

Version 9.1

*IBM InfoSphere Optim  
Utilisation d'Optim Designer*

**IBM**



Version 9.1

*IBM InfoSphere Optim  
Utilisation d'Optim Designer*

**IBM**

**Important**

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 129.

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.can.ibm.com> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France  
Direction Qualité  
17, avenue de l'Europe  
92275 Bois-Colombes Cedex*

© Copyright IBM Corporation 1996, 2012.

# Table des matières

## Avis aux lecteurs canadiens . . . . . v

## Chapitre 1. Utilisation d'InfoSphere Optim Designer . . . . . 1

Mise en route . . . . .	1
Perspective Optim . . . . .	1
Utilisation d'Optim Manager à partir d'Optim Designer. . . . .	4
Données d'échantillon . . . . .	4
Base de données prise en charge . . . . .	13
Fonctions d'accessibilité . . . . .	13

## Chapitre 2. Gestion des connexions de source de données . . . . . 15

Utilisation d'un alias de magasin de données . . . . .	15
Définition d'un alias de magasin de données . . . . .	15
Edition d'un alias de magasin de données . . . . .	15
Connexion à un alias de magasin de données . . . . .	16
Utilisation du référentiel Optim . . . . .	16
Connexion au référentiel Optim . . . . .	16
Edition d'une connexion à un référentiel. . . . .	16
Changement d'une connexion à un référentiel . . . . .	16
Association d'un répertoire Optim au référentiel . . . . .	17
Transformation d'une demande en service . . . . .	17

## Chapitre 3. Gestion des modèles de données . . . . . 19

Utilisation des définitions d'accès . . . . .	19
Création d'une définition d'accès . . . . .	20
Gestion des tables dans une définition d'accès. . . . .	20
Gestion des critères de sélection . . . . .	22
Gestion des relations dans une définition d'accès . . . . .	23
Gestion des variables dans une définition d'accès . . . . .	25
Gestion des listes de pointage et de découpage . . . . .	26
Utilisation des relations Optim . . . . .	26
Règles de compatibilité pour les relations . . . . .	28
Création d'une relation Optim . . . . .	30
Modification d'une relation Optim. . . . .	30
Utilisation des clés primaires Optim . . . . .	32
Création d'une clé principale . . . . .	33
Edition d'une clé primaire . . . . .	33
Utilisation des mappes de table. . . . .	34
Création d'une mappe de table . . . . .	34
Modification d'une mappe de table . . . . .	35
Utilisation des mappes de colonne. . . . .	36
Règles de compatibilité pour les mappes de colonne. . . . .	37
Création d'une mappe de colonne . . . . .	38
Edition d'une mappe de colonne . . . . .	38

## Chapitre 4. Conception de services de gestion des données . . . . . 41

Utilisation des services d'extraction . . . . .	41
Création d'un service d'extraction . . . . .	41

Modification d'un service d'extraction . . . . .	42
Utilisation des services de conversion . . . . .	44
Création d'un service de conversion . . . . .	44
Modification d'un service de conversion. . . . .	45
Utilisation des services d'insertion. . . . .	45
Création d'un service d'insertion . . . . .	46
Edition d'un service d'insertion. . . . .	46
Utilisation des services de chargement . . . . .	48
Création d'un service de chargement . . . . .	49
Modification d'un service de chargement . . . . .	50
Test d'un service de gestion des données . . . . .	51

## Chapitre 5. Masquage des données . . . 53

Application d'une fonction de masquage des données . . . . .	53
Fonctions de masquage des données . . . . .	53
Fonctions de recherche . . . . .	53
Fonction de mélange . . . . .	63
Fonction TRANS SSN . . . . .	65
Fonction TRANS CCN. . . . .	68
Fonction TRANS EML. . . . .	71
Fonction TRANS COL. . . . .	75
TRANS NID . . . . .	77
Fonction de sous-chaîne . . . . .	88
Fonction aléatoire . . . . .	89
Fonction séquentielle . . . . .	89
Fonction d'identité ou de série . . . . .	90
Fonction de séquence Oracle. . . . .	91
Fonction Propager la valeur de clé principale ou externe . . . . .	92
Expressions concaténées . . . . .	93
Expressions numériques . . . . .	94
Fonctions littérales et de valeur. . . . .	94
Fonction Age . . . . .	95
Fonction Devise . . . . .	98
Nom d'e-mail généré automatiquement. . . . .	100
Nom d'e-mail formaté . . . . .	100
Fonction de numérotation aléatoire . . . . .	101
Fonction de numérotation séquentielle . . . . .	101
Utilisation des routines d'exit . . . . .	101
Ecriture de routines d'exit . . . . .	102
Routine d'exit standard . . . . .	103
Exit de format source. . . . .	105
Exit de format cible . . . . .	107
Ecriture de procédures de mappe de colonne avec le scriptage Lua . . . . .	110
Fonctions LUA pour les procédures de mappe de colonne . . . . .	110
Limitations des procédures de mappe de colonne . . . . .	112
Exemple de procédure de mappe de colonne : Procédure générique . . . . .	112
Exemple de procédure de mappe de colonne : consultation commutée . . . . .	114

Exemple de procédure de mappe de colonne :	
Masquage du numéro de carte d'identité . . . .	117

**Chapitre 6. Gestion de données . . . . 119**

Exploration des données. . . . .	119
Edition de données . . . . .	119
Comparaison de données . . . . .	120
Définition d'une demande de comparaison . . .	120
Edition d'une demande de comparaison . . .	121
Exécution d'un processus de comparaison . . .	121
Création de tables . . . . .	122

**Chapitre 7. Gestion de données avec Optim Designer . . . . . 123**

Extraction de données avec Optim Designer . . .	123
Création d'un dossier dans l'explorateur de référentiel . . . . .	123

Création d'une définition d'accès . . . . .	123
Définition des critères de sélection . . . . .	124
Création d'un service d'extraction . . . . .	125
Masquage de données avec Optim Designer . . .	125
Création d'une mappe de table . . . . .	126
Modification d'une mappe de table . . . . .	126
Création d'une mappe de colonne . . . . .	126
Application d'une fonction de masquage des données . . . . .	127
Ajout d'une mappe de colonne à une mappe de table . . . . .	128
Création d'un service de conversion . . . . .	128

**Index . . . . . 133**

---

## Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

### Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

### Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

### Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.

### OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

### Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

<b>France</b>	<b>Canada</b>	<b>Etats-Unis</b>
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

### Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

### Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

---

# Chapitre 1. Utilisation d'InfoSphere Optim Designer

IBM® InfoSphere Optim Designer vous permet de définir des modèles de données, des règles de confidentialité des données, des services de gestion des données et des magasins de données. Vous pouvez définir des objets dans Optim Designer et les partager dans le référentiel Optim. Utilisez l'explorateur de référentiel pour gérer les objets du référentiel. Utilisez l'explorateur de répertoires pour afficher des objets et transformer des requêtes dans un répertoire Optim.

## Services de gestion des données

Utilisez un service de gestion des données pour extraire, convertir, charger ou insérer des données. Un service utilise des définitions d'accès pour définir les données à extraire. Un service utilise des mappes de table et de colonne pour mapper des données source et cible. Vous pouvez également transformer les requêtes de processus d'un répertoire Optim en services de gestion des données. Utilisez Optim Manager en mode imbriqué pour tester les services.

## Masquage des données

Vous pouvez masquer les données traitées par un service de gestion des données. Utilisez le masquage des données pour transformer les données telles que les numéros nationaux d'identité, les numéros de carte de crédit, les dates, les valeurs numériques et les informations personnelles. Utilisez une mappe de colonne pour entrer une fonction de masquage de données ou créer une procédure de mappe de colonne avec un script LUA.

## Définitions d'accès

La définition d'accès vous permet de définir des modèles et des sous-ensembles de données dans des schémas similaires ou différents. Une définition d'accès permet d'identifier les critères de sélection et les relations d'un service de gestion de données. Vous pouvez définir des relations en dehors d'un SGBD et utiliser le code SQL pour définir des critères de sélection.

## Optim Manager

Vous pouvez ouvrir Optim Manager à partir d'Optim Designer (mode imbriqué), ce qui vous permet de tester les services de gestion des données.

---

## Mise en route

Pour démarrer, vous devez utiliser l'explorateur de référentiel pour vous connecter à un référentiel Optim.

## Perspective Optim

Dans Optim Designer, la perspective Optim fournit les outils nécessaires pour définir des modèles de données et des services de gestion des données et pour se connecter à un répertoire Optim. Lorsque vous ouvrez Optim Designer pour la première fois, la perspective Optim s'affiche par défaut.

La perspective Optim comprend les vues suivantes :

### Explorateur de référentiel

Utilisez l'explorateur de référentiel pour définir des alias de magasin de données, des définitions d'accès, des services de gestion de données et d'autres objets stockés dans un référentiel Optim.

### Explorateur de répertoire

Utilisez l'explorateur de répertoires pour vous connecter à un répertoire Optim.

Pour ouvrir la perspective Optim, cliquez sur **Fenêtre > Ouvrir la perspective > Autre**. Dans la fenêtre Ouvrir la perspective, sélectionnez **Optim**.

## Explorateur de référentiel

Utilisez l'explorateur de référentiel pour définir des alias de magasin de données, des définitions d'accès, des services de gestion de données et d'autres objets stockés dans un référentiel Optim.

Lorsque Optim Designer est connecté à un référentiel Optim, l'explorateur de référentiel comprend les objets suivants :

### Alias de magasin de données

Un alias de magasin de données est un objet défini par l'utilisateur et associé à une base de données. Lorsque vous définissez un alias de magasin de données, vous fournissez des paramètres utilisés par Optim pour communiquer avec cette base de données. Ces paramètres incluent le type et la version du système de gestion des bases de données (DBMS) et les propriétés d'une connexion client et d'une connexion JDBC.

Un nom d'alias de magasin de données sert de qualificatif de haut niveau, qui vous permet d'accéder à une base de données spécifique pour exécuter les fonctions demandées. Par exemple, dans une définition d'accès, vous devez qualifier le nom d'une table avec un nom d'alias de magasin de données. L'alias du magasin de données référencé fournit les paramètres nécessaires à la connexion à la base de données dans laquelle réside la table.

### Clés primaires Optim

Une clé primaire est la/les colonne(s) contenant des valeurs identifiant de manière unique chaque ligne d'une table. Une table de base de données doit posséder une clé primaire pour Optim afin d'insérer, de mettre à jour, de restaurer ou de supprimer des données à partir d'une table de base de données. Optim utilise des clés primaires définies dans la base de données. Toutefois, vous pouvez également définir des clés primaires Optim pour compléter les clés primaires dans la base de données.

### Relations Optim

Optim utilise des relations pour déterminer les données à extraire des tables associées et se base sur les relations définies dans la base de données (si disponibles). Toutefois, vous pouvez également définir des relations Optim pour compléter les relations dans la base de données.

### Définitions d'accès

Utilisez des définitions d'accès pour définir les tables, la traversée de relation et les critères de sélection des données à traiter.

### Mappes de colonne

Une mappe de colonne fournit les spécifications nécessaires pour faire correspondre ou exclure des colonnes du traitement dans un service de gestion de données. La conversion, l'insertion et le chargement des services doivent référencer une mappe de table, qui peut faire référence à une ou plusieurs mappes. Vous pouvez également utiliser une mappe de colonne pour définir les transformations de données avec des fonctions ou des procédures de mappe de colonne.

### Mappes de table

Utilisez une mappe de table pour définir les spécifications des tables source et cible correspondantes contenant des données compatibles. Vous pouvez mapper des tables qui possèdent des noms différents, modifier les noms de table, exclure des tables d'un processus, ou inclure des mappes de colonne afin de mieux contrôler les données.

### Services

Utilisez un service de gestion des données pour extraire, convertir, charger ou insérer des données. Un service utilise des définitions d'accès pour définir les données à extraire. Un service utilise des mappes de table et de colonne pour mapper des données source et cible. Vous pouvez masquer les données en appliquant une règle de confidentialité des données à une entité traitée par un service. Vous pouvez également transformer les requêtes de processus d'un répertoire Optim en services de gestion des données. Utilisez Optim Manager en mode imbriqué pour tester les services.

### Dossiers

Un dossier d'explorateur de référentiel contient des services, des définitions d'accès, des mappes de colonne et des mappes de table. Utilisez les dossiers pour organiser ces objets.

### Création d'un dossier pour l'explorateur de référentiel :

Utilisez la fenêtre Nouveau dossier pour créer un dossier dans l'explorateur de référentiel.

Un dossier d'explorateur de référentiel contient des services, des définitions d'accès, des mappes de colonne et des mappes de table. Utilisez les dossiers pour organiser ces objets.

Pour créer un dossier dans l'explorateur de référentiel.

1. En fonction de l'existence d'un dossier, procédez selon l'une des méthodes suivantes :
  - a. Pour créer un dossier initial, cliquez sur **Cliquer pour créer un nouveau dossier** dans l'explorateur de référentiel.
  - b. Pour créer un dossier supplémentaire, cliquez avec le bouton droit de la souris sur la vue Explorateur de référentiel et cliquez sur **Nouveau > Dossier**.

La fenêtre Nouveau dossier s'ouvre.

2. Entrez un nom de dossier, puis cliquez sur **OK**.

Le nouveau dossier s'affiche dans l'explorateur de référentiel.

### Explorateur de répertoire

Utilisez l'explorateur de répertoires pour vous connecter à un répertoire Optim. Seul un répertoire Optim peut être associé au référentiel.

Utilisez l'explorateur de répertoires pour transformer les demandes de processus dans des services de gestion de données.

Les définitions suivantes sont disponibles.

Icône	Définition
	définition d'accès
	demande d'archivage
	colonne
	mappe de colonne
	demande de conversion
	ID de créateur
	alias de base de données
	demande de suppression
	demande d'extraction
	demande d'insertion
	demande de chargement

Icône	Définition
	clé primaire
	relation
	demande de restauration
	table
	mappe de table
	variable

## Utilisation d'Optim Manager à partir d'Optim Designer

Utilisez Optim Manager à partir d'Optim Designer pour tester les services de gestion de données et les services d'interopérabilité Optim. L'utilisation d'Optim Manager à partir d'Optim Designer est également appelée Optim Manager en mode imbriqué.

Optim Manager est une application Web que vous pouvez utiliser pour tester les services. Optim Manager s'affiche dans un navigateur interne fourni par Optim Designer ou dans un navigateur externe. Vous pouvez sélectionner le navigateur qu'Optim doit utiliser pour Optim Manager en cliquant sur **Fenêtre > Préférences > Général > Navigateur Web** dans Optim Designer.

## Ouverture d'Optim Manager à partir d'Optim Designer

Vous pouvez ouvrir Optim Manager en testant un service. Vous pouvez également ouvrir Optim Manager en saisissant l'adresse URL suivante dans un navigateur Web : `http://localhost:numéro_port/console`, où *numéro\_port* est le numéro de port affecté à Optim Manager. Le numéro de port par défaut est 60000.

Si le navigateur affiche un message "Page introuvable" lorsque vous ouvrez Optim Manager, un conflit de port peut exister et vous devez changer le numéro de port d'Optim Manager.

Vous pouvez également ouvrir Optim Manager depuis la barre d'outils d'Optim Designer.

## Changement du numéro de port d'Optim Manager

Pour modifier le numéro de port, vous devez modifier la propriété suivante dans le fichier `eclipse.ini` situé dans le répertoire d'installation d'Optim Designer :

```
-Dorg.eclipse.equinox.http.jetty.http.port=numéro_port
```

où *numéro\_port* est le nouveau numéro de port d'Optim Manager. Si Optim Designer est ouvert, vous devez redémarrer l'application pour appliquer le nouveau numéro de port.

## Données d'échantillon

Optim fournit des données d'échantillon pour le masquage des données ainsi que des sources de données source et cible prédéfinies.

## Exemples de structure et de tables de base de données

En tant que groupe, les tables de base de données comprennent des informations sur les clients et les commandes, avec des instructions sur l'expédition. Les exemples de tables comprennent également des

informations sur les ventes et l'inventaire. Il existe des différences mineures dans les types de données, selon le SGBD que vous utilisez pour installer les exemples de tables. Le diagramme suivant présente la structure de base d'un exemple de base de données.

En tant que groupe, les tables de base de données comprennent des informations sur les clients et les commandes, avec des instructions sur l'expédition. Les exemples de tables comprennent également des informations sur les ventes et l'inventaire. Il existe des différences mineures dans les types de données, selon le SGBD que vous utilisez pour installer les exemples de tables.

Un ensemble de tables supplémentaire est également installé avec l'exemple de base de données. Les tables de l'ensemble supplémentaire possèdent les mêmes noms que les tables du premier ensemble et portent le suffixe "2". Les quatre tables de cet ensemble supplémentaire sont les suivantes :

- OPTIM\_CUSTOMERS2
- OPTIM\_ORDERS2
- OPTIM\_DETAILS2
- OPTIM\_ITEMS2

Les tables de l'ensemble supplémentaire ne contiennent pas de données. Elles permettent de démontrer les fonctions dans Optim.

#### **Table OPTIM\_SALES :**

La table OPTIM\_SALES identifie chaque vendeur par son nom, son numéro d'identificateur et son directeur.

La table OPTIM\_SALES contient les colonnes suivantes :

##### **SALESMAN\_ID**

CHAR ; jusqu'à 6 caractères ; ne peut pas contenir une valeur nulle.

##### **FIRST\_NAME**

VARCHAR ; jusqu'à 15 caractères ; ne peut pas contenir une valeur nulle.

##### **LAST\_NAME**

VARCHAR ; jusqu'à 15 caractères ; ne peut pas contenir une valeur nulle.

##### **NATIONALITY**

VARCHAR ; jusqu'à 30 caractères

##### **NATIONAL\_ID**

VARCHAR ; jusqu'à 30 caractères

##### **PHONE\_NUMBER**

VARCHAR ; jusqu'à 20 caractères ; peut contenir une valeur nulle.

##### **EMAIL\_ADDRESS**

VARCHAR ; jusqu'à 70 caractères ; ne peut pas contenir une valeur nulle.

**AGE** SMALLINT ; ne peut pas contenir une valeur nulle ; possède une valeur par défaut.

**SEX** CHAR ; 1 caractère ; ne peut pas contenir une valeur nulle ; possède une valeur par défaut.

##### **TERRITORY**

VARCHAR ; jusqu'à 14 caractères ; ne peut pas contenir une valeur nulle.

##### **MANAGER\_ID**

VARCHAR ; jusqu'à 6 caractères.

#### **Clés primaires**

La colonne SALESMAN\_ID est la colonne de clé principale.

## Relations avec les autres tables

La table OPTIM\_SALES est parent de :

- la table OPTIM\_CUSTOMERS, via une clé externe sur la colonne SALESMAN\_ID ;
- la table OPTIM\_MALE\_RATES, via une relation menée par les données Optim avec la colonne AGE lorsque SEX = 'M'.
- la table OPTIM\_FEMALE\_RATES, via une relation menée par les données Optim avec la colonne AGE lorsque SEX = 'F' ;
- la table OPTIM\_STATE\_LOOKUP, via une relation de sous-chaîne Optim grâce à SUBSTR(SALESMAN\_ID,1,2).

### Table OPTIM\_CUSTOMERS :

La table OPTIM\_CUSTOMERS contient les noms de clients, les numéros d'identificateur et les adresses.

La table OPTIM\_CUSTOMERS contient les colonnes suivantes :

#### **CUST\_ID**

CHAR ; jusqu'à 5 caractères ; ne peut pas contenir une valeur nulle ; contient une contrainte de vérification.

#### **CUSTNAME**

CHAR ; jusqu'à 20 caractères ; ne peut pas contenir une valeur nulle.

#### **ADDRESS1**

VARCHAR ; jusqu'à 100 caractères ; ne peut pas contenir une valeur nulle.

#### **ADDRESS2**

VARCHAR ; jusqu'à 100 caractères ; ne peut pas contenir une valeur nulle.

#### **LOCALITY**

VARCHAR ; jusqu'à 56 caractères

**CITY** VARCHAR ; jusqu'à 60 caractères

#### **STATE**

VARCHAR ; jusqu'à 30 caractères

#### **COUNTRY\_CODE**

CHAR ; jusqu'à 2 caractères

#### **POSTAL\_CODE**

VARCHAR ; jusqu'à 15 caractères

#### **POSTAL\_CODE\_PLUS4**

CHAR ; jusqu'à 4 caractères ; peut contenir une valeur nulle.

#### **EMAIL\_ADDRESS**

VARCHAR ; jusqu'à 70 caractères

#### **PHONE\_NUMBER**

VARCHAR ; jusqu'à 20 caractères

#### **YTD\_SALES**

DECIMAL ; montant en dollars ; ne peut pas contenir une valeur nulle ; possède une valeur par défaut.

#### **SALESMAN\_ID**

CHAR ; jusqu'à 6 caractères

#### **NATIONALITY**

VARCHAR ; jusqu'à 30 caractères

**NATIONAL\_ID**  
VARCHAR ; jusqu'à 30 caractères

**CREDITCARD\_NUMBER**  
VARCHAR ; 19 caractères

**CREDITCARD\_TYPE**  
VARCHAR ; jusqu'à 30 caractères

**CREDITCARD\_EXP**  
CHAR ; 4 caractères

**CREDITCARD\_CVV**  
VARCHAR ; jusqu'à 4 caractères

**DRIVER\_LICENSE**  
VARCHAR ; jusqu'à 30 caractères

**CUSTOMER\_INFO**  
XMLTYPE

### Clés primaires

La colonne CUST\_ID est la colonne de clé principale.

### Relations avec les autres tables

La table OPTIM\_CUSTOMERS est parent de :

- la table OPTIM\_ORDERS, via une clé externe sur la colonne CUST\_ID ;
- la table OPTIM\_SHIP\_TO, via une relation Optim sur la colonne CUST\_ID.

La table OPTIM\_CUSTOMERS est un enfant de :

- la table OPTIM\_SALES, via sa clé externe sur la colonne SALESMAN\_ID.

### Table OPTIM\_ORDERS :

La table OPTIM\_ORDERS contient des informations pour les commandes, comprenant le numéro de commande, l'ID du client et le vendeur.

La table OPTIM\_ORDERS contient les colonnes suivantes :

**ORDER\_ID**  
DECIMAL ; numéro d'ID de commande ; ne peut pas contenir une valeur nulle.

**CUST\_ID**  
CHAR ; numéro d'ID du client ; jusqu'à 5 caractères ; ne peut pas contenir une valeur nulle.

**ORDER\_DATE**  
TIMESTAMP ; date de commande ; ne peut pas contenir une valeur nulle ; possède une valeur par défaut.

**ORDER\_TIME**  
TIMESTAMP ; heure du jour ; ne peut pas contenir une valeur nulle ; possède une valeur par défaut.

**FREIGHT\_CHARGES**  
DECIMAL ; montant en dollars

**ORDER\_SALESMAN**  
CHAR ; jusqu'à 6 caractères

**ORDER\_POSTED\_DATE**

TIMESTAMP ; ne peut pas contenir une valeur nulle ; possède une valeur par défaut.

**ORDER\_SHIP\_DATE**

CHAR ; date à laquelle la commande est envoyée ; jusqu'à 8 caractères ; ne peut pas contenir une valeur nulle ; possède une valeur par défaut.

**Clés primaires**

La colonne ORDER\_ID est la colonne de clé principale.

**Relations avec les autres tables**

La table OPTIM\_ORDERS est un parent de la table OPTIM\_DETAILS, via une clé externe sur la colonne ORDER\_ID.

La table OPTIM\_ORDERS est un enfant de la table OPTIM\_CUSTOMERS, via sa clé externe sur la colonne CUST\_ID.

**Table OPTIM\_DETAILS :**

La table OPTIM\_DETAILS contient des renseignements supplémentaires pour chaque commande de la table OPTIM\_ORDERS.

La table OPTIM\_DETAILS contient les colonnes suivantes :

**ORDER\_ID**

DECIMAL ; numéro d'ID de commande ; ne peut pas contenir une valeur nulle.

**ITEM\_ID**

CHAR ; jusqu'à 5 caractères ; numéro d'ID d'élément ; ne peut pas contenir une valeur nulle.

**ITEM\_QUANTITY**

DECIMAL ; nombre d'éléments ; ne peut pas contenir une valeur nulle.

**DETAIL\_UNIT\_PRICE**

DECIMAL ; prix unitaire ; montant en dollars ; ne peut pas contenir de valeur nulle.

**Clés primaires**

Les colonnes ORDER\_ID et ITEM\_ID sont la clé primaire.

**Relations avec les autres tables**

La table OPTIM\_DETAILS est un enfant de :

- la table OPTIM\_ORDERS, via sa clé externe sur la colonne ORDER\_ID ;
- la table OPTIM\_ITEMS, via sa clé externe sur la colonne ITEM\_ID.

**Table OPTIM\_ITEMS :**

La table OPTIM\_ITEMS contient des informations sur chaque élément pour une commande, comprenant la description, le prix et la quantité dans l'inventaire.

La table OPTIM\_ITEMS contient les colonnes suivantes :

**ITEM\_ID**

CHAR ; jusqu'à 5 caractères ; ne peut pas contenir une valeur nulle.

**ITEM\_DESCRIPTION**

VARCHAR ; jusqu'à 72 caractères ; ne peut pas contenir une valeur nulle.

**CATEGORY**

VARCHAR ; jusqu'à 14 caractères ; ne peut pas contenir une valeur nulle.

**RATING**

CHAR ; jusqu'à 4 caractères ; ne peut pas contenir une valeur nulle.

**UNIT\_PRICE**

DECIMAL ; montant en dollars ; ne peut pas contenir de valeur nulle.

**ON\_HAND\_INVENTORY**

INTEGER ; ne peut pas contenir de valeur nulle.

**Clés primaires**

La colonne ITEM\_ID est la colonne de clé principale.

**Relations avec les autres tables**

La table OPTIM\_ITEMS est un parent de la table OPTIM\_DETAILS, via une clé externe sur la colonne ITEM\_ID.

**Table OPTIM\_SHIP\_TO :**

La table OPTIM\_SHIP\_TO contient des informations d'expédition de commande.

La table OPTIM\_SHIP\_TO contient les colonnes suivantes :

**CUST\_ID**

CHAR ; jusqu'à 5 caractères ; ne peut pas contenir une valeur nulle.

**SHIP\_ID**

DECIMAL ; ne peut pas contenir de valeur nulle.

**ADDRESS1**

VARCHAR ; jusqu'à 100 caractères

**ADDRESS2**

VARCHAR ; jusqu'à 100 caractères

**LOCALITY**

VARCHAR ; jusqu'à 56 caractères

**CITY** VARCHAR ; jusqu'à 30 caractères

**STATE**

VARCHAR ; jusqu'à 30 caractères

**COUNTRY\_CODE**

CHAR ; abréviation à 2 caractères

**POSTAL\_CODE**

VARCHAR ; jusqu'à 15 caractères

**POSTAL\_CODE\_PLUS4**

CHAR ; 4 caractères

**IN\_CARE\_OF**

VARCHAR ; jusqu'à 31 caractères

**SHIPPING\_CHANGE\_DT**

TIMESTAMP ; ne peut pas contenir une valeur nulle ; possède une valeur par défaut.

## Clés primaires

La colonne SHIP\_ID est la colonne de clé principale.

## Relations avec les autres tables

La table OPTIM\_SHIP\_TO est un parent de la table OPTIM\_SHIP\_INSTR, via une relation Optim sur la colonne SHIP\_ID.

La table OPTIM\_SHIP\_TO est un enfant de la table OPTIM\_CUSTOMERS, via sa relation Optim avec la colonne CUST\_ID.

## Table OPTIM\_SHIP\_INSTR :

La table OPTIM\_SHIP\_INSTR contient des informations détaillées sur l'envoi de commandes.

La table OPTIM\_SHIP\_INSTR contient les colonnes suivantes :

### SHIP\_ID

DECIMAL

### SHIP\_INSTR\_ID

INTEGER

### ORDER\_SHIP\_INSTR

VARCHAR ; jusqu'à 254 caractères

### SHIP\_UPDATED

TIMESTAMP ; ne peut pas contenir une valeur nulle ; possède une valeur par défaut.

## Clés primaires

La colonne SHIP\_INSTR\_ID est la colonne de clé principale.

## Relations avec les autres tables

La table OPTIM\_SHIP\_INSTR est un enfant de la table OPTIM\_SHIP\_TO, via sa relation Optim avec la colonne SHIP\_ID.

## Table OPTIM\_MALE\_RATES :

La table OPTIM\_MALE\_RATES contient des taux d'assurance basés sur l'âge.

La table OPTIM\_MALE\_RATES contient les colonnes suivantes :

### AGE SMALLINT

### RATE\_PER\_1000

DECIMAL ; taux en montant en dollars

## Clés primaires

La colonne RATE\_PER\_1000 est la colonne de clé principale.

## Relations avec les autres tables

La table OPTIM\_MALE\_RATES est un enfant de la table OPTIM\_SALES, via sa relation menée par les données Optim avec la colonne AGE.

### **Table OPTIM\_FEMALE\_RATES :**

La table OPTIM\_FEMALE\_RATES contient des taux d'assurance basés sur l'âge.

La table OPTIM\_FEMALE\_RATES contient les colonnes suivantes :

**AGE** SMALLINT

**RATE\_PER\_1000**

DECIMAL ; taux en montant en dollars

#### **Clés primaires**

La colonne RATE\_PER\_1000 est la colonne de clé principale.

#### **Relations avec les autres tables**

La table OPTIM\_FEMALE\_RATES est un enfant de la table OPTIM\_SALES, via sa relation menée par les données Optim avec la colonne AGE.

### **Table OPTIM\_STATE\_LOOKUP :**

La table OPTIM\_STATE\_LOOKUP contient les codes d'état et les abréviations correspondantes.

La table OPTIM\_STATE\_LOOKUP contient les colonnes suivantes :

**DIST\_CODE**

CHAR ; 3 caractères ; peut contenir une valeur nulle.

**DISTRICT**

CHAR ; 2 caractères ; peut contenir une valeur nulle.

#### **Clés primaires**

La table OPTIM\_STATE\_LOOKUP ne possède pas de clé primaire.

#### **Relations avec les autres tables**

La table OPTIM\_STATE\_LOOKUP est un enfant de la table OPTIM\_SALES via une relation de sous-chaîne avec la colonne DISTRICT grâce à SUBSTR(SALESMAN\_ID,1,2).

### **Tables de confidentialité de données**

Les tables de confidentialité de données sont disponibles pour les clients possédant une licence de confidentialité des données Optim. Utilisez les tables pour masquer les données personnelles et de la société, telles que les noms d'employés, les noms de clients, les numéros de sécurité sociale, les numéros de carte de crédit et les adresses électroniques. Utilisez les tables pour générer des données transformées uniques et valides dans le contexte de l'application.

Vous pouvez utiliser les tables de confidentialité de données pour :

- éviter les violations de confidentialité interne en désidentifiant ou en masquant les données disponibles pour les développeurs, les testeurs d'assurance qualité et d'autres membres du personnel ;
- améliorer les initiatives de conformité à la confidentialité en remplaçant les données client par des données précises contextuellement, mais fictives ;
- protéger les informations client confidentielles et les données des employées dans vos environnements de développement et de test d'applications ;
- assurer des résultats de test valides en propageant des éléments masqués sur les tables associées pour vous assurer de l'intégrité référentielle de la base de données.

Chaque catégorie de données personnelles est fournie dans une table distincte pour les pays suivants (abréviations entre parenthèses) : Australie (AU), Canada (CA), France (FR), Allemagne (DE), Italie (IT), Japon (JP), Espagne (ES) Royaume-Uni (UK) et Etats-Unis (US). Chaque table comprend une colonne de numéros séquentiels qui est utilisée avec des règles de consultation utilisant des valeurs hachées pour sélectionner une ligne dans la table de consultation.

Chaque nom de table se compose d'un préfixe d'abréviation de pays et de la catégorie (*abréviationpays\_catégorie*). Par exemple, la table d'adresse du Canada est appelée CA\_ADDRESSES et celle de l'Allemagne, DE\_ADDRESSES.

Le schéma comprend les catégories suivantes :

#### **ADDRESSES**

Tables qui comprennent des colonnes pour le nom de la rue, la ville, la localité (par ex. état ou province) et le code postal.

#### **FIRSTNAME**

Tables qui comprennent une colonne avec des prénoms masculins et féminins donnés.

#### **FIRSTNAME\_F**

Tables qui comprennent une colonne avec des prénoms féminins donnés.

#### **FIRSTNAME\_M**

Tables qui comprennent une colonne avec des prénoms masculins donnés.

#### **LASTNAME**

Tables qui comprennent une colonne avec les noms de famille.

#### **PERSON**

Tables qui comprennent des colonnes pour la date de naissance, le nom donné, le nom de famille, le sexe, le numéro de téléphone, le numéro de carte d'identité, le nom de la société et l'adresse électronique.

**CCN** Tables qui comprennent un numéro de carte de crédit pour l'émetteur associé (MasterCard, VISA, etc.).

#### **DOMAIN\_NAMES**

Table qui comprend les noms de domaine pour le masquage des adresses électroniques.

### **Création de tables de données d'échantillon**

Utilisez l'utilitaire de configuration pour créer des table de données d'échantillon.

Pour créer des tables de données d'échantillon :

1. Dans Optim Designer, cliquez sur **Utilitaires > Configuration** L'utilitaire de configuration s'ouvre.
2. Cliquez sur **Tâches > Charger/Supprimer des données d'échantillon** L'assistant Charger/Supprimer des données d'échantillon s'ouvre.
3. Suivez les étapes de l'assistant.

Vous devez sélectionner un répertoire Optim. Le nom du répertoire Optim par défaut pour le référentiel Optim est PODREPO.

Lorsqu'il vous est demandé de sélectionner un alias de base de données, sélectionnez un nom d'alias de magasin de données.

### **Création de tables de confidentialité des données**

Utilisez l'utilitaire de configuration pour créer des tables de confidentialité des données.

Pour créer des tables de confidentialité des données :

1. Dans Optim Designer, cliquez sur **Utilitaires > Configuration** L'utilitaire de configuration s'ouvre.

2. Cliquez sur **Tâches > Charger/Supprimer des données de confidentialité de données** L'assistant Charger/Supprimer des données de confidentialité de données s'ouvre.
3. Suivez les étapes de l'assistant.

Vous devez sélectionner un répertoire Optim. Le nom du répertoire Optim par défaut pour le référentiel Optim est PODREPO.

Lorsqu'il vous est demandé de sélectionner un alias de base de données, sélectionnez un nom d'alias de magasin de données.

## Base de données prise en charge

Optim Designer fournit une prise en charge pour plusieurs systèmes de gestion de base de données.

Optim Designer prend en charge les bases de données suivantes :

- DB2 for z/OS V8.1, V9.1, V10.1
- DB2 for Linux, UNIX, and Windows V8.2, V9.1, V9.5, V9.7
- DB2 pour i V5.4
- Informix V10
- Microsoft SQL Server 2005, 2008
- Oracle V10.2, V11, V11.2
- Sybase V12.5, V15, V15.5
- Teradata V2.6, V12, V13
- IBM Informix V11.5, V.11.7

### Prérequis DB2

Pour qu'Optim puisse obtenir des métadonnées JDBC complètes à partir d'une instance de DB2 z/OS, vous devez définir la valeur DESCSTAT de ZPARMS sur YES. Vous devez également exécuter le travail DSNTIJMS pour installer les procédures mémorisées requises par JDBC, associer les packages nécessaires et définir les droits de sécurité. Des définitions Workload Manager (WLM) sont également nécessaires afin que WLM puisse démarrer l'espace adresse de procédure mémorisée lorsque DB2 le requiert.

---

## Fonctions d'accessibilité

Les fonctions d'accessibilité permettent aux utilisateurs ayant un handicap physique, par exemple les personnes à mobilité réduite ou malvoyantes, ou ayant d'autres besoins spéciaux, d'utiliser correctement les logiciels.

Optim Designer utilise les fonctions d'accessibilité disponibles avec l'environnement Eclipse.

Les fonctions d'accessibilité permettent aux utilisateurs ayant un handicap physique, par exemple les personnes à mobilité réduite ou malvoyantes, ou ayant des besoins spéciaux, d'utiliser correctement les logiciels.



---

## Chapitre 2. Gestion des connexions de source de données

Utilisez l'explorateur de référentiel pour créer des alias de magasin de données. Utilisez l'explorateur de répertoires pour vous connecter à un répertoire Optim.

---

### Utilisation d'un alias de magasin de données

Un alias de magasin de données est un objet défini par l'utilisateur et associé à une base de données. Lorsque vous définissez un alias de magasin de données, vous fournissez des paramètres utilisés par Optim pour communiquer avec cette base de données. Ces paramètres incluent le type et la version du système de gestion des bases de données (DBMS) et les propriétés d'une connexion client et d'une connexion JDBC.

Un nom d'alias de magasin de données sert de qualificatif de haut niveau, qui vous permet d'accéder à une base de données spécifique pour exécuter les fonctions demandées. Par exemple, dans une définition d'accès, vous devez qualifier le nom d'une table avec un nom d'alias de magasin de données. L'alias du magasin de données référencé fournit les paramètres nécessaires à la connexion à la base de données dans laquelle réside la table.

Un alias de magasin de données identifie une base de données spécifique et sert de préfixe dans les noms qualifiés complets des clés primaires, des tables de bases de données et des relations. Les alias de magasin de données sont des éléments essentiels à la gestion de vos bases de données. Les règles suivantes s'appliquent aux noms d'alias de magasin de données :

- Chaque base de données ne peut posséder qu'un alias de magasin de données.
- Chaque nom d'alias de magasin de données doit être unique.
- Les objets du référentiel ne peuvent pas posséder le même nom qu'un alias de magasin de données.

### Définition d'un alias de magasin de données

L'assistant Nouvel alias de magasin de données vous permet de définir un nouvel alias de magasin de données.

Pour définir un alias de magasin de données, vous avez besoin de la chaîne de connexion pour une connexion client et du fichier .jar pour une connexion JDBC.

Pour définir un alias de magasin de données :

1. Dans l'explorateur de référentiel, cliquez avec le bouton droit de la souris sur **Alias de magasin de données** et cliquez sur **Nouvel alias de magasin de données**. L'assistant Nouvel alias de magasin de données s'ouvre.
2. Suivez les étapes de l'assistant.

Vous devez entrer les propriétés associées à une connexion client et à une connexion JDBC. Vous devez également identifier le jeu de caractères utilisé par la base de données et la méthode de gestion des packages pour accéder aux objets Optim.

### Edition d'un alias de magasin de données

L'assistant Propriétés de connexion vous permet d'éditer un alias de magasin de données.

Pour éditer un alias de magasin de données :

1. Dans l'explorateur de référentiel, développez le noeud **Alias de magasin de données**, cliquez avec le bouton droit de la souris sur l'alias de magasin de données et cliquez sur **Ouvrir**. L'assistant Propriétés de connexion s'ouvre.

2. Suivez les étapes de l'assistant.

## Connexion à un alias de magasin de données

Utilisez l'explorateur de référentiel pour vous connecter à un alias de magasin de données.

Pour se connecter à un alias de magasin de données :

Dans l'explorateur de référentiel, développez le noeud **Alias de magasin de données**, cliquez avec le bouton droit de la souris sur l'alias de magasin de données et cliquez sur **Connecter**.

---

## Utilisation du référentiel Optim

Le référentiel Optim contient les définitions, telles que les alias de magasin de données, les services et les définitions d'accès, qui sont partagées entre les composants Optim.

Seul un répertoire Optim peut être associé au référentiel.

## Connexion au référentiel Optim

Utilisez la fenêtre Nouvelle connexion à un référentiel pour définir une connexion au référentiel Optim.

Le référentiel Optim doit être en cours d'exécution pour pouvoir créer une connexion.

Pour définir une connexion au référentiel Optim :

1. Dans l'explorateur de référentiel, cliquez avec le bouton droit de la souris sur **Référentiel Optim** et cliquez sur **Nouveau > Connexion à un référentiel**. La fenêtre Nouvelle connexion à un référentiel s'ouvre.
2. Entrez les informations de connexion ainsi que les données d'identification de l'utilisateur pour le référentiel.
3. Cliquez sur **Tester la connexion** afin de vérifier la connexion.
4. Cliquez sur **OK**.

Lorsque Optim Designer se connecte au référentiel, l'explorateur de référentiel affiche le contenu du référentiel.

## Edition d'une connexion à un référentiel

Utilisez la fenêtre Editer une connexion à un référentiel pour définir une connexion au référentiel Optim.

Pour définir une connexion au référentiel Optim :

1. Dans l'explorateur de référentiel, cliquez avec le bouton droit de la souris sur **Référentiel Optim** et cliquez sur **Ouvrir**. La fenêtre Editer une connexion à un référentiel s'ouvre.
2. Entrez les informations de connexion ainsi que les données d'identification de l'utilisateur pour le référentiel.
3. Cliquez sur **Tester la connexion** afin de vérifier la connexion.
4. Cliquez sur **OK**.

Lorsque Optim Designer se connecte au référentiel, l'explorateur de référentiel affiche le contenu du référentiel.

## Changement d'une connexion à un référentiel

Utilisez la fenêtre Changer de Référentiel pour vous connecter à un autre référentiel Optim.

Le référentiel Optim s'exécute pour changer une connexion.

Pour changer une connexion au profit du référentiel Optim :

1. Dans l'explorateur de référentiel, cliquez avec le bouton droit de la souris sur **référentiel Optim**, puis cliquez sur **Changer de référentiel**. La fenêtre Changer de référentiel s'ouvre.
2. Sélectionnez une connexion à un référentiel.
3. Cliquez sur **OK**.

Lorsque Optim Designer se connecte au référentiel, l'explorateur de référentiel affiche le contenu du référentiel.

## Association d'un répertoire Optim au référentiel

Utilisez la fenêtre Répertoire associé pour associer un répertoire Optim au référentiel.

Une connexion de répertoire Optim doit être définie dans le registre de l'ordinateur sur lequel Optim Designer est installé. Utilisez le programme de configuration Optim pour définir une connexion dans le registre.

Un seul répertoire Optim peut être associé au référentiel.

Pour associer un répertoire Optim au référentiel :

1. Dans l'explorateur de répertoire, cliquez sur **Cliquez pour associer un répertoire Optim à une connexion à un référentiel**. La fenêtre Répertoire associé s'ouvre.
2. Sélectionnez un répertoire Optim.
3. Cliquez sur **Terminer**.

Lorsque Optim Designer se connecte au répertoire Optim, l'explorateur de répertoire affiche le contenu du répertoire Optim.

## Transformation d'une demande en service

Utilisez l'assistant Transformer la demande en service Optim pour transformer la demande d'un répertoire Optim en un service de gestion de données.

Un répertoire Optim doit être associé au référentiel.

Pour transformer la demande d'un répertoire Optim en un service de gestion de données :

1. Dans l'explorateur d'annuaire, cliquez sur **Transformer**. L'assistant Transformer la demande en un service Optim s'ouvre.
2. Dans la page Alias de base de données et Connexions à la base de données, consultez les informations de connexion. Si un alias de magasin de données n'est pas défini pour un alias de base de données associé à la demande, cliquez sur **Modifier la connexion** pour définir un alias de magasin de données. Cliquez sur **Suivant**.
3. Si la transformation aboutit, cliquez sur **Terminer**. Si la transformation n'aboutit pas, cliquez sur **Suivant** pour consulter les informations relatives à l'erreur.

Un dossier possédant le nom du répertoire Optim et contenant la demande sera créé dans l'explorateur de référentiel. La demande transformée sera répertoriée comme service dans le dossier.



---

## Chapitre 3. Gestion des modèles de données

Vous pouvez créer des modèles de données définissant les données à extraire et mappant les données cible pour les services de gestion. Vous pouvez également créer des relations Optim et des clés primaires venant compléter les relations et les clés primaires de la base de données.

---

### Utilisation des définitions d'accès

Utilisez des définitions d'accès pour définir les tables, la traversée de relation et les critères de sélection des données à traiter.

Une définition d'accès comprend les éléments suivants.

**Tables** Une définition d'accès doit faire référence à au moins une table, une vue, un alias ou un synonyme. La table, vue, alias ou synonyme à partir duquel/de laquelle les lignes sont sélectionnées en premier lieu s'appelle table initiale. Vous pouvez entrer le nom d'une table initiale et inclure facilement les noms de toutes les tables associées à la table initiale (24 000 tables maximum).

#### Table initiale

La table initiale est la première table à utiliser lors de l'extraction des données. Vous pouvez indiquer une table dans la définition d'accès comme table initiale, à l'exception d'une table de référence. Si vous n'indiquez pas de façon explicite une table initiale, la première table de la liste des tables est utilisée dans ce sens.

#### Table de référence

Tables à partir desquelles toutes les lignes sont sélectionnées, à moins que des critères de sélection ne soient spécifiés pour la table de référence. Définissez une table de référence autre que la table initiale.

### Relations

Les relations déterminent le chemin de traversée pour sélectionner les données à partir des tables. Par défaut, les relations sont traversées de parent vers enfant, mais vous pouvez contrôler le sens de traversée via les paramètres de la définition d'accès. Les relations entre les tables référencées par la définition d'accès sont répertoriées dans l'onglet Relations (24 000 tables maximum). Vous pouvez sélectionner les relations à utiliser dans le traitement et le sens de traversée.

### Critères de sélection

Les critères de sélection définissent un ensemble de données spécifique à utiliser depuis les tables dans une définition d'accès. Vous pouvez indiquer des opérateurs SQL et des valeurs et utiliser des variables de substitution avec les valeurs par défaut.

### Pointage et découpage

Utilisez une liste de pointage et de découpage pour sélectionner des lignes spécifiques à partir d'une table initiale qui sera incluse dans un service.

### Variabes

Les variables sont des valeurs par défaut définies par l'utilisateur et spécifiées dans une définition d'accès. Vous pouvez utiliser ces variables de substitution pour indiquer des critères de sélection de colonne ou y créer une clause SQL WHERE.

### Paramètres supplémentaires

Utilisez des paramètres supplémentaire pour extraire des lignes correspondant à une valeur de colonne particulière dans la table initiale ou paramètres d'utilisation d'une fréquence d'échantillonnage indiquée (toutes les x lignes).

## Conventions de dénomination

Nom de définition d'accès qualifié complet sous la forme *identifier.name*, où :

*identifier*

Qualificateur attribué à la définition d'accès (1 à 8 caractères).

*name* Nom de base attribué à la définition d'accès (1 à 12 caractères).

Un ensemble logique de conventions de dénomination peut identifier l'utilisation de chaque définition d'accès et être utilisé pour les organiser et en faciliter l'accès.

## Création d'une définition d'accès

L'assistant Nouvelle définition d'accès vous permet de créer une définition d'accès.

Avant de créer une définition d'accès, un alias de magasin de données doit exister pour la base de données qui contient la table initiale.

Pour créer une définition d'accès :

1. Développez un dossier dans l'explorateur de référentiel qui contiendra la définition d'accès, cliquez avec le bouton droit de la souris sur **Définitions d'accès**, puis cliquez sur **Nouvelle définition d'accès**. L'assistant Nouvelle définition d'accès s'ouvre.
2. Suivez les étapes de l'assistant.  
Vous devez sélectionner un alias de magasin de données ainsi qu'une table initiale à partir de l'alias. Vous pouvez choisir d'ajouter des tables associées à la table initiale.

La nouvelle définition d'accès s'affiche dans l'éditeur de définition d'accès.

## Gestion des tables dans une définition d'accès

Utilisez l'onglet **Tables** de l'éditeur de définition d'accès pour ajouter ou supprimer des tables dans une définition d'accès. Vous pouvez également remplacer les tables par des tables liées ou de référence.

### Ajout de tables à une définition d'accès

Utilisez l'assistant Ajouter des tables pour ajouter des tables à une définition d'accès dans un alias de magasin de données.

Pour ajouter des tables à une définition d'accès :

1. Développez un dossier dans l'explorateur de référentiel qui contient la définition d'accès, développez le noeud **Définitions d'accès** et cliquez deux fois sur la définition d'accès. L'éditeur de définition d'accès s'ouvre.
2. Sélectionnez l'onglet **Tables**.
3. Cliquez sur **Ajouter des tables**. L'assistant Ajouter des tables s'ouvre.
4. Suivez les étapes de l'assistant.  
Vous devez sélectionner un alias de magasin de données qui contient les tables à ajouter. Vous pouvez ajouter des tables de référence ou des tables associées à une table sélectionnée.
5. Sauvegardez la définition d'accès.

### Modification des tables en tables de référence ou en tables associées

Utilisez l'éditeur de définition d'accès pour modifier une table dans une définition d'accès en table associée ou en table de référence.

Pour modifier une table en table associée ou en table de référence :

1. Développez le dossier dans l'explorateur de référentiel qui contient la définition d'accès, développez le noeud **Définitions d'accès** et cliquez deux fois sur la définition d'accès. L'éditeur de définition d'accès s'ouvre.
2. Sélectionnez l'onglet **Tables**.
3. Sélectionnez une table à modifier.
4. Cliquez sur **Modifier en table de référence** ou sur **Modifier en table associée**. La colonne **Type** contient le nouveau type de table.
5. Sauvegardez la définition d'accès.

### **Suppression de tables d'une définition d'accès**

L'éditeur de définition d'accès vous permet de supprimer des tables d'une définition d'accès.

Pour supprimer des tables d'une définition d'accès :

1. Développez le dossier dans l'explorateur de référentiel qui contient la définition d'accès, développez le noeud **Définitions d'accès** et cliquez deux fois sur la définition d'accès. L'éditeur de définition d'accès s'ouvre.
2. Sélectionnez l'onglet **Tables**.
3. Sélectionnez une table à supprimer.
4. Cliquez sur **Supprimer la table**.
5. Sauvegardez la définition d'accès.

### **Affichage et édition des étapes de traversée**

La fenêtre Etapes de traversée vous permet d'afficher et d'éditer les étapes de traversée d'une définition d'accès.

Vous pouvez évaluer les tables, les relations et les étapes effectuées pour sélectionner des données. Cette évaluation peut garantir que vous récupérez l'ensemble de données souhaité.

Les étapes d'un processus peuvent être répétées un certain nombre de fois en fonction des relations que vous sélectionnez et des critères que vous spécifiez.

Toute table peut être revisitée plusieurs fois dans des étapes successives. Des cycles peuvent également être impliqués. Un cycle a pour effet la traversée répétée d'un ensemble de tables tant qu'un passage complet à travers le cycle implique de sélectionner des lignes supplémentaires.

Lorsqu'une relation est traversée pour sélectionner des lignes parentes pour des lignes enfant déjà sélectionnées, tout critère de sélection de la table parente est ignoré.

Pour afficher et éditer des étapes de traversée :

1. Développez un dossier dans l'explorateur de référentiel qui contient la définition d'accès, développez le noeud **Définitions d'accès** et cliquez deux fois sur la définition d'accès. L'éditeur de définition d'accès s'ouvre.
2. Sélectionnez l'onglet **Tables**.
3. Cliquez sur **Afficher les étapes**. La fenêtre Etapes de traversée s'ouvre.
4. Consultez les étapes et éditez l'ordre dans lequel les tables sont sélectionnées.
5. Cliquez sur **OK**.
6. Sauvegardez la définition d'accès.

## Gestion des critères de sélection

Les critères de sélection permettent de mettre en évidence un ensemble spécifique de données en définissant une clause SQL WHERE et en utilisant des variables de substitution avec les valeurs par défaut.

Les critères de sélection doivent respecter la syntaxe SQL et inclure des opérateurs relationnels ou logiques. Les opérateurs logiques et la syntaxe diffèrent en fonction des SGBD. Consultez la documentation appropriée à votre SGBD pour plus d'informations.

Pour sélectionner l'ensemble de données de votre choix pour une table, vous pouvez avoir besoin d'une combinaison d'opérateurs logiques AND et OR.

Si une liste de pointage et découpage directe est également utilisée, elle est incluse de manière logique aux autres critères à l'aide d'un opérateur logique OR.

Lorsqu'une relation est traversée de l'enfant au parent, tout critère de sélection de la table parente est ignoré.

### Critères de date

Un opérateur unique, **BEFORE**, vous permet de sélectionner des données sur la base de valeurs dans une colonne DATE. La syntaxe de cet opérateur est la suivante :

**BEFORE** (*nD nW nM nY*)

Les arguments **D**, **W**, **M** et **Y** sont à combiner afin d'indiquer le nombre de jours, de semaines, de mois et d'années soustrait de la date lors de l'exécution. Si aucun argument n'est indiqué, la date du jour est utilisée. Les lignes dont la date est antérieure à la date calculée sont extraites ou archivées. Le multiplicateur *n* est un entier qui peut éventuellement être précédé par le signe + ou -.

### Définition des critères de sélection

Utilisez la fenêtre Editer les critères de sélection pour définir des critères de sélection avec une clause SQL WHERE.

Pour définir des critères de sélection :

1. Développez un dossier dans l'explorateur de référentiel qui contient la définition d'accès, développez le noeud **Définitions d'accès** et cliquez deux fois sur la définition d'accès. L'éditeur de définition d'accès s'ouvre.
2. Sélectionnez l'onglet **Tables**.
3. Sélectionnez la table dans laquelle vous allez ajouter les critères de sélection.
4. Cliquez sur **Editer les critères de sélection**. La fenêtre Editer les critères de sélection s'ouvre et affiche une clause WHERE pour la table.
5. Entrez les critères de sélection.  
Pour afficher la liste des colonnes et des opérateurs, cliquez avec le bouton droit de la souris sur la clause WHERE, puis cliquez sur **Assistant de contenu**.  
Pour ajouter une variable, sélectionnez un **Délimiteur de variable** et cliquez sur **Insérer une variable**. Cliquez sur **Vérifier la syntaxe** pour vérifier la syntaxe et identifier les erreurs.
6. Cliquez sur **OK**.  
Si la syntaxe est incorrecte, une invite s'ouvre et identifie les erreurs. Vous ne pouvez pas sauvegarder les critères s'ils contiennent des erreurs.
7. Sauvegardez la définition d'accès.

## Définition des options de groupement des données

Utilisez l'onglet **Propriétés de groupe de données** de l'éditeur de définition d'accès pour spécifier la sélection du groupe, qui extrait un nombre de lignes en fonction des valeurs d'une colonne spécifique dans la table initiale. Les lignes dans chaque groupe ont les mêmes valeurs dans la colonne sélectionnée.

Lorsque les critères sont utilisés avec les paramètres de sélection de groupe, les critères s'appliquent d'abord avec les paramètres de sélection appliqués au résultat.

Pour spécifier les options de groupement des données :

1. Développez un dossier dans l'explorateur de référentiel qui contient la définition d'accès, développez le noeud **Définitions d'accès** et cliquez deux fois sur la définition d'accès. L'éditeur de définition d'accès s'ouvre.
2. Sélectionnez l'onglet **Propriétés des groupes de données**.
3. Sélectionnez une colonne et spécifiez les options de groupement.
4. Sauvegardez la définition d'accès.

## Définition des options d'échantillonnage des données et du nombre limite de lignes

Utilisez l'onglet **Tables** dans l'éditeur de définition d'accès pour spécifier les options d'extraction d'un échantillonnage de lignes ou de limitation du nombre de lignes à extraire.

Si vous utilisez une liste de pointage et découpage pour sélectionner les lignes de la table initiale, le service d'extraction ignore les paramètres d'échantillonnage de données et du nombre limite de lignes pour la table initiale.

Pour spécifier les options d'échantillonnage des données et de limite du nombre de lignes :

1. Développez un dossier dans l'explorateur de référentiel qui contient la définition d'accès, développez le noeud **Définitions d'accès** et cliquez deux fois sur la définition d'accès. L'éditeur de définition d'accès s'ouvre.
2. Sélectionnez l'onglet **Tables**.
3. Entrez les options suivantes :

### Tous les nième nombres de lignes

Entrez une valeur numérique pour spécifier un facteur d'échantillonnage pour une table. Par exemple, si vous entrez 5, le processus extrait une ligne sur cinq dans la table, en commençant par la cinquième ligne. Les valeurs admises sont comprises entre 1 et 9999.

### Nombre max. de lignes

Entrez une valeur numérique pour limiter le nombre de lignes extraites d'une table. Les valeurs valides sont comprises entre 1 et 999999999999.

4. Sauvegardez la définition d'accès.

## Gestion des relations dans une définition d'accès

Utilisez l'onglet **Relations** de l'éditeur de définition d'accès pour ajouter ou supprimer des relations dans une définition d'accès. Vous pouvez également indiquer des options de traversée de relation et d'accès aux tables.

### Sélection des options de traversée d'une relation

L'éditeur de définition d'accès vous permet de gérer les options de traversée d'une relation dans une définition d'accès.

Vous pouvez spécifier les options suivantes :

- Nombre maximal de lignes de la table enfant à sélectionner pour une relation.
- Nombre maximal de consultations de clé effectuées simultanément pour une table.

L'augmentation de la limite de consultation de clé peut améliorer les performances de manière significative. Par exemple, si vous définissez la limite de consultation de clé sur 5 et que la clé possède une seule colonne, les valeurs de clé 5 sont recherchées dans une requête unique auprès du SGBD.

- Les chemins de traversée :
  - Traversent la relation de l'enfant au parent, afin de sélectionner une ligne parente pour chaque ligne enfant pour garantir l'intégrité de la relation. (Option 1)
  - Traversent la relation afin de sélectionner des lignes enfants supplémentaires pour chaque ligne parente sélectionnée suite à une traversée de l'enfant au parent. (Option 2)

Les options 1 et 2 s'avèrent pertinentes lorsque la table initiale est une table enfant ou lorsqu'une table a plusieurs tables parent référencées dans la définition d'accès.

L'option 2 doit être utilisée uniquement si vous traversez une relation de l'enfant au parent. Par exemple, si un processus effectue une traversée de l'enfant au parent (option 1) et qu'une ligne parente est sélectionnée, l'option 2 entraîne le processus à sélectionner des lignes enfants supplémentaires pour cette ligne parente.

Si vous sélectionnez l'option 2 pour une relation, réfléchissez à une limite enfant pour restreindre le nombre de lignes enfants à extraire.

Pour gérer les options de traversée d'une relation dans une définition d'accès :

1. Développez le dossier dans l'explorateur de référentiel qui contient la définition d'accès, développez le noeud **Définitions d'accès** et cliquez deux fois sur la définition d'accès. L'éditeur de définition d'accès s'ouvre.
2. Sélectionnez l'onglet **Relations**, puis l'onglet **Options de traversée**.
3. Entrez les options de traversée.
4. Sauvegardez la définition d'accès.

## Sélection des options d'accès à une table

L'éditeur de définition d'accès vous permet de sélectionner la méthode d'accès à la table enfant ou parente pour chaque relation.

Vous pouvez spécifier les options suivantes :

### Par défaut

Optim détermine la meilleure méthode. Une consultation de clé est utilisée lorsqu'un index de SGBD est disponible et une analyse est effectuée lorsqu'aucun index n'est disponible. Toutefois, si vous souhaitez accéder à une grande partie de la table, la méthode par défaut consiste à effectuer une analyse, même si un index existe.

### Imposer l'analyse

Lit toutes les lignes d'une table en une seule fois.

### Imposer la consultation de clé

Localise les lignes à l'aide d'une clause WHERE pour rechercher des valeurs de clé primaire ou externe.

**Remarque :** Cette méthode remplace la méthode par défaut uniquement si les informations statistiques du rapport de processus indiquent que la méthode par défaut est moins efficace.

Pour gérer les options d'accès à une table :

1. Développez le dossier dans l'explorateur de référentiel qui contient la définition d'accès, développez le noeud **Définitions d'accès** et cliquez deux fois sur la définition d'accès. L'éditeur de définition d'accès s'ouvre.
2. Sélectionnez l'onglet **Relations**, puis l'onglet **Accès aux tables**.
3. Entrez les options d'accès à la table.
4. Sauvegardez la définition d'accès.

## Gestion des variables dans une définition d'accès

Les variables sont des valeurs par défaut définies par l'utilisateur dans une définition d'accès. Vous pouvez utiliser ces variables de substitution pour créer une clause SQL WHERE.

En affectant des variables, vous fournissez les valeurs des variables à chaque traitement de la définition d'accès. Vous pouvez aussi fournir des valeurs par défaut pour les variables de substitution. Les variables sont enregistrées avec la définition d'accès.

### Valeurs par défaut

Lorsque vous créez une variable, vous pouvez entrer une valeur par défaut facultative à utiliser lorsqu'aucune valeur n'est indiquée pour la variable au moment de l'exécution.

Le type de données et la taille des valeurs par défaut doivent être appropriés à la colonne et se conformer à la syntaxe SQL. Par exemple, prenons une variable **ST** (state) avec le symbole deux-points ( : ) comme délimiteur et une colonne nécessitant des données de type caractères.

- Si vous utilisez la variable avec des guillemets simples dans les critères de sélection, vous devez indiquer la valeur sans guillemets simples :

Critères de sélection	Valeur
= 'ST'	CA

- Si vous utilisez la variable avec des guillemets simples dans les critères de sélection, vous devez indiquer la valeur avec des guillemets simples :

Critères de sélection	Valeur
= :ST	'CA'

**Remarque :** Les valeurs par défaut ne sont pas validées jusqu'à l'exécution. Si une valeur possède un type de données ou une taille incorrect pour la colonne ou ne se conforme pas à la syntaxe SQL, des erreurs de traitement peuvent se produire.

### Chaîne d'invite

Vous devez entrer le texte demandant la valeur de la variable au moment de l'exécution. Saisissez la chaîne d'invite que vous souhaitez voir s'afficher dans la boîte de dialogue de demande de traitement (jusqu'à 50 caractères). Cette invite s'affiche avant l'exécution du processus.

### Création d'une variable

Utilisez la fenêtre Ajouter une variable pour créer une variable dans une définition d'accès.

Pour créer une variable :

1. Développez le dossier dans l'explorateur de référentiel qui contient la définition d'accès, développez le noeud **Définitions d'accès** et cliquez deux fois sur la définition d'accès. L'éditeur de définition d'accès s'ouvre.
2. Sélectionnez l'onglet **Variables**.
3. Cliquez sur **Ajouter**. La fenêtre Ajouter une variable s'ouvre.
4. Entrez les informations sur la variable. Vous devez entrer un nom de variable et un texte d'invite. Cliquez sur **OK**.
5. Sauvegardez la définition d'accès.

## Gestion des listes de pointage et de découpage

Utilisez une liste de pointage et de découpage pour sélectionner des lignes spécifiques à partir d'une table initiale qui sera incluse dans un service.

Une liste de pointage et de découpage est incluse dans une définition d'accès. Lorsque vous utilisez une liste de pointage et de découpage à partir de la table initiale, les clés primaires de ces lignes sont stockées dans un fichier de pointage et de découpage. La demande de service utilise les clés primaires pour identifier les lignes à traiter en premier lieu.

### Création d'un fichier liste de pointage et de découpage

Utilisez la fenêtre Nouveau fichier de pointage et découpage pour créer un fichier liste de pointage et découpage pour une table dans un alias de magasin de données.

Pour créer un fichier liste de pointage et découpage :

1. Dans l'explorateur de référentiel, développez le noeud **Alias de magasin de données** et l'alias de magasin de données cible jusqu'à ce que la table à laquelle vous allez ajouter le fichier de pointage et découpage s'affiche.
2. Développez la table, cliquez avec le bouton droit de la souris sur **Listes de pointage et de découpage** et cliquez sur **Nouveau fichier de pointage et découpage**. La fenêtre Nouveau fichier de pointage et découpage s'ouvre.
3. Entrez un nom de fichier et cliquez sur **OK**.

Le fichier liste de pointage et découpage s'affiche sous **Listes de pointage et de découpage**.

Vous devez utiliser l'éditeur de pointage et découpage pour sélectionner des lignes ou la liste de pointage et découpage.

### Sélection d'une liste de pointage et découpage

L'éditeur de définition d'accès vous permet de sélectionner une liste de pointage et découpage pour une définition d'accès.

Pour sélectionner une liste de pointage et découpage, un fichier de pointage et découpage doit être disponible pour la table initiale.

Pour sélectionner une liste de pointage et découpage :

1. Développez le dossier dans l'explorateur de référentiel qui contient la définition d'accès, développez le noeud **Définitions d'accès** et cliquez deux fois sur la définition d'accès. L'éditeur de définition d'accès s'ouvre.
2. Sélectionnez l'onglet **Pointage et découpage**.
3. Sélectionnez une liste de pointage et découpage.
4. Sauvegardez la définition d'accès.

---

## Utilisation des relations Optim

Optim utilise des relations pour déterminer les données à extraire des tables associées et se base sur les relations définies dans la base de données (si disponibles). Toutefois, vous pouvez également définir des relations Optim pour compléter les relations dans la base de données.

Avec les relations Optim, plusieurs restrictions de base de données sont levées. Par exemple :

- Les clés primaires et les clés externes ne sont pas requises.
- Il n'est pas nécessaire que les colonnes correspondantes soient identiques, mais elles doivent être compatibles.

- Au moins une colonne par paire de colonnes correspondantes doit être indiquée par nom de colonne. Toutefois, vous pouvez utiliser une expression pour évaluer ou définir la valeur dans la seconde colonne. Les expressions peuvent inclure des littéraux chaîne, des constantes numériques, des valeurs NULL, des concaténations et des sous-chaînes.

Les relations Optim les plus flexibles sont les relations “étendues”. Les relations étendues peuvent répliquer des relations implicites ou gérées via des applications dans votre base de données, qui vous permettent de manipuler des ensembles de données relationnelles de la même manière que dans votre environnement de production.

De plus, une relation Optim peut être stockée dans le répertoire Optim comme suit :

- Une relation **explicite**, utilisée pour une paire de tables unique.
- Une relation **générique**, utilisée pour une ou plusieurs paires de tables possédant le même nom de base, les mêmes noms de colonnes et les mêmes attributs, mais différents ID de créateur.

Les relations génériques sont utiles lorsque seul l’ID de créateur permet de différencier les ensembles de tables. Par exemple, dans un environnement de test, chaque programmeur peut utiliser une copie des mêmes tables de production. Chaque ensemble de tables peut être différencié par son ID de créateur. Les relations génériques vous permettent de définir un ensemble de relations s’appliquant à tous les ensembles de tables. De plus, lorsqu’un ensemble de ces tables est ajouté, les relations génériques s’appliquent automatiquement.

## Restrictions

Bien que les règles de création de relations Optim soient plus flexibles que les règles de création de relations définies par la base de données, il existe quelques restrictions :

- Vous devez référencer au moins une colonne pour chaque table de la relation.
- Vous pouvez référencer au maximum 64 colonnes pour toute table de la relation.
- Vous ne pouvez pas faire correspondre un littéral ou une constante à un littéral ou une constante.
- Vous ne pouvez pas utiliser un objet LOB ou une colonne de variante SQL.
- La longueur totale de toutes les valeurs indiquées dans la table parent ou la table enfant ne peut pas dépasser 3 584 octets.
- Vous ne pouvez pas créer une relation à l’aide d’une colonne de variante SQL.

Dans une définition de relation pour base de données à plusieurs octets ou Unicode :

- Vous ne pouvez pas utiliser la fonction de sous-chaîne.
- Vous ne pouvez pas concaténer les données de type caractères (CHAR ou NCHAR) avec des données binaires (RAW).
- Si la sémantique d’un caractère Oracle est utilisée pour une colonne CHAR, toutes les colonnes CHAR de la relation doivent posséder la sémantique du caractère ou un type de données NCHAR.

EXEMPLES :

Parent	Pris en charge/Non pris en charge	Enfant	Description
CHAR		CHAR	Pris en charge, la sémantique doit correspondre
NCHAR		NCHAR	Pris en charge, sémantique non pertinente.

Parent	Pris en charge/Non pris en charge	Enfant	Description
CHAR		NCHAR	Non pris en charge
CHAR		VARCHAR	Pris en charge, la sémantique doit correspondre.
NCHAR		NVARCHAR	Pris en charge, sémantique non pertinente.
CHAR   NCHAR		NCHAR   CHAR	Pris en charge, si sémantique de type caractères ; non pris en charge si sémantique de type octets.
CHAR   NCHAR		NCHAR   NCHAR	Pris en charge, si sémantique de type caractères ; non pris en charge si sémantique de type octets.

## Règles de compatibilité pour les relations

Lorsque vous définissez une relation Optim, les valeurs correspondantes doivent être compatibles.

Type de colonne	Compatible avec
Caractère	<ul style="list-style-type: none"> <li>• Colonne de type caractère</li> <li>• Colonne de type numérique</li> <li>• Littéral chaîne</li> <li>• Expression de type caractère</li> </ul>
Numérique	<ul style="list-style-type: none"> <li>• Colonne de type numérique</li> <li>• Constante numérique</li> <li>• Colonne de type caractère</li> </ul>
Binaire	<ul style="list-style-type: none"> <li>• Colonne de type binaire</li> <li>• Littéral hexadécimal</li> <li>• Expression binaire</li> </ul>
Booléen	<ul style="list-style-type: none"> <li>• Colonne de type booléen</li> <li>• Constante booléenne (True ou False)</li> </ul>
Date-heure	Colonne de type date-heure
Date	Colonne de type date
Heure	Colonne de type heure
Intervalle	Colonne de type intervalle

### Remarque :

- Au cours du traitement, une valeur est convertie en type de données requis pour sélectionner les lignes associées. Par défaut, le résultat de la conversion d'une valeur numérique en type de données de caractères est une valeur justifiée à droite avec des zéros en début de chaîne. Les paramètres de registre spéciaux vous permettent de modifier la valeur par défaut sur une justification à gauche avec des espaces de début ou de fin de chaîne. Une concordance caractère/numérique nécessite également que la colonne de type numérique ait une échelle égale à 0.

- Vous pouvez utiliser NULL pour n'importe quelle colonne compatible.
- Les colonnes de type Unicode ou multi-octets doivent posséder le même jeu de caractères.

## Types de données

Les classes de données et leurs types de données associés suivants sont pris en charge. Ces classes de données jouent un rôle important pour la compatibilité des données lorsque vous utilisez des valeurs de colonne dans des relations.

### Classe de caractères

SGBD	Types de données
DB2	CHAR, VARCHAR, CLOB
Oracle	CHAR, VARCHAR2, LONG, CLOB, NCLOB, NCHAR, NVARCHAR
Sybase ASE	CHAR, VARCHAR, TXT
SQL Server	CHAR, VARCHAR, TXT
Informix	CHAR, VARCHAR, TXT

**Remarque :** Les colonnes contenant des caractères mono-octets ne sont pas compatibles avec les colonnes contenant des caractères Unicode ou multi-octets.

### Classe numérique

SGBD	Types de données
DB2	INTEGER, SMALLINT, DECIMAL, FLOAT, DOUBLE
Oracle	NUMBER, FLOAT
Sybase ASE	TINYINT, INT, SMALLINT, DECIMAL, FLOAT, REAL, MONEY, SMALL MONEY
SQL Server	TINYINT, INT, SMALLINT, DECIMAL, FLOAT, REAL, MONEY, SMALL MONEY
Informix	INTEGER, SMALLINT, DECIMAL, FLOAT, REAL, DOUBLE PRECISION, SMALLFLOAT, SERIAL, MONEY, NUMERIC

### Classe binaire

SGBD	Types de données
DB2	CHAR (pour les données binaires), VARCHAR (pour les données binaires), BLOB
Oracle	RAW, LONG RAW
Sybase ASE	BINARY, VARBINARY, IMAGE
SQL Server	BINARY, VARBINARY, IMAGE
Informix	BYTE

### Classe booléenne

SGBD	Types de données
Sybase ASE	BOOLEAN (TRUE ou FALSE)

### Date-heure

SGBD	Types de données
DB2	TIMESTAMP
Oracle	DATE, TIMESTAMP, TIMESTAMP WITH LOCAL TIME ZONE, TIMESTAMP WITH TIME ZONE
Sybase ASE	DATETIME, SMALL DATE TIME
SQL Server	DATETIME, SMALL DATE TIME
Informix	DATE, DATETIME

### Classe de date

SGBD	Types de données
DB2	DATE
Oracle	DATE
Informix	DATE

### Classe d'heure

SGBD	Types de données
DB2	TIME

### Classe d'intervalle

SGBD	Types de données
Oracle	YEAR/MONTH INTERVAL, DAY/SECOND INTERVAL
Informix	YEAR/MONTH INTERVAL, DAY/TIME INTERVAL

## Création d'une relation Optim

L'assistant Nouvelle relation Optim permet de créer une relation Optim.

Avant de créer une relation Optim, un alias de magasin de données doit exister pour les tables de la relation.

Pour créer une relation Optim :

1. Dans l'explorateur de référentiel, cliquez avec le bouton droit de la souris sur **Relations Optim** et cliquez sur **Nouvelle relation Optim**. L'assistant Nouvelle relation Optim s'ouvre.
2. Suivez les étapes de l'assistant.  
Vous devez sélectionner les tables parente et enfant dans la relation.

La nouvelle relation Optim s'affiche dans l'éditeur de relation.

Vous devez utiliser l'éditeur de relation pour définir une expression de colonne qui identifie les colonnes parente et enfant dans la relation.

## Modification d'une relation Optim

Utilisez l'éditeur de relations pour définir une expression de colonne identifiant les colonnes parent et enfant dans une relation Optim. Vous pouvez également utiliser l'éditeur pour créer une relation générique et modifier des colonnes.

## Création d'une expression de colonne

Utilisez la fenêtre Ajouter une expression de colonne pour créer une expression de colonne qui identifie les colonnes parente et enfant dans une relation Optim.

Pour créer une expression de colonne dans une relation Optim :

1. Dans l'explorateur de référentiel, développez le noeud **Relations Optim** et cliquez deux fois sur la relation à éditer. L'éditeur de relation s'ouvre.
2. Cliquez sur **Ajouter une expression de colonne...** La fenêtre Ajouter une expression de colonne s'affiche.
3. Cliquez sur **Sélectionner une colonne parente**. La fenêtre Sélectionner une colonne parente s'ouvre.
4. Sélectionnez une colonne parente et cliquez sur **OK** pour revenir à la fenêtre Ajouter une expression de colonne.
5. Cliquez sur **Sélectionner une colonne enfant**. La fenêtre Sélectionner une colonne enfant s'ouvre.
6. Sélectionnez une colonne enfant et cliquez sur **OK** pour revenir à la fenêtre Ajouter une expression de colonne.
7. Cliquez sur **OK** pour revenir à l'éditeur. Les colonnes enfant et parente sont répertoriées dans l'éditeur.
8. Sauvegardez la relation.

## Edition des colonnes dans une relation

Utilisez les fenêtres Sélectionner une colonne parente et Sélectionner une colonne enfant pour éditer les colonnes dans une relation Optim.

Pour éditer les colonnes dans une relation Optim :

1. Dans l'explorateur de référentiel, développez le noeud **Relations Optim** et cliquez deux fois sur la relation à éditer. L'éditeur de relation s'ouvre.
2. Sélectionnez la ligne qui contient une colonne à éditer. Pour éditer une colonne parente :
  - a. Cliquez sur **Sélectionner une colonne parente**. La fenêtre Sélectionner une colonne parente s'ouvre.
  - b. Sélectionnez une colonne parente et cliquez sur **OK** pour revenir à l'éditeur. Pour éditer une colonne enfant :
  - a. Cliquez sur **Sélectionner une colonne enfant**. La fenêtre Sélectionner une colonne enfant s'ouvre.
  - b. Sélectionnez une colonne enfant et cliquez sur **OK** pour revenir à l'éditeur. Les éléments sélectionnés sont répertoriés dans l'éditeur.
3. Sauvegardez la relation.

## Modification de l'ordre des colonnes dans une relation

Utilisez l'éditeur de relation pour modifier l'ordre des colonnes dans une relation Optim.

Pour modifier l'ordre des colonnes dans une relation Optim :

1. Dans l'explorateur de référentiel, développez le noeud **Relations Optim** et cliquez deux fois sur la relation à éditer. L'éditeur de relation s'ouvre.
2. Sélectionnez la ligne qui contient les colonnes à réorganiser.
3. Cliquez sur **Déplacer la ligne vers le haut** ou sur **Déplacer la ligne vers le bas** pour modifier l'ordre.
4. Sauvegardez la relation.

## Création d'une relation générique

Utilisez l'éditeur de relation pour créer une relation Optim générique.

Certaines bases de données contiennent des ensembles de tables identiques, à l'exception de l'ID de créateur. Plutôt que de définir une relation pour chaque ensemble de tables, vous pouvez définir une relation générique qui s'applique à tous les ensembles de tables qui portent le même nom de base, quel que soit leur ID de créateur. Vous pouvez modifier les tables de base dans une relation générique.

Pour créer une relation Optim générique :

1. Dans l'explorateur de référentiel, développez le noeud **Relations Optim** et cliquez deux fois sur la relation à éditer. L'éditeur de relation s'ouvre.
2. Cliquez sur **Générique**.
3. Sauvegardez la relation.

---

## Utilisation des clés primaires Optim

Une clé primaire est la/les colonne(s) contenant des valeurs identifiant de manière unique chaque ligne d'une table. Une table de base de données doit posséder une clé primaire pour Optim afin d'insérer, de mettre à jour, de restaurer ou de supprimer des données à partir d'une table de base de données. Optim utilise des clés primaires définies dans la base de données. Toutefois, vous pouvez également définir des clés primaires Optim pour compléter les clés primaires dans la base de données.

Une clé primaire est nécessaire :

- Par exemple, pour toute table visitée plusieurs fois dans un processus, une table enfant possédant deux ou plusieurs tables parent référencées dans la définition d'accès.
- Pour activer la fonction de pointage et de découpage d'une table initiale.

**Remarque :** Si une clé primaire n'est pas définie et est requise pour exécuter une tâche spécifique, un message d'erreur se produit.

### Types de clé primaire Optim

Vous pouvez définir deux types de clé primaire :

- Une clé primaire **explicite** s'applique à une table unique.
- Une clé primaire **générique** s'applique à toute table possédant le même nom, les mêmes noms de colonne, les mêmes spécifications d'attributs, mais différents ID de créateur.

Il n'y a aucune différence de fonction ou d'apparence entre les clés primaires génériques et explicites. Toutefois, si une table possède les deux types de clé, la clé primaire explicite est utilisée.

### Conventions de dénomination

Le nom qualifié complet d'une clé primaire est identique à celui de la table de base de données pour laquelle elle est définie. Ce nom se présente sous la forme suivante : *alias.creatorid.tablename*.

*alias* Alias identifiant la base de données où se trouve la table (1 à 12 caractères).

*creatorid*

Identificateur de créateur attribué à la table (1 à 64 caractères).

*tablename*

Nom de la table de base (1 à 64 caractères).

**Remarque :**

- La longueur totale des colonnes d'une clé primaire est limitée à 3 584 octets.

## Création d'une clé principale

Utilisez l'assistant Nouvelle clé primaire pour créer une clé primaire pour une table.

Avant de créer une clé primaire, un alias de magasin de données doit exister pour la table.

Pour créer une clé primaire :

1. Dans l'explorateur de référentiel, cliquez avec le bouton droit de la souris sur **Clés primaires Optim** et cliquez sur **Nouvelle clé primaire**. L'assistant Nouvelle clé primaire s'ouvre.
2. Suivez les étapes de l'assistant.  
Vous devez sélectionner une table.

La nouvelle clé primaire s'affiche dans l'éditeur de clé primaire.

Vous devez utiliser l'éditeur de clé primaire pour sélectionner les colonnes de clé primaire pour la table.

## Edition d'une clé primaire

L'éditeur de clé primaire vous permet d'éditer une clé primaire pour une table. Vous pouvez sélectionner des colonnes de clé et définir une clé primaire générique.

### Sélection de colonnes de clé primaire

L'éditeur de clé primaire vous permet de sélectionner des colonnes de clé primaire pour une table. Vous pouvez également définir l'ordre des colonnes de clé.

Pour sélectionner des colonnes de clé primaire pour une table :

1. Dans l'explorateur de référentiel, développez le noeud **Clés primaires Optim** et cliquez deux fois sur la clé primaire à éditer. L'éditeur de clé primaire s'ouvre.
2. Dans la liste **Colonnes disponibles**, sélectionnez les colonnes de clé.
3. Cliquez sur >> pour déplacer les colonnes vers la liste **Colonnes de clé**.  
Pour définir l'ordre des colonnes de clé :
  - a. Dans la liste **Colonnes de clé**, sélectionnez une colonne à déplacer.
  - b. Cliquez sur **Déplacer vers le haut** pour déplacer une colonne vers le haut de la liste, ou sur **Déplacer vers le bas** pour déplacer une colonne vers le bas de la liste.
4. Sauvegardez la clé primaire.

### Création d'une clé primaire générique

Utilisez l'éditeur de clé primaire pour créer une clé primaire Optim générique.

Certaines bases de données contiennent des ensembles de tables identiques, à l'exception de l'ID de créateur. Plutôt que de définir une clé primaire explicite pour chaque table, définissez une clé primaire générique pour toutes les tables qui portent le même nom de base, quel que soit leur ID de créateur.

Pour créer une clé primaire Optim générique :

1. Dans l'explorateur de référentiel, développez le noeud **Clés primaires Optim** et cliquez deux fois sur la clé primaire à éditer. L'éditeur de clé primaire s'ouvre.
2. Cliquez sur **Générique**.
3. Sauvegardez la clé primaire.

---

## Utilisation des mappes de table

Utilisez une mappe de table pour définir les spécifications des tables source et cible correspondantes contenant des données compatibles. Vous pouvez mapper des tables qui possèdent des noms différents, modifier les noms de table, exclure des tables d'un processus, ou inclure des mappes de colonne afin de mieux contrôler les données.

Utilisez une mappe de table pour :

- Placer les données dans un service de conversion, d'insertion ou de chargement ;
- Exclure une ou plusieurs tables d'un service de conversion, d'insertion ou de chargement ;
- Inclure une mappe de colonne.

En fonction du processus pour lequel vous utilisez une mappe de table, les deux ensembles de tables sont respectivement appelés tables source et tables cibles :

- Les **tables source** sont des tables extraites qui contiennent des données à utiliser dans un processus de conversion, d'insertion ou de chargement.
- Les **tables cibles** sont les tables dans lesquelles les données sont converties, insérées ou chargées.

**Remarque :** Les tables correspondantes peuvent avoir des ID de créateur ou des noms différents.

Les mappes de table sont stockées dans le référentiel Optim ou imbriquées dans un service de gestion des données.

Vous pouvez éventuellement utiliser une mappe de colonne pour n'importe quelle paire de tables dans une mappe de table. Les mappes de colonne identifient et mettent des colonnes en correspondance dans une paire de tables. Une mappe de colonne doit être utilisée lorsque les noms de colonne ou les attributs sont différents ou lorsque des transformations de données sont requises.

## Conventions de dénomination

Le nom qualifié complet d'une mappe de table possède le format suivant : *identificateur.nom*.

*identificateur*

Identificateur attribué à la mappe de table (1 à 8 caractères).

*nom*

Nom attribué à la mappe de table (1 à 12 caractères).

Il convient d'utiliser un ensemble logique de conventions de dénomination afin d'identifier leur propre utilisation et d'organiser les définitions pour en faciliter l'accès.

## Création d'une mappe de table

Utilisez l'assistant Nouvelle mappe de table pour créer une mappe de table.

Avant de créer une mappe de table, un alias de magasin de données de fichier doit exister pour la table source.

Pour créer une mappe de table :

1. Développez un dossier dans l'explorateur de référentiel qui contiendra la mappe de table, cliquez avec le bouton droit de la souris sur **Mappes de table**, puis cliquez sur **Nouvelle mappe de table**.  
L'assistant Nouvelle mappe de table s'ouvre.
2. Suivez les étapes de l'assistant.  
Vous devez sélectionner un alias de magasin de données de fichier pour la table source.

La nouvelle mappe de table s'affiche dans l'éditeur de mappe de table.

Vous devez utiliser l'éditeur pour définir des magasins de données cibles et des schémas avant de sauvegarder la mappe de table.

## Modification d'une mappe de table

Utilisez l'éditeur de mappe de table pour modifier les données cible et ajouter une mappe de colonne.

### Définition de données cibles par défaut dans une mappe de table

Utilisez l'éditeur de mappe de table pour définir les magasins de données cibles et les schémas par défaut dans une mappe de table.

Avant de définir des données cibles de mappe de table par défaut, un alias de magasin de données doit exister pour les données cibles.

Pour définir des données cibles par défaut dans une mappe de table :

1. Développez un dossier dans l'explorateur de référentiel qui contient la mappe de table, développez le noeud **Mappes de table** et cliquez deux fois sur la mappe de table. L'éditeur de mappe de table s'ouvre.
2. Sélectionnez l'onglet **Mappe de table**.
3. Dans la zone **Alias de magasin de données et mappe de schéma**, sélectionnez la ligne qui contient le magasin de données source.
4. Pour définir un alias de magasin de données cible, cliquez sur la cellule **Alias de magasin de données cible** et sélectionnez l'alias de magasin de données cible dans la liste.
5. Pour définir un schéma cible, cliquez sur la cellule **Schéma cible** et sélectionnez le schéma cible dans la liste.
6. Sauvegardez la mappe de table.

### Edition des données cibles dans une mappe de table

L'éditeur de mappe de table vous permet d'éditer des magasins de données cibles, des schémas et des tables pour les tables source d'une mappe de table.

Avant d'éditer les données cibles d'une mappe de table, un alias de magasin de données cible par défaut doit exister pour les données cibles.

Pour éditer des données cibles dans une mappe de table :

1. Développez un dossier dans l'explorateur de référentiel qui contient la mappe de table, développez le noeud **Mappes de table** et cliquez deux fois sur la mappe de table. L'éditeur de mappe de table s'ouvre.
2. Sélectionnez l'onglet **Mappe de table**.
3. Dans la zone **Mappe de table**, sélectionnez la ligne qui contient la table source.
4. Pour définir un alias de magasin de données cible, cliquez sur la cellule **Magasin de données cible** et entrez le nom d'un alias de magasin de données cible.
5. Pour définir un schéma cible, cliquez sur la cellule **Schéma cible** et sélectionnez le schéma cible dans la liste.
6. Pour définir une table cible, cliquez sur la cellule **Table cible** et sélectionnez la table cible dans la liste.
7. Sauvegardez la mappe de table.

### Ajout d'une mappe de colonne à une mappe de table

Utilisez l'éditeur de mappe de table pour ajouter une mappe de colonne à une paire de tables mappées.

Pour ajouter une mappe de colonne à une mappe de table :

1. Développez un dossier dans l'explorateur de référentiel qui contient la mappe de table, développez le noeud **Mappes de table** et cliquez deux fois sur la mappe de table. L'éditeur de mappe de table s'ouvre.
2. Sélectionnez l'onglet **Mappe de table**.
3. Dans la zone **Mappe de table**, sélectionnez la ligne qui contient les tables pour la mappe de colonne.
4. Cliquez sur **Ajouter une mappe de colonne**. La fenêtre Nouvelle mappe de colonne s'ouvre et présente une liste des mappes de colonne qui contient les tables sélectionnées.
5. Sélectionnez une mappe de colonne ou **Créer une mappe de colonne** pour créer une mappe de colonne en fonction des tables sélectionnées. Cliquez sur **OK**.
  - a. Si vous choisissez de créer une mappe de colonne, l'éditeur de mappe de colonne s'ouvre ; il est renseigné avec les colonnes provenant des tables sélectionnées.
  - b. Editez et sauvegardez la nouvelle mappe de colonne.
6. Sauvegardez la mappe de table.

---

## Utilisation des mappes de colonne

Une mappe de colonne fournit les spécifications nécessaires pour faire correspondre ou exclure des colonnes du traitement dans un service de gestion de données. La conversion, l'insertion et le chargement des services doivent référencer une mappe de table, qui peut faire référence à une ou plusieurs mappes. Vous pouvez également utiliser une mappe de colonne pour définir les transformations de données avec des fonctions ou des procédures de mappe de colonne.

Une mappe de colonne doit être utilisée lorsque les noms ou les attributs de colonne ne sont pas semblables, lorsque des conversions de données sont nécessaires ou lorsque vous excluez une ou plusieurs colonnes du traitement. Une mappe de colonne référencée dans un service de conversion, d'insertion ou de chargement peut modifier les données, évaluer l'ancienneté des dates ou convertir les devises. Vous pouvez définir des conversions de données via des fonctions ou des procédures de mappe de colonne.

Lorsque vous créez une mappe de colonne, vous devez choisir un alias de magasin de données de fichiers comme source des colonnes à mapper. De même, vous devez spécifier un alias de magasin de données pour les données cible.

Les mappes de colonne stockées dans le référentiel Optim peuvent être réutilisées ou partagées avec d'autres utilisateurs. Une mappe de colonne locale est stockée dans un service de gestion de données. Si ce n'est pas le cas, elle n'est pas disponible pour d'autres services. Si la mappe de table associée est définie en local pour un service, la mappe de table et la mappe de colonne ne sont disponibles que pour le service spécifique.

## Conventions de dénomination

Le nom qualifié complet d'une mappe de colonne se compose de deux parties : *identifier.name*.

*identifier*

Identificateur attribué à la mappe de colonne (1 à 8 caractères).

*name* Nom attribué à la mappe de colonne (1 à 12 caractères).

Lorsque vous créez des mappes de colonne, il est utile d'utiliser un ensemble logique de conventions de dénomination pour identifier et organiser les définitions et en faciliter l'accès.

## Règles de compatibilité pour les mappes de colonne

Les classes de données et leurs types de données associés suivants sont pris en charge. Ces classes de données jouent un rôle important pour la compatibilité des données lorsque vous spécifiez des valeurs de colonne dans des relations et des mappes de colonne.

### Caractère

SGBD	Types de données
DB2	CHAR, VARCHAR, CLOB
Oracle	CHAR, VARCHAR2, LONG, CLOB, NCLOB, NCHAR, NVARCHAR
Sybase ASE	CHAR, VARCHAR, TXT
SQL Server	CHAR, VARCHAR, TXT
Informix	CHAR, VARCHAR, TXT

**Remarque :** Les colonnes contenant des caractères mono-octets ne sont pas compatibles avec les colonnes contenant des caractères Unicode ou multi-octets.

### Numérique

SGBD	Types de données
DB2	INTEGER, SMALLINT, DECIMAL, FLOAT, DOUBLE
Oracle	NUMBER, FLOAT
Sybase ASE	TINYINT, INT, SMALLINT, DECIMAL, FLOAT, REAL, MONEY, SMALL MONEY
SQL Server	TINYINT, INT, SMALLINT, DECIMAL, FLOAT, REAL, MONEY, SMALL MONEY
Informix	INTEGER, SMALLINT, DECIMAL, FLOAT, REAL, DOUBLE PRECISION, SMALLFLOAT, SERIAL, MONEY, NUMERIC

### Binaire

SGBD	Types de données
DB2	CHAR (pour les données binaires), VARCHAR (pour les données binaires), BLOB
Oracle	RAW, LONG RAW
Sybase ASE	BINARY, VARBINARY, IMAGE
SQL Server	BINARY, VARBINARY, IMAGE
Informix	BYTE

### Booléen

SGBD	Types de données
Sybase ASE	BOOLEAN (TRUE ou FALSE)

### Date-heure

SGBD	Types de données
DB2	TIMESTAMP
Oracle	DATE, TIMESTAMP, TIMESTAMP WITH LOCAL TIME ZONE, TIMESTAMP WITH TIME ZONE
Sybase ASE	DATETIME, SMALL DATE TIME
SQL Server	DATETIME, SMALL DATE TIME

SGBD	Types de données
Informix	DATE, DATETIME

### Date

SGBD	Types de données
DB2	DATE
Oracle	DATE
Informix	DATE

### Heure

SGBD	Types de données
DB2	TIME

### Intervalle

SGBD	Types de données
Oracle	YEAR/MONTH INTERVAL, DAY/SECOND INTERVAL
Informix	YEAR/MONTH INTERVAL, DAY/TIME INTERVAL

## Création d'une mappe de colonne

Utilisez l'assistant Nouvelle mappe de colonne pour créer une mappe de colonne.

Avant de créer une mappe de colonne, un alias de magasin de données de fichier doit exister pour les données source et cible.

Pour créer une mappe de colonne :

1. Développez un dossier dans l'explorateur de référentiel qui contiendra la mappe de colonne, cliquez avec le bouton droit de la souris sur **Mappes de colonne**, puis cliquez sur **Nouvelle mappe de colonne**. L'assistant Nouvelle mappe de colonne s'ouvre.
2. Suivez les étapes de l'assistant.  
Vous devez sélectionner un alias de magasin de données de fichier ainsi qu'une table pour les données source. Vous devez également sélectionner un alias de magasin de données ainsi qu'une table pour les données cibles.

La nouvelle mappe de colonne s'affiche dans l'éditeur de mappe de colonne.

## Edition d'une mappe de colonne

Utilisez l'éditeur de mappe de colonne pour éditer une mappe de colonne et appliquer les fonctions de masquage des données.

### Application d'une fonction de masquage des données

Utilisez l'éditeur de mappe de colonne pour appliquer et éditer une fonction pour une colonne.

Pour appliquer une fonction de masquage des données :

1. Développez un dossier dans l'explorateur de référentiel qui contient la mappe de colonne, développez le noeud **Mappes de colonne** et cliquez deux fois sur la mappe de colonne. L'éditeur de mappe de colonne s'ouvre.
2. Sélectionnez la colonne associée à la règle.

3. Cliquez sur **Appliquer la fonction**. La fenêtre Appliquer la fonction s'ouvre.
4. Sélectionnez la fonction à appliquer. Cliquez sur **OK**. Le nom de la fonction apparaît dans la colonne associée et l'éditeur de fonction s'ouvre dans l'éditeur de mappe de colonne.
5. Selon la fonction, exécutez l'une des étapes suivantes :  
Sélectionnez l'onglet Expression de fonction et éditez l'expression de fonction.  
Sélectionnez un onglet d'option et sélectionnez des options pour la fonction.
6. Sauvegardez la mappe de colonne.

### Mappage d'une colonne source

L'éditeur de mappe de colonne vous permet de mapper une colonne source à une cible. Les colonnes source et cible portant le même nom et possédant des types de données compatibles sont automatiquement mappées.

Pour mapper une colonne source :

1. Développez un dossier dans l'explorateur de référentiel qui contient la mappe de colonne, développez le noeud **Mappes de colonne** et cliquez deux fois sur la mappe de colonne. L'éditeur de mappe de colonne s'ouvre.
2. Cliquez sur une colonne source et sélectionnez un nom de colonne dans la liste.
3. Sauvegardez la mappe de colonne.

### Utilisation de procédures de mappe de colonne

Une procédure de mappe de colonne est une procédure utilisée pour masquer ou transformer les données d'une colonne lorsque vous exécutez un service. Comme le nom l'indique, vous devez ajouter des procédures de mappe de colonne à la mappe de colonne. Vous pouvez écrire les procédures de mappe de colonne à l'aide du langage de script Lua.

#### Création d'une procédure de mappe de colonne :

Utilisez l'éditeur de script Lua pour créer une procédure de mappe de colonne.

Pour créer une procédure de mappe de colonne :

1. Développez un dossier dans l'explorateur de référentiel qui contient la mappe de colonne, développez le noeud **Mappes de colonne** et cliquez deux fois sur la mappe de colonne. L'éditeur de mappe de colonne s'ouvre.
2. Sélectionnez la colonne pour la procédure de mappe de colonne.
3. Cliquez sur **Ajouter une procédure**. L'éditeur de script Lua s'ouvre.
4. Créez et sauvegardez la procédure. La colonne associée avec la procédure affiche **Procédure**.
5. Sauvegardez la mappe de colonne.

#### Edition d'une procédure de mappe de colonne :

Utilisez l'éditeur de script Lua pour éditer une procédure de mappe de colonne.

Pour éditer une procédure de mappe de colonne :

1. Développez un dossier dans l'explorateur de référentiel qui contient la mappe de colonne, développez le noeud **Mappes de colonne** et cliquez deux fois sur la mappe de colonne. L'éditeur de mappe de colonne s'ouvre.
2. Sélectionnez la colonne associée à la procédure de mappe de colonne.
3. Cliquez sur **Editer la procédure**. L'éditeur de script Lua s'ouvre.
4. Editer et sauvegarder la procédure.
5. Sauvegardez la mappe de colonne.

### Saisie des valeurs pour les paramètres dans une procédure de mappe de colonne :

Utilisez l'éditeur de mappe de colonne pour entrer les valeurs pour les paramètres dans une procédure de mappe de colonne.

Pour entrer des valeurs pour les paramètres dans une procédure de mappe de colonne :

1. Développez un dossier dans l'explorateur de référentiel qui contient la mappe de colonne, développez le noeud **Mappes de colonne** et cliquez deux fois sur la mappe de colonne. L'éditeur de mappe de colonne s'ouvre.
2. Sélectionnez la colonne qui contient la procédure. L'éditeur **Procédure** s'ouvre dans l'éditeur de mappe de colonne.
3. Dans la zone **Paramètres de procédure**, entrez les valeurs des colonnes séparément pour les paramètres. Par exemple : abc, def.
4. Sauvegardez la mappe de colonne.

---

## Chapitre 4. Conception de services de gestion des données

Utilisez un service de gestion des données pour extraire, convertir, charger ou insérer des données. Un service utilise des définitions d'accès pour définir les données à extraire. Un service utilise des mappes de table et de colonne pour mapper des données source et cible. Vous pouvez masquer les données en appliquant une règle de confidentialité des données à une entité traitée par un service. Vous pouvez également transformer les requêtes de processus d'un répertoire Optim en services de gestion des données. Utilisez Optim Manager en mode imbriqué pour tester les services.

---

### Utilisation des services d'extraction

Utilisez un service d'extraction pour copier un ensemble de lignes liées à partir d'une ou de plusieurs tables et enregistrer ces lignes dans un magasin de données de fichiers.

Le service d'extraction indique l'ensemble de paramètres nécessaires à l'extraction des données et des définitions d'objets depuis les tables source et le magasin de données de fichiers dans lequel stocker les informations extraites.

Le service d'extraction extrait toujours les définitions des tables et des colonnes. Ces définitions sont utilisées pour créer les tables cible, si nécessaire. Vous pouvez également choisir d'extraire d'autres définitions d'objets, notamment des clés primaires, des relations et des index.

### Conventions de dénomination

Le nom qualifié complet d'un service d'extraction se compose comme suit : *identifieur.name*.

*identifieur*

Identificateur servant de préfixe pour le nom du service d'extraction (1 à 8 caractères).

*name* Nom attribué au service d'extraction (1 à 12 caractères).

Lorsque vous créez des services d'extraction, il peut être utile d'utiliser un ensemble logique de conventions de dénomination afin d'identifier l'utilisation de chaque service et de les organiser pour en faciliter l'accès.

### Création d'un service d'extraction

Utilisez l'assistant Nouveau service pour créer un service d'extraction.

Un service d'extraction nécessite une définition d'accès. Au cours de la création d'un service, vous pouvez sélectionner ou créer une définition d'accès.

Pour créer un service d'extraction :

1. Développez un dossier dans l'explorateur de référentiel qui contiendra le service d'extraction, cliquez avec le bouton droit de la souris sur **Services**, puis cliquez sur **Nouveau service**. L'assistant Nouveau service s'ouvre.
2. Suivez les étapes de l'assistant.  
Vous devez sélectionner ou créer une définition d'accès. Vous devez également entrer un nom pour le magasin de données de fichier cible.

Le nouveau service d'extraction s'affiche dans l'éditeur de service d'extraction.

## Modification d'un service d'extraction

Utilisez l'éditeur de service d'extraction pour indiquer des options de traitement de service d'extraction, notamment les objets de données à extraire, la conversion, la compression de fichiers et les remplacements de variables.

### Définition des options du processus d'extraction

L'onglet **Propriétés du service** de l'éditeur de service d'extraction vous permet de spécifier les options de traitement du service d'extraction.

Vous pouvez spécifier les options de traitement du service d'extraction suivantes :

- Gérez le nombre de connexions à la base de données. L'augmentation des connexions à la base de données améliore les performances lorsque de grandes quantités de données sont traitées, en permettant à plusieurs unités d'exécution d'extraire des lignes simultanément.
- Limitez le nombre de lignes extraites.
- Autorisez les fichiers joints.
- Incluez des informations statistiques dans le rapport du service d'extraction.

Pour spécifier les options de traitement du service d'extraction :

1. Développez un dossier dans l'explorateur de référentiel qui contient le service d'extraction, développez le noeud **Services** et cliquez deux fois sur le service d'extraction à éditer. L'éditeur de service d'extraction s'ouvre.
2. Sélectionnez l'onglet **Propriétés du service**.
3. Editez les options de traitement du service.
4. Sauvegardez le service d'extraction.

### Définition des options d'objet et de groupement

Utilisez l'onglet **Données et objets** dans l'éditeur de service d'extraction pour spécifier les objets de données à extraire.

Vous pouvez spécifier les options suivantes :

- Indiquez si le service devrait extraire des données uniquement, des objets uniquement, ou les deux.
- Sélectionnez les objets à extraire.

Pour spécifier les objets de données à extraire :

1. Développez un dossier dans l'explorateur de référentiel qui contient le service d'extraction, développez le noeud **Services** et cliquez deux fois sur le service d'extraction à éditer. L'éditeur de service d'extraction s'ouvre.
2. Sélectionnez l'onglet **Données et objets**.
3. Editez les options de traitement du service.
4. Sauvegardez le service d'extraction.

### Définition des options de conversion

L'onglet **Conversion** de l'éditeur de service d'extraction vous permet de spécifier les options de conversion. Vous pouvez convertir des données extraites dans le service.

Vous pouvez spécifier les options suivantes :

- Indiquez si le service devrait convertir les données extraites.
- Nombre maximal de lignes supprimées, qui ne peuvent pas être converties, à autoriser avant d'arrêter le service.

Pour spécifier les options de conversion :

1. Développez un dossier dans l'explorateur de référentiel qui contient le service d'extraction, développez le noeud **Services** et cliquez deux fois sur le service d'extraction à éditer. L'éditeur de service d'extraction s'ouvre.
2. Sélectionnez l'onglet **Conversion**.
3. Editez les options de traitement du service.
4. Sauvegardez le service d'extraction.

### **Définition des options de compression de fichier**

L'onglet **Options de compression de fichier** de l'éditeur de service d'extraction vous permet de spécifier les options de compression du magasin de données de fichier ou des tables extraites.

Vous pouvez spécifier les options suivantes :

- Indiquez si le fichier ou les tables spécifiques sont compressés.
- Pour chaque table, vous pouvez choisir d'utiliser un seuil de compression ou la compression illimitée. La valeur de seuil est le volume minimal de réduction en taille que vous prévoyez d'atteindre en compressant la table. Entrez une valeur comprise entre 1 et 99 pour définir une valeur de seuil pour cette table.

Les méthodes de compression suivantes sont disponibles pour un magasin de données de fichier :

#### **Compression en ligne**

Les données sont compressées à mesure qu'elles sont extraites et avant qu'elles ne soient écrites dans le magasin de données de fichier. La compression en ligne possède un débit d'entrées-sorties plus faible en comparaison à la post-compression, mais utilise des ressources de base de données pour la durée du service d'extraction.

La compression en ligne nécessite un volume de ressources de stockage moins élevé au cours du processus d'extraction en comparaison à la post-compression.

#### **Post-compression**

Les données sont extraites et écrites dans un magasin de données de fichier non compressé. Au cours de la deuxième étape, Optim lit le magasin de données de fichier non compressé et écrit une version compressée du magasin de données de fichier. L'avantage de la post-compression réside dans le fait que ces connexions à la base de données sont fermées plus tôt avec la post-compression qu'avec la compression en ligne. Toutefois, la post-compression implique une augmentation du temps total écoulé, car le magasin de données de fichier non compressé doit être fermé, lu, puis une version compressée doit être créée.

Les sites touchés par les conflits de ressource de base de données sont susceptibles de trouver la post-compression utile car elle raccourcit la durée pendant laquelle les ressources de base de données sont requises. Toutefois, la post-compression augmente le temps écoulé et les exigences en termes de stockage pour le traitement du service d'extraction. Bien qu'un espace de stockage plus volumineux soit nécessaire au cours de l'opération de compression, cet espace de stockage temporaire obligatoire est libéré une fois la compression terminée.

Pour spécifier les options de compression de fichier :

1. Développez un dossier dans l'explorateur de référentiel qui contient le service d'extraction, développez le noeud **Services** et cliquez deux fois sur le service d'extraction à éditer. L'éditeur de service d'extraction s'ouvre.
2. Sélectionnez l'onglet **Options de compression de fichier**.
3. Editez les options de traitement du service.
4. Sauvegardez le service d'extraction.

## Définition des options de variable

L'onglet **Variables** de l'éditeur de service d'extraction vous permet de remplacer les valeurs par défaut des variables.

Pour spécifier les options de variable :

1. Développez un dossier dans l'explorateur de référentiel qui contient le service d'extraction, développez le noeud **Services** et cliquez deux fois sur le service d'extraction à éditer. L'éditeur de service d'extraction s'ouvre.
2. Sélectionnez l'onglet **Variables**.
3. Entrez les valeurs de remplacement des valeurs par défaut.
4. Sauvegardez le service d'extraction.

## Modification d'une définition d'accès pour un service d'extraction

Utilisez l'assistant Modification d'une définition d'accès pour modifier une définition d'accès associée à un service d'extraction.

Pour modifier une définition d'accès associée à un service d'extraction :

1. Développez un dossier dans l'explorateur de référentiel qui contient le service d'extraction, développez le noeud **Services** et cliquez deux fois sur le service d'extraction à éditer. L'éditeur de service d'extraction s'ouvre.
2. Cliquez sur **Modifier**. L'assistant Modifier la définition d'accès s'ouvre.
3. Suivez les étapes de l'assistant.  
Sélectionnez une définition d'accès ou créer une définition d'accès locale.
4. Sauvegardez le service d'extraction.

---

## Utilisation des services de conversion

Utilisez un service de conversion pour transformer les données dans un magasin de données de fichiers. Vous pouvez transformer les données pour garantir la confidentialité des données ou transformer systématiquement les données afin de répondre aux exigences de test de votre application.

Indiquez une mappe de table pour faire correspondre les tables du magasin de données de fichiers source à celles du magasin de données de fichiers cible ou pour exclure les tables du service de conversion.

Utilisez les mappes de colonne de la mappe de table pour indiquer les données à convertir et la méthode de conversion.

## Conventions de dénomination

**Remarque :** Le nom qualifié complet d'un service de conversion se compose comme suit : *identifieur.name*.

*identifieur*

Identificateur servant de préfixe pour le nom du service de conversion (1 à 8 caractères).

*name* Nom attribué au service de conversion (1 à 12 caractères).

Lorsque vous créez un service de conversion, il peut être utile d'utiliser un ensemble logique de conventions de dénomination afin d'identifier l'utilisation de chaque service et de les organiser pour en faciliter l'accès.

## Création d'un service de conversion

Utilisez l'assistant Nouveau service pour créer un service de conversion.

Un service de conversion nécessite une mappe de table. Au cours de la création d'un service, vous pouvez sélectionner ou créer une mappe de table.

Pour créer un service de conversion :

1. Développez un dossier dans l'explorateur de référentiel qui contiendra le service, cliquez avec le bouton droit de la souris sur **Services**, puis cliquez sur **Nouveau service**. L'assistant Nouveau service s'ouvre.
2. Suivez les étapes de l'assistant.  
Vous devez sélectionner ou créer une mappe de table. Vous pouvez entrer un nom pour le magasin de données de fichier cible ou choisir d'écraser le magasin de données de fichier source.

Le nouveau service de conversion s'affiche dans l'éditeur de service de conversion.

## Modification d'un service de conversion

Utilisez l'éditeur de service de conversion pour indiquer des options de traitement de service de conversion.

### Définition des options du processus de conversion

L'onglet **Options de traitement** de l'éditeur de service de conversion vous permet de spécifier les options de traitement du service de conversion.

Vous pouvez spécifier les options de traitement du service de conversion suivantes :

- Définissez une limite de ligne à écarter.
- Comprimez le fichier cible.
- Autorisez les fichiers joints.

Pour spécifier les options de traitement du service de conversion :

1. Développez un dossier dans l'explorateur de référentiel qui contient le service d'extraction, développez le noeud **Services** et cliquez deux fois sur le service de conversion à éditer. L'éditeur de service de conversion s'ouvre.
2. Sélectionnez l'onglet **Options de traitement**.
3. Editez les options de traitement du service.
4. Sauvegardez le service de conversion.

### Modification d'une mappe de table pour un service de conversion

Utilisez l'assistant Modifier la mappe de table pour modifier une mappe de table associée à un service de conversion.

Pour modifier une mappe de table associée à un service de conversion :

1. Développez un dossier dans l'explorateur de référentiel qui contient le service de conversion, développez le noeud **Services** et cliquez deux fois sur le service de conversion à éditer. L'éditeur de service de conversion s'ouvre.
2. Cliquez sur **Modifier**. L'assistant Modifier une mappe de table s'ouvre.
3. Suivez les étapes de l'assistant.  
Sélectionnez une mappe de table ou choisissez de créer une mappe de table locale.
4. Sauvegardez le service de conversion.

---

## Utilisation des services d'insertion

Utilisez un service d'insertion pour insérer les données d'un magasin de données de fichiers dans une base de données cible.

Utilisez les mappes de table et de colonne pour mapper les données de la source vers la cible. Vous devez utiliser une mappe de table pour indiquer la cible des tables dans le magasin de données de fichiers. Vous pouvez également utiliser les mappes de colonne d'une ou de plusieurs tables cible. Les mappes de colonne vous permettent d'indiquer les données source de chaque colonne et de les transformer éventuellement avant leur insertion.

Si des tables du magasin de données de fichiers n'existent pas dans la cible, utilisez l'utilitaire de création pour les créer.

## Conventions de dénomination

Le nom qualifié complet d'un service d'insertion se compose comme suit : *identifier.name*.

*identifier*

Identificateur servant de préfixe pour le nom du service d'insertion (1 à 8 caractères).

*name* Nom attribué au service d'insertion (1 à 12 caractères).

Lorsque vous créez des services d'insertion, il peut être utile d'utiliser un ensemble logique de conventions de dénomination pour identifier l'utilisation de chaque service et les organiser pour en faciliter l'accès.

## Création d'un service d'insertion

Utilisez l'assistant Nouveau service pour créer un service d'insertion.

Un service d'insertion nécessite une mappe de table. Au cours de la création d'un service, vous pouvez sélectionner ou créer une mappe de table.

Pour créer un service d'insertion :

1. Développez un dossier dans l'explorateur de référentiel qui contiendra le service, cliquez avec le bouton droit de la souris sur **Services**, puis cliquez sur **Nouveau service**. L'assistant Nouveau service s'ouvre.
2. Suivez les étapes de l'assistant.  
Vous devez sélectionner ou créer une mappe de table.

Le nouveau service d'insertion s'affiche dans l'éditeur de service d'insertion.

## Edition d'un service d'insertion

L'éditeur de service d'insertion vous permet de spécifier les options de traitement du service d'insertion. Vous pouvez spécifier des options permettant d'insérer et de supprimer des lignes, mais également de gérer des déclencheurs et des contraintes.

### Définition des options du processus d'insertion

L'onglet **Options de traitement** de l'éditeur de service d'insertion vous permet de spécifier les options de traitement du service d'insertion.

Vous pouvez spécifier les options de traitement suivantes :

- Sélectionnez le type de traitement à exécuter. Spécifiez les paramètres permettant de verrouiller les tables, définir une fréquence de validation et définir une limite de ligne à écarter.
- Supprimez des lignes de certaines ou de toutes les tables. Cette suppression s'avère utile pour restaurer des données au cours du test. Si une ligne ne peut pas être supprimée pour quelque raison, toutes les lignes supprimées depuis la dernière validation sont restaurées et le processus d'insertion prend fin.
- Désactivez les déclencheurs de base de données.
- Désactivez les contraintes de base de données.

Le traitement des lignes inclut les options suivantes :

### Insérer

Insère de nouvelles lignes dans les tables.

- Si la clé primaire d'une ligne des données source *ne correspond pas* à la clé primaire d'une ligne de la table cible, la ligne est insérée.
- Si la clé primaire d'une ligne des données source *correspond* à la clé primaire d'une ligne de la table cible, la ligne est ignorée et marquée comme étant annulée.

**Mixte** Met à jour, insère ou met à jour/insère en fonction de votre sélection pour chaque table dans la fenêtre Spécifier par table. Pour utiliser la fenêtre Spécifier par table, cliquez sur **Spécifier par table** et sélectionnez une option de traitement pour chaque table.

- Si vous sélectionnez **Mixte**, mais n'effectuez aucune sélection dans la fenêtre Spécifier par table, ou si vous définissez toutes les tables dans la même sélection, l'option de traitement est modifiée et indique le processus utilisé pour toutes les tables.

**Remarque :** Vous ne pouvez pas sélectionner **Mixte** si l'option **Toutes les lignes seront supprimées** est sélectionnée dans la zone **Options de suppression de ligne**.

### Mettre à jour

Met à jour les lignes dans les tables. Les tables doivent posséder une clé primaire.

- Si la clé primaire d'une ligne des données source *correspond* à la clé primaire d'une ligne de la table cible, la ligne est mise à jour.
- Si la clé primaire d'une ligne des données source *ne correspond pas* à la clé primaire d'une ligne de la table cible, la ligne est signalée comme étant en échec.

**Remarque :** Vous ne pouvez pas sélectionner **Mettre à jour** si l'option **Toutes les lignes seront supprimées** est sélectionnée dans la zone **Options de suppression de ligne**.

### Mettre à jour/Insérer

Met à jour et insère des lignes dans les tables. Les tables doivent posséder une clé primaire.

- Si la clé primaire d'une ligne des données source *ne correspond pas* à la clé primaire d'une ligne de la table cible, la ligne est insérée.
- Si la clé primaire d'une ligne des données source *correspond* à la clé primaire d'une ligne de la table cible, la ligne est mise à jour.

**Remarque :** Vous ne pouvez pas sélectionner **Mettre à jour/Insérer** si l'option **Toutes les lignes seront supprimées** est sélectionnée dans la zone **Options de suppression de ligne**.

Pour spécifier les options du processus d'insertion :

1. Développez un dossier dans l'explorateur de référentiel qui contient le service d'insertion, développez le noeud **Services** et cliquez deux fois sur le service d'insertion à éditer. L'éditeur de service d'insertion s'ouvre.
2. Sélectionnez l'onglet **Options de traitement**.
3. Editez les options de traitement du service.
4. Sauvegardez le service d'insertion.

## Modification d'une mappe de table pour un service d'insertion

Utilisez l'assistant Modifier la mappe de table pour modifier une mappe de table associée à un service d'insertion.

Pour modifier une mappe de table associée à un service d'insertion :

1. Développez un dossier dans l'explorateur de référentiel qui contient le service d'insertion, développez le noeud **Services** et cliquez deux fois sur le service d'insertion à éditer. L'éditeur de service d'insertion s'ouvre.

2. Cliquez sur **Modifier**. L'assistant Modifier une mappe de table s'ouvre.
3. Suivez les étapes de l'assistant.  
Sélectionnez une mappe de table ou choisissez de créer une mappe de table locale.
4. Sauvegardez le service d'insertion.

---

## Utilisation des services de chargement

Utilisez un service de chargement pour convertir le contenu d'un magasin de données de fichiers en un format approprié pour un chargeur DBMS particulier, puis démarrer l'utilitaire de chargement de la base de données correspondante si cela a été spécifié.

Le service de chargement génère un fichier de données au format approprié pour chaque table dans le magasin de données de fichiers et un fichier SQL ou BAT (exécution par lots) en fonction du SGBD qui contient la syntaxe nécessaire au démarrage du chargeur de bases de données.

Un service de chargement contient les paramètres utilisés pour préparer les données d'un chargeur de SGBD et les instructions requises au traitement du chargement. Indiquez une mappe de table dans le service de chargement pour mapper la cible des données à charger. Utilisez des mappes de colonne facultatives dans le service de chargement pour transformer les données avant le chargement.

Les données à charger doivent se trouver dans un magasin de données de fichiers.

### Chargement et insertion

Optim peut déplacer les données dans une base de données à l'aide d'un service de chargement ou d'insertion. Tenez compte des points suivants lorsque vous choisissez la méthode à employer :

- Le volume des données et la vitesse d'utilisation de l'utilitaire de chargement des bases de données peuvent contrebalancer les avantages du service d'insertion.
- Les données peuvent contenir des cycles d'intégrité référentielle (RI) dépassant les capacités du service d'insertion pour insérer correctement toutes les données.
- L'utilitaire de chargement des bases de données nécessite le contrôle exclusif de la base de données et empêche l'accès de l'utilisateur lors du service de chargement. Les autres utilisateurs peuvent accéder à la base de données lorsque le service d'insertion est exécuté.
- L'utilitaire de chargement des bases de données insère les nouvelles données ou remplace les données existantes. Le service d'insertion permet de traiter la mise à jour/l'insertion en une étape.

### Noms des fichiers de sortie

Un service de chargement génère les types de fichier suivants pour prendre en charge le processus de chargement de l'utilitaire de base de données :

#### Fichiers de données

Les fichiers de données contiennent les données à charger, préparées dans un format approprié au SGBD que vous utilisez. Optim génère un fichier de données pour chaque table dans le magasin de données de fichiers. Les fichiers de données sont nommés de la même manière que le magasin de données de fichiers, mais contiennent des extensions de noms de fichiers numérotées de façon séquentielle. Par exemple, un magasin de données de fichiers nommé *demo.xf* contenant trois tables génère trois fichiers de données nommés : *demo.001*, *demo.002*, et *demo.003*.

#### Fichiers message

Les fichiers message contiennent des informations que l'utilitaire de chargement des bases de données génère lors du service de chargement. Généralement, le service de chargement dispose d'un fichier message. Le fichier message est nommé de la même manière que le magasin de données de fichiers, mais contient l'extension *.msg*. Par exemple, si le magasin de données de fichiers est nommé *demo.xf*, le fichier message est nommé *demo.msg*.

## Fichier SQL

Pour DB2, un fichier SQL est généré avec une instruction pour chaque table cible contenant la syntaxe de chargeur afin d'exécuter manuellement le chargeur. Le fichier SQL est nommé de la même manière que le magasin de données de fichiers, mais possède l'extension *.sql*.

## Fichier BAT

Pour Oracle, Sybase ASE, SQL Server et Informix, un fichier BAT, contenant la syntaxe pour exécuter manuellement le chargeur pour chaque table, est généré. Un fichier BAT est généré pour chaque alias de magasin de données indiqué dans la mappe de table. Chaque fichier BAT se trouve dans le répertoire avec le magasin de données de fichiers correspondant converti. Si vous choisissez d'exécuter manuellement le chargeur, le fichier BAT doit être modifié (dans le bloc-notes, par exemple) pour remplacer une chaîne de huit points d'interrogation par des informations spécifiques au mot de passe (excepté Informix).

De plus, un fichier de format est généré pour chaque fichier de données. Mis à part son extension, un fichier de format possède le même nom que le fichier de données correspondant. Si moins de 500 tables doivent être chargées, l'extension du nom du fichier de format correspond à l'extension du nom du fichier de données, à laquelle sont ajoutés 500. Par exemple, si trois fichiers de données nommés *demo.001*, *demo.002* et *demo.003*, les fichiers de format correspondants sont respectivement nommés *demo.501*, *demo.502* et *demo.503*.) Si le nombre de tables est supérieur à 500, un algorithme de génération d'extensions de fichiers plus complexe est employé.

**Remarque :** Si votre serveur de fichiers n'autorise pas des noms de fichiers supérieurs à 8 caractères et que le nom du magasin de données de fichiers est long, le service de chargement échoue. La meilleure solution consiste à éviter d'utiliser de longs noms de fichiers pour les magasins de données de fichiers. Si nécessaire, vous pouvez copier et renommer un fichier avant de l'utiliser pour un service de chargement.

## Conventions de dénomination

Le nom qualifié complet d'un service de chargement se compose comme suit : *identifier.name*.

*identifier*

Identificateur servant de préfixe pour le nom du service (1 à 8 caractères).

*name* Nom attribué au service (1 à 12 caractères).

Lorsque vous créez des services de chargement, il peut être utile d'utiliser un ensemble logique de conventions de dénomination pour identifier l'utilisation de chaque service et les organiser pour en faciliter l'accès.

## Création d'un service de chargement

Utilisez l'assistant Nouveau service pour créer un service de chargement.

Un service de chargement nécessite une mappe de table. Au cours de la création d'un service, vous pouvez sélectionner ou créer une mappe de table.

Pour créer un service de chargement :

1. Développez un dossier dans l'explorateur de référentiel qui contiendra le service d'extraction, cliquez avec le bouton droit de la souris sur **Services**, puis cliquez sur **Nouveau service**. L'assistant Nouveau service s'ouvre.
2. Suivez les étapes de l'assistant.  
Vous devez sélectionner ou créer une mappe de table.

Le nouveau service de chargement s'affiche dans l'éditeur de service de chargement.

## Modification d'un service de chargement

Utilisez l'éditeur de service de chargement pour indiquer des options de traitement de chargement. Vous pouvez indiquer des options pour le processus de chargement et pour le chargeur SGBD de chaque magasin de données cible.

### Définition des options de chargement

L'onglet **Options de traitement** de l'éditeur de service de chargement vous permet de spécifier les options de traitement du service de chargement.

Vous pouvez spécifier les options suivantes :

- Sélectionnez la méthode d'exécution de plusieurs chargeurs de SGBD lorsque plusieurs alias de magasin de données sont utilisés : parallèle (pour exécuter différents chargeurs de SGBD simultanément) ou séquentielle (pour exécuter différents chargeurs de SGBD successivement).
- Arrêtez un chargeur de SGBD si une erreur se produit. Si plusieurs chargeurs de SGBD s'exécutent en séquence, le traitement prend fin pour le chargeur de SGBD avec une erreur et tous les chargeurs de SGBD suivants s'arrêtent.
- Désactivez les déclencheurs de base de données.
- Désactivez les contraintes de base de données.

Pour spécifier les options de traitement du service de chargement :

1. Développez un dossier dans l'explorateur de référentiel qui contient le service de chargement, développez le noeud **Services** et cliquez deux fois sur le service de chargement à éditer. L'éditeur de service de chargement s'ouvre.
2. Sélectionnez l'onglet **Options de traitement**.
3. Editez les options de traitement du service.
4. Sauvegardez le service de chargement.

### Définition des options du chargeur de SGBD

L'onglet **Options de chargement** de l'éditeur de service de chargement vous permet de spécifier les options de chaque chargeur de SGBD.

Vous pouvez spécifier les options suivantes :

- Un mode de traitement.
- Les options spécifiques à chaque alias de magasin de données cible.

En fonction du SGBD du magasin de données de fichier cible sélectionné, les modes de traitement suivants sont disponibles :

#### Insérer

Insère des lignes à partir du magasin de données de fichier source dans des tables cibles vides. Si les tables cibles contiennent des données, le chargeur renvoie une erreur.

#### Remplacer

Efface et remplace toutes les lignes existantes dans les tables cibles par les lignes provenant du magasin de données de fichier source. (L'option **Remplacer** est susceptible de consommer plus de ressources que l'option **Tronquer** car aucune journalisation n'est effectuée.)

#### Ajouter

Insère les lignes à partir du magasin de données de fichier source dans les tables cibles. Si les valeurs de clé primaire correspondent, les lignes en double sont supprimées ou insérées dans la table d'exceptions (si spécifiée).

## Tronquer

L'option **Tronquer** est identique à l'option **Remplacer**, à la différence près que la base de données ne journalise pas les lignes en cours de suppression et que l'option Tronquer nécessite que les contraintes RI soient désactivées.

Pour spécifier les options du chargeur de SGBD :

1. Développez un dossier dans l'explorateur de référentiel qui contient le service de chargement, développez le noeud **Services** et cliquez deux fois sur le service de chargement à éditer. L'éditeur de service de chargement s'ouvre.
2. Sélectionnez l'onglet **Options de chargement**.
3. Sélectionnez l'alias de magasin de données cible.
4. Editez les options de traitement du service.
5. Sauvegardez le service de chargement.

## Modification d'une mappe de table pour un service de chargement

Utilisez l'assistant Modifier la mappe de table pour modifier une mappe de table associée à un service de chargement.

Pour modifier une mappe de table associée à un service de chargement :

1. Développez un dossier dans l'explorateur de référentiel qui contient le service de chargement, développez le noeud **Services** et cliquez deux fois sur le service de chargement à éditer. L'éditeur de service de chargement s'ouvre.
2. Cliquez sur **Modifier**. L'assistant Modifier une mappe de table s'ouvre.
3. Suivez les étapes de l'assistant.  
Sélectionnez une mappe de table ou choisissez de créer une mappe de table locale.
4. Sauvegardez le service de chargement.

---

## Test d'un service de gestion des données

Utilisez Optim Manager en mode imbriqué pour tester un service de gestion de données.

Pour tester un service de gestion de données :

1. Développez un dossier dans l'explorateur de référentiel contenant le service d'extraction et développez le noeud **Services**.
2. Cliquez à l'aide du bouton droit de la souris sur le service, puis cliquez sur **Exécuter le service**. Optim Manager s'ouvre et la fenêtre Exécuter le service s'affiche.
3. Cliquez sur **Exécuter**. Pour contrôler la progression du service, utilisez l'onglet **Contrôle du service**.



---

## Chapitre 5. Masquage des données

Vous pouvez masquer des données telles que les numéros nationaux d'identité, les numéros de carte de crédit, les dates, les valeurs numériques et les informations personnelles. Utilisez une mappe de colonne pour entrer une fonction de masquage de données ou créer une procédure de mappe de colonne avec un script LUA. Utilisez un service de conversion pour transformer les données.

---

### Application d'une fonction de masquage des données

Utilisez l'éditeur de mappe de colonne pour appliquer et éditer une fonction pour une colonne.

Pour appliquer une fonction de masquage des données :

1. Développez un dossier dans l'explorateur de référentiel qui contient la mappe de colonne, développez le noeud **Mappes de colonne** et cliquez deux fois sur la mappe de colonne. L'éditeur de mappe de colonne s'ouvre.
2. Sélectionnez la colonne associée à la règle.
3. Cliquez sur **Appliquer la fonction**. La fenêtre Appliquer la fonction s'ouvre.
4. Sélectionnez la fonction à appliquer. Cliquez sur **OK**. Le nom de la fonction apparaît dans la colonne associée et l'éditeur de fonction s'ouvre dans l'éditeur de mappe de colonne.
5. Selon la fonction, exécutez l'une des étapes suivantes :
  - Sélectionnez l'onglet Expression de fonction et éditez l'expression de fonction.
  - Sélectionnez un onglet d'option et sélectionnez des options pour la fonction.
6. Sauvegardez la mappe de colonne.

---

### Fonctions de masquage des données

Les fonctions de masquage des données permettent de transformer ou de masquer les données sensibles de différentes manières.

#### Fonctions de recherche

Les fonctions de recherche permettent de sélectionner des valeurs à partir d'une table de recherche et de renseigner la table cible. Les fonctions de recherche et de hachage permettent de sélectionner des valeurs en fonction de la valeur source. La fonction de recherche aléatoire permet sinon de sélectionner des valeurs à partir d'une table de recherche, indépendamment de la valeur source.

#### Fonction de recherche

La fonction de recherche obtient la valeur pour une colonne cible à partir d'une table de consultation, selon la valeur de la colonne source. Il existe deux formes de fonction de recherche : à colonne simple et à colonnes multiples.

La fonction de recherche à colonne simple insère une valeur dans une colonne cible unique. La fonction de recherche à colonnes multiples insère des valeurs de plusieurs colonnes de tables de recherche dans les colonnes de destination correspondantes.

Vous pouvez entrer la fonction de recherche à colonnes multiples pour toute colonne source qui sera remplacée par une valeur de table de recherche. Toutefois, vous devez modifier la mappe de colonne pour supprimer les noms des colonnes source restantes qui seront également remplacées.

Le paramètre *ignore* vous permet d'ignorer la table de recherche et d'utiliser une valeur source lorsqu'une ligne de la colonne source indiquée contient une valeur spécifiée (NULL, SPACES (pour colonnes CHAR) ou valeur VARCHAR de longueur nulle).

Vous pouvez utiliser le paramètre *preserve* pour ignorer la table de recherche et utiliser une valeur source lorsqu'une ligne de la colonne source spécifiée contient une valeur précise (NULL, SPACES (pour colonnes CHAR ou valeur VARCHAR de longueur nulle). La variable *preserve* peut également être utilisée pour ignorer la table de recherche si une colonne source ne contient pas de valeur.

Si une correspondance n'est pas trouvée dans la table de recherche, une erreur de conversion est signalée.

La syntaxe est la suivante :

```
LOOKUP ( [sourcesearchcol, | SRCSEARCH=(sourcecoll,...,sourcecoln)]
         [dest=(coll, coln) , ]
         lktablename ( {LookupTableSearchcol
         | LKPSEARCH=(LookupTableSearchColl,...,LookupTableSearchColn),
         {value | values=(coll, coln) }
         [,cache | ,nocache ] )
         [,ignore=(colname ( spaces, null, zero_len ), )
         | PRESERVE=( [ NOT_FOUND, ] colname (spaces, null, zero_len),... ) ] )
```

*sourcesearchcol*

Pour les recherches à colonne unique, nom de la colonne de la table source contenant la valeur de recherche (facultatif). S'il n'est pas spécifié, le nom de la colonne cible est utilisé.

*SRCSEARCH=*

Pour les recherches à colonnes multiples, noms des colonnes de table source contenant les valeurs de recherche. Les noms de colonne doivent être inclus entre parenthèses et séparés par des virgules.

*dest=* Noms des colonnes de table cible dans lesquelles les valeurs provenant de la table de recherche sont insérées. (obligatoire pour les recherches à colonnes multiples.)

*coll, coln*

Noms des colonnes de la table cible. L'ordre des noms de colonnes doit correspondre aux colonnes de la table de recherche dans le paramètre *values=*.

*lktablename*

Nom de la table de recherche. Vous pouvez définir le nom de la table de recherche sous les formes suivantes : *dbalias.creatorid.tablename*, *creatorid.tablename* ou *tablename*. Si vous ne qualifiez pas complètement le nom de la table, les qualificatifs de la table cible sont utilisés.

*LookupTableSearchcol*

Pour les recherches à colonne unique, nom de la colonne d'une table de recherche qui contient une valeur à faire correspondre à la valeur de recherche de la colonne source.

*value* Nom de la colonne d'une table de recherche contenant la valeur de recherche convertie à insérer dans la cible. (obligatoire pour les recherches à colonne unique.)

*values=*

Noms des colonnes de table de recherche contenant les valeurs à insérer à la destination. (obligatoire pour les recherches à colonnes multiples.)

*coll, coln*

Noms des colonnes de la table de recherche. L'ordre des noms de colonnes doit correspondre aux colonnes de la table cible dans le paramètre *dest=*.

*cache* | *nocache*

Indiquez *cache* (par défaut) pour tenir à jour une table des valeurs de recherche trouvées en mémoire ou *nocache* pour annuler les valeurs trouvées. L'utilisation de la variable *cache* est plus rapide lors de l'extraction multiple d'une valeur, mais nécessite davantage de mémoire.

*ignore=*

Liste des colonnes source avec des valeurs insérées dans la cible au lieu de la valeur de la recherche lorsque la colonne contient une ligne avec une valeur définie (nulle, espaces, zéro ou varchar de longueur nulle).

*col* Nom de la colonne source.

Pour les recherches à colonne unique, entrez uniquement un nom de colonne.

Pour les recherches à colonnes multiples, l'ordre des noms de colonnes doit correspondre aux colonnes de la table cible dans le paramètre *dest=*. Le nombre de colonnes doit correspondre à celui défini dans le paramètre *dest=* et au moins une colonne doit contenir des valeurs. Pour ne pas spécifier de valeurs dans une colonne, n'entrez pas de valeur. Par exemple, *coln()*.

*null* Ignorez la table de recherche si la ligne de la colonne source contient une valeur nulle.

*SPACES*

Ignorez la table de recherche si la ligne de la colonne source contient une valeur SPACES. Pour colonnes CHAR uniquement.

*ZERO\_LEN*

Ignorez la table de recherche si la ligne de la colonne source contient une valeur VARCHAR de longueur nulle.

*preserve=*

Liste des colonnes source avec les valeurs insérées à la destination au lieu de la valeur de recherche lorsque la colonne source contient une valeur définie (NOT\_FOUND, null, spaces ou zero-length varchar).

*NOT\_FOUND*

Ignorez la table de recherche si aucune correspondance n'est trouvée pour la ligne de la colonne source.

#### **Remarque :**

*preserve=* et *ignore=* s'excluent mutuellement. *ignore=* ne sera plus utilisé dans une prochaine version.

Les opérandes *col*, *null*, *spaces* et *zero\_len* ont le même effet lorsqu'elles sont utilisées avec *preserve=* ou *ignore=*.

## **Exemple de colonne unique**

Utilisez la fonction de recherche pour convertir la valeur source d'une table de recherche en une valeur correspondante dans une autre table.

Par exemple, supposons que la colonne source STATE contienne des abréviations d'états (par exemple, NJ) et que la colonne cible doive contenir le nom de l'état complet (dans cet exemple, **New Jersey**). Une table de recherche nommée STATE\_LOOKUP contient une colonne (CODE) pour les abréviations ou codes d'état et une colonne (NAME) pour les noms correspondants.

Pour obtenir la valeur de la colonne cible à l'aide de la table STATE\_LOOKUP, indiquez ce qui suit :

```
LOOKUP(STATE,STATE_LOOKUP(CODE,NAME))
```

La fonction de recherche recherche une valeur dans la colonne CODE de la table STATE\_LOOKUP correspondant à la valeur (NJ) de la colonne STATE de la table source. Lorsqu'une correspondance est disponible, la fonction insère la valeur correspondante à partir de la colonne NAME (**New Jersey**) de la colonne cible.

## Exemple de colonnes multiples

Utilisez la fonction de recherche pour insérer des valeurs à partir de colonnes d'une ligne de table de recherche dans une ligne de table cible, en fonction d'une valeur contenue dans une colonne source.

Par exemple, à partir d'une colonne source (SOC\_SEC) contenant des numéros de sécurité sociale, vous pouvez remplacer les valeurs dans les colonnes de destination (FIRST\_NAME et LAST\_NAME) par les prénoms et noms d'une table de recherche. Une table nommée NAME\_LOOKUP contient une colonne (SSN) dotée des numéros de sécurité sociale de la table source, ainsi que des colonnes (FIRST\_MASK and LAST\_MASK) pour masquer les noms correspondants dans la cible.

Pour remplacer les noms dans la table cible en fonction d'un numéro de sécurité sociale, indiquez ce qui suit :

```
LOOKUP(SOC_SEC,DEST=(FIRST_NAME, LAST_NAME),  
NAME_LOOKUP(SSN,VALUES=(FIRST_MASK, LAST_MASK)))
```

La fonction de recherche recherche une valeur dans la colonne SSN de la table NAME\_LOOKUP correspondant à la valeur dans la table source de la colonne SOC\_SEC. Lorsqu'une correspondance est disponible, la fonction insère les valeurs correspondantes des colonnes FIRST\_MASK et LAST\_MASK de la table de recherche dans les colonnes cible correspondantes.

## Exemple de fonction Ignore

Utilisez l'instruction suivante pour étendre l'exemple de colonne unique, où vous souhaitez utiliser les valeurs *NULL* et *SPACES* au lieu des valeurs de la table de recherche :

```
LOOKUP(STATE,STATE_LOOKUP(CODE,NAME),  
IGNORE=(STATE(NULL,SPACES)))
```

## Exemple de fonction NoCache

Utilisez l'instruction suivante pour étendre l'exemple de colonne unique, où vous ne souhaitez pas conserver une table de valeurs de recherche trouvées en mémoire :

```
LOOKUP(STATE,STATE_LOOKUP(CODE,NAME),NOCACHE)
```

## Fonction de recherche hachée

La fonction de recherche hachée obtient la valeur pour une colonne cible à partir d'une table de consultation, selon la valeur hachée dérivée d'une colonne source. La fonction de recherche hachée vous permet de masquer des données avec cohérence lorsque vous utilisez les mêmes tables source et de recherche dans un environnement.

La colonne source hachée ne doit pas forcément être une colonne qui sera remplacée par des valeurs de la table de consultation.

La fonction de recherche hachée est sensible à la casse. Par exemple, les valeurs source John et JOHN seront hachées sur des valeurs différentes. Vous pouvez utiliser le paramètre TRIM pour convertir la valeur source en majuscules avant qu'elle ne soit hachée.

La fonction de recherche hachée possède deux formats : colonne unique et colonne multiple. Le format en colonne unique permet d'insérer une valeur dans une colonne cible unique. Le format en colonne multiple permet d'insérer des valeurs provenant de plusieurs colonnes de table de consultation dans les colonnes cibles correspondantes, en se basant sur une valeur hachée unique à partir d'une colonne source.

Vous pouvez entrer dans la fonction de recherche hachée au format en colonne multiple n'importe quelle colonne source qui sera remplacée par des valeurs de table de consultation. Vous devez toutefois modifier la mappe de colonne de sorte à supprimer les noms des colonnes source restantes qui seront également remplacées.

La table de consultation doit contenir une colonne clé contenant des valeurs de chiffre séquentiel sans espace et les colonnes restantes contiennent des valeurs de remplacement. La colonne clé doit être un type de données numérique. La table de consultation est généralement indexée. La fonction hache une colonne source pour faire dériver les nombres séquentiels de 1 vers la valeur maximale de la colonne clé de la table de consultation. La valeur hachée de la table source est comparée aux nombres séquentiels de la table de consultation et les valeurs de la ligne de la table de consultation correspondante sont insérées au niveau de la cible.

Si la colonne source utilisée pour dériver la valeur hachée contient certaines valeurs (NULL, espaces (pour les colonnes CHAR) et VARCHAR de longueur zéro), la valeur n'est pas hachée et les valeurs réservées suivantes sont utilisées comme clés dans la table de consultation :

Valeur source	Clé de la table de consultation
NULL	-1
Espaces (CHAR ou VARCHAR)	-2
VARCHAR de longueur zéro	-3

La table de consultation doit contenir une ligne pour chacun de ces nombres, vous permettant d'insérer une valeur de consultation pour chacune de ces valeurs source. Si l'une de ces valeurs source est identifiée, mais qu'aucun nombre correspondant ne se trouve dans la table de consultation, une erreur de conversion est signalée.

Le paramètre *ignore* vous permet d'ignorer la table de consultation et d'utiliser une valeur source lorsqu'une ligne d'une colonne source spécifique contient une valeur spécifiée (NULL, SPACES (pour les colonnes CHAR) ou VARCHAR de longueur égale à zéro).

Le paramètre *preserve* vous permet d'ignorer la table de consultation et d'utiliser une valeur source lorsqu'une ligne d'une colonne source spécifique contient une valeur spécifiée (NULL, SPACES (pour les colonnes CHAR) ou VARCHAR de longueur égale à zéro). Le paramètre *preserve* peut également être utilisé pour ignorer la table de consultation si une colonne source ne contient aucune valeur.

Le paramètre *trim* vous permet d'indiquer des caractères qui seront découpés à partir de la valeur source avant qu'elle ne soit hachée. Par exemple, si vous choisissez de découper des virgules à partir d'une valeur source, les valeurs Smith, John et Smith John seront hachées sur la même valeur. Vous pouvez également utiliser ce paramètre pour convertir la valeur source en majuscules avant qu'elle ne soit hachée.

Si la valeur source est convertie en majuscules, les caractères de découpage sont également convertis en majuscules.

Vous pouvez utiliser le paramètre *seed* pour varier le calcul réalisé par l'algorithme de hachage. La valeur hachée à partir de la colonne source et la valeur *seed* sont mises en correspondance avec un numéro séquentiel provenant de la table de consultation afin d'obtenir la valeur de remplacement pour la colonne cible.

La syntaxe est la suivante :

```
HASH_LOOKUP( [colsource,] [trim=( [car1car2 ] [\u] ),]
  dest=(col1, coln), nomtableconsult (recherche,
  { valeur | values=(col1, coln) } ) [ ,cache | ,nocache ]
  [,ignore=(col (spaces, null, zero_len ), )
  | PRESERVE=( [ NOT_FOUND, ] nomcol (spaces, null, zero_len), ) ] [,seed=n])
```

*colsource*

Nom de la colonne de table source à partir de laquelle les valeurs hachées sont dérivées (facultatif). S'il n'est pas spécifié, le nom de la colonne cible est utilisé.

*trim=* Liste des caractères à tronquer à partir de la valeur source avant qu'elle-même ne soit hachée en tant qu'option pour convertir la valeur source en majuscules avant qu'elle ne soit hachée. Si la valeur source de sortie est NULL ou si tous les espaces après les caractères ont été tronqués, la valeur source ne sera pas hachée et la valeur réservée appropriée lui sera affectée (-1 ou -2).

*car1car2...*

Caractères à découper à partir de la valeur source avant de la hacher. La liste est sensible à la casse. Vous pouvez spécifier un espace ou une virgule. Après la première occurrence d'un caractère, les occurrences suivantes dans la liste sont ignorées.

Pour spécifier une barre oblique inversée “\” ou une parenthèse droite “)”, vous devez faire précéder le caractère d'un caractère d'échappement de type barre oblique inversée. Par exemple, pour spécifier une parenthèse droite, entrez : *trim=(\)*.

Vous pouvez uniquement utiliser le caractère d'échappement avec une barre oblique inversée, une parenthèse droite ou dans le cadre de l'indicateur en majuscule.

*\u* Indique que la valeur source doit être convertie en majuscules avant de la hacher. Les caractères à découper sont également convertis en majuscules.

*dest=* Noms des colonnes de table cibles dans lesquelles les valeurs provenant de la table de consultation sont insérées. (Obligatoire pour la consultation dans plusieurs colonnes.)

*col1,coln*

Noms des colonnes de la table cible. L'ordre des noms de colonne doit correspondre à celui des colonnes de la table de consultation indiquées dans le paramètre *values=*.

*nomtableconsult*

Nom de la table de consultation. Vous pouvez spécifier le nom de la table de consultation sous la forme **aliasbdd.IDcréateur.nomtable**, **IDcréateur.nomtable** ou **nomtable**. Si vous n'attribuez pas un nom qualifié complet à la table, les qualificatifs de la table cible sont utilisés.

*recherche*

Nom de la colonne dans la table de consultation qui contient des valeurs séquentielles à mettre en correspondance avec les valeurs hachées provenant de la colonne source.

*valeur* Nom de la colonne de la table de consultation qui contient la valeur de recherche convertie à insérer dans la cible. (Obligatoire pour la consultation dans une seule colonne.)

*values=*

Noms des colonnes de la table de consultation qui contiennent des valeurs à insérer dans la cible. (Obligatoire pour la consultation dans plusieurs colonnes.)

*col1,coln*

Noms des colonnes de la table de consultation. L'ordre des noms de colonne doit correspondre à celui des colonnes de la table cible indiquées dans le paramètre *dest=*.

*cache | nocache*

Indiquez *cache* (valeur par défaut) pour conserver en mémoire une table de valeurs de consultation trouvées ou *nocache* pour supprimer les valeurs trouvées. L'utilisation de *cache* est plus rapide lorsque vous récupérez une valeur plusieurs fois, mais nécessite davantage de mémoire.

*ignore=*

Liste des colonnes source contenant les valeurs insérées dans la cible, plutôt que la valeur de consultation lorsque la colonne contient une ligne avec une valeur citée (NULL, SPACES, ZERO ou VARCHAR de longueur égale à zéro).

*col* Nom de la colonne source.

Pour la consultation dans une seule colonne, entrez un seul nom de colonne.

Pour la consultation dans plusieurs colonnes, l'ordre des noms de colonne doit correspondre à celui des colonnes de la table cible indiquées dans le paramètre *dest=*. Le nombre de colonnes doit être égal au nombre de colonnes indiqué dans le paramètre *dest=* et au moins une colonne doit contenir des valeurs. Pour n'indiquer aucune valeur dans une colonne, n'entrez aucune valeur. Par exemple, *coln()*.

**NULL** Ignore la table de consultation si une ligne de la colonne source contient une valeur NULL.

**SPACES**

Ignore la table de consultation si une ligne de la colonne source contient une valeur SPACES. S'applique uniquement aux colonnes CHAR.

**ZERO\_LEN**

Ignore la table de consultation si une ligne de la colonne source contient une valeur VARCHAR de longueur égale à zéro.

*preserve=*

Liste des colonnes source contenant les valeurs insérées dans la cible, plutôt que la valeur de consultation lorsque la colonne contient une valeur citée (NOT\_FOUND, NULL, SPACES ou VARCHAR de longueur égale à zéro).

**NOT\_FOUND**

Ignore la table de consultation si aucune correspondance n'est trouvée pour la ligne de la colonne source.

**Remarque :**

*preserve=* et *ignore=* sont mutuellement exclusifs. *ignore=* sera considéré comme obsolète dans la prochaine édition.

Les opérandes *col*, *null*, *spaces* et *zero\_len* ont le même effet lorsqu'ils sont utilisés avec *preserve=* ou *ignore=*.

*seed=* Utilisez *seed=* pour varier le calcul de l'algorithme de hachage. Les valeurs comprises entre 1 et 2 000 000 000 peuvent être utilisées. Si vous utilisez une valeur de 0, le paramètre *seed=* est ignoré.

## Exemple avec une seule colonne

Utilisez la fonction de recherche hachée pour insérer des valeurs provenant d'une colonne de table de consultation dans une colonne de table cible, en fonction d'une valeur hachée à partir d'une colonne source.

Prenons par exemple la colonne source *FIRST\_NAME*, qui contient des prénoms et la colonne cible qui va inclure les prénoms de remplacement provenant de la table de consultation. Une table de consultation, *NAME\_LOOKUP*, contient une colonne (*FIRST*) qui inclut les prénoms et une autre colonne (*SEQ*) qui contient les valeurs séquentielles.

Pour obtenir les valeurs de la colonne cible à l'aide de la table *NAME\_LOOKUP*, indiquez :

```
HASH_LOOKUP(FIRST_NAME,NAME_LOOKUP(SEQ, FIRST))
```

La fonction de recherche hachée met en correspondance les valeurs hachées de la colonne source avec les valeurs de la colonne SEQ de la table NAME\_LOOKUP. Lorsqu'une correspondance est trouvée, la fonction insère la valeur correspondante provenant de la colonne FIRST dans la colonne cible.

### Exemple avec plusieurs colonnes

Utilisez la fonction de recherche hachée pour insérer des valeurs provenant de colonnes de table de consultation dans les colonnes d'une table cible, en fonction d'une valeur hachée à partir d'une colonne source.

Prenons l'exemple de valeurs hachées à partir d'une colonne source (FIRST\_NAME) qui contient des prénoms ; vous pouvez remplacer les valeurs dans les colonnes cibles (FIRST et LAST) par des noms et des prénoms provenant d'une table de consultation. Une table de consultation nommée NAME\_LOOKUP contient une colonne (SEQ) incluant des valeurs séquentielles ainsi que des colonnes (FIRST\_MASK et LAST\_MASK) pour masquer les valeurs dans la cible.

Pour remplacer les noms dans la table cible en fonction des valeurs hachées à partir d'une colonne source, indiquez :

```
HASH_LOOKUP(FIRST_NAME,DEST=(FIRST, LAST), NAME_LOOKUP(SEQ,VALUES=(FIRST_MASK, LAST_MASK)))
```

La fonction de recherche hachée met en correspondance les valeurs hachées de la colonne FIRST\_NAME source avec les valeurs de la colonne SEQ de la table NAME\_LOOKUP. Lorsqu'une correspondance est trouvée, la fonction insère les valeurs correspondantes à partir des colonnes FIRST\_MASK et LAST\_MASK de la table de consultation dans les colonnes cibles correspondantes.

### Exemple d'utilisation du paramètre Ignore

Utilisez l'instruction suivante pour développer l'exemple avec une seule colonne, dans lequel vous souhaitez utiliser les valeurs NULL et SPACES source plutôt que les valeurs de la table de consultation :

```
HASH_LOOKUP(FIRST_NAME,NAME_LOOKUP(SEQ, FIRST),IGNORE=(FIRST_NAME(NULL,SPACES)))
```

### Exemple d'utilisation du paramètre NoCache

Utilisez l'instruction suivante pour développer l'exemple avec une seule colonne, dans lequel vous ne souhaitez pas conserver en mémoire une table de valeurs de consultation :

```
HASH_LOOKUP(FIRST_NAME,NAME_LOOKUP(SEQ, FIRST),NOCACHE)
```

### Exemple d'utilisation du paramètre Trim

Utilisez l'instruction suivante pour développer l'exemple avec une seule colonne, dans lequel vous souhaitez tronquer des espaces et des virgules à partir de la valeur source et convertir la valeur source en majuscules avant de la hacher :

```
HASH_LOOKUP(FIRST_NAME, TRIM=( ,\u),NAME_LOOKUP(SEQ,FIRST))
```

### Fonction de recherche aléatoire

La fonction de recherche aléatoire sélectionne une valeur aléatoirement à partir d'une table de consultation spécifiée pour l'insérer dans une colonne cible. La fonction génère un nombre aléatoire entre 1 et la limite ou le nombre de lignes dans la table de consultation utilisée comme sous-script dans la table. La ou les valeurs de la colonne à partir de la ligne qui correspond au sous-script est/sont insérée(s) dans la colonne cible.

Il existe deux formes de fonction de recherche aléatoire : à colonne simple et à colonnes multiples. La fonction de recherche à colonne simple insère une valeur dans une colonne cible unique. La fonction de recherche à colonnes multiples insère des valeurs de plusieurs colonnes de tables de recherche dans les colonnes cible correspondantes.

Vous pouvez entrer la fonction de recherche aléatoire à colonnes multiples pour toute colonne source qui sera remplacée par une valeur de table de recherche. Toutefois, vous devez modifier la mappe de colonne pour supprimer les noms des colonnes source restantes qui seront également remplacés.

Le paramètre *ignore* vous permet d'ignorer la table de recherche et d'utiliser une valeur source lorsqu'une ligne de la colonne source indiquée contient une valeur spécifiée (NULL, SPACES (pour colonnes CHAR) ou valeur VARCHAR de longueur nulle).

Vous pouvez utiliser le paramètre *preserve* pour ignorer la table de recherche et utiliser une valeur source lorsque la ligne d'une colonne source spécifiée contient une valeur précise (NULL, SPACES (pour colonnes CHAR) ou valeur VARCHAR de longueur nulle).

La syntaxe est la suivante :

```
RAND_LOOKUP(lktablename, { columnname | dest=(col1,coln) },values=(col1,coln) }  
[,limit] [,ignore=(col(spaces, null, zero_len), ) | PRESERVE=( colname (spaces, null, zero_len), ) ] )
```

*lktablename*

Nom de la table de recherche. Vous pouvez définir le nom de la table de recherche sous les formes suivantes : *dbalias.creatorid.tablename*, *creatorid.tablename* ou *tablename*. Si le nom de la table n'est pas qualifié complètement, les qualificatifs de la table cible sont utilisés.

*columnname*

Nom de la colonne d'une table de recherche contenant les valeurs à sélectionner de façon aléatoire dans le cadre de l'insertion dans la destination (obligatoire pour les recherches à colonne unique).

*dest*= Noms des colonnes de table cible dans lesquelles les valeurs provenant de la table de recherche sont insérées. (obligatoire pour les recherches à colonnes multiples.)

*col1,coln*

Noms des colonnes de la table cible. L'ordre des noms de colonnes doit correspondre aux colonnes de la table de recherche dans le paramètre *values*=.

*values*=

Noms des colonnes d'une table de recherche contenant les valeurs à insérer dans la cible (obligatoire pour les recherches à colonnes multiples).

*col1,coln*

Noms des colonnes de la table de recherche. L'ordre des noms de colonnes doit correspondre aux colonnes de la table cible dans le paramètre *dest*=.

*limit*

Limite facultative du nombre de lignes de la table de recherche utilisée pour sélectionner les valeurs des colonnes. Indiquez un entier pouvant atteindre une valeur maximale de 2 000 000 000. Si aucune limite n'est indiquée, toutes les lignes sont utilisées.

**Remarque :** Une table des valeurs des colonnes est générée en mémoire. La taille de cette table peut être limitée par les ressources système.

*ignore*=

Liste des colonnes source avec des valeurs insérées à la cible au lieu de la valeur de recherche lorsque la colonne contient une ligne avec une valeur définie (NULL, SPACES ou valeur VARCHAR nulle).

*col*

Nom de la colonne source.

Pour les recherches à colonne unique, entrez uniquement un nom de colonne.

Pour les recherches à colonnes multiples, l'ordre des noms de colonnes doit correspondre aux colonnes de la table cible dans le paramètre *dest=*. Le nombre de colonnes doit correspondre à celui défini dans le paramètre *dest=* et au moins une colonne doit contenir des valeurs. Pour ne pas spécifier de valeurs dans une colonne, n'entrez pas de valeur. Par exemple, *coln()*.

*null* Ignorez la table de recherche si la ligne de la colonne source contient une valeur nulle.

*spaces* Ignorez la table de recherche si la ligne de la colonne source contient une valeur SPACES. Pour colonnes CHAR uniquement.

*zero\_len*

Ignorez la table de recherche si la ligne de la colonne source contient une valeur VARCHAR de longueur nulle.

*preserve=*

Liste des colonnes source avec les valeurs insérées à la cible au lieu de la valeur de recherche lorsque la colonne contient une valeur définie (NOT\_FOUND, null, spaces ou valeur VARCHAR nulle).

*NOT\_FOUND*

Ignorez la table de recherche si aucune correspondance n'est trouvée pour la ligne de la colonne source.

#### **Remarque :**

*preserve=* et *ignore=* s'excluent mutuellement. *ignore=* ne sera plus utilisé dans une prochaine version.

Les opérandes *col*, *null*, *spaces* et *zero\_len* ont le même effet lorsqu'elles sont utilisées avec *preserve=* ou *ignore=*.

### **Exemple de colonne unique**

Pour sélectionner une valeur de façon aléatoire à partir de la colonne STATE dans les 50 premières lignes d'une table nommée STATE\_LOOKUP et l'insérer dans la colonne cible, indiquez ce qui suit :

```
RAND_LOOKUP(STATE_LOOKUP,STATE,50)
```

### **Exemple de colonnes multiples**

Pour sélectionner les valeurs depuis les colonnes CITY, STATE et ZIPCODE dans une ligne aléatoire d'une table nommée STATE\_LOOKUP et les insérer dans les colonnes cible correspondantes, indiquez ce qui suit :

```
RAND_LOOKUP(STATE_LOOKUP,  
DEST=(CITY,STATE,ZIPCODE),  
VALUES=(CITY,STATE,ZIP))
```

### **Exemple de fonction Ignore**

Utilisez l'instruction suivante pour étendre l'exemple de colonne unique, où la colonne source est nommée STATES et utiliser les valeurs NULL et SPACES source au lieu des valeurs de la table de recherche :

```
RAND_LOOKUP(STATE_LOOKUP,STATE,50, IGNORE=(STATES(NULL,SPACES)))
```

## Fonction de mélange

La fonction de mélange remplace une valeur source par une autre valeur à partir de la colonne qui est ensuite insérée dans la colonne cible. La ligne source et la ligne qui contient la valeur de remplacement ne seront jamais les mêmes, toutefois, en fonction de vos données, la valeur source et la valeur de remplacement peuvent être identiques.

Vous pouvez indiquer le nombre de fois où la fonction extrait à nouveau une valeur de remplacement jusqu'à ce qu'une valeur ne correspondant pas à la valeur source soit détectée ("nouvelle tentative"), sinon vous pouvez permettre à une valeur de remplacement de correspondre à la source. Chaque fonction de mélange opère de façon indépendante dans une mappe de colonne.

Il existe deux formes de fonctions de mélange : à colonne simple et à colonnes multiples. La fonction de mélange à colonne simple insère une valeur de remplacement dans une colonne cible unique. La fonction de mélange à colonnes multiples insère des valeurs de remplacement depuis plusieurs colonnes d'une ligne dans des colonnes cible. Une colonne ne peut pas être incluse dans plusieurs fonctions de mélange d'une mappe de colonne. Si la fonction de nouvelle tentative est utilisée avec une fonction de mélange à colonnes multiples, la fonction recherche une autre ligne de remplacement si une valeur de la colonne de la ligne source correspond à la valeur de la colonne de la ligne de remplacement correspondante. La fonction à colonnes multiples ne peut pas être utilisée dans une fonction de propagation.

Pour créer une fonction de mélange à colonnes multiples, entrez la fonction pour une colonne source qui sera remplacée par des valeurs mélangées et modifiez la mappe de colonne pour supprimer les noms des autres colonnes source contenant des valeurs qui seront également remplacées.

Le paramètre *ignore* empêche la fonction de remplacer une ligne source ou d'utiliser une ligne de remplacement si l'une des lignes contient une valeur spécifiée (NULL, SPACES (pour colonnes CHAR), ou valeur VARCHAR nulle). Si aucune nouvelle tentative n'est autorisée, le paramètre *ignore* ne s'applique pas à la ligne de remplacement.

La syntaxe est la suivante :

```
SHUFFLE [ ( dest=(col1,coln) ) ] |
[ ( dest=(col1,coln) , retry[=number] ) ] |
[ ( dest=(col1,coln) [ , retry[=number] ] , ignore=( col1 ( [spaces] | [spaces,null]
| [spaces,null,zero_len] | [null] | [null,zero_len] | [zero_len] ) , coln (...) ) ) ] |
[ ( retry[=number] ) ] |
[ ( retry[=number] , ignore=( col ( [spaces] | [spaces,null] | [spaces,null,zero_len]
| [null] | [null,zero_len] | [zero_len] ) ) ) ] |
[ ( ignore=( col ( [spaces] | [spaces,null] | [spaces,null,zero_len]
| [null] | [null,zero_len] | [zero_len] ) ) ) ]
```

où :

*dest=* Nom des colonnes de table cible dans lesquelles les valeurs de remplacement sont insérées (obligatoire pour la fonction de mélange à colonnes multiples).

*col1, coln, ...*

Noms des colonnes de la table cible.

*retry* Nombre de fois où la valeur de remplacement doit rechercher une valeur ne correspondant pas à la ligne source. Entrez 0 pour permettre à la valeur de remplacement de correspondre à la source.

**Remarque :** L'utilisation d'une valeur de nouvel essai élevée avec des valeurs contenant beaucoup de valeurs dupliquées rallonge la durée du traitement. Pour ces colonnes, il peut être plus judicieux d'utiliser une valeur de nouvel essai égale à zéro.

*=number*

Entrez une valeur comprise entre 0 et 1 000. Entrez 0 pour permettre à la valeur de remplacement de correspondre à la source.

*ignore=*

Liste des colonnes pour lesquelles la fonction ne remplacera pas une valeur source ou n'utilisera pas une valeur de remplacement si l'une d'elles est une valeur spécifiée (NULL, SPACES (pour colonnes CHAR) ou valeur VARCHAR nulle). Si une valeur de remplacement est ignorée, la fonction recherche une autre valeur de remplacement. Si aucune nouvelle tentative n'est autorisée, le paramètre *ignore=* ne s'applique pas aux valeurs de remplacement.

*col* Nom de la colonne source.

Pour la fonction de mélange à colonne unique, entrez uniquement un nom de colonne.

Pour la fonction de mélange à colonnes multiples, l'ordre des noms de colonnes doit correspondre aux colonnes de la table cible dans le paramètre *dest=*. Le nombre de colonnes doit correspondre à celui défini dans le paramètre *dest=* et au moins une colonne doit contenir des valeurs. Pour ne pas spécifier de valeurs dans une colonne, n'entrez pas de valeur. Par exemple, *coln()*.

*null* Ne remplacez pas la valeur source ou utilisez une valeur de remplacement si l'une d'elles possède la valeur NULL.

*spaces* Ne remplacez pas la valeur source ou utilisez une valeur de remplacement si l'une d'elles possède la valeur SPACES. Pour colonnes CHAR uniquement.

*zero\_len*

Ne remplacez pas la valeur source ou utilisez une valeur de remplacement si l'une d'elles possède une valeur VARCHAR nulle.

## Exemple de fonction à colonne unique

L'exemple suivant insère des valeurs mélangées dans une colonne unique.

```
SHUFFLE
```

## Exemple de fonction de nouvel essai à colonne unique

L'exemple suivant insère des valeurs mélangées dans une colonne unique et recherche une valeur de remplacement ne correspondant pas à la source jusqu'à 12 fois.

```
SHUFFLE(RETRY=12)
```

## Exemple de colonnes multiples

L'exemple suivant insère des valeurs mélangées dans les colonnes STATE et ZIP et recherche une valeur de remplacement ne correspondant pas à la source jusqu'à 12 fois.

```
SHUFFLE(DEST=(STATE,ZIP),RETRY=12)
```

## Exemple de fonction Ignore

L'exemple suivant insère des valeurs mélangées dans les colonnes STATE et ZIP et recherche une valeur de remplacement ne correspondant pas à la source jusqu'à 12 fois. L'exemple ne remplace pas une valeur source ou n'utilise pas une valeur de remplacement pour la colonne STATE si une ligne source ou de remplacement contient une valeur NULL ou SPACES, mais n'ignore pas les lignes source ou de remplacement de la colonne ZIP.

```
SHUFFLE(DEST=(STATE,ZIP),RETRY=12,  
IGNORE=(STATE(NULL,SPACES),ZIP()))
```

## Fonction TRANS SSN

Utilisez la fonction TRANS SSN pour générer un numéro de sécurité sociale américain (SSN) valide et unique. Par défaut, la fonction TRANS SSN génère de façon algorithmique un SSN cible modifié de façon constante et basé sur le SSN source. La fonction TRANS SSN peut également générer un SSN aléatoire lorsque les données source ne contiennent pas de valeur SSN ou qu'il n'est pas nécessaire de transformer le SSN source de façon constante.

Un numéro SSN comporte trois zones secondaires. Les trois premiers chiffres (zone) représentent une zone en général déterminée par l'Etat dans lequel est émis le SSN. Les 2 chiffres suivants (groupe) définissent un numéro de groupe correspondant au numéro de zone. Les 4 derniers chiffres (série) sont un numéro de série séquentiel. Quel que soit le type de traitement (par défaut ou aléatoire), la fonction TRANS SSN génère un SSN doté d'un numéro de groupe approprié au numéro de la zone.

La méthode de traitement par défaut génère un SSN contenant le numéro de zone source, ainsi qu'un groupe modifié et des numéros de série basés sur le SSN source.

La méthode de traitement aléatoire génère un SSN pouvant inclure le numéro de zone source et utilise un numéro de groupe émis il y a peu de temps par la Sécurité sociale pour le numéro de la zone cible. Les numéros de série commencent à 0001 et sont incrémentés de 1 pour chaque SSN supplémentaire généré pour le numéro de la zone. Lorsque le numéro de série atteint 9999, la numérotation repart à 0001 et le numéro de groupe précédant le dernier numéro émis pour le numéro de la zone est utilisé.

La syntaxe de la fonction TRANS SSN est la suivante :

**TRANS SSN** [( '*flags*' ] [*sourcecol* [**preserve=invalid**'] ]

*flags* Vous pouvez indiquer un ou plusieurs indicateurs d'option de traitement insensibles à la casse.

*n* Générez un SSN aléatoire non basé sur une valeur source.

*m* Utilisez le groupe maximum de toutes les valeurs de zone SSN, comprenant les valeurs comprises entre 773 et 899 et excluant les numéros de zone non valides.

*r* Générez un SSN doté d'un numéro de zone aléatoire correspondant au même état que le SSN d'entrée.

*v* Validez le numéro de groupe source en le comparant à des numéros utilisés par la Sécurité sociale.

- Le SSN cible doit contenir des tirets séparant les zones (par exemple, 123-45-6789). Nécessite une colonne cible d'au moins 11 caractères.

*sourcecol*

Nom de la colonne source. Si aucun nom de colonne source n'est indiqué, la colonne cible n'est pas utilisé. Si aucun nom de colonne source n'est indiqué et que le nom de la colonne cible ne correspond à un nom de colonne dans la table source, une erreur se produit lors du traitement.

*preserve=invalid*

Si la colonne source contient un SSN non valide, ne le remplacez pas par une valeur générée. La valeur de la colonne source est utilisée dans la colonne cible.

## Types de données autorisés

Les types de données source et cible suivants sont autorisés :

### CHAR

La longueur des données dans la colonne doit être comprise entre 9 et 256 caractères.

### DECIMAL

La précision de la colonne doit être comprise entre 9 et 20 et l'échelle doit être égale à 0.

## INTEGER

Aucune restriction.

## VARCHAR

La longueur des données dans la colonne doit être comprise entre 9 et 254 caractères.

Si une colonne source ou cible ne se conforme pas à ces restrictions, une erreur se produit lors du traitement.

## Règles de traitement cible

Les règles suivantes s'appliquent à la valeur SSN cible, en fonction du type de données ou de la valeur cible :

### CHAR

Si la valeur source est égale à 0, contient des espaces ou une valeur VARCHAR nulle, la valeur cible contient des espaces.

Si une valeur source contient 11 caractères ou plus et des tirets imbriqués (-) ou que l'indicateur '-' est présent, la valeur cible contient des tirets si la colonne cible contient 11 caractères ou plus.

### DECIMAL, INTEGER

Si la valeur source est égale à 0, contient des espaces ou une valeur VARCHAR nulle, la valeur cible est égale à 0.

### VARCHAR

Si la valeur source est égale à 0, contient des espaces ou une valeur VARCHAR nulle, la longueur cible est égale à 0.

Si une valeur source contient 11 caractères ou plus et des tirets imbriqués (-) ou que l'indicateur '-' est présent, la valeur cible contient des tirets si la colonne cible contient 11 caractères ou plus.

**NULL** Si la valeur source est NULL, la valeur cible l'est également.

## Lignes ignorées

Une ligne source peut être ignorée et non écrite dans la cible dans les cas suivants :

- La valeur source est NULL et la colonne cible n'autorise pas ce type de valeur.
- La colonne source est de type CHAR ou VARCHAR et la valeur source contient moins de 9 caractères, un caractère non numérique (autre que les tirets entre les trois zones) ou est trop importante.
- Le numéro de zone source n'a pas été utilisé par la Sécurité sociale.
- Le numéro de groupe source n'a pas été utilisé avec le numéro de zone par la Sécurité sociale (uniquement si l'indicateur 'v' a été indiqué).
- Le numéro de série source est 0000 ou le SSN est une valeur réservée non émise par la Sécurité sociale (par exemple, 078-05-1120).
- La valeur source ne peut pas être convertie au format pris en charge par TRANS SSN.

## Messages d'erreur

Les messages d'erreur suivants peuvent être émis :

### SSN01

*Parm on Col cccc ("ppp") is invalid*

#### Explication

La colonne contient une fonction TRANS dotée d'un indicateur d'option de traitement non valide.

**Action utilisateur**

Assurez-vous que la fonction TRANS de la colonne indiquée utilise un indicateur d'option de traitement valide (n, r, v, -).

**SSN02**

*Col cccc not on source*

**Explication**

La colonne saisie comme paramètre *sourcecol* ou le nom de la colonne cible (si le paramètre *sourcecol* a été omis) est introuvable dans la table source.

**Action utilisateur**

Vérifiez la table source et résolvez les différences ou les colonnes manquantes.

**SSN03**

*Source Col cccc-aaa invalid*

**Explication**

Le format de la colonne source n'est pas pris en charge car l'attribut indiqué est non valide.

**Action utilisateur**

Vérifiez la colonne source et assurez-vous que les valeurs de type, longueur, précision et échelle sont correctes.

**SSN04**

*Dest Col cccc-aaa invalid*

**Explication**

Le format de la colonne cible n'est pas pris en charge car l'attribut indiqué est non valide.

**Action utilisateur**

Vérifiez la colonne cible pour vous assurer que les valeurs de type, longueur, précision et échelle sont correctes.

**SSN05**

*Get col cccc data-rc=nmn*

**Explication**

Une erreur interne inattendue s'est produite lors de la récupération de la valeur depuis la colonne source.

**Action utilisateur**

Vérifiez les valeurs des colonnes source et cible et assurez-vous que les valeurs de type, longueur, précision et échelle sont correctes. Si le problème persiste, contactez IBM Software Support.

**SSN08**

*Put col cccc data-rc=nmn*

**Explication**

Une erreur interne inattendue s'est produite lors de la définition de la valeur dans la colonne cible.

**Action utilisateur**

Vérifiez les valeurs des colonnes source et cible et assurez-vous que les valeurs de type, longueur, précision et échelle sont correctes. Si le problème persiste, contactez IBM Software Support.

Si une autre erreur est émise, contactez le support technique.

## Exemple 1

L'exemple suivant utilise une colonne source correspondant à la colonne cible et générant un SSN aléatoire non basé sur la valeur source :

```
TRANS SSN ('=n')
```

## Exemple 2

L'exemple suivant utilise un nom de colonne source (NATIONAL\_ID) qui diffère de la colonne cible et génère un SSN contenant des tirets via la méthode de traitement par défaut :

```
TRANS SSN ('=- NATIONAL_ID')
```

## Fonction TRANS CCN

Utilisez la fonction TRANS CCN pour générer un numéro de carte de crédit (CCN) valide et unique. Par défaut, la fonction TRANS CCN génère de façon algorithmique un NCC constant modifié basé sur le NCC source. La fonction TRANS CCN peut également générer une valeur aléatoire lorsque les données source ne possèdent pas de valeur NCC ou qu'il n'est pas nécessaire de transformer le NCC source de façon constante.

Un numéro de carte de crédit (CCN), tel que le définit la norme ISO 7812, se compose d'un identificateur de l'émetteur à 6 chiffres, suivi d'un numéro de compte de longueur variable et d'un chiffre de vérification unique qui est le dernier chiffre. Le chiffre de vérification contrôle l'exactitude du CCN. Il est généré par la transmission de l'identificateur de l'émetteur et des numéros de compte via l'algorithme de Luhn. La longueur maximale d'un CCN est de 19 chiffres.

La méthode de traitement par défaut génère un NCC en incluant les quatre premiers chiffres de l'identificateur d'émetteur du NCC source et en modifiant les deux chiffres restants du numéro de l'identificateur d'émetteur et le numéro de compte basé sur le NCC source. Un chiffre clé valide est également attribué.

La méthode de traitement aléatoire génère un NCC pouvant inclure les quatre premiers chiffres du numéro d'identificateur d'émetteur ou un numéro d'identificateur d'émetteur attribué à une carte American Express, Discover, MasterCard ou VISA. Un chiffre clé valide est également attribué. Si les quatre premiers chiffres d'un numéro d'identificateur d'émetteur sont inclus, le premier numéro de compte basé sur ces chiffres commence par le 1. De plus, chaque NCC supplémentaire utilisant ces chiffres incrémente le numéro de compte de 1.

La syntaxe de la fonction TRANS CCN est la suivante :

```
TRANS CCN [( '=flag' [sourcecol] [preserve=invalid]' )]
```

*flag* Indiquez un indicateur d'option pour générer un NCC aléatoire.

*n* Générez un NCC aléatoire non basé sur une valeur source et incluant un numéro d'identificateur d'émetteur attribué à une carte American Express, Discover, MasterCard ou VISA.

*r* Générez un NCC incluant les quatre premiers chiffres du numéro d'identificateur d'émetteur.

*sourcecol*

Nom de la colonne source. Si aucun nom de colonne source n'est indiqué, le nom de colonne cible est utilisé.

Si aucun nom de colonne source n'est indiqué et que le nom de la colonne cible ne correspond à aucun nom de colonne de la table source, une erreur se produit lors du traitement.

*preserve=invalid*

Si la colonne source contient un NCC non valide, ne le remplacez pas par une valeur générée. La valeur de la colonne source est utilisée dans la colonne cible.

## Types de données autorisés

Les types de données source et cible suivants sont autorisés :

### CHAR

La longueur de la colonne doit être comprise entre 13 et 256 caractères.

### VARCHAR

La longueur de la colonne doit être comprise entre 13 et 254 caractères.

### DECIMAL

La précision de la colonne doit être comprise entre 13 et 254 et l'échelle doit être égale à 0.

Si une colonne source ou cible ne se conforme pas à ces restrictions, un message d'erreur est émis.

## Règles de traitement cible

Les règles suivantes s'appliquent à la valeur NCC cible, en fonction du type de données ou de la valeur cible :

### CHAR

Si la valeur source contient des espaces ou une valeur VARCHAR nulle, la valeur cible contient des espaces.

### VARCHAR

Si la valeur source contient des espaces ou une valeur VARCHAR nulle, la longueur cible est définie sur 0.

### DECIMAL

Si la valeur source est égale à 0, la valeur cible l'est également.

NULL Si la valeur source est NULL, la valeur cible l'est également.

## Lignes ignorées

Une ligne source peut être ignorée et non écrite dans la cible dans les cas suivants :

- La valeur source est NULL et la colonne cible n'autorise pas ce type de valeur.
- La valeur source est inférieure à 13 caractères, contient un caractère non numérique, est trop importante ou possède un chiffre clé incorrect.
- La longueur de la valeur source est non valide pour l'émetteur de la carte de crédit.
- La valeur source ne peut pas être convertie dans un format pris en charge par la fonction TRANS CCN.

## Messages d'erreur

Les messages d'erreur suivants peuvent être émis :

### CCN01

*Parm on Col ccccc ("ppp") is invalid*

#### Explication

La colonne indiquée contient une fonction TRANS dotée d'un indicateur d'option non valide.

**Action utilisateur**

Assurez-vous que la fonction TRANS de la colonne indiquée utilise un indicateur d'option de traitement valide (n, r, 6).

**CCN02**

*Col cccc not on source*

**Explication**

La colonne saisie comme paramètre *sourcecol* ou le nom de la colonne cible (si le paramètre *sourcecol* a été omis) est introuvable dans la table source.

**Action utilisateur**

Vérifiez la table source et résolvez les différences ou les colonnes manquantes.

**CCN03**

*Source Col cccc-aaa invalid*

**Explication**

Le format de la colonne source n'est pas pris en charge car l'attribut indiqué est non valide.

**Action utilisateur**

Vérifiez la colonne source et assurez-vous que les valeurs de type, longueur, précision et échelle sont correctes.

**CCN04**

*Dest Col cccc-aaa invalid*

**Explication**

Le format de la colonne cible n'est pas pris en charge car l'attribut indiqué est non valide.

**Action utilisateur**

Vérifiez la colonne cible pour vous assurer que les valeurs de type, longueur, précision et échelle sont correctes.

**CCN05**

*Get col cccc data-rc=nnn*

**Explication**

Une erreur interne inattendue s'est produite lors de la récupération de la valeur depuis la colonne source.

**Action utilisateur**

Vérifiez les valeurs des colonnes source et cible et assurez-vous que les valeurs de type, longueur, précision et échelle sont correctes. Si le problème persiste, contactez IBM Software Support.

**CCN08**

*Put col cccc data-rc=nnn*

**Explication**

Une erreur interne inattendue s'est produite lors de la définition de la valeur dans la colonne cible.

**Action utilisateur**

Vérifiez les valeurs des colonnes source et cible et assurez-vous que les valeurs de type, longueur, précision et échelle sont correctes. Si le problème persiste, contactez IBM Software Support.

Si une autre erreur est émise, contactez le support technique.

## Exemple 1

L'exemple suivant utilise un nom de colonne source (CREDITCARD) qui diffère de la colonne cible et génère un NCC aléatoire basé sur la valeur source :

```
TRANS CCN ('=n CREDITCARD')
```

## Exemple 2

L'exemple suivant utilise un nom de colonne source (CREDITCARD) qui diffère de la colonne cible et génère un NCC via la méthode de traitement par défaut :

```
TRANS CCN ('CREDITCARD')
```

## Fonction TRANS EML

Utilisez la fonction TRANS EML pour générer une adresse de courrier électronique. Une adresse de courrier électronique se compose de deux parties : un nom d'utilisateur suivi par un nom de domaine, séparés par le symbole '@'. Par exemple, utilisateur@domaine.com.

La fonction TRANS EML génère une adresse de courrier électronique avec un nom d'utilisateur basé sur un littéral concaténé avec un numéro séquentiel. Le nom de domaine peut se baser sur une adresse de courrier électronique dans les données source, sur un littéral ou être sélectionné de façon aléatoire à partir d'une liste de grands fournisseurs de services de messagerie électronique. L'adresse de courrier électronique peut également être convertie en majuscules ou en minuscules.

La fonction TRANS EML peut générer un nom d'utilisateur basé sur les valeurs d'une ou de deux colonnes de la table cible (contenant généralement le nom d'un utilisateur). Les options de traitement vous permettent d'utiliser uniquement le premier caractère de la valeur dans la première colonne (par exemple, l'initiale d'un prénom) et de séparer les valeurs des deux colonnes par un point ou un trait de soulignement.

Si le nom d'utilisateur est basé sur une valeur de colonne cible unique ou un littéral, le nom est concaténé avec un numéro séquentiel. Si un nom d'utilisateur est basé sur les valeurs de deux colonnes de la table cible et qu'un point ou un trait de soulignement n'est pas utilisé, les valeurs sont concaténées. Si aucun paramètre n'est fourni pour le nom d'utilisateur, le nom se compose du littéral "email" concaténé avec un numéro séquentiel. Les numéros séquentiels des noms d'utilisateur sont des suffixes commençant à 1 et incrémentés de 1.

La syntaxe de la fonction TRANS EML est la suivante :

```
TRANS EML [( '[=flags] , [{sourcecol | "domain" | , }  
[ {name1col[name2col] | "userpfx"} ] [preserve=invalid]' )]
```

*flags* Vous pouvez indiquer un ou plusieurs indicateurs d'option de traitement insensibles à la casse.

*n* Générez un nom de domaine aléatoire à partir d'une liste d'importants fournisseurs de services de messagerie.

*.* Séparez les valeurs *name1col* et *name2col* par un point.

*\_* Séparez les valeurs *name1col* et *name2col* par un trait de soulignement.

*i* N'utilisez que le premier caractère de la valeur *name1col*.

*l* Convertissez l'adresse de courrier électronique en minuscules.

*u* Convertissez l'adresse de courrier électronique en majuscules.

*sourcecol*

Nom de la colonne source avec adresses de courrier électronique pour fournir le nom de domaine.

Si ni l'indicateur 'n', ni le paramètre *domain* ne sont définis, le nom de domaine situé dans la colonne source est utilisé (si *sourcecol* n'est pas défini, le nom de la colonne source est basé sur le nom de la colonne cible).

Si aucun nom de colonne source n'est indiqué et que le nom de la colonne cible ne correspond pas à un nom de colonne dans la table source, une erreur se produit lors du traitement.

*domain* Littéral pouvant contenir jusqu'à 31 caractères et formant le nom de domaine.

, Une virgule est requise si ni le paramètre *sourcecol*, ni le paramètre *domain* n'est défini et que vous définissez un littéral ou un/des nom(s) de colonne pour le nom de domaine.

*name1col*

Nom de colonne de la table cible contenant des valeurs utilisées pour former la première (ou unique) partie du nom d'utilisateur.

*name2col*

Nom de colonne de la table cible contenant des valeurs utilisées pour former la deuxième partie du nom d'utilisateur.

*userpfx* Littéral pouvant contenir jusqu'à 31 caractères, concaténé avec un numéro séquentiel pour former le nom d'utilisateur.

*preserve=invalid*

Si la colonne source contient une adresse de courrier électronique non valide, ne la remplacez pas par une valeur générée. La valeur de la colonne source est utilisée dans la colonne cible.

## Types de données autorisés

Les types de données source et cible suivants sont autorisés :

### CHAR

La longueur de la colonne doit être comprise entre 3 et 4 096 caractères.

### VARCHAR

La longueur de la colonne doit être comprise entre 3 et 4 094 caractères.

Si une colonne source ou cible ne se conforme pas à ces restrictions, un message d'erreur est émis.

## Règles de traitement cible

Les règles suivantes s'appliquent à la valeur du courrier électronique cible, en fonction du type de données ou de la valeur cible :

### CHAR

Si la valeur source contient des espaces ou une valeur VARCHAR nulle, la valeur cible contient des espaces.

### VARCHAR

Si la valeur source contient des espaces ou une valeur VARCHAR nulle, la longueur cible est définie sur 0.

NULL Si la valeur source est NULL, la valeur cible l'est également.

## Lignes ignorées

Une ligne source peut être ignorée et non écrite dans la cible dans les cas suivants :

- La valeur source est NULL et la colonne cible n'autorise pas ce type de valeur.

- La valeur source est une valeur VARCHAR inférieure à 3 caractères.
- La valeur du courrier électronique source ne contient pas le symbole '@'.
- La valeur source ne peut pas être convertie dans un format pris en charge par la fonction TRANS EML.

## Messages d'erreur

Les messages d'erreur suivants peuvent être émis :

### EML01

*Parm on Col ccccc ("ppp") is invalid*

#### Explication

La colonne indiquée contient une fonction TRANS dotée d'un indicateur d'option non valide.

#### Action utilisateur

Assurez-vous que la fonction TRANS de la colonne indiquée utilise un indicateur d'option de traitement valide (n, ., -, i, l, u).

### EML02

*Col ccccc not on source*

#### Explication

La colonne saisie comme paramètre *sourcecol* ou le nom de la colonne cible (si le paramètre *sourcecol* a été omis) est introuvable dans la table source.

#### Action utilisateur

Vérifiez la table source et résolvez les différences ou les colonnes manquantes.

### EML03

*Source Col ccccc-aaa invalid*

#### Explication

Le format de la colonne source n'est pas pris en charge car l'attribut indiqué est non valide.

#### Action utilisateur

Vérifiez la colonne source et assurez-vous que les valeurs de type, longueur, précision et échelle sont correctes.

### EML04

*Dest Col ccccc-aaa invalid*

#### Explication

Le format de la colonne cible n'est pas pris en charge car l'attribut indiqué est non valide.

#### Action utilisateur

Vérifiez la colonne cible pour vous assurer que les valeurs de type, longueur, précision et échelle sont correctes.

### EML05

*Get col ccccc data-rc=nmn*

#### Explication

Une erreur interne inattendue s'est produite lors de la récupération de la valeur depuis la colonne source.

#### Action utilisateur

Vérifiez les valeurs des colonnes source et cible et assurez-vous que les valeurs de type, longueur, précision et échelle sont correctes. Si le problème persiste, contactez IBM Software Support.

**EML08**

*Put col cccc data-rc=nnn*

**Explication**

Une erreur interne inattendue s'est produite lors de la définition de la valeur dans la colonne cible.

**Action utilisateur**

Vérifiez les valeurs des colonnes source et cible et assurez-vous que les valeurs de type, longueur, précision et échelle sont correctes. Si le problème persiste, contactez IBM Software Support.

**EML09**

*Domain literal ssss too long*

**Explication**

La chaîne indiquée comme littéral de nom de domaine (domaine) dépasse la limite maximale de 31 caractères.

**Action utilisateur**

Indiquez un nom de domaine contenant 31 caractères au maximum.

**EML10**

*User literal ssss too long*

**Explication**

La chaîne spécifiée comme littéral de nom d'utilisateur (*userpfx*) dépasse la limite maximale de 31 caractères.

**Action utilisateur**

Indiquez un nom d'utilisateur contenant 31 caractères au maximum.

**EML11**

*Name1 Col cccc not on dest*

**Explication**

Pour exécuter la fonction TRANS indiquée, une colonne *name1col* doit être indiquée dans la table cible.

**Action utilisateur**

Vérifiez que la colonne *name1col* indiquée correspond à la colonne *name1col* indiquée dans la fonction TRANS.

Le nom de colonne *name1col* est introuvable dans la table cible.

**EML12**

*Name1 Col cccc-aaa invalid*

**Explication**

Pour exécuter la fonction TRANS indiquée, le format de la colonne *name1col* indiquée doit être valide.

**Action utilisateur**

Vérifiez que la disponibilité, le type et la longueur de la colonne *name1col* sont corrects.

**EML13**

*Name2 Col cccc not on dest*

**Explication**

Pour exécuter la fonction TRANS indiquée, une colonne *name2col* doit être indiquée dans la table cible.

**Action utilisateur**

Vérifiez que la colonne *name2col* indiquée correspond à la colonne *name2col* indiquée dans la fonction TRANS.

Le nom de colonne *name2col* est introuvable dans la table cible.

#### EML14

*Name2 Col ccccc-aaa invalid*

#### Explication

Pour exécuter la fonction TRANS indiquée, le format de la colonne *name2col* indiquée doit être valide.

#### Action utilisateur

Vérifiez que la disponibilité, le type et la longueur de la colonne *name2col* sont corrects.

Si une autre erreur est émise, contactez le support technique.

### Exemple 1

L'exemple suivant utilise un littéral (optim.com) pour former le nom de domaine et deux colonnes de la table cible (NAME\_FIRST and NAME\_LAST) pour former un nom d'utilisateur contenant un trait de soulignement :

```
TRANS EML ('=_ "optim.com" NAME_FIRST NAME_LAST')
```

### Exemple 2

L'exemple suivant utilise un nom de domaine dans la colonne source et un littéral (OptimUser) pour former un nom d'utilisateur doté d'un numéro séquentiel en guise de suffixe :

```
TRANS EML (' , "OptimUser" ')
```

## Fonction TRANS COL

La fonction TRANS COL peut masquer les données qui n'ont pas de format inhérent ou qui ont un format peu répandu. TRANS COL conserve le format et le type de caractères de données source dans la cible.

Si les données source contiennent des caractères alphabétiques en majuscules, la fonction TRANS COL génère des caractères alphabétiques en majuscules dans la cible. Cette fonction masque les caractères alphabétiques et numériques, mais les autres caractères des données source sont copiés dans la cible sans être modifiés. Vous pouvez utiliser la fonction TRANS COL pour masquer les valeurs CHAR, VARCHAR et les types de données numériques non flottantes. Vous pouvez générer des valeurs uniques, une valeur différente pour chaque occurrence de la même source et des valeurs de longueurs différentes par rapport à la source.

La syntaxe de la fonction TRANS COL est la suivante :

```
TRANS COL ( '{ unique | hash }[ source=colname ]  
[ copy=( start, len [, "lit" ] )... ]  
[ seed= {"lit" | var (variable) | RANDOM} ]  
[ length=n | max ] [ preserve=( [ null ] [ spaces ] [ zero_len ] ) ]  
[ TRIM=(char1 [ charn,...] [\u] [\r] ) ] [num]' )
```

*unique* Générez une valeur cible unique. La longueur de la valeur cible est identique à celle de la valeur source.

*hash* Générez une valeur cible en hachant la valeur source. Lorsque le hachage est utilisé, différentes valeurs source peuvent produire les mêmes valeurs cible à chaque exécution du processus.

#### Remarque :

Pour la même valeur source, il est possible d'obtenir la même valeur cible lorsque le paramètre

unique ou le paramètre de hachage est utilisé. Utilisez le hachage avec le paramètre de valeur de départ pour générer différentes valeurs cible à chaque exécution du processus.

**source=colname**

Utilisez ce paramètre pour indiquer le nom de la colonne source si la colonne cible diffère du nom de la colonne source. La valeur indiquée est convertie en majuscules. Pour éviter que la valeur soit convertie en majuscules, insérez la valeur entre guillemets.

**copy=** Une ou plusieurs paires de sous-chaînes à copier dans la cible sans être masquées. Si vous fournissez une chaîne littérale, les caractères source des positions indiquées sont remplacés. Le paramètre *copy=* est valide uniquement pour une colonne de type de données de caractères.

**seed=** Valeur utilisée pour modifier le comportement des algorithmes de masquage. Indiquez une chaîne littérale, une référence à une variable d'environnement ou la valeur RANDOM.

*"lit"* Pour indiquer une chaîne littérale, insérez la chaîne entre guillemets.

**var (variable)**

Indiquez une variable d'environnement insérée entre parenthèses. Le nom de la variable et sa valeur ne peuvent pas contenir de guillemets.

**RANDOM**

Générez une valeur de départ aléatoire à partir de la date et de l'heure système en cours.

**length={n | max }**

Générez une valeur cible possédant une longueur différente de la longueur de la valeur source. Utilisez *length=max* pour générer une valeur cible qui remplira totalement la colonne. La spécification d'une longueur plus courte que la valeur source génère une valeur source tronquée lorsqu'elle est écrite dans la cible. La valeur sélectionnée pour *n* ne peut pas dépasser la longueur définie pour la colonne cible. Le paramètre *length=* n'est valide qu'avec le paramètre *hash=*.

**preserve=**

Répertoriez une ou plusieurs valeurs source à ne pas remplacer dans la cible.

**null** Si la colonne source possède une valeur nulle, ne remplacez pas la valeur dans la cible.

**spaces** Si la colonne source possède une valeur d'espaces, ne remplacez pas la valeur dans la cible. Pour colonnes CHAR uniquement.

**zero\_len**

Si la colonne source possède une valeur VARCHAR nulle, ne remplacez pas la valeur dans la cible.

**TRIM=(char1 [ charn,...])**

Le(s) caractère(s) de la colonne source indiquée n'est/ne sont pas masqué(s) et écrit(s) dans la cible. Par exemple, si vous indiquez *TRIM=(x,y,z,1,2,3)* et qu'un caractère x, y, z, 1, 2, 3 apparaît dans la colonne source, ce caractère n'est ni masqué, ni écrit dans la cible.

**[u]** Utilisez ce paramètre pour convertir le(s) caractère(s) en majuscules avant le masquage. Si un caractère ne possède pas de représentation en majuscules, il reste inchangé. Par exemple, *TRIM=(x,y \u)* ne masque pas les caractères x et y s'ils se trouvent dans la colonne source et convertit toutes les autres caractères de la colonne source en majuscules avant de les masquer.

**[r]** Utilisez cette opérande pour supprimer les espaces de fin de chaîne. Par exemple, *TRIM=(x,y \u \r)* ne masque pas les caractères x et y s'ils apparaissent dans la colonne source, convertit les autres caractères de la colonne source en majuscules avant de les masquer, puis supprime les espaces de fin de chaîne avant le masquage.

**num** Utilisez ce paramètre pour obtenir des entiers d'une colonne de type de données de caractères identiques à ceux d'une colonne de type de données numérique. Le paramètre *num* est valide

uniquement pour les valeurs numériques d'une colonne de type de données de caractères. Ainsi, l'intégrité de la clé externe est conservée dans différents types de données. Si vous utilisez ce paramètre, ne définissez pas *copy=* ou *length=*.

Exemples :

- La fonction TRANS COL UNIQUE de la valeur source **CDE-7834** peut générer une valeur cible de : **ZWQ-4598**
- La fonction TRANS COL UNIQUE SEED=RANDOM de la valeur source **CDE-7834-2008** peut générer une valeur cible de : **SWX-3162-8451**
- La fonction TRANS COL UNIQUE de la valeur source **Smith, John** peut générer une valeur cible de : **Fnxwq, Lrzp**
- La fonction TRANS COL UNIQUE de la valeur source **SMITH JOHN** peut générer une valeur cible de : **FNXWQ LRZP**
- La fonction TRANS COL UNIQUE (COPY=((1,3)(10,4))) de la valeur source **CDE-7834-2008** peut générer une valeur cible de : **CDE-4032-2008**
- La fonction TRANS COL HASH (LENGTH=13) de la valeur source **CDE-7834** peut générer une valeur cible de : **ZWQ-4598RN7A0**
- La fonction TRANS COL UNIQUE PRESERVE=(spaces) de la valeur source **XYZ 477 6835** peut générer une valeur cible de : **LMN 623 0972**
- La fonction TRANS COL HASH TRIM=(e \u) de la valeur source **InfoSphere** peut générer une valeur cible de : **RBIWACRL**

## TRANS NID

Utilisez la fonction TRANS NID pour masquer les numéros d'ID nationaux.

Vous pouvez masquer les numéros d'ID nationaux via une méthode reproductible conservant la partie de la valeur source ou une méthode aléatoire ne conservant pas la partie de la valeur source.

Vous pouvez également indiquer le type de séparateur utilisé dans les valeurs de sortie (tirets, points, espaces ou pas de séparateurs).

La fonction TRANS NID utilise la syntaxe suivante :

```
[TRANS] NID ('SWI=country_code, [FMT=(output_format)], [MTD={mask|random}],  
            [SRC=column_name], [VAL={Y|N}], [PRE=INV]')
```

**SWITCH** ou **SWI**

Valeur de deux caractères indiquant le type d'ID national à masquer. Ce paramètre est requis. Une seule valeur de permutation est autorisée.

Les valeurs de deux caractères suivantes sont valides :

**Canada : Numéro d'assuré social canadien (SIN)**

CA

**France : Numéro d'Institut National de la Statistique et des Etudes Economiques (INSEE) français**

FR

**Italie : Numéro de code fiscal (CF) italien**

IT

**Espagne : Numéro d'identification fiscale (NIF) espagnol**

ES

**Royaume-Uni : Numéro d'assurance sociale (NINO) britannique**

UK

## Etats-Unis d'Amérique : Numéro de sécurité sociale (SSN) américain US

### FMT ou FORMAT

Détermine le format de sortie et les parties de la valeur source à masquer. La syntaxe de ce paramètre est déterminée par l'ID national masqué :

- Numéro d'assuré social canadien (SIN)
- Numéro d'Institut National de la Statistiques et des Etudes Economiques français (INSEE)
- Numéro de code fiscal italien (CF)
- Numéro d'identification fiscale (NIF)/Numéro d'identification des étrangers (NIE) espagnol
- Numéro d'assurance sociale britannique (NINO)
- Numéro de sécurité sociale américain (SSN)

Si ce paramètre est omis et que *MTD=mask*, le format source est utilisé et les zones par défaut sont masquées.

Si ce paramètre est omis et que *MTD=random*, les valeurs de sortie n'incluent pas de séparateurs.

Si la colonne cible n'est pas suffisamment développée pour contenir un format de sortie utilisant des séparateurs, les séparateurs ne sont pas inclus.

### MTD ou METHOD

Détermine la méthode de masquage à utiliser (reproductible ou aléatoire). Si ce paramètre est omis, un masquage reproductible est exécuté (*MTD=mask*).

*mask* les valeurs source sont masquées de façon reproductible. Les valeurs de sortie sont basées sur les valeurs source. Par défaut.

*random*

Les valeurs de sortie sont générées par un algorithme de masquage aléatoire. Les valeurs de sortie ne sont pas basées sur les valeurs source.

#### Remarque :

- *MTD=random* n'est pas compatible avec les paramètres suivants : *VAL=Y* et *PRE=INV*.
- Si *MTD=random* et le paramètre *FMT* indiquent qu'une partie de la valeur source doit être copiée dans la valeur de sortie, la valeur de sortie n'inclut pas de valeurs source. Toutefois, les séparateurs indiqués dans le paramètre *FMT* sont inclus dans la valeur de sortie.

### SRC ou SOURCE

Nom de la colonne contenant les valeurs source. Utilisez ce paramètre uniquement si les noms des colonnes source et cible ne correspondent pas.

### VAL ou VALIDATE

Détermine si la validation spécifique à un pays est exécutée sur les valeurs source. Si ce paramètre est omis, aucune validation n'est exécutée (*VAL=N*).

*Y* Validez les valeurs source.

**Remarque :** L'option ne peut pas être utilisée si *MTD=random*.

*N* Ne validez pas les valeurs source. Par défaut.

### PRE ou PRESERVE

Détermine si des valeurs source non valides sont copiées dans la colonne cible. Si ce paramètre est omis, les valeurs source non valides ne sont pas copiées dans la colonne cible et les lignes contenant ces valeurs sont ignorées. Ce paramètre ne contient qu'une option : *INV* ou *INVALID*.

*INV* ou *INVALID*

Copiez les valeurs source non valides dans la colonne cible.

**Remarque :** L'option ne peut pas être utilisée si *MTD=random*.

## Exemples

La syntaxe suivante masque les numéros de sécurité sociale américains (SSN) de façon reproductible. La fonction copie les trois premiers chiffres du SSN source et contient un tiret en guise de séparateur dans la valeur de sortie. La validation est exécutée dans la valeur de sortie.

```
NID('SWITCH=US, FMT=(US=3C-2X-4X), MTD=MASK, VAL=Y')
```

La syntaxe suivante masque les numéros d'Institut National de la Statistique et des Etudes Economiques (INSEE) via une méthode aléatoire. La fonction utilise le format de sortie par défaut, qui ne contient pas de séparateurs.

```
NID('SWITCH=FR, MTD=RANDOM')
```

La syntaxe suivante masque les numéros d'identification fiscale espagnols (NIF) via une méthode reproductible. La fonction préserve les valeurs source non valides et utilise le format de sortie par défaut.

```
NID('SWITCH=ES, MTD=MASK, PRE=INV')
```

## Masquage des numéros d'assurance sociale canadiens

Vous pouvez utiliser la fonction TRANS NID pour masquer les numéros d'assurance sociale canadiens.

Un numéro d'assurance sociale SIN se compose de neuf chiffres : un code de région d'un chiffre et un numéro de série de huit chiffres. Les trois premiers chiffres constituent l'en-tête. Le dernier chiffre du numéro de série constitue le chiffre clé.

La fonction TRANS NID génère un numéro d'assurance sociale masqué possédant un chiffre calculé en fonction des huit chiffres masqués précédents de la valeur de sortie.

### Formats de sortie (FMT=)

Un numéro d'assurance sociale peut posséder les formats de sortie suivants.

C indique les valeurs qui sont copiées. X indique les valeurs qui sont masquées. Par exemple, *3C4X* indique que les trois premiers caractères sont copiés et que les quatre suivants sont masqués.

Zones à masquer	Format sans séparateur	Format avec tiret	Format avec espace	Format avec point
Numéro de série sans chiffres d'en-tête ( <i>MTD=mask</i> par défaut)	CA=3C6X	CA=3C-3X-3X	CA=3C 3X 3X	CA=3C.3X.3X
Chiffres du numéro de série et de l'en-tête	CA=9X	CA=3X-3X-3X	CA=3X 3X 3X	CA=3X.3X.3X

## Types de données

Les types de données suivants sont autorisés pour les colonnes source et cible. Les restrictions relatives à chaque type de données sont fournies.

### CHAR et NCHAR

La longueur de la colonne doit faire au moins 9 caractères.

### DECIMAL

La précision de la colonne doit se situer entre 9 et 20 et l'échelle doit être égale à 0.

## **INTEGER**

Aucune restriction.

## **VARCHAR et NVARCHAR**

La longueur de la colonne doit faire au moins 9 caractères.

### **Validation (VAL=Y)**

Si le paramètre (VAL=Y) est utilisé, une ligne source sera ignorée dans les cas suivants :

- Le premier chiffre est le huit.
- Trois zéros consécutifs sont présents aux positions 1-3, 4-6 ou 7-9.
- Le chiffre clé est non valide.

### **Traitement spécial**

Les vérifications suivantes seront effectuées lors du traitement :

- la valeur source est égale à 0 (type de données INTEGER ou DECIMAL), les espaces ou une valeur VARCHAR ou NVARCHAR nulle et
  - la colonne cible est un type de données INTEGER ou DECIMAL, la valeur de sortie est égale à 0.
  - la colonne cible est un type de données CHAR ou NCHAR, la valeur de sortie sera représentée par des espaces.
  - la colonne cible est un type de données VARCHAR ou NVARCHAR, la longueur cible sera égale à 0.
- Si la longueur de la colonne cible fait 11 caractères ou plus et que des séparateurs ont été spécifiés pour la valeur de sortie, les séparateurs sont inclus si la colonne cible est un type de données CHAR, NCHAR, VARCHAR ou NVARCHAR.
- Si la valeur source est NULL, la valeur de sortie l'est également.

### **Lignes ignorées**

Une ligne source est ignorée et n'est pas écrite dans la table cible dans les cas suivants :

#### **Valeur nulle**

La valeur source est NULL mais la colonne cible n'autorise pas ce type de valeur.

#### **Numéro d'assurance sociale non valide**

- La colonne source est de type CHAR, NCHAR, VARCHAR ou NVARCHAR mais la valeur source contient moins de 9 caractères (séparateurs non inclus).
- La colonne source est de type CHAR, NCHAR, VARCHAR ou NVARCHAR mais la valeur contient plus de 9 caractères (séparateurs non inclus).
- La valeur source contient un caractère non numérique.

### **Masquage des numéros INSEE**

Vous pouvez utiliser la fonction TRANS NID pour masquer les numéros INSEE.

Un numéro INSEE est un nombre à 15 chiffres possédant le format suivant : SYMMDDCCCCOOCK.

**S** Sexe et civilité.

**YY** Deux derniers chiffres de l'année de naissance.

**MM** Mois de naissance.

**DD** Département d'origine.

**CCC** Commune d'origine.

**OOO** Numéro d'ordre.

**KK** Clé de contrôle ou chiffre clé.

La fonction TRANS NID génère un numéro INSEE masqué selon les règles suivantes :

- Si la zone Département est masquée, la zone Commune est également masquée avec une valeur compatible.
- La zone d'ordre est toujours masquée.
- La zone Chiffre clé est calculée en fonction des 13 chiffres de la valeur de sortie précédemment masqués.

### Formats de sortie (FMT=)

Un numéro INSEE peut posséder les formats de sortie suivants.

Tous les formats masquent les zones Ordre et Chiffre clé. Si la zone Département est masquée, la zone Commune est également masquée avec une valeur compatible.

C indique les valeurs qui sont copiées. X indique les valeurs qui sont masquées. Par exemple, 3C4X indique que les trois premiers caractères sont copiés et que les quatre suivants sont masqués.

Zones à masquer (outre Ordre et Chiffre clé)	Format sans séparateur	Format avec tiret	Format avec espace
Sexe, année, mois, commune (MTD=mask par défaut)	FR=5X2C8X	FR=5X2C6X-2X	FR=5X2C6X 2X
Sexe	FR=1X9C5X	FR=1X9C3X-2X	FR=1X9C3X 2X
Sexe, année	FR=3X7C5X	FR=3X7C3X-2X	FR=3X7C3X 2X
Sexe, mois	FR=1X2C2X5C5X	FR=1X2C2X5C3X-2X	FR=1X2C2X5C3X 2X
Sexe, commune	FR=1X6C8X	FR=1X6C6X-2X	FR=1X6C6X 2X
Sexe, département	FR=1X4C8X	FR=1X4C6X-2X	FR=1X4C6X 2X
Sexe, année, mois	FR=5X5C5X	FR=5X5C3X-2X	FR=5X5C3X 2X
Sexe, année, commune	FR=3X4C8X	FR=3X4C6X-2X	FR=3X4C6X 2X
Sexe, année, Département, Commune	FR=3X2C10X	FR=3X2C8X-2X	FR=3X2C8X 2X
Sexe, mois, commune	FR=1X2C2X2C8X	FR=1X2C2X2C6X-2X	FR=1X2C2X2C6X 2X
Sexe, mois, département, commune	FR=1X2C12X	FR=1X2C10X-2X	FR=1X2C10X 2X
Sexe, année, mois, département, commune	FR=15X	FR=13X-2X 13X	FR=13X 2X
Année	FR=1C2X7C5X	FR=1C2X7C3X-2X	FR=1C2X7C3X 2X
Année, mois	FR=1C4X5C5X	FR=1C4X5C3X-2X	FR=1C4X5C3X 2X
Année, commune	FR=1C2X4C8X	FR=1C2X4C6X-2X	FR=1C2X4C6X 2X
Année, département	FR=1C2X2C10X	FR=1C2X2C8X-2X	FR=1C2X2C8X 2X
Année, mois, commune	FR=1C4X2C8X	FR=1C4X2C6X-2X	FR=1C4X2C6X 2X
Année, mois, département	FR=1C14X	FR=1C12X-2X	FR=1C12X 2X
Mois	FR=3C2X5C5X	FR=3C2X5C3X-2X	FR=3C2X5C3X 2X
Mois, commune	FR=3C2X2C8X	FR=3C2X2C6X-2X	FR=3C2X2C6X 2X
Mois, département	FR=3C12X	FR=3C10X-2X	FR=3C10X 2X
Commune	FR=7C8X	FR=7C6X-2X	FR=7C6X 2X

Zones à masquer (autre Ordre et Chiffre clé)	Format sans séparateur	Format avec tiret	Format avec espace
Département	FR=5C10X	FR=5C8X-2X	FR=5C8X 2X

## Types de données

Les types de données suivants sont autorisés pour les colonnes source et cible. Les restrictions relatives à chaque type de données sont fournies.

### CHAR et NCHAR

La longueur de la colonne doit faire au moins 15 caractères.

### VARCHAR et NVARCHAR

La longueur de la colonne doit faire au moins 15 caractères.

### Validation (VAL=Y)

Si le paramètre (VAL=Y) est utilisé, une ligne source sera ignorée dans les cas suivants :

- La valeur de la zone Commune source est non valide.
- La zone Chiffre clé source est non valide.

### Traitement spécial

Les vérifications suivantes seront effectuées lors du traitement :

- Si la valeur source contient des espaces ou une valeur VARCHAR ou NVARCHAR nulle et
  - la colonne cible est un type de données CHAR ou NCHAR, la valeur de sortie sera représentée par des espaces.
  - la colonne cible est un type de données VARCHAR ou NVARCHAR, la longueur de destination sera égale à 0.
- Si la valeur source est NULL, la valeur de sortie l'est également.
- Si la longueur d'une colonne cible fait 16 caractères ou plus et que des séparateurs ont été spécifiés pour la valeur de sortie, les séparateurs sont inclus.

### Lignes ignorées

Une ligne source est ignorée et n'est pas écrite dans la table cible si la valeur source contient les erreurs suivantes :

#### Valeur nulle

La valeur source est NULL mais la colonne cible n'autorise pas ce type de valeur.

#### Valeur INSEE non valide

- La taille de la valeur INSEE est trop importante ou trop faible pour être valide.
- La valeur INSEE de saisie contient des séparateurs non valides ou incorrectement positionnés.
- La zone Sexe ne contient pas l'une des valeurs suivantes : 1, 2, 7 ou 8.

### Masquage de numéro de code fiscal italien

Vous pouvez utiliser la fonction TRANS NID pour masquer les codes fiscaux italiens (CF).

Un CF est une valeur alphanumérique de 16 caractères au format suivant : *FFF-NNN-YYMDD-RRRRC*.

**FFF** Prénom codé.

**NNN** Nom donné codé.

**YY** Année de naissance.

- M** Mois de naissance.
- DD** Jour de naissance.
- RRRR** Code région.
- C** Caractère de contrôle.

La fonction TRANS NID génère un CF masqué selon les règles suivantes :

- Toute consonne apparaissant dans les zones Nom ou Prénom fournies est masquée comme consonne. Il en est de même pour les voyelles. Si un X apparaît après une voyelle, il doit être copié dans la valeur de sortie.
- La zone Caractère de contrôle est calculée en fonction des 15 chiffres masqués précédents de la valeur de sortie.

### Formats de sortie (FMT=)

Un CF peut posséder les formats de sortie suivants.

C indique les valeurs qui sont copiées. X indique les valeurs qui sont masquées. Par exemple, 3C4X indique que les trois premiers caractères sont copiés et que les quatre suivants sont masqués.

Zones à masquer	Format sans séparateur	Format avec tiret	Format avec espace
Date de naissance, Région (MTD=mask par défaut)	IT=6C10X	IT=3C-3C-5X-5X	IT=3C 3C 5X 5X
Prénom, nom, région	IT=6X5C5X	IT=3X-3X-5C-5X	IT=3X 3X 5C 5X
Prénom, nom donné, date de naissance	IT=11X4C1X	IT=3X-3X-5X-4C1X	IT=3X 3X 5X 4C1X
Prénom, nom donné	IT=6X9C1X	IT=3X-3X-5C-4C1X	IT=3X 3X 5C 4C1X
Date de naissance	IT=6C5X4C1X	IT=3C-3C-5X-4C1X	IT=3C 3C 5X 4C1X
Région	IT=11C5X	IT=3C-3C-5C-5X	IT=3C 3C 5C 5X
Prénom, nom donné, date de naissance, région	IT=16X	IT=3X-3X-5X-5X	IT=3X 3X 5X 5X

### Types de données

Les types de données suivants sont autorisés pour les colonnes source et cible. Les restrictions relatives à chaque type de données sont fournies.

#### CHAR et NCHAR

La colonne doit contenir au moins 16 caractères.

#### VARCHAR et NVARCHAR

La colonne doit contenir au moins 16 caractères.

### Validation (VAL=Y)

Si le paramètre VAL=Y est utilisé, une ligne source est ignorée en cas de présence d'un caractère de contrôle non valide.

### Traitement spécial

Les vérifications suivantes seront effectuées lors du traitement :

- Si la valeur source contient des espaces ou une valeur VARCHAR ou NVARCHAR nulle et

- la colonne cible est un type de données CHAR ou NCHAR, la valeur de sortie sera représentée par des espaces.
- la colonne cible est un type de données VARCHAR ou NVARCHAR, la longueur de destination sera égale à 0.
- Si la valeur source est NULL, la valeur de sortie l'est également.
- Si une colonne cible fait 19 caractères ou plus et que des séparateurs ont été spécifiés pour la valeur de sortie, les séparateurs seront inclus.

## Lignes ignorées

Une ligne source est ignorée et n'est pas écrite dans la table cible dans les cas suivants :

### Valeur nulle

La valeur source est NULL mais la colonne cible n'autorise pas ce type de valeur.

### CF non valide

La colonne source possède une valeur source non valide ou la valeur source est trop importante.

La valeur source est inférieure à 16 caractères.

## Masquage du numéro d'identification fiscale espagnol et du numéro d'identité des étrangers

Vous pouvez utiliser la fonction TRANS NID pour masquer les numéros d'identification fiscale (NIF) et les numéros d'identité des étrangers (NIE).

Le NIF est une valeur composée de huit caractères au format suivant : *NNNNNNN-A*. Les sept premiers caractères sont le numéro de série et le caractère final est un suffixe alphabétique. Le suffixe est un chiffre clé.

Les espagnols d'origine étrangère utilisent le numéro d'identité des étrangers (NIE). Il s'agit d'une valeur de neuf caractères utilisant le même format que le numéro d'identification fiscale, excepté qu'il est précédé d'un X. Un NIE utilise le format suivant : *X-NNNNNNN-A*.

La fonction TRANS NID génère un NIF ou un NIE masqué, doté d'un chiffre clé calculé en fonction des 7 chiffres masqués précédents de la valeur de sortie.

### Formats de sortie (FMT=)

Un NIF ou un NIE peut posséder les formats de sortie suivants.

Pour chaque format, tous les caractères sont masqués. Les NIF et NIE utilisent les mêmes options de format. Une valeur source NIE inclut toujours un préfixe X dans la valeur de sortie.

Zones à masquer	Format sans séparateur	Format avec tiret	Format avec espace
Numéro de série, suffixe ( <i>MTD=mask</i> par défaut)	ES=8X	ES=7X-1X	ES=7X 1X

## Types de données

Les types de données suivants sont autorisés pour les colonnes source et cible. Les restrictions relatives à chaque type de données sont fournies.

### CHAR et NCHAR

La longueur de la colonne doit faire au moins 8 caractères.

### VARCHAR et NVARCHAR

La longueur de la colonne doit faire au moins 8 caractères.

## **Validation (VAL=Y)**

Si le paramètre *VAL=Y* est utilisé, une ligne source est ignorée en cas de présence d'un suffixe non valide.

## **Traitement spécial**

Les vérifications suivantes seront effectuées lors du traitement :

- Si la valeur source contient des espaces ou une valeur VARCHAR ou NVARCHAR nulle et
  - la colonne cible est un type de données CHAR ou NCHAR, la valeur de sortie sera représentée par des espaces.
  - la colonne cible est un type de données VARCHAR ou NVARCHAR, la longueur de destination sera égale à 0.
- Si la valeur source est NULL, la valeur de sortie l'est également.
- Si la longueur d'une colonne cible fait 11 caractères ou plus et que des séparateurs ont été spécifiés pour la valeur de sortie, les séparateurs sont inclus.

## **Lignes ignorées**

Une ligne source est ignorée et n'est pas écrite dans la table cible dans les cas suivants :

### **Longueur non valide**

La valeur en entrée ne contient pas de séparateurs et est inférieure à 8 caractères (NIF) ou 9 caractères (NIE).

La valeur en entrée contient des séparateurs et est inférieure à 9 caractères (NIF) ou 11 caractères (NIE).

### **Modèle non valide**

La longueur de la valeur source est valide, mais la valeur ne correspond pas au modèle d'un NIF ou d'un NIE.

### **Séparateurs**

Différents séparateurs ont été trouvés aux positions 2 et 10 d'une valeur source NIE.

La valeur source contient un séparateur non valide.

### **Valeur nulle**

La valeur source est NULL mais la colonne cible n'autorise pas ce type de valeur.

## **Masquage des numéros d'assurance sociale britanniques**

Vous pouvez utiliser la fonction TRANS NID pour masquer les numéros d'assurance sociale britanniques (NINO).

Un NINO se compose de trois parties : deux lettres (préfixe), six chiffres (nombre) et une lettre facultative (suffixe).

## **Formats de sortie (FMT=)**

Un NINO peut posséder les formats de sortie suivants.

Un NINO peut être masqué avec ou sans séparateur dans un format en trois ou cinq parties.

C indique les valeurs qui sont copiées. X indique les valeurs qui sont masquées. Par exemple, 3C4X indique que les trois premiers caractères sont copiés et que les quatre suivants sont masqués.

Pour créer un NINO sans séparateur, utilisez les paramètres suivants :

Zones à masquer	Format sans séparateur
Préfixe, numéro	UK=8X1C
Numéro ( <i>MTD=mask</i> par défaut)	UK=2C6X1C
Préfixe, numéro, suffixe ( <i>MTD=random</i> uniquement)	UK=9X

Pour créer un NINO avec un format à trois ou cinq parties, utilisez les paramètres suivants :

Zones à masquer	Format avec tiret	Format avec espace	Format avec point
Préfixe, numéro (trois parties)	UK=2X-6X-1C	UK=2X 6X 1C	UK=2X.6X.1C
Préfixe, numéro (cinq parties)	UK=2X-2X-2X-2X-1C	UK=2X 2X 2X 2X 1C	UK=2X.2X.2X.2X.1C
Numéro (trois parties)	UK=2C-6X-1C	UK=2C 6X 1C	UK=2C.6X.1C
Numéro (cinq parties)	UK=2C-2X-2X-2X-1C	UK=2C 2X 2X 2X 1C	UK=2C.2X.2X.2X.1C
Préfixe, numéro, suffixe (trois parties) ( <i>MTD=random</i> uniquement)	UK=2X-6X-1X	UK=2X 6X 1X	UK=2X.6X.1X
Préfixe, numéro, suffixe (cinq parties) ( <i>MTD=random</i> uniquement)	UK=2X-2X-2X-2X-1X	UK=2X 2X 2X 2X 1X	UK=2X.2X.2X.2X.1X

## Types de données

Les types de données suivants sont autorisés pour les colonnes source et cible. Les restrictions relatives à chaque type de données sont fournies.

### CHAR et NCHAR

La longueur de la colonne doit faire au moins 9 caractères.

### VARCHAR et NVARCHAR

La longueur de la colonne doit faire au moins 9 caractères.

### Validation (*VAL=Y*)

Le paramètre *VAL=Y* est non valide pour le masquage des NINO et est ignoré.

### Traitement spécial

Les vérifications suivantes seront effectuées lors du traitement :

- Si la valeur source contient des espaces ou une valeur VARCHAR ou NVARCHAR nulle et
  - la colonne cible est un type de données CHAR ou NCHAR, la valeur de sortie sera représentée par des espaces.
  - la colonne cible est un type de données VARCHAR ou NVARCHAR, la longueur cible sera égale à 0.
- Si la valeur source est NULL, la valeur de sortie l'est également.
- Si la colonne cible ne peut contenir de séparateur, la valeur de sortie n'en inclut pas.

## Lignes ignorées

Une ligne source est ignorée et n'est pas écrite dans la table cible dans les cas suivants :

### Valeur nulle

La valeur source est NULL mais la colonne cible n'autorise pas ce type de valeur.

### NINO non valide

- La taille de la valeur source est inférieure ou supérieure à la taille d'un NINO valide.
- La valeur source contient des séparateurs incorrectement positionnés.
- La valeur source contient un séparateur non valide.
- La valeur source contient un préfixe non valide.
- La valeur source contient un suffixe autre que A, B, C ou D.
- La valeur source contient une zone numérique non comprise entre 000001 et 999999.

## Masquage des numéros de sécurité sociale américains

Vous pouvez utiliser la fonction TRANS NID pour masquer les numéros de sécurité sociale américains (SSN).

Un SSN contient trois sous-zones au format suivant : AAAGGSSSS.

**AAA** Numéro de zone. La zone est généralement déterminée par l'état dans lequel le SSN est émis.

**GG** Numéro de groupe. Un numéro de groupe est attribué en fonction du numéro de la zone.

**SSSS** Numéro de série.

La fonction TRANS NID génère un SSN masqué en fonction des règles suivantes :

- Un numéro de groupe approprié pour le numéro de zone est généré. Le numéro de groupe est le dernier utilisé par la Sécurité sociale pour la zone.
- Les numéros de série commencent à 0001 et sont incrémentés de 1 pour chaque SSN supplémentaire généré pour le numéro de la zone. Lorsque le numéro de série atteint 9999, la numérotation repart à 0001 et le numéro de groupe précédant le dernier numéro émis pour le numéro de la zone est utilisé.
- Lorsque  $MTD=mask$ , la valeur de sortie inclut un numéro de zone correspondant au même état que le numéro de la zone source.

## Formats de sortie (FMT=)

Un SSN peut posséder les formats de sortie suivants.

C indique les valeurs qui sont copiées. X indique les valeurs qui sont masquées. Par exemple, 3C4X indique que les trois premiers caractères sont copiés et que les quatre suivants sont masqués.

Zones à masquer	Format sans séparateur	Format avec tiret	Format avec espace	Format avec point
Groupe, numéro de série (MTD=mask par défaut)	US=3C6X	US=3C-2X-4X	US=3C 2X 4X	US=3C.2X.4X
Zone, groupe, numéro de série	US=9X	US=3X-2X-4X	US=3X 2X 4X	US=3X.2X.4X

## Types de données

Les types de données suivants sont autorisés pour les colonnes source et cible. Les restrictions relatives à chaque type de données sont fournies.

## CHAR et NCHAR

La longueur de la colonne doit faire au moins 9 caractères.

## DECIMAL

La précision de la colonne doit se situer entre 9 et 20 et l'échelle doit être égale à 0.

## INTEGER

Aucune restriction.

## VARCHAR et NVARCHAR

La longueur de la colonne doit faire au moins 9 caractères.

## Validation (VAL=Y)

Si le paramètre (VAL=Y) est utilisé, une ligne source sera ignorée dans les cas suivants :

- Le numéro de la zone source dépasse la valeur maximale.
- Le numéro de zone source n'a pas été utilisé par la Sécurité sociale.
- Le numéro de groupe source n'a pas été utilisé pour le numéro de zone source.

## Traitement spécial

Les vérifications suivantes seront effectuées lors du traitement :

- la valeur source est égale à 0 (type de données INTEGER ou DECIMAL), les espaces ou une valeur VARCHAR ou NVARCHAR nulle et
  - la colonne cible est un type de données INTEGER ou DECIMAL, la valeur de sortie est égale à 0.
  - la colonne cible est un type de données CHAR ou NCHAR, la valeur de sortie sera représentée par des espaces.
  - la colonne cible est un type de données VARCHAR ou NVARCHAR, la longueur de destination sera égale à 0.
- Si la longueur de la colonne cible fait 11 caractères ou plus et que des séparateurs ont été indiqués pour la valeur de sortie, les séparateurs sont inclus.
- Si la valeur source est NULL, la valeur de sortie l'est également.

## Lignes ignorées

Une ligne source est ignorée et n'est pas écrite dans la table cible dans les cas suivants :

### Valeur nulle

La valeur source est NULL mais la colonne cible n'autorise pas ce type de valeur.

### SSN non valide

- La colonne source est de type CHAR, NCHAR, VARCHAR ou NVARCHAR, mais la valeur source contient plus ou moins que 9 caractères (séparateurs non inclus).
- La valeur source contient un caractère non numérique.
- La valeur source est égale à 0 ou est une valeur réservée, telle que 078-05-1120 et 457-55-5462.

## Fonction de sous-chaîne

La fonction de sous-chaîne renvoie une sous-chaîne du contenu de la colonne nommée.

SUBSTR(*columnname*, *start*, [*length*])

*columnname*

Nom d'une colonne de caractères ou binaire.

*start* Position du premier caractère dans la chaîne.

*length* Nombre de caractères à utiliser.

- Si l'environnement local utilise une virgule comme séparateur décimal, vous devez laisser un espace après chaque virgule séparant les paramètres numériques (par exemple, après la virgule située entre *start* et *length*).
- Les valeurs *start* et *length* sont supérieures ou égales à 1.
- Les valeurs *start* et *length* ne peuvent pas dépasser la longueur de données totale + 1.
- Les valeurs *column-name* et *start* sont obligatoires. Si vous n'indiquez qu'un entier, celui-ci est utilisé comme valeur *start*. La sous-chaîne commence à la valeur *start* et inclut le reste de la valeur de la colonne.

## Exemple

Si la colonne PHONE\_NUMBER possède la valeur CHAR(10), vous pouvez utiliser la fonction de sous-chaîne pour mapper le code de la zone. Pour obtenir une sous-chaîne des trois premières positions du numéro de téléphone (code de zone) de la colonne cible, indiquez ce qui suit :

```
SUBSTR(PHONE_NUMBER, 1, 3)
```

## Fonction aléatoire

La fonction aléatoire renvoie un nombre sélectionné de façon aléatoire dans la plage indiquée par les valeurs inférieure et supérieure.

```
RAND(low, high)
```

*low* La plus petite valeur aléatoire possible.

*high* La plus grande valeur aléatoire possible.

- Utilisez la fonction aléatoire avec des données de caractères ou numériques.
- Si l'environnement local utilise une virgule comme séparateur décimal, vous devez laisser un espace après la virgule.
- *low* et *high* sont des entiers compris entre **-2,147,483,648** et **2,147,483,647**.
- Les valeurs *low* et *high* sont de plus limitées par le type et la longueur des données de la colonne cible.
- La valeur *low* doit être inférieure à la valeur *high*.
- Lorsque vous utilisez la fonction aléatoire dans une expression concaténée, une chaîne de caractères de longueur variable est renvoyée.

## Exemple

Vous pouvez utiliser la fonction aléatoire pour masquer ou modifier les données commerciales d'une base de données de test. Supposons que la colonne YTD\_SALES possède la valeur DECIMAL(7,2). Le nombre maximal de chiffres précédant la décimale est 5 ; la plage possible pour cette colonne est donc comprise entre -99999 et 99999. Pour créer des données de test dans une plage comprise entre 1000 (*low*) et 89999 (*high*), indiquez ce qui suit :

```
RAND(1000, 89999)
```

Dans cet exemple, la fonction renvoie des valeurs commerciales aléatoires dans la plage spécifiée comprise entre 1000.00 et 89999.99.

## Fonction séquentielle

La fonction séquentielle renvoie un nombre qui est incrémenté de manière séquentielle. La syntaxe est la suivante :

`SEQ(start, step)`

*start* Valeur de début.

*step* Valeur incrémentielle.

- Utilisez la fonction séquentielle avec des données de caractère et numériques.
- Si l'environnement local utilise une virgule comme séparateur décimal, vous devez laisser un espace après la virgule.
- *start* et *step* sont des entiers compris entre **-2,147,483,648** et **2,147,483,647**.
- Les valeurs *start* et *step* sont de plus limitées par le type et la longueur des données de la colonne cible.
- Si la valeur calculée dépasse la longueur de la colonne cible, la valeur *start* est automatiquement attribuée à la fonction.
- Lorsque vous utilisez la fonction séquentielle dans une expression concaténée, une chaîne de caractères de longueur variable est renvoyée.

## Exemple 1

Vous pouvez utiliser la fonction séquentielle pour modifier les données client d'une base de données de test. Supposons que la colonne `CUST_ID` possède la valeur `CHAR(5)`. Pour l'incrémenter de 50 à partir de 1, indiquez ce qui suit :

```
SEQ(1, 50)
```

Dans cet exemple, la fonction renvoie les valeurs `CUST_ID` à partir de '00001' et l'incrémente de 50 pour générer '00051', '00101', etc. Lorsque le résultat dépasse '99951', une valeur *start* de 1 est attribuée à la fonction.

## Exemple 2

Vous pouvez utiliser la fonction séquentielle dans une mappe de colonne pour masquer les données commerciales d'une base de données de test. Supposons que la colonne `YTD_SALES` possède la valeur `DECIMAL(7,2)`. Pour l'incrémenter de 100 à partir de 1 000, indiquez ce qui suit :

```
SEQ(1000, 100)
```

Dans cet exemple, la fonction renvoie les valeurs `YTD_SALES` à partir de 1 000 et l'incrémente de 100 pour générer 1 100, 1 200, etc. Lorsque le résultat dépasse 99 999, une valeur *start* de 1 000 est attribuée à la valeur.

## Exemple 3

Supposons que la colonne `SALESMAN_ID` possède la valeur `CHAR(6)`. Pour insérer des valeurs commençant par 'NJ,' suivies d'un nombre à partir de 50 et incrémenté de 10, utilisez la fonction dans une expression concaténée :

```
'NJ' || SEQ(50, 10)
```

Dans cet exemple, la fonction renvoie des valeurs `SALESMAN_ID` commençant par 'NJ50 ' et l'incrémente de 10 pour générer 'NJ60 ', 'NJ70 ', etc. Lorsque le résultat dépasse 'NJ9990', une valeur *start* est attribuée à la fonction.

## Fonction d'identité ou de série

Les fonctions d'identité et de série indiquent au SGBD de fournir une valeur séquentielle (entier) à une colonne cible. La syntaxe de ces fonctions est la suivante :

IDENTITY( )

SERIAL ( )

- Utilisez la fonction d'identité pour les colonnes d'identité contenues dans les bases de données DB2, Sybase ASE et SQL Server.
- Utilisez la fonction de série pour les colonnes de série contenues dans les bases de données Informix.
- Les deux fonctions sont valides pour traitement d'insertion (mettre à jour/insérer) et de chargement, mais ne sont pas valides pour le traitement de conversion.
- Si les lignes sont mises à jour dans un processus d'insertion (mettre à jour/insérer), la colonne cible, ciblée par la fonction d'identité ou de série conserve la valeur initiale. De plus, si la colonne cible fait partie de la clé primaire, la valeur de la colonne reste la même lorsque la ligne est mise à jour.
- Vous pouvez utiliser la fonction d'identité ou de série avec la fonction Propager pour le traitement d'insertion ; vous ne pouvez toutefois pas propager les colonnes d'identité ou de série dans un processus de chargement.

## Fonction de séquence Oracle

La fonction de séquence Oracle affecte une valeur à la colonne cible via une séquence Oracle.

*schema.seqname*.NEXTVAL [INCL\_UPD]

*schema* Qualificateur du nom de la séquence Oracle.

*seqname*

Nom de la séquence oracle qui attribue les valeurs séquentielles.

*NEXTVAL*

Mot-clé insérant la valeur Oracle suivante dans la colonne cible.

*INCL\_UPD*

Mot-clé facultatif mettant à jour une valeur de séquence attribuée à une colonne lorsque des lignes sont mises à jour dans le cadre d'un processus d'insertion. S'il n'est pas indiqué (par défaut), la valeur de la colonne n'est pas modifiée lorsque la ligne est mise à jour.

- Vous pouvez utiliser la fonction de séquence Oracle pour attribuer des valeurs séquentielles uniques aux lignes à insérer dans une base de données Oracle.
- La fonction de séquence Oracle est valide lorsqu'elle est utilisée dans une mappe de colonne dans le cadre d'un processus d'insertion ou de chargement, mais n'est pas valide dans le cadre d'un traitement de conversion.
- Si des lignes sont mises à jour dans un processus d'insertion (insertion/mise à jour) et que la colonne cible fait partie de la clé primaire, la valeur de la colonne n'est pas modifiée lorsque la ligne de la table cible est mise à jour. Pour utiliser la séquence Oracle lors de l'exécution d'une mise à jour/insertion, insérez la variable *INCL\_UPD* dans la fonction.
- Lors d'un processus de chargement, le processus utilise la fonction séquentielle Oracle pour attribuer une nouvelle valeur à chaque ligne cible. Le chargement appelle le SGBD pour obtenir ces valeurs. Si vous choisissez de ne pas exécuter le chargeur Oracle, ces valeurs de séquence ne sont jamais utilisées.

### Exemple 1

Pour attribuer une valeur séquentielle et incrémenter les numéros client, où le nom de la séquence Oracle est **schema.numeven**, indiquez ce qui suit :

*schema.numeven*.NEXTVAL

## Exemple 2

Pour développer le premier exemple et mettre à jour une valeur de séquence existante, indiquez ce qui suit :

```
schema.numeven.NEXTVAL(INCL_UPD)
```

## Fonction Propager la valeur de clé principale ou externe

La fonction de clé affecte une valeur à la clé principale ou la clé externe de la colonne et propage cette valeur à toutes les tables associées.

La syntaxe est la suivante :

```
PROP( { value [, columnname] ] EXIT exitname |  
PROC { LOCAL | identifieur.name } } )
```

*value* Valeur à attribuer à la colonne. Indiquez une valeur source de mappe de colonne valide (par exemple, un nom de colonne, un littéral chaîne, une expression ou une fonction). La valeur doit être appropriée pour la colonne.

*columnname*

Nom de la colonne source qui contient la valeur objet de la fonction. La valeur obtenue est insérée dans la colonne cible de la table mappée et dans la colonne cible appropriée des tables associées participantes.

Le nom de la colonne est obligatoire *uniquement* si aucune colonne source ne correspond à la colonne cible des types de nom et de données. S'il n'est pas spécifié, le nom de la colonne cible est utilisé.

*exitname*

Nom exit d'une mappe de colonne.

*identifieur.name*

Nom de procédure de mappe de colonne.

- Si l'environnement local utilise une virgule comme séparateur décimal, vous devez laisser un espace après chaque virgule séparant les paramètres numériques.
- La fonction Propager est valide dans une mappe de colonne pour les traitements d'insertion (mais pas de mise à jour ou mise à jour/insertion), de chargement ou de conversion.
- Lorsque vous utilisez la fonction Propager, vous devez inclure au moins une table associée dans le processus. Vous pouvez utiliser la fonction Propager plusieurs fois pour le même processus.
- Vous pouvez utiliser la fonction Propager pour une colonne de clé primaire ou la colonne de clé externe qui lui correspond, mais pas les deux.
- Si plusieurs colonnes définissent une relation, vous pouvez utiliser la fonction Propager pour une ou plusieurs de ces colonnes. Toutefois, dans une relation étendue Optim, vous pouvez indiquer la fonction Propager uniquement dans des relations entre colonnes.
- Vous pouvez utiliser la fonction Identifier ou la fonction Série dans la fonction Propager dans le cadre d'un traitement d'insertion. Toutefois, vous ne pouvez pas propager la fonction Identité dans un processus de chargement.
- Les paramètres indiqués dans la fonction Propager ne sont pas validés jusqu'à l'exécution. En cas de conflits, le processus n'est pas exécuté.
- L'insertion peut contenir des cycles de propagation. Toutefois, les traitements de chargement et de conversion peuvent ne pas aboutir à des cycles de propagation. Les cycles sont détectés lors de la validation du processus au moment de l'exécution. Si une demande de chargement ou de conversion génère un cycle de propagation, le processus n'est pas exécuté.
- Optim mémorise les valeurs source et celles attribuées aux colonnes cible correspondantes. Par conséquent, vous pouvez effectuer la propagation vers les colonnes cible dans lesquelles la source est

une expression. Lorsque l'expression évaluée correspond à une valeur source, l'attribut Optim affecte la valeur cible correspondante. Lorsque l'expression évaluée ne correspond pas aux valeurs source, une erreur de conversion se produit.

Avant l'exécution d'une insertion, d'un chargement ou d'une conversion, vous pouvez analyser la mappe de colonne pour vérifier comment la fonction Propager est utilisée dans le processus.

### Exemple 1

Vous pouvez générer un numéro aléatoire, l'attribuer à la colonne cible par défaut et propager le numéro dans les colonnes cible des tables associées. Pour générer une valeur entre 10 000 et 99 999, l'insérer dans la colonne cible mappée et la propager dans les colonnes cible des tables associées, indiquez ce qui suit :

```
PROP(RAND(10000, 99999))
```

### Exemple 2

Vous pouvez exécuter la même fonction que dans l'exemple 1 lorsque les noms des colonnes source et cible ne correspondent pas. Pour inclure le nom de la colonne source (CUST\_NUMBER) dans la fonction Propager, indiquez ce qui suit :

```
PROP(RAND(10000, 99999), CUST_NUMBER)
```

### Exemple 3

Vous pouvez utiliser une séquence Oracle pour générer la valeur de la colonne cible et propager cette valeur dans les colonnes cible des tables associées. Pour propager la séquence Oracle **schema.numeven**, indiquez ce qui suit :

```
PROP(schema.numeven.NEXTVAL)
```

## Expressions concaténées

La concaténation vous permet de combiner des valeurs de colonne ou de combiner une valeur de colonne avec une autre valeur, à l'aide d'un opérateur de concaténation (CONCAT, ||, or +). Une expression concaténée peut inclure des valeurs de caractères ou des valeurs binaires, mais pas les deux :

#### Valeurs des caractères

Les valeurs des caractères concaténés peuvent être des colonnes de caractères, des littéraux chaîne, des sous-chaînes de valeurs dans des colonnes de caractères, la fonction séquentielle ou la fonction aléatoire.

#### Valeurs binaires

Les valeurs des caractères concaténés peuvent être des colonnes de caractères, des littéraux hexadécimaux, des sous-chaînes de colonnes binaires, la fonction séquentielle ou la fonction aléatoire.

- Une expression concaténée ne peut pas inclure de littéral chaîne nul (' '), de registre spécial ou la fonction Age.

### Exemple

Supposons que la table CUSTOMERS possède une adresse dans deux colonnes : ADDRESS1 et ADDRESS2. La table SHIP\_TO stocke une adresse dans une colonne : ADDRESS. Vous pouvez utiliser une expression concaténée pour combiner les informations d'adresse de deux colonnes d'une table dans la colonne d'une autre table.

Pour combiner l'adresse, spécifiez l'un des exemples suivants :

#### Table CUSTOMERS

ADDRESS1 || ADDRESS2  
ADDRESS1 CONCAT ADDRESS2  
ADDRESS1 + ADDRESS2

#### Table SHIP\_TO

ADDRESS  
ADDRESS  
ADDRESS

## Expressions numériques

Utilisez une expression numérique pour indiquer une valeur dans la colonne source lorsque les types de données des colonnes source et cible correspondantes sont compatibles.

Une expression numérique possède la syntaxe suivante:

operand1 operator operand2

Chaque opérande doit être une colonne ou une constante numérique. L'opérateur spécifie s'il faut ajouter ( + ), soustraire ( - ), diviser ( / ) ou multiplier ( \* ).

### Exemple 1

Pour augmenter de 10 % la valeur DECIMAL(5,2) d'une colonne UNIT\_PRICE, indiquez l'exemple suivant :

1.1 \* UNIT\_PRICE

### Exemple 2

Pour diviser de moitié la valeur d'une colonne INTEGER ON\_HAND\_INVENTORY, indiquez ce qui suit :

ON\_HAND\_INVENTORY / 2.

## Fonctions littérales et de valeur

Utilisez les fonctions littérales et de valeur pour spécifier des littéraux tels qu'une chaîne ou une valeur hexadécimale et des valeurs telles qu'une valeur NULL ou un registre spécial.

Fonction	Description
Nom de la colonne	Nom de colonne explicite (les noms de colonne ne sont pas sensibles à la casse).
NULL	NULL. La colonne cible doit accepter les valeurs NULL.
Constante numérique	Une constante numérique. Une valeur constante doit correspondre à la colonne cible telle que définie par son type de données, sa précision et son échelle.
Constante booléenne	Une constante booléenne (VRAI ou FAUX).

Fonction	Description
Registre spécial	<p>Un registre spécial :</p> <ul style="list-style-type: none"> <li>• CURRENT DATE</li> <li>• CURRENT_DATE</li> <li>• CURRENT TIME</li> <li>• CURRENT_TIME</li> <li>• CURRENT TIMESTAMP</li> <li>• CURRENT_TIMESTAMP</li> <li>• CURRENT SQLID</li> <li>• CURRENT_SQLID</li> <li>• CURDATE( )</li> <li>• CURTIME( )</li> <li>• GETDATE( )</li> <li>• GETTIME( )</li> <li>• SYSDATE( )</li> <li>• NOW( )</li> <li>• WORKSTATION_ID</li> <li>• USER</li> </ul>
Littéral chaîne	<p>Un littéral chaîne, entre guillemets simples. La colonne cible doit contenir des données de type caractères.</p> <p>Par exemple : "CA" ou "90210".</p>
Littéral hexadécimal	<p>Un littéral hexadécimal.</p> <p>Par exemple : X'1234567890ABCDEF' ou 0X1234567890ABCDEF</p>
Littéral de date/heure	<p>Un littéral de date/heure entre guillemets simples. Séparez la date et l'heure par un espace. Pour formater la date/l'heure avec une fraction décimale, placez une virgule après l'heure, suivi par la fraction. Le format de la date est déterminé par les paramètres des <b>options régionales</b> du panneau de configuration de votre ordinateur.</p> <p><b>Remarque :</b> Pour l'horodateur Oracle avec les colonnes de fuseau horaire, vous devez spécifier le suffixe du fuseau horaire en dernier.</p>

## Fonction Age

La fonction Age permet de dater les valeurs d'une colonne source. La colonne source peut contenir des données de type caractère, numériques, de date ou d'horodatage. Une colonne CHAR ou VARCHAR possède une longueur maximale de 256 octets.

La fonction Age est formatée comme suit :

AGE(*paramètres*)

- Définissez la fonction Age de sorte à inclure un ou plusieurs paramètres d'âge.
- Utilisez des virgules ou des espaces pour séparer les paramètres dans la fonction Age.
- Les paramètres peuvent être spécifiés dans n'importe quel ordre.

Paramètre	Format	Valeurs admises
Nom de colonne : indiquez le nom de la colonne source s'il est différent de celui de la colonne cible.	SC= <i>nom-colonne</i> SRCCOL= <i>nom-colonne</i>	Nom de colonne

Paramètre	Format	Valeurs admises
Par défaut : calculer l'ancienneté en fonction de la valeur d'ajustement de date indiquée dans une demande de processus.	DEF	Utilise la valeur d'ajustement de date indiquée dans la demande de processus.
Aucune : la valeur n'est pas datée.	NONE	la valeur ne doit pas être datée malgré les spécifications de la demande de processus.
Incrémentiel – Le calcul incrémentiel de l'âge se base sur une unité de temps connue. Optim prend en charge le calcul de l'âge pour une date dans les unités uniques (20 ans, par exemple) ou dans les unités multiples (2 ans, 3 mois, 2 jours, par exemple).	+ ou - ] nY [ + ou - ] nM [ + ou - ] nW [ + ou - ] nD (Le signe plus [+] est facultatif.)	nY -2500 à +1581 nM -30000 à +30000 nW -30000 à +30000 nD -99999 à +99999
Année spécifique : calculer l'ancienneté en fonction d'une année à quatre chiffres spécifique entrée au format souhaité	nnnnY	1582 - 3999
Multiple/règle : calculer l'ancienneté en fonction du nombre d'applications d'une règle métier. Si vous définissez la fonction Age à l'aide du paramètre Multiple/règle, vous devez également inclure le paramètre RULE.	nnnnnR	1 - 30000

## Ancienneté sémantique

L'ancienneté sémantique se base sur un ensemble de règles que vous définissez pour gérer les dates qui tombent en période de vacances, les week-ends, etc. Vous pouvez utiliser l'ancienneté sémantique pour ajuster les dates de sorte qu'elles tombent pendant des jours ouvrables valides.

### Calendrier –

Nom du calendrier définissant les dates spéciales auxquelles s'appliquent les règles. Si vous utilisez le paramètre CALENDAR, vous devez également indiquer une règle.

CA=*nom-calendrier*

CALENDAR=*nom-calendrier*

### Règle –

Nom de la règle définissant l'ajustement pour les dates spéciales. Si DEF est indiqué, la règle par défaut spécifiée dans la demande de processus est utilisée.

RU=*nom-règle*

RULE=*nom-règle*

RU=DEF

RULE=DEF

### Siècle pivot –

Détermine le siècle sous forme d'année à deux chiffres. Entrez une valeur comprise entre 00 et 99.

PI=*nn*

PIVOT=*nn*

- Si vous indiquez **AGE(RU=DEF)**, la règle spécifiée dans une demande de processus est utilisée. Vous devez indiquer des valeurs pour les autres paramètres de la fonction Age.
- Si vous utilisez une règle, mais n'indiquez aucun calendrier, la fonction Age utilise le calendrier par défaut que vous spécifiez dans une demande de processus.

- Si vous n'incluez aucun paramètre CALENDAR, RULE et PIVOT aux emplacements requis dans la fonction Age, les valeurs par défaut que vous spécifiez dans une demande de processus s'appliquent.
- Pour spécifier le siècle approprié pour une année à deux chiffres, vous devez inclure le paramètre PIVOT dans la fonction Age.
- Si vous indiquez une valeur PIVOT, toutes les années à deux chiffres supérieures ou égales à la valeur PIVOT sont placées dans le 20e siècle (19xx). Toutes les années à deux chiffres inférieures à la valeur PIVOT sont placées dans le 21e siècle (20xx). La valeur PIVOT par défaut est 65.

## Formats de date

Le format de date source et le format de date cible doivent contenir un format de date unique valide et doivent être inférieurs ou égaux à la longueur de la colonne cible. La chaîne de format doit être délimitée par des guillemets simples.

### Format de date source –

Applique la chaîne de format de la colonne source aux colonnes d'âge numérique et de type caractère.

SF='chaîne-format'

SRCFMT='chaîne-format'

Si la colonne source est de type caractère ou numérique, vous devez utiliser SRCFMT ou une routine d'exit source (SRCEXIT) pour décrire le contenu de la colonne. Ces paramètres sont mutuellement exclusifs.

### Format de date cible –

Applique la chaîne de format de la colonne cible aux colonnes d'âge numérique et de type caractère.

DF='chaîne-format'

DSTFMT='chaîne-format'

Si la colonne cible est de type caractère ou numérique, vous pouvez spécifier DSTFMT ou une routine d'exit cible (DSTEXIT). Si vous n'indiquez pas de format pour la cible, la fonction de calcul de l'âge pour une date utilise SRCFMT par défaut. La colonne cible d'une fonction AGE ne peut pas être binaire.

Utilisez les chaînes de caractères suivantes pour spécifier les composants du format de date :

Année	Mois	Jour	Heure	Composants/seconde
YYYY	MONTH	DDD	HH	FFFFFF
CCYY	MMM	DD	MI	FFFFFF
YY	MM	D	SS	FFFF
	M			FFF
				FF
				F

- Si vous indiquez un point d'interrogation (?) dans une chaîne de format, la fonction Age mappe la valeur de caractère telle quelle. (Utilisez le point d'interrogation pour inclure des barres obliques, des tirets, des points, etc, dans le format de date.)
- Si vous indiquez un astérisque (\*) dans une chaîne de format, la fonction Age mappe les caractères restants dans la colonne source à la colonne cible. (Utilisez l'astérisque lorsque la valeur de colonne est une date concaténée à des caractères supplémentaires.)

**Remarque :** Vous pouvez utiliser l'utilitaire Calendrier pour définir un séparateur par défaut et une année de sortie par défaut. Ces valeurs par défaut s'appliquent lorsque les formats source et cible nécessitent des séparateurs ou une année spécifique.

## Exemple 1

Pour donner un âge à une colonne de date de 2 ans, 6 mois, 40 semaines et 15 jours, puis appliquer une règle, formatez la fonction Age comme suit :

```
AGE(+2Y,+6M,+40W,+15D,RU=NEXTPAYDAY)
```

## Exemple 2

Pour donner un âge uniquement à la partie 'année' d'une colonne de date (année 2020), puis appliquer une règle, formatez la fonction Age comme suit :

```
AGE(2020Y,RU=NEXTWORKDAY)
```

## Exemple 3

Pour donner un âge à une colonne de date à l'aide du paramètre Multiple/règle, de sorte à incrémenter de 5 occurrences une règle appelée NEXTSTRTQTR, en utilisant un calendrier appelé PSAPRULE, formatez la fonction Age comme suit :

```
AGE(CA=PSAPRULE,RU=NEXTSTRTQTR,5R)
```

## Exemple 4

Pour donner un âge aux données d'une colonne de type caractère ou numérique, utilisez les paramètres suivants :

- Colonne source nommée.
- Format source, qui utilise les deux premiers caractères pour les deux derniers chiffres de l'année et les 3 chiffres restants pour le jour dans le calendrier julien.
- Siècle pivot afin de déterminer le siècle correct, étant donné que la source est formatée avec une année à deux chiffres. Le siècle pivot de cet exemple est 42. Toutes les années à deux chiffres supérieures ou égales à 42 sont placées dans le 20<sup>e</sup> siècle (19xx). Toutes les années à deux chiffres inférieures à 42 sont placées dans le 21<sup>e</sup> siècle (20xx).
- Donner un âge à une date tous les 5 ans.

Formatez la fonction Age comme suit :

```
AGE(5Y,SC=ORDER_DATE,SF='YYDDD',PI=42)
```

## Fonction Devise

Utilisez la fonction Devise pour convertir une valeur de devise contenue dans une colonne source en autre devise. La colonne source doit être définie au format numérique, pas en virgule flottante. Deux méthodes de conversion sont disponibles :

### Conversion directe

Cette méthode fournit des paramètres de conversion en fonction des valeurs définies dans une définition de devise. Utilisez la fonction Devise pour convertir une valeur monétaire contenue dans une colonne (en remplaçant la valeur précédente) ou, en définissant différentes colonnes source et cible, conservez la valeur initiale et la valeur convertie. Vous pouvez définir de manière explicite les types de devise source et cible ou identifier une colonne de référence afin d'indiquer le type de devise.

La première préférence en termes de calcul consiste à utiliser le taux de conversion pour passer de la devise source à la devise cible. La seconde préférence en termes de calcul consiste à utiliser le taux de conversion pour passer de la devise cible vers la devise source.

## Triangulation

Convertissez la valeur contenue dans une colonne de la devise source en euro-dollar, puis convertissez la valeur euro-dollar en devise cible. Les deux taux doivent être fournis dans la table des taux : un taux entre l'euro-dollar et la devise source, et un taux entre l'euro-dollar et la devise cible. L'expression de spécification est TRIANG ou TR.

La fonction Devise est formatée comme suit :

```
CURRENCY( {ST=code | SS=(nom-colonne,numéro de la table des types)}  
{DT=code | DS=(nom-colonne,numéro de la table des types)}  
[SC=nom-colonne] [TR] [CU=nom de la définition de devise]  
[TD=nom-colonne-date-transaction] [DF='format']  
[NS=échelle] )
```

La fonction Devise doit inclure au moins une combinaison du type de devise source (ST) ou de la spécification source (SS) avec le type de devise cible (DT) ou la spécification cible (DS). Tous les autres paramètres sont facultatifs.

Les types de devise source et cible peuvent être spécifiés de l'une des manières suivantes :

1. Utilisez les mots-clés ST/DT pour permettre la spécification explicite de la devise en utilisant le code de devise ISO 4217 à trois caractères.
2. Utilisez les mots-clés SS/DS pour permettre la spécification indirecte du type de devise, où une valeur contenue dans une colonne nommée de la ligne est utilisée comme clé. La clé est mise en correspondance avec un type de devise, comme indiqué dans la table des types de définition de devise spécifiée.

Si vous indiquez une date de transaction (TD) et que la colonne contenant la date de transaction n'utilise pas le format DATE, vous devez également indiquer un format de date (DF). Si une date de transaction spécifiée n'est pas comprise dans la plage de dates spécifiée dans la table des taux de définition de devise, la plage de dates la plus proche est utilisée pour les calculs de conversion.

**Remarque :** Si certaines données obligatoires (les types de devise ou les taux par exemple) sont manquantes, des erreurs de conversion sont générées lors de la phase d'exécution.

Le tableau ci-dessous décrit le format valide et les valeurs autorisées pour les paramètres de la fonction Devise. Les paramètres peuvent être spécifiés dans n'importe quel ordre. Utilisez des virgules ou des espaces pour séparer les paramètres dans la fonction Devise.

Paramètre	Format
Colonne source	SC=nom-colonne SRCCOL=nom-colonne
Type de devise source	ST=code SRCTYP=code où code = code de devise ISO 4217
Type de devise cible	DT=code DSTYP=code où code = code de devise ISO 4217
Spécification source	SS=nom de colonne, numéro de la table des types SRCSPC=nom de colonne, numéro de la table des types où nom de colonne, numéro de la table des types = colonne Table des types et numéro (défini dans la définition de devise) à utiliser pour spécifier le type de devise source

<b>Paramètre</b>	<b>Format</b>
<b>Spécification cible</b>	DS= <i>nom de colonne, numéro de la table des types</i> DSTSPC= <i>nom de colonne, numéro de la table des types</i> où <i>nom de colonne, numéro de la table des types = colonne Table des types et numéro (défini dans la définition de devise) à utiliser pour spécifier le type de devise cible</i>
<b>Triangulation (applique une conversion via l'euro-dollar)</b>	TR TRIANG
<b>Définition de devise</b>	CU= <i>nom de la définition de devise</i> CURTBL= <i>nom de la définition de devise</i> où <i>nom de la définition de devise = définition de devise qui contient les paramètres de conversion appropriés.</i>
<b>Date de transaction</b>	TD= <i>nom de colonne</i> TRNDAT= <i>nom de colonne</i> où <i>nom de colonne = nom de la colonne Date de transaction permettant d'identifier la date de conversion</i>
<b>Format de date</b>	DF= <i>'format'</i> DATEFMT= <i>'format' où format = format de la colonne Date de transaction ; type de date, le cas échéant.</i>
<b>Echelle numérique</b>	NS= <i>échelle</i> NUMSCL= <i>échelle</i> où <i>échelle = échelle à appliquer aux colonnes cibles numériques Oracle dont l'échelle n'est pas définie.</i>

## Exemple 1

Pour convertir des marks finlandais en euros-dollars, formatez la fonction Devise comme suit :

```
CURRENCY(ST=FIM DT=EUR)
```

Si la valeur initiale doit être conservée, utilisez la fonction Devise pour entrer une valeur pour une autre colonne dans la table cible.

## Exemple 2

Pour convertir des marks finlandais en euros-dollars et créer une colonne qui contiendra la valeur source initiale (en marks finlandais) nommée ITEM\_COST, formatez la fonction Devise comme suit :

```
CURRENCY(ST=FIM DT=EUR SC=ITEM_COST)
```

## Nom d'e-mail généré automatiquement

La fonction de nom d'e-mail généré automatiquement génère une adresse électronique avec un nom d'utilisateur basé sur un littéral concaténé avec un numéro séquentiel. Les numéros séquentiels sont des suffixes commençant par 1 et qui sont incrémentés de 1. La fonction utilise le nom de domaine à partir d'une adresse électronique dans une colonne source spécifiée.

## Nom d'e-mail formaté

La fonction de nom d'e-mail formaté génère une adresse électronique avec un nom d'utilisateur basé sur les valeurs obtenues auprès d'un ou deux attributs. Cette règle utilise le nom de domaine issu d'une adresse e-mail figurant dans une colonne source spécifiée.

## Fonction de numérotation aléatoire

La fonction de numérotation aléatoire génère des nombres sélectionnés de manière aléatoire dans la plage indiquée par les valeurs hautes et basses.

Vous pouvez utiliser la fonction de numérotation aléatoire pour remplacer un caractère ou des données numériques. Les valeurs hautes et basses doivent être des entiers compris dans la plage allant de -2 147 483 648 à 2 147 483 647. La valeur basse doit être inférieure à la valeur haute.

## Fonction de numérotation séquentielle

La fonction de numérotation séquentielle génère des chiffres incrémentés de manière séquentielle.

Vous pouvez utiliser la fonction de numérotation séquentielle pour remplacer un caractère ou des données numériques. Vous devez entrer une valeur de début et une valeur d'incrément des chiffres. Les valeurs de début et incrémentielles doivent être des entiers compris dans la plage allant de -2 147 483 648 à 2 147 483 647.

La valeur générée est limitée par le type de données et la longueur de la colonne cible. Si la valeur générée dépasse la longueur de la colonne cible, la fonction est automatiquement réinitialisée à la valeur de début.

---

## Utilisation des routines d'exit

Lorsque vous créez un service pour convertir, créer, insérer, charger, ou restaurer des données, vous pouvez spécifier une mappe de table qui contient une ou plusieurs mappes de colonne, afin de dériver les valeurs appropriées pour les colonnes cibles.

Il existe différentes manières de spécifier une valeur de colonne dans une mappe de colonne. L'une de ces méthodes consiste à spécifier une routine d'exit en tant que source ; l'exit définit les valeurs qui ne peuvent pas être définies d'une autre manière pour les colonnes cibles. Une autre méthode consiste à utiliser les routines d'exit pour exclure des lignes du traitement. Vous pouvez utiliser trois types de routine d'exit :

### Exit standard

La routine d'exit standard est appelée pour dériver la valeur d'une colonne cible dans une mappe de colonne. Ce type de routine d'exit s'avère utile lorsque vous souhaitez effectuer des transformations de données qui vont au-delà des mappes de colonne. Par exemple, un exit peut modifier le numéro de département d'un employé pour les lignes sélectionnées, conformément à un algorithme complexe, ou sélectionner des lignes spécifiques à traiter et ignorer toutes les autres lignes.

Un exit standard peut obtenir un segment de sous-chaîne d'une colonne LOB source. Pour insérer une nouvelle valeur LOB dans une colonne LOB cible, un exit peut créer un fichier et retransmettre le nom du fichier au processeur de mappe de colonne.

### Exit de format source

L'exit de format source est appelé pour formater la colonne source dans une fonction Age qui n'aurait autrement pas été prise en charge dans une mappe de colonne. Cette routine d'exit examine la date source dans une colonne de type entier ou caractère et la convertit dans un format de date pouvant être utilisé comme entrée dans la fonction Age.

### Exit de format cible

L'exit de format cible est appelé pour formater la colonne cible dans une fonction Age qui n'aurait autrement pas été prise en charge dans une mappe de colonne. Cette routine d'exit convertit une date

dans l'un des quatre différents formats cibles. Le type de données de la colonne cible détermine le format.

## Exit dans une mappe de colonne

Pour utiliser une routine d'exit dans une mappe de colonne, vous devez indiquer l'un des paramètres suivants dans la colonne source appropriée :

### Exit standard

```
EXIT nomdll[(param[,]param)... ]
```

### Exit de format source

```
AGE(SRCEXIT=nomdll)
```

### Exit de format cible

```
AGE(DSTEXIT=nomdll)
```

Le processus appelle la routine d'exit de la mappe de colonne, une fois pour chaque ligne de données traitée, puis passe un appel de fin une fois la dernière ligne traitée. Les paramètres facultatifs spécifiés avec une routine d'exit standard sont transmis à l'exit ; il doit s'agir de littéraux de chaîne (entre guillemets simples) ou numériques (limite : 8).

## Ecriture de routines d'exit

Vous pouvez écrire des routines d'exit dans n'importe quel langage de programmation ; toutefois, les appels passés à des sous-routines doivent être conformes à des conventions utilisées dans le langage de programmation C.

## Fichiers d'en-tête

Pour définir les paramètres, les structures et les codes retour utilisés dans la routine d'exit, une routine d'exit doit inclure deux fichiers d'en-tête de programme C, PSTEXIT.H et, en fonction du format alphanumérique des métadonnées (noms de table, noms de colonne, etc.), PSTCMXIT.H ou PSTCMWXT.H:

### PSTEXIT.H

Indique les types de données, les codes retour et les structures des types de données définis par Optim.

### PSTCMXIT.H

Pour les métadonnées au format mono-octet (par ex., ASCII).

Fournit les prototypes pour la routine de rappel de la mappe de colonne et indique les *définitions* et les *définitions de structure* des paramètres d'exit de la mappe de colonne.

### PSTCMWXT.H

Pour les métadonnées au format UTF-16 (WCHAR).

Fournit les prototypes pour la routine de rappel de la mappe de colonne et indique les *définitions* et les *définitions de structure* des paramètres d'exit de la mappe de colonne.

Ces fichiers d'en-tête se trouvent dans le même répertoire que les fichiers d'application d'IBM InfoSphere Optim.

**Remarque :** Si un paramètre est aux formats mono-octet et UTF-16, le présent chapitre présente le format UTF-16 entre parenthèses.

## Utilisation des bibliothèques de liaison dynamique (DLL)

Vous devez compiler et lier chaque routine d'exit en tant que DLL distincte. Optim charge chaque DLL de manière dynamique lors de la phase d'exécution. Une DLL peut contenir une seule occurrence d'un type d'exit de mappe de colonne particulier. Toutefois, vous pouvez inclure une routine d'exit de mappe de colonne de chaque type dans la même DLL.

Vous devez utiliser le nom de la DLL dans la mappe de colonne. Indiquez le nom de la DLL tel qu'il est défini sur la plateforme (n'utilisez pas son nom générique). Ecrivez la DLL à nommer et exportez les fonctions réelles qui implémentent la routine d'exit :

### Exit standard

```
PSTColMapExit  
(PSTColMapWExit)
```

### Exit de format source

```
PSTColMapAgeSrcExit  
(PSTColMapAgeSrcWExit)
```

### Exit de format cible

```
PSTColMapAgeDstExit  
(PSTColMapAgeDstWExit)
```

## Exigences

Chaque routine d'exit de mappe de colonne doit satisfaire les exigences suivantes :

- L'exit doit générer une valeur appropriée à la colonne cible et ne doit modifier aucune autre zone de stockage.
- Optim utilise la clé primaire pour générer les zones de travail internes. Si la routine d'exit affecte une valeur à une colonne qui fait partie de la clé primaire, et que cette routine est appelée plusieurs fois pour la même ligne source, vérifiez que la routine d'exit génère la même valeur de sortie à chaque fois.
- Un exit peut être appelé fréquemment ; évitez les pertes de temps système inutiles. Par exemple, l'exit devrait initialiser le traitement au cours du premier appel, puis sauvegarder les informations pour les appels suivants dans la zone de travail.

## Routine d'exit standard

Lorsque vous écrivez un exit standard, vous spécifiez la fonction d'exit, PSTColMapExit (PSTColMapWExit), ainsi que les paramètres suivants :

### Mono-octet

```
PSTColMapExit  
(PST_STRUCT_CM_EXIT_PARM * pInputParms,  
 PST_STRUCT_CM_EXIT_COL_LIST * pSrcColList,  
 PST_STRUCT_CM_EXIT_COL_LIST * pDstColList)
```

### UTF-16

```
PSTColMapWExit  
(PST_STRUCT_CM_WEXIT_PARM * pInputParms,  
 PST_STRUCT_CM_WEXIT_COL_LIST * pSrcColList,  
 PST_STRUCT_CM_WEXIT_COL_LIST * pDstColList)
```

## Paramètres

Lorsqu'un processus appelle une routine d'exit standard, le processus transmet les paramètres suivants à chaque appel :

### pInputParms

Pointeur vers PST\_STRUCT\_CM\_EXIT\_PARM (PST\_STRUCT\_CM\_WEXIT\_PARM). Cette

structure contient des informations sur les tables source et cible, la nature de l'appel en cours, le nombre de paramètres facultatifs spécifiés avec l'exit, les pointeurs vers les zones de travail des fonctions de rappel et les paramètres facultatifs.

La première zone de cette structure est la zone **FuncCode**, identifiée par PST\_CM\_FUNC\_TRANSFORM (PST\_CMW\_FUNC\_TRANSFORM) ou PST\_CM\_FUNC\_TERMINATE (PST\_CMW\_FUNC\_TERMINATE).

La zone *NumParms* contient le nombre de paramètres facultatifs spécifiés avec l'exit de mappe de colonne (de 0 à 8).

La zone *pParm* contient un tableau de pointeurs vers chaque paramètre facultatif spécifié avec l'exit de mappe de colonne.

#### **pScrColList**

Pointeur vers PST\_STRUCT\_CM\_EXIT\_COL\_LIST (PST\_STRUCT\_CM\_WEXIT\_COL\_LIST). Cette structure décrit les colonnes source.

#### **pDstColList**

Pointeur vers PST\_STRUCT\_CM\_EXIT\_COL\_LIST (PST\_STRUCT\_CM\_WEXIT\_COL\_LIST). Cette structure décrit les colonnes cibles.

## **Fonctions de rappel**

Chaque fois qu'un processus Optim appelle une routine d'exit standard, le processus transmet les adresses des fonctions de rappel suivantes :

#### **pPSTGetColValue( )**

Récupère des données pour toutes les colonnes source et pour la plupart des colonnes cibles dans la ligne de données en cours. En règle générale, la routine d'exit appelle cette fonction une seule fois pour récupérer les données d'une colonne source. Toutefois, la routine d'exit peut appeler cette fonction plusieurs fois afin de récupérer les données d'autres colonnes.

#### **pPSTPutColValue( )**

Indique les données de la colonne cible dans la ligne de données en cours. La routine d'exit détermine la valeur de la colonne cible et renvoie la valeur à Optim. Cette fonction doit être appelée à moins que la ligne ne soit rejetée ou que le processus ne soit abandonné.

## **Traitement**

Le traitement standard de la routine d'exit standard est résumé dans les étapes suivantes :

1. A chaque appel d'Optim, la routine d'exit effectue une vérification pour un premier appel. Lors du premier appel, l'exit effectue certaines tâches d'initialisation ainsi qu'un traitement normal (étape 2). Au cours des appels suivants, seul le traitement normal est effectué (étape 2).
2. Optim ne transmet pas les données de colonne à la routine d'exit standard ; toutefois, la routine d'exit peut passer des appels à la fonction de rappel **pPSTGetColValue( )** afin d'obtenir les données des colonnes source. Ces valeurs sont nécessaires pour déterminer la valeur de la colonne cible.
3. Une fois que la routine d'exit génère la valeur cible, l'exit appelle la fonction de rappel **pPSTPutColValue( )** afin d'enregistrer la valeur dans la colonne cible, ou transmet un code retour approprié, décrivant le processus permettant d'ignorer la ligne de données ou d'abandonner.
4. Une fois la dernière ligne de données traitée, Optim passe un appel de fin à la routine d'exit, identifié par la valeur PST\_CM\_FUNC\_TERMINATE (PST\_CMW\_FUNC\_TERMINATE) dans la zone FuncCode. Cet appel invite la routine d'exit à libérer de l'espace de stockage alloué de manière dynamique. Lorsque les tâches finales sont terminées, la routine d'exit transmet un code retour à Optim.

## Codes retour

Les codes retour suivants s'appliquent aux routines d'exit standard :

PST\_CM\_EXIT\_SUCCESS  
(PST\_CMW\_EXIT\_SUCCESS)

Une valeur est affectée à la colonne cible ou cette dernière est transformée avec succès.

PST\_CM\_EXIT\_REJECT\_ROW  
(PST\_CMW\_EXIT\_REJECT\_ROW)

Une valeur ne peut pas être affectée à la colonne cible ou cette dernière ne peut pas être transformée. Annulez la ligne.

PST\_CM\_EXIT\_ABORT\_PROCESS  
(PST\_CMW\_EXIT\_ABORT\_PROCESS)

Erreur fatale. Arrêtez le traitement. Pour renvoyer un message d'erreur, placez le message dans la zone de travail et définissez l'espace inutilisé sur des blancs ou une valeur NULL.

## Exit de format source

Lorsque vous écrivez une routine d'exit de format source pour Optim, vous indiquez la fonction d'exit, PSTColMapAgeSrcExit (PSTColMapAgeSrcWExit), ainsi que les paramètres suivants :

### Mono-octet

```
PSTColMapAgeSrcExit  
(PST_STRUCT_CM_AGE_SRCFMT_PARM * pInputParms,  
 PST_STRUCT_CM_EXIT_COL_LIST * pSrcColList,  
 PST_STRUCT_CM_EXIT_COL_LIST * pDstColList)
```

### UTF-16

```
PSTColMapAgeSrcWExit  
(PST_STRUCT_CM_AGE_SRCFMT_WPARAM *  
 pInputParms,  
 PST_STRUCT_CM_WEXIT_COL_LIST * pSrcColList,  
 PST_STRUCT_CM_WEXIT_COL_LIST * pDstColList)
```

## Paramètres

Lorsqu'un processus Optim appelle une routine d'exit de format source, le processus transmet les paramètres suivants :

### pInputParms

Pointeur vers PST\_STRUCT\_CM\_AGE\_SRCFMT\_PARM (PST\_STRUCT\_CM\_AGE\_SRCFMT\_WPARAM). Cette structure contient des informations sur les tables source et cible, la nature de l'appel en cours et les pointeurs vers une fonction de rappel et les zones de travail.

La première zone de cette structure est la zone **FuncCode**, identifiée par PST\_CM\_SRCFMT\_TRANSFORM (PST\_CMW\_SRCFMT\_TRANSFORM) ou PST\_CM\_SRCFMT\_TERMINATE (PST\_CMW\_SRCFMT\_TERMINATE).

### pSrcColList

Pointeur vers PST\_STRUCT\_CM\_EXIT\_COL\_LIST (PST\_STRUCT\_CM\_WEXIT\_COL\_LIST). Cette structure décrit les colonnes source.

### pDstColList

Pointeur vers PST\_STRUCT\_CM\_EXIT\_COL\_LIST (PST\_STRUCT\_CM\_WEXIT\_COL\_LIST). Cette structure décrit les colonnes cibles.

## Fonction de rappel

Chaque fois qu'un processus Optim appelle une routine d'exit de format source, le processus transmet l'adresse de la fonction de rappel suivante :

### **pPSTGetColValue( )**

Récupère des données pour toutes les colonnes source et pour la plupart des colonnes cibles dans la ligne de données en cours. En règle générale, la routine d'exit n'a pas besoin d'appeler cette fonction car les données de la colonne source sont fournies dans le premier paramètre. Toutefois, la routine d'exit peut appeler cette fonction afin de récupérer les données d'autres colonnes.

## Traitement

Le traitement standard de la routine d'exit de format source est résumé dans les étapes suivantes :

1. A chaque appel d'Optim, la routine d'exit effectue une vérification pour un premier appel. Lors du premier appel, l'exit effectue certaines tâches d'initialisation ainsi qu'un traitement normal (étape 2). Au cours des appels suivants, seul le traitement normal est effectué (étape 2).
2. L'exit reçoit la valeur de la colonne source, comme indiqué dans la fonction Age définie dans la mappe de colonne. Optim n'applique pas la fonction Age avant d'appeler la routine d'exit et enregistre la valeur brute dans l'une des zones d'union **InputValue**, comme indiqué par la zone **ValueType** du fichier d'en-tête.
3. Si l'exit a besoin d'examiner d'autres colonnes afin de calculer la valeur de la colonne cible, il doit appeler la fonction de rappel **pPSTGetColValue( )** afin d'obtenir la valeur de ces colonnes.
4. Une fois la valeur cible générée, la routine d'exit doit formater la valeur et la placer dans la zone **OutputTimeStamp** ou **OutputSybDateTime**. Optim valide cette valeur et applique la fonction Age. L'exit doit transmettre un code retour approprié indiquant la zone où les données sont sauvegardées, ou décrivant le processus permettant d'ignorer la ligne de données ou d'abandonner.
5. Une fois la dernière ligne de données traitée, Optim passe un appel de fin à la routine d'exit, identifié par la valeur **PST\_CM\_SRCFMT\_TERMINATE** (**PST\_CMW\_SRCFMT\_TERMINATE**) dans la zone **FuncCode**. Cet appel invite la routine d'exit à libérer de l'espace de stockage alloué de manière dynamique. Lorsque les tâches finales sont terminées, la routine d'exit transmet un code retour à Optim.

## Modes d'abandon

La routine d'exit de format source peut abandonner le traitement de plusieurs manières :

- Traiter les lignes dont les dates sont ignorées ou non valides. Si vous sélectionnez l'une des ces options dans une demande de processus, et que les colonnes source et cible ont les mêmes attributs, la colonne source est copiée dans la colonne cible sans être modifiée. Si vous ne sélectionnez aucune option, la ligne est rejetée.
- Rejeter la ligne. Rejetez la ligne sans tenir compte des options de processus pour les dates ignorées ou non valides, en fonction des spécifications de la routine d'exit.
- Abandonner l'intégralité du processus, en fonction des spécifications de la routine d'exit.

La routine d'exit transmet un code retour indiquant le format de date ou le mode d'abandon à utiliser.

Au cours d'un processus, il se peut que la routine d'exit interroge une colonne provenant de la ligne d'entrée et certaines colonnes provenant de la ligne cible. Toutefois, la routine d'exit ne peut pas interroger une colonne cible qui contient une routine d'exit et qui est définie dans la mappe de colonne après la colonne cible en cours. Toutes les autres colonnes cibles sont disponibles.

## Codes retour

Les codes retour suivants s'appliquent aux exits de format source :

```
PST_CM_SRCFMT_USE_TIMESTAMP  
(PST_CMW_SRCFMT_USE_TIMESTAMP)
```

Une valeur est affectée à la colonne cible dans la zone **OutputTimeStamp** du premier paramètre transmis à l'exit.

```
PST_CM_SRCFMT_USE_SYB_DATETIME  
(PST_CMW_SRCFMT_USE_SYB_DATETIME)
```

Une valeur est affectée à la colonne cible dans la zone **OutputSybDateTime** du premier paramètre transmis à l'exit.

```
PST_CM_SRCFMT_SKIP  
(PST_CMW_SRCFMT_SKIP)
```

La fonction Age n'est pas appliquée. Si vous ne sélectionnez pas l'option **Process rows with skipped dates** (Traiter les lignes dont les dates sont ignorées), la ligne est rejetée. Sinon, les données sont copiées dans la source, tant que la source et la cible sont compatibles. Si ce n'est plus le cas, la ligne est rejetée.

```
PST_CM_SRCFMT_COL_INVALID  
(PST_CMW_SRCFMT_COL_INVALID)
```

La fonction Age n'est pas appliquée. Si vous ne sélectionnez pas l'option **Process rows with invalid dates** (Traiter les lignes dont les dates ne sont pas valides), la ligne est rejetée. Sinon, les données sont copiées dans la source, tant que la source et la cible sont compatibles. Si ce n'est plus le cas, la ligne est rejetée.

```
PST_CM_SRCFMT_REJECT_ROW  
(PST_CMW_SRCFMT_REJECT_ROW)
```

Une valeur ne peut pas être affectée à la colonne source. Rejetez (annulez) la ligne.

```
PST_CM_SRCFMT_ABORT_PROCESS  
(PST_CMW_SRCFMT_ABORT_PROCESS)
```

Erreur fatale. Arrêtez le processus. Pour renvoyer un message d'erreur, placez le message dans la zone de travail et définissez l'espace inutilisé sur des blancs ou une valeur NULL.

## Exit de format cible

Lorsque vous écrivez une routine d'exit de format cible pour Optim, vous indiquez la fonction d'exit, `PSTColMapAgeDstExit` (`PSTColMapAgeDstWExit`), ainsi que les paramètres suivants :

### Mono-octet

```
PSTColMapAgeDstExit  
(PST_STRUCT_CM_AGE_DSTFMT_PARM * pInputParms,  
 PST_STRUCT_CM_EXIT_COL_LIST * pSrcColList,  
 PST_STRUCT_CM_EXIT_COL_LIST * pDstColList)
```

### UTF-16

```
PSTColMapAgeDstWExit  
(PST_STRUCT_CM_AGE_DSTFMT_WPARAM * pInputParms,  
 PST_STRUCT_CM_WEXIT_COL_LIST * pSrcColList,  
 PST_STRUCT_CM_WEXIT_COL_LIST * pDstColList)
```

## Paramètres

Lorsqu'un processus Optim appelle une routine d'exit de format cible, le processus transmet les paramètres suivants :

### pInputParms

Pointeur vers `PST_STRUCT_CM_AGE_DSTFMT_PARM` (`PST_STRUCT_CM_AGE_DSTFMT_WPARAM`). Cette structure contient des informations sur les tables source et cible, la nature de l'appel en cours et les pointeurs vers une fonction de rappel et les zones de travail.

La première zone de cette structure correspond à la zone **FuncCode**, identifiée par :

PST\_CM\_DSTFMT\_TO\_CHAR (PST\_CMW\_DSTFMT\_TO\_WCHAR),  
PST\_CM\_DSTFMT\_TO\_INTEGER (PST\_CMW\_DSTFMT\_TO\_INTEGER),  
PST\_CM\_DSTFMT\_TO\_TIMESTAMP (PST\_CMW\_DSTFMT\_TO\_TIMESTAMP),  
PST\_CM\_DSTFMT\_TO\_SYB\_DATETIME (PST\_CMW\_DSTFMT\_TO\_SYB\_DATETIME), ou  
PST\_CM\_DSTFMT\_TERMINATE (PST\_CMW\_DSTFMT\_TERMINATE).

#### **pSrcColList**

Pointeur vers PST\_STRUCT\_CM\_EXIT\_COL\_LIST (PST\_STRUCT\_CM\_WEXIT\_COL\_LIST). Cette structure décrit les colonnes source.

#### **pDstColList**

Pointeur vers PST\_STRUCT\_CM\_EXIT\_COL\_LIST (PST\_STRUCT\_CM\_WEXIT\_COL\_LIST). Cette structure décrit les colonnes cibles.

## **Fonction de rappel**

Chaque fois qu'un processus Optim appelle une routine d'exit de format cible, le processus transmet l'adresse de la fonction de rappel suivante :

#### **pPSTGetColValue( )**

Récupère des données pour toutes les colonnes source et pour la plupart des colonnes cibles dans la ligne de données en cours. En règle générale, la routine d'exit n'a pas besoin d'appeler cette fonction car les données de la colonne source datée sont fournies dans le premier paramètre. Toutefois, la routine d'exit peut appeler cette fonction afin de récupérer les données d'autres colonnes.

## **Formats**

La date d'entrée est aux formats PST\_C\_TIMESTAMP et PST\_C\_SYB\_DATETIME. L'exit transforme cette date dans l'un des formats suivants, en fonction du type de données de la colonne cible.

#### **PST\_C\_CHAR\_SZ**

Colonnes cibles CHAR et VARCHAR.

#### **PST\_C\_INTEGER\_CHAR\_SZ**

Colonnes cibles NUMERIC.

#### **PST\_C\_TIMESTAMP**

Colonnes DATE/TIME DB2 et Oracle.

#### **PST\_C\_SYB\_DATETIME**

Colonnes DATETIME Sybase ASE.

## **Traitement**

Le traitement standard de la routine d'exit de format cible est résumé dans les étapes suivantes :

1. A chaque appel d'Optim, la routine d'exit effectue une vérification pour un premier appel. Lors du premier appel, l'exit effectue certaines tâches d'initialisation ainsi qu'un traitement normal (étape 2). Au cours des appels suivants, seul le traitement normal est effectué (étape 2).
2. L'exit reçoit la valeur de la colonne source, comme indiqué dans la fonction Age définie dans la mappe de colonne. Optim applique la fonction Age avant d'appeler l'exit de mappe de colonne et enregistre la valeur datée dans les zones **InputTimeStamp** et **InputSybaDateTime** du fichier d'en-tête.
3. Si l'exit a besoin d'examiner d'autres colonnes afin de calculer la valeur de la colonne cible, il doit appeler la fonction de rappel **pPSTGetColValue( )** afin d'obtenir la valeur de ces colonnes.
4. Une fois la valeur cible générée, la routine d'exit doit formater la valeur et la placer dans l'une des zones de l'union **OutputValue**. La zone **FuncCode** indique la zone de l'union **OutputValue** dans laquelle

la valeur doit être placée. L'exit doit renvoyer un code approprié indiquant la zone où les données sont sauvegardées, ou décrivant le processus permettant d'ignorer la ligne ou d'abandonner.

5. Une fois la dernière ligne de données traitée, Optim passe un appel de fin à la routine d'exit, identifié par la valeur `PST_CM_DSTFMT_TERMINATE` (`PST_CMW_DSTFMT_TERMINATE`) dans la zone **FuncCode**. Cet appel invite la routine d'exit à libérer de l'espace de stockage alloué de manière dynamique. Lorsque les tâches finales sont terminées, la routine d'exit transmet un code retour à Optim.

## Modes d'abandon

La routine d'exit de format cible peut abandonner le traitement de plusieurs manières :

- Traiter les lignes dont les dates sont ignorées ou non valides. Si vous sélectionnez l'une des ces options dans une demande de processus, et que les colonnes source et cible ont les mêmes attributs, la colonne source est copiée dans la colonne cible sans être modifiée. Si vous ne sélectionnez aucune option, la ligne est rejetée.
- Rejeter la ligne. Rejetez la ligne sans tenir compte des options de processus pour les dates ignorées ou non valides, en fonction des spécifications de la routine d'exit.
- Abandonner l'intégralité du processus, en fonction des spécifications de la routine d'exit.

La routine d'exit transmet un code retour indiquant si la conversion a abouti ou quel mode d'abandon utiliser.

Au cours d'un processus, il se peut que la routine d'exit interroge une colonne provenant de la ligne d'entrée et certaines colonnes provenant de la ligne cible. Toutefois, la routine d'exit ne peut pas interroger une colonne cible qui contient une routine d'exit et qui est définie dans la mappe de colonne après la colonne cible en cours. Toutes les autres colonnes cibles sont disponibles.

## Codes retour

Les codes retour suivants s'appliquent aux exits de format cible :

`PST_CM_DSTFMT_SUCCESS`  
(`PST_CMW_DSTFMT_SUCCESS`)

Une valeur est affectée à la colonne cible, comme indiqué dans la zone **FuncCode** du premier paramètre transmis à l'exit.

`PST_CM_DSTFMT_SKIP`  
(`PST_CMW_DSTFMT_SKIP`)

Si vous ne sélectionnez pas l'option **Process rows with skipped dates** (Traiter les lignes dont les dates sont ignorées), la ligne est rejetée. Sinon, les données source sont copiées dans la cible, tant que la source et la cible sont compatibles. Si ce n'est plus le cas, la ligne est rejetée.

`PST_CM_DSTFMT_COL_INVALID`  
(`PST_CMW_DSTFMT_COL_INVALID`)

Si vous ne sélectionnez pas l'option **Process rows with invalid dates** (Traiter les lignes dont les dates ne sont pas valides), la ligne est rejetée. Sinon, les données sont copiées dans la source, tant que la source et la cible sont compatibles. Si ce n'est plus le cas, la ligne est rejetée.

`PST_CM_DSTFMT_REJECT_ROW`  
(`PST_CMW_DSTFMT_REJECT_ROW`)

Une valeur ne peut pas être affectée à la colonne cible. Rejetez (annulez) la ligne.

`PST_CM_DSTFMT_ABORT_PROCESS`  
(`PST_CMW_DSTFMT_ABORT_PROCESS`)

Erreur fatale. Arrêtez le processus. Pour renvoyer un message d'erreur, placez le message dans la zone de travail et définissez l'espace inutilisé sur des blancs ou une valeur NULL.

---

## Écriture de procédures de mappe de colonne avec le scriptage Lua

Une procédure de mappe de colonne est une procédure utilisée pour masquer ou transformer les données dans une colonne lorsque vous exécutez un service. Comme le nom l'indique, vous devez ajouter des procédures de mappe de colonne à une mappe de colonne. Vous pouvez écrire des procédures de mappe de colonne grâce au langage de script Lua.

Pour plus d'informations sur le langage de programmation Lua, voir le site Web Lua à l'adresse <http://www.lua.org>.

## Fonctions LUA pour les procédures de mappe de colonne

Les procédures de mappe de colonne prennent en charge la plupart des fonctions LUA standard. Les procédures de mappe de colonne prennent également en charge les fonctions spécifiques à Optim.

### Noms de fonction standard pour les fonctions définies par l'utilisateur

Utilisez les noms de fonction suivants dans vos procédures de mappe de colonne LUA. Chacune de ces fonctions est appelée automatiquement au point indiqué. Vous pouvez également créer des fonctions avec d'autres noms de fonction et les appeler à partir des fonctions standard.

Nom	Description	Requis
<code>cm_load()</code>	Cette fonction est appelée avant qu'une table ne soit traitée.	Non
<code>cm_unload()</code>	Cette fonction est appelée après que toutes les tables aient été traitées.	Non
<code>cm_starttable()</code>	Cette fonction est appelée au début du traitement pour chaque table.	Non
<code>cm_endtable()</code>	Cette fonction est appelée à la fin du traitement pour chaque table.	Non
<code>cm_transform()</code>	Cette fonction est appelée pour chaque ligne traitée.	Oui

Chaque procédure de mappe de colonne doit contenir une fonction `cm_transform()`. Si la fonction `cm_transform()` n'existe pas, le code a pour résultat une expression complexe.

### Fonctions globales

Les fonctions suivantes sont disponibles dans tous les contextes d'exécution d'une procédure de mappe de colonne :

Nom	Description
<code>numparms()</code>	Permet d'obtenir le nombre de paramètres transmis à la procédure de mappe de colonne.
<code>parms.get(<i>n</i>)</code>	Permet d'obtenir la valeur du paramètre à l'index <i>n</i> .
<code>print()</code>	Permet d'imprimer les messages dans le rapport de processus.
<code>rejectrow()</code>	Permet d'ignorer une ligne et de passer à la ligne suivante.

## Fonctions du magasin de données

Utilisez les fonctions suivantes pour obtenir des informations sur les magasins de données source et cible :

Nom	Description
source.getdbalias()	Permet d'obtenir l'alias de base de données du magasin de données source.
source.getcreatorid()	Permet d'obtenir l'ID créateur du magasin de données source.
target.getdbalias()	Permet d'obtenir l'alias de base de données du magasin de données cible.
target.getcreatorid()	Permet d'obtenir l'ID créateur du magasin de données cible.

## Fonctions de table

Utilisez les fonctions suivantes pour obtenir des informations sur les tables source et cible :

Nom	Description
source.gettablename()	Permet d'obtenir le nom de la table source.
target.gettablename()	Permet d'obtenir le nom de la table cible.

## Fonctions de colonne

Utilisez les fonctions suivantes pour obtenir des informations sur les colonnes source et cible, transformer les données de colonne, écrire le résultat dans la colonne cible et quitter :

Nom	Description
source.column.getvalue()	Permet d'obtenir une valeur non numérique à partir d'une colonne source.  Une erreur d'exécution est générée si vous utilisez la fonction <code>source.column.getvalue()</code> pour obtenir des valeurs à partir de colonnes du type de données <code>PST_SQL_BINARY</code> .
source.column.getasdouble()	Permet d'obtenir une valeur numérique à partir d'une colonne source (les procédures de mappe de colonne traitent les données numériques au format à double précision).  Utilisez cette fonction pour les colonnes des types de données suivants : <code>PST_SQL_DECIMAL</code> , <code>PST_SQL_DOUBLE</code> , <code>PST_SQL_FLOAT</code> , <code>PST_SQL_INFX_DEC_FLOAT</code> et <code>PST_SQL_ORA_NUMBER</code> .
source.column.getlength()	Permet d'obtenir la longueur de la chaîne dans une colonne source.
source.column.getname()	Permet d'obtenir le nom de la colonne source.
source.column.gettype()	Permet d'obtenir le type de données de la colonne source.

Nom	Description
<code>target.column.setvalue()</code>	Permet de définir la valeur d'une colonne cible.  Une erreur d'exécution est générée si vous utilisez la fonction <code>target.column.setvalue()</code> pour définir des valeurs dans des colonnes du type de données <code>PST_SQL_BINARY</code> .
<code>target.column.getlength()</code>	Permet d'obtenir la longueur de la chaîne dans une colonne cible.
<code>target.column.getname()</code>	Permet d'obtenir le nom de la colonne cible.
<code>target.column.gettype()</code>	Permet d'obtenir le type de données de la colonne cible.
<code>optimmask()</code>	Permet d'appeler un fournisseur Optim Data Privacy Providers (ODPP).
<code>userexit()</code>	Permet d'appeler un exit utilisateur.

## Fonctions non prises en charge

Les catégories de fonctions suivantes ne sont pas prises en charge dans les procédures de mappe de colonne.

- Les fonctions d'entrée et de sortie générées dans la bibliothèque io LUA
- `string.dump()`

## Limitations des procédures de mappe de colonne

N'oubliez pas les limitations ci-après lorsque vous écrivez une procédure de mappe de colonne.

### Données numériques traitées au format à double précision

Les procédures de mappe de colonne traitent les données numériques au format à double précision. Vous devez ainsi utiliser la fonction `source.column.getasdouble()` pour obtenir des données numériques à partir de colonnes source.

### Codage

Les procédures de mappe de colonne utilisent le codage UTF-16 dans leur traitement interne.

### Type de données `PST_SQL_BINARY` non pris en charge

Une erreur d'exécution est générée si vous utilisez la fonction `source.column.getvalue()` pour obtenir des valeurs à partir de colonnes du type de données `PST_SQL_BINARY`. Une erreur d'exécution est également générée si vous utilisez la fonction `target.column.setvalue()` pour définir des valeurs dans des colonnes du type de données `PST_SQL_BINARY`.

## Fonctions non prises en charge

Les catégories de fonctions suivantes ne sont pas prises en charge dans les procédures de mappe de colonne.

- Les fonctions d'entrée et de sortie générées dans la bibliothèque io LUA
- `string.dump()`

## Exemple de procédure de mappe de colonne : Procédure générique

L'exemple de procédure de mappe de colonne illustre la structure d'une procédure de mappe de colonne avec ses fonctions standard : `cm_load`, `cm_unload`, `cm_starttable`, `cm_endtable` et `cm_transform`.

```

-----
--
-- Exemple de procédure de mappe de colonne IBM Optim
--
-- Nom :          OptimSample
--
-- Révision :     1.0
--
-- Description :  Présente toutes les capacités des procédures de mappe de colonne Optim/Lua
--
-- Entrée :       Zéro ou plusieurs paramètres, qui seront simplement répercutés vers le
--                rapport de processus Optim
--
--
-- Sortie :       Section dans le rapport de processus Optim qui affiche des informations
--                provenant de cette procédure de mappe de colonne. La colonne en elle-même
--                n'est pas modifiée.
--
--
-----

-- Fonction cm_load - Appelée avant que les tables ne soient traitées
-----
function cm_load()

  print(" *** Début du processus ***")

  colinfoshown = false

  -- Afficher les paramètres transmis à partir de la mappe de colonne

  print(" Nombre d'arguments : " .. string.format("%d", numparms()))
  for i = 1, numparms(), 1 do
    print("   Argument " .. string.format("%d", i-1) .. " " .. parms.get(i-1))
  end

end

end

-----

-- Fonction cm_unload - Appelée après que toutes les tables aient été traitées
-----
function cm_unload()

  print(" *** Fin du processus ***")

end

end

-----

-- Fonction cm_starttable - Appelée au début du traitement pour chaque table
-----
function cm_starttable()

  print(" \nDébut du traitement de la table")
  fullname = source.getdbalias() .. "." .. source.getcreatorid() .. "." .. source.gettablename()
  print("   Table source : " .. fullname)
  fullname = target.getdbalias() .. "." .. target.getcreatorid() .. "." .. target.gettablename()
  print("   Table cible : " .. fullname)

end

end

-----

-- Fonction cm_endtable - Appelée à la fin du traitement pour chaque table
-----
function cm_endtable()

```

```

print(" \nFin du traitement de la table")

end

-----
-- Fonction cm_transform - Appelée pour chaque ligne traitée
-----
function cm_transform()

if (not colinfoshown) then
  colinfoshown = true
  print(" Traitement de la colonne " .. source.column.getname())
  print("      Type : " .. source.column.gettype())
  print("      Longueur : " .. source.column.getlength())
end

-- Cette instruction obtient la valeur dans la colonne pour laquelle cm_transform a été appelée
-- Le nom d'une autre colonne peut éventuellement être spécifié, par exemple :
-- source.column.getvalue("COL1") renvoie la valeur dans la colonne COL1
oldvalue = source.column.getvalue()

-- Ce code affecte à la colonne cible la même valeur que la colonne source
-- La logique de modification de la valeur serait placée ici.
-- Si vous ne souhaitez PAS insérer cette ligne dans la table cible, appelez la fonction rejectrow()
newvalue = oldvalue
target.column.setvalue(newvalue)
end

```

## Exemple de procédure de mappe de colonne : consultation commutée

L'exemple de procédure de mappe de colonne illustre la façon dont la fonction mask\_parms masque une colonne à l'aide des données d'une table de consultation.

```

-----
--
-- Exemple de procédure de mappe de colonne IBM Optim
--
-- Nom :          OptimSwitchedLookup
--
-- Révision:      1.0
--
-- Description :  Masque une colonne grâce à la recherche de table. La table de recherche à
--                utiliser est déterminée par la valeur d'une autre colonne.
--
-- Entrée :       Paramètre 1 (obligatoire) :
--                Chaîne qui indique le type de recherche à utiliser :
--                HASH, RANDOM ou LOOKUP
--
--                Paramètre 2 .... n-1 (obligatoire)
--                Expression qui indique la table de recherche à utiliser. Le format est
--                COND(colonne-nom=valeur) DATASOURCE(paramètres_source_de_données)
--                Ce paramètre peut être répété plusieurs fois. Si une ligne ne
--                satisfait aucun des paramètres COND, elle ne sera pas insérée
--                dans la table cible.
--
--                Paramètre n (facultatif) :
--                Chaîne contenant des paramètres supplémentaires à
--                copier dans l'appel optimmask.
--                Il s'agit d'un ajout à la valeur datasource_parameters dans
--                la clause COND du paramètre 2 et la zone mask_parms_constant
--                déclarée au début de ce script.
--
--
-- Sortie :       - Données de colonne masquée telles que définies par la fonction target.column.setvalue
--                - Texte orienté vers le rapport Optim par la fonction d'impression

```

```

--
-- Codes retour : 0 - Exécution réussie
--               1 - Rejeter ligne (utiliser dans cm_transform ; la ligne ne sera pas insérée
--                   dans la table de destination)
--               2 - Abandonner le processus
--
-- Deux fonctions d'aide programmable sont spécifiées pour indiquer les
-- conditions autres que le code retour 0.
-- Il n'est pas nécessaire de coder une instruction de retour lors de l'utilisation
-- de ces fonctions.
-- error(string) - Cet appel cause l'abandon du processus Optim
--                et la chaîne s'affiche dans le rapport Optim
--                en tant que message d'erreur.
-- rejectrow()   - Cet appel pousse Optim à rejeter la ligne
--                en cours de traitement. La ligne n'est
--                pas insérée dans la table de destination.
--
--
--
-----

```

```

function cm_transform()

-- Changez cette zone pour qu'elle contienne des paramètres qui doivent
-- être placés dans tous les appels optimmask
local mask_parms_constant = 'CACHE=Y,WHENNF=PRE'

--
-- Validez le nombre de paramètres :
--
nparm = numparms()
if (nparm < 2) then
    process_error("La procédure d'appel à la mappe de colonne OptimSwitchedLookup
    doit avoir 2 paramètres ou plus")
end

--
-- Paramètre de processus 1 (type de recherche)
parm = parms.get(0)
if (string.lower(parm) == "hash") then
    provalue = "HASH_LOOKUP"
elseif (string.lower(parm) == "random") then
    provalue = "RANDOM_LOOKUP"
elseif (string.lower(parm) == "lookup") then
    provalue = "LOOKUP"
else
    process_error("Paramètre non valide. HASH, RANDOM ou LOOKUP attendu. Trouvé " .. parm)
end

--
-- Paramètres de traitement COND/DATASOURCE
--
gotcond = false
for parmptr = 1, nparm-1 do
    parm = parms.get(parmptr)
    if (string.lower(string.sub(parm, 1, 5)) == "cond()") then
        gotcond = true
        datasource_parameters = process_cond()
        if (datasource_parameters > "") then
            break;
        end
    end
end
if (not gotcond) then
    process_error("Aucun paramètre COND trouvé")
end
end

```

```

-- Aucun COND ne correspondait à cette ligne, rejetez la ligne
if (datasource_parameters <= "") then
  rejectrow()
end

--
-- Traitez le paramètre optimmask supplémentaire et facultatif
--
lastparm = parms.get(nparm-1)
if (string.lower(string.sub(lastparm, 1, 5)) == "cond(") then
  optimmask_additional_parms = ""
else
  optimmask_additional_parms = lastparm
end

--
-- Générez l'appel à optimmask, émettez l'appel
-- et placez une nouvelle valeur dans la colonne cible
--
mask_parms = "PRO=" .. *provalue .. "," .. mask_parms_constant

-- L'utilisation des variables d'environnement pour stocker l'ID utilisateur et le mot de passe
-- pour la table système est fournie à but illustratif uniquement.
-- Pour plus de sécurité, stockez ces informations dans des variables d'environnement
-- au format chiffré.
userid = os.getenv("optimmaskuserid")
if (userid) then
  mask_parms = mask_parms .. ",USER=" .. userid
end
password = os.getenv("optimmaskpswd")
if (password) then
  mask_parms = mask_parms .. ",PASS=" .. password
end

mask_parms = mask_parms .. "," .. optimmask_additional_parms

oldvalue = source.column.getvalue()
newvalue = optimmask(oldvalue, mask_parms)
target.column.setvalue(newvalue)
end

function process_cond()

  strptr = 6    -- Pointez le premier caractère après "COND("

  -- Obtenir le nom de colonne
  equalsign = string.find(parm, "=", strptr, true)
  if (not equalsign) then
    process_error("Erreur de syntaxe autour du caractère " .. strptr .. " dans l'expression : " .. parm)
  end
  colname = string.sub(parm, strptr, equalsign-1)

  -- Obtenir la valeur de la colonne
  strptr = equalsign + 1
  closeparen = string.find(parm, ")", strptr, true)
  if (not closeparen) then
    process_error("L'expression COND n'a pas de parenthèse de fermeture dans l'expression : " .. parm)
  end
  colvalue = string.sub(parm, strptr, closeparen-1)

  --Debug - Imprimer les résultats de l'analyse :
  --print ("Colname trouvé=" .. colname .. " dans parm : " .. parm)
  --print ("Colvalue trouvé=" .. colvalue .. " dans parm : " .. parm)

```

```

-- Si la condition COND(column-name=value) n'est pas remplie pour cette ligne,
-- ne poursuivez pas le traitement de ce paramètre. Renvoyez "" pour indiquer zéro occurrence.
if (source.column.getvalue(colname) ~= colvalue) then
    return ""
end

-- Correspondance trouvée sur COND, obtenir la valeur DATASOURCE
strptr = closeparen + 1
datasourceValuePtr = string.find(parm, "DATASOURCE(", strptr, true)
if (not datasourceValuePtr) then
    process_error("Clause DATASOURCE non trouvée dans l'expression : " .. parm)
end
-- (Remarque : cette analyse est très simple ; elle part du principe qu'il
-- n'existe pas de texte errant entre les clauses COND et DATASOURCE)
closeparen = string.find(parm, ")", datasourceValuePtr, true)
if (not closeparen) then
    process_error("L'expression COND n'a pas de parenthèse de fermeture dans l'expression : " .. parm)
end
return string.sub(parm, datasourceValuePtr+11, closeparen-1)

end
function process_error(msg)
    errprefix = "Error in column map procedure for column " .. source.column.getname()
    errprefix = errprefix .. " in table " .. source.getdbalias() .. "."
    .. source.getcreatorid() .. "." .. source.gettablename()
    error(errprefix .. ":\n " .. msg)
end
end

```

## Exemple de procédure de mappe de colonne : Masquage du numéro de carte d'identité

L'exemple de procédure de mappe de colonne illustre comment la fonction `mask_parms` masque une colonne contenant un numéro de carte d'identité.

```

-----
--
-- Exemple de procédure de mappe de colonne IBM Optim
--
-- Nom :          OptimNID
--
-- Révision :    1.0
--
-- Description :  Masque une colonne contenant un numéro de carte d'identité en fonction de la
--                valeur d'une autre colonne ou d'un argument de script avec le code pays.
--
-- Entrée :      Paramètre 1 (obligatoire) :
--                Code pays de type chaîne (par ex. FR pour la France)
--                --OU--
--                COL(nom-colonne) où nom-colonne correspond au nom d'une colonne
--                qui contient le code pays
--
--                Paramètre 2 (facultatif) :
--                Chaîne contenant des paramètres supplémentaires à copier dans
--                l'appel optimmask
--                Il s'agit d'un ajout à la zone mask_parms_constant qui est
--                déclaré au début de ce script.
--
--
-- Sortie :      - Données de la colonne masquées, comme défini par la fonction
--                target.column.setvalue
--                - Texte redirigé vers le rapport Optim par la fonction d'affichage
--
-- Codes retour : 0 - Exécution réussie
--                1 - Rejeter la ligne (à utiliser dans cm_transform ; la ligne ne sera pas
--                insérée dans la table cible)
--                2 - Abandonner le processus

```

```

--
--      Deux fonctions d'aide sont spécifiées de sorte à définir les conditions autres
--      que le code retour 0.
--      Aucun code retour n'est requis pour une instruction de retour lorsque vous
--      utilisez ces fonctions.
--      error(string) - Cette appel entraîne l'abandon du processus Optim
--                    et la chaîne apparaît dans le rapport Optim
--                    en tant que message d'erreur.
--      rejectrow()  - Cet appel amène Optim à rejeter la ligne en cours de
--                    traitement. La ligne ne sera pas insérée dans
--                    la table cible.-
--
--
-----

```

```

function cm_transform()

```

```

-- Modifiez cette zone de sorte qu'elle contienne les paramètres qui devraient
-- être placés dans tous les appels optimmask
local mask_parms_constant = 'MTD=REPEATABLE'

--
-- Obtenez les paramètres :
--
nparm = numparms()
if nparm ~= 1 and nparm ~= 2 then
    msg = "Appel de la procédure de mappe de colonne pour la colonne "
        msg = msg .. source.column.getname()
    msg = msg .. " 1 ou 2 paramètres sont obligatoires"
    error(msg)
end
parm1 = parms.get(0)
if (nparm == 2) then
    parm2 = parms.get(1)
end

--
-- Obtenez le code pays dans la zone swivalue en fonction des paramètres
--
if string.lower(string.sub(parm1, 1, 4)) == 'col(' then
    closeparen = string.find(parm1, ')', 5, true)
    if (not closeparen) then
        error("Aucune parenthèse fermante trouvée dans l'expression : " .. parm1)
    end
    colname = string.sub(parm1, 5, closeparen-1)
    swivalue = source.column.getvalue(colname)
else
    swivalue = parm1
end

--
-- Créez l'appel à optimmask, passez l'appel
-- et placez la nouvelle valeur dans la colonne cible
--
mask_parms = 'PRO=NID, SWI=' .. swivalue .. ', ' .. mask_parms_constant
if (parm2) then
    mask_parms = mask_parms .. ', ' .. parm2
end
oldvalue = source.column.getvalue()
newvalue = optimmask(oldvalue, mask_parms)
target.column.setvalue(newvalue)
end

```

---

## Chapitre 6. Gestion de données

Utilisez les utilitaires Optim Designer pour modifier, parcourir et comparer des données. Vous pouvez également créer des tables.

---

### Exploration des données

Utilisez l'utilitaire Explorer pour consulter le contenu du magasin de données d'un fichier sans avoir à le restaurer dans la base de données.

Utilisez la fonction d'aide dans l'utilitaire Explorer pour obtenir plus d'informations sur une rubrique ou une fonction spécifique. Vous pouvez ouvrir une fenêtre d'aide en sélectionnant **Aide > Contenu**, en cliquant avec le bouton droit de la souris sur un élément et en sélectionnant **Qu'est-ce que c'est**, ou en appuyant sur F1.

Pour explorer des données :

1. Dans Optim Designer, cliquez sur **Outils > Explorer** La fenêtre Explorer s'ouvre.
2. Cliquez sur **Fichier > Ouvrir** pour ouvrir la fenêtre Ouvrir et sélectionnez un fichier de données créé par un service. Les fichiers sont stockés dans le répertoire de données spécifié dans l'utilitaire de configuration.
3. Cliquez sur **Ouvrir**. Les tables du fichier de données sont répertoriées dans la fenêtre Explorer.
4. Cliquez sur **Outils > Créer tous les objets sélectionnés**. La table s'ouvre dans la fenêtre Explorer les données.

---

### Edition de données

Utilisez l'éditeur de table pour parcourir et éditer des ensembles de données dont les relations sont intactes dans les tables des bases de données. L'éditeur gère un modèle de donnée volontairement complexe et contenant de nombreuses tables et relations tout en garantissant un ensemble de données dont les références sont intactes.

Utilisez la fonction d'aide de l'éditeur de table pour obtenir plus d'informations sur une rubrique ou une fonction spécifique. Vous pouvez ouvrir une fenêtre d'aide en sélectionnant **Aide > Contenu**, en cliquant avec le bouton droit de la souris sur un élément et en sélectionnant **Qu'est-ce** ou en appuyant sur la touche F1.

Pour éditer des données :

1. Dans Optim Designer, cliquez sur **Outils > Editer** La fenêtre Options d'édition s'ouvre.
2. Entrez un **Nom de table** ou une **définition d'accès nommée**, sélectionnez les options **Affichage initial** ou **Mode**, puis cliquez sur **OK**. La fenêtre de l'éditeur de table s'ouvre.
3. Editez une ou plusieurs tables simultanément.

Validez les changements dans la base de données en déplaçant le pointeur vers une ligne différente ou à l'aide des options du menu. Chaque instance d'une validation compte comme un niveau d'annulation. Un niveau d'annulation est défini comme un changement à une ligne validée dans la base de données.

Lorsqu'un cas d'erreur survient quand vous essayez de valider des données dans la base de données, les données ne sont pas validées mais la tentative compte malgré tout comme un niveau d'annulation. L'éditeur vous permet de restaurer les données modifiées à un point de validation spécifique.

Un ensemble d'extraction est l'ensemble des lignes lues par l'éditeur à partir d'une table unique de base de données.

- Pour annuler les changements et restaurer les lignes dans la version d'origine obtenue dans l'ensemble d'extraction en cours défini pour toutes les tables, sélectionnez **Outils > Annuler**.
- Pour annuler un changement de ligne, cliquez avec le bouton droit de la souris sur la ligne et cliquez sur **Annuler**.
- Pour afficher les versions successives de la ligne validée dans l'ensemble d'extraction en cours et la version d'origine de la ligne dans l'ensemble d'extraction, cliquez avec le bouton droit de la souris sur la ligne et cliquez sur **Annuler...**

---

## Comparaison de données

Utilisez l'utilitaire de comparaison pour comparer les données d'un ensemble de tables source avec les données d'un autre. Vous pouvez définir une demande de comparaison stockée dans le référentiel Optim et enregistrer les résultats d'une comparaison dans un fichier de comparaison.

### Définition d'une demande de comparaison

Utilisez l'éditeur de demande de comparaison pour définir les sources de données à comparer et les options de traitement.

Utilisez la fonction d'aide de l'utilitaire de comparaison pour obtenir plus d'informations sur une rubrique ou une fonction spécifique. Vous pouvez ouvrir une fenêtre d'aide en sélectionnant **Aide > Contenu**, en cliquant avec le bouton droit de la souris sur un élément et en sélectionnant **Qu'est-ce** ou en appuyant sur la touche F1.

Pour définir une demande de comparaison :

1. Dans l'explorateur de référentiel, cliquez avec le bouton droit de la souris sur **Comparaison**, puis cliquez sur **Nouvelle comparaison**. L'éditeur de demande de comparaison s'ouvre.
2. Dans l'onglet **Général**, procédez comme suit :
  - a. Entrez un nom de fichier de comparaison dans la zone **Fichier de comparaison**.
  - b. sélectionnez **Parcourir immédiatement les résultats** pour afficher les résultats du processus de comparaison lorsqu'il se termine.
  - c. Sélectionnez un mode de **Comparaison**.

#### Table unique

##### Fichier source – fichier source

Comparer les données d'une table unique dans un fichier source avec les données d'un autre fichier source.

##### Fichier source – Table de base de données

Comparer les données d'une table unique dans un fichier source avec les données d'une table unique dans la base de données.

##### Table de base de données – table de base de données

Comparer les données d'une table unique dans une base de données avec les données d'une table unique dans une autre base de données.

#### Tables multiples

##### Fichier source – fichier source

Comparer les données de plusieurs tables dans un fichier source avec les données d'un autre fichier source.

##### Fichier source – définition d'accès

Comparer les données de plusieurs tables d'un fichier source avec les données spécifiées d'une définition d'accès.

### **Fichier source – toutes les tables d'une base de données**

Comparer les données de plusieurs tables d'un fichier source avec les données d'une base de données.

**Remarque :** Ce choix est très utile lors du test d'une application de base de données. Le fichier source représente l'image "avant" par rapport aux tables de la base de données, qui représentent l'image "après".

### **Définition d'accès - définition d'accès**

Comparer les données spécifiée d'une définition d'accès avec les données spécifiées d'une autre définition d'accès.

### **Définition d'accès – Toutes les tables d'une base de données**

Comparer les données spécifiées d'une définition d'accès avec les données d'une base de données.

3. Dans l'onglet **Source**, sélectionnez les sources de données pour comparer en fonction du mode de comparaison.
4. Cliquez sur **File > Exécuter** pour démarrer le processus de comparaison.
5. Cliquez sur **Fichier > Sauvegarder** pour sauvegarder la demande de comparaison. Entrez un nom de demande au format IDENTIFICATEUR.NOM.

Le processus sera créé dans un fichier de comparaison du répertoire de données spécifié dans l'utilitaire de configuration. Utilisez la fonction Parcourir pour afficher les résultats de la comparaison.

## **Edition d'une demande de comparaison**

Utilisez l'éditeur de demande de comparaison pour éditer une demande de comparaison.

Utilisez la fonction d'aide de l'utilitaire de comparaison pour obtenir plus d'informations sur une rubrique ou une fonction spécifique. Vous pouvez ouvrir une fenêtre d'aide en sélectionnant **Aide > Contenu**, en cliquant avec le bouton droit de la souris sur un élément et en sélectionnant **Qu'est-ce** ou en appuyant sur la touche F1.

Pour éditer une demande de comparaison :

1. Dans l'explorateur de référentiel, développez le noeud **Comparaison**, cliquez avec le bouton droit de la souris sur la demande que vous souhaitez éditer, puis cliquez sur **Editer**. L'éditeur de demande de comparaison s'ouvre.
2. Editez la demande.
3. Cliquez sur **Fichier > Sauvegarder** pour sauvegarder la demande de comparaison.

## **Exécution d'un processus de comparaison**

Vous pouvez exécuter un processus de comparaison pour générer un fichier de comparaison que vous pouvez afficher avec la fonction Parcourir.

Pour exécuter une demande de comparaison :

Exécutez la demande avec l'une des méthodes suivantes :

Pour exécuter le processus de comparaison à partir de l'explorateur de référentiel, cliquez avec le bouton droit de la souris sur une demande de comparaison et cliquez sur **Exécuter**.

Pour exécuter le processus de comparaison depuis l'éditeur de demande de comparaison, cliquez sur **Fichier > Exécuter**.

Le processus sera créé dans un fichier de comparaison du répertoire de données spécifié dans l'utilitaire de configuration. Utilisez la fonction Parcourir pour afficher les résultats de la comparaison.

---

## Création de tables

Utilisez l'utilitaire de création pour créer des objets en fonction de définitions issues d'un fichier de données source.

Utilisez la fonction d'aide de l'utilitaire de création pour obtenir plus d'informations sur une rubrique ou une fonction spécifique. Vous pouvez ouvrir une fenêtre d'aide en sélectionnant **Aide > Contenu**, en cliquant avec le bouton droit de la souris sur un élément et en sélectionnant **Qu'est-ce** ou en appuyant sur la touche F1.

Pour créer des tables :

1. Dans Optim Designer, cliquez sur **Outils > Créer**. La fenêtre Options de création s'ouvre.
2. Sélectionnez un fichier de données source créé par un service. Les fichiers sont stockés dans le répertoire de données spécifié dans l'utilitaire de configuration.
3. Sélectionnez une des **Options de mappe de table**.

Si vous choisissez une mappe de table locale, l'éditeur de mappe de table local s'ouvre. Effectuez les étapes suivantes :

- a. Dans la zone **Qualificateur**, entrez un alias de magasin de données cible et l'ID de créateur au format `data_store_alias.creator_id`.
- b. Modifiez les noms des tables de destination.
- c. Cliquez sur **Fichier > Mettre à jour et revenir**.

Si vous sélectionnez une mappe de table nommée, procédez comme suit :

- a. Dans la zone **Nom de mappe de table**, entrez ou sélectionnez un nom de mappe de table.
4. Cliquez sur **OK**. La fenêtre Création s'ouvre.
5. Editez et sélectionnez les objets à créer.
6. Cliquez sur **Outils > Créer tous les objets sélectionnés** pour commencer le processus de création.

---

## Chapitre 7. Gestion de données avec Optim Designer

Optim Designer fournit une interface de conception unique pour la confidentialité des données et pour tester les solutions de gestion des données Optim.

---

### Extraction de données avec Optim Designer

Ce tutoriel explique comment utiliser Optim Designer pour créer un service d'extraction Optim. Vous y apprendrez comment définir une définition d'accès et un service d'extraction.

Au terme de ce tutoriel, vous serez capable d'exécuter un service d'extraction.

#### Objectifs d'apprentissage

A la fin de ces exercices, vous saurez :

- Créer un dossier dans l'explorateur de référentiel
- Créer une définition d'accès définissant les données à extraire
- Editer une définition d'accès pour qu'elle définisse des critères de sélection
- Créer un service d'extraction
- Tester un service d'extraction

#### Durée

Ce module dure environ 60 minutes.

#### Prérequis

Ce tutoriel requiert un environnement Optim Designer disposant d'une connexion à une connexion à un référentiel Optim et un alias de magasin de données qui contient les données d'échantillon Optim.

Ce tutoriel peut être réalisé dans l'environnement Optim Designer.

### Création d'un dossier dans l'explorateur de référentiel

Au cours de cet exercice, vous allez créer un dossier dans l'explorateur de référentiel. Un dossier d'explorateur de référentiel contient des services, des définitions d'accès, des mappes de colonne et des mappes de table. Utilisez les dossiers pour organiser ces objets.

Pour créer un plan d'accès aux données et une règle de sélection :

1. Cliquez avec le bouton droit de la souris sur la vue de l'explorateur de référentiel, puis cliquez sur **Nouveau > Dossier**. La fenêtre Nouveau dossier s'ouvre.
2. Dans **Nom**, tapez Tutoriel, puis cliquez sur **OK**.

Vous avez créé un dossier nommé Tutoriel dans l'explorateur de référentiel.

### Création d'une définition d'accès

Au cours de cet exercice, vous allez créer une définition d'accès. Utilisez les définitions d'accès pour spécifier les tables, la traversée de relation et les critères de sélection des données à traiter.

Avant de créer une définition d'accès, un alias de magasin de données doit exister pour la base de données qui contient la table initiale.

Pour créer une définition d'accès :

1. Développez le dossier Tutoriel dans l'explorateur du référentiel, cliquez avec le bouton droit de la souris sur **Définitions d'accès**, puis cliquez sur **Nouvelle définition d'accès**. L'assistant Nouvelle définition d'accès s'ouvre.
2. Sur la page Entrer un nom de définition d'accès, entrez `SAMPLE.AD`. Cliquez sur **Suivant**.
3. Sur la page Sélectionner un alias de magasin de données contenant les données d'échantillon Optim. Cliquez sur **Suivant**.
4. Sur la page Sélectionner une table initiale, procédez comme suit :
  - a. Dans la zone **Table initiale**, entrez le modèle de recherche suivant : `SCHEMA.OPTIM_CUSTOMERS`, où `SCHEMA` est le schéma contenant les données d'échantillon Optim. Par exemple : `SAMPLE.OPTIM_CUSTOMERS`.
  - b. Cliquez sur **Rechercher des tables**. La liste table affiche les tables qui correspondent au modèle.
  - c. Sélectionnez la table `OPTIM_CUSTOMERS`.
  - d. Cliquez sur **Suivant**.
5. Sur la page Spécifier une méthode de sélection de table, cliquez sur **Rechercher les tables associées**. Cliquez sur **Suivant**.
6. Sur la page Entrer un modèle de table et sélectionner des tables associées, procédez comme suit :
  - a. Cliquez sur **Rechercher des tables**. La liste affiche les tables associées à `OPTIM_CUSTOMERS`.
  - b. Cliquez sur **Sélectionner tout**.
  - c. Cliquez sur **Terminer**.

La nouvelle définition d'accès s'affiche dans l'éditeur de définition d'accès.

## Définition des critères de sélection

Au cours de cet exercice, vous définirez des critères de sélection dans la définition d'accès. Les critères de sélection permettent de mettre en évidence un ensemble spécifique de données en définissant une clause SQL WHERE et en utilisant des variables de substitution avec les valeurs par défaut.

Les critères de sélection doivent respecter la syntaxe SQL et inclure des opérateurs relationnels ou logiques. Les opérateurs logiques et la syntaxe diffèrent en fonction des SGBD. Consultez la documentation appropriée à votre SGBD pour plus d'informations.

Pour sélectionner l'ensemble de données de votre choix pour une table, vous pouvez avoir besoin d'une combinaison d'opérateurs logiques AND et OR.

Pour définir des critères de sélection :

1. Développez le dossier Tutoriel dans l'explorateur de référentiel, développez le noeud **Définitions d'accès**, puis cliquez deux fois sur la définition d'accès `SAMPLE.AD`. L'éditeur de définition d'accès s'ouvre.
2. Sélectionnez l'onglet **Tables**.
3. Sélectionnez la table `OPTIM_CUSTOMERS` dans la liste table.
4. Cliquez sur **Editer les critères de sélection**. La fenêtre Editer les critères de sélection s'ouvre et affiche une clause WHERE pour la table.
5. Entrez les critères suivants : `COUNTRY_CODE='US'`.  
Cliquez sur **Vérifier la syntaxe** pour vérifier la syntaxe et identifier les erreurs.
6. Cliquez sur **OK** pour revenir à l'éditeur de définition d'accès.  
Si la syntaxe est incorrecte, une invite s'ouvre et identifie les erreurs. Vous ne pouvez pas sauvegarder les critères s'ils contiennent des erreurs.
7. Dans le menu principal, cliquez sur **Fichier > Sauvegarder** pour sauvegarder la définition d'accès.

Vous avez défini des critères de sélection qui sélectionneront uniquement des lignes depuis la table OPTIM\_CUSTOMERS dans laquelle la valeur de la colonne COUNTRY\_CODE est "US".

## Création d'un service d'extraction

Au cours de cet exercice, vous allez créer un service d'extraction basé sur les données spécifiées dans une définition d'accès. Utilisez un service d'extraction pour copier un ensemble de lignes liées à partir d'une ou de plusieurs tables et enregistrer ces lignes dans un magasin de données de fichiers.

Pour créer un service d'extraction :

1. Développez le dossier Tutoriel dans l'explorateur de référentiel, cliquez avec le bouton droit de la souris sur **Services**, puis cliquez sur **Nouveau service**. L'assistant Nouveau service s'ouvre.
2. Sur la page Entrer un nom de service et sélectionner un service, procédez comme suit :
  - a. Dans la zone **Nom**, entrez SAMPLE.EXTRACT.
  - b. Dans la liste des types de service, sélectionnez **Extraction**.
  - c. Cliquez sur **Suivant**.
3. Sur la page Sélectionner une définition d'accès, sélectionnez SAMPLE.AD. Cliquez sur **Suivant**.
4. Sur la page Entrer un nom de magasin de données de fichier cible, entrez SAMPLE\_EXT dans la zone **Nom de magasin de données de fichier cible**.
5. Cliquez sur **Terminer**.

Le nouveau service d'extraction s'affiche dans l'éditeur de service d'extraction. Utilisez ce service pour créer un magasin de données de fichier pour un service de conversion, d'insertion ou de chargement. Vous pouvez également utiliser le magasin de données de fichier pour créer une mappe de colonne ou une mappe de table.

---

## Masquage de données avec Optim Designer

Ce tutoriel explique comment utiliser Optim Designer pour créer un service de conversion qui transforme les données d'un magasin de données de fichier.

Ce tutoriel requiert le magasin de données de fichier SAMPLE\_EXT créé durant le tutoriel «Extraction de données avec Optim Designer», à la page 123.

Au terme de ce module, vous serez capable d'appliquer une fonction de confidentialité et d'exécuter un service de conversion.

### Objectifs d'apprentissage

A la fin de ces exercices, vous saurez :

- Créer un dossier dans l'explorateur de référentiel
- Créer une définition d'accès définissant les données à extraire
- Editer une définition d'accès pour qu'elle définisse des critères de sélection
- Créer un service d'extraction
- Tester un service d'extraction

### Durée

Ce module dure environ 60 minutes.

## Prérequis

Ce tutoriel requiert un environnement Optim Designer disposant d'une connexion à une connexion à un référentiel Optim et un alias de magasin de données qui contient les données d'échantillon Optim.

Ce tutoriel peut être réalisé dans l'environnement Optim Designer.

## Création d'une mappe de table

Au cours de cet exercice, vous allez créer une mappe de table. Utilisez une mappe de table pour définir les spécifications permettant de mettre en relation les tables source et cible des données compatibles. Vous pouvez mapper des tables dont les noms sont différents, modifier les noms de table, exclure des tables d'un processus ou exclure des mappes de colonne pour accroître le contrôle sur vos données.

Pour créer une mappe de table :

1. Développez le dossier Tutoriel dans l'explorateur de référentiel, cliquez avec le bouton droit de la souris sur **Mappes de table**, puis cliquez sur **Nouvelle mappe de table**. L'assistant Nouvelle mappe de table s'ouvre.
2. Sur la page Entrer un nom de mappe de table, entrez SAMPLE.TMAP dans la zone **Nom**. Cliquez sur **Suivant**.
3. Sur la page Sélectionner un alias de magasin de données, sélectionnez SAMPLE.EXT.
4. Cliquez sur **Terminer**.

La nouvelle mappe de table s'affiche dans l'éditeur de mappe de table.

Vous devez utiliser l'éditeur pour définir des magasins de données cibles et des schémas avant de sauvegarder la mappe de table.

## Modification d'une mappe de table

Au cours de cet exercice, vous utiliserez l'éditeur de mappe de table pour définir des magasins de données cible et des schémas pour les tables source dans une mappe de table.

Pour éditer des données cibles dans une mappe de table :

1. Développez le dossier Tutoriel dans l'explorateur de référentiel, développez le noeud **Mappes de table** et cliquez deux fois sur la mappe de table SAMPLE.TMAP. L'éditeur de mappe de table s'ouvre.
2. Sélectionnez l'onglet **Mappe de table**.
3. Dans la zone **Alias de magasin de données et mappe de schéma**, sélectionnez la ligne qui contient le magasin de données source.
4. Cliquez sur la cellule **Alias de magasin de données cible**, puis sélectionnez l'alias de magasin de données contenant les données d'échantillon Optim.
5. Cliquez sur la cellule **Schéma cible** et sélectionnez le schéma contenant les données d'échantillon Optim.
6. Pour le menu principal, cliquez sur **Fichier > Sauvegarder** pour sauvegarder la mappe de table.

La mappe de table utilise les mêmes tables source et cible pour masquer les données source et conserver le même schéma et le même alias de magasin de données.

## Création d'une mappe de colonne

Au cours de cet exercice, vous allez créer une mappe de colonne. Une mappe de colonne fournit les spécifications requises pour faire correspondre ou exclure des colonnes du processus d'un service de gestion des données. La conversion, l'insertion et le chargement des services doivent référencer une mappe de table, qui peut faire référence à une ou plusieurs mappes. Utilisez une mappe de colonne pour définir les transformations de données avec des fonctions ou des procédures de mappe de colonne.

Pour créer une mappe de colonne :

1. Développez le dossier Tutoriel dans l'explorateur de référentiel, cliquez avec le bouton droit de la souris sur **Mappes de colonne**, puis cliquez sur **Nouvelle mappe de colonne**. L'assistant Nouvelle mappe de colonne s'ouvre.
2. Sur la page Entrer un nom de mappe de colonne, entrez `SAMPLE.CMAP` dans la zone Nom. Cliquez sur **Suivant**.
3. Sur la page Sélectionner un alias de magasin de données de fichier source, sélectionnez `SAMPLE_EXT`. Cliquez sur **Suivant**.
4. Sur la page Sélectionner une table source, procédez comme suit :
  - a. Dans la zone **Modèle de recherche de table**, entrez le modèle de recherche suivant : `ALIAS_MAGASIN_DONNEES.SCHEMA.OPTIM_CUSTOMERS`, où `ALIAS_MAGASIN_DONNEES.SCHEMA` sont les alias des magasins de données et le schéma contenant les données d'échantillon Optim. Par exemple : `OPTIM.SAMPLE.OPTIM_CUSTOMERS`.
  - b. Cliquez sur **Rechercher des tables**. La liste table affiche les tables qui correspondent au modèle.
  - c. Sélectionnez la table `OPTIM_CUSTOMERS`.
  - d. Cliquez sur **Suivant**.
5. Sur la page Sélectionner un alias de magasin de données cible, sélectionnez l'alias de magasin de données qui contient les données d'échantillon Optim. Cliquez sur **Suivant**.
6. Sur la page Sélectionnez une table cible, procédez comme suit :
  - a. Dans la zone **Modèle de recherche de table**, entrez le modèle de recherche suivant : `SCHEMA.OPTIM_CUSTOMERS`, où `SCHEMA` est le schéma qui contient les données d'échantillon Optim. Par exemple : `SAMPLE.OPTIM_CUSTOMERS`.
  - b. Cliquez sur **Rechercher des tables**. La liste table affiche les tables qui correspondent au modèle.
  - c. Sélectionnez la table `OPTIM_CUSTOMERS`.
7. Cliquez sur **Terminer**.

La nouvelle mappe de colonne s'affiche dans l'éditeur de mappe de colonne.

## Application d'une fonction de masquage des données

Au cours de cet exercice, vous appliquerez une fonction de masquage des données à une colonne dans une mappe de colonne. Vous pouvez masquer des données telles que les numéros nationaux d'identité, les numéros de carte de crédit, les dates, les valeurs numériques et les informations personnelles.

Pour appliquer une fonction de masquage des données :

1. Développez le dossier Tutoriel dans l'explorateur de référentiel, développez le noeud **Mappes de colonne** et cliquez deux fois sur la mappe de colonne `SAMPLE.CMAP`. L'éditeur de mappe de colonne s'ouvre.
2. Sélectionnez la colonne `PHONE_NUMBER`.
3. Cliquez sur **Appliquer la fonction**. La fenêtre Appliquer la fonction s'ouvre.
4. Sélectionnez la fonction **Shuffle**. Cliquez sur **OK**. Le nom de la fonction apparaît dans la colonne **Colonne source** et l'éditeur de fonction s'ouvre dans l'éditeur de mappe de colonne.
5. Dans la zone **Expression de mappe de colonne** de l'éditeur de fonction, entrez `SHUFFLE(RETRY=12)`.
6. Dans le menu principal, cliquez sur **Fichier > Sauvegarder** pour sauvegarder la mappe de colonne.

Vous avez appliqué la fonction Shuffle à la colonne `PHONE_NUMBER`. La fonction masquera les données en remplaçant les valeurs de la colonne par d'autres valeurs de la colonne. La fonction recherchera jusqu'à douze fois une valeur de remplacement ne correspondant pas à la valeur source.

## Ajout d'une mappe de colonne à une mappe de table

Dans cet exercice, vous ajouterez une mappe de colonne à une mappe de table. Une mappe de table est requise pour un service de conversion. Utilisez la mappe de colonne associée pour autoriser le service de conversion à exécuter la fonction de masquage des données définie dans la mappe de colonne.

Pour ajouter une mappe de colonne à la mappe de table :

1. Développez le dossier Tutoriel dans l'explorateur de référentiel, développez le noeud **Mappes de table** et cliquez deux fois sur la mappe de table SAMPLE.TMAP. L'éditeur de mappe de table s'ouvre.
2. Sélectionnez l'onglet **Mappe de table**.
3. Dans la zone **Mappe de table**, sélectionnez la table OPTIM\_CUSTOMERS.
4. Cliquez sur **Ajouter une mappe de colonne**. La fenêtre Nouvelle mappe de colonne s'ouvre et présente une liste des mappes de colonne qui contient les tables sélectionnées.
5. Sélectionnez la mappe de colonne SAMPLE.CMAP. Cliquez sur **OK**.
6. Pour le menu principal, cliquez sur **Fichier > Sauvegarder** pour sauvegarder la mappe de table.

Vous avez ajouté la mappe de colonne SAMPLE.CMAP (et sa fonction de masquage de données associée) à la mappe de table SAMPLE.TMAP, qui autorise un service de conversion à transformer des données.

## Création d'un service de conversion

Au cours de cet exercice, vous allez créer un service de conversion pour masquer des données.

Pour créer un service de conversion :

1. Développez le dossier Tutoriel dans l'explorateur de référentiel, cliquez avec le bouton droit de la souris sur **Services**, puis cliquez sur **Nouveau service**. L'assistant Nouveau service s'ouvre.
2. Sur la page Entrer un nom de service et sélectionner un service, procédez comme suit :
  - a. Dans la zone **Nom**, entrez SAMPLE.CONVERT.
  - b. Dans la liste des types de service, sélectionnez **Conversion**.
  - c. Cliquez sur **Suivant**.
3. Sur la page Sélectionner une mappe de table, sélectionnez la mappe de table SAMPLE.TMAP. Cliquez sur **Suivant**.
4. Sur la page Entrer les propriétés cible, entrez SAMPLE\_CONV dans la zone **Nom du magasin de données de fichier cible**.
5. Cliquez sur **Terminer**.

Le nouveau service de conversion s'affiche dans l'éditeur de service de conversion. Vous pouvez utiliser ce service pour masquer la colonne PHONE\_NUMBER dans la table OPTIM\_CUSTOMERS.

---

## Remarques

Le présent document a été développé pour des produits et services offerts aux Etats-Unis.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service IBM puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations  
IBM Canada Ltd.  
3600 Steeles Avenue East  
Markham, Ontario  
L3R 9Z7  
Canada

Pour obtenir les informations sur les licences concernant les produits utilisant un jeu de caractères double octet, contactez le service IBM Intellectual Property Department de votre pays ou envoyez vos demandes par écrit à l'adresse suivante :

IBM World Trade Asia Corporation  
Licensing 2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan

**Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun autre pays dans lequel il serait contraire aux lois locales.** LE PRESENT DOCUMENT EST LIVRE «EN L'ETAT» SANS AUCUNE GARANTIE, EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et, sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation  
Software Interoperability Coordinator, Department 49XA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions du Contrat sur les produits et services IBM, des Conditions internationales d'utilisation de logiciels IBM, ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Tous les tarifs indiqués sont les prix de vente actuels suggérés par IBM et sont susceptibles d'être modifiés sans préavis. Les tarifs appliqués peuvent varier selon les revendeurs.

Ces informations sont fournies uniquement à titre de planification. Elles sont susceptibles d'être modifiées avant la mise à disposition des produits décrits.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

---

## Marques

Les termes qui suivent sont des marques d'International Business Machines Corporation aux Etats-Unis et/ou dans certains autres pays :

IBM

Logo IBM

DB2

AIX

Informix

InfoSphere

Optim

Netezza est une marque d'IBM International Group B.V., une société IBM.

Adobe, Acrobat, PostScript et toutes les marques incluant Adobe sont des marques d'Adobe Systems Incorporated aux Etats-Unis et/ou dans certains autres pays.

Apache Derby est une marque d'Apache Software Foundation.

Eclipse est une marque d'Eclipse Foundation, Inc.

Microsoft, Windows, Windows NT, et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

Java et toutes les marques incluant Java sont des marques de Sun Microsystems, Inc. aux Etats-Unis et/ou dans certains autres pays.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

UNIX est une marque enregistrée de The Open Group aux Etats-Unis et/ou dans certains autres pays.

Les autres noms de sociétés, de produits et de services peuvent appartenir à des tiers.



# Index

## A

- alias de magasin de données
  - présentation 15
- Alias de magasin de données
  - Assistant Nouvel alias de magasin de données 15
  - Assistant Propriétés de connexion 15
  - connexion 16
  - définition 15
  - édition 15
- Assistant Ajout de tables 20
- Assistant Nouveau service 41, 45, 46, 49
- Assistant Nouvel alias de magasin de données 15
- Assistant Nouvelle clé primaire 33
- Assistant Nouvelle définition d'accès 20
- Assistant Nouvelle mappe de colonne 38
- Assistant Nouvelle mappe de table 34
- Assistant Nouvelle relation Optim 30
- Assistant Propriétés de connexion 15
- Assistant Transformation d'une demande en un service Optim 17

## C

- Clés primaires Optim
  - Assistant Nouvelle clé primaire 33
  - création 33
  - Editeur de clé primaire 33
  - édition 33
  - explicite 32
  - générique 32, 33
  - présentation 32
  - sélection de colonnes de clé 33
  - valeur de fonction Propager 92
- codage
  - procédures de mappe de colonne 112
- comparaison de données 120
- consultation commuté
  - exemple 114
- critères de sélection
  - définition 22
  - Fenêtre Editer les critères de sélection 22
  - gestion des relations 23
  - groupement de données 23
  - liste de pointage et découpage 26
  - options d'accès à une table 24
  - options d'échantillonnage des données et du nombre limite de lignes 23
  - présentation 22
  - traversée de relation 23
  - variables 25

## D

- définition d'accès
  - Assistant Nouvelle définition d'accès 20

- définition d'accès (*suite*)
  - création 20
- définitions d'accès
  - ajout de tables 20
  - Assistant Ajout de tables 20
  - critères de sélection 22
  - Editeur de définition d'accès 21, 23
  - éditeur de définition d'accès 21
  - étapes de traversée 21
  - Fenêtre Editer les critères de sélection 22
  - Fenêtre Etapes de traversée 21
  - gestion des relations 23
  - gestion des tables 20
  - liste de pointage et découpage 26
  - liste de sélection directe 26
  - modification des tables en tables de référence ou en tables associées 21
  - options d'accès à une table 24
  - présentation 19
  - suppression de tables 21
  - traversée de relation 23
  - variables 25
- dossier de l'explorateur de référentiel 3

## E

- Editeur de clé primaire 33
- Editeur de définition d'accès 21
  - Onglet de pointage et découpage 26
  - Onglet Propriétés de groupe de données 23
  - onglet Relations 23, 24
  - Onglet Tables 23
- éditeur de définition d'accès 21
  - onglet Relations 23
- Editeur de demande de comparaison 120, 121
- Editeur de mappe de colonne 38, 39
  - Editeur de script Lua 39
  - Fenêtre Application d'une fonction 38, 53
- Editeur de mappe de table 35
  - Fenêtre Nouvelle mappe de colonne 36
  - Onglet de mappe de table 35
- Editeur de relation 32
  - Fenêtre Ajouter une expression de colonne 31
  - Fenêtre Sélectionner une colonne parente 31
  - Fenêtre Sélectionner une colonne enfant 31
  - Fenêtre Sélectionner une colonne parente 31
- Editeur de relations 31
- Editeur de script Lua 39
- Editeur de service d'extraction 42
  - Assistant Modifier la définition d'accès 44
  - Onglet Conversion 42

- Editeur de service d'extraction (*suite*)
  - Onglet Données et objets 42
  - Onglet Options de compression de fichier 43
  - Onglet Propriétés de service 42
  - Onglet Variables 44
- Editeur de service d'insertion 46
  - Assistant Modifier la mappe de table 47
  - Onglet Options de traitement 46
- Editeur de service de chargement 50
  - Assistant Modifier la mappe de table 51
  - Onglet Options de chargement 50
  - Onglet Options de traitement 50
- Editeur de service de conversion 45
  - Assistant Modifier la mappe de table 45
  - Onglet Options de traitement 45
- Editeur de table 119
- édition de table 119
- Explorateur de référentiel
  - dossier de l'explorateur de référentiel 3
  - présentation 2
- Explorateur de répertoire 3
- exploration de données 119
- expressions concaténées 93
- expressions numériques 94

## F

- Fenêtre Ajouter une variable 25
- Fenêtre Application d'une fonction 38, 53
- fenêtre Associer le répertoire 17
- Fenêtre Changer de référentiel 16
- Fenêtre Editer les critères de sélection 22
- Fenêtre Editer une connexion à un référentiel 16
- Fenêtre Etapes de traversée 21
- Fenêtre Exécuter service 51
- Fenêtre Nouveau fichier de pointage et découpage 26
- Fenêtre Nouvelle connexion à un référentiel 16
- Fenêtre Nouvelle mappe de colonne 36
- fonction Age 95
  - ancienneté sémantique 96
  - calcul incrémentiel de l'âge 96
  - exit de format cible 101
  - exit de format source 101
- Fonction aléatoire 89
- fonction d'identité 91
- Fonction de mélange 63, 64
- Fonction de nom d'e-mail formaté 101
- Fonction de nom d'e-mail généré automatiquement 100
- fonction de numérotation aléatoire 101

- fonction de numérotation séquentielle 101
- Fonction de recherche 53
- Fonction de recherche aléatoire 61
- Fonction de recherche hachée 56
- Fonction de séquence Oracle 91
- fonction de série 91
- Fonction de sous-chaîne 88
- fonction Devise 98
  - conversion directe 98
  - triangulation 99
- fonction Propager 92
- Fonction séquentielle 90
- Fonction TRANS CCN 68
- Fonction TRANS COL 75
- Fonction TRANS EML 71
- Fonction TRANS NID
  - INSEE France 80
  - Italie CF 82
  - NIF Espagne 84
  - numéros d'assurance sociale du Canada 79
  - présentation 77
  - R.U. NINO 85
  - SSN américain 87
- Fonction TRANS SSN 65
- Fonctions de recherche 53
- Fonctions littérales et de valeur 94

## G

- gestion de données
  - comparaison de données 120
  - exploration de données 119
  - utilitaire de comparaison 120
  - Utilitaire Explorer 119
  - Utilitaires Optim Designer 119
- gestion des données
  - création de tables 122
  - définition d'une demande de comparaison 120
  - Editeur de demande de comparaison 120, 121
  - Editeur de table 119
  - édition d'une demande de comparaison 121
  - édition de données 119
  - exécution d'une demande de comparaison 121
  - Utilitaire de création 122

## L

- liste de pointage et découpage
  - création 26
  - Fenêtre Nouveau fichier de pointage et découpage 26
  - Onglet de pointage et découpage 26
  - présentation 26
  - sélection 26

## M

- mappes de colonne
  - ajout aux mappes de table 36

- mappes de colonne (*suite*)
  - application d'une fonction de masquage des données 38, 53
  - Assistant Nouvelle mappe de colonne 38
  - création 38
  - création d'une procédure de mappe de colonne 39
  - Editeur de mappe de colonne 38
  - éditeur de script Lua 39
  - édition 38
  - édition d'une procédure de mappe de colonne 39
  - édition de paramètres dans une procédure de mappe de colonne 40
  - Fenêtre Application d'une fonction 38, 53
  - mappage de colonne source 39
  - présentation 36
  - procédure de mappe de colonne 39
  - règles de compatibilité des données 37
  - routines d'exit 101
  - types de données 37
- mappes de table
  - ajout d'une mappe de colonne 36
  - Assistant Nouvelle mappe de table 34
  - création 34
  - Editeur de mappe de table 35, 36
  - édition 35
  - mappes de colonne 36
  - présentation 34
- Mappes de table
  - données cible par défaut 35
  - Editeur de mappe de table 35
  - édition de données cible 35
- masquage des données
  - application d'une fonction de masquage des données 38, 53
  - Constante booléenne 94
  - Constante numérique 94
  - création d'une procédure de mappe de colonne 39
  - édition d'une procédure de mappe de colonne 39
  - édition de paramètres dans une procédure de mappe de colonne 40
  - expressions concaténées 93
  - expressions numériques 94
  - fonction Age 95
  - Fonction aléatoire 89
  - fonction d'identité 91
  - Fonction de mélange 63
  - Fonction de nom d'e-mail généré automatiquement 100
  - fonction de numérotation aléatoire 101
  - fonction de numérotation séquentielle 101
  - Fonction de recherche 53
  - Fonction de recherche aléatoire 61
  - Fonction de recherche hachée 56
  - Fonction de séquence Oracle 91
  - fonction de série 91
  - Fonction de sous-chaîne 88
  - fonction Devise 98

- masquage des données (*suite*)
  - fonction Propager 92
  - Fonction séquentielle 90
  - Fonction TRANS CCN 68
  - Fonction TRANS COL 75
  - Fonction TRANS EML 71
  - Fonction TRANS NID 77
    - INSEE France 80
    - Italie CF 82
    - NIF Espagne 84
    - numéros d'assurance sociale du Canada 79
    - R.U. NINO 85
    - SSN américain 87
  - Fonction TRANS SSN 65
  - fonctions 53
  - Fonctions de recherche 53
  - Fonctions littérales et de valeur 94
  - Littéral chaîne 94
  - Littéral de date/heure 94
  - Littéral hexadécimal 94
  - Nom d'e-mail formaté 101
  - NULL 94
  - présentation 53
  - Registre spécial 94
- masquage du numéro de carte d'identité exemple 117
- masquer les données 53
- Mode intégré Optim Manager 4, 51
- modèle de données
  - création de tables d'échantillon Optim 12
  - création de tables de confidentialité des données 12
  - exemple de données Optim 5
  - présentation 4
  - Table OPTIM\_CUSTOMERS 6
  - Table OPTIM\_DETAILS 8
  - Table OPTIM\_FEMALE\_RATES 11
  - Table OPTIM\_ITEMS 8
  - Table OPTIM\_MALE\_RATES 10
  - Table OPTIM\_ORDERS 7
  - Table OPTIM\_SALES 5
  - Table OPTIM\_SHIP\_INSTR 10
  - Table OPTIM\_SHIP\_TO 9
  - Table OPTIM\_STATE\_LOOKUP 11
  - tables de confidentialité de données 11
- modèles de données
  - définitions d'accès 19
  - présentation 19
- modélisation de données
  - définitions d'accès 19
  - présentation 19

## O

- Optim Designer
  - bases de données prises en charge 13
  - connexions à la base de données 15
  - Explorateur de référentiel 2
  - Explorateur de répertoire 3
  - fonctions d'accessibilité 13
  - mise en route 1
  - perspective Optim 1
  - présentation 1

## P

- perspective Optim
  - Explorateur de référentiel 2
  - Explorateur de répertoire 3
  - présentation 1
- procédure de mappe de colonne 110
  - codage 112
  - création 39
  - édition 39
  - édition de paramètres 40
  - exemple 113, 114, 117
  - fonctions 110
  - format de données numériques 112
  - limitations 112
  - mappes de colonne 39
  - noms réservés 110
- propager la valeur de la clé primaire 92

## R

- référentiel Optim
  - changement de connexion 16
  - connexion 16
  - édition de connexion 16
  - Fenêtre Changer de référentiel 16
  - Fenêtre Editer une connexion à une référentiel 16
  - Fenêtre Nouvelle connexion à un référentiel 16
  - présentation 16
  - raccordement d'annuaire Optim 17
  - transformation d'une demande en un service 17
- relations Optim
  - Assistant Nouvelle relation Optim 30
  - compatibilité des données 28
  - création 30
  - édition de colonnes 31
  - relations génériques 32
  - restrictions 27
- Relations Optim
  - Editeur de relations 31
  - édition 31
  - étendues 27
  - explicite 27
  - expressions de colonne 31
  - générique 27
  - ordre de colonne 31
  - présentation 26
- répertoire Optim
  - Assistant Transformation d'une demande en un service Optim 17
  - connexion 17
  - fenêtre Associer le répertoire 17
  - transformation d'une demande en un service 17
- routines d'exit
  - configurations requises 103
  - écriture 102
  - exemples de fichier d'en-tête 102
  - exit de format cible
    - codes retour 109
    - entrée dans la fonction Age 101
    - fonction 107
    - fonction de rappel 108
    - formats 108

- routines d'exit (*suite*)
  - exit de format cible (*suite*)
    - modes d'abandon 109
    - paramètres 107
    - traitement 108
  - exit de format source
    - codes retour 106
    - entrée dans la fonction Age 101
    - fonction 105
    - fonction de rappel 106
    - modes d'abandon 106
    - paramètres 105
    - traitement 106
  - exit standard 103
    - codes retour 105
    - fonctions de rappel 104
    - paramètres 103
    - traitement 104
  - présentation 101
  - utilisation des bibliothèques de liaison dynamique (DLL) 103

## S

- service d'extraction
  - Assistant Nouveau service 41
  - création 41
  - Editeur de service d'extraction 42, 43, 44
  - édition 42
  - modification de la définition d'accès 44
  - options d'objets et de groupement 42
  - options de compression de fichier 43
  - options de conversion 42
  - options de traitement 42
  - Options de variable 44
  - présentation 41
- service d'insertion
  - Assistant Nouveau service 46
  - création 46
  - Editeur de service d'insertion 46, 47
  - édition 46
  - modification de la mappe de table 47
  - options de traitement 46
  - présentation 46
- service de chargement
  - Assistant Nouveau service 49
  - création 49
  - Editeur de service de chargement 50, 51
  - édition 50
  - fichiers de sortie 48
  - modification de la mappe de table 51
  - Options Chargeur de SGBD 50
  - options de traitement 50
  - présentation 48
- service de conversion
  - Assistant Nouveau service 45
  - création 45
  - Editeur de service de conversion 45
  - édition 45
  - modification de la mappe de table 45
  - options de traitement 45
  - présentation 44
- services de gestion des données
  - Mode intégré Optim Manager 51

- services de gestion des données (*suite*)
  - présentation 41
  - service d'extraction 41
  - service d'insertion 46
  - service de chargement 48
  - service de conversion 44
  - test 51

## T

- test de services 51

## U

- utilitaire de comparaison
  - présentation 120
- Utilitaire de création 122
- Utilitaire Explorer 119
- Utilitaires Optim Designer
  - comparaison de données 120
  - création de tables 122
  - définition d'une demande de comparaison 120
  - Editeur de demande de comparaison 120, 121
  - Editeur de table 119
  - édition d'une demande de comparaison 121
  - édition de données 119
  - exécution d'une demande de comparaison 121
  - exploration de données 119
  - présentation 119
  - utilitaire de comparaison 120
  - Utilitaire de création 122
  - Utilitaire Explorer 119

## V

- variables
  - création 25
  - Fenêtre Ajouter une variable 25
  - présentation 25





