>Enregistrer l'affichage des indicateurs de l'éditeur</i> . . . . .	311
Source de la correspondance . . . . .	311
Dessin de la correspondance . . . . .	313
 <b>Dénormaliser les modèles logiques et physiques</b> . . . . .	 <b>315</b>
 <b>Principes de la dénormalisation</b> . . . . .	 <b>316</b>
Dénormalisation : cohérence des modèles. . . . .	316
<i>Report des correspondances</i> . . . . .	316
<i>Supprimer les objets sources</i> . . . . .	316
Synchronisation et dénormalisation . . . . .	316
<i>Combinaison des options de dénormalisation et synchronisation</i> . . . . .	317

Dénormalisation : cas d'emploi . . . . .	317
<b>Dénormalisation logique . . . . .</b>	<b>320</b>
Lancer une dénormalisation logique . . . . .	320
<i>Exemple de dénormalisation logique . . . . .</i>	<i>320</i>
Liste des assistants de dénormalisation logique . . . . .	322
<i>Transformer une association en entité . . . . .</i>	<i>322</i>
<i>Transformer une entité en association . . . . .</i>	<i>322</i>
<i>Transformer une généralisation en association . . . . .</i>	<i>323</i>
<i>Transformer une association en généralisation . . . . .</i>	<i>324</i>
<i>Partition verticale d'une entité . . . . .</i>	<i>324</i>
<i>Partition horizontale d'une entité . . . . .</i>	<i>325</i>
<i>Fusion d'entités . . . . .</i>	<i>326</i>
<i>Fusion d'entités ascendantes . . . . .</i>	<i>326</i>
<i>Fusion d'entités descendantes . . . . .</i>	<i>327</i>
<i>Copier/Coller d'attributs . . . . .</i>	<i>327</i>
<b>Dénormalisation physique . . . . .</b>	<b>328</b>
Lancer une dénormalisation physique . . . . .	328
<i>Exemple de dénormalisation physique . . . . .</i>	<i>328</i>
Liste des assistants de dénormalisation physique . . . . .	329
<i>Partition verticale d'une table . . . . .</i>	<i>329</i>
<i>Partition horizontale d'une table . . . . .</i>	<i>330</i>
<i>Fusion de tables . . . . .</i>	<i>331</i>
<i>Transformation d'une clé étrangère en table . . . . .</i>	<i>332</i>
<i>Transformation d'une table en clé étrangère . . . . .</i>	<i>333</i>
<i>Copier/Coller des colonnes . . . . .</i>	<i>333</i>
 <b>Générer des scripts SQL . . . . .</b>	 <b>335</b>
<b>Lancer la génération SQL . . . . .</b>	<b>336</b>
Objets de la génération SQL . . . . .	336
Lancer l'assistant de génération . . . . .	336
<b>Génération incrémentale . . . . .</b>	<b>338</b>
Objets de la génération incrémentale . . . . .	338
Lancer la génération incrémentale . . . . .	338
<i>Options de la génération . . . . .</i>	<i>338</i>
<i>Lancer l'assistant de génération . . . . .</i>	<i>339</i>
<b>Paramétrer la génération SQL . . . . .</b>	<b>341</b>
Paramétrer la version de SGBD . . . . .	341
<i>SGBD et versions supportées . . . . .</i>	<i>341</i>
<i>Modifier les propriétés de la version de SGBD . . . . .</i>	<i>341</i>
Paramétrer la génération d'une base de données . . . . .	342
Préfixer le nom des objets . . . . .	344
<i>Héritage . . . . .</i>	<i>344</i>
<i>SGBD concernés . . . . .</i>	<i>344</i>
<b>Syntaxe supportée . . . . .</b>	<b>345</b>
Instruction CREATE TABLE . . . . .	345
<i>Gestion de NOT NULL . . . . .</i>	<i>345</i>
<i>Clause PRIMARY KEY . . . . .</i>	<i>346</i>
<i>Clause FOREIGN KEY . . . . .</i>	<i>346</i>

<i>Clause UNIQUE</i> . . . . .	348
Instruction CREATE INDEX (Oracle, Sybase, SQL Server) . . . . .	348
<i>Définition d'un index</i> . . . . .	348
<i>Traitement et génération des ordres SQL</i> . . . . .	349
<i>Clause CREATE VIEW</i> . . . . .	349
<b>Définir les vues d'une base de données</b> . . . . .	<b>350</b>
Créer les vues d'une base de données . . . . .	350
<i>Ajouter une table ou une colonne de vue</i> . . . . .	351
Définition SQL . . . . .	351
<i>Jointures de vue</i> . . . . .	351
<i>Mode utilisateur</i> . . . . .	352
<i>Champs</i> . . . . .	352
Définir un groupement de données. . . . .	352
<b>Définir les triggers pour une base de données.</b> . . . . .	<b>354</b>
Créer un trigger . . . . .	354
<i>Déclenchement du trigger.</i> . . . . .	354
<i>Références</i> . . . . .	354
<i>Définition SQL</i> . . . . .	355
Intégrité référentielle . . . . .	355
<b>Utiliser des procédures stockées</b> . . . . .	<b>356</b>
<b>Ajouter des propriétés physiques aux objets d'une base de données</b> . . . . .	<b>358</b>
SGBD cibles . . . . .	358
Créer des propriétés physiques . . . . .	358
<i>Objets contenant des paramètres physiques</i> . . . . .	359
<i>Créer une nouvelle clause.</i> . . . . .	360
<i>Relier une clause</i> . . . . .	360
<i>Nommage des clauses</i> . . . . .	360
<i>Exemple de personnalisation d'un modèle physique</i> . . . . .	362
Générer le fichier SQL . . . . .	364
 <b>Rétro-générer des tables</b> . . . . .	 <b>365</b>
<b>Lancer la rétro-génération</b> . . . . .	<b>366</b>
<b>Reconnaissance des Datatypes par ODBC</b> . . . . .	<b>367</b>
Problèmes de reconnaissance des datatypes . . . . .	369
<b>Rétro-génération des propriétés physiques</b> . . . . .	<b>370</b>
Valeurs par défaut . . . . .	370
Élimination des valeurs redondantes et transverses . . . . .	370
Cas spécifiques . . . . .	371
<i>Propriétés physiques des tablespaces.</i> . . . . .	371
<i>Rétro-génération des clusters</i> . . . . .	371
<b>Extraire la description de schémas de bases de données à partir de sources de données</b> <b>372</b>	
Configuration des sources de données requise . . . . .	372
Télécharger HOPEX Data Source Extractor . . . . .	372
Lancer l'extraction de données . . . . .	372
Fichier de compte-rendu de l'extraction . . . . .	378
Fichier résultat de l'extraction . . . . .	379

Personnaliser l'extraction ODBC . . . . .	379
<i>Utilisation du fichier Odwdbex.ini et des requêtes personnalisées . . . . .</i>	<i>380</i>
<i>Utilisation des API standard ODBC . . . . .</i>	<i>380</i>
Format des clauses SELECT . . . . .	381
<i>Clés primaires . . . . .</i>	<i>381</i>
<i>Clés étrangères . . . . .</i>	<i>382</i>
<i>Index . . . . .</i>	<i>383</i>
<i>Colonnes . . . . .</i>	<i>384</i>

---

## Tableaux de correspondances entre types pivots et datatypes. . . . . 387

---

## Rapports fournis dans HOPEX Data Architecture. . . . . 419

Accéder aux rapports . . . . .	419
Rapports de description . . . . .	419
<i>Rapport de vue . . . . .</i>	<i>419</i>
<i>Rapport de glossaire . . . . .</i>	<i>419</i>
<i>Carte de domaines de données . . . . .</i>	<i>421</i>
<i>Graphe de réalisation . . . . .</i>	<i>422</i>
<i>Dépendances entre domaines de données . . . . .</i>	<i>422</i>
<i>Dendrogramme des catégories de données . . . . .</i>	<i>422</i>
Rapports nuages de mot . . . . .	423
<i>Nombre d'informations des domaines d'information . . . . .</i>	<i>423</i>
<i>Etendue de la description de l'information . . . . .</i>	<i>423</i>
<i>Utilisation des données dans les domaines de données . . . . .</i>	<i>423</i>
Rapports d'utilisation des données . . . . .	423
<i>Utilisation des informations détenues par un conteneur . . . . .</i>	<i>423</i>
<i>Utilisation des informations d'un domaine . . . . .</i>	<i>424</i>
<i>Utilisation des informations d'une carte d'informations . . . . .</i>	<i>424</i>
<i>Utilisation de l'information . . . . .</i>	<i>425</i>
<i>Utilisation des informations des domaines d'un conteneur . . . . .</i>	<i>425</i>
Report DataSets . . . . .	426
<i>Créer un Report DataSet . . . . .</i>	<i>426</i>
<i>Exemple de Report Dataset . . . . .</i>	<i>426</i>

---

## Workflow de validation des données. . . . . 429

<i>Etapes du workflow de validation . . . . .</i>	<i>429</i>
<i>Générer un rapport de workflow . . . . .</i>	<i>430</i>

---

## Import et export de données. . . . . 431

<b>Importer des données métier à partir d'un fichier Excel . . . . .</b>	<b>432</b>
Télécharger le modèle de fichier Excel . . . . .	432
Contenu du modèle Excel . . . . .	432
<i>FeuilleTerm</i> . . . . .	432
<i>Feuille Concept</i> . . . . .	433
<i>Feuille Synonym</i> . . . . .	434
<i>Feuille Component</i> . . . . .	435
<i>Feuille State Concept</i> . . . . .	435
<b>Importer des données logiques à partir d'un fichier Excel . . . . .</b>	<b>436</b>
Télécharger le modèle de fichier Excel . . . . .	436
Contenu du modèle Excel . . . . .	436
<i>Feuille Data Dictionary</i> . . . . .	436
<i>Feuille Data Type</i> . . . . .	437
<i>Feuille Data Type Component</i> . . . . .	438
<i>Feuille Class</i> . . . . .	438
<i>Feuille Attribute</i> . . . . .	439
<i>Feuille Relationship</i> . . . . .	440
<i>Feuille Generalization</i> . . . . .	440
<b>Importer des évaluations de données . . . . .</b>	<b>441</b>
Exemple d'import . . . . .	441
Contenu du modèle Excel . . . . .	442
Télécharger le modèle Excel . . . . .	442
Importer un fichier Excel d'évaluations de données . . . . .	442



# INTRODUCTION À HOPEX DATA ARCHITECTURE



**Hopex Data Architecture** 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 Architecture](#)
- ✓ [Se connecter à Hopex Data Architecture](#)
- ✓ [Le bureau de HOPEX Information Architecture](#)

# PÉRIMÈTRE DE LA SOLUTION HOPEX DATA ARCHITECTURE

---

## Trois niveaux de modélisation

La solution **Hopex Data Architecture** 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 œuvre. 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 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](#).

---

## 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](#).

---

## 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 Architecture** fournit des rapports types par défaut qui vous permettent de vérifier la qualité, l'utilisation et la conformité de vos données.

Voir [Rapports fournis dans Hopex Data Architecture](#).

# SE CONNECTER À HOPEX DATA ARCHITECTURE

Pour se connecter à **Hopex Data Architecture**, 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 Architecture** dépendent du profil avec lequel vous êtes connecté.

---

## Les profils de Hopex Data Architecture

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

Profil	Description
<b>Architecte de données</b> (Data Architect)	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.
<b>Administrateur fonctionnel de données</b> (Data 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 : <ul style="list-style-type: none"><li>• les dictionnaires,</li><li>• les domaines d'information,</li><li>• les concepts, vues de concept</li><li>• etc.</li></ul>

## Les rôles métier de Hopex Data Architecture

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

Rôle métier	Description
<b>Directeur de données (DCO)</b>	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.
<b>Propriétaire de données</b>	Le propriétaire de données est l'autorité qui décide de l'accès aux données et de leur utilisation. Il/elle peut être le concepteur des données, un de ses utilisateurs ou une tierce partie. Les coordinateurs de données 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.
<b>Concepteur de données</b>	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.
<b>Data engineer</b>	Le data engineer construit et maintient les outils et les infrastructures nécessaires à l'analyse des données par les scientifiques de données. 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.
<b>Scientifique de données</b>	Le scientifique de données est en charge du rapprochement entre le concepteur des données (métier et logiques) et les responsables des processus qui utilisent ces données.
<b>Responsable qualité des données</b>	Le responsable qualité 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.
<b>Coordinateur de données</b>	Le coordinateur de données (en 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.

Voir aussi [Responsabilité des données](#).

# LE BUREAU DE HOPEX INFORMATION ARCHITECTURE

---

## Page d'accueil de Hopex Data Architecture

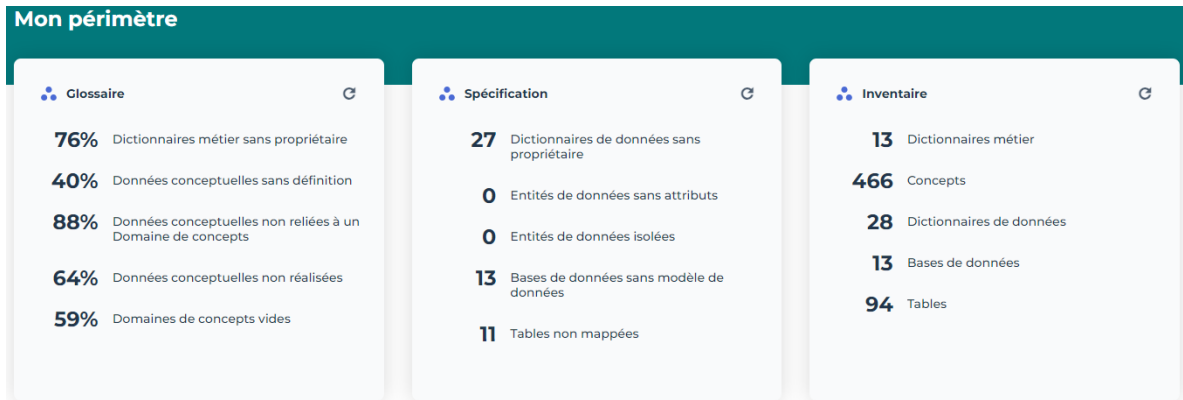
La page d'accueil de la solution **Hopex Data Architecture** se compose des sections suivantes.

- L'en-tête présente quelques informations d'intérêt général.
  - ☛ *Celles-ci peuvent être définies dans le menu **Administration** > **Domaines méthodologiques** 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 :

- Glossaire
- Spécification
- Inventaire



### Glossaire

Cette vignette recense les objets suivants :

- **Dictionnaires métier sans propriétaire** : affiche les dictionnaires métier sans conteneur détenteur.  
 La conteneur détenteur d'un dictionnaire métier peut être une bibliothèque, un projet AE, une entreprise ou un catalogue éditeur. Il est visible dans la page **Caractéristiques** de ses propriétés.
- **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.
- **Données conceptuelles non réalisées** : affiche les données métier qui ne sont pas associées à des éléments du SI.  
 Pour définir quels éléments du SI réalisent des concepts, voir [Relier les concepts métier à l'architecture logique et physique](#).

## Spécification

Cette vignette recense les objets suivants :

- **Dictionnaires de données sans propriétaire**
- **Entités de données sans attributs**
  - ☛ Pour le détail des éléments d'un dictionnaire de données, voir [Modéliser les dictionnaires de données](#).
- **Entités de données isolées** : liste les classes qui ne participent pas à une association, une relation orientée (partie) ou un héritage (généralisation) avec d'autres classes.
- **Bases de données sans modèles de données** : affiche les bases de données qui ne sont pas associées à un modèle logique
  - ☛ Vous pouvez relier un modèle de données à une base de données. Le modèle permet de représenter la structure de la base de données. Voir aussi : [Modéliser les bases de données](#).
- **Tables non mappées**
  - ☛ Voir [Synchroniser les modèles logiques et physiques](#).

## Inventaire

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

- Dictionnaires métier
- Concepts
- Dictionnaires de données
- Bases de données
- Tables

---

## Afficher l'environnement de travail d'une entreprise

Un référentiel peut être partitionné en *Entreprises*.

Une entreprise est un projet d'entreprise, ayant pour finalité la mise à disposition de biens et de services en adéquation avec la mission de l'entreprise et son environnement. L'entreprise établit des buts de l'entreprise 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 **Hopex Data Architecture** ; 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 des données.

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


1. Cliquez sur le menu de navigation puis sur **Environnement**.
2. Dans le volet de navigation cliquez sur **Navigation standard**.

3. Dans la zone d'édition cliquez sur la tuile **Entreprises**.
4. Cliquez sur **Nouveau**.
5. Dans l'assistant de création, indiquez le nom de l'entreprise.
6. Pour créer en même temps l'environnement de l'entreprise, sélectionnez le type d'environnement "Information Architecture".
7. 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 Architecture** :

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**.
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](#).

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



**Hopex Data Governance** et **Hopex Data Architecture** offrent une solution de gestion et de partage du vocabulaire spécifique à votre entreprise. Elles permettent de 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, elles 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.

## 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 fourni pour assurer le travail d'analyse et d'organisation des concepts 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 travail de réalisation des concepts est assuré par l'Architecte de données.

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.

Glossaire métier

Rechercher

Chercher dans

Objets principaux Tous les objets Catalogue de données Glossaire métier

☐ Tous les types d'objet sont pris en compte

☐ Concept

☐ Concept d'état

☐ Concept type

☐ Concept événement

☐ Propriété de concept

☐ Vue de concept

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.  
*Ensuite, la dernière liste sélectionnée devient la liste par défaut.*
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\*' as the selected term. The main content area is divided into two columns. The left column lists search results, including 'Client' (highlighted in yellow) with a 'Mis à jour Aujourd'hui' (Updated today) label and a comment: 'Commentaire : Une personne ou une organisation qui achète...'. Below this, another 'Client' entry is shown with a 'Mis à jour Aujourd'hui' label and a comment: 'Commentaire : Reverse Finance Data'. Further down, a 'Réservation' entry is visible with a comment: 'Commentaire : Action de réserver un forfait voyage compre...'. The right column shows a detailed view of the 'Client' term, with a blue header bar containing the title 'Client' and a close button. Below the header, there are two tabs: 'Vue globale' (Global view) and 'Avancés' (Advanced). The 'Vue globale' tab is active, showing a 'Responsables' (Responsible) section with a table of roles and their status. The 'Définition' (Definition) section is also visible, providing a description of the term.

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](#).

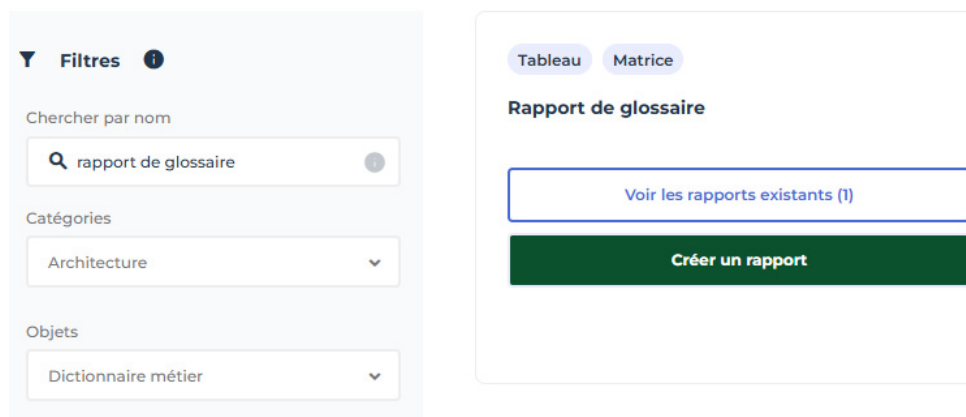
## GÉNÉRER UN GLOSSAIRE

**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](#)

---

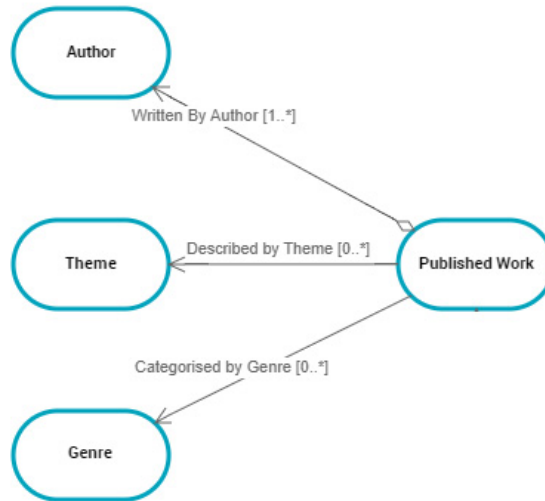
## Les liens entre les concepts

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.

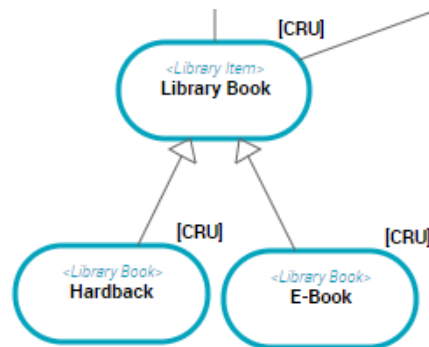
 Pour plus de détails, voir [Composants d'un concept](#).

## 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**.

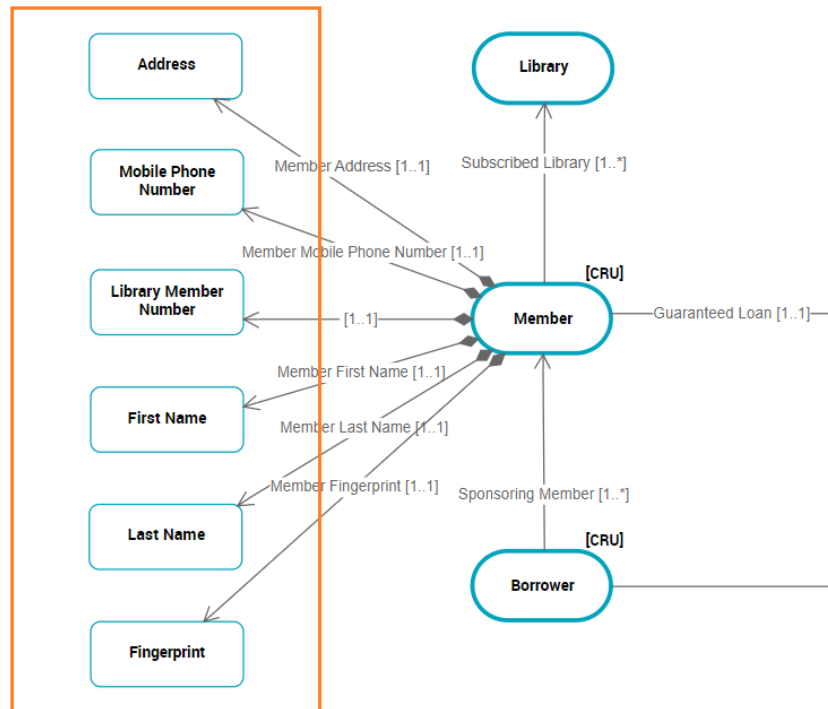
Par exemple, le "Souscripteur" est aussi un "Adhérent".

➡ 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.

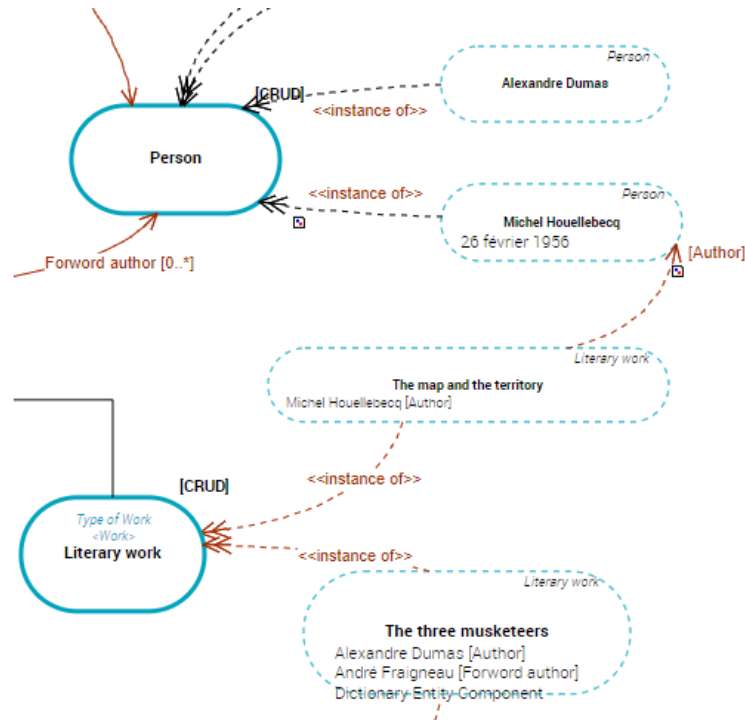
➡ Pour plus de détails, voir [Propriétés de concept](#).

## 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.*

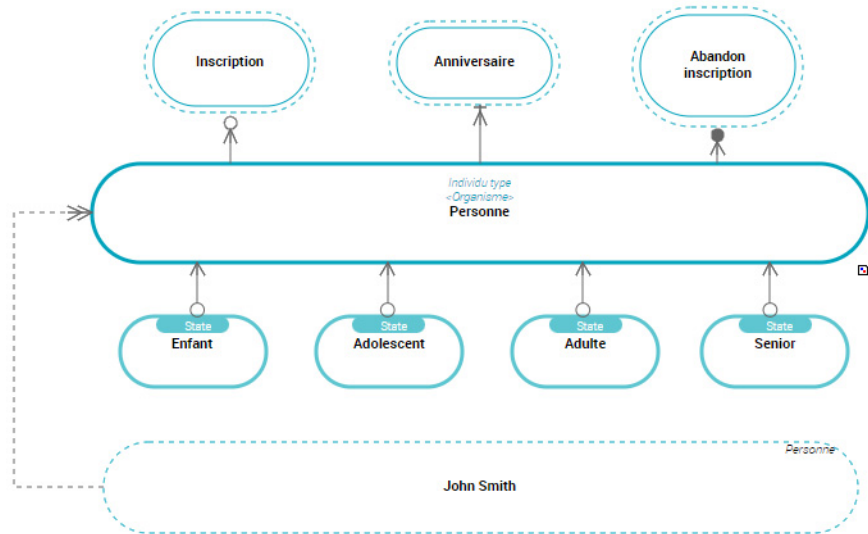
Les **Concepts d'état** et les **Concepts événement** sont des concepts à part entière qui peuvent être décrits comme n'importe quel concept.

## 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**.

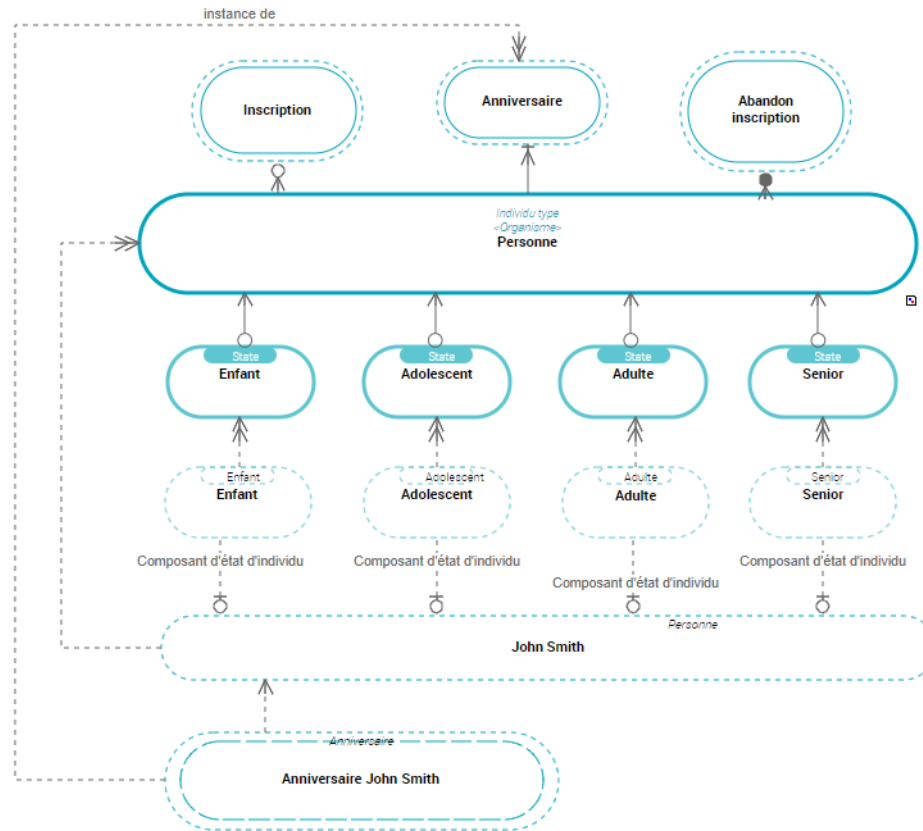
 Pour plus de détails, voir [Etats d'un concept ou d'un individu](#).

## 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**.

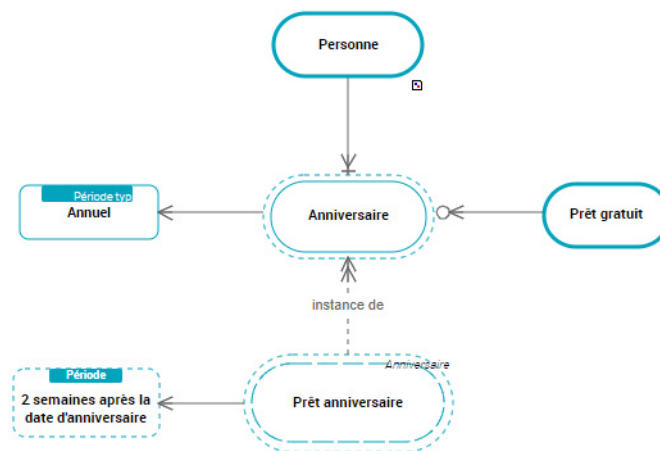


Un composant d'entité permet de relier un individu à un élément de dictionnaire.

## 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.

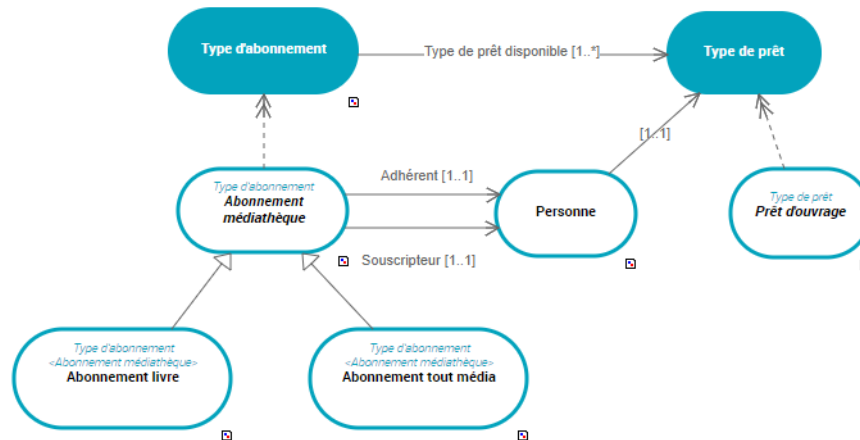


Pour plus de détails, voir [Utiliser les périodes](#).

## 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

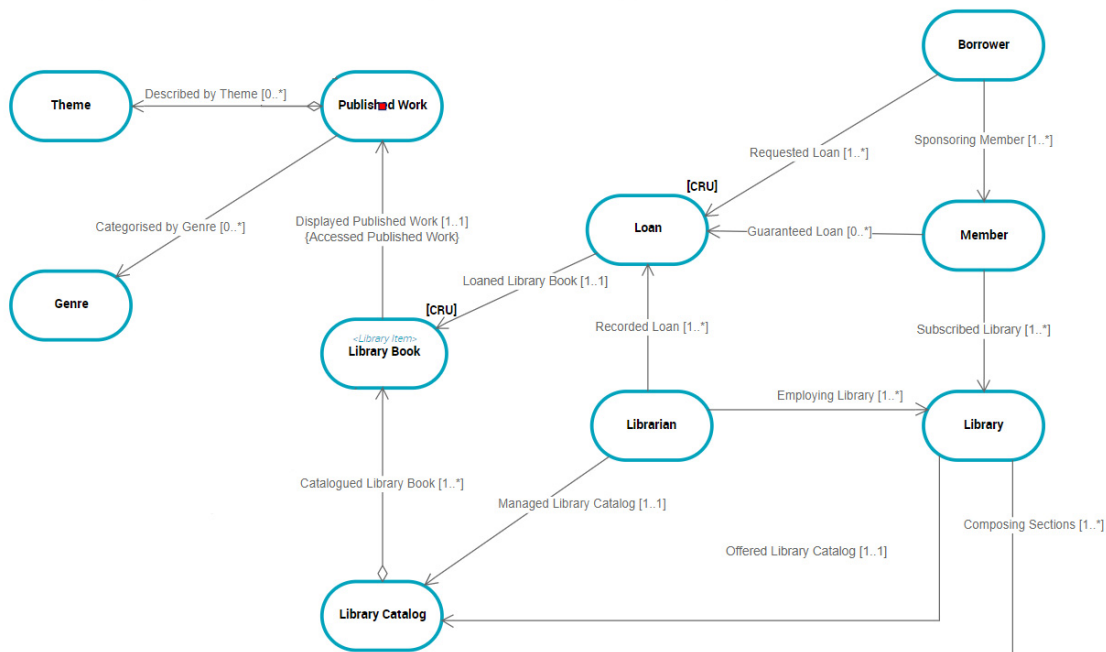
Pour la définition des données métier, **Hopex Data Governance** et **Hopex Data Architecture** fournissent différents types de diagramme.

## 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".

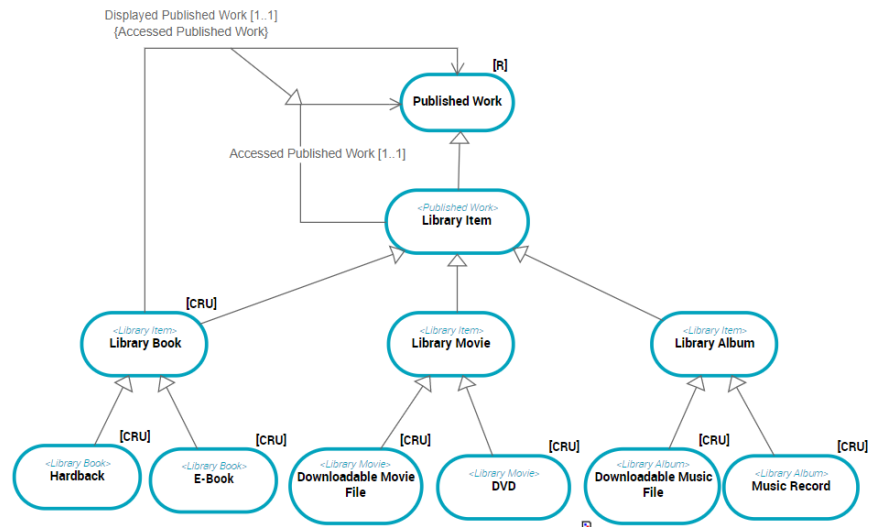


➤ Pour plus de détails, voir [Construire un diagramme de concepts](#).

---

## 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***

<b>Métadonnées physiques</b>	<b>Objets métier</b>
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**

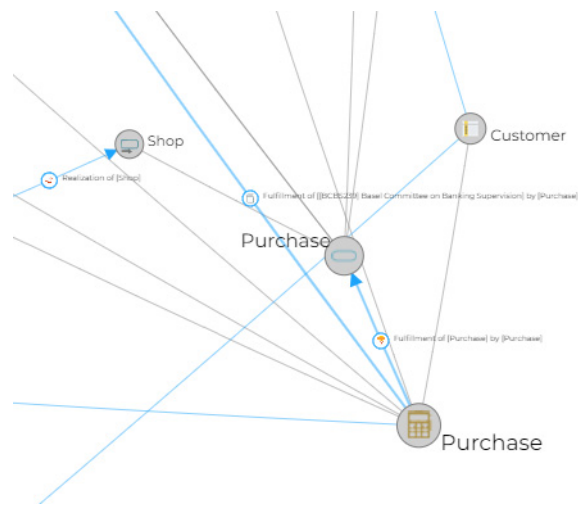
Il s'agit ici de définir les données en entrée du rapport.

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".

Notez que les réalisations des composants structurels des concepts passés en paramètre sont également affichées.



## 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).

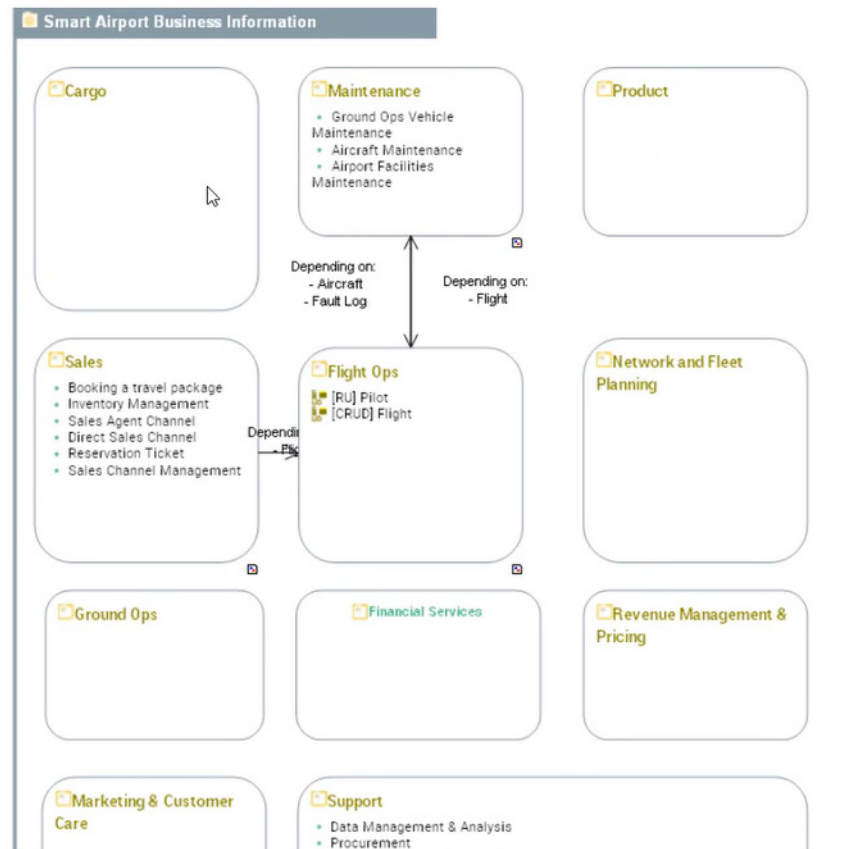
Les composants externes sont ceux qui sont utilisés dans la carte mais qui ne font pas partie du périmètre étudié.

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

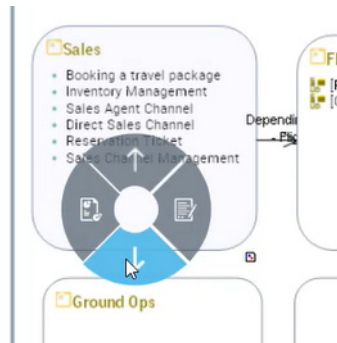
Ci-dessous la carte des domaines de concepts d'un aéroport. Des liens entre les domaines de concepts indiquent les dépendances qui existent entre des concepts de ces domaines.



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 :

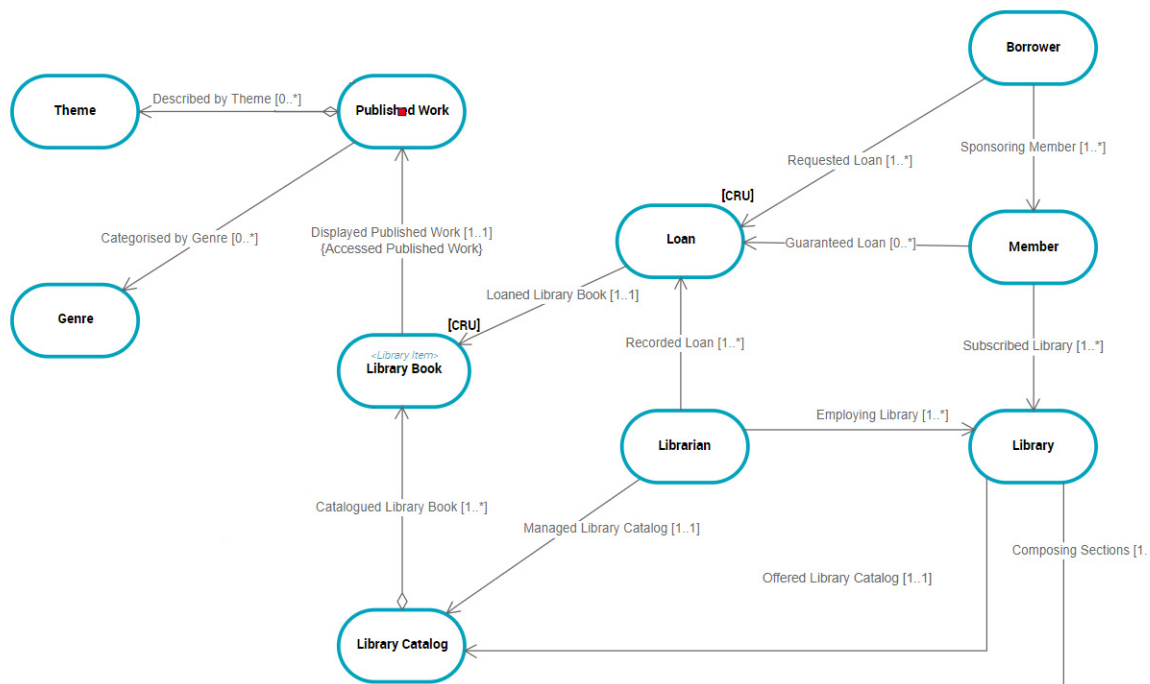
1. Dans la zone d'édition, passez la souris sur le domaine de concepts et cliquez sur le bouton **Créer un diagramme**.
2. Sélectionnez **Diagramme du domaine de concepts**.
3. Cliquez sur **OK**.

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.

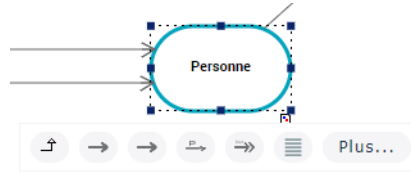
➡ Pour plus de détails sur la création de concept, voir [Concept](#)

### 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.

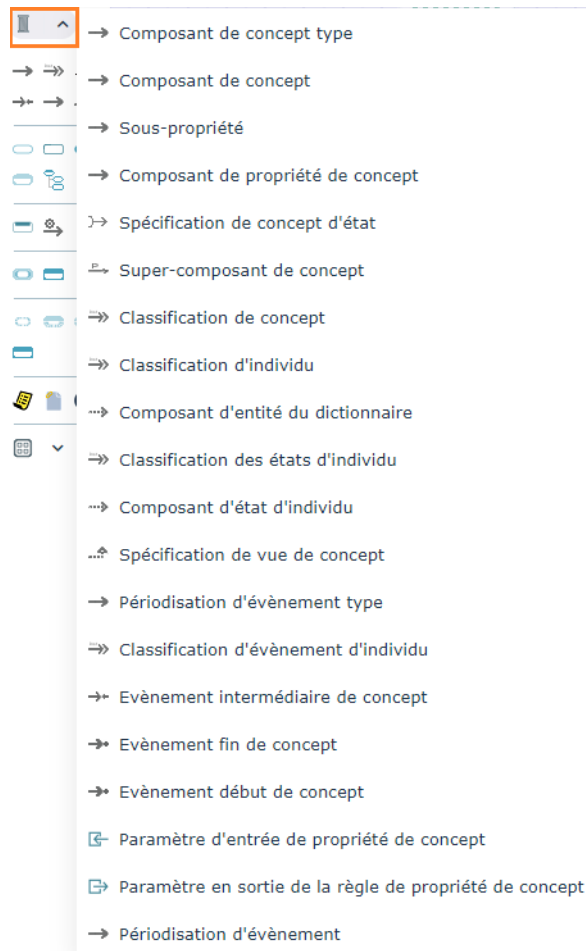
Par exemple : **Composant de concept**

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	

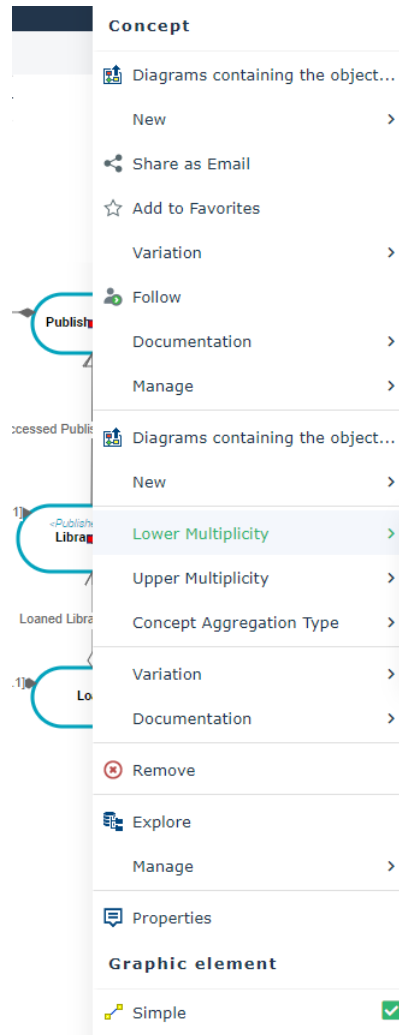
### Accéder aux propriétés d'un lien dans un diagramme de concepts

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		
<div> <div>Général</div> <div>Caractéristiques</div> <div>Entités du domaine</div> <div>Relations</div> </div>		
<div> <div>Nouveau</div> <div>Réordonner</div> <div>Hérité</div> <div>Exclu</div> </div>		
<input type="checkbox"/>	Nom Local ↑	Accès aux données
<input type="checkbox"/>	Body of work (EN)	CRUD
<input type="checkbox"/>	Cinematographic work (EN)	CRUD
<input type="checkbox"/>	Literary work (EN)	CRUD
<input checked="" type="checkbox"/>	Musical work (EN)	CRUD
<input type="checkbox"/>	Person (EN)	CRUD
<input type="checkbox"/>	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.

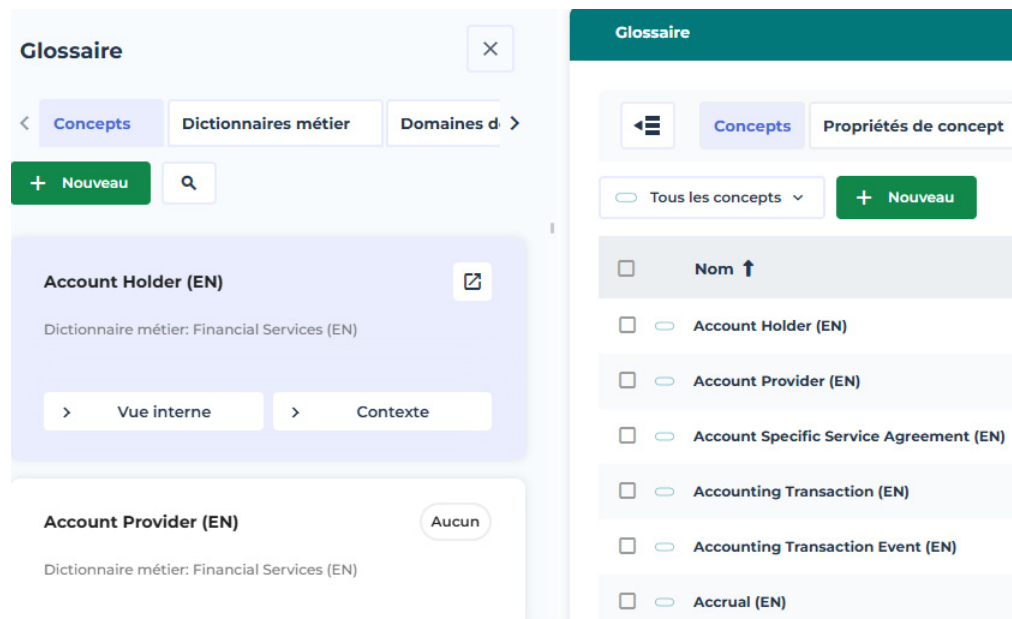
Un concept est associé à un ou des terme(s) qui désignent le concept dans une langue donnée. Voir [Consulter le glossaire métier](#).

## 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) section of the Hopex Data Governance application. 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 arrows for selection and navigation.


Nom ↑
<input type="checkbox"/> Account Holder (EN)
<input type="checkbox"/> Account Provider (EN)
<input type="checkbox"/> Account Specific Service Agreement (EN)
<input type="checkbox"/> Accounting Transaction (EN)
<input type="checkbox"/> Accounting Transaction Event (EN)
<input type="checkbox"/> 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

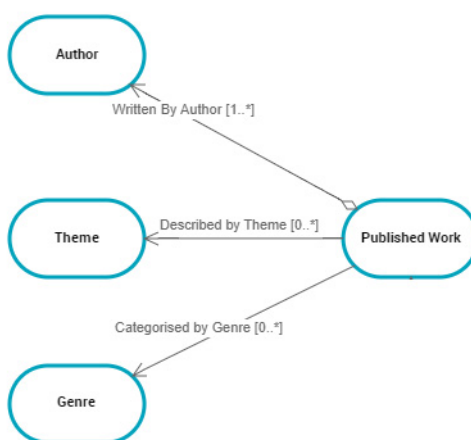
Il est possible de relier un concept à un autre concept pour le caractériser.

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](#).

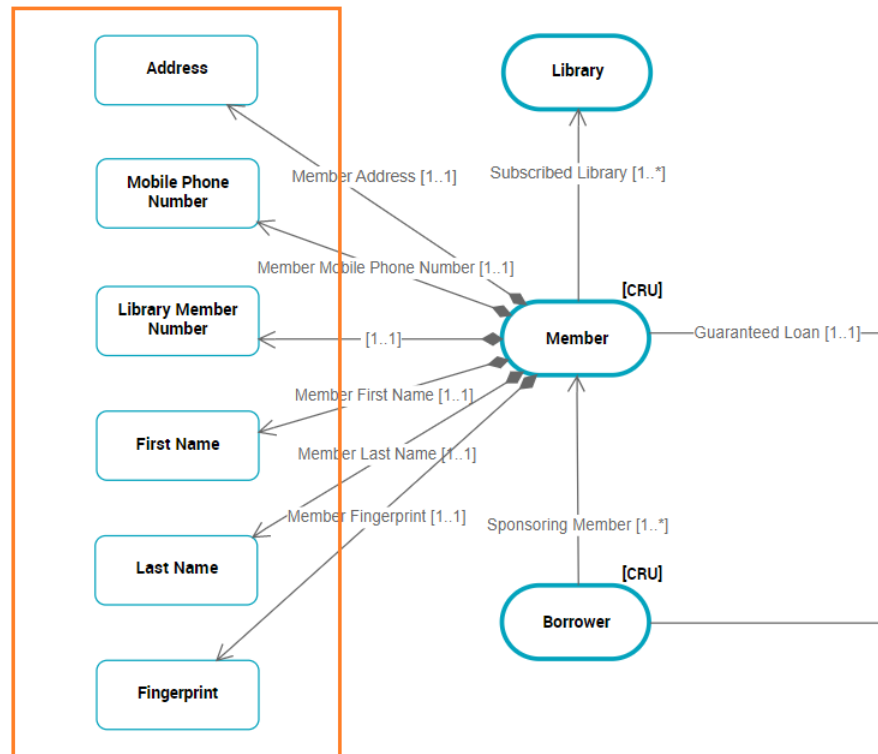
## PROPRIÉTÉS DE CONCEPT

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.*

---

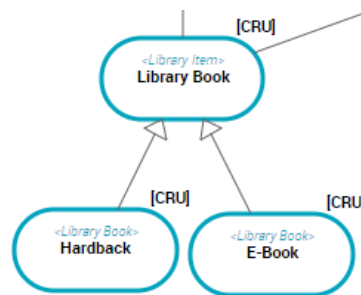
## Créer une propriété de concept calculée

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.

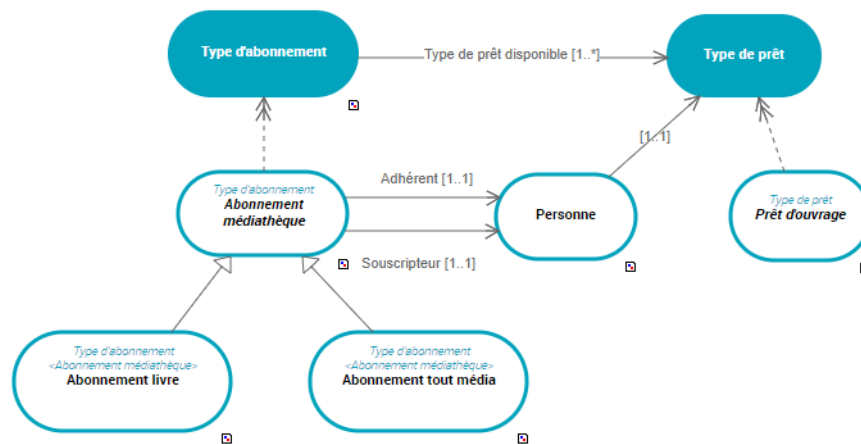
☛ La substitution est représentée par un lien mais il s'agit d'un objet **Hopex**.

## 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 :

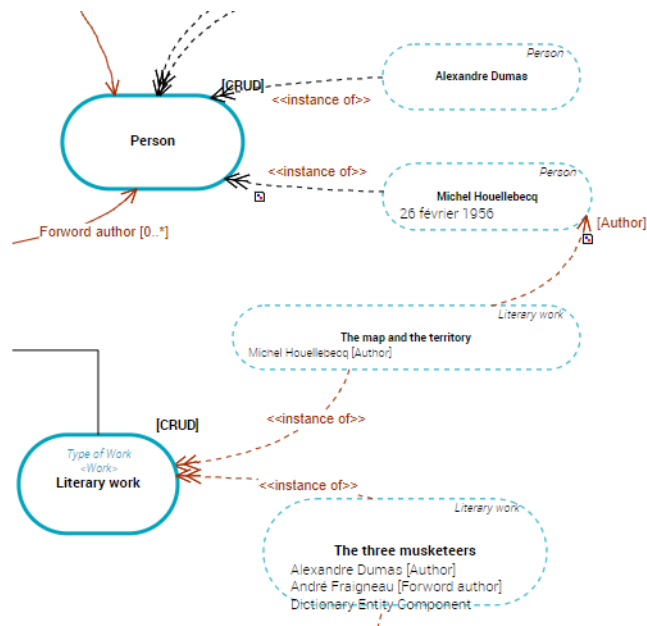
- Ses **Caractéristiques** principales avec son **Nom Local** et le dictionnaire détenteur.
- Les **Classifications**, qui indiquent quel concept caractérise l'individu.  
 Pour plus de détails, voir [Créer une classification d'individu](#).
- Les **Etats de l'individu** qui présentent les différents 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**.

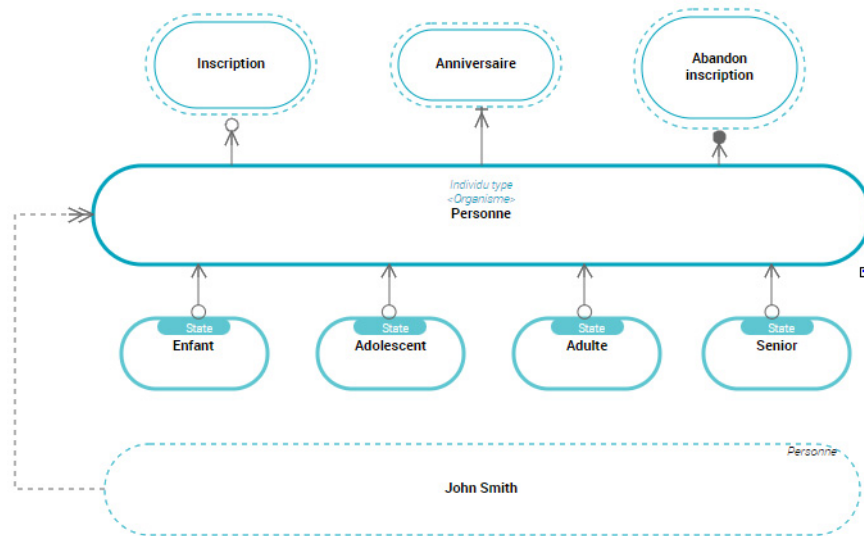


*Un concept événement représente un fait se produisant durant la vie d'un concept, par exemple un changement de saison. Un concept événement permet de marquer l'impact sur un concept d'un phénomène interne ou externe au concept. On peut distinguer les événements de*

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

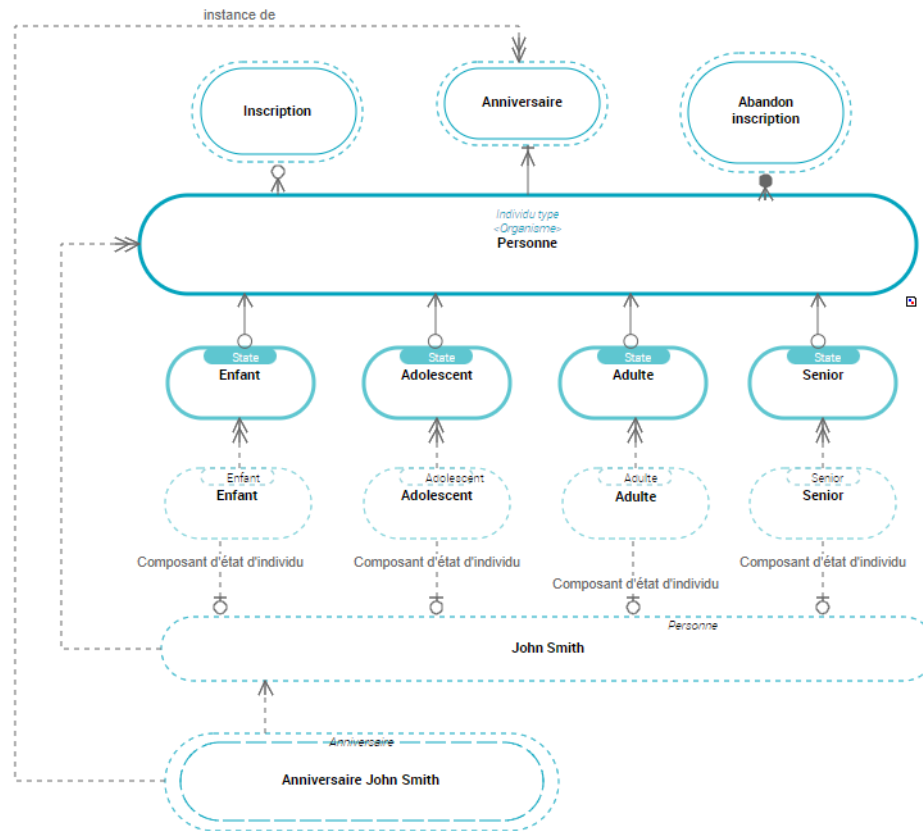
Si un concept est associé à des états, les occurrences de ce concept peuvent également être associées à des états. Ainsi, **Hopex Data Governance** et **Hopex Data Architecture** propose l'**Etat d'individu**.

📖 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.

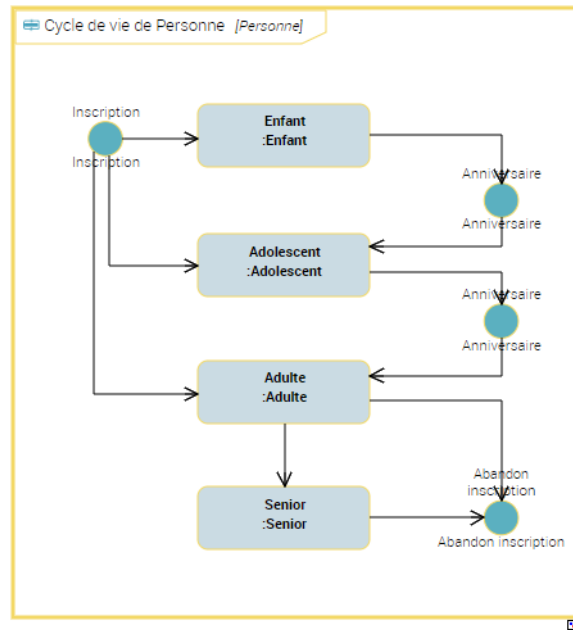
---

## 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.

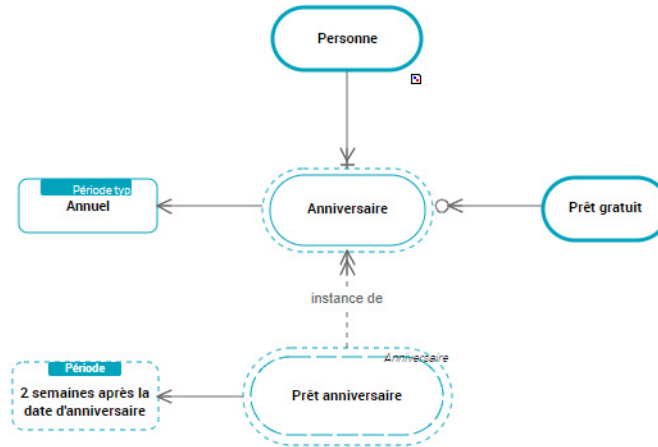
Une **Période type** permet de préciser un **concept événement**.



*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

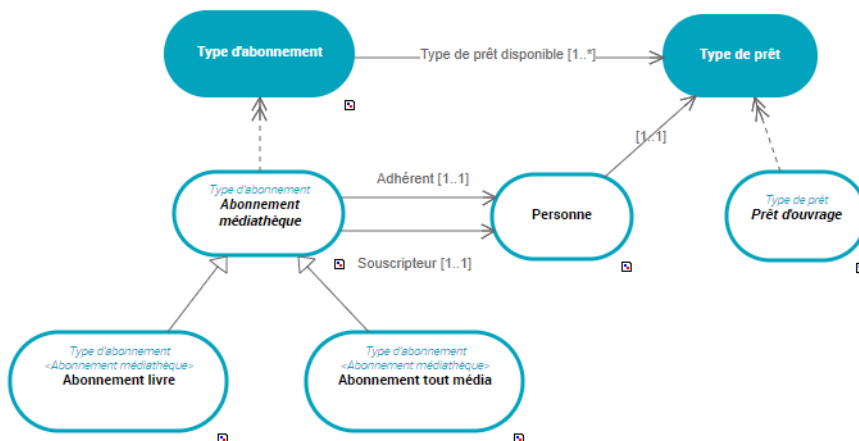
Il est possible de relier un concept type à un autre concept type pour le caractériser.

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 :

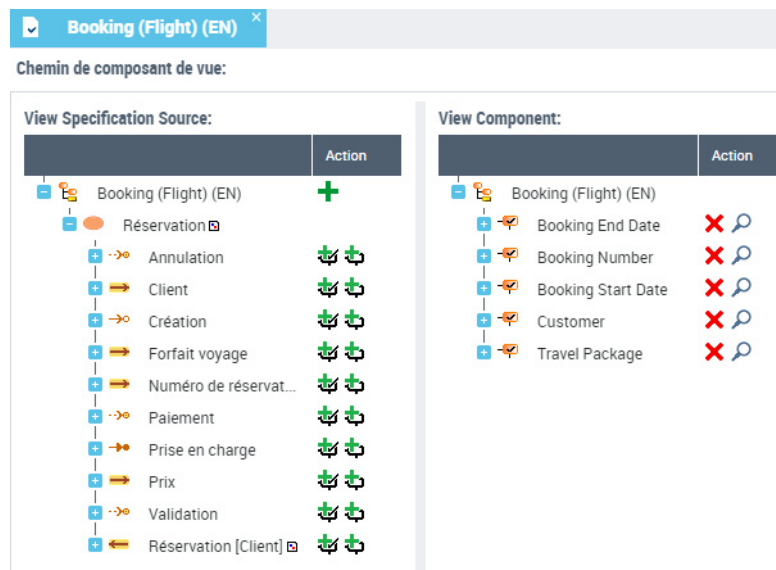
1. Faites un clic droit sur le nom du concept et sélectionnez **Nouveau > Vue de concept**.  
L'assistant de création d'une vue de concept apparaît.
2. Saisissez son **Nom**.
3. Le champ **Termes existants** dresse la liste des termes portant déjà le nom de la vue.  
 *Un terme est un mot ou groupe de 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.



### 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.

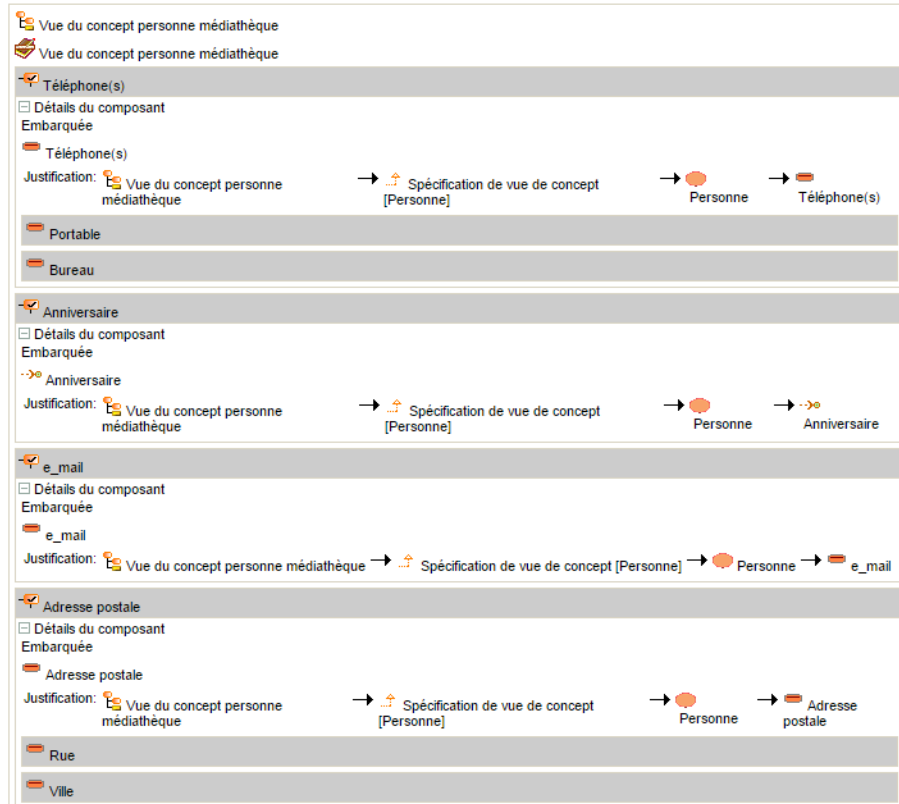
### Paramètres du rapport

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

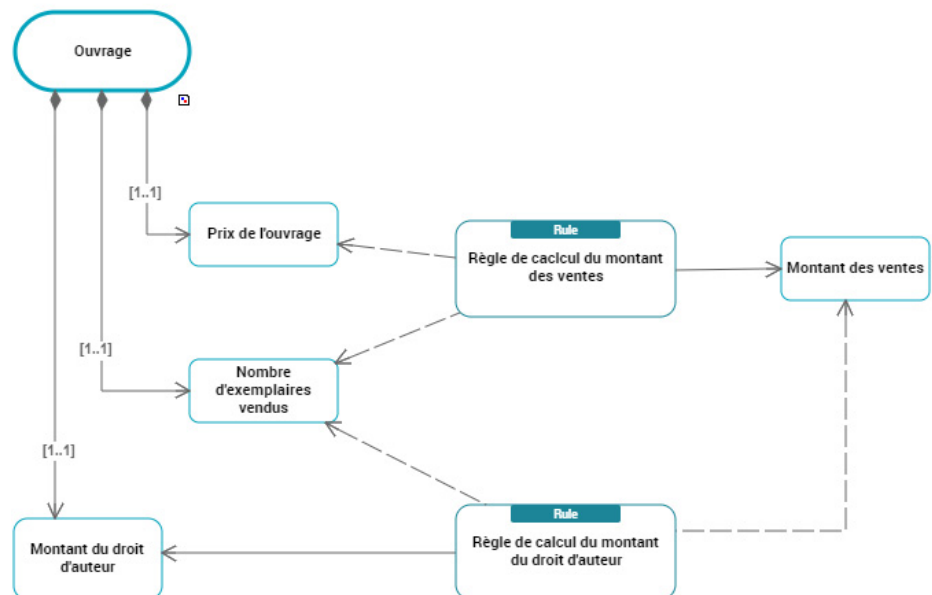
La règle de calcul prend en entrée des concepts ou propriétés de concept et en sortie la propriété de concept calculée.

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.



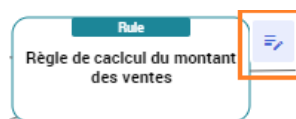
Pour définir la propriété de concept cible :

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 définis dans **Hopex Data Architecture** 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 puis sur **Glossaire > Données métier**.
2. Dans la zone d'édition, cliquez sur l'onglet **Concepts**.

 Dans le bureau **Information Architecture**, cliquez sur **Glossaire > Information métier** et recherchez sous le dictionnaire le concept voulu.

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.

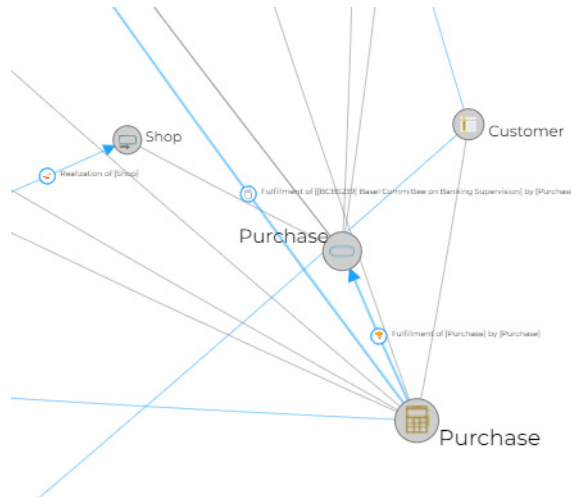
Pour accéder à ce rapport :

1. Ouvrez la fenêtre des propriétés du concept.
2. Cliquez sur la liste déroulante puis sur **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. Dans le bureau Information Architecture, cliquez sur le menu de navigation puis sur **Architecture des données > Données logiques**.
2. Dans la zone d'édition, cliquez sur **Dictionnaires de données** puis sur **Vue hiérarchique**.  
La liste des paquetages du référentiel apparaît dans la fenêtre d'édition.
3. Dépliez le dossier du paquetage qui vous intéresse.
4. Sélectionnez la classe que vous voulez relier à un concept et ouvrez sa fenêtre de propriétés.
5. Cliquez sur la page **Caractéristiques > Réalisation**.
6. Sous l'onglet **Objets réalisés**, cliquez sur **Nouveau**.  
La fenêtre **Ajout d'une réalisation détenue** apparaît.
7. Dans le champ **Type d'objet**, sélectionnez "Réalisation d'information métier" et cliquez sur **Suivant**.
8. Dans le champ **MetaClasse** sélectionnez "Concept".
9. Dans le champ **Information métier réalisée**, sélectionnez le concept qui vous intéresse.

10. 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**

### **Réalisation de données métier par des données logiques**

A travers cette matrice vous pouvez indiquer que des données logiques (classes, vue de données, etc.) réalisent des informations métier (concepts, types de concept, etc.).

---

## **Les différents niveaux de réalisation**

### ***Niveau métier***

#### **Réalisation de données métier par d'autres données métier**

Cette matrice permet d'indiquer que des données métier (concepts, types de concept, etc.) réalisent d'autres informations métier.

### ***Niveau logique***

#### **Réalisation de données logiques par d'autres données logiques**

#### **Réalisation de données métier par des données logiques**

#### **Réalisation de cartes d'informations métier par des cartes de données logiques**

#### **Réalisation de domaines d'informations métier par des domaines de données logiques**

#### **Réalisation de domaines de données logiques par des domaines de données applicatifs**

### ***Niveau physique***

#### **Réalisation de données métier par des données physiques**

#### **Réalisation de domaines d'informations métier par des domaines de données physiques**

#### **Réalisation de cartes d'informations métier par des cartes de données logiques**

#### **Réalisation de domaines de données logiques et applicatifs par des domaines de données physiques**





# **Conception des données et des bases de données**





# 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

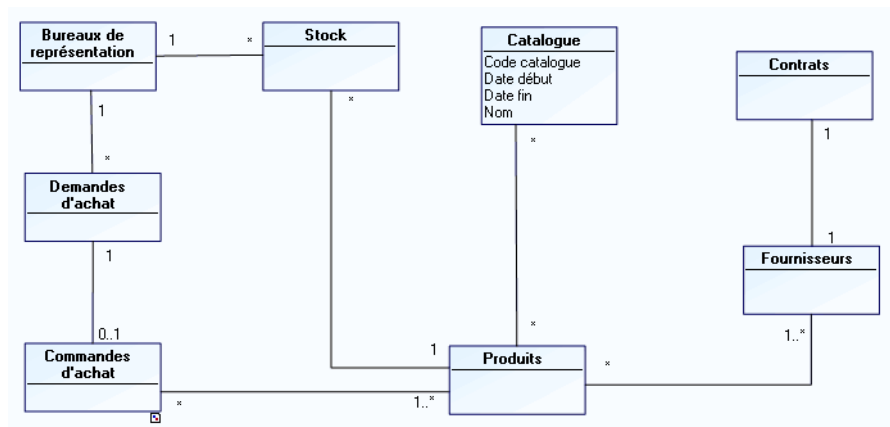
Lorsque vous faites le choix de travailler avec le formalisme "Modèle de données", les dictionnaires métier sont représentés par des modèles de données (et non des dictionnaires de données).

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

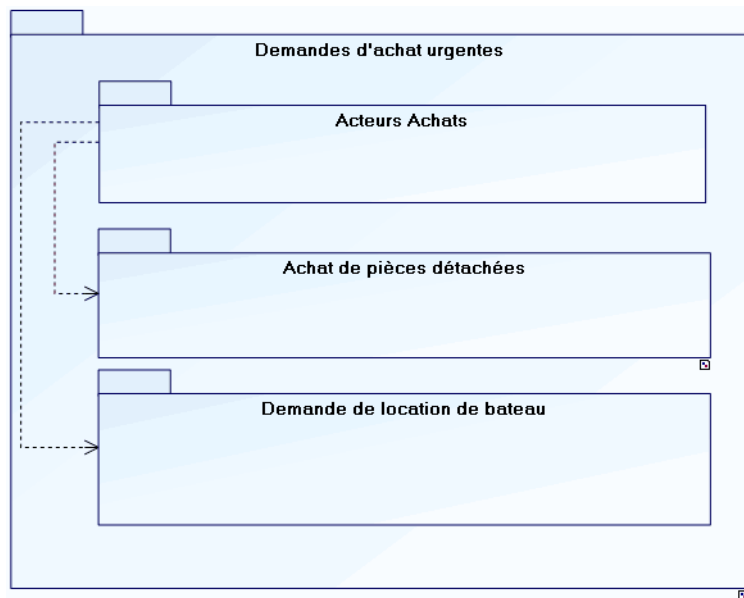
Un dictionnaire de données rassemble et détient un ensemble de données logiques. Il peut être découpé en domaines de données logiques. Voir [Domaine de données logiques](#).

## 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

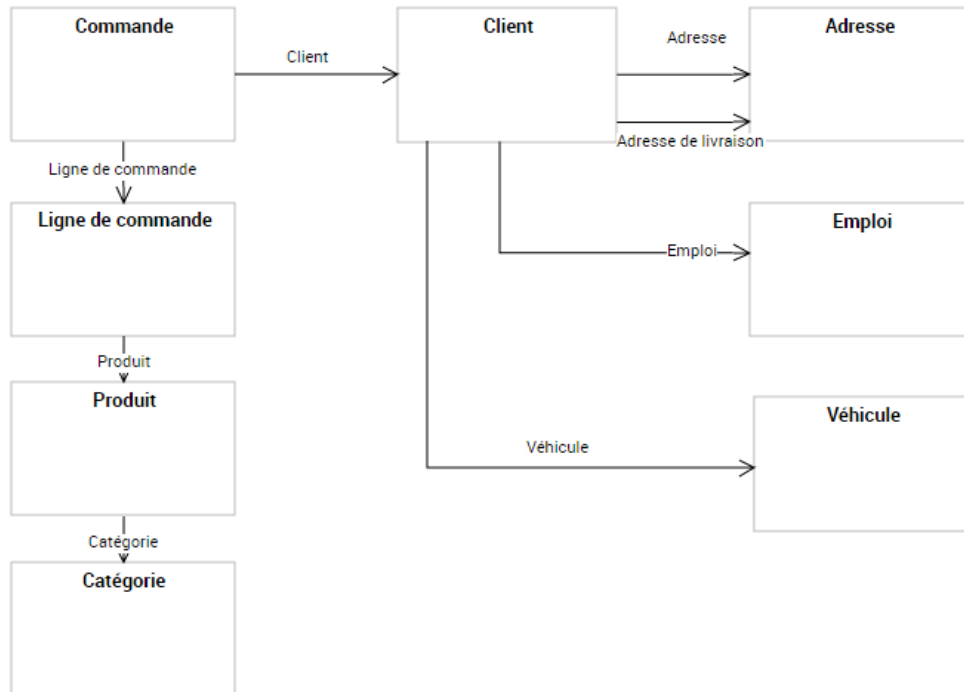
Les domaines de données / domaines de données logiques peuvent être décrits par un diagramme.

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
Vue de données Customer	
Customer (EN)	
Customer Id (EN)	
Customer Name (EN)	
Address (EN)	
Client [Account (EN)]	
Customer (EN) [Order (EN)]	

View Component:

	Action
Vue de données Customer	
Address (EN)	

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.

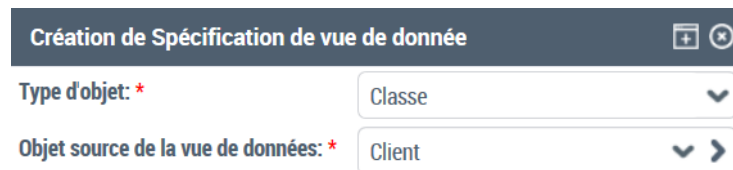
## Créer une vue de données (à partir de la liste des vues)

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**



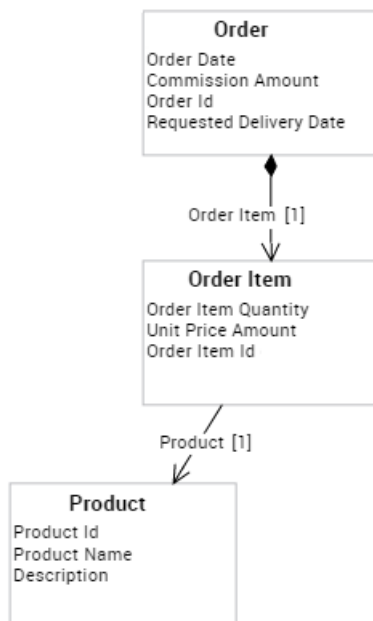
**Partie (classe  
composante)**



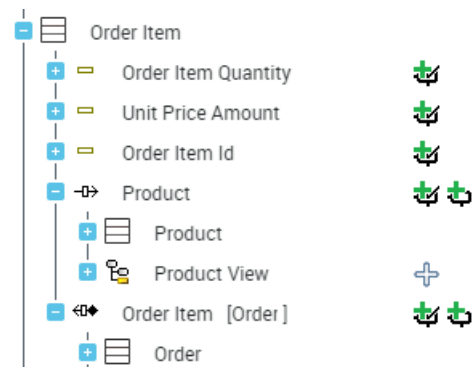
**Partie (classe  
composée)**

### Exemple

#### Modèle logique



#### Vue de données logiques




## Définir les composants de la vue de données

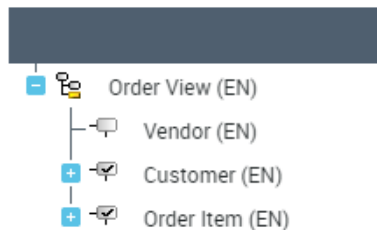
### Composant embarqué

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.

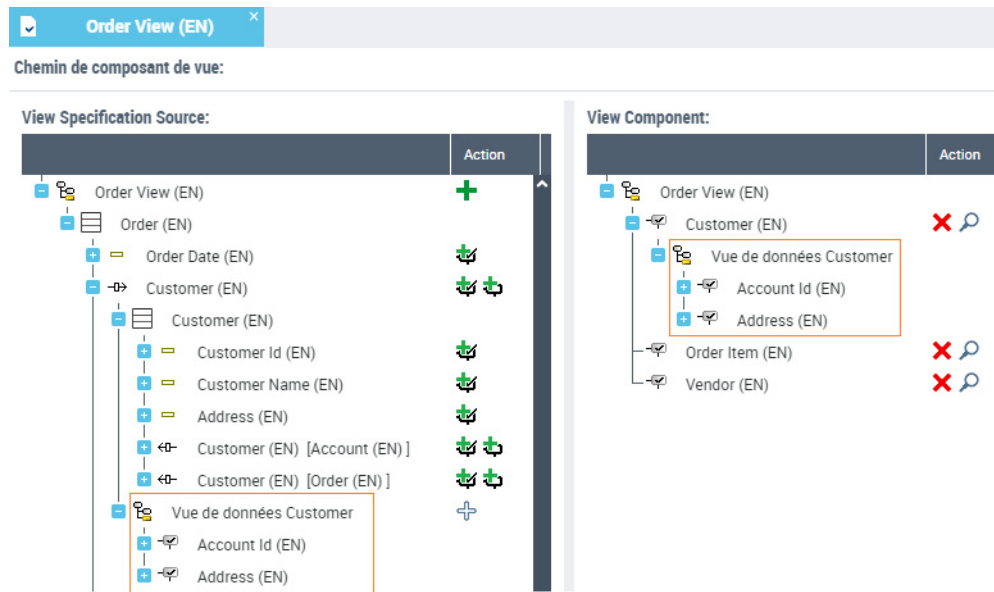
Pour référencer un objet dans la vue :

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.

---

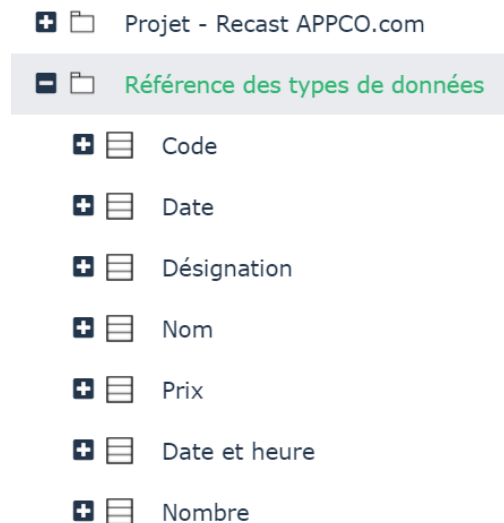
## Paquetage de types de données

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.

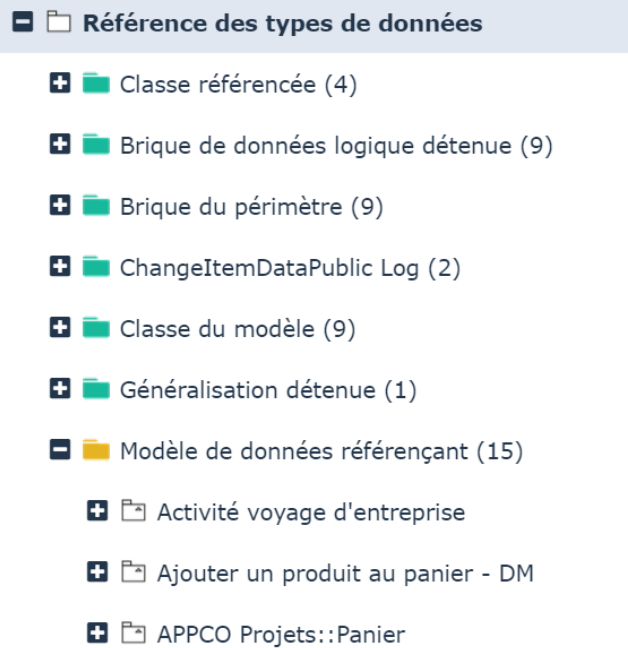
Pour typer les attributs d'une entité, ne sont proposés que les types de données définis pour le modèle de données qui contient cette entité.

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 packaging 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 *packaging de types de données*.

Le packaging 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 [Packaging de types de données](#).

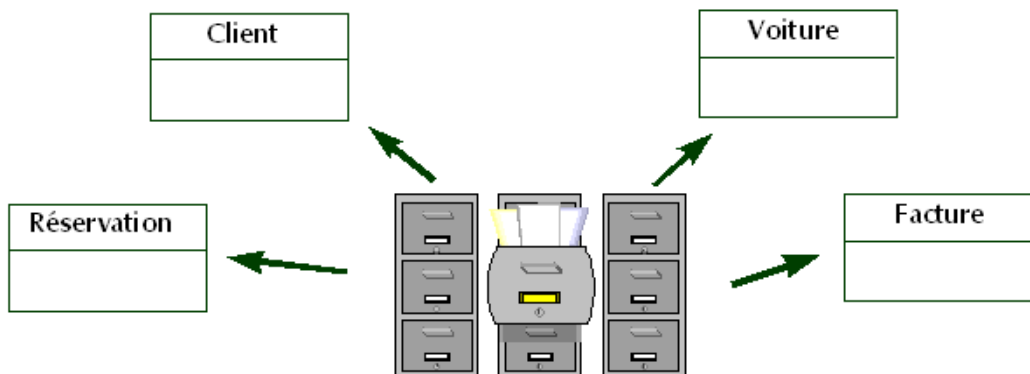
## Les entités

 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.

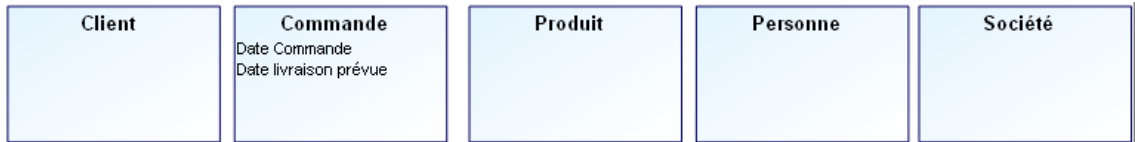
Une entité est reliée à d'autres entités via des associations. L'ensemble des entités et des associations constitue le noyau d'un diagramme de données.

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

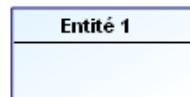
## Créer une entité

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.

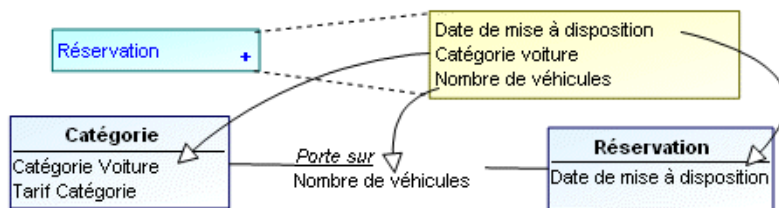
4. Cliquez sur **Ajouter**.  
L'entité apparaît dans le diagramme.



😊 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.

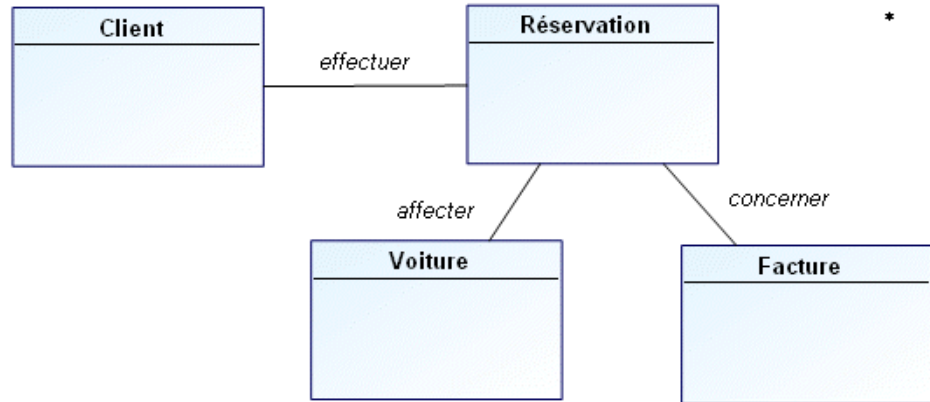
Voir [Généralisations](#).

---

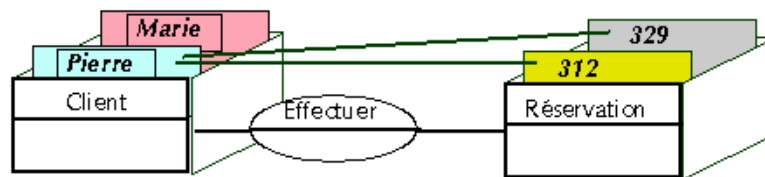
## Les associations

 *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.

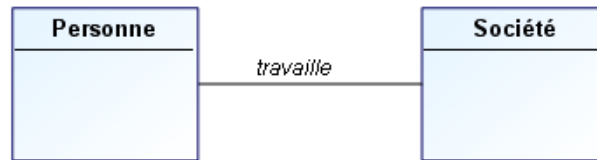
☛ *Le diagramme ne doit pas permettre de représenter des situations irréalistes ou aberrantes.*

Exemples d'association :

- Un client passe une commande.
- Une commande comprend plusieurs produits.




- Une personne travaille pour une société.



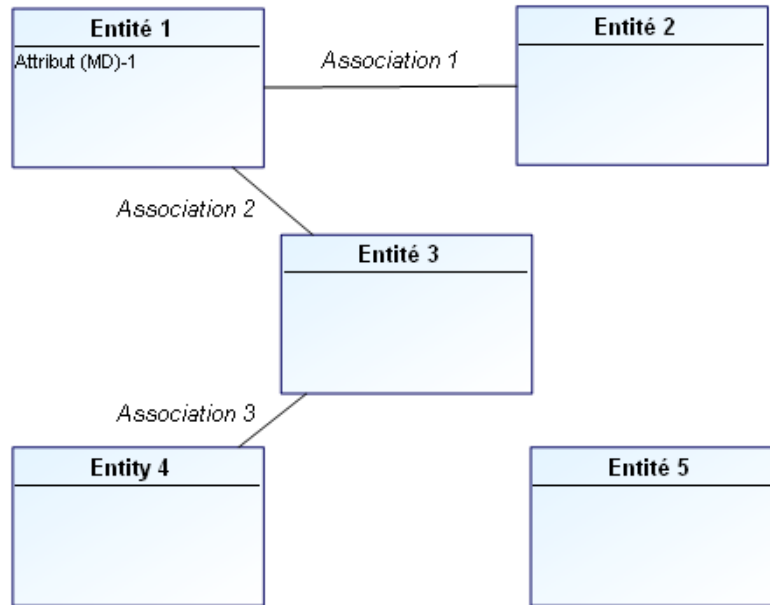
- Une alarme est déclenchée par un capteur.
- Un capteur couvre une zone.
- Une fenêtre affiche une chaîne de caractères.

## 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.

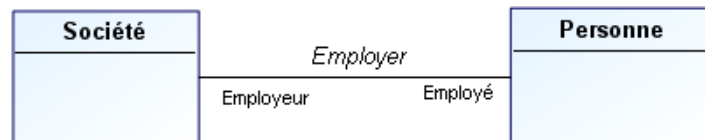
### Définir le rôle des associations



*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.

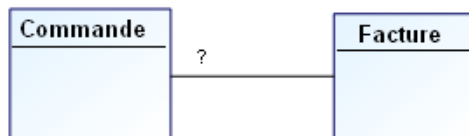
La multiplicité exprime le nombre de participations minimum et maximum d'un objet donné d'une entité à une association.

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

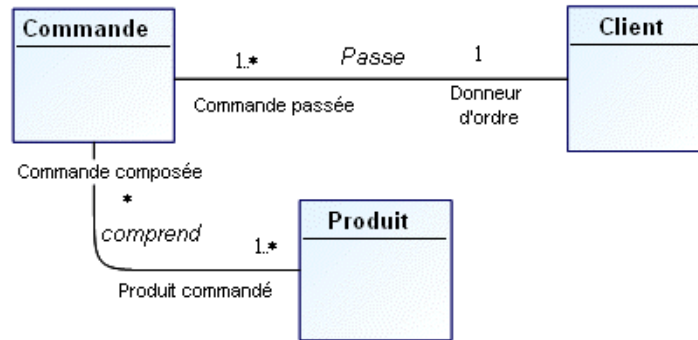
Exemple :



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 :

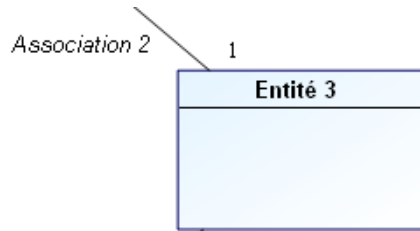


- 1..\* Un client peut passer une ou plusieurs commandes.
- 1 Une commande est passée par un et un seul client.
- 1..\* Une commande comprend un ou plusieurs produits.
- \* Un produit peut faire partie de plusieurs commandes, ou d'aucune.
- 0..1 Une personne travaille pour une société.
- 1..\* Une alarme 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.

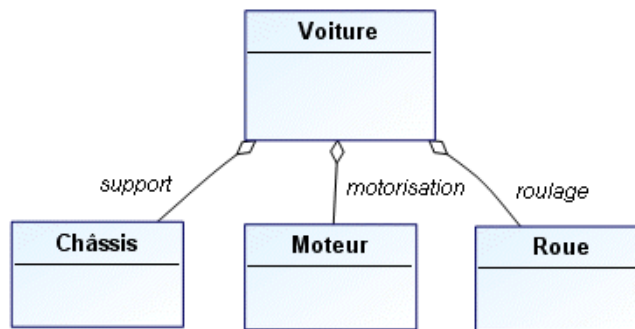


## Agrégation

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.

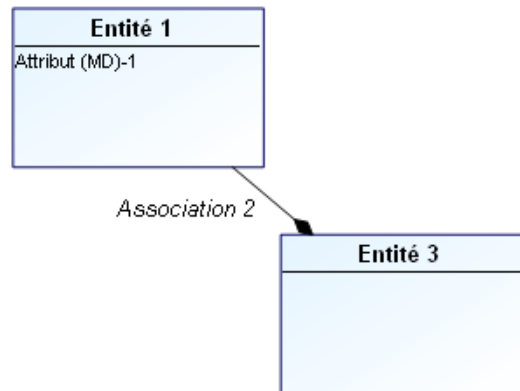
## Composition

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.



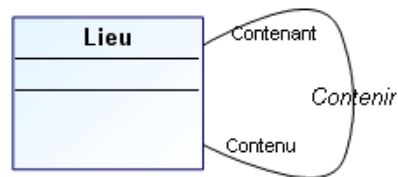
Pour préciser la composition d'un rôle :

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".

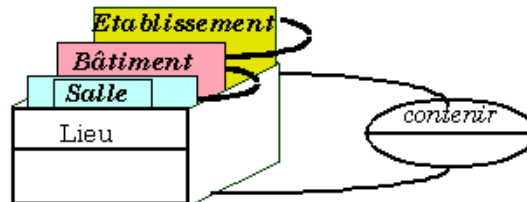
---

## 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.



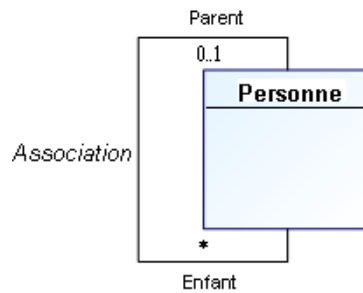
Une salle de classe est contenue dans un bâtiment, lui-même contenu dans un établissement scolaire.

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.

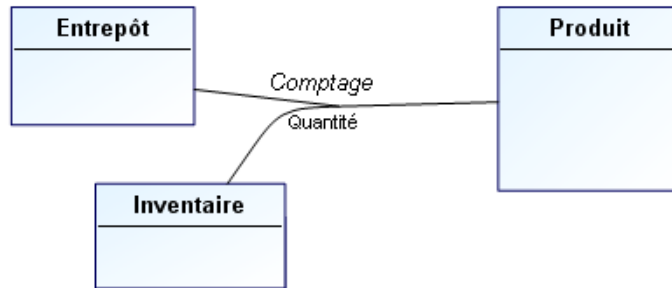
📖 Un rôle permet d'indiquer une des entités concernées par l'association. L'indication des rôles est particulièrement importante dans le cas d'une association entre une entité et elle-même.

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.

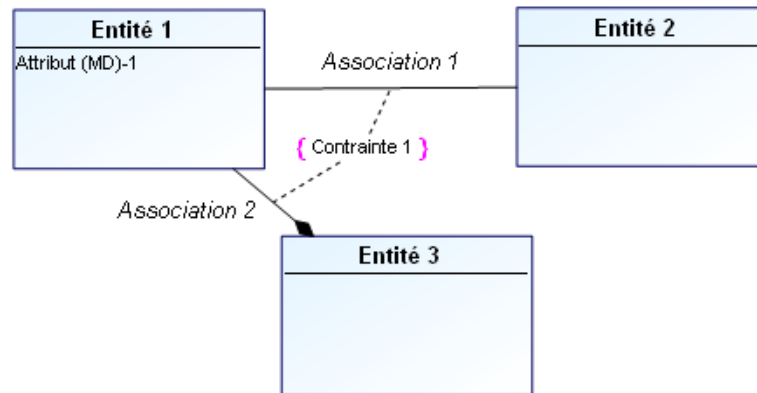
La date de livraison doit être postérieure à la date de commande.


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.



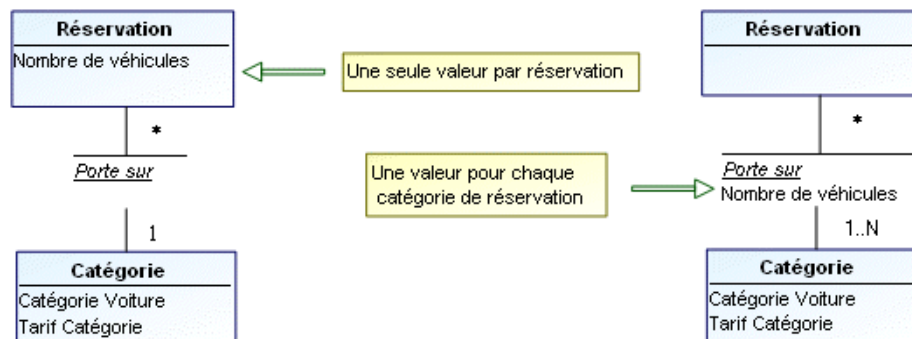
😊 Enregistrez régulièrement votre dessin à l'aide du bouton **Enregistrer** 

## 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.

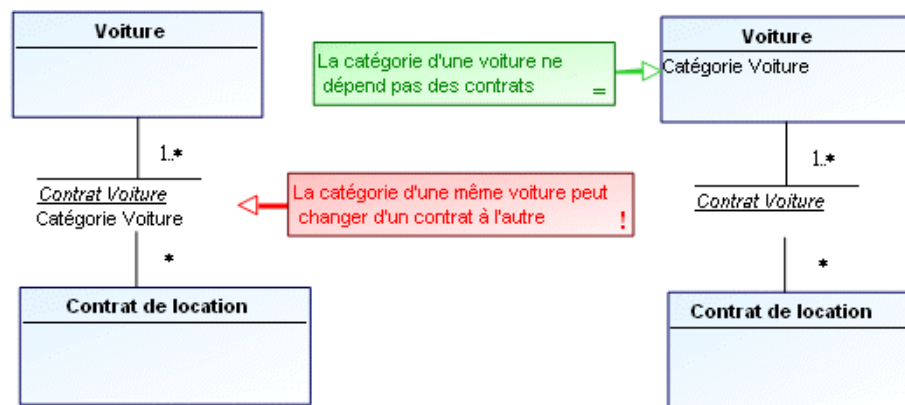


Si le nombre de véhicules est porté par l'entité "Réservation", on ne peut indiquer que le nombre total de véhicules pour une réservation. On doit donc faire une réservation par catégorie de véhicule loué (multiplicité 1).

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.

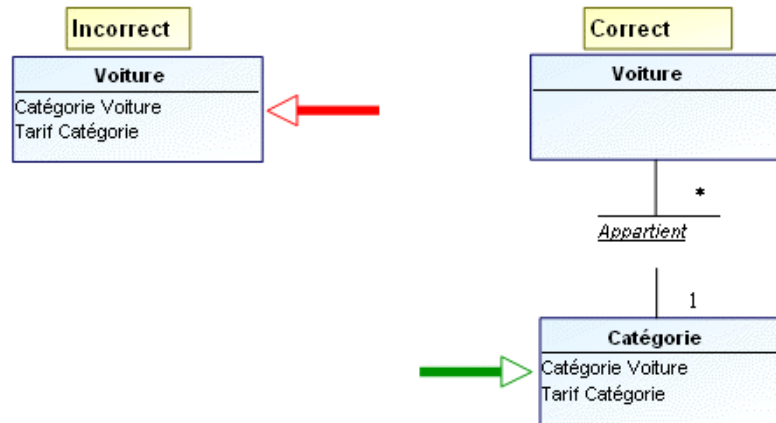


Le fait que la catégorie de voiture porte sur l'association "Catégorie Voiture" suppose que la catégorie de la voiture puisse changer d'un contrat à l'autre, ce qui ne serait pas très honnête.

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.

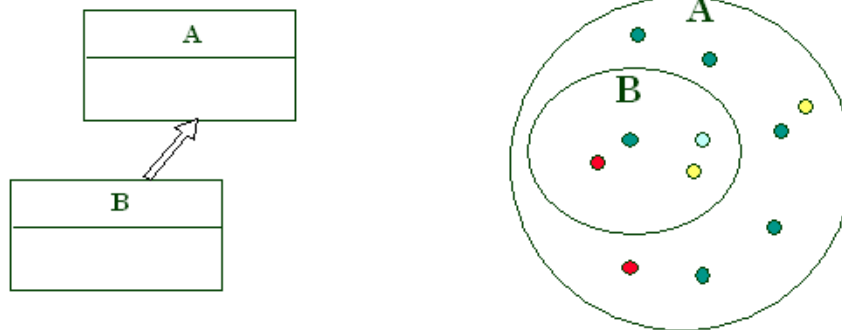
Pour éviter cela, il faut créer une entité "Catégorie" qui portera le tarif.

☺ 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.

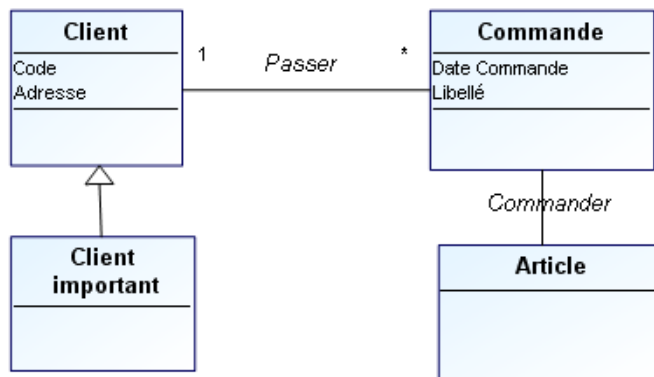
B étant un sous-ensemble de A, les objets de l'entité B "héritent" des caractéristiques de ceux de l'entité A.

Il n'est donc pas nécessaire de décrire de nouveau pour l'entité B :

- Ses attributs
- Ses associations

Exemple :

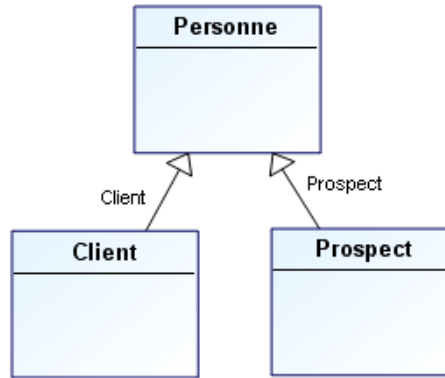
L'entité "Client important" qui représente les clients dont le "C.A. sur les 12 derniers mois" dépasse 1 million d'euros, peut être une spécialisation de l'entité "Client".



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".

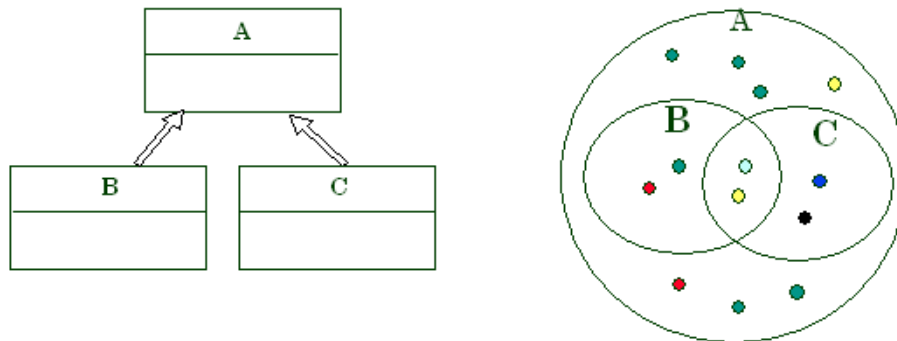
"polygone", "ellipse" et "cercle" sont des sous-entités de l'entité "forme".

"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".

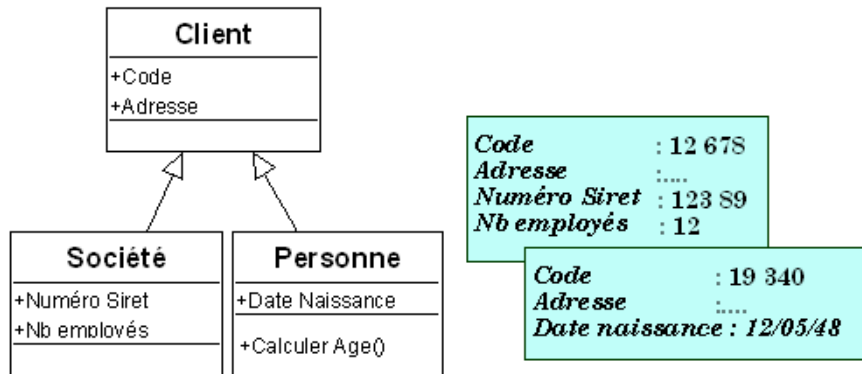
## Cas de plusieurs sous-entités



Plusieurs sous-entités d'une même entité :

- Ne sont pas forcément exclusives.
- Ne forment pas nécessairement une partition.

### Intérêt des sous-entités

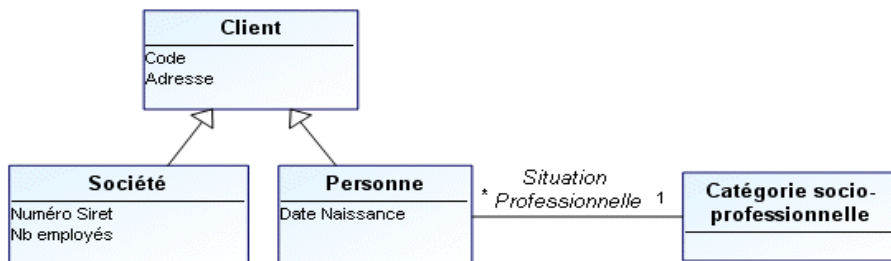


Une sous-entité hérite de tous les attributs et associations de sa super-entité, mais elle peut avoir des attributs ou des associations que ne possède pas sa super-entité.

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).

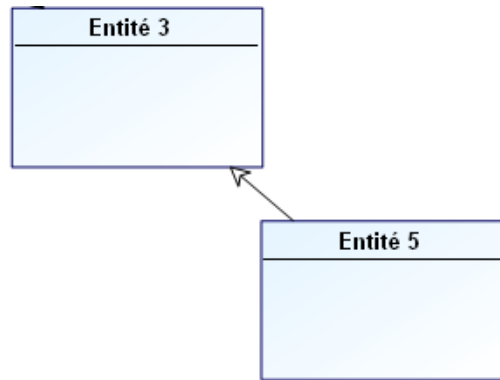
## Héritage multiple

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* :

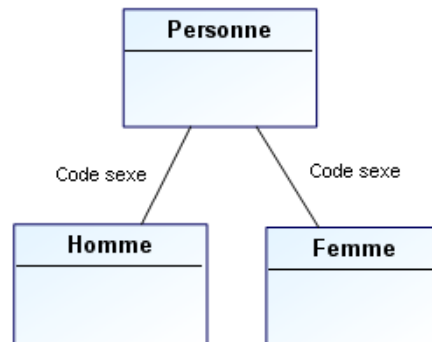
1. Dans la barre d'insertion du diagramme de données, cliquez sur le bouton **Généralisation** 
2. Cliquez dans la sous-entité, par exemple "Entité 5", et faites glisser la souris jusqu'à l'entité générale, par exemple "Entité 3", avant de relâcher votre pression.  
La généralisation apparaît dans le diagramme, matérialisée par une flèche.



## Discriminant

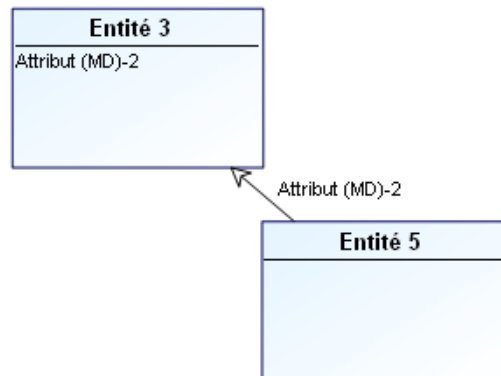
Le discriminant est l'attribut de l'entité générale dont la valeur permet de répartir les objets entre les sous-entités associées à la généralisation.

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.

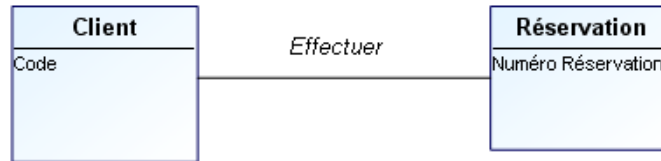
---

## 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	✓	✗
<b>Catalog</b>	✓	
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, déployez 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 :

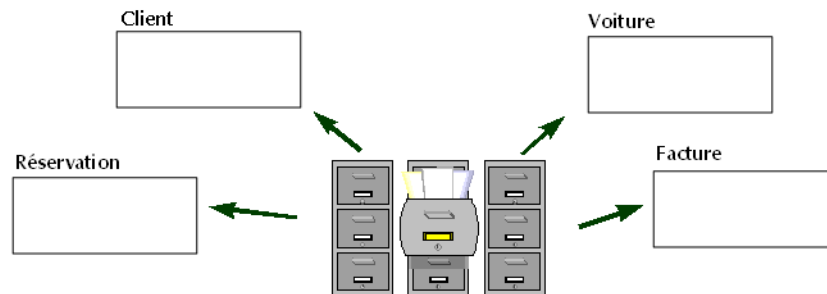
1. Déplacez la souris sur le modèle de données et cliquez sur le bouton **Créer un diagramme**  qui apparaît à droite.
2. Sélectionnez le type de diagramme **Diagramme de données (IDEF1X)**.  
Le diagramme apparaît.

---

## 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**.

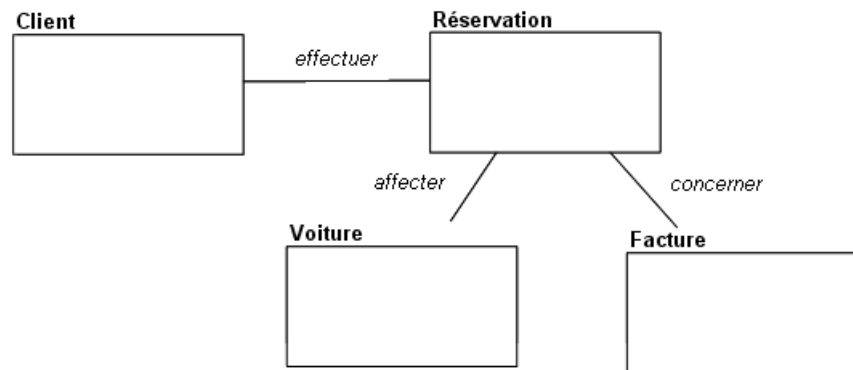
➡ Pour plus de détails, voir [Identifiant d'une entité](#).

---

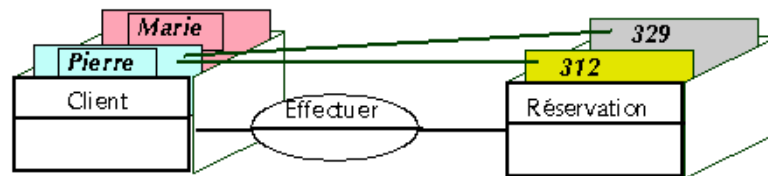
## Associations (IDEF1X)

📖 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.

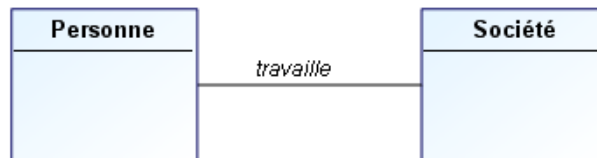
➡ Le diagramme ne doit pas permettre de représenter des situations irréalistes ou aberrantes.

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

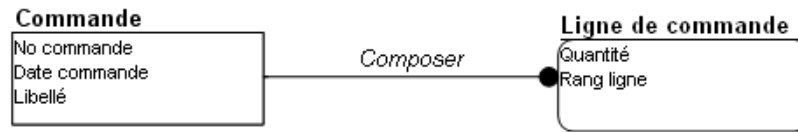
 Une relation identifiante obligatoire est une association entre des entités dans laquelle chaque instance de la première entité est associée à zéro, une ou plusieurs instances de la seconde entité et chaque instance de la seconde entité est associée à une instance de la première entité et est identifiée 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.

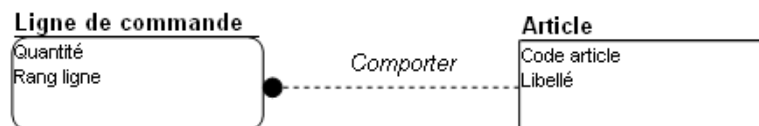
## Relation non-identifiante obligatoire

 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.

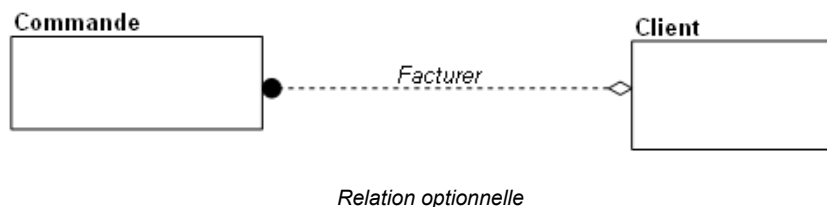
Dans une **relation optionnelle** non-identifiante, chaque instance de l'entité fille est reliée à zéro ou une instance de l'entité parente.

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.

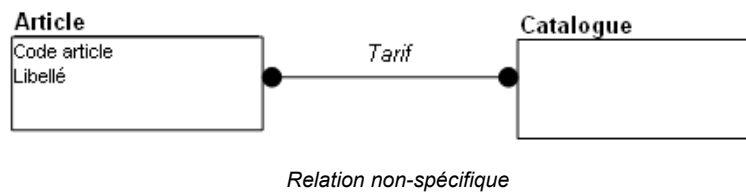
Dans le développement initial d'un modèle, il est souvent pratique d'identifier des "relations non-spécifiques" entre les entités. Ces relations non-spécifiques sont raffinées dans les phases de développement ultérieures du modèle.

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

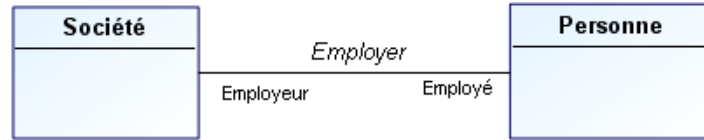
Dans l'exemple ci-dessus, un article peut faire l'objet d'une remise pour zéro, un ou plusieurs clients, et un client peut bénéficier de remises sur zéro, un ou plusieurs articles. Dans chaque cas, le taux de remise est indiqué sur l'entité associative.

## 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.

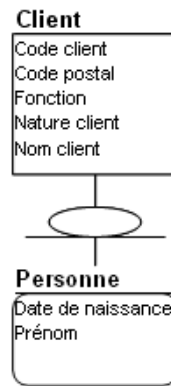
Les catégories sont aussi toujours des entités dépendantes.

## 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.



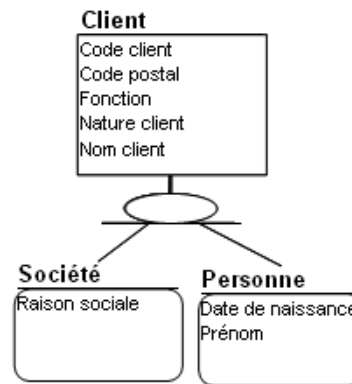
*Relation de catégorisation*

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.

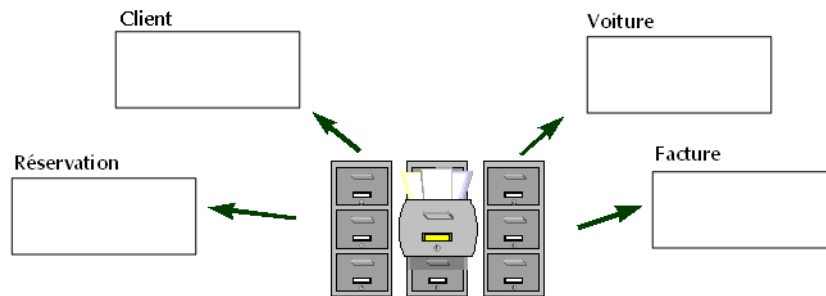
---

## 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.

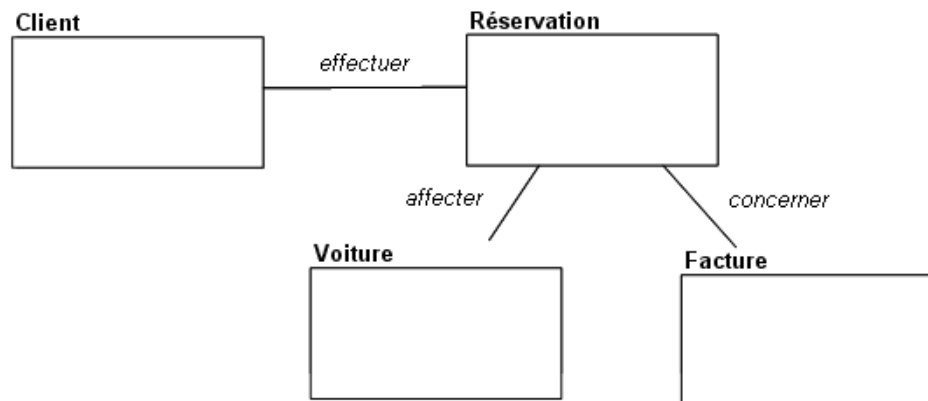
## Associations (I.E)



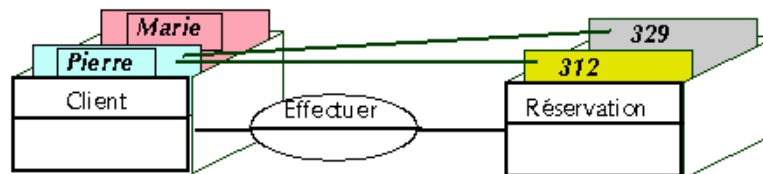
Une association représente un lien significatif entre deux objets. Les associations sont utilisées pour saisir des données au sujet de la relation qui existe entre les deux objets.

### 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.

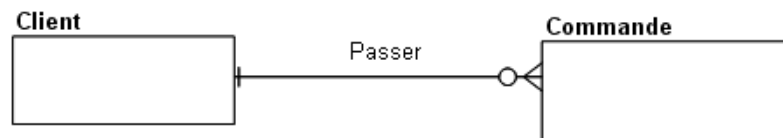
Combinée avec une multiplicité de 0 ou 1, un trait "|" est souvent utilisé pour représenter une multiplicité maximum de 1.

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.

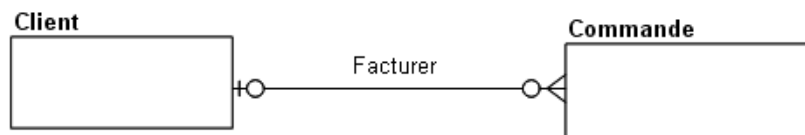


Dans l'exemple ci-dessus, un client peut passer zéro, une ou plusieurs commandes, mais une commande est toujours passée par un et un seul client.

### 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.

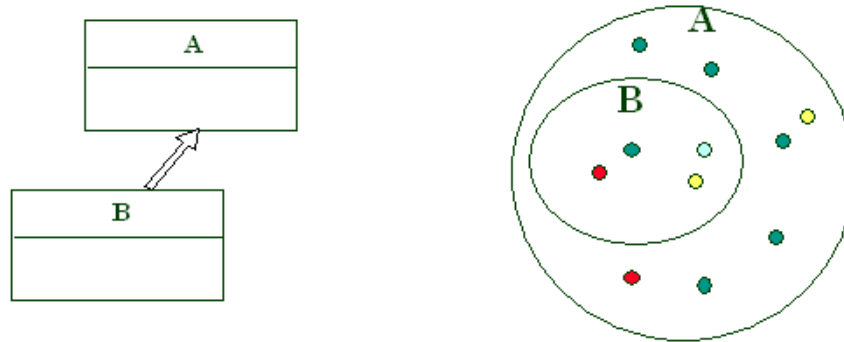
La représentation de l'association change en fonction de la nouvelle valeur de ses multiplicités.

 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.

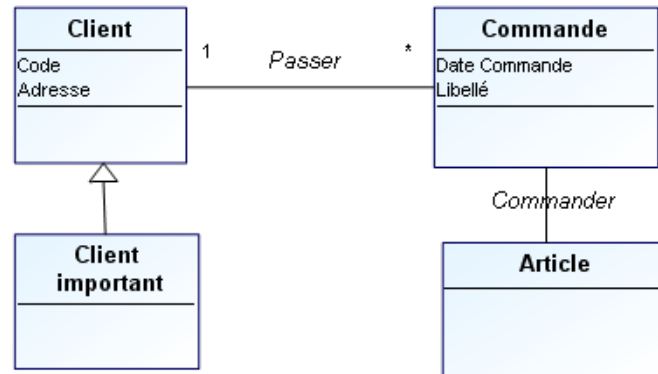
B étant un sous-ensemble de A, les instances de l'entité B "héritent" des caractéristiques de celles de l'entité A.

Il n'est donc pas nécessaire de décrire de nouveau pour l'entité B :

- ses attributs
- ses associations

Exemple :

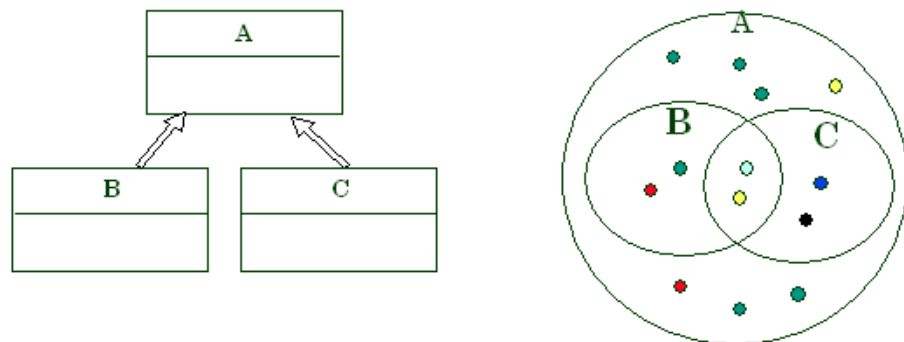
L'entité "Client important" qui représente les clients dont le "C.A. sur les 12 derniers mois" dépasse 1 million d'euros, peut être un sous-type de l'entité client.



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".

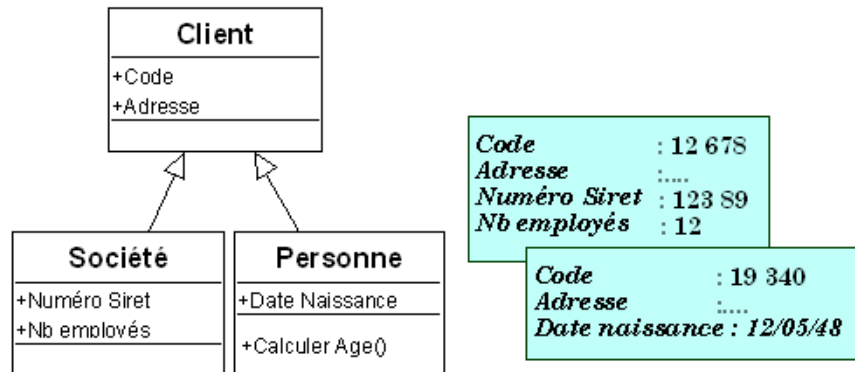
## Cas de plusieurs sous-types



Plusieurs sous-types d'une même entité :

- ne sont pas forcément exclusifs.
- ne forment pas nécessairement une partition du type.

## Intérêt des sous-types

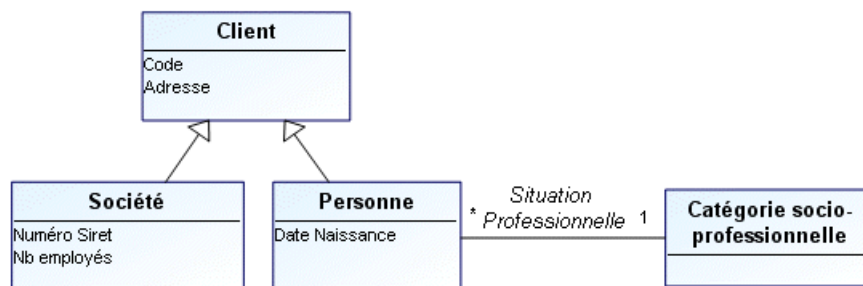


Une entité sous-type hérite de tous les attributs et associations de son sur-type, mais elle peut avoir des attributs ou des associations que ne possède pas son sur-type.

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.

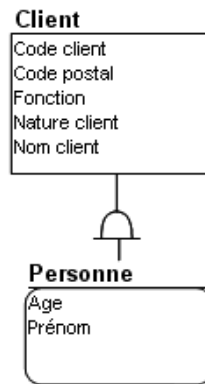
## Héritage multiple

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.

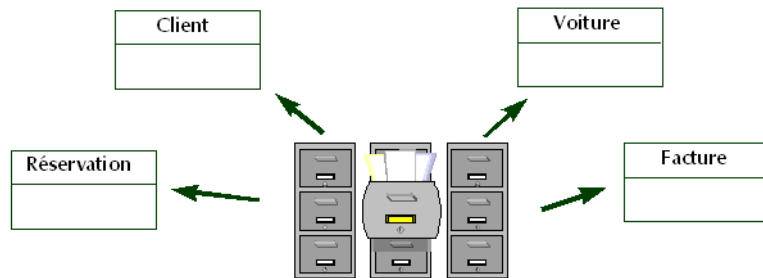
---

## 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)

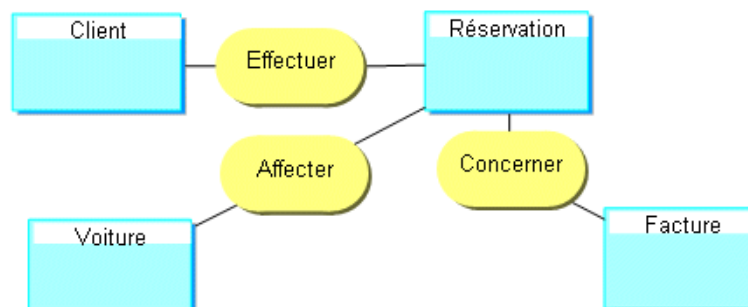
Une association représente un lien significatif entre deux objets. Les associations sont utilisées pour saisir des données au sujet de la relation qui existe entre les deux objets.

### Exemples d'associations

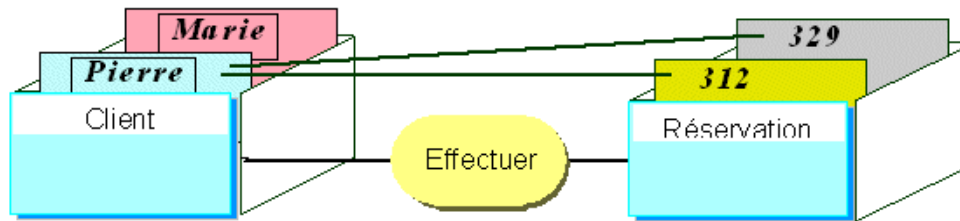
Pour modéliser qu'un "salarié" est responsable d'un "service" et préciser la "date début" de ses fonctions, est créé le modèle de données suivant, où "date début" est une propriété de l'association.



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.



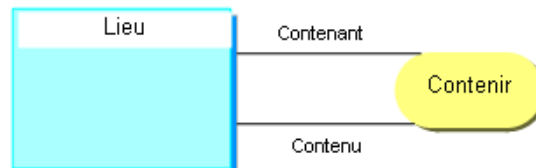
Pierre et Marie sont des clients. Pierre a effectué les réservations numéros 312 et 329.

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.*

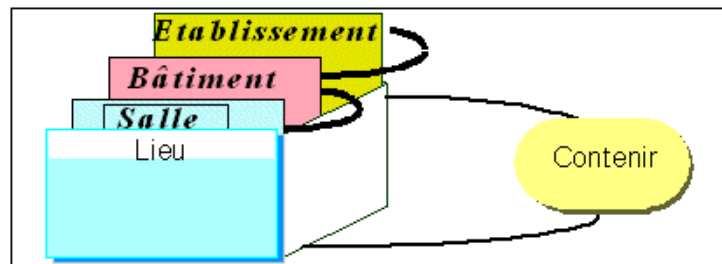
## Relation réflexive

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.

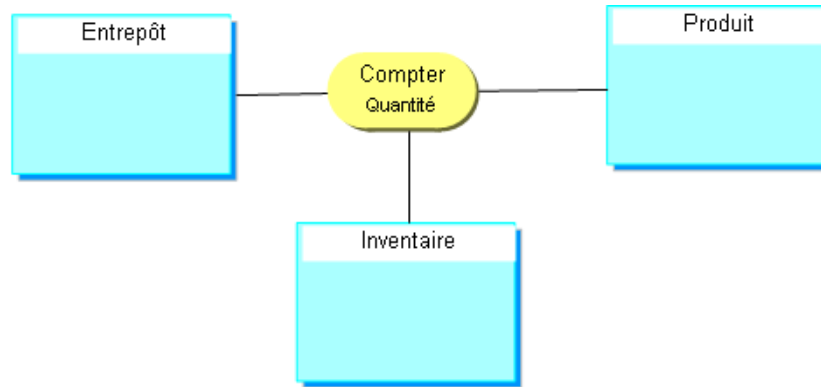
## Relation plus que binaire

Certaines associations associent non pas deux, mais davantage d'entités.

Ces associations sont, en principe, rares.

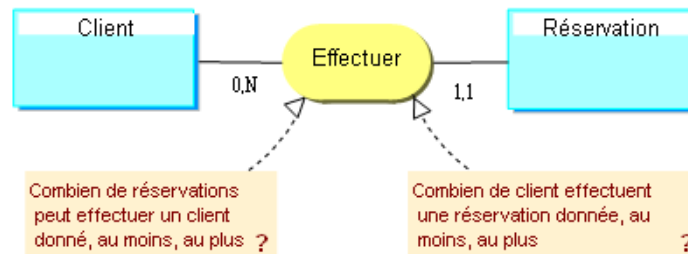
### Exemple

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.

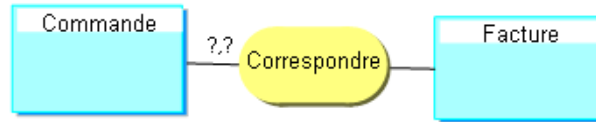


Les participations ou cardinalités usuelles sont 0,1 1,1 0,N 1,N.

- 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.

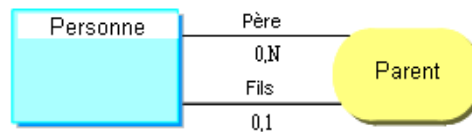
### Relation réflexive

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 déroulante)
- Tout/Partie: (chambre déroulante)
- Multiplicité: (chambre déroulante)
- 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".

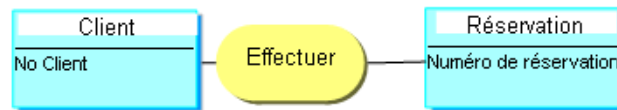
### Exemples d'attributs

"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

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 ".

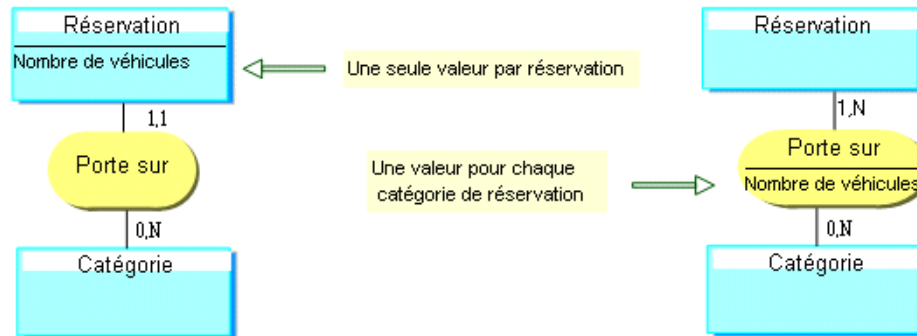
Quand vous avez terminé, fermez la fenêtre de propriétés.

## 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).**

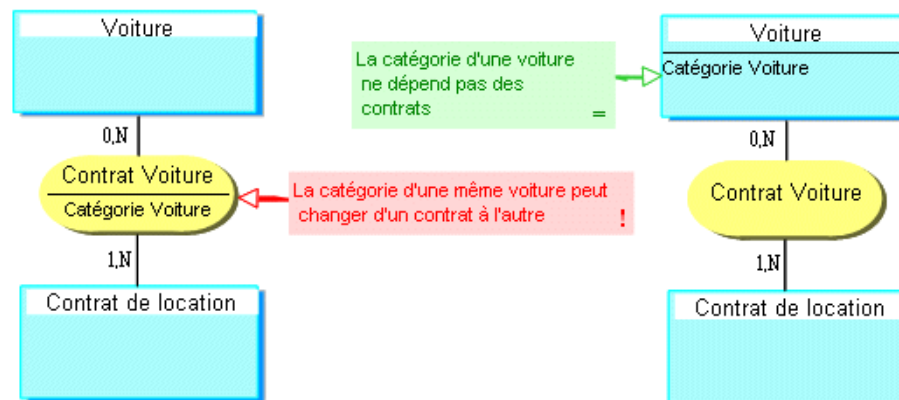


Si le nombre de véhicules est porté par l'entité "Réservation", on ne peut indiquer que le nombre total de véhicules pour une réservation. On doit donc faire une réservation par catégorie de véhicule loué (cardinalités 1,1).

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.**

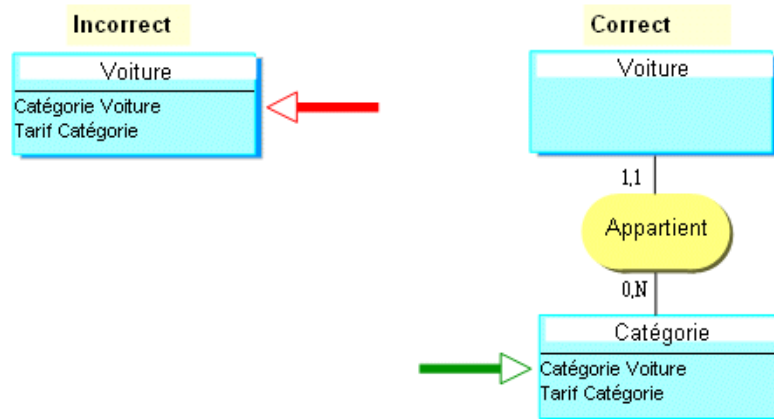


Le fait que la catégorie de voiture porte sur l'association "Catégorie Voiture" suppose que la catégorie de la voiture puisse changer d'un contrat à l'autre, ce qui ne serait pas très honnête.

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.

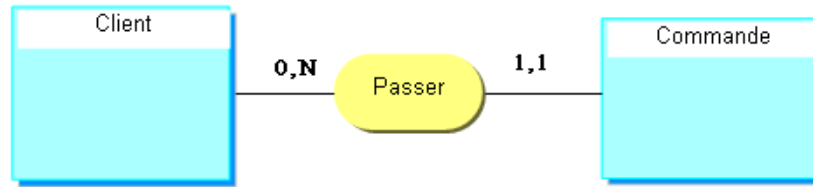
Quand plusieurs rôles, c'est-à-dire plusieurs liaisons, existent entre une entité et une association, les cardinalités sont définies pour chaque rôle.

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.

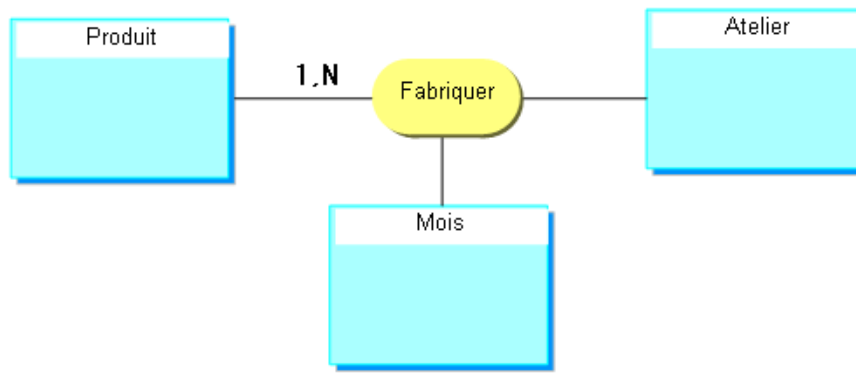
☛ Si l'expression des cardinalités n'est pas suffisante pour décrire la liaison qui existe entre une entité et une association, par exemple quand une cardinalité dépend d'un contexte d'organisation, il est possible d'utiliser les contraintes de cardinalité, qui permettent une description plus précise.

### Exemples



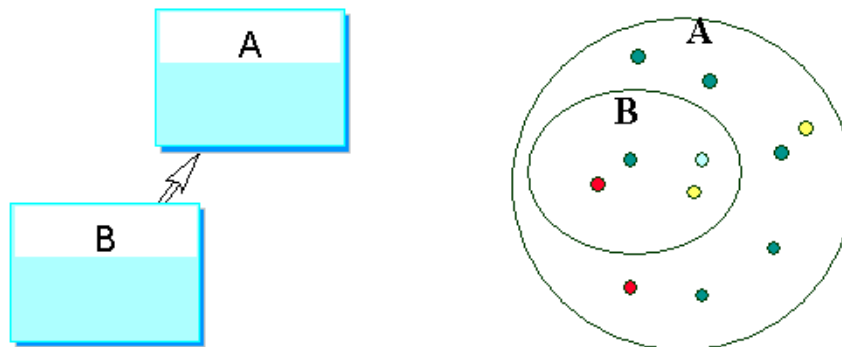
0,N : Le client peut ne pas passer de commande, il peut passer un maximum de N commandes (N indéterminé).

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.

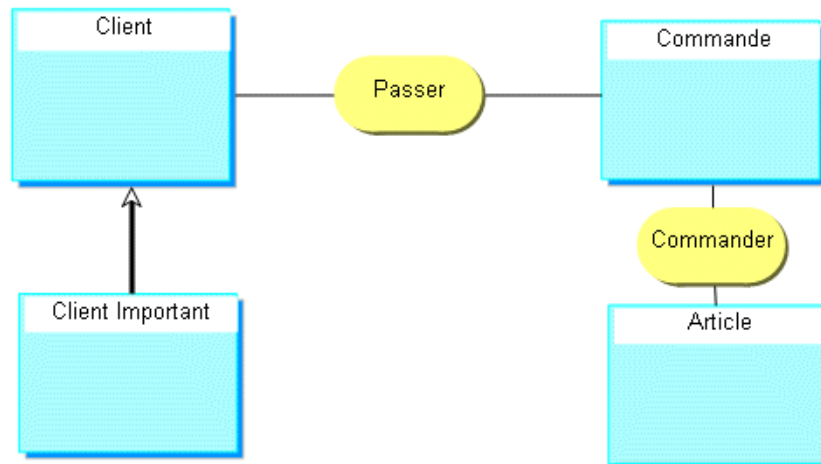
B étant un sous-ensemble de A, les exemplaires de l'entité B " héritent " des caractéristiques de ceux de l'entité A.

Il n'est donc pas nécessaire de décrire de nouveau pour l'entité B :

- ses propriétés,
- ses associations.

### Exemple

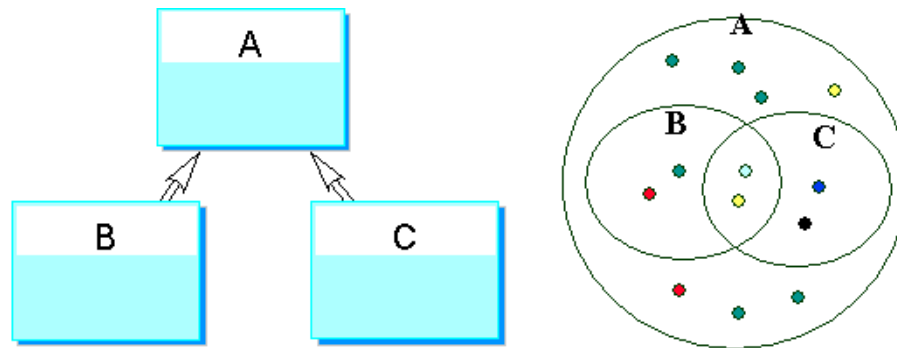
L'entité "Client important" qui représente les clients dont le "C.A. sur les 12 derniers mois" dépasse 1 million d'euros, peut être un sous-type de l'entité client (origine).



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".

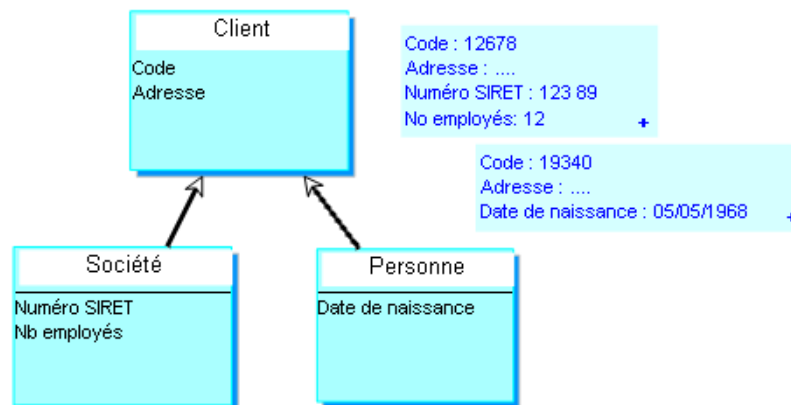
### Cas de plusieurs sous-types



Plusieurs sous-types d'une même entité

- ne sont pas forcément exclusifs.
- ne forment pas nécessairement une partition du type.

### Intérêt des sous-types



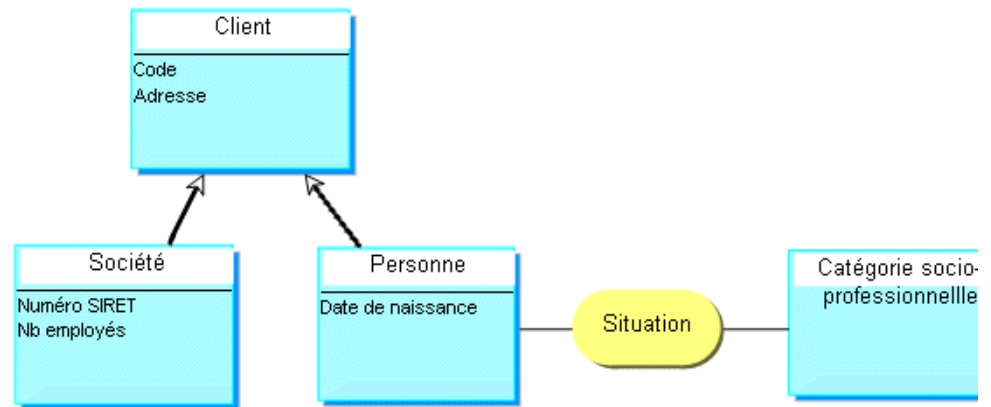
Une entité sous-type peut avoir des propriétés spécifiques. Celles-ci n'ont de sens que pour un sous-type particulier. Dans l'exemple ci-dessus :

- 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

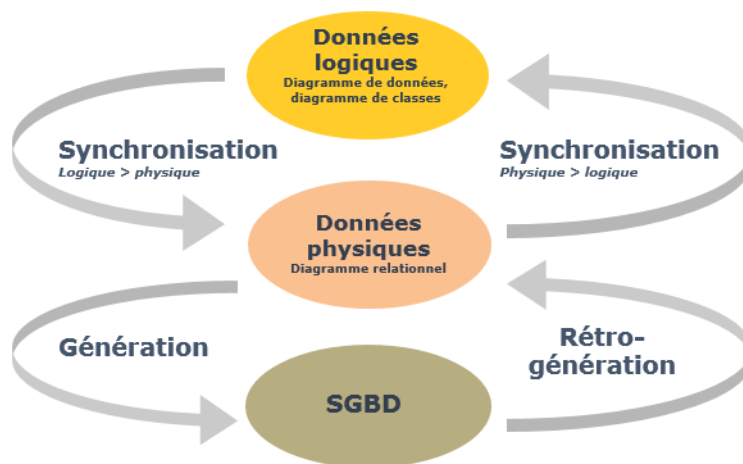


La base de données est l'objet physique qui permet de stocker et d'organiser les données logiques en vue de leur utilisation par des programmes correspondant à des applications distinctes, et de manière à faciliter l'évolution indépendante des données et des programmes.

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**

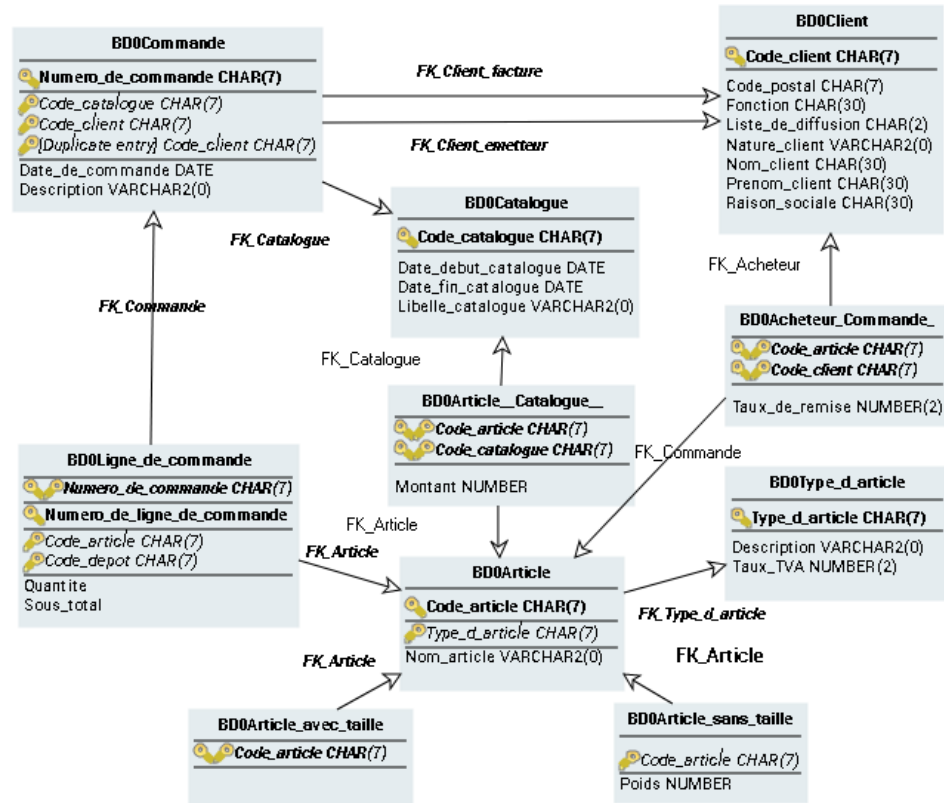
Sur les composants de type table ou vue physique vous pouvez définir le mode d'accès à l'objet référencé (création, lecture, suppression, etc.).

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 recrée 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**).

---

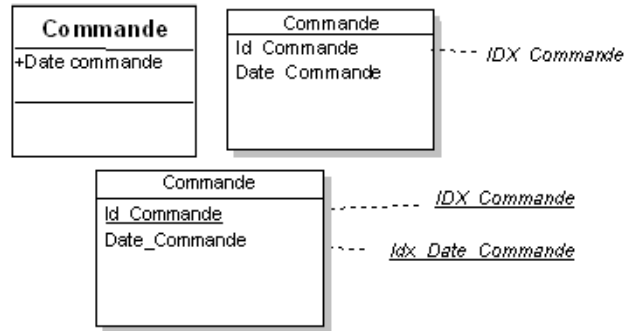
## Créer un index

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 :

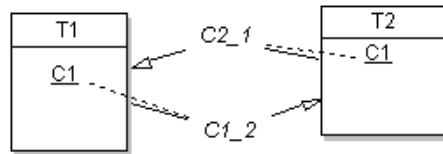
1. Cochez la case de la colonne **Périmètre** qui correspond à la clé.
2. Cliquez sur le bouton **Appliquer**.  
Le bouton **Appliquer** retire de la liste des propositions les clés explicitement acceptées ou refusées.

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

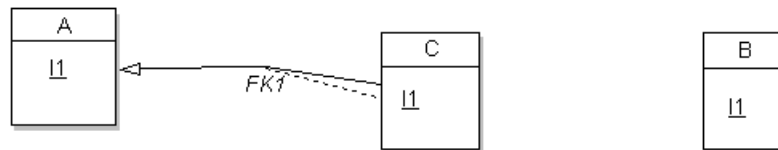
Quand la même colonne est clé primaire de deux tables, la proposition de clé étrangères propose la création de chacune des deux clés.



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

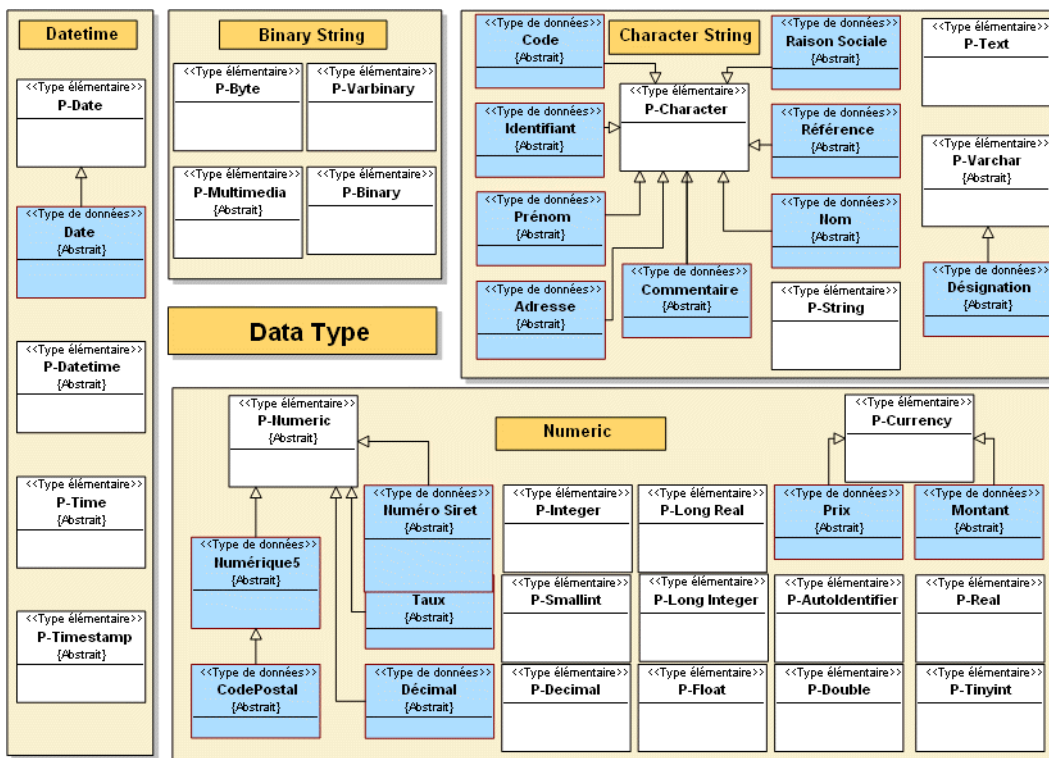
Si vous utilisez la notation UML et les diagrammes de classes pour modéliser vos données - et à titre de compatibilité avec les versions antérieures de **Hopex Database** - d'autres méthodes de référencement des types pivots sont possibles.

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.

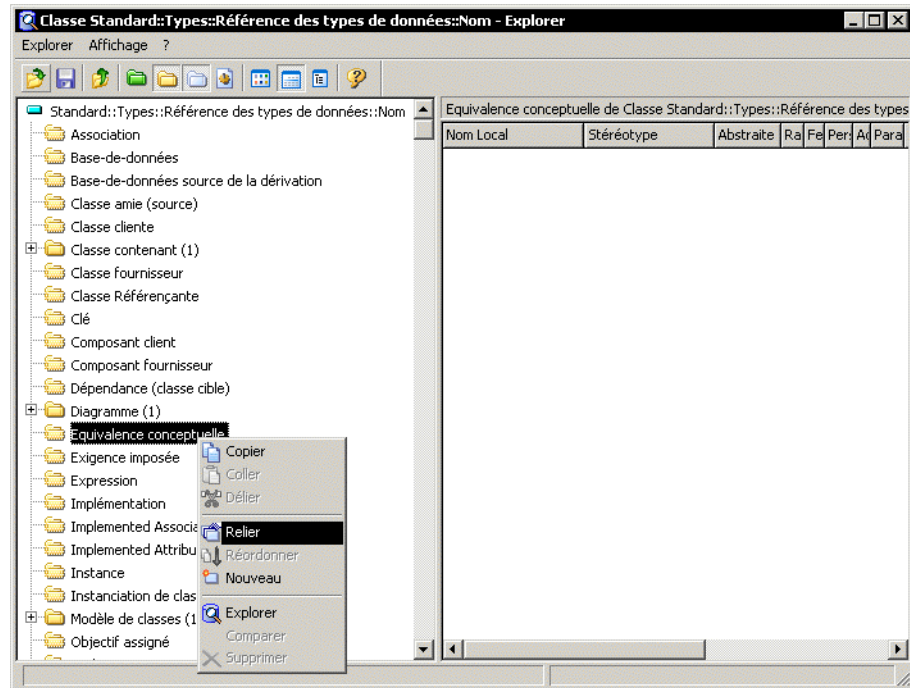
## Par un lien de correspondance

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.

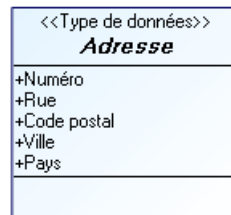
☛ *Seuls les types pivots du paquetage Standard::Types::Pivot sont proposés dans la liste.*

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.

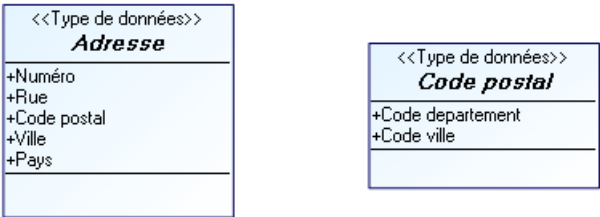


Ici le type "Adresse" est composé du numéro, de la rue, du code postal, de la ville et du pays.

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

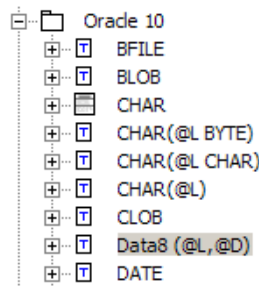
Pour créer un nouveau datatype dans **Hopex Data Governance** :

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 **Stéréotype** 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 :

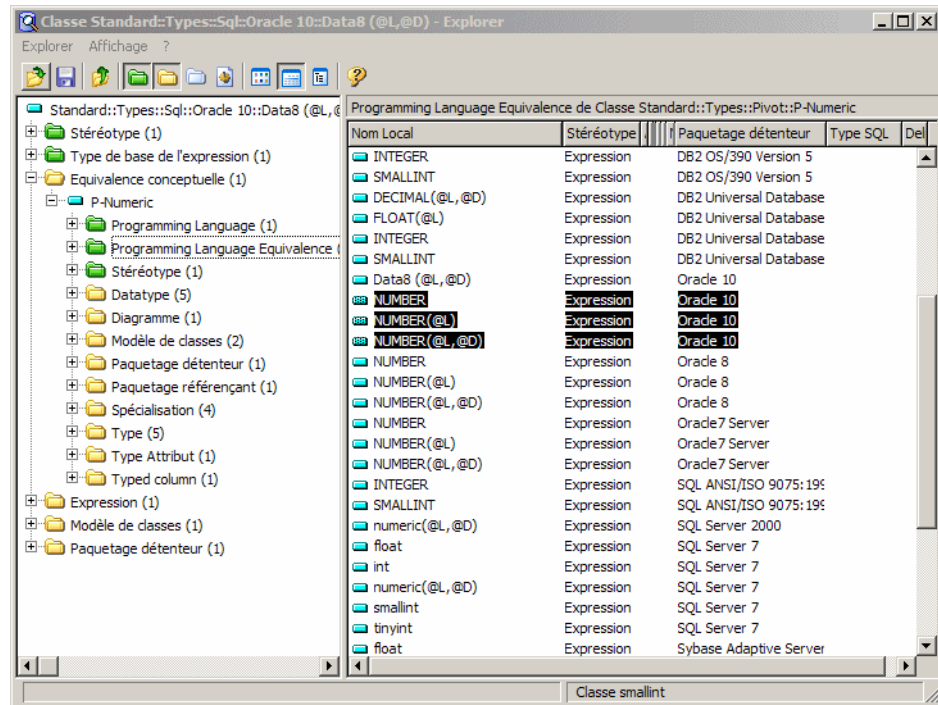
1. Ouvrez la fenêtre de propriétés du datatype "Data8 (@L,@D)".
2. Cliquez sur la page **Compléments**.
3. Faites un clic droit sur le dossier "Equivalence conceptuelle" et sélectionnez **Relier**.
4. Dans la fenêtre de recherche, sélectionnez la classe "P-Numeric".

## 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".



# 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

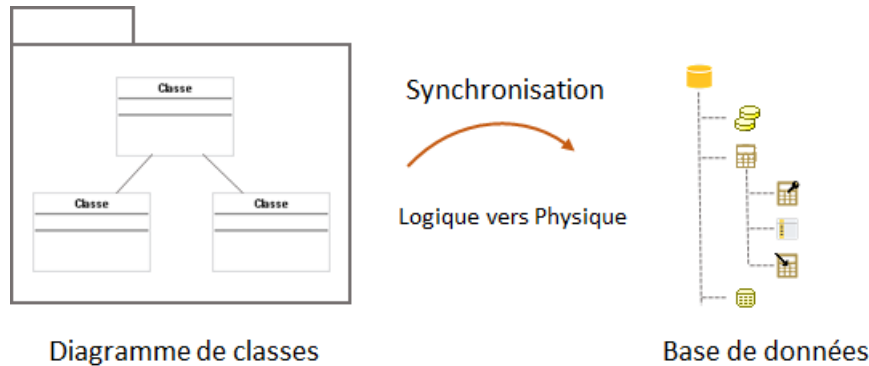
Certaines options de synchronisation sont filtrées par défaut. Pour les afficher :

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](#).

---

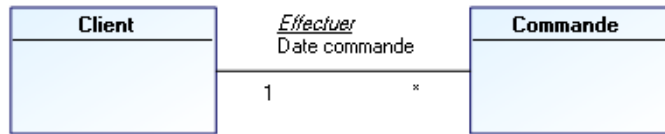
## Synchronisation logique > physique : les Associations

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)

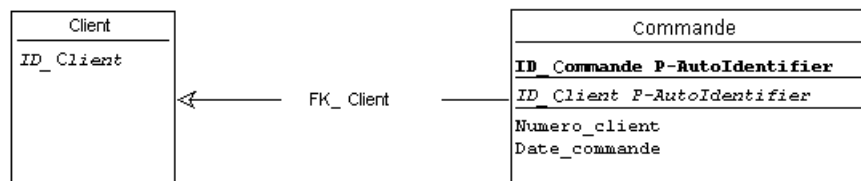
Une association contrainte est une association binaire dont la multiplicité maximum d'un de ses rôles est 1. Dans ce cas, il n'est pas nécessaire de créer une table correspondant à cette association. Une colonne rajoutée dans la table correspondant à l'entité suffit.

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".

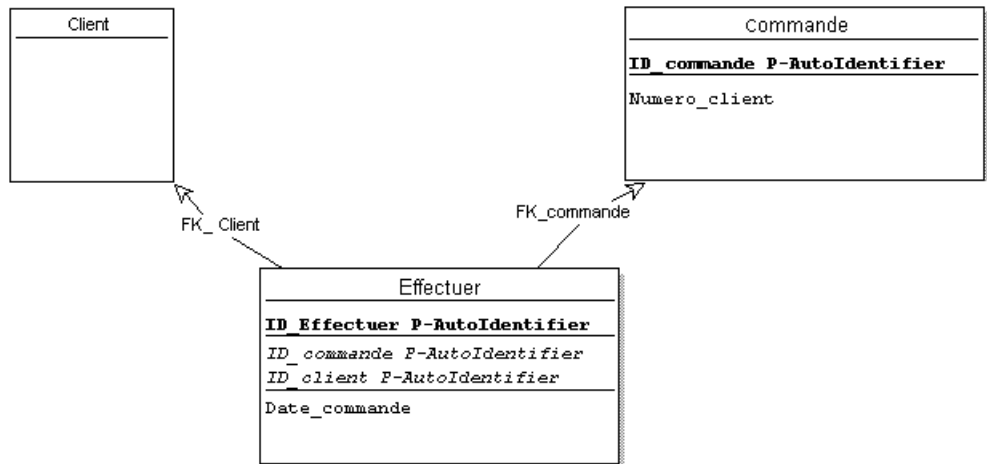
La clé étrangère ("foreign key") est créée à partir de l'identifiant de l'entité.

### ***L'association est transformée en table***

Pour que l'association se transforme en table :

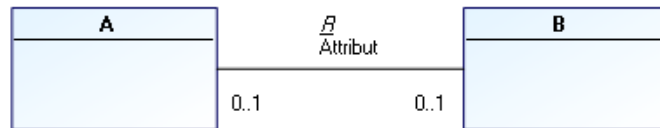
1. Ouvrez la fenêtre de propriétés de l'association.
2. Cliquez sur l'onglet **Caractéristiques**.

3. Dans le champ **Correspondance potentielle**, sélectionnez la valeur "Table".



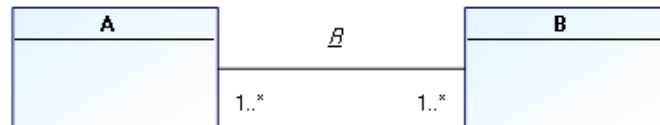
### Association contrainte (multiplicités : 0,1 et 0,1)

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 :

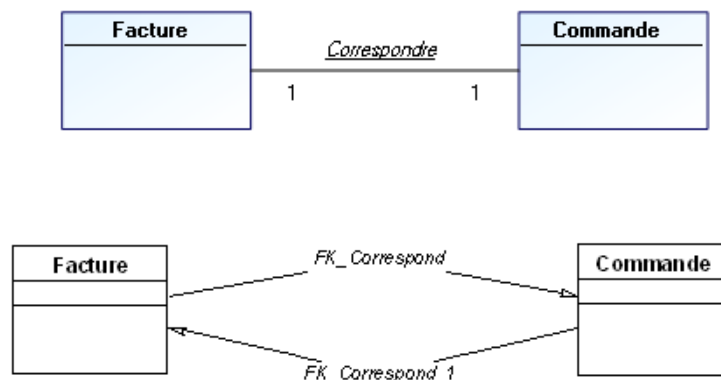
- Un objet de type A doit exister pour pouvoir créer un objet de type B et le lui relier.
- Réciproquement, un objet de type B doit exister pour pouvoir créer un objet de type A et le lui relier.

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.

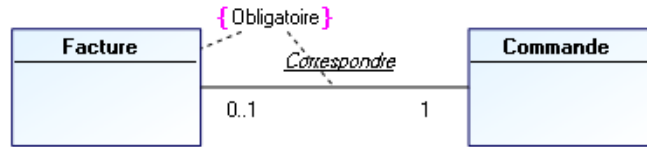
Certains SGBD interdisent complètement la création de tables de ce type.

Pour permettre une synchronisation correcte, il vous faut éviter les situations de ce genre.

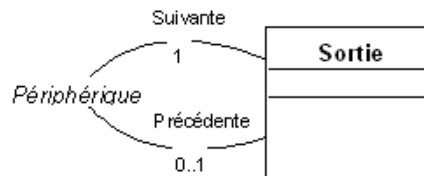
☛ Choisissez une des clés étrangères pour laquelle vous mettez la multiplicité 1 et sur l'autre, mettez 0..1. Vous pouvez également retirer la clé étrangère de la table après la synchronisation, mais c'est moins pratique.

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.

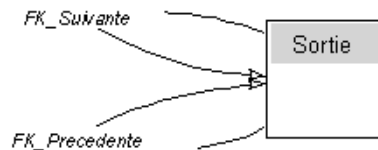


Voici un autre exemple dans le cas d'une nomenclature : Nomenclature 1, 1 ou 0..1, 1.



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](#).




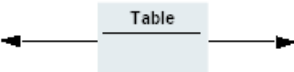
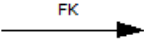
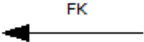
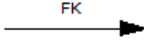
Classe associative

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.

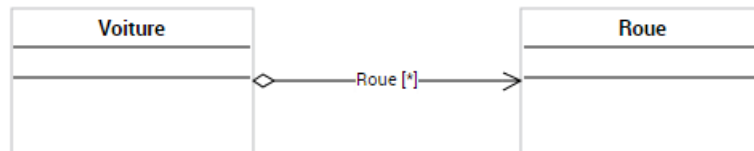


Après synchronisation, la partie "Voiture" donne lieu à une table :

- 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 / \*

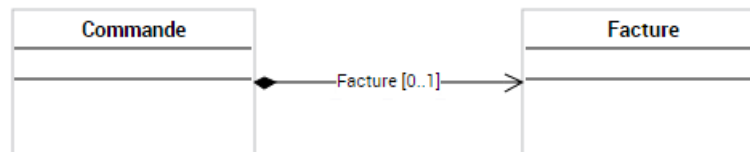
Une voiture peut avoir une ou plusieurs roues.



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
<b>Commandes</b>	<input checked="" type="checkbox"/>		<b>Commandes</b>	
Contrat d'achat	<input checked="" type="checkbox"/>		Achat::Contrat d'ac...	
Ordre d'achat	<input checked="" type="checkbox"/>		Achat::Ordre d'achat	
Diagramme de classes	<input checked="" type="checkbox"/>		Achat::Ordre d'acha...	
Diagramme de données	<input checked="" type="checkbox"/>		Achat::Ordre d'acha...	
Article Commandé	<input checked="" type="checkbox"/>		Achat::Article Com...	
Contrat	<input checked="" type="checkbox"/>		Achat::Contrat	
Employé	<input checked="" type="checkbox"/>		Employé::Employé	
Fournisseur	<input checked="" type="checkbox"/>		Achat::Fournisseur	
Ordre d'achat	<input checked="" type="checkbox"/>		Achat::Ordre d'achat	
Produit	<input checked="" type="checkbox"/>		Achat::Produit	
Site (EN)	<input checked="" type="checkbox"/>		Achat::Site (EN)	
Fournisseur , Site fourni...	<input checked="" type="checkbox"/>		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.

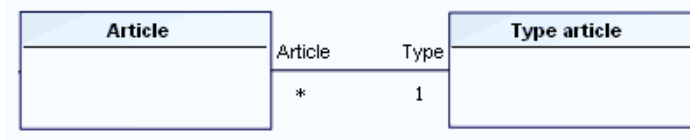
Un objet non réalisé ne donne pas lieu à la création d'un objet lors de la synchronisation mais il est traité comme un objet abstrait. Voir [Entité abstraite](#).

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.

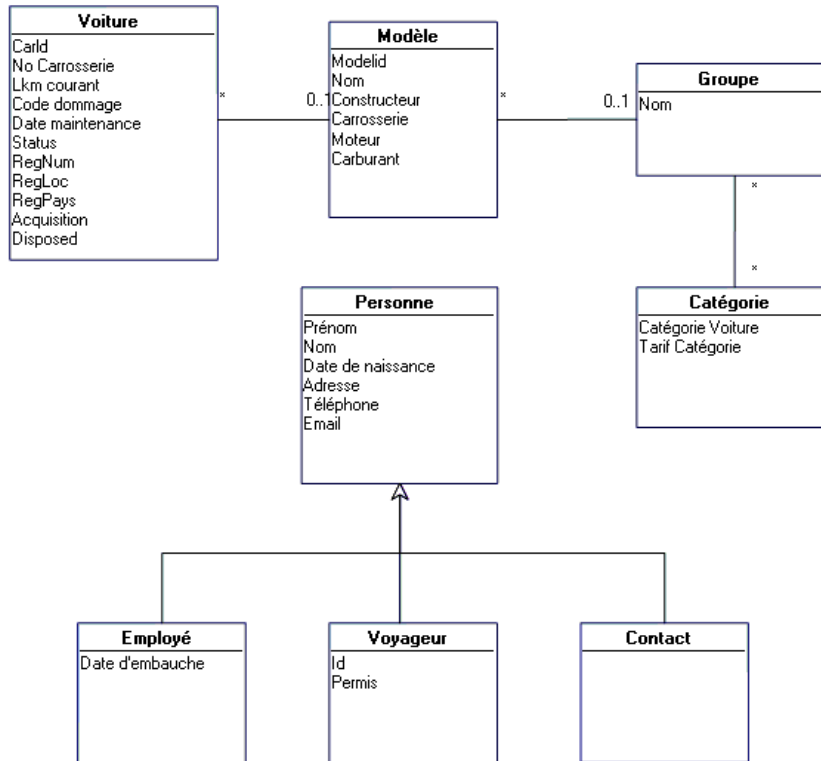
### Impact de l'objet synchronisé sur les autres objets

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

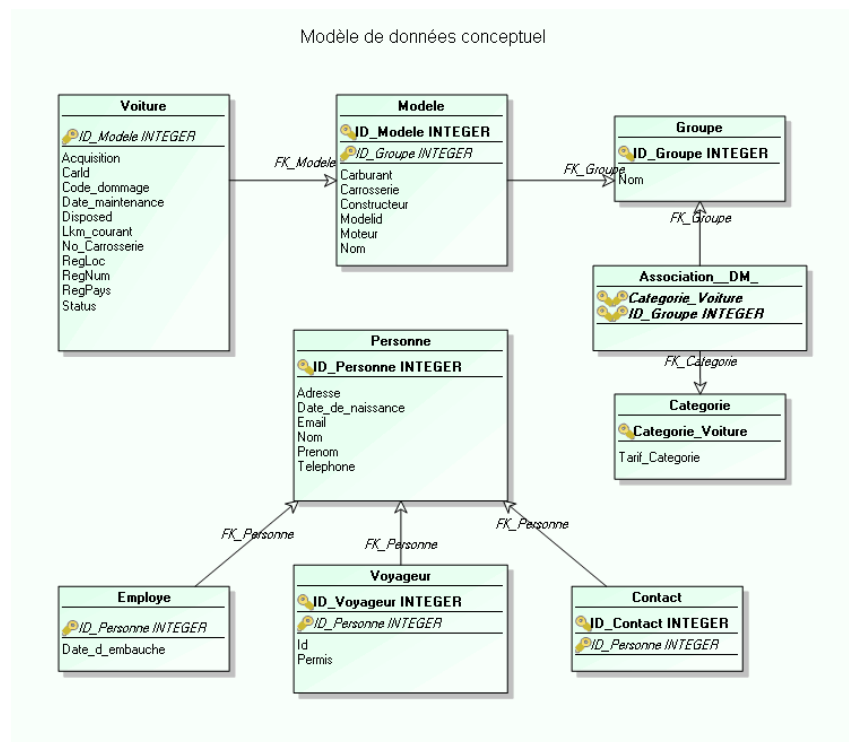
associés à l'entité source et qui sont nécessaires à la mise à jour de la table correspondante.

### 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

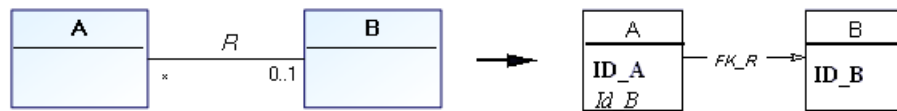
Les modifications des caractéristiques d'un attribut (type, longueur, décimales,...) sont reportées sur la colonne correspondante du diagramme relationnel.

Si la valeur d'une caractéristique de la colonne a été modifiée directement dans le diagramme relationnel, cette valeur est conservée.

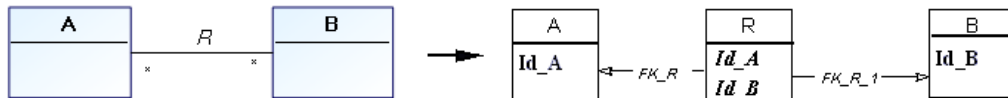
## Modification du nom d'un attribut, d'une entité, ou d'une association

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.

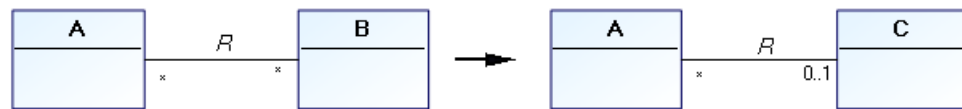
## Modification de la multiplicité maximum d'une association



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](#)).

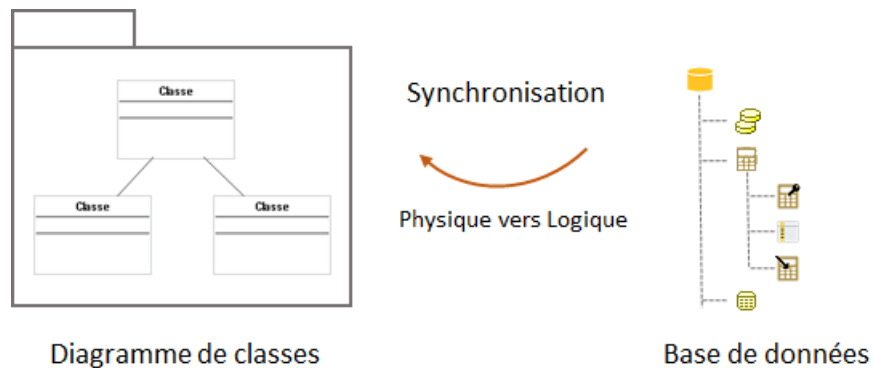
## DU MODÈLE PHYSIQUE AU MODÈLE LOGIQUE

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 :

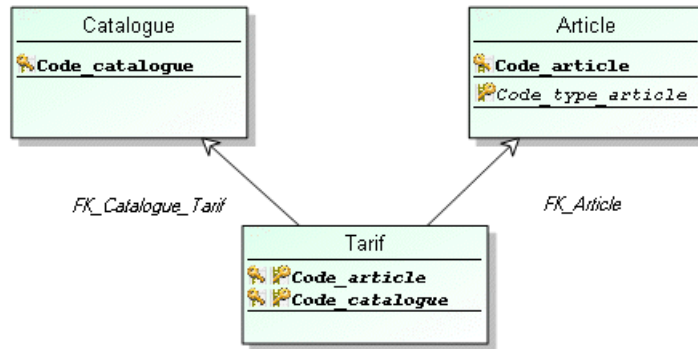


Résultat au niveau logique :



- 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 :



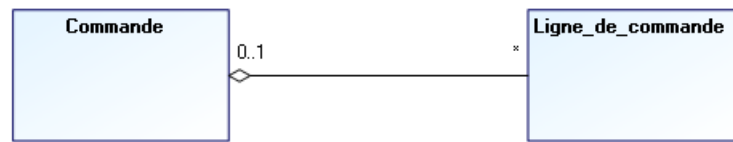
- Une table dont la clé primaire contient des clés étrangères et au moins une colonne qui n'est pas clé étrangère devient une entité. Une association agrégée est créée entre cette entité et l'entité correspondant à la table vers laquelle pointe chacune des clés étrangères.

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 :

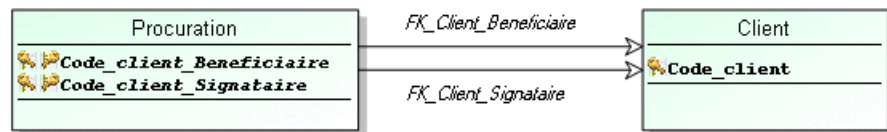


Résultat niveau logique :

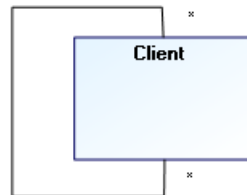


- Une table dont la clé primaire est constituée uniquement de deux clés étrangères pointant sur une même table devient une association réflexive de multiplicités (\*..\*).

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

Options - Synchronisation

Noms de Table: 128,DB^ROOT

Noms de colonnes: 128,^ROOT

Noms colonnes de FK: 128,^ROOT

Noms de PK: 128,PK\_^ROOT

Noms de FK: 128,FK\_^ROOT

Noms des Index de PK: 128,IDX\_^ROOT

Noms des Index de FK: 128,IDX\_^ROOT

Noms des index: 128,IDX\_^ROOT

Noms cols PK auto: 128,ID\_^TBL

Colonnes Not Null: Not Null

OnUpdate:

OnDelete:

Not Null PkColumns: Not Null

- **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

**Location**

Rechercher...

Location véhicule  
+ Eléments Détenus  
+ Club fidélité  
+ Compte fidélité  
+ Enregistrement Local  
+ Agence  
+ Catégorie durée  
+ Conducteur  
+ **Location**  
+ Taux location  
+ Logistique  
+ Médiathèque - Emprunts

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 :

<b>Table</b>	Préfixe de la base de données + nom* de l'entité ou de l'association
<b>Colonne</b>	Nom* de l'attribut
<b>Clé primaire</b>	"PK_" + nom* de l'entité ou de l'association
<b>Clé étrangère</b>	"FK_" + nom* de l'entité cible ou du rôle si renseigné
<b>Index de clé primaire</b>	"IDX_" + nom* de l'entité ou de l'association
<b>Index de clé étrangère</b>	"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.

### Noms des Index de FK

128,IDX

### Noms des index

128,IDX

### Noms cols PK auto

128,ID\_ ^TBL

Modifier

Réinitialiser le paramètre

Réinitialiser tous les paramètres

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 prefixer les noms. Ces éléments sont les suivants :

<b>^ROOT</b>	<ul style="list-style-type: none"> <li>• Pour une table : nom* de la classe, de l'association ou de la partie dont elle provient.</li> <li>• Pour une colonne : nom* de l'attribut ou nom de l'identifiant.</li> <li>• Pour une clé primaire : nom* de la classe, de l'association ou partie dont elle provient.</li> <li>• Pour une clé étrangère : nom* de la classe cible ou du rôle si renseigné.</li> <li>• Pour une colonne de FK : nom* de l'attribut.</li> <li>• Pour une colonne de PK auto : nom* de l'identifiant de la classe.</li> <li>• Pour un index sur clé primaire : nom* de la classe, de l'association ou de la partie.</li> <li>• Pour un index sur clé étrangère : nom* de la classe cible ou du rôle si renseigné.</li> <li>• Pour un index : nom* de l'attribut, du rôle ou de la classe.</li> </ul>
<b>^DB</b>	Préfixe de la base de données
<b>^EXT</b>	<ul style="list-style-type: none"> <li>• Pour une clé étrangère : nom* de l'association, de la partie ou de la généralisation</li> <li>• Pour un index sur clé étrangère : nom* de l'association, de la partie ou de la généralisation</li> </ul>
<b>^TBL</b>	Nom local de la table ou nom local de la table de référence
<b>^TBO</b>	<ul style="list-style-type: none"> <li>• Pour une clé étrangère : nom* de la classe, de l'association ou de la partie dont elle provient</li> <li>• Pour un index sur clé étrangère : nom* de la classe, de l'association ou de la partie d'origine</li> </ul>
<b>^TBR</b>	Nom de la table de référence
<b>^KEY</b>	Nom de la clé étrangère
<b>^CPT</b>	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)

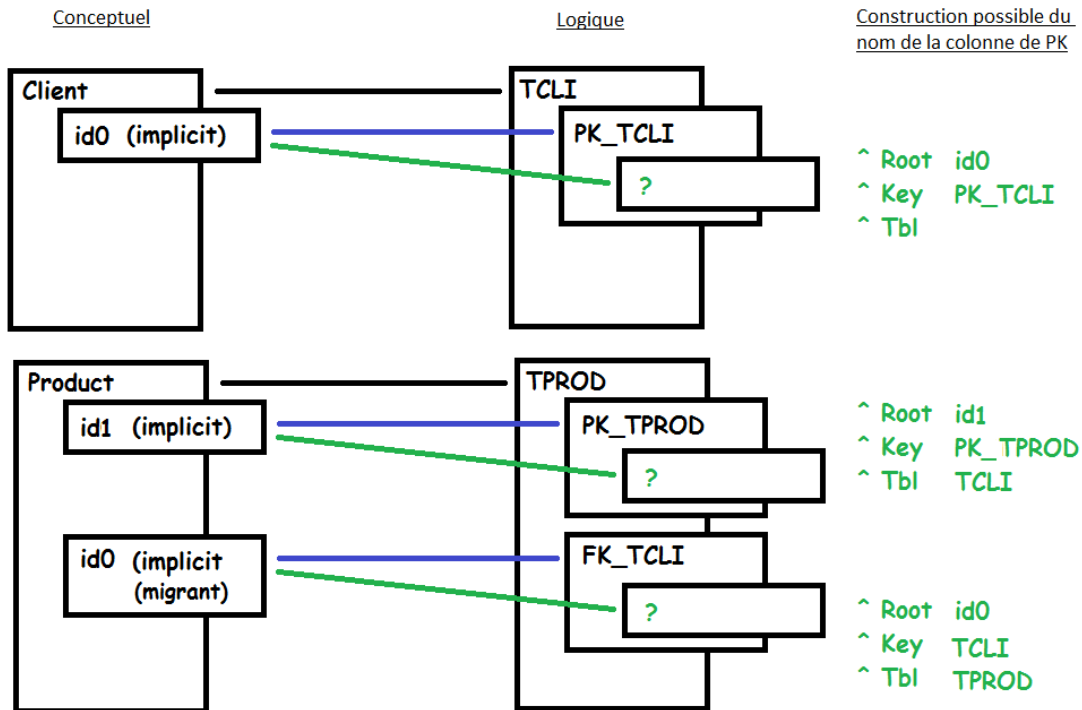
Lors de la synchronisation en mode Logique > Physique, l'identifiant de l'entité devient la clé primaire de la table. Si l'identifiant est implicite, une colonne est automatiquement créée. Pour plus de détails, voir [Règles de synchronisation "logique vers physique"](#).

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

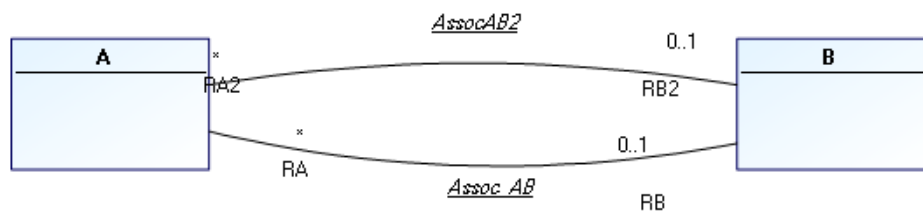
Vous pouvez modifier les règles de construction et construire le nom de ces colonnes avec le mot clé ^KEY qui correspond au nom de la clé étrangère (sans " FK\_ ") si

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](#).

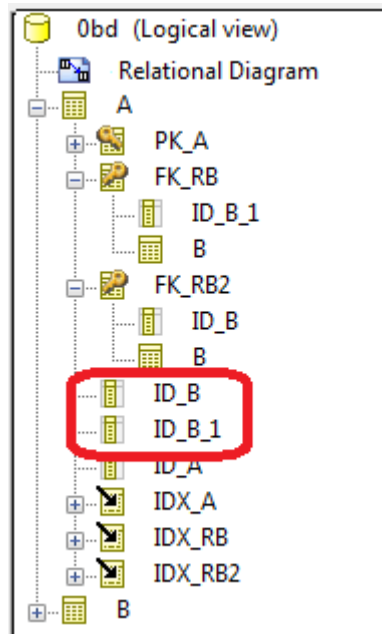


### Exemple

Lorsqu'il existe deux associations contraintes entre deux entités comme ci-dessous :



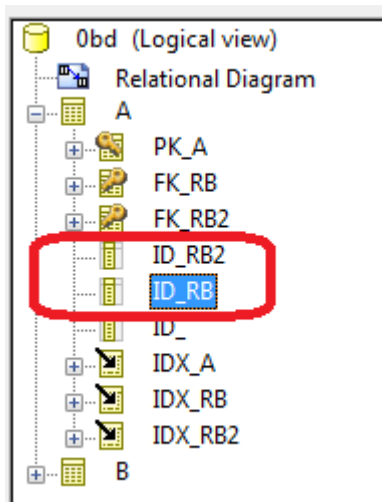
Par défaut, après synchronisation, vous obtenez deux colonnes avec des noms identiques seulement différenciés par le préfixe "1".



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.

---

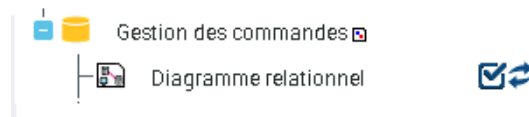
### Cas de mise à jour des diagrammes lors d'une synchronisation

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.



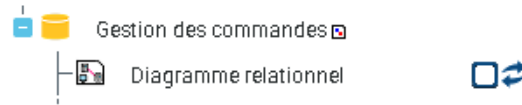
#### Après modification du diagramme cible

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.

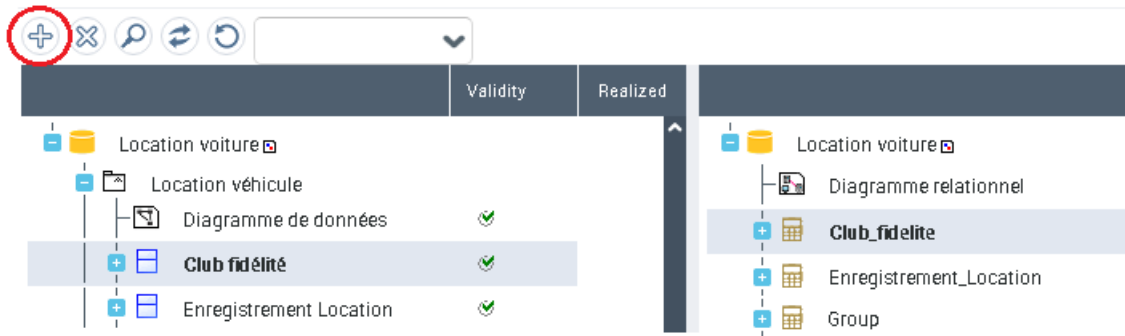
---

## 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**.

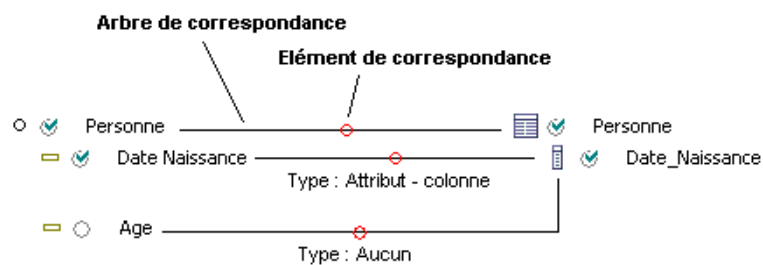


La correspondance se crée à partir du dernier objet sélectionné. Ainsi, pour créer une correspondance du modèle logique vers physique, autrement dit pour définir un objet du modèle physique à partir d'un objet du modèle logique, vous devez sélectionner l'objet du modèle logique puis celui du modèle physique, 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 :

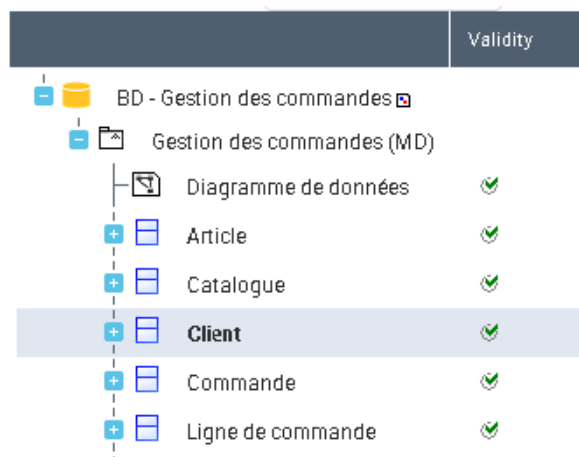
- Sélectionnez l'objet en question et cliquez sur le bouton **Enlever la correspondance**.

## 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](#).

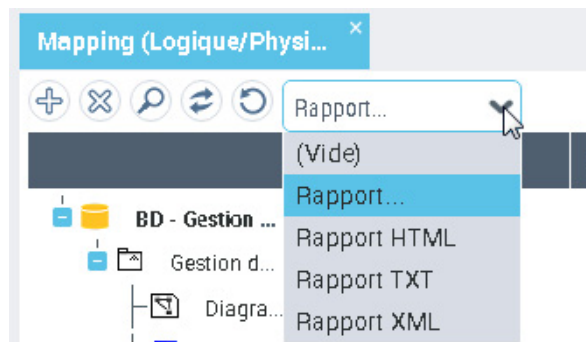
## Rapport des correspondances

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.

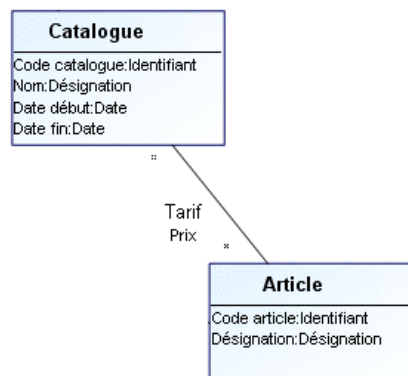
## Exemple de correspondance

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	✓	<b>Tarif</b>	✓
<b>Tarif</b>	✓		

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. Ils 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.**

## Dénormarisation : 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.

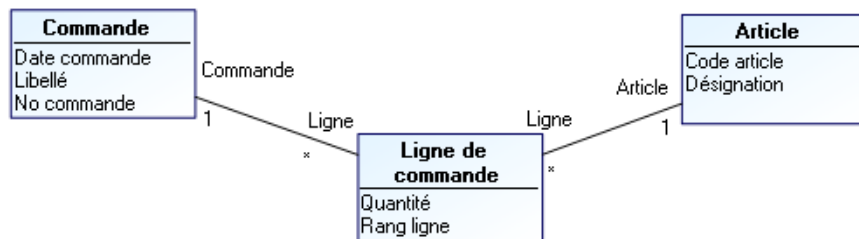
## Lancer une dénormalisation logique

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 :

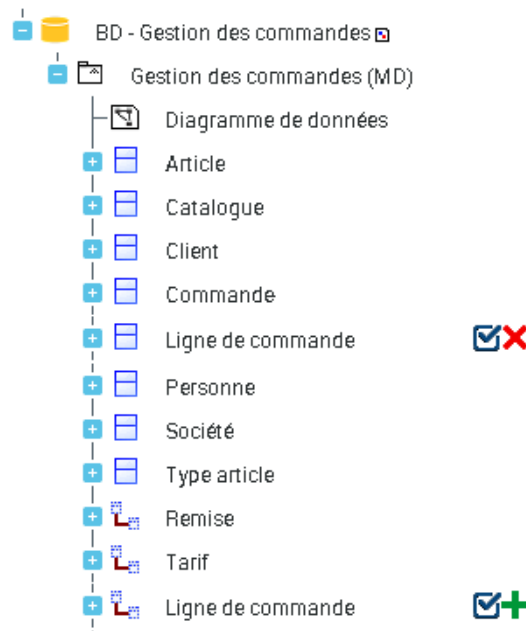
1. Faites un clic droit sur la base de données "Gestion des commandes" qui contient cette entité et sélectionnez **Dénormaliser**.  
Un assistant apparaît.
2. Cliquez sur la dénormalisation **Logique**.
3. Cliquez sur **Suivant**.
4. Dans le champ **Sélectionnez le type de dénormalisation**, sélectionnez "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.

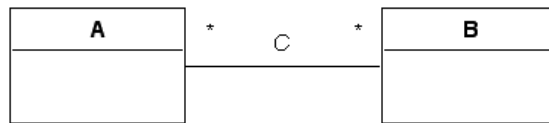
---

## Liste des assistants de dénormalisation logique

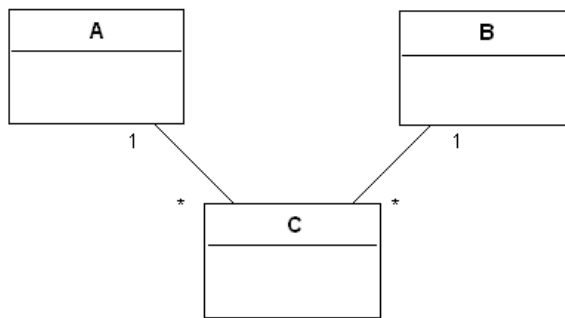
### Transformer une association en entité

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**



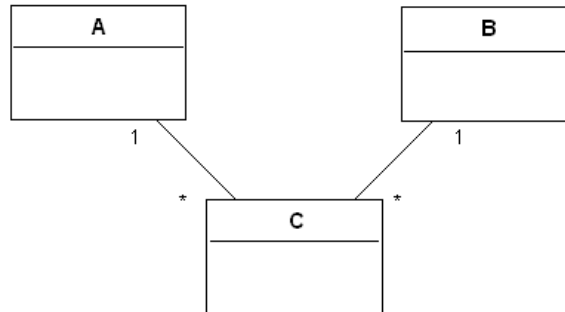
**Après**



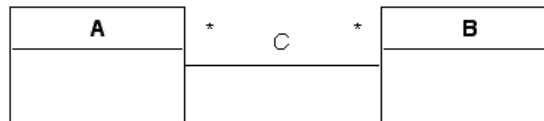
### Transformer une entité en association

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



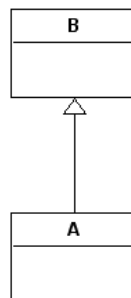
### Après



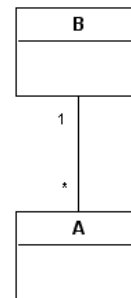
## Transformer une généralisation en association

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



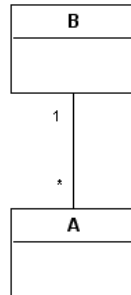
### Après



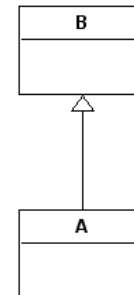
## Transformer une association en généralisation

Cette dénormalisation permet de transformer une association 1,\* en généralisation. Une généralisation est créée entre les deux entités, et l'association est supprimée.

**Avant**



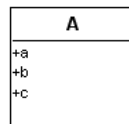
**Après**



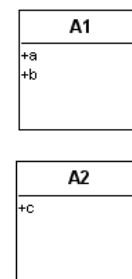
## Partition verticale d'une entité

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 remarquerez que les deux sont bidirectionnelles, ce qui signifie que entités et table se mettent à jour dans les deux sens de la synchronisation.

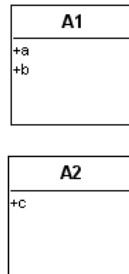
Si vous effectuez des modifications logiques sur ces entités puis relancez la synchronisation, l'éditeur affiche un signal sur la table cible ; en effet il ne sait quelle entité prendre pour effectuer les mises à jour.

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.

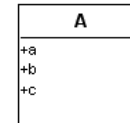
## Fusion d'entités

Cette dénormalisation permet de fusionner des entités.

**Avant**



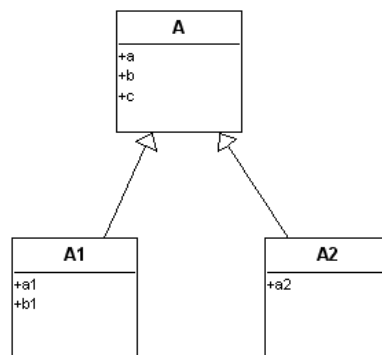
**Après**



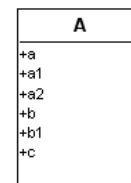
## Fusion d'entités ascendantes

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**



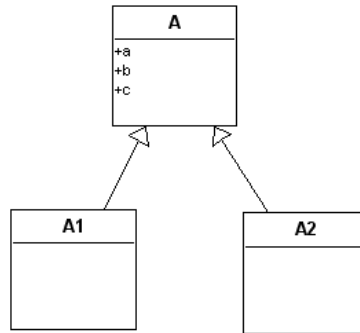
**Après**



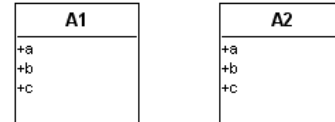
## Fusion d'entités descendantes

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**



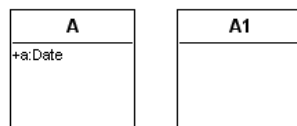
**Après**



## 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

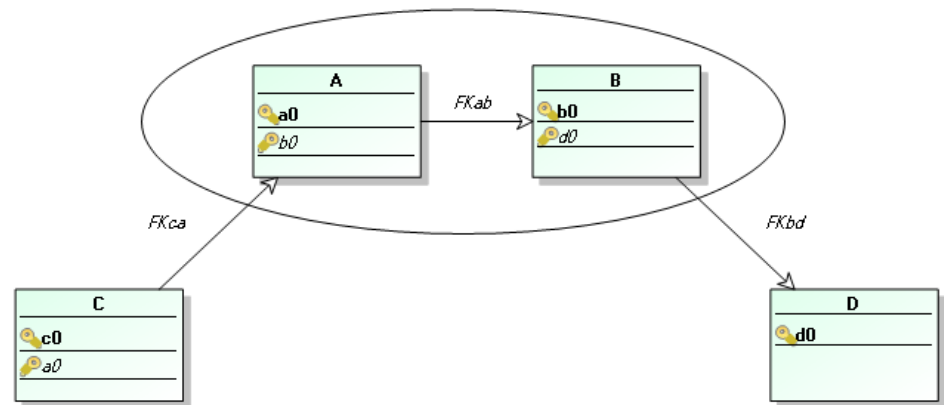
### Option des clés primaires

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

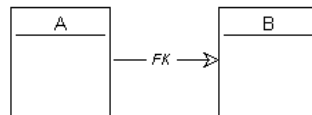
autres clés étrangères, FKca et FKbd, sont bien reportées dans le diagramme relationnel.



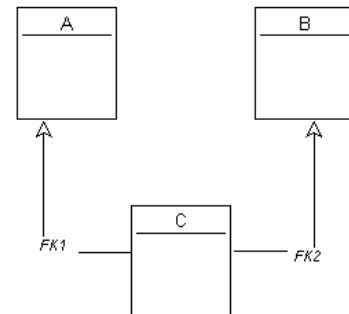
## Transformation d'une clé étrangère en table

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**



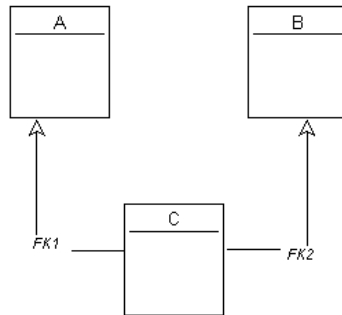
**Après**



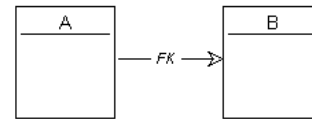
## Transformation d'une table en clé étrangère

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



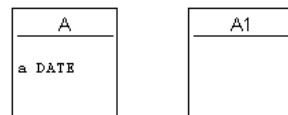
### Après



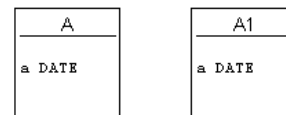
## Copier/Coller des colonnes

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](#).

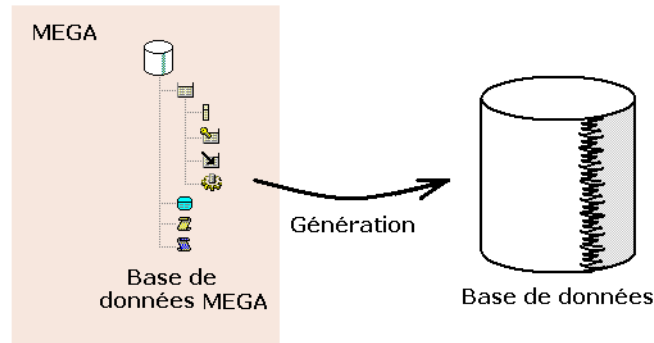
Enfin, le niveau logique peut être complété par la génération d'objets physiques propres à chaque base de données pour un SGBD, comme les vues logiques, les procédures stockées et les triggers.

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
<ul style="list-style-type: none"> <li>TBSTABLE</li> <li>FORCE LOGGING</li> </ul>	<ul style="list-style-type: none"> <li>↕</li> <li>⚠</li> </ul>	<ul style="list-style-type: none"> <li>TBSTABLE</li> <li>NO FORCE LOGGING</li> </ul>	<a href="#">Aperçu</a>
	<ul style="list-style-type: none"> <li>✗</li> <li>✗</li> </ul>	<ul style="list-style-type: none"> <li>TBSPARTITION</li> <li>TBSOVERFLOW</li> </ul>	
<ul style="list-style-type: none"> <li>DEPARTEMENT_TEST1               <ul style="list-style-type: none"> <li>DEPTNO</li> <li>DOM</li> <li>NAME</li> <li>PAYS</li> <li>REGION</li> <li>SEGMENT</li> <li>LOGGING</li> <li>COMPRESS</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>↕</li> <li>↕</li> </ul>	<ul style="list-style-type: none"> <li>DEPARTEMENT_TEST1               <ul style="list-style-type: none"> <li>DEPTNO</li> <li>DOM</li> <li>NAME</li> <li>PAYS</li> <li>REGION</li> <li>SEGMENT</li> <li>NOLOGGING</li> </ul> </li> </ul>	<a href="#">Aperçu</a>

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.

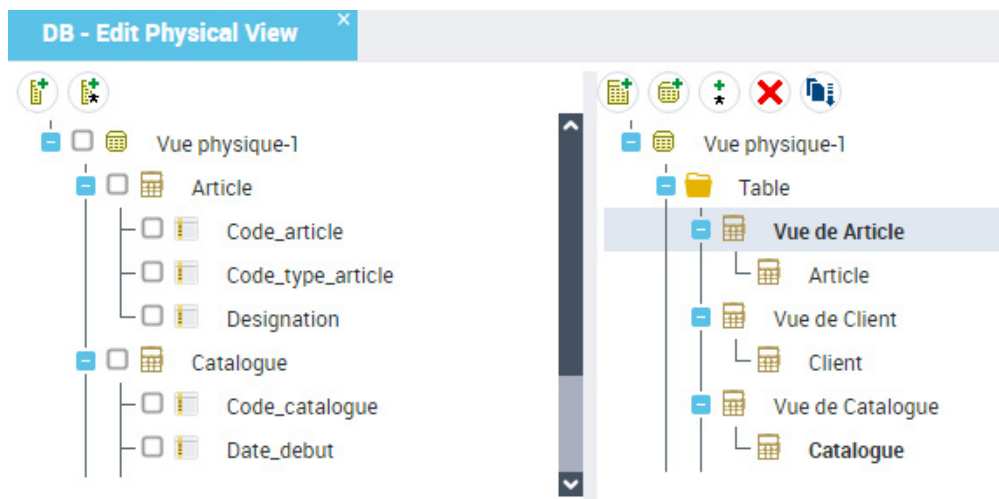
La création des *vues* d'une base de données se fait par l'intermédiaire d'un arbre, qui permet de générer automatiquement une partie de la définition de la vue, et peut être complétée par l'utilisateur.

### 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**.

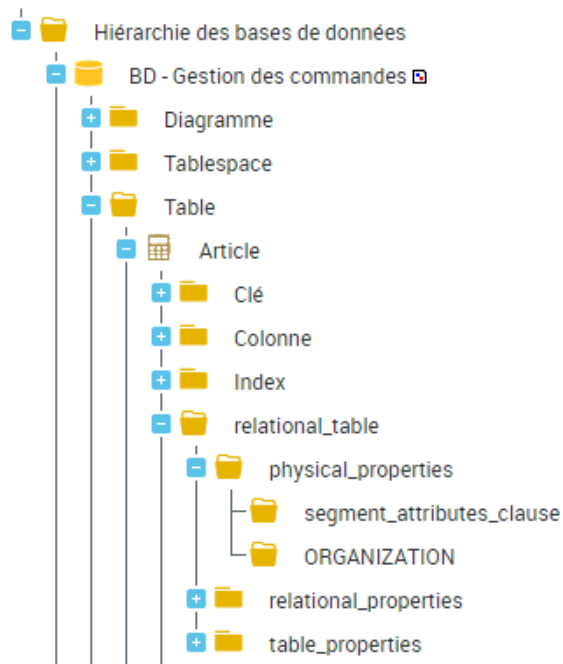
➡ Dans **Hopex Data Architecture**, cliquez sur le menu de navigation puis sur **Architecture de données > Données physiques**.

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 :

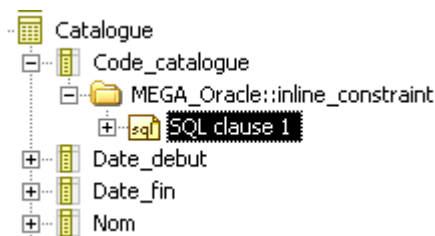
1. Cliquez avec le bouton droit sur le groupe de paramètres correspondant et sélectionnez **Nouveau > SQL clause**.
2. Ouvrez la fenêtre de propriétés de la clause et saisissez la valeur du paramètre à définir.

## Relier une clause

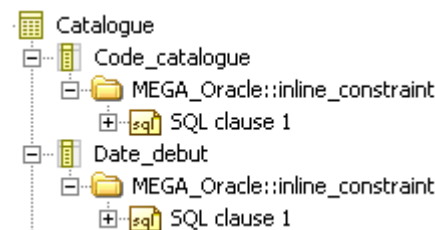
Vous pouvez affecter une même clause à plusieurs objets, à condition de relier le bon type de clause.

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

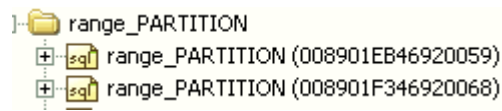
Vous avez la possibilité de donner un nom spécifique à une clause. Dans ce cas, le nom de la clause devient statique; le changement de clause type ne modifie pas le nom de la clause.

☛ 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.



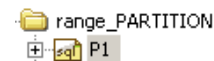
## Nommage à partir d'une propriété

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	

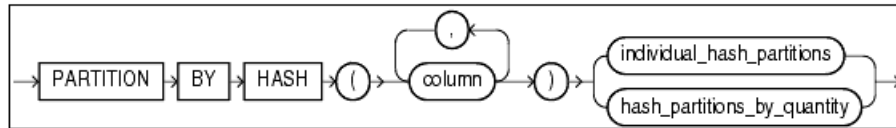


Le nom des clauses SQL n'est pas pris en compte dans la génération SQL. Dans l'exemple fourni, c'est bien la valeur de la propriété PARTITION qui alimente les scripts SQL générés.

## 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**.

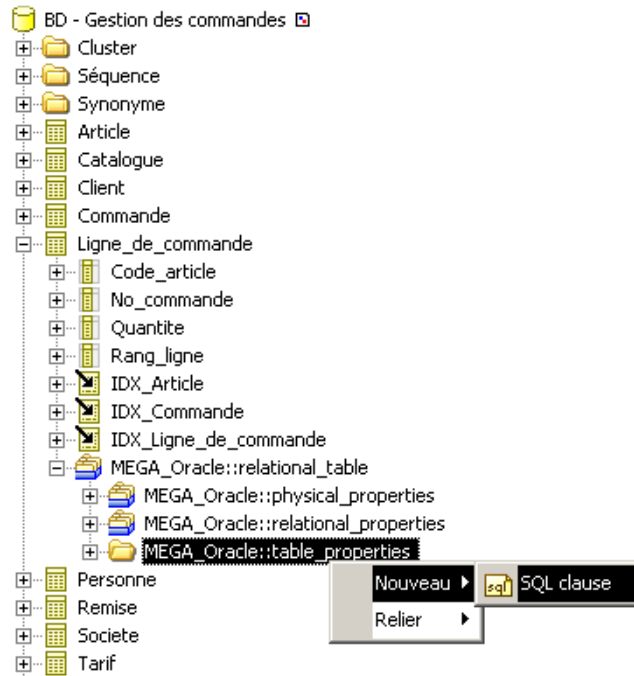
Affichez les propriétés physiques de la base de données "Gestion des commandes" :

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.

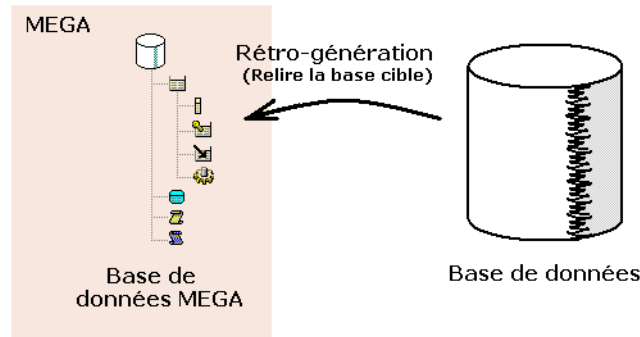
Cette rétro-génération est possible par extraction préalable. Voir [Extraire la description de schémas de bases de données à partir de sources de données](#).

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** :

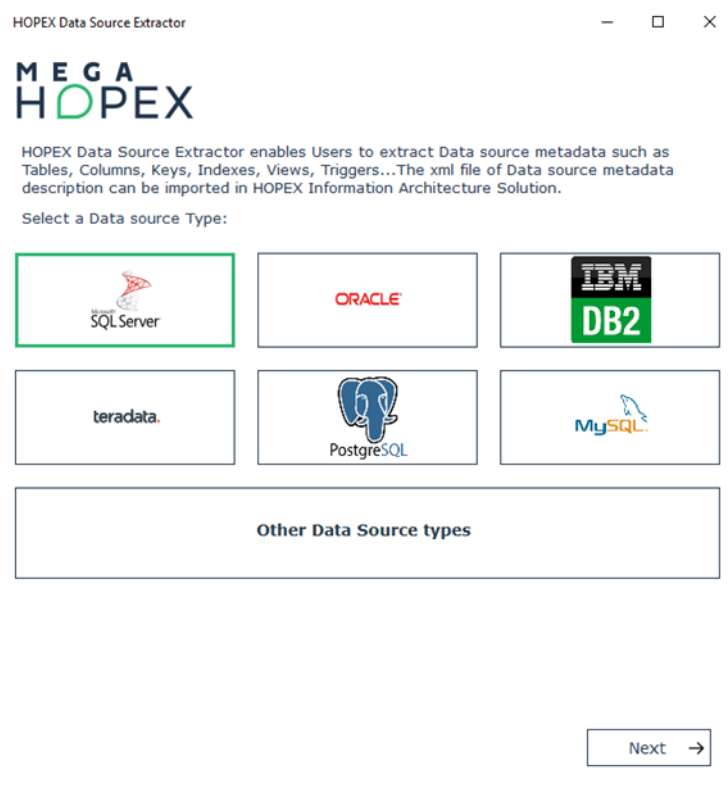
1. Lancez l'outil **ODBC Data Source Administrator**.
2. Cliquez sur l'onglet **Drivers**.
3. Sélectionnez le pilote et cliquez sur **OK**.

➡ 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**.*

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.

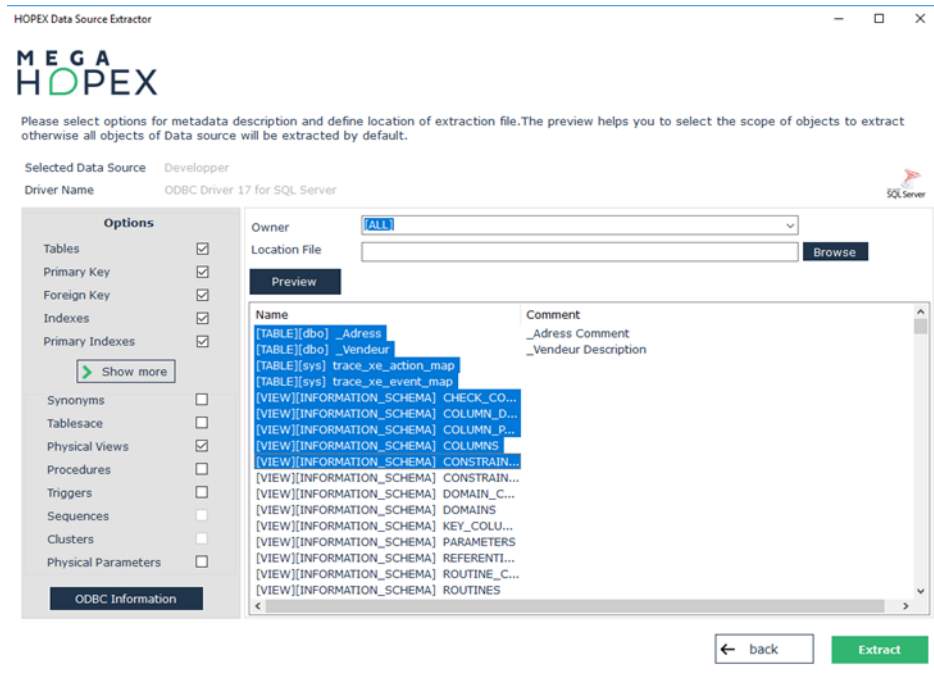
Le champ **Destination file** permet de préciser le chemin et le nom du fichier d'extraction ; le bouton **Browse** permet de parcourir les dossiers.

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.

▼ <XMDB NameId="Xmdb" Version="2.0">
  <DATABASE NameId="Database" DatabaseName="Airport Services">
    <QUALIFIER NameId="QUALIFIER.SQLPrmSpecific032Qualifier" QualifierValue="" </QUALIFIER>
    ▼ <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<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<0	NATIONAL VARCHAR BINARY
P-Numeric	L=0 or L $\emptyset$	NUMERIC
	L>0 and D $\emptyset$	NUMERIC (@L)
	L>0 and D not $\emptyset$	NUMERIC (@L,@D)
P-Real	L=0 or L $\emptyset$	REAL
	L>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	$L < 0$	VARBINARY
	L is numeric and $L \geq 0$	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 ∅	timestamp
	L <> 0	timestamp(@L)
P-Varchar	L=0 or L ∅	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



# RAPPORTS FOURNIS DANS HOPEX DATA ARCHITECTURE



**Hopex Data Architecture** 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 Architecture**.

---

## Accéder aux rapports

Pour accéder aux rapports d'**Hopex Data Architecture** :

- 1 Dans la barre de navigation, cliquez sur **Rapports**.

Certains rapports d'analyse sont 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 **Rapports** > **Rapports de description** > **Carte de domaines de données** sont également accessibles dans les propriétés d'une carte de domaines de données.

---

## Rapports de description

### Rapport de vue

Voir [Le rapport de vue](#).

### Rapport de glossaire

**Hopex Data Architecture** propose un rapport de glossaire prêt à l'emploi pour construire 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	<p><b>Élément de</b></p> <p>1. (Synonyms) Media library member (Hyponymes) Souscripteur</p> <p><b>Etats</b></p> <p>1. (Synonyms) Media library member, Media library subscriber (Produits d'architecture définis) <input type="checkbox"/> Abonné</p>
Achat d'ouvrage	<p><b>Événement</b></p> <p>1. (Synonyms) Publication purchase</p>
Adhérent médiathèque	<p><b>Etats</b></p> <p>1. (Synonyms) Media library member, Media library subscriber (Produits d'architecture définis) <input type="checkbox"/> Abonné</p>

## 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

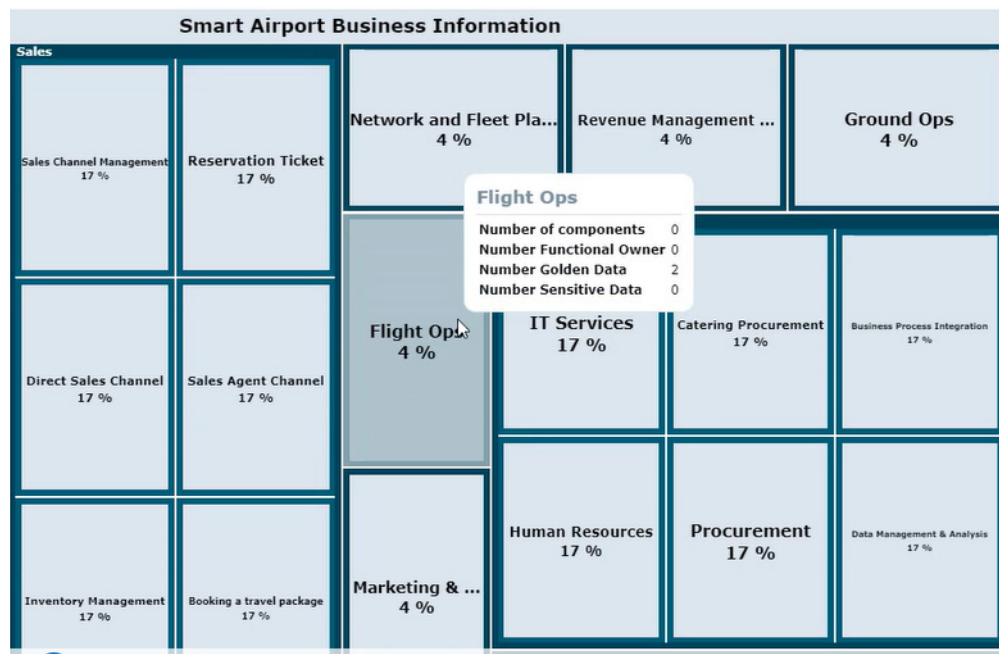
Dans ce rapport des filtres permettent de visualiser :

- 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 également les domaines qui composent la carte. Pour chaque domaine sont disponibles les informations suivantes :

- le nombre de données sensibles
- le nombre de données de référence
- la moyenne de la qualité des données

## Graphe de réalisation

Voir [Afficher le graphe de réalisation](#).

## Dépendances entre domaines de données

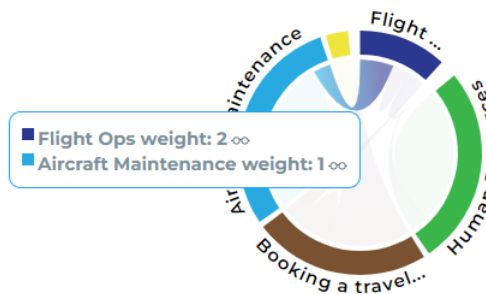
A partir d'une carte de données, ce rapport présente sous forme de graphique à cordes les dépendances structurelles des données utilisées entre les domaines de données.

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".



## Dendrogramme des catégories de données

Le rapport **Dendrogramme des catégories de données** vous permet de visualiser où sont utilisées les données d'une catégorie dans l'inventaire des applications.

Pour lancer ce rapport :

1. Cliquez sur le menu de navigation puis sur **Rapports > Rapports de description**.
2. Dans la zone d'édition, cliquez sur la tuile **Dendrogramme des catégories de données**.
3. Sélectionnez les catégories de données concernées.
4. Rafraîchissez le rapport.

---

## 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 d'information métier 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 événement 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.

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

---

## **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 Architecture** fournit différents types de Report DataSets.

### **Créer un Report DataSet**

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 'Matrice' (Matrix) report configuration interface. On the left, there are dropdown menus for 'Ligne:' (set to 'Concept'), 'Colonne:' (set to 'Term'), 'Cell Display:' (set to 'Value'), 'Appliquer le calcul sur:' (set to 'Concept'), and 'Calculer:' (set to 'Count'). The main area shows a matrix with columns for terms: 'Ville', 'Birthday', 'Book subscription', 'Abandonment subscription', 'Work', 'Abandon inscription', 'compte cible', 'Available Type of Loan', and 'Virement Planifié'. Rows are listed on the left: 'Book subscription (EN)', 'Work (EN)', 'compte cible', and 'Virement Planifié'. The matrix cells contain counts, with '1' appearing in several cells. On the right, the 'Propriétés de Report DataSet-3' panel shows the 'Données' (Data) tab. It includes a 'Term List' with checkboxes for various terms, a 'Report DataSet' section with buttons for 'Rafraîchir', 'PDF', 'Excel', and 'Rapport instantané', and a table listing terms and their concepts.

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.](#)
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.](#)
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

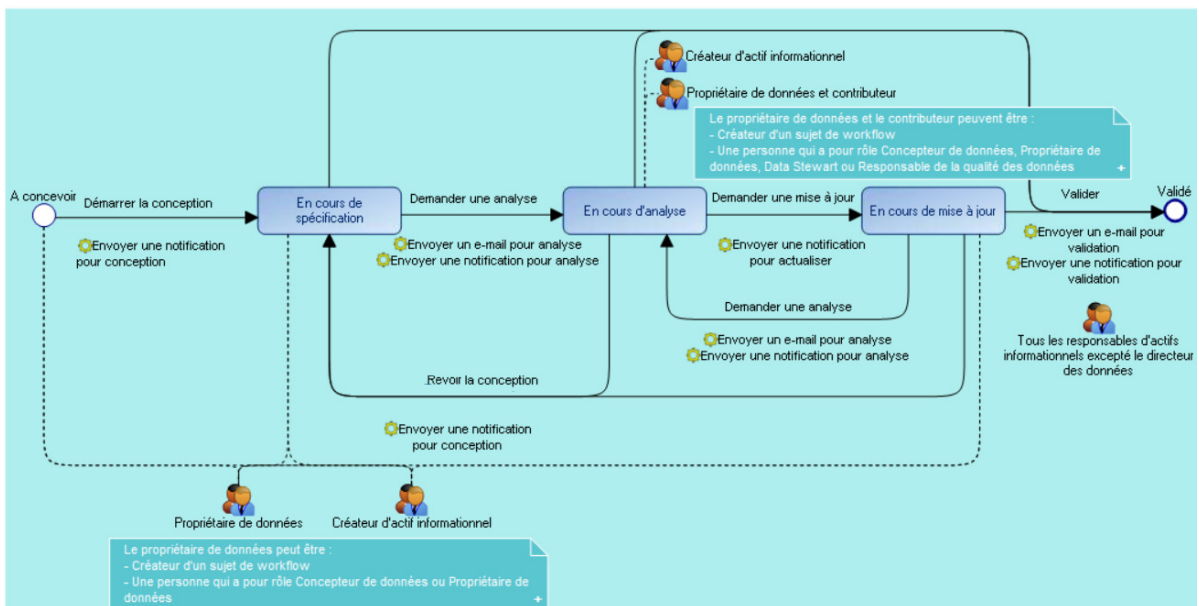
**Hopex Data Architecture** comprend un workflow standard pour gérer la progression de la conception d'actifs informationnels, de leur création jusqu'à la fin de leur validation.

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 optionnelle et non automatique. 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 Data 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 Architecture** fournit 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

**Hopex Data Architecture** fournit un modèle de fichier Excel pour 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 **Hopex Data Architecture**, 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 :

- Business Dictionary
- Data Category
- Term
- Concept
- Synonym
- Hypernym
- Component
- State concept

### FeuilleTerm

La feuille **Term** permet d'importer un ensemble de termes avec leur nom, la langue et leur définition.

Elle contient les colonnes suivantes.

**Term\_Ident**

Cette propriété permet d'identifier le terme lorsqu'il est référencé dans d'autres feuilles du fichier. Si seule la feuille **Term** est utilisée dans le fichier Excel, il n'est pas obligatoire de définir cette propriété.

**Term\_Name**

Cette propriété définit le nom du terme.

**Business Dictionary**

Dans cette colonne vous devez indiquer le nom du dictionnaire métier dans lequel vont être créés les Termes, Concepts et autres objets métier importés.

Il n'est pas possible d'indiquer des noms de dictionnaire métier différents dans un même fichier Excel, de même il est important de saisir le même nom dans toutes les lignes d'objets des feuilles à importer.

**Language**

Cette propriété indique l'abréviation de la langue associée au terme, par exemple FR, EN, etc. Cette abréviation est utilisée pour identifier la langue de l'objet dans Hopex.

Lorsque vous cliquez dans la colonne correspondante, une liste de langues est proposée.

**Text Definition**

Cette propriété contient la définition des termes et est utilisée pour créer dans Hopex les concepts qui correspondent aux termes saisis dans la feuille.

La propriété **Text Definition** est à compléter lorsque seule la feuille **Term** est utilisée ou bien lorsque le terme concerné dispose d'une seule définition. Si le terme dispose de plusieurs définitions, celles-ci doivent être déclarées au niveau de la feuille **Concept**. Ainsi chacun des concepts référence le terme associé (via son identifiant déclaré dans la feuille **Term**) et porte sa définition dans la propriété **Text Définition** de la feuille **Concept** (voir plus bas l'exemple de la feuille Concept).

Exemple de feuille Term :

B	C	D	E	
Term_Name	Term_Ident	Business Dictionary	Language	Text Definition
0000000040000063	9B089CFF5C947952	C69DCA055C931225	C69DC9445C9311DC	869F9ABB5CA43A1E
Order	T0	Import Business Glossary	EN	
club	T1	Import Business Glossary	EN	
society	T2	Import Business Glossary	EN	

**Feuille Concept**

La feuille **Concept** permet de relier des concepts aux termes définis dans la feuille **Term**, et de donner leur définition.

### **Concept\_Ident**

Cette propriété identifie le concept qui correspond au terme.

### **Term\_Ident**

Cette propriété identifie le terme dont est issu le concept et qui est défini dans la feuille **Term**.

### **Concept\_Name**

Cette propriété est optionnelle ; elle est utilisée à titre d'information. Le nom du concept est celui du terme associé.

### **Text Definition**

Cette propriété contient la définition du terme dont est issu le concept.

Exemple :  
La feuille ci-dessous donne l'ensemble des concepts et définitions associés au terme "Order" (défini dans la feuille des termes avec l'identifiant "T0") :

Business Dictionary	Concept Name	Definition	Properties
D81D0FEC64553EDF	D81D10BF64553F7F	D81D10D464553FB0	D81D10F164553FE1
r 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
r 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

## **Feuille Synonym**

La feuille **Synonym** permet de relier des termes définis dans la feuille **Term** à des définitions saisies dans la feuille **Concept**, en désignant les termes comme synonymes de la définition (la définition étant portée par un concept).

### **Concept\_Ident**

Cette propriété identifie le concept qui correspond au terme.

### **Concept\_Name**

Cette propriété est optionnelle et fournie à titre d'information. Elle donne le nom du concept.

### **Term\_Ident**

Cette propriété identifie le terme associé au synonyme, qui est défini dans la feuille **Term**.

### ***Term\_Name***

Cette propriété est optionnelle et fournie à titre d'information, elle donne le nom du terme.

## **Feuille Component**

La feuille **Component** permet de définir les relations entre les concepts.

### ***Concept\_Component\_Ident***

Cette propriété identifie les composants de concept.

### ***Term\_Ident***

Cette propriété est à définir si le composant doit désigner le terme. Par défaut le nom du composant est initialisé à partir du concept référencé.

### ***Owner\_Concept\_Ident***

Cette propriété est obligatoire ; elle identifie le concept détenteur.

### ***Referenced\_Concept\_Ident***

Cette propriété est obligatoire ; elle identifie le concept référencé.

## **Feuille State Concept**

La feuille **State Concept** permet d'importer un ensemble d'états de concept associés à des termes.

### ***Concept\_State\_Ident***

Cette propriété identifie l'état de concept à importer.

### ***Term\_Ident***

Cette propriété identifie le terme associé, qui est défini dans la feuille **Term**.

### ***Concept\_State\_Name***

Cette propriété est optionnelle et fournie à titre d'information. Le nom de l'état de concept est issu du terme qui lui correspond.

### ***Text Definition***

Cette propriété contient la définition du terme associé.

### ***StateOf\_Ident***

Cette propriété identifie le concept de l'état. Si cette propriété est nulle, l'état de concept est créé sans concept.

Voir aussi : [Importer des données logiques à partir d'un fichier Excel](#).

# IMPORTER DES DONNÉES LOGIQUES À PARTIR D'UN FICHIER EXCEL

**Hopex Data Architecture** fournit 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 **Hopex Data Architecture**, 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

<input type="checkbox"/>	Nœud d'évaluation	Date d'évaluat...	Exhaustivité	Unicité	Fraicheur	Validité	Exactit...	Consistance
<input type="checkbox"/>	 Artifact (EN) (Concept)	21/08/2020	<div><div></div>Very Low</div>	<div><div></div>Low</div>	<div><div></div>Medium</div>	<div><div></div>Very High</div>	<div><div></div>High</div>	<div><div></div>Medium</div>

---

## 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**.

