



Guide de configuration pour des sources de données fédérées



Guide de configuration pour des sources de données fédérées

Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant dans la section «Remarques», à la page 413.

Première édition - juillet 2009

Réf. US : SC19-1034-02

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
Tour Descartes
92066 Paris-La Défense Cedex 50*

© Copyright IBM France 2009. Tous droits réservés.

© **Copyright International Business Machines Corporation 1998, 2009.**

Table des matières

Avis aux lecteurs canadiens vii

Chapitre 1. Configuration du serveur fédéré 1

Emission de commandes dans un système fédéré	1
Vérification de la configuration du serveur fédéré	1
Autorisation du serveur fédéré à accéder aux sources de données	1
Définition des variables d'environnement requises pour les clients d'une source de données	2
Vérification de la liaison entre les fichiers de bibliothèque et le logiciel client de la source de données	4
Gestion de la clé et des règles de licence d'utilisation du produit	7
Création d'une base de données fédérée	7
Jeux de codes, séquences de classement et support de langue nationale	8
Configuration du serveur fédéré afin d'accéder aux sources de données	12
Sensibilité à la casse et utilisation correcte des guillemets	15
Configuration de plusieurs serveurs fédérés pour accéder aux sources de données	18

Chapitre 2. Configuration des sources de données 21

Configuration de l'accès aux sources de données BioRS	21
encapsuleur BioRS	21
Ajout de sources de données BioRS à un serveur fédéré	22
Fonctions et requêtes BioRS personnalisées	38
Optimisation des performances de l'encapsuleur BioRS	51
Configuration de l'accès aux sources de données DB2	57
Catalogage d'une entrée de noeud DB2	57
Catalogage de la base de données DB2 distante	58
Enregistrement de l'encapsuleur DB2	58
Enregistrement de définitions de serveur pour les sources de données DB2	60
Création de mappages utilisateur pour les sources de données DB2	62
Test de la connexion au serveur de source de données DB2	63
Enregistrement de pseudonymes pour les tables et les vues DB2	64
Configuration de l'accès aux sources de données Excel	66
Encapsuleur Excel	66
Ajout de sources de données Excel à un serveur fédéré	70
Configuration de l'accès aux sources de données Informix	78

Configuration et test du fichier de configuration client Informix	79
Définition des variables d'environnement Informix	80
Enregistrement de l'encapsuleur Informix	83
Enregistrement des définitions de serveur pour une source de données Informix	85
Création de mappages utilisateur pour une source de données Informix	88
Test de la connexion au serveur Informix	89
Enregistrement de pseudonymes pour les tables, vues et synonymes Informix	91
Configuration de l'accès aux sources de données JDBC	92
Préparation du serveur fédéré pour accéder aux sources de données via JDBC	94
Enregistrement de l'encapsuleur JDBC	95
Enregistrement des définitions de serveur des sources de données JDBC	97
Création de mappages utilisateur pour les sources de données JDBC	100
Test des connexions aux serveurs de sources de données JDBC	101
Enregistrements des pseudonymes des tables et des vues de source de données JDBC	102
Configuration de l'accès aux sources de données Microsoft SQL Server	104
Préparation du serveur fédéré pour l'accès aux sources de données Microsoft SQL Server (Windows)	104
Préparation du serveur fédéré pour l'accès aux sources de données du serveur Microsoft SQL (Linux, UNIX)	105
Définition des variables d'environnement Microsoft SQL Server	106
Enregistrement de l'encapsuleur Microsoft SQL Server	108
Enregistrement des définitions de serveur pour une source de données Microsoft SQL Server	110
Création des mappages utilisateur pour une source de données Microsoft SQL Server	112
Test de la connexion au serveur distant Microsoft SQL Server	114
Enregistrement de pseudonymes pour les tables et vues Microsoft SQL Server	115
Utilisation des informations de traçage ODBC pour identifier et résoudre les incidents de connexion aux sources de données Microsoft SQL Server	116
Configuration de l'accès aux sources de données ODBC	118
Préparation du serveur fédéré pour accéder aux sources de données avec ODBC (Windows)	119
Préparation du serveur fédéré pour accéder aux sources de données avec ODBC (Linux, UNIX)	120
Enregistrement de l'encapsuleur ODBC	121

Enregistrement des définitions de serveur pour une source de données ODBC	124	Prise en charge de l'encapsuleur Sybase pour Adaptive Server Enterprise (ASE)	187
Création d'un mappage utilisateur pour une source de données ODBC	126	Définition des variables d'environnement Sybase.	188
Test de la connexion au serveur de source de données ODBC.	128	Paramétrage et test du fichier de configuration client Sybase (Windows).	191
Enregistrement de pseudonymes pour les tables et vues de la source de données ODBC.	129	Configuration et test du fichier de configuration client Sybase (UNIX)	192
Optimisation des performances de l'encapsuleur ODBC avec l'utilitaire d'optimisation ODBC (db2fedsvr cfg)	131	Enregistrement de l'encapsuleur Sybase	193
Accès aux données Excel à l'aide de l'encapsuleur ODBC	134	Enregistrement des définitions de serveur pour une source de données Sybase.	194
Configuration de l'accès ODBC aux sources de données IBM InfoSphere Classic Federation Server for z/OS	138	Création des mappages utilisateur pour une source de données Sybase	196
Enregistrement de l'encapsuleur ODBC.	140	Test de la connexion au serveur Sybase.	198
Enregistrement des définitions de serveur pour une source de données ODBC.	143	Enregistrement de pseudonymes pour les tables et vues Sybase	199
Création d'un mappage utilisateur pour une source de données ODBC	145	Identification et résolution des incidents de configuration de l'encapsuleur Sybase	200
Test de la connexion au serveur de source de données ODBC.	146	Configuration de l'accès aux sources de données du fichier structuré de la table.	202
Enregistrement de pseudonymes pour les tables et vues de la source de données ODBC.	147	Fichiers structurés sous forme de tables - présentation	202
Optimisation des performances de l'encapsuleur ODBC avec l'utilitaire d'optimisation ODBC (db2fedsvr cfg)	149	Attributs des fichiers structurés sous forme de tables	203
Configuration de l'accès aux sources de données OLE DB	152	Encapsuleur des fichiers structurés sous forme de tables	203
Enregistrement de l'encapsuleur OLE DB	153	Ajout de sources de données de fichiers structurés sous forme de tables à un serveur fédéré	204
Enregistrement des définitions de serveur pour une source de données OLE DB	155	Modèle de contrôle d'accès aux fichiers pour l'encapsuleur de fichier structuré sous forme de tables	214
Création des mappages utilisateur pour une source de données OLE DB.	156	Instructions pour optimiser les performances d'interrogation de l'encapsuleur de fichier structuré sous forme de tables.	214
Configuration de l'accès aux sources de données Oracle	157	Configuration de l'accès aux sources de données Teradata	215
Définition des variables d'environnement Oracle	158	Test de la connexion au serveur Teradata	216
Paramétrage et test du fichier de configuration client Oracle.	162	Vérifier que la bibliothèque Teradata est activée pour la liaison d'exécution (AIX)	216
Enregistrement de l'encapsuleur Oracle.	163	Définition des variables d'environnement Teradata	217
Enregistrement des définitions de serveur pour une source de données Oracle.	164	Enregistrement de l'encapsuleur Teradata	221
Création des mappages utilisateur pour une source de données Oracle	166	Enregistrement des définitions de serveur pour une source de données Teradata	223
Test de la connexion au serveur Oracle	167	Création d'un mappage utilisateur pour une source de données Teradata	225
Enregistrement des pseudonymes pour les tables et les vues Oracle.	169	Test de la connexion au serveur Teradata	226
Configuration de l'accès aux scripts en tant que sources de données	170	Enregistrement de pseudonymes pour les tables et vues Teradata	228
Présentation de l'encapsuleur de script	171	Identification et résolution des incidents avec la configuration de la source de données Teradata	230
Ajout de scripts en tant que sources de données d'un système fédéré	172	Configuration de l'accès aux sources de données des services Web	232
Enregistrement de pseudonymes pour les scripts (ligne de commande DB2)	180	Les services Web et l'encapsuleur de services Web	233
Requêtes SQL avec l'encapsuleur de script	185	Enregistrement de l'encapsuleur de services Web	239
Optimisation des performances de l'encapsuleur de script	186	Enregistrement de la définition de serveur pour des sources de données de services Web	240
Configuration de l'accès aux sources de données Sybase.	186		

Activation de la sécurité à travers l'encapsuleur de services Web	242	Mappages de types de données aval par défaut pour les sources de données Informix	380
Enregistrement de pseudonymes pour les sources de données de services Web	243	Mappages de types de données aval par défaut pour les sources de données JDBC	381
Restrictions de requête pour les encapsuleurs de services Web	256	Mappages de types de données aval par défaut pour les sources de données Microsoft SQL Server	383
Sources de données de services Web - exemples de requêtes	259	Mappages de types de données aval par défaut pour les sources de données ODBC	384
Configuration de l'accès aux sources de données XML	262	Mappages de types de données aval par défaut pour les sources de données Oracle NET8	385
encapsuleur XML	262	Mappages de types de données aval par défaut pour les sources de données Sybase	386
Ajout de données XML à un système fédéré	266	Mappages de types de données aval par défaut pour les sources de données Teradata	388
Requêtes pour des sources de données XML	284	Echantillons de mappages de types de données aval	389
Chapitre 3. Support de source de données pour des fonctions fédérées	291	Mappages de types de données inversés par défaut	391
Chapitre 4. Référence des options des sources de données	295	Mappages de types de données inversés par défaut pour les sources de données d'une base de données DB2 pour Linux, UNIX et Windows	392
Référence des options BioRS	295	Mappages de types de données inversés par défaut pour les sources de données DB2 for System i	392
Référence des options de base de données DB2	299	Mappages de types de données inversés par défaut pour les sources de données DB2 pour VM et VSE	393
Référence des options Excel	306	Mappages de types de données inversés par défaut pour les sources de données DB2 for z/OS	394
Référence des options Informix	307	Mappages de types de données inversés par défaut pour les sources de données Informix	394
Références des options JDBC	313	Mappages de types de données inversés par défaut pour les sources de données JDBC	395
Référence des options Microsoft SQL Server	320	Mappages de types de données inversés par défaut pour les sources de données Microsoft SQL Server	396
Référence des options ODBC	325	Mappages de types de données inverses par défaut pour les sources de données ODBC	397
Référence des options Oracle	331	Mappages de types de données inversés par défaut pour les sources de données Oracle NET8	397
Référence des options de script	336	Mappages de types de données inversés par défaut pour les sources de données Sybase	398
Référence des options Sybase	341	Mappages de types de données inversés par défaut pour les sources de données Teradata	399
Référence des options Teradata	347	Mappages de types de données par défaut Unicode	400
Référence des options des fichiers structurés sous forme de tables	351	Mappages de types de données aval par défaut Unicode pour les sources de données JDBC	400
Référence des options des services Web	354	Mappages de types de données aval par défaut Unicode pour les sources de données Microsoft SQL Server	401
Référence des options XML	362	Mappages de types de données inversés par défaut Unicode pour les sources de données Microsoft SQL Server	401
Chapitre 5. Vues de la table du catalogue global contenant des informations fédérées.	369	Mappages de types de données aval par défaut Unicode pour les sources de données NET8	402
Chapitre 6. Options de mappage de fonctions pour les systèmes fédérés	373		
Chapitre 7. Types de serveur valides dans les instructions SQL	375		
Chapitre 8. Mappages de type de données	377		
Mappages de types de données aval par défaut	377		
Mappages de types de données aval par défaut pour les sources de données d'une base de données DB2 pour Linux, UNIX et Windows	377		
Mappages de types de données aval par défaut pour les sources de données DB2 for System i	378		
Mappages de types de données aval par défaut pour les sources de données DB2 pour VM et VSE	379		
Mappages de types de données aval par défaut pour les sources de données DB2 for z/OS	380		

Mappages de types de données inversés par défaut Unicode pour les sources de données NET8	402
Mappages de types de données aval par défaut Unicode pour les sources de données ODBC	402
Mappages de types de données inversés par défaut Unicode pour les sources de données ODBC	403
Mappages de types de données aval par défaut Unicode pour les sources de données Sybase	403
Mappages de types de données inversés par défaut Unicode pour les sources de données Sybase.	404

Types de données pris en charge pour les sources de données non relationnelles	404
------------------------------------------------------------------------------------------	-----

Documentation du produit.	409
------------------------------------------	------------

Documentation accessible.	411
------------------------------------------	------------

Remarques	413
----------------------------	------------

Marques	416
-------------------	-----

Index	417
------------------------	------------

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.

France	Canada	Etats-Unis
 (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. Configuration du serveur fédéré

Avant de configurer les sources de données, configurez le serveur fédéré et vérifiez que la configuration est correcte.

Procédure

Pour configurer le serveur fédéré, procédez comme suit :

Emission de commandes dans un système fédéré

Si vous êtes novice en matière de fédération, consultez la méthode d'accès au Centre de contrôle DB2 et la méthode de démarrage de la ligne de commande DB2. Utilisez ces interfaces pour exécuter plusieurs tâches au sein d'un système fédéré.

Le Centre de contrôle affiche des boîtes de dialogue et des assistants qui vous aident à exécuter une tâche et ne nécessitent aucune connaissance des commandes DB2 requises pour exécuter une tâche. Si vous êtes novice dans l'utilisation de DB2 et en matière de fédération, vous pouvez avoir recours au Centre de contrôle, dont l'utilisation est intuitive et simple. Pour utiliser la ligne de commande DB2, entrez les commandes DB2 requises pour exécuter une tâche. Si vous savez déjà comment utiliser DB2, il est alors préférable d'utiliser la ligne de commande. Lorsque les deux interfaces sont disponibles pour exécuter une tâche, la documentation décrit les étapes spécifiques à suivre, ainsi que les commandes à saisir.

Pour accéder à une interface, procédez comme suit :

- Pour lancer la ligne de commande DB2, entrez `db2` à l'apparition de l'invite du système d'exploitation. Ensuite, vous pouvez entrer des commandes, une à une. Lorsque vous avez terminé d'utiliser la ligne de commande DB2, entrez `quit`.
- Pour utiliser le Centre de contrôle DB2, développez le dossier **Objets fédérés** pour afficher la liste des objets que vous pouvez utiliser. Ensuite, sélectionnez un objet et utilisez les panneaux et les assistants pour exécuter une tâche.

Vérification de la configuration du serveur fédéré

Lors de l'installation de la fédération, le logiciel essaie de configurer le serveur fédéré à votre place. Pour éviter tout problème lors de la configuration des sources de données, vérifiez que le serveur fédéré est correctement configuré.

Pour vérifier la configuration du serveur fédéré, procédez comme suit :

Autorisation du serveur fédéré à accéder aux sources de données

Le paramètre `FEDERATED` doit être défini sur `YES` pour permettre au serveur fédéré d'accéder aux sources de données.

Le programme d'installation essaie de définir le paramètre `FEDERATED`, mais il est recommandé de confirmer qu'il a été correctement défini sur `YES`. Si le

paramètre FEDERATED n'a pas été correctement défini, toutes les tentatives d'accès aux sources de données vont échouer et afficher le message SQL20076, code raison 1.

Procédure

Pour vérifier que le paramètre FEDERATED a été correctement défini et activé (le cas échéant), procédez comme suit :

1. A partir de la ligne de commande DB2, émettez la commande suivante pour afficher les paramètres et leur configuration actuelle :
GET DATABASE MANAGER CONFIGURATION
2. Vérifiez la valeur du paramètre MAX_CONNECTIONS afin de déterminer si le concentrateur est activé :
 - Si MAX_CONNECTIONS est égal à MAX_COORDAGENTS, le concentrateur est désactivé.
 - Si MAX_CONNECTIONS est supérieur à MAX_COORDAGENTS, le concentrateur est activé.

Le paramètre MAX_CONNECTIONS ne peut pas être actif lorsque le paramètre FEDERATED est défini sur YES. Si le concentrateur est activé, modifiez la valeur de MAX_CONNECTIONS afin qu'elle soit égale à celle de MAX_COORDAGENTS.

3. Vérifiez la configuration du paramètre FEDERATED. S'il est défini sur NO, choisissez alors valeur YES. Émettez la commande suivante pour modifier le paramètre :
UPDATE DATA.BASE MANAGER CONFIGURATION USING FEDERATED YES

Définition des variables d'environnement requises pour les clients d'une source de données

Vérifiez que les variables d'environnement requises sont définies dans le fichier db2dj.ini. Cette tâche est obligatoire pour les sources de données Informix, Microsoft® SQL Server, Oracle, Sybase et Teradata, ainsi que pour les applications qui utilisent les encapsuleurs JDBC et ODBC.

Avant de commencer

L'administrateur système exécute cette tâche.

Si vous installez le logiciel client avant d'installer le serveur fédéré, le programme d'installation définit automatiquement les variables d'environnement requises pour les sources de données sélectionnées. Si vous installez le logiciel client après avoir installé le serveur fédéré, effectuez une mise à niveau vers une nouvelle version du logiciel client ou migrez vers un nouveau matériel, vous devez alors définir manuellement les variables d'environnement requises.

La table suivante répertorie ces variables, par source de données.

Tableau 1. Variables d'environnement requises, par source de données

Source de données	Variable requise	Valeur
Informix	INFORMIXDIR INFORMIXSERVER	Pour INFORMIXDIR, le répertoire contenant le logiciel client Informix. . Pour INFORMIXSERVER, le nom du serveur Informix par défaut.

Tableau 1. Variables d'environnement requises, par source de données (suite)

Source de données	Variable requise	Valeur
JDBC	Varie, en fonction du pilote JDBC installé. Si vous ne spécifiez pas le module du pilote JDBC dans le paramètre du serveur DRIVER_PACKAGE de l'instruction CREATE SERVER, vous devez spécifier le module du pilote JDBC dans la variable d'environnement système "CLASSPATH". Pour obtenir des informations complémentaires sur la définition des variables d'environnement système, reportez-vous à la documentation de votre pilote JDBC.	Nom de fichier et répertoire de la bibliothèque du pilote JDBC.
Microsoft SQL Server	DJX_ODBC_LIBRARY_PATH	Répertoire contenant la bibliothèque ODBC.
ODBC	Varie, en fonction de l'application ODBC.	Varie, en fonction de l'application ODBC.
Oracle	ORACLE_HOME	Répertoire contenant le logiciel client Oracle.
Sybase	SYBASE, SYBASE_OCS	Pour SYBASE, le répertoire contenant le logiciel client Sybase . Pour SYBASE_OCS, le répertoire, la version et l'édition du logiciel client .
Teradata	COPYLIB	Répertoire contenant le logiciel client Teradata.

Procédure

Pour définir manuellement les variables d'environnement, procédez comme suit :

1. Ouvrez le fichier db2dj.ini.

Le fichier db2dj.ini est ajouté au serveur fédéré au cours de l'installation. Par défaut, le fichier se trouve à l'emplacement spécifié par la variable de registre DB DB2_DJ_INI. Si cette dernière n'est pas définie, le fichier se trouve dans les emplacements suivants :

- Sous UNIX®, le fichier se trouve dans *instance_de_base/sqllib/cfg/db2dj.ini*, où *instance_de_base* correspond au répertoire de base du propriétaire de l'instance.
- Sous Microsoft Windows®, le fichier se trouve dans *%CHEMINDB2%\cfg\db2dj.ini*, où *%CHEMINDB2%* correspond au répertoire dans lequel le système de base de données DB2 est installé. Par exemple, C:\Program Files\IBM\sqllib.

2. Utilisez la syntaxe *nom_variable=valeur_variable* pour définir la valeur de la variable d'environnement. Si *valeur_variable* correspond à un nom de fichier ou de répertoire, vous devez spécifier le chemin qualifié complet. Par exemple, pour définir la variable INFORMIXDIR du répertoire informix dans le répertoire de base /home/user1, insérez l'entrée suivante dans le fichier db2dj.ini :
INFORMIXDIR=/home/user1/informix
3. A partir de la ligne de commande DB2, émettez les commandes suivantes :
db2stop
db2start
Arrêtez, puis redémarrez l'instance de base de données pour vérifier que les variables d'environnement sont effectives.
4. Si vous configurez un système multipartition, copiez le fichier db2dj.ini dans chaque partition.

Restrictions pour le fichier db2dj.ini

Le fichier db2dj.ini contient les variables d'environnement requises et facultatives pour une source de données.

Lors de la modification du fichier db2dj.ini, n'oubliez pas les restrictions suivantes.

Pour spécifier chaque variable, utilisez le format *nom_variable=valeur_variable*

où

- *nom_variable* correspond au nom de la variable d'environnement.
- *valeur_variable* correspond à la valeur.

Pour spécifier un nom de fichier ou un nom de répertoire en tant que *valeur_variable*, spécifiez le nom qualifié complet. Ce nom ne peut pas contenir de méta-caractères pour les noms de fichier, comme ~ (tilde), et de variables d'environnement, comme \$HOME. Par exemple, pour définir la variable d'environnement INFORMIXDIR dans le répertoire informix et dans le répertoire de base /home/user1, spécifiez cette valeur dans le fichier db2dj.ini :

```
INFORMIXDIR=/home/user1/informix
```

N'oubliez pas les restrictions suivantes :

- la longueur maximale du nom d'une variable d'environnement est de 255 octets
- la longueur maximale de la valeur d'une variable d'environnement est de 756 octets
- la longueur maximale d'une ligne dans le fichier db2dj.ini est de 1021 octets ; les données situées au-delà de cette limite sont ignorées.

Vérification de la liaison entre les fichiers de bibliothèque et le logiciel client de la source de données

Pour obtenir des résultats optimaux, vérifiez toujours que les fichiers de la bibliothèque sont correctement liés au logiciel client de la source de données avant d'utiliser la commande CREATE WRAPPER pour enregistrer l'encapsuleur. Cette tâche doit uniquement être exécutée lorsque le serveur fédéré utilise UNIX et lorsque le client de la source de données est Informix, Microsoft SQL Server, Oracle, Sybase ou Teradata.

Lors de l'installation de la fédération, vous pouvez choisir de configurer automatiquement les encapsuleurs. Si vous installez le logiciel client de la source de données sur le serveur fédéré avant d'installer la fédération sur le serveur et

que vous choisissez de configurer automatiquement l'encapsuleur au cours du programme d'installation, les fichiers de bibliothèque d'encapsuleur sont liés au logiciel client de la source de données correspondante. Si vous installez le logiciel client de la source de données après avoir installé la fédération, par exemple pour effectuer une mise à jour vers une nouvelle version du logiciel client, vous devez lier manuellement les fichiers de bibliothèque d'encapsuleur au logiciel client de la source de données.

Les raisons principales expliquant l'échec d'une liaison sont les suivantes :

- Une variable d'environnement requise est indisponible au cours de l'installation.
- Le client de la source de données n'est pas installé sur le serveur fédéré.
- Le client de la source de données installé sur le serveur fédéré ne se trouve pas à un niveau pris en charge.

Procédure

Pour vérifier que les fichiers de bibliothèque sont correctement liés, procédez comme suit :

1. Ouvrez le fichier de message de la source de données. Le fichier de message se trouve dans le répertoire où le serveur fédéré est installé, dans le sous-répertoire lib32 ou lib64. Cette table répertorie les sources de données, les fichiers de messages et les fichiers de bibliothèque.

Tableau 2. Sources de données, fichiers de messages et fichiers de bibliothèque

Source de données	Fichier de message	Fichiers de bibliothèque
Informix	djxlinkInformix.out	AIX - libdb2informix.a HP-UX - libdb2informix.so Linux - libdb2informix.so Solaris - libdb2informix.so
Microsoft SQL Server	djxlinkMssql.out	AIX - libdb2mssql3.a HP-UX - libdb2mssql3.so Linux - libdb2mssql3.so Solaris - libdb2mssql3.so
Oracle	djxlinkOracle.out	AIX - libdb2net8.a HP-UX - libdb2net8.so Linux - libdb2net8.so Solaris - libdb2net8.so
Sybase	djxlinkSybase.out	AIX - libdb2ctlib.a HP-UX - libdb2ctlib.so Linux - libdb2ctlib.so Solaris - libdb2ctlib.so
Teradata	djxlinkTeradata.out	AIX - libdb2teradata.a HP-UX - libdb2teradata.so Linux - libdb2teradata.so Solaris - libdb2teradata.so

2. Évaluez le contenu du fichier de message :

- Si la liaison a réussi, le fichier de bibliothèque s'affiche dans le répertoire où la fédération est installée et le fichier de message affiche un message confirmant la réussite de la liaison.
- Si la liaison a échoué, le fichier de message affiche un message d'erreur. Si ni le fichier de bibliothèque, ni le fichier de message ne s'affiche dans le répertoire où la fédération est installée, vous devez lier manuellement le fichier de bibliothèque au logiciel client de la source de données.

Liaison manuelle des fichiers de bibliothèque au logiciel client de la source de données

Si les fichiers de bibliothèque ne s'affichent pas dans le chemin de répertoire, vous devez exécuter un script pour les lier au logiciel client de la source de données.

Avant de commencer

- Le logiciel client de la source de données doit être installé et configuré sur le serveur fédéré.
- Par défaut, les messages des scripts sont émis en anglais. Pour émettre des messages dans une langue autre, une base de données fédérée doit être configurée afin de pouvoir utiliser la page de codes dans une langue autre que l'anglais.
- Les variables d'environnement requises de la source de données doivent être définies.

Tableau 3. Sources de données, scripts et variables d'environnement requises

Source de données	Script	Variable d'environnement requise
Informix	djxlinkInformix	INFORMIXDIR
Microsoft SQL Server	djxlinkMssql	DJX_ODBC_LIBRARY_PATH
Oracle	djxlinkOracle	ORACLE_HOME
Sybase	djxlinkSybase	SYBASE, SYBASE_OCS
Teradata	djxlinkTeradata	COPLIB

Procédure

Pour lier les fichiers de bibliothèque au logiciel client de la source de données, procédez comme suit :

1. Ouvrez une invite de commande UNIX et entrez la commande suivante :
`cd /opt/IBM/db2/V9.5/bin`
2. A partir de la ligne de commande DB2, entrez les commandes suivantes :
`db2 disconnect all`
`db2stop`
3. Exécutez un script pour chaque source de données à laquelle vous voulez accéder. Pour obtenir les noms des scripts, reportez-vous à la table ci-dessus.
4. Si vous installez un environnement composé de plusieurs partitions, répétez les étapes 1 à 3 pour chaque instance de la base de données.
5. Pour chaque instance de la base de données, accédez au répertoire *rép_install*/instance, où *rép_install* correspond au répertoire d'installation DB2 et entrez la commande suivante :
`./db2iupdt nom_instance`
Cette commande met à jour la configuration de l'instance et lui permet d'accéder aux sources de données.
6. Pour chaque instance de la base de données, affichez les autorisations d'accès aux fichiers dans les fichiers de bibliothèque. Vérifiez que le propriétaire d'instance de la base de données est autorisé à lire et à exécuter des fichiers.

Gestion de la clé et des règles de licence d'utilisation du produit

Chaque ordinateur où IBM® InfoSphere Federation Server est installé doit disposer d'un fichier de licence spécifiant la clé de licence d'utilisation du produit. Les règles relatives à une licence contrôlent et surveillent le nombre d'utilisateurs autorisés à se connecter au serveur fédéré.

La clé de licence se trouve dans le répertoire de licences d'utilisation du logiciel d'installation. Si elle n'est pas enregistrée, vous devez l'enregistrer manuellement.

1. Localisez votre fichier de licences dans le répertoire license du logiciel d'installation du produit :
 - isfs.lic** correspond au fichier de licence d'utilisation du produit complet
 - isfs_t.lic** correspond au fichier de licence d'utilisation du produit d'essai
 - isfs_d.lic** correspond au fichier de licence de l'édition développeur
2. Si le fichier de licences ne figure pas dans le répertoire, vous devez enregistrer manuellement une clé de licence, pour cela reportez-vous à la section .
3. Spécifiez les règles relatives à la licence pour contrôler et surveiller les utilisateurs autorisés à se connecter au serveur fédéré.

Création d'une base de données fédérée

Avant de configurer le serveur fédéré pour accéder aux sources de données, vous devez créer une base de données fédérée.

Avant de commencer

- Vous devez disposer d'un droit SYSADM ou SYSCTRL pour créer une base de données.
- La fédération doit être installée sur un serveur qui va agir comme serveur fédéré.

Si l'instance de base de données utilise une configuration composée de plusieurs partitions, toutes les partitions répertoriées dans le fichier db2nodes.cfg sont affectées lors de la création de la base de données. La partition de base de données à partir de laquelle la commande CREATE DATABASE est émise devient la partition de catalogue de la nouvelle base de données.

Procédure

Pour créer une base de données fédérée, procédez comme suit :

1. Déterminez le jeu de codes et la séquence de classement à spécifier lors de la création de la base de données fédérée.
2. A partir de la ligne de commande DB2, émettez la commande CREATE DATABASE.

Par exemple, pour créer une base de données nommée "fédérée" qui utilise le jeu de codes ISO8859-15 sur le territoire brésilien, émettez la commande suivante :

```
CREATE DATABASE fédérée
USING CODESET ISO8859-15
TERRITORY BR
```

Jeux de codes, séquences de classement et support de langue nationale

Lorsque de la création d'une base de données fédérée, spécifiez le jeu de codes, le territoire et la séquence de classement. Ces informations contrôlent la langue dans laquelle les données sont stockées et la séquence dans laquelle les données de type caractères sont triées.

Un *jeu de codes* correspond à un jeu de modèles binaires uniques qui procède à un mappage vers les caractères d'une langue naturelle spécifique. Les produits IBM utilisent le terme *page de codes* comme synonyme pour "jeu de codes". Un *territoire* identifie un environnement local et spécifie des informations propres à une région pour le jeu de codes spécifié. Si aucune de ces options n'est spécifiée, la base de données applique la langue et la séquence de classement du client DB2 utilisés pour créer la base de données.

Avant de créer une base de données fédérée, déterminez le jeu de codes et les valeurs de territoire à spécifier. Une fois la base de données créée, ces valeurs ne peuvent pas être modifiées. Pour sélectionner un jeu de codes pour la base de données fédérée, évaluez celui spécifié par la source de données éloignée à laquelle la base de données fédérée pourra accéder. Sélectionnez un jeu de codes pour la base de données fédérée qui correspond à celui qu'utilisent les sources de données éloignées. Si la base de données fédérée accède à plusieurs sources de données, évaluez les jeux de codes spécifiés par toutes les sources de données éloignées. Si les sources de données utilisent différents jeux de codes ou des jeux de codes incompatibles, spécifiez Unicode comme jeu de codes pour le serveur fédéré.

Pour des sources de données différentes, un encapsuleur exécute les tâches suivantes lorsqu'il se connecte à une source de données pour la première fois :

1. Détermine la page de codes et le territoire de la base de données fédérée.
2. Mappe le jeu de codes et le territoire vers un environnement local client de la source de données, à condition que la source de données le prenne en charge.
3. Définit une variable d'environnement, appelle la source de données d'une interface de programme d'application pour informer la source de données de l'environnement local client ou se prépare à exécuter la conversion d'un jeu de codes.

La conversion d'une page de codes consiste à convertir des données de type caractères entre la page de codes de la base de données de la source de données et la page de codes de la base de données fédérée. Certaines sources de données procèdent à la conversion d'une page de codes. Pour certaines sources de données qui ne procèdent pas à la conversion de pages de codes, l'encapsuleur se charge de la conversion. Par exemple, si la base de données fédérée utilise la page de codes 819, territoire américain, l'environnement local client Oracle équivalent est `American_America.WE8ISO8859P1`. L'encapsuleur Oracle définit automatiquement la variable d'environnement `NLS_LANG` sur la valeur de l'environnement local client Oracle. Puis, lorsque les données sont envoyées de la base de données Oracle vers l'encapsuleur, la base de données Oracle convertit les données du jeu de codes `American_America.WE8ISO8859P1` vers la page de codes 819. Lorsque les données sont envoyées de l'encapsuleur vers la base de données Oracle, le serveur ou le client Oracle convertit les données de la page de codes 819 vers le jeu de codes qu'utilise la base de données Oracle.

La séquence de classement est associée à la langue que le serveur fédéré et le serveur de la source de données prennent en charge. Pour spécifier la séquence de

classement de la base de données fédérée, incluez l'option `COLLATE USING` dans la commande `CREATE DATABASE`. Si la base de données fédérée et la source de données utilisent une séquence de classement identique, définissez l'option de serveur `COLLATING_SEQUENCE` sur "Y" lorsque vous émettez une instruction `CREATE SERVER`. La séquence de classement spécifiée pour la base de données fédérée influe sur les requêtes procédant au tri ou à la comparaison de caractères. Par défaut, la base de données fédérée utilise une séquence de classement sensible à la casse. Toutefois, certaines sources de données utilisent par défaut une séquence de classement que ne tient pas compte de la casse. Une source de données peut également autoriser la personnalisation d'une séquence de classement ou proposer diverses options de configuration de la page de codes par défaut. Si la séquence de classement de la base de données fédérée et la source de données sont différentes, une requête risque de ne pas renvoyer les résultats souhaités. Par exemple, si une requête implique le tri de caractères, les résultats corrects sont renvoyés, mais pas dans l'ordre souhaité. Si une requête implique des comparaisons de caractères, vous risquez d'obtenir des résultats incorrects.

L'emplacement où les opérations basées sur les caractères sont exécutées influe sur les performances d'une requête. Lorsque les séquences de classement diffèrent, le serveur fédéré procède localement au tri et à des comparaisons de caractères afin que des ensembles de résultats triés de façon cohérente soient renvoyés. Pour obtenir des résultats optimaux, configurez la séquence de classement de la base de données fédérée par rapport à la même séquence utilisée par la source de données. Puis, chaque fois que possible, l'optimiseur de requêtes minimise les opérations basées sur les caractères par rapport à la source de données, afin que cette dernière, et non pas la base de données fédérée, exécute les opérations.

Configuration d'Unicode pour les systèmes fédérés

Vous pouvez exécuter des encapsuleurs relationnels et non relationnels et des fonctions définies par l'utilisateur sur une base de données dans la page de codes Unicode (UTF-8). Une base de données dans la page de codes Unicode fournit un environnement de serveur fédéré indépendant de la plate-forme.

Prise en charge du format Unicode pour les systèmes fédérés

Tous les encapsuleurs relationnels et non relationnels et les fonctions définies par l'utilisateur peuvent s'exécuter sur une base de données dans la page de codes Unicode (UTF-8).

La base de données dans la page de codes Unicode fournit des environnements de serveur fédéré indépendants de la plate-forme. La base de données peut manipuler les données qui sont stockées dans différentes pages de codes sur différentes sources de données.

Dans la figure 1, à la page 10, une société a des agences dans différents pays. Chaque agence stocke les données de ses clients avec ses propres bases de données dans ses propres pages de codes. La base de données Microsoft SQL Server stocke les données dans la page de codes A. La base de données Oracle stocke les données dans la page de codes B. La page de codes A et la page de codes B se trouvent dans des territoires différents. Pour intégrer les données des différents territoires, la société peut définir la page de codes de la base de données fédérée sur Unicode. La société peut alors joindre les tables pour voir le nombre total d'ordres d'achat, indépendamment du territoire.

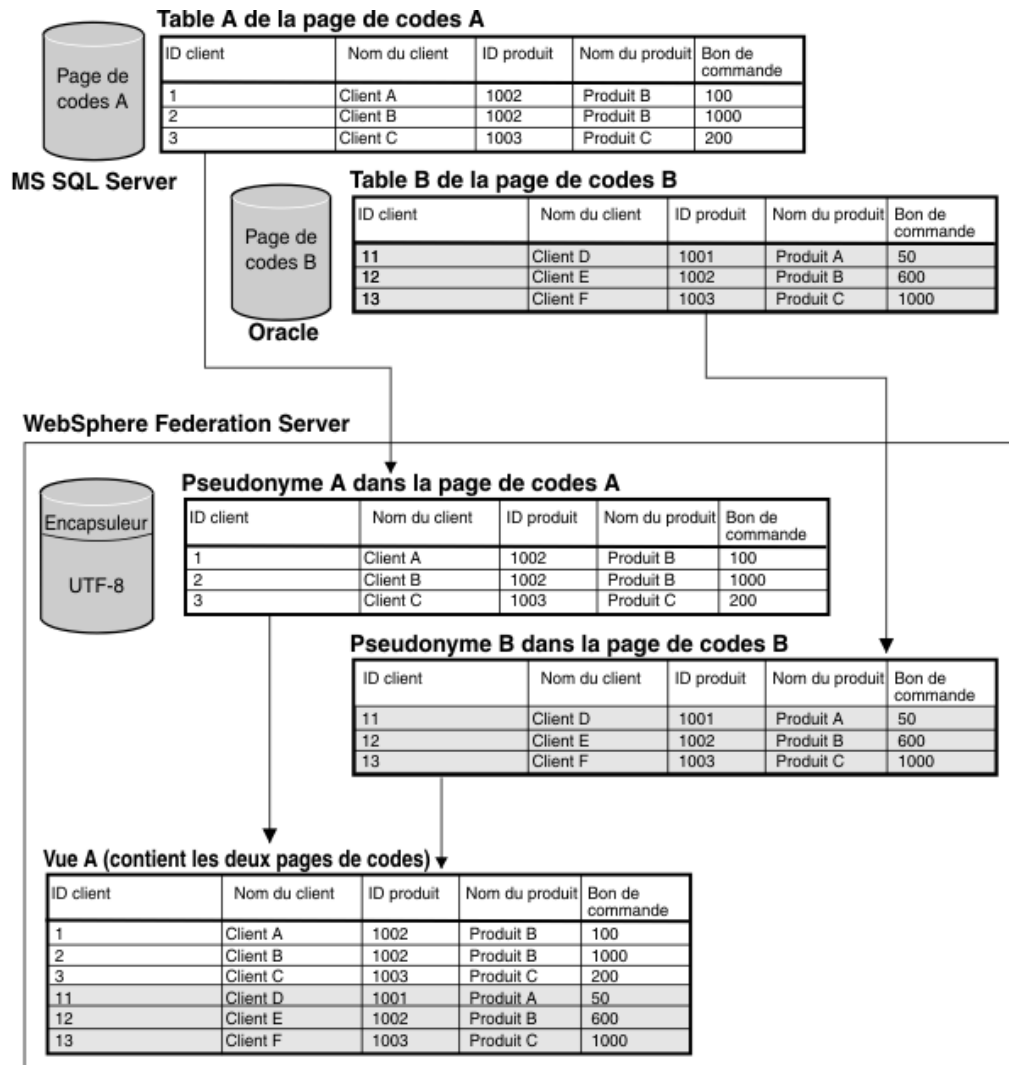


Figure 1. Exemple Unicode

Indication de la page de codes de client pour la prise en charge du format Unicode de Microsoft SQL Server et des sources de données ODBC

Pour assurer une conversion correcte de la page de codes pour Microsoft SQL Server et les sources de données ODBC, vous devez indiquer la page de codes client si la page de codes diffère de la page de codes de la base de données fédérée.

Procédure

Pour indiquer la page de codes client, émettez une instruction CREATE SERVER avec l'option CODEPAGE définie sur la valeur du numéro de la page de codes client. La page de codes client est la page de codes du client de la source de données.

Exemple : si la source de données est Microsoft SQL Server, que le serveur fédéré est sous Windows et que l'environnement local par défaut du système d'exploitation est défini sur japonais (codage SJIS), l'option du serveur CODEPAGE

doit être définie sur 943 (codage SJIS) ou sur 1202 (UTF-16LE). Pour indiquer la page de codes 1202 pour la source de données Microsoft SQL Server nommée FEDSERVERW, émettez l'instruction suivante :

```
CREATE SERVER FEDSERVERW TYPE MSSQLSERVER VERSION 2000 WRAPPER MSSQLODBC3
    OPTIONS(NODE 'SAMPLE', DBNAME 'TESTDB', CODEPAGE '1202');
```

Exemple : si la source de données est Microsoft SQL Server, que le serveur fédéré est exécuté sous UNIX et que le paramètre IANAAppCodePage du client de connexion DataDirect est 6 (codage SJIS), l'option du serveur CODEPAGE doit être définie sur 943 (codage SJIS) ou 1208 (UTF-8). Pour indiquer la page de codes 1208 pour la source de données Microsoft SQL Server nommée FEDSERVERU, émettez l'instruction suivante :

```
CREATE SERVER FEDSERVERU TYPE MSSQLSERVER VERSION 2000 WRAPPER MSSQLODBC3
    OPTIONS(NODE 'SAMPLE', DBNAME 'TESTDB', CODEPAGE '1208');
```

Pages de codes Unicode prises en charge pour l'option CODEPAGE de l'encapsuleur MSSQL et ODBC

Les valeurs de page de codes valides sont celles prises en charge par les bases de données DB2 pour Linux®, UNIX et Windows, plus celles indiquées dans la table suivante.

Tableau 4. Pages de codes Unicode prises en charge pour l'option CODEPAGE de l'encapsuleur MSSQL et ODBC

Valeur de l'option CODEPAGE	Description
1200	Codepage1200 - UCS-2 (big endian)
1202	Codepage1202 - UCS-2 (little endian)
1208	Codepage1208 - UTF-8
1232	Codepage1232 - UTF-32 (big endian)
1234	Codepage1234 - UTF-32 (little endian)

Indication de la page de codes de fichier pour la prise en charge du format Unicode de sources de données de fichiers structurés sous forme de tables

Pour assurer une conversion correcte de la page de codes pour des sources de données de fichiers structurés sous forme de tables, vous devez indiquer la page de codes de fichier si la page de codes diffère de la page de codes de la base de données fédérée.

Restrictions

Vous ne pouvez utiliser l'option CODEPAGE que dans une base de données fédérée Unicode.

A propos de cette tâche

Les valeurs valides sont celles prises en charge par les bases de données DB2 pour Linux, UNIX et Windows. La valeur par défaut est la page de codes de la base de données fédérée.

Procédure

Pour indiquer la page de codes de fichiers structurés sous forme de tables, émettez l'instruction CREATE NICKNAME avec le jeu d'options CODEPAGE du numéro de la page de codes des données dans les fichiers structurés sous forme de tables.

Exemple : la page de codes des données d'un fichier nommé DRUGDATA1.TXT est 943. Pour indiquer 943 comme page de codes de fichiers structurés sous forme de tables, émettez l'instruction CREATE NICKNAME suivante :

```
CREATE NICKNAME DRUGDATA1(Dcode Integer NOT NULL, Drug CHAR(20),
    Manufactuer CHAR(20))
FOR SERVER biochem_lab
OPTIONS(FILE_PATH '/usr/pat/DRUGDATA1.TXT',CODEPAGE '943',
COLUMN_DELIMITER '.',
SORTED 'Y', KEY_COLUMN 'DCODE', VALIDATE_DATA_FILE 'Y');
```

Indication de la page de codes de fichier pour la prise en charge du format Unicode de sources de données de fichiers structurés sous forme de tables - exemple

L'exemple suivant vous montre comment indiquer la page de codes de fichiers structurés sous forme de tables.

La page de codes des données dans un fichier nommé DRUGDATA1.TXT est 943. Pour indiquer 943 comme page de codes de fichiers structurés sous forme de tables, émettez l'instruction CREATE NICKNAME suivante :

```
CREATE NICKNAME DRUGDATA1(Dcode Integer NOT NULL, Drug CHAR(20),
    Manufactuer CHAR(20))
FOR SERVER biochem_lab
OPTIONS(FILE_PATH '/usr/pat/DRUGDATA1.TXT',CODEPAGE '943',
COLUMN_DELIMITER '.',
SORTED 'Y', KEY_COLUMN 'DCODE', VALIDATE_DATA_FILE 'Y');
```

Erreurs lorsque les tailles de point des codes distant et fédéré sont différentes

Lorsque la taille de point du code diffère entre la base de données fédérée et la source de données distante, des données tronquées peuvent être renvoyées ou des échecs d'insertion ou de mise à jour peuvent se produire.

Lorsque vous sélectionnez des données dans la source de données distante, ces données sont tronquées si la conversion de la chaîne de caractères produit un plus grand nombre d'octets que la taille de la colonne de pseudonymes. Si les données tronquées se terminent par un caractère sans soutien, des blancs remplissent les octets restants. Vous pouvez aussi insérer ou mettre à jour des données qui sont plus grandes que la taille de la colonne de pseudonymes si la taille des données converties est inférieure ou égale à la taille de la colonne distante.

Si la base de données fédérée a une taille de point du code inférieure à la source de données distante, l'insertion ou la mise à jour des données peut échouer. Les insertions ou les mises à jour échouent si la conversion de la chaîne de caractères produit un plus grand nombre d'octets que la taille de la colonne de la source de données distante.

Ajustez la taille de la colonne de pseudonymes ou la taille de la colonne de table distante pour empêcher la troncature des données ou une erreur de troncature.

Configuration du serveur fédéré afin d'accéder aux sources de données

Après avoir installé et configuré le serveur fédéré et créé une base de données fédérée, planifiez puis configurez un accès aux sources de données.

Le processus de configuration d'un accès aux sources de données est identique, peu importe la source de données. Les paramètres spécifiques appliqués à mesure

que vous terminez une tâche de configuration pour chaque source de données sont les seuls éléments variables du processus. Le processus décrit ici est générique ; pour plus d'informations sur la configuration d'une source de données spécifique, reportez-vous aux informations de configuration détaillées de la source de données.

Contrôle de la dénomination et de la spécification des objets

Lors du processus de configuration, utilisez des instructions SQL DB2 pour enregistrer des objets. Avant d'exécuter toute tâche de configuration, veillez à avoir consulté les règles de dénomination fédérée et compris dans quelle mesure l'utilisation des guillemets dans les instructions SQL influe sur le respect de la casse des objets spécifiés.

Enregistrement de l'encapsuleur

Après avoir utilisé le programme d'installation de fédération pour installer la prise en charge d'une source de données, vous devez enregistrer l'encapsuleur correspondant. Un encapsuleur correspond à un ensemble de fichiers de bibliothèque que le serveur fédéré utilise pour communiquer avec la source de données et pour y récupérer des données. Pour chaque type de source de données auquel vous voulez accéder, vous devez enregistrer un encapsuleur. Par exemple, pour accéder à une table dans DB2 for Linux, UNIX et Windows, une table dans DB2 for iSeries et une table dans Teradata, vous devez enregistrer deux encapsuleurs : l'encapsuleur DRDA pour les sources de données DB2 et l'encapsuleur Teradata pour les sources de données Teradata.

Dans le cadre de votre planification, définissez si vous souhaitez utiliser ou pas le nom par défaut de l'encapsuleur ou si vous souhaitez lui attribuer un nom autre, puis contrôlez ses options qui sont disponibles pour chaque source de données que vous configurez. Chacune d'entre elle dispose d'une ou de plusieurs options d'encapsuleur requises devant être définies.

Enregistrement des définitions de serveur

Avant de pouvoir accéder aux objets d'une source de données spécifique, vous devez enregistrer une ou plusieurs définitions de serveur. Pour une source de données relationnelle, une définition de serveur représente une base de données éloignée, une partition de base de données ou un noeud. Pour une source de données non-relationnelle, une définition de serveur procède souvent à un mappage vers d'autres types d'objets de données externes. Chaque source de données exige que des paramètres facultatifs soient spécifiés lors de l'enregistrement de la définition de serveur.

Dans le cadre de votre planification, contrôlez les options de serveur qui sont disponibles pour la source de données spécifique que vous configurez. Chaque source de données dispose d'une ou de plusieurs options de serveur obligatoires devant être définie.

Enregistrement des mappages utilisateur

Si une source de données éloignée exige une authentification d'utilisateur et si l'ID utilisateur et le mot de passe distants d'un utilisateur différent de ceux utilisés pour se connecter à la base de données fédérée, définissez un mappage utilisateur. Il s'agit d'une association entre l'ID autorisation d'un serveur de fédération et un

ID utilisateur et un mot de passe d'une source de données. Par défaut, les mappages utilisateur sont stockés dans le catalogue du serveur fédéré.

Dans le cadre de votre planification, indiquez si vous souhaitez stocker des informations de mappage utilisateur dans un référentiel externe, comme un serveur LDAP ou un fichier. Pour utiliser un référentiel externe, vous devez créer un module d'extension qui fournit une interface au serveur fédéré dans le référentiel.

Mise à jour des statistiques de la source de données

Pour chaque source de données relationnelle à laquelle vous avez l'intention d'accéder, utilisez une commande équivalente à la commande DB2 RUNSTATS afin de mettre à jour les statistiques de la source de données éloignée. Puis, lors de la création de pseudonymes, les informations statistiques les plus récentes sont ajoutées au catalogue système dans la base de données fédérée. Ensuite, lorsque vous exécutez une requête dans la source de données, l'optimiseur de requêtes utilise ces informations pour déterminer la méthode la plus efficace permettant d'exécuter cette requête.

Une fois les pseudonymes créés, les statistiques de la source de données peuvent changer. Lorsque les statistiques d'une source de données relationnelles changent, exécutez la procédure mémorisée SYSPROC.NNSTAT afin de mettre à jour les informations statistiques dans le catalogue système. Lorsque les statistiques d'une source de données non relationnelles changent, utilisez l'outil disponible dans la source de données non relationnelles ou effectuez une mise à jour manuelle des statistiques dans les vues SYSTAT du catalogue.

Enregistrement de pseudonymes

Créez un pseudonyme pour chaque objet de source de données relationnelles auquel vous voulez accéder. Pour certaines sources de données non relationnelles, définissez une liste détaillée des colonnes d'entrée et de sortie lors de l'enregistrement du pseudonyme. Chaque colonne spécifiée est mappée vers une zone, une colonne ou un élément spécifique dans l'objet de source de données.

Dans le cadre de votre planification, contrôlez le pseudonyme et les options de colonne disponibles pour la source de données que vous configurez. Certaines sources de données exigent que vous définissiez un pseudonyme et les options de colonne.

Exécution de tâches de configuration supplémentaires

En fonction de la méthode d'exploitation de la source de données utilisée, vous pouvez exécuter les tâches de configuration supplémentaires suivantes :

Création de spécifications d'index

Vous pouvez définir une spécification d'index pour les objets dépourvus d'index. Par exemple, créez une spécification d'index lorsqu'une table fait l'acquisition d'un nouvel index ou si un objet de source de données, tel qu'une vue, est dépourvu d'index.

Définition de mappages de types de données alternatives

Dans le système fédéré, il existe des mappages par défaut entre les types de données de la source de données et les types de données de la base de données fédérée. Pour les sources de données relationnelles, vous pouvez définir des mappages de types de données alternatives. Par exemple, vous

pouvez modifier un mappage type pour tous les objets de source de données figurant sur un serveur spécifique ou modifier un mappage type pour un objet de source de données, un type de source de données ou un objet et un type de source de données spécifiques.

Définition de mappages de fonctions alternatifs

Dans le système fédéré, il existe des mappages de fonctions par défaut entre les fonctions de source de données intégrées et les fonctions de base de données fédérées intégrées. Pour les sources de données relationnelles, vous pouvez définir des mappages de fonctions alternatifs. Par exemple, vous pouvez définir un mappage de fonction alternatif lorsque vous souhaitez utiliser une nouvelle fonction intégrée ou une fonction définie par un utilisateur qui est disponible à la source de données, mais pour laquelle la base de données fédérée ne dispose pas d'une fonction mappée.

Sensibilité à la casse et utilisation correcte des guillemets

Lorsque vous spécifiez des valeurs d'option et des objets dans les instructions DB2 SQL, vous devez savoir à quel moment les guillemets sont obligatoires, quel type de guillemet doit être utilisé et dans quelle mesure ils influent sur la sensibilité à la casse.

Le nom que vous donnez aux objets à leur création influe sur la casse des caractères du nom de l'objet et détermine la façon dont vous attribuez des noms aux objets et des valeurs d'option dans les commandes. Par exemple, si, lors de la création d'un pseudonyme, vous ne saisissez pas le nom entre des guillemets doubles, le catalogue système mémorise le pseudonyme en caractères majuscules, peu importe la casse de caractères utilisée lors de l'attribution du nom à un objet. Si vous utilisez des guillemets doubles lors de la création du pseudonyme, le catalogue mémorise les caractères du nom de l'objet dans la casse précisément spécifiée. Puis, lorsque vous utilisez le nom de l'objet en tant que valeur d'option, vous devez spécifier cette casse avec précision. Par exemple, l'option de colonne FOREIGN_KEY que les encapsuleurs Script, Web services et XML prennent en charge exige que vous spécifiez le pseudonyme pour la colonne de clé étrangère en tant que valeur d'option. Lors de la saisie de la valeur d'option, vous devez utiliser une casse identique à celle que le catalogue du serveur fédéré utilise pour mémoriser le pseudonyme.

La table suivante décrit l'utilisation correcte de la casse et des guillemets lorsque vous spécifiez des valeurs d'option et des objets dans les instructions DB2 SQL.

Tableau 5. Utilisation correcte de la casse et des guillemets

Identificateur	Utilisation de la casse et des guillemets	Exemples
Valeur d'option	Utilisez la casse requise par la valeur d'option et placez la valeur d'option entre des guillemets simples.	<p>Cette instruction crée une table de source de données appelée <code>remote_schema.remote_table</code> (en minuscules) :</p> <pre>CREATE TABLE newton.my_nick (c1 int) OPTIONS (remote_server 'MY_SERVER' remote_schema 'remote_schema', remote_tabname 'remote_table');</pre> <p>Cette instruction crée une table de source de données appelée <code>REMOTE_SCHEMA.REMOTE_TABLE</code> (en majuscules) :</p> <pre>CREATE TABLE newton.my_nick (c1 int) OPTIONS (remote_server 'MY_SERVER' remote_schema 'REMOTE_SCHEMA', remote_tabname 'REMOTE_TABLE');</pre>
Objet contenant uniquement des caractères en minuscules	Utilisez tous les caractères en minuscules et placez l'identificateur entre des guillemets doubles.	<p>Cette instruction crée un pseudonyme sur une table de source de données appelée <code>infx_user.remote_table</code> (en minuscules) :</p> <pre>CREATE NICKNAME my_nick FOR infx_server. "infx_user"."remote_table";</pre> <p>Remarque : Certaines sources de données, comme Informix et Teradata, utilisent par défaut des noms en minuscules.</p>
Objet contenant uniquement des caractères en majuscules, des numéros et des caractères de soulignement (<code>_</code>)	<p>Choisissez l'une des deux méthodes suivantes :</p> <ul style="list-style-type: none"> Utilisez tous les caractères en majuscules et placez l'identificateur entre des guillemets doubles. Utilisez la casse de votre choix et ne placez pas l'identificateur entre des guillemets doubles. 	<p>Chacune de ces instructions crée le pseudonyme <code>MY_NICK</code> (en majuscules) :</p> <pre>CREATE NICKNAME my_nick FOR infx_server. "infx_user"."remote_table"; CREATE NICKNAME "MY_NICK" FOR infx_server. "infx_user"."remote_table";</pre>

Pour les ID autorisation et les mots de passe de la source de données, vous pouvez également utiliser les options de serveur `FOLD_ID` et `FOLD_PW` pour convertir l'ID et le mot de passe dans la casse correcte.

A partir d'une invite de commande du système d'exploitation UNIX

Si vous placez une valeur sensible à la casse entre guillemets à l'apparition de l'invite de commande UNIX sur le serveur fédéré, vous devez vous assurer que l'analyse syntaxique des guillemets a été correctement effectuée :

Instructions SQL contenant des guillemets doubles, mais aucun guillemet simple

Si l'instruction SQL contient des guillemets doubles, mais aucun guillemet simple, placez l'instruction entière entre des guillemets simples.

Par exemple, pour émettre l'instruction SQL suivante :

```
CREATE NICKNAME mon_pseudonyme FOR mon_serveur."propriétaire"."ma_table"
```

Saisissez le texte suivant à l'apparition de l'invite de commande UNIX suivante :

```
db2 'CREATE NICKNAME mon_pseudonyme FOR mon_serveur."propriétaire"."ma_table"'
```

Instructions SQL contenant des guillemets simples, mais aucun guillemet double

Si l'instruction SQL contient des guillemets simples, mais aucun guillemet double, placez l'instruction entière entre des guillemets doubles.

Par exemple, pour émettre l'instruction SQL suivante :

```
CREATE USER MAPPING FOR UTILISATEUR SERVER mon_serveur  
  OPTIONS(REMOTE_AUTHID 'mon_id', REMOTE_PASSWORD 'mon_mot_de_passe')
```

Saisissez le texte suivant à l'apparition de l'invite de commande UNIX suivante :

```
db2 "CREATE USER MAPPING FOR UTILISATEUR SERVER mon_serveur  
  OPTIONS(REMOTE_AUTHID 'mon_id', REMOTE_PASSWORD 'mon_mot_de_passe')"
```

Instructions SQL contenant des guillemets simples et doubles

Si l'instruction SQL contient des guillemets simples et doubles, procédez comme suit :

- Placez l'instruction entière entre des guillemets doubles
- Faites précéder chaque guillemet double par une barre oblique inverse dans l'instruction.

Par exemple, pour émettre l'instruction SQL suivante :

```
CREATE USER MAPPING FOR "id_local" SERVER mon_serveur  
  OPTIONS(REMOTE_AUTHID 'mon_id', REMOTE_PASSWORD 'mon_mot_de_passe')
```

Saisissez le texte suivant à l'apparition de l'invite de commande UNIX suivante :

```
db2 "CREATE USER MAPPING FOR \"id_local"\ SERVER mon_serveur  
  OPTIONS(REMOTE_AUTHID 'mon_id', REMOTE_PASSWORD 'mon_mot_de_passe')"
```

Les exemples mentionnés ci-dessus impliquent la saisie des instructions SQL dans l'invite de commande UNIX et transfèrent l'instruction vers la commande DB2 sans utiliser l'option `-f`. Pour utiliser la commande DB2 avec l'option `-f` afin de saisir des instructions SQL à partir d'un fichier, saisissez les instructions comme illustré dans la première occurrence de chaque exemple.

A partir d'une invite de commande du système d'exploitation Windows

Pour conserver les valeurs sensibles à la casse lorsque vous saisissez des commandes à partir d'une invite de commande Microsoft Windows sur le serveur

fédéré, faites précéder chaque guillemet double par une barre oblique inverse. Par exemple, vous souhaitez créer le pseudonyme *pseudo1* pour la table *salaires_hebdomadaire* de Microsoft SQL Server. La table se trouve dans la base de données *NORBASE*. Le nom du schéma local est *mon_schéma*.

À l'apparition de l'invite Windows sur le serveur fédéré, tapez :

```
db2 CREATE NICKNAME pseudo1
    FOR NORBASE.\"mon_schéma\".\"salaires_hebdomadaire\"
```

A partir de la ligne de commande DB2 ou d'une application

Lorsque vous spécifiez une valeur à partir de la ligne de commande DB2 ou d'une application, vous pouvez conserver les valeurs sensibles à la casse en plaçant les valeurs entre des guillemets correspondants.

Par exemple, vous souhaitez créer un mappage utilisateur pour l'ID utilisateur *id_local*. L'ID utilisateur distant est *mon_id* et le mot de passe distant est *mon_mot_de_passe*. Vous voulez que ces trois valeurs restent en minuscules. À l'apparition de l'invite de commande DB2, tapez :

```
CREATE USER MAPPING FOR "id_local" SERVER mon_serveur
    OPTIONS(REMOTE_AUTHID 'mon_id', REMOTE_PASSWORD 'mon_mot_de_passe')
```

Configuration de plusieurs serveurs fédérés pour accéder aux sources de données

Un système fédéré peut se composer de plusieurs serveurs fédérés. Au lieu de configurer chaque serveur fédéré séparément, gagnez du temps en utilisant le Centre de contrôle pour configurer les serveurs fédérés.

Avant de commencer

- La fédération doit être installée sur un serveur qui va agir comme serveur fédéré.
- Une base de données fédérée doit être disponible sur le serveur fédéré.

Lorsque vous configurez le premier serveur, la fenêtre **Sortie d'action** capture les instructions DDL émises lors de la création d'objets fédérés. Vous pouvez réutiliser ou modifier ces instructions et les appliquer afin de configurer rapidement des serveurs fédérés supplémentaires.

La fenêtre **Sortie d'action** reste active pour la session en cours. Si vous fermez la fenêtre **Sortie d'action**, les instructions DDL de la session en cours continuent à être stockées dans la fenêtre **Sortie d'action**. Toutefois, si vous fermez le Centre de contrôle DB2, toutes les instructions DDL de la session en cours sont supprimées de la fenêtre **Sortie d'action**.

Procédure

Pour configurer plusieurs serveurs fédérés afin d'accéder aux sources de données, procédez comme suit :

1. Utilisez le Centre de contrôle afin de configurer le premier serveur fédéré pour les sources de données auxquelles vous voulez accéder. Chaque instruction DDL est ainsi capturée.
2. Affichez la page **Sortie d'action** dans la fenêtre **Sortie d'action**. Si vous avez fermé la fenêtre **Sortie d'action**, cliquez avec le bouton droit de la souris sur le

dossier **Objets de la base de données fédérée** et cliquez sur **Afficher les actions** pour ouvrir la fenêtre **Sortie d'action**.

3. Supprimez toutes les instructions DDL que vous ne souhaitez pas utiliser sur les autres serveurs fédérés. Pour cela, cliquez avec le bouton droit de la souris sur l'instruction et cliquez sur **Supprimer**. Par exemple, vous pouvez supprimer les instructions affichant Echec dans la colonne Etat de la page **Sortie d'action**.
4. Copiez les instructions à utiliser dans les autres serveurs fédérés sur la page **Editeur de commande**.
 - a. Sélectionnez les instructions à copier. Pour sélectionner plusieurs instructions, appuyez sur la touche Ctrl.
 - b. Cliquez avec le bouton droit de la souris sur les instructions sélectionnées et cliquez sur **Copier vers l'éditeur de commande**. La page **Editeur de commande** s'ouvre.
5. Dans la page **Editeur de commande**, modifiez les instructions DDL que vous voulez utiliser sur les autres serveurs fédérés. Par exemple, vous pouvez modifier une instruction spécifiant un schéma local.

Vous devez modifier les mappages utilisateur afin de spécifier les mots de passe adéquats pour le serveur fédéré. Lorsque le DDL des instructions CREATE USER MAPPING est capturé dans la fenêtre **Sortie d'action**, les mots de passe sont masqués par des astérisques. Vous devez remplacer les astérisques par les mots de passe adéquats.
6. Après avoir modifié les instructions DDL qui ont été générées dans la fenêtre **Sortie d'action**, exécutez les instructions DDL sur le serveur fédéré suivant.

Chapitre 2. Configuration des sources de données

Vous pouvez configurer le serveur fédéré pour accéder aux sources de données relationnelles et non relationnelles.

Configuration de l'accès aux sources de données BioRS

Vous pouvez intégrer les données qui sont dans les banques de données BioRS avec des informations figurant dans d'autres sources à l'aide d'un système fédéré.

Procédure

Pour configurer un serveur fédéré pour accéder aux sources de données BioRS, vous devez fournir au serveur fédéré des informations sur les sources de données et les objets auxquels vous voulez accéder. Après avoir configuré le serveur fédéré, vous pouvez créer des requêtes et utiliser les fonctions personnalisées pour accéder aux sources de données BioRS.

encapsuleur BioRS

BioRS est un système de requêtes et d'extraction que vous pouvez utiliser pour rechercher des informations dans plusieurs sources de données. L'encapsuleur BioRS utilise les mêmes API que l'interface Web BioRS pour exécuter les requêtes.

BioRS est un système de requêtes et d'extraction mis au point par Biomax Informatics. Vous pouvez utiliser BioRS pour rechercher des informations dans plusieurs sources de données, notamment dans des fichiers à plat et des bases de données relationnelles. Vous téléchargez généralement les données publiques, telles que SwissProt et GenBank, sous forme de fichiers à plat dans votre système BioRS. BioRS peut intégrer des sources de données publiques et des sources de données privées (par exemple des bases de données privées qui sont gérées par votre organisation) dans un environnement commun.

Une fois qu'une source de données est intégrée dans le système BioRS, elle est appelée *banque de données*. Les éléments contenus dans chaque entrée de banque de données sont appelés collectivement un *schéma*. Les éléments d'une banque de données qui sont indexés peuvent être utilisés dans les fonctions BIOR.S.CONTAINS, BIOR.S.CONTAINS_GE et BIOR.S.CONTAINS_LE. Les fonctions BioRS sont indiquées dans la clause WHERE de l'instruction SELECT. Les éléments qui ne sont pas indexés peuvent être référencés dans la liste de sélection et dans d'autres prédicats de la clause WHERE. Les éléments qui ne sont pas indexés sont traités par le serveur fédéré.

Vous pouvez établir des relations entre les entrées des banques de données, de façon à pouvoir relier les banques de données à l'intérieur du système BioRS.

Les banques de données BioRS peuvent avoir une relation parent-enfant (les banques de données peuvent être imbriquées). Dans une telle relation, la banque de données enfant contient un élément du type de données Référence appelé PARENT. L'élément PARENT renvoie à l'élément `_ID_` de la banque de données parent. Hormis la présence de cet élément PARENT prédéfini, les banques de données imbriquées contiennent les mêmes données que les banques de données non imbriquées.

BioRS fournit une interface Web qui permet aux utilisateurs d'appliquer des requêtes aux données des banques de données BioRS. L'encapsuleur BioRS utilise les mêmes interfaces de programme d'application (API) que l'interface Web BioRS pour exécuter les requêtes.

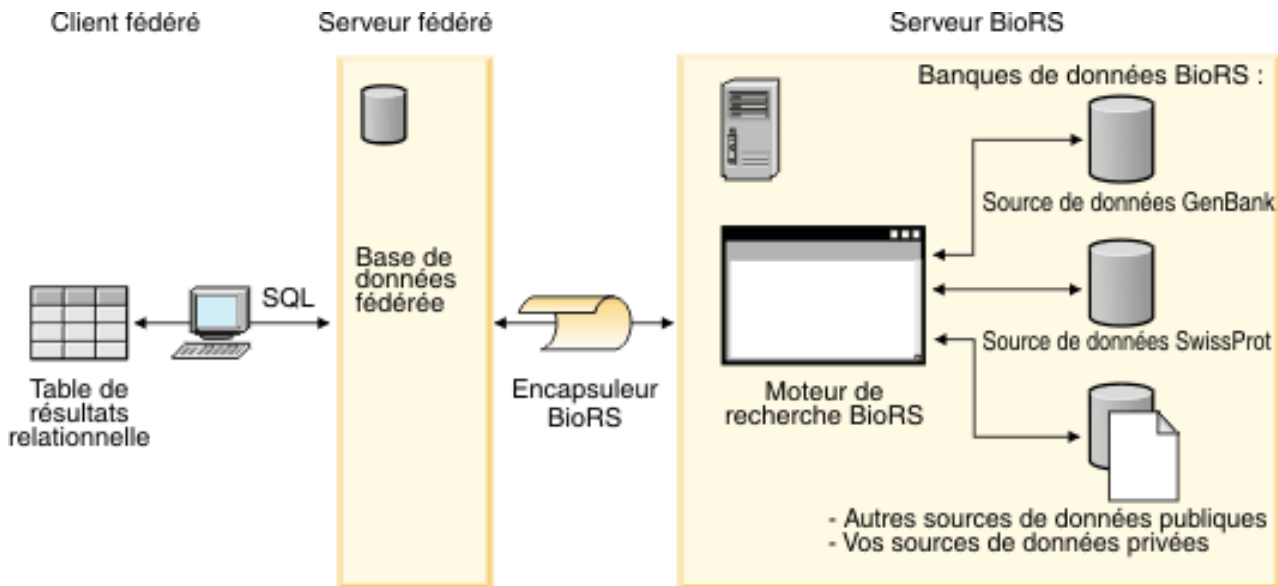


Figure 2. Fonctionnement de l'encapsuleur BioRS

A partir du client, les utilisateurs ou applications soumettent une requête à l'aide d'instructions SQL. Ensuite, la requête est envoyée à votre système fédéré sur lequel l'encapsuleur BioRS est installé. Suivant la façon dont la requête est construite, aussi bien le serveur fédéré que votre serveur BioRS sont susceptibles d'être utilisés pour traiter la requête. Le serveur BioRS peut être sur un ordinateur autre que le système fédéré. Les informations d'authentification doivent être fournies par le système fédéré au serveur BioRS pour chaque requête. Ces informations peuvent être une combinaison ID utilisateur et mot de passe ou bien une indication non authentifiée (habituellement un compte d'invité).

Pour des informations détaillées sur BioRS, voir le site Web de Biomax à l'adresse : <http://www.biomax.com>.

Ajout de sources de données BioRS à un serveur fédéré

Pour configurer un serveur fédéré pour accéder à des sources de données BioRS, vous devez fournir au serveur fédéré des informations sur les sources de données et les objets auxquels vous voulez accéder.

Avant de commencer

- La fédération doit être installée sur un serveur qui fera office de serveur fédéré.
- Il doit y avoir une base de données sur le serveur fédéré.

A propos de cette tâche

Vous pouvez configurer un serveur fédéré pour accéder aux données qui sont stockées dans des sources de données BioRS à l'aide du Centre de contrôle ou en émettant des instructions SQL sur la ligne de commande. Le Centre de contrôle comporte un assistant pour vous guider à travers les étapes nécessaires pour configurer les objets fédérés obligatoires.

Procédure

Pour ajouter des sources de données BioRS à un serveur fédéré :

1. Enregistrez les fonctions personnalisées pour l'encapsuleur BioRS.
2. Enregistrez l'encapsuleur BioRS.
3. Enregistrez les définitions de serveur BioRS.
4. Facultatif : créez les mappages utilisateur BioRS.
5. Enregistrez des pseudonymes pour les banques de données BioRS.

Enregistrement des fonctions personnalisées pour l'encapsuleur BioRS

Vous devez enregistrer les fonctions personnalisées BioRS avant d'enregistrer l'encapsuleur BioRS. Les fonctions BioRS personnalisées sont utilisées avec l'encapsuleur BioRS pour envoyer des prédicats au moteur de requête BioRS.

A propos de cette tâche

Vous devez enregistrer toutes les fonctions personnalisées sur chaque instance de base de données fédérée sur laquelle l'encapsuleur BioRS est installé.

Toutes les fonctions personnalisées pour l'encapsuleur BioRS doivent être enregistrées sous le nom de schéma biors.

Procédure

Pour enregistrer les fonctions personnalisées BioRS :

Pour chacune des fonctions personnalisées BioRS, émettez l'instruction CREATE FUNCTION. Vous devez indiquer deux types de données dans chaque fonction personnalisée. Le premier type de données que vous indiquez est la colonne indexée. Le deuxième type de données que vous indiquez est le terme recherché. Vous devez inclure les mots clé AS TEMPLATE, DETERMINISTIC et NO EXTERNAL ACTION dans l'instruction CREATE FUNCTION.

L'exemple suivant montre la syntaxe de la fonction BIOR.SCONTAINS :

```
CREATE FUNCTION biors.contains (type_données_colonne, type_données_terme_recherché)
  RETURNS INTEGER AS TEMPLATE
  DETERMINISTIC NO EXTERNAL ACTION;
```

Conseil : Utilisez le fichier modèle create_function_mappings.ddl pour enregistrer les fonctions personnalisées BioRS. Le fichier modèle se trouve dans le répertoire sqllib/samples/lifesci/biors sur le serveur fédéré. Le fichier modèle contient les instructions CREATE FUNCTION pour chacune des combinaisons de types de données possibles. Pour enregistrer les fonctions personnalisées, éditez le fichier create_function_mappings.ddl pour indiquer les types de données des colonnes d'index et des termes recherchés pour chaque fonction personnalisée. Vous devez ensuite exécuter le fichier create_function_mappings.ddl sur chaque instance de base de données fédérée sur laquelle l'encapsuleur BioRS est installé.

Fonctions personnalisées pour l'encapsuleur BioRS :

Descriptions et exemples de fonctions personnalisées utilisées avec l'encapsuleur BioRS.

L'environnement fédéré utilise deux moteurs de requête. Pour l'encapsuleur BioRS, ces moteurs de requête sont le moteur de requête de la base de données fédérée et le moteur de requête BioRS. Vous pouvez indiquer que les prédicats sont envoyés au moteur BioRS à l'aide des fonctions personnalisées BioRS.

Les fonctions personnalisées pour l'encapsuleur BioRS sont :

- BIRS.CONTAINS
- BIRS.CONTAINS_LE
- BIRS.CONTAINS_GE
- BIRS.SEARCH_TERM

Vous utilisez l'instruction CREATE FUNCTION pour enregistrer les fonctions personnalisées BioRS.

La table suivante présente les quatre fonctions personnalisées BioRS avec des exemples de types de données valides que vous pouvez indiquer lorsque vous enregistrez les fonctions.

Tableau 6. Fonctions personnalisées pour l'encapsuleur BioRS

Fonction	Description
BIRS.CONTAINS (VARCHAR(), VARCHAR()) BIRS.CONTAINS (VARCHAR(), CHAR()) BIRS.CONTAINS (VARCHAR(), DATE) BIRS.CONTAINS (VARCHAR(), TIMESTAMP)	Recherche dans une colonne indexée les valeurs qui sont identiques (d'après la sémantique de la requête BioRS) à la valeur que vous indiquez. Le premier argument doit être une référence à la colonne indexée et le deuxième argument est la valeur de recherche que vous indiquez.
BIRS.CONTAINS_LE (VARCHAR(), VARCHAR()) BIRS.CONTAINS_LE (VARCHAR(), SMALLINT) BIRS.CONTAINS_LE (VARCHAR(), BIGINT) BIRS.CONTAINS_LE (VARCHAR(), DECIMAL) BIRS.CONTAINS_LE (VARCHAR(), DOUBLE) BIRS.CONTAINS_LE (VARCHAR(), REAL)	Recherche dans une colonne indexée les valeurs qui sont inférieures ou égales (d'après la sémantique de la requête BioRS) à la valeur que vous indiquez. Le premier argument doit être une référence à la colonne indexée et le deuxième argument est la valeur de recherche que vous indiquez.
BIRS.CONTAINS_GE (CHAR(), CHAR()) BIRS.CONTAINS_GE (CHAR(), DATE) BIRS.CONTAINS_GE (CHAR(), TIMESTAMP) BIRS.CONTAINS_GE (CHAR(), INTEGER) BIRS.CONTAINS_GE (CHAR(), SMALLINT) BIRS.CONTAINS_GE (CLOB(), DATE)	Recherche dans une colonne indexée les valeurs qui sont supérieures ou égales (d'après la sémantique de la requête BioRS) à la valeur que vous indiquez. Le premier argument doit être une référence à la colonne indexée et le deuxième argument est la valeur de recherche que vous indiquez.
BIRS.SEARCH_TERM (VARCHAR(), VARCHAR()) BIRS.SEARCH_TERM (VARCHAR(), CHAR()) BIRS.SEARCH_TERM (CHAR(), VARCHAR()) BIRS.SEARCH_TERM (CHAR(), CHAR())	Transmet une expression de requête BioRS au moteur de recherche BioRS.

Enregistrement de l'encapsuleur BioRS

Vous devez enregistrer un encapsuleur pour accéder aux sources de données BioRS. Les serveurs fédérés utilisent des encapsuleurs pour dialoguer avec les sources de données et en extraire des données. Les encapsuleurs sont mis en oeuvre comme un ensemble de fichiers de bibliothèque.

Si vous utilisez un serveur proxy pour accéder aux fichiers BioRS, vous devez indiquer les informations du serveur proxy comme options lorsque vous enregistrez l'encapsuleur. Par défaut, les options de l'encapsuleur sont utilisées lorsque vous interrogez un fichier BioRS. Cependant, si vous indiquez aussi des options du serveur proxy lorsque vous enregistrez une définition de serveur, les options de serveur prévalent.

Procédure

Pour enregistrer l'encapsuleur BioRS :

Choisissez la méthode que vous voulez utiliser pour enregistrer l'encapsuleur BioRS :

Méthode	Procédure.
Utilisation du Centre de contrôle	Démarrez l'assistant des objets fédérés. Cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée et cliquez sur Créer des objets fédérés .
Depuis la ligne de commande	<p>Émettez l'instruction CREATE WRAPPER.</p> <pre>CREATE WRAPPER <i>nom_encapsuleur</i> LIBRARY <i>nom_bibliothèque</i>;</pre> <p>Si vous utilisez un serveur proxy pour accéder aux documents BioRS, l'instruction que vous émettez est :</p> <pre>CREATE WRAPPER <i>nom_encapsuleur</i> LIBRARY <i>nom_bibliothèque</i> OPTIONS (PROXY_TYPE '<i>type</i>', PROXY_SERVER_NAME '<i>server_name</i>', PROXY_SERVER_PORT '<i>numéro_port</i>');</pre>

Vous devez indiquer le paramètre LIBRARY dans l'instruction CREATE WRAPPER. Le nom du fichier de bibliothèque de l'encapsuleur que vous indiquez dépend du système d'exploitation du serveur fédéré. Voir la liste des fichiers de bibliothèque de l'encapsuleur BioRS pour le nom de bibliothèque correct à indiquer dans l'instruction CREATE WRAPPER.

Fichiers de bibliothèque de l'encapsuleur BioRS :

Les fichiers de bibliothèque de l'encapsuleur BioRS sont ajoutés au serveur fédéré lorsque vous installez le serveur fédéré.

Lorsque vous installez le serveur fédéré, trois fichiers de bibliothèque sont ajoutés dans le chemin de répertoire par défaut. Par exemple, si le serveur fédéré fonctionne sous AIX, les fichiers de bibliothèque de l'encapsuleur qui sont ajoutés dans le chemin de répertoire sont libdb2lsbiors.a, libdb2lsbiorsFa et libdb2lsbiorsU.a. Le fichier de bibliothèque de l'encapsuleur par défaut est libdb2lsbiors.a. Les autres fichiers de bibliothèque de l'encapsuleur sont utilisés avec des options spécifiques de l'encapsuleur.

Vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER et indiquer le nom de fichier de bibliothèque de l'encapsuleur par défaut.

Les chemins des répertoires par défaut et les noms de fichier de bibliothèque de l'encapsuleur par défaut sont indiqués dans la table suivante.

Tableau 7. Emplacements et noms de fichier de la bibliothèque de l'encapsuleur BioRS

Système d'exploitation	Chemin de répertoire	Nom de fichier de bibliothèque d'encapsuleur
AIX	/usr/opt/<chemin_installation>/lib32/ /usr/opt/<chemin_installation>/lib64/	libdb2lsbiors.a
Linux	/opt/IBM/db2/<chemin_installation>/lib32 /opt/IBM/db2/<chemin_installation>/lib64	libdb2lsbiors.so
Solaris	/opt/IBM/db2/<chemin_installation>/lib32 /opt/IBM/db2/<chemin_installation>/lib64	libdb2lsbiors.so
Windows	%DB2PATH%\bin	db2lsbiors.dll

<chemin_installation> est le chemin du répertoire dans lequel le serveur fédéré est installé sous UNIX ou Linux.

%DB2PATH% est la variable d'environnement qui est utilisée pour indiquer le chemin de répertoire dans lequel le serveur fédéré est installé sous Windows. Le chemin Windows par défaut est C:\Program Files\IBM\SQLLIB.

Instruction CREATE WRAPPER - exemples pour l'encapsuleur BioRS :

Utiliser l'instruction CREATE WRAPPER pour enregistrer l'encapsuleur BioRS. Les exemples montrent les options qui sont obligatoires pour accéder à des documents BioRS avec et sans serveur proxy.

Enregistrement d'un encapsuleur

Si vous n'utilisez pas de serveur proxy pour accéder aux documents BioRS, émettez l'instruction suivante pour enregistrer l'encapsuleur :

```
CREATE WRAPPER encapsuleur_biors LIBRARY 'libdb2lsbiors.a';
```

encapsuleur_biors

Nom que vous attribuez à l'encapsuleur BioRS. Les doublons de noms d'encapsuleur ne sont pas autorisés.

LIBRARY '*libdb2lsbiors.a*'

Nom du fichier de bibliothèque d'encapsuleur pour les serveurs fédérés qui utilisent les systèmes d'exploitation AIX.

Enregistrement d'un encapsuleur pour un serveur proxy HTTP

Pour enregistrer un encapsuleur et indiquer un serveur proxy HTTP, utilisez l'instruction suivante :

```
CREATE WRAPPER proxy_biors LIBRARY 'libdb2lsbiors.a'
  OPTIONS (PROXY_TYPE 'HTTP',
          PROXY_SERVER_NAME 'proxy.mysite.com',
          PROXY_SERVER_PORT '81');
```

PROXY_TYPE '*HTTP*'

Spécifie le type de proxy utilisé pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont 'NONE', 'HTTP' ou 'SOCKS'.

PROXY_SERVER_NAME '*proxy.mysite.com*'

Indique le nom de serveur proxy ou l'adresse IP. Cette option est obligatoire si la valeur de l'option de serveur PROXY_TYPE est 'HTTP' ou 'SOCKS'.

PROXY_SERVER_PORT '81'

Indique le numéro de port du serveur proxy. Cette option est obligatoire si la valeur de l'option de serveur PROXY_TYPE est 'HTTP' ou 'SOCKS'.

Enregistrement d'un encapsuleur pour un serveur proxy SOCKS

Pour enregistrer un encapsuleur et indiquer un serveur proxy SOCKS sans informations d'authentification, utilisez l'instruction suivante :

```
CREATE WRAPPER encapsuleur_biors LIBRARY 'libdb2lsbiors.so'  
  OPTIONS (PROXY_TYPE 'SOCKS',  
          PROXY_SERVER_NAME 'proxy_socks',  
          PROXY_SERVER_PORT '1081');
```

LIBRARY '*libdb2lsbiors.so*'

Nom du fichier de bibliothèque d'encapsuleur pour les serveurs fédérés qui utilisent les systèmes d'exploitation Linux et Solaris.

PROXY_TYPE 'SOCKS'

Spécifie le type de proxy utilisé pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont 'NONE', 'HTTP' ou 'SOCKS'.

PROXY_SERVER_NAME '*proxy_socks*'

Indique le nom de serveur proxy ou l'adresse IP. Cette option est obligatoire si la valeur de l'option de serveur PROXY_TYPE est 'HTTP' ou 'SOCKS'.

PROXY_SERVER_PORT '1081'

Indique le numéro de port du serveur proxy. Cette option est obligatoire si la valeur de l'option de serveur PROXY_TYPE est 'HTTP' ou 'SOCKS'.

Enregistrement de la définition de serveur pour une source de données BioRS

Vous devez enregistrer chaque serveur BioRS auquel vous voulez accéder dans la base de données fédérée.

Si vous utilisez un serveur proxy pour accéder à des fichiers BioRS, vous pouvez indiquer les informations du serveur proxy comme options lorsque vous enregistrez la définition de serveur. Si vous indiquez les informations du serveur proxy dans la définition de serveur, les options de serveur prévalent sur les options que vous avez indiquées lorsque vous avez enregistré l'encapsuleur.

Procédure

Pour enregistrer une définition de serveur pour une source de données BioRS :

Choisissez la méthode que vous voulez utiliser pour enregistrer la définition de serveur :

Méthode	Procédure.
Utilisation du Centre de contrôle	Démarrez l'assistant des objets fédérés. Cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée et cliquez sur Créer des objets fédérés .

Méthode	Procédure.
Depuis la ligne de commande	<p>Émettez l'instruction CREATE SERVER.</p> <pre>CREATE SERVER <i>nom_définition_serveur</i> VERSION <i>version_number</i> WRAPPER <i>nom_encapsuleur</i> OPTIONS (NODE '<i>nom_noeud</i>');</pre> <p>Si vous utilisez un serveur proxy pour accéder à des fichiers BioRS, vous devez indiquer plusieurs options lorsque vous enregistrez l'encapsuleur BioRS ou la définition de serveur. Pour indiquer les informations du serveur proxy lorsque vous enregistrez l'encapsuleur BioRS, vous émettez cette instruction :</p> <pre>CREATE SERVER <i>nom_définition_serveur</i> VERSION <i>version_number</i> WRAPPER <i>nom_encapsuleur</i> OPTIONS (PROXY_TYPE '<i>type</i>', PROXY_SERVER_NAME '<i>server_name</i>', PROXY_SERVER_PORT '<i>numéro_port</i>');</pre>

Lorsque vous enregistrez la définition de serveur, vous indiquez des options de serveur dans l'instruction CREATE SERVER. Il y a des options de serveur obligatoires et des options de serveur facultatives. L'option de serveur NODE est obligatoire.

Une fois la définition de serveur enregistrée, utilisez l'instruction ALTER SERVER pour ajouter ou déplacer des options de serveur.

Instruction CREATE SERVER - exemples pour l'encapsuleur BioRS :

Utiliser l'instruction CREATE SERVER pour enregistrer des définitions de serveur pour l'encapsuleur BioRS. Les exemples montrent les paramètres obligatoires, les paramètres facultatifs et les options de serveur supplémentaires que vous pouvez indiquer.

Enregistrement d'une définition de serveur avec les paramètres obligatoires

L'exemple suivant vous montre comment enregistrer une définition de serveur pour un encapsuleur BioRS en émettant l'instruction CREATE SERVER :

```
CREATE SERVER serveur_biors VERSION '5.2' WRAPPER encapsuleur_biors
OPTIONS(NODE 'biors_myco.com');
```

serveur_biors

Nom que vous attribuez au serveur BioRS. Les doublons de noms de définition de serveur ne sont pas autorisés.

VERSION '*5.2*'

La version du serveur BioRS auquel vous voulez accéder. Les versions de BioRS prises en charge sont les versions 5.0.14 et 5.2. Si vous accédez à un serveur BioRS de la version 5.2, vous devez indiquer '*5.2*' comme valeur du paramètre VERSION. Vous n'avez pas besoin d'indiquer cette option si vous utilisez la version 5.0.14. La valeur par défaut '*1.0*', qui équivaut à la version 5.0.14, est utilisée pour ce paramètre si vous n'indiquez pas de valeur.

WRAPPER *encapsuleur_biors*

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

NODE *'biors_myco.com'*

Indique le nom d'hôte du système sur lequel l'utilitaire d'interrogation BioRS est disponible. La valeur par défaut est *localhost*. Cette valeur fait la distinction entre les majuscules et les minuscules.

Bien que le nom du noeud soit indiqué comme option dans l'instruction CREATE SERVER, il doit être indiqué obligatoirement pour les sources de données BioRS.

Enregistrement d'une définition de serveur à l'aide de paramètres et d'options de serveur facultatifs

L'exemple suivant montre des paramètres et options de serveur supplémentaires que vous pouvez indiquer lorsque vous enregistrez une définition de serveur pour un encapsuleur BioRS :

```
CREATE SERVER serveur_biors TYPE BioRS VERSION '5.2'  
  WRAPPER encapsuleur_biors  
  OPTIONS (NODE 'biors_server2.com', PORT '5555', TIMEOUT 30 ,  
  CASE_SENSITIVE 'N');
```

TYPE *BioRS*

Indique le type de serveur de source de données auquel vous configurez l'accès. Pour l'encapsuleur BioRS, le type de serveur doit être BioRS. Ce paramètre est facultatif.

PORT *'5555'*

Indique le numéro du port à utiliser pour se connecter au serveur BioRS. La valeur par défaut est *'5014'*.

TIMEOUT *30*

Indique la durée, en minutes, pendant laquelle l'encapsuleur BioRS attend une réponse du serveur BioRS. La valeur par défaut est *10*. Ce paramètre est facultatif.

CASE_SENSITIVE *'N'*

Indique si le serveur BioRS traite les noms d'une manière sensible à la casse. Les valeurs valides sont *'Y'* ou *'N'*. La valeur par défaut est *'Y'*.

Dans BioRS, un paramètre de configuration contrôle la sensibilité à la casse des données stockées sur la machine du serveur BioRS. L'option CASE_SENSITIVE est la contrepartie sur le serveur fédéré de ce paramètre de configuration sur le système BioRS. Vous devez synchroniser les paramètres de configuration du respect de la casse sur le serveur BioRS entre votre système BioRS et le serveur fédéré. Si vous ne préservez pas la synchronisation des paramètres de configuration du respect de la casse entre le serveur BioRS et le serveur fédéré, des erreurs se produiront lorsque vous essaierez d'accéder aux données BioRS à l'aide du serveur fédéré.

Important : Vous ne pouvez pas modifier ou supprimer l'option CASE_SENSITIVE après avoir créé une nouvelle définition BioRS. Si vous avez besoin de modifier l'option CASE_SENSITIVE, vous devez supprimer, puis recréer la définition de serveur. Si vous supprimez la définition de serveur BioRS, vous devez aussi créer tous les pseudonymes BioRS qui

référençaient cette définition de serveur. Le serveur fédéré supprime automatiquement tous les pseudonymes qui correspondent à un serveur supprimé.

Enregistrement d'une définition de serveur comportant un serveur proxy

```
CREATE SERVER serv_proxy_biors VERSION 5.2 WRAPPER proxy_biors
  OPTIONS (NODE 'biors.mysite.com',
    PORT '5555',
    PROXY_TYPE 'HTTP'
    PROXY_SERVER_NAME 'proxy.mysite.com'
    PROXY_SERVER_PORT '81'
```

PROXY_TYPE '*HTTP*'

Indique le type de serveur proxy qui est utilisé pour accéder à Internet derrière un pare-feu. Les valeurs valides sont 'NONE', 'HTTP' ou 'SOCKS'.

PROXY_SERVER_NAME '*proxy.mysite.com*'

Indique le nom de serveur proxy ou l'adresse IP. Cette option est obligatoire si la valeur de l'option de serveur PROXY_TYPE est 'HTTP' ou 'SOCKS'.

PROXY_SERVER_PORT '*81*'

Indique le numéro de port du serveur proxy. Cette option est obligatoire si la valeur de l'option de serveur PROXY_TYPE est 'HTTP' ou 'SOCKS'.

Enregistrement d'une définition de serveur comportant un serveur proxy avec des informations d'authentification

Pour enregistrer une définition de serveur et indiquer un serveur proxy SOCKS avec des informations d'authentification, utilisez l'instruction suivante :

```
CREATE SERVER serv_proxy_biors VERSION 5.2 WRAPPER proxy_biors
  OPTIONS (NODE 'biors.mysite.com',
    PORT '5555',
    PROXY_TYPE 'SOCKS'
    PROXY_SERVER_NAME 'proxy_socks',
    PROXY_SERVER_PORT '1081',
    PROXY_AUTHID 'argle'
    PROXY_PASSWORD 'bargle')
```

PROXY_AUTHID '*argle*'

Indique l'ID utilisateur sur le serveur proxy. Cette option de serveur est obligatoire lorsque la valeur de l'option de serveur PROXY_TYPE est 'SOCKS'.

PROXY_PASSWORD '*bargle*'

Indique le mot de passe sur le serveur proxy qui est associé au nom d'utilisateur '*argle*'. Cette option de serveur est obligatoire lorsque la valeur de l'option de serveur PROXY_TYPE est 'SOCKS'.

Création des mappages utilisateur pour une source de données BioRS

Lorsque vous essayez d'accéder à un serveur BioRS, le serveur fédéré établit une connexion avec le serveur BioRS. Suivant les méthodes d'accès au compte qui sont utilisées dans votre système BioRS, vous pouvez ne pas avoir besoin de créer des mappages utilisateur.

Un mappage utilisateur est une association entre chaque ID utilisateur et mot de passe de serveur fédéré et l'ID utilisateur et le mot de passe de la source de données correspondants.

Il y a deux méthodes pour indiquer les mappages utilisateur avec les systèmes fédérés. Vous pouvez utiliser un référentiel externe, tel que LDAP, pour stocker les mappages utilisateur ou bien vous pouvez créer les mappages utilisateur dans le catalogue de la base de données fédérée.

Avant de commencer

Les méthodes d'accès au compte qui sont utilisées dans votre système BioRS détermineront si vous avez besoin de créer des mappages utilisateur :

- Si votre serveur BioRS est configuré pour un accès d'invité à tous les comptes utilisateur, vous n'avez pas besoin de créer des mappages utilisateur. Le serveur fédéré utilise un compte d'invité pour accéder au serveur BioRS.
- Si vous avez un référentiel externe, tel que LDAP, pour stocker les mappages utilisateur, vous n'avez pas besoin de créer des mappages utilisateur. Vous devez indiquer l'option DB2_UM_PLUGIN sur l'encapsuleur BioRS. Vous pouvez indiquer cette option lorsque vous enregistrez ou modifiez l'encapsuleur. Le schéma dans le référentiel externe doit comporter l'accès d'invité.
- Si votre serveur BioRS est configuré pour authentifier les comptes utilisateur avec des ID et des mots de passe, vous devez créer des mappages utilisateur dans la base de données fédérée pour les comptes utilisateur qui utiliseront l'encapsuleur BioRS pour accéder à des sources de données BioRS.
- Si votre serveur BioRS est configuré pour utiliser un panachage de comptes utilisateur invités et authentifiés, vous devez créer des mappages utilisateur pour les comptes utilisateur authentifiés qui utiliseront l'encapsuleur BioRS pour accéder à des sources de données BioRS.

A propos de cette tâche

Les mappages utilisateur permettent d'authentifier l'accès des utilisateurs ou des applications qui interrogent une source de données BioRS avec l'encapsuleur BioRS. Si un mappage utilisateur n'est pas défini pour un utilisateur ou une application, le serveur fédéré utilise un compte d'invité. Si une banque de données qui est interrogée exige une authentification, un message d'erreur peut être renvoyé.

Pour garantir que l'ID utilisateur et le mot de passe corrects soient transmis au serveur BioRS, créez des mappages utilisateur dans votre base de données fédérée pour les utilisateurs qui sont autorisés à consulter des sources de données BioRS. Lorsque vous créez un mappage utilisateur, le mot de passe distant est stocké sous un format chiffré dans une table du catalogue système de la base de données fédérée.

Procédure

Pour mapper un ID utilisateur local à l'ID utilisateur et au mot de passe du serveur BioRS :

Choisissez la méthode que vous voulez utiliser pour créer les mappages utilisateur.

Méthode	Procédure.
Utilisation du Centre de contrôle	Démarrez l'assistant des objets fédérés. Cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée et cliquez sur Créer des objets fédérés .

Méthode	Procédure.
Utilisation de la ligne de commande	<p>Emettez l'instruction CREATE USER MAPPING. Par exemple :</p> <pre>CREATE USER MAPPING FOR <i>IDutilisateur_local</i> SERVER <i>nom_définition_serveur</i> OPTIONS (REMOTE_AUTHID ' <i>IDutilisateur_distant'</i>, REMOTE_PASSWORD '<i>motdepasse_distant'</i>) PROXY_AUTHID '<i>IDutilisateur</i> <i>_serveur_proxy'</i>, PROXY_PASSWORD '<i>motdepasse_serveur</i> <i>_proxy'</i>;</pre>

Si vous indiquez des informations d'authentification pour le serveur proxy lorsque vous enregistrez une définition de serveur et un mappage utilisateur, les valeurs que vous indiquez dans l'instruction CREATE USER MAPPING prévalent sur les valeurs que vous indiquez dans l'instruction CREATE SERVER.

Par exemple, vous avez dix personnes dans votre organisation et vous indiquez des informations d'authentification lorsque vous enregistrez la définition de serveur. Vous créez des mappages utilisateur pour trois des dix personnes. Lorsque ces trois personnes accèdent au système fédéré, ce sont les informations d'authentification que vous avez indiquées lorsque vous avez créé les mappages utilisateur qui sont utilisées. Pour les sept autres personnes, ce sont les informations d'authentification que vous avez indiquées lorsque vous avez enregistré la définition de serveur qui sont utilisées.

Instruction CREATE USER MAPPING - exemples pour l'encapsuleur BioRS :

Utiliser l'instruction CREATE USER MAPPING pour mapper un ID utilisateur du serveur fédéré à un ID utilisateur et à un mot de passe du serveur BioRS.

Vous pouvez créer un mappage utilisateur en indiquant un compte utilisateur invité, un compte utilisateur authentifié, le registre spécial USER ou un serveur proxy.

Création d'un mappage utilisateur pour un compte utilisateur invité

L'option utilisateur GUEST indique si l'encapsuleur BioRS doit utiliser un compte d'invité pour accéder au serveur BioRS.

L'exemple suivant montre comment indiquer qu'un compte utilisateur invité est utilisé pour accéder au serveur BioRS :

```
CREATE USER MAPPING FOR charlie SERVER serveur_biors
  OPTIONS (GUEST 'Y');
```

charlie Indique l'identificateur utilisateur local que vous mappez à un ID utilisateur distant, qui est défini sur le serveur BioRS.

SERVER *serveur_biors*

Indique le nom de définition de serveur que vous avez enregistré dans l'instruction CREATE SERVER pour le serveur BioRS.

GUEST 'Y'

Indique que l'encapsuleur BioRS utilise un compte utilisateur invité pour authentifier cet utilisateur.

Création d'un mappage utilisateur pour un compte utilisateur authentifié

L'exemple suivant montre comment mapper un ID utilisateur du serveur fédéré à un ID utilisateur et à un mot de passe du serveur BioRS :

```
CREATE USER MAPPING FOR charlie SERVER serveur_biors
  OPTIONS (REMOTE_AUTHID 'charlene',
          REMOTE_PASSWORD 'all4one');
```

charlie Indique l'identificateur utilisateur local que vous mappez à un ID utilisateur et à un mot de passe distants, qui sont définis sur le serveur BioRS.

SERVER *serveur_biors*

Indique le nom de définition de serveur que vous avez enregistré dans l'instruction CREATE SERVER pour le serveur BioRS.

REMOTE_AUTHID '*charlene*'

Indique l'ID utilisateur sur le serveur BioRS auquel vous mappez *charlie*. Cet ID distant doit être d'un format prévu par le serveur BioRS.

REMOTE_PASSWORD '*all4one*'

Indique le mot de passe qui est associé à '*charlene*'.

Création d'un mappage utilisateur à l'aide d'un registre spécial

Vous pouvez utiliser le registre spécial USER de la base de données fédérée pour mapper l'identificateur utilisateur de la personne qui émet l'instruction CREATE USER MAPPING à l'identificateur utilisateur de la source de données qui est indiqué dans l'option utilisateur REMOTE_AUTHID.

L'exemple suivant montre une instruction CREATE USER MAPPING qui comporte le registre spécial USER :

```
CREATE USER MAPPING FOR USER SERVER serveur_biors
  OPTIONS (REMOTE_AUTHID 'charlene', REMOTE_PASSWORD 'all4one');
```

Création d'un mappage utilisateur pour un serveur proxy

L'exemple suivant montre comment mapper un ID utilisateur du serveur fédéré à un ID utilisateur et à un mot de passe du serveur BioRS :

Pour enregistrer une définition de serveur et indiquer un serveur proxy SOCKS avec des informations d'authentification, utilisez l'instruction suivante :

```
CREATE USER MAPPING FOR charlie SERVER proxy_biors
  OPTIONS (REMOTE_AUTHID 'charlene',
          REMOTE_PASSWORD 'all4one'
          PROXY_AUTHID 'chuck'
          PROXY_PASSWORD 'them2us');
```

PROXY_AUTHID '*chuck*'

Indique l'ID utilisateur sur le serveur proxy. Cette option de mappage utilisateur est obligatoire lorsque le serveur proxy exige une authentification.

PROXY_PASSWORD '*them2us*'

Indique le mot de passe sur le serveur proxy qui est associé au nom

d'utilisateur '*chuck*'. Cette option de mappage utilisateur est obligatoire lorsque le serveur proxy exige une authentification.

Enregistrement de pseudonymes pour les sources de données BioRS

Dans chaque définition de serveur BioRS que vous enregistrez, vous devez enregistrer un pseudonyme pour chaque banque de données à laquelle vous voulez accéder. Utilisez ces pseudonymes, au lieu des noms des banques de données, lorsque vous interrogez les serveurs BioRS.

Avant de commencer

- Si un nom de banque de données BioRS ne respecte pas la syntaxe attendue par l'instruction CREATE NICKNAME, vous devez utiliser l'option de pseudonyme REMOTE_OBJECT lorsque vous enregistrez le pseudonyme.
- Si un nom d'élément BioRS ne respecte pas la syntaxe attendue par l'instruction CREATE NICKNAME, vous devez utiliser l'option de colonne ELEMENT_NAME lorsque vous enregistrez le pseudonyme.

Restrictions

N'utilisez pas l'élément AllText de BioRS comme première colonne pour un pseudonyme. Vous pouvez utiliser l'élément AllText de BioRS dans n'importe quelle autre position de colonne (par exemple comme deuxième ou troisième colonne).

A propos de cette tâche

Une fois qu'une source de données a été intégrée dans le système BioRS, elle est désignée comme une *banque de données* dans BioRS. Les banques de données dans BioRS équivalent aux pseudonymes dans un système fédéré.

Les noms que vous donnez aux pseudonymes peuvent comporter jusqu'à 128 caractères.

Vous pouvez enregistrer un pseudonyme à l'aide du Centre de contrôle ou de la ligne de commande. Le Centre de contrôle comporte un assistant pour vous guider à travers les étapes nécessaires pour enregistrer le pseudonyme.

Procédure

Pour enregistrer un pseudonyme pour une banque de données BioRS :

Choisissez la méthode que vous voulez utiliser pour enregistrer le pseudonyme :

Méthode	Procédure.
Utilisation du Centre de contrôle	Démarrez l'assistant des objets fédérés. Cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée et cliquez sur Créer des objets fédérés .
Utilisation de la ligne de commande	Emettez l'instruction CREATE NICKNAME. Par exemple : <pre>CREATE NICKNAME <i>pseudonyme</i> (<i>nom_colonne type_données</i> OPTIONS (<i>options_colonne_pseudonyme</i>), <i>nom_colonne type_données</i> OPTIONS (<i>options_colonne_pseudonyme</i>), <i>nom_colonne type_données</i> OPTIONS (<i>options_colonne_pseudonyme</i>)) FOR SERVER <i>nom_définition_serveur</i> OPTIONS (<i>options_pseudonyme</i>);</pre>

Lorsque vous créez un pseudonyme BioRS, vous définissez une liste de colonnes de pseudonymes. Les colonnes de pseudonymes indiquées doivent correspondre aux éléments d'un format de banque de données BioRS spécifique. BioRS définit cinq types de données possibles pour les éléments : texte, numérique, date, auteur et référence. Les types de données BioRS peuvent être mappés uniquement aux types de données CHAR, CLOB ou VARCHAR qui sont utilisés par la base de données fédérée.

Répétez cette étape pour chaque banque de données BioRS pour laquelle vous voulez créer un pseudonyme.

Instruction CREATE NICKNAME - exemples pour l'encapsuleur BioRS :

Utiliser l'instruction CREATE NICKNAME pour enregistrer un pseudonyme pour une banque de données BioRS à laquelle vous voulez accéder.

Création de pseudonymes simples, mais limités, pour les banques de données

La manière la plus simple d'enregistrer un pseudonyme pour une banque de données BioRS est de donner au pseudonyme le même nom que la banque de données BioRS.

Par exemple :

```
CREATE NICKNAME SwissProt
  (ID VARCHAR(32) OPTIONS (ELEMENT_NAME '_ID_'),
  ALLTEXT VARCHAR(128),
  ENTRYDATE VARCHAR (64))
FOR SERVER serveur_biors;
```

Le nom du pseudonyme est *SwissProt*, qui est le même nom que le nom de la banque de données BioRS correspondante.

L'utilisation de cette syntaxe CREATE NICKNAME simple vous limite de deux manières :

1. Vous êtes limité à une famille de pseudonymes pour chaque schéma de base de données fédérée. Par exemple, vous avez deux banques de données liées par une relation parent-enfant. Les banques de données sont SWISSPROT et SPFEAT. Ces banques de données constituent une famille. Si vous utilisez la syntaxe par défaut de l'instruction CREATE NICKNAME, vous aurez le pseudonyme SWISSPROT pour la banque de données SWISSPROT et le pseudonyme SPFEAT pour la banque de données SPFEAT. Pour avoir plusieurs pseudonymes pour SWISSPROT dans le schéma, vous devez utiliser l'option de pseudonyme REMOTE_OBJECT lorsque vous enregistrez le pseudonyme.
2. Vous êtes limité aux banques de données dont les noms peuvent être utilisés comme des pseudonymes. Le nom de banque de données doit respecter la syntaxe qui est prise en charge par le serveur fédéré. Par exemple, si le nom de banque de données comporte un point ou un espace, vous devez utiliser l'option de pseudonyme REMOTE_OBJECT lorsque vous enregistrez le pseudonyme.

Création de plusieurs pseudonymes pour la même banque de données

L'option de pseudonyme REMOTE_OBJECT indique le nom de la banque de données BioRS qui est associé au pseudonyme. Le nom que vous indiquez dans l'option de pseudonyme REMOTE_OBJECT détermine le schéma et la banque de données BioRS pour le pseudonyme. L'option de pseudonyme REMOTE_OBJECT indique aussi la relation entre le pseudonyme et les autres pseudonymes.

L'exemple suivant montre le même ensemble de caractéristiques de pseudonyme que l'exemple précédent, mais en modifiant le pseudonyme. L'exemple utilise l'option de pseudonyme REMOTE_OBJECT pour indiquer la banque de données BioRS pour laquelle le pseudonyme est défini :

```
CREATE NICKNAME NewSP
  (ID VARCHAR(32) OPTIONS (ELEMENT_NAME '_ID_'),
  ALLTEXT VARCHAR(128),
  ENTRYDATE VARCHAR (64))
  FOR SERVER serveur_biors
  OPTIONS (REMOTE_OBJECT 'SwissProt');
```

Création de pseudonymes pour des banques de données qui ne respectent pas la syntaxe fédérée

L'exemple suivant montre comment créer un pseudonyme pour une banque de données BioRS distante qui ne respecte pas la syntaxe requise par le serveur fédéré :

```
CREATE NICKNAME SwissFT
  (ID VARCHAR(32) OPTIONS (ELEMENT_NAME '_ID_'),
  ALLTEXT VARCHAR (128),
  ENTRYDATE VARCHAR (64),
  FtLength VARCHAR (16))
  FOR SERVER biors1
  OPTIONS (REMOTE_OBJECT 'SwissProt.Features');
```

SwissFT

Pseudonyme unique utilisé pour identifier la banque de données BioRS.

ID VARCHAR(32) OPTIONS (ELEMENT_NAME '_ID_')

Nom et type de données d'une colonne de table. L'option de colonne ELEMENT_NAME est indiquée pour la colonne ID.

L'option de colonne ELEMENT_NAME indique le nom d'élément BioRS. La sensibilité à la casse de ce nom dépend de la sensibilité à la casse du serveur BioRS et de la valeur de l'option du serveur CASE_SENSITIVE. Vous devez spécifier le nom de l'élément BioRS uniquement s'il est différent du nom de colonne. Les valeurs de l'option de colonne doivent être placées entre apostrophes.

En général, vous utiliserez l'option de colonne ELEMENT_NAME dans les cas suivants :

- Lorsqu'un nom d'élément BioRS contient des caractères, tels que les points et les espaces, qui ne respectent pas de syntaxe fédérée valide. Par exemple, si votre banque de données contient un élément nommé Pub.Date, vous ne pouvez pas utiliser le nom d'élément comme nom de colonne. Les caractères tels que les points et les espaces ne sont pas pris en charge. Vous devez mapper le nom d'élément à un nom de colonne valide.
- Lorsque la syntaxe d'un nom d'élément BioRS ne respecte pas les normes que vous ou votre organisation avez définies pour votre système fédéré. Par exemple, si votre organisation a défini que les conventions des schémas, pseudonymes et colonnes doivent comporter un préfixe, un nom d'élément BioRS risque de ne pas pouvoir être utilisé comme nom de colonne.
- Lorsque le nom d'élément BioRS risque d'être ambigu pour les utilisateurs fédérés.

ALLTEXT VARCHAR(128)

Nom et type de données d'une colonne de table.

ENTRYDATE VARCHAR(64)

Nom et type de données d'une colonne de table.

FtLength VARCHAR(16)

Nom et type de données d'une colonne de table.

SERVER *biors1*

Nom que vous avez affecté au serveur BioRS dans l'instruction CREATE SERVER.

OPTIONS (REMOTE_OBJECT 'SwissProt.Features')

Spécifie le nom de la banque de données BioRS associée au pseudonyme. Ce nom détermine le schéma et la banque de données BioRS pour le pseudonyme. Ce nom indique aussi la relation entre le pseudonyme et les autres pseudonymes.

La sensibilité à la casse de ce nom dépend de la sensibilité à la casse du serveur BioRS et de la valeur de l'option du serveur CASE_SENSITIVE.

Important : Vous ne pouvez pas modifier ou supprimer ce nom avec l'instruction ALTER NICKNAME. Si le nom d'une banque de données BioRS change, vous devez supprimer le pseudonyme et le recréer.

Vous devez indiquer l'option de pseudonyme REMOTE_OBJECT lorsque le nom d'une banque de données BioRS ne respecte pas une syntaxe fédérée valide. Dans cet exemple, le nom de banque de données "SwissProt.Features" ne respecte pas la syntaxe pour plusieurs raisons. Le nom de banque de données contient un caractère, un point, qui n'est pas admis par la syntaxe fédérée, ainsi qu'un panachage de majuscules et de minuscules.

En général, vous utiliserez l'option de pseudonyme `REMOTE_OBJECT` dans les cas suivants :

- Lorsqu'un nom de banque de données BioRS contient des caractères, tels que les points et les espaces, qui ne respectent pas la syntaxe fédérée valide. Vous devez mapper le nom de banque de données à un nom fédéré valide.
- Lorsque le respect de la casse d'un nom de banque de données BioRS ne respecte pas les normes que vous ou votre organisation avez définies pour votre système fédéré. Par exemple, si votre organisation a défini que les conventions des schémas, pseudonymes et colonnes doivent comporter un préfixe, un nom de banque de données BioRS risque de ne pas pouvoir être utilisé comme nom.
- Lorsque le nom de banque de données BioRS risque d'être ambigu pour les utilisateurs fédérés.

Création de pseudonymes pour une banque de données liée à une autre banque de données BioRS

L'exemple suivant montre comment créer un pseudonyme pour une table qui utilise une banque de données BioRS qui est liée à une autre banque de données BioRS :

```
CREATE NICKNAME SwissFT2
  (ID VARCHAR(32) OPTIONS (ELEMENT_NAME '_ID_'),
  ALLTEXT VARCHAR (1200),
  FtKey VARCHAR (32),
  FtLength VARCHAR (64),
  FtDescription VARCHAR (128),
  Parent VARCHAR (32) OPTIONS (REFERENCED_OBJECT 'SwissProt'))
FOR SERVER biors1
OPTIONS (REMOTE_OBJECT 'SwissProt.Features');
```

Le nom de ce pseudonyme est `SwissFT2`. Les colonnes de table sont `ID`, `ALLTEXT`, `FtKey`, `FtLength`, `FtDescription` et `Parent`. L'option de colonne `ELEMENT_NAME` est indiquée pour la colonne `ID`. L'option `REMOTE_OBJECT` est utilisée pour indiquer le nom de la banque de données BioRS à laquelle le pseudonyme correspond.

En outre, la colonne `Parent` utilise l'option `REFERENCED_OBJECT`. Vous devez indiquer cette option pour les colonnes qui correspondent aux éléments du type de données Référence BioRS. L'option `REFERENCED_OBJECT` indique le nom de la banque de données BioRS à laquelle la colonne renvoie. Dans ce cas, l'élément `Parent` renvoie à la banque de données `SwissProt BioRS`.

Fonctions et requêtes BioRS personnalisées

Les fonctions BioRS personnalisées sont utilisées avec l'encapsuleur BioRS pour envoyer des prédicats au moteur de requête BioRS.

L'environnement fédéré utilise deux moteurs de requête. Pour l'encapsuleur BioRS, ces moteurs de requête sont la base de données fédérée et BioRS. Vous pouvez indiquer que les prédicats sont envoyés au moteur BioRS à l'aide des quatre fonctions personnalisées BioRS.

Toutes les fonctions personnalisées de l'encapsuleur BioRS doivent être enregistrées sous le nom de schéma `BIORS`. Vous devez inclure le schéma `BIORS` chaque fois que vous utilisez les fonctions.

Les fonctions personnalisées de l'encapsuleur BIORs sont :

- BIORs.CONTAINS
- BIORs.CONTAINS_LE
- BIORs.CONTAINS_GE
- BIORs.SEARCH_TERM

Les fonctions CONTAINS de BioRS

Les fonctions personnalisées BIORs.CONTAINS, BIORs.CONTAINS_LE et BIORs.CONTAINS_GE attendent un argument de colonne du terme recherché et un argument de texte de requête. L'exemple suivant présente une instruction BIORs.CONTAINS :

```
BIORs.CONTAINS (colonne_terme_recherché,terme_requête)
```

La valeur de l'argument de la colonne de terme recherché doit se référer à une colonne BioRS indexée. L'utilisation d'une colonne non indexée produit le message d'erreur SQL30090N ("Opération non valide pour l'environnement d'exécution de l'application").

La valeur de l'argument de terme de requête est la valeur qui est utilisée pour rechercher l'élément indexé indiqué dans l'argument de colonne du terme recherché.

La valeur de l'argument de terme de requête ne peut être qu'un littéral, une variable hôte ou une référence de colonne. Vous ne pouvez pas utiliser de concaténation arithmétique ni de concaténation de chaînes. En outre, la valeur de l'argument de terme de requête ne peut pas être NULL, même si la colonne de terme recherché qui est utilisée est définie comme autorisant les valeurs nulles.

L'écriture en majuscules et minuscules de l'argument de terme de requête est sans importance.

Types de données valides

Les types de données et formats valides de l'argument de terme de requête dépendent du type de données BioRS de la colonne du terme recherché qui est utilisée. BioRS définit cinq types de données possibles : texte, auteur, date, numérique et référence.

Les types de données BioRS et les termes de la fonction de recherche qui sont valides pour chaque type de données sont indiqués dans la table suivante.

Tableau 8. Types de données BioRS et termes de la fonction de recherche

Type de données de la colonne du terme recherché	Terme recherché valide	Format
Texte	VARCHAR() ou CHAR()	Le terme de texte BioRS, avec les caractères génériques.

Tableau 8. Types de données BioRS et termes de la fonction de recherche (suite)

Type de données de la colonne du terme recherché	Terme recherché valide	Format
Auteur	VARCHAR() ou CHAR()	La référence d'auteur BioRS sous la forme "<nom>, <init>". "<nom>" est le nom de l'auteur. "<init>" sont les initiales de l'auteur, sans points. Un blanc entre la virgule et les initiales est accepté. <nom> peut aussi être indiqué seul, sans virgule, ni initiales.
Date	VARCHAR(), CHAR(), DATE ou TIMESTAMP	Si chaîne de caractères, format de date valide pour la base de données fédérée, aaaa/mm/jj.
Nombre	VARCHAR() ou CHAR(), INTEGER, SMALLINT, BIGINT REAL, DOUBLE, DECIMAL	Formats de nombres valides pour la base de données fédérée.
Référence	VARCHAR() ou CHAR()	Terme de texte BioRS.

Toutes les autres combinaisons de colonnes de terme recherché du type de données BioRS et d'arguments de terme recherché produisent le message d'erreur SQL30090N ("Opération non valide pour l'environnement d'exécution de l'application").

Utilisation des caractères génériques

L'argument de terme recherché pour les colonnes de terme recherché des types de données texte, auteur et référence doit correspondre à un modèle de langage de requête BioRS. Dans BioRS, les arguments de terme recherché peuvent se composer de chaînes alphanumériques et de caractères génériques. La fonction BIOR.S.CONTAINS prend en charge deux caractères génériques : ? (le point d'interrogation) et * (l'astérisque).

Le caractère générique ? correspond à un seul caractère. Par exemple, le prédicat BIOR.S.CONTAINS (description, 'bactéri?')=1 correspond au terme bactérie, mais pas au terme bactériel.

Le caractère générique * correspond à zéro, un ou plusieurs caractères. Par exemple, le prédicat BIOR.S.CONTAINS (description, 'bactéri*')=1 correspond aux termes bactéri, bactérie et bactériologique.

Pour des informations détaillées sur les modèles du langage de requête BioRS, consultez votre documentation BioRS.

Indication de fonctions CONTAINS BioRS dans les requêtes

La fonction BIOR.S.CONTAINS peut être indiquée pour tous les types de colonne BioRS.

Les fonctions personnalisées BIOR.S.CONTAINS_GE et BIOR.S.CONTAINS_LE peuvent être indiquées uniquement pour les colonnes dont le type de données BioRS sous-jacent est Nombre ou Date. La fonction BIOR.S.CONTAINS_GE sélectionne les lignes dans lesquelles la colonne contient une valeur supérieure ou

égale à la valeur représentée par l'argument de terme recherché. La fonction `BIORS.CONTAINS_LE` sélectionne les lignes dans lesquelles la colonne contient une valeur inférieure ou égale à la valeur représentée par l'argument de terme recherché.

Les fonctions `BIORS.CONTAINS`, `BIORS.CONTAINS_GE` et `BIORS.CONTAINS_LE` renvoient un résultat entier. Lorsque l'une des trois fonctions `CONTAINS` est utilisée dans un prédicat, la valeur de retour doit être comparée à la valeur 1 à l'aide des opérateurs `=` ou `<>`. Par exemple :

```
SELECT * FROM s.MySP WHERE BIOR.S.CONTAINS (s.AllText, 'muscus') = 1;
```

L'expression `NOT (BioRS.Contains (col,value) = 1)` est équivalente à l'expression `BioRS.CONTAINS (col,value) <> 1`.

La fonction `SEARCH_TERM` de BioRS

La fonction personnalisée `BIORS.SEARCH_TERM` attend un argument de colonne du terme recherché et un terme recherché. L'exemple suivant illustre la syntaxe de la fonction personnalisée `SEARCH_TERM` :

```
BIORS.SEARCH_TERM (colonne_terme_recherché,terme_requête)
```

La valeur de l'argument de colonne du terme recherché doit se référer à la colonne qui représente l'élément `_ID_`.

La valeur de l'argument de terme recherché est une expression qui peut référencer plusieurs éléments.

La valeur de l'argument de terme de requête ne peut être qu'un littéral, une variable hôte ou une référence de colonne. Vous ne pouvez pas utiliser de concaténation arithmétique ni de concaténation de chaînes. En outre, la valeur de l'argument de terme de requête ne peut pas être `NULL`, même si la colonne de terme recherché qui est utilisée est définie comme autorisant les valeurs nulles.

L'écriture en majuscules et minuscules de l'argument de terme de requête est sans importance.

Indication de la fonction `SEARCH_TERM` de BioRS dans les requêtes

Vous pouvez exécuter des requêtes qui ne seraient pas possibles sinon en émettant la fonction `BIORS.SEARCH_TERM`. Vous pouvez utiliser cette fonction pour indiquer un terme recherché à l'aide du format BioRS. La fonction `BIORS.SEARCH_TERM` attend deux arguments. Le premier argument est une référence à la colonne `_ID_` du pseudonyme auquel le terme doit être appliqué. Le deuxième argument est une chaîne de caractères qui contient le terme sans nom de banque de données.

L'exemple suivant sélectionne toutes les colonnes pour les entrées dans la banque de données MyEMBL dans lesquelles l'élément `SeqLength` contient une valeur supérieure ou égale à 100.

```
SELECT * FROM MyEMBL s WHERE  
  BIOR.S.SEARCH_TERM (s.ID, '[SeqLength GREATER number:100;]') = 1;
```

L'exemple suivant sélectionne la colonne `MolWeight` dans le pseudonyme Swiss dans laquelle la valeur de l'élément `MolWeight` est supérieure ou égale à 100368.

```
SELECT s.molweight FROM Swiss s WHERE
      BIORs.SEARCH_TERM (s.ID, '[MolWeight GREATER number:100368;]') = 1;
```

Opérations d'égalité dans les requêtes BioRS

Vous pouvez utiliser un opérateur d'égalité (=) dans les expressions littérales ou dans les requêtes de jointure, avec certaines limitations.

Si vous utilisez l'opérateur d'égalité dans une expression littérale ou dans une requête de jointure, l'opérateur d'égalité doit référencer l'élément `_ID_` d'une banque de données BioRS pour que la requête soit envoyée au serveur BioRS. Les requêtes qui comportent un opérateur d'égalité, mais qui ne référencent pas l'élément `_ID_` ne sont pas envoyées pour traitement par le serveur BioRS.

Vous pouvez utiliser l'opérateur d'égalité dans une expression littérale. Par exemple :

```
ID = 'swissprot:100K_RAT'
```

Vous pouvez utiliser un prédicat d'égalité dans une jointure entre une banque de données BioRS et une autre table locale ou des pseudonymes non-BioRS. Par exemple :

```
SELECT n.ID, n.EntryDate, t.C1 FROM w46851_n1 n, w46851_t1 t WHERE t.ID = n.ID
```

Une jointure entre des banques de données BioRS doit référencer l'élément `_ID_` d'une banque de données et un élément de type référence pour l'autre banque de données.

Cependant, l'utilisation d'un prédicat d'égalité peut renvoyer des résultats qui diffèrent des résultats prévus sous ces conditions :

Correspondance insensible à la casse

L'opération n'est pas sensible à la casse. Par exemple, `ID='100k_rat'` correspond aux deux chaînes suivantes :

- '100k_rat'
- '100K_RAT'

Correspondance avec caractères génériques

L'instruction `ID='100K_R*'` correspond aussi bien à '100K_RAT' qu'à '100K_RODENT.'

Préfixation de la banque de données

L'opération renvoie un préfixe qui indique la banque de données source. Par exemple, `ID='100K_RAT'` dans une jointure sur la banque de données SwissProt pourrait renvoyer la valeur 'swissprot:100K_RAT.'

Remarque : Ne créez pas d'applications qui aient besoin de l'un des comportements décrits.

L'exemple suivant illustre le comportement du prédicat d'égalité dans une jointure.

La table locale, `w46851_t1`, contient les valeurs suivantes :

ID	C1
swissprot:100K_RAT	0
swissprot:RAT	1
swissprot:100K_R	2
swissprot:100K_R*	3
swissprot:100k_rat	4
100K_RAT	100

RAT	101
100K_R	102
100K_R*	103
100k_rat	104

Vous pouvez joindre la table w46851_t1 avec un pseudonyme w46851_n1 basé sur la banque de données SwissProt. L'instruction suivante montre la requête de jointure avec une opération d'égalité :

```
SELECT n.ID, n.EntryDate, t.C1 FROM w46851_n1 n, w46851_t1 t WHERE t.ID = n.ID
```

Les résultats suivants sont renvoyés :

ID	ENTRYDATE	C1
swissprot:100K_RAT	01-NOV-1997	0
swissprot:100K_RAT	01-NOV-1997	3
swissprot:100K_RAT	01-NOV-1997	4
swissprot:100K_RAT	01-NOV-1997	100
swissprot:100K_RAT	01-NOV-1997	103
swissprot:100K_RAT	01-NOV-1997	104

6 enregistrement(s) sélectionné(s).

Cependant, le comportement prévu est que seule la ligne 0 serait renvoyée.

Prédicats d'équijointure pour l'encapsuleur BioRS

Vous devez indiquer des prédicats pour le moteur BioRS lorsque vous utilisez les fonctions personnalisées BioRS, à une exception près. Cette exception est lorsque vous exécutez des opérations d'équijointure pendant une requête.

Une opération de *jointure* comprend l'extraction de données de plusieurs tables d'après des valeurs de colonne correspondantes. Une *équijointure* est une opération de jointure dont la condition de jonction a la forme *expression = expression*. Pour les requêtes BioRS, les termes d'équijointure doivent contenir l'élément *_ID_* d'une banque de données et un élément de type Référence d'une autre banque de données.

Exemple de définitions de pseudonyme et de requête d'équijointure

Cet exemple montre des exemples de définitions de pseudonyme et une requête d'équijointure qui utilise les exemples de pseudonymes.

Vous voulez interroger deux banques de données BioRS, SwissProt et SwissProt.features. La banque de données SwissProt.features est un enfant de la banque de données SwissProt et contient un élément appelé Parent. L'élément Parent contient des références à des entrées identifiées par l'élément *_ID_* de SwissProt. Vous enregistrez les deux définitions de pseudonyme pour les deux banques de données.

Définition de pseudonyme pour la banque de données SwissProt

La banque de données SwissProt est la banque de données parent.

```
CREATE NICKNAME tc600sprot (
  ID VARCHAR (32) OPTIONS (ELEMENT_NAME '_ID_'),
  AllText    VARCHAR (128),
  EntryDate  VARCHAR (128),
  Update     VARCHAR (128),
  Description VARCHAR (1200),
  Crossreference VARCHAR (32),
  Authors    VARCHAR (256),
```

```

Journal          VARCHAR (256),
JournalIssue     VARCHAR (64) OPTIONS (IS_INDEXED 'N'),
PublicationYear  VARCHAR (1024),
Gene             VARCHAR (20) OPTIONS (IS_INDEXED 'Y'),
Remarks         VARCHAR (1200),
RemarkType      CHAR (20),
CatalyticActivity VARCHAR (20),
CoFactor         VARCHAR (64),
Disease         VARCHAR (128),
Function        VARCHAR (128),
Pathway         VARCHAR (128),
Similarity      VARCHAR (128),
Complex         VARCHAR (64),
FtKey           VARCHAR (32),
FtDescription   VARCHAR (128),
FtLength        VARCHAR (256),
MolWeight       VARCHAR (64),
ProteinLen      VARCHAR (32) OPTIONS (ELEMENT_NAME 'Protein_length'),
Sequence        CLOB,
AccNumber       VARCHAR (32),
Taxonomy        VARCHAR (128),
Organelle       VARCHAR (128),
Organism        VARCHAR (128),
Keywords        VARCHAR (1200),
Localization    VARCHAR (128),
FtKey_count     VARCHAR (32))
FOR SERVER biors_server_600
  OPTIONS (REMOTE_OBJECT 'SwissProt');

```

Définition de pseudonyme pour la banque de données SwissProt.features

La banque de données SwissProt.features est une banque de données enfant de la banque de données SwissProt. Ce pseudonyme contient l'élément **Parent**.

```

CREATE NICKNAME tc600feat (
  ID          VARCHAR (32) OPTIONS (ELEMENT_NAME '_ID_'),
  AllText     VARCHAR (1200),
  FtKey       VARCHAR (32),
  FtLength    VARCHAR (64),
  FtDescription VARCHAR (128),
  Parent      VARCHAR (32) OPTIONS (REFERENCED_OBJECT 'SwissProt'))
FOR SERVER biors_server_600
  OPTIONS (REMOTE_OBJECT 'SwissProt.features');

```

La requête qui référence les deux pseudonymes d'une équijointure

Dans cette requête, deux prédicats sont appliqués au pseudonyme tc600sprot (banque de données SwissProt). Ces deux prédicats filtrent les lignes qui contiennent le terme anophèles et ayant 1997 comme année de publication. Un prédicat est appliqué au pseudonyme tc600feat (banque de données SwissProt.features) pour filtrer les lignes dont l'élément FtKey contient le terme signal. Les deux pseudonymes sont joints à l'aide du terme *f.Parent = s.ID*.

```

SELECT s.ID, f.ID, f.FtKey FROM tc600sprot s, tc600feat f
  WHERE BIORS.CONTAINS (s.AllText, 'anophèles') = 1
  AND BIORS.CONTAINS (s.PublicationYear, 1997) = 1
  AND BIORS.CONTAINS (f.FtKey, 'signal') = 1
  AND f.Parent = s.ID;

```

L'ensemble de résultats final contient uniquement les lignes qui répondent à ces critères et les entrées de la banque de données SwissProt.features qui référencent une entrée correspondante dans la banque de données SwissProt.

L'élément AllText de BioRS

Chaque banque de données du système BioRS contient un élément appelé AllText. L'élément AllText est un élément indexé que BioRS crée automatiquement pour toutes les banques de données.

Avec l'élément AllText, vous pouvez rechercher dans tout le texte d'une entrée, et pas seulement dans certains éléments indexés. Par exemple, la recherche du terme muscus peut renvoyer des entrées dans lesquelles le mot muscus apparaît dans le titre, l'extrait, la description ou l'organisme.

Pour utiliser l'élément AllText dans une requête fédérée, vous devez mapper l'élément AllText à une colonne de pseudonymes. Vous mappez l'élément AllText à une colonne de pseudonymes lorsque vous indiquez des colonnes dans l'instruction CREATE NICKNAME. Une colonne de pseudonymes qui est mappée à l'élément AllText renvoie une valeur NULL dans les instructions SELECT. Lorsque vous indiquez une colonne comme élément AllText, la colonne ne doit pas être la première colonne déclarée dans une instruction CREATE NICKNAME.

Une fois l'élément AllText correctement mappé à une colonne de pseudonymes, vous pouvez utiliser cette colonne de pseudonymes dans une fonction personnalisée BIOR.SCONTAINS.

Source de données BioRS - exemples de requêtes

Ces exemples comportent un ensemble complet d'exemples de requêtes que vous pouvez utiliser pour accéder aux sources de données BioRS et vous montrent les instructions nécessaires pour créer les pseudonymes utilisés dans les exemples.

Ces exemples vous montrent comment :

- Structurer vos requêtes pour optimiser les performances du système
- Utiliser les fonctions personnalisées et les caractères génériques dans vos requêtes
- Utiliser des requêtes pour accéder aux colonnes de types de données BioRS spécifiques
- Utiliser des prédicats relationnels pour constituer une équijointure entre les pseudonymes parent et enfant

Les exemples de requête utilisent les pseudonymes swiss et swissft.

Instruction CREATE NICKNAME pour le pseudonyme swiss

Le pseudonyme parent swiss a été enregistré pour la banque de données SwissProt à l'aide de l'instruction CREATE NICKNAME suivante :

```
CREATE NICKNAME swiss
(
  ID                CHAR (30) OPTIONS (ELEMENT_NAME '_ID_'),
  EntryDate         VARCHAR (15),
  Update            CLOB (15),
  Description        CLOB (15),
  Crossreference     CLOB (15),
  Authors            CLOB (15),
  Journal            VARCHAR (15),
  JournalIssue      VARCHAR (15),
  PublicationYear    CLOB (15),
  PublicationTitle   CLOB (15),
  Gene               CLOB (15),
  Remarks            CLOB (15),
  RemarkType        VARCHAR (15),
```

```

CatalyticActivity VARCHAR (15),
CoFactor          VARCHAR (15),
Disease           VARCHAR (15),
Function          CLOB (15),
Pathway           VARCHAR (15),
Similarity        CLOB (15),
Complex           VARCHAR (15),
FtKey             VARCHAR (15),
FtDescription     CLOB (15),
FtLength          VARCHAR (15),
MolWeight         CHAR (15),
Protein_Length   VARCHAR (15),
Sequence          CLOB (15),
AccNumber         VARCHAR (15),
Taxonomy          CLOB (15),
Organelle         VARCHAR (15),
Organism          VARCHAR (15),
Keywords          VARCHAR (15),
Localization      VARCHAR (15),
FtKey_count       VARCHAR (15),
AllText           CLOB (15)
)
FOR SERVER serveur_biors
  OPTIONS (REMOTE_OBJECT 'swissprot');

```

Instruction CREATE NICKNAME pour le pseudonyme swissft

Le pseudonyme enfant swissft a été enregistré pour la banque de données SwissProt.Features à l'aide de l'instruction CREATE NICKNAME suivante :

```

CREATE NICKNAME swissft
(
  ID          VARCHAR (30) OPTIONS (ELEMENT_NAME '_ID_'),
  FtKey       VARCHAR (15),
  FtLength    VARCHAR (15),
  FtDescription VARCHAR (15),
  Parent      VARCHAR (30) OPTIONS (REFERENCED_OBJECT 'swissprot'),
  AllText     CLOB (15)
)
FOR SERVER serveur_biors
  OPTIONS (REMOTE_OBJECT 'swissprot.features');

```

Les structures de requête ont un effet sur les performances du serveur fédéré

Les requêtes et résultats de la table suivante vous montrent comment vous pouvez structurer vos requêtes pour optimiser la charge de travail entre le système fédéré et le serveur BioRS.

Tableau 9. Exemples de différentes requêtes produisant des résultats identiques

Recherche	Résultat
<pre> SELECT s.id FROM Swiss s WHERE BIORS.CONTAINS(s.id, '100K_RAT') = 1 FETCH FIRST 3 ROWS ONLY; </pre>	<pre> Client ----- 100K_RAT 1 enregistrement sélectionné. </pre>
<pre> SELECT s.id FROM Swiss s WHERE s.id LIKE '%100K_RAT%' FETCH FIRST 3 ROWS ONLY; </pre>	<pre> Client ----- 100K_RAT 1 enregistrement sélectionné. </pre>

Les deux requêtes de cette table produisent les mêmes résultats. Cependant, la première requête est exécutée beaucoup plus rapidement que la deuxième requête. La première requête utilise la fonction BIOR.S.CONTAINS pour indiquer le prédicat d'entrée. Le résultat est que le serveur BioRS sélectionne les données dans la banque de données SwissProt, puis transmet les données sélectionnées au serveur fédéré. Dans la deuxième requête, le prédicat d'entrée LIKE est indiqué directement dans le pseudonyme swiss. Le résultat est que le serveur BioRS transfère toute la banque de données SwissProt au serveur fédéré. Une fois le contenu de la banque de données transféré, le serveur fédéré sélectionne les données.

Les performances des requêtes sont généralement bien meilleures lorsque les prédicats sont envoyés à la source de données pour traitement.

Requêtes utilisant des caractères génériques dans la fonction personnalisée BIOR.S.CONTAINS

Les requêtes et résultats de la table suivante montrent l'utilisation de caractères génériques dans la fonction personnalisée BIOR.S.CONTAINS. Tous les résultats des requêtes sont identiques, bien que des caractères génériques différents soient utilisés.

Tableau 10. Exemples de requêtes utilisant des caractères génériques dans la fonction personnalisée BIOR.S.CONTAINS

Recherche	Résultat
SELECT s.crossreference FROM Swiss s WHERE BIOR.S.CONTAINS(s.crossreference, 'MEDLINE') = 1 FETCH FIRST 3 ROWS ONLY;	CROSSREFERENCE ----- NCBI_TaxID=1011 NCBI_TaxID=5875 NCBI_TaxID=4081 3 enregistrement(s) sélectionné(s).
SELECT s.crossreference FROM Swiss s WHERE BIOR.S.CONTAINS(s.crossreference, '?ED?IN?') = 1 FETCH FIRST 3 ROWS ONLY;	CROSSREFERENCE ----- NCBI_TaxID=1011 NCBI_TaxID=5875 NCBI_TaxID=4081 3 enregistrement(s) sélectionné(s).
SELECT s.crossreference FROM Swiss s WHERE BIOR.S.CONTAINS(s.crossreference, '*D*N*') = 1 FETCH FIRST 3 ROWS ONLY;	CROSSREFERENCE ----- NCBI_TaxID=1011 NCBI_TaxID=5875 NCBI_TaxID=4081 3 enregistrement(s) sélectionné(s)

Requêtes accédant aux colonnes du type de données auteur BioRS

Les requêtes et résultats de la table suivante montrent comment vous pouvez accéder aux informations dans les éléments du type de données auteur BioRS avec la fonction personnalisée BIOR.S.CONTAINS. La syntaxe de toutes les requêtes est presque identique. La seule différence est la présence ou l'absence de la première initiale dans le terme recherché et le volume d'espace entre le prénom et la dernière initiale.

Tableau 11. Exemples de requêtes accédant aux colonnes du type de données auteur BioRS

Recherche	Résultat
SELECT s.authors FROM Swiss s WHERE BIORS.CONTAINS(s.authors, 'Mueller') = 1 FETCH FIRST 3 ROWS ONLY;	AUTHORS ----- Mueller D. Rehb Mayer K.F.X. Sc Zemmour J. Litt 3 enregistrement(s) sélectionné(s).
SELECT s.authors FROM Swiss s WHERE BIORS.CONTAINS(s.authors, 'Mueller,D') = 1 FETCH FIRST 3 ROWS ONLY;	AUTHORS ----- 0 enregistrement(s) sélectionné(s).
SELECT s.authors FROM Swiss s WHERE BIORS.CONTAINS(s.authors, 'Mueller ,D') = 1 FETCH FIRST 3 ROWS ONLY;	AUTHORS ----- 0 enregistrement(s) sélectionné(s).
SELECT s.authors FROM Swiss s WHERE BIORS.CONTAINS(s.authors, 'Mueller, D') = 1 FETCH FIRST 3 ROWS ONLY;	AUTHORS ----- Mueller D. Rehb Zou P.J. Borovo Davies J.D. Mue 3 enregistrement(s) sélectionné(s).

Requêtes accédant aux colonnes du type de données date BioRS

Les requêtes et résultats de la table suivante montrent comment vous pouvez accéder aux informations dans les éléments du type date BioRS avec la fonction personnalisée BIOR.S.CONTAINS.

Lorsqu'une zone du type date BioRS contient une séquence de dates, les résultats peuvent contenir des informations supplémentaires, comme le montre le deuxième exemple de la table. Les éléments des types de données numériques BioRS (date et nombre) peuvent contenir plusieurs valeurs. Les résultats des requêtes appliquées à des éléments de date ou à des éléments numériques BioRS peuvent donc aussi contenir plusieurs valeurs. Lorsqu'il y a plusieurs valeurs, elles sont toujours séparées par des espaces.

Tableau 12. Exemples de requêtes accédant aux colonnes du type de données date BioRS

Recherche	Résultat
SELECT e.entrydate FROM embl e WHERE BIORS.CONTAINS(e.entrydate, date('11/01/1997')) = 1 FETCH FIRST 3 ROWS ONLY;	ENTRYDATE ----- 01-NOV-1997 01-NOV-1997 01-NOV-1997 3 enregistrement(s) sélectionné(s).
SELECT g.update FROM gen g WHERE BIORS.CONTAINS(g.update, date('11/01/1997')) = 1 FETCH FIRST 3 ROWS ONLY;	UPDATE ----- 01-NOV-1997 11- 01-NOV-1997 12- 01-NOV-1997 06- 3 enregistrement(s) sélectionné(s).

Requêtes utilisant les fonctions personnalisées BIORS.CONTAINS_LE et BIORS.CONTAINS_GE

Les requêtes et résultats de la table suivante montrent comment vous pouvez utiliser les fonctions personnalisées BIORS.CONTAINS_LE et BIORS.CONTAINS_GE.

Tableau 13. Exemples de requêtes utilisant les fonctions personnalisées BIORS.CONTAINS_LE et BIORS.CONTAINS_GE

Recherche	Résultat
SELECT s.molweight FROM Swiss s WHERE BIORS.CONTAINS_LE(s.molweight, 100368) = 1 FETCH FIRST 3 ROWS ONLY;	MOLWEIGHT ----- 100368 10576 8523 3 enregistrement(s) sélectionné(s).
SELECT s.molweight FROM Swiss s WHERE BIORS.CONTAINS_GE(s.molweight, 100368) = 1 FETCH FIRST 3 ROWS ONLY;	MOLWEIGHT ----- 100368 103625 132801 3 enregistrement(s) sélectionné(s).

Tableau 13. Exemples de requêtes utilisant les fonctions personnalisées BIOR.S.CONTAINS_LE et BIOR.S.CONTAINS_GE (suite)

Recherche	Résultat
SELECT s.journalissue FROM Swiss s WHERE BIORS.CONTAINS_GE(s.journalissue, 172) = 1 FETCH FIRST 3 ROWS ONLY;	JOURNALISSUE ----- 172 21 242 196 3 enregistrement(s) sélectionné(s).

Requêtes utilisant la fonction personnalisée BIOR.S.SEARCH_TERM

Les requêtes et résultats de la table suivante montrent comment vous pouvez utiliser la fonction personnalisée BIOR.S.SEARCH_TERM pour indiquer un terme recherché à l'aide du format BioRS.

Tableau 14. Exemples de requêtes utilisant la fonction personnalisée BIOR.S.SEARCH_TERM

Recherche	Résultat
SELECT s.publicationyear FROM Swiss s WHERE BIORS.SEARCH_TERM (s.id, '[PublicationYear EQ number:1997;]')=1 FETCH FIRST 10 ROWS ONLY;	PUBLICATIONYEAR ----- 1997 1997 2000 1988 1991 1997 1994 1997 1997 1998 1994 1995 1997 1997 1999 1997 1994 1994 1995 1993 1992 1997 10 enregistrement(s) sélectionné(s).
SELECT s.molweight FROM Swiss s WHERE BIORS.SEARCH_TERM (s.id, '[MolWeight EQ number:100368;]') = 1 FETCH FIRST 10 ROWS ONLY;	MOLWEIGHT ----- 100368 100368 2 record(s) selected.

Tableau 14. Exemples de requêtes utilisant la fonction personnalisée `BIORS.SEARCH_TERM` (suite)

Recherche	Résultat
<pre>SELECT s.molweight FROM Swiss s WHERE BIORS.SEARCH_TERM (s.id, '[MolWeight GREATER number:100368;]') = 1 FETCH FIRST 10 ROWS ONLY;</pre>	<pre>MOLWEIGHT ----- 100368 103625 132801 194328 130277 287022 289130 135502 112715 112599 10 enregistrement(s) sélectionné(s).</pre>

Utilisation de prédicats relationnels pour constituer une équijointure entre deux banques de données liées par une relation parent-enfant

La requête suivante montre comment utiliser des prédicats relationnels pour constituer une équijointure entre deux banques de données liées par une relation parent-enfant :

```
SELECT s.id, f.id, f.parent FROM Swiss s, Swissft f
WHERE (f.parent = s.id) FETCH FIRST 10 ROWS ONLY;
```

Dans les résultats de requête suivants, l'enregistrement `100K_RAT` est un parent de neuf enregistrements enfant (`100K_RAT.1` à `100K_RAT.9`).

ID	ID	PARENT
100K_RAT	100K_RAT.1	swissprot:100K_RAT
100K_RAT	100K_RAT.2	swissprot:100K_RAT
100K_RAT	100K_RAT.3	swissprot:100K_RAT
100K_RAT	100K_RAT.4	swissprot:100K_RAT
100K_RAT	100K_RAT.5	swissprot:100K_RAT
100K_RAT	100K_RAT.6	swissprot:100K_RAT
100K_RAT	100K_RAT.7	swissprot:100K_RAT
100K_RAT	100K_RAT.8	swissprot:100K_RAT
100K_RAT	100K_RAT.9	swissprot:100K_RAT
104K_THEPA	104K_THEPA.1	swissprot:104K_THEPA

10 enregistrement(s) sélectionné(s).

Optimisation des performances de l'encapsuleur BioRS

Vous pouvez améliorer les performances d'interrogation des sources de données BioRS en optimisant les performances de l'encapsuleur BioRS.

Instructions pour optimiser les performances de l'encapsuleur BioRS

La structure de vos requêtes et informations statistiques sur les banques de données BioRS ont un effet sur les performances d'interrogation.

Réduisez le volume de données transféré entre moteurs de recherche.

L'environnement fédéré utilise deux moteurs de recherche. Pour l'encapsuleur BioRS, ces moteurs de recherche sont la base de données

fédérée et BioRS. Certaines configurations comportent plusieurs moteurs de base de données fédérée. Le moteur de base de données fédérée traite les prédicats (opérateurs relationnel, tels que =, BETWEEN, LIKE et <>) indiqués dans les colonnes de pseudonymes. Le moteur BioRS traite les prédicats qui sont indiqués à l'aide de quatre fonctions personnalisées pour l'encapsuleur BioRS.

Pour réduire le volume de données transféré entre les deux moteurs de recherche, organisez vos requêtes de façon à ce que le traitement de l'information soit envoyé au système BioRS chaque fois que possible.

Si vous avez besoin d'exécuter des opérations de jointure dans une requête, tirez parti des relations parent-enfant qui existent déjà dans les banques de données BioRS et exécutez des opérations d'équijointure chaque fois que possible. Les opérations d'équijointure sont traitées dans BioRS, ce qui réduit aussi le volume de données transférées entre la base de données fédérée et les moteurs de recherche BioRS.

Important : N'interrompez pas les requêtes fédérées à BioRS, par exemple à l'aide de Ctrl-D ou Ctrl-Z dans l'interpréteur de commandes ou en arrêtant un programme d'application. L'interruption d'une requête laisse des processus "morts" s'exécuter sur le serveur BioRS. Ces processus "morts" dégraderont rapidement les performances à la fois du serveur BioRS et du serveur fédéré. Si un certain nombre de ces processus "morts" s'exécutent, des erreurs imprévues peuvent se produire pendant le traitement de la requête fédérée. Par exemple, une requête valide pourrait renvoyer 0 ligne lorsque des lignes sont prévues. Dans des situations extrêmes, le serveur BioRS, le serveur fédéré ou les deux serveurs peuvent s'arrêter ou se terminer de façon anormale.

Gérez des informations statistiques BioRS dans l'environnement fédéré.

Dans un système fédéré, la base de données fédérée exploite les statistiques de catalogue sur les objets dotés de pseudonymes pour optimiser le traitement des requêtes. La gestion de statistiques actualisées sur les sources de données BioRS est essentielle pour optimiser les performances de l'encapsuleur BioRS. Si les caractéristiques de structure ou de données statistiques d'un objet distant pour lequel un pseudonyme est défini sont modifiées, vous devez mettre à jour les statistiques de cardinalité de la colonne de pseudonymes correspondante dans votre système fédéré.

Pour optimiser les performances de l'encapsuleur BioRS, exécutez ces mises à jour sur le serveur fédéré à intervalles réguliers.

Informations statistiques sur BioRS

Des informations statistiques BioRS actualisées sont essentielles pour optimiser les performances de l'encapsuleur BioRS.

Dans un système fédéré, la base de données fédérée exploite les statistiques de catalogue sur les objets dotés de pseudonymes pour optimiser le traitement des requêtes. Ces statistiques sont extraites des sources de données BioRS lorsque vous enregistrez un pseudonyme à l'aide de l'instruction CREATE NICKNAME. La base de données fédérée vérifie la présence de l'objet sur la source de données, puis tente de regrouper les données statistiques existantes de la source de données. Les informations sont lues dans les catalogues de source de données et placées dans le catalogue système de la base de données fédérée.

Pour les sources de données BioRS, les informations statistiques critiques sont les suivantes :

- La cardinalité d'un pseudonyme. Pour les sources de données BioRS, la cardinalité du pseudonyme est équivalente au nombre d'entrées dans la banque de données BioRS correspondante.
- La cardinalité de la colonne qui correspond à l'élément `_ID_` de BioRS. La cardinalité de cette colonne doit correspondre à la cardinalité du pseudonyme dans lequel la colonne est référencée.
- La cardinalité de toutes les colonnes que l'encapsuleur BioRS pourrait avoir besoin d'utiliser.

Conseil : Pour optimiser les performances de l'encapsuleur BioRS, vous devez gérer des statistiques actualisées sur les sources de données BioRS. Si les caractéristiques de structure ou de données statistiques d'un objet distant pour lequel un pseudonyme est défini sont modifiées, vous devez mettre à jour les statistiques de cardinalité correspondantes dans votre système fédéré. Les statistiques de cardinalité sont stockées dans le catalogue système de la base de données fédérée dans les vues `SYSSTAT.TABLES` et `SYSSTAT.COLUMNS`.

Pour gérer les statistiques de cardinalité BioRS dans votre système fédéré :

1. Déterminez les statistiques de cardinalité pour le pseudonyme, si nécessaire.
2. Mettez à jour les statistiques de cardinalité pour le pseudonyme dans les vues catalogue.
3. Mettez à jour les statistiques de cardinalité pour les colonnes dans les vues catalogue.

Etablissement des statistiques de cardinalité de la banque de données BioRS

Vous devez déterminer les statistiques de cardinalité de la banque de données BioRS avant de pouvoir mettre à jour les statistiques de pseudonymes ou avant de pouvoir mettre à jour la cardinalité de la colonne qui correspond à l'élément `_ID_` de BioRS.

Procédure

Pour déterminer les statistiques de cardinalité pour une banque de données spécifique dans BioRS :

Utilisez le programme utilitaire `admin_find` de BioRS ou bien le programme utilitaire `www_find.cgi` et indiquez l'option `-c`, qui s'applique à la cardinalité. Pour de plus amples informations sur ces deux programmes utilitaires de BioRS, consultez votre documentation BioRS.

Mise à jour des statistiques de cardinalité de pseudonyme BioRS

Vous devez mettre à jour les statistiques de cardinalité de pseudonyme lorsque le contenu d'une banque de données BioRS subit des modifications importantes.

Avant de commencer

Vous devez déterminer le nombre de cardinalité de la banque de données BioRS qui correspond au pseudonyme dont vous voulez mettre à jour les statistiques.

A propos de cette tâche

Pour mettre à jour les statistiques de cardinalité de pseudonyme BioRS dans votre système fédéré, vous devez modifier la vue catalogue SYSSTAT.TABLES.

La gestion de statistiques de cardinalité correctes pour les pseudonymes permet à l'optimiseur et à l'encapsuleur BioRS de choisir le plan d'accès aux données le plus performant.

Procédure

Pour mettre à jour les statistiques de cardinalité de pseudonyme BioRS :

Emettez l'instruction UPDATE pour modifier la vue catalogue SYSSTAT.TABLES et indiquer le nombre de cardinalité approprié. La syntaxe de l'instruction UPDATE est :

```
UPDATE SYSSTAT.TABLES SET CARD=numéro_cardinalité
      WHERE TABSCHEMA=schéma_pseudonyme
      AND TABNAME=nom_pseudonyme;
```

Par exemple, si le pseudonyme est JONES.SWISS, vous utiliserez l'instruction UPDATE suivante pour mettre à jour les statistiques :

```
UPDATE SYSSTAT.TABLES SET CARD=15312191
      WHERE TABSCHEMA='JONES'
      AND TABNAME='SWISS';
```

SYSSTAT.TABLES

Vue catalogue système dans la base de données fédérée dans laquelle les statistiques de pseudonymes sont stockées.

SET CARD=15312191

Nombre de cardinalité de la banque de données BioRS qui correspond au pseudonyme pour lequel vous mettez à jour les statistiques.

TABSCHEMA= 'JONES'

Nom du schéma pour le pseudonyme que vous voulez mettre à jour.

TABNAME='SWISS'

Nom du pseudonyme que vous voulez mettre à jour.

Mise à jour des statistiques de cardinalité de la colonne BioRS

Pour mettre à jour les statistiques de cardinalité de la colonne BioRS dans votre système fédéré, vous devez modifier la vue catalogue SYSSTAT.COLUMNS.

Vous pouvez mettre à jour les statistiques de cardinalité de colonne BioRS avant de créer les pseudonymes BioRS ou bien vous pouvez mettre à jour les statistiques de cardinalité de colonne BioRS lorsque vous voulez améliorer les performances d'interrogation des sources de données BioRS.

Restrictions

N'utilisez pas cette procédure pour mettre à jour les statistiques de cardinalité pour des colonnes qui correspondent à l'élément `_ID_` de BioRS.

Vous devez veiller à ce que les statistiques de cardinalité des colonnes BioRS soient actualisées de façon à ce que l'optimiseur et l'encapsuleur BioRS puissent choisir le meilleur plan d'accès aux données pendant le traitement des requêtes.

A propos de cette tâche

Procédure

Pour mettre à jour les statistiques de cardinalité de colonne BioRS :

Emettez l'instruction UPDATE pour modifier la vue catalogue SYSSTAT.COLUMNS. La syntaxe de l'instruction UPDATE est :

```
UPDATE SYSSTAT.COLUMNS SET COLCARD=(SELECT COUNT(DISTINCT nom_colonne)
    FROM schéma_pseudonyme.nom_pseudonyme)
    WHERE
    TABSCHEMA=schéma_pseudonyme
    AND TABNAME=nom_pseudonyme
    AND COLNAME=nom_colonne;
```

SYSSTAT.COLUMNS

Vue catalogue système de la base de données fédérée dans laquelle les statistiques de colonne sont stockées.

COLCARD=(SELECT COUNT(DISTINCT *nom_colonne*

Nom de la colonne dans le pseudonyme pour laquelle vous mettez à jour les statistiques.

TABSCHEMA= *schéma_pseudonyme*

Nom du schéma pour le pseudonyme que vous voulez mettre à jour.

TABNAME=*nom_pseudonyme*

Nom du pseudonyme que vous voulez mettre à jour.

COLNAME=*nom_colonne*

Nom de la colonne dont vous voulez mettre à jour les statistiques de cardinalité.

L'exécution de la requête peut prendre plusieurs minutes, car toutes les entrées de la banque de données qui sont associées au pseudonyme doivent être extraites. Si une colonne contient plusieurs valeurs (par exemple l'élément PublicationYear du format de base de données SwissProt), le calcul devient trop complexe pour utiliser une requête SQL. Pour de telles colonnes, vous devez calculer manuellement la valeur de cardinalité, puis mettre à jour la vue catalogue SYSSTAT.COLUMNS. Pour calculer la valeur de cardinalité, divisez le nombre de valeurs distinctes dans la colonne par le nombre moyen de valeurs par ligne. La valeur de cardinalité calculée ne peut pas être supérieure à la cardinalité de la table.

Par exemple, si le pseudonyme a les trois lignes de valeurs suivantes pour la colonne PublicationYear, il y a neuf valeurs distinctes et le nombre moyen de valeurs dans une ligne est de quatre.

- 1997 1992 1985
- 1997 1992 1982
- 1992 1991 1990 1976 1974 1971

La cardinalité de cette colonne PublicationYear est 9 divisé par 4, soit 3 (2,25 arrondi au prochain entier supérieur). Vous pouvez mettre à jour la vue catalogue SYSSTAT.COLUMNS à l'aide de l'instruction UPDATE suivante :

```

UPDATE SYSSTAT.COLUMNS SET CARDCOL=3
WHERE
  TABSCHEMA=schéma_pseudonyme
  AND TABNAME=nom_pseudonyme
  AND COLNAME=nom_colonne;

```

Mise à jour de la cardinalité de la colonne `_ID_` de BioRS

Pour mettre à jour les statistiques de cardinalité de colonne BioRS pour la colonne mappée à l'élément `_ID_` de BioRS, vous devez modifier la vue catalogue `SYSSTAT.COLUMNS`.

Avant de commencer

Vous devez déterminer le nombre de cardinalité de la banque de données BioRS qui correspond au pseudonyme dans lequel la colonne est référencée. Le nombre de cardinalité de la colonne mappée à l'élément `_ID_` de BioRS doit correspondre à la cardinalité du pseudonyme dans lequel la colonne est référencée.

Vous devez veiller à ce que les statistiques de cardinalité pour la colonne mappée à l'élément `_ID_` de BioRS soient actualisées. L'optimiseur et l'encapsuleur BioRS utilisent ces statistiques pour choisir le meilleur plan d'accès aux données pour traiter vos requêtes.

A propos de cette tâche

Pour mettre à jour la cardinalité de la colonne `_ID_` de BioRS, vous devez sélectionner les entrées de la vue `SYSCAT.COLOPTIONS` qui contiennent l'option `ELEMENT_NAME`. Cet encapsuleur BioRS utilise cette option pour mapper entre les noms de colonne de pseudonyme dans la base de données fédérée et les noms d'élément dans le serveur BioRS.

Procédure

Pour mettre à jour les statistiques de cardinalité de la colonne `_ID_` de BioRS :

Emettez l'instruction `UPDATE` pour modifier la vue catalogue.

Par exemple :

```

UPDATE SYSSTAT.COLUMNS SET COLCARD=nombre_cardinalité
WHERE
  TABSCHEMA=schéma_pseudonyme
  AND TABNAME=nom_pseudonyme
  AND COLNAME=nom_colonne
  IN (SELECT nom_colonne FROM SYSCAT.COLOPTIONS
      WHERE
        TABSCHEMA=schéma_pseudonyme
        AND TABNAME=nom_pseudonyme
        AND OPTION='ELEMENT_NAME';
      AND SETTING='_ID_')

```

SYSSTAT.COLUMNS

Vue catalogue système de la base de données fédérée dans laquelle les statistiques de colonne sont stockées.

SET COLCARD=nombre_cardinalité

Le nombre de cardinalité de la banque de données BioRS qui correspond au pseudonyme de la colonne pour laquelle vous mettez à jour les statistiques.

TABSCHEMA= schéma_pseudonyme

Nom du schéma pour le pseudonyme que vous voulez mettre à jour.

TABNAME=*nom_pseudonyme*

Nom du pseudonyme que vous voulez mettre à jour.

COLNAME=*nom_colonne*

Nom de la colonne dont vous voulez mettre à jour les statistiques de cardinalité.

IN (SELECT *nom_colonne* **FROM SYSCAT.COLOPTIONS**

Cette instruction SELECT détermine le nom de la colonne mappée à l'élément *_ID_* de BioRS. SYSCAT.COLOPTIONS est la vue catalogue système de la base de données fédérée dans laquelle les options de colonne sont stockées.

OPTION='ELEMENT_NAME'

La valeur des lignes de la vue SYSCAT.COLOPTIONS qui indique qu'un nom de colonne de pseudonyme est mappé au nom d'élément BioRS.

SETTING='_ID_'

Indique que la colonne pour l'option ELEMENT_NAME est '_ID_'.

Configuration de l'accès aux sources de données DB2

Pour configurer un serveur fédéré en vue d'accéder aux sources de données de la famille DB2, vous devez fournir au serveur fédéré des informations sur les sources de données et sur les objets auxquels vous voulez accéder.

Avant de commencer

- Vérifiez la configuration du serveur fédéré.

Procédure

Pour configurer le serveur fédéré en vue d'accéder aux sources de données DB2, procédez comme suit :

1. Cataloguez une entrée de noeud DB2.
2. Cataloguez la base de données DB2 distante.
3. Enregistrez l'encapsuleur DB2.
4. Enregistrez les définitions de serveur pour une source de données DB2.
5. Créez les mappages utilisateur pour une source de données DB2.
6. Testez la connexion au serveur de source de données DB2.
7. Enregistrez les pseudonymes des tables et des vues DB2.

Catalogage d'une entrée de noeud DB2

Vous devez cataloguer une entrée de noeud indiquant le protocole utilisé par le serveur fédéré pour se connecter à la source de données DB2.

Procédure

Pour cataloguer une entrée de noeud, procédez comme suit :

Exécutez la commande CATALOG TCPIP NODE.

Par exemple :

```
CATALOG TCPIP NODE noeud_db2 REMOTE system42 SERVER db2tcp42
```

où :

- *noeud_db2* correspond au nom attribué au noeud

- *system42* correspond au nom d'hôte du système sur lequel réside la source de données
- *db2tcp42* correspond au nom de service ou au numéro de port principal de l'instance du gestionnaire de bases de données du serveur

Catalogage de la base de données DB2 distante

Pour identifier la base de données à laquelle le serveur fédéré se connecte, vous devez cataloguer la base de données DB2 distante dans le répertoire système des bases de données du serveur fédéré.

Procédure

Pour cataloguer la base de données distante, procédez comme suit :

1. Pour cataloguer la base de données, utilisez le programme d'aide à la configuration ou exécutez la commande CATALOG DATABASE.

Par exemple :

```
CATALOG DATABASE DB2DB390 AS CLIENTS390 AT NODE DB2NODE AUTHENTICATION SERVER
```

où :

- *DB2DB390* correspond au nom de la base de données distante à cataloguer
 - *CLIENTS390* correspond à l'alias de la base de données distante en cours de catalogage. Si vous ne spécifiez pas d'alias, le gestionnaire de base de données utilise comme alias le nom de la base de données distante.
 - *DB2NODE* correspond au nom du noeud précédemment catalogué
 - *AUTHENTICATION SERVER* indique que l'authentification se déroule sur le noeud de source de données DB2
2. Si le nom de la base de données distante comporte plus de huit caractères, exécutez la commande CATALOG DCS DATABASE pour créer une entrée de répertoire DCS.

Par exemple :

```
CATALOG DCS DATABASE SALES400 AS SALES_DB2DB400
```

où :

- *SALES400* correspond à l'alias de la base de données distante à cataloguer. L'alias doit correspondre au nom d'une entrée du répertoire système des bases de données du serveur fédéré associé au noeud distant. L'alias est identique au nom spécifié dans la commande CATALOG DATABASE.
- *SALES_DB2DB400* correspond au nom de la base de données de l'hôte cible à cataloguer

Enregistrement de l'encapsuleur DB2

Pour accéder aux sources de données de la famille DB2, vous devez enregistrer un encapsuleur. Le serveur fédéré utilise l'encapsuleur pour communiquer et extraire les données des sources de données. Un encapsuleur est implémenté sous la forme d'un ensemble de fichiers de bibliothèque.

Procédure

Pour enregistrer un encapsuleur, procédez comme suit :

Emettez l'instruction CREATE WRAPPER et spécifiez le nom de l'encapsuleur. Le nom de l'encapsuleur par défaut pour les sources de données de la famille DB2 est DRDA.

Par exemple :

```
CREATE WRAPPER DRDA
```

Lorsque vous utilisez le nom par défaut pour enregistrer l'encapsuleur, vous n'êtes pas tenu de spécifier le nom de la bibliothèque puisque le serveur fédéré utilise automatiquement le nom de la bibliothèque par défaut associée à l'encapsuleur. Si le nom de l'encapsuleur entre en conflit avec un nom d'encapsuleur existant dans la base de données fédérée, vous pouvez remplacer le nom d'encapsuleur par défaut par le nom de votre choix. Toutefois, si vous procédez ainsi, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER.

Par exemple, pour enregistrer un encapsuleur intitulé encapsuleur_db2 sur un serveur fédéré qui utilise le système d'exploitation AIX, émettez l'instruction suivante :

```
CREATE WRAPPER encapsuleur_db2 LIBRARY 'libdb2drda.a'
```

Le nom de bibliothèque par défaut est spécifique au système d'exploitation du serveur fédéré. Pour plus d'informations, consultez la liste des fichiers de bibliothèque d'encapsuleur DB2.

Fichiers de bibliothèque d'encapsuleur DB2

Lors de l'installation du serveur fédéré, les fichiers de bibliothèque d'encapsuleur sont ajoutés au chemin de répertoire par défaut.

Si vous n'utilisez pas le nom d'encapsuleur par défaut lors de l'enregistrement d'un encapsuleur, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER et préciser le nom de fichier de bibliothèque d'encapsuleur par défaut.

Ce tableau répertorie les chemins de répertoire par défaut et les noms des fichiers de bibliothèque d'encapsuleur par défaut. Dans le tableau, *chemin_installation* correspond au chemin du répertoire sur lequel le serveur de fédération est installé sous UNIX ou Linux, et *%DB2PATH%* correspond à la variable d'environnement spécifiant le chemin de répertoire sur lequel le serveur de fédération est installé sous Microsoft Windows. Le chemin de répertoire Windows par défaut est C:\Program Files\IBM\sqlib.

Tableau 15. Noms des chemins de répertoires et des fichiers de bibliothèque DB2 par système d'exploitation

Système d'exploitation	Chemin de répertoire	Nom de fichier de bibliothèque
AIX	/usr/opt/chemin_installation/lib32/ /usr/opt/chemin_installation/lib64/	libdb2drda.a
HP-UX	/opt/IBM/db2/chemin_installation/lib32/ /opt/IBM/db2/chemin_installation/lib64	libdb2drda.so
Linux	/opt/IBM/db2/chemin_installation/lib32/ /opt/IBM/db2/chemin_installation/lib64	libdb2drda.so

Tableau 15. Noms des chemins de répertoires et des fichiers de bibliothèque DB2 par système d'exploitation (suite)

Système d'exploitation	Chemin de répertoire	Nom de fichier de bibliothèque
Solaris	/opt/IBM/db2/ <i>chemin_installation</i> /lib32 /opt/IBM/db2/ <i>chemin_installation</i> /lib64	db2drda.so
Windows	%DB2PATH%\bin	db2drda.dll

Pour chaque système d'exploitation, trois fichiers de bibliothèque sont installés : un fichier de bibliothèque par défaut et deux fichiers supplémentaires utilisés uniquement avec des options d'encapsuleur spécifiques.

Enregistrement de définitions de serveur pour les sources de données DB2

Pour se connecter à chaque serveur DB2, le serveur fédéré nécessite les informations relatives aux droits et au mot de passe . Dans la mesure où ces informations ne sont pas conservées dans le catalogue global, vous devez les incorporer dans chaque définition de serveur.

Procédure

Pour enregistrer une définition de serveur, utilisez l'une des méthodes suivantes :

- Utilisez l'assistant des objets fédérés du centre de contrôle DB2 Universal Database. Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés.
- Emettez l'instruction CREATE SERVER.

Lorsque vous enregistrer le serveur, vous devez inclure certaines options de serveur obligatoires. Cet exemple comprend uniquement les options de serveur requises pour l'enregistrement d'un serveur DB2 :

```
CREATE SERVER nom_de_définition_de_serveur TYPE type_de_serveur
VERSION numéro_de_version WRAPPER DRDA
AUTHORIZATION "IDutilisateur" PASSWORD "motdepasse"
OPTIONS (DBNAME 'nom_de_basededonnées')
```

L'option de serveur DBNAME est obligatoire. La valeur de DBNAME correspond à l'alias de la base de données DB2 à laquelle vous voulez accéder. L'alias est défini lors du catalogage de la base de données.

Remarque : pour l'option VERSION, si vous utilisez DB2 for z/OS Version 8 afin de créer la base de données en mode compatibilité, vous devez indiquer la version 7.

Lors de l'enregistrement du serveur, vous pouvez spécifier des options de serveur supplémentaires dans l'instruction CREATE SERVER. Ces options incluent les options de serveur générales et les options de serveur spécifiques à la source de données. Pour plus d'informations, consultez les informations de référence des options DB2.

Une fois le serveur enregistré, utilisez l'instruction ALTER SERVER pour ajouter ou supprimer des options de serveur

Instruction CREATE SERVER - Exemples pour l'encapsuleur DB2

L'instruction CREATE SERVER permet d'enregistrer les définitions de serveur DB2. Cette rubrique inclut un exemple complet avec les options obligatoires, et un exemple présentant l'utilisation des options de serveur supplémentaires.

Exemple complet

L'exemple suivant présente la procédure d'enregistrement d'une définition de serveur pour un encapsuleur DRDA à l'aide de l'instruction CREATE SERVER :

```
CREATE SERVER DB2SERVER TYPE DB2/ZOS VERSION 7 WRAPPER DRDA
  AUTHORIZATION "spalten" PASSWORD "db2guru"
  OPTIONS (DBNAME 'CLNTS390')
```

DB2SERVER

Nom attribué au serveur de base de données DB2. Les noms des définitions de serveur en double ne sont pas autorisés. Cette option de serveur est obligatoire.

TYPE DB2/ZOS

Spécifie le type de serveur de source de données pour lequel vous configurez l'accès.

VERSION 7

Version du serveur de base de données DB2 auquel vous voulez accéder.

Remarque : si vous utilisez DB2 for z/OS version 8 pour créer la base de données en mode compatibilité, vous devez indiquer la version 7.

WRAPPER DRDA

Nom spécifié dans l'instruction CREATE WRAPPER.

AUTHORIZATION "spalten"

Identificateur utilisateur au niveau de la source de données. Cet identificateur doit disposer du droit BINDADD sur la source de données. Cette valeur fait la distinction entre les majuscules et les minuscules.

PASSWORD "db2guru"

Mot de passe associé à l'identificateur utilisateur au niveau de la source de données. Cette valeur fait la distinction entre les majuscules et les minuscules.

DBNAME 'CLNTS390'

Alias de la base de données DB2 à laquelle vous voulez accéder. Vous avez défini cet alias au moment du catalogage de la base de données avec la commande CATALOG DATABASE. Cette valeur fait la distinction entre les majuscules et les minuscules.

Bien que la variable du nom de base de données soit spécifiée en tant qu'option de l'instruction CREATE SERVER, elle est obligatoire pour les sources de données DB2.

Exemple avec les options de serveur

Lors de l'enregistrement de la définition de serveur, vous pouvez spécifier des options de serveur supplémentaires dans l'instruction CREATE SERVER. Ces options incluent les options de serveur générales et les options de serveur DB2 spécifiques aux sources de données. Pour en savoir plus, consultez les informations relatives aux références d'option.

L'option CPU_RATIO indique dans quelle mesure la vitesse d'exécution de l'unité centrale de la source de données est plus élevée ou plus faible que celle de l'unité centrale fédérée. Si vous définissez l'option CPU_RATIO sur '0.001', l'unité centrale au niveau de la source de données distante a une capacité disponible 1000 fois supérieure à l'unité centrale du serveur fédéré.

L'option SAME_DECFLT_ROUNDING indique si le mode d'arrondi du serveur fédéré et du serveur distant utilise le même paramètre DECFLOAT. Si vous définissez l'option SAME_DECFLT_ROUNDING sur 'Y', le paramètre DECFLOAT de mode d'arrondi est alors identique pour les deux serveurs.

Par exemple :

```
CREATE SERVER DB2SERVER TYPE DB2/CS VERSION 9.7 WRAPPER DRDA
  AUTHORIZATION "spalten" PASSWORD "db2guru"
  OPTIONS (DBNAME 'CLNTS390', CPU_RATIO '0.001', SAME_DECFLT_ROUNDING 'Y')
```

Création de mappages utilisateur pour les sources de données DB2

Un mappage utilisateur définit une association entre un ID utilisateur et un mot de passe sur le serveur fédéré et l'ID utilisateur et le mot de passe correspondants sur le serveur de source de données.

La configuration de l'environnement fédéré détermine la nécessité, ou non, de mappages utilisateur pour les sources de données DB2. Si l'environnement utilise des contextes sécurisés fédérés et une authentification proxy, aucun ou seulement quelques mappages peuvent être requis. Pour de meilleurs résultats, prévoyez et configurez des contextes sécurisés fédérés avant de créer des mappages utilisateur.

Procédure

Pour mapper l'ID utilisateur local sur l'ID utilisateur et le mot de passe du serveur DB2, procédez comme suit :

Emettez une instruction CREATE USER MAPPING.

Par exemple :

```
CREATE USER MAPPING FOR IDutilisateur_local SERVER nom_de_definition_de_serveur
  OPTIONS (REMOTE_AUTHID 'IDutilisateur_distant', REMOTE_PASSWORD
'motdepasse_distant')
```

REMOTE_AUTHID correspond à l'identificateur utilisateur de connexion, pas à l'identificateur de liaison.

Instruction CREATE USER MAPPING - Exemples pour l'encapsuleur DB2

L'instruction CREATE USER MAPPING permet de mapper un identificateur utilisateur de serveur fédéré avec un ID utilisateur et un mot de passe de serveur DB2.

Exemple complet

L'exemple suivant présente la procédure de mappage d'un identificateur utilisateur de serveur fédéré avec un ID utilisateur et un mot de passe DB2 distants :

```
CREATE USER MAPPING FOR ALONZO SERVER DB2SERVER
  OPTIONS (REMOTE_AUTHID 'al', REMOTE_PASSWORD 'day2night')
```


ALONZO

Mappe l'identificateur utilisateur local avec l'ID utilisateur et le mot de passe distants.

SERVER DB2SERVER

Indique le nom du serveur de source de données de la famille DB2 défini dans l'instruction CREATE SERVER.

REMOTE_AUTHID 'al'

Indique l'ID utilisateur de connexion au serveur de source de données de la famille DB2 avec lequel vous mappez ALONZO. La valeur est sensible à la casse, sauf si vous définissez l'option de serveur FOLD_ID sur 'U' ou 'L' dans l'instruction CREATE SERVER.

REMOTE_PASSWORD 'day2night'

Indique le mot de passe associé à 'al'. La valeur est sensible à la casse, sauf si vous définissez l'option de serveur FOLD_PW sur 'U' ou 'L' dans l'instruction CREATE SERVER.

Exemple de registre spécial

Voici un exemple d'instruction CREATE USER MAPPING comportant le registre spécial USER :

```
CREATE USER MAPPING FOR USER SERVER DB2SERVER
  OPTIONS (REMOTE_AUTHID 'al', REMOTE_PASSWORD 'day2night')
```

Vous pouvez utiliser le registre spécial DB2 USER pour mapper l'identificateur utilisateur de la personne à l'origine de l'émission de l'instruction CREATE USER MAPPING avec l'ID utilisateur de source de données spécifié dans l'option utilisateur REMOTE_AUTHID.

Test de la connexion au serveur de source de données DB2

Le test de la connexion au serveur de source de données DB2 permet de déterminer si le serveur fédéré est correctement configuré pour accéder au serveur de source de données DB2.

Pour tester la connexion au serveur DB2, ouvrez une session passe-système et émettez une instruction SQL SELECT sur les tables système DB2.

Source de données DB2	Exemple
DB2 for z/OS	SET PASSTHRU <i>nom_définition_serveur</i> SELECT count(*) FROM sysibm.systables SET PASSTHRU RESET
DB2 for Linux, UNIX et Windows	SET PASSTHRU <i>nom_définition_serveur</i> SELECT count(*) FROM syscat.systables SET PASSTHRU RESET
DB2 for System i	SET PASSTHRU <i>nom_définition_serveur</i> SELECT count(*) FROM qsys2.systables SET PASSTHRU RESET

Si l'instruction SQL SELECT renvoie un nombre, l'accès à la source de données est correctement configuré.

Identification et résolution des erreurs de connexion à la source de données

Une connexion de test au serveur de source de données peut renvoyer une erreur pour plusieurs raisons. Il y a des mesures que vous pouvez prendre pour déterminer pourquoi l'erreur s'est produite.

Symptôme

Une erreur est renvoyée lorsque vous essayez de vous connecter à la source de données.

Cause

Il y a plusieurs causes possibles à un incident de connexion.

Résolution de l'incident

Pour identifier et résoudre les erreurs de connexion à la source de données, recherchez les problèmes éventuels avec les éléments suivants :

- Vérifiez que la source de données est disponible.
- S'il y a lieu, vérifiez que le serveur de source de données est configuré pour les connexions entrantes.
- Vérifiez que les paramètres de votre mappage utilisateur pour les options `REMOTE_AUTHID` et `REMOTE_PASSWORD` sont valides pour les connexions à la source de données. Modifiez le mappage utilisateur ou créez un autre mappage utilisateur si nécessaire.
- S'il y a lieu, vérifiez que le logiciel client de la source de données sur le serveur fédéré est installé et configuré correctement pour se connecter à la source de données.
- Pour les sources de données ODBC, vérifiez que le pilote ODBC sur le serveur fédéré est installé et configuré correctement pour se connecter au serveur de source de données ODBC. Sur les serveurs fédérés qui fonctionnent sous Windows, utilisez l'outil Administrateur de sources de données ODBC pour vérifier le pilote. Sur les serveurs fédérés qui fonctionnent sous UNIX, consultez la documentation du fournisseur de client ODBC.
- Vérifiez que les valeurs des variables définies sur le serveur fédéré sont appropriées à la source de données. Ces variables sont les variables d'environnement système, les variables du fichier `db2dj.ini` et les variables du registre de profil DB2 (`db2set`).
- Vérifiez votre définition de serveur. Si nécessaire, supprimez la définition de serveur et recréez-la.

Enregistrement de pseudonymes pour les tables et les vues DB2

Pour chaque définition de serveur DB2, enregistrez un pseudonyme pour chacune des tables et des vues auxquelles vous voulez accéder. Ensuite, utilisez les pseudonymes au lieu des noms des objets de source de données lorsque vous interrogez la base de données DB2.

Avant de commencer

Avant d'enregistrer un pseudonyme, utilisez la commande DB2 RUNSTATS pour mettre à jour les statistiques sur la source de données DB2. Le serveur fédéré utilise les statistiques de la source de données pour optimiser le traitement des requêtes.

Restrictions

Vous ne pouvez pas créer de pseudonyme sur un alias de base de données DB2.

Procédure

Pour enregistrer un pseudonyme, procédez comme suit :

Emettez l'instruction CREATE NICKNAME.

Par exemple :

```
CREATE NICKNAME pseudonyme FOR nom_définition_serveur."schéma_distant"."table.distante"
```

Lorsque vous créez le pseudonyme, le serveur fédéré interroge le catalogue de la source de données à l'aide du pseudonyme. Cette requête teste la connexion à la source de données. Si la connexion ne fonctionne pas, un message d'erreur s'affiche.

Instruction CREATE NICKNAME - Exemples pour les sources de données DB2

L'instruction CREATE NICKNAME et les options de pseudonyme obligatoires permettent d'enregistrer un pseudonyme pour une table ou une vue DB2 à laquelle vous voulez accéder.

Cet exemple présente l'utilisation des options obligatoires pour l'instruction CREATE NICKNAME. Vous pouvez également inclure des options de colonne et des options de pseudonyme supplémentaires.

```
CREATE NICKNAME DB2SALES FOR DB2SERVER.VINNIE.EUROPE
```

DB2SALES

Pseudonyme unique identifiant la table ou la vue DB2. Le pseudonyme peut comporter un schéma et le pseudonyme. Si vous omettez le schéma, l'identificateur de l'utilisateur à l'origine de l'enregistrement du pseudonyme est utilisé.

DB2SERVER.VINNIE.EUROPE

Identificateur en trois parties de l'objet distant :

- *DB2SERVER* correspond au nom que vous avez attribué au serveur de base de données DB2 dans l'instruction CREATE SERVER.
- *VINNIE* correspond à l'ID utilisateur du propriétaire de la table ou de la vue. Cette valeur fait la distinction entre les majuscules et les minuscules.
- *EUROPE* correspond au nom de la table ou de la vue distante à laquelle vous voulez accéder.

Configuration de l'accès aux sources de données Excel

Vous pouvez intégrer les données qui sont dans des sources de données Excel avec des informations d'autres sources à l'aide d'un système fédéré.

Procédure

Pour configurer un serveur fédéré pour accéder aux sources de données Excel, vous devez fournir au serveur fédéré des informations sur les sources de données et les objets auxquels vous voulez accéder. Après avoir configuré le serveur fédéré, vous pouvez créer des requêtes pour accéder aux sources de données Excel.

Encapsuleur Excel

Un classeur Excel est un fichier qui est créé à l'aide de l'application Microsoft Excel et qui a une extension de fichier xls. L'encapsuleur Excel est utilisé pour exécuter des recherches dans les fichiers Excel.

Vous utilisez des fichiers Excel pour stocker des informations qui se présentent mieux sous forme d'une table, avec des lignes et des colonnes correspondantes. Les classeurs Excel se composent d'une ou plusieurs feuilles de calcul ou *feuilles de travail*. Les feuilles de calcul sont souvent utilisées pour effectuer des calculs.

La figure suivante montre comment l'encapsuleur Excel connecte vos feuilles de calcul au système fédéré.

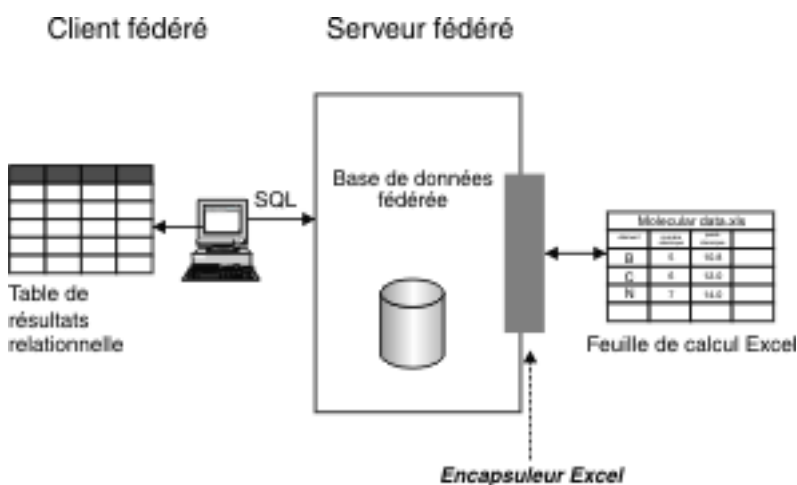


Figure 3. Fonctionnement de l'encapsuleur Excel

L'encapsuleur Excel utilise l'instruction CREATE NICKNAME pour mapper les colonnes de vos feuilles de calcul Excel aux colonnes de votre système fédéré. La table suivante montre un exemple de données de feuille de calcul qui sont stockées dans un fichier appelé Compound_Master.xls.

Tableau 16. Exemple de feuille de calcul pour Compound_Master.xls

	A	B	C	D
1	COMPOUND_NAME	WEIGHT	MOL_COUNT	WAS_TESTED
2	compound_A	1.23	367	testé
3	compound_G		210	
4	compound_F	0.000425536	174	testé

Tableau 16. Exemple de feuille de calcul pour *Compound_Master.xls* (suite)

	A	B	C	D
5	compound_Y	1.00256		testé
6	compound_Q		1024	
7	compound_B	33.5362		
8	compound_S	0.96723	67	testé
9	compound_O	1.2		testé

Normalement, vous ne pouvez pas accéder aux informations d'une feuille de calcul Excel à l'aide des commandes SQL standard. Si l'encapsuleur Excel est installé et enregistré sur votre serveur fédéré, vous pouvez accéder à ces informations comme si c'était une source de données relationnelle classique. Par exemple, si vous vouliez connaître toutes les données composées dans lesquelles le comptage moléculaire est supérieur à 100, vous exécuteriez la requête SQL suivante :

```
SELECT * FROM compound_master WHERE mol_count > 100
```

Les résultats de la requête sont affichés dans la table suivante.

Tableau 17. Résultats de la requête

COMPOUND_NAME	WEIGHT	MOL_COUNT	WAS_TESTED
compound_A	1.23	367	testé
compound_G		210	
compound_F	0.000425536	174	testé
compound_Q		1024	

Méthodes d'accès aux données Excel

Vous pouvez accéder aux données des feuilles de calcul Microsoft Excel à l'aide de l'encapsuleur Excel ou de l'encapsuleur ODBC.

Pour interroger des données Excel, les deux encapsuleurs nécessitent un serveur fédéré qui puisse ouvrir et lire les feuilles de calcul du classeur Excel. Le classeur Excel doit donc être sur le même ordinateur que le serveur fédéré ou sur une unité accessible du réseau.

Si vous utilisez l'encapsuleur Excel, l'application Excel doit être installée sur le serveur fédéré.

Si vous utilisez l'encapsuleur ODBC, le pilote ODBC Excel doit être sur le serveur fédéré. Ce pilote est installé automatiquement avec Microsoft Windows®. Il n'est pas nécessaire que l'application Excel soit installée sur le serveur fédéré.

Chaque encapsuleur impose certaines exigences pour l'emplacement et la mise en page des données dans les classeurs Excel. Avec l'encapsuleur Excel, seules les données de la première feuille de calcul du classeur sont accessibles. Avec l'encapsuleur ODBC, vous pouvez accéder aux données à partir de n'importe quelle feuille de calcul du classeur.

Les exemples suivants montrent les exigences de mise en page de la feuille de calcul pour ces deux encapsuleurs.

Exemple de feuille de calcul contenant des lignes d'étiquettes et une formule

Cet exemple montre une feuille de calcul qui contient plusieurs lignes d'étiquettes en haut de la feuille de calcul, des lignes vides et une formule dans la ligne 13. Pour accéder aux données de la feuille de calcul, vous devez identifier la plage de cellules à laquelle vous voulez accéder.

	A	B	C	D
1	Analyse du composé			
2				
3	Nom du composé	Weight	Nombre moléculaire	Testé ?
4	compound_A	1.23	367	testé
5	compound_G		210	
6	compound_F	0.000425536	174	testé
7	compound_Y	1.000256		testé
8	compound_Q		1024	
9	compound_B	33.5362		
10	compound_S	0.96723	67	testé
11	compound_O	1.2		testé
12				
13		Total composés testés		5

Figure 4. Feuille de calcul contenant plusieurs lignes d'étiquettes et une formule

Si vous utilisez l'encapsuleur Excel

Vous indiquez la plage de cellules dans l'instruction CREATE NICKNAME à l'aide de l'option RANGE. Incluez uniquement les données dans la plage que vous indiquez. N'incluez pas les étiquettes de colonne dans la plage. Les cellules qui contiennent des formules, telles que SUM, renvoient le résultat de la formule et non la formule. A moins que vous ne vouliez faire renvoyer les résultats de formule, n'incluez pas dans la plage les cellules qui contiennent des formules. Dans cet exemple, la plage de cellules que vous incluez dans l'option RANGE est A4:D11.

Si vous utilisez l'encapsuleur ODBC

Vous devez créer un nom pour la plage de cellules afin de désigner explicitement l'emplacement des données à l'intérieur de la feuille de calcul. Sous Excel, cette plage de cellules est appelée une *plage nommée*. Le pilote ODBC Excel reconnaît une seule ligne d'étiquettes, la première ligne de la plage. Aucune ligne vide n'est autorisée entre les étiquettes et les données. La plage nommée doit comporter une seule ligne d'étiquettes de colonne. Vous indiquez la plage nommée dans l'instruction CREATE NICKNAME. Vous devez inclure une ligne d'étiquettes de colonne dans la plage que vous nommez. Si vous n'incluez pas une ligne d'étiquettes de colonne dans la plage nommée, la première ligne de données est traitée

comme des étiquettes de colonne. Les cellules qui contiennent des formules, telles que SUM, renvoient le résultat de la formule et non la formule. A moins que vous ne vouliez faire renvoyer les résultats de formule, n'incluez pas dans la plage les cellules qui contiennent des formules. Dans cet exemple, la plage de cellules que vous nommez est A3:D11.

Exemple de feuille de calcul contenant une ligne d'étiquettes

Cet exemple montre une feuille de calcul qui contient une seule ligne d'étiquettes de colonne en haut de la feuille de calcul. La mise en page ne comporte pas de lignes supplémentaires avec des étiquettes, des lignes vides, ni de cellules avec des formules.

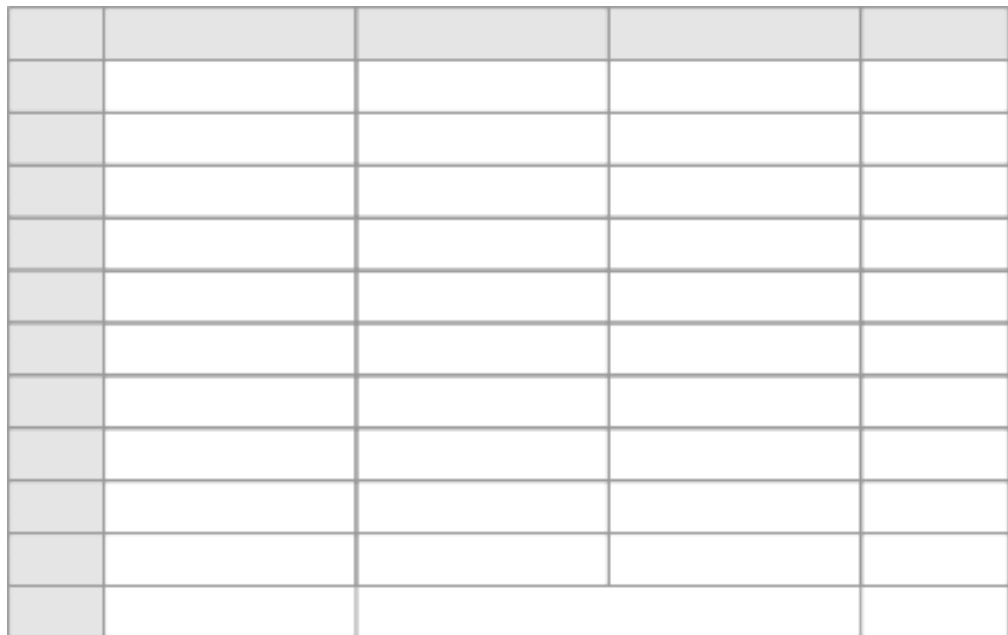


Figure 5. Feuille de calcul contenant une ligne d'étiquettes de colonne dans la ligne 1

Si vous utilisez l'encapsuleur Excel

Vous devez indiquer la plage de cellules dans l'instruction CREATE NICKNAME à l'aide de l'option RANGE. La plage ne peut pas comporter d'étiquettes de colonne dans la ligne 1. La plage de cellules que vous indiqueriez est A2:D9.

Si vous utilisez l'encapsuleur ODBC

Vous pouvez accéder à ces données sans créer de plage nommée. Vous indiquez le nom de la feuille de calcul dans l'instruction CREATE NICKNAME. L'encapsuleur lit la première ligne non vide comme des étiquettes et utilise les informations comme noms de colonne pour le pseudonyme. Les lignes ultérieures sont lues comme des données.

Exemple de feuille de calcul contenant uniquement des données

Cet exemple montre une feuille de calcul qui contient uniquement des données. Il n'y a aucune ligne d'étiquettes de colonne, aucune ligne vide et aucune cellule avec des formules.

	compound_A	1.23	367	testé
	compound_G		210	
	compound_F	0.000425536	174	testé
	compound_Y	1.000256		testé
	compound_Q		1024	
	compound_B	33.5362		
	compound_S	0.96723	67	testé
	compound_O	1.2		testé

Figure 6. Feuille de calcul contenant uniquement des données

Si vous utilisez l'encapsuleur Excel

Si les données sont dans la première feuille de calcul du classeur, l'encapsuleur accédera aux données sans utiliser l'option RANGE. Si les données sont dans une autre feuille de calcul du classeur, vous devez indiquer l'option RANGE dans l'instruction CREATE NICKNAME.

Si vous utilisez l'encapsuleur ODBC

Lorsque vous utilisez l'encapsuleur ODBC pour accéder à des données Excel, l'encapsuleur est limité par ce qui est pris en charge par le pilote ODBC Excel. Le pilote ODBC Excel exige un format spécifique de feuille de calcul. Le pilote suppose que la première ligne non vide contient les étiquettes de colonne. Si la première ligne non vide contient des données, les données dans cette ligne sont traitées comme les étiquettes de colonne des autres données. Si la feuille de calcul ne contient pas de ligne d'étiquettes de colonne, la première ligne est utilisée comme étiquettes et non comme des données. Vous perdez donc la première ligne de données. Vous pouvez contourner cette exigence en modifiant votre feuille de calcul. Insérez une nouvelle ligne avant les données et ajoutez des étiquettes pour chaque colonne de données, de façon à obtenir une présentation comme dans l'exemple qui contient une ligne d'étiquettes.

Ajout de sources de données Excel à un serveur fédéré

Pour configurer un serveur fédéré pour accéder aux sources de données Excel, vous devez fournir au serveur fédéré des informations sur les sources de données et les objets auxquels vous voulez accéder.

Avant de commencer

- Les données des feuilles de calcul Excel doivent être structurées correctement pour que l'encapsuleur Excel puisse accéder aux données.
- La fédération doit être installée sur un serveur qui fera office de serveur fédéré.
- Il doit y avoir une base de données sur le serveur fédéré.

Restrictions

- L'encapsuleur Excel est disponible uniquement pour les versions du système d'exploitation Microsoft Windows qui sont prises en charge par le serveur fédéré.
- L'application Excel doit être installée sur le serveur fédéré.
- Le classeur Excel doit être sur le même ordinateur que le serveur fédéré ou sur une unité accessible du réseau.
- Seules les données de la première feuille de calcul du classeur Excel sont accessibles avec l'encapsuleur Excel.
- La page de codes définie sur la base de données fédérée doit correspondre au jeu de caractères du fichier Excel, sinon vos requêtes risquent de produire des résultats inattendus.
- Les sessions passe-système ne sont pas autorisées.

A propos de cette tâche

Vous pouvez configurer un serveur fédéré pour accéder aux données qui sont stockées dans des sources de données Excel à l'aide du Centre de contrôle ou en émettant des instructions SQL sur la ligne de commande. Le Centre de contrôle comporte un assistant pour vous guider à travers les étapes nécessaires pour configurer les objets fédérés obligatoires.

Procédure

Pour ajouter les sources de données Excel à un serveur fédéré :

1. «Enregistrement de l'encapsuleur Excel».
2. «Enregistrement de la définition de serveur pour une source de données Excel», à la page 72.
3. «Enregistrement de pseudonymes pour des sources de données Excel», à la page 73.

Enregistrement de l'encapsuleur Excel

Vous devez enregistrer un encapsuleur pour accéder aux sources de données Excel.

A propos de cette tâche

Les encapsuleurs sont utilisés par les serveurs fédérés pour dialoguer avec les sources de données et extraire des données des sources de données. Les encapsuleurs sont mis en oeuvre comme un ensemble de fichiers de bibliothèque.

Vous pouvez enregistrer un encapsuleur à l'aide du Centre de contrôle ou à l'aide de la ligne de commande. Le Centre de contrôle comporte un assistant pour vous guider à travers les étapes nécessaires pour enregistrer l'encapsuleur.

Procédure

Pour enregistrer l'encapsuleur Excel :

Choisissez la méthode que vous voulez utiliser pour enregistrer l'encapsuleur Excel :

Méthode	Procédure.
Utilisation du Centre de contrôle	Démarrez l'assistant des objets fédérés. Cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée et cliquez sur Créer des objets fédérés .
Utilisation de la ligne de commande	Emettez l'instruction CREATE WRAPPER. Par exemple : <pre>CREATE WRAPPER encapsuleur_excel LIBRARY 'db2lsxls.dll';</pre> Vous devez indiquer le paramètre LIBRARY dans l'instruction CREATE WRAPPER.

Fichiers de bibliothèque de l'encapsuleur Excel :

Les fichiers de bibliothèque de l'encapsuleur Excel sont ajoutés au serveur fédéré lorsque vous installez le serveur fédéré.

Lorsque vous installez le serveur fédéré, trois fichiers de bibliothèque sont ajoutés dans le chemin de répertoire par défaut. Par exemple, si le serveur fédéré fonctionne sous Windows, les fichiers de bibliothèque de l'encapsuleur qui sont ajoutés dans le chemin de répertoire sont db2lsxls.dll, db2lsxlsF.dll et db2lsxlsU.dll. Le fichier de bibliothèque de l'encapsuleur par défaut est db2lsxls.dll. Les autres fichiers de bibliothèque de l'encapsuleur sont utilisés avec des options spécifiques de l'encapsuleur.

Lorsque vous enregistrez l'encapsuleur Excel, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER et indiquer le nom de fichier de bibliothèque de l'encapsuleur par défaut.

Le chemin du répertoire par défaut et le nom de fichier de bibliothèque par défaut de l'encapsuleur sont indiqués dans la table suivante.

Tableau 18. Emplacement et nom de fichier de la bibliothèque de l'encapsuleur Excel

Système d'exploitation	Chemin de répertoire	Nom de fichier de bibliothèque d'encapsuleur
Windows	%DB2PATH%\bin	db2lsxls.dll

%DB2PATH% est la variable d'environnement qui est utilisée pour indiquer le chemin de répertoire dans lequel le serveur fédéré est installé sous Windows. Le chemin Windows par défaut est C:\Program Files\IBM\SQLLIB.

Enregistrement de la définition de serveur pour une source de données Excel

Vous devez enregistrer une définition de serveur parce que la hiérarchie des objets fédérés exige que les fichiers de classeur Excel, qui sont identifiés par des pseudonymes, soient associés à un objet de définition de serveur spécifique.

A propos de cette tâche

Vous pouvez enregistrer une définition de serveur à l'aide du Centre de contrôle ou à l'aide de la ligne de commande. Le Centre de contrôle comporte un assistant pour vous guider à travers les étapes nécessaires pour enregistrer la définition de serveur.

Procédure

Pour enregistrer une définition de serveur pour une source de données Excel :

Choisissez la méthode que vous voulez utiliser pour enregistrer la définition de serveur :

Méthode	Procédure.
Utilisation du Centre de contrôle	Démarrez l'assistant des objets fédérés. Cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée et cliquez sur Créer des objets fédérés .
Utilisation de la ligne de commande	Emettez l'instruction CREATE SERVER. Par exemple : <code>CREATE SERVER <i>nom_définition_serveur</i> WRAPPER <i>encapsuleur_excel</i>;</code>

Instruction CREATE SERVER - exemples pour l'encapsuleur Excel :

Utiliser l'instruction CREATE SERVER pour enregistrer des définitions de serveur pour l'encapsuleur Excel.

L'exemple suivant vous montre comment enregistrer une définition de serveur appelée `biochem_lab` pour un classeur contenant des données biochimiques.

L'instruction CREATE SERVER que vous émettez est :

```
CREATE SERVER biochem_lab WRAPPER encapsuleur_excel;
```

biochem_lab

Nom que vous attribuez à la définition de serveur Excel. Les doublons de noms de définition de serveur ne sont pas autorisés.

WRAPPER *Encapsuleur_excel*

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

Enregistrement de pseudonymes pour des sources de données Excel

Dans chaque définition de serveur Excel que vous enregistrez, vous devez enregistrer un pseudonyme pour chaque feuille de calcul Excel à laquelle vous voulez accéder. Utilisez ces pseudonymes, au lieu des noms des feuilles de calcul, lorsque vous interrogez les sources de données Excel.

A propos de cette tâche

Lorsque vous créez un pseudonyme pour une feuille de calcul Excel, les informations de la feuille de calcul sont mappées à une table relationnelle.

Les cellules vides dans la feuille de calcul sont interprétées comme NULL.

Jusqu'à 10 lignes vides consécutives peuvent exister dans la feuille de calcul et seront incluses dans le fichier. Plus de 10 lignes vides consécutives sont interprétées comme la fin du fichier.

Des colonnes vides peuvent exister dans la feuille de calcul. Cependant, ces colonnes doivent être enregistrées et décrites comme des zones valides même si elles ne sont pas utilisées.

Procédure

Pour enregistrer un pseudonyme pour une feuille de calcul Excel :

Choisissez la méthode que vous voulez utiliser pour enregistrer le pseudonyme. La longueur des pseudonymes peut aller jusqu'à 128 caractères.

Méthode	Procédure.
Utilisation du Centre de contrôle	Démarrez l'assistant des objets fédérés. Cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée et cliquez sur Créer des objets fédérés . Suivez les étapes de l'assistant.
Depuis la ligne de commande	Emettez l'instruction CREATE NICKNAME. Par exemple : <pre>CREATE NICKNAME <i>pseudonyme</i> (nom_colonne <i>type_données</i> OPTIONS (<i>options_colonne_pseudonyme</i>), nom_colonne <i>type_données</i> OPTIONS (<i>options_colonne_pseudonyme</i>), nom_colonne <i>type_données</i> OPTIONS (<i>options_colonne_pseudonyme</i>)) FOR SERVER <i>nom_définition_serveur</i> OPTIONS (<i>options_pseudonyme</i>);</pre>

Répétez cette étape pour chaque feuille de calcul Excel pour laquelle vous voulez créer un pseudonyme.

Instruction CREATE NICKNAME - exemples pour l'encapsuleur Excel :

Utiliser l'instruction CREATE NICKNAME pour enregistrer un pseudonyme pour une feuille de calcul Excel à laquelle vous voulez accéder. Ces exemples montrent les paramètres obligatoires et les options de pseudonyme facultatives.

```
CREATE NICKNAME Compounds
(
  Compound_ID INTEGER,
  CompoundName VARCHAR(50),
  MolWeight FLOAT
)
FOR SERVER biochem_lab
OPTIONS (FILE_PATH 'C:\My Documents\CompoundMaster.xls',
        RANGE 'B2:D25');
```

Compounds

Pseudonyme unique utilisé pour identifier la feuille de calcul Excel.

Important : Le pseudonyme est un nom en deux parties se composant du schéma et du nom du pseudonyme. Si vous omettez le schéma lorsque

vous enregistrez le pseudonyme, l'identificateur utilisateur de l'utilisateur qui enregistre le pseudonyme est utilisé comme schéma de pseudonyme.

Compound_ID *INTEGER*

Nom et type de données d'une colonne de la feuille de calcul contenant les identificateurs de composé.

CompoundNAME *VARCHAR(50)*

Nom et type de données d'une colonne de la feuille de calcul contenant les noms des composés.

MolWeight *FLOAT*

Nom et type de données d'une colonne de la feuille de calcul contenant le poids moléculaire des composés.

FOR SERVER *biochem_lab*

Nom que vous avez affecté à la définition de serveur Excel dans l'instruction CREATE SERVER.

FILE_PATH *'C:\My Documents\CompoundMaster.xls'*

Indique le chemin de répertoire et le nom de fichier qualifiés complets du classeur Excel qui contient les données auxquelles vous voulez accéder. Les données doivent être dans la première feuille de calcul du classeur.

OPTIONS (RANGE 'B2:D25')

Indique la plage de cellules à laquelle vous voulez accéder dans le classeur que vous avez indiqué dans l'option de pseudonyme FILE_PATH.

Toute syntaxe ou erreur sémantique dans la valeur de l'option de plage entraîne un message SQL1882E. Les erreurs possibles sont :

- La plage n'est pas une plage valide. Par exemple, si la cellule supérieure gauche indiquée dans la plage est au-dessous ou à droite de la cellule inférieure droite.
- Le nombre de colonnes désignées par la valeur de plage ne correspond pas au nombre de colonnes indiquées dans l'instruction CREATE NICKNAME.
- Un caractère incorrect ou une autre erreur de syntaxe a été trouvé.

Sources de données Excel - exemples de requête

Pour accéder aux données Excel, vous utilisez le pseudonyme et les colonnes de pseudonymes définies dans vos instructions SQL de la même manière que vous utiliseriez un nom de table normal et des colonnes de table.

Ces exemples vous montrent comment structurer les requêtes pour accéder à des données Excel à l'aide des composés de pseudonyme.

Sélection d'une colonne spécifique d'informations

La requête suivante affiche tous les compound_ID dont le poids moléculaire est supérieur à 2000 :

```
SELECT compound_ID FROM compounds
WHERE molweight > 2000;
```

Utilisation d'une condition OR (OU) dans vos instructions SELECT

La requête suivante affiche tous les enregistrements dans lesquels le nom du composé ou le poids moléculaire sont nuls :

```
SELECT * FROM compounds
WHERE compoundname IS NULL OR molweight IS NULL;
```

Utilisation des conditions LIKE (COMME) et AND (ET) dans vos instructions SELECT

La requête suivante affiche tous les enregistrements dans lesquels le nom du composé contient la chaîne ase et le poids moléculaire est supérieur ou égal à 300 :

```
SELECT * FROM compounds
WHERE compoundname LIKE '%ase%' AND molweight >= 300;
```

Source de données Excel - exemple de scénario

Ce scénario montre les instructions SQL qui sont nécessaires pour enregistrer les objets fédérés qui sont utilisés pour accéder à une feuille de calcul Excel. Ce scénario comprend plusieurs requêtes que vous pouvez exécuter à l'aide du pseudonyme que vous créez.

Informations de la feuille de calcul Excel

Ce scénario commence par une feuille de calcul qui contient des informations sur divers composés. Le nom du classeur dans lequel se trouve la feuille de calcul est Compound_Master.xls et le classeur a été créé dans Excel. Le chemin d'accès complet du classeur est C:\Data\Compound_Master.xls.

La première feuille de calcul du classeur contient quatre colonnes et neuf lignes de données. Les colonnes indiquent les noms des composés, le poids des composés, le comptage moléculaire du composé et si le composé a été testé.

Le contenu de la feuille de calcul est affiché dans la table suivante.

Tableau 19. Exemple de feuille de calcul Compound_Master.xls

Â	A	B	C	D
1	COMPOUND_NAME	WEIGHT	MOL_COUNT	WAS_TESTED
2	compound_A	1.23	367	testé
3	compound_G	Â	210	Â
4	compound_F	0.000425536	174	testé
5	compound_Y	1.00256	Â	testé
6	compound_Q	Â	1024	Â
7	compound_B	33.5362	Â	Â
8	compound_S	0.96723	67	testé
9	compound_O	1.2	Â	testé

Enregistrer les objets fédérés

Pour accéder à la feuille de calcul à l'aide de l'encapsuleur Excel, vous devez enregistrer les objets sur le serveur fédéré :

1. Enregistrez l'encapsuleur Excel.

Par exemple :

```
CREATE WRAPPER Excel LIBRARY 'db2lsls.dll';
```

2. Enregistrez la définition de serveur :

Par exemple :

```
CREATE SERVER biochem_lab WRAPPER Excel;
```

3. Enregistrez un pseudonyme qui renvoie à la feuille de calcul Excel :

Par exemple :

```
CREATE NICKNAME Compound_Master
  (compound_name VARCHAR(40),
   weight         FLOAT,
   mol_count      INTEGER,
   was_tested     VARCHAR(20))
FOR SERVER biochem_lab
OPTIONS (FILE_PATH 'C:\Data\Compound_Master.xls');
```

Le processus d'enregistrement est terminé. La feuille de calcul Excel fait maintenant partie du système fédéré et peut être utilisée dans les requêtes SQL.

Les exemples suivants montrent les requêtes SQL et les résultats renvoyés du pseudonyme *Compound_Master*.

Requête renvoyant toutes les données qui correspondent à une condition de clause WHERE spécifique

Pour renvoyer toutes les données des composés qui ont un comptage moléculaire supérieur à 100, émettez la requête suivante :

```
SELECT * FROM Compound_Master
WHERE mol_count > 100;
```

Requête renvoyant des colonnes spécifiques de la feuille de calcul

Pour renvoyer les noms et les comptages moléculaires de tous les composés dont le comptage moléculaire n'a pas encore été déterminé, émettez la requête suivante :

```
SELECT compound_name, mol_count FROM Compound_Master
WHERE mol_count IS NULL;
```

Toutes les colonnes des lignes 2, 3, 4, 6 et 8 sont renvoyées.

Les colonnes *compound_name* et *mol_count* des lignes 5, 7 et 10 sont renvoyées.

Requête comptant le nombre de lignes qui correspondent à des conditions de clause WHERE spécifiques

Pour renvoyer le nombre de composés ayant un poids supérieur à 1 et n'ayant pas été testés, émettez la requête suivante :

```
SELECT count(*) FROM Compound_Master
WHERE was_tested IS NULL AND weight > 1
```

Le comptage d'enregistrement 1 est renvoyé. Le composé de la ligne 7 correspond aux critères d'interrogation.

Requête renvoyant des colonnes spécifiques de la feuille de calcul et comportant une instruction de sous-requête

Pour renvoyer les noms et les comptages moléculaires de tous les composés dont le comptage moléculaire a été déterminé et dont le comptage moléculaire est inférieur au comptage moléculaire moyen, émettez la requête suivante :

```
SELECT compound_name, mol_count FROM Compound_Master
WHERE mol_count IS NOT NULL
AND mol_count <
(SELECT AVG(mol_count) FROM Compound_Master
WHERE mol_count IS NOT NULL AND was_tested IS NOT NULL);
```

La sous-requête renvoie 368 comme moyenne du comptage moléculaire. La requête principale utilise la moyenne pour renvoyer les résultats de requête qui sont affichés dans la table suivante :

Tableau 20. Résultats de la requête

COMPOUND_NAME	MOL_COUNT
compound_A	367
compound_G	210
compound_F	174
compound_S	67

Modèle de contrôle d'accès aux fichiers pour l'encapsuleur Excel

Pour accéder à un fichier Excel, l'encapsuleur exige une identité de l'utilisateur à des fins de sécurité. L'encapsuleur Excel utilise l'identité de l'utilisateur qui est associée au service de base de données fédérée. Le nom du service de base de données fédérée dépend du nom de l'instance du gestionnaire de bases de données. Par exemple, si le nom de l'instance du gestionnaire de bases de données est DB2, alors le nom de service sera DB2 - DB2. Pour déterminer l'identité de l'utilisateur qui est associée au service de base de données fédérée, utilisez le Panneau de configuration dans Windows pour afficher les services. Cliquez deux fois sur le nom de service et affichez la page Propriétés de connexion.

Configuration de l'accès aux sources de données Informix

Pour configurer un serveur fédéré pour accéder aux sources de données Informix, vous devez fournir au serveur fédéré des informations sur les sources de données et les objets auxquels vous voulez accéder.

Avant de commencer

- Le logiciel Informix Client SDK doit être installé et configuré sur le serveur qui fera office de serveur fédéré.
- Vérifiez la configuration du serveur fédéré.
- Vérifiez le paramètre fédéré pour vérifier que la fédération est activée.
- Sur les serveurs fédérés AIX, la bibliothèque AIX Base Application Development Math Library doit être installée. Vous pouvez déterminer si la bibliothèque est installée en émettant la commande AIX `lspp -l bos.adt.libm`.

Vous pouvez configurer un serveur fédéré pour accéder aux données qui sont stockées dans des sources de données Informix à l'aide du Centre de contrôle DB2 ou en émettant des instructions SQL depuis la ligne de commande DB2. Le Centre de contrôle DB2 comporte un assistant pour vous guider à travers les étapes nécessaires pour configurer les objets fédérés obligatoires.

Procédure

Pour ajouter des sources de données Informix à un serveur fédéré :

1. Configurez et testez le fichier de configuration client Informix.
2. Définissez les variables d'environnement Informix.
3. Enregistrez l'encapsuleur.
4. Enregistrez la définition de serveur.
5. Créez les mappages utilisateur.

6. Testez la connexion au serveur.
7. Enregistrez des pseudonymes pour les tables, vues et synonymes Informix.

Configuration et test du fichier de configuration client Informix

Le fichier de configuration client Informix est utilisé pour se connecter aux bases de données Informix, à l'aide des bibliothèques client qui sont installées sur le serveur fédéré.

Avant de commencer

Le logiciel Informix Client SDK doit être installé sur le serveur fédéré.

Le fichier de configuration client indique l'emplacement de chaque serveur de la base Informix et le type de connexion (protocole) du serveur de la base.

L'emplacement par défaut du fichier de configuration client dépend du système d'exploitation qui est utilisé par le serveur fédéré.

- Sur les serveurs fédérés qui fonctionnent sous UNIX, l'emplacement et le nom du fichier par défaut sont \$INFORMIXDIR/etc/sqlhosts. Le fichier sqlhosts est installé avec Informix Client SDK.
- Sur les serveurs fédérés qui fonctionnent sous Windows, l'emplacement par défaut du registre sqlhosts est l'ordinateur local.

Le format de sqlhosts est décrit dans le *Guide d'administration d'Informix Dynamic Server*.

Procédure

Pour configurer et tester le fichier de configuration client Informix :

1. Configurez Informix Client SDK.
 - Sur les serveurs fédérés qui fonctionnent sous UNIX, vous pouvez configurer Informix Client SDK en éditant le fichier sqlhosts. Vous pouvez aussi copier le fichier sqlhosts à partir d'un autre système sur lequel Informix Connect ou Informix Client SDK est installé.
 - Sur les serveurs fédérés qui fonctionnent sous Windows, vous pouvez configurer Informix Client SDK avec l'utilitaire Informix Setnet32. L'utilitaire Setnet32 configure le registre sqlhosts.
2. Vérifiez l'emplacement du fichier ou registre sqlhosts.
 - Sur les serveurs fédérés qui fonctionnent sous UNIX, le fichier sqlhosts se trouve dans le répertoire \$INFORMIXDIR/etc/.
 - Sur les serveurs fédérés qui fonctionnent sous Windows, les informations de sqlhosts sont conservées dans la clé suivante du registre Windows :
HKEY_LOCAL_MACHINE\SOFTWARE\INFORMIX\SQLHOSTS
3. Si vous voulez placer le fichier ou registre sqlhosts dans un chemin autre que le chemin de recherche par défaut, définissez la variable d'environnement INFORMIXSQLHOSTS pour indiquer l'adresse du fichier. Utilisez une des options suivantes pour définir la variable d'environnement INFORMIXSQLHOSTS.
 - Sur les serveurs fédérés qui fonctionnent sous UNIX, définissez la variable d'environnement INFORMIXSQLHOSTS sur le nom qualifié complet du fichier sqlhosts.

- Sur les serveurs fédérés qui fonctionnent sous Windows, utilisez l'utilitaire Setnet32 pour définir la variable d'environnement INFORMIXSQLHOSTS sur le nom de l'ordinateur Windows qui stocke le registre.
4. Testez la connexion pour vérifier que le logiciel client peut se connecter au serveur Informix. Si l'utilitaire Informix dbaccess se trouve sur le serveur fédéré, utilisez-le pour tester la connexion. Sinon, exécutez le programme de l'exemple de démonstration Informix pour tester la configuration du client.

Définition des variables d'environnement Informix

Les variables d'environnement Informix doivent être définies dans le fichier db2dj.ini sur le serveur fédéré.

Il y a des variables d'environnement obligatoires et des variables d'environnement facultatives pour les sources de données Informix. Si vous avez installé le logiciel client Informix avant d'avoir installé l'encapsuleur Informix, les variables d'environnement Informix nécessaires sont définies dans le fichier db2dj.ini.

Vous devez définir les variables d'environnement à l'aide des étapes de cette tâche si vous n'avez pas installé le logiciel client Informix avant d'installer l'encapsuleur Informix ou que vous voulez définir l'une des variables d'environnement facultatives.

Pour définir les variables d'environnement Informix :

1. Utilisez l'une des méthodes suivantes pour définir les variables d'environnement Informix que vous voulez utiliser :

Méthode	Etape
Définir les variables d'environnement à l'aide du Centre de contrôle DB2	Utilisez l'assistant des objets fédérés dans le Centre de contrôle DB2. Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés.
Définir les variables d'environnement automatiquement	Réexécutez le programme de configuration DB2 et indiquez l'option de l'installation Personnalisée . Suivez les instructions de l'assistant. Important : La réexécution du programme d'installation définira uniquement les variables d'environnement obligatoires. Les variables d'environnement facultatives doivent être définies manuellement.
Pour définir les variables d'environnement manuellement	Modifiez le fichier db2dj.ini. <ul style="list-style-type: none"> • Sur les serveurs fédérés qui fonctionnent sous UNIX, le fichier db2dj.ini se trouve dans le répertoire sqllib/cfg. • Sur les serveurs fédérés qui fonctionnent sous Windows, le fichier db2dj.ini se trouve dans le répertoire %DB2PATH%\cfg.

Le fichier db2dj.ini contient des informations de configuration sur le logiciel client Informix qui est installé sur votre serveur fédéré. Si le fichier n'existe pas, vous pouvez créer un fichier portant le nom db2dj.ini à l'aide de tout éditeur de texte. Dans le fichier db2dj.ini, vous devez indiquer le chemin qualifié

complet des variables d'environnement, sinon vous rencontrerez des erreurs. Les variables d'environnement suivantes montrent à quoi pourrait ressembler une entrée du fichier db2dj.ini sous UNIX :

```
INFORMIXDIR=/informix/csdk  
INFORMIXSERVER=inf10
```

2. Définissez les variables d'environnement de la conversion de page de codes Informix (si nécessaire).
3. Pour vérifier que les variables d'environnement sont définies sur le serveur fédéré, recyclez l'instance de la base de données fédérée.

Emettez les commandes suivantes pour recycler l'instance de la base de données fédérée :

```
db2stop  
db2start
```

Les variables d'environnement Informix

Il y a des variables d'environnement obligatoires et des variables d'environnement facultatives pour les sources de données Informix. Ces variables sont définies dans le fichier db2dj.ini.

Les variables d'environnement valides pour Informix sont :

- INFORMIXDIR
- INFORMIXSERVER
- INFORMIXSQLHOSTS (facultative)
- CLIENT_LOCALE (facultative)
- DB_LOCALE (facultative)
- DBNLS (facultative)

Les variables d'environnement CLIENT_LOCALE, DB_LOCALE et DBNLS sont des variables d'environnement de la page de codes.

Description des variables

INFORMIXDIR

Indique le chemin du répertoire dans lequel le logiciel Informix Client SDK est installé.

Par exemple :

- Sur les serveurs fédérés qui fonctionnent sous UNIX, définissez le chemin sur :

```
INFORMIXDIR=/informix/csdk
```

- Sur les serveurs fédérés qui fonctionnent sous Windows, définissez le chemin sur :

```
INFORMIXDIR=C:\informix\csdk
```

INFORMIXSERVER

Identifie le nom du serveur Informix par défaut. Ce paramètre doit être une entrée valide dans le fichier sqlhosts (sous UNIX) ou dans la clé de registre SQLHOSTS (sous Windows). Pour obtenir une valeur pour la variable INFORMIXSERVER, lisez le fichier sqlhosts. Sélectionnez une des valeurs de *dbservername*. La valeur *dbservername* est la première valeur de chaque entrée dans le fichier sqlhosts.

Par exemple :

```
INFORMIXSERVER=inf10
```

Condition requise : Bien que l'encapsuleur Informix n'utilise pas la valeur de cette variable d'environnement, le client Informix requiert la définition de cette variable d'environnement. L'encapsuleur utilise la valeur de l'option du serveur NODE, qui indique le serveur de la base Informix auquel vous voulez accéder.

INFORMIXSQLHOSTS

Si vous utilisez le chemin par défaut pour le fichier sqlhosts Informix, vous n'avez pas besoin de définir cette variable d'environnement. Cependant, si vous utilisez un autre chemin pour le fichier sqlhosts Informix, alors vous devez définir cette variable d'environnement. Définissez la variable INFORMIXSQLHOSTS sur le chemin d'accès complet dans lequel se trouve le fichier sqlhosts Informix.

- Sur les serveurs fédérés qui fonctionnent sous UNIX, le chemin par défaut est \$INFORMIXDIR/etc/sqlhosts.
- Sur les serveurs fédérés qui fonctionnent sous Windows, si la clé de registre SQLHOSTS ne réside pas sur l'ordinateur local, alors la valeur de la variable d'environnement INFORMIXSQLHOSTS est le nom de l'ordinateur Windows qui stocke le registre.

Voici un exemple sous UNIX de définition de cette variable d'environnement sur un autre chemin :

```
INFORMIXSQLHOSTS=/informix/csdk/etc/my_sqlhosts
```

Conversion de page de codes Informix :

Chaque fois que l'encapsuleur Informix se connecte à une source de données Informix, l'encapsuleur détermine quelle valeur de page de codes utiliser pour cette connexion. L'encapsuleur Informix peut définir la valeur de page de codes ou bien vous pouvez désigner une page de codes en définissant la variable d'environnement CLIENT_LOCALE.

Les variables d'environnement qui indiquent la conversion de la page de codes Informix sont définies dans le fichier db2dj.ini sur votre serveur fédéré.

Pour la conversion de la page de codes Informix, vous pouvez définir les variables d'environnement facultatives suivantes :

- CLIENT_LOCALE
- DB_LOCALE
- DBNLS

Les variables d'environnement de la page de codes Informix sont :

CLIENT_LOCALE

Indique l'environnement local Informix que vous voulez utiliser. Utilisez cette variable lorsque vous ne voulez pas que l'encapsuleur Informix détermine automatiquement la valeur de la variable.

Par exemple :

```
CLIENT_LOCALE=valeur_environnement_local_client_Informix
```

- Si la variable CLIENT_LOCALE est définie dans le fichier db2dj.ini sur le serveur fédéré, l'encapsuleur utilise la valeur de page de codes dans le fichier db2dj.ini.
- Si la variable CLIENT_LOCALE n'est pas définie sur le serveur fédéré, l'encapsuleur détermine le territoire et la page de codes de la base de données fédérée. L'encapsuleur définit la variable CLIENT_LOCALE sur

l'environnement local Informix correspondant le plus. S'il n'y a pas d'environnement local Informix correspondant, l'encapsuleur définit la variable CLIENT_LOCALE sur l'environnement local en_us.8859-1 pour les systèmes UNIX et sur l'environnement local en_us.CP1252 pour les systèmes Windows.

Vous pouvez voir la liste des environnements locaux Informix valides en émettant la commande glfiles sur le serveur Informix.

Voir le *Guide Informix des fonctionnalités GLS* pour de plus amples informations sur les conversions de page de codes.

DB_LOCALE

Indique que la base de données Informix utilise une page de codes différente de celle qu'utilise votre environnement local client. Utilisez cette variable lorsque vous voulez qu'Informix exécute des conversions entre les deux pages de codes. Définissez la variable d'environnement DB_LOCALE sur le nom de l'environnement local de la base de données Informix.

Par exemple :

```
DB_LOCALE=valeur_environnement_local_bd_Informix
```

DBNLS

Indique qu'Informix vérifie que le paramètre DB_LOCALE correspond à l'environnement local réel de la base de données Informix. Définissez cette variable d'environnement sur 1.

Par exemple :

```
DBNLS=1
```

Imposer à Informix d'exécuter la conversion de la page de codes

La base de données Informix utilise une page de codes différente de celle de votre environnement local client et vous voulez qu'Informix effectue des conversions entre les deux pages de codes. Vous devez :

1. Définir la variable d'environnement Informix DB_LOCALE sur le nom de l'environnement local de la base de données Informix. Vous définissez cette variable dans le fichier db2dj.ini sur le serveur fédéré.
2. Pour vérifier que le paramètre DB_LOCALE correspond à l'environnement local réel de la base de données Informix, définissez la variable d'environnement Informix DBNLS sur 1. Vous définissez cette variable dans le fichier db2dj.ini sur le serveur fédéré.

Données Informix utilisant la page de codes chinois GB 18030

Pour accéder à des données qui utilisent la page de codes chinois GB 18030, utilisez la page de codes UTF-8 sur votre base de données fédérée et ajoutez le paramètre suivant à votre fichier db2dj.ini, afin qu'Informix convertisse correctement les données GB 18030 vers Unicode.

```
DB_LOCALE=zh_cn.GB18030-2000
```

Enregistrement de l'encapsuleur Informix

Vous devez enregistrer un encapsuleur pour accéder aux sources de données Informix. Les encapsuleurs sont utilisés par les serveurs fédérés pour dialoguer avec les sources de données et extraire des données des sources de données. Les encapsuleurs sont mis en oeuvre comme un ensemble de fichiers de bibliothèque.

Procédure

Pour enregistrer l'encapsuleur Informix :

Emettez l'instruction CREATE WRAPPER et indiquez le nom par défaut de l'encapsuleur Informix.

Par exemple :

```
CREATE WRAPPER INFORMIX;
```

Recommandation : Utilisez le nom d'encapsuleur par défaut. Le nom d'encapsuleur par défaut pour Informix est INFORMIX. Lorsque vous enregistrez l'encapsuleur avec le nom par défaut, le serveur fédéré utilise automatiquement la bibliothèque de l'encapsuleur Informix appropriée au système d'exploitation sous lequel votre serveur fédéré fonctionne.

En cas de conflit entre le nom d'encapsuleur par défaut et un nom d'encapsuleur existant dans la base de données fédérée, vous pouvez remplacer le nom d'encapsuleur par défaut par un nom que vous choisissez. Lorsque vous utilisez un nom différent du nom par défaut, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER.

Par exemple, pour enregistrer un encapsuleur avec le nom encapsuleur_informix sur un serveur fédéré qui utilise le système d'exploitation AIX, émettez l'instruction suivante :

```
CREATE WRAPPER encapsuleur_informix LIBRARY 'libdb2informix.a';
```

Le nom du fichier de bibliothèque de l'encapsuleur que vous indiquez dépend du système d'exploitation du serveur fédéré. Voir la liste des fichiers de bibliothèque de l'encapsuleur Informix pour le nom de bibliothèque approprié à indiquer dans l'instruction CREATE WRAPPER.

Fichiers de bibliothèque de l'encapsuleur Informix

Les fichiers de bibliothèque de l'encapsuleur Informix sont ajoutés au serveur fédéré lorsque vous installez la fédération.

Lorsque vous installez la fédération, trois fichiers de bibliothèque sont ajoutés dans le chemin de répertoire par défaut. Par exemple, si le serveur fédéré fonctionne sous AIX, les fichiers de bibliothèque d'encapsuleur qui sont ajoutés dans le chemin de répertoire sont libdb2informix.a, libdb2informixF.a et libdb2informixU.a. Le fichier de bibliothèque de l'encapsuleur par défaut est libdb2informix.a. Les autres fichiers de bibliothèque de l'encapsuleur sont utilisés sur le plan interne par la bibliothèque de l'encapsuleur par défaut.

Si vous décidez de ne pas utiliser le nom d'encapsuleur par défaut lorsque vous enregistrez l'encapsuleur, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER et indiquer le nom de fichier de bibliothèque de l'encapsuleur par défaut.

Les chemins des répertoires par défaut et les noms de fichier de bibliothèque de l'encapsuleur par défaut sont indiqués dans la table suivante.

Tableau 21. Emplacements et noms des fichiers de bibliothèque de l'encapsuleur Informix

Système d'exploitation	Chemin de répertoire	Nom de fichier de bibliothèque
AIX	/usr/opt/chemin_installation/lib32/ /usr/opt/chemin_installation/lib64/	libdb2informix.a

Tableau 21. Emplacements et noms des fichiers de bibliothèque de l'encapsuleur Informix (suite)

Système d'exploitation	Chemin de répertoire	Nom de fichier de bibliothèque
HP-UX	 /opt/IBM/db2/chemin_installation/lib32 /opt/IBM/db2/chemin_installation/lib64	libdb2informix.so
Linux	 /opt/IBM/db2/chemin_installation/lib32 /opt/IBM/db2/chemin_installation/lib64	libdb2informix.so
Solaris	 /opt/IBM/db2/chemin_installation/lib32 /opt/IBM/db2/chemin_installation/lib64	libdb2informix.so
Windows	%DB2PATH%\bin	db2informix.dll

- *chemin_installation* est le chemin du répertoire dans lequel la fédération est installée sous UNIX ou Linux.
- %DB2PATH% est la variable d'environnement qui est utilisée pour indiquer le chemin du répertoire dans lequel la fédération est installée sous Windows. Le chemin Windows par défaut est C:\Program Files\IBM\SQLLIB.

Enregistrement des définitions de serveur pour une source de données Informix

Vous devez enregistrer chaque serveur Informix auquel vous voulez accéder dans la base de données fédérée.

Procédure

Pour enregistrer une définition de serveur pour une source de données Informix :

1. Recherchez le nom du noeud dans le fichier ou registre sqlhosts Informix.

Exemple de fichier sqlhosts :

```
inf10an onsoctcp anaconda inmx10
inf10bo onsoctcp boa ifmx10
inf10py onsoctcp python ifmx10
```

- La première valeur de chaque ligne est le *nom_noeud*, par exemple inf10an.
- La deuxième valeur de chaque ligne est le *type_de_reseau* ou type de connexion. Dans cet exemple, onsoctcp indique que c'est une connexion TCP/IP.
- La troisième valeur de chaque ligne est le nom d'hôte, par exemple anaconda, boa et python.
- La quatrième valeur de chaque ligne est le nom de service, par exemple inmx10. La zone du nom de service dépend du *type_de_reseau* indiqué dans la deuxième valeur.

Pour de plus amples informations sur le format du fichier sqlhosts et la signification de ces zones, voir le manuel Informix *Guide d'administration d'Informix Dynamic Server*.

2. Utilisez l'une des méthodes suivantes pour créer la définition de serveur.
 - Utilisez l'assistant des objets fédérés du centre de contrôle DB2 Universal Database. Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés.
 - Emettez l'instruction CREATE SERVER.

Par exemple :


```
CREATE SERVER nom_définition_serveur TYPE informix
  VERSION numéro_version WRAPPER INFORMIX
  OPTIONS (NODE 'nom_noeud', DBNAME 'nom_base_de_données');
```

Bien que les variables '*nom_noeud*' et '*nom_base_de_données*' soient indiquées comme options dans l'instruction CREATE SERVER, ces options doivent être indiquées obligatoirement pour les sources de données Informix.

Une fois la définition de serveur enregistrée, utilisez l'instruction ALTER SERVER pour ajouter ou déplacer des options de serveur.

Instruction CREATE SERVER - exemples pour l'encapsuleur Informix

Utiliser l'instruction CREATE SERVER pour enregistrer des définitions de serveur pour l'encapsuleur Informix. Cette rubrique comporte un exemple complet avec les paramètres obligatoires et des exemples avec des options de serveur supplémentaires.

Exemple complet

L'exemple suivant vous montre comment enregistrer une définition de serveur pour un encapsuleur Informix en émettant l'instruction CREATE SERVER :

```
CREATE SERVER asia TYPE informix VERSION 10 WRAPPER INFORMIX
  OPTIONS (NODE 'abc', DBNAME 'sales');
```

asia Nom que vous attribuez au serveur de la base Informix. Les doublons de noms de définition de serveur ne sont pas autorisés.

TYPE *informix*

Indique le type de serveur de source de données auquel vous configurez l'accès. Pour l'encapsuleur Informix, le type de serveur doit être *informix*.

VERSION *10*

Version du serveur de la base Informix auquel vous voulez accéder.

WRAPPER *INFORMIX*

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

NODE '*abc*'

Nom du noeud sur lequel le serveur de la base Informix se trouve. Obtenez le nom du noeud à partir du fichier sqlhosts. Cette valeur fait la distinction entre les majuscules et les minuscules.

Bien que le nom du noeud soit indiqué comme option dans l'instruction CREATE SERVER, il doit être indiqué obligatoirement pour les sources de données Informix.

DBNAME '*sales*'

Nom de la base de données Informix à laquelle vous voulez accéder. Cette valeur fait la distinction entre les majuscules et les minuscules.

Bien que le nom de la base de données soit indiqué comme option dans l'instruction CREATE SERVER, il doit être indiqué obligatoirement pour les sources de données Informix.

Options de serveur supplémentaires

Lorsque vous créez une définition de serveur, vous pouvez indiquer des options de serveur supplémentaires dans l'instruction CREATE SERVER. Les options de serveur peuvent être des options de serveur générales et des options de serveur spécifiques à Informix.

Options de serveur FOLD_ID et FOLD_PW

Lorsque le serveur fédéré se connecte à une source de données, le serveur fédéré essaie de se connecter à l'aide de toutes les combinaisons possibles de majuscules et minuscules pour l'ID utilisateur et le mot de passe, en plus de la combinaison actuelle. Le serveur fédéré peut effectuer jusqu'à neuf tentatives de connexion avant de se connecter avec succès au serveur de source de données. Ces tentatives peuvent augmenter les délais de connexion et peuvent entraîner un verrouillage de l'ID utilisateur. Vous pouvez empêcher les verrouillages en indiquant des valeurs pour les options de serveur FOLD_ID et FOLD_PW. Vous pouvez définir les options de serveur FOLD_ID et FOLD_PW sur 'N' (ne pas convertir l'ID utilisateur et le mot de passe).

Si vous définissez les options de serveur FOLD_ID et FOLD_PW sur 'N', vous devez indiquer l'ID utilisateur et le mot de passe en respectant l'écriture en majuscules et en minuscules. L'avantage de la définition de ces options de serveur sur 'N' est que lorsqu'un ID utilisateur ou un mot de passe non valide est indiqué, l'encapsuleur n'essaiera pas les autres combinaisons de majuscules et de minuscules. Ces deux options de serveur peuvent réduire le risque de dépassement du nombre maximum d'échecs de tentatives de connexion entraînant un verrouillage de l'ID.

L'exemple suivant montre une définition de serveur Informix avec ces options de serveur :

```
CREATE SERVER asia TYPE informix VERSION 10 WRAPPER INFORMIX
      OPTIONS (NODE 'abc', DBNAME 'sales', FOLD_ID 'N', FOLD_PW 'N');
```

Exemple avec l'option de serveur IUD_APP_SVPT_ENFORCE

L'option IUD_APP_SVPT_ENFORCE indique si le serveur fédéré doit imposer la détection ou la génération d'instructions de point de sauvegarde de l'application. Informix ne prend pas en charge les instructions de point de sauvegarde de l'application. Lorsque cette option est définie sur 'N', le serveur fédéré n'annule pas les transactions lorsqu'une erreur est rencontrée. Votre application doit traiter la correction d'erreurs.

L'option de serveur IUD_APP_SVPT_ENFORCE doit être définie sur 'N' pour activer la réplication vers ou à partir des sources de données Informix. L'exemple suivant montre une définition de serveur Informix avec l'option de serveur IUD_APP_SVPT_ENFORCE.

```
CREATE SERVER asia TYPE informix VERSION 10 WRAPPER INFORMIX
      OPTIONS (NODE 'abc', DBNAME 'sales', IUD_APP_SVPT_ENFORCE 'N');
```

Options INFORMIX_DB_LOCALE et INFORMIX_CLIENT_LOCALE

L'option INFORMIX_DB_LOCALE définit la variable d'environnement d'environnement local de la base de données (DB_LOCALE) qui est utilisée pour la connexion entre le serveur fédéré et le serveur de source de données. Si l'option INFORMIX_DB_LOCALE n'est pas indiquée, la variable d'environnement DB_LOCALE Informix est définie sur la valeur qui est indiquée dans le fichier db2dj.ini. Si le fichier db2dj.ini n'indique pas la variable d'environnement DB_LOCALE, la variable d'environnement DB_LOCALE Informix n'est pas définie. Les valeurs valides sont tout environnement local Informix valide. Cette option est facultative. La valeur par défaut est None.

L'option `INFORMIX_CLIENT_LOCALE` définit la variable d'environnement de l'environnement local client (`CLIENT_LOCALE`) à utiliser pour la connexion entre le serveur fédéré et le serveur de source de données. Si l'option `INFORMIX_CLIENT_LOCALE` n'est pas indiquée, la variable d'environnement `CLIENT_LOCALE` Informix est définie sur la valeur indiquée dans le fichier `db2dj.ini`. Si `db2dj.ini` n'indique pas `CLIENT_LOCALE`, alors la variable d'environnement `CLIENT_LOCALE` Informix est définie sur l'environnement local Informix correspondant le plus étroitement à la page de codes et au territoire de la base de données fédérée. Les valeurs valides sont tout environnement local Informix valide. Cette option est facultative. La valeur par défaut est `None`.

L'exemple suivant montre une définition de serveur Informix avec les options `INFORMIX_DB_LOCALE` et `INFORMIX_CLIENT_LOCALE`.

```
CREATE SERVER asia TYPE informix VERSION 10 WRAPPER INFORMIX
  OPTIONS (NODE 'abc', DBNAME 'sales', INFORMIX_DB_LOCALE 'en_us.8859-1',
  INFORMIX_CLIENT_LOCALE 'en_us.CP1252');
```

Création de mappages utilisateur pour une source de données Informix

Lorsque vous essayez d'accéder à un serveur Informix, le serveur fédéré établit une connexion au serveur Informix à l'aide d'un ID utilisateur et d'un mot de passe qui sont valides pour cette source de données. Vous devez définir une association (un mappage utilisateur) entre chaque ID utilisateur et mot de passe de serveur fédéré et l'ID utilisateur et le mot de passe de la source de données correspondante.

Créez un mappage utilisateur pour chaque ID utilisateur qui accédera au système fédéré pour envoyer des requêtes réparties à la source de données Informix.

Procédure

Pour mapper un ID utilisateur local à l'ID utilisateur et au mot de passe du serveur Informix :

Emettez une instruction `CREATE USER MAPPING`.

Par exemple :

```
CREATE USER MAPPING FOR IDutilisateur_local SERVER nom_définition_serveur
  OPTIONS (REMOTE_AUTHID 'IDutilisateur_distant', REMOTE_PASSWORD
  'motdepasse_distant');
```

Bien que les variables `REMOTE_AUTHID` et `REMOTE_PASSWORD` soient indiquées comme options dans l'instruction `CREATE USER MAPPING`, ces options doivent être indiquées obligatoirement pour accéder aux sources de données Informix.

Instruction `CREATE USER MAPPING` - exemples pour l'encapsuleur Informix

Utiliser l'instruction `CREATE USER MAPPING` pour mapper un ID utilisateur du serveur fédéré à un ID utilisateur et un mot de passe du serveur Informix. Cette rubrique comporte un exemple complet avec les paramètres obligatoires et un exemple qui vous montre comment utiliser le registre spécial `DB2 USER` avec l'instruction `CREATE USER MAPPING`.

Exemple complet

L'exemple suivant montre comment mapper un ID utilisateur du serveur fédéré à un ID utilisateur et un mot de passe du serveur Informix :

```
CREATE USER MAPPING FOR VINCENT SERVER asia
  OPTIONS (REMOTE_AUTHID 'vinnie', REMOTE_PASSWORD 'close2call');
```

VINCENT

Indique l'ID utilisateur local que vous mappez à un ID utilisateur qui est défini sur le serveur Informix.

SERVER *asia*

Indique le nom de définition de serveur que vous avez enregistré dans l'instruction CREATE SERVER pour le serveur Informix.

REMOTE_AUTHID *'vinnie'*

Indique l'ID utilisateur sur le serveur de la base Informix auquel vous mappez *VINCENT*. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_ID sur 'U' ou 'L' dans l'instruction CREATE SERVER.

Bien que l'ID utilisateur distant soit indiqué comme option dans l'instruction CREATE USER MAPPING, il doit être indiqué obligatoirement pour les sources de données Informix.

REMOTE_PASSWORD *'close2call'*

Indique le mot de passe qui est associé à *'vinnie'*. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_PW sur 'U' ou 'L' dans l'instruction CREATE SERVER.

Bien que le mot de passe distant soit indiqué comme option dans l'instruction CREATE USER MAPPING, il doit être indiqué obligatoirement pour les sources de données Informix.

Exemple de registre spécial

Vous pouvez utiliser le registre spécial DB2 USER pour mapper l'identificateur utilisateur de la personne qui émet l'instruction CREATE USER MAPPING vers l'identificateur utilisateur de la source de données qui est indiqué dans l'option utilisateur REMOTE_AUTHID.

L'exemple suivant montre une instruction CREATE USER MAPPING qui comporte le registre spécial USER :

```
CREATE USER MAPPING FOR USER SERVER asia
  OPTIONS (REMOTE_AUTHID 'vinnie', REMOTE_PASSWORD 'close2call');
```

Test de la connexion au serveur Informix

Tester la connexion au serveur de source de données Informix pour déterminer si le serveur fédéré est correctement configuré pour accéder aux sources de données Informix.

Vous pouvez tester la connexion au serveur Informix à l'aide de la définition de serveur et des mappages utilisateur que vous avez définis.

Procédure

Pour tester la connexion au serveur Informix :

Ouvrez une session passe-système et émettez une instruction SELECT sur les tables système Informix. Si l'instruction SELECT renvoie un comptage, votre définition de serveur et votre mappage utilisateur sont configurés correctement.

Par exemple :

```
SET PASSTHRU nom_définition_serveur
SELECT count(*) FROM informix.systables
SET PASSTHRU RESET
```

Si l'instruction SELECT renvoie une erreur, vous devez identifier et résoudre les erreurs de connexion.

Identification et résolution des erreurs de connexion à la source de données

Une connexion de test au serveur de source de données peut renvoyer une erreur pour plusieurs raisons. Il y a des mesures que vous pouvez prendre pour déterminer pourquoi l'erreur s'est produite.

Symptôme

Une erreur est renvoyée lorsque vous essayez de vous connecter à la source de données.

Cause

Il y a plusieurs causes possibles à un incident de connexion.

Résolution de l'incident

Pour identifier et résoudre les erreurs de connexion à la source de données, recherchez les problèmes éventuels avec les éléments suivants :

- Vérifiez que la source de données est disponible.
- S'il y a lieu, vérifiez que le serveur de source de données est configuré pour les connexions entrantes.
- Vérifiez que les paramètres de votre mappage utilisateur pour les options REMOTE_AUTHID et REMOTE_PASSWORD sont valides pour les connexions à la source de données. Modifiez le mappage utilisateur ou créez un autre mappage utilisateur si nécessaire.
- S'il y a lieu, vérifiez que le logiciel client de la source de données sur le serveur fédéré est installé et configuré correctement pour se connecter à la source de données.
- Pour les sources de données ODBC, vérifiez que le pilote ODBC sur le serveur fédéré est installé et configuré correctement pour se connecter au serveur de source de données ODBC. Sur les serveurs fédérés qui fonctionnent sous Windows, utilisez l'outil Administrateur de sources de données ODBC pour vérifier le pilote. Sur les serveurs fédérés qui fonctionnent sous UNIX, consultez la documentation du fournisseur de client ODBC.
- Vérifiez que les valeurs des variables définies sur le serveur fédéré sont appropriées à la source de données. Ces variables sont les variables d'environnement système, les variables du fichier db2dj.ini et les variables du registre de profil DB2 (db2set).
- Vérifiez votre définition de serveur. Si nécessaire, supprimez la définition de serveur et recréez-la.

Réglage des performances pour l'encapsuleur Informix

Vous pouvez utiliser les options de serveur FOLD_ID et FOLD_PW pour améliorer la connectivité entre le serveur fédéré et les sources de données Informix.

Lorsque le serveur fédéré se connecte à une source de données, le serveur essaie de se connecter à l'aide de toutes les combinaisons possibles de majuscules et minuscules pour l'ID utilisateur et le mot de passe. Le serveur peut effectuer jusqu'à neuf tentatives de connexion avant de se connecter avec succès au serveur de source de données. Ces tentatives peuvent augmenter les délais de connexion et peuvent entraîner un verrouillage de l'ID utilisateur.

Vous pouvez améliorer les performances en indiquant des valeurs pour les options de serveur FOLD_ID et FOLD_PW.

- Si tous vos ID utilisateur et mots de passe Informix sont en minuscules, la définition des options de serveur FOLD_ID et FOLD_PW sur la valeur 'L' peut améliorer vos délais de connexion.

Par exemple :

```
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_ID 'L');
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_PW 'L');
```

- Le serveur fédéré essaie chaque combinaison de valeurs majuscules et minuscules pour l'ID utilisateur et le mot de passe. Vous pouvez réduire le risque de dépassement du nombre maximum d'échecs de tentatives de connexion en définissant ces options sur 'N' (ne pas convertir l'ID utilisateur et le mot de passe). Si vous définissez ces paramètres, alors vous devrez toujours indiquer l'ID utilisateur et le mot de passe en respectant l'écriture en majuscules et en minuscules. Si un ID utilisateur et un mot de passe non valides sont indiqués, l'encapsuleur n'essaiera pas les autres combinaisons.

Par exemple :

```
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_ID 'N');
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_PW 'N');
```

Enregistrement de pseudonymes pour les tables, vues et synonymes Informix

Dans chaque définition de serveur Informix que vous enregistrez, vous devez enregistrer un pseudonyme pour chaque table, vue ou synonyme auquel vous voulez accéder. Utilisez ces pseudonymes, au lieu des noms des objets de source de données, lorsque vous interrogez les serveurs Informix.

Avant de commencer

Mettez à jour les statistiques sur la source de données Informix avant d'enregistrer un pseudonyme. La base de données fédérée exploite les statistiques du catalogue de la source de données pour optimiser le traitement des requêtes. Vous pouvez utiliser la commande Informix UPDATE STATISTICS, qui équivaut à la commande RUNSTATS DB2 pour mettre à jour les statistiques de la source de données.

Procédure

Pour enregistrer un pseudonyme pour une table, une vue ou un synonyme Informix :

Emettez l'instruction CREATE NICKNAME. Les pseudonymes peuvent avoir jusqu'à 128 octets de longueur.

Par exemple :

```
CREATE NICKNAME pseudonyme FOR nom_définition_serveur."schéma_distant".
```

```
"table.distante" ;
```

Lorsque vous créez le pseudonyme, le serveur fédéré interroge le catalogue de la source de données à l'aide du pseudonyme. Cette requête teste la connexion à la table, à la vue ou au synonyme de la source de données. Si la connexion ne fonctionne pas, vous recevrez un message d'erreur.

Répétez cette étape pour chaque table, vue ou synonyme Informix pour lequel vous voulez créer un pseudonyme.

Instruction CREATE NICKNAME - exemples pour l'encapsuleur Informix

Utiliser l'instruction CREATE NICKNAME pour enregistrer un pseudonyme pour une table, une vue ou un synonyme Informix auquel vous voulez accéder. Cette rubrique comporte un exemple complet avec les paramètres obligatoires.

Exemple complet

```
CREATE NICKNAME JPSALES FOR asia."vinnie".japan ;
```

JPSALES

Pseudonyme unique utilisé pour identifier la table, la vue ou le synonyme Informix.

Important : Le pseudonyme est un nom composé de deux parties, le schéma et le pseudonyme. Si vous omettez le schéma lorsque vous enregistrez le pseudonyme, le schéma du pseudonyme sera l'identificateur utilisateur de l'utilisateur qui enregistre le pseudonyme.

asia."*vinnie*".*japan*"

Identificateur en trois parties pour l'objet distant :

- *asia* est le nom de définition de serveur que vous avez affecté au serveur de la base Informix dans l'instruction CREATE SERVER.
- *vinnie* est le nom du propriétaire auquel la table, la vue ou le synonyme appartient, à moins que la base de données soit compatible avec ANSI. Dans une base de données compatible avec ANSI, c'est le nom de schéma.
- *japan* est le nom de la table, de la vue ou du synonyme distant auquel vous voulez accéder.

Le serveur fédéré convertit les noms des schémas et tables Informix en majuscules, à moins que vous ne placiez les noms entre apostrophes.

Configuration de l'accès aux sources de données JDBC

Pour configurer le serveur fédéré pour accéder aux sources de données JDBC, vous devez fournir au serveur fédéré les informations sur les sources de données et les objets auxquels vous voulez accéder.

Les sources de données accessibles via l'API JDBC s'appellent des sources de données JDBC dans ce document.

Avant de commencer

- Le pilote JDBC doit être installé et configuré sur l'ordinateur qui fait office de serveur fédéré.
- Vérifiez la configuration du serveur fédéré.
- Vérifiez le paramètre fédéré pour vérifier que la fédération est activée.
- Définissez les variables d'environnement système, les variables du fichier db2dj.ini et les variables DB2 Profile Registry (db2set). Voir la documentation de la source de données JDBC pour les variables de client JDBC. Il se peut que vous deviez définir la variable d'environnement LIBPATH.

Restrictions

- L'encapsuleur n'est pris en charge qu'en mode isolé.
- L'encapsuleur JDBC ne prend pas en charge les fonctions et instructions suivantes :
 - Instructions LOCK TABLE avec des pseudonymes
 - Isolement au niveau de l'instruction
- Les sources de données JDBC ne prennent pas en charge les opérations de mise à jour et de suppression positionnées.
- L'encapsuleur JDBC ne prend pas en charge les instructions INSERT, UPDATE et DELETE sur les sources de données qui limitent le nombre d'instructions actives pour chaque connexion. Voir la documentation de la source de données pour déterminer si elle limite le nombre d'instructions actives pour chaque connexion.
- L'encapsuleur JDBC ne prend pas en charge les opérations sur les tables qui contiennent des colonnes avec des types de données qui utilisent des indicateurs de type de données SQL de pilote. Les types d'opérations non pris en charge incluent les instruction CREATE NICKNAME et SELECT dans les sessions passe-système. L'encapsuleur JDBC prend en charge uniquement les indicateurs de types de données SQL définis par la spécification JDBC 3.0 et les spécifications suivantes. Voir la documentation du pilote JDBC pour les spécifications JDBC.
- L'encapsuleur JDBC ne prend pas en charge le type de données LOB dans les sessions passe-système.
- Restrictions concernant le mappage de types et la conversion de types de données :
 - Types de données non pris en charge : ARRAY, DATALINK, DISTINCT, JAVA_OBJECT, REF, STRUCT et OTHER
 - Types de données avec prise en charge limitée :
 - La prise en charge du type de données XML est limitée. Le serveur fédéré traite le type de données aCLOB uniquement si le type de données JDBC associé est CLOB ou SQLXML (JDBC 4.0). Autrement, le type de données XML n'est pas pris en charge.
 - L'encapsuleur JDBC stocke les types de données DBCS et UNICODE sous la forme UCS-2.
 - Le type de données DECFLOAT dans les bases de données DB2 et le type de données NUMBER dans les bases de données Oracle ont une plus grande portée et ils peuvent avoir un format différent de celui des types de données d'encapsuleur JDBC correspondants. Le mappage à un type de données DECFLOAT ou NUMBER peut produire des résultats inexacts.

A propos de cette tâche

Vous pouvez configurer le serveur fédéré pour accéder aux sources de données JDBC en utilisant le Centre de contrôle DB2 ou en exécutant des instructions SQL sur la ligne de commande DB2. Le Centre de contrôle DB2 contient un assistant qui vous aide pas à pas à configurer les objets fédérés nécessaires.

Procédure

Pour ajouter des sources de données JDBC à un serveur fédéré :

1. Préparez le serveur fédéré pour accéder aux sources de données via JDBC.
2. Enregistrez l'encapsuleur JDBC.
3. Enregistrez les définitions de serveur d'une source de donnée JDBC.
4. Créez un mappage utilisateur pour une source de données JDBC.
5. Testez la connexion au serveur de source de données JDBC.
6. Enregistrez les pseudonymes des tables et vues de source de données JDBC.

Préparation du serveur fédéré pour accéder aux sources de données via JDBC

Le serveur fédéré doit pouvoir accéder aux sources de données JDBC. Pour préparer le serveur fédéré, vous devez déterminer si vous devez définir la variable d'environnement CLASSPATH.

Avant de commencer

Si vous utilisez un pilote JDBC autre que le pilote JDBC par défaut DB2 dans le fichier db2jcc.jar, il se peut que vous deviez ajouter les informations du pilote JDBC dans la variable d'environnement CLASSPATH. Vous pouvez éventuellement spécifier vos package de pilote JDBC en utilisant le paramètre DRIVER_PACKAGE de l'instruction CREATE SERVER lorsque vous enregistrez les définitions de serveur.

Procédure

Pour définir la variable d'environnement CLASSPATH :

Enregistrez le fichier Java .jar qui contient le pilote JDBC dans la variable d'environnement CLASSPATH.

Option	Description
Pour Linux et UNIX	Exécutez la commande export pour enregistrer le pilote JDBC. Par exemple, si vous spécifiez le pilote JDBC DB2, exécutez la commande suivante : <pre>export CLASSPATH=\$CLASSPATH:db2_instance_dir/sql1lib/java/db2jcc.jar</pre> , où <i>db2_instance_dir</i> correspond au chemin de fichier qualifié complet dans lequel est installée l'instance du système de base de données DB2.

Option	Description
Pour Windows	<p>Définissez le pilote JDBC dans la variable d'environnement système CLASSPATH :</p> <ol style="list-style-type: none"> 1. Connectez-vous en tant qu'administrateur. 2. Ouvrez le Panneau de configuration et accédez à Système → Variables d'environnement. 3. Ajoutez le nom de fichier et le répertoire du pilote JDBC à la variable système CLASSPATH. Séparez la nouvelle entrée des entrées existantes en utilisant un point-virgule. <p>Reportez-vous à la documentation du pilote JDBC pour plus d'informations sur l'enregistrement des variables d'environnement système.</p>

Après avoir effectué cette tâche, vous devez enregistrer l'encapsuleur.

Enregistrement de l'encapsuleur JDBC

Vous devez enregistrer un encapsuleur pour pouvoir accéder aux sources de données JDBC. Les serveurs fédérés utilisent des encapsuleurs pour dialoguer avec les sources de données et en extraire des données. Les encapsuleurs sont mis en oeuvre comme un ensemble de fichiers de bibliothèque.

Procédure

Pour enregistrer l'encapsuleur JDBC :

Utilisez l'une des méthodes suivantes :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .
Exécutez l'instruction CREATE WRAPPER et spécifiez le nom par défaut de l'encapsuleur JDBC.	<p>Par exemple :</p> <pre>CREATE WRAPPER JDBC;</pre> <p>Lorsque vous enregistrez l'encapsuleur en utilisant le nom par défaut, JDBC, le serveur fédéré utilise automatiquement la bibliothèque d'encapsuleur JDBC correspondant au système d'exploitation sous lequel le serveur fédéré est exécuté.</p>
Exécutez l'instruction CREATE WRAPPER et spécifiez un nom alternatif pour l'encapsuleur JDBC.	<p>En cas de conflit entre le nom d'encapsuleur par défaut et un nom d'encapsuleur existant dans la base de données fédérée, vous pouvez remplacer le nom d'encapsuleur par défaut par un nom que vous choisissez. Lorsque vous utilisez un nom différent du nom par défaut, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER.</p> <p>Par exemple, pour enregistrer l'encapsuleur <i>jdbc_Wrapper</i> sur un serveur fédéré qui utilise AIX, exécutez l'instruction suivante :</p> <pre>CREATE WRAPPER jdbc_Wrapper LIBRARY 'libdb2rcjdbc.a';</pre> <p>Le fichier de bibliothèque d'encapsuleur que vous définissez dépend du système d'exploitation du serveur fédéré.</p>

Après avoir effectué cette tâche, vous devez enregistrer les définitions de serveur.

Fichiers de bibliothèque de l'encapsuleur JDBC

Les fichiers de bibliothèque de l'encapsuleur JDBC sont ajoutés au serveur lorsque vous installez l'encapsuleur.

Lorsque vous installez l'encapsuleur JDBC, les fichiers de bibliothèque sont ajoutés au chemin de répertoire par défaut. Si, par exemple, le serveur fédéré est exécuté sous AIX, les fichiers de bibliothèque libdb2rcjdbc.a, libdb2rcjdbcF.a, libdb2rcjdbcU.a et db2qgjdbc.jar de l'encapsuleur sont ajoutés au chemin de répertoire. Le fichier de bibliothèque d'encapsuleur par défaut est libdb2rcjdbc.a. Les autres fichiers de bibliothèque d'encapsuleur sont utilisés en interne par l'encapsuleur JDBC.

Si vous n'utilisez pas le nom d'encapsuleur par défaut lorsque vous enregistrez un encapsuleur, vous devez spécifier le paramètre **LIBRARY** dans l'instruction CREATE WRAPPER.

Utilisez les chemins de répertoire et les noms de fichier de bibliothèque d'encapsuleur par défaut suivants pour spécifier le paramètre **LIBRARY** dans l'instruction CREATE WRAPPER :

Tableau 22. Chemins de répertoire de bibliothèque client JDBC et noms de fichier

Système d'exploitation	Chemin de répertoire	Fichier de bibliothèque d'encapsuleur
AIX	/usr/opt/install_path/lib32/ /usr/opt/chemin_installation/lib64/	libdb2rcjdbc.a
Linux	/opt/IBM/db2/chemin_installation/lib32 /opt/IBM/db2/chemin_installation/lib64	libdb2rcjdbc.so
Solaris	/opt/IBM/db2/chemin_installation/lib32 /opt/IBM/db2/chemin_installation/lib64	libdb2rcjdbc.so
Windows	%DB2PATH%\bin	db2rcjdbc.dll

, où *install_path* est le chemin de répertoire dans lequel la fédération est installée sous Linux ou UNIX.

Instruction CREATE WRAPPER - Exemples pour l'encapsuleur JDBC

Utilisez l'instruction CREATE WRAPPER pour enregistrer l'encapsuleur JDBC.

Pour enregistrer un encapsuleur avec le nom par défaut sur un serveur fédéré, exécutez l'instruction CREATE WRAPPER avec le nom d'encapsuleur JDBC, comme dans l'exemple suivant :

```
CREATE WRAPPER JDBC;
```

Dans cet exemple, *jdbc_Wrapper* est le nom alternatif que vous affectez à l'encapsuleur que vous enregistrez dans la base de données fédérée.

Serveurs fédérés Linux et Solaris

L'exemple suivant montre comment enregistrer un encapsuleur avec un nom alternatif :

```
CREATE WRAPPER jdbc_Wrapper LIBRARY 'libdb2rcjdbc.so';
```

Serveur fédéré AIX

L'exemple suivant montre comment enregistrer un encapsuleur avec un nom alternatif :

```
CREATE WRAPPER jdbcr_wrapper LIBRARY 'libdb2rcjdbc.a';
```

Serveur fédéré Windows

L'exemple suivant montre comment enregistrer un encapsuleur avec un nom alternatif :

```
CREATE WRAPPER jdbcr_wrapper LIBRARY 'db2rcjdbc.dll';
```

Enregistrement des définitions de serveur des sources de données JDBC

Vous devez enregistrer chaque serveur JDBC auquel vous voulez accéder dans la base de données fédérée.

Avant de commencer

Si vous utilisez un pilote JDBC autre que le pilote JDBC par défaut DB2, il se peut que vous deviez définir la variable d'environnement CLASSPATH pour spécifier les fichiers de package de pilote JDBC. Vous pouvez éventuellement spécifier vos packages de pilote JDBC dans l'instruction CREATE SERVER avec l'option de serveur DRIVER_PACKAGE.

Procédure

Pour enregistrer le serveur JDBC, vous devez spécifier le nom du package du pilote JDBC depuis la bibliothèque de pilotes JDBC et la chaîne de connexion du serveur distant.

Pour enregistrer une définition de serveur pour une source de données JDBC :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .
Exécutez l'instruction CREATE SERVER.	Exécutez l'instruction CREATE SERVER pour chaque serveur à enregistrer. Par exemple : <pre>CREATE SERVER <i>server_Definition_Name</i> TYPE <i>jdbcr_datasource_Type</i> VERSION <i>version_Number</i> WRAPPER <i>jdbcr_wrapper_Name</i> OPTIONS (DRIVER_CLASS '<i>jdbcr_driver_Class_Path</i>', URL '<i>jdbcr_url_Connect_String</i>');</pre> Important : Lorsque vous exécutez l'instruction CREATE SERVER, elle ne crée pas une connexion à la source de données tant que vous n'exécutez pas l'instruction CREATE NICKNAME. Si vous spécifiez des informations de connexion erronées dans le paramètre OPTIONS, vous ne recevez aucun message d'erreur jusqu'à ce que vous exécutiez l'instruction CREATE NICKNAME ou passe-système.

Après avoir effectué cette tâche, vous devez créer un mappage utilisateur.

Instruction CREATE SERVER - Exemples pour l'encapsuleur JDBC

Utilisez l'instruction CREATE SERVER pour enregistrer les définitions de serveur de pour l'encapsuleur JDBC. Cet exemple indique les paramètres nécessaires et des paramètres de serveur supplémentaires.

L'exemple suivant explique comment enregistrer une définition de serveur pour une source de données DB2 en exécutant l'instruction CREATE SERVER :

```
CREATE SERVER jdbc_server1
  TYPE JDBC
  VERSION 3.0
  WRAPPER jdbc_wrapper1
  OPTIONS (
    DRIVER_PACKAGE '/home/My_LIB/JDBC_driver/derbyclient.jar',
    DRIVER_CLASS 'com.ibm.db2.jcc.DB2Driver',
    URL 'jdbc:db2://server.example.com:50471/testdb');
```

Paramètres

jdbc_server1

Spécifie le nom que vous attribuez au serveur de source de données JDBC. Les noms de définition de serveur dupliqués ne sont pas autorisés.

TYPE JDBC

Spécifie le type du serveur de source de données auquel vous voulez accéder. Ce paramètre est facultatif.

VERSION 3.0

Spécifie la version de la source de données JDBC à laquelle vous voulez accéder. Ce paramètre est facultatif.

WRAPPER jdbc_wrapper1

Spécifie le nom d'encapsuleur que vous avez défini dans l'instruction CREATE WRAPPER.

DRIVER_PACKAGE '/home/My_LIB/JDBC_driver/derbyclient.jar'

Spécifie les packages de pilote JDBC.

DRIVER_CLASS 'com.ibm.db2.jcc.DB2Driver'

Spécifie la bibliothèque des pilotes JDBC.

URL 'jdbc:db2://matthaus.cn.ibm.com:50471/testdb'

Spécifie la chaîne de connexion JDBC du serveur distant.

Paramètres de serveur

Lorsque vous créez la définition du serveur, vous pouvez spécifier des options de serveur supplémentaires dans l'instruction CREATE SERVER. Les options de serveur peuvent inclure des options de serveur générales et des paramètres de serveur JDBC.

En règle générale, les valeurs par défaut des paramètres de serveur ont une fonctionnalité limitée. Vous pouvez utiliser les paramètres de serveur pour optimiser la configuration.

Pour accéder aux sources de données JDBC, vous devez spécifier les paramètres de serveur DRIVER_CLASS et URL dans l'instruction CREATE SERVER. Les

paramètres de serveur DRIVER_PACKAGE et JDBC_LOG sont facultatifs. La syntaxe du paramètre OPTIONS ci-dessous spécifie tous les paramètres de serveur JDBC :

```
OPTIONS (  
  DRIVER_PACKAGE '/path1/file1.jar: /path2/file2.jar',  
  DRIVER_CLASS 'com.ibm.db2.jcc.DB2Driver',  
  URL 'jdbc:db2://server.example.com:50471/testdb',  
  JDBC_LOG 'Y');
```

Paramètres

DRIVER_PACKAGE 'path1/file1.jar: path2/file2.jar'

Spécifie les packages de pilote JDBC et définit la variable d'environnement CLASSPATH.

DRIVER_CLASS 'com.ibm.db2.jcc.DB2Driver'

Spécifie la bibliothèque des pilotes JDBC DB2.

URL 'jdbc:db2://server.example.com:50471/testdb'

Spécifie la chaîne de connexion JDBC qui est constituée de trois parties séparées par deux points :

- Protocole de base de données
- Nom de type de base de données ou nom de pilote de connectivité
- Identité de base de données via un alias ou un sous-nom

Exemples

Pour les bases de données DB2

```
jdbc:db2://server.example.com:50471/testdb
```

, où jdbc correspond au protocole, db2 représente le type de base de données et

//server.example.com:50471/testdb indique l'alias de base de données qui fait référence à une entrée du catalogue de base de données DB2 sur le client DB2.

Pour Oracle

```
jdbc:oracle:thin:@//myhost:1521/orcl
```

, où jdbc correspond au protocole, oracle représente le type de base de données et thin:@//myhost:1521/orcl définit les informations du client JDBC Oracle pour accéder au serveur Oracle.

JDBC_LOG 'Y'

Indique de créer des fichiers journaux pour le traçage des erreurs. La valeur par défaut de cette option de serveur est N.

Exemple

```
CREATE SERVER jdbc_server1  
  TYPE JDBC  
  VERSION 3.0  
  WRAPPER jdbc_wrapper1  
  OPTIONS (  
    DRIVER_PACKAGE '/home2/JDBC_driver/derbyclient.jar',  
    DRIVER_CLASS 'org.apache.derby.jdbc.ClientDriver',  
    URL 'jdbc:derby://9.181.139.129:1527/testdb9;create=true;',  
    JDBC_LOG 'Y');
```

Création de mappages utilisateur pour les sources de données JDBC

Vous devez définir une association (mappage utilisateur) entre chaque ID utilisateur de serveur fédéré et l'ID utilisateur de source de données correspondant.

A propos de cette tâche

Lorsque vous tentez d'accéder à un serveur JDBC, le serveur fédéré établit une connexion au serveur JDBC en utilisant un ID utilisateur et un mot de passe pour la source de données.

Créez un mappage utilisateur pour chaque ID utilisateur qui accède au système fédéré pour envoyer des requêtes réparties à la source de données JDBC.

Procédure

Pour créer des mappages utilisateur pour les sources de données JDBC :

Exécutez l'instruction `CREATE USER MAPPING` pour associer un ID utilisateur local à l'ID utilisateur et au mot de passe de la source de données JDBC :

```
CREATE USER MAPPING FOR local_userID
  SERVER server_definition_name
  OPTIONS (
    REMOTE_AUTHID 'remote_userID',
    REMOTE_PASSWORD 'remote_password');
```

Les paramètres de mappage `REMOTE_AUTHID` et `REMOTE_PASSWORD` sont nécessaires.

Après avoir effectué cette tâche, vous pouvez tester la connexion à la source de données JDBC.

Instruction `CREATE USER MAPPING` - Exemples pour l'encapsuleur JDBC

Cet exemple explique comment utiliser le registre spécial DB2 USER avec l'instruction `CREATE USER MAPPING`.

L'exemple suivant explique comment associer un identificateur utilisateur fédéré à un ID utilisateur et un mot de passe de source de données JDBC :

```
CREATE USER MAPPING FOR arturo
  SERVER jdbc_server1
  OPTIONS (
    REMOTE_AUTHID 'art',
    REMOTE_PASSWORD 'red4blue');
```

Paramètres

SERVER arturo

Spécifie l'identificateur utilisateur local que vous associez à l'ID utilisateur et au mot de passe distants qui sont définis au niveau de la source de données JDBC.

OPTIONS jdbc_server1

Spécifie le nom de définition de serveur que vous avez défini dans l'instruction `CREATE SERVER` pour la source de données JDBC.

REMOTE_AUTHID 'art'

Spécifie l'ID utilisateur distant auquel vous associez arturo. La valeur est sensible à la casse si vous n'avez pas affecté 'U' ou 'L' au paramètre de serveur FOLD_ID dans l'instruction CREATE SERVER.

REMOTE_PASSWORD 'red4blue'

Spécifie le mot de passe distant associé à 'art'. La valeur est sensible à la casse si vous n'avez pas affecté 'U' ou 'L' à l'option de serveur FOLD_PW dans l'instruction CREATE SERVER.

Registre spécial DB2 USER

Vous pouvez utiliser le registre spécial DB2 USER pour associer l'identificateur utilisateur de la personne qui exécute l'instruction CREATE USER MAPPING à l'identificateur utilisateur de source de données défini dans l'option utilisateur REMOTE_AUTHID.

L'exemple suivant montre une instruction CREATE USER MAPPING contenant le registre spécial USER :

```
CREATE USER MAPPING FOR USER
  SERVER jdbc_server1
  OPTIONS (
    REMOTE_AUTHID 'art',
    REMOTE_PASSWORD 'red4blue');
```

Test des connexions aux serveurs de sources de données JDBC

Vous pouvez tester la connexion au serveur de source de données JDBC en utilisant la définition de serveur et les mappages utilisateur que vous avez spécifiés.

Procédure

Pour tester la connexion au serveur de source de données JDBC :

Ouvrez une session passe-système et exécutez l'instruction SELECT sur les tables du système de source de données JDBC. Si l'instruction SELECT renvoie un nombre, cela implique que la définition de serveur et les mappages utilisateur sont correctement définis.

```
SET PASSTHRU nom_définition_serveur
SELECT count(*) FROM nom_schéma.nom_table
SET PASSTHRU RESET
```

Si l'instruction SELECT envoie une erreur, identifiez et corrigez l'erreur de connexion.

Après avoir effectué cette tâche, vous devez enregistrer les pseudonymes des tables et des vues de la source de données JDBC.

Identification et résolution des erreurs de connexion à la source de données

Une connexion de test au serveur de source de données peut renvoyer une erreur pour plusieurs raisons. Il y a des mesures que vous pouvez prendre pour déterminer pourquoi l'erreur s'est produite.

Symptôme

Une erreur est renvoyée lorsque vous essayez de vous connecter à la source de données.

Cause

Il y a plusieurs causes possibles à un incident de connexion.

Résolution de l'incident

Pour identifier et résoudre les erreurs de connexion à la source de données, recherchez les problèmes éventuels avec les éléments suivants :

- Vérifiez que la source de données est disponible.
- S'il y a lieu, vérifiez que le serveur de source de données est configuré pour les connexions entrantes.
- Vérifiez que les paramètres de votre mappage utilisateur pour les options REMOTE_AUTHID et REMOTE_PASSWORD sont valides pour les connexions à la source de données. Modifiez le mappage utilisateur ou créez un autre mappage utilisateur si nécessaire.
- S'il y a lieu, vérifiez que le logiciel client de la source de données sur le serveur fédéré est installé et configuré correctement pour se connecter à la source de données.
- Pour les sources de données ODBC, vérifiez que le pilote ODBC sur le serveur fédéré est installé et configuré correctement pour se connecter au serveur de source de données ODBC. Sur les serveurs fédérés qui fonctionnent sous Windows, utilisez l'outil Administrateur de sources de données ODBC pour vérifier le pilote. Sur les serveurs fédérés qui fonctionnent sous UNIX, consultez la documentation du fournisseur de client ODBC.
- Vérifiez que les valeurs des variables définies sur le serveur fédéré sont appropriées à la source de données. Ces variables sont les variables d'environnement système, les variables du fichier db2dj.ini et les variables du registre de profil DB2 (db2set).
- Vérifiez votre définition de serveur. Si nécessaire, supprimez la définition de serveur et recréez-la.

Enregistrements des pseudonymes des tables et des vues de source de données JDBC

Pour chaque définition de serveur JDBC que vous enregistrez, vous devez enregistrer un pseudonyme pour chaque table ou vue à laquelle vous voulez accéder. Utilisez ces pseudonymes et non pas les noms des objets de source de données lorsque vous interrogez les sources de données JDBC.

Avant de commencer

Mettez à jour les statistiques sur la source de données JDBC avant d'enregistrer un pseudonyme. La base de données fédérée exploite les statistiques du catalogue de la source de données pour optimiser le traitement des requêtes. Utilisez la commande de source de données équivalant à la commande RUNSTATS DB2 pour mettre à jour les statistiques de la source de données.

Procédure

Pour enregistrer un pseudonyme pour une table ou une vue de source de données JDBC :

Utilisez l'une des méthodes suivantes :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .
Exécutez l'instruction CREATE NICKNAME .	Par exemple : <pre>CREATE NICKNAME <i>pseudonyme</i> FOR <i>nom_définition_serveur</i>."<i>schéma_distant</i>". <i>"table.distante"</i>;</pre> <p>La longueur des pseudonymes peut aller jusqu'à 128 caractères.</p>

Lorsque vous créez le pseudonyme, le serveur fédéré interroge le catalogue de la source de données. Cette requête teste la connexion à la table ou la vue de la source de données. Si la connexion ne fonctionne pas, vous recevez un message d'erreur.

Répétez cette étape pour chaque table ou vue JDBC pour laquelle vous voulez créer un pseudonyme.

Instruction CREATE NICKNAME - Exemples pour l'encapsuleur JDBC

L'exemple montre comment enregistrer un pseudonyme pour une table ou une vue JDBC en utilisant l'instruction CREATE NICKNAME.

Cette instruction spécifie la définition de serveur et le schéma et la table distants :

```
CREATE NICKNAME cust_europe FOR jdbc_server."vinnie"."italy"
```

cust_europe

Pseudonyme unique qui identifie la table ou la vue JDBC. Le pseudonyme doit être unique dans le schéma.

Important : Le pseudonyme est constitué de deux parties : du schéma et du pseudonyme. Si vous omettez le schéma lors de la création du pseudonyme, le schéma du pseudonyme correspond à l'identificateur utilisateur qui enregistre le pseudonyme.

Si la source de données JDBC ne prend pas en charge les schémas, omettez le schéma dans l'instruction CREATE NICKNAME. Par exemple :

```
CREATE NICKNAME cust_europe FOR jdbc_server."italy"
```

jdbc_server."vinnie"."italy"

Identificateur en trois parties pour l'objet distant :

jdbc_server

Nom de définition de serveur que vous avez affecté au serveur de source de données JDBC dans l'instruction CREATE SERVER.

vinnie ID utilisateur du propriétaire de la table ou de la vue.

italy Nom de la table ou de la vue distante à laquelle vous voulez accéder.

Le serveur fédéré met les noms des schémas et des tables JDBC en majuscules si vous ne les placez pas entre guillemets doubles.

Configuration de l'accès aux sources de données Microsoft SQL Server

Pour configurer le serveur fédéré pour accéder aux sources de données Microsoft SQL Server, vous devez fournir au serveur fédéré des informations sur les sources de données et les objets auxquels vous voulez accéder.

Avant de commencer

- Le pilote ODBC doit être installé et configuré sur le serveur fédéré.
- La fédération doit être installée sur un serveur qui fait office de serveur fédéré.
- Vérifiez la configuration du serveur fédéré.
- Vérifiez le paramètre fédéré pour vérifier que la fédération est activée.

Vous pouvez configurer le serveur fédéré pour accéder aux sources de données Microsoft SQL Server à l'aide du Centre de contrôle DB2 ou en émettant des instructions SQL sur la ligne de commande DB2.

Procédure

Pour configurer l'accès aux sources de données Microsoft SQL Server :

1. Utilisez une des méthodes suivantes pour préparer le serveur fédéré et la base de données fédérée en fonction de votre système d'exploitation.
 - Préparez le serveur fédéré et la base de données fédérée (Windows).
 - Préparez le serveur fédéré et la base de données fédérée (UNIX)
2. Définissez les variables d'environnement pour l'encapsuleur Microsoft SQL Server.
3. Enregistrez l'encapsuleur.
4. Enregistrez la définition de serveur.
5. Créez les mappages utilisateur.
6. Testez la connexion au serveur distant Microsoft SQL Server
7. Enregistrez les pseudonymes pour les tables et vues Microsoft SQL Server.

Préparation du serveur fédéré pour l'accès aux sources de données Microsoft SQL Server (Windows)

Sur les serveurs fédérés qui fonctionnent sous Windows, le serveur fédéré doit pouvoir accéder aux sources de données Microsoft SQL Server. Pour préparer le serveur fédéré, vous devez vérifier les paramètres DSN système ODBC et tester la connexion aux sources de données Microsoft SQL Server.

Procédure

Pour préparer le serveur fédéré à accéder aux sources de données Microsoft SQL Server :

1. Vérifiez que le DSN système ODBC est défini pour se connecter à la source de données Microsoft SQL Server. Dans le Panneau de configuration, trouvez l'entrée DSN existante pour le serveur distant Microsoft SQL Server ou créez une entrée DSN.

L'entrée DSN pour le serveur distant Microsoft SQL Server est la valeur que vous utilisez pour l'option de serveur NODE lorsque vous enregistrez la définition de serveur dans la base de données fédérée.

2. Utilisez une des méthodes suivantes pour tester la connexion à la source de données Microsoft SQL Server :
 - Choisissez **Configurer** dans la fenêtre Administrateur de sources de données ODBC.
 - Utilisez l'utilitaire d'interrogation Microsoft SQL Server.

Une fois cette tâche terminée, vous pouvez définir les variables d'environnement.

Préparation du serveur fédéré pour l'accès aux sources de données du serveur Microsoft SQL (Linux, UNIX)

Sur les serveurs fédérés qui fonctionnent sous Linux ou UNIX, le serveur fédéré doit pouvoir accéder aux sources de données Microsoft SQL Server. Pour préparer le serveur fédéré, vous devez vérifier les paramètres dans le fichier `odbc.ini`, créer des liens symboliques et tester la connexion aux sources de données Microsoft SQL Server.

Procédure

Pour préparer le serveur fédéré à accéder aux sources de données Microsoft SQL Server :

1. Vérifiez que le fichier `odbc.ini` est mis à jour sur le serveur fédéré. Si le fichier `odbc.ini` n'existe pas sur le serveur fédéré, vous pouvez le créer dans un éditeur de texte. Consultez la documentation du fournisseur du client ODBC pour toute information sur le fichier `odbc.ini`.

A faire : Placez le fichier `odbc.ini` ou une copie de ce fichier dans le répertoire personnel du propriétaire de l'instance DB2, afin de vérifier qu'il est accessible si le propriétaire d'instance n'est pas le superutilisateur.

2. Vérifiez que le chemin du fichier `odbc.ini` figure dans la variable d'environnement `ODBCINI`.

A partir d'une invite de commande du système d'exploitation, émettez la commande suivante :

```
export ODBCINI=$HOME/.odbc.ini
```

3. Créez les liens symboliques appropriés :

Système d'exploitation du serveur fédéré	Etape
Linux	<p>Créez les liens symboliques suivants :</p> <pre>ln -s \$DJX_ODBC_LIBRARY_PATH/./ /locale/usr/local/locale</pre> <pre>ln -s \$DJX_ODBC_LIBRARY_PATH/libodbcinst.so/ usr/lib/libodbcinst.so</pre> <p>Si vous utilisez le pilote ODBC DataDirect Technologies Connect, vérifiez le nom de bibliothèque et créez le lien symbolique. Le nom de la bibliothèque varie suivant la version du pilote et suivant que vous utilisez un pilote 32 bits ou 64 bits.</p> <p>Par exemple, si vous utilisez DataDirect version 4.2, vous créez le lien suivant :</p> <pre>ln -s \$DJX_ODBC_LIBRARY_PATH/libivicu19.so/ usr/lib/libivicu19.so</pre> <p>Si vous utilisez DataDirect version 5.0, vous créez le lien suivant :</p> <pre>ln -s \$DJX_ODBC_LIBRARY_PATH/libivicu20.so/ usr/lib/libivicu20.so</pre> <p>Si vous utilisez DataDirect et que vous n'incluez pas le lien symbolique, CREATE WRAPPER MSSQLODBC3 pourrait échouer avec le message d'erreur suivant :</p> <p>SQL10013N Le nom de bibliothèque indiqué n'a pas pu être chargé.</p>
Solaris	<p>Créez le lien symbolique suivant :</p> <pre>ln -s \$DJX_ODBC_LIBRARY_PATH/./ locale \$HOME/sqllib/locale</pre> <p>\$HOME est le répertoire de base du propriétaire d'instance DB2.</p>

4. Exécutez le script /opt/odbc/odbc.sh. Ce script configure plusieurs variables d'environnement spécifiques au système d'exploitation.
5. Testez la connexion du serveur fédéré à la source de données Microsoft SQL Server à l'aide de l'utilitaire demoodbc de DataDirect Connect ODBC. L'utilitaire demoodbc se trouve dans le sous-répertoire /demo des bibliothèques DataDirect Connect ODBC.

Une fois cette tâche terminée, vous pouvez définir les variables d'environnement.

Définition des variables d'environnement Microsoft SQL Server

Les variables d'environnement Microsoft SQL Server doivent être définies dans le fichier db2dj.ini sur le serveur fédéré.

Restrictions

Lisez les restrictions pour le fichier db2dj.ini avant de commencer cette tâche.

Le fichier db2dj.ini contient des informations de configuration sur le pilote ODBC Microsoft SQL Server installé sur votre serveur fédéré.

Il y a des variables d'environnement obligatoires et facultatives pour les sources de données Microsoft SQL Server.

Si vous avez installé le logiciel client Microsoft SQL Server avant d'installer l'encapsuleur Microsoft SQL Server, les variables d'environnement Microsoft SQL Server obligatoires sont définies dans le fichier db2dj.ini.

Vous devez définir les variables d'environnement à l'aide des étapes de cette tâche si vous n'avez pas installé le logiciel client Microsoft SQL Server avant d'installer l'encapsuleur Microsoft SQL Server ou que vous voulez définir l'une des variables d'environnement facultatives.

Procédure

Pour définir les variables d'environnement Microsoft SQL Server :

1. Utilisez l'une des méthodes suivantes :

Méthode	Etape
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .
Définissez automatiquement les variables d'environnement.	Exécutez l'assistant d'installation IBM InfoSphere Federation Server. Suivez les instructions de l'assistant. Important : Définissez les variables d'environnement obligatoires en exécutant l'assistant d'installation. Les variables d'environnement facultatives doivent être définies manuellement.
Définissez manuellement les variables d'environnement.	Modifiez le fichier db2dj.ini. <ul style="list-style-type: none">Sur les serveurs fédérés qui fonctionnent sous UNIX, ce fichier se trouve dans le répertoire sqllib/cfg.Sur les serveurs fédérés qui fonctionnent sous Windows, ce fichier se trouve dans le répertoire %DB2PATH%\cfg. Si le fichier n'existe pas, vous pouvez créer un fichier portant le nom db2dj.ini à l'aide de tout éditeur de texte. Dans le fichier db2dj.ini, vous devez indiquer le chemin qualifié complet dans la valeur des variables d'environnement, sinon vous rencontrerez des erreurs. Par exemple : DJX_ODBC_LIBRARY_PATH=/opt/odbc/lib ODBCINI=/opt/odbc/.odbc.ini

2. Pour garantir que les variables d'environnement soient définies sur le serveur fédéré, recyclez l'instance DB2 avec les commandes suivantes :

```
db2stop  
db2start
```

Une fois cette tâche terminée, vous pouvez enregistrer l'encapsuleur.

Variables d'environnement Microsoft SQL Server

Il y a des variables d'environnement obligatoires et facultatives pour les sources de données Microsoft SQL Server. Ces variables sont définies dans le fichier db2dj.ini.

Les variables d'environnement suivantes sont valides pour Microsoft SQL Server :

- DJX_ODBC_LIBRARY_PATH
- ODBCINI
- LD_LIBRARY_PATH (Solaris uniquement)

Description des variables

DJX_ODBC_LIBRARY_PATH

Indique le chemin de répertoire des fichiers de bibliothèque ODBC. Cette variable doit aussi être indiquée sur les serveurs fédérés qui fonctionnent sous Solaris.

Par exemple :

```
DJX_ODBC_LIBRARY_PATH=répertoire_unité_ODBC/lib
```

répertoire_unité_ODBC est le chemin du répertoire d'installation du pilote ODBC.

ODBCINI

Indique le chemin de répertoire dans lequel votre fichier de configuration ODBC (odbc.ini) se trouve.

Par exemple :

```
ODBCINI=/home/db2inst1/.odbc.ini
```

Ne définissez pas la variable d'environnement ODBCINI comme une variable système.

LD_LIBRARY_PATH (Solaris uniquement)

Sur les serveurs fédérés qui fonctionnent sous Solaris, indiquez le chemin de répertoire des fichiers de bibliothèque ODBC.

Par exemple :

```
LD_LIBRARY_PATH=répertoire_unité_ODBC/lib
```

Enregistrement de l'encapsuleur Microsoft SQL Server

Vous devez enregistrer un encapsuleur pour accéder aux sources de données Microsoft SQL Server. Les serveurs fédérés utilisent des encapsuleurs pour dialoguer avec les sources de données et en extraire des données. Les encapsuleurs sont mis en oeuvre comme un ensemble de fichiers de bibliothèque.

Procédure

Pour enregistrer l'encapsuleur Microsoft SQL Server :

Utilisez l'une des méthodes suivantes :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .

Méthode	Description
<p>Émettez l'instruction CREATE WRAPPER et indiquez le nom par défaut pour l'encapsuleur Microsoft SQL Server.</p>	<p>Par exemple :</p> <pre>CREATE WRAPPER MSSQLODBC3;</pre> <p>A faire : Lorsque vous enregistrez l'encapsuleur à l'aide du nom par défaut MSSQLODBC3, le serveur fédéré utilise automatiquement la bibliothèque de l'encapsuleur Microsoft SQL Server appropriée au système d'exploitation sous lequel fonctionne votre serveur fédéré.</p> <p>En cas de conflit entre le nom d'encapsuleur par défaut et un nom d'encapsuleur existant dans la base de données fédérée, vous pouvez remplacer le nom d'encapsuleur par défaut par un nom que vous choisissez. Lorsque vous utilisez un nom différent du nom par défaut, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER.</p> <p>Par exemple, pour enregistrer un encapsuleur avec le nom encapsuleur_serveursql sur un serveur fédéré qui utilise AIX, émettez l'instruction suivante :</p> <pre>CREATE WRAPPER encapsuleur_serveursql LIBRARY 'libdb2mssql3.a';</pre> <p>Le fichier de bibliothèque de l'encapsuleur que vous indiquez dépend du système d'exploitation du serveur fédéré.</p>

Une fois cette tâche terminée, vous pouvez enregistrer la définition de serveur.

Fichiers de bibliothèque de l'encapsuleur Microsoft SQL Server

Les fichiers de bibliothèque de l'encapsuleur Microsoft SQL Server sont ajoutés au serveur fédéré lorsque vous installez l'encapsuleur.

Lorsque vous installez l'encapsuleur Microsoft SQL Server, trois fichiers de bibliothèque sont ajoutés dans le chemin de répertoire par défaut. Par exemple, si le serveur fédéré fonctionne sous AIX, les fichiers de bibliothèque de l'encapsuleur qui sont ajoutés dans le chemin de répertoire sont libdb2mssql3.a, libdb2mssql3F.a et libdb2mssql3U.a. Le fichier de bibliothèque de l'encapsuleur par défaut est libdb2mssql3.a. Les autres fichiers de bibliothèque de l'encapsuleur sont utilisés sur le plan interne par l'encapsuleur Microsoft SQL Server.

Si vous n'utilisez pas le nom d'encapsuleur par défaut lorsque vous enregistrez un encapsuleur, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER et indiquer le nom de fichier de bibliothèque de l'encapsuleur par défaut.

Les chemins des répertoires par défaut et les noms de fichier de bibliothèque de l'encapsuleur par défaut sont indiqués dans la table suivante.

Tableau 23. Emplacements et noms de fichier de la bibliothèque du client Microsoft SQL Server

Système d'exploitation	Chemin de répertoire	Nom de fichier de bibliothèque
AIX	/usr/opt/chemin_installation/lib32/ /usr/opt/chemin_installation/lib64/	libdb2mssql3.a
Linux	/opt/IBM/db2/chemin_installation/lib32 /opt/IBM/db2/chemin_installation/lib64	libdb2mssql3.so
Solaris	/opt/IBM/db2/chemin_installation/lib32 /opt/IBM/db2/chemin_installation/lib64	libdb2mssql3.so
Windows	%DB2PATH%\bin	db2mssql3.dll

chemin_installation est le chemin du répertoire dans lequel le serveur fédéré est installé sous UNIX ou Linux.

Enregistrement des définitions de serveur pour une source de données Microsoft SQL Server

Vous devez enregistrer chaque serveur distant Microsoft SQL Server auquel vous voulez accéder dans la base de données fédérée.

Procédure

Pour enregistrer une définition de serveur pour une source de données Microsoft SQL Server :

1. Recherchez le nom du noeud du serveur Microsoft SQL Server.
 - Sur les serveurs fédérés qui fonctionnent sous Windows, le nom du noeud est le nom DSN système que vous avez indiqué pour le serveur distant Microsoft SQL Server auquel vous accédez.
 - Sur les serveurs fédérés qui fonctionnent sous UNIX, le nom du noeud est défini dans le fichier .odbc.ini.

Au début du fichier .odbc.ini se trouve une section intitulée ODBC Data Sources, qui comporte la liste des noeuds. Chacun des noeuds a une section dans le fichier .odbc.ini qui décrit ce noeud.

L'exemple suivant est un fichier .odbc.ini sous AIX. Les noms de noeud sont [rawilson] et [medusa].

```
[ODBC Data Sources]
rawilson=MS SQL Server 2000
medusa=MS SQL Server 2000
[rawilson]
Driver=/opt/odbc/lib/ddmsss20.so
Description=MS SQL Server Driver for AIX
Address=9.112.30.39,1433
[medusa]
Driver=/opt/odbc/lib/ddmsss20.so
Description=MS SQL Server Driver for AIX
Address=9.112.98.123,1433
[ODBC]
InstallDir=/opt/odbc
```

2. Utilisez l'une des méthodes suivantes pour créer la définition de serveur.

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .

Méthode	Description
Emettez l'instruction CREATE SERVER	<p>Par exemple :</p> <pre>CREATE SERVER <i>nom_définition_serveur</i> TYPE MSSQLSERVER VERSION <i>version_number</i> WRAPPER <i>nom_encapsuleur</i> OPTIONS (NODE '<i>nom_noeud</i>', DBNAME '<i>nom_base_de_données</i>');</pre> <p>Bien que les variables '<i>nom_noeud</i>' et '<i>nom_bd</i>' soient indiquées comme options dans l'instruction CREATE SERVER, ces options doivent être indiquées obligatoirement pour les sources de données Microsoft SQL Server.</p> <p>Une fois la définition de serveur enregistrée, utilisez l'instruction ALTER SERVER pour ajouter ou déplacer des options de serveur.</p>

Une fois cette tâche terminée, vous pouvez créer des mappages utilisateur.

Instruction CREATE SERVER - exemples pour l'encapsuleur Microsoft SQL Server

Utiliser l'instruction CREATE SERVER pour enregistrer les définitions de serveur pour l'encapsuleur Microsoft SQL Server. Cette rubrique fournit un exemple complet avec les paramètres obligatoires et un exemple avec des options de serveur supplémentaires.

L'exemple suivant vous montre comment enregistrer une définition de serveur pour un encapsuleur Microsoft SQL Server en émettant l'instruction CREATE SERVER :

```
CREATE SERVER sqlserver TYPE MSSQLSERVER VERSION 2000 WRAPPER nom_encapsuleur
OPTIONS (NODE 'sqlnode', DBNAME 'africa');
```

sqlserver

Nom que vous attribuez au serveur distant Microsoft SQL Server. Les doublons de noms de définition de serveur ne sont pas autorisés.

TYPE MSSQLSERVER

Indique le type de source de données dont vous configurez l'accès. Pour l'encapsuleur Microsoft SQL Server, le type de serveur doit être MSSQLSERVER.

VERSION 2000

Version du serveur de la base Microsoft SQL Server auquel vous voulez accéder.

WRAPPER *nom_encapsuleur*

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

NODE '*sqlnode*'

Nom du noeud sur lequel le serveur distant Microsoft SQL Server se trouve. Sur les serveurs fédérés qui fonctionnent sous Windows, le nom DSN système du serveur distant Microsoft SQL Server auquel vous accédez. Sur les serveurs fédérés qui fonctionnent sous UNIX, le noeud défini dans le fichier .odbc.ini.

Cette valeur fait la distinction entre les majuscules et les minuscules.

Bien que le nom du noeud soit indiqué comme option dans l'instruction CREATE SERVER, il doit être indiqué obligatoirement pour les sources de données Microsoft SQL Server.

DBNAME *'africa'*

Nom de la base de données Microsoft SQL Server à laquelle vous voulez accéder. Cette valeur fait la distinction entre les majuscules et les minuscules.

Bien que le nom de la base de données soit indiqué comme option dans l'instruction CREATE SERVER, il doit être indiqué obligatoirement pour les sources de données Microsoft SQL Server.

Options de serveur

Lorsque vous créez une définition de serveur, vous pouvez indiquer des options de serveur supplémentaires dans l'instruction CREATE SERVER. Les options de serveur peuvent être des options de serveur générales et des options de serveur spécifiques à Microsoft SQL Server.

L'option de serveur COLLATING_SEQUENCE indique si la source de données utilise la même séquence de classement que le serveur fédéré ou bien une séquence de classement différente. Sur un serveur de la base Microsoft SQL Server fonctionnant sous Windows NT® ou Windows 2000, la séquence de classement par défaut n'est pas sensible à la casse (par exemple, 'STEWART' et 'Stewart' seront considérés comme identiques). Pour garantir des résultats corrects du serveur fédéré, définissez l'option de serveur COLLATING_SEQUENCE sur 'I'. Ce paramètre indique que la source de données Microsoft SQL Server n'est pas sensible à la casse.

Le serveur fédéré ne transfère pas les requêtes si les résultats qui sont renvoyés par les sources de données sont différents des résultats renvoyés si la requête est traitée sur le serveur fédéré. Lorsque vous définissez l'option de serveur COLLATING_SEQUENCE sur 'I', le serveur fédéré ne transfère pas les requêtes qui contiennent des données ou expressions de chaîne et contiennent aussi l'une des clauses suivantes, l'un des prédicats ou fonctions suivants :

- Clauses GROUP BY
- Clauses DISTINCT
- Prédicats de base, tels que égale (=)
- Des fonctions d'agrégation, telles que MIN ou MAX

L'exemple suivant montre comment indiquer l'option de serveur COLLATING_SEQUENCE :

```
CREATE SERVER sqlserver TYPE MSSQLSERVER VERSION 2000 WRAPPER mssqlodbc3
OPTIONS (NODE 'sqlnode', DBNAME 'africa', COLLATING_SEQUENCE 'I');
```

Création des mappages utilisateur pour une source de données Microsoft SQL Server

Lorsque vous essayez d'accéder à un serveur distant Microsoft SQL Server, le serveur fédéré établit une connexion au serveur distant Microsoft SQL Server à l'aide d'un ID utilisateur et d'un mot de passe valides pour cette source de données.

Procédure

Pour mapper un identificateur utilisateur local à un ID utilisateur et à un mot de passe Microsoft SQL Server distants :

Emettez une instruction CREATE USER MAPPING.

Par exemple :

```
CREATE USER MAPPING FOR IDutilisateur_local SERVER nom_définition_serveur
    OPTIONS (REMOTE_AUTHID 'IDutilisateur_distant', REMOTE_PASSWORD
'motdepasse_distant');
```

Une fois cette tâche terminée, vous pouvez tester la connexion aux tables et vues Microsoft SQL Server.

Instruction CREATE USER MAPPING - exemples pour l'encapsuleur Microsoft SQL Server

Utiliser l'instruction CREATE USER MAPPING pour mapper un identificateur utilisateur fédéré à un ID utilisateur et à un mot de passe Microsoft SQL Server distants. Cette rubrique fournit un exemple complet avec les paramètres obligatoires et un exemple qui vous montre comment utiliser le registre spécial USER DB2 avec l'instruction CREATE USER MAPPING.

L'exemple suivant montre comment mapper un identificateur utilisateur fédéré à un ID utilisateur et à un mot de passe du serveur distant Microsoft SQL Server :

```
CREATE USER MAPPING FOR elizabeth SERVER sqlserver
    OPTIONS (REMOTE_AUTHID 'liz', REMOTE_PASSWORD 'abc123')
```

elizabeth

Indique l'identificateur utilisateur que vous mappez à un ID utilisateur et à un mot de passe distants, qui sont définis sur le serveur distant Microsoft SQL Server.

SERVER *sqlserver*

Indique le nom de définition de serveur que vous avez enregistré dans l'instruction CREATE SERVER pour le serveur distant Microsoft SQL Server.

REMOTE_AUTHID '*liz*'

Indique l'ID utilisateur distant auquel vous mappez *elizabeth*. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_ID sur 'U' ou 'L' dans l'instruction CREATE SERVER.

REMOTE_PASSWORD '*abc123*'

Indique le mot de passe distant qui est associé à '*liz*'. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_PW sur 'U' ou 'L' dans l'instruction CREATE SERVER.

registre spécial USER DB2

Vous pouvez utiliser le registre spécial USER DB2 pour mapper l'identificateur utilisateur de la personne qui émet l'instruction CREATE USER MAPPING à l'ID utilisateur distant qui est indiqué dans l'option REMOTE_AUTHID.

L'exemple suivant montre une instruction CREATE USER MAPPING qui comporte le registre spécial USER :

```
CREATE USER MAPPING FOR USER SERVER sqlserver
    OPTIONS (REMOTE_AUTHID 'liz', REMOTE_PASSWORD 'abc123');
```

Test de la connexion au serveur distant Microsoft SQL Server

Tester la connexion au serveur distant Microsoft SQL Server pour déterminer si le serveur fédéré est correctement configuré pour accéder aux sources de données Microsoft SQL Server.

Vous pouvez tester la connexion au serveur distant Microsoft SQL Server à l'aide de la définition de serveur et des mappages utilisateur que vous avez définis.

Procédure

Pour tester la connexion au serveur distant Microsoft SQL Server :

Ouvrez une session passe-système et émettez une instruction SELECT sur les tables système Microsoft SQL Server. Si l'instruction SELECT renvoie un comptage, votre définition de serveur et votre mappage utilisateur sont configurés correctement.

Par exemple :

```
SET PASSTHRU nom_définition_serveur
SELECT count(*) FROM dbo.sysobjects
SET PASSTHRU RESET
```

Une fois cette tâche terminée, vous pouvez enregistrer des pseudonymes.

Identification et résolution des erreurs de connexion à la source de données

Une connexion de test au serveur de source de données peut renvoyer une erreur pour plusieurs raisons. Il y a des mesures que vous pouvez prendre pour déterminer pourquoi l'erreur s'est produite.

Symptôme

Une erreur est renvoyée lorsque vous essayez de vous connecter à la source de données.

Cause

Il y a plusieurs causes possibles à un incident de connexion.

Résolution de l'incident

Pour identifier et résoudre les erreurs de connexion à la source de données, recherchez les problèmes éventuels avec les éléments suivants :

- Vérifiez que la source de données est disponible.
- S'il y a lieu, vérifiez que le serveur de source de données est configuré pour les connexions entrantes.
- Vérifiez que les paramètres de votre mappage utilisateur pour les options REMOTE_AUTHID et REMOTE_PASSWORD sont valides pour les connexions à la source de données. Modifiez le mappage utilisateur ou créez un autre mappage utilisateur si nécessaire.
- S'il y a lieu, vérifiez que le logiciel client de la source de données sur le serveur fédéré est installé et configuré correctement pour se connecter à la source de données.
- Pour les sources de données ODBC, vérifiez que le pilote ODBC sur le serveur fédéré est installé et configuré correctement pour se connecter au serveur de source de données ODBC. Sur les serveurs fédérés qui fonctionnent sous

Windows, utilisez l'outil Administrateur de sources de données ODBC pour vérifier le pilote. Sur les serveurs fédérés qui fonctionnent sous UNIX, consultez la documentation du fournisseur de client ODBC.

- Vérifiez que les valeurs des variables définies sur le serveur fédéré sont appropriées à la source de données. Ces variables sont les variables d'environnement système, les variables du fichier db2dj.ini et les variables du registre de profil DB2 (db2set).
- Vérifiez votre définition de serveur. Si nécessaire, supprimez la définition de serveur et recréez-la.

Enregistrement de pseudonymes pour les tables et vues Microsoft SQL Server

Dans chaque définition de serveur Microsoft SQL Server distant que vous enregistrez, vous devez enregistrer un pseudonyme pour chaque table ou vue à laquelle vous voulez accéder. Utilisez ces pseudonymes, au lieu des noms des objets de source de données, lorsque vous interrogez les serveurs distants Microsoft SQL Server.

Avant de commencer

Pour garantir que la base de données fédérée ait des statistiques actualisées et complètes, exécutez la procédure mémorisée `sp_createstats` de Microsoft SQL Server et la commande `CREATE STATISTICS` de Microsoft SQL Server dans la base de données Microsoft SQL Server avant de créer le pseudonyme.

La procédure mémorisée `sp_createstats` collecte les statistiques de toutes les colonnes par défaut dans une table d'une source de données Microsoft SQL Server, mais ne collecte pas de statistiques pour les colonnes qui apparaissent en premier à l'intérieur d'un index. Pour garantir que la base de données fédérée ait des statistiques complètes sur la table Microsoft SQL Server, vous devez aussi utiliser la commande `CREATE STATISTICS` de Microsoft SQL Server pour collecter des statistiques pour chaque colonne qui apparaît en premier dans un index.

Lorsque vous utilisez la commande `CREATE STATISTICS` dans la base de données Microsoft SQL Server, vous devez donner aux statistiques le même nom pour la colonne dans laquelle les statistiques sont collectées. En donnant aux statistiques le même nom que la colonne, vous êtes sûr que lorsque vous enregistrez le pseudonyme avec l'instruction `CREATE NICKNAME`, la base de données fédérée lit les statistiques collectées par la commande `CREATE STATISTICS` de Microsoft SQL Server.

Procédure

Pour enregistrer un pseudonyme pour une table ou une vue Microsoft SQL Server :

Utilisez l'une des méthodes suivantes :

Méthode	Description
Utilisez le Centre de contrôle.	Démarrez l'assistant des objets fédérés. Cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée et cliquez sur Créer des objets fédérés . Suivez les étapes de l'assistant.

Méthode	Description
Emettez l'instruction CREATE NICKNAME.	Par exemple : CREATE NICKNAME <i>pseudonyme</i> FOR <i>nom_définition_serveur</i> ." <i>schéma_distant</i> ". <i>"table.distante"</i> ;

Lorsque vous créez le pseudonyme, le serveur fédéré interroge le catalogue de la source de données à l'aide du pseudonyme. Cette requête teste la connexion à la table, à la vue ou au synonyme de la source de données. Si la connexion ne fonctionne pas, vous recevez un message d'erreur.

Répétez cette étape pour chaque table ou vue Microsoft SQL Server pour laquelle vous voulez créer un pseudonyme.

Instruction CREATE NICKNAME - exemples pour l'encapsuleur Microsoft SQL Server

Utiliser l'instruction CREATE NICKNAME pour enregistrer un pseudonyme pour une table ou une vue Microsoft SQL Server à laquelle vous voulez accéder. Cette rubrique fournit un exemple complet avec les paramètres obligatoires.

L'exemple suivant montre comment enregistrer un pseudonyme pour une table ou une vue Microsoft SQL Server à l'aide de l'instruction CREATE NICKNAME.

```
CREATE NICKNAME cust_africa FOR sqlserver."vinnie"."egypt"
```

cust_africa

Pseudonyme unique utilisé pour identifier la table ou vue Microsoft SQL Server.

Important : Le pseudonyme est un nom composé de deux parties, le schéma et le pseudonyme. Si vous omettez le schéma lorsque vous enregistrez le pseudonyme, le schéma du pseudonyme sera l'identificateur utilisateur de l'utilisateur qui enregistre le pseudonyme.

```
sqlserver."vinnie"."egypt"
```

Identificateur en trois parties pour l'objet distant :

- *sqlserver* est le nom de définition de serveur que vous avez affecté au serveur distant Microsoft SQL Server dans l'instruction CREATE SERVER.
- *vinnie* est l'ID utilisateur du propriétaire de la table ou vue.
- *egypt* est le nom de la table ou vue distante à laquelle vous voulez accéder.

Le serveur fédéré convertit les noms des schémas Microsoft SQL Server et des tables en majuscules à moins que vous ne placiez les noms entre apostrophes.

Utilisation des informations de traçage ODBC pour identifier et résoudre les incidents de connexion aux sources de données Microsoft SQL Server

Si vous rencontrez des problèmes de connexion à la source de données, vous pouvez obtenir des informations de traçage ODBC pour analyser et résoudre les problèmes.

Symptôme

Cependant, l'activation de la fonction de trace a une incidence sur les performances du système. Vous devez désactiver le traçage une fois que vous avez résolu les problèmes de connectivité.

Si vous ne parvenez pas à vous connecter à la source de données avec l'encapsuleur Microsoft SQL Server, l'exécution de la fonction trace pourrait vous aider à diagnostiquer le problème.

Cause

La cause du problème est peut-être une erreur dans la configuration de l'encapsuleur.

Identification de l'incident

Pour diagnostiquer le problème sur un serveur fédéré fonctionnant sous Windows :

1. Dans le Panneau de configuration, ouvrez le dossier **Outils d'administration**.
2. Cliquez sur **Sources de données (ODBC)** pour ouvrir la fenêtre Administrateur de sources de données ODBC.
3. Cliquez sur l'onglet **Traçage**.
4. Cliquez sur **Démarrer le traçage** pour démarrer l'utilitaire de traçage.

Sur un serveur fédéré fonctionnant sous UNIX :

1. Modifiez le fichier `odbc.ini`.

Par exemple, si vous utilisez le pilote DataDirect ODBC 3.x, recherchez l'exemple de fichier `odbc.ini` dans le répertoire du client. Le fichier `odbc.ini` contient un exemple des paramètres qui sont nécessaires pour activer les fichiers de trace :

```
[ODBC]
Trace=1
TraceFile=/home/user1/rép_trace/nomdefichier.xxx
TraceDll=répertoire_unité_ODBC/odbctrac.so
InstallDir=/opt/odbc
```

Pour activer le traçage, définissez la première ligne sur `Trace=1`. Pour désactiver le traçage, définissez la première ligne sur `Trace=0`. La valeur du paramètre `TraceFile` est le chemin et le nom du fichier auquel l'instance de base de données fédérée a un accès en écriture.

Résolution de l'incident

Recherchez d'éventuels incidents dans le fichier de trace.

Sous Windows, ouvrez l'Administrateur de sources de données ODBC et cliquez sur l'onglet **Traçage**. Le chemin du fichier de trace est affiché dans la zone **Chemin du fichier journal**.

Sous UNIX ouvrez le fichier `odbc.ini`. Le chemin du fichier de trace est indiqué par le paramètre `TraceFile`.

Configuration de l'accès aux sources de données ODBC

Pour configurer le serveur fédéré pour accéder aux sources de données ODBC, vous devez fournir au serveur fédéré des informations sur les sources de données et les objets auxquels vous voulez accéder.

Avant de commencer

- Le pilote ODBC doit être installé et configuré sur le serveur qui fait office de serveur fédéré.
- La fédération doit être installée sur le serveur qui fait office de serveur fédéré.
- Vérifiez la configuration du serveur fédéré.
- Vérifiez le paramètre fédéré pour vérifier que la fédération est activée.
- La configuration appropriée des variables d'environnement système, des variables du fichier db2dj.ini et des variables du registre de profil DB2 (db2set). Examinez la documentation fournie par la source de données ODBC pour connaître les variables qui sont obligatoires pour votre client ODBC. La variable d'environnement LIBPATH peut être obligatoire.

Restrictions

- L'encapsuleur ODBC ne peut pas être utilisé pour accéder aux sources de données de la famille DB2. Utilisez l'encapsuleur de l'architecture de base de données pour accéder à des sources de données de la famille DB2.
- L'encapsuleur ODBC ne prend pas en charge les fonctions et instructions suivantes :
 - Instructions LOCK TABLE avec des pseudonymes
 - Fonctions obsolètes sous ODBC 3.x
 - Pilotes X/Open ou SQL/interface de ligne de commande
 - Pseudonymes de procédure mémorisée
 - Mise en application de l'atomicité au niveau de l'instruction à l'aide d'instructions de point de sauvegarde distantes
 - Curseurs WITH HOLD
- Pour les sources de données qui ne prennent pas en charge les opérations de mise à jour et de suppression positionnées, les instructions UPDATE et DELETE positionnées et certaines instructions UPDATE et DELETE recherchées pour un pseudonyme échoueront s'il n'existe pas, dans les colonnes pour lesquelles null n'est pas admis, d'index à entrées uniques dans le pseudonyme ou dans la table distante correspondante. L'erreur SQL30090 avec le code anomalie 21 est renvoyée lorsque ces instructions échouent.
- L'encapsuleur ODBC ne prend pas en charge les instructions INSERT, UPDATE ou DELETE pour les sources de données qui restreignent le nombre d'instructions actives pour chaque connexion. Consultez la documentation de votre source de données pour déterminer si la source de données limite le nombre d'instructions actives pour chaque connexion. L'une des sources de données ODBC auxquelles cette restriction s'applique est IBM Red Brick Warehouse.
- L'encapsuleur ODBC ne prend pas en charge les opérations sur les tables qui contiennent des colonnes avec des types de données qui utilisent des indicateurs de type de données SQL spécifiques au pilote. Parmi les types d'opération qui ne sont pas pris en charge figurent les instructions CREATE NICKNAME et SELECT en mode transparent. L'encapsuleur ODBC prend en charge uniquement les indicateurs de type de données SQL qui sont définis par la norme ODBC dans *Microsoft ODBC Programmer's Reference*.

A propos de cette tâche

Vous pouvez configurer le serveur fédéré pour accéder aux sources de données ODBC à l'aide du Centre de contrôle DB2 ou en émettant des instructions SQL sur la ligne de commande DB2. Le Centre de contrôle DB2 comporte un assistant pour vous guider à travers les étapes nécessaires pour configurer les objets fédérés obligatoires.

Les sources de données accessibles à l'aide de l'API ODBC sont appelées, dans ce texte, sources de données ODBC.

Suivant vos besoins, vous pouvez accéder aux données Excel à l'aide de l'encapsuleur ODBC au lieu de l'encapsuleur Excel. Pour configurer l'encapsuleur ODBC pour accéder aux données Excel, voir Accès aux données Excel à l'aide de l'encapsuleur ODBC.

Recommandation : Pour Microsoft SQL Server, vous devez utiliser l'encapsuleur Microsoft SQL Server, au lieu de l'encapsuleur ODBC, pour accéder aux sources de données Microsoft SQL Server. L'encapsuleur Microsoft SQL Server fournit de meilleures performances d'interrogation et plus de fonctionnalités pour Microsoft SQL Server. Pour des informations sur la configuration de l'encapsuleur Microsoft SQL Server, voir Configuration de l'accès aux sources de données Microsoft SQL Server.

Procédure

Pour ajouter des sources de données ODBC à un serveur fédéré :

1. Utilisez une des méthodes suivantes pour préparer le serveur fédéré et la base de données fédérée suivant votre système d'exploitation :
 - Préparez le serveur fédéré pour accéder aux sources de données avec ODBC (Windows).
 - Préparez le serveur fédéré pour accéder aux sources de données avec ODBC (Linux, UNIX).
2. Enregistrez l'encapsuleur ODBC.
3. Enregistrez les définitions de serveur pour une source de données ODBC.
4. Créez un mappage utilisateur pour une source de données ODBC.
5. Testez la connexion au serveur de source de données ODBC.
6. Enregistrez des pseudonymes pour les tables et vues de la source des données ODBC.

Préparation du serveur fédéré pour accéder aux sources de données avec ODBC (Windows)

Sur les serveurs fédérés qui fonctionnent sous Windows, le serveur fédéré doit pouvoir accéder aux sources de données ODBC. Pour préparer le serveur fédéré, vous devez vérifier les paramètres du DSN système ODBC.

Procédure

Pour préparer le serveur fédéré pour accéder à des sources de données avec ODBC :

Vérifiez que le pilote ODBC 3.x a été installé et configuré sur le serveur fédéré. Le nom du noeud de la source de données ODBC doit être défini dans le DSN

système. Voir la documentation du pilote ODBC pour les procédures d'installation et de configuration.

Si vous avez utilisé la fenêtre Administrateur de sources de données ODBC de Microsoft pour configurer le DSN, vous pouvez vérifier ce paramètre dans le Panneau de configuration. Vérifiez que la source de données ODBC est enregistrée comme un DSN système. Sinon, le serveur fédéré risque de ne pas pouvoir trouver le DSN.

Une fois cette tâche terminée, vous pouvez enregistrer l'encapsuleur.

Préparation du serveur fédéré pour accéder aux sources de données avec ODBC (Linux, UNIX)

Sur les serveurs fédérés qui fonctionnent sous Linux ou UNIX, le serveur fédéré doit pouvoir accéder aux sources de données ODBC. Pour préparer le serveur fédéré, vous devez vérifier les paramètres dans le fichier `odbc.ini`, créer des liens symboliques et tester la connexion aux sources de données ODBC.

Avant de commencer

Pour le système d'exploitation Linux pour System z, vous devez définir le paramètre **EnableDescribeParam** sur 1 dans le fichier de configuration `odbc.ini` du DSN lorsque vous utilisez l'encapsuleur ODBC pour vous connecter aux sources de données Sybase ou Oracle avec les pilotes DataDirect Sybase Wire Protocol, Oracle Wire Protocol ou Oracle Client :

```
EnableDescribeParam=1
```

Si vous ne définissez pas le paramètre **EnableDescribeParam** sur 1, le message d'erreur 1822N est généré pour lier les variables hôte non-char et varchar lorsque l'objet ODBC SERVER est défini sur le mode PUSHDOWN.

Pour de plus amples informations sur la connexion à une source de données avec un pilote DataDirect, voir <http://media.datadirect.com/download/docs/odbc/allodbc/userguide/ase6.html>.

Procédure

Pour préparer le serveur fédéré pour accéder à des sources de données avec ODBC :

1. Vérifiez que le fichier `odbc.ini` a été mis à jour sur le serveur fédéré. Si le fichier n'existe pas, vous pouvez le créer dans un éditeur de texte. Consultez la documentation du fournisseur du client ODBC pour toute information sur le fichier `odbc.ini`.
2. Configurez le client ODBC.
Consultez la documentation du fournisseur du client ODBC pour des instructions sur la manière de configurer le client ODBC.
3. Si le client est DataDirect ODBC ou RedBrick, vérifiez que les liens symboliques appropriés ont été créés. Dans les liens symboliques suivants, `ODBC_CLIENT_DIR` est le répertoire dans lequel le client ODBC est installé.

Systeme d'exploitation du serveur fédéré	Etape
Linux	<p>Créez les liens symboliques suivants :</p> <pre>ln -s \$ODBC_CLIENT_DIR/./locale /usr/local/locale</pre> <pre>ln -s \$ODBC_CLIENT_DIR/libodbcinst.so /usr/lib/libodbcinst.so</pre> <p>Si vous utilisez le pilote DataDirect Technologies Connect for ODBC 4.2, vous devez aussi créer le lien symbolique suivant :</p> <pre>ln -s \$ODBC_CLIENT_DIR/libivicu19.so /usr/lib/libivicu19.so</pre>
Solaris	<p>Créez le lien symbolique suivant :</p> <pre>ln -s \$ODBC_CLIENT_DIR/./locale \$HOME/sql/lib/locale</pre> <p>\$HOME est le répertoire de base du propriétaire d'instance DB2.</p>

4. Si le client est DataDirect ODBC, vous pouvez tester la connexion entre le serveur fédéré et la source de données à l'aide de l'outil demoodbc de DataDirect Connect ODBC.
 - a. Exécutez le script /opt/odbc/odbc.sh. Ce script configure plusieurs variables d'environnement spécifiques au système d'exploitation.
 - b. Testez la connexion à la source de données ODBC à l'aide de l'outil demoodbc de DataDirect Connect ODBC. L'outil demoodbc se trouve dans le sous-répertoire /demo des bibliothèques Connect ODBC.

Une fois cette tâche terminée, vous pouvez enregistrer l'encapsuleur.

Enregistrement de l'encapsuleur ODBC

Vous devez enregistrer un encapsuleur pour accéder aux sources de données ODBC. Les serveurs fédérés utilisent des encapsuleurs pour dialoguer avec les sources de données et en extraire des données. Les encapsuleurs sont mis en oeuvre comme un ensemble de fichiers de bibliothèque.

Procédure

Pour enregistrer l'encapsuleur ODBC :

Utilisez l'une des méthodes suivantes :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .

Méthode	Description
<p>Émettez l'instruction CREATE WRAPPER et indiquez le nom par défaut pour l'encapsuleur ODBC.</p>	<p>Par exemple :</p> <pre>CREATE WRAPPER ODBC;</pre> <p>A faire : Lorsque vous enregistrez l'encapsuleur à l'aide du nom par défaut, ODBC, le serveur fédéré utilise automatiquement la bibliothèque de l'encapsuleur ODBC appropriée au système d'exploitation sous lequel votre serveur fédéré fonctionne.</p> <p>Vous devez indiquer l'option d'encapsuleur MODULE sur les serveurs fédérés qui fonctionnent sous Linux ou UNIX. L'option d'encapsuleur MODULE indique le chemin d'accès complet de la bibliothèque qui contient le gestionnaire du pilote ODBC.</p> <p>Pour AIX et Solaris, si vous configurez l'accès à des sources de données IBM InfoSphere Classic Federation Server for z/OS, vous devez indiquer les options DB2_FENCED et DB2_SOURCE_CLIENT_MODE comme illustré dans l'exemple suivant.</p> <pre>CREATE WRAPPER ODBC LIBRARY 'libdb2rcodbc.a' OPTIONS (DB2_FENCED 'Y', DB2_SOURCE_CLIENT_MODE '32BIT', MODULE '/opt/IBM/DB2IIClassic82/cli/lib/cacsqcli.so')</pre>
<p>Émettez l'instruction CREATE WRAPPER et indiquez un nom différent pour l'encapsuleur ODBC.</p>	<p>En cas de conflit entre le nom d'encapsuleur par défaut et un nom d'encapsuleur existant dans la base de données fédérée, vous pouvez remplacer le nom d'encapsuleur par défaut par un nom que vous choisissez. Lorsque vous utilisez un nom différent du nom par défaut, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER.</p> <p>Par exemple, pour enregistrer un encapsuleur avec le nom encapsuleur_odbc sur un serveur fédéré qui utilise AIX, émettez l'instruction suivante :</p> <pre>CREATE WRAPPER encapsuleur_odbc LIBRARY 'libdb2rcodbc.a';</pre> <p>Le fichier de bibliothèque de l'encapsuleur que vous indiquez dépend du système d'exploitation du serveur fédéré.</p>

Une fois cette tâche terminée, vous pouvez enregistrer les définitions de serveur.

Fichiers de bibliothèque de l'encapsuleur ODBC

Les fichiers de bibliothèque de l'encapsuleur ODBC sont ajoutés au serveur fédéré lorsque vous installez l'encapsuleur.

Lorsque vous installez l'encapsuleur ODBC, trois fichiers de bibliothèque sont ajoutés dans le chemin de répertoire par défaut. Par exemple, si le serveur fédéré fonctionne sous AIX, les fichiers de bibliothèque de l'encapsuleur ajoutés dans le chemin de répertoire sont libdb2rcodbc.a, libdb2rcodbc.F.a et libdb2rcodbcU.a. Le fichier de bibliothèque de l'encapsuleur par défaut est libdb2rcodbc.a. Les autres fichiers de bibliothèque de l'encapsuleur sont utilisés sur le plan interne par l'encapsuleur ODBC.

Si vous n'utilisez pas le nom d'encapsuleur par défaut lorsque vous enregistrez un encapsuleur, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER et indiquer le nom de fichier de bibliothèque de l'encapsuleur par défaut.

Les chemins des répertoires par défaut et les noms de fichier de bibliothèque de l'encapsuleur par défaut sont indiqués dans la table suivante.

Tableau 24. Emplacements et noms de fichier de la bibliothèque du client ODBC

Système d'exploitation	Chemin de répertoire	Fichier de bibliothèque d'encapsuleur
AIX	/usr/opt/chemin_installation/lib32/ /usr/opt/chemin_installation/lib64/	libdb2codbc.a
Linux	/opt/IBM/db2/chemin_installation/lib32 /opt/IBM/db2/chemin_installation/lib64	libdb2codbc.so
Solaris	/opt/IBM/db2/chemin_installation/lib32 /opt/IBM/db2/chemin_installation/lib64	libdb2codbc.so
Windows	%DB2PATH%\bin	db2codbc.dll

chemin_installation est le chemin du répertoire dans lequel la fédération est installée sous UNIX ou Linux.

Instruction CREATE WRAPPER - exemples pour l'encapsuleur ODBC

Utiliser l'instruction CREATE WRAPPER pour enregistrer l'encapsuleur ODBC. Cette rubrique fournit des exemples pour Linux, UNIX et Windows.

Dans les exemples suivants, *encapsuleur_odbc* est le nom que vous attribuez à l'encapsuleur que vous enregistrez dans la base de données fédérée.

Serveurs fédérés Linux et Solaris

L'exemple suivant vous montre comment enregistrer un encapsuleur avec le nom par défaut sur un serveur fédéré qui fonctionne sous Linux ou Solaris :

```
CREATE WRAPPER odbc OPTIONS (MODULE '/opt/lib/odbc.so');
```

Vous devez indiquer l'option d'encapsuleur MODULE sur les serveurs fédérés qui fonctionnent sous Linux ou UNIX. L'option d'encapsuleur MODULE indique le chemin d'accès complet de la bibliothèque qui contient le gestionnaire du pilote ODBC.

L'exemple suivant vous montre comment enregistrer un encapsuleur avec un nom différent sur un serveur fédéré fonctionnant sous Linux ou Solaris :

```
CREATE WRAPPER encapsuleur_odbc LIBRARY 'libdb2codbc.so'  
OPTIONS (MODULE '/opt/lib/odbc.so');
```

Serveur fédéré AIX

L'exemple suivant vous montre comment enregistrer un encapsuleur avec le nom par défaut sur un serveur fédéré qui fonctionne sous AIX :

```
CREATE WRAPPER odbc  
OPTIONS (MODULE '/usr/lib/odbc.a');
```

Vous devez indiquer l'option d'encapsuleur MODULE sur les serveurs fédérés qui fonctionnent sous UNIX. L'option d'encapsuleur MODULE indique le chemin d'accès complet de la bibliothèque qui contient le gestionnaire du pilote ODBC.

L'exemple suivant vous montre comment enregistrer un encapsuleur avec un nom différent sur un serveur fédéré qui fonctionne sous AIX :

```
CREATE WRAPPER encapsuleur_odbc LIBRARY 'libdb2rcodbc.a'  
  OPTIONS (MODULE '/usr/lib/odbc.a');
```

IBM InfoSphere Classic Federation Server for z/OS (AIX)

L'exemple suivant vous montre comment enregistrer un encapsuleur ODBC en émettant l'instruction CREATE WRAPPER sous un système d'exploitation AIX. Sous AIX et Solaris, les options DB2_FENCED et DB2_SOURCE_CLIENT_MODE doivent être indiquées comme illustré dans l'exemple suivant.

```
CREATE WRAPPER odbc  
  OPTIONS (DB2_FENCED 'Y', DB2_SOURCE_CLIENT_MODE '32BIT',  
  MODULE '/opt/IBM/DB2IIClassic82/cli/lib/cacsq1cli.so');
```

Vous devez indiquer l'option d'encapsuleur MODULE sur les serveurs fédérés qui fonctionnent sous UNIX. L'option d'encapsuleur MODULE indique le chemin d'accès complet de la bibliothèque qui contient le gestionnaire du pilote ODBC.

L'exemple suivant vous montre comment enregistrer un encapsuleur pour accéder aux sources de données IBM InfoSphere Classic Federation Server for z/OS avec un nom différent :

```
CREATE WRAPPER encapsuleur_odbc LIBRARY 'libdb2rcodbc.a'  
  OPTIONS (DB2_FENCED 'Y', DB2_SOURCE_CLIENT_MODE '32BIT',  
  MODULE '/opt/IBM/DB2IIClassic82/cli/lib/cacsq1cli.so');
```

Serveur fédéré Windows

L'exemple suivant vous montre comment enregistrer un encapsuleur avec le nom par défaut sur un serveur fédéré qui fonctionne sous Windows :

```
CREATE WRAPPER odbc;
```

L'exemple suivant vous montre comment enregistrer un encapsuleur avec un nom différent sur un serveur fédéré qui fonctionne sous Windows :

```
CREATE WRAPPER encapsuleur_odbc LIBRARY 'db2rcodbc.dll';
```

Enregistrement des définitions de serveur pour une source de données ODBC

Vous devez enregistrer chaque serveur ODBC auquel vous voulez accéder dans la base de données fédérée.

Procédure

Pour enregistrer une définition de serveur pour une source de données ODBC :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .

Méthode	Description
Emettez l'instruction CREATE SERVER.	<p>1. Emettez l'instruction CREATE SERVER et indiquez les options de serveur obligatoires :</p> <pre>CREATE SERVER <i>nom_définition_serveur</i> TYPE <i>type_source_données</i> VERSION <i>numéro_version</i> WRAPPER <i>nom_encapsuleur</i> OPTIONS (NODE '<i>nom_noeud</i>', CODEPAGE '<i>nom_page_codes</i>', DBNAME '<i>nom_base_de_données</i>');</pre> <p>Bien que NODE, CODEPAGE et DBNAME soient des options de serveur dans l'instruction CREATE SERVER, elles doivent être indiquées obligatoirement pour les sources de données ODBC :</p> <ul style="list-style-type: none"> • L'option de serveur NODE doit être définie sur la source de données indiquée avec le mot clé DATASOURCE dans le fichier cac.ini. <p>Exemple Si le fichier cac.ini indique DATASOURCE = CACSAMP tcp/150.45.37.49/5000, alors l'option NODE doit être définie sur CACSAMP.</p> <ul style="list-style-type: none"> • L'option de serveur CODEPAGE doit être définie sur le numéro de la page de codes client indiquée dans le fichier cac.ini. <p>Exemple Si le fichier cac.ini indique CLIENT CODEPAGE = IBM-850, alors l'option CODEPAGE doit être définie sur 850.</p> <ul style="list-style-type: none"> • L'option de serveur DBNAME doit être définie sur le nom de la base de données de la source de données à laquelle vous voulez accéder. <p>Exemple Si le nom de la source de données ODBC est venice, l'option DBNAME doit être définie sur <i>venice</i>.</p> <p>2. Une fois la définition de serveur créée, utilisez l'instruction ALTER SERVER pour ajouter ou supprimer des options de serveur.</p>

Une fois cette tâche terminée, vous pouvez créer un mappage utilisateur.

Instruction CREATE SERVER - exemples avec l'encapsuleur ODBC

Utiliser l'instruction CREATE SERVER pour enregistrer des définitions de serveur pour l'encapsuleur ODBC. Cette rubrique fournit un exemple complet avec les paramètres obligatoires et un exemple avec des options de serveur supplémentaires.

L'exemple suivant vous montre comment enregistrer une définition de serveur pour une source de données MySQL en émettant l'instruction CREATE SERVER :

```
CREATE SERVER monserveur_sql TYPE mysql
VERSION 4.0 WRAPPER nom_encapsuleur
OPTIONS (NODE 'noeud_odbc', DBNAME 'venice)
```

monserveur_sql

Nom que vous attribuez au serveur de source de données ODBC. Les doublons de noms de définition de serveur ne sont pas autorisés.

TYPE *mysql*

Indique le type de serveur de source de données auquel vous configurez l'accès. Ce paramètre est facultatif.

VERSION *4.0*

La version de la source de données ODBC à laquelle vous voulez accéder. Ce paramètre est facultatif.

WRAPPER *nom_encapsuleur*

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

NODE '*noeud_odbc*'

Le nom du noeud (nom DSN système) qui a été affecté à la source de données ODBC lorsque le DSN a été défini. Sur les serveurs fédérés qui fonctionnent sous Windows, cette valeur doit être le nom d'un DSN système dans la fenêtre Administrateur de sources de données ODBC. Sur les serveurs fédérés qui fonctionnent sous UNIX, le nom du noeud est le DSN défini dans le fichier de configuration ODBC. Le fichier de configuration ODBC est habituellement appelé `odbc.ini`.

Bien que le nom du noeud soit indiqué comme option dans l'instruction CREATE SERVER, il doit être indiqué obligatoirement pour les sources de données ODBC.

DBNAME '*venice*'

Facultatif. Le nom de la source de données ODBC à laquelle vous voulez accéder.

Options de serveur

Lorsque vous créez la définition de serveur, vous pouvez indiquer des options de serveur supplémentaires dans l'instruction CREATE SERVER. Les options de serveur peuvent être des options de serveur générales et des options de serveur spécifiques à ODBC.

Certaines sources de données ODBC (par exemple MySQL) ne peuvent pas traiter les apostrophes autour des noms de table et des noms de colonne dans les instructions SQL. Pour accéder à ces sources de données, vous devez inclure les options de serveur suivantes dans l'instruction CREATE SERVER :

- `DB2_TABLE_QUOTE_CHAR '``'`
- `DB2_ID_QUOTE_CHAR '``'`
- `DB2_AUTHID_QUOTE_CHAR '``'`

Le caractère ``` est le délimiteur utilisé pour les identificateurs tels que les noms de schéma, les noms de table et les noms de colonne.

Par exemple :

```
CREATE SERVER monserveur_sql TYPE mysql
  VERSION 4.0 WRAPPER nom_encapsuleur
  OPTIONS (NODE 'monnoeud_sql', DB2_TABLE_QUOTE_CHAR '``',
  DB2_ID_QUOTE_CHAR '``', DB2_AUTHID_QUOTE_CHAR '``')
```

Création d'un mappage utilisateur pour une source de données ODBC

Lorsque vous essayez d'accéder à un serveur ODBC, le serveur fédéré établit une connexion au serveur ODBC à l'aide d'un ID utilisateur et d'un mot de passe qui sont valides pour cette source de données. Pour les sources de données exigeant un mappage utilisateur, vous devez définir une association (un mappage utilisateur) entre chaque ID utilisateur et mot de passe de serveur fédéré et l'ID utilisateur et le mot de passe de la source de données correspondante.

A propos de cette tâche

Créer un mappage utilisateur pour chaque ID utilisateur qui accédera au système fédéré pour envoyer des demandes réparties à la source de données ODBC.

Procédure

Pour mapper un ID utilisateur local à l'ID utilisateur et au mot de passe de la source de données ODBC :

Emettez une instruction CREATE USER MAPPING.

Par exemple :

```
CREATE USER MAPPING FOR IDutilisateur_local SERVER nom_définition_serveur
    OPTIONS (REMOTE_AUTHID 'IDutilisateur_distant', REMOTE_PASSWORD
'motdepasse_distant');
```

Bien que REMOTE_AUTHID et REMOTE_PASSWORD soient des options du mappage utilisateur dans l'instruction CREATE USER MAPPING, ces options doivent être indiquées obligatoirement pour accéder aux sources de données ODBC :

Le mappage utilisateur doit mapper l'ID authentification DB2 à l'ID utilisateur et au mot de passe indiqués dans le fichier cac.ini.

Exemple

Si le fichier cac.ini indique USERID = *MON_ID_UTILISATEUR* et USERPASSWORD = *MON_MOT_DE_PASSE*, alors les options de l'instruction CREATE USER MAPPING doivent être indiquées comme suit :

```
REMOTE_AUTHID = MON_ID_UTILISATEUR
REMOTE_PASSWORD = MON_MOT_DE_PASSE
```

Une fois cette tâche terminée, vous pouvez tester la connexion à la source de données ODBC.

Instruction CREATE USER MAPPING - exemples pour l'encapsuleur ODBC

Utiliser l'instruction CREATE USER MAPPING pour mapper un ID utilisateur du serveur fédéré à un ID utilisateur et à un mot de passe de la source de données ODBC. Cette rubrique fournit un exemple complet avec les paramètres obligatoires et un exemple qui vous montre comment utiliser le registre spécial USER DB2 avec l'instruction CREATE USER MAPPING.

L'exemple suivant montre comment mapper un identificateur utilisateur fédéré à un ID utilisateur et à un mot de passe de la source de données ODBC :

```
CREATE USER MAPPING FOR arturo SERVER monserveur_sql
    OPTIONS (REMOTE_AUTHID 'art', REMOTE_PASSWORD 'red4blue')
```

arturo Indique l'identificateur utilisateur local que vous mappez à l'ID utilisateur et au mot de passe distants, qui sont définis sur la source de données ODBC.

monserveur_sql

Indique le nom de définition de serveur que vous avez défini dans l'instruction CREATE SERVER pour la source de données ODBC.

'art' Indique l'ID utilisateur distant auquel vous mappez *arturo*. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_ID sur 'U' ou 'L' dans l'instruction CREATE SERVER.

'red4blue'

Indique le mot de passe distant associé à *'art'*. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_PW sur 'U' ou 'L' dans l'instruction CREATE SERVER.

registre spécial USER DB2

Vous pouvez utiliser le registre spécial USER DB2 pour mapper l'identificateur utilisateur de la personne qui émet l'instruction CREATE USER MAPPING à l'identificateur utilisateur de la source de données qui est indiqué dans l'option utilisateur REMOTE_AUTHID.

L'exemple suivant montre une instruction CREATE USER MAPPING qui comporte le registre spécial USER :

```
CREATE USER MAPPING FOR USER SERVER monserveur_sql
  OPTIONS (REMOTE_AUTHID 'art', REMOTE_PASSWORD 'red4blue');
```

Test de la connexion au serveur de source de données ODBC

Tester la connexion au serveur de source de données ODBC pour déterminer si le serveur fédéré est correctement configuré pour accéder aux sources de données ODBC.

A propos de cette tâche

Vous pouvez tester la connexion au serveur de source de données ODBC à l'aide de la définition de serveur et des mappages utilisateur que vous avez définis.

Procédure

Pour tester la connexion au serveur de source de données ODBC :

Ouvrez une session passe-système et émettez une instruction SELECT sur les tables système de la source de données ODBC. Si l'instruction SELECT renvoie un comptage, votre définition de serveur et votre mappage utilisateur sont configurés correctement.

```
SET PASSTHRU nom_définition_serveur
SELECT count(*) FROM nom_schéma.nom_table
SET PASSTHRU RESET
```

Si l'instruction SELECT renvoie une erreur, vous devez identifier et résoudre les erreurs de connexion

Une fois cette tâche terminée, vous pouvez enregistrer des pseudonymes pour les tables et vues de la source de données ODBC.

Identification et résolution des erreurs de connexion à la source de données

Une connexion de test au serveur de source de données peut renvoyer une erreur pour plusieurs raisons. Il y a des mesures que vous pouvez prendre pour déterminer pourquoi l'erreur s'est produite.

Symptôme

Une erreur est renvoyée lorsque vous essayez de vous connecter à la source de données.

Cause

Il y a plusieurs causes possibles à un incident de connexion.

Résolution de l'incident

Pour identifier et résoudre les erreurs de connexion à la source de données, recherchez les problèmes éventuels avec les éléments suivants :

- Vérifiez que la source de données est disponible.
- S'il y a lieu, vérifiez que le serveur de source de données est configuré pour les connexions entrantes.
- Vérifiez que les paramètres de votre mappage utilisateur pour les options REMOTE_AUTHID et REMOTE_PASSWORD sont valides pour les connexions à la source de données. Modifiez le mappage utilisateur ou créez un autre mappage utilisateur si nécessaire.
- S'il y a lieu, vérifiez que le logiciel client de la source de données sur le serveur fédéré est installé et configuré correctement pour se connecter à la source de données.
- Pour les sources de données ODBC, vérifiez que le pilote ODBC sur le serveur fédéré est installé et configuré correctement pour se connecter au serveur de source de données ODBC. Sur les serveurs fédérés qui fonctionnent sous Windows, utilisez l'outil Administrateur de sources de données ODBC pour vérifier le pilote. Sur les serveurs fédérés qui fonctionnent sous UNIX, consultez la documentation du fournisseur de client ODBC.
- Vérifiez que les valeurs des variables définies sur le serveur fédéré sont appropriées à la source de données. Ces variables sont les variables d'environnement système, les variables du fichier db2dj.ini et les variables du registre de profil DB2 (db2set).
- Vérifiez votre définition de serveur. Si nécessaire, supprimez la définition de serveur et recréez-la.

Enregistrement de pseudonymes pour les tables et vues de la source de données ODBC

Dans chaque définition de serveur ODBC que vous enregistrez, vous devez enregistrer un pseudonyme pour chaque table ou vue à laquelle vous voulez accéder. Utilisez ces pseudonymes, au lieu des noms des objets de source de données, lorsque vous interrogez les sources de données ODBC.

Avant de commencer

Mettez à jour les statistiques sur la source de données ODBC avant d'enregistrer un pseudonyme. La base de données fédérée exploite les statistiques du catalogue de la source de données pour optimiser le traitement des requêtes. Utilisez la commande de source de données équivalant à la commande RUNSTATS DB2 pour mettre à jour les statistiques de la source de données.

Procédure

Pour enregistrer un pseudonyme pour une table ou une vue de la source de données ODBC, utilisez une des méthodes suivantes.

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .
Émettez l'instruction CREATE NICKNAME. La longueur des pseudonymes peut aller jusqu'à 128 caractères.	Par exemple : <pre>CREATE NICKNAME <i>pseudonyme</i> FOR <i>nom_définition_serveur</i>."<i>schéma_distant</i>". "<i>table.distante</i>" ;</pre>

Lorsque vous créez le pseudonyme, le serveur fédéré interroge le catalogue de la source de données à l'aide du pseudonyme. Cette requête teste la connexion à la table ou à la vue de la source de données. Si la connexion ne fonctionne pas, vous recevez un message d'erreur.

Répétez cette étape pour chaque table ou vue ODBC pour laquelle vous voulez créer un pseudonyme.

Instruction CREATE NICKNAME - exemples pour l'encapsuleur ODBC

Utiliser l'instruction CREATE NICKNAME pour enregistrer un pseudonyme pour une table ou une vue ODBC à laquelle vous voulez accéder. Cette rubrique fournit un exemple complet avec les paramètres obligatoires.

L'exemple suivant montre comment enregistrer un pseudonyme pour une table ou une vue ODBC à l'aide de l'instruction CREATE NICKNAME.

```
CREATE NICKNAME cust_europe FOR monserveur_sql."vinnie"."italy"
```

cust_europe

Pseudonyme unique utilisé pour identifier la table ou la vue ODBC. Le pseudonyme doit être unique à l'intérieur du schéma.

Important : Le pseudonyme est un nom composé de deux parties, le schéma et le pseudonyme. Si vous omettez le schéma lorsque vous enregistrez le pseudonyme, le schéma du pseudonyme sera l'identificateur utilisateur de l'utilisateur qui enregistre le pseudonyme.

Si votre source de données ODBC ne prend pas en charge les schémas, omettez le schéma de l'instruction CREATE NICKNAME.

```
monserveur_sql."vinnie"."italy"
```

Identificateur en trois parties pour l'objet distant :

- *monserveur_sql* est le nom de définition de serveur que vous avez affecté au serveur de source de données ODBC dans l'instruction CREATE SERVER.
- *vinnie* est l'ID utilisateur du propriétaire de la table ou vue.
- *italy* est le nom de la table ou de la vue distante à laquelle vous voulez accéder.

Le serveur fédéré convertit les noms des schémas et tables ODBC en majuscules, à moins que vous ne placiez les noms entre apostrophes.

Optimisation des performances de l'encapsuleur ODBC avec l'utilitaire d'optimisation ODBC (db2fedsvr cfg)

Vous pouvez optimiser les performances de l'encapsuleur ODBC avec l'utilitaire d'optimisation ODBC, db2fedsvr cfg. Cet utilitaire applique un ensemble de requêtes prédéfinies à la source de données et teste la précision des résultats. L'utilitaire crée un ensemble d'instructions ALTER SERVER que vous pouvez appliquer au serveur pour définir les options de serveur afin d'obtenir des performances optimales.

Avant de commencer

Les éléments suivants doivent être configurés sur votre serveur fédéré :

- fédération
- L'encapsuleur ODBC
- Le logiciel client ODBC
- Les variables pour l'environnement système. Les variables ODBCINI et LIBPATH peuvent être obligatoires.

Pour des informations sur l'installation, la configuration et les exigences pour ODBC, voir la documentation fournie avec chacun de ces produits.

Procédure

Pour optimiser les performances de l'encapsuleur ODBC avec l'utilitaire d'optimisation ODBC :

1. Si votre serveur ODBC n'a pas été défini, utilisez le Centre de contrôle DB2 ou l'interpréteur de commandes DB2 pour vous connecter au serveur fédéré et créer un encapsuleur et un serveur ODBC.
2. Facultatif : si les tables de test existent déjà sur votre source de données ODBC, connectez-vous à votre source de données ODBC et supprimez-les.
3. Exécutez l'utilitaire d'optimisation ODBC depuis la ligne de commande DB2 avec les options que vous voulez utiliser. L'exécution de l'utilitaire peut prendre un certain temps.
4. Vérifiez que l'utilitaire a été exécuté avec succès. Si l'utilitaire a été exécuté avec succès, vous voyez le message suivant dans la fenêtre de commande ou dans le fichier de sortie que vous avez indiqué :

```
ALTER SERVER "DS1" OPTIONS (ADD option1, 'value1')
ALTER SERVER "DS1" OPTIONS (ADD option2, 'value2')
ALTER SERVER "DS1" OPTIONS (ADD option3, 'value3')
.....
```

La commande db2fedsvr cfg a été exécutée avec succès.

Si la commande n'a pas été exécutée avec succès, vous voyez un message d'erreur indiquant la cause de l'erreur. Corrigez le problème et réexécutez la commande.

Vous devez créer les tables de test manuellement dans les cas suivants :

- Si vous voulez utiliser des noms de table différents des noms par défaut
 - Si la source de données à laquelle vous accédez avec ODBC est en lecture seule
 - Si l'utilitaire d'optimisation ODBC ne parvient pas à créer les tables de test qui sont attendues par l'utilitaire
5. Connectez-vous au serveur fédéré sur lequel la source de données est définie.

6. Utilisez l'utilitaire db2look pour enregistrer vos paramètres de serveur existants avant d'exécuter le fichier créé par l'utilitaire d'optimisation ODBC. Voir la documentation de l'utilitaire db2look pour des informations sur l'enregistrement de vos paramètres de serveur existants.
7. Facultatif : si votre serveur ODBC est défini, vous pouvez vous connecter au serveur fédéré et supprimer les options de serveur. L'utilitaire crée des instructions ALTER SERVER au format décrit à l'étape 4. Si ces options de serveur ont déjà été ajoutées, les instructions ALTER SERVER échoueront.
8. Utilisez la commande suivante pour exécuter les instructions ALTER SERVER qui ont été générées par l'utilitaire pour la base de données fédérée. Les instructions ALTER SERVER qui ont été créées par l'utilitaire d'optimisation ODBC se trouvent dans le fichier db2fedsvr.cfg.sql.
db2 -tvf db2fedsvr.cfg.sql
9. Vérifiez les résultats des instructions ALTER SERVER. Si l'une des instructions a échoué, vous pouvez modifier les instructions dans le fichier db2fedsvr.cfg.sql et réexécuter les instructions jusqu'à ce qu'elles réussissent.
10. Une fois que vous avez terminé l'optimisation du serveur avec l'utilitaire d'optimisation ODBC, définissez l'option de serveur PUSHDOWN sur 'Y' pour terminer le processus d'optimisation.

Syntaxe de la commande db2fedsvr.cfg - utilitaire d'optimisation ODBC

Utiliser la commande db2fedsvr.cfg pour améliorer les performances de l'encapsuleur ODBC.

Syntaxe

```
db2fedsvr.cfg -s serverName [-m odbcDriverManagerLibrary] -dsn odbcDSNname
[-dbname dsDBname] [-u userid] [-p password] [-noprep] [-prefix tableNamePrefix] [-suffix
tableNameSuffix] [-dscp codePage] [-v] [-o outputFile] [-h]
```

Paramètres

Utilisez la commande db2fedsvr.cfg32 lorsque vous utilisez des pilotes ODBC 32 bits sous AIX ou Solaris. Sinon, utilisez la commande db2fedsvr.cfg.

-dbname dsDBname

Nom de la base de données de la source de données.

-dscp codePage

Identificateur de la page de codes pour la source de données. Si cette option n'est pas indiquée, l'utilitaire utilise la page de codes de l'environnement de l'utilisateur. Ce paramètre est facultatif.

-dsn odbcDSNname

Nom de source de données système (DSN) de la source de données.

-h Affiche l'aide détaillée. Ce paramètre est facultatif.

-m odbcDriverManagerLibrary

Nom qualifié complet du fichier de la bibliothèque du gestionnaire de pilote ODBC. Le nom du fichier de bibliothèque du gestionnaire de pilote ODBC est facultatif sous Windows.

-noprep

Empêche la création des tables de tests sur la source de données avant les tests. Ce paramètre est facultatif.

-o outputFile

Nom qualifié complet du fichier de sortie de l'utilitaire d'optimisation ODBC. Le fichier de sortie contient les instructions ALTER SERVER qui sont utilisées pour ajuster les performances de l'encapsuleur ODBC. Ce paramètre est facultatif. Si ce paramètre n'est pas indiqué, la sortie s'affiche dans la fenêtre de commande.

-p password

Mot de passe pour se connecter à la source de données. Ce paramètre est facultatif.

-prefix tableNamePrefix

Le préfixe des noms de table de la source de données ODBC que l'utilitaire utilise pour l'analyse. Si aucun préfixe n'est indiqué, c'est le préfixe par défaut, IITEST, qui est utilisé. Ce paramètre est facultatif.

-s serverName

Nom du serveur fédéré.

-suffix tableNameSuffix

Le suffixe des noms de table de la source de données ODBC que l'utilitaire utilise pour l'analyse. Si aucun suffixe n'est indiqué, c'est une chaîne vide qui est utilisée.

-u userid

Nom d'utilisateur pour se connecter à la source de données. Ce paramètre est facultatif.

-v Indique que la sortie de l'utilitaire est prolixe. Ce paramètre est facultatif.

Exemple

L'exemple suivant montre la commande qui est appliquée au magasin de données, la source de données ODBC. Dans cet exemple, les tables de test sont nommées ABC n XYZ, où n est un nombre de 1 à 7.

```
db2fedsvr cfg -s DS1 -m "/usr/lib/odbc.a"  
-dsn datastore -dbname db1 -u authid -p password -noprep  
-prefix ABC -suffix XYZ -o "/home/user1/db2fedsvr cfg.sql"
```

Testez les définitions de table pour l'utilitaire d'optimisation ODBC (db2fedsvr cfg)

Dans certains cas, vous devez créer manuellement les tables de test pour l'utilitaire d'optimisation ODBC. Les définitions de table de test sont décrites dans cette rubrique.

Vous devez créer les tables de test pour l'utilitaire d'optimisation ODBC dans les cas suivants :

- Si vous voulez utiliser des noms de table différents des noms par défaut
- Si la source de données à laquelle vous accédez avec ODBC est en lecture seule
- Si l'utilitaire d'optimisation ODBC ne parvient pas à créer les tables de test qui sont attendues par l'utilitaire

La définition de table suivante s'applique aux sept tables de test qui sont nécessaires (IITEST1 à IITEST7). Le préfixe du nom de table par défaut est IITEST et le suffixe par défaut est une chaîne vide. Si vous indiquez un préfixe et un suffixe différents, vous devez indiquer les options *-prefix* et *-suffix* lorsque vous exécutez l'utilitaire d'optimisation ODBC.

Tableau 25. Définition de table de test de l'utilitaire d'optimisation ODBC pour la table IITEST1

Nom de colonne	Type de données SQL	Identificateur de type de données SQL	Longueur
IT1C1	integer	SQL_INTEGER	
IT1C2	integer	SQL_INTEGER	
IT1C3	char(1)	SQL_CHAR	1
IT1C4	char(3)	SQL_CHAR	3
IT1C5	char(10)	SQL_CHAR	10
IT1C6	varchar(10)	SQL_VARCHAR	10
IT1C7	char(100)	SQL_CHAR	100

Tableau 26. Définition de table de test de l'utilitaire d'optimisation ODBC pour la table IITEST2

Nom de colonne	Type de données SQL	Identificateur de type de données SQL	Longueur
IT2C1	integer	SQL_INTEGER	
IT2C2	integer	SQL_INTEGER	
IT2C3	char(30)	SQL_CHAR	30

Tableau 27. Définition de table de test de l'utilitaire d'optimisation ODBC pour les tables IITEST3 à IITEST7

Nombre de colonnes dans chaque table	Nom de colonne	Type de données SQL	Identificateur de type de données SQL	Longueur
66	ITxCn	char(100)	SQL_CHAR	100

x Nombre correspondant à la table à définir.

n Numéro de colonne.

Par exemple, IT3C1 est le nom de colonne de la première colonne de la table IITEST3.

Accès aux données Excel à l'aide de l'encapsuleur ODBC

Vous pouvez accéder aux classeurs Microsoft Excel avec l'encapsuleur ODBC en utilisant le pilote ODBC Excel.

Avant de commencer

- Le pilote ODBC Excel doit être sur le serveur fédéré.
- Le serveur fédéré doit pouvoir ouvrir et lire les feuilles de calcul dans le classeur Excel pour extraire les données. Les classeurs Excel doivent donc être sur le même ordinateur que le serveur fédéré ou sur une unité réseau mappée accessible.
- Mettez en forme les colonnes d'après le type de données que la colonne est censée recevoir.
- Les données insérées dans les colonnes doivent correspondre au type de format défini pour la colonne.
- Si les huit premières lignes du tableur n'ont pas de données, veillez à ce qu'elles soient bien vides. Pour garantir qu'une cellule est vide, ouvrez la feuille de calcul dans Microsoft Excel et sélectionnez **Edition** → **Effacer Tout**.
- Veillez à ce que les données insérées dans les colonnes de la feuille de calcul correspondent au type défini.

Restrictions

- L'encapsuleur ODBC ne peut pas accéder à une feuille de calcul lorsque le classeur est déjà ouvert par un utilisateur ou par une application en mode lecture-écriture. Cependant, si l'encapsuleur ODBC ouvre le classeur avant qu'un utilisateur ou une application ne l'ouvre, l'utilisateur ou l'application peut ouvrir le classeur en mode lecture seule.
- Le pilote ODBC Excel suppose que la première ligne non vide contient les étiquettes des colonnes de la feuille de calcul. Vous devez insérer une ligne d'étiquettes de colonne dans la feuille de calcul si la feuille de calcul n'a pas d'étiquettes.
- Le pilote ODBC Excel étant uniquement disponible pour les systèmes d'exploitation Windows, vous ne pouvez utiliser l'encapsuleur ODBC pour accéder à des données Excel que sur des serveurs fédérés qui fonctionnent sous Windows.
- Vous pouvez appliquer des opérations d'insertion et de mise à jour à des feuilles de calcul Excel, mais vous ne pouvez pas exécuter des opérations de suppression. Le pilote ODBC Excel ne prend pas en charge les opérations de suppression. Pour supprimer des données de la feuille de calcul, vous devez ouvrir la feuille de calcul dans Excel pour effectuer les modifications.

A propos de cette tâche

Il n'est pas nécessaire que l'application Excel soit installée sur le serveur fédéré. Le pilote ODBC Excel est automatiquement installé avec Windows.

Avec l'encapsuleur ODBC et le pilote ODBC Excel, vous pouvez accéder aux données de chacune des feuilles de calcul à l'intérieur d'un classeur. Le pilote ODBC Excel interprète un classeur comme une base de données et interprète chaque feuille de calcul à l'intérieur du classeur comme une table.

Le pilote ODBC Excel prend en charge les versions antérieures des classeurs Excel, même si l'assistance n'est plus assurée pour la version de l'application Excel qui a produit les classeurs. Par exemple, Microsoft n'assure plus aucune assistance technique pour les feuilles de calcul créées sous Excel version 4.0, mais le pilote prend en charge les feuilles de calcul Excel qui ont été créées sous cette version.

Procédure

Pour accéder à des feuilles de calcul Excel avec l'encapsuleur ODBC :

1. Vérifiez que le classeur Excel auquel vous voulez accéder est sur le serveur fédéré ou sur une unité réseau mappée accessible.
2. Si vos données Excel sont partagées à travers un réseau Windows transitoire qui utilise des groupes de travail, vous devez configurer vos droits d'accès sur la source de données Excel.
3. Si nécessaire, modifiez la mise en page des données dans les feuilles de calcul Excel pour l'ajuster aux exigences du pilote ODBC Excel. Répétez cette étape pour chaque feuille de calcul ou plage nommée à laquelle vous voulez accéder.
4. Si nécessaire, créez les plages nommées auxquelles vous voulez accéder.
5. Créez un DSN système pour le classeur auquel vous voulez accéder. Vous pouvez utiliser l'Administrateur de sources de données ODBC pour configurer le DSN système. Le nom qui a été indiqué lorsque vous avez créé le DSN système est affecté comme valeur de l'option NODE dans l'instruction CREATE SERVER.

Si votre source de données Excel est partagée à travers un réseau Windows qui utilise des groupes de travail, vous devez indiquer le nom de la base de données du DSN système avec la syntaxe suivante :

```
\\nom_ordinateur\sousrépertoire_nomfichier
```

où *nom_ordinateur* est le nom de l'ordinateur de la source de données Excel et *sousrépertoire_nomfichier* est le sous-répertoire et le nom du fichier Excel.

Exemple

Si le nom de l'ordinateur de la source de données Excel est XLSQLS et que le répertoire réseau du fichier Excel est E : \share\test.xls, vous indiquerez le nom de base de données DSN suivant :

```
\\XLSQLS\share\test.xls
```

où le répertoire principal du répertoire réseau E: est remplacé par \\ et par le nom de l'ordinateur XLSQLS.

6. Emettez l'instruction CREATE WRAPPER.
7. Indiquez l'emplacement du classeur en enregistrant un objet de serveur dans le catalogue système de la base de données fédérée. Pour l'encapsuleur ODBC, vous avez besoin d'un objet de serveur pour chaque DSN. Le DSN est associé au classeur lorsque le pilote ODBC Excel est utilisé. Le NODE *compounds_workbook_dsn* est le DSN système que vous avez créé. L'option NODE est obligatoire pour que l'encapsuleur ODBC puisse accéder aux feuilles de calcul Excel.

Pour indiquer l'emplacement du classeur, émettez l'instruction CREATE SERVER et utilisez le DSN comme DSN système pour l'option NODE.

Par exemple :

```
CREATE SERVER compounds_workbook WRAPPER odbc
      OPTIONS (NODE 'compounds_workbook_dsn', PASSWORD 'n')
```

Répétez cette étape pour chaque classeur auquel vous prévoyez d'accéder.

8. Emettez l'instruction CREATE NICKNAME pour créer un pseudonyme pour la feuille de calcul à laquelle vous voulez accéder. La syntaxe est :

```
CREATE NICKNAME pseudonyme FOR server_name.table_distante
```
9. Si vous avez créé une plage nommée pour accéder aux données, indiquez le nom de la plage comme élément *table_distante* de l'instruction CREATE NICKNAME.

Par exemple, si le nom de la plage est *testing*, l'instruction CREATE NICKNAME sera :

```
CREATE NICKNAME pseudonyme_composé FOR compounds_workbook.testing
```

Pour accéder aux données de toute la feuille de calcul et pas seulement d'une plage, vous indiquez le nom de la feuille de calcul suivi du symbole \$.

Par exemple, si le nom de la feuille de calcul est *Feuille1*, l'instruction CREATE NICKNAME sera :

```
CREATE NICKNAME compounds_nick FOR compounds_workbook.Feuille1$
```

Configuration des droits d'accès aux données Excel dans un groupe de travail avec utilisation de l'encapsuleur ODBC

Vous pouvez configurer vos droits d'accès pour accéder aux données Excel distantes qui sont partagées à travers un réseau Windows utilisant des groupes de travail.

A propos de cette tâche

Tous les exemples de cette tâche sont conçus pour Windows XP Professionnel. Reportez-vous à la documentation du système d'exploitation de votre source de données pour des informations spécifiques sur la manière de définir vos droits de compte utilisateur.

Procédure

Pour configurer les droits d'accès à votre source de données Excel :

1. Définissez le modèle de partage et de sécurité dans le panneau **Accès réseau : Modèle de partage et de sécurité pour les comptes locaux**. Vous pouvez sélectionner un des modèles de partage et de sécurité suivants :

Invité seul : les utilisateurs locaux s'identifient en tant qu'Invité.

Vous ne vous connectez à la source de données qu'à travers le compte utilisateur Invité. Vous recevez le niveau d'accès que vous configurez pour le compte utilisateur Invité et vous devez utiliser votre mot de passe d'utilisateur DB2.

Classique : les utilisateurs locaux s'identifient avec leur propre compte.

Vous pouvez utiliser vos ID utilisateur et mot de passe DB2 pour vous connecter à la source de données ou bien vous pouvez utiliser le compte utilisateur Invité. Pour l'ID utilisateur DB2, vous recevez le niveau d'accès à la source de données correspondant au niveau d'accès de votre ID utilisateur DB2. Pour le compte utilisateur Invité, vous recevez le niveau d'accès que vous configurez pour le compte utilisateur Invité et vous devez utiliser votre mot de passe d'utilisateur DB2.

Exemple

Pour sélectionner le modèle de partage et de sécurité **Classique : les utilisateurs locaux s'identifient avec leur propre compte**, vous ouvrez les **Outils d'administration** et naviguez jusqu'à **Stratégie de sécurité locale** → **Stratégies locales** → **Options de sécurité** → **Accès réseau : Modèle de partage et de sécurité pour les comptes locaux**

2. Créez le compte utilisateur dans le panneau **Accéder à cet ordinateur à partir des propriétés de réseau** :

Option	Description
Pour le modèle de sécurité Invité seul : les utilisateurs locaux s'identifient en tant qu'Invité	Indiquez le groupe de travail Windows et créez le compte utilisateur Invité. Vous devez indiquer le mot de passe à l'aide de votre mot de passe DB2.
Pour le modèle de sécurité Classique : les utilisateurs locaux s'identifient avec leur propre compte	<ul style="list-style-type: none">• Indiquez le groupe de travail Windows et créez le compte utilisateur à l'aide de vos ID utilisateur et mot de passe DB2.• Si vous voulez utiliser le compte utilisateur Invité, indiquez le groupe de travail Windows et créez le compte utilisateur Invité. Vous devez indiquer le mot de passe pour le compte utilisateur Invité à l'aide de votre mot de passe DB2.

Exemple

Pour créer le compte utilisateur Invité :

- a. Ouvrez les **Outils d'administration** et naviguez jusqu'à **Stratégie de sécurité locale** → **Stratégies locales** → **Attribution des droits d'utilisateurs** → **Accéder à l'ordinateur via le réseau**
 - b. Indiquez le groupe de travail Windows dans la zone **A partir de cet emplacement**.
 - c. Indiquez l'ID utilisateur DB2 dans la zone **Entrez les noms d'objet à sélectionner**.
 - d. Cliquez ensuite sur **Propriétés**.
 - e. Dans le panneau **Entrer le mot de passe**, indiquez l'ID utilisateur et le mot de passe DB2.
3. Définissez les droits de partage dans le panneau **Propriétés de partage** en définissant les droits **Modifier** et **Lire** sur **Autoriser**.

Exemple

Pour définir les droits de partage du répertoire réseau
E:\share\test.xls :

- Cliquez avec le bouton droit de la souris sur le dossier E:\share, puis cliquez sur **Partage et sécurité**.
 - Cliquez avec le bouton droit de la souris sur le dossier **Partager ce dossier** et sélectionnez **Autorisations**.
 - Choisissez l'utilisateur, puis sélectionnez **Autoriser** à la fois pour les droits **Modifier** et **Lire**.
4. Définissez les droits NTFS dans le panneau **Propriétés** en définissant les droits **Lire et exécuter** et **Lire** sur **Autoriser**.

Exemple

Pour définir les droits NTFS du fichier E:\share\test.xls :

- Cliquez avec le bouton droit de la souris sur le fichier test.xls, puis cliquez sur **Propriétés**.
- Dans l'onglet **Sécurité**, sélectionnez l'utilisateur, puis sélectionnez **Autoriser** à la fois pour les droits **Lire et exécuter** et **Lire**.

Terminez la procédure pour accéder aux données Excel à l'aide de l'encapsuleur ODBC.

Configuration de l'accès ODBC aux sources de données IBM InfoSphere Classic Federation Server for z/OS

L'encapsuleur ODBC a été optimisé pour accéder aux sources de données IBM InfoSphere Classic Federation Server for z/OS. L'encapsuleur ODBC détecte le pilote ODBC et configure automatiquement les options de performances.

Avant de commencer

Les éléments suivants doivent être configurés sur votre serveur fédéré :

- Serveur fédéré
- IBM InfoSphere Classic Federation Server for z/OS client
- Base de données fédérée
- Variables dans le fichier db2dj.ini d'environnement système du registre de profil DB2 (db2set) :
 - Pour les systèmes Linux et UNIX, la variable CAC_CONFIG doit indiquer le répertoire du fichier cac.ini, par exemple :

CAC_CONFIG=/home/db2inst1/cac.ini

- La variable d’environnement DB2LIBPATH peut être obligatoire suivant les répertoires d’installation de la bibliothèque du gestionnaire de pilote ODBC et de la bibliothèque du pilote ODBC.
- Pour les systèmes AIX et Solaris, l’option de l’encapsuleur DB2_FENCED doit être définie sur Y et l’option de l’encapsuleur DB2_SOURCE_CLIENT_MODE doit être définie sur 32BIT.

Pour des informations sur l’installation, la configuration et les exigences pour ODBC, voir la documentation fournie avec chacun de ces produits.

Restrictions

L’encapsuleur ODBC ne prend pas en charge l’instruction suivante avec les sources de données IBM InfoSphere Classic Federation Server for z/OS :

- CREATE TABLE

Les types de données suivants ne sont pas pris en charge avec les sources de données IBM InfoSphere Classic Federation Server for z/OS :

- BLOB
- CLOB
- DBCLOB
- CHAR FOR BIT DATA
- VARCHAR FOR BIT DATA

Procédure

Pour configurer l’accès ODBC aux sources de données IBM InfoSphere Classic Federation Server for z/OS :

1. Exécutez une des tâches suivantes suivant votre système d’exploitation :
 - Sous Windows, vérifiez que le nom de noeud de la source de données est défini comme un DSN système. Si vous avez utilisé l’Administrateur de sources de données ODBC Microsoft pour définir le DSN, vous pouvez utiliser le **Panneau de configuration** pour vérifier qu’il est enregistré comme un DSN système.
 - Sous UNIX, configurez le pilote ODBC.
2. Vous pouvez avoir besoin d’ajouter le répertoire des bibliothèques du client de la source de données à la variable d’environnement DB2LIBPATH pour que les bibliothèques du client se chargent correctement. Exécutez la commande db2set pour définir la variable d’environnement DB2LIBPATH et indiquer le répertoire d’installation de la bibliothèque du client de la source de données :

```
db2set DB2LIBPATH="répertoire_bibliothèque_client"
```

Où *répertoire_bibliothèque_client* est le répertoire de la bibliothèque du client de la source de données.

Exemple

```
db2set DB2LIBPATH="/opt/IBM/DB2IIClassic82/cli/lib"
```

3. Enregistrez l’encapsuleur ODBC.
4. Enregistrez les définitions de serveur pour une source de données ODBC.
5. Créez un mappage utilisateur pour une source de données ODBC.
6. Testez la connexion au serveur de source de données ODBC.

- Enregistrez des pseudonymes pour les tables et vues de la source des données ODBC.

Enregistrement de l'encapsuleur ODBC

Vous devez enregistrer un encapsuleur pour accéder aux sources de données ODBC. Les serveurs fédérés utilisent des encapsuleurs pour dialoguer avec les sources de données et en extraire des données. Les encapsuleurs sont mis en oeuvre comme un ensemble de fichiers de bibliothèque.

Procédure

Pour enregistrer l'encapsuleur ODBC :

Utilisez l'une des méthodes suivantes :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .
Emettez l'instruction CREATE WRAPPER et indiquez le nom par défaut pour l'encapsuleur ODBC.	<p>Par exemple :</p> <pre>CREATE WRAPPER ODBC;</pre> <p>A faire : Lorsque vous enregistrez l'encapsuleur à l'aide du nom par défaut, ODBC, le serveur fédéré utilise automatiquement la bibliothèque de l'encapsuleur ODBC appropriée au système d'exploitation sous lequel votre serveur fédéré fonctionne.</p> <p>Vous devez indiquer l'option d'encapsuleur MODULE sur les serveurs fédérés qui fonctionnent sous Linux ou UNIX. L'option d'encapsuleur MODULE indique le chemin d'accès complet de la bibliothèque qui contient le gestionnaire du pilote ODBC.</p> <p>Pour AIX et Solaris, si vous configurez l'accès à des sources de données IBM InfoSphere Classic Federation Server for z/OS, vous devez indiquer les options DB2_FENCED et DB2_SOURCE_CLIENT_MODE comme illustré dans l'exemple suivant.</p> <pre>CREATE WRAPPER ODBC LIBRARY 'libdb2rcodbc.a' OPTIONS (DB2_FENCED 'Y', DB2_SOURCE_CLIENT_MODE '32BIT', MODULE '/opt/IBM/DB2IIClassic82/cli/lib/cacsqcli.so')</pre>
Emettez l'instruction CREATE WRAPPER et indiquez un nom différent pour l'encapsuleur ODBC.	<p>En cas de conflit entre le nom d'encapsuleur par défaut et un nom d'encapsuleur existant dans la base de données fédérée, vous pouvez remplacer le nom d'encapsuleur par défaut par un nom que vous choisissez. Lorsque vous utilisez un nom différent du nom par défaut, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER.</p> <p>Par exemple, pour enregistrer un encapsuleur avec le nom <code>encapsuleur_odbc</code> sur un serveur fédéré qui utilise AIX, émettez l'instruction suivante :</p> <pre>CREATE WRAPPER encapsuleur_odbc LIBRARY 'libdb2rcodbc.a';</pre> <p>Le fichier de bibliothèque de l'encapsuleur que vous indiquez dépend du système d'exploitation du serveur fédéré.</p>

Une fois cette tâche terminée, vous pouvez enregistrer les définitions de serveur.

Fichiers de bibliothèque de l'encapsuleur ODBC

Les fichiers de bibliothèque de l'encapsuleur ODBC sont ajoutés au serveur fédéré lorsque vous installez l'encapsuleur.

Lorsque vous installez l'encapsuleur ODBC, trois fichiers de bibliothèque sont ajoutés dans le chemin de répertoire par défaut. Par exemple, si le serveur fédéré fonctionne sous AIX, les fichiers de bibliothèque de l'encapsuleur ajoutés dans le chemin de répertoire sont libdb2rcodbc.a, libdb2rcodbcF.a et libdb2rcodbcU.a. Le fichier de bibliothèque de l'encapsuleur par défaut est libdb2rcodbc.a. Les autres fichiers de bibliothèque de l'encapsuleur sont utilisés sur le plan interne par l'encapsuleur ODBC.

Si vous n'utilisez pas le nom d'encapsuleur par défaut lorsque vous enregistrez un encapsuleur, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER et indiquer le nom de fichier de bibliothèque de l'encapsuleur par défaut.

Les chemins des répertoires par défaut et les noms de fichier de bibliothèque de l'encapsuleur par défaut sont indiqués dans la table suivante.

Tableau 28. Emplacements et noms de fichier de la bibliothèque du client ODBC

Système d'exploitation	Chemin de répertoire	Fichier de bibliothèque d'encapsuleur
AIX	/usr/opt/ <i>chemin_installation</i> /lib32/ /usr/opt/ <i>chemin_installation</i> /lib64/	libdb2rcodbc.a
Linux	/opt/IBM/db2/ <i>chemin_installation</i> /lib32 /opt/IBM/db2/ <i>chemin_installation</i> /lib64	libdb2rcodbc.so
Solaris	/opt/IBM/db2/ <i>chemin_installation</i> /lib32 /opt/IBM/db2/ <i>chemin_installation</i> /lib64	libdb2rcodbc.so
Windows	%DB2PATH%\bin	db2rcodbc.dll

chemin_installation est le chemin du répertoire dans lequel la fédération est installée sous UNIX ou Linux.

Instruction CREATE WRAPPER - exemples pour l'encapsuleur ODBC

Utiliser l'instruction CREATE WRAPPER pour enregistrer l'encapsuleur ODBC. Cette rubrique fournit des exemples pour Linux, UNIX et Windows.

Dans les exemples suivants, *encapsuleur_odbc* est le nom que vous attribuez à l'encapsuleur que vous enregistrez dans la base de données fédérée.

Serveurs fédérés Linux et Solaris

L'exemple suivant vous montre comment enregistrer un encapsuleur avec le nom par défaut sur un serveur fédéré qui fonctionne sous Linux ou Solaris :

```
CREATE WRAPPER odbc OPTIONS (MODULE '/opt/lib/odbc.so');
```


Vous devez indiquer l'option d'encapsuleur MODULE sur les serveurs fédérés qui fonctionnent sous Linux ou UNIX. L'option d'encapsuleur MODULE indique le chemin d'accès complet de la bibliothèque qui contient le gestionnaire du pilote ODBC.

L'exemple suivant vous montre comment enregistrer un encapsuleur avec un nom différent sur un serveur fédéré fonctionnant sous Linux ou Solaris :

```
CREATE WRAPPER encapsuleur_odbc LIBRARY 'libdb2rcodbc.so'  
  OPTIONS (MODULE '/opt/lib/odbc.so');
```

Serveur fédéré AIX

L'exemple suivant vous montre comment enregistrer un encapsuleur avec le nom par défaut sur un serveur fédéré qui fonctionne sous AIX :

```
CREATE WRAPPER odbc  
  OPTIONS (MODULE '/usr/lib/odbc.a');
```

Vous devez indiquer l'option d'encapsuleur MODULE sur les serveurs fédérés qui fonctionnent sous UNIX. L'option d'encapsuleur MODULE indique le chemin d'accès complet de la bibliothèque qui contient le gestionnaire du pilote ODBC.

L'exemple suivant vous montre comment enregistrer un encapsuleur avec un nom différent sur un serveur fédéré qui fonctionne sous AIX :

```
CREATE WRAPPER encapsuleur_odbc LIBRARY 'libdb2rcodbc.a'  
  OPTIONS (MODULE '/usr/lib/odbc.a');
```

IBM InfoSphere Classic Federation Server for z/OS (AIX)

L'exemple suivant vous montre comment enregistrer un encapsuleur ODBC en émettant l'instruction CREATE WRAPPER sous un système d'exploitation AIX. Sous AIX et Solaris, les options DB2_FENCED et DB2_SOURCE_CLIENT_MODE doivent être indiquées comme illustré dans l'exemple suivant.

```
CREATE WRAPPER odbc  
  OPTIONS (DB2_FENCED 'Y', DB2_SOURCE_CLIENT_MODE '32BIT',  
  MODULE '/opt/IBM/DB2IIClassic82/cli/lib/cacsq1cli.so');
```

Vous devez indiquer l'option d'encapsuleur MODULE sur les serveurs fédérés qui fonctionnent sous UNIX. L'option d'encapsuleur MODULE indique le chemin d'accès complet de la bibliothèque qui contient le gestionnaire du pilote ODBC.

L'exemple suivant vous montre comment enregistrer un encapsuleur pour accéder aux sources de données IBM InfoSphere Classic Federation Server for z/OS avec un nom différent :

```
CREATE WRAPPER encapsuleur_odbc LIBRARY 'libdb2rcodbc.a'  
  OPTIONS (DB2_FENCED 'Y', DB2_SOURCE_CLIENT_MODE '32BIT',  
  MODULE '/opt/IBM/DB2IIClassic82/cli/lib/cacsq1cli.so');
```

Serveur fédéré Windows

L'exemple suivant vous montre comment enregistrer un encapsuleur avec le nom par défaut sur un serveur fédéré qui fonctionne sous Windows :

```
CREATE WRAPPER odbc;
```

L'exemple suivant vous montre comment enregistrer un encapsuleur avec un nom différent sur un serveur fédéré qui fonctionne sous Windows :

```
CREATE WRAPPER encapsuleur_odbc LIBRARY 'db2rcodbc.dll';
```


Enregistrement des définitions de serveur pour une source de données ODBC

Vous devez enregistrer chaque serveur ODBC auquel vous voulez accéder dans la base de données fédérée.

Procédure

Pour enregistrer une définition de serveur pour une source de données ODBC :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .
Emettez l'instruction CREATE SERVER.	<p>1. Emettez l'instruction CREATE SERVER et indiquez les options de serveur obligatoires :</p> <pre>CREATE SERVER nom_définition_serveur TYPE type_source_données VERSION numéro_version WRAPPER nom_encapsuleur OPTIONS (NODE 'nom_noeud', CODEPAGE 'nom_page_codes', DBNAME 'nom_base_de_données');</pre> <p>Bien que NODE, CODEPAGE et DBNAME soient des options de serveur dans l'instruction CREATE SERVER, elles doivent être indiquées obligatoirement pour les sources de données ODBC :</p> <ul style="list-style-type: none"> • L'option de serveur NODE doit être définie sur la source de données indiquée avec le mot clé DATASOURCE dans le fichier cac.ini. <p>Exemple</p> <p>Si le fichier cac.ini indique DATASOURCE = CACSAMP tcp/150.45.37.49/5000, alors l'option NODE doit être définie sur CACSAMP.</p> <ul style="list-style-type: none"> • L'option de serveur CODEPAGE doit être définie sur le numéro de la page de codes client indiquée dans le fichier cac.ini. <p>Exemple</p> <p>Si le fichier cac.ini indique CLIENT CODEPAGE = IBM-850, alors l'option CODEPAGE doit être définie sur 850.</p> <ul style="list-style-type: none"> • L'option de serveur DBNAME doit être définie sur le nom de la base de données de la source de données à laquelle vous voulez accéder. <p>Exemple</p> <p>Si le nom de la source de données ODBC est venice, l'option DBNAME doit être définie sur venice.</p> <p>2. Une fois la définition de serveur créée, utilisez l'instruction ALTER SERVER pour ajouter ou supprimer des options de serveur.</p>

Une fois cette tâche terminée, vous pouvez créer un mappage utilisateur.

Instruction CREATE SERVER - exemples avec l'encapsuleur ODBC

Utiliser l'instruction CREATE SERVER pour enregistrer des définitions de serveur pour l'encapsuleur ODBC. Cette rubrique fournit un exemple complet avec les paramètres obligatoires et un exemple avec des options de serveur supplémentaires.

L'exemple suivant vous montre comment enregistrer une définition de serveur pour une source de données MySQL en émettant l'instruction CREATE SERVER :

```
CREATE SERVER monserveur_sql TYPE mysql  
VERSION 4.0 WRAPPER nom_encapsuleur  
OPTIONS (NODE 'noeud_odbc', DBNAME 'venice')
```

monserveur_sql

Nom que vous attribuez au serveur de source de données ODBC. Les doublons de noms de définition de serveur ne sont pas autorisés.

TYPE *mysql*

Indique le type de serveur de source de données auquel vous configurez l'accès. Ce paramètre est facultatif.

VERSION 4.0

La version de la source de données ODBC à laquelle vous voulez accéder. Ce paramètre est facultatif.

WRAPPER *nom_encapsuleur*

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

NODE '*noeud_odbc*'

Le nom du noeud (nom DSN système) qui a été affecté à la source de données ODBC lorsque le DSN a été défini. Sur les serveurs fédérés qui fonctionnent sous Windows, cette valeur doit être le nom d'un DSN système dans la fenêtre Administrateur de sources de données ODBC. Sur les serveurs fédérés qui fonctionnent sous UNIX, le nom du noeud est le DSN défini dans le fichier de configuration ODBC. Le fichier de configuration ODBC est habituellement appelé *odbc.ini*.

Bien que le nom du noeud soit indiqué comme option dans l'instruction CREATE SERVER, il doit être indiqué obligatoirement pour les sources de données ODBC.

DBNAME '*venice*'

Facultatif. Le nom de la source de données ODBC à laquelle vous voulez accéder.

Options de serveur

Lorsque vous créez la définition de serveur, vous pouvez indiquer des options de serveur supplémentaires dans l'instruction CREATE SERVER. Les options de serveur peuvent être des options de serveur générales et des options de serveur spécifiques à ODBC.

Certaines sources de données ODBC (par exemple MySQL) ne peuvent pas traiter les apostrophes autour des noms de table et des noms de colonne dans les instructions SQL. Pour accéder à ces sources de données, vous devez inclure les options de serveur suivantes dans l'instruction CREATE SERVER :

- DB2_TABLE_QUOTE_CHAR ' ` '
- DB2_ID_QUOTE_CHAR ' ` '
- DB2_AUTHID_QUOTE_CHAR ' ` '

Le caractère ` est le délimiteur utilisé pour les identificateurs tels que les noms de schéma, les noms de table et les noms de colonne.

Par exemple :

```
CREATE SERVER monserveur_sql TYPE mysql
  VERSION 4.0 WRAPPER nom_encapsuleur
  OPTIONS (NODE 'monnoeud_sql', DB2_TABLE_QUOTE_CHAR '^',
    DB2_ID_QUOTE_CHAR '^', DB2_AUTHID_QUOTE_CHAR '^')
```

Création d'un mappage utilisateur pour une source de données ODBC

Lorsque vous essayez d'accéder à un serveur ODBC, le serveur fédéré établit une connexion au serveur ODBC à l'aide d'un ID utilisateur et d'un mot de passe qui sont valides pour cette source de données. Pour les sources de données exigeant un mappage utilisateur, vous devez définir une association (un mappage utilisateur) entre chaque ID utilisateur et mot de passe de serveur fédéré et l'ID utilisateur et le mot de passe de la source de données correspondante.

A propos de cette tâche

Créer un mappage utilisateur pour chaque ID utilisateur qui accédera au système fédéré pour envoyer des demandes réparties à la source de données ODBC.

Procédure

Pour mapper un ID utilisateur local à l'ID utilisateur et au mot de passe de la source de données ODBC :

Emettez une instruction CREATE USER MAPPING.

Par exemple :

```
CREATE USER MAPPING FOR IDutilisateur_local SERVER nom_définition_serveur
  OPTIONS (REMOTE_AUTHID 'IDutilisateur_distant', REMOTE_PASSWORD
    'motdepasse_distant');
```

Bien que REMOTE_AUTHID et REMOTE_PASSWORD soient des options du mappage utilisateur dans l'instruction CREATE USER MAPPING, ces options doivent être indiquées obligatoirement pour accéder aux sources de données ODBC :

Le mappage utilisateur doit mapper l'ID authentification DB2 à l'ID utilisateur et au mot de passe indiqués dans le fichier cac.ini.

Exemple

Si le fichier cac.ini indique USERID = MON_ID_UTILISATEUR et USERPASSWORD = MON_MOT_DE_PASSE, alors les options de l'instruction CREATE USER MAPPING doivent être indiquées comme suit :

```
REMOTE_AUTHID = MON_ID_UTILISATEUR
REMOTE_PASSWORD = MON_MOT_DE_PASSE
```

Une fois cette tâche terminée, vous pouvez tester la connexion à la source de données ODBC.

Instruction CREATE USER MAPPING - exemples pour l'encapsuleur ODBC

Utiliser l'instruction CREATE USER MAPPING pour mapper un ID utilisateur du serveur fédéré à un ID utilisateur et à un mot de passe de la source de données ODBC. Cette rubrique fournit un exemple complet avec les paramètres obligatoires et un exemple qui vous montre comment utiliser le registre spécial USER DB2 avec l'instruction CREATE USER MAPPING.

L'exemple suivant montre comment mapper un identificateur utilisateur fédéré à un ID utilisateur et à un mot de passe de la source de données ODBC :

```
CREATE USER MAPPING FOR arturo SERVER monserveur_sql
  OPTIONS (REMOTE_AUTHID 'art', REMOTE_PASSWORD 'red4blue')
```

arturo Indique l'identificateur utilisateur local que vous mappez à l'ID utilisateur et au mot de passe distants, qui sont définis sur la source de données ODBC.

monserveur_sql

Indique le nom de définition de serveur que vous avez défini dans l'instruction CREATE SERVER pour la source de données ODBC.

'art' Indique l'ID utilisateur distant auquel vous mappez *arturo*. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_ID sur 'U' ou 'L' dans l'instruction CREATE SERVER.

'red4blue'

Indique le mot de passe distant associé à *'art'*. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_PW sur 'U' ou 'L' dans l'instruction CREATE SERVER.

registre spécial USER DB2

Vous pouvez utiliser le registre spécial USER DB2 pour mapper l'identificateur utilisateur de la personne qui émet l'instruction CREATE USER MAPPING à l'identificateur utilisateur de la source de données qui est indiqué dans l'option utilisateur REMOTE_AUTHID.

L'exemple suivant montre une instruction CREATE USER MAPPING qui comporte le registre spécial USER :

```
CREATE USER MAPPING FOR USER SERVER monserveur_sql
  OPTIONS (REMOTE_AUTHID 'art', REMOTE_PASSWORD 'red4blue');
```

Test de la connexion au serveur de source de données ODBC

Tester la connexion au serveur de source de données ODBC pour déterminer si le serveur fédéré est correctement configuré pour accéder aux sources de données ODBC.

A propos de cette tâche

Vous pouvez tester la connexion au serveur de source de données ODBC à l'aide de la définition de serveur et des mappages utilisateur que vous avez définis.

Procédure

Pour tester la connexion au serveur de source de données ODBC :

Ouvrez une session passe-système et émettez une instruction SELECT sur les tables système de la source de données ODBC. Si l'instruction SELECT renvoie un comptage, votre définition de serveur et votre mappage utilisateur sont configurés correctement.

```
SET PASSTHRU nom_définition_serveur
SELECT count(*) FROM nom_schéma.nom_table
SET PASSTHRU RESET
```

Si l'instruction SELECT renvoie une erreur, vous devez identifier et résoudre les erreurs de connexion

Une fois cette tâche terminée, vous pouvez enregistrer des pseudonymes pour les tables et vues de la source de données ODBC.

Identification et résolution des erreurs de connexion à la source de données

Une connexion de test au serveur de source de données peut renvoyer une erreur pour plusieurs raisons. Il y a des mesures que vous pouvez prendre pour déterminer pourquoi l'erreur s'est produite.

Symptôme

Une erreur est renvoyée lorsque vous essayez de vous connecter à la source de données.

Cause

Il y a plusieurs causes possibles à un incident de connexion.

Résolution de l'incident

Pour identifier et résoudre les erreurs de connexion à la source de données, recherchez les problèmes éventuels avec les éléments suivants :

- Vérifiez que la source de données est disponible.
- S'il y a lieu, vérifiez que le serveur de source de données est configuré pour les connexions entrantes.
- Vérifiez que les paramètres de votre mappage utilisateur pour les options REMOTE_AUTHID et REMOTE_PASSWORD sont valides pour les connexions à la source de données. Modifiez le mappage utilisateur ou créez un autre mappage utilisateur si nécessaire.
- S'il y a lieu, vérifiez que le logiciel client de la source de données sur le serveur fédéré est installé et configuré correctement pour se connecter à la source de données.
- Pour les sources de données ODBC, vérifiez que le pilote ODBC sur le serveur fédéré est installé et configuré correctement pour se connecter au serveur de source de données ODBC. Sur les serveurs fédérés qui fonctionnent sous Windows, utilisez l'outil Administrateur de sources de données ODBC pour vérifier le pilote. Sur les serveurs fédérés qui fonctionnent sous UNIX, consultez la documentation du fournisseur de client ODBC.
- Vérifiez que les valeurs des variables définies sur le serveur fédéré sont appropriées à la source de données. Ces variables sont les variables d'environnement système, les variables du fichier db2dj.ini et les variables du registre de profil DB2 (db2set).
- Vérifiez votre définition de serveur. Si nécessaire, supprimez la définition de serveur et recréez-la.

Enregistrement de pseudonymes pour les tables et vues de la source de données ODBC

Dans chaque définition de serveur ODBC que vous enregistrez, vous devez enregistrer un pseudonyme pour chaque table ou vue à laquelle vous voulez accéder. Utilisez ces pseudonymes, au lieu des noms des objets de source de données, lorsque vous interrogez les sources de données ODBC.

Avant de commencer

Mettez à jour les statistiques sur la source de données ODBC avant d'enregistrer un pseudonyme. La base de données fédérée exploite les statistiques du catalogue de la source de données pour optimiser le traitement des requêtes. Utilisez la commande de source de données équivalant à la commande RUNSTATS DB2 pour mettre à jour les statistiques de la source de données.

Procédure

Pour enregistrer un pseudonyme pour une table ou une vue de la source de données ODBC, utilisez une des méthodes suivantes.

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .
Emettez l'instruction CREATE NICKNAME. La longueur des pseudonymes peut aller jusqu'à 128 caractères.	Par exemple : CREATE NICKNAME <i>pseudonyme</i> FOR <i>nom_définition_serveur</i> ." <i>schéma_distant</i> ". " <i>table.distante</i> " ;

Lorsque vous créez le pseudonyme, le serveur fédéré interroge le catalogue de la source de données à l'aide du pseudonyme. Cette requête teste la connexion à la table ou à la vue de la source de données. Si la connexion ne fonctionne pas, vous recevez un message d'erreur.

Répétez cette étape pour chaque table ou vue ODBC pour laquelle vous voulez créer un pseudonyme.

Instruction CREATE NICKNAME - exemples pour l'encapsuleur ODBC

Utiliser l'instruction CREATE NICKNAME pour enregistrer un pseudonyme pour une table ou une vue ODBC à laquelle vous voulez accéder. Cette rubrique fournit un exemple complet avec les paramètres obligatoires.

L'exemple suivant montre comment enregistrer un pseudonyme pour une table ou une vue ODBC à l'aide de l'instruction CREATE NICKNAME.

```
CREATE NICKNAME cust_europe FOR monserveur_sql."vinnie".italy"
```

cust_europe

Pseudonyme unique utilisé pour identifier la table ou la vue ODBC. Le pseudonyme doit être unique à l'intérieur du schéma.

Important : Le pseudonyme est un nom composé de deux parties, le schéma et le pseudonyme. Si vous omettez le schéma lorsque vous enregistrez le pseudonyme, le schéma du pseudonyme sera l'identificateur utilisateur de l'utilisateur qui enregistre le pseudonyme.

Si votre source de données ODBC ne prend pas en charge les schémas, omettez le schéma de l'instruction CREATE NICKNAME.

```
monserveur_sql."vinnie".italy"
```

Identificateur en trois parties pour l'objet distant :

- *monserveur_sql* est le nom de définition de serveur que vous avez affecté au serveur de source de données ODBC dans l'instruction CREATE SERVER.
- *vinnie* est l'ID utilisateur du propriétaire de la table ou vue.
- *italy* est le nom de la table ou de la vue distante à laquelle vous voulez accéder.

Le serveur fédéré convertit les noms des schémas et tables ODBC en majuscules, à moins que vous ne placiez les noms entre apostrophes.

Optimisation des performances de l'encapsuleur ODBC avec l'utilitaire d'optimisation ODBC (db2fedsvrcfg)

Vous pouvez optimiser les performances de l'encapsuleur ODBC avec l'utilitaire d'optimisation ODBC, db2fedsvrcfg. Cet utilitaire applique un ensemble de requêtes prédéfinies à la source de données et teste la précision des résultats. L'utilitaire crée un ensemble d'instructions ALTER SERVER que vous pouvez appliquer au serveur pour définir les options de serveur afin d'obtenir des performances optimales.

Avant de commencer

Les éléments suivants doivent être configurés sur votre serveur fédéré :

- fédération
- L'encapsuleur ODBC
- Le logiciel client ODBC
- Les variables pour l'environnement système. Les variables ODBCINI et LIBPATH peuvent être obligatoires.

Pour des informations sur l'installation, la configuration et les exigences pour ODBC, voir la documentation fournie avec chacun de ces produits.

Procédure

Pour optimiser les performances de l'encapsuleur ODBC avec l'utilitaire d'optimisation ODBC :

1. Si votre serveur ODBC n'a pas été défini, utilisez le Centre de contrôle DB2 ou l'interpréteur de commandes DB2 pour vous connecter au serveur fédéré et créer un encapsuleur et un serveur ODBC.
2. Facultatif : si les tables de test existent déjà sur votre source de données ODBC, connectez-vous à votre source de données ODBC et supprimez-les.
3. Exécutez l'utilitaire d'optimisation ODBC depuis la ligne de commande DB2 avec les options que vous voulez utiliser. L'exécution de l'utilitaire peut prendre un certain temps.
4. Vérifiez que l'utilitaire a été exécuté avec succès. Si l'utilitaire a été exécuté avec succès, vous voyez le message suivant dans la fenêtre de commande ou dans le fichier de sortie que vous avez indiqué :

```
ALTER SERVER "DS1" OPTIONS (ADD option1, 'value1')
ALTER SERVER "DS1" OPTIONS (ADD option2, 'value2')
ALTER SERVER "DS1" OPTIONS (ADD option3, 'value3')
.....
```

La commande db2fedsvrcfg a été exécutée avec succès.

Si la commande n'a pas été exécutée avec succès, vous voyez un message d'erreur indiquant la cause de l'erreur. Corrigez le problème et réexécutez la commande.

Vous devez créer les tables de test manuellement dans les cas suivants :

- Si vous voulez utiliser des noms de table différents des noms par défaut
 - Si la source de données à laquelle vous accédez avec ODBC est en lecture seule
 - Si l'utilitaire d'optimisation ODBC ne parvient pas à créer les tables de test qui sont attendues par l'utilitaire
5. Connectez-vous au serveur fédéré sur lequel la source de données est définie.
 6. Utilisez l'utilitaire db2look pour enregistrer vos paramètres de serveur existants avant d'exécuter le fichier créé par l'utilitaire d'optimisation ODBC. Voir la documentation de l'utilitaire db2look pour des informations sur l'enregistrement de vos paramètres de serveur existants.
 7. Facultatif : si votre serveur ODBC est défini, vous pouvez vous connecter au serveur fédéré et supprimer les options de serveur. L'utilitaire crée des instructions ALTER SERVER au format décrit à l'étape 4. Si ces options de serveur ont déjà été ajoutées, les instructions ALTER SERVER échoueront.
 8. Utilisez la commande suivante pour exécuter les instructions ALTER SERVER qui ont été générées par l'utilitaire pour la base de données fédérée. Les instructions ALTER SERVER qui ont été créées par l'utilitaire d'optimisation ODBC se trouvent dans le fichier db2fedsvr.cfg.sql.
db2 -tvf db2fedsvr.cfg.sql
 9. Vérifiez les résultats des instructions ALTER SERVER. Si l'une des instructions a échoué, vous pouvez modifier les instructions dans le fichier db2fedsvr.cfg.sql et réexécuter les instructions jusqu'à ce qu'elles réussissent.
 10. Une fois que vous avez terminé l'optimisation du serveur avec l'utilitaire d'optimisation ODBC, définissez l'option de serveur PUSHDOWN sur 'Y' pour terminer le processus d'optimisation.

Syntaxe de la commande db2fedsvr.cfg - utilitaire d'optimisation ODBC

Utiliser la commande db2fedsvr.cfg pour améliorer les performances de l'encapsuleur ODBC.

Syntaxe

```
db2fedsvr.cfg -s serverName [-m odbcDriverManagerLibrary] -dsn odbcDSNname  
[-dbname dsDBname] [-u userid] [-p password] [-noprep] [-prefix tableNamePrefix] [-suffix  
tableNameSuffix] [-dscp codePage] [-v] [-o outputFile] [-h]
```

Paramètres

Utilisez la commande db2fedsvr.cfg32 lorsque vous utilisez des pilotes ODBC 32 bits sous AIX ou Solaris. Sinon, utilisez la commande db2fedsvr.cfg.

-dbname dsDBname

Nom de la base de données de la source de données.

-dscp codePage

Identificateur de la page de codes pour la source de données. Si cette option n'est pas indiquée, l'utilitaire utilise la page de codes de l'environnement de l'utilisateur. Ce paramètre est facultatif.

-dsn odbcDSNname

Nom de source de données système (DSN) de la source de données.

-h Affiche l'aide détaillée. Ce paramètre est facultatif.

-m odbcDriverManagerLibrary

Nom qualifié complet du fichier de la bibliothèque du gestionnaire de pilote ODBC. Le nom du fichier de bibliothèque du gestionnaire de pilote ODBC est facultatif sous Windows.

-noprep

Empêche la création des tables de tests sur la source de données avant les tests. Ce paramètre est facultatif.

-o outputFile

Nom qualifié complet du fichier de sortie de l'utilitaire d'optimisation ODBC. Le fichier de sortie contient les instructions ALTER SERVER qui sont utilisées pour ajuster les performances de l'encapsuleur ODBC. Ce paramètre est facultatif. Si ce paramètre n'est pas indiqué, la sortie s'affiche dans la fenêtre de commande.

-p password

Mot de passe pour se connecter à la source de données. Ce paramètre est facultatif.

-prefix tableNamePrefix

Le préfixe des noms de table de la source de données ODBC que l'utilitaire utilise pour l'analyse. Si aucun préfixe n'est indiqué, c'est le préfixe par défaut, IITEST, qui est utilisé. Ce paramètre est facultatif.

-s serverName

Nom du serveur fédéré.

-suffix tableNameSuffix

Le suffixe des noms de table de la source de données ODBC que l'utilitaire utilise pour l'analyse. Si aucun suffixe n'est indiqué, c'est une chaîne vide qui est utilisée.

-u userid

Nom d'utilisateur pour se connecter à la source de données. Ce paramètre est facultatif.

-v Indique que la sortie de l'utilitaire est prolixe. Ce paramètre est facultatif.

Exemple

L'exemple suivant montre la commande qui est appliquée au magasin de données, la source de données ODBC. Dans cet exemple, les tables de test sont nommées ABC n XYZ, où n est un nombre de 1 à 7.

```
db2fedsvrcfg -s DS1 -m "/usr/lib/odbc.a"  
-dsn datastore -dbname db1 -u authid -p password -noprep  
-prefix ABC -suffix XYZ -o "/home/user1/db2fedsvrcfg.sql"
```

Testez les définitions de table pour l'utilitaire d'optimisation ODBC (db2fedsvrcfg)

Dans certains cas, vous devez créer manuellement les tables de test pour l'utilitaire d'optimisation ODBC. Les définitions de table de test sont décrites dans cette rubrique.

Vous devez créer les tables de test pour l'utilitaire d'optimisation ODBC dans les cas suivants :

- Si vous voulez utiliser des noms de table différents des noms par défaut
- Si la source de données à laquelle vous accédez avec ODBC est en lecture seule
- Si l'utilitaire d'optimisation ODBC ne parvient pas à créer les tables de test qui sont attendues par l'utilitaire

La définition de table suivante s'applique aux sept tables de test qui sont nécessaires (IITEST1 à IITEST7). Le préfixe du nom de table par défaut est IITEST et le suffixe par défaut est une chaîne vide. Si vous indiquez un préfixe et un suffixe différents, vous devez indiquer les options -prefix et -suffix lorsque vous exécutez l'utilitaire d'optimisation ODBC.

Tableau 29. Définition de table de test de l'utilitaire d'optimisation ODBC pour la table IITEST1

Nom de colonne	Type de données SQL	Identificateur de type de données SQL	Longueur
IT1C1	integer	SQL_INTEGER	
IT1C2	integer	SQL_INTEGER	
IT1C3	char(1)	SQL_CHAR	1
IT1C4	char(3)	SQL_CHAR	3
IT1C5	char(10)	SQL_CHAR	10
IT1C6	varchar(10)	SQL_VARCHAR	10
IT1C7	char(100)	SQL_CHAR	100

Tableau 30. Définition de table de test de l'utilitaire d'optimisation ODBC pour la table IITEST2

Nom de colonne	Type de données SQL	Identificateur de type de données SQL	Longueur
IT2C1	integer	SQL_INTEGER	
IT2C2	integer	SQL_INTEGER	
IT2C3	char(30)	SQL_CHAR	30

Tableau 31. Définition de table de test de l'utilitaire d'optimisation ODBC pour les tables IITEST3 à IITEST7

Nombre de colonnes dans chaque table	Nom de colonne	Type de données SQL	Identificateur de type de données SQL	Longueur
66	ITxCn	char(100)	SQL_CHAR	100

x Nombre correspondant à la table à définir.

n Numéro de colonne.

Par exemple, IT3C1 est le nom de colonne de la première colonne de la table IITEST3.

Configuration de l'accès aux sources de données OLE DB

Pour configurer le serveur fédéré pour accéder aux sources de données OLE DB, vous devez fournir au serveur fédéré des informations sur les fournisseurs OLE DB.

Avant de commencer

- La fédération doit être installée sur un serveur qui fait office de serveur fédéré.
- Vérifiez la configuration du serveur fédéré.

- Vérifiez le paramètre fédéré pour vérifier que la fédération est activée.

Microsoft OLE DB est un ensemble d'interfaces de modèles COM qui permettent aux applications d'accéder à des informations qui ne sont généralement pas stockées dans des bases de données. Un utilisateur OLE DB est tout élément de code système ou de code d'application qui utilise une interface OLE DB. Un fournisseur OLE DB est tout composant de logiciel qui présente une interface OLE DB.

Vous pouvez configurer un serveur fédéré pour accéder aux données qui sont stockées dans les sources de données OLE DB en émettant des instructions SQL sur la ligne de commande DB2.

Une fois l'accès à la source de données OLE DB configuré, utilisez l'instruction CREATE FUNCTION pour enregistrer une fonction de table externe OLE DB définie par l'utilisateur dans la base de données fédérée.

Procédure

Pour configurer l'accès aux sources de données OLE DB d'un serveur fédéré :

1. Enregistrez l'encapsuleur.
2. Enregistrez la définition de serveur.
3. Créez les mappages utilisateur.

Enregistrement de l'encapsuleur OLE DB

Vous devez enregistrer un encapsuleur pour accéder aux sources de données OLE DB. Les serveurs fédérés utilisent des encapsuleurs pour dialoguer avec les sources de données et en extraire des données. Les encapsuleurs sont mis en oeuvre comme un ensemble de fichiers de bibliothèque.

A propos de cette tâche

L'encapsuleur OLE DB vous permet d'accéder aux fournisseurs OLE DB compatibles avec Microsoft OLE DB version 2.0 (ou ultérieure).

L'encapsuleur OLE DB sert uniquement à vous aider à enregistrer des fonctions de table externes OLE DB définies par l'utilisateur dans la base de données fédérée. Contrairement à d'autres encapsuleurs, l'encapsuleur OLE DB n'utilise pas de pseudonymes pour accéder aux données qui sont stockées dans les sources de données.

Procédure

Pour enregistrer l'encapsuleur OLE DB :

Utilisez l'une des méthodes suivantes pour enregistrer l'encapsuleur OLE DB :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .

Méthode	Description
<p>Emettez l'instruction CREATE WRAPPER et indiquez le nom par défaut pour l'encapsuleur OLE DB.</p>	<p>Par exemple :</p> <pre>CREATE WRAPPER OLEDB</pre> <p>A faire : Lorsque vous enregistrez l'encapsuleur à l'aide du nom par défaut, OLEDB, le serveur fédéré utilise automatiquement la bibliothèque de l'encapsuleur OLE DB approprié au système d'exploitation sous lequel votre serveur fédéré fonctionne.</p> <p>En cas de conflit entre le nom d'encapsuleur par défaut et un nom d'encapsuleur existant dans la base de données fédérée, vous pouvez remplacer le nom d'encapsuleur par défaut par un nom que vous choisissez. Lorsque vous utilisez un nom différent du nom par défaut, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER.</p> <p>Par exemple, pour enregistrer un encapsuleur avec le nom encapsuleur_oledb sur un serveur fédéré qui fonctionne sous Windows, émettez l'instruction suivante :</p> <pre>CREATE WRAPPER encapsuleur_oledb LIBRARY 'db2oledb.dll'</pre> <p>Le fichier de bibliothèque de l'encapsuleur que vous indiquez dépend du système d'exploitation du serveur fédéré.</p>

Une fois cette tâche terminée, vous pouvez enregistrer la définition de serveur.

Fichiers de bibliothèque de l'encapsuleur OLE DB

Les fichiers de bibliothèque de l'encapsuleur OLE DB sont ajoutés au serveur fédéré lorsque vous installez l'encapsuleur.

Lorsque vous installez l'encapsuleur OLE DB, le fichier de bibliothèque est ajouté dans le chemin de répertoire par défaut.

Si vous n'utilisez pas le nom d'encapsuleur par défaut lorsque vous enregistrez l'encapsuleur, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER et indiquer le nom de fichier de bibliothèque.

Le chemin du répertoire par défaut et le nom de fichier de bibliothèque de l'encapsuleur par défaut sont indiqués dans la table suivante.

Tableau 32. Emplacements et noms de fichier de bibliothèque du client OLE DB

Système d'exploitation	Chemin de répertoire	Nom de fichier de bibliothèque
Windows	%DB2PATH%\bin	db2oledb.dll

Enregistrement des définitions de serveur pour une source de données OLE DB

Vous devez enregistrer chaque serveur de source de données OLE DB auquel vous voulez accéder dans la base de données fédérée.

A propos de cette tâche

Vous enregistrez une définition de serveur pour une source de données OLE DB dans la ligne de commande DB2.

Procédure

Pour enregistrer une définition de serveur pour une source de données OLE DB :

Emettez l'instruction CREATE SERVER.

Par exemple :

```
CREATE SERVER nom_définition_serveur WRAPPER nom_encapsuleur
      OPTIONS (CONNECTSTRING 'keyword=value;keyword=value');
```

Bien que la variable CONNECTSTRING soit indiquée comme option dans l'instruction CREATE SERVER, cette option est obligatoire pour les sources de données OLE DB.

Une fois cette tâche terminée, vous pouvez créer un mappage utilisateur.

Instruction CREATE SERVER - exemples pour l'encapsuleur OLE DB

Utiliser l'instruction CREATE SERVER pour enregistrer des définitions de serveur pour l'encapsuleur OLE DB. Cette rubrique fournit un exemple complet avec les paramètres obligatoires et un exemple avec des options de serveur supplémentaires.

L'exemple suivant vous montre comment enregistrer une définition de serveur pour un encapsuleur OLE DB en émettant l'instruction CREATE SERVER :

```
CREATE SERVER Nwind WRAPPER OLEDB
      OPTIONS (CONNECTSTRING 'Provider=Microsoft.Jet.OLEDB.4.0;
      Data Source=c:\msdasdk\bin\oledb\nwind.mdb');
```

Nwind Nom que vous attribuez à la source de données OLE DB. Les doublons de noms de définition de serveur ne sont pas autorisés.

WRAPPER *OLEDB*

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

CONNECTSTRING '*Provider=Microsoft.Jet.OLEDB.4.0; Data Source=c:\msdasdk\bin\oledb\nwind.mdb*'

Fournit les propriétés d'initialisation qui sont nécessaires pour se connecter à un fournisseur OLE DB.

La valeur de l'option CONNECTSTRING contient une série de paires de mot clé et de valeur qui sont séparées par des points-virgule. Le signe égale (=) sépare chaque mot clé et sa valeur. Les mots clé sont les descriptions des propriétés d'initialisation d'OLE DB (ensemble de propriétés DBPROPSET_DBINT) ou des mots clé spécifiques au fournisseur.

Pour la syntaxe et la sémantique complètes de l'option `CONNECTSTRING`, voir *Microsoft OLE DB 2.0 Programmer's Reference and Data Access SDK*, Microsoft Press, 1998.

Options de serveur

Lorsque vous créez la définition de serveur, vous pouvez indiquer des options de serveur supplémentaires dans l'instruction `CREATE SERVER`. Les options de serveur peuvent être des options de serveur générales et des options de serveur spécifiques à OLE DB.

L'option de serveur `COLLATING_SEQUENCE` indique si la source de données utilise la même séquence de classement que le serveur fédéré. La valeur par défaut de l'option de serveur `COLLATING_SEQUENCE` est 'N'. Si la source de données OLE DB utilise la même séquence de classement que le serveur fédéré, définissez l'option de serveur `COLLATING_SEQUENCE` sur 'Y'.

L'exemple suivant montre une définition de serveur OLE DB qui comporte l'option de serveur `COLLATING_SEQUENCE` :

```
CREATE SERVER Nwind WRAPPER OLEDB
  OPTIONS (CONNECTSTRING 'Provider=Microsoft.Jet.OLEDB.4.0;
    Data Source=c:\msdasdk\bin\oledb\nwind.mdb',
  COLLATING_SEQUENCE 'Y')
```

Création des mappages utilisateur pour une source de données OLE DB

Lorsque vous essayez d'accéder à un serveur OLE DB, le serveur fédéré établit une connexion au serveur OLE DB à l'aide d'un ID utilisateur et d'un mot de passe qui sont valides pour cette source de données. Vous devez définir une association (un mappage utilisateur) entre chaque identificateur utilisateur et mot de passe fédérés et l'ID utilisateur et le mot de passe de la source de données correspondante.

Procédure

Pour mapper un identificateur utilisateur local à l'ID utilisateur et au mot de passe OLE DB distants :

Emettez une instruction `CREATE USER MAPPING`.

Par exemple :

```
CREATE USER MAPPING FOR local_userid SERVER nom_définition_serveur
  OPTIONS (REMOTE_AUTHID 'IDutilisateur_distant', REMOTE_PASSWORD
  'motdepasse_distant');
```

Si la longueur du mot de passe sur la source de données OLE DB ou du mot de passe sur le serveur fédéré est inférieure à huit caractères, les instructions SQL qui accèdent à la source de données OLE DB échoueront. Le message d'erreur suivant apparaîtra :

```
SQL30082N La tentative d'établir la connexion a échoué pour la cause de sécurité "15"
("PROCESSING FAILURE"). SQLSTATE=08001
```

Pour éviter ce problème, modifiez le mot de passe de la source de données OLE DB ou modifiez le mot de passe sur le serveur fédéré pour qu'il compte au moins huit caractères.

Instruction CREATE USER MAPPING - exemples pour l'encapsuleur OLE DB

Utiliser l'instruction CREATE USER MAPPING pour mapper un identificateur utilisateur fédéré à un ID utilisateur et à un mot de passe OLE DB distants. Cette rubrique comporte un exemple complet avec les paramètres obligatoires et un exemple qui vous montre comment utiliser le registre spécial USER DB2 avec l'instruction CREATE USER MAPPING.

L'exemple suivant montre comment mapper un identificateur utilisateur fédéré à un ID utilisateur et à un mot de passe de la source de données OLE distants :

```
CREATE USER MAPPING FOR laura SERVER Nwind
  OPTIONS (REMOTE_AUTHID 'lulu', REMOTE_PASSWORD 'raiders')
```

laura Indique l'identificateur utilisateur local que vous mappez à un ID utilisateur et à un mot de passe distants, qui sont définis sur la source de données OLE DB.

SERVER *Nwind*

Indique le nom de définition de serveur que vous avez enregistré dans l'instruction CREATE SERVER pour le serveur OLE DB.

REMOTE_AUTHID '*lulu*'

Indique l'ID utilisateur sur le serveur de la base OLE DB auquel vous mappez *lulu*. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_ID sur 'U' ou 'L' dans l'instruction CREATE SERVER.

REMOTE_PASSWORD '*raiders*'

Indique le mot de passe associé à '*lulu*'. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_PW sur 'U' ou 'L' dans l'instruction CREATE SERVER.

registre spécial USER DB2

Vous pouvez utiliser le registre spécial USER DB2 pour mapper l'identificateur utilisateur de la personne qui émet l'instruction CREATE USER MAPPING à l'identificateur utilisateur de la source de données qui est indiqué dans l'option utilisateur REMOTE_AUTHID.

L'exemple suivant montre une instruction CREATE USER MAPPING qui comporte le registre spécial USER :

```
CREATE USER MAPPING FOR USER SERVER Nwind
  OPTIONS (REMOTE_AUTHID 'lulu', REMOTE_PASSWORD 'raiders');
```

Configuration de l'accès aux sources de données Oracle

Lors de la configuration d'un système fédéré en vue d'accéder aux sources de données Oracle, vous devez fournir au système fédéré des informations sur les sources de données et sur les objets auxquels vous voulez accéder.

Avant de commencer

- Le logiciel client Oracle doit être installé et configuré sur le serveur agissant en tant que serveur fédéré.
- La fédération doit être installée sur un serveur qui fait office de serveur fédéré.
- Vérifiez la configuration du serveur fédéré.
- Vérifiez le paramètre fédéré pour vérifier que la fédération est activée.

Procédure

Pour ajouter des sources de données Oracle à un système fédéré, procédez comme suit :

1. Définissez les variables d'environnement Oracle.
2. Paramétrez et testez le fichier de configuration client Oracle.
3. Enregistrez l'encapsuleur Oracle.
4. Enregistrez des définitions de serveur pour une source de données Oracle.
5. Créez les mappages utilisateur.
6. Testez la connexion au serveur Oracle.
7. Enregistrez les pseudonymes.

Définition des variables d'environnement Oracle

Les variables d'environnement Oracle doivent être définies dans le fichier db2dj.ini sur le serveur fédéré.

Restrictions

Lisez les restrictions pour le fichier db2dj.ini.

Le fichier db2dj.ini contient des informations de configuration sur le logiciel client Oracle installé sur le serveur fédéré.

Il existe des variables d'environnement obligatoires et facultatives pour les sources de données Oracle.

Si vous avez installé le logiciel client Oracle avant d'installer l'encapsuleur Oracle, les variables d'environnement Oracle obligatoires sont définies sur le fichier db2dj.ini.

Si vous n'avez pas installé le logiciel client Oracle avant d'installer l'encapsuleur Oracle, ou si vous voulez définir l'une des variables d'environnement facultatives, vous devez définir les variables d'environnement en suivant les étapes de cette tâche.

Procédure

Pour définir les variables d'environnement Oracle, procédez comme suit :

1. Utilisez l'une des méthodes suivantes :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .
Définissez automatiquement les variables d'environnement.	Exécutez l'assistant d'installation IBM InfoSphere Federation Server. Suivez les instructions de l'assistant. Important : Définissez les variables d'environnement obligatoires en exécutant l'assistant d'installation. Les variables d'environnement facultatives doivent être définies manuellement.

Méthode	Description
Définissez manuellement les variables d'environnement.	<p>Modifiez le fichier db2dj.ini :</p> <p>Le fichier db2dj.ini se trouve dans le répertoire indiqué par la variable de registre DB2 DB2_DJ_INI. Si la variable DB2_DJ_INI n'est pas définie, le fichier db2dj.ini figure dans un des chemins par défaut suivants selon le système d'exploitation :</p> <ul style="list-style-type: none"> Sous Linux et UNIX : <i>nominstance</i>/sqlib/cfg/db2dj.ini. <i>instancehome</i> Le répertoire de base du propriétaire d'instance. Sous Windows : %DB2PATH%\cfg\db2dj.ini %DB2PATH% Le répertoire dans lequel le système de la base de données DB2 est installé, par exemple, C:\Program Files\IBM\sqlib. <p>Si le fichier n'existe pas, vous pouvez créer un fichier portant le nom db2dj.ini à l'aide de tout éditeur de texte. Dans le fichier db2dj.ini, vous devez indiquer le chemin qualifié complet dans la valeur des variables d'environnement, sinon vous rencontrerez des erreurs.</p>

- Sur les serveurs fédérés sous UNIX, ajoutez la variable d'environnement Oracle au fichier .profile de l'instance DB2. Par exemple :

```
export ORACLE_HOME=répertoire_principale_oracle
export PATH=$ORACLE_HOME/bin:$PATH
```

Où *répertoire_principale_oracle* correspond au répertoire d'installation du logiciel client Oracle.

- Sur les serveurs fédérés sous Linux ou UNIX, exécutez le fichier .profile d'instance DB2 en saisissant :

```
. $HOME/.profile
```

- Pour vous assurer que les variables d'environnement sont définies sur le serveur fédéré, recyclez l'instance DB2 avec les commandes suivantes :

```
db2stop
db2start
```

Une fois cette tâche terminée, vous pouvez configurer et tester le client Oracle.

Variables d'environnement Oracle

Il existe des variables d'environnement obligatoires et facultatives pour les sources de données Oracle. Ces variables sont définies dans le fichier db2dj.ini.

Les variables d'environnement suivantes sont valides pour Oracle :

- ORACLE_HOME
- ORACLE_BASE (facultative)
- ORA_NLS (facultative)
- NLS_LANG (facultative)
- TNS_ADMIN (facultative)

Descriptions des variables

ORACLE_HOME

Définissez la variable d'environnement ORACLE_HOME sur le chemin du répertoire dans lequel le logiciel client Oracle est installé. Spécifiez le

chemin d'accès qualifié complet de la variable d'environnement :
ORACLE_HOME=répertoire_principal_oracle. Par exemple, si le répertoire principal Oracle est \usr\oracle\8.1.7, l'entrée du fichier db2dj.ini est
ORACLE_HOME=\usr\oracle\8.1.7.

Si un utilisateur individuel de l'instance fédérée définit localement la variable d'environnement ORACLE_HOME, l'instance fédérée n'utilise pas ce paramètre. L'instance fédérée utilise uniquement la valeur de ORACLE_HOME, qui est définie dans le fichier db2dj.ini.

ORACLE_BASE

La variable ORACLE_BASE représente la racine de l'arborescence de répertoires client Oracle. Si vous définissez la variable d'environnement ORACLE_BASE lors de l'installation du logiciel client Oracle, définissez-la sur le serveur fédéré.

Par exemple :

```
ORACLE_BASE=répertoire_principal_oracle
```

ORA_NLS

Si plusieurs versions d'Oracle sont en cours d'exécution sur votre système, assurez-vous des points suivants :

- La variable d'environnement ORA_NLS adéquate est définie
- Les fichiers de données NLS correspondants pour les versions que vous utilisez sont disponibles

Les données spécifiques aux emplacements sont stockées dans un répertoire indiqué par la variable d'environnement ORA_NLS. Chaque version d'Oracle a un répertoire de données ORA_NLS différent.

Tableau 33. Variable d'environnement ORA_NLS par version

Version Oracle	Variable d'environnement
8.x, 9.x	ORA_NLS33
10.x	ORA_NLS10

Par exemple, sur les serveurs fédérés sous UNIX qui ont accès aux sources de données Oracle 8.1, le paramètre de la variable d'environnement ORA_NLS33 est le suivant :

```
ORA_NLS33=répertoire_principal_oracle/ocommon/nls/admin/<data>
```

NLS_LANG

La variable d'environnement NLS_LANG est une variable d'environnement de pages de codes. Pour plus d'informations sur le paramétrage de cette variable, consultez la documentation Oracle adéquate.

TNS_ADMIN

Sur les serveurs fédérés sous Windows

Le client Oracle recherche le fichier tnsnames.ora dans le répertoire %ORACLE_HOME%\NETWORK\ADMIN, où %ORACLE_HOME% est définie dans le fichier db2dj.ini. Si le fichier tnsnames.ora ne figure pas dans ce répertoire, vous devez définir la variable d'environnement TNS_ADMIN sur le serveur fédéré. Définissez cette variable d'environnement dans le fichier db2dj.ini sur le chemin dans lequel figure le fichier tnsnames.ora.

Sur les serveurs fédérés sous AIX ou Linux

Le client Oracle recherche le fichier tnsnames.ora dans le répertoire

/etc. Si le fichier tnsnames.ora n'est pas ce répertoire, le client Oracle recherche alors le fichier tnsnames.ora dans le répertoire \$ORACLE_HOME/network/admin, où \$ORACLE_HOME est définie dans le fichier db2dj.ini. Si le fichier tnsnames.ora n'est pas dans ce répertoire, vous devez définir la variable d'environnement TNS_ADMIN sur le serveur fédéré. Vous définissez la variable d'environnement dans le fichier db2dj.ini sur le chemin où figure le fichier tnsnames.ora.

Par exemple, si le fichier tnsnames.ora réside dans le répertoire /principal/oracle, définissez la variable d'environnement comme suit :

```
TNS_ADMIN=/principal/oracle
```

Sur les serveurs fédérés sous Solaris

Le client Oracle recherche le fichier tnsnames.ora dans le répertoire /var/opt/oracle. Si le fichier tnsnames.ora ne figure pas dans ce répertoire, le client Oracle recherche alors le fichier tnsnames.ora dans le répertoire \$ORACLE_HOME/network/admin, où \$ORACLE_HOME est définie dans le fichier db2dj.ini. Si le fichier tnsnames.ora n'est pas dans ce répertoire, vous devez définir la variable d'environnement TNS_ADMIN. Vous définissez la variable dans le fichier db2dj.ini sur le chemin où figure le fichier tnsnames.ora.

Par exemple, si le fichier tnsnames.ora réside dans le répertoire /principal/oracle, définissez la variable d'environnement comme suit :

```
TNS_ADMIN=/principal/oracle
```

Conversion de pages de codes Oracle :

Chaque fois que l'encapsuleur Oracle se connecte à une source de données Oracle, l'encapsuleur détermine la valeur de page de codes à utiliser pour cette connexion. Vous pouvez indiquer que l'encapsuleur Oracle définit la valeur de page de codes ou vous pouvez désigner une page de codes en définissant la variable d'environnement NLS_LANG.

Les variables d'environnement précisant la conversion de pages de codes Oracle sont définies dans le fichier db2dj.ini sur votre serveur fédéré.

Si la variable d'environnement NLS_LANG est définie dans le fichier db2dj.ini sur le serveur fédéré, l'encapsuleur utilise la valeur de pages de codes du fichier db2dj.ini.

Si la variable d'environnement NLS_LANG n'est pas définie dans le fichier db2dj.ini sur le serveur fédéré, l'encapsuleur détermine le territoire et la page de codes de la base de données fédérée. L'encapsuleur définit la variable d'environnement NLS_LANG sur l'environnement local Oracle correspondant le plus proche. Si aucun environnement local proche ne correspond, la variable d'environnement NLS_LANG est définie sur American_America.US7ASCII.

Pour obtenir la liste des paramètres nationaux valides, consultez la documentation jointe à votre logiciel Oracle.

Exemple de la page de codes chinoise GB 18030 :

Si vous accédez à une source de données Oracle contenant des données codées avec la page de codes chinoise GB 18030, et que votre base de données fédérée utilise la page de codes UTF-8, l'encapsuleur Oracle définit la variable d'environnement Oracle NLS_LANG sur :

```
NLS_LANG=Simplified Chinese_China.UTF8
```

Ce paramètre est correct si vous utilisez un client Oracle 8, mais si vous utilisez la version 9i (ou une version ultérieure) du client Oracle, vous devez définir la variable d'environnement NLS_LANG pour remplacer le paramètre par défaut de l'encapsuleur Oracle. Définissez la variable d'environnement NLS_LANG sur Simplified Chinese_China.AL32UTF8, de sorte que le client Oracle 9i convertisse correctement les données GB 18030 en Unicode.

Par exemple :

```
NLS_LANG=Simplified Chinese_China.AL32UTF8
```

Paramétrage et test du fichier de configuration client Oracle

Le fichier de configuration client Oracle permet de se connecter aux bases de données Oracle à l'aide des bibliothèques client installées sur le système fédéré.

A propos de cette tâche

Le fichier de configuration client spécifie l'emplacement de chaque serveur de base de données Oracle ainsi que le type de connexion (protocole) du serveur de base de données.

Le nom par défaut du fichier de configuration client Oracle est tnsnames.ora.

L'emplacement par défaut du fichier de configuration client dépend du système d'exploitation utilisé par le système fédéré :

- Sur les systèmes Linux et UNIX, l'emplacement par défaut du fichier est le suivant : \$ORACLE_HOME/network/admin.
- Sur les systèmes Windows, l'emplacement par défaut du fichier est le suivant : %ORACLE_HOME%\NETWORK\ADMIN.

Procédure

Pour paramétrer et tester le fichier de configuration client Oracle, procédez comme suit :

1. Créez le fichier tnsnames.ora à l'aide de l'utilitaire de configuration Oracle NET8/NET, qui est fourni avec le logiciel client Oracle.
Dans le fichier tnsnames.ora, SID (or SERVICE_NAME) correspond au nom de l'instance Oracle, et HOST correspond au nom de l'hôte sur lequel le serveur Oracle se situe.
2. Si vous souhaitez placer le fichier tnsnames.ora sur un chemin différent du chemin de recherche par défaut, définissez la variable d'environnement TNS_ADMIN pour spécifier l'emplacement du fichier :
 - a. Editez le fichier db2dj.ini dans le répertoire sqllib/cfg, et définissez la variable d'environnement TNS_ADMIN. Par exemple :

```
TNS_ADMIN=x:/chemin/
```

- b. Emettez les commandes suivantes pour recycler l'instance DB2 et vérifier que la variable d'environnement est définie dans le programme :


```
db2stop
db2start
```
3. Testez la connexion pour vérifier que le logiciel client peut se connecter au serveur Oracle. Pour cela, utilisez l'utilitaire Oracle sqlplus.

Une fois cette tâche terminée, vous pouvez enregistrer l'encapsuleur Oracle.

Enregistrement de l'encapsuleur Oracle

Pour accéder aux sources de données Oracle, vous devez enregistrer un encapsuleur. Les serveurs fédérés utilisent des encapsuleurs pour dialoguer avec les sources de données et en extraire des données. Les encapsuleurs sont mis en oeuvre comme un ensemble de fichiers de bibliothèque.

Procédure

Pour enregistrer un encapsuleur Oracle, procédez comme suit :

Utilisez l'une des méthodes suivantes :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .
Emettre l'instruction CREATE WRAPPER et spécifier le nom par défaut de l'encapsuleur Oracle.	<p>Par exemple :</p> <pre>CREATE WRAPPER NET8 ;</pre> <p>A faire : lorsque vous enregistrez l'encapsuleur avec le nom par défaut, NET8, le serveur fédéré utilise automatiquement la bibliothèque d'encapsuleur Oracle adéquate pour le système d'exploitation sur lequel le serveur fédéré s'exécute.</p> <p>Si vous n'utilisez pas le nom d'encapsuleur par défaut lors de l'enregistrement d'un encapsuleur, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER et préciser le nom de fichier de bibliothèque d'encapsuleur par défaut.</p> <p>Par exemple, pour enregistrer un encapsuleur intitulé encapsuleur_oracle sur le serveur fédéré exécuté sur AIX, émettez l'instruction suivante :</p> <pre>CREATE WRAPPER encapsuleur_oracle LIBRARY 'libdb2net8.a'</pre> <p>Le fichier de bibliothèque d'encapsuleur spécifié dépend du système d'exploitation du serveur fédéré.</p>

Une fois cette tâche terminée, vous pouvez enregistrer les définitions de serveur.

fichiers de bibliothèque d'encapsuleur Oracle

Les fichiers de bibliothèque d'encapsuleur Oracle sont ajoutés au serveur fédéré lorsque vous installez l'encapsuleur.

Lors de l'installation de l'encapsuleur Oracle, trois fichiers de bibliothèque sont ajoutés au chemin de répertoire par défaut. Par exemple, si le serveur fédéré s'exécute sous AIX, les fichiers de bibliothèque d'encapsuleur ajoutés au chemin de répertoire suivant sont libdb2net8.a, libdb2net8F.a et libdb2net8U.a. Le fichier de bibliothèque d'encapsuleur par défaut est libdb2net8.a. Les autres fichiers de bibliothèque d'encapsuleur sont utilisés en interne par l'encapsuleur Oracle.

Les chemins des répertoires par défaut et les noms de fichier de bibliothèque de l'encapsuleur par défaut sont indiqués dans la table suivante.

Tableau 34. Emplacements des bibliothèques d'encapsuleur Oracle et noms des fichiers

Système d'exploitation	Chemin de répertoire	Noms de fichier de bibliothèque
AIX	/usr/opt/chemin_installation/lib32/	libdb2net8.a
	/usr/opt/chemin_installation/lib64/	
Linux	/opt/IBM/db2/chemin_installation/lib32	libdb2net8.so
	/opt/IBM/db2/chemin_installation/lib64	
Solaris	/opt/IBM/db2/chemin_installation/lib32 /opt/IBM/db2/chemin_installation/lib64	libdb2net8.so
Windows	%DB2PATH%\bin	db2net8.dll

chemin_installation est le chemin du répertoire dans lequel le serveur fédéré est installé sous UNIX ou Linux.

Enregistrement des définitions de serveur pour une source de données Oracle

Vous devez enregistrer chaque serveur Oracle auquel vous souhaitez accéder dans la base de données fédérée.

Procédure

Pour enregistrer une définition de serveur pour une source de données Oracle, procédez comme suit :

1. Recherchez le nom de noeud dans le fichier Oracle tnsnames.ora.

Exemple de fichier tnsnames :

```
noeud_paris =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = somehost)(PORT = 1521)))
    (CONNECT_DATA = (SERVICE_NAME = ora9i.seel)))
```

Dans cet exemple, le nom de noeud à utiliser dans l'instruction CREATE SERVER est le suivant : noeud_paris.

2. Pour créer la définition de serveur, utilisez une des méthodes suivantes :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .

Méthode	Description
Emettez l'instruction CREATE SERVER	<p>Par exemple :</p> <pre>CREATE SERVER <i>nom_de_définition_de_serveur</i> TYPE oracle VERSION <i>numéro_version</i> WRAPPER <i>nom_encapsuleur</i> OPTIONS (NODE '<i>nom_de_noeud</i>') ;</pre> <p>Même si la variable <i>nom_de_noeud</i> est spécifiée en tant qu'option dans l'instruction CREATE SERVER, elle est obligatoire pour les sources de données Oracle.</p> <p>Une fois la définition de serveur enregistrée, utilisez l'instruction ALTER SERVER pour ajouter ou déplacer des options de serveur.</p>

Une fois cette tâche terminée, vous pouvez créer les mappages utilisateur.

Instruction CREATE SERVER - Exemples pour l'encapsuleur Oracle

L'instruction CREATE SERVER permet d'enregistrer des définitions de serveur pour l'encapsuleur Oracle. Cette rubrique fournit un exemple complet avec les paramètres obligatoires et un exemple avec des options de serveur supplémentaires.

L'exemple suivant présente le mode d'enregistrement d'une définition de serveur, pour un encapsuleur Oracle, par émission de l'instruction CREATE SERVER :

```
CREATE SERVER oraserver TYPE oracle VERSION 8.1.7 WRAPPER nom_encapsuleur
  OPTIONS (NODE 'noeud_paris') ;
```

serveurora

Nom attribué au serveur de base de données Oracle. Les doublons de noms de définition de serveur ne sont pas autorisés.

TYPE *oracle*

Indique le type de serveur de source de données auquel vous configurez l'accès. Pour l'encapsuleur Oracle, le type de serveur doit être *oracle*.

VERSION *8.1.7*

Version du serveur de base de données Oracle auquel vous souhaitez accéder.

WRAPPER *nom_encapsuleur*

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

NODE '*noeud_paris*'

Nom du noeud sur lequel le serveur de base de données Oracle réside. Récupérez le nom du noeud dans le fichier tnsnames.ora. Cette valeur fait la distinction entre les majuscules et les minuscules.

Bien que le nom du noeud soit spécifié en tant qu'option de l'instruction CREATE SERVER, il est obligatoire pour les sources de données Oracle.

Options de serveur

Lorsque vous créez la définition de serveur, vous pouvez indiquer des options de serveur supplémentaires dans l'instruction CREATE SERVER. Ces options de serveur peuvent être générales ou spécifiques à Oracle.

Le serveur fédéré suppose que l'ensemble des colonnes Oracle VARCHAR contiennent des blancs de fin. Si vous êtes sûr que toutes les colonnes VARCHAR de la base de données Oracle ne contiennent pas de blancs de fin, vous pouvez définir une option de serveur indiquant que la source de données utilise une sémantique de comparaison VARCHAR sans blanc.

L'exemple suivant présente une définition de serveur Oracle avec l'option de serveur VARCHAR_NO_TRAILING_BLANKS :

```
CREATE SERVER oraserver TYPE oracle VERSION 8.1.7 WRAPPER nom_encapsuleur
  OPTIONS (NODE 'noeud_paris', VARCHAR_NO_TRAILING_BLANKS 'Y') ;
```

Utilisez l'option de serveur VARCHAR_NO_TRAILING_BLANKS lorsqu'aucune colonne ne contient de blanc de fin. Si seulement quelques colonnes VARCHAR contiennent des blancs de fin, vous pouvez définir une option sur ces colonnes avec l'instruction ALTER NICKNAME.

Création des mappages utilisateur pour une source de données Oracle

Pour accéder à un serveur Oracle, le serveur fédéré établit une connexion vers le serveur Oracle à l'aide d'un ID utilisateur et d'un mot de passe valides pour cette source de données.

Restrictions

L'ID utilisateur au niveau de la source de données Oracle doit être créé à l'aide de la commande create user Oracle avec la clause `identifié par`, au lieu de la clause `identifié en externe`.

Procédure

Pour créer les mappages utilisateurs pour une source de données Oracle, procédez comme suit :

Emettez une instruction CREATE USER MAPPING. Par exemple :

```
CREATE USER MAPPING FOR IDutilisateur_local SERVER nom_définition_serveur
  OPTIONS (REMOTE_AUTHID 'IDutilisateur_distant', REMOTE_PASSWORD
  'motdepasse_distant') ;
```

Une fois cette tâche terminée, testez la connexion au serveur Oracle.

Instruction CREATE USER MAPPING - Exemples pour l'encapsuleur Oracle

L'instruction CREATE USER MAPPING permet de mapper un identificateur utilisateur fédéré avec un ID utilisateur et un mot de passe Oracle distants. Cette rubrique comporte un exemple complet avec les paramètres obligatoires et un exemple qui vous montre comment utiliser le registre spécial USER DB2 avec l'instruction CREATE USER MAPPING.

L'exemple suivant présente la procédure de mappage d'un identificateur utilisateur fédéré avec un ID utilisateur et un mot de passe Oracle :

```
CREATE USER MAPPING FOR robert SERVER serveurora
  OPTIONS (REMOTE_AUTHID 'rob', REMOTE_PASSWORD 'then4now') ;
```

robert Indique l'identificateur utilisateur que vous mappez vers un ID utilisateur et un mot de passe distants, définis sur le serveur Oracle.

SERVER *serveurora*

Indique le nom de la définition de serveur enregistré dans l'instruction CREATE SERVER pour le serveur Oracle.

REMOTE_AUTHID *'rob'*

Indique l'ID utilisateur distant vers lequel vous mappez *robert*. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_ID sur 'U' ou 'L' dans l'instruction CREATE SERVER.

REMOTE_PASSWORD *'then4now'*

Indique le mot de passe distant associé à *rob*. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_PW sur 'U' ou 'L' dans l'instruction CREATE SERVER.

registre spécial USER DB2

Vous pouvez utiliser le registre spécial USER DB2 pour mapper l'identificateur utilisateur de la personne qui émet l'instruction CREATE USER MAPPING à l'identificateur utilisateur de la source de données qui est indiqué dans l'option utilisateur REMOTE_AUTHID.

L'exemple suivant montre une instruction CREATE USER MAPPING qui comporte le registre spécial USER :

```
CREATE USER MAPPING FOR USER SERVER oraserver
  OPTIONS (REMOTE_AUTHID 'rob', REMOTE_PASSWORD 'then4now') ;
```

Test de la connexion au serveur Oracle

Le test de la connexion au serveur Oracle permet de déterminer si le serveur fédéré est correctement configuré pour pouvoir accéder aux sources de données Oracle.

A propos de cette tâche

Vous pouvez tester la connexion au serveur Oracle en utilisant la définition de serveur et les mappages utilisateur que vous avez définis.

Procédure

Pour tester la connexion au serveur Oracle, procédez comme suit :

Ouvrez une session passe-système et émettez une instruction SELECT sur les tables système Oracle. Si l'instruction SELECT renvoie un nombre, votre définition de serveur et votre mappage utilisateur sont correctement configurés.

Par exemple :

```
SET PASSTHRU nom_du_serveur_distant
SELECT count(*) FROM sys.all_tables
SET PASSTHRU RESET
```

Si l'instruction SELECT renvoie une erreur, vous devez identifier et résoudre les erreurs de connexion.

Une fois cette tâche terminée, vous pouvez enregistrer les pseudonymes pour les tables et les vues Oracle.

Identification et résolution des incidents de connectivité avec les sources de données Oracle

L'incident le plus courant que vous êtes susceptible de rencontrer lors de la configuration du serveur fédéré en vue d'accéder aux sources de données Oracle est la connectivité.

Symptôme

Si vous ne parvenez pas à vous connecter à une source de données Oracle depuis un serveur fédéré, vous devriez peut-être mettre à jour le fichier hôte TCP/IP.

Cause

Ce problème peut être provoqué par un fichier hôte TCP/IP périmé.

Résolution de l'incident

Pour chaque hôte de la section DESCRIPTION du fichier tnsnames.ora, vous devriez peut-être mettre à jour le fichier hôte TCP/IP. La mise à jour de ce fichier dépend du mode de configuration du protocole TCP/IP sur votre réseau. Un composant du réseau doit convertir le nom de l'hôte distant, spécifié dans la section DESCRIPTION du fichier tnsnames.ora, en adresse.

Si votre réseau comporte un serveur de noms qui reconnaît le nom d'hôte, vous n'êtes pas tenu de mettre à jour le fichier hôte TCP/IP. Si votre réseau n'a pas de serveur de noms qui reconnaît le nom d'hôte, vous devez ajouter une entrée au fichier hôte TCP/IP pour l'hôte distant.

L'emplacement du fichier hôte TCP/IP dépend du système d'exploitation utilisé par le système fédéré :

Sur les serveurs fédérés sous Linux ou UNIX

Le fichier hôte figure dans le répertoire /etc/hosts.

Sur les serveurs fédérés sous Windows

Le fichier hôte figure dans le répertoire x:\winnt\system32\drivers\etc\hosts.

Identification et résolution des erreurs de connexion à la source de données

Une connexion de test au serveur de source de données peut renvoyer une erreur pour plusieurs raisons. Il y a des mesures que vous pouvez prendre pour déterminer pourquoi l'erreur s'est produite.

Symptôme

Une erreur est renvoyée lorsque vous essayez de vous connecter à la source de données.

Cause

Il y a plusieurs causes possibles à un incident de connexion.

Résolution de l'incident

Pour identifier et résoudre les erreurs de connexion à la source de données, recherchez les problèmes éventuels avec les éléments suivants :

- Vérifiez que la source de données est disponible.
- S'il y a lieu, vérifiez que le serveur de source de données est configuré pour les connexions entrantes.
- Vérifiez que les paramètres de votre mappage utilisateur pour les options REMOTE_AUTHID et REMOTE_PASSWORD sont valides pour les connexions à la source de données. Modifiez le mappage utilisateur ou créez un autre mappage utilisateur si nécessaire.
- S'il y a lieu, vérifiez que le logiciel client de la source de données sur le serveur fédéré est installé et configuré correctement pour se connecter à la source de données.
- Pour les sources de données ODBC, vérifiez que le pilote ODBC sur le serveur fédéré est installé et configuré correctement pour se connecter au serveur de source de données ODBC. Sur les serveurs fédérés qui fonctionnent sous Windows, utilisez l'outil Administrateur de sources de données ODBC pour vérifier le pilote. Sur les serveurs fédérés qui fonctionnent sous UNIX, consultez la documentation du fournisseur de client ODBC.
- Vérifiez que les valeurs des variables définies sur le serveur fédéré sont appropriées à la source de données. Ces variables sont les variables d'environnement système, les variables du fichier db2dj.ini et les variables du registre de profil DB2 (db2set).
- Vérifiez votre définition de serveur. Si nécessaire, supprimez la définition de serveur et recréez-la.

Enregistrement des pseudonymes pour les tables et les vues Oracle

Pour chaque définition de serveur Oracle que vous enregistrez, vous devez enregistrer un pseudonyme pour chaque table ou vue à laquelle vous voulez accéder. Lorsque vous interrogez les serveurs Oracle, utilisez ces pseudonymes au lieu des noms des objets de source de données.

Avant de commencer

Avant d'enregistrer un pseudonyme, mettez à jour les statistiques de la source de données Oracle. La base de données fédérée exploite les statistiques du catalogue de la source de données pour optimiser le traitement des requêtes. Utilisez la commande de source de données équivalant à la commande RUNSTATS DB2 pour mettre à jour les statistiques de la source de données.

Pour les objets de source de données qui utilisent Oracle Label Security, les données des pseudonymes ne peuvent pas être placées dans la mémoire cache. Vous pouvez activer ou désactiver la mise en cache à l'aide de l'instruction ALTER NICKNAME.

Restriction : la création d'un pseudonyme sur un synonyme Oracle d'un synonyme n'est pas prise en charge et provoque l'émission du message d'erreur SQL0204.

Procédure

Pour enregistrer un pseudonyme pour une table ou une vue Oracle, utilisez l'une des méthodes suivantes :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .
Emettez l'instruction CREATE NICKNAME.	Par exemple : <pre>CREATE NICKNAME <i>pseudonyme</i> FOR <i>nom_définition_serveur</i>."<i>schéma_distant</i>". "<i>table.distante</i>" ;</pre>

Lors de la création du pseudonyme, le serveur fédéré interroge le catalogue de la source de données à l'aide du pseudonyme. Cette requête teste la connexion à la table ou à la vue de source de données. Si la connexion ne fonctionne pas, un message d'erreur s'affiche. Répétez cette étape pour chaque table ou vue Oracle pour laquelle vous voulez créer un pseudonyme.

Instruction CREATE NICKNAME - Exemples pour l'encapsuleur Oracle

L'instruction CREATE NICKNAME permet d'enregistrer un pseudonyme pour une table ou une vue Oracle à laquelle vous voulez accéder. Cette rubrique fournit un exemple complet avec les paramètres obligatoires.

L'exemple suivant présente la procédure d'enregistrement d'un pseudonyme pour une table ou une vue Oracle avec l'instruction CREATE NICKNAME.

```
CREATE NICKNAME PARISINV FOR oraserver."vinnie".inventory ;
```

PARISINV

Pseudonyme unique utilisé pour identifier la table ou la vue Oracle. Le pseudonyme est composé du schéma et du pseudonyme. Si vous omettez le schéma lorsque vous enregistrez le pseudonyme, le schéma du pseudonyme sera l'identificateur utilisateur de l'utilisateur qui enregistre le pseudonyme.

```
oraserver."vinnie".inventory"
```

Identificateur en trois parties pour l'objet distant :

- *oraserver* correspond au nom de la définition de serveur que vous avez attribué au serveur de base de données Oracle dans l'instruction CREATE SERVER.
- *vinnie* est l'ID utilisateur du propriétaire de la table ou vue.
- *inventory* correspond au nom de la table ou de la vue distante à laquelle vous souhaitez accéder.

Le serveur fédéré met les noms des tables et des schémas Oracle en majuscules, sauf si vous les entourez de guillemets.

Configuration de l'accès aux scripts en tant que sources de données

Pour configurer un système fédéré pour accéder aux scripts en tant que sources de données, vous devez enregistrer des fonctions personnalisées, un encapsuleur, une définition de serveur et des pseudonymes pour les scripts.

Avant de commencer

- La fédération doit être installée sur un serveur qui fait office de serveur fédéré.

- Il doit y avoir une base de données fédérée sur le serveur fédéré.

A propos de cette tâche

Vous pouvez configurer un serveur fédéré pour accéder aux données à travers des scripts en émettant des instructions SQL sur la ligne de commande DB2.

Procédure

Pour ajouter des scripts comme sources de données à un serveur fédéré :

1. Identifiez ou écrivez un script.
2. Enregistrez la fonction personnalisée.
3. Configurez le démon Script.
4. Démarrez le démon Script.
5. Enregistrez l'encapsuleur de script.
6. Enregistrez la définition de serveur pour une source de données à laquelle il est accédé par un script.
7. Enregistrez des pseudonymes pour les sources de données.

Présentation de l'encapsuleur de script

Vous pouvez utiliser des scripts, tels que des scripts Perl, pour accéder aux informations des bases de données ou pour générer des données. Vous pouvez intégrer ces données avec des données provenant d'autres sources de données fédérées à l'aide de l'encapsuleur de script.

Vous pouvez avoir des scripts existants qui renvoient des données à partir de sources de données telles que des banques de données de biologie ou qui génèrent leurs propres données. L'encapsuleur de script permet d'utiliser des scripts comme s'ils étaient des sources de données fédérées. L'encapsuleur de script permet d'accéder à des données qui peuvent alors être intégrées à l'aide d'un système fédéré. Les scripts doivent renvoyer des résultats en XML.

Les pseudonymes de l'encapsuleur de script peuvent comporter des colonnes d'entrée et de sortie. Ces pseudonymes utilisent des modèles de fonction dans des prédicats pour transmettre au script des valeurs en entrée. Les données de sortie du script sont représentées en XML sous une forme hiérarchique, qui peut alors être mappée à des pseudonymes à l'aide de clés primaires et externes.

L'encapsuleur de script est un encapsuleur en lecture seule. L'encapsuleur de script ne peut pas écrire de données dans une source de données.

Dans le diagramme suivant, les données passent d'un script à travers l'encapsuleur de script et le démon Script jusqu'à la base de données fédérée, où les données peuvent être intégrées avec des données provenant d'autres sources et visualisées avec le client fédéré. Si cela est souhaité, le système fédéré peut accéder au script à travers un serveur proxy et un pare-feu.

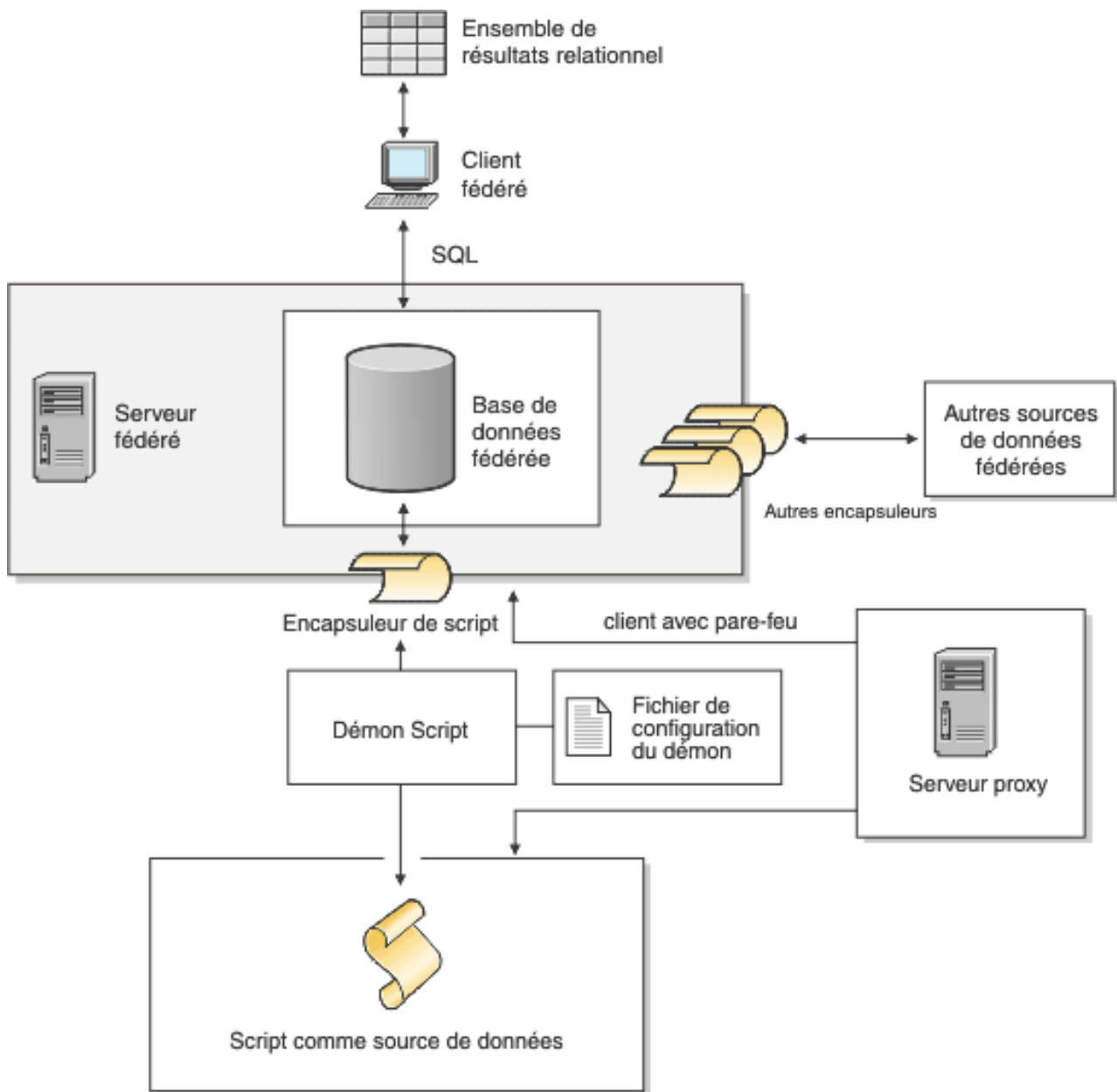


Figure 7. L'encapsuleur de script dans un système fédéré

Un script est appelé à partir du répertoire qui contient le démon Script. Si le script extrait les données de son propre fichier de données et ne peut pas trouver le fichier de données, le script peut comporter des chemins relatifs. Utilisez des chemins d'accès absolus dans des scripts.

Ajout de scripts en tant que sources de données d'un système fédéré

Pour configurer un serveur fédéré pour accéder à des scripts comme sources de données, vous devez configurer le démon Script, un encapsuleur, une définition de serveur et des pseudonymes pour les scripts.

Enregistrement de la fonction personnalisée pour le script

Vous devez enregistrer la fonction personnalisée de script WSSCRIPT.ARGS avant d'enregistrer l'encapsuleur de script.

A propos de cette tâche

Vous devez enregistrer la fonction personnalisée sur chaque instance de base de données fédérée sur laquelle l'encapsuleur de script est installé.

La fonction personnalisée pour l'encapsuleur de script doit être enregistrée sous le nom de schéma WSSCRIPT.

Vous devez inclure des mots clé spécifiques lorsque vous enregistrez la fonction personnalisée pour l'encapsuleur de script. Incluez les mots clé AS TEMPLATE, DETERMINISTIC et NO EXTERNAL ACTION dans l'instruction CREATE FUNCTION.

L'environnement fédéré utilise deux moteurs de requête. Pour l'encapsuleur de script, ces moteurs de requête sont le moteur de requête de la base de données fédérée et le moteur de requête de l'encapsuleur de script. Vous pouvez indiquer que les prédicats seront envoyés au moteur de l'encapsuleur de script en utilisant des fonctions personnalisées de l'encapsuleur de script dans la clause WHERE de votre instruction SELECT.

Le fichier create_function_mappings.ddl dans le répertoire sqllib/exemples/lifesci/script sur le serveur fédéré indique les types de données pour la fonction personnalisée.

Procédure

Pour enregistrer la fonction personnalisée de script :

Exécutez le fichier create_function_mappings.ddl sur chaque instance de base de données fédérée sur laquelle l'encapsuleur de script est installé.

L'exemple suivant montre la syntaxe de la fonction WSSCRIPT.ARGS :

```
CREATE FUNCTION WSSCRIPT.ARGS (type_données  
  
_colonne_entrée(), type_données_colonne_entrée())  
  RETURNS INTEGER AS TEMPLATE  
  DETERMINISTIC NO EXTERNAL ACTION;
```

Types de données pour la fonction personnalisée de l'encapsuleur de script :

Vous devez indiquer le type de données de la colonne d'entrée deux fois dans chaque fonction personnalisée.

Enregistrez une fonction personnalisée WSSCRIPT distincte pour chaque type de données valide pour l'argument nom_colonne. La fonction personnalisée WSSCRIPT a les types de données suivants :

- VARCHAR
- INTEGER
- CLOB
- DOUBLE

- DATE

Les deux paramètres de la fonction personnalisée doivent être du même type de données, c'est-à-dire du type de données de la colonne d'entrée correspondante. Lorsqu'il est utilisé dans un prédicat de requête, le premier paramètre est le nom de la colonne d'entrée du commutateur. La deuxième colonne est la valeur à transmettre au script pour ce commutateur.

Configuration du démon Script

L'encapsuleur de script exige un démon Script à l'écoute des demandes de travail de script provenant de l'encapsuleur de script. Le démon Script doit être configuré avant que vous n'enregistriez l'encapsuleur de script.

Avant de commencer

Le démon Script a les prérequis suivants :

- Il dispose d'un accès en écriture à un répertoire dans lequel le démon puisse écrire des fichiers temporaires.
- Il est exécuté sur un serveur auquel vous pouvez accéder par TCP/IP depuis votre système fédéré. Ce serveur peut être le même serveur qui fait office de serveur fédéré ou un serveur de script distinct.
- Il nécessite un fichier de configuration qui doit être sur le même serveur que le démon Script.
-
- Il est exécuté séparément dans l'encapsuleur de script et la base de données fédérée.

Procédure

Pour configurer le démon Script :

1. Vérifiez que les fichiers exécutables du démon Script sont sur le bon serveur. Vous pouvez avoir besoin de copier les fichiers exécutables du démon Script sur un autre serveur.

Pendant l'installation de IBM InfoSphere Federation Server, les fichiers exécutables du démon Script sont installés sur le serveur fédéré. Le nom et l'emplacement du fichier sont les suivants :

UNIX db2script_daemon est installé dans le répertoire \$DB2PATH/bin.
\$DB2PATH est le répertoire dans lequel le serveur fédéré est installé.

Windows

db2script_daemon.exe est installé dans le répertoire %DB2PATH%\bin.
%DB2PATH% est le répertoire dans lequel le serveur fédéré est installé, habituellement C:\SQLLIB\bin.

Si vous utilisez un serveur de script distinct, copiez les fichiers exécutables et de configuration du démon Script du serveur fédéré vers le serveur de script. Les fichiers exécutables du démon Script peuvent être exécutés dans n'importe quel répertoire du serveur de script qui ne contient pas d'espaces dans les noms du chemin de répertoire.

2. Vérifiez que le fichier de configuration du démon Script est sur le bon serveur. Pendant l'installation du système fédéré, un exemple de fichier de configuration pour le démon Script est installé sur le serveur fédéré. Le nom de l'exemple de fichier de configuration est SCRIPT_DAEMON.config. L'emplacement du fichier est le suivant :

UNIX Le fichier de configuration du démon est installé dans le répertoire \$DB2PATH/bin.

Windows

Le fichier de configuration du démon est installé dans le répertoire %DB2PATH%\bin.

Par défaut, le démon recherche le fichier de configuration dans le répertoire de travail à partir duquel le démon a été démarré. Vous pouvez copier le fichier de configuration dans un autre emplacement. Si vous utilisez un serveur de script, copiez le fichier de configuration du démon de son répertoire sur le serveur fédéré vers un répertoire sur le serveur de script. Vous pouvez copier le fichier de configuration du démon dans n'importe quel répertoire sur le serveur de script auquel le démon peut accéder.

3. Modifiez l'exemple de fichier de configuration pour le démon Script.
 - a. Renommez le fichier de configuration de façon à pouvoir réutiliser le fichier modèle.
 - b. Vérifiez que la première ligne du fichier de configuration est un signe égale (=). Si le signe égale est absent, le démon ne démarre pas. Un message d'erreur indiquera que DAEMON_PORT n'a pas été indiqué.
 - c. Vérifiez que la dernière ligne du fichier de configuration se termine par un saut de ligne.

L'exemple de fichier de configuration qui est fourni avec le système fédéré se termine par un caractère de saut de ligne. Si la dernière ligne ne se termine pas par un caractère de saut de ligne, vous recevez un message d'erreur lorsque vous essayez d'exécuter votre première requête de script qui utilise la source de données indiquée sur la dernière ligne.

- d. Vérifiez qu'il n'y a pas d'espaces supplémentaires après les chemins de répertoire ou à la fin du fichier de configuration.
- e. Indiquez les options suivantes dans le fichier de configuration. Pour les options attendant des chemins, vous pouvez indiquer des chemins relatifs. Les chemins relatifs sont relatifs au répertoire à partir duquel le processus démon a été démarré.

DAEMON_PORT=*numéro_port*

Le port réseau sur lequel le démon est à l'écoute des demandes de travail de script qui sont soumises par l'encapsuleur. La valeur par défaut est 4099.

MAX_PENDING_REQUESTS=*nombre_de_demandes*

Le nombre maximum de demandes de travail de script pouvant être bloquées sur le démon à un moment donné. Ce nombre ne représente pas le nombre de travaux de script qui s'exécutent en même temps, mais seulement le nombre de demandes de travail qui peuvent être bloquées à la fois. Définissez cette valeur sur un nombre supérieur à cinq. Le démon Script ne restreint pas le nombre de travaux de script qui peuvent être exécutés en même temps.

DAEMON_LOGFILE_DIR=*dir*

Le répertoire dans lequel le démon crée son fichier journal. Ce fichier contient les informations de statut et d'erreur générées par le démon Script.

SCRIPT_OUT_DIR_PATH=*path*

Le répertoire dans lequel le démon crée le fichier temporaire pour stocker les données de sortie du script. Le démon lit les données de

ce fichier et retransmet les données à l'encapsuleur à travers la connexion réseau. Une fois les données transmises à l'encapsuleur, le démon efface le fichier temporaire

script specification entry=entry

Liste d'entrées indiquant le nom et l'emplacement des scripts qui peuvent être appelés par l'encapsuleur de script. L'entrée a le format suivant :

nom_script=chemin_script_qualifié_complet

Les exemples suivants s'appliquent au système d'exploitation indiqué :

UNIX Par exemple, pour indiquer un script accédant à une source de données Oracle, ajoutez la ligne suivante au fichier de configuration du démon :

`oracle=/dsk/1/data/oracle`

Windows

Par exemple, pour indiquer un script accédant à une source de données Oracle, ajoutez la ligne suivante au fichier de configuration du démon :

`oracle=c:\data\oracle.a`

L'exemple suivant montre un fichier SCRIPT_DAEMON.cfg pour quatre scripts :

```
=  
DAEMON_PORT=4099  
MAX_PENDING_REQUESTS=10  
DAEMON_LOGFILE_DIR=./  
SCRIPT_OUT_DIR_PATH=./  
fee=/home/user_id/fee  
fie=/home/user_id/fie  
foe=/home/user_id/foe  
fum=/home/user_id/fum
```

L'exemple de fichier de configuration du démon Script fournit un exemple de configuration du démon Script.

Démarrage du démon Script

Avant de pouvoir accéder aux sources de données à l'aide de scripts, vous devez démarrer le démon Script.

Avant de commencer

Vous devez avoir un accès en écriture à tous les chemins qui sont indiqués dans le fichier de configuration du démon pour les options DAEMON_LOGFILE_DIR et SCRIPT_OUT_DIR_PATH.

A propos de cette tâche

Le fichier exécutable du démon Script démarre un nouveau processus dans lequel le démon Script est exécuté.

Procédure

Pour démarrer le démon Script :

1. Ouvrez le répertoire dans lequel le fichier exécutable du démon se trouve.

2. Emettez la commande `db2script_daemon` pour exécuter les fichiers exécutables, en incluant les options appropriées.

L'exemple suivant comporte des options :

```
db2script_daemon -a action -c fichier_configuration -d niveau_débogage -u
```

```
id_utilisateur -p motdepasse
```

Commande `db2script_daemon` - options et exemples :

La commande `db2script_daemon` démarre le démon Script. Vous pouvez démarrer le démon Script avec plusieurs options.

La commande `db2script_daemon` peut être utilisée sur des serveurs UNIX ou Windows. Certaines des options indiquées dans la syntaxe peuvent être utilisées uniquement sur des serveurs Windows.

Les options qui sont indiquées avec l'action `start` n'ont d'effet que sur l'instance actuelle du démon et remplacent les valeurs indiquées avec l'action `install`.

Options - commande `db2script_daemon`

La commande `db2script_daemon` a les options suivantes :

-a action (Windows uniquement)

Exécute l'activité indiquée. Les actions valides sont `status`, `install`, `start`, `stop` et `remove`.

-c fichier_configuration

Demande au service démon d'utiliser le fichier de configuration indiqué. Si vous n'indiquez pas le fichier de configuration, le démon recherche le fichier `SCRIPT_DAEMON.config` dans le répertoire dans lequel les fichiers exécutables du démon sont installés. Vous pouvez utiliser cette option avec les actions `install` et `start`.

-d [1 | 2 | 3]

Définit le niveau de débogage pour le service démon sur la valeur indiquée. Une valeur de 1 active la consignation, 2 trace toutes les commandes et 3 active la consignation, trace toutes les commandes et sauvegarde tous les fichiers temporaires pour capturer la sortie XML. Vous pouvez utiliser cette option avec les actions `install` et `start`.

-u user_id (Windows uniquement)

Définit le service démon pour qu'il soit exécuté sous l'ID utilisateur indiqué. Vous pouvez utiliser cette option avec l'action `install`.

-p motdepasse (Windows uniquement)

Indique le mot de passe de l'ID utilisateur fourni. Le mot de passe n'est valide et obligatoire que si vous indiquez l'option `-u`. Si l'option `-p` n'est pas indiquée lorsque vous définissez l'option `-u`, le programme vous demande le mot de passe. Vous pouvez utiliser cette option avec l'action `install`.

Exemples - commande `db2script_daemon`

Les exemples suivants montrent comment utiliser les options du démon Script.

Démarrez le démon

Pour démarrer le démon sous UNIX, émettez la commande suivante :

```
db2script_daemon
```

Cette commande suppose que le fichier de configuration du démon est dans le même répertoire que le fichier exécutable.

Pour installer et démarrer le démon sous Windows, émettez les commandes suivantes :

```
db2script_daemon -a install
```

```
db2script_daemon -a start
```

Indiquez le fichier de configuration du démon

Si vous avez modifié le nom du fichier de configuration du démon ou si le fichier de configuration n'est pas dans le même répertoire que le fichier exécutable du démon, vous devez utiliser l'option -c lorsque vous exécutez le fichier exécutable. Cette option indique le chemin de répertoire et le nom du fichier de configuration du démon.

Dans cet exemple, les informations de configuration du démon sont dans un fichier appelé SCRIPT_D.config dans le sous-répertoire cfg sur un serveur UNIX. Entrez la commande suivante :

```
db2script_daemon -c cfg/SCRIPT_D.config
```

Indiquez le niveau de débogage

Si vous voulez démarrer le démon avec le débogage activé sur un niveau de débogage de 2, émettez les commandes suivantes :

```
db2blast_daemon -a install -d 2
```

```
db2blast_daemon -a start
```

Vérifier le statut du démon (Windows)

Pour vérifier le statut du démon sur un serveur Windows, émettez la commande suivante :

```
db2blast_daemon -a status
```

Cette commande suppose que le fichier de configuration du démon est dans le même répertoire que le fichier exécutable.

Arrêter le démon

Pour arrêter le démon sous UNIX, affichez l'ID processus du démon en émettant la commande suivante :

```
ps -ef | grep db2script
```

Utilisez ensuite l'ID processus pour arrêter le démon en émettant la commande suivante :

```
kill ID_processus
```

Cette commande suppose que le fichier de configuration du démon est dans le même répertoire que le fichier exécutable.

Pour arrêter le démon sous Windows, émettez la commande suivante :

```
db2script_daemon -a stop
```

Supprimer le démon

Vous pouvez supprimer le démon Script lorsque vous ne voulez plus utiliser l'encapsuleur de script.

Pour supprimer le démon, émettez la commande suivante :

```
db2script_daemon -a remove
```

Enregistrement de l'encapsuleur de script

Vous devez enregistrer l'encapsuleur de script pour accéder aux sources de données à l'aide de scripts. L'encapsuleur de script est mis en oeuvre comme un fichier de bibliothèque.

Procédure

Pour enregistrer l'encapsuleur de script :

Emettez l'instruction CREATE WRAPPER, en indiquant un nom pour l'encapsuleur de script et le nom du fichier de bibliothèque d'encapsuleur.

Par exemple, pour enregistrer un encapsuleur sous le nom encapsuleur_script sur un serveur fédéré qui fonctionne sous AIX, émettez l'instruction suivante :

```
CREATE WRAPPER encapsuleur_script LIBRARY 'libdb2lsscript.a';
```

Le nom du fichier de bibliothèque de l'encapsuleur que vous indiquez dépend du système d'exploitation du serveur fédéré.

Il n'y a pas d'options spécifiques à l'encapsuleur de script pour l'instruction CREATE WRAPPER de l'encapsuleur de script. L'encapsuleur est exécuté par défaut de manière non isolée.

Fichier de bibliothèque de l'encapsuleur de script :

Pour enregistrer l'encapsuleur de script, indiquez le fichier de bibliothèque de l'encapsuleur de script pour le système d'exploitation du serveur fédéré.

Lorsque vous installez la fédération, un fichier de bibliothèque de l'encapsuleur de script est ajouté dans le chemin de répertoire par défaut.

Les chemins des répertoires par défaut et les noms de fichier de bibliothèque de l'encapsuleur par défaut sont indiqués dans la table suivante.

Tableau 35. Emplacements et noms de fichier de bibliothèque de l'encapsuleur de script

Système d'exploitation	Chemin de répertoire	Nom de fichier de bibliothèque d'encapsuleur
AIX	/usr/opt/ <i>chemin_installation</i> /lib	libdb2lsscript.a
Linux	/opt/IBM/db2/ <i>chemin_installation</i> /lib	libdb2lsscript.so
Solaris	/opt/IBM/db2/ <i>chemin_installation</i> /lib	libdb2lsscript.so
Windows	%DB2PATH%\bin	db2lsscript.dll

- *chemin_installation* est le chemin du répertoire dans lequel la fédération est installée sous UNIX ou Linux.
- %DB2PATH% est la variable d'environnement qui est utilisée pour indiquer le chemin de répertoire dans lequel la fédération a été installée sous Windows. Le chemin de répertoire Windows par défaut est C:\Program Files\IBM\SQLLIB.

Enregistrement de la définition de serveur pour un script en tant que source de données (ligne de commande DB2)

Vous devez enregistrer chaque serveur auquel vous voulez accéder dans la base de données fédérée.

Procédure

Pour enregistrer une définition de serveur pour un script :

Emettez l'instruction CREATE SERVER.

Par exemple :

```
CREATE SERVER serveur_script WRAPPER encapsuleur_script  
  OPTIONS (NODE 'myserver.example.com', DAEMON_PORT '4099');
```

Les options de serveur NODE et DAEMON_PORT sont obligatoires pour les scripts comme sources de données.

Une fois la définition de serveur enregistrée, utilisez l'instruction ALTER SERVER pour ajouter ou déplacer des options de serveur.

Instruction CREATE SERVER - exemples pour l'encapsuleur de script :

Les exemples montrent l'utilisation des options obligatoires et des options de serveur supplémentaires.

Exemple d'options obligatoires

L'exemple suivant vous montre comment enregistrer une définition de serveur pour l'encapsuleur de script en émettant l'instruction CREATE SERVER :

```
CREATE SERVER server1_scriptn WRAPPER encapsuleur_script  
  OPTIONS(NODE 'big_rs.company.com');
```

server1_scriptn

Nom que vous attribuez au serveur de script. Les doublons de noms de définition de serveur ne sont pas autorisés.

WRAPPER *encapsuleur_script*

Le nom d'encapsuleur.

NODE '*big_rs.company.com*'

Nom d'hôte du système sur lequel le processus démon Script est exécuté. Cette valeur fait la distinction entre les majuscules et les minuscules.

Bien que le nom du noeud soit indiqué comme option dans l'instruction CREATE SERVER, il doit être indiqué obligatoirement pour l'encapsuleur de script.

Exemple d'option facultative

L'exemple suivant montre une option de serveur supplémentaire que vous pouvez indiquer lorsque vous enregistrez une définition de serveur pour l'encapsuleur de script :

```
CREATE SERVER server1_scriptn WRAPPER encapsuleur_script  
  OPTIONS(NODE 'big_rs.company.com', DAEMON_PORT '4088');
```

DAEMON_PORT '*4088*'

Indique le numéro du port sur lequel le démon est à l'écoute des demandes de travail de script. Le numéro de port doit être le même que celui que vous avez indiqué dans l'option DAEMON_PORT du fichier de configuration du démon. Le numéro de port par défaut est 4099.

Enregistrement de pseudonymes pour les scripts (ligne de commande DB2)

Vous devez enregistrer un pseudonyme distinct pour chaque script. Utilisez ces pseudonymes lorsque vous interrogez la source de données à laquelle il est accédé à l'aide d'un script.

A propos de cette tâche

L'encapsuleur de script associe les données XML à des pseudonymes. Les pseudonymes parent et enfant correspondent aux éléments racine et aux éléments imbriqués d'un document XML. Les pseudonymes parent et enfant sont connectés à l'aide de clés primaires et externes qui sont indiquées dans l'instruction CREATE NICKNAME. Chaque pseudonyme est défini par des expressions XPath qui identifient les données élémentaires XML et indiquent comment extraire les valeurs de colonne de chaque élément.

La source de données est indiquée par l'instruction CREATE NICKNAME et associée à un nom de script avec l'option de pseudonyme DATASOURCE. Vous devez créer une colonne pour chaque argument d'entrée à transmettre au script. Utilisez les options de colonne d'entrée pour contrôler la syntaxe des commutateurs sur la ligne de commande. La valeur de chaque commutateur doit être incluse dans le prédicat de requête à l'aide de la fonction personnalisée ARGS en phase d'exécution.

Les scripts simples qui n'attendent pas d'arguments de la ligne de commande obligatoires n'ont pas besoin de colonnes d'entrée.

Procédure

Pour enregistrer un pseudonyme pour un script :

Emettez l'instruction CREATE NICKNAME. Les pseudonymes peuvent comporter jusqu'à 128 caractères.

Par exemple :

```
CREATE NICKNAME pseudonyme
(
  nom_colonne type_données OPTIONS ('options_colonne_pseudonyme'),
  nom_colonne type_données OPTIONS ('options_colonne_pseudonyme'),
  nom_colonne type_données OPTIONS ('options_colonne_pseudonyme')
)
FOR SERVER nom_définition_serveur
OPTIONS (options_pseudonyme);
```

Emettez l'instruction pour chaque script pour lequel vous voulez créer un pseudonyme.

Instruction CREATE NICKNAME - exemples pour l'encapsuleur de script

Les exemples vous montrent comment utiliser l'instruction CREATE NICKNAME pour enregistrer des pseudonymes pour l'encapsuleur de script.

L'exemple suivant crée un pseudonyme parent pour les données XML renvoyées par un script appelé fee :

```
CREATE NICKNAME customers
(
  argle double OPTIONS(SWITCH '-argle', POSITION 1, DEFAULT 1.0 ),
  argfile CLOB() OPTIONS(SWITCH '-file', INPUT_MODE 'FILE_INPUT', POSITION 2),
  argpos varchar() OPTIONS(SWITCH ' ', POSITION 3),
  id VARCHAR(5) OPTIONS(XPATH './@id'),
  name VARCHAR(16) OPTIONS(XPATH './name'),
  address VARCHAR(30) OPTIONS(XPATH './address/@street'),
  cid VARCHAR(16) FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER serveur_script
OPTIONS(DATASOURCE 'fee',
  XPATH '/doc/customer', STREAMING 'YES');
```


L'exemple suivant crée un pseudonyme appelé orders. Le pseudonyme orders est un enfant du pseudonyme appelé customers, qui a été créé dans l'exemple précédent :

```
CREATE NICKNAME orders
(
  amount INTEGER      OPTIONS(XPATH './amount'),
  date   VARCHAR(10)  OPTIONS(XPATH './date'),
  oid    VARCHAR(16)  OPTIONS(PRIMARY_KEY 'YES'),
  cid    VARCHAR(16)  FOR BIT DATA NOT NULL OPTIONS(FOREIGN_KEY 'CUSTOMERS'))
FOR SERVER serveur_script
OPTIONS( XPATH './order');
```

Options de pseudonyme de l'encapsuleur de script

Vous pouvez indiquer des options lorsque vous créez un pseudonyme pour un script. Seul le pseudonyme racine peut comporter des colonnes d'entrée.

Options de pseudonyme

La liste suivante décrit les options de pseudonyme :

DATASOURCE

Le nom du script qui sera appelé. Le script doit être indiqué dans le fichier de configuration du démon Script. Cette option est obligatoire pour les sources de données du pseudonyme parent. Cette option s'applique uniquement au pseudonyme racine.

NAMESPACES

Liste séparée par des virgules de paires nom=valeur que l'encapsuleur utilise pour résoudre les préfixes d'espace de nom dans l'expression XPath du pseudonyme.

TIMEOUT

La durée maximale, en minutes, pendant laquelle l'encapsuleur de script attend les résultats du démon Script. La valeur par défaut est de 60 minutes. Cette option s'applique uniquement au pseudonyme racine.

VALIDATE

Indique si le document de base XML doit être validé avant l'extraction des données XML. Si cette option est définie sur YES, le système de la base de données DB2 vérifie que la structure du document de base est conforme à un schéma XML ou à une définition de type de document (DTD). Cette option n'est acceptée que pour les colonnes du pseudonyme racine (le pseudonyme qui identifie les éléments au niveau supérieur du document XML). La valeur par défaut est NO.

Le document de base XML n'est pas validé si l'encapsuleur de script ne peut pas trouver le fichier schéma XML ou le fichier DTD (.xsd ou .dtd). Le système de la base de données DB2 n'émet pas de message d'erreur si la validation ne s'effectue pas. Vérifiez que le fichier schéma XML ou le fichier DTD est dans l'emplacement indiqué dans le document de base XML. Ne définissez pas le paramètre VALIDATE sur YES si vous avez défini le paramètre STREAMING sur YES.

STREAMING

Indique si le document de base XML est séparé en fragments logiques correspondant au noeud qui correspond à l'expression XPath du pseudonyme. L'encapsuleur de script traite alors la source de données XML fragment par fragment, ce qui réduit l'utilisation totale de mémoire. Cette option n'est acceptée que pour les colonnes du pseudonyme racine (le pseudonyme qui identifie les éléments au niveau supérieur du document XML). La valeur de

flot de données (STREAMING) par défaut est NO. Ne définissez pas le paramètre STREAMING sur YES si vous définissez le paramètre VALIDATE sur YES.

XPATH

Expression XPath identifiant l'élément XML qui représente des blocs de données donnés dans la source de données. L'encapsuleur évalue l'option de pseudonyme XPATH pour un pseudonyme enfant dans le contexte du chemin indiqué par l'option de pseudonyme XPATH du pseudonyme parent. Cette expression Xpath est utilisée comme un contexte permettant d'évaluer les valeurs de colonne identifiées par les options de colonne du pseudonyme XPATH.

Options de colonne de pseudonymes

Les options de colonne de pseudonyme sont décrites ci-dessous :

DEFAULT

Valeur par défaut d'une colonne d'entrée. Cette option ne s'applique qu'aux colonnes d'entrée.

La valeur par défaut est utilisée si aucune valeur n'est fournie par la requête SQL. Cette option n'est pas obligatoire.

FOREIGN_KEY

Indique que ce pseudonyme est un pseudonyme enfant et spécifie le pseudonyme parent correspondant.

Un pseudonyme peut avoir au maximum une option de colonne FOREIGN_KEY. La valeur de cette option est sensible à la casse. La colonne désignée par l'option FOREIGN_KEY contient une clé qui est générée par l'encapsuleur. La valeur de la colonne ne peut pas être extraite d'une requête SELECT et l'option XPATH ne doit pas être indiquée. La colonne ne peut être utilisée que pour associer des pseudonymes parent et des pseudonymes enfant. Une instruction CREATE NICKNAME avec une option FOREIGN_KEY échoue si le pseudonyme parent a un nom de schéma différent. A moins que le pseudonyme auquel il est fait référence dans une clause FOREIGN_KEY ait été explicitement défini en minuscules ou dans une casse mixte en étant placé entre apostrophes dans l'instruction CREATE NICKNAME correspondante, vous devez spécifier le pseudonyme en majuscules lorsque vous faites référence à ce pseudonyme dans la clause FOREIGN_KEY.

Les colonnes de clé externe doivent être désignées comme FOR BIT DATA et NOT NULL.

INPUT_MODE

Indique le mode entrée d'une colonne. Les valeurs valides sont CONFIG ou FILE_INPUT. L'encapsuleur transmet la valeur spécifiée au démon Script.

CONFIG

La valeur est traitée comme un paramètre configurable.

FILE_INPUT

Un fichier est créé et stocke la valeur. Le nom de fichier est transmis comme argument de la ligne de commande.

POSITION

Valeur entière pour les paramètres positionnels. La séquence de position commence à 1. Cette option s'applique uniquement aux colonnes d'entrée.

Si cette valeur positionnelle est définie sur un entier, alors cette entrée doit être dans cette position dans la ligne de commande. Si cette option est définie, le commutateur est inséré à l'emplacement approprié lorsque la requête est exécutée. Si POSITION est défini sur -1, l'option est ajoutée comme la dernière option de ligne de commande. Par exemple, si une valeur de colonne doit être à la fin de la ligne de commande et qu'il n'y a pas d'option SWITCH, si vous définissez la valeur de POSITION sur -1, la valeur est ajoutée à la fin de la ligne de commande. Les valeurs entières de POSITION ne peuvent pas être dupliquées dans un pseudonyme. Cette option n'est pas obligatoire.

PRIMARY_KEY

Indique que ce pseudonyme est un pseudonyme parent. Le type de données de colonne doit être VARCHAR(16). Un pseudonyme peut avoir au maximum une option de colonne PRIMARY_KEY. La seule valeur valide est Y.

La colonne qui est désignée comme PRIMARY_KEY contient une clé qui est générée par l'encapsuleur. La valeur de la colonne ne peut pas être extraite d'une requête SELECT et l'option XPATH ne doit pas être indiquée. La colonne ne peut être utilisée que pour associer des pseudonymes parent et des pseudonymes enfant. Les colonnes de clé primaire doivent être désignées comme FOR BIT DATA et NOT NULL.

SWITCH

Chaîne de caractères indiquant un paramètre pour le script sur la ligne de commande. Cette option ne s'applique qu'aux colonnes d'entrée.

Sur la ligne de commande, la valeur de cette option précède la valeur de la colonne fournie par WSSCRIPT.ARGS ou la valeur par défaut, s'il y a lieu. Si la valeur du commutateur est une chaîne vide et qu'une valeur par défaut existe pour la colonne, la valeur par défaut est ajoutée sans les informations SWITCH lorsque la ligne de commande est générée. Si aucune valeur par défaut n'est fournie et qu'aucune valeur pour la colonne n'est fournie par la requête SQL, alors cette colonne d'entrée est ignorée lorsque la ligne de commande est générée. Cette option est obligatoire pour une colonne d'entrée.

SWITCH_ONLY

Permet l'utilisation de commutateurs sans argument de ligne de commande.

Si l'option SWITCH_ONLY est indiquée avec une valeur de Y, alors les valeurs en entrée valides sont Y ou N. Pour une valeur en entrée de Y, seul le commutateur est ajouté à la ligne de commande. Pour une valeur d'entrée N, aucune valeur n'est ajoutée à la ligne de commande.

VALID_VALUES

Ensemble de valeurs valides séparées par des points-virgules pour une colonne.

XPATH

Indique l'expression XPath dans le document XML qui contient les données correspondant à cette colonne. L'encapsuleur de script évalue l'expression XPath une fois que l'instruction CREATE NICKNAME a appliqué cette expression XPath provenant de cette option de pseudonyme XPATH. Si vous appliquez une requête à un nom de colonne ayant une référence de balise XPATH configurée de manière incorrecte, par exemple une écriture en majuscules/minuscules incorrecte, votre requête renvoie des valeurs NULL dans cette colonne pour toutes les lignes renvoyées.

Requêtes SQL avec l'encapsuleur de script

Les requêtes SQL effectuées à l'aide de l'encapsuleur de script utilisent la fonction personnalisée pour transmettre les valeurs de paramètre en entrée pour le script.

Toute instruction SELECT qui transmet des valeurs de paramètre à un script à l'aide de l'encapsuleur de script doit contenir au moins un prédicat avec une fonction personnalisée pour recevoir les valeurs de paramètre en entrée pour le script.

Pseudonymes racine

Par exemple, l'instruction suivante crée un pseudonyme racine pour un script nommé myscript :

```
CREATE NICKNAME customers (  
  argle double OPTIONS(SWITCH '-argle', POSITION 1, DEFAULT 1.0),  
  argfile CLOB() OPTIONS(SWITCH '-file', INPUT_MODE 'FILE_INPUT', POSITION 2),  
  argpos varchar() OPTIONS(SWITCH ' ', POSITION 3),  
  id varchar(10) OPTIONS(XPATH './@id'),  
  name varchar OPTIONS(XPATH '/name'))  
FOR SERVER serveur_script  
  OPTIONS(DATASOURCE 'myscript', XPATH 'doc/customer',  
    TIMEOUT '300', VALIDATE 'YES');
```

Les instructions de la fonction personnalisée sont les suivantes :

```
CREATE FUNCTION wsscript.args (varchar(), varchar())  
  RETURNS INTEGER AS TEMPLATE  
  DETERMINISTIC NO EXTERNAL ACTION;
```

```
CREATE FUNCTION wsscript.args (date(), date())  
  RETURNS INTEGER AS TEMPLATE  
  DETERMINISTIC NO EXTERNAL ACTION;
```

```
CREATE FUNCTION wsscript.args (integer(), integer())  
  RETURNS INTEGER AS TEMPLATE  
  DETERMINISTIC NO EXTERNAL ACTION;
```

```
CREATE FUNCTION wsscript.args (CLOB(), CLOB())  
  RETURNS INTEGER AS TEMPLATE  
  DETERMINISTIC NO EXTERNAL ACTION;
```

```
CREATE FUNCTION wsscript.args (double(), double())  
  RETURNS INTEGER AS TEMPLATE  
  DETERMINISTIC NO EXTERNAL ACTION;
```

Le fichier de configuration indique les paramètres de configuration suivants :

```
SCRIPT_OUT_DIR_PATH=C:\temp  
myscript=C:\perl\bin\perl myscript.pl -model
```

Pour exécuter une requête et envoyer le contenu de la table t1.bigdata au démon et au fichier local C:\temp\f12345, émettez la requête suivante et la commande wsscript.args :

```
SELECT id, name FROM customers, t1  
  WHERE wsscript.args (customers.argfile, t1.bigdata) = 1
```

La requête précédente produit la ligne de commande suivante :

```
C:\perl\bin\perl myscript.pl -model -argle 1.0 -file C:\temp\f12345
```

Pour exécuter une requête qui utilise les valeurs implicites pour tous les paramètres, appliquez une requête au pseudonyme sans aucun prédicat. Pour

éviter des problèmes de sortie excessive, incluez l'option STREAMING dans le pseudonyme.

Pseudonymes enfant

L'encapsuleur de script mappe l'ensemble de résultats XML du script à des pseudonymes liés par une relation parent-enfant. Pour extraire les données d'un pseudonyme enfant, associez le pseudonyme enfant à son pseudonyme parent jusqu'à la racine. Les instructions SELECT qui référencent un pseudonyme enfant doivent être jointes avec le pseudonyme parent du pseudonyme enfant à l'aide de colonnes de clés primaires et externes.

La requête suivante affiche les noms et les montants du client pour chaque commande (order) de chaque client :

```
SELECT c.name, o.amount FROM customers c, orders o
WHERE c.cid=o.cid
AND wsscript.args (customers.argfile, t1.bigdata) = 1
AND wsscript.args (customers.argpos, VARCHAR('test1')) = 1
AND wsscript.args (customers.argle, FLOAT('3.5')) = 1
```

Indiquez l'instruction c.cid=o.cid de jointure pour désigner la relation parent-enfant entre les pseudonymes de client et les pseudonymes d'ordre. Si vous joignez un pseudonyme enfant à lui-même, un message d'erreur est renvoyé.

Optimisation des performances de l'encapsuleur de script

L'emplacement du démon Script peut avoir un effet sur les performances d'interrogation.

Pour améliorer les performances de communication sur le réseau, utilisez un serveur de script distinct pour le démon Script. Placez le serveur fédéré et le serveur de script sur des serveurs séparés. En outre, placez le démon Script sur le serveur de script.

Configuration de l'accès aux sources de données Sybase

Pour configurer un serveur fédéré pour accéder aux sources de données Sybase, vous devez fournir au serveur fédéré des informations sur les sources de données et les objets auxquels vous voulez accéder.

Avant de commencer

- Le logiciel Sybase Client SDK doit être installé sur un serveur qui fera office de serveur fédéré. Lorsque vous installez le client Sybase sous Windows, vous devez indiquer l'option **Complète** ou **Personnalisée**. Si vous indiquez l'option personnalisée, vous devez indiquer l'option **XA Interface Library for ASE Distributed Transaction Manager**.
- IBM InfoSphere Federation Server doit être installé sur un serveur qui fait office de serveur fédéré.
- Vérifiez la configuration du serveur fédéré.
- Vérifiez le paramètre fédéré pour vérifier que la fédération est activée.

Vous pouvez configurer un serveur fédéré pour accéder aux données qui sont stockées dans les sources de données Sybase à l'aide du Centre de contrôle DB2 ou en émettant des instructions SQL sur la ligne de commande DB2. Le Centre de contrôle DB2 comporte un assistant pour vous guider à travers les étapes nécessaires pour configurer les objets fédérés obligatoires.

Procédure

Pour configurer l'accès aux sources de données Sybase :

1. Définissez les variables d'environnement Sybase.
2. Installez et testez le fichier de configuration client Sybase à l'aide de l'une des méthodes suivantes selon votre système d'exploitation :
 - Installez et testez le fichier de configuration client Sybase (Linux, UNIX).
 - Installez et testez le fichier de configuration client (Windows).
3. Enregistrez l'encapsuleur.
4. Enregistrez la définition de serveur.
5. Créez les mappages utilisateur.
6. Testez la connexion au serveur Sybase.
7. Enregistrez des pseudonymes pour les tables et vues Sybase.

Prise en charge de l'encapsuleur Sybase pour Adaptive Server Enterprise (ASE)

L'encapsuleur Sybase prend en charge Sybase Adaptive Server Enterprise (ASE) 15.0, ainsi qu'ASE 12.5 et ASE 12.0.

Clients pris en charge pour Sybase ASE 15.0

Vous pouvez vous connecter à ASE 15.0 en utilisant l'encapsuleur Sybase avec le client Sybase, version 12.5.1 ou ultérieure.

Si vous utilisez SDK (Software Developer Kit : kit de développement de logiciels) version 12.5.1 comme client Sybase, IBM vous recommande d'installer Electronic Software Distribution (ESD) #12 ou une version ultérieure pour le kit de développement de logiciels (SDK) sur le serveur fédéré qui est associé à l'encapsuleur Sybase. Si vous utilisez SDK version 15.0 comme client Sybase, IBM vous recommande d'installer ESD #3 ou une version ultérieure. Si vous n'installez pas ESD, vous risquez de rencontrer une erreur imprévue en utilisant l'encapsuleur Sybase.

Mise à jour des bibliothèques Sybase pour Sybase client version 15.0

UNIX : si vous utilisez Sybase client version 15.0, vous pouvez exécuter le script Insyblibs Sybase pour mettre à jour les noms de bibliothèque Sybase afin de préserver la cohérence entre les versions prises en charge des fichiers de bibliothèque Sybase. Le script Insyblibs crée des liens symboliques entre les nouveaux noms de bibliothèque et les noms de bibliothèque antérieurs, ce qui permet aux applications de la pré-version 15.0 d'utiliser des bibliothèques renommées.

Windows : si vous utilisez Sybase client version 15.0, vous pouvez exécuter le fichier copylibs.bat pour copier les fichiers *.dll obligatoires, qui permettent aux applications de la pré-version 15.0 d'utiliser des bibliothèques renommées.

Erreur de script Insyblibs

Le script Insyblibs actuel contient un problème. Lorsque vous exécutez 'Insyblibs create', le message d'erreur suivant se produit :

```
"libsyb*.s[o: Fichier ou répertoire inexistant"
```

Comme solution palliative à cet incident, vous pouvez supprimer le | (caractère de barre) entre [] (crochets) à la ligne 34 du script. Sybase est au courant de ce problème. Pour plus d'informations, consultez l'assistance technique Sybase.

Types de données non pris en charge

Vous ne pouvez pas créer de pseudonymes pour les objets de source de données qui contiennent des types de données non pris en charge. L'encapsuleur Sybase ne prend pas en charge les types de données suivants, qui ont été introduits dans ASE version 12.5.1 :

- DATE
- TIME

L'encapsuleur Sybase ne prend pas en charge les types de données suivants, qui ont été introduits dans ASE version 15.0 :

- BIGINT
- LONGSYSNAME
- UNITEXT
- UNSIGNED BIGINT
- UNSIGNED INT
- UNSIGNED SMALLINT

Définition des variables d'environnement Sybase

Les variables d'environnement Sybase doivent être définies dans le fichier db2dj.ini sur le serveur fédéré.

Restrictions

Lisez les restrictions pour le fichier db2dj.ini.

Le fichier db2dj.ini contient des informations de configuration sur le logiciel Sybase Open Client installé sur votre serveur fédéré.

Il y a des variables d'environnement obligatoires et facultatives pour les sources de données Sybase.

Si vous avez installé Sybase Open Client avant d'installer l'encapsuleur Sybase, les variables d'environnement Sybase obligatoires sont définies dans le fichier db2dj.ini.

Vous devez définir les variables d'environnement à l'aide des étapes de cette tâche si vous n'avez pas installé Sybase Open Client avant d'installer l'encapsuleur Sybase ou que vous voulez définir l'une des variables d'environnement facultatives.

Procédure

Pour définir les variables d'environnement Sybase :

1. Utilisez l'une des méthodes suivantes :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .

Méthode	Description
Définissez automatiquement les variables d'environnement.	Exécutez l'assistant d'installation IBM InfoSphere Federation Server. Suivez les instructions de l'assistant. Important : Définissez les variables d'environnement obligatoires en exécutant l'assistant d'installation. Les variables d'environnement facultatives doivent être définies manuellement.
Définissez manuellement les variables d'environnement.	Modifiez le fichier db2dj.ini : Le fichier db2dj.ini se trouve dans le répertoire indiqué par la variable de registre DB2 DB2_DJ_INI. Si la variable DB2_DJ_INI n'est pas définie, le fichier db2dj.ini figure dans un des chemins par défaut suivants selon le système d'exploitation : <ul style="list-style-type: none"> Sous Linux et UNIX : <i>instancehome</i>/sqlib/cfg/db2dj.ini. <i>instancehome</i> Le répertoire de base du propriétaire d'instance. Sous Windows : %DB2PATH%\cfg\db2dj.ini %DB2PATH% Le répertoire dans lequel le système de la base de données DB2 est installé, par exemple C:\Program Files\IBM\sqlib. <p>Si le fichier n'existe pas, vous pouvez créer un fichier portant le nom db2dj.ini à l'aide de tout éditeur de texte.</p> <p>Dans le fichier db2dj.ini, vous devez indiquer le chemin qualifié complet comme valeur de la variable d'environnement SYBASE, sinon vous rencontrerez des erreurs. Par exemple :</p> <pre>SYBASE=/opt/sybase SYBASE_OCS=OCS-12_5</pre>

- Sous Linux et UNIX, mettez à jour le fichier .profile qui est sur l'instance de base de données fédérée. Indiquez les informations sur les variables d'environnement Sybase que vous avez ajoutées au fichier db2dj.ini. Sous Windows, le fichier est mis à jour lorsque vous installez Sybase Open Client. Emettez les commandes suivantes pour mettre à jour le fichier .profile :

```
export SYBASE=répertoire_base_sybase
export SYBASE_OCS=OCS-version_release
export PATH=${SYBASE}/bin:$PATH
```
- A partir du répertoire de base, exécutez le fichier .profile qui est sur l'instance de base de données fédérée. Emettez la commande suivante pour exécuter le fichier .profile :

```
. .profile
```
- Sur certains systèmes d'exploitation, tels que Linux, vous devez ajouter le chemin de la bibliothèque Sybase client à la variable DB2LIBPATH db2set, par exemple :

```
db2set DB2LIBPATH=/opt/sybase125/OCS-12_5/lib
```
- Pour être sûr que les variables d'environnement sont définies sur le serveur fédéré, recyclez l'instance de base de données fédérée avec les commandes suivantes :

```
db2stop
db2start
```

Une fois cette tâche terminée, vous devez enregistrer l'encapsuleur.

Variables d'environnement Sybase

Il y a des variables d'environnement obligatoires et facultatives pour les sources de données Sybase. Ces variables sont définies dans le fichier db2dj.ini.

Les variables d'environnement suivantes sont valides pour Sybase :

- SYBASE
- SYBASE_OCS
- SYBASE_CHARSET (facultative)

Description des variables

SYBASE

Indique le chemin du répertoire dans lequel Sybase Open Client est installé. Indique le chemin qualifié complet de cette variable d'environnement.

Par exemple, si la version Sybase Open Client 12.5 est installée dans le chemin de répertoire D:\djxclient\sybase\V125, indiquez la variable d'environnement SYBASE suivante :

```
SYBASE=D:\djxclient\sybase\V125
```

Si Sybase Open Client Version 12.0 est installé dans le chemin de répertoire D:\djxclient\sybase\V12, indiquez la variable d'environnement SYBASE suivante :

```
SYBASE=D:\djxclient\sybase\V12
```

SYBASE_OCS

Indique le répertoire, la version et l'édition du logiciel Sybase Open Client installé. N'indiquez pas le chemin qualifié complet lorsque vous indiquez cette variable d'environnement.

```
SYBASE_OCS=OCS-version_release
```

Par exemple, si Sybase Open Client Version 12.0 est installé dans le chemin de répertoire D:\djxclient\sybase\V12\OCS-12_0, indiquez la valeur suivante pour la variable d'environnement SYBASE_OCS :

```
SYBASE_OCS=OCS-12_0
```

Si Sybase Open Client Version 12.5 est installé dans le chemin de répertoire D:\djxclient\sybase\V125\OCS-12_5, indiquez la valeur suivante pour la variable d'environnement SYBASE_OCS :

```
SYBASE_OCS=OCS-12_5
```

SYBASE_CHARSET

Indique le nom du jeu de caractères que vous voulez utiliser. Définissez la variable d'environnement SYBASE_CHARSET sur le jeu de codes que vous indiquez dans le paramètre **CODESET** du serveur fédéré. Une liste des noms de jeu de caractères valides figure dans le répertoire \$SYBASE\charsets.

Les jeux de codes sont pas indiqués de la même manière dans le paramètre **CODESET** et dans la variable d'environnement SYBASE_CHARSET. Par exemple, si vous définissez le paramètre **CODESET** sur le format de transformation Unicode 8 bits, UTF-8, indiquez UTF8 dans la variable d'environnement SYBASE_CHARSET :

```
SYBASE_CHARSET=utf8
```


Si vous ne définissez pas la variable d'environnement SYBASE_CHARSET, l'encapsuleur utilise le jeu de caractères Sybase qui correspond à celui qui est indiqué dans la page de codes de la base de données fédérée. S'il n'y a pas de jeu de caractères Sybase correspondant, l'encapsuleur utilise le jeu de caractères iso_1.

Paramétrage et test du fichier de configuration client Sybase (Windows)

Le fichier de configuration client Sybase est utilisé pour se connecter aux bases de données Sybase à l'aide des bibliothèques de client qui sont installées sur le serveur fédéré.

Avant de commencer

Sybase Client SDK doit être installé sur le serveur fédéré.

A propos de cette tâche

Le fichier de configuration client indique l'emplacement de chaque serveur SQL Sybase et de chaque instance Adaptive Server Enterprise, ainsi que le type de connexion (protocole) du serveur de la base.

Vous devez configurer un fichier de configuration client sur chaque instance du serveur fédéré qui sera utilisé pour se connecter à Sybase.

Procédure

Pour configurer et tester le fichier de configuration client Sybase sur les serveurs fédérés qui fonctionnent sous Windows :

1. Installez le fichier de configuration client à l'aide de l'utilitaire fourni avec Sybase Open Client. Voir la documentation Sybase pour de plus amples informations sur l'utilisation de cet utilitaire.

Le fichier de configuration client est créé dans le répertoire %SYBASE%\ini. Le nom du fichier est sql.ini.

2. Testez la connexion pour vérifier que Sybase Open Client peut se connecter au serveur Sybase.

Utilisez un utilitaire d'interrogation Sybase approprié, tel qu'isql, pour tester la connexion.

Par exemple, si Sybase Open Client est installé dans le chemin de répertoire D :\dxclient\sybase\V125, vous pouvez émettre les commandes suivantes depuis une invite de commande :

```
cd D:\dxclient\sybase\V125\OCS-12_5\bin
isql -Ssybnode -Umary
```

Vous pouvez aussi émettre la commande suivante depuis une invite de commande :

```
%SYBASE%\%SYBASE_OCS%\bin\isql -Ssybnode -Umary
```

Indication du chemin du fichier d'interfaces

: Si vous voulez utiliser un fichier d'interfaces autre que le fichier par défaut, utilisez l'option de serveur IFILE pour indiquer le chemin. L'encapsuleur Sybase recherche le fichier d'interfaces dans les espaces suivants, dans l'ordre indiqué :

- a. option de serveur IFILE

- b. %DB2PATH%\interfaces
- c. %SYBASE%\ini\sql.ini

Une fois cette tâche terminée, vous pouvez définir les variables d'environnement.

Configuration et test du fichier de configuration client Sybase (UNIX)

Le fichier de configuration client Sybase est utilisé pour se connecter aux bases de données Sybase à l'aide des bibliothèques de client qui sont installées sur le serveur fédéré.

Avant de commencer

Sybase Client SDK doit être installé sur le serveur fédéré.

Le fichier de configuration client indique l'emplacement de chaque serveur SQL Sybase et de chaque instance Adaptive Server Enterprise, ainsi que le type de connexion (protocole) du serveur de la base.

Vous devez configurer un fichier de configuration client sur chaque instance du serveur fédéré qui sera utilisé pour se connecter à Sybase.

Procédure

Pour configurer et tester le fichier de configuration client Sybase sur des serveurs fédérés qui fonctionnent sous UNIX :

1. Installez le fichier de configuration client à l'aide de l'utilitaire fourni avec Sybase Open Client.

Le fichier de configuration client est créé dans le répertoire \$SYBASE. Le nom par défaut du fichier est interfaces. Voir la documentation Sybase pour de plus amples informations sur l'utilisation de cet utilitaire.

2. Testez la connexion pour vérifier que Sybase Open Client peut se connecter au serveur Sybase.

Utilisez un utilitaire d'interrogation Sybase approprié, tel qu'isql, pour tester la connexion.

Par exemple, si Sybase Open Client est installé dans le chemin de répertoire /opt/djxclient/sybase/V125, vous pouvez émettre la commande suivante depuis une invite UNIX :

```
cd /opt/djxclient/sybase/V125/OCS-12_5
isql -Ssybnode -Umary
```

Vous pouvez aussi émettre la commande suivante depuis une invite UNIX :

```
$SYBASE/$SYBASE_OCS/bin/isql -Ssybnode -Umary
```

Indication du chemin du fichier d'interfaces

: Si vous voulez utiliser un fichier d'interfaces autre que le fichier par défaut, utilisez l'option de serveur IFILE pour indiquer le chemin. L'encapsuleur Sybase recherche le fichier d'interfaces dans les espaces suivants, dans l'ordre indiqué :

- a. option de serveur IFILE
- b. sqllib/interfaces
- c. \$SYBASE/interfaces

Une fois cette tâche terminée, vous pouvez définir les variables d'environnement.

Enregistrement de l'encapsuleur Sybase

Vous devez enregistrer un encapsuleur pour accéder aux sources de données Sybase. Les serveurs fédérés utilisent des encapsuleurs pour dialoguer avec les sources de données et en extraire des données. Les encapsuleurs sont mis en oeuvre comme un ensemble de fichiers de bibliothèque.

Procédure

Pour enregistrer l'encapsuleur Sybase :

Utilisez l'une des méthodes suivantes :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .
Emettez l'instruction CREATE WRAPPER et indiquez le nom par défaut de l'encapsuleur Sybase.	<p>Par exemple :</p> <pre>CREATE WRAPPER CTLIB</pre> <p>A faire : Lorsque vous enregistrez l'encapsuleur à l'aide du nom par défaut, CTLIB, le serveur fédéré utilise automatiquement la bibliothèque de l'encapsuleur Sybase appropriée au système d'exploitation sous lequel votre serveur fédéré fonctionne.</p> <p>En cas de conflit entre le nom d'encapsuleur par défaut et un nom d'encapsuleur existant dans la base de données fédérée, vous pouvez remplacer le nom d'encapsuleur par défaut par un nom que vous choisissez. Lorsque vous utilisez un nom différent du nom par défaut, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER.</p> <p>Par exemple, pour enregistrer un encapsuleur avec le nom encapsuleur_sybase sur un serveur fédéré qui fonctionne sous AIX, émettez l'instruction suivante :</p> <pre>CREATE WRAPPER encapsuleur_sybase LIBRARY 'libdb2ctlib.a';</pre> <p>Le fichier de bibliothèque de l'encapsuleur que vous indiquez dépend du système d'exploitation du serveur fédéré.</p>

Une fois cette tâche terminée, vous pouvez enregistrer la définition de serveur.

Fichiers de bibliothèque de l'encapsuleur Sybase

Les fichiers de bibliothèque de l'encapsuleur Sybase sont ajoutés au serveur fédéré lorsque vous installez l'encapsuleur.

Lorsque vous installez l'encapsuleur Sybase, trois fichiers de bibliothèque sont ajoutés dans le chemin de répertoire par défaut. Par exemple, si le serveur fédéré fonctionne sous AIX, les fichiers de bibliothèque de l'encapsuleur qui sont ajoutés dans le chemin de répertoire sont libdb2ctlib.a, libdb2ctlibF.a et libdb2ctlibU.a. Les autres fichiers de bibliothèque de l'encapsuleur sont utilisés sur le plan interne par l'encapsuleur Sybase.

Si vous n'utilisez pas le nom d'encapsuleur par défaut lorsque vous enregistrez l'encapsuleur, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER et indiquer le nom de fichier de bibliothèque par défaut.

Les chemins des répertoires par défaut et les noms de fichier de bibliothèque de l'encapsuleur par défaut sont indiqués dans la table suivante.

Tableau 36. Emplacements et noms de fichier de la bibliothèque de l'encapsuleur Sybase

Système d'exploitation	Chemin de répertoire	Nom de fichier de bibliothèque
AIX	/usr/opt/chemin_installation/lib32/ /usr/opt/chemin_installation/lib64/	libdb2ctlib.a
Linux	/opt/IBM/db2/chemin_installation/lib32 /opt/IBM/db2/chemin_installation/lib64	libdb2ctlib.so
Solaris	/opt/IBM/db2/chemin_installation/lib32 /opt/IBM/db2/chemin_installation/lib64	libdb2ctlib.so
Windows	%DB2PATH%\bin	db2ctlib.dll

chemin_installation est le chemin du répertoire dans lequel IBM InfoSphere Federation Server est installé sous UNIX ou Linux.

Enregistrement des définitions de serveur pour une source de données Sybase

Vous devez enregistrer chaque serveur Sybase auquel vous voulez accéder dans la base de données fédérée.

Procédure

Pour enregistrer une définition de serveur pour une source de données Sybase :

1. Recherchez le nom du noeud dans le fichier interfaces de Sybase. Les exemples suivants montrent les entrées des fichiers d'interfaces pour des serveurs fédérés qui fonctionnent sous UNIX ou Windows :

UNIX :

```
sybase125
query tcp ether anaconda 4100
```

Windows :

```
[sybase125]
query=TCP,anaconda,4100
```

- La première ligne de chaque exemple est le nom du noeud, par exemple sybase125.
- La deuxième ligne indique le type de connexion, le nom d'hôte et le numéro de port. Dans cet exemple, TCP indique qu'il s'agit d'une connexion TCP/IP, anaconda est le nom d'hôte et 4100 est le numéro de port. .

2. Pour créer la définition de serveur, utilisez une des méthodes suivantes :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .
Emettez l'instruction CREATE SERVER.	<p>Par exemple :</p> <pre>CREATE SERVER <i>nom_définition_serveur</i> TYPE SYBASE VERSION <i>numéro_version_client_Sybase</i> WRAPPER <i>nom_encapsuleur</i> OPTIONS (NODE '<i>nom_noeud</i>', DBNAME '<i>nom_base_de_données</i>');</pre> <p>Bien que les variables '<i>nom_noeud</i>' et '<i>nom_base_de_données</i>' soient indiquées comme options dans l'instruction CREATE SERVER, ces options doivent être indiquées obligatoirement pour les sources de données Sybase.</p> <p>Important : Si vous n'avez pas renommé le fichier sql.ini en interfaces lorsque vous avez configuré le fichier de configuration client Sybase, vous devez inclure l'option de serveur IFILE lorsque vous enregistrez la définition de serveur.</p> <p>Une fois la définition de serveur enregistrée, utilisez l'instruction ALTER SERVER pour ajouter ou déplacer des options de serveur.</p>

Une fois cette tâche terminée, vous pouvez créer des mappages utilisateur.

Instruction CREATE SERVER - exemples pour l'encapsuleur Sybase

Utiliser l'instruction CREATE SERVER pour enregistrer des définitions de serveur pour l'encapsuleur Sybase. Cette rubrique fournit un exemple complet avec les paramètres obligatoires et un exemple avec des options de serveur supplémentaires.

L'exemple suivant vous montre comment enregistrer une définition de serveur pour un encapsuleur Sybase en émettant l'instruction CREATE SERVER :

```
CREATE SERVER SYBSERVER TYPE SYBASE VERSION 12.0 WRAPPER CTLIB
  OPTIONS (NODE 'sybnode', DBNAME 'sybdb');
```

SYBSERVER

Nom que vous attribuez au serveur Sybase. Les doublons de noms de définition de serveur ne sont pas autorisés.

TYPE SYBASE

Indique le type de serveur de source de données auquel vous configurez l'accès. Pour l'encapsuleur CTLIB, le type de serveur doit être SYBASE.

VERSION 12.0

La version du logiciel du client de base de données Sybase qui est utilisée pour la connexion fédérée.

WRAPPER CTLIB

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

NODE 'sybnode'

Le nom du noeud sur lequel le serveur Sybase se trouve. Obtenez le nom du noeud à partir du fichier interfaces. Cette valeur fait la distinction entre les majuscules et les minuscules.

Bien que le nom du noeud soit indiqué comme option dans l'instruction CREATE SERVER, il doit être indiqué obligatoirement pour les sources de données Sybase.

DBNAME 'sybdb'

Le nom de la base de données Sybase à laquelle vous voulez accéder. Cette valeur fait la distinction entre les majuscules et les minuscules.

Bien que le nom de la base de données soit indiqué comme option dans l'instruction CREATE SERVER, il doit être indiqué obligatoirement pour les sources de données Sybase.

Options de serveur

Lorsque vous créez une définition de serveur, vous pouvez indiquer des options de serveur supplémentaires dans l'instruction CREATE SERVER. Les options de serveur peuvent être des options de serveur générales et des options de serveur spécifiques à Sybase.

Option de serveur IFILE

Si vous n'utilisez pas le fichier d'interfaces par défaut, vous devez créer un fichier interfaces et inclure l'option de serveur IFILE dans l'instruction CREATE SERVER. Le fichier d'interfaces par défaut est \$SYBASE/interfaces pour Linux et UNIX et %SYBASE%\ini\sql.ini pour Windows.

La valeur que vous indiquez pour l'option de serveur IFILE est le chemin d'accès complet et le nom du fichier Sybase Open Client sql.ini.

L'exemple suivant montre comment utiliser l'option de serveur IFILE lorsque vous enregistrez une définition de serveur sur un serveur fédéré qui fonctionne sous Windows :

```
CREATE SERVER SYBSERVER TYPE SYBASE VERSION 12.0 WRAPPER CTLIB
  OPTIONS (NODE 'sybnode', DBNAME 'sybdb',
  IFILE 'C:\Sybase\ini\sql.ini');
```

Option de serveur TIMEOUT

L'option de serveur TIMEOUT définit le nombre de secondes pendant lequel l'encapsuleur attend une réponse du serveur Sybase. Utilisez l'option TIMEOUT pour éviter les blocages de transactions.

L'exemple suivant montre comment indiquer l'option de serveur TIMEOUT lorsque vous enregistrez une définition de serveur :

```
CREATE SERVER SYBSERVER TYPE SYBASE VERSION 12.0 WRAPPER CTLIB
  OPTIONS (NODE 'sybnode', DBNAME 'sybdb',
  TIMEOUT '60');
```

Les options de serveur supplémentaires spécifiques à Sybase sont :

- LOGIN_TIMEOUT
- PACKET_SIZE

Création des mappages utilisateur pour une source de données Sybase

Lorsque vous essayez d'accéder à un serveur Sybase, le serveur fédéré établit une connexion avec le serveur Sybase à l'aide d'un ID utilisateur et d'un mot de passe

qui sont valides pour cette source de données. Vous devez définir une association (un mappage utilisateur) entre chaque ID utilisateur et mot de passe de serveur fédéré et l'ID utilisateur et le mot de passe de la source de données correspondante.

Créez un mappage utilisateur pour chaque ID utilisateur qui accédera au système fédéré pour envoyer des demandes réparties à la source de données Sybase.

Procédure

Créez des mappages utilisateur pour une source de données Sybase :

Emettez une instruction CREATE USER MAPPING.

Par exemple :

```
CREATE USER MAPPING FOR IDutilisateur_local SERVER nom_définition_serveur
    OPTIONS (REMOTE_AUTHID 'IDutilisateur_distant', REMOTE_PASSWORD
'motdepasse_distant');
```

Une fois cette tâche terminée, testez la connexion au serveur Sybase.

Instruction CREATE USER MAPPING - exemples pour l'encapsuleur Sybase

Utiliser l'instruction CREATE USER MAPPING pour mapper un ID utilisateur du serveur fédéré à un ID utilisateur et à un mot de passe du serveur Sybase. Cette rubrique fournit un exemple complet avec les paramètres obligatoires et un exemple qui vous montre comment utiliser le registre spécial USER DB2 avec l'instruction CREATE USER MAPPING.

L'exemple suivant montre comment mapper un ID utilisateur du serveur fédéré à un ID utilisateur et à un mot de passe du serveur Sybase :

```
CREATE USER MAPPING FOR maria SERVER SYBSERVER
    OPTIONS (REMOTE_AUTHID 'mary', REMOTE_PASSWORD 'day2night');
```

maria Indique l'ID utilisateur local que vous mappez à un ID utilisateur qui est défini sur le serveur Sybase.

SERVER SYBSERVER

Indique le nom de définition de serveur que vous avez enregistré dans l'instruction CREATE SERVER pour le serveur Sybase.

REMOTE_AUTHID 'mary'

Indique l'ID utilisateur sur le serveur Sybase auquel vous mappez *maria*. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_ID sur 'U' ou 'L' dans l'instruction CREATE SERVER.

Bien que l'ID utilisateur distant soit indiqué comme option dans l'instruction CREATE SERVER, il doit être indiqué obligatoirement pour les sources de données Sybase.

REMOTE_PASSWORD 'day2night'

Indique le mot de passe qui est associé à '*mary*'. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_PW sur 'U' ou 'L' dans l'instruction CREATE SERVER.

Bien que le mot de passe distant soit indiqué comme option dans l'instruction CREATE SERVER, il doit être indiqué obligatoirement pour les sources de données Sybase.

registre spécial USER DB2

Vous pouvez utiliser le registre spécial USER DB2 pour mapper l'identificateur utilisateur de la personne qui émet l'instruction CREATE USER MAPPING à l'identificateur utilisateur de la source de données qui est indiqué dans l'option utilisateur REMOTE_AUTHID.

L'exemple suivant montre une instruction CREATE USER MAPPING qui comporte le registre spécial USER :

```
CREATE USER MAPPING FOR USER SERVER SYBSERVER
  OPTIONS (REMOTE_AUTHID 'mary', REMOTE_PASSWORD 'day2night');
```

Test de la connexion au serveur Sybase

Tester la connexion au serveur de source de données Sybase pour déterminer si le serveur fédéré est correctement configuré pour accéder aux sources de données Sybase.

Vous pouvez tester la connexion au serveur Sybase à l'aide de la définition de serveur et des mappages utilisateur que vous avez définis.

Procédure

Pour tester la connexion au serveur Sybase :

Ouvrez une session passe-système et émettez une instruction SELECT sur les tables système Sybase. Si l'instruction SELECT renvoie un comptage, votre définition de serveur et votre mappage utilisateur sont configurés correctement.

Par exemple :

```
SET PASSTHRU nom_définition_serveur
SELECT count(*) FROM dbo.sysobjects
SET PASSTHRU RESET
```

Si l'instruction SELECT renvoie une erreur, vous devez identifier et résoudre les erreurs de connexion.

Une fois cette tâche terminée, vous pouvez enregistrer des pseudonymes pour les tables et vues Sybase.

Identification et résolution des erreurs de connexion à la source de données

Une connexion de test au serveur de source de données peut renvoyer une erreur pour plusieurs raisons. Il y a des mesures que vous pouvez prendre pour déterminer pourquoi l'erreur s'est produite.

Symptôme

Une erreur est renvoyée lorsque vous essayez de vous connecter à la source de données.

Cause

Il y a plusieurs causes possibles à un incident de connexion.

Résolution de l'incident

Pour identifier et résoudre les erreurs de connexion à la source de données, recherchez les problèmes éventuels avec les éléments suivants :

- Vérifiez que la source de données est disponible.
- S'il y a lieu, vérifiez que le serveur de source de données est configuré pour les connexions entrantes.
- Vérifiez que les paramètres de votre mappage utilisateur pour les options REMOTE_AUTHID et REMOTE_PASSWORD sont valides pour les connexions à la source de données. Modifiez le mappage utilisateur ou créez un autre mappage utilisateur si nécessaire.
- S'il y a lieu, vérifiez que le logiciel client de la source de données sur le serveur fédéré est installé et configuré correctement pour se connecter à la source de données.
- Pour les sources de données ODBC, vérifiez que le pilote ODBC sur le serveur fédéré est installé et configuré correctement pour se connecter au serveur de source de données ODBC. Sur les serveurs fédérés qui fonctionnent sous Windows, utilisez l'outil Administrateur de sources de données ODBC pour vérifier le pilote. Sur les serveurs fédérés qui fonctionnent sous UNIX, consultez la documentation du fournisseur de client ODBC.
- Vérifiez que les valeurs des variables définies sur le serveur fédéré sont appropriées à la source de données. Ces variables sont les variables d'environnement système, les variables du fichier db2dj.ini et les variables du registre de profil DB2 (db2set).
- Vérifiez votre définition de serveur. Si nécessaire, supprimez la définition de serveur et recréez-la.

Enregistrement de pseudonymes pour les tables et vues Sybase

Dans chaque définition de serveur Sybase que vous enregistrez, vous devez enregistrer un pseudonyme pour chaque table ou vue à laquelle vous voulez accéder. Utilisez ces pseudonymes, au lieu des noms des objets de source de données, lorsque vous interrogez les serveurs Sybase.

Avant de commencer

Mettez à jour les statistiques sur la source de données Sybase avant d'enregistrer un pseudonyme. La base de données fédérée exploite les statistiques du catalogue de la source de données pour optimiser le traitement des requêtes. Utilisez la commande de la source de données équivalant à la commande RUNSTATS de DB2 pour mettre à jour les statistiques de la source de données.

Procédure

Pour enregistrer un pseudonyme pour des tables ou vues Sybase, utilisez une des méthodes suivantes :

Méthodes	Descriptions
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .

Méthodes	Descriptions
Emettez l'instruction CREATE NICKNAME. La longueur des pseudonymes peut aller jusqu'à 128 caractères.	Par exemple : <pre>CREATE NICKNAME <i>pseudonyme</i> FOR <i>nom_définition_serveur</i>."<i>schéma_distant</i>". "<i>table.distante</i>" ;</pre>

Lorsque vous créez le pseudonyme, le serveur fédéré interroge le catalogue de la source de données à l'aide du pseudonyme. Cette requête teste la connexion à la table ou à la vue de la source de données. Si la connexion ne fonctionne pas, vous recevez un message d'erreur.

Répétez cette étape pour chaque table ou vue Sybase pour laquelle vous voulez créer un pseudonyme.

Instruction CREATE NICKNAME - exemples pour l'encapsuleur Sybase

Utiliser l'instruction CREATE NICKNAME pour enregistrer un pseudonyme pour une table ou une vue Sybase à laquelle vous voulez accéder. Cette rubrique comporte un exemple complet avec les paramètres obligatoires.

L'exemple suivant vous montre comment enregistrer un pseudonyme pour une table ou une vue Sybase à l'aide de l'instruction CREATE NICKNAME.

```
CREATE NICKNAME SYBSALES FOR SYBSERVER."vinnie". "europa";
```

SYBSALES

Pseudonyme unique utilisé pour identifier la table ou la vue Sybase.

Important : Le pseudonyme est un nom composé de deux parties, le schéma et le pseudonyme. Si vous omettez le schéma lorsque vous enregistrez le pseudonyme, le schéma du pseudonyme sera l'identificateur utilisateur de l'utilisateur qui enregistre le pseudonyme.

```
SYBSERVER."vinnie". "europa"
```

Identificateur en trois parties pour l'objet distant :

- *SYBSERVER* est le nom de définition de serveur que vous avez affecté au serveur de la base Sybase dans l'instruction CREATE SERVER.
- *vinnie* est l'ID utilisateur du propriétaire de la table ou vue.
- *europa* est le nom de la table ou de la vue distante à laquelle vous voulez accéder.

Le serveur fédéré convertit les noms des schémas et tables Sybase en majuscules, à moins que vous ne placiez les noms entre apostrophes.

Identification et résolution des incidents de configuration de l'encapsuleur Sybase

Problèmes de chargement de la bibliothèque de l'encapsuleur Sybase

Lorsque vous créez l'encapsuleur Sybase, vous pouvez rencontrer des erreurs liées à l'installation de Sybase Open Client qui empêchent le chargement de la bibliothèque de l'encapsuleur Sybase.

Symptôme

Lorsque vous créez l'encapsuleur Sybase, l'erreur SQL suivante est émise :
SQL10013N La bibliothèque spécifiée "db2ctlibF.dll"
n'a pas pu être chargée.

Cause

Sybase XA Interface Library for ASE Distributed Transaction Manager n'a pas été installé sur le système Windows avec Sybase Open Client.

Résolution de l'incident

Réinstallez Sybase Open Client sous Windows et sélectionnez l'option d'installation **Complète** ou **Personnalisée**. Si vous sélectionnez l'option d'installation **Personnalisée**, indiquez l'option **XA Interface Library for ASE Distributed Transaction Manager**.

Variable d'environnement SYBASE absente

Si le fichier db2dj.ini n'est pas dans le bon répertoire ou qu'il est absent, vous rencontrerez une erreur SQL.

Symptôme

L'erreur SQL suivante est émise :

SQL1822N Code d'erreur imprévue "" reçu de la source de données
"nom de serveur". Le texte et les jetons associés sont "Variable SYBASE non définie".

Cause

Le fichier db2dj.ini n'a pas été trouvé ou le fichier ne contient pas la variable d'environnement SYBASE. Le fichier db2dj.ini se trouve dans le répertoire indiqué par la variable de registre DB2_DB2_DJ_INI. Si la variable DB2_DB2_DJ_INI n'est pas définie, le fichier db2dj.ini figure dans un des chemins par défaut suivants selon le système d'exploitation :

- Sous UNIX : *instancehome*/sqllib/cfg/db2dj.ini.

instancehome

Le répertoire de base du propriétaire d'instance.

- Sous Windows : %DB2PATH%\cfg\db2dj.ini

%DB2PATH%

Le répertoire dans lequel le système de la base de données DB2 est installé, par exemple C:\Program Files\IBM\sqllib.

Résolution de l'incident

Créez un fichier db2dj.ini avec les variables d'environnement Sybase obligatoires et placez-le dans le répertoire approprié à votre système d'exploitation. Vous pouvez utiliser un éditeur de texte pour créer le fichier db2dj.ini.

Nom du noeud Sybase absent

Si le fichier de configuration client Sybase n'est pas correctement configuré, le système fédéré risque de ne pas trouver le nom du noeud Sybase.

Symptôme

L'erreur SQL suivante est émise :

SQL1097N Le nom du noeud n'a pas été trouvé dans le répertoire des noeuds.

Cause

Le nom du noeud Sybase n'a pas été trouvé.

Résolution de l'incident

Pour résoudre l'incident :

- Si l'option de serveur IFILE a été indiquée, vérifiez que le nom du noeud est déclaré dans le fichier qu'il indique.
- Si le fichier d'interfaces existe dans le répertoire sqllib, vérifiez que le nom du noeud y est déclaré.
- Si l'option de serveur IFILE n'est pas indiquée et que le fichier interfaces n'existe pas dans le répertoire sqllib, vérifiez que le nom du noeud est déclaré dans le fichier %SYBASE%\ini\sql.ini sous Windows ou dans le fichier \$SYBASE/interfaces sous UNIX.

Configuration de l'accès aux sources de données du fichier structuré de la table

Vous pouvez intégrer les données qui sont dans le fichier structuré sous forme de tables avec des informations d'autres sources à l'aide d'un système fédéré.

Procédure

Pour configurer un serveur fédéré pour accéder aux sources de données de fichiers structurés sous forme de tables, vous devez fournir au serveur fédéré des informations sur les sources de données et les objets auxquels vous voulez accéder. Après avoir configuré le serveur fédéré, vous pouvez créer des requêtes et utiliser les fonctions personnalisées pour accéder aux sources de données de fichiers structurés sous forme de tables.

Fichiers structurés sous forme de tables - présentation

Un fichier structuré sous forme de tables est un fichier dont la structure normale se compose d'une série d'enregistrements ou de lignes d'informations. Chaque enregistrement contient le même nombre de zones. Les données des zones sont séparées par un délimiteur, tel qu'une virgule.

L'exemple suivant montre le contenu d'un fichier appelé DRUGDATA1.TXT. Il contient trois enregistrements, chacun avec trois zones, séparées par des virgules :

```
234,DrugnameA,Manufacturer1  
332,DrugnameB,Manufacturer2  
333,DrugnameC,Manufacturer2
```

La première zone est le numéro d'identification unique du médicament. La deuxième zone est le nom du médicament. La troisième zone est le nom du fabricant qui produit le médicament.

Le délimiteur de zone peut comporter plusieurs caractères. Un guillemet simple ne peut pas être utilisé comme délimiteur. Le délimiteur doit être cohérent sur

l'ensemble du fichier. Une valeur NULL est représentée par deux délimiteurs l'un à côté de l'autre ou par un délimiteur suivi d'un caractère de fin de ligne, si la zone NULL est la dernière de la ligne. Le délimiteur de colonne ne peut pas exister en tant que donnée valide d'une colonne.

Par exemple :

```
234,DrugnameA,Manufacturer1
332,DrugnameB,Manufacturer2
333,DrugnameC,Manufacturer2
356,,Manufacturer1
```

Attributs des fichiers structurés sous forme de tables

Les enregistrements ou lignes des fichiers structurés sous forme de tables peuvent être triés ou non triés. L'encapsuleur des fichiers structurés sous forme de tables peut effectuer des recherches plus efficaces dans les fichiers triés que dans les fichiers qui ne sont pas triés.

Si les données des fichiers structurés sous forme de tables sont triées, le tri doit s'effectuer par ordre croissant d'après la colonne de clé. Vous devez définir l'option SORTED sur Y dans la colonne de clé lorsque vous définissez les colonnes pour le pseudonyme. Sinon, l'encapsuleur traite les données des fichiers structurés sous forme de tables comme des données non triées.

Fichiers triés

DRUGDATA1.TXT contient des enregistrements triés. Le fichier est trié d'après la première zone, le numéro d'identification unique du médicament. Cette zone est la clé primaire parce qu'elle est unique pour chaque médicament. Les fichiers triés doivent être triés par ordre croissant.

```
234,DrugnameA,Manufacturer1
332,DrugnameB,Manufacturer2
333,DrugnameC,Manufacturer2
```

Fichiers non triés

DRUGDATA2.TXT contient des enregistrements non triés. Les enregistrements sont indiqués dans le fichier dans n'importe quel ordre.

```
556,DrugnameB,Manufacturer2
234,DrugnameA,Manufacturer1
721,DrugnameC,Manufacturer2
```

Encapsuleur des fichiers structurés sous forme de tables

Les données dans les fichiers structurés sous forme de tables peuvent être associées avec des données d'autres fichiers structurés sous forme de tables, avec des données relationnelles ou non relationnelles et avec des données non structurées.

A l'aide d'un encapsuleur, le serveur fédéré peut traiter des instructions SQL qui recherchent les données dans les fichiers structurés sous forme de tables comme si les données figuraient dans une table ou une vue relationnelle habituelle.

La figure suivante montre comment le serveur fédéré fonctionne avec les fichiers structurés sous forme de tables.

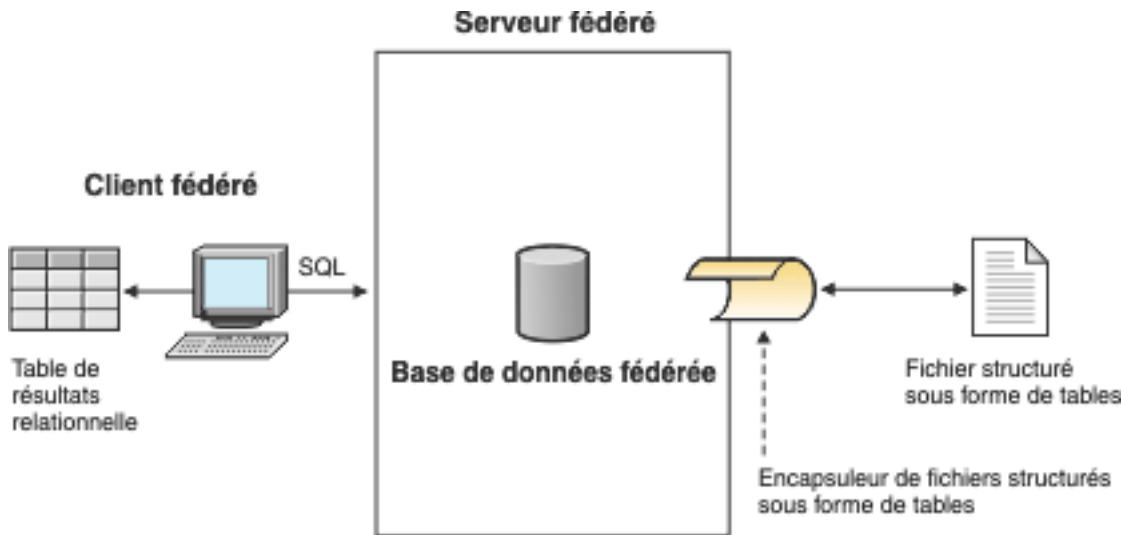


Figure 8. Fonctionnement de l'encapsuleur de fichiers à structure de table

Par exemple, le fichier structuré sous forme de tables DRUGDATA2.TXT se trouve sur un ordinateur de votre laboratoire. Les données du fichier sont :

```
556,DrugnameB,Manufacturer1
234,DrugnameA,Manufacturer2
721,DrugnameC,Manufacturer2
```

Il est fastidieux d'essayer d'interroger ces données et de les associer à d'autres tables dans d'autres sources de données que vous utilisez.

Une fois que vous avez enregistré le fichier DRUGDATA2.TXT sur le serveur fédéré, l'encapsuleur de fichier structuré sous forme de tables peut accéder aux données du fichier comme si les données étaient dans une table relationnelle.

Par exemple, vous pouvez exécuter la requête suivante :

```
SELECT * FROM DRUGDATA2 ORDER BY DCODE
```

Cette requête produit les résultats suivants :

DCODE	DRUG	MANUFACTURER
234	DrugnameA	Manufacturer2
556	DrugnameB	Manufacturer1
721	DrugnameC	Manufacturer2

Vous pouvez associer les données du fichier DRUGDATA2.TXT aux données d'autres sources de données relationnelles et non relationnelles et analyser toutes les données ensemble.

Ajout de sources de données de fichiers structurés sous forme de tables à un serveur fédéré

Pour configurer un serveur fédéré pour accéder aux sources de données de fichiers structurés sous forme de tables, vous devez fournir au serveur fédéré des informations sur les sources de données et les objets auxquels vous voulez accéder.

Avant de commencer

- La fédération doit être installée sur un serveur qui fera office de serveur fédéré.
- Il doit y avoir une base de données sur le serveur fédéré.

Vous pouvez configurer un serveur fédéré pour accéder aux données qui sont stockées dans les sources de données de fichiers structurés sous forme de tables à l'aide du Centre de contrôle ou en émettant des instructions SQL sur la ligne de commande. Le Centre de contrôle comporte un assistant pour vous guider à travers les étapes nécessaires pour configurer les objets fédérés obligatoires.

Procédure

Pour ajouter les sources de données de fichiers structurés sous forme de tables à un serveur fédéré :

1. Enregistrez l'encapsuleur de fichier structuré sous forme de tables.
2. Enregistrez la définition de serveur du fichier structuré sous forme de tables.
3. Enregistrez des pseudonymes pour les fichiers structurés sous forme de tables.

Enregistrement de l'encapsuleur de fichier structuré sous forme de tables

Vous devez enregistrer un encapsuleur pour accéder aux sources de données de fichiers structurés sous forme de tables. Les encapsuleurs sont utilisés par les serveurs fédérés pour dialoguer avec les sources de données et extraire des données des sources de données. Les encapsuleurs sont mis en oeuvre comme un ensemble de fichiers de bibliothèque.

Vous pouvez enregistrer un encapsuleur à l'aide du Centre de contrôle ou depuis la ligne de commande. Le Centre de contrôle comporte un assistant pour vous guider à travers les étapes nécessaires pour enregistrer l'encapsuleur.

Procédure

Pour enregistrer l'encapsuleur de fichier structuré sous forme de tables :

Choisissez la méthode que vous voulez utiliser pour enregistrer l'encapsuleur de fichier structuré sous forme de tables :

Méthode	Procédure.
Utilisation du Centre de contrôle	Démarrez l'assistant des objets fédérés. Cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée et cliquez sur Créer des objets fédérés .
Depuis la ligne de commande	<p>Émettez l'instruction CREATE WRAPPER.</p> <pre>CREATE WRAPPER nom_encapsuleur LIBRARY nom_bibliothèque;</pre> <p>Par exemple, pour enregistrer un encapsuleur avec le nom encapsuleur_fichiers_a_plat sur le serveur fédéré fonctionnant sous le système d'exploitation AIX, émettez l'instruction suivante :</p> <pre>CREATE WRAPPER encapsuleur_fichiers_a_plat LIBRARY 'libdb2lsfile.a';</pre>

Vous devez indiquer le paramètre LIBRARY dans l'instruction CREATE

WRAPPER. Le nom du fichier de bibliothèque de l'encapsuleur que vous indiquez dépend du système d'exploitation du serveur fédéré. Voir dans la liste des fichiers de bibliothèque de l'encapsuleur de fichier structuré sous forme de tables le nom de bibliothèque à indiquer dans l'instruction CREATE WRAPPER.

Fichiers de bibliothèque de l'encapsuleur des fichiers structurés sous forme de tables :

Les fichiers de bibliothèque de l'encapsuleur des fichiers structurés sous forme de tables sont ajoutés au serveur fédéré lorsque vous installez IBM InfoSphere Federation Server.

Lorsque vous installez IBM InfoSphere Federation Server, trois fichiers de bibliothèque sont ajoutés dans le chemin de répertoire par défaut. Par exemple, si le serveur fédéré fonctionne sous AIX, les fichiers de bibliothèque de l'encapsuleur qui sont ajoutés dans le chemin de répertoire sont libdb2lsfile.a, libdb2lsfileF.a et libdb2lsfileU.a. Le fichier de bibliothèque de l'encapsuleur par défaut est libdb2lsfile.a. Les autres fichiers de bibliothèque de l'encapsuleur sont utilisés avec des options spécifiques de l'encapsuleur.

Vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER et indiquer le nom de fichier de bibliothèque de l'encapsuleur par défaut.

Les chemins des répertoires par défaut et les noms de fichier de bibliothèque de l'encapsuleur par défaut sont indiqués dans la table suivante.

Tableau 37. Emplacements et noms de fichier de bibliothèque du client de fichiers structurés sous forme de tables

Systeme d'exploitation	Chemin de répertoire	noms de fichier de bibliothèque d'encapsuleur
AIX	/usr/opt/<chemin_installation>/lib32/ /usr/opt/<chemin_installation>/lib64/	libdb2lsfile.a
Linux	/opt/IBM/db2/<chemin_installation>/lib32 /opt/IBM/db2/<chemin_installation>/lib64	libdb2lsfile.so
Solaris	/opt/IBM/db2/<chemin_installation>/lib32 /opt/IBM/db2/<chemin_installation>/lib64	libdb2lsfile.so
Windows	%DB2PATH%\bin	db2lsfile.dll

<chemin_installation> est le chemin de répertoire dans lequel IBM InfoSphere Federation Server est installé sous Linux ou UNIX.

%DB2PATH% est la variable d'environnement qui est utilisée pour indiquer le chemin de répertoire dans lequel IBM InfoSphere Federation Server est installé sous Windows. Le chemin Windows par défaut est C:\Program Files\IBM\SQLLIB.

Enregistrement de la définition de serveur pour les fichiers structurés sous forme de tables

Pour les fichiers structurés sous forme de tables, vous devez enregistrer une définition de serveur parce que la hiérarchie des objets fédérés exige que les fichiers structurés sous forme de tables, qui sont identifiés par des pseudonymes, soient associés à un objet de définition de serveur spécifique.

Vous pouvez enregistrer une définition de serveur à l'aide du Centre de contrôle ou dans la ligne de commande. Le Centre de contrôle comporte un assistant pour vous guider à travers les étapes nécessaires pour enregistrer la définition de serveur.

Procédure

Pour enregistrer une définition de serveur pour une source de données de fichier structuré sous forme de tables :

Choisissez la méthode que vous voulez utiliser pour enregistrer la définition de serveur :

Méthode	Procédure.
Utilisation du Centre de contrôle	Démarrez l'assistant des objets fédérés. Cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée et cliquez sur Créer des objets fédérés .
Depuis la ligne de commande	Emettez l'instruction CREATE SERVER. Par exemple : CREATE SERVER <i>nom_définition_serveur</i> WRAPPER <i>nom_encapsuleur</i> ;

Instruction CREATE SERVER - exemple pour l'encapsuleur de fichier structuré sous forme de tables :

Utiliser l'instruction CREATE SERVER pour enregistrer la définition de serveur pour l'encapsuleur de fichier structuré sous forme de tables. Cet exemple montre les paramètres obligatoires.

L'exemple suivant vous montre comment enregistrer une définition de serveur appelée *biochem_lab* pour un fichier texte qui contient des données biochimiques. L'instruction CREATE SERVER que vous émettez est :

```
CREATE SERVER biochem_lab WRAPPER encapsuleur_fichiers_a_plat;
```

biochem_lab

Nom que vous attribuez à la définition de serveur du fichier structuré sous forme de tables. Les doublons de noms de définition de serveur ne sont pas autorisés.

WRAPPER *encapsuleur_fichiers_a_plat*

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

Enregistrement de pseudonymes pour les fichiers structurés sous forme de tables

Vous devez enregistrer un pseudonyme pour chaque fichier structuré sous forme de tables auquel vous voulez accéder. Utilisez ces pseudonymes, au lieu des noms des fichiers, lorsque vous interrogez des sources de données de fichiers structurés sous forme de tables.

Restrictions

- Si une zone non numérique est trop longue pour son type de colonne, les données qui dépassent sont tronquées.

- Si une zone décimale du fichier a plus de chiffres après le séparateur décimal qu'il n'est autorisé, les données qui dépassent sont tronquées. Par exemple, si le nombre est 10.123456 et que le type de données indique que 3 chiffres sont autorisés après le séparateur décimal, le nombre est tronqué à 10.123. Le séparateur décimal est déterminé par l'élément RADIXCHAR de la catégorie de support de langue nationale LC_NUMERIC. Le paramètre d'échelle du type des données de colonne indique le nombre de chiffres autorisés après le séparateur décimal.
- Si vous essayez d'accéder à des sources de données de fichiers structurés sous forme de tables qui se trouvent sur une unité partagée par le serveur fédéré fonctionnant sous Windows 2003, vos requêtes risquent d'échouer. C'est une limitation de Windows 2003. Vous pouvez éviter ce problème en indiquant le chemin d'accès absolu dans l'option FILE_PATH de l'instruction CREATE NICKNAME.
- La longueur maximum pour une ligne de données est de 10 mégaoctets (10 485 760 octets).
-

Lorsque vous créez un pseudonyme pour un fichier structuré sous forme de tables, les informations du fichier sont mappées à une table relationnelle. Vous créez des pseudonymes pour votre fichier structuré sous forme de tables de deux manières :

- En indiquant le fichier structuré sous forme de tables lorsque vous créez le pseudonyme à l'aide de l'option de pseudonyme FILE_PATH.
- En indiquant le fichier structuré sous forme de tables lorsque vous interrogez la source de données, à l'aide de l'option de colonne de pseudonyme DOCUMENT. Lorsque cette option est utilisée, le pseudonyme peut être utilisé pour représenter les données de tout fichier structuré sous forme de tables dont le schéma correspond à la définition de pseudonyme.

Les noms que vous donnez aux pseudonymes peuvent comporter jusqu'à 128 caractères.

Vous pouvez enregistrer un pseudonyme à l'aide du Centre de contrôle ou de la ligne de commande. Le Centre de contrôle comporte un assistant pour vous guider à travers les étapes nécessaires pour enregistrer le pseudonyme.

Procédure

Pour enregistrer un pseudonyme pour un fichier structuré sous forme de tables :

Choisissez la méthode que vous voulez utiliser pour enregistrer le pseudonyme :

Méthode	Procédure.
Utilisation du Centre de contrôle	Démarrez l'assistant des objets fédérés. Cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée et cliquez sur Créer des objets fédérés .

Méthode	Procédure.
Utilisation de la ligne de commande	<p>Emettez l'instruction CREATE NICKNAME. Par exemple :</p> <pre>CREATE NICKNAME <i>pseudonyme</i> (nom_colonne <i>type_données</i>, nom_colonne <i>type_données</i>, nom_colonne <i>type_données</i>) FOR SERVER <i>nom_définition_serveur</i> OPTIONS (<i>options_pseudonyme</i>);</pre>

Répétez cette étape pour chaque fichier structuré sous forme de tables pour lequel vous voulez créer un pseudonyme.

Instruction CREATE NICKNAME - exemples pour l'encapsuleur de fichier structuré sous forme de tables :

Utiliser l'instruction CREATE NICKNAME pour enregistrer un pseudonyme pour un fichier structuré sous forme de tables auquel vous voulez accéder.

Vous devez indiquer l'option de pseudonyme FILE_PATH ou bien l'option de colonne de pseudonyme DOCUMENT lorsque vous enregistrez un pseudonyme pour un fichier structuré sous forme de tables.

Création d'un pseudonyme avec l'option de pseudonyme FILE_PATH

L'exemple suivant montre une instruction CREATE NICKNAME pour le fichier structuré sous forme de tables DRUGDATA1.TXT :

```
CREATE NICKNAME DRUGDATA1
(
  Dcode INTEGER NOT NULL,
  Drug CHAR(20),
  Manufacturer CHAR(20)
)
FOR SERVER biochem_lab
OPTIONS(FILE_PATH '/usr/pat/DRUGDATA1.TXT')
```

DRUGDATA1

Pseudonyme unique utilisé pour identifier le fichier structuré sous forme de tables.

Le pseudonyme est un nom composé de deux parties, le schéma et le pseudonyme. Si vous omettez le schéma lorsque vous enregistrez le pseudonyme, le schéma du pseudonyme est l'identificateur utilisateur de l'utilisateur qui enregistre le pseudonyme.

Dcode INTEGER NOT NULL

Les nom et type de données d'une colonne du fichier structuré sous forme de tables qui contient un code de médicament.

Drug CHAR(20)

Les nom et type de données d'une colonne du fichier structuré sous forme de tables qui contient le nom d'un médicament.

Manufacturer CHAR(20)

Les nom et type de données d'une colonne du fichier structuré sous forme de tables qui contient le nom du fabricant du médicament.

FOR SERVER *biochem_lab*

Le nom que vous avez affecté à la définition de serveur du fichier structuré sous forme de tables dans l'instruction CREATE SERVER.

FILE_PATH *'/usr/pat/DRUGDATA1.TXT'*

Indique le chemin de répertoire qualifié complet et le nom de fichier du fichier structuré sous forme de tables qui contient les données auxquelles vous voulez accéder. Le chemin doit être placé entre apostrophes.

Création d'un pseudonyme avec l'option de colonne de pseudonyme DOCUMENT

Lorsque vous créez un pseudonyme à l'aide de l'option de colonne de pseudonyme DOCUMENT, vous indiquez que le nom du fichier structuré sous forme de tables sera fourni lorsque vous exécuterez une requête utilisant le pseudonyme. Vous ne pouvez indiquer l'option de colonne de pseudonyme DOCUMENT que sur une seule colonne lorsque vous enregistrez le pseudonyme. La colonne associée à l'option DOCUMENT doit être un type de données VARCHAR ou CHAR. Vous devez inclure le chemin d'accès complet du fichier lorsque vous exécutez une requête qui utilise le pseudonyme.

L'exemple suivant montre une instruction CREATE NICKNAME qui indique l'option de colonne de pseudonyme DOCUMENT :

```
CREATE NICKNAME customers
(
  doc VARCHAR(100) OPTIONS(DOCUMENT 'FILE'),
  name VARCHAR(16),
  address VARCHAR(30),
  id VARCHAR(16))
FOR SERVER biochem_lab
```

customers

Nom unique pour le pseudonyme.

Le pseudonyme est un nom composé de deux parties, le schéma et le pseudonyme. Si vous omettez le schéma lorsque vous enregistrez le pseudonyme, le schéma du pseudonyme est l'identificateur utilisateur de l'utilisateur qui enregistre le pseudonyme.

doc VARCHAR(100) OPTIONS(DOCUMENT 'FILE')

Les nom et type de données d'une colonne servant à indiquer le nom du fichier structuré sous forme de tables auquel vous voulez accéder. Vous indiquez le nom de fichier lorsque vous exécutez la requête.

name VARCHAR(16)

Les nom et type de données d'une colonne du fichier structuré sous forme de tables qui contient le nom du client.

address VARCHAR(30)

Les nom et type de données d'une colonne du fichier structuré sous forme de tables qui contient l'adresse du client.

id VARCHAR(16)**FOR SERVER** *biochem_lab*

Le nom que vous avez affecté à la définition de serveur du fichier structuré sous forme de tables dans l'instruction CREATE SERVER.

FILE_PATH *'/usr/pat/DRUGDATA1.TXT'*

Indique le chemin de répertoire qualifié complet et le nom de fichier du fichier structuré sous forme de tables qui contient les données auxquelles

vous voulez accéder. Vous devez indiquer l'option de pseudonyme FILE_PATH ou DOCUMENT dans l'instruction CREATE NICKNAME. Le chemin doit être placé entre apostrophes.

Création d'un pseudonyme avec les paramètres facultatifs

L'exemple suivant montre une instruction CREATE NICKNAME pour le fichier structuré sous forme de tables DRUGDATA1.TXT :

```
CREATE NICKNAME DRUGDATA1
(
  Dcode INTEGER NOT NULL,
  Drug CHAR(20),
  Manufacturer CHAR(20)
)
FOR SERVER biochem_lab
OPTIONS(FILE_PATH '/usr/pat/DRUGDATA1.TXT',
  COLUMN_DELIMITER ',',
  SORTED 'Y',
  KEY_COLUMN 'DCODE',
  VALIDATE_DATA_FILE 'Y')
```

DRUGDATA1

Pseudonyme unique utilisé pour identifier le fichier structuré sous forme de tables.

Le pseudonyme est un nom composé de deux parties, le schéma et le pseudonyme. Si vous omettez le schéma lorsque vous enregistrez le pseudonyme, le schéma du pseudonyme est l'identificateur utilisateur de l'utilisateur qui enregistre le pseudonyme.

Dcode INTEGER NOT NULL

Les nom et type de données d'une colonne du fichier structuré sous forme de tables qui contient un code de médicament.

Drug CHAR(20)

Les nom et type de données d'une colonne du fichier structuré sous forme de tables qui contient le nom d'un médicament.

Manufacturer CHAR(20)

Les nom et type de données d'une colonne du fichier structuré sous forme de tables qui contient le nom du fabricant du médicament.

FOR SERVER biochem_lab

Le nom que vous avez affecté à la définition de serveur du fichier structuré sous forme de tables dans l'instruction CREATE SERVER.

FILE_PATH '/usr/pat/DRUGDATA1.TXT'

Indique le chemin de répertoire qualifié complet et le nom de fichier du fichier structuré sous forme de tables qui contient les données auxquelles vous voulez accéder. Vous devez indiquer l'option de pseudonyme FILE_PATH ou DOCUMENT dans l'instruction CREATE NICKNAME. Le chemin doit être placé entre apostrophes.

COLUMN_DELIMITER ','

Indique le délimiteur qui est utilisé pour séparer les zones dans un fichier structuré sous forme de tables. La valeur du délimiteur doit être placée entre apostrophes. Le délimiteur de colonne peut comporter plusieurs caractères. Si vous n'indiquez pas de délimiteur de colonne, le délimiteur par défaut est une virgule. L'apostrophe ne peut pas être utilisée comme délimiteur. Le délimiteur de colonne doit être cohérent sur l'ensemble du fichier. Une valeur NULL est représentée par deux délimiteurs l'un à côté de l'autre ou par un délimiteur suivi d'un caractère de fin de ligne, si la

zone NULL est la dernière de la ligne. Le délimiteur de colonne ne peut pas exister en tant que donnée valide d'une colonne. Par exemple, la virgule ne peut être utilisée comme délimiteur de colonne si une des colonnes contient des données avec des virgules intégrées.

SORTED 'Y'

Indique que le fichier de source de données est trié. Les sources de données triées doivent être triées dans l'ordre croissant conformément à la séquence de classement de l'environnement local en cours, comme défini par les paramètres de la catégorie Prise en charge de la langue nationale LC_COLLATE. Si vous spécifiez que la source de données est triée, définissez l'option VALIDATE_DATA_FILE sur 'Y'. La valeur par défaut pour le paramètre SORTED est 'N'.

KEY_COLUMN 'DCODE'

Le nom de la colonne du fichier qui représente la clé d'après laquelle le fichier est trié. La valeur de la colonne de clé doit être placée entre apostrophes. Indiquez cette option uniquement si vous indiquez l'option de pseudonyme SORTED. Une colonne désignée avec l'option de colonne de pseudonyme DOCUMENT ne doit pas être indiquée comme colonne de clé. La valeur doit être le nom d'une colonne définie dans l'instruction CREATE NICKNAME. La colonne de clé ne doit pas contenir de valeurs NULL. Seules les clés à colonne unique sont prises en charge. Les clés de plusieurs colonnes ne sont pas autorisées. La colonne doit être triée dans l'ordre croissant. Si la valeur n'est pas indiquée pour un pseudonyme trié, la valeur par défaut est la première colonne du fichier dotée d'un pseudonyme.

VALIDATE_DATA_FILE 'Y'

Indique si l'encapsuleur vérifie que la colonne de clé est triée par ordre croissant et contrôle la présence de clés ayant la valeur NULL. Les valeurs valides pour cette option sont Y ou N. Cette option n'est pas autorisée si l'option de colonne de pseudonyme DOCUMENT est utilisée pour le chemin.

Désignation des colonnes de clé lorsque vous enregistrez un pseudonyme

Vous pouvez désigner une colonne de clé en indiquant la contrainte NOT NULL dans l'instruction de pseudonyme :

```
CREATE NICKNAME tox (tox_id INTEGER NOT NULL, toxicity VARCHAR(100))
FOR SERVER tox_server1
  OPTIONS (FILE_PATH'/tox_data.txt', SORTED 'Y')
```

```
CREATE NICKNAME weights (mol_id INTEGER, wt VARCHAR(100) NOT NULL)
FOR SERVER wt_server
  OPTIONS (FILE_PATH'/wt_data.txt', SORTED 'Y', KEY_COLUMN 'WT')
```

NOT NULL

Indique que la colonne peut pas comporter de valeurs NULL ou vides.

L'encapsuleur n'impose pas la contrainte NOT NULL, mais la base de données fédérée l'impose. Si vous créez un pseudonyme et associez une contrainte NOT NULL à une colonne, puis sélectionnez une ligne contenant une valeur NULL pour la colonne, la base de données fédérée émet une erreur SQL0407N indiquant que vous ne pouvez pas attribuer une valeur NULL à une colonne NOT NULL.

Les pseudonymes triés font exception à cette règle. La colonne de clé des pseudonymes triés ne peut pas être NULL. Si une colonne de clé NULL est trouvée pour un pseudonyme trié, l'erreur SQL1822N est émise, indiquant que la colonne de clé est absente.

Noms de colonne sensibles à la casse

La base de données fédérée convertit les noms de colonne en majuscules, à moins que la colonne soit définie avec des guillemets. L'exemple suivant ne fonctionnera pas correctement parce que la valeur de l'option KEY_COLUMN est placée entre apostrophes. Dans cet exemple, le nom de colonne sera converti par la base de données fédérée en EMPNO. Le résultat est que si vous indiquez empno dans une requête, la colonne n'est pas reconnue par la base de données fédérée.

```
CREATE NICKNAME depart (  
  empno char(6) NOT NULL)  
  FOR SERVER DATASTORE  
  OPTIONS(FILE_PATH'data.txt', SORTED 'Y', KEY_COLUMN 'empno');
```

Serveurs fédérés Windows 2003

Si vous essayez d'accéder à des sources de données de fichiers structurés sous forme de tables qui se trouvent sur une unité partagée d'un serveur fédéré fonctionnant sous Windows 2003, votre requête risque d'échouer avec le message d'erreur suivant :

```
SQL1822N Code d'erreur imprévue "ERRNO = 2" reçu de la source de données  
"SERVERNAME1". Le texte et les jetons associés sont "Impossible de lire le fichier".  
SQLSTATE=560BD
```

C'est une limitation de Windows 2003. Vous pouvez éviter ce problème en indiquant le chemin d'accès absolu dans l'option FILE_PATH de l'instruction CREATE NICKNAME.

L'exemple suivant montre une instruction CREATE NICKNAME avec un chemin abrégé indiqué dans l'option FILE_PATH :

```
CREATE NICKNAME pseudonyme  
  (  
    COL1 CHAR (10) NOT NULL  
  )  
  FOR SERVER servername1  
  OPTIONS (FILE_PATH 'X:\textfile1.txt');
```

où X:\ est l'unité mappée à la machine distante. Les requêtes qui utilisent ce pseudonyme risquent d'échouer parce que vous avez indiqué le chemin abrégé.

Pour le serveur fédéré fonctionnant sous Windows 2003, indiquez le chemin d'accès absolu dans l'option FILE_PATH de l'instruction CREATE NICKNAME.

Par exemple :

```
CREATE NICKNAME pseudonyme  
  (  
    COL1 CHAR (10) NOT NULL  
  )  
  FOR SERVER servername1  
  OPTIONS (FILE_PATH '\\host.svl.ibm.com\D$\textfile1.txt') ;
```


Modèle de contrôle d'accès aux fichiers pour l'encapsuleur de fichier structuré sous forme de tables

L'encapsuleur accède aux fichiers structurés sous forme de tables à l'aide des informations d'autorisation du propriétaire d'instance de la base de données fédérée. L'encapsuleur peut uniquement accéder aux fichiers qui peuvent être lus par cet ID utilisateur ou ID groupe. L'identificateur utilisateur qui établit la connexion à la base de données fédérée n'est pas utilisé pour accéder aux fichiers structurés sous forme de tables.

Sur un serveur fédéré, tout fichier structuré sous forme de tables pour lequel un pseudonyme a été créé doit être accessible par le même chemin d'accès depuis chaque noeud. Il n'est pas nécessaire que le fichier soit sur un noeud de la base de données fédérée dès lors que le fichier est accessible de n'importe quel noeud à travers un chemin commun.

Pour accéder à un fichier structuré sous forme de tables, l'encapsuleur exige une identité de l'utilisateur à des fins de sécurité. L'encapsuleur de fichier structuré sous forme de tables utilise l'identité de l'utilisateur qui est associée au service de base de données fédérée. Le nom du service de base de données fédérée dépend du nom de l'instance du gestionnaire de bases de données. Par exemple, si le nom de l'instance du gestionnaire de bases de données est DB2, alors le nom de service est DB2 - DB2. Pour déterminer l'identité de l'utilisateur qui est associée à un service de base de données fédérée, utilisez le Panneau de configuration de Windows pour afficher les services. Cliquez deux fois sur le nom de service et affichez la page Propriétés de connexion.

Fichiers structurés sous forme de tables situés sur des unités distantes

Les fichiers structurés sous forme de tables auxquels vous voulez accéder doivent être sur une unité locale ou mappée.

Réseaux sur lesquels un domaine Windows est configuré

Le compte d'ouverture de session pour le service de base de données fédérée doit être un compte du domaine ayant accès au dossier partagé sur l'unité mappée sur laquelle les fichiers structurés sous forme de tables se trouvent.

Réseaux sans domaine Windows configuré

Le compte d'ouverture de session du service de base de données fédérée doit avoir les mêmes nom d'utilisateur et mot de passe qu'un utilisateur valide sur l'ordinateur qui partage ce dossier. Cet utilisateur doit être sur la liste des autorisations pour le dossier partagé avec au moins un accès en lecture

Instructions pour optimiser les performances d'interrogation de l'encapsuleur de fichier structuré sous forme de tables

Vous pouvez améliorer les performances des requêtes dans les fichiers structurés sous forme de tables en ayant des fichiers triés et en créant des statistiques pour vos pseudonymes.

Utilisez les astuces suivantes pour améliorer les performances d'interrogation :

- Triez les données dans vos fichiers. Le serveur fédéré peut rechercher les données dans les fichiers triés beaucoup plus efficacement que dans les fichiers qui ne sont pas triés.
- Pour les fichiers triés, indiquez une valeur ou plage de valeurs pour la colonne de clé lorsque vous soumettez une requête.
- Les statistiques des pseudonymes de fichiers structurés sous forme de tables doivent être mises à jour manuellement en mettant à jour les vues SYSSTAT et SYSCAT. Utilisez la fonction de mise à jour des statistiques du pseudonyme pour mettre à jour les statistiques des pseudonymes des fichiers structurés sous forme de tables.

Configuration de l'accès aux sources de données Teradata

Pour configurer un serveur fédéré pour accéder aux sources de données Teradata, vous devez fournir au serveur fédéré des informations sur les sources de données et les objets auxquels vous voulez accéder.

Avant de commencer

- Le logiciel client Teradata doit être installé et configuré sur le serveur qui fait office de serveur fédéré.
- La fédération doit être installée sur un serveur qui fait office de serveur fédéré.
- Vérifiez la configuration du serveur fédéré.
- Vérifiez le paramètre fédéré pour vérifier que la fédération est activée.

A propos de cette tâche

Vous pouvez configurer un serveur fédéré pour accéder aux données qui sont stockées dans les sources de données Teradata à l'aide du Centre de contrôle DB2 ou en émettant des instructions SQL sur la ligne de commande DB2. Le Centre de contrôle DB2 comporte un assistant pour vous guider à travers les étapes nécessaires pour configurer les objets fédérés obligatoires.

Procédure

Pour configurer l'accès aux sources de données Teradata :

1. Testez la connexion au serveur Teradata.
2. Vérifiez que la bibliothèque Teradata est activée pour la liaison d'exécution (AIX).
3. Définissez les variables d'environnement pour l'encapsuleur Teradata.
4. Enregistrez l'encapsuleur.
5. Enregistrez la définition de serveur.
6. Créez les mappages utilisateur.
7. Testez la connexion au serveur Teradata.
8. Enregistrez des pseudonymes pour les tables et vues Teradata.

Test de la connexion au serveur Teradata

Tester la connexion au serveur Teradata pour vérifier que le logiciel client Teradata est correctement configuré sur le serveur fédéré.

Avant de commencer

L'utilitaire Basic Teradata Query (BTEQ) et Teradata Data Connector Application Program Interface (PIOM) doivent être installés sur le serveur fédéré. L'utilitaire BTEQ et PIOM sont installés sur le serveur fédéré lorsque vous installez le logiciel client Teradata.

A propos de cette tâche

Vous utilisez l'utilitaire BTEQ pour soumettre une requête SQL afin de vérifier que le serveur fédéré peut se connecter au serveur Teradata. Voir la documentation Teradata pour de plus amples informations sur l'utilitaire BTEQ.

Procédure

Pour tester la connexion au serveur Teradata :

1. Démarrez une session de l'utilitaire BTEQ et connectez-vous au serveur Teradata.
2. Emettez une commande SQL pour vérifier que vous pouvez vous connecter avec succès au serveur Teradata.

Par exemple :

```
select count(*) from dbc.tables;
```

Si la connexion est réussie, vous voyez la sortie de la requête.

Si la connexion échoue, vous recevez une erreur. Vérifiez que le logiciel client Teradata est correctement installé et configuré sur le serveur fédéré.

3. Déconnectez-vous du serveur Teradata et fermez la session de l'utilitaire BTEQ.

Une fois cette tâche terminée, vérifiez que la bibliothèque Teradata est activée pour la liaison d'exécution.

Vérifier que la bibliothèque Teradata est activée pour la liaison d'exécution (AIX)

Lorsque vous ajoutez une source de données Teradata à votre serveur fédéré sous AIX, vous devez vérifier que la liaison d'exécution est activée avant d'enregistrer des encapsuleurs ou des serveurs.

Procédure

Pour vérifier que la bibliothèque Teradata est activée pour la liaison d'exécution :

1. Accédez au répertoire dans lequel le fichier libcliv2.so se trouve.
Le fichier libcliv2.so est installé avec le logiciel client Teradata. Par défaut, il est installé dans le répertoire /usr/lib.
2. A partir d'une invite de commande, émettez la commande UNIX suivante pour vérifier si la liaison d'exécution est activée :

```
dump -H libcliv2.so | grep libtli.a
```
3. Vérifiez les noms de fichier qui sont renvoyés. Si le nom de fichier libtli.a est renvoyé, la bibliothèque Teradata est activée pour la liaison d'exécution.

Si le nom de fichier libtli.a n'est pas renvoyé, ouvrez une fenêtre de commande et émettez les commandes UNIX suivantes pour activer la liaison d'exécution pour la bibliothèque Teradata :

```
rtl_enable libcliv2.so -F libtli.a
mv libcliv2.so libcliv2.so.old
mv libcliv2.so.new libcliv2.so
chmod a+r libcliv2.so
```

Une fois cette tâche terminée, vous pouvez définir les variables d'environnement.

Définition des variables d'environnement Teradata

Les variables d'environnement Teradata doivent être définies dans le fichier db2dj.ini sur le serveur fédéré.

Restrictions

Lisez les restrictions pour le fichier db2dj.ini.

Le fichier db2dj.ini contient des informations de configuration sur le logiciel client Teradata qui est installé sur votre serveur fédéré.

Il y a des variables d'environnement obligatoires et facultatives pour les sources de données Teradata.

Si vous avez installé le logiciel client Teradata avant d'installer l'encapsuleur Teradata, les variables d'environnement Teradata obligatoires sont définies dans le fichier db2dj.ini.

Vous devez définir les variables d'environnement à l'aide des étapes de cette tâche si vous n'avez pas installé le logiciel client Teradata avant d'installer l'encapsuleur Teradata ou que vous voulez définir l'une des variables d'environnement facultatives.

Procédure

Pour définir les variables d'environnement Teradata :

1. Utilisez l'une des méthodes suivantes :

Méthode	Etape
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .
Définissez automatiquement les variables d'environnement.	Exécutez l'assistant d'installation IBM InfoSphere Federation Server. Suivez les instructions de l'assistant. Important : Définissez les variables d'environnement obligatoires en exécutant l'assistant d'installation. Les variables d'environnement facultatives doivent être définies manuellement.

Méthode	Etape
Définissez manuellement les variables d'environnement.	<p>Modifiez le fichier db2dj.ini :</p> <p>Le fichier db2dj.ini se trouve dans le répertoire que la variable de registre DB2_DB2_DJ_INI indique. Si la variable DB2_DJ_INI n'est pas définie, le fichier db2dj.ini figure dans un des chemins par défaut suivants selon le système d'exploitation :</p> <ul style="list-style-type: none"> Sous UNIX : <i>instancehome</i>/sqlib/cfg/db2dj.ini. <i>instancehome</i> Le répertoire de base du propriétaire d'instance. Sous Windows : %DB2PATH%\cfg\db2dj.ini %DB2PATH% Le répertoire dans lequel le système de la base de données DB2 est installé, par exemple, C:\Program Files\IBM\sqlib. <p>Si le fichier n'existe pas, vous pouvez créer un fichier portant le nom db2dj.ini à l'aide de tout éditeur de texte. Dans le fichier db2dj.ini, vous devez indiquer le chemin qualifié complet dans la valeur des variables d'environnement, sinon vous rencontrerez des erreurs. Par exemple :</p> <p>COPLIB=/usr/lib</p>

- Définissez les variables d'environnement de conversion de la page de codes Teradata (si nécessaire).
- Pour être sûr que les variables d'environnement sont définies sur le serveur fédéré, recyclez l'instance de base de données fédérée avec les commandes suivantes :

```
db2stop
db2start
```

Une fois cette tâche terminée, vous pouvez enregistrer l'encapsuleur.

Variables d'environnement Teradata

Il y a des variables d'environnement obligatoires et facultatives pour les sources de données Teradata. Ces variables sont définies dans le fichier db2dj.ini.

Les variables d'environnement suivantes sont valides pour Teradata :

- COPLIB
- COPERR
- TERADATA_CHARSET (facultative)
- NETRACE (facultative)
- COPANOMLOG (facultative)

Description des variables

COPLIB

Indique le chemin de répertoire sur le serveur fédéré pour le fichier libcliv2.so. Indique le chemin qualifié complet pour la variable COPLIB.

Par exemple :

```
COPLIB=/usr/lib
```

Les fichiers libcliv2.so et errmsg.cat se trouvent généralement dans le même répertoire.

COPERR

Indique le chemin de répertoire sur le serveur fédéré pour le fichier `errmsg.cat`. Indique le chemin qualifié complet pour la variable `COPERR`.

Par exemple :

```
COPERR=/usr/lib
```

TERADATA_CHARSET

Indique le jeu de caractères de la page de codes à utiliser avec les sources de données Teradata.

Chaque fois que le serveur fédéré se connecte à une source de données Teradata, l'encapsuleur Teradata détermine quel jeu de caractères de la page de codes utiliser pour cette connexion. Le jeu de caractères de la page de codes peut être défini par l'encapsuleur Teradata ou bien vous pouvez désigner une page de codes en définissant la variable d'environnement `TERADATA_CHARSET`.

Si la variable d'environnement `TERADATA_CHARSET` est définie dans le fichier `db2dj.ini` sur le serveur fédéré, l'encapsuleur utilise le jeu de caractères de la page de codes dans le fichier `db2dj.ini`. La valeur de la variable d'environnement `TERADATA_CHARSET` n'est pas validée, mais si la variable d'environnement n'est pas définie sur une valeur valide, la source de données Teradata renvoie une erreur.

Si la variable d'environnement `TERADATA_CHARSET` n'est pas définie dans le fichier `db2dj.ini` sur le serveur fédéré, l'encapsuleur détecte le jeu de caractères du client d'après la page de codes de la base de données.

Sur les serveurs fédérés qui fonctionnent sous UNIX, les valeurs suivantes sont valides pour la variable d'environnement `TERADATA_CHARSET` :

- `HANGULKSC5601_2R4`
- `KanjiEUC_0U`
- `LATIN1_0A`
- `LATIN9_0A`
- `LATIN1252_0A`
- `SCHGB2312_1T0`
- `TCHBIG5_1R0`
- `UTF8`
- `ASCII`

Sur les serveurs fédérés qui fonctionnent sous Windows, les valeurs suivantes sont valides pour la variable d'environnement `TERADATA_CHARSET` :

- `HANGULKSC5601_2R4`
- `KanjiSJIS_0S`
- `LATIN1_0A`
- `LATIN1252_0A`
- `SCHGB2312_1T0`
- `TCHBIG5_1R0`
- `UTF8`
- `ASCII`

NETRACE

Facultatif. Active la fonction de traçage du logiciel client Teradata. Cette variable n'est nécessaire que pour le débogage.

COPANOMLOG

Facultatif. Active la fonction de consignation du logiciel client Teradata.
Cette variable n'est nécessaire que pour le débogage.

Vérification du jeu de caractères sur le serveur Teradata

Si le jeu de caractères approprié n'est pas indiqué sur le serveur Teradata, vous pouvez recevoir des erreurs de connexion. Vérifiez que le jeu de caractères que vous voulez utiliser est installé sur le serveur Teradata.

Procédure

Pour vérifier que le jeu de caractères que vous voulez utiliser est installé sur le serveur Teradata :

1. Connectez-vous au serveur Teradata à l'aide de l'utilitaire BTEQ ou de n'importe quel autre utilitaire de connexion valide.
2. Emettez l'instruction suivante pour afficher la table `dbc.chartranslations` :
`select * from dbc.chartranslations;`
3. Vérifiez la valeur dans la troisième colonne, `InstallFlag`, de la table renvoyée. La valeur 'Y' dans la troisième colonne indique que le jeu de caractères est installé et utilisé sur le serveur Teradata.

Utilisez la table suivante pour déterminer si le jeu de caractères approprié est installé :

Tableau 38. Jeux de caractères pour Teradata

Jeu de caractères codés sur deux octets	Jeu de caractères mono-octet	Jeu de caractères Teradata	Langue	Jeu de codes DB2 IBM
941	897	"KanjiSJIS_0S"	Japonais	IBM-943
1362	1126	"HANGULKSC5601_2R4"	Coréen	1363
1385	1114	"SCHGB2312_1T0"	Chinois simplifié	GBk
380	1115	"SCHGB2312_1T0"	Chinois simplifié	IBM-1381
947	1114	"TCHBIG5_1R0"	Chinois traditionnel	big5
1200	1208	"UTF8"	Unicode	UTF-8
0	819	"Latin1_0A"	Anglais (latin 1)	ISO8859-1
0	1252	"Latin1252_0A"	Anglais (Win Latin)	ISO8859-1/15
0	819	"ASCII"	Anglais (ASCII)	ISO8859-1

4. Si le jeu de caractères obligatoire n'est pas installé, installez-le pour utiliser l'encapsuleur Teradata.
 - Le code ASCII est activé sur le serveur Teradata, mais il n'est pas catalogué dans la table `dbc.chartranslations`. Si toutes les valeurs renvoyées pour `InstallFlag` sont 'N', le code ASCII est le seul jeu de caractères valide sur le serveur Teradata et la variable d'environnement `TERADATA_CHARSET` doit être définie sur ASCII dans le fichier `db2dj.ini`.
 - Si le jeu de caractères que vous voulez utiliser est indiqué dans la table `dbc.chartranslations`, mais que la valeur `InstallFlag` est définie sur 'N', émettez l'instruction suivante pour modifier l'indicateur `InstallFlag` sur 'Y' :
`update dbc.chartranslations
set installflag='Y' where CharSetName= 'nom_jeu_caractères';`
 - Si le jeu de caractères que vous voulez utiliser n'est pas indiqué dans la table `dbc.chartranslations`, contactez le service clients Teradata.

- Redémarrez le serveur Teradata pour mettre à jour la liste des jeux de caractères. Dans une fenêtre Teradata de commande, entrez : tpareset -f raison_du_redémarrage

Identification et résolution des incidents avec les jeux de caractères pour les sources de données Teradata :

Lorsque vous définissez la variable d'environnement TERADATA_CHARSET pour une source de données Teradata, vous risquez de rencontrer des erreurs si le jeu de caractères approprié n'est pas indiqué.

Symptôme

Si le jeu de caractères approprié n'est pas indiqué pour la source de données Teradata, vous rencontrerez l'erreur suivante :

```
SQL 1822N Code d'erreur imprévue "227"  
reçu de la source de données "<chaîne>". Les texte et jetons  
associés sont "MTDP: EM_CHARNAME(227) : nom jeu de caractères  
indiqué invalide". SQLSTATE=560BD
```

Cause

Le jeu de caractères qui est indiqué dans la variable d'environnement TERADATA_CHARSET est incorrect.

Résolution de l'incident

Vérifiez que le jeu de caractères approprié est installé et indiqué sur le serveur Teradata et dans le fichier db2dj.ini.

Enregistrement de l'encapsuleur Teradata

Vous devez enregistrer un encapsuleur pour accéder aux sources de données Teradata. Les serveurs fédérés utilisent des encapsuleurs pour dialoguer avec les sources de données et en extraire des données. Les encapsuleurs sont mis en oeuvre comme un ensemble de fichiers de bibliothèque.

Procédure

Pour enregistrer l'encapsuleur Teradata :

Utilisez l'une des méthodes suivantes :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés.

Méthode	Description
<p>Emettez l'instruction CREATE WRAPPER et indiquez le nom par défaut pour l'encapsuleur Teradata.</p>	<p>Par exemple :</p> <pre>CREATE WRAPPER TERADATA;</pre> <p>A faire : Lorsque vous enregistrez l'encapsuleur à l'aide du nom par défaut, TERADATA, le serveur fédéré utilise automatiquement la bibliothèque de l'encapsuleur Teradata appropriée au système d'exploitation sous lequel votre serveur fédéré fonctionne.</p> <p>En cas de conflit entre le nom d'encapsuleur par défaut et un nom d'encapsuleur existant dans la base de données fédérée, vous pouvez remplacer le nom d'encapsuleur par défaut par un nom que vous choisissez. Lorsque vous utilisez un nom différent du nom par défaut, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER.</p> <p>Par exemple, pour enregistrer un encapsuleur avec le nom encapsuleur_tera sur un serveur fédéré qui fonctionne sous AIX, émettez l'instruction suivante :</p> <pre>CREATE WRAPPER encapsuleur_tera LIBRARY 'libdb2teradata.a' OPTIONS (DB2_FENCED 'Y');</pre> <p>Le fichier de bibliothèque de l'encapsuleur que vous indiquez dépend du système d'exploitation du serveur fédéré.</p> <p>Options obligatoires pour AIX et Solaris Remarque : L'option d'encapsuleur DB2_FENCED dans l'instruction CREATE WRAPPER est obligatoire pour AIX et Solaris parce que l'encapsuleur Teradata ne prend en charge que les clients 32 bits.</p> <p>Par exemple :</p> <pre>CREATE WRAPPER TERADATA OPTIONS (DB2_FENCED 'Y')</pre>

Une fois cette tâche terminée, vous pouvez enregistrer la définition de serveur.

Fichiers de bibliothèque de l'encapsuleur Teradata

Les fichiers de bibliothèque de l'encapsuleur Teradata sont ajoutés au serveur fédéré lorsque vous installez l'encapsuleur.

Lorsque vous installez l'encapsuleur Teradata, trois fichiers de bibliothèque sont ajoutés dans le chemin de répertoire par défaut. Par exemple, si le serveur fédéré fonctionne sous AIX, les fichiers de bibliothèque de l'encapsuleur qui sont ajoutés dans le chemin de répertoire sont libdb2teradata.a, libdb2teradataF.a et libdb2teradataU.a. Le fichier de bibliothèque de l'encapsuleur par défaut est libdb2teradata.a. Les autres fichiers de bibliothèque de l'encapsuleur sont utilisés sur le plan interne par l'encapsuleur Teradata.

Si vous n'utilisez pas le nom d'encapsuleur par défaut lorsque vous enregistrez l'encapsuleur, vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER et indiquer le nom de fichier de bibliothèque de l'encapsuleur par défaut.

Les chemins des répertoires par défaut et les noms de fichier de bibliothèque de l'encapsuleur par défaut sont indiqués dans la table suivante.

Tableau 39. Emplacements et noms de fichier de la bibliothèque de l'encapsuleur Teradata

Système d'exploitation	Chemin de répertoire	Noms de fichier de bibliothèque
AIX	/usr/opt/chemin_installation/lib32/ /usr/opt/chemin_installation/lib64/	libdb2teradata.a
Solaris	/opt/IBM/db2/chemin_installation/lib32 /opt/IBM/db2/chemin_installation/lib64	libdb2teradata.so
Windows	%DB2PATH%\bin	db2teradata.dll

chemin_installation est le chemin de répertoire dans lequel le serveur fédéré est installé sous UNIX.

Enregistrement des définitions de serveur pour une source de données Teradata

Vous devez enregistrer chaque serveur Teradata auquel vous voulez accéder dans la base de données fédérée.

Procédure

Pour enregistrer une définition de serveur pour une source de données Teradata :

1. Recherchez le fichier hosts.
 - Sur les serveurs fédérés qui fonctionnent sous AIX, le fichier hosts se trouve dans le répertoire /etc/hosts.
 - Sur les serveurs fédérés qui fonctionnent sous Windows, le fichier hosts se trouve dans le répertoire %WINDIR%\system32\drivers\etc\hosts.
2. Recherchez dans le fichier hosts l'alias du serveur distant.
Cet alias commence par une chaîne alphabétique et se termine par le suffixe COP*n*. La valeur *n* est le numéro du processeur d'application qui est associé au processeur de transmission Teradata.
3. Recherchez la première zone non numérique sur la ligne du fichier hosts qui contient l'alias.

Exemple de fichier hosts :

```
127.0.0.1      localhost

9.22.5.77     nodexyz      nodexyzCOP1   # teradata server

9.66.111.133  rtplib05.data.xxx.com aap
9.66.111.161  rtpscm11.data.xxx.com aaprwr
9.66.111.161  rtpscm11.data.xxx.com accessm
```

Dans cet exemple, nodexyz est le nom du noeud.

4. Pour créer le serveur, utilisez une des méthodes suivantes :

Méthodes	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .

Méthodes	Description
Emettez l'instruction CREATE SERVER.	<p>Par exemple :</p> <pre>CREATE SERVER <i>nom_définition_serveur</i> TYPE TERADATA VERSION <i>version_number</i> WRAPPER <i>nom_encapsuleur</i> OPTIONS (NODE '<i>nom_noeud</i>');</pre> <p>Bien que la variable '<i>nom_noeud</i>' soit indiquée comme option dans l'instruction CREATE SERVER, cette option est obligatoire pour les sources de données Teradata.</p>

Instruction CREATE SERVER - exemples pour l'encapsuleur Teradata

Utiliser l'instruction CREATE SERVER pour enregistrer des définitions de serveur pour l'encapsuleur Teradata. Cette rubrique comporte un exemple complet avec les paramètres obligatoires et un exemple avec des options de serveur supplémentaires.

L'exemple suivant vous montre comment enregistrer une définition de serveur pour un encapsuleur Teradata en émettant l'instruction CREATE SERVER :

```
CREATE SERVER serveur_tera TYPE TERADATA
  VERSION 2.5 WRAPPER mon_encapsuleur
  OPTIONS (NODE 'noeud_tera');
```

serveur_tera

Nom que vous attribuez au serveur de la base Teradata. Les doublons de noms de définition de serveur ne sont pas autorisés.

TYPE TERADATA

Indique le type de serveur de source de données auquel vous configurez l'accès. Pour l'encapsuleur Teradata, le type de serveur doit être TERADATA.

VERSION 2.5

La version du serveur de la base Teradata auquel vous voulez accéder.

WRAPPER TERADATA

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

NODE '*noeud_tera*'

Le nom du noeud sur lequel le serveur de la base Teradata se trouve. Obtenez le nom du noeud à partir du fichier hosts. Cette valeur fait la distinction entre les majuscules et les minuscules.

Bien que le nom du noeud soit indiqué comme option dans l'instruction CREATE SERVER, il doit être indiqué obligatoirement pour les sources de données Teradata.

Options de serveur

Lorsque vous créez une définition de serveur, vous pouvez indiquer des options de serveur supplémentaires dans l'instruction CREATE SERVER. Les options de serveur peuvent être des options de serveur générales et des options de serveur spécifiques à Teradata.

Les options de serveur CPU_RATIO et IO_RATIO fournissent des informations statistiques sur le serveur Teradata à l'optimiseur de requêtes. Pour indiquer que les ressources de l'unité centrale du serveur fédéré sont deux fois plus puissantes

que les ressources de l'unité centrale du serveur Teradata, définissez la valeur de l'option de serveur CPU_RATIO sur 2.0. Pour indiquer que les unités d'entrée-sortie du serveur fédéré traitent les données trois fois plus vite que les unités d'entrée-sortie du serveur Teradata, définissez l'option de serveur IO_RATIO sur 3.0.

L'exemple suivant montre une définition de serveur Teradata avec les options suivantes :

```
CREATE SERVER serveur_tera TYPE TERADATA
  VERSION 2.5 WRAPPER mon_encapsuleur
  OPTIONS (NODE 'noeud_tera', CPU_RATIO '2.0', IO_RATIO '3.0');
```

Création d'un mappage utilisateur pour une source de données Teradata

Lorsque vous essayez d'accéder à un serveur Teradata, le serveur fédéré établit une connexion avec le serveur Teradata à l'aide d'un ID utilisateur et d'un mot de passe qui sont valides pour cette source de données.

A propos de cette tâche

Créez un mappage utilisateur pour chaque ID utilisateur qui accédera au système fédéré pour envoyer des requêtes réparties à la source de données Teradata.

Procédure

Pour créer les mappages utilisateur pour une source de données Teradata :

Emettez une instruction CREATE USER MAPPING.

Par exemple :

```
CREATE USER MAPPING FOR IDutilisateur_local SERVER nom_définition_serveur
  OPTIONS (REMOTE_AUTHID 'IDutilisateur_distant', REMOTE_PASSWORD
'motdepasse_distant')
```

Bien que les variables REMOTE_AUTHID et REMOTE_PASSWORD soient indiquées comme options dans l'instruction CREATE USER MAPPING, ces options doivent être indiquées obligatoirement pour accéder aux sources de données Teradata.

Une fois cette tâche terminée, testez la connexion entre le serveur fédéré et le serveur Teradata.

Instruction CREATE USER MAPPING - exemples pour l'encapsuleur Teradata

Utiliser l'instruction CREATE USER MAPPING pour mapper un identificateur utilisateur du serveur fédéré à un ID utilisateur et à un mot de passe Teradata distants. Cette rubrique comporte un exemple complet avec les paramètres obligatoires et un exemple qui vous montre comment utiliser le registre spécial USER DB2 avec l'instruction CREATE USER MAPPING.

L'exemple suivant montre comment mapper un identificateur utilisateur local à un ID utilisateur et à un mot de passe Teradata distants :

```
CREATE USER MAPPING FOR MICHAEL SERVER serveur_tera
  OPTIONS (REMOTE_AUTHID 'mike', REMOTE_PASSWORD 'passxyz123');
```

MICHAEL

Indique l'identificateur utilisateur local que vous mappez à l'ID utilisateur et au mot de passe distants qui sont définis sur le serveur Teradata.

SERVER *serveur_tera*

Indique le nom de définition de serveur que vous avez enregistré dans l'instruction CREATE SERVER pour le serveur Teradata.

REMOTE_AUTHID *'mike'*

Indique l'ID utilisateur Teradata distant auquel vous mappez *MICHAEL*. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_ID sur 'U' ou 'L' dans l'instruction CREATE SERVER.

Bien que l'ID utilisateur distant soit indiqué comme option dans l'instruction CREATE SERVER, il doit être indiqué obligatoirement pour les sources de données Teradata.

REMOTE_PASSWORD *'passxyz123'*

Indique le mot de passe Teradata distant qui est associé à *'mike'*. La valeur est sensible à la casse, à moins que vous ne définissiez l'option de serveur FOLD_PW sur 'U' ou 'L' dans l'instruction CREATE SERVER.

Bien que le mot de passe distant soit indiqué comme option dans l'instruction CREATE SERVER, il doit être indiqué obligatoirement pour les sources de données Teradata.

registre spécial USER DB2

Vous pouvez utiliser le registre spécial USER DB2 pour mapper l'identificateur utilisateur de la personne qui émet l'instruction CREATE USER MAPPING à l'identificateur utilisateur de la source de données qui est indiqué dans l'option utilisateur REMOTE_AUTHID.

L'exemple suivant montre une instruction CREATE USER MAPPING qui comporte le registre spécial USER :

```
CREATE USER MAPPING FOR USER SERVER serveur_tera
  OPTIONS (REMOTE_AUTHID 'mike', REMOTE_PASSWORD 'passxyz123');
```

Test de la connexion au serveur Teradata

Testez la connexion au serveur de source de données Teradata pour déterminer si le serveur fédéré est correctement configuré pour accéder aux sources de données Teradata.

A propos de cette tâche

Vous pouvez tester la connexion au serveur Teradata à l'aide de la définition de serveur et des mappages utilisateur que vous avez définis.

Procédure

Pour tester la connexion au serveur Teradata :

Ouvrez une session passe-système et émettez une instruction SELECT sur les tables système Teradata. Si l'instruction SELECT renvoie un comptage, votre définition de serveur et votre mappage utilisateur sont configurés correctement.

Par exemple :

```
SET PASSTHRU nom_définition_serveur
SELECT count(*) FROM dbc.tables
SET PASSTHRU RESET
```

Si l'instruction SELECT renvoie une erreur, vous devez identifier et résoudre les erreurs de connexion.

Une fois cette tâche terminée, vous pouvez enregistrer les pseudonymes pour les tables et vues Teradata.

Identification et résolution des erreurs de connexion à la source de données

Une connexion de test au serveur de source de données peut renvoyer une erreur pour plusieurs raisons. Il y a des mesures que vous pouvez prendre pour déterminer pourquoi l'erreur s'est produite.

Symptôme

Une erreur est renvoyée lorsque vous essayez de vous connecter à la source de données.

Cause

Il y a plusieurs causes possibles à un incident de connexion.

Résolution de l'incident

Pour identifier et résoudre les erreurs de connexion à la source de données, recherchez les problèmes éventuels avec les éléments suivants :

- Vérifiez que la source de données est disponible.
- S'il y a lieu, vérifiez que le serveur de source de données est configuré pour les connexions entrantes.
- Vérifiez que les paramètres de votre mappage utilisateur pour les options REMOTE_AUTHID et REMOTE_PASSWORD sont valides pour les connexions à la source de données. Modifiez le mappage utilisateur ou créez un autre mappage utilisateur si nécessaire.
- S'il y a lieu, vérifiez que le logiciel client de la source de données sur le serveur fédéré est installé et configuré correctement pour se connecter à la source de données.
- Pour les sources de données ODBC, vérifiez que le pilote ODBC sur le serveur fédéré est installé et configuré correctement pour se connecter au serveur de source de données ODBC. Sur les serveurs fédérés qui fonctionnent sous Windows, utilisez l'outil Administrateur de sources de données ODBC pour vérifier le pilote. Sur les serveurs fédérés qui fonctionnent sous UNIX, consultez la documentation du fournisseur de client ODBC.
- Vérifiez que les valeurs des variables définies sur le serveur fédéré sont appropriées à la source de données. Ces variables sont les variables d'environnement système, les variables du fichier db2dj.ini et les variables du registre de profil DB2 (db2set).
- Vérifiez votre définition de serveur. Si nécessaire, supprimez la définition de serveur et recréez-la.

Enregistrement de pseudonymes pour les tables et vues Teradata

Dans chaque définition de serveur Teradata que vous enregistrez, vous devez enregistrer un pseudonyme pour chaque table ou vue à laquelle vous voulez accéder. Utilisez ces pseudonymes, au lieu des noms des objets de source de données, lorsque vous interrogez les serveurs Teradata.

Avant de commencer

La base de données fédérée exploite les statistiques du catalogue de la source de données pour optimiser le traitement des requêtes. Pour garantir que la base de données fédérée dispose de statistiques complètes sur les tables Teradata, utilisez la commande Teradata COLLECT STATISTICS avant d'enregistrer un pseudonyme.

A partir du serveur Teradata, utilisez la commande Teradata COLLECT STATISTICS pour collecter les statistiques sur une ou plusieurs colonnes ou index d'une table.

Lorsque vous enregistrez le pseudonyme avec l'instruction CREATE NICKNAME, la base de données fédérée lit les statistiques dans le catalogue système Teradata et met à jour les statistiques locales pour le pseudonyme.

A propos de cette tâche

Lorsque vous enregistrez un pseudonyme dans une vue Teradata, la base de données fédérée reconnaît toutes les colonnes de la vue comme NULL admis, même si les colonnes de la vue Teradata n'autorisent pas les valeurs NULL. Il n'y a pas de solution palliative à cette limitation.

Procédure

Pour enregistrer un pseudonyme pour une table ou une vue Teradata :

Méthode	Description
Utilisez l'assistant des objets fédérés du centre de contrôle DB2.	Pour démarrer l'assistant, cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée, puis cliquez sur Créer des objets fédérés .
Emettez l'instruction CREATE NICKNAME. La longueur des pseudonymes peut aller jusqu'à 128 caractères.	Par exemple : <pre>CREATE NICKNAME <i>pseudonyme</i> FOR <i>nom_définition_serveur.schema_distant.table_distante</i>;</pre>

Lorsque vous créez le pseudonyme, le serveur fédéré interroge le catalogue de la source de données à l'aide du pseudonyme. Cette requête teste la connexion à la table ou à la vue de la source de données. Si la connexion ne fonctionne pas, vous recevez un message d'erreur.

Répétez cette étape pour chaque table ou vue Teradata pour laquelle vous voulez créer un pseudonyme.

Pseudonymes Teradata sur les serveurs fédérés

Lorsque vous interrogez une source de données Teradata depuis un serveur fédéré, vous utilisez un pseudonyme dans la requête pour identifier les table et vue Teradata auxquelles vous voulez accéder.

Lorsque vous créez un pseudonyme pour une table ou une vue Teradata, le serveur fédéré se connecte au serveur Teradata associé à la table ou à la vue. Le serveur fédéré utilise le pseudonyme pour vérifier la connexion au serveur Teradata. La base de données fédérée vérifie la présence de la table ou de la vue sur la source de données, puis essaie de collecter des données statistiques sur la table ou la vue Teradata dans le catalogue sur le serveur Teradata. Les statistiques qui sont regroupées sur l'objet doté d'un pseudonyme sont stockées dans le catalogue global sur le serveur fédéré.

Le serveur fédéré exploite les statistiques qu'il collecte sur les objets dotés d'un pseudonyme pour optimiser le traitement des requêtes. Tout ou partie des informations du catalogue Teradata pouvant être utilisées par l'optimiseur de requêtes, vous devez mettre à jour les statistiques sur le serveur Teradata avant de créer un pseudonyme. Mettez à jour les statistiques sur le serveur Teradata à l'aide d'une commande ou d'un utilitaire équivalant à la commande RUNSTATS de DB2.

Vous ne pouvez pas soumettre une instruction INSERT, UPDATE ou DELETE à un pseudonyme référençant une vue Teradata modifiable à moins que l'instruction SQL puisse être complètement envoyée à la source de données Teradata.

Instruction CREATE NICKNAME - exemples pour l'encapsuleur Teradata

Utiliser l'instruction CREATE NICKNAME pour enregistrer un pseudonyme pour une table ou une vue Teradata à laquelle vous voulez accéder. Cette rubrique comporte un exemple complet avec les paramètres obligatoires.

Cet exemple montre comment créer un pseudonyme pour une table ou une vue Teradata sur le serveur Teradata :

```
CREATE NICKNAME TERASALES FOR serveur_tera.vinnie.europe ;
```

TERASALES

Pseudonyme unique utilisé pour identifier la table ou la vue Teradata.

Important : Le pseudonyme est un nom composé de deux parties, le schéma et le pseudonyme. Si vous omettez le schéma lorsque vous enregistrez le pseudonyme, le schéma du pseudonyme sera l'identificateur utilisateur de l'utilisateur qui enregistre le pseudonyme. L'identificateur utilisateur se rapporte au serveur fédéré, non à la source de données Teradata distante.

serveur_tera.vinnie.europe

Identificateur en trois parties pour l'objet distant :

- *serveur_tera* est le nom de définition de serveur que vous avez affecté au serveur de la base Teradata dans l'instruction CREATE SERVER.
- *vinnie* est l'ID utilisateur du propriétaire de la table ou vue. Cette valeur fait la distinction entre les majuscules et les minuscules.
- *europe* est le nom de la table ou de la vue distante à laquelle vous voulez accéder.

Identification et résolution des incidents avec la configuration de la source de données Teradata

Activation de la liaison d'exécution pour libcliv2.so (AIX)

Si vous exécutez le fichier `djxlinkTeradata.sh` pour créer un lien vers la bibliothèque partagée Teradata appelée `libcliv2.so`, vous risquez de recevoir un message d'erreur lorsque vous émettez une instruction `CREATE NICKNAME`.

Symptôme

Voici un exemple de message d'erreur que vous risquez de recevoir :

```
DB21034E La commande a été traitée comme une instruction SQL, car ce n'était pas
une commande de l'interpréteur de commandes valide. Pendant le traitement SQL,
```

elle a renvoyé :

```
SQL30081N Une erreur de communication a été détectée. Protocole de communication
utilisé : "TCP/IP". API de communication actuellement utilisée : "SOCKETS".
```

Emplacement

```
où l'erreur a été détectée : "9.112.26.28". Fonction de communication détectant
l'erreur : "recv". Code(s) d'erreur spécifique(s) au protocole : "*", "*", "0".
SQLSTATE=08001
```

Cause

La fonction `OsCall` a entraîné l'arrêt du serveur fédéré.

Résolution de l'incident

Si vous recevez un message d'erreur, recherchez d'éventuels fichiers de déroutement dans le répertoire `/sqlib/db2dump`. Les noms de fichier de déroutement commencent par la lettre `t` et se terminent par le suffixe `000`. Par exemple :

```
t123456.000
```

Recherchez dans les informations de trace du fichier de déroutement d'éventuelles références de fonction `OsCall` qui indiquent que la fonction `OsCall` a entraîné l'arrêt du serveur fédéré.

L'exemple suivant montre des informations de trace avec une référence de fonction `OsCall` que vous pourriez trouver dans un fichier de déroutement :

```
*** Démarrer la trace ***
```

```
0x239690E0 OsCall + 0x28C
0x23973FB0 mtdpassn + 0x8A4
0x239795A4 mtdp + 0x208
0x2395A928 MTDPIO + 0x28C
0x239609C4 CLICON + 0xD50
0x23962350 DBCHCL + 0xC4
```

Si vous trouvez une référence de fonction `OsCall` dans un des fichiers de déroutement, vous devez activer la liaison d'exécution pour la bibliothèque partagée `libcliv2.so`.

Émettez les commandes UNIX suivantes pour activer la liaison d'exécution :


```
cd /usr/lib
rtl_enable libcliv2.so -F libtli.a
mv libcliv2.so libcliv2.so.old
mv libcliv2.so.new libcliv2.so
chmod a+r libcliv2.so
```

Identification et résolution des incidents avec les jeux de caractères pour les sources de données Teradata

Lorsque vous définissez la variable d'environnement TERADATA_CHARSET pour une source de données Teradata, vous risquez de rencontrer des erreurs si le jeu de caractères approprié n'est pas indiqué.

Symptôme

Si le jeu de caractères approprié n'est pas indiqué pour la source de données Teradata, vous rencontrerez l'erreur suivante :

```
SQL 1822N Code d'erreur imprévue "227"
reçu de la source de données "<chaîne>". Les texte et jetons
associés sont "MTDP: EM_CHARNAME(227) : nom jeu de caractères
indiqué invalide". SQLSTATE=560BD
```

Cause

Le jeu de caractères qui est indiqué dans la variable d'environnement TERADATA_CHARSET est incorrect.

Résolution de l'incident

Vérifiez que le jeu de caractères approprié est installé et indiqué sur le serveur Teradata et dans le fichier db2dj.ini.

Identification et résolution des erreurs des opérations UPDATE ou DELETE sur des pseudonymes

Par défaut, les lignes ne sont pas identifiées de manière unique sur les tables des sources de données Teradata. Vous pourriez recevoir un message d'erreur SQL lorsque vous essayez de mettre à jour ou de supprimer un pseudonyme associé à une table ou à une vue Teradata.

Symptôme

L'erreur SQL30090N, RC="21" est renvoyée lorsque vous essayez de mettre à jour ou de supprimer un pseudonyme associé à une table ou à une vue Teradata.

Cause

Ce problème est occasionné parce que des lignes ne sont pas définies de manière unique sur la table source Teradata.

Résolution de l'incident

Pour résoudre l'incident :

1. Supprimez et recréez le pseudonyme.
2. Créez au moins un index à entrées uniques sur la table Teradata à mettre à jour ou à supprimer et réessayez l'opération.

Utilisation de la consignation de l'accès à Teradata

Teradata fournit une fonction de consignation d'accès qui génère des entrées de journal lorsque Teradata vérifie les privilèges de sécurité spécifiques de divers utilisateurs sur une ou plusieurs bases de données. Bien que la consignation d'accès fournisse des informations de sécurité considérables et significatives, cette fonction augmente de façon importante l'utilisation du processeur et peut dégrader les performances du système.

Si vous avez besoin d'améliorer les performances du système, évaluez les règles de contrôle des privilèges que vous avez définies pour la consignation d'accès. Puis arrêtez les règles superflues en définissant des instructions END LOGGING.

Pour obtenir les meilleures performances, désactivez toute consignation d'accès. Supprimez la macro Teradata DBC.AccLogRules, puis imposez une réinitialisation de l'application parallèle digne de confiance (TPA : trusted parallel application) pour interrompre complètement la consignation d'accès.

Voir la documentation Teradata pour de plus amples informations.

Configuration de l'accès aux sources de données des services Web

Pour configurer le système fédéré pour accéder aux sources de données de services Web, vous devez fournir au serveur fédéré des informations sur les sources de données et les objets auxquels vous voulez accéder, par exemple un document WSDL valide.

Avant de commencer

- La fédération doit être installée sur un serveur qui fera office de serveur fédéré.
- Il doit y avoir une base de données sur le serveur fédéré.

Vous pouvez configurer le serveur fédéré pour accéder aux sources de données de services Web à l'aide du Centre de contrôle DB2 ou de la ligne de commande DB2. Le Centre de contrôle DB2 comporte un assistant pour vous guider à travers les étapes nécessaires pour configurer le serveur fédéré.

Procédure

Pour configurer l'accès aux sources de données de services Web :

1. Enregistrez l'encapsuleur de services Web.
2. Enregistrez la définition de serveur pour les sources de données de services Web.
3. Enregistrez des mappages utilisateur pour activer la sécurité pour l'authentification HTTP (facultatif)
4. Enregistrez des pseudonymes pour les sources de données de services Web :
 - Enregistrez des pseudonymes pour les sources de données de services Web à l'aide de la ligne de commande DB2.
 - Enregistrez des pseudonymes pour les sources de données de services Web à l'aide du Centre de contrôle DB2.
5. Créez des vues fédérées pour les pseudonymes de services Web.

Les services Web et l'encapsuleur de services Web

Les fournisseurs de services Web sont décrits par les documents WSDL. Vous pouvez utiliser l'encapsuleur de services Web pour accéder aux fournisseurs de services Web.

Le diagramme figure 9 montre l'architecture des services Web.

1. Un fournisseur de services Web implémente un service et publie les informations WSDL dans un courtier de service, tel qu'UDDI.
2. L'utilisateur du service peut alors utiliser le courtier de service pour rechercher un fournisseur de services Web.
3. Lorsque l'utilisateur du service recherche un fournisseur de services Web, l'utilisateur du service se relie au fournisseur de services pour pouvoir utiliser le service Web.
4. L'utilisateur appelle le service en échangeant des messages SOAP (Simple Object Access Protocol) entre le demandeur et le fournisseur.

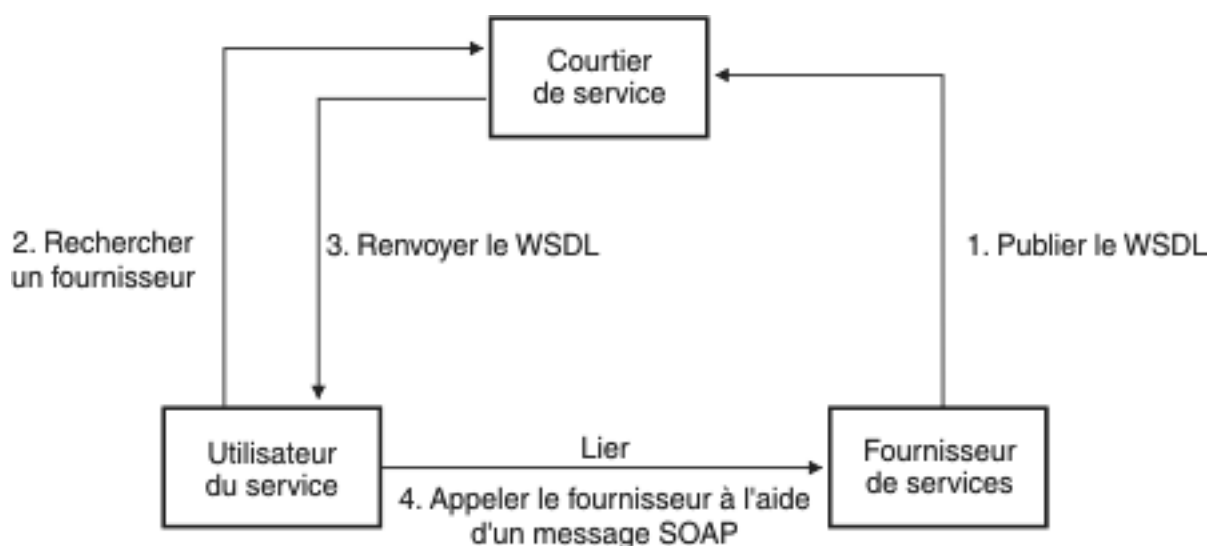


Figure 9. Services Web : une architecture orientée services

La spécification SOAP définit la mise en page d'un message de format XML. Un message SOAP est placé dans une enveloppe SOAP. L'enveloppe se compose d'un en-tête SOAP facultatif et d'un corps SOAP obligatoire. L'en-tête SOAP peut contenir des informations sur le message, par exemple des informations de chiffrement ou des informations d'authentification. Le corps SOAP contient le message. La spécification SOAP définit aussi un codage par défaut pour programmer des liaisons entre langages, qui est appelé codage SOAP.

Le document WSDL et le service Web

La clé du service Web est le document WSDL. Le document WSDL est un document XML qui décrit les services Web en décrivant les messages qu'ils envoient et qu'ils reçoivent. Les messages sont décrits à l'aide d'un système de types, qui est généralement le schéma XML. Une opération de service Web associe un modèle d'échange de messages à un ou plusieurs messages. Un modèle d'échange de messages identifie la séquence et la cardinalité des messages qui sont envoyés ou reçus, ainsi que les destinataires ou expéditeurs logiques des messages. Une interface regroupe des opérations indépendamment du transport ou du format

WF. Une liaison WSDL indique les caractéristiques du transport et du format WF pour une ou plusieurs interfaces. Un noeud final associe une adresse réseau à une liaison. Un service regroupe des noeuds finaux qui implémentent une interface commune. Les messages peuvent contenir des informations orientées document ou des informations orientées processus, que l'on appelle aussi appels de procédure distants (RPC : Remote Procedure Call). Un document WSDL peut contenir un ou plusieurs services Web.

L'exemple dans figure 10, à la page 235 montre la définition WSDL d'un service simple qui fournit des cotations boursières. Le service Web prend en charge une opération unique nommée GetLastTradePrice. Le service est accessible à travers le protocole SOAP 1.1 sur HTTP. La demande lit en entrée un symbole boursier, qui est un type de données chaîne et renvoie le prix, qui est un type de données à virgule flottante. Les types de données chaîne et numérique à virgule flottante sont des types prédéfinis dans les normes du schéma XML. Un service Web peut aussi définir des types de données et utiliser ces types de données définis par l'utilisateur dans les messages. Les types de données XML prédéfinis et définis par l'utilisateur sont mappés aux colonnes des pseudonymes. L'exemple complet et la spécification WSDL figurent sur le site Web du consortium World Wide Web.

```

<?xml version="1.0"?>
<definitions name="StockQuote"
...

<types>
  <schema targetNamespace="http://example.com/stockquote.xsd"
    xmlns="http://www.w3.org/2000/10/XMLSchema">
    <element name="TradePriceRequest">
      <complexType>
        <all>
          <element name="tickerSymbol" type="string"/>
        </all>
      </complexType>
    </element>
    <element name="TradePrice">
      <complexType>
        <all>
          <element name="price" type="float"/>
        </all>
      </complexType>
    </element>
  </schema>
</types>

<message name="GetLastTradePriceInput">
...
</message>

  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetLastTradePriceInput"/>
      <output message="tns:GetLastTradePriceOutput"/>
    </operation>
  </portType>

  <binding name="StockQuoteSoapBinding"
    type="tns:StockQuotePortType">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="GetLastTradePrice">
      <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>

  <service name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteBinding">
      <soap:address location="http://example.com/stockquote"/>
    </port>
  </service>
</definitions>

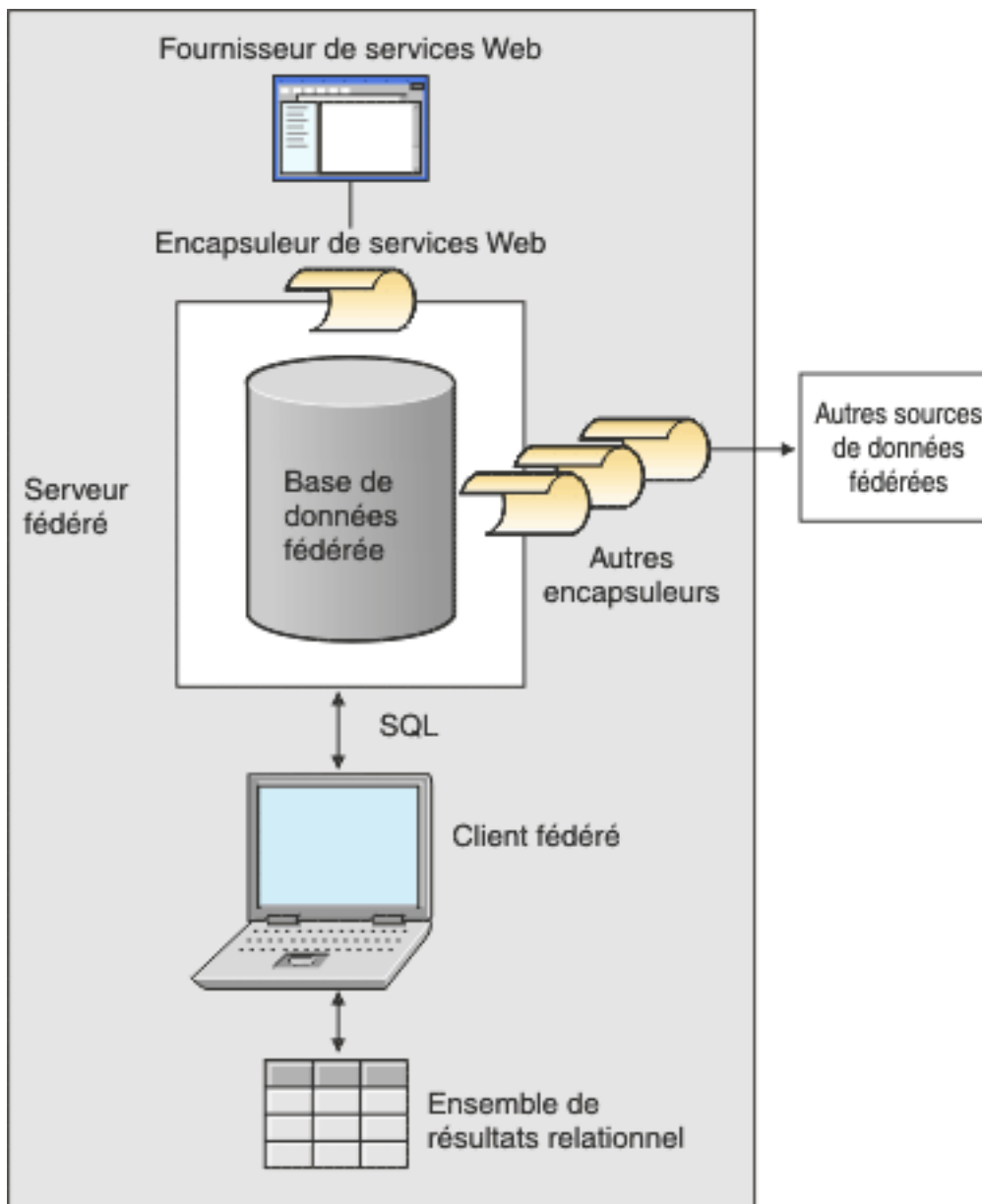
```

Figure 10. Exemple de document WSDL

Le document WSDL, l'encapsuleur de services Web et les pseudonymes

L'encapsuleur de services Web utilise les opérations d'un type de port qui ont une liaison SOAP avec un transport HTTP. Les messages d'entrée dans l'opération et les types ou éléments associés deviennent des colonnes dans le pseudonyme. Les messages de sortie dans l'opération sont extraits vers la hiérarchie des pseudonymes. Vous pouvez créer une hiérarchie séparée de pseudonymes pour chaque opération dans le document WSDL.

Avec l'encapsuleur de services Web, vous pouvez utiliser les fonctions des systèmes fédérés pour associer des données des services Web à des données sur d'autres sources de données fédérées.



L'exemple figure 11, à la page 237 utilise un document WSDL qui contient un portType avec le nom d'opération GETTEMP. Avec ce service Web, vous entrez un

code postal comme entrée et recevez une température pour ce code postal.

```
<?xml version="1.0"?>
<definitions name="TemperatureService" targetNamespace=http://www.xmethods.net/
  sd/TemperatureService.wsdl"
  xmlns:tns="http://www.xmethods.net/sd/TemperatureService.wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <message name="getTempRequest">
    <part name="zipcode" type="xsd:string"/>
  </message>
  <message name="getTempResponse">
    <part name="return" type="xsd:float"/>
  </message>
  <portType name="TemperaturePortType">
    <operation name="getTemp">
      <input message="tns:getTempRequest"/>
      <output message="tns:getTempResponse"/>
    </operation>
  </portType>
  <binding name="TemperatureBinding" type="tns:TemperaturePortType">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="getTemp">
      <soap:operation soapAction="" />
      <input>
        <soap:body use="encoded" namespace="urn:xmethods-Temperature"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </input>
      <output>
        <soap:body use="encoded" namespace="urn:xmethods-Temperature"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </output>
    </operation>
  </binding>
  <service name="TemperatureService">
    <documentation>
      Renvoie la température actuelle pour un code postal américain donné
    </documentation>
    <port name="TemperaturePort" binding="tns:TemperatureBinding">
      <soap:address
        location="http://services.xmethods.net:80/soap/servlet/rpcrouter" />
    </port>
  </service>
</definitions>
```

Figure 11. Service Web GETTEMP

La valeur en entrée est décrite par la colonne code postal du pseudonyme. La valeur de sortie est décrite par la colonne de retour du pseudonyme. Dans le document WSDL, ces colonnes sont identifiées dans l'élément messages. L'élément messages représente la définition logique des données qui sont envoyées entre le fournisseur de services Web et l'utilisateur de services Web. Si des explications supplémentaires sur les informations de l'élément de message sont nécessaires, le document WSDL peut aussi contenir un élément de type. L'élément de type peut se référer à des types prédéfinis reposant sur les spécifications du schéma XML ou à des types qui sont définis par un utilisateur.

L'exemple figure 12, à la page 238 montre le pseudonyme que l'utilitaire DB2[®] Control Center Discovery produit à partir du document WSDL du service Web GETTEMP. La colonne code postal est une colonne d'entrée obligatoire à cause de

la syntaxe TEMPLATE du pseudonyme :

```
CREATE NICKNAME GETTEMP (
  ZIPCODE VARCHAR (48) OPTIONS(TEMPLATE '&column'),
  RETURN VARCHAR (48) OPTIONS(XPATH './return/text()')
)
FOR SERVER "EHPWSSERV"
OPTIONS(URL 'http://services.xmethods.net:80/soap/servlet/rpcrouter',
  SOAPACTION ' ' ,
  TEMPLATE '<soapenv:Envelope>
    <soapenv:Body>
      <ns2:getTemp>
        <zipcode>&zipcode[1,1]</zipcode>
      </ns2:getTemp>
    </soapenv:Body>
  </soapenv:Envelope>',
  XPATH '/soapenv:Envelope/soapenv:Body/*' ,
  NAMESPACES ' ns1="http://www.xmethods.net/sd/TemperatureService.wsdl",
              ns2="urn:xmethods-Temperature" ,
              soapenv="http://schemas.xmlsoap.org/soap/envelope/" ');
```

Figure 12. GETTEMP pseudonyme

Les options de pseudonyme URL et SOAPACTION de l'encapsuleur de services Web permettent de remplacer le noeud final ou l'adresse que vous avez indiquée lorsque vous avez créé le pseudonyme. Lorsque vous utilisez les colonnes URLCOLUMN ou SOAPACTIONCOLUMN activées dans une requête, vous pouvez utiliser des adresses dynamiques avec les mêmes pseudonymes. Si vous définissez les options de pseudonyme URL et SOAPACTION lorsque vous créez un pseudonyme et activez URLCOLUMN et SOAPACTIONCOLUMN dans l'option de colonne, alors vous utilisez les fonctions d'attribution a posteriori des encapsuleurs de services Web. La valeur de l'option de pseudonyme SOAPACTION devient un attribut dans l'en-tête HTTP. La valeur de l'option de pseudonyme URL est l'URL HTTP à laquelle la demande est envoyée.

Les options de pseudonyme URL et SOAPACTION fournissent des associations de pseudonymes dynamiques. Ces adresses dynamiques sont utiles si plusieurs sociétés implémentent un service Web portType. L'encapsuleur de services Web exige que les seules différences entre les documents WSDL soient des URL et SOAPACTIONS différentes. Vous pouvez utiliser la fonction d'attribution a posteriori pour créer et utiliser le même pseudonyme pour des noeuds finaux de service différents que différentes sociétés pourraient vouloir utiliser. Les valeurs URL et SOAPACTION sont dérivées du document WSDL.

L'exemple suivant montre comment vous pouvez utiliser les options de colonne URLCOLUMN et SOAPACTIONCOLUMN :

```
CREATE NICKNAME GetPartQuote(
  partnumber INTEGER OPTIONS (TEMPLATE'&column'),
  price FLOAT OPTIONS (XPATH './price'),
  urlcol VARCHAR(100) OPTIONS (URLCOLUMN 'Y'),
  soapactioncol VARCHAR(100) OPTIONS (SOAPACTIONCOLUMN 'Y'),
FOR SERVER myServer
OPTIONS (
  ...
  SOAPACTION 'http://example.com/GetPartPrice' ,
  URL 'http://mycompany.com:9080/GetPartPrice',
  ...
)
```

Figure 13. Pseudonyme GetPartQuote

L'exemple suivant utilise les colonnes URLCOL et SOAPACTIONCOL qui ont été définies avec l'option de colonne URLCOLUMN activée et l'option de colonne SOAPACTIONCOLUMN activée :

```
SELECT * FROM supplier_endpoints p,
  GetPartQuote q
WHERE partnumber=1234 AND
  p.url=q.urlcol AND
  p.soapaction=q.soapactioncol;
```

L'application SQL peut différer le choix des noeuds finaux à utiliser jusqu'au moment où une requête est exécutée, au lieu de définir un noeud final spécifique au moment où le pseudonyme est créé.

L'encapsuleur de services Web peut diviser un gros volume de données de documents WSDL en fragments pour réduire la mémoire totale utilisée. Indiquez l'option **STREAMING** dans la page Paramètres de la fenêtre Propriétés du Centre de contrôle DB2 lorsque vous créez un pseudonyme de services Web. L'encapsuleur de services Web traite le flux résultant des données XML, puis extrait les informations qui sont demandées par un fragment de requête. L'encapsuleur de services Web fait l'analyse syntaxique d'un fragment à la fois. N'utilisez l'option **STREAMING** que pour faire une analyse syntaxique des grands documents XML.

Enregistrement de l'encapsuleur de services Web

Vous devez enregistrer un encapsuleur pour accéder aux sources de données de services Web. Les encapsuleurs sont utilisés par les serveurs fédérés pour dialoguer avec les sources de données et extraire des données des sources de données. Les encapsuleurs sont mis en oeuvre comme un ensemble de fichiers de bibliothèque.

Avant de commencer

Voir dans la liste des fichiers de bibliothèque de l'encapsuleur de services Web le nom à indiquer dans l'instruction CREATE WRAPPER.

Le nom du fichier de bibliothèque de l'encapsuleur que vous indiquez dépend du système d'exploitation du serveur fédéré.

Procédure

Pour enregistrer un encapsuleur :

Émettez une instruction CREATE WRAPPER avec le nom de l'encapsuleur et le nom du fichier de bibliothèque de l'encapsuleur. Par exemple, pour enregistrer un encapsuleur avec le nom encapsuleur_websr sur un serveur fédéré qui fonctionne sous Windows, émettez l'instruction suivante :

```
CREATE WRAPPER encapsuleur_websr LIBRARY 'db2ws.dll';
```

Fichiers de bibliothèque de l'encapsuleur de services Web

Les fichiers de bibliothèque de l'encapsuleur de services Web sont ajoutés au serveur fédéré lorsque vous installez le serveur fédéré.

Lorsque vous installez IBM InfoSphere Federation Server, les fichiers de bibliothèque sont ajoutés dans le chemin de répertoire par défaut. Par exemple, si le serveur fédéré fonctionne sous AIX, les fichiers de bibliothèque de l'encapsuleur ajoutés dans le chemin de répertoire sont libdb2ws.a, libdb2wsF.a et libdb2wsU.a. Le fichier de bibliothèque de l'encapsuleur par défaut est libdb2ws.a. Les autres fichiers de bibliothèque de l'encapsuleur sont utilisés avec des options spécifiques de l'encapsuleur.

Vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER et indiquer le nom de fichier de bibliothèque de l'encapsuleur par défaut.

Les chemins des répertoires par défaut et les noms de fichier de bibliothèque de l'encapsuleur par défaut sont indiqués dans la table suivante.

Tableau 40. Emplacements et noms de fichier de bibliothèque pour CREATE WRAPPER

Système d'exploitation	Chemin de répertoire	Fichier de bibliothèque d'encapsuleur
AIX	/usr/opt/<chemin_installation>/lib32/ /usr/opt/<chemin_installation>/lib64/	libdb2ws.a
Linux	/opt/IBM/db2/<chemin_installation>/lib32 /opt/IBM/db2/<chemin_installation>/lib64	libsb2ws.so
Solaris	/opt/IBM/db2/<chemin_installation>/lib32 /opt/IBM/db2/<chemin_installation>/lib64	libdb2ws.so
Windows	%DB2PATH%\bin	db2ws.dll

- <chemin_installation> est le chemin de répertoire dans lequel le serveur fédéré est installé sous Linux ou UNIX.
%DB2PATH% est la variable d'environnement qui est utilisée pour indiquer le chemin de répertoire dans lequel le serveur fédéré est installé sous Windows. Le chemin Windows par défaut est C:\Program Files\IBM\SQLLIB.

Enregistrement de la définition de serveur pour des sources de données de services Web

Vous pouvez enregistrer une définition de serveur à l'aide du Centre de contrôle DB2 ou à l'aide de la ligne de commande DB2. Cette tâche décrit comment enregistrer une définition de serveur de services Web depuis la ligne de commande DB2.

Une définition de serveur doit être enregistrée pour chaque service Web auquel vous voulez accéder.

Procédure

Pour enregistrer une définition de serveur sur le système fédéré pour l'encapsuleur de services Web, émettez l'instruction CREATE SERVER.

Par exemple, pour enregistrer une définition de serveur de services Web nommée ws_server sous Windows, émettez l'instruction suivante :

```
CREATE SERVER ws_server WRAPPER encapsuleur_websr;
```

Vous pouvez définir des paramètres de délai d'attente et de serveur proxy facultatifs pour l'instruction CREATE SERVER.

Instruction CREATE SERVER - exemples pour l'encapsuleur de services Web

Utiliser l'instruction CREATE SERVER pour enregistrer des définitions de serveur pour l'encapsuleur de services Web avec des paramètres de délai d'attente et de serveur proxy.

Même si vous n'utilisez pas un serveur proxy pour accéder aux documents des services Web, vous devez toujours enregistrer une définition de serveur. La hiérarchie des objets fédérés exige que les fichiers de services Web soient associés à un objet de définition de serveur spécifique. L'instruction que vous émettez pour enregistrer une définition de serveur est :

```
CREATE SERVER mon_serveur WRAPPER mon_encapsuleur  
    OPTIONS (TIMEOUT '60');
```

mon_serveur

Nom que vous attribuez à la définition de serveur de services Web. Les doublons de noms de définition de serveur ne sont pas autorisés.

WRAPPER mon_encapsuleur

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

TIMEOUT '60'

Indique la durée, en minutes, pendant laquelle le serveur fédéré doit attendre un transfert sur le réseau et le calcul d'un résultat.

Définitions de serveur lorsqu'un serveur proxy est utilisé

Vous devez utiliser les options de serveur proxy dans l'instruction CREATE SERVER si toutes les conditions suivantes sont vraies :

- Vous voulez extraire des données à l'aide d'un identificateur URI.
- L'identificateur URI extrait les données derrière un pare-feu, à travers un serveur proxy.
- Le pare-feu ou serveur proxy est HTTP ou SOCKS.

Les options que vous indiquez dépendent du type de serveur proxy auquel vous voulez accéder.

Vérifiez avec votre administrateur de réseau les informations sur le type de serveur proxy que vous utilisez et les paramètres que vous devez indiquer dans les options du serveur proxy.

Enregistrement d'une définition de serveur pour un serveur proxy HTTP

Pour enregistrer une définition de serveur et indiquer un serveur proxy HTTP, utilisez l'instruction suivante :

```
CREATE SERVER ws_server_http
  WRAPPER ws_wrapper
  OPTIONS (PROXY_TYPE 'HTTP', PROXY_SERVER_NAME 'proxy_http',
    PROXY_SERVER_PORT '8080');
```

ws_server_http

Nom que vous attribuez à la définition de serveur de services Web. Les doublons de noms de définition de serveur ne sont pas autorisés.

WRAPPER *ws_wrapper*

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

PROXY_TYPE '*HTTP*'

Indique le type de serveur proxy qui est utilisé pour accéder à Internet derrière un pare-feu.

PROXY_SERVER_NAME '*proxy_http*'

Indique le nom de serveur proxy ou l'adresse IP.

PROXY_SERVER_PORT '*8080*'

Indique le numéro de port du serveur proxy.

Enregistrement d'une définition de serveur pour un serveur proxy SOCKS avec des informations d'authentification

Pour enregistrer une définition de serveur et indiquer un serveur proxy SOCKS avec des informations d'authentification, émettez l'instruction suivante :

```
CREATE SERVER ws_server_socks
  WRAPPER ws_wrapper
  OPTIONS (PROXY_TYPE 'SOCKS', PROXY_SERVER_NAME 'proxy_socks',
    PROXY_SERVER_PORT '1081', PROXY_AUTHID 'Sampson',
    PROXY_PASSWORD 'sailing4me');
```

PROXY_AUTHID '*Sampson*'

Indique le nom d'utilisateur sur le serveur proxy.

PROXY_PASSWORD '*sailing4me*'

Indique le mot de passe sur le serveur proxy qui est associé au nom d'utilisateur *Sampson*.

Activation de la sécurité à travers l'encapsuleur de services Web

L'encapsuleur de services Web prend en charge l'authentification HTTP à l'aide de l'instruction CREATE USER MAPPING.

L'encapsuleur de services Web prend en charge le protocole de transport HTTP sur SSL pour les messages SOAP. Le document WSDL qui est généré par le fournisseur des services Web contient https:// dans l'URL. Les messages SOAP sont donc chiffrés dans la requête HTTP ou dans la réponse HTTP.

Si le service Web utilise le protocole de transport HTTP sur SSL, vous pouvez configurer l'encapsuleur pour valider les certificats SSL que le serveur envoie pour l'identification, à l'aide de l'encapsuleur ou de l'option de serveur

SSL_VERIFY_SERVER_CERTIFICATE. L'encapsuleur de services Web peut appeler les services Web avec des certificats d'auto-signature.

Le mappage utilisateur est facultatif. Si vous n'indiquez pas de mappage utilisateur, vous risquez de voir une erreur si le fournisseur de services Web prévoit des informations d'authentification. Certains serveurs risquent d'utiliser l'authentification pour restreindre l'accès à un service. Le besoin d'authentification n'est pas apparent d'après les informations dans le document WSDL.

Procédure

Pour mapper un ID utilisateur du serveur fédéré à un ID utilisateur et à un mot de passe de services Web :

Emettez l'instruction CREATE USER MAPPING.

Par exemple, avec l'instruction CREATE USER MAPPING suivante, lors de l'accès au pseudonyme de services Web sur le serveur S1, la requête HTTP est envoyée avec l'ID utilisateur SYSTEM et le mot de passe MANAGER.

```
CREATE USER MAPPING
FOR RSPALTEN SERVER S1
OPTIONS ( REMOTE_AUTHID 'SYSTEM', REMOTE_PASSWORD 'MANAGER'
PROXY_AUTHID 'ID' PROXY_PASSWORD 'PWD'
SSL_CLIENT_CERTIFICATE_LABEL 'LABEL');
```

Enregistrement de pseudonymes pour les sources de données de services Web

Dans chaque définition de serveur de services Web que vous enregistrez, vous devez enregistrer un pseudonyme pour chaque source de données à laquelle vous voulez accéder. Vous pouvez enregistrer un pseudonyme à l'aide de la ligne de commande ou du Centre de contrôle DB2.

Pour enregistrer des pseudonymes pour les sources de données de services Web :

Choisissez une des méthodes suivantes :

- «Enregistrement de pseudonymes pour les sources de données de services Web (ligne de commande DB2)»
- «Enregistrement de pseudonymes pour les sources de données de services Web (Centre de contrôle DB2)», à la page 244

Enregistrement de pseudonymes pour les sources de données de services Web (ligne de commande DB2)

Dans chaque définition de serveur de services Web que vous enregistrez, vous devez enregistrer un pseudonyme pour chaque source de données à laquelle vous voulez accéder. Utilisez ces pseudonymes au lieu des noms de sources de données lorsque vous interrogez les sources de données de services Web.

Avant de commencer

Vous devez avoir accès à un document WSDL valide qui décrit le service Web avec lequel vous voulez dialoguer.

Restrictions

- Seules les opérations de demande-réponse sont prises en charge.
- La seule liaison prise en charge est une liaison SOAP avec un transport HTTP.

- Vous devez utiliser l'option TEMPLATE ou bien l'option XPATH pour chaque colonne, sauf dans les colonnes spéciales avec les options SOAPACTIONCOLUMN, URLCOLUMN, PRIMARY_KEY ou FOREIGN_KEY.

Vous créez une hiérarchie des pseudonymes pour chaque opération de service Web qui est définie dans le document WSDL. Le pseudonyme parent identifie l'enveloppe SOAP; les pseudonymes enfant identifient les éléments du corps SOAP et l'en-tête SOAP facultatif.

Procédure

Pour enregistrer des pseudonymes pour les sources de données de services Web depuis la ligne de commande DB2 :

Émettez une instruction CREATE NICKNAME. Par exemple, pour enregistrer les pseudonymes sous Windows pour un service Web nommé GETTEMP, émettez l'instruction suivante :

```
CREATE NICKNAME GETTEMP (
  ZIPCODE VARCHAR (48) OPTIONS(TEMPLATE '&column'),
  RETURN VARCHAR (48) OPTIONS(XPATH './return/text()')
)
FOR SERVER "EHPWSSERV"
  OPTIONS(URL 'http://services.xmethods.net:80/soap/servlet/rpcrouter',
    SOAPACTION ' ',
    TEMPLATE '<soapenv:Envelope>
      <soapenv:Body>
        <ns2:getTemp>
          <zipcode>&zipcode[1,1]</zipcode>
        </ns2:getTemp>
      </soapenv:Body>
    </soapenv:Envelope>',
    XPATH '/soapenv:Envelope/soapenv:Body/*' ,
    NAMESPACES ' ns1="http://www.xmethods.net/sd/TemperatureService.wsdl",
      ns2="urn:xmethods-Temperature" ,
      soapenv="http://schemas.xmlsoap.org/soap/envelope/" );
```

Enregistrement de pseudonymes pour les sources de données de services Web (Centre de contrôle DB2)

Dans chaque définition de serveur de services Web que vous enregistrez, vous devez enregistrer un pseudonyme pour chaque source de données à laquelle vous voulez accéder. Utilisez ces pseudonymes au lieu des noms de sources de données lorsque vous interrogez les sources de données de services Web.

Avant de commencer

Vous devez avoir accès à un document WSDL valide qui décrit le service Web avec lequel vous voulez dialoguer.

Restrictions

- Seules les opérations de demande-réponse sont prises en charge.
- La seule liaison prise en charge est une liaison SOAP avec un transport HTTP.
- Vous devez utiliser l'option TEMPLATE ou bien l'option XPATH sur chaque colonne, sauf dans les colonnes spéciales avec les options SOAPACTIONCOLUMN, URLCOLUMN, PRIMARY_KEY ou FOREIGN_KEY.

Vous créez une hiérarchie des pseudonymes pour chaque opération de service Web qui est définie dans le document WSDL. Le pseudonyme parent identifie l'enveloppe SOAP; les pseudonymes enfant identifient les éléments du corps SOAP et l'en-tête SOAP facultatif.

Dans le Centre de contrôle DB2, vous pouvez utiliser l'utilitaire Discovery pour créer les pseudonymes. L'entrée pour l'utilitaire Discovery est une URL de l'emplacement d'un document WSDL. L'utilitaire Discovery crée des pseudonymes à la suite du traitement du document WSDL. Le Centre de contrôle DB2 génère des pseudonymes uniques en regroupant les désignations de pièce ou noms d'élément avec le nom de colonne du document WSDL. Le document WSDL peut contenir des définitions de schéma intégrées dans le fichier WSDL ou dans un fichier schéma XML externe qui est importé dans le fichier WSDL. Ces définitions de schéma sont importées à l'aide d'une adresse URL.

Lorsque le Centre de contrôle DB2 génère un pseudonyme enfant qui est uniquement utilisé pour l'entrée, l'instruction XPATH contient un point, comme dans l'exemple suivant : XPATH ' . '

Procédure

Pour enregistrer des pseudonymes pour des sources de données de services Web du Centre de contrôle DB2 :

1. Développez le dossier **Objets de base de données fédérée**.
2. Développez le dossier d'encapsuleur pour lequel vous voulez enregistrer des pseudonymes.
3. Développez le dossier **Définitions de serveur**.
4. Développez le dossier de serveur pour lequel vous voulez enregistrer des pseudonymes.
5. Cliquez avec le bouton droit sur le dossier **Pseudonymes** et sélectionnez **Créer**.
6. Dans la fenêtre Créer des pseudonymes, cliquez sur **Découvrir** pour définir le critère de recherche qui vous aidera à sélectionner les objets dans la source de données.
7. Indiquez le document WSDL qui contient la définition de service Web à laquelle vous voulez que les utilisateurs du système fédéré accèdent. Le document WSDL peut être un document local ou bien vous pouvez indiquer son emplacement à l'aide d'une URL.
8. Cliquez sur le bouton **OK** pour créer le pseudonyme d'après le document WSDL sélectionné.

Le Centre de contrôle DB2 extrait le document WSDL vers plusieurs instructions DDL create-nickname, avec les définitions de relations parent-enfant appropriées. L'instruction DDL (langage de définition de données) que le Centre de contrôle DB2 génère mappe tous les éléments d'entrée aux colonnes du pseudonyme racine dans la hiérarchie des pseudonymes.

Instruction CREATE NICKNAME - exemples pour l'encapsuleur de services Web

Lorsque vous créez un pseudonyme pour accéder à un service Web, vous créez une colonne d'entrée pour chaque valeur dans le message d'entrée d'une opération de service Web et une colonne de sortie pour chaque valeur dans le message de sortie d'une opération de service Web. Vous contrôlez les définitions des colonnes d'entrée et sortie avec les définitions d'options de colonne de pseudonyme.

L'option de colonne TEMPLATE indique qu'une colonne est une colonne d'entrée. L'option de colonne XPATH indique qu'une colonne est une colonne de sortie. Lorsque l'option de pseudonyme TEMPLATE contient une notation entre crochets ([1,1]), la colonne est une colonne d'entrée obligatoire.

L'option de pseudonyme NAMESPACES est une liste séparée par des virgules de paires valeur-nom qu'un système fédéré utilise pour résoudre les espaces de nom pour les éléments des documents XML d'entrée et sortie. Les espaces de nom sont utilisés dans la demande du message de façon à ce que les préfixes soient définis dans l'option de pseudonyme TEMPLATE. L'option de pseudonyme NAMESPACES résout les préfixes en expressions XPath avec les identificateurs URI d'espace de nom qui sont définis dans les schémas WSDL ou XML. Les expressions XPath sont appliquées au document XML qui est renvoyé par le service Web.

Exemple 1 : colonnes d'entrée obligatoires

L'exemple suivant montre un pseudonyme pour un service Web nommé getQuote. Le service Web lit un symbole boursier d'action et renvoie une cotation. L'instruction DDL suivante est créée par l'utilitaire Discovery dans le Centre de contrôle DB2.

```
CREATE NICKNAME "stockquote.stockquoteport_getquote_nn" (
  symbol VARCHAR (48) OPTIONS(TEMPLATE '&column'),
  result VARCHAR (48) OPTIONS(XPATH './Result/text()'))
FOR SERVER "xmethods_server" OPTIONS(
  URL 'http://66.28.98.121:9090/soap' ,
  SOAPACTION 'urn:xmethods-delayed-quotes#getQuote' ,
  TEMPLATE '<soapenv:Envelope>
    <soapenv:Body>
      <ns2:getQuote>
        <symbol>&symbol [1,1]</symbol>
      </ns2:getQuote>
    </soapenv:Body>
  </soapenv:Envelope>',
  XPATH '/soapenv:Envelope/soapenv:Body/*' ,
  NAMESPACES 'ns2="urn:xmethods-delayed-quotes" ,
  ns1="http://www.example.com/wsdl/
  net.xmethods.services.stockquote.StockQuote/" ,
  soapenv="http://schemas.xmlsoap.org/soap/envelope/" ');
```

L'option TEMPLATE du pseudonyme indique la colonne SYMBOL comme colonne d'entrée obligatoire, parce que la colonne contient la désignation [1,1]. Dans l'option TEMPLATE du pseudonyme, l'enveloppe SOAP complète est indiquée pour le service Web. La valeur en entrée getQuote figure dans l'enveloppe SOAP et dans les éléments de corps. L'option de pseudonyme XPATH contient les informations pour trouver la valeur du prix de négociation à travers l'enveloppe SOAP et les balises de corps.

Utilisez le pseudonyme stockquote.stockquoteport_getquote_nn pour accéder au service Web, comme dans la requête suivante :

```
SELECT * FROM "stockquote.stockquoteport_getquote_nn"
  WHERE symbol='IBM';
```

Vous devez utiliser le prédicat symbol='IBM' dans cette instruction, parce que le symbole est une colonne d'entrée obligatoire. Le prédicat d'égalité est le seul prédicat valide dans les colonnes d'entrée. Chacun des prédicats d'égalité définit une valeur dans le message d'entrée. Si la colonne d'entrée est facultative, il n'y a pas besoin d'un prédicat d'égalité dans cette colonne. Si la colonne d'entrée est obligatoire, alors vous devez émettre la requête avec un prédicat d'égalité. Vous

pouvez utiliser une valeur de littéral telle qu'IBM dans une expression d'égalité ou bien une valeur provenant d'une table de jointure ou d'un pseudonyme.

Exemple 2 : éléments et pseudonymes enfant répétitifs

L'exemple suivant utilise un service Web nommé getZooReport qui produit un rapport pour les zoos. La valeur en entrée est un identificateur de zoo. La valeur de sortie est un rapport qui est décrit par le schéma suivant :

```

<wsdl:definitions name="Name"
  targetNamespace="http://myzoo.com"
  ...
<wsdl:types>
  <xsd:schema elementFormDefault="qualified" targetNamespace="http://myzoo.com"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="Animal">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="tns:Name"/>
          <xsd:element ref="tns:Species"/>
          <xsd:element ref="tns:Lot"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="AnimalCareList">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element maxOccurs="unbounded" minOccurs="1" ref="tns:Animal"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="Count" type="xsd:string"/>
    <xsd:element name="LastModified" type="xsd:string"/>
    <xsd:element name="Lot" type="xsd:string"/>
    <xsd:element name="Name" type="xsd:string"/>
    <xsd:element name="NumberOfCages" type="xsd:string"/>
    <xsd:element name="Species" type="xsd:string"/>
    <xsd:element name="Zoo">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="tns:ZooName"/>
          <xsd:element ref="tns:Count"/>
          <xsd:element ref="tns:LastModified"/>
          <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:Zookeeper"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string" use="optional"/>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="ZooName" type="xsd:string"/>
    <xsd:element name="Zookeeper">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="tns:Name"/>
          <xsd:element ref="tns:NumberOfCages"/>
          <xsd:element ref="tns:AnimalCareList"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string" use="optional"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
</wsdl:types>
  ...

```

Figure 14. Service Web *getZooReport*

L'instruction DDL suivante est générée par l'utilitaire Discovery du Centre de contrôle DB2 qui repose sur le WSDL qui contient le schéma :

```

CREATE NICKNAME zooport_getzooreport_nn (
  zoo_id VARCHAR (48) OPTIONS(TEMPLATE '&column'),
  report_zooname VARCHAR (48) OPTIONS(XPATH './ns1:Zoo/@ns1:id'),
  report_count VARCHAR (48) OPTIONS(XPATH './ns1:Zoo/ns1:ZooName/text()'),
  report_lastmodified VARCHAR (48)
    OPTIONS(XPATH './ns1:Zoo/ns1:LastModified/text()'),
  zooport_getzooreport_pkey VARCHAR (16) FOR BIT DATA NOT NULL
    OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER "zooserver" OPTIONS(
  URL 'http://localhost:9080/MaelstromTest/services/ZooPort' ,
  SOAPACTION 'http://myzoo.com/getZooReport' ,
  TEMPLATE '<soapenv:Envelope>
    <soapenv:Body>
      <zooId>&zooId[1,1]</zooId>
    </soapenv:Body>
  </soapenv:Envelope>' ,
  XPATH '/soapenv:Envelope/soapenv:Body' ,
  NAMESPACES ' soapenv="http://schemas.xmlsoap.org/soap/envelope/" ,
              ns1="http://myzoo.com " ');

```

Figure 15. Parent nickname - zooport_getzooreport_nn

```

CREATE NICKNAME zooport_getzooreport_report_nn (
  zooport_getzooreport_fkey VARCHAR (16)
    FOR BIT DATA NOT NULL
    OPTIONS(FOREIGN_KEY 'ZOOPORT_GETZOOREPORT_NN'),
  zookeeper_id VARCHAR (48) OPTIONS(XPATH './ns1:Zookeeper/@ns1:id'),
  report_name VARCHAR (48) OPTIONS(XPATH './ns1:Zookeeper/ns1:Name/text()'),
  report_numberofcages VARCHAR (48)
    OPTIONS(XPATH './ns1:Zookeeper/ns1:NumberOfCages/text()'),
  zooport_getzooreport_pkey VARCHAR (16)
    FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER "zooserver" OPTIONS(
  XPATH './ns1:Zoo' ,
  NAMESPACES ' soapenv="http://schemas.xmlsoap.org/soap/envelope/" ,
              ns1="http://myzoo.com" ');

```

Figure 16. Enfant du pseudonyme zooport_getzooreport_nn

```

CREATE NICKNAME zooport_getzooreport_report_report_nn (
  zooport_getzooreport_fkey VARCHAR (16) FOR BIT DATA NOT NULL
    OPTIONS(FOREIGN_KEY 'zooport_getzooreport_report_nn'),
  report_name VARCHAR (48)
    OPTIONS(XPATH './ns1:Animal/ns1:Name/text()'),
  report_species VARCHAR (48)
    OPTIONS(XPATH './ns1:Animal/ns1:Species/text()'),
  report_lot VARCHAR (48) OPTIONS(XPATH './ns1:Animal/ns1:Lot/text()'))
FOR SERVER "zooserver" OPTIONS(
  XPATH './ns1:Zookeeper/ns1:AnimalCareList' ,
  NAMESPACES ' soapenv="http://schemas.xmlsoap.org/soap/envelope/" ,
              ns1="http://myzoo.com" ');

```

Figure 17. Enfant de zooport_getzooreport_report_nn

Le schéma comporte certains éléments qui sont répétés, que l'on appelle éléments de séquence. Ces éléments répétés deviennent des pseudonymes enfant du pseudonyme parent, comme le montrent figure 15, figure 16 et figure 17. Par

exemple, zooname, count, lastmodified et zookeeper sont tous des éléments de zoo. L'élément zoo contient 0 ou plus éléments zookeeper. Le pseudonyme racine, zoo, contient les colonnes zooname, count et lastmodified. Un pseudonyme enfant, zookeeper, est créé par l'utilitaire Discovery du Centre de contrôle DB2 pour décrire les éléments répétés de zookeeper. Le troisième élément de la colonne zookeeper, animalcarelist, contient aussi 0 ou plus éléments et devient ainsi un pseudonyme enfant, zooport_getzooreport_report_report_nn. La figure suivante montre la hiérarchie des pseudonymes :

Pseudonyme racine : zooport_getzooreport_nn

Zoo (parent) :

- ZooName
- Count
- LastModified
- ZooKeeper (il y a 0 ou plus éléments ZooKeeper)

Pseudonyme enfant : zooport_getzooreport_report_nn

Éléments ZooKeeper

- Nom
- NumberOfCages
- AnimalCareList (il y a 0 ou plus éléments Animal)

Pseudonyme enfant : zooport_getzooreport_report_report_nn

Animal

- Nom
- Species
- Lot

Figure 18. Hiérarchies de pseudonymes parent -> enfant ->

L'instruction suivante est un exemple de requête que vous pourriez émettre sur les pseudonymes pour accéder au service Web de rapport du zoo. Lorsque vous émettez cette instruction, vous extrayez les informations du rapport du zoo d'après un identificateur spécifique et d'après les emplacements où les clés primaires et externes des rapports de zoo de pseudonymes enfant correspondent.

```
SELECT * FROM zooport_getzooreport_nn ,
        zooport_getzooreport_report_nn zk ,
        zooport_getzooreport_report_report_nn a
WHERE zoid='1'AND zooport_getzooreport_pkey=zk.zooport_getzooreport_fkey
and zk.zooport_getzooreport_pkey=a.zooport_getzooreport_fkey;
```

Exemple 3 : attribution a posteriori

L'exemple suivant montre comment vous pouvez utiliser l'option d'attribution a posteriori. Vous pouvez utiliser cette option à partir du Centre de contrôle DB2 ou d'une ligne de commande DB2. Si vous définissez les options de pseudonyme URL et SOAPACTION et que vous activez les options de colonne URLCOLUMN et SOAPACTIONCOLUMN lorsque vous créez un pseudonyme, vous utilisez les fonctions d'attribution a posteriori. Le Centre de contrôle DB2 crée les options de colonne URLCOLUMN et SOAPACTIONCOLUMN et définit les valeurs des colonnes sur yes.

L'exemple suivant concerne un service Web qui fournit des tarifs pour des pièces qui sont réalisées par tous les fournisseurs d'une société. Voici l'instruction CREATE NICKNAME qui comporte les définitions URLCOLUMN et SOAPACTIONCOLUMN :

```
CREATE NICKNAME GetPartQuote(
  partnumber INTEGER OPTIONS (TEMPLATE '&column'),
  price FLOAT OPTIONS (XPATH './price'),
  urlcol VARCHAR(100) OPTIONS (URLCOLUMN 'Y'),
  soapactioncol VARCHAR(100) OPTIONS (SOAPACTIONCOLUMN 'Y'),
  FOR SERVER myServer
  OPTIONS (
    ...
    SOAPACTION 'http://example.com/GetPartPrice' ,
    URL 'http://mycompany.com:9080/GetPartPrice'',
    ...
  )
```

Pour obtenir les tarifs de tous les fournisseurs avec une seule requête, il faut disposer des valeurs que chaque fournisseur utilise pour les options de colonne SOAPACTION et URL. La requête ressemble à ceci :

```
SELECT * FROM supplier_endpoints p,
  GetPartQuote q
  WHERE partnumber=1234 AND
        p.url=q.urlcol AND
        p.soapaction=q.soapactioncol;
```

La table locale supplier_endpoints contient toutes les URL et adresses SOAP avec lesquelles vous pouvez appeler le service Web. Vous pouvez inclure une clause "ORDER BY price" pour déterminer le fournisseur le moins cher pour cette pièce.

Exemple 4 : option de colonne ESCAPE_INPUT

Vous pouvez inclure des fragments XML comme valeurs en entrée dans votre requête. Lorsque vous enregistrez un pseudonyme, incluez l'option de colonne ESCAPE_INPUT=N. Cette option gère les caractères spéciaux, tels que les signes supérieur ou inférieur (< et >) dans les fragments XML des valeurs en entrée.

Lorsqu'un schéma contient des valeurs en entrée répétées qui exigent que vous envoyiez XML dans le cadre du message SOAP, vous pouvez utiliser l'option de colonne ESCAPE_INPUT pour construire le message de sortie avec le XML approprié.

Par exemple, le service Web du zoo comporte une opération pour ajouter un nouveau gardien de zoo et les animaux associés à ce gardien de zoo. Dans le schéma pour cet exemple, une liste AnimalCareList peut avoir plusieurs animaux.

```
CREATE NICKNAME add_zookeeper(
  zookeeper_id VARCHAR(48) OPTIONS(TEMPLATE '...'),
  name VARCHAR(48) OPTIONS(TEMPLATE '...'),
  numberofcages VARCHAR(48) OPTIONS(TEMPLATE '...'),
  animals VARCHAR(3000) OPTIONS( TEMPLATE '...' , ESCAPE_INPUT 'N')
  ...
```

Pour ajouter un nouveau gardien de zoo avec deux animaux, émettez une requête telle que l'exemple suivant :

```
SELECT * FROM add_zookeeper
  WHERE zookeeper_ID='37' AND
        name='Amit Kapoor' AND
        numberofcages='3' AND
        animals='<AnimalCareList xmlns="http://myzoo.com">
```

```

<Animal>
  <Name>Larry</Name>
  <Species>Gorilla</Species>
  <Lot>7</Lot>
</Animal>
<Animal>
  <Name>Bill</Name>
  <Species>Chimpanzee</Species>
  <Lot>8H</Lot>
</Animal>
</AnimalCareList>;

```

Le pseudonyme `add_zookeeper` est une opération de service Web qui peut modifier l'état du service Web ou mettre à jour des informations. Bien que les encapsuleurs non relationnels ne puissent pas être mis à jour, l'instruction `SELECT` dans cet exemple met à jour les informations du zoo pour ajouter un gardien de zoo.

Vous pouvez aussi utiliser l'option de colonne `ESCAPE_INPUT` pour un schéma avec un élément tel que `xsd:anyType`. Dans ce cas, le type d'élément est inconnu. Vous pouvez utiliser l'option de colonne `ESCAPE_INPUT` dans la colonne d'entrée de cet élément, afin de pouvoir indiquer des fragments XML arbitraires pour votre entrée.

Accès à un en-tête SOAP

Utilisez l'encapsuleur de service Web pour accéder aux valeurs d'en-tête SOAP (Simple Object Access Protocol) incorporées dans un message SOAP.

Un message SOAP, que ce soit une demande ou une réponse, se compose d'une enveloppe qui contient les espaces de nom qui sont utilisés par le reste du message SOAP, un en-tête facultatif et un corps. Un en-tête SOAP contient généralement des métadonnées sur le message SOAP ou d'autres informations facultatives. Le message SOAP se déplace d'une source vers une destination en passant à travers une série de services intermédiaires qui traitent le message ou ses effets secondaires. Par exemple, un message peut passer à travers un service de transaction qui fournit au client un appel garanti si le réseau échoue ou bien une fonction de sécurité figurant sur un portail d'entreprise peut fournir des informations d'authentification. Ces intermédiaires utilisent souvent les métadonnées figurant dans l'en-tête SOAP.

Les en-têtes SOAP contiennent souvent des informations facultatives ou prennent en charge des interfaces utilisateur évolutives. Par exemple, une banque permettra à ses clients d'utiliser des cartes bancaires pour retirer de l'argent sur leur compte courant. Cependant, certains clients ont plusieurs comptes. Si un client a plusieurs comptes, la banque doit pouvoir identifier le compte sur lequel le prélèvement est effectué.

A partir de cet exemple, voici l'élément d'en-tête qui indique un compte bancaire :

```
<xsd:element name="AccountID" type="xsd:int"/>
```

Voici un message SOAP qui utilise cet élément d'en-tête. En l'occurrence, le client a choisi de retirer 1000 dollars sur le compte 1 et doit des frais de transaction de 50 cents.

```

<SOAP-ENV:Envelope>
  <SOAP-ENV:Header>
    <tns:AccountID>1</tns:AccountID>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>

```

```

        <tns:Action>withdrawal</tns:Action>
        <tns:Amount>1000</tns:Amount>
        <tns:Fee>0.50</tns:Fee>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Les en-têtes SOAP ont les exigences suivantes :

- soapenv:Header est un élément facultatif sous soapenv:Envelope.
- S'il est utilisé, l'en-tête SOAP doit être le premier élément.
- Tous les enfants sous soapenv:Header doivent être qualifiés par un espace de nom.
- soapenv:Header peut avoir les attributs suivants : actor, mustUnderstand, et encoding style.

La hiérarchie des pseudonymes pour accéder aux en-têtes SOAP exige que le pseudonyme racine pointe sur l'enveloppe SOAP. Les éléments d'en-tête et de corps sont des pseudonymes enfant du pseudonyme racine. Vous pouvez créer les pseudonymes vous-même ou utiliser le Centre de contrôle pour générer les pseudonymes automatiquement. Vous pouvez ensuite créer des requêtes qui accèdent aux informations qui sont stockées dans l'en-tête SOAP.

Exemple

Les instructions CREATE NICKNAME suivantes illustrent comment créer les pseudonymes racine, de corps et d'en-tête.

```

CREATE NICKNAME STOCKSERVICE_GETLASTSELLPRICE_ROOT_NN
    (NN_PKEY VARCHAR(16) NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER "WSSERVER_HDR"
OPTIONS
    (URL 'http://wswrap.svl.ibm.com:9081/StockPrice/services/StockService' ,
    SOAPACTION ' ' , TEMPLATE
    '<soapenv:Envelope>&STOCKSERVICE_GETLASTSELLPRICE_HEADER_NN[1,1]
    &STOCKSERVICE_GETLASTSELLPRICE_BODY_NN[1,1]</soapenv:Envelope>', XPATH
    '/soapenv:Envelope',
    NAMESPACES 'soapenv="http://schemas.xmlsoap.org/soap/envelope/",
    ns1="http://soapheader.ibm.com");

CREATE NICKNAME STOCKSERVICE_GETLASTSELLPRICE_HEADER_NN
    (NN_PKEY VARCHAR(16) NOT NULL OPTIONS(PRIMARY_KEY 'YES'),
    NN_FKEY VARCHAR(16) NOT NULL OPTIONS(FOREIGN_KEY
    'STOCKSERVICE_GETLASTSELLPRICE_ROOT_NN'),
    QUOTE_TIMESTAMP VARCHAR(30) OPTIONS(TEMPLATE '&column'))
FOR SERVER "WSSERVER_HDR"
OPTIONS
    (XPATH './soapenv:Header', TEMPLATE '<soapenv:Header>
    <ns1:quote_timestamp>&quote_timestamp[1,1]
    </ns1:quote_timestamp></soapenv:Header>',
    NAMESPACES 'ns1="http://soapheader.ibm.com", soapenv=
    "http://schemas.xmlsoap.org/soap/envelope/" ');

CREATE NICKNAME STOCKSERVICE_GETLASTSELLPRICE_BODY_NN
    (NN_PKEY VARCHAR(16) NOT NULL
    OPTIONS(PRIMARY_KEY 'YES'), NN_FKEY VARCHAR(16) NOT NULL
    OPTIONS(FOREIGN_KEY 'STOCKSERVICE_GETLASTSELLPRICE_ROOT_NN'),
    GETLASTSELLPRICE_TICKER VARCHAR(48)
    OPTIONS(TEMPLATE '&column'), GETLASTSELLPRICEReturn VARCHAR(48)
    OPTIONS(XPATH
    './ns1:getLastSellPriceResponse/ns1:getLastSellPriceReturn/text()'))
FOR SERVER "WSSERVER_HDR"
OPTIONS
    (TEMPLATE '<soapenv:Body><ns1:getLastSellPrice>
    <ns1:ticker>&getLastSellPrice_ticker[1,1]

```

```

</ns1:ticker></ns1:getLastSellPrice></soapenv:Body>',
XPath './soapenv:Body',
NAMESPACES ' soapenv="http://schemas.xmlsoap.org/soap/envelope/" ,
ns1="http://soapheader.ibm.com" ');

```

La requête suivante renvoie le prix de l'action IBM pour une date et une heure spécifiques :

```

SELECT * from
  STOCKSERVICE_GETLASTSELLPRICE_ROOT_NN as a,
  STOCKSERVICE_GETLASTSELLPRICE_HEADER_NN as b,
  STOCKSERVICE_GETLASTSELLPRICE_BODY_NN as c
where
  a.nn_pkey = b.nn_fkey and
  a.nn_pkey = c.nn_fkey and
  b.QUOTE_TIMESTAMP='2007-05-09T18:30:20.000Z' and
  c.GETLASTSELLPRICE_TICKER='IBM';

```

Pseudonymes et expressions XPath

L'encapsuleur de services Web utilise des pseudonymes qui correspondent à la structure arborescente des données XML. Chaque pseudonyme est défini par des expressions XPath qui représentent des valeurs de sortie.

Les pseudonymes correspondent à la structure arborescente de vos données de documents XML. Les pseudonymes parents et les pseudonymes enfant correspondent à la structure racine et aux éléments imbriqués de la structure arborescente de données. Ces pseudonymes parent et enfant sont connectés par des clés primaires et externes qui sont indiquées avec l'instruction CREATE NICKNAME.

Chaque pseudonyme est défini par des expressions XPath qui représentent des valeurs de sortie. L'encapsuleur de services Web utilise des expressions XPath pour établir une correspondance entre les données d'un document XML et les lignes d'une table relationnelle. Ces expressions XPath identifient les valeurs dans le document XML et déterminent comment ces valeurs correspondent aux colonnes de chaque ligne. L'encapsuleur lit uniquement les données XML, sans mettre à jour le document. L'option XPath contient les informations pour rechercher les messages SOAP à travers l'enveloppe SOAP et les balises du corps SOAP. Le message getQuote figure dans l'enveloppe SOAP et les éléments de corps.

L'expression XPath pour l'option NICKNAME pointe vers des balises répétitives qui sont dans l'élément de sortie. L'expression XPath détermine combien ou quelles lignes seront dans le pseudonyme. L'expression XPath de l'option de colonne se réfère à l'expression XPath NICKNAME. L'option de colonne XPath identifie les valeurs dans une ligne. Une option NICKNAME XPath dans un pseudonyme enfant se réfère à une expression XPath de l'option NICKNAME dans un pseudonyme parent.

Lorsque vous créez un pseudonyme, vous choisissez des options dans un document WSDL qui indique l'association entre le pseudonyme et le document XML.

Option TEMPLATE aux niveaux du pseudonyme et de la colonne

Lorsqu'un pseudonyme est créé, l'encapsuleur de services Web exige les fragments de modèle au niveau du pseudonyme et au niveau de la colonne, c'est-à-dire l'option TEMPLATE dans l'instruction CREATE NICKNAME.

L'encapsuleur de services Web génère des documents XML qui sont attendus par l'environnement de services Web. L'encapsuleur utilise ces informations pendant les phases de planification et d'exécution de la requête.

L'option **TEMPLATE** pour l'encapsuleur de services Web :

L'encapsuleur de services Web exige l'option **TEMPLATE** dans l'instruction **CREATE NICKNAME** lorsque le pseudonyme est créé.

Encapsuleur de services Web

Pour l'encapsuleur de services Web, les attributs obligatoires et facultatifs dépendent des définitions du document WSDL et de la façon dont une colonne est dérivée. Une colonne peut être dérivée d'un élément ou d'un attribut d'un élément.

- Si la colonne est dérivée d'un élément, la valeur `minOccurs` détermine si une colonne est facultative.
- Si la valeur de `minOccurs` égale 0, la colonne est facultative.
- Si la valeur de `minOccurs` égale 1, la colonne est obligatoire.
- Si la colonne est dérivée d'un attribut d'un élément, alors la valeur utilisée sur l'attribut détermine si une colonne est facultative.
- Si un attribut contient la valeur `use=optional`, la colonne est facultative.
- Si un attribut contient la valeur `use=required`, la colonne est obligatoire.

L'exemple suivant est un attribut, dans une définition de schéma, qui est associé à une colonne :

```
<xsd:complexType>
  <xsd:sequence>
    <xsd:element ref="tns:ZooName" />
    <xsd:element ref="tns:Count" />
    <xsd:element ref="tns:LastModified" />
    <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:Zookeeper" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string" use="optional" />
</xsd:complexType>
```

Création de vues fédérées pour les pseudonymes de services Web

Créer des vues fédérées pour les pseudonymes de services Web afin de garantir que les requêtes joignant des éléments d'une hiérarchie de pseudonymes de services Web s'exécutent correctement.

Vous pouvez définir des vues fédérées pour la hiérarchie de pseudonymes qui décrit un document de services Web. Une vue fédérée garantit que les requêtes qui joignent des éléments d'une hiérarchie de pseudonymes de services Web puissent s'exécuter correctement.

L'encapsuleur de services Web exige que les jointures de tout pseudonyme enfant comportent tous les pseudonymes en amont jusqu'au pseudonyme parent. Une instruction **SELECT** sur un seul pseudonyme enfant échoue si le pseudonyme parent n'est pas inclus dans l'instruction. La méthode recommandée pour inclure tous les pseudonymes obligatoires dans une requête consiste à définir une vue fédérée.

Procédure

Pour créer des vues fédérées pour les pseudonymes de services Web :

1. Définissez une vue incluant tous les pseudonymes qui sont associés à une opération dans le service Web si vous voulez joindre ces pseudonymes.
2. Dans la clause WHERE de la vue, utilisez des prédicats de jointure pour toutes les colonnes associées avec les options de colonne PRIMARY_KEY et FOREIGN_KEY.

Dans l'exemple suivant, la clé primaire se trouve sur la colonne ooport_getzooreport_pk dans le pseudonyme zooport_getzooreport_report_nn. La clé externe se trouve sur la colonne ooport_getzooreport_fkey dans le pseudonyme zooport_getzooreport_report_report_nn.

```
CREATE VIEW zooreport
(zooïd, zooname, number_of_zookeeper,
lastmodified,zookeeper_id, zookeeper_name,
cages, animal_name, animal_species, animal_lot)
AS ( SELECT zooïd, report_zooname,
report_count, report_lastmodified,
zookeeper_id, zk.report_name, report_numbercages,
a.report_name, report_species,
report_lot
FROM zooport_getzooreport_nn ,
zooport_getzooreport_report_nn as zk,
zooport_getzooreport_report_report_nn as a
WHERE zk.ooport_getzooreport_pkey=a.ooport_getzooreport_fkey
AND zooport_getzooreport_pkey=zk.ooport_getzooreport_fkey);
```

Vous pouvez obtenir des informations de tous les pseudonymes avec l'instruction SELECT suivante :

```
SELECT * FROM zooreport WHERE zooïd='1';
```

Restrictions de requête pour les encapsuleurs de services Web

L'utilisation de prédicats et de jointures de requête est restreinte pour les services Web comme le montrent les exemples fournis.

Prédicats d'égalité

Les seuls prédicats valides dans les colonnes d'entrée sont les prédicats d'égalité. Pour les colonnes de sortie, aucun prédicat n'est valide.

L'exemple suivant renvoie une erreur avec un message qui indique que le prédicat n'est pas pris en charge pour cette colonne. Dans cet exemple, la colonne code postal (zipcode) est une colonne d'entrée :

```
SELECT return FROM gettemp WHERE zipcode<'95141'
```

L'exemple suivant montre une requête valide utilisant un prédicat d'égalité dans les colonnes d'entrée. Le pseudonyme customers (clients) est associé à une table locale qui contient les ID client. La requête contient un prédicat supplémentaire dans la colonne Sales (ventes), qui est une colonne de sortie uniquement.

```
SELECT a.name, a.address
FROM customers a, local_table b
WHERE
a.customer_id=b.custid AND
a.Sales > 300000;
```

Prédicats des colonnes d'entrée obligatoires

Vous devez fournir des valeurs de prédicat d'égalité pour toutes les colonnes d'entrée obligatoires dans vos requêtes SQL pour la hiérarchie des pseudonymes que vous référencez. L'encapsuleur renvoie un code SQL -901 pour toutes les requêtes qui ne respectent pas cette restriction.

Prédicats IN ou OR

Avec les encapsuleurs de services Web, aucun prédicat IN ou OR n'est autorisé pour les colonnes d'entrée.

Les exemples suivants montrent des requêtes non valides. Le pseudonyme customers a une colonne d'entrée obligatoire, customer_id :

```
SELECT * FROM customers
  WHERE customer_id IN (12345, 67890, 11223);
SELECT * FROM customers
  WHERE customer_id IN (SELECT custid FROM local_table); )
```

Jointures dans les colonnes d'entrée facultatives

Les exemples suivants illustrent une restriction pour la jointure des colonnes d'entrée facultatives et l'utilisation de variables hôte. Vous ne pouvez pas joindre des colonnes d'entrée facultatives d'une table ou d'un pseudonyme local. Si WSDL génère une colonne de pseudonymes d'entrée comme colonne facultative et que vous avez besoin d'utiliser cette colonne dans une jointure, alors vous devez éditer l'instruction DDL pour modifier la colonne en une colonne d'entrée obligatoire.

Dans cet exemple, un ordre nommé de pseudonyme de l'encapsuleur de services Web est créé avec shipping_method comme colonne d'entrée facultative. L'instruction suivante est une requête valide parce qu'elle utilise un littéral dans le prédicat :

```
SELECT * FROM order
  WHERE part="hammer" AND shipping_method="FEDEX";
```

Cependant, si vous incluez une table locale nommée orderparts, qui définit des pièces et des méthodes de livraison, dans la requête et que la table contient une colonne nommée shipping_method qui est facultative, l'instruction n'est pas valide :

```
SELECT * FROM
  order o, orderparts op
  WHERE
    o.part="hammer" AND
    o.shipping_method=op.shipping_method
```

Si vous utilisez des variables hôte dans des requêtes avec des prédicats pour des colonnes d'entrée facultatives, l'instruction n'est pas valide non plus :

```
SELECT * FROM
  order o
  WHERE
    o.part="hammer" AND
    o.shipping_method=:ashipping_method
```

Pour obtenir des résultats valides, les colonnes d'entrée jointes doivent être des colonnes obligatoires pour les encapsuleurs de services Web.

Jointures externes

Les jointures externes entre des pseudonymes utilisant la clé primaire d'un pseudonyme parent et la clé externe de colonnes de pseudonymes enfant ne sont pas prises en charge.

Lorsqu'un élément parent dans un document XML ne contient aucun élément enfant et que vous utilisez une jointure interne entre le pseudonyme parent et le pseudonyme enfant, alors aucune ligne n'est renvoyée pour cet élément. Par exemple, pour un client donné, si les coordonnées bancaires ne figurent pas dans le système SAP, alors aucune ligne n'est renvoyée pour le pseudonyme `sap_bapi_customer_getdetail2_sap_customerbankdetail_NN` du client concerné.

Les instructions CREATE NICKNAME suivantes définissent les colonnes qui sont utilisées dans l'exemple de requête :

```
CREATE NICKNAME sap_bapi_customer_getdetail2_NN(
...
  NAME VARCHAR(35)
    OPTIONS(XPATH './ns3:sap_customeraddress/
      ns1:sap_customeraddress/ns1:NAME/text()'),
...
  NN_PKEY VARCHAR(16) OPTIONS(PRIMARY_KEY 'YES'),
  COMPANYCODE VARCHAR(4) OPTIONS(XPATH './ns3:COMPANYCODE/text()',
    TEMPLATE '<ns3:COMPANYCODE>&column</ns3:COMPANYCODE>'),
  CUSTOMERNO VARCHAR(10) OPTIONS(XPATH './ns3:CUSTOMERNO/text()',
    TEMPLATE '<ns3:CUSTOMERNO>&column</ns3:CUSTOMERNO>'),
...
  FOR SERVER sap_server
  OPTIONS(XPATH '//ns3:sap_bapi_customer_getdetail2',
    TEMPLATE '<ns3:sap_bapi_customer_getdetail2>
      &sap_bapi_customer_getdetail2_sap_customerbankdetail_NN[0,1]
      &COMPANYCODE[0,1]
      &CUSTOMERNO[1,1]
    </ns3:sap_bapi_customer_getdetail2>',
...
)
```

Figure 19. Extrait du pseudonyme `getdetail2`

```

CREATE NICKNAME sap_bapi_customer_getdetail2_sap_customerbankdetail_NN(
  CUSTOMER VARCHAR(10) OPTIONS(XPATH './ns2:CUSTOMER/text()'),
    TEMPLATE '<ns2:CUSTOMER>&column</ns2:CUSTOMER>'),
  BANK_KEY VARCHAR(15) OPTIONS(XPATH './ns2:BANK_KEY/text()'),
    TEMPLATE '<ns2:BANK_KEY>&column</ns2:BANK_KEY>'),
  BANK_ACCT VARCHAR(18) OPTIONS(XPATH './ns2:BANK_ACCT/text()'),
    TEMPLATE '<ns2:BANK_ACCT>&column</ns2:BANK_ACCT>'),
  CTRL_KEY VARCHAR(2) OPTIONS(XPATH './ns2:CTRL_KEY/text()'),
    TEMPLATE '<ns2:CTRL_KEY>&column</ns2:CTRL_KEY>'),
  BANK_REF VARCHAR(20) OPTIONS(XPATH './ns2:BANK_REF/text()'),
    TEMPLATE '<ns2:BANK_REF>&column</ns2:BANK_REF>'),
  NN_FKEY VARCHAR(16) OPTIONS(FOREIGN_KEY 'SAP_BAPI_CUSTOMER_GETDETAIL2_NN'))
FOR SERVER sap_server
OPTIONS(XPATH './ns3:sap_customerbankdetail/ns2:sap_customerbankdetail',
  TEMPLATE '<ns3:sap_customerbankdetail>
    <ns2:sap_customerbankdetail>
      &CUSTOMER[0,1]
      &BANK_KEY[0,1]
      &BANK_ACCT[0,1]
      &CTRL_KEY[0,1]
      &BANK_REF[0,1]
    </ns2:sap_customerbankdetail>
  </ns3:sap_customerbankdetail>',
  ...

```

Figure 20. Extrait du pseudonyme customer bank detail (coordonnées bancaires du client)

Dans l'exemple suivant, la requête ne renvoie aucune ligne parce qu'il y a une condition de jonction interne entre les deux pseudonymes :

```

SELECT a.name, b.bank_key
  FROM sap_bapi_customer_getdetail2_NN a,
       sap_bapi_customer_getdetail2_sap_customerbankdetail_NN b
 WHERE a.customerno='1234567890'
       AND a.NN_PKEY=b.NN_FKEY;

```

Si une définition de pseudonyme de l'encapsuleur de services Web contient des colonnes d'entrée obligatoires, alors une jointure externe gauche entre ce pseudonyme et une autre table locale ou d'autres pseudonymes n'est pas prise en charge.

Sources de données de services Web - exemples de requêtes

Des exemples de requêtes SQL pour l'encapsuleur de services Web sont affichés.

Exemple 1 : utilisation de tables de requêtes matérialisées

Vous utilisez des tables de requêtes matérialisées pour placer localement en mémoire cache les résultats d'une requête et pour améliorer les performances des requêtes. Vous pouvez utiliser des pseudonymes de sources de données de services Web pour créer des tables de requêtes matérialisées. Pour certaines requêtes, la base de données peut déterminer automatiquement si la table de requêtes matérialisée peut répondre à une requête sans accéder aux tables de base. La procédure suivante montre comment créer et remplir une table de requêtes matérialisée :

1. Créez une table locale ou de base :

```
CREATE TABLE mystocks(ticker VARCHAR(10));
```

Vous pouvez utiliser la table locale pour gérer toutes les valeurs que vous voulez mettre en cache.

2. Insérez toutes les valeurs que vous voulez mettre en cache dans la table :

```
INSERT INTO mystocks VALUES('IBM');
INSERT INTO mystocks VALUES('MSFT');
...
```

3. Créez un pseudonyme de services Web :

```
CREATE NICKNAME stockquote_nn (
    ticker VARCHAR(40) OPTIONS (TEMPLATE '&column'),
    price VARCHAR(16) OPTIONS (XPATH './Result/text()')
)
FOR SERVER stock_server
OPTIONS (TEMPLATE '<ticker>&column</ticker>'
        XPATH './Result/text()');
```

4. Créez une vue qui se compose du pseudonyme et de la table locale :

```
CREATE VIEW stock_quote_view (ticker, price)
AS (
    SELECT nn.ticker, nn.price
    FROM stockquote_nn nn, mystocks s
    WHERE nn.ticker=s.ticker
);
```

5. Créez une table de requêtes matérialisée :

```
CREATE TABLE stockquote_MQT (ticker, ticker2, price)
as (SELECT nn.ticker,s.ticker as ticker2, nn.price
FROM stockquote_nn nn, mystocks s
WHERE nn.ticker=s.ticker )
DATA INITIALLY DEFERRED REFRESH DEFERRED;
```

Incluez toutes les colonnes VARCHAR dans le prédicat de jointure (nn.ticker et s.ticker) de la liste de sortie de la table de requêtes matérialisée pour maximiser les chances que la table de requêtes matérialisée soit utilisée par la base de données fédérée.

Pour différer la régénération de la table de requêtes matérialisée, indiquez le mot clé REFRESH DEFERRED. Les tables de requêtes matérialisées qui sont indiquées avec le mot clé REFRESH DEFERRED ne reflètent pas de modifications de la table de base sous-jacente. Utilisez la clause DATA INITIALLY DEFERRED pour que vos données ne soient pas insérées dans la table dans le cadre de l'instruction CREATE TABLE.

6. Emettez une instruction REFRESH TABLE pour remplir la table. Les données de la table reflètent le résultat de la requête sous forme d'une image instantanée au moment où vous émettez l'instruction REFRESH TABLE. L'exemple suivant remplit la table stockquote_MQT et définit une valeur pour le registre spécial avec l'ancienneté de régénération actuelle.

```
REFRESH TABLE stockquote_MQT;

SET CURRENT REFRESH AGE any;
```

Les requêtes appliquées aux données de la table de requêtes matérialisée sont plus rapides que les requêtes appliquées aux données d'une table de base. Lorsque vous voulez utiliser la table de requêtes matérialisée, vous vous référez à la vue et non au pseudonyme :

```
SELECT * FROM stock_quote_view
WHERE ticker='IBM';
```

Si vous émettez une requête pour sélectionner une valeur qui n'a pas été placée en mémoire cache, 0 lignes sont renvoyées.

Exemple 2 : émission de jointures à l'aide des clés primaires et externes

Les colonnes PRIMARY_KEY et FOREIGN_KEY définissent les relations entre les pseudonymes parent et enfant. Chaque pseudonyme parent doit avoir une option de colonne de clé primaire. Vous définissez les enfants d'un pseudonyme parent avec l'option de colonne de clé externe qui référence la colonne de clé primaire d'un pseudonyme parent. Un pseudonyme peut avoir plusieurs enfants, mais un seul parent.

Comme ces colonnes contiennent uniquement des données binaires, les colonnes sont définies avec les mots clé FOR BIT DATA NOT NULL. Le Centre de contrôle DB2 génère cette définition lorsque vous créez le pseudonyme. Vous pouvez définir explicitement les colonnes PRIMARY_KEY et FOREIGN_KEY comme FOR BIT DATA NOT NULL lorsque vous créez le pseudonyme.

L'exemple suivant montre comment l'encapsuleur de services Web utilise les colonnes PRIMARY_KEY et FOREIGN_KEY pour associer des pseudonymes parent et enfant.

```
CREATE NICKNAME zooport_getzooreport_nn (
  zooid VARCHAR (48) OPTIONS(TEMPLATE '&column'),
  zoo_id VARCHAR (48) OPTIONS(XPATH './ns1:Zoo/@id'),
  report_zoo_zooname VARCHAR (48)
    OPTIONS(XPATH './ns1:Zoo/ns1:ZooName/text()'),
  report_zoo_count VARCHAR (48)
    OPTIONS(XPATH './ns1:Zoo/ns1:Count/text()'),
  report_zoo_lastmodified VARCHAR (48)
    OPTIONS(XPATH './ns1:Zoo/ns1:LastModified/text()'),
  nn_pk VARCHAR (16) NOT NULL OPTIONS(PRIMARY_KEY 'YES'),
  url VARCHAR (256) OPTIONS(URLCOLUMN 'Y'),
  soapaction VARCHAR (256) OPTIONS(SOAPACTIONCOLUMN 'Y')
) FOR SERVER "mytestsrvr"
OPTIONS (
  URL 'http://localhost:9080/MaelstromTest/services/ZooPort',
  SOAPACTION 'http://myzoo.com/getZooReport' ,
  TEMPLATE '<soapenv:Envelope>
    <soapenv:Body>
      <zooId>&zooId[1,1]</zooId>
    </soapenv:Body>
  </soapenv:Envelope>',
  XPATH '/soapenv:Envelope/soapenv:Body' ,
  NAMESPACE ' soapenv="http://schemas.xmlsoap.org/soap/envelope/",
  ns1="http://myzoo.com" ');

CREATE NICKNAME zooport_getzooreport_report_zookeeper_nn (
  nn_fk VARCHAR (16) NOT NULL
    OPTIONS(FOREIGN_KEY 'ZOOPORT_GETZOOREPORT_NN'),
  zookeeper_id VARCHAR (48) OPTIONS(XPATH './@id'),
  report_zookeeper_name VARCHAR (48) OPTIONS(XPATH './ns1:Name/text()'),
  zookeeper_numbercages VARCHAR(48)
    OPTIONS(XPATH './ns1:NumberOfCages/text()'),
  nn_pk VARCHAR (16) NOT NULL OPTIONS(PRIMARY_KEY 'YES')
)
FOR SERVER "MYTESTSRVR" OPTIONS(
  XPATH './ns1:Zoo/ns1:Zookeeper' ,
  NAMESPACE ' soapenv="http://schemas.xmlsoap.org/soap/envelope/",
  ns1="http://myzoo.com" ');
```

La clé externe, nn_fk, dans le pseudonyme zooport_getzooreport_report_zookeeper_nn, se réfère au pseudonyme parent, zooport_getzooreport_nn, dans l'option de clé externe. Les colonnes de pseudonymes de clés primaires et externes désignées ne correspondent pas aux

données dans votre document WSDL, car ces colonnes de pseudonymes contiennent des clés qui sont générées par l'encapsuleur. Ces clés identifient une relation entre les pseudonymes parent et enfant qui n'est unique qu'à l'intérieur d'une requête. Si le pseudonyme enfant contient une colonne d'entrée, le modèle de l'option de pseudonyme parent renvoie à ce pseudonyme enfant dans la structure de modèle avec l'option de pseudonyme.

L'instruction SQL suivante associe les pseudonymes parent et enfant :

```
SELECT *
FROM   zooport_getzooreport_nn a,
       zooport_getzooreport_report_zookeeper_nn z,

WHERE  a.nn_pk = z.nn_fk
       AND a.zooid = 100
       ;
```

La description suivante explicite comment l'encapsuleur de services Web utilise les pseudonymes TEMPLATE et XPATH et les options de colonne pendant l'exécution de la requête. Elle n'est pas conçue comme un exemple d'implémentation spécifique.

Lorsque vous joignez les colonnes de clés primaires et externes, l'encapsuleur de services Web envoie un message au fournisseur des services Web et un ensemble de lignes est renvoyé par le fournisseur des services Web. L'encapsuleur génère un message pour le pseudonyme parent en remplaçant par les valeurs de la colonne d'entrée (a.zooid = 100) dans la requête la référence dans le modèle d'option de colonne (ZOOID VARCHAR (48) OPTIONS(TEMPLATE '&column')), puis toutes les références de colonne dans l'option du modèle de pseudonyme (<zooId>&zooId[1,1]</zooId>). L'option du modèle de pseudonyme peut comporter des références de colonne ou des références de pseudonyme enfant. Le message est alors envoyé au service Web.

L'encapsuleur génère les lignes pour un pseudonyme en appliquant l'option de pseudonyme XPATH au document que le service Web renvoie. Si l'option de pseudonyme XPATH renvoie plusieurs fragments XML, alors le pseudonyme contient plusieurs lignes. L'option XPATH de colonne est appliquée aux fragments XML résultants, qui représentent les lignes pour obtenir les valeurs de colonne. Si un pseudonyme a un ou plusieurs parents indirects, toutes les expressions XPATH du pseudonyme parent sont appliquées du haut vers le bas de la hiérarchie avant que l'option de pseudonyme XPATH et l'option de colonne XPATH soient appliquées à ce pseudonyme.

Configuration de l'accès aux sources de données XML

Vous pouvez intégrer les données qui figurent dans les sources de données XML avec les informations d'autres sources à l'aide d'un système fédéré.

Pour configurer un serveur fédéré pour accéder à des sources de données XML, vous devez fournir au serveur fédéré des informations sur les sources de données et les objets auxquels vous voulez accéder. Après avoir configuré le serveur fédéré, vous pouvez créer des requêtes pour accéder aux sources de données XML.

encapsuleur XML

Le langage XML (Extensible Markup Language) est un format universel pour les documents et les données structurés. XML utilise des balises pour structurer les données dans les documents.

Les fichiers XML utilisent l'extension de fichier `.xml`. De même qu'HTML, XML utilise des mots placés entre signes inférieur et supérieur (< >) comme des balises. Les balises structurent les données qui figurent dans le document.

Le document suivant est un exemple de document XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <customer id='123'>
    <name>...</name>
    <address>...</address>
    ...
    <order>
      <amount>...</amount>
      <date>...</date>
      <item quant='12'>
        <name>...</name>
      </item>
      <item quant='4'>...</item>
      ...
    </order>
    <order>...</order>
    ...
    <payment>
      <number>...</number>
      <date>...</date>
    </payment>
    <payment>>...</payment>
    ...
  </customer>
  <customer id='124'>...</customer>
</doc>
```

Figure 21. Exemple de document XML

Fonctionnement de l'encapsuleur XML

L'encapsuleur XML vous permet d'utiliser SQL pour interroger les types de données suivants :

- Documents XML externes stockés dans un fichier unique
- Plusieurs fichiers XML dans un chemin de répertoire
- Fichiers XML distants référencés avec un identificateur URI (Uniform Reference Identifier)
- Documents XML stockés dans des colonnes relationnelles

La figure suivante montre comment l'encapsuleur XML fonctionne avec votre système fédéré.

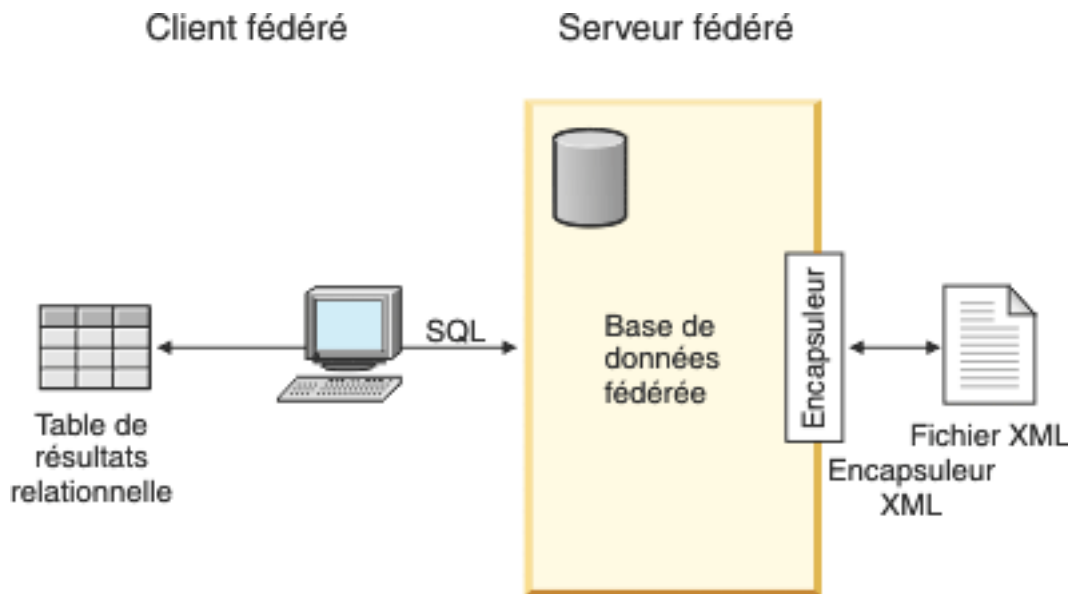


Figure 22. Fonctionnement de l'encapsuleur XML

Avec l'encapsuleur XML, vous pouvez mapper des données XML d'une source de données externe vers un schéma relationnel composé d'un ensemble de pseudonymes. La structure d'un document XML équivaut logiquement à un schéma relationnel dans lequel les éléments imbriqués et répétés sont modélisés comme des tables séparées avec des clés externes.

Les pseudonymes qui correspondent à un document XML sont organisés dans une structure arborescente dans laquelle les pseudonymes enfant sont mappés aux éléments imbriqués à l'intérieur de l'élément qui correspond au pseudonyme parent.

Un pseudonyme racine est un pseudonyme se situant au niveau supérieur d'une hiérarchie des pseudonymes. Un pseudonyme non-racine est un pseudonyme qui a un pseudonyme parent dans une hiérarchie des pseudonymes. Vous pouvez avoir des pseudonymes racine qui ne sont pas l'élément de niveau supérieur d'un document XML.

Lorsque des éléments imbriqués sont répétés ou ont des identités distinctes avec des structures complexes, vous pouvez fournir des pseudonymes séparés pour chaque élément imbriqué.

Les pseudonymes enfant et parent sont connectés par des clés primaires et externes qui sont générées par l'encapsuleur.

Des expressions XPath sont utilisées pour mapper un document XML vers un schéma relationnel composé d'un ensemble de pseudonymes. XPath est un mécanisme d'adressage pour identifier les composants d'un fichier XML (par exemple, les groupes de noeuds et d'attributs à l'intérieur d'une arborescence de documents XML). La syntaxe XPath de base est similaire à l'adressage du système de fichiers.

Chaque pseudonyme est défini par une expression XPath qui identifie les éléments XML représentant les différents blocs de données et un ensemble d'expressions XPath qui indique comment extraire les valeurs de colonne de chaque élément.

Exemple de mappage de document XML

L'exemple suivant illustre comment :

- L'exemple de document XML est mappé vers un ensemble de pseudonymes
- Les relations parent et enfant sont établies à l'aide de clés primaires et externes
- Les expressions XPath sont utilisées pour définir les différents blocs de données et colonnes à l'intérieur de chaque élément du document
- Une requête peut être appliquée au document XML une fois le document enregistré sur votre système fédéré

L'exemple de document XML contient un ensemble d'éléments client. Chaque élément contient plusieurs éléments de commande et de paiement.

Les éléments de commande contiennent plusieurs éléments d'article.

La relation entre les éléments est représentée dans la figure suivante.

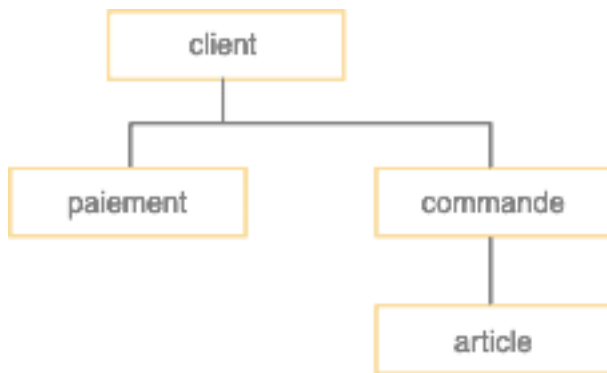


Figure 23. Structure arborescente de l'exemple de document XML

A partir de cette structure, vous pouvez utiliser l'instruction CREATE NICKNAME pour mapper le document XML vers un schéma relationnel comportant quatre pseudonymes :

- customers (clients)
- orders (commandes)
- payments (paiements)
- items (articles)

Vous définissez des relations entre les pseudonymes en désignant chaque pseudonyme comme pseudonyme parent ou pseudonyme enfant à l'aide d'options de colonne de pseudonyme spéciales qui indiquent des clés primaires et externes. Chaque pseudonyme parent doit avoir une colonne spéciale désignée par une option de colonne de clé primaire. Vous définissez les enfants d'un pseudonyme parent avec l'option de colonne de clé externe qui référence la colonne de clé primaire d'un pseudonyme parent. Les colonnes de pseudonyme primaire et externe désignées ne correspondent pas aux données dans votre document XML, car ces colonnes de pseudonymes contiendront des clés générées par l'encapsuleur. Un pseudonyme peut avoir plusieurs enfants, mais un seul parent. Le pseudonyme racine n'a pas de parent.

Dans l'exemple de document XML, une clé primaire est définie pour le pseudonyme customers et des clés externes pointant sur le pseudonyme parent

sont définies pour les pseudonymes orders, payments et items. Les clés externes des pseudonymes orders et payments pointent sur le pseudonyme customers et la clé externe du pseudonyme items pointe sur le pseudonyme orders.

Pour identifier les éléments XML qui représentent les différents blocs de données, vous créez une expression XPath. Dans cet exemple, tous les éléments client sont référencés à l'aide de l'expression XPath '/doc/customer' et tous les éléments de commande sont référencés à l'aide de l'expression XPath './order'. Le point dans l'expression XPath './order' indique que les blocs de données de chaque élément de commande sont imbriqués à l'intérieur des blocs de données de l'élément client correspondant.

Vous créez un ensemble d'expressions XPath pour indiquer comment extraire les valeurs de colonne de chaque élément. Dans cet exemple, l'attribut id des éléments client, qui est maintenant une colonne définie dans le pseudonyme, est référencé à l'aide de l'expression XPath './@id'. L'élément nom des éléments client est référencé à l'aide de l'expression XPath './name'. L'élément adresse des éléments client est référencé à l'aide de l'expression XPath './address/@street'.

Une fois le document XML mappé vers un ensemble de pseudonymes à l'aide de l'instruction CREATE NICKNAME, vous définissez chaque pseudonyme comme un parent ou un enfant à l'aide de clés primaires et externes. Vous indiquez les expressions XPath dans ces clés primaires et externes qui définissent les différents blocs de données et colonnes à l'intérieur de chaque élément du document. Vous pouvez alors appliquer des requêtes SQL au document XML.

Ajout de données XML à un système fédéré

Pour configurer un serveur fédéré pour accéder à des sources de données XML, vous devez fournir au serveur fédéré des informations sur les sources de données et les objets auxquels vous voulez accéder.

Avant de commencer

- La fédération doit être installée sur un serveur qui fera office de serveur fédéré.
- Il doit y avoir une base de données sur le serveur fédéré.

A propos de cette tâche

Vous pouvez configurer un serveur fédéré pour accéder à des données stockées dans des sources de données en XML à l'aide du Centre de contrôle ou en émettant des instructions SQL sur la ligne de commande. Le Centre de contrôle comporte un assistant pour vous guider à travers les étapes nécessaires pour configurer les objets fédérés obligatoires.

Procédure

Pour ajouter des sources de données XML à un serveur fédéré :

1. Enregistrez l'encapsuleur XML.
2. Enregistrez la définition de serveur XML.
3. Enregistrez des pseudonymes pour les sources de données XML.
4. Créez des vues fédérées pour les pseudonymes non-racine.

Enregistrement de l'encapsuleur XML

Vous devez enregistrer un encapsuleur pour accéder à des sources de données XML.

A propos de cette tâche

Les encapsuleurs sont utilisés par les serveurs fédérés pour dialoguer avec les sources de données et extraire des données des sources de données. Les encapsuleurs sont mis en oeuvre comme un ensemble de fichiers de bibliothèque.

Vous pouvez enregistrer un encapsuleur à l'aide du Centre de contrôle ou depuis la ligne de commande. Le Centre de contrôle comporte un assistant pour vous guider à travers les étapes nécessaires pour enregistrer l'encapsuleur.

Si vous utilisez un serveur proxy et un fichier de clés pour accéder à des fichiers XML, vous pouvez indiquer les informations de serveur proxy comme options lorsque vous enregistrez l'encapsuleur ou la définition de serveur. Si vous indiquez les informations de serveur proxy lorsque vous enregistrez la définition de serveur, ces paramètres remplacent les paramètres du serveur proxy que vous indiquez lorsque vous enregistrez l'encapsuleur.

Procédure

Pour enregistrer l'encapsuleur XML :

Choisissez la méthode que vous voulez utiliser pour enregistrer l'encapsuleur XML :

Méthode	Procédure.
Utilisation du Centre de contrôle	Démarrez l'assistant des objets fédérés. Cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée et cliquez sur Créer des objets fédérés .
Depuis la ligne de commande	Emettez l'instruction CREATE WRAPPER. <pre>CREATE WRAPPER <i>nom_encapsuleur</i> LIBRARY <i>nom_bibliothèque</i>;</pre> <p>Si vous utilisez un serveur proxy ou un fichier de clés pour accéder à des fichiers XML, vous devez indiquer plusieurs options lorsque vous enregistrez l'encapsuleur XML ou la définition de serveur. Pour indiquer les informations de serveur proxy lorsque vous enregistrez l'encapsuleur XML, l'instruction que vous émettez sera :</p> <pre>CREATE WRAPPER <i>nom_encapsuleur</i> LIBRARY <i>nom_bibliothèque</i> OPTIONS (PROXY_TYPE '<i>type</i>', PROXY_SERVER_NAME '<i>server_name</i>', PROXY_SERVER_PORT '<i>numéro_port</i>');</pre>

Vous devez indiquer le paramètre LIBRARY dans l'instruction CREATE WRAPPER. Le nom du fichier de bibliothèque de l'encapsuleur que vous indiquez dépend du système d'exploitation du serveur fédéré. Voir dans la liste des fichiers de bibliothèque de l'encapsuleur XML le nom de bibliothèque à indiquer dans l'instruction CREATE WRAPPER.

Si vous utilisez un serveur proxy ou un fichier de clés pour accéder à des fichiers XML, vous devez indiquer plusieurs options lorsque vous enregistrez l'encapsuleur XML ou la définition de serveur.

Fichiers de bibliothèque de l'encapsuleur XML :

Les fichiers de bibliothèque de l'encapsuleur XML sont ajoutés au serveur fédéré lorsque vous installez le serveur fédéré.

Lorsque vous installez IBM InfoSphere Federation Server, trois fichiers de bibliothèque sont ajoutés dans le chemin de répertoire par défaut. Par exemple, si le serveur fédéré fonctionne sous AIX, les fichiers de bibliothèque de l'encapsuleur qui sont ajoutés dans le chemin de répertoire sont libdb2lsxml.a, libdb2lsxmlF.a et libdb2lsxmlU.a. Le fichier de bibliothèque de l'encapsuleur par défaut est libdb2lsxml.a. Les autres fichiers de bibliothèque de l'encapsuleur sont utilisés avec des options spécifiques de l'encapsuleur.

Vous devez inclure le paramètre LIBRARY dans l'instruction CREATE WRAPPER et indiquer le nom de fichier de bibliothèque de l'encapsuleur par défaut.

Les chemins des répertoires par défaut et les noms de fichier de bibliothèque de l'encapsuleur par défaut sont indiqués dans la table suivante.

Tableau 41. Emplacements et noms de fichier de la bibliothèque de l'encapsuleur XML

Système d'exploitation	Chemin de répertoire	Nom de fichier de bibliothèque d'encapsuleur
AIX	/usr/opt/<chemin_installation>/lib32/ /usr/opt/<chemin_installation>/lib64/	libdb2lsxml.a
Linux	/opt/IBM/db2/<chemin_installation>/lib32 /opt/IBM/db2/<chemin_installation>/lib64	libdb2lsxml.so
Solaris	/opt/IBM/db2/<chemin_installation>/lib32 /opt/IBM/db2/<chemin_installation>/lib64	libdb2lsxml.so
Windows	%DB2PATH%\bin	db2lsxml.dll

<chemin_installation> est le chemin de répertoire dans lequel le serveur fédéré est installé sous Linux ou UNIX.

%DB2PATH% est la variable d'environnement qui est utilisée pour indiquer le chemin de répertoire dans lequel le serveur fédéré est installé sous Windows. Le chemin Windows par défaut est C:\Program Files\IBM\SQLLIB.

Instruction CREATE WRAPPER - exemples pour l'encapsuleur XML :

Utiliser l'instruction CREATE WRAPPER pour enregistrer l'encapsuleur XML. Les exemples montrent les paramètres qui sont obligatoires pour accéder à des documents XML avec et sans serveur proxy.

Enregistrement d'un encapsuleur

Si vous n'utilisez pas un serveur proxy pour accéder aux documents XML, l'instruction que vous émettez pour enregistrer l'encapsuleur est :

```
CREATE WRAPPER encapsuleur_xml LIBRARY 'libdb2lsxml.a';
```

encapsuleur_xml

Nom que vous attribuez à l'encapsuleur XML. Les doublons de noms d'encapsuleur ne sont pas autorisés.

LIBRARY '*libdb2lsxml.a*'

Nom du fichier de bibliothèque d'encapsuleur pour les serveurs fédérés qui utilisent les systèmes d'exploitation AIX.

Enregistrement d'un encapsuleur pour un serveur proxy HTTP

Pour enregistrer un encapsuleur et indiquer un serveur proxy HTTP, utilisez l'instruction suivante :

```
CREATE WRAPPER encapsuleur_xml LIBRARY 'libdb2lxml.a'  
  OPTIONS (PROXY_TYPE 'HTTP', PROXY_SERVER_NAME 'proxy_http',  
          PROXY_SERVER_PORT '8080');
```

PROXY_TYPE '*HTTP*'

Indique le type de serveur proxy qui est utilisé pour accéder à Internet derrière un pare-feu. Les valeurs valides sont 'NONE', 'HTTP' ou 'SOCKS'.

PROXY_SERVER_NAME '*proxy_http*'

Indique le nom de serveur proxy ou l'adresse IP. Cette option est obligatoire si la valeur de l'option de serveur PROXY_TYPE est 'HTTP' ou 'SOCKS'.

PROXY_SERVER_PORT '*8080*'

Indique le numéro de port du serveur proxy. Cette option est obligatoire si la valeur de l'option de serveur PROXY_TYPE est 'HTTP' ou 'SOCKS'.

Enregistrement d'un encapsuleur pour un serveur proxy SOCKS

Pour enregistrer un encapsuleur et indiquer un serveur proxy SOCKS sans informations d'authentification, utilisez l'instruction suivante :

```
CREATE WRAPPER encapsuleur_xml LIBRARY 'libdb2lxml.so'  
  OPTIONS (PROXY_TYPE 'SOCKS', PROXY_SERVER_NAME 'proxy_socks',  
          PROXY_SERVER_PORT '1081');
```

LIBRARY '*libdb2lxml.so*'

Indique le nom du fichier de bibliothèque de l'encapsuleur pour les serveurs fédérés fonctionnant sous les systèmes d'exploitation Linux et Solaris.

PROXY_TYPE '*SOCKS*'

Indique le type de serveur proxy qui est utilisé pour accéder à Internet derrière un pare-feu. Les valeurs valides sont 'NONE', 'HTTP' ou 'SOCKS'.

PROXY_SERVER_NAME '*proxy_socks*'

Indique le nom de serveur proxy ou l'adresse IP. Cette option est obligatoire si la valeur de l'option de serveur PROXY_TYPE est 'HTTP' ou 'SOCKS'.

PROXY_SERVER_PORT '*1081*'

Indique le numéro de port du serveur proxy. Cette option est obligatoire si la valeur de l'option de serveur PROXY_TYPE est 'HTTP' ou 'SOCKS'.

Enregistrement de la définition de serveur pour une source de données XML

Vous devez enregistrer une définition de serveur pour extraire des documents XML. La définition de serveur que vous enregistrez est différente selon que vous utilisez ou non un serveur proxy pour accéder aux documents XML.

A propos de cette tâche

Pour extraire des documents XML qui sont derrière un pare-feu à l'aide d'un serveur proxy et d'un identificateur URI, vous devez inclure les options de serveur proxy dans l'instruction CREATE SERVER.

Si vous n'utilisez pas de serveur proxy, vous devez quand même enregistrer une définition de serveur parce que la hiérarchie des objets fédérés exige que les fichiers XML soient associés à un objet de définition de serveur spécifique.

Vous pouvez enregistrer une définition de serveur à l'aide du Centre de contrôle ou dans la ligne de commande. Le Centre de contrôle comporte un assistant pour vous guider à travers les étapes de l'enregistrement de la définition de serveur.

Procédure

Pour enregistrer une définition de serveur pour une source de données XML :

Choisissez la méthode que vous voulez utiliser pour enregistrer la définition de serveur :

Méthode	Procédure.
Utilisation du Centre de contrôle	Utilisez l'assistant Objets fédérés. Cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée et cliquez sur Créer des objets fédérés . Suivez les étapes de l'assistant.
Depuis la ligne de commande	<p>Émettez l'instruction CREATE SERVER.</p> <p>Si vous n'utilisez pas de serveur proxy pour accéder aux documents XML, l'instruction que vous émettez sera :</p> <pre>CREATE SERVER <i>nom_définition_serveur</i> WRAPPER <i>nom_encapsuleur</i>;</pre> <p>Si vous utilisez un serveur proxy pour accéder aux documents XML, l'instruction que vous émettez sera :</p> <pre>CREATE SERVER <i>nom_définition_serveur</i> WRAPPER <i>nom_encapsuleur</i> OPTIONS (PROXY_TYPE '<i>type</i>', PROXY_SERVER_NAME '<i>server_name</i>', PROXY_SERVER_PORT '<i>numéro_port</i>');</pre>

Si vous utilisez un serveur proxy pour accéder à des documents XML, le protocole que vous utilisez peut imposer que vous indiquiez un identificateur utilisateur et un mot de passe pour le serveur proxy. Vous indiquez les informations d'authentification comme options lorsque vous enregistrez la définition de serveur.

Instruction CREATE SERVER - exemples pour l'encapsuleur XML :

Utiliser l'instruction CREATE SERVER pour enregistrer des définitions de serveur pour l'encapsuleur XML. Les exemples montrent les paramètres qui sont obligatoires pour accéder à des documents XML avec et sans serveur proxy.

Enregistrement d'une définition de serveur

Même si vous n'utilisez pas un serveur proxy pour accéder à des documents XML, vous devez toujours enregistrer une définition de serveur. La hiérarchie des objets fédérés exige que les fichiers XML soient associés à un objet de définition de serveur spécifique. L'instruction que vous émettez pour enregistrer une définition de serveur est :

```
CREATE SERVER serveur_xml WRAPPER encapsuleur_xml;
```


serveur_xml

Nom que vous attribuez à la définition de serveur XML. Les doublons de noms de définition de serveur ne sont pas autorisés.

WRAPPER *encapsuleur_xml*

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

Définitions de serveur lorsqu'un serveur proxy est utilisé

Vous devez utiliser les options de serveur proxy dans l'instruction CREATE SERVER si toutes les conditions suivantes sont vraies :

- Vous voulez extraire des données à l'aide d'un identificateur URI
- L'identificateur URI utilisé extraira les données derrière un pare-feu, à travers un serveur proxy
- Le pare-feu ou serveur proxy utilisé est HTTP ou SOCKS

Les options que vous indiquez dépendent du type de serveur proxy auquel vous voulez accéder.

Vérifiez avec votre administrateur de réseau les informations sur le type de serveur proxy que vous utilisez et les paramètres que vous devez indiquer dans les options du serveur proxy.

Enregistrement d'une définition de serveur pour un serveur proxy HTTP

Pour enregistrer une définition de serveur et indiquer un serveur proxy HTTP, utilisez l'instruction suivante :

```
CREATE SERVER serveur_xml_http
  WRAPPER encapsuleur_xml
  OPTIONS (PROXY_TYPE 'HTTP', PROXY_SERVER_NAME 'proxy_http',
    PROXY_SERVER_PORT '8080');
```

serveur_xml_http

Nom que vous attribuez à la définition de serveur XML. Les doublons de noms de définition de serveur ne sont pas autorisés.

WRAPPER *encapsuleur_xml*

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

PROXY_TYPE '*HTTP*'

Indique le type de serveur proxy qui est utilisé pour accéder à Internet derrière un pare-feu. Les valeurs valides sont 'NONE', 'HTTP' ou 'SOCKS'.

PROXY_SERVER_NAME '*proxy_http*'

Indique le nom de serveur proxy ou l'adresse IP. Cette option est obligatoire si la valeur de l'option de serveur PROXY_TYPE est 'HTTP' ou 'SOCKS'.

PROXY_SERVER_PORT '*8080*'

Indique le numéro de port du serveur proxy. Cette option est obligatoire si la valeur de l'option de serveur PROXY_TYPE est 'HTTP' ou 'SOCKS'.

Enregistrement d'une définition de serveur pour un serveur proxy SOCKS

Pour enregistrer une définition de serveur et indiquer un serveur proxy SOCKS lorsque des informations d'authentification ne sont pas exigées, utilisez l'instruction suivante :

```
CREATE SERVER serveur_xml_socks
  WRAPPER encapsuleur_xml
  OPTIONS (PROXY_TYPE 'SOCKS', PROXY_SERVER_NAME 'proxy_socks',
    PROXY_SERVER_PORT '1081');
```

serveur_xml_socks

Nom que vous attribuez à la définition de serveur XML. Les doublons de noms de définition de serveur ne sont pas autorisés.

WRAPPER *encapsuleur_xml*

Le nom d'encapsuleur que vous avez indiqué dans l'instruction CREATE WRAPPER.

PROXY_TYPE '*SOCKS*'

Indique le type de serveur proxy qui est utilisé pour accéder à Internet derrière un pare-feu. Les valeurs valides sont '*NONE*', '*HTTP*' ou '*SOCKS*'.

PROXY_SERVER_NAME '*proxy_socks*'

Indique le nom de serveur proxy ou l'adresse IP. Cette option est obligatoire si la valeur de l'option de serveur PROXY_TYPE est '*HTTP*' ou '*SOCKS*'.

PROXY_SERVER_PORT '*1081*'

Indique le numéro de port du serveur proxy. Cette option est obligatoire si la valeur de l'option de serveur PROXY_TYPE est '*HTTP*' ou '*SOCKS*'.

Enregistrement d'une définition de serveur pour un serveur proxy SOCKS avec des informations d'authentification

Pour enregistrer une définition de serveur et indiquer un serveur proxy SOCKS avec des informations d'authentification, utilisez l'instruction suivante :

```
CREATE SERVER serveur_xml_socks
  WRAPPER encapsuleur_xml
  OPTIONS (PROXY_TYPE 'SOCKS', PROXY_SERVER_NAME 'proxy_socks',
    PROXY_SERVER_PORT '1081', PROXY_AUTHID 'Martin',
    PROXY_PASSWORD 'not4me');
```

PROXY_AUTHID '*Martin*'

Indique le nom d'utilisateur sur le serveur proxy. Cette option de serveur est obligatoire lorsque la valeur de l'option de serveur PROXY_TYPE est '*SOCKS*'.

PROXY_PASSWORD '*not4me*'

Indiquez le mot de passe sur le serveur proxy qui est associé au nom d'utilisateur '*Martin*'. Cette option de serveur est obligatoire lorsque la valeur de l'option de serveur PROXY_TYPE est '*SOCKS*'.

Indication d'une limite de temps pour les réponses du serveur proxy

Outre les options de serveur obligatoires pour les serveurs proxy, vous pouvez aussi indiquer l'option de serveur SOCKET_TIMEOUT lorsque vous enregistrez une définition de serveur. L'option de serveur SOCKET_TIMEOUT indique la durée maximale, en minutes, pendant laquelle le serveur fédéré attendra les résultats du serveur proxy. Si vous n'indiquez pas l'option de serveur SOCKET_TIMEOUT, aucune limite de temps ne s'applique. Le serveur fédéré attendra les résultats du serveur proxy pendant une durée indéfinie.

Pour enregistrer une définition de serveur et indiquer combien de temps le serveur fédéré attendra une réponse du serveur proxy, utilisez l'instruction suivante :

```
CREATE SERVER serveur_xml_http
  WRAPPER encapsuleur_xml
  OPTIONS (PROXY_TYPE 'HTTP', PROXY_SERVER_NAME 'proxy_http',
    PROXY_SERVER_PORT '8080', SOCKET_TIMEOUT '12');
```

SOCKET_TIMEOUT 12

Indique que le serveur fédéré attendra pendant 12 minutes une réponse du serveur proxy.

Accéder aux fichiers XML à l'aide d'un serveur proxy

Si votre réseau utilise un serveur proxy, vous devez fournir des informations sur le serveur proxy lorsque vous enregistrez l'encapsuleur ou la définition de serveur pour les sources de données XML.

Les options que vous indiquez dépendent du type de serveur proxy auquel vous voulez accéder et du protocole, SSL ou TLS, utilisé.

Vous pouvez indiquer le serveur proxy et les options SSL lorsque vous enregistrez un encapsuleur ou une définition de serveur :

- Si vous indiquez ces options lorsque vous enregistrez l'encapsuleur, les pseudonymes qui sont associés à cet encapsuleur utiliseront les options qui sont définies pour l'encapsuleur.
- Si vous indiquez ces options lorsque vous enregistrez une définition de serveur, les pseudonymes qui sont associés à cette définition de serveur utiliseront les options qui sont définies pour la définition de serveur.
- Si vous indiquez des valeurs différentes pour ces options lorsque vous enregistrez l'encapsuleur et la définition de serveur, les valeurs définies pour la définition de serveur prévalent sur les valeurs définies pour l'encapsuleur.

L'encapsuleur XML comporte une fonction de validation qui risque de subir des limitations lorsqu'elle est utilisée avec le serveur proxy. Les conditions dans lesquelles vous constaterez cette limitation sont les suivantes :

- Vous utilisez la fonction de serveur proxy au niveau de serveur et vous avez défini les diverses options de serveur proxy
- Le document d'instance XML contient une référence à un schéma XML externe qui se trouve à l'extérieur du pare-feu

Si vous êtes dans l'une de ces situations, essayez de modifier l'emplacement de votre schéma XML vers un emplacement à l'intérieur du pare-feu. Si vous modifiez l'emplacement du schéma XML, vous devez mettre à jour le document d'instance XML avec le nouvel emplacement du schéma XML.

Options de serveur proxy

La table suivante présente la liste des options que vous devez indiquer pour chaque type de serveur proxy :

Tableau 42. Options obligatoires avec les serveurs proxy

Type de serveur proxy	Encapsuleur ou options de serveur obligatoires
HTTP ou SOCKS sans authentification	PROXY_TYPE PROXY_SERVER_NAME PROXY_SERVER_PORT

Tableau 42. Options obligatoires avec les serveurs proxy (suite)

Type de serveur proxy	Encapsuleur ou options de serveur obligatoires
HTTP ou SOCKS avec authentification	PROXY_TYPE PROXY_SERVER_NAME PROXY_SERVER_PORT PROXY_AUTHID PROXY_PASSWORD

Options de serveur SSL

La table suivante présente la liste des options que vous devez indiquer lorsque vous utilisez les protocoles SSL :

Tableau 43. Options obligatoires avec les protocoles SSL

Objet fédéré	Options obligatoires
Encapsuleur	SSL_KEYSTORE_FILE SSL_KEYSTORE_PASSWORD SSL_VERIFY_SERVER_CERTIFICATE
Définition de serveur	SSL_KEYSTORE_FILE SSL_KEYSTORE_PASSWORD SSL_VERIFY_SERVER_CERTIFICATE SSL_CLIENT_CERTIFICATE_LABEL

Pseudonymes pour les sources de données XML

Vous devez enregistrer un pseudonyme pour chaque document XML auquel vous voulez accéder. Utilisez ces pseudonymes, au lieu des noms de document, lorsque vous interrogez des sources de données XML.

Vous pouvez enregistrer des pseudonymes pour des sources de données XML à l'aide du Centre de contrôle ou de la ligne de commande. Le Centre de contrôle simplifie le processus de création des pseudonymes XML. Avant d'enregistrer les pseudonymes, vous devez comprendre :

Associations de données entre pseudonymes et documents XML :

Les pseudonymes correspondent à la structure arborescente de vos données de documents XML. Les pseudonymes parents et les pseudonymes enfant correspondent à la structure racine et aux éléments imbriqués de la structure arborescente de données. Ces pseudonymes parent et enfant sont connectés par des clés primaires et externes qui sont indiquées avec l'instruction CREATE NICKNAME.

Chaque pseudonyme est défini par des expressions XPath qui exécutent les fonctions suivantes :

- Identifie les éléments XML qui représentent différents blocs de données
- Indique comment extraire les valeurs de colonne de chaque élément

L'encapsuleur XML utilise des expressions XPath pour établir une correspondance entre les données dans le document XML et les lignes dans une table relationnelle. Ces expressions XPath identifient les valeurs à l'intérieur du document XML et

déterminent comment ces valeurs correspondent aux colonnes de chaque ligne. L'encapsuleur XML lit uniquement les données de documents XML. L'encapsuleur XML ne met pas à jour ces données.

Lorsque vous créez un pseudonyme, vous choisissez des options qui indiquent l'association entre le pseudonyme et le document XML. Les pseudonymes sont associés à vos documents XML d'une manière fixe ou bien avec des noms de source que vous indiquez.

Avec une association fixe, le pseudonyme représente les données de documents XML spécifiques. Ces documents XML sont :

Un fichier local

Vous indiquez un fichier XML comme votre document XML.

Plusieurs fichiers locaux dans un chemin de répertoire

Vous indiquez un chemin de répertoire dans lequel se trouvent plusieurs fichiers XML. Les fichiers XML dans ce chemin de répertoire fournissent les données de documents XML au pseudonyme. Tous les fichiers XML doivent avoir la même configuration. Si un fichier XML dans le répertoire a une configuration différente de la configuration du pseudonyme, l'encapsuleur XML renvoie des valeurs NULL lorsqu'il traite ce fichier de données XML. Le répertoire doit être local pour le serveur fédéré ou accessible à partir d'un système de fichiers partagés.

Lors de l'analyse du répertoire, l'encapsuleur XML ne sélectionne pour l'analyse syntaxique que les fichiers dotés d'une extension .xml. L'encapsuleur XML ignore tous les autres fichiers, notamment les fichiers avec une extension .txt, les fichiers avec une extension .xsd et les fichiers sans extension.

Utilisez l'option FILE_PATH dans l'instruction CREATE NICKNAME pour indiquer des données fixes d'un fichier. Utilisez l'option DIRECTORY_PATH pour indiquer des données fixes d'un répertoire.

Si les données source sont indiquées pendant l'exécution de la requête, vous pouvez utiliser le pseudonyme pour représenter les données de toute source de documents XML dont le schéma correspond à la définition du pseudonyme. Ces documents XML sont :

Identificateurs référence uniforme (identificateurs URI)

Un fichier XML distant désigné par un identificateur URI fournit les données de documents XML au pseudonyme. Indiquez cette source de document à l'aide de l'option de colonne de pseudonyme DOCUMENT 'URI'.

Colonnes relationnelles

Les colonnes d'une table relationnelle, d'une vue ou d'un pseudonyme sont utilisées comme entrée pour votre document XML. Indiquez cette source de document à l'aide de l'option de colonne de pseudonyme DOCUMENT 'COLUMN'.

Fichier

Un fichier unique qui contient des données XML est fourni comme entrée pendant l'exécution de la requête. Indiquez cette source de document à l'aide de l'option de colonne de pseudonyme DOCUMENT 'FILE'.

Répertoire

Plusieurs fichiers XML dans le chemin de répertoire indiqué fournissent les

données pendant l'exécution de la requête. Indiquez cette source de document à l'aide de l'option de colonne de pseudonyme DOCUMENT 'DIRECTORY'.

Vous indiquez l'option de colonne DOCUMENT pour indiquer que les données source sont fournies au moment de l'interrogation. Indiquez URI, COLUMN, FILE ou DIRECTORY avec la colonne DOCUMENT pour indiquer le type de source de documents XML.

Vous ne pouvez pas indiquer une option FILE_PATH ou une option DIRECTORY_PATH avec une option de colonne DOCUMENT.

Utilisez l'option STREAMING pour séparer les données de documents XML en fragments. Vous pouvez utiliser l'option STREAMING avec des données de format fixe ou des données de noms de source que vous indiquez lorsque vous exécutez une requête. L'encapsuleur XML traite le flux résultant des données XML et extrait les informations qui sont demandées par un fragment de requête. L'encapsuleur XML fait une analyse syntaxique d'un fragment à la fois. Les fragments étant analysés un par un, l'utilisation totale de la mémoire est réduite. Le temps de traitement nécessaire pour exécuter la requête complète augmente suivant la capacité de mémoire de votre serveur. N'utilisez, donc, l'option STREAMING que pour une analyse syntaxique de gros documents XML de 50 mégaoctets ou plus.

Vous pouvez aussi choisir des valeurs d'option de pseudonyme qui vous aideront à optimiser les requêtes qui extraient d'importants volumes de données XML ou de données contenant plusieurs éléments imbriqués. Ces options sont :

- INSTANCE_PARSE_TIME
- XPATH_EVAL_TIME
- NEXT_TIME

Vous pouvez définir des valeurs pour ces options pour tester et optimiser la requête XML. Ces valeurs d'option contrôlent le temps de traitement nécessaire pour trouver des éléments et faire une analyse syntaxique des données figurant dans les lignes du document XML.

La fonction de modèle de coût pour l'encapsuleur XML :

L'encapsuleur XML fournit une fonction de modèle de coût pour optimiser les requêtes de pseudonymes qui correspondent à vos documents de base XML.

Lorsque vous créez un pseudonyme à l'aide de l'instruction CREATE NICKNAME, vous pouvez indiquer les options de pseudonyme suivantes pour prendre en charge la fonction de modèle de coût :

- INSTANCE_PARSE_TIME
- XPATH_EVAL_TIME

Vous pouvez utiliser les valeurs par défaut pour ces options de pseudonyme. Ou bien vous pouvez définir les valeurs de ces options de pseudonyme pour optimiser les requêtes sur les pseudonymes racine et non-racine que vous créez.

L'option de pseudonyme INSTANCE_PARSE_TIME est la durée, en millisecondes, nécessaire pour lire un élément principal produisant des lignes du pseudonyme racine et en faire une analyse syntaxique. La durée de l'analyse syntaxique comprend tous les éléments non-racine produisant des lignes. Par exemple, si le pseudonyme racine est customers (clients), les éléments non-racine sont tous les

éléments qui correspondent aux commandes, paiements et articles de chaque client. L'encapsuleur XML génère dans la mémoire une structure pour représenter ces éléments racine et non-racine produisant des lignes.

L'option de pseudonyme XPATH_EVAL_TIME est la durée, en millisecondes, nécessaire pour évaluer les expressions XPath qui trouvent les données correspondant à une ligne du pseudonyme. Les expressions XPath qui sont évaluées comprennent les expressions XPath qui trouvent les lignes elles-mêmes et les expressions XPath qui trouvent les valeurs de colonne à l'intérieur de ces lignes.

Espaces de nom pour les sources de données XML :

Utiliser l'option de colonne de pseudonyme NAMESPACES pour identifier les éléments ou attributs qui font partie d'un espace de nom.

Vous pouvez indiquer l'option de pseudonyme NAMESPACES lorsque vous enregistrez des pseudonymes. La valeur de l'option de colonne de pseudonyme NAMESPACES est une liste séparée par des virgules de paires nom et valeur. L'encapsuleur XML utilise les paires nom et valeur pour résoudre les préfixes d'espace de nom qui figurent dans les expressions XPath de pseudonyme. Les préfixes qui sont utilisés dans les expressions XPath sont traités par le processeur XPath.

Dans l'exemple suivant, le document XML comporte les informations de nom, code et de description pour trois produits. Le document XML déclare deux espaces de nom, `http://www.one.com` et `http://www.two.com` et a un espace de nom par défaut `http://www.default.com`. L'élément `product` est associé à l'espace de nom `ns1`. L'élément `product` contient les attributs `name` et `code` et l'élément `desc`. L'attribut `name` n'est pas associé à un espace de nom. L'attribut `code` est associé à l'espace de nom `ns2`. L'élément `desc` est associé à l'espace de nom `default`.

```
<?xml version="1.0" encoding="UTF-8"?>
<doc xmlns:ns1="http://www.one.com" xmlns:ns2="http://www.two.com"
  xmlns="http://www.default.com">
<ns1:product name="Ordinateur" ns2:code="ABC123"
  <desc>"Description du produit Ordinateur"</desc>
<ns1:product name="Clavier" ns2:code="EFG456"
  <desc>"Description du produit Clavier"</desc>
<ns1:product name="Souris" ns2:code="HIJ789"
  <desc>"Description du produit Souris"</desc>
</ns1:product>
</doc>
```

La table suivante montre l'espace de nom qui est associé à chaque élément et attribut dans le document XML.

Tableau 44. Les éléments, attributs et espaces de nom dans le document XML

Élément ou attribut	Espace de nom dans le document XML	Notes
product : un élément du document XML.	ns1="http://www.one.com"	Aucune
name : un attribut de l'élément product dans le document XML.	Aucune	L'attribut name n'est pas associé à un espace de nom.
code : un attribut de l'élément product dans le document XML.	ns2="http://www.two.com"	Aucune

Tableau 44. Les éléments, attributs et espaces de nom dans le document XML (suite)

Élément ou attribut	Espace de nom dans le document XML	Notes
desc : un élément à l'intérieur de l'élément product dans le document XML.	"http://www.default.com"	L'élément desc utilise l'espace de nom default, qui ne contient pas de préfixe.

Lorsque vous enregistrez le pseudonyme pour le document XML, vous définissez trois colonnes pour qu'elles correspondent aux éléments et attributs dans le document XML. Vous indiquez les informations d'espace de nom dans l'option de pseudonyme NAMESPACEs. Par exemple :

```
CREATE NICKNAME products
  (name VARCHAR(16) OPTIONS (XPATH '@name'),
   code VARCHAR(16) OPTIONS (XPATH '@pre2:code')
  description VARCHAR (256) OPTIONS (XPATH './default:desc'))
FOR SERVER serveur_xml
OPTIONS (FILE_PATH '/home/mbreining/sql/xml/namespaces.xml',
        XPATH '//pre1:name',
        NAMESPACEs 'pre1="http://www.one.com", pre2="http://www.two.com",
                    default="http://www.default.com");
```

La table suivante montre comment les espaces de nom de document XML correspondent aux valeurs que vous indiquez dans l'instruction CREATE NICKNAME.

Tableau 45. Espaces de nom de document XML et valeurs correspondantes dans l'instruction CREATE NICKNAME.

Espace de nom dans le document XML	Nom de colonne dans l'instruction CREATE NICKNAME	Valeur dans l'option de colonne XPATH	Valeur dans l'option de pseudonyme NAMESPACEs
L'espace de nom est ns1="http://www.one.com".	Aucune	Aucune	La valeur est pre1="http://www.one.com". La valeur est un préfixe que vous définissez (pre1) et l'identificateur unique de l'espace de nom ("http://www.one.com").
Néant. L'attribut n'est pas associé à un espace de nom.	Nom	La valeur est @name. La valeur est un attribut qui figure dans le document XML (name).	Aucune
L'espace de nom est ns2="http://www.two.com"	Code	La valeur est @pre2:code. La valeur est un préfixe que vous définissez (pre2) et un attribut qui figure dans le document XML (code).	La valeur est pre2="http://www.two.com". La valeur est un préfixe que vous définissez (pre2) et l'identificateur unique de l'espace de nom ("http://www.two.com").

Tableau 45. Espaces de nom de document XML et valeurs correspondantes dans l'instruction CREATE NICKNAME. (suite)

Espace de nom dans le document XML	Nom de colonne dans l'instruction CREATE NICKNAME	Valeur dans l'option de colonne XPATH	Valeur dans l'option de pseudonyme NAMESPACES
L'espace de nom est "http://www.default.com". L'espace de nom par défaut ne contient pas de préfixe.	Description	La valeur est ./default:desc. La valeur est un préfixe que vous définissez (default) et un élément qui figure dans le document XML (desc).	La valeur est default="http://www.default.com". La valeur est un préfixe que vous définissez (default) et l'identificateur unique de l'espace de nom ("http://www.default.com").

L'option de pseudonyme NAMESPACES utilise des descripteurs condensés pour prendre en charge les chaînes de plus de 256 caractères.

Pour de plus amples informations sur les espaces de nom XML, voir l'explication des espaces de nom sur le site Web du consortium World Wide Web.

Enregistrement de pseudonymes pour des sources de données XML :

Dans chaque définition de serveur XML que vous enregistrez, vous devez enregistrer un pseudonyme pour chaque document XML auquel vous voulez accéder. Utilisez ces pseudonymes, au lieu des noms de document XML, lorsque vous interrogez les sources de données XML.

Restrictions

Si vous essayez d'accéder à des sources de données XML qui sont sur une unité partagée d'un serveur fédéré fonctionnant sous Windows 2003, vos requêtes risquent d'échouer. C'est une limitation de Windows 2003. Vous pouvez éviter ce problème en indiquant le chemin d'accès absolu dans l'option FILE_PATH de l'instruction CREATE NICKNAME.

Vous devez créer des pseudonymes qui correspondent à la structure arborescente de votre source de données XML. Les pseudonymes parent correspondent à la structure racine de l'arbre. Les pseudonymes enfant correspondent aux éléments qui sont imbriqués à l'intérieur de l'élément pour le pseudonyme parent.

Vous pouvez enregistrer des pseudonymes à l'aide du Centre de contrôle ou depuis la ligne de commande. Le Centre de contrôle comporte un assistant pour vous guider à travers les étapes de l'enregistrement de la définition de serveur.

Procédure

Pour enregistrer des pseudonymes pour des sources de données XML :

Choisissez la méthode que vous voulez utiliser pour enregistrer la définition de serveur :

Méthode	Procédure.
Utilisation du Centre de contrôle	Utilisez l'assistant Objets fédérés. Cliquez avec le bouton droit de la souris sur le dossier Objets de base de données fédérée et cliquez sur Créer des objets fédérés . Suivez les étapes de l'assistant.
Depuis la ligne de commande	Emettez l'instruction CREATE NICKNAME. La longueur des pseudonymes peut aller jusqu'à 128 caractères.

Instruction CREATE NICKNAME - exemples pour l'encapsuleur XML :

Utiliser l'instruction CREATE NICKNAME pour enregistrer des pseudonymes pour des documents XML. Un exemple complet montre comment créer des pseudonymes parent et enfant et des exemples pour des options de colonne spécifiques.

Recommandation : N'utilisez pas l'opérateur self ou descendant // lorsque vous indiquez des colonnes XPATH et des options de pseudonyme dans vos requêtes. L'opérateur self ou descendant est un opérateur XPath. L'utilisation de l'opérateur self ou descendant peut réduire les performances du serveur fédéré.

Exemple complet

L'exemple suivant montre comment créer des pseudonymes pour des sources de données XML à l'aide de l'exemple de fichier XML présenté dans l'exemple de fichier XML suivant.

```

<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <customer id='123'>
    <name>...</name>
    <address>...</address>
    ...
    <order>
      <amount>...</amount>
      <date>...</date>
      <item quant='12'>
        <name>...</name>
      </item>
      <item quant='4'>...</item>
      ...
    </order>
  </customer>
  <customer id='124'>...</customer>
</doc>

```

Figure 24. Exemple de fichier XML

Le pseudonyme parent

La première étape consiste à créer le pseudonyme parent, customers. Pour créer le pseudonyme, émettez l'instruction suivante :

```

CREATE NICKNAME customers(
  id          VARCHAR(5)    OPTIONS(XPATH './@id')
  name       VARCHAR(16)   OPTIONS(XPATH './name'),
  address    VARCHAR(30)   OPTIONS(XPATH './address/@street'),
  cid       VARCHAR(16)   FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER serveur_xml
OPTIONS(DIRECTORY_PATH '/home/db2user',
        XPATH '/doc/customer', STREAMING 'YES');

```

Cette instruction crée le pseudonyme customers dans plusieurs fichiers XML sous le chemin de répertoire indiqué, /home/db2user.

L'option de pseudonyme STREAMING indique que les données source XML sont séparées et traitées par noeud, dans cet exemple par enregistrement de client. Si l'option de pseudonyme STREAMING est utilisée, l'encapsuleur ne stocke pas tout le document XML dans la mémoire. L'encapsuleur XML divise, en effet, le document en plusieurs sections qui sont soumises à une analyse syntaxique une par une et de façon séquentielle. L'option de pseudonyme STREAMING doit être utilisée uniquement avec les documents XML volumineux. Les performances de vos requêtes sont impactées lorsque vous utilisez cette option.

Les pseudonymes enfant

L'étape suivante consiste à créer les pseudonymes enfant pour les éléments orders, payments et items.

Émettez l'instruction suivante pour créer le pseudonyme enfant orders :

```

CREATE NICKNAME orders(
  amount INTEGER      OPTIONS(XPATH './amount'),
  date   VARCHAR(10)  OPTIONS(XPATH './date'),
  oid    VARCHAR(16)  OPTIONS(PRIMARY_KEY 'YES'),
  cid    VARCHAR(16)  FOR BIT DATA NOT NULL OPTIONS(FOREIGN_KEY 'CUSTOMERS'))
FOR SERVER serveur_xml
OPTIONS( XPATH './order');

```

Emettez l'instruction suivante pour créer le pseudonyme enfant payments :

```

CREATE NICKNAME payments(
  number INTEGER      OPTIONS(XPATH './number'),
  date   VARCHAR(10)  OPTIONS(XPATH './date'),
  cid    VARCHAR(16)  FOR BIT DATA NOT NULL OPTIONS(FOREIGN_KEY 'CUSTOMERS'))
FOR SERVER serveur_xml
OPTIONS( XPATH './payment');

```

Emettez l'instruction suivante pour créer le pseudonyme enfant items :

```

CREATE NICKNAME items(
  name     VARCHAR(20)  OPTIONS(XPATH './name'),
  quantity INTEGER      OPTIONS(XPATH './@quant'),
  oid      VARCHAR(16)  FOR BIT DATA NOT NULL OPTIONS(FOREIGN_KEY 'ORDERS'))
FOR SERVER serveur_xml
OPTIONS( XPATH './item');

```

Exemples d'options de colonne de pseudonyme

Les exemples suivants vous montrent comment inclure les options de colonne de pseudonyme DOCUMENT lorsque vous créez des pseudonymes. Les exemples vous montrent aussi comment ces options sont utilisées dans les requêtes.

Exemple avec DOCUMENT 'FILE'

Les exemples suivants vous montrent comment inclure les options de colonne de pseudonyme DOCUMENT lorsque vous créez des pseudonymes. Les exemples vous montrent aussi comment utiliser ces options dans des requêtes.

L'exemple suivant, avec CREATE NICKNAME, montre l'utilisation de l'option de colonne de pseudonyme DOCUMENT 'FILE' :

```

CREATE NICKNAME customers(
  doc     VARCHAR(100)  OPTIONS(DOCUMENT 'FILE'),
  name    VARCHAR(16)   OPTIONS(XPATH './name'),
  address VARCHAR(30)   OPTIONS(XPATH './address/@street'),
  cid     VARCHAR(16)  FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER serveur_xml
OPTIONS(XPATH '/doc/customer');

```

Vous pouvez alors appliquer la requête suivante au pseudonyme customers en indiquant l'emplacement du document XML dans la clause WHERE :

```

SELECT * FROM customers WHERE doc = '/home/db2user/Customers.xml';

```

Exemple avec DOCUMENT 'DIRECTORY'

L'exemple suivant, avec CREATE NICKNAME, montre l'utilisation de l'option de colonne de pseudonyme DOCUMENT 'DIRECTORY' :

```

CREATE NICKNAME customers(
  doc     VARCHAR(100)  OPTIONS(DOCUMENT 'DIRECTORY'),
  name    VARCHAR(16)   OPTIONS(XPATH './name'),

```

```

address VARCHAR(30)  OPTIONS(XPATH './address/@street'),
cid      VARCHAR(16) FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER serveur_xml
OPTIONS(XPATH '/doc/customer');

```

Vous pouvez alors appliquer la requête suivante au pseudonyme customers :

```
SELECT name FROM customers WHERE doc = '/home/data/xml';
```

Cette requête extrait les documents XML qui se trouvent dans le chemin de répertoire /home/data/xml, qui est indiqué dans la clause WHERE.

Exemple avec DOCUMENT 'URI'

L'exemple suivant, avec CREATE NICKNAME, montre l'utilisation de l'option de colonne de pseudonyme DOCUMENT 'URI' :

```

CREATE NICKNAME customers(
  doc      VARCHAR(100)  OPTIONS(DOCUMENT 'URI'),
  name     VARCHAR(16)   OPTIONS(XPATH './name'),
  address  VARCHAR(30)  OPTIONS(XPATH './address/@street'),
  cid      VARCHAR(16)  FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER serveur_xml
OPTIONS(XPATH '/doc/customer');

```

Vous pouvez alors appliquer la requête suivante au pseudonyme customers pour extraire les données XML de l'emplacement distant :

```
SELECT * FROM customers WHERE doc = 'http://www.lg-mv.org/foo.xml';
```

Exemple avec DOCUMENT 'COLUMN'

L'exemple suivant, avec CREATE NICKNAME, montre l'utilisation de l'option de colonne de pseudonyme DOCUMENT 'COLUMN' :

```

CREATE NICKNAME emp(
  doc      VARCHAR(500)  OPTIONS(DOCUMENT 'COLUMN')
  fname   VARCHAR(16)   OPTIONS(XPATH '@first'),
  lname   VARCHAR(16)   OPTIONS(XPATH '@last'))
FOR SERVER serveur_xml
OPTIONS(XPATH '/doc/name');

```

Vous pouvez alors appliquer une des requêtes suivantes au pseudonyme emp pour extraire les données XML :

```

SELECT * FROM emp WHERE doc = '<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <title> employees </title>
  <name first="David" last="Marston"/>
  <name first="Donald" last="Leslie"/>
  <name first="Emily" last="Farmer"/>
  <name first="Myriam" last="Midy"/>
  <name first="Lee" last="Tran"/>
  <name first="Lili" last="Farmer"/>
  <name first="Sanjay" last="Kumar"/>
</doc>';

```

ou

```
SELECT * FROM emp WHERE doc = (SELECT * FROM xml_tab);
```

La table xml_tab contient une colonne qui est remplie avec les données XML.

Requêtes pour des sources de données XML

Avant de créer des requêtes pour accéder à des sources de données XML, il y a des mesures que vous pouvez prendre pour optimiser les performances d'interrogation.

Vues fédérées

Vous pouvez utiliser des vues fédérées pour garantir que les requêtes qui joignent les éléments d'une hiérarchie des pseudonymes XML s'exécutent correctement.

Eviter l'opérateur self ou descendant

N'utilisez pas l'opérateur self ou descendant // si vous indiquez des colonnes XPATH et des options de pseudonyme lorsque vous créez les pseudonymes XML. L'opérateur self ou descendant est un opérateur XPath et l'utilisation de l'opérateur self ou descendant peut réduire les performances du serveur fédéré.

Serveurs fédérés Windows 2003

Si vous essayez d'accéder à des sources de données XML qui figurent sur une unité partagée d'un serveur fédéré qui fonctionne sous Windows 2003, votre requête risque d'échouer avec le message d'erreur suivant :

```
SQL1822N Code d'erreur imprévue "ERRNO = 2" reçu de la source de données
"XML_SERVER". Le texte et les jetons associés sont "Impossible de lire le fichier".
SQLSTATE=560BD
```

C'est une limitation de Windows 2003. Vous pouvez éviter ce problème en indiquant le chemin d'accès absolu dans l'option FILE_PATH de l'instruction CREATE NICKNAME.

L'exemple suivant montre une instruction CREATE NICKNAME avec un chemin abrégé indiqué dans l'option FILE_PATH :

```
CREATE NICKNAME customers
(
  id      VARCHAR(5)  OPTIONS(XPATH './@id'),
  name    VARCHAR(16) OPTIONS(XPATH './name'),
  address VARCHAR(30) OPTIONS(XPATH './address/@street'),
  cid     VARCHAR(16) FOR BIT DATA NOT NULL
          OPTIONS(PRIMARY_KEY 'YES'))
  FOR SERVER serveur_xml
          OPTIONS(DIRECTORY_PATH '\home\db2user',
                  XPATH '/doc/customer', STREAMING 'YES');
```

Les requêtes qui utilisent ce pseudonyme risquent d'échouer parce que vous avez indiqué le chemin abrégé.

Pour les serveurs fédérés qui fonctionnent sous Windows 2003, indiquez le chemin d'accès absolu dans l'option FILE_PATH de l'instruction CREATE NICKNAME.

Par exemple :

```
CREATE NICKNAME customers
(
  id      VARCHAR(5)  OPTIONS(XPATH './@id'),
  name    VARCHAR(16) OPTIONS(XPATH './name'),
  address VARCHAR(30) OPTIONS(XPATH './address/@street'),
  cid     VARCHAR(16) FOR BIT DATA NOT NULL
          OPTIONS(PRIMARY_KEY 'YES'))
```

```
FOR SERVER serveur_xml
  OPTIONS(DIRECTORY_PATH '\\host.svl.ibm.com\D$\home\db2user',
  XPATH '/doc/customer', STREAMING 'YES');
```

Création de vues fédérées pour les pseudonymes de l'encapsuleur XML

Vous pouvez créer des vues fédérées de la hiérarchie des pseudonymes qui décrivent un document XML. La définition de vues fédérées garantit que les requêtes qui joignent les éléments d'une hiérarchie des pseudonymes XML s'exécutent correctement.

A propos de cette tâche

Une *vue fédérée* est une vue de la base de données fédérée qui référence un pseudonyme au lieu d'une table de la source de données.

Dans la hiérarchie des pseudonymes XML, le pseudonyme racine et les requêtes qui joignent des colonnes, en dehors des colonnes spéciales PRIMARY_KEY et FOREIGN_KEY, ne sont pas impactés par l'utilisation des vues fédérées.

Lorsque vous créez des vues fédérées pour des pseudonymes XML, vous devez inclure tous les prédicats obligatoires et le chemin d'accès complet du répertoire principal.

Procédure

Pour créer des vues fédérées pour des pseudonymes XML :

Utilisez l'instruction CREATE VIEW pour définir une vue pour chaque pseudonyme non-racine. La vue doit être une jointure de tous les pseudonymes sur le chemin du pseudonyme racine.

1. Dans la clause WHERE de la vue, définissez les colonnes PRIMARY_KEY et FOREIGN_KEY comme les prédicats de jointure.
2. Dans la liste de sélection, incluez toutes les colonnes du pseudonyme non-racine sauf la colonne qui est désignée par l'option de colonne de pseudonyme FOREIGN_KEY. Dans la liste de sélection, incluez la colonne du pseudonyme parent désigné avec l'option PRIMARY_KEY.

Instruction CREATE VIEW - exemples pour l'encapsuleur XML

Utiliser l'instruction CREATE VIEW pour créer des vues fédérées pour des pseudonymes non-racine. Cet exemple comporte un exemple de fichier XML, les instructions que vous utilisez pour créer les vues et montre comment vous pouvez utiliser les vues dans une requête.

Vous pouvez créer des vues fédérées de la hiérarchie des pseudonymes qui décrivent un document XML pour garantir le bon fonctionnement des requêtes qui joignent des éléments d'une hiérarchie des pseudonymes XML. Lorsque vous indiquez une vue fédérée dans une requête, les données sont extraites de la source de données distante.

Les exemples suivants vous montrent comment créer des vues fédérées pour des pseudonymes non-racine afin de décrire des documents de base XML.

Exemple de fichier XML

Les exemples suivants reposent sur cet exemple de fichier XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <customer id='123'>
    <name>...</name>
    <address>...</address>
    ...
    <order>
      <amount>...</amount>
      <date>...</date>
      <item quant='12'>
        <name>...</name>
      </item>
      <item quant='4'>...</item>
      ...
    </order>
    <order>...</order>
    ...
    <payment>
      <number>...</number>
      <date>...</date>
    </payment>
    <payment>...</payment>
    ...
  </customer>
  <customer id='124'>...</customer>
</doc>
```

Figure 25. Exemple de fichier XML

Instructions CREATE VIEW pour les pseudonymes non-racine

L'exemple suivant montre comment créer une vue fédérée pour le pseudonyme non-racine order :

```
CREATE VIEW order_view AS
  SELECT o.amount, o.date, o.oid, c.cid
  FROM customers c, orders o
  WHERE c.cid = o.cid;
```

L'exemple suivant montre comment créer une vue fédérée pour le pseudonyme non-racine payment :

```
CREATE VIEW payment_view AS
  SELECT p.number, p.date, c.cid
  FROM customers c, payments p
  WHERE c.cid = p.cid;
```

L'exemple suivant montre comment créer une vue fédérée pour le pseudonyme non-racine item :

```
CREATE VIEW item_view AS
  SELECT i.quantity, i.name, o.oid
  FROM customers c, orders o, items i
  WHERE c.cid = o.cid AND o.oid = i.oid;
```


Une requête qui utilise les vues fédérées

Les requêtes qui sont soumises aux vues fédérées sont traitées correctement parce que le chemin de jointure vers le répertoire principal est indiqué dans la clause WHERE.

Par exemple, la requête suivante utilise le numéro d'identification de client et la date à laquelle une commande a été passée pour renvoyer la quantité qui a été commandée et le montant du paiement dû. Au lieu d'utiliser les pseudonymes dans la requête, les vues sont indiquées dans la clause FROM.

```
SELECT o.amount, p.amount
FROM order_view o, payment_view p
WHERE p.date = o.date AND
      p.cid = o.cid;
```

Astuces pour optimiser les requêtes avec la fonction de modèle de coût XML

La fonction de modèle de coût pour l'encapsuleur XML aide à optimiser les requêtes pour les pseudonymes que vous créez.

La fonction de modèle de coût utilise les options de pseudonyme suivantes de l'instruction CREATE NICKNAME :

- INSTANCE_PARSE_TIME
- XPATH_EVAL_TIME

Vous pouvez indiquer des valeurs pour ces options de pseudonyme lorsque vous émettez une instruction CREATE NICKNAME afin d'enregistrer un pseudonyme pour une source de données XML.

La fonction de modèle de coût utilise ces valeurs de paramètre pour déterminer la durée nécessaire pour faire une analyse syntaxique des données dans chaque ligne d'un document de base XML. Les valeurs de paramètre sont aussi utilisées pour évaluer l'expression XPath pour le pseudonyme.

Vous pouvez utiliser les valeurs par défaut pour ces options de pseudonyme. Cependant, si vous voulez optimiser les requêtes sur des structures de sources XML volumineuses ou complexes pour les pseudonymes que vous créez, utilisez l'exemple suivant comme guide.

Exemple d'optimisation d'une requête volumineuse

Votre document XML a un schéma relationnel avec quatre pseudonymes :

- customers (clients)
- orders (commandes)
- payments (paiements)
- items (articles)

Le pseudonyme racine est customers.

Appliquez des requêtes à chaque pseudonyme. Appliquez chaque requête à un échantillon des données XML représentatif de votre environnement.

Par exemple :

```
SELECT * from customers;  
SELECT * from orders;  
SELECT * from payments;  
SELECT * from items;
```

Notez la durée (en millisecondes) nécessaire pour exécuter chaque requête à l'aide de la commande db2batch ou d'une commande ou d'un utilitaire équivalent. Vous pouvez utiliser la commande db2batch pour obtenir un fichier de sortie contenant le temps nécessaire pour exécuter les requêtes. Notez le nombre de blocs de données renvoyés.

Pour chaque pseudonyme, utilisez les formules suivantes pour déterminer les valeurs optimales pour les options de pseudonyme INSTANCE_PARSE_TIME et XPATH_EVAL_TIME :

```
INSTANCE_PARSE_TIME =  
  (75 % x durée d'exécution de la requête SELECT *) ÷ nombre de blocs de données renvoyés  
XPATH_EVAL_TIME     =  
  (25 % x durée d'exécution de la requête SELECT *) ÷ nombre de blocs de données renvoyés
```

Pour le pseudonyme racine (dans cet exemple, customers), utilisez les valeurs calculées pour les options de pseudonyme INSTANCE_PARSE_TIME et XPATH_EVAL_TIME.

Pour les pseudonymes non-racine, (dans cet exemple orders, payments et items), utilisez uniquement la valeur calculée pour le paramètre XPATH_EVAL_TIME. La valeur du paramètre INSTANCE_PARSE_TIME n'est pas applicable aux pseudonymes non-racine.

Vous pouvez utiliser ces formules pour vous guider dans l'optimisation de vos requêtes. Les valeurs optimales pour ces options de pseudonyme dépendent aussi de la complexité de vos documents de sources XML et de la vitesse du processeur que vous utilisez.

Source de données XML - exemple de requêtes

Exemples de requêtes avec les pseudonymes XML.

Ces exemples utilisent les pseudonymes customers, orders et items.

Requête renvoyant une valeur spécifique à partir des documents XML

Lorsque l'instruction SELECT suivante est exécutée, l'encapsuleur renvoie les noms de tous les clients :

```
SELECT name FROM customers;
```

Requête renvoyant tous les enregistrements pour un client spécifique

Lorsque l'instruction SELECT suivante est exécutée, l'encapsuleur renvoie tous les enregistrements dans lesquels le nom du client est Chang :

```
SELECT * FROM customers  
  WHERE name='Chang';
```

Requête renvoyant des valeurs spécifiques d'après une condition de jonction entre un pseudonyme parent et un pseudonyme enfant

Lorsque l'instruction SELECT suivante est exécutée, l'encapsuleur renvoie les noms de client et montants pour chaque commande passée par chaque client. Vous devez

indiquer la jointure, `c.cid=o.cid`, pour indiquer la relation parent-enfant entre le pseudonyme `customers` et le pseudonyme `orders`.

```
SELECT c.name, o.amount FROM customers c, orders o
WHERE c.cid=o.cid;
```

Requête montrant comment indiquer des conditions de jonction entre un pseudonyme parent et plusieurs pseudonymes enfant

Lorsque l'instruction `SELECT` suivante est exécutée, l'encapsuleur renvoie les adresses du client, les montants de commande et les désignations d'article pour chaque commande et chaque article de chaque client. Vous devez indiquer les deux jointures pour gérer les relations parent-enfant.

```
SELECT c.address, o.amount, i.name FROM customers c, orders o, items i
WHERE c.cid=o.cid AND o.oid=i.oid;
```

Requêtes montrant comment indiquer un document XML dans une requête

Les exemples suivants vous montrent comment écrire des requêtes à l'aide d'un pseudonyme qui indique une option de colonne de pseudonyme `DOCUMENT` au lieu d'une option de pseudonyme `FILE_PATH`.

L'instruction `CREATE NICKNAME` qui est utilisée pour créer le pseudonyme `customers` est :

```
CREATE NICKNAME customers
(
  doc      VARCHAR(100)  OPTIONS(DOCUMENT 'FILE'),
  name     VARCHAR(16)   OPTIONS(XPATH './name'),
  address  VARCHAR(30)  OPTIONS(XPATH './address/@street'),
  cid      VARCHAR(16)  OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER serveur_xml
OPTIONS(XPATH '/doc/customer');
```

La requête suivante sélectionne toutes les données du fichier `Customers.xml` avec le chemin `/home/db2user/Customers.xml` :

```
SELECT * FROM customers
WHERE doc='/home/db2user/Customers.xml';
```

La requête suivante sélectionne les noms de clients et les dates de leurs commandes dans le fichier `Customers.xml` pour chaque commande d'un montant supérieur à 1000. Le chemin `/home/db2user/Customers.xml` indique l'emplacement du fichier `Customers.xml`.

```
SELECT c.name, o.date FROM customers c, orders o
WHERE c.doc='/home/db2user/Customers.xml' AND o.amount > 1000;
```


Chapitre 3. Support de source de données pour des fonctions fédérées

Reportez-vous à ce tableau pour savoir si une source de données prend ou non en charge une fonction fédérée spécifique.

Avant de pouvoir utiliser certaines de ces fonctions, vous devrez éventuellement définir des options d'encapsuleur ou de serveur spécifiques, ou exécuter d'autres tâches pour les activer. Pour plus d'informations, reportez-vous aux rubriques spécifiques à chaque fonction.

Tableau 46. Fonctions et sources de données prises en charge

Fonction	Sources de données
Points de sauvegarde d'application avec opérations WRITE sur les pseudonymes	DB2 for Linux, UNIX et Windows
Optimisation de l'état asynchrone	Toutes les sources de données
Tables de la mémoire cache	famille DB2 Informix Microsoft SQL Server Oracle Sybase
Importation de données dans des pseudonymes	famille DB2 Informix Microsoft SQL Server Oracle Sybase Teradata
Tolérances des erreurs dans les expressions de tables imbriquées	famille DB2 Informix JDBC Microsoft SQL Server ODBC Oracle Sybase Teradata
Référentiel de mappage utilisateur externe	Toutes les sources de données
Indicateurs d'état de santé fédérés	famille DB2 Excel Informix JDBC Microsoft SQL Server ODBC Oracle Sybase Fichiers structurés en tableaux Teradata XML (uniquement pour les pseudonymes racines)

Tableau 46. Fonctions et sources de données prises en charge (suite)

Fonction	Sources de données
Procédures fédérées	famille DB2, en mode sécurisé Oracle, en mode sécurisé Microsoft SQL Server, en mode sécurisé Sybase, en mode isolé, avec le serveur fédéré installé sous UNIX Sybase, en mode isolé ou mode sécurisé, avec le serveur fédéré installé sous Linux ou Microsoft Windows
Contextes accrédités fédérés	DB2 for Linux, UNIX et Windows version 9.5 DB2 for z/OS version 9 Oracle
Proxy HTTP	Services Web XML
Isolement au niveau de la connexion	famille DB2 Informix JDBC Microsoft SQL Server ODBC Oracle Sybase
Contrôle d'accès basé sur les labels	DB2 for Linux, UNIX et Windows version 9.1 et 9.5 Oracle
Opérations LOB en lecture et écriture	DB2 for z/OS DB2 for Linux, UNIX et Windows DB2 for System i Oracle
Opérations LOB en lecture seule	BioRS Informix JDBC Microsoft SQL Server ODBC Script Sybase Teradata Services Web XML
Tables de requêtes matérialisées	Toutes les sources de données, avec des restrictions spécifiques

Tableau 46. Fonctions et sources de données prises en charge (suite)

Fonction	Sources de données
Fonction de mise à jour des statistiques de pseudonyme	BioRS famille DB2 Excel Informix JDBC Microsoft SQL Server ODBC Oracle Sybase Fichiers structurés en tableaux Teradata XML (uniquement pour les pseudonymes racines)
Sessions passe-système	DRDA Informix Oracle Microsoft SQL Server Sybase Teradata
Type de données XML distant	DB2 for Linux, UNIX et Windows encapsuleur XML
Proxy SOCKS	BioRS Script Services Web XML
Secure Socket Layer (SSL)	Services Web XML
Isolement au niveau de l'instruction	famille DB2 Microsoft SQL Server
Transactions de validation en deux phases	DB2 for Linux, UNIX et Windows, en mode sécurisé DB2 for System i, en mode sécurisé DB2 for z/OS, en mode sécurisé Informix, en mode sécurisé Microsoft SQL Server, en mode sécurisé, avec le serveur fédéré installé sous Microsoft Windows Oracle, en mode sécurisé Sybase, en mode sécurisé, avec le serveur fédéré installé sous Microsoft Windows Sybase, en mode isolé, avec le serveur fédéré installé sous UNIX
prise en charge du format Unicode	Toutes les sources de données

Chapitre 4. Référence des options des sources de données

Chaque source de données prend en charge des options d'encapsuleur, de serveur, de mappage utilisateur, de pseudonyme et de colonne spécifiques.

Référence des options BioRS

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur, un pseudonyme et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier dans les instructions CREATE WRAPPER et CREATE SERVER.

Tableau 47. Options d'encapsuleur pour BioRS

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java™, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Cette option est obligatoire si la valeur PROXY_TYPE est HTTP ou SOCKS. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le port ou le nom de service du service proxy sur le serveur proxy. Cette option est obligatoire si la valeur PROXY_TYPE est HTTP ou SOCKS. Les valeurs valides sont un numéro de port décimal allant de 1 à 32760 ou un nom de service.

Tableau 47. Options d'encapsuleur pour BioRS (suite)

Nom	Description
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE, HTTP et SOCKS. La valeur par défaut est NONE. Si vous définissez cette option sur HTTP ou SOCKS, vous devez également spécifier PROXY_SERVER_NAME et PROXY_SERVER_PORT.

Options de serveur

Tableau 48. Options de serveur pour BioRS

Nom	Description
CASE_SENSITIVE	Spécifie si le serveur BioRS traite les noms en étant sensible à la casse. Les valeurs valides sont Y et N. La valeur par défaut est Y ; les noms sont traités avec sensibilité à la casse. Dans le produit BioRS, un paramètre de configuration contrôle la sensibilité à la casse des données stockées sur le serveur BioRS. L'option CASE_SENSITIVE et le paramètre de configuration doivent spécifier la même sensibilité à la casse. Si vous avez besoin de modifier la valeur de l'option CASE_SENSITIVE après avoir créé la définition de serveur, vous devez supprimer la définition de serveur et créer à nouveau la définition et tous les pseudonymes.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Tableau 48. Options de serveur pour BioRS (suite)

Nom	Description
NODE	Obligatoire. Spécifie le nom d'hôte DNS ou l'adresse IP du système sur lequel l'utilitaire d'interrogation BioRS est disponible. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré. La valeur par défaut est localhost.
PORT	Spécifie le port des connexions vers le serveur BioRS. Les valeurs valides sont un port numérique ou un nom de service TCP/IP. La valeur par défaut est 5014.
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le port ou le nom de service du service proxy sur le serveur proxy. Les valeurs valides sont un numéro de port décimal allant de 1 à 32760 ou un nom de service.
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE, HTTP et SOCKS. La valeur par défaut est NONE.
TIMEOUT	Spécifie la durée maximale, en minutes, au cours de laquelle le serveur fédéré attend une réponse du serveur distant. La valeur par défaut est 10.

Options de mappage utilisateur

Tableau 49. Options de mappage utilisateur pour BioRS

Option	Description
GUEST	Spécifie que l'ID d'authentification BioRS GUEST est utilisée pour effectuer des opérations. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'ID BioRS GUEST n'est pas utilisé. Cette option n'est pas valide si vous spécifiez les options REMOTE_AUTHID et REMOTE_PASSWORD.
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.

Tableau 49. Options de mappage utilisateur pour BioRS (suite)

Option	Description
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy. Le mot de passe est chiffré lorsqu'il est stocké dans le catalogue de la base de données fédérée.
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.

Options de pseudonyme

Tableau 50. Options de pseudonyme pour BioRS

Option	Description
REMOTE_OBJECT	Spécifie le nom de la banque de données BioRS associée au pseudonyme. Ce nom détermine le schéma et la banque de données BioRS pour le pseudonyme. Le nom spécifie également la relation du pseudonyme avec les autres pseudonymes. La sensibilité à la casse de cette option dépend de la sensibilité à la casse du serveur BioRS et de la valeur de l'option du serveur CASE_SENSITIVE. Vous ne pouvez pas utiliser l'instruction ALTER NICKNAME pour modifier ou supprimer ce nom. Si le nom d'une banque de données BioRS change, vous devez supprimer le pseudonyme et le recréer.
TIMEOUT	Spécifie la durée d'attente maximale, en minutes, d'une réponse provenant du serveur de source de données. La valeur par défaut est 10.

Options de colonne

Tableau 51. Options de colonne pour BioRS

Option	Description
ELEMENT_NAME	Spécifie le nom de l'élément BioRS. La sensibilité à la casse de ce nom dépend de la sensibilité à la casse du serveur BioRS et de la valeur de l'option du serveur CASE_SENSITIVE. Vous devez spécifier le nom de l'élément BioRS uniquement si il est différent du nom de colonne.

Tableau 51. Options de colonne pour BioRS (suite)

Option	Description
IS_INDEXED	Spécifie si la colonne correspondante est indexée et, par conséquent, si elle peut être référencée dans un prédicat. Les valeurs valides sont Y et N. La valeur par défaut est N. La colonne n'est pas indexée.
REFERENCED_OBJECT	Spécifie le nom de la banque de données BioRS référencée par la colonne en cours. La sensibilité à la casse de ce nom dépend de la sensibilité à la casse du serveur BioRS et de la valeur de l'option du serveur CASE_SENSITIVE. Cette option n'est valide que pour les colonnes ayant le type de données de référence BioRS.

Référence des options de base de données DB2

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur, un pseudonyme et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent aux sources de données DB2 et identifient les options obligatoires que vous devez spécifier.

Tableau 52. Options d'encapsuleur pour les sources de données DB2

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Options de serveur

Tableau 53. Options de serveur pour les sources de données DB2

Nom	Description
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.
COMM_RATE	Spécifie le débit de communication, en mégaoctets par seconde, entre le serveur fédéré et le serveur source de données. Les valeurs valides sont supérieures à 0 et inférieures à 2147483648. La valeur par défaut est 2.
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.
DATE_COMPAT	Indique si le paramètre date_compat est appliqué à la base de données. Les valeurs valides sont Y et N. La valeur par défaut est N. Cette option de serveur est uniquement valide pour DB2 Database pour Linux, UNIX et Windows, version 9.7 ou ultérieure.
DBNAME	Obligatoire. Indique la base de données spécifique à utiliser pour la connexion initiale à la base de données DB2 distante. Cette base de données spécifique représente l'alias de base de données pour la base de données DB2 distante qui est cataloguée sur le serveur fédéré à l'aide de la commande CATALOG DATABASE ou de l'Assistant Configuration de DB2.

Tableau 53. Options de serveur pour les sources de données DB2 (suite)

Nom	Description
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
DB2_TWO_PHASE_COMMIT	Spécifie si le serveur fédéré se connecte à la source de données dans un protocole de validation à deux phases ou un protocole de validation à une phase. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur fédéré utilise le protocole de validation à une phase pour se connecter. Y spécifie que le serveur fédéré utilise un protocole de validation à deux phases pour se connecter.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
FED_PROXY_USER	Spécifie l'ID d'autorisation à utiliser pour établir toutes les connexions accréditées sortantes lorsque la connexion entrante est non accréditée. L'utilisateur dont l'ID est spécifié avec cette option doit avoir un mappage utilisateur spécifiant les options REMOTE_AUTHID et REMOTE_PASSWORD. Restriction : Cette option de serveur est uniquement valide pour DB2 Database pour Linux, UNIX et Windows version 9.5 et ultérieure et DB2 for z/OS version 9 et ultérieure.

Tableau 53. Options de serveur pour les sources de données DB2 (suite)

Nom	Description
FOLD_ID	Spécifie la casse de l'ID utilisateur envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie l'ID utilisateur en majuscule ; si l'ID utilisateur en majuscule échoue, le serveur envoie alors l'ID utilisateur en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
FOLD_PW	Spécifie la casse du mot de passe envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie le mot de passe en majuscule ; si le mot de passe en majuscule échoue, le serveur envoie alors le mot de passe en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
IO_RATIO	Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.
NO_EMPTY_STRING	Indique si le serveur source de données distant peut contenir des chaînes vides. Les valeurs valides sont Y et N. La valeur par défaut varie selon votre source de données distante. Pour les sources de données distantes Oracle, la valeur par défaut est Y ; toutes les valeurs de chaîne vides sont converties en valeurs NULL. Pour toutes les autres sources de données distantes, la valeur par défaut est N ; la source de données peut contenir des chaînes vides. Vous pouvez améliorer les performances de vos systèmes en définissant cette option sur Y dans les configurations système où le serveur fédéré est en mode compatible VARCHAR2, mais où la source de données distante n'est pas compatible VARCHAR2.

Tableau 53. Options de serveur pour les sources de données DB2 (suite)

Nom	Description
NUMBER_COMPAT	<p>Indique si le serveur de source de données prend en charge le type de données NUMBER. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur de source de données ne prend pas en charge le type de données NUMBER. Dans les systèmes où le serveur fédéré ne prend pas en charge le type de données NUMBER contrairement au serveur de source de données, vous devez définir l'option NUMBER_COMPAT sur Y, car le serveur de source de données peut renvoyer des résultats DECFLOAT hors de l'intervalle du type de données DECIMAL et générer l'erreur SQLSTATE 560BD.</p> <p>Restriction : Cette option de serveur est uniquement valide pour DB2 Database pour Linux, UNIX et Windows version 9.7 et ultérieure.</p>
OLD_NAME_GEN	<p>Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.</p>
PUSHDOWN	<p>Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y ; la source de données évalue les opérations. N spécifie que le serveur fédéré envoie les instructions SQL qui ne contiennent que SELECT avec les noms de colonne. Les prédicats, comme WHERE= ; les fonctions de colonne et fonctions scalaires, comme MAX et MIN ; les tris, comme ORDER BY OR GROUP BY ; et les jointures ne sont pas inclus dans les instructions SQL que le serveur fédéré envoie à la source de données.</p>

Tableau 53. Options de serveur pour les sources de données DB2 (suite)

Nom	Description
SAME_DECFLT_ROUNDING	<p>Indique si le mode d'arrondissement du serveur fédéré et du serveur de source de données utilise les mêmes paramètres de mode d'arrondissement DECFLOAT. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur fédéré et le serveur distant ont les mêmes paramètres de mode d'arrondissement DECFLOAT.</p> <p>Important : Si vous définissez cette option sur Y lorsque les modes d'arrondissement diffèrent entre le serveur fédéré et le serveur de source de données, il est possible que vous obteniez des résultats d'arrondissement DECFLOAT incorrects.</p> <p>Pour configurer un serveur fédéré et un serveur de source de données existants possédant le même paramètre d'arrondissement DECFLOAT, utilisez l'instruction ALTER SERVER.</p> <p>Restriction : Cette option de serveur est uniquement valide pour DB2 Database pour Linux, UNIX et Windows version 9.5 et ultérieure.</p>
VARCHAR2_COMPAT	<p>Indique si la source de données distante est compatible VARCHAR2. Les valeurs valides sont Y et N. La valeur par défaut varie selon votre source de données distante. Pour les sources de données Oracle, la valeur par défaut est Y ; la source de données est compatible VARCHAR2. Pour toutes les sources de données distantes, la valeur par défaut est N ; la source de données n'est pas définie sur le mode compatible VARCHAR2.</p> <p>Vous devez définir cette option de serveur sur Y si la source de données DB2 Database pour Linux, UNIX et Windows, ODBC ou JDBC est configurée en mode compatible VARCHAR2.</p>

Options de mappage utilisateur

Tableau 54. Options de mappage utilisateur pour les sources de données DB2

Option	Description
FED_PROXY_USER	<p>Spécifie l'ID d'autorisation à utiliser pour établir toutes les connexions accréditées sortantes lorsque la connexion entrante est non accréditée. L'utilisateur dont l'ID est spécifié avec cette option doit avoir un mappage utilisateur spécifiant un REMOTE_AUTHID et un REMOTE_PASSWORD. Si vous spécifiez l'option de mappage utilisateur FED_PROXY_USER, vous devez également spécifier l'option de serveur FED_PROXY_USER.</p> <p>Restriction : Cette option du serveur est uniquement valide pour DB2 Database pour Linux, UNIX et Windows version 9.5 et ultérieure et DB2 for z/OS Version 9 et ultérieure.</p>
ACCOUNTING_STRING	<p>Obligatoire si les informations comptables doivent être transmises. Spécifie une chaîne comptable DRDA. Les valeurs valides incluent toute chaîne comprenant jusqu'à 255 caractères.</p>
REMOTE_AUTHID	<p>Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.</p>
REMOTE_PASSWORD	<p>Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.</p>
USE_TRUSTED_CONTEXT	<p>Indique si le mappage utilisateur est accrédité. Les valeurs valides sont Y et N. La valeur par défaut est N ; le mappage utilisateur n'est pas accrédité et peut être uniquement utilisé dans des connexions sortantes fédérées non accréditées. Y indique que le mappage utilisateur est accrédité et peut être utilisé à la fois dans des connexions fédérées sortantes accréditées et non accréditées.</p> <p>Restriction : Cette option de serveur est uniquement valide pour DB2 Database pour Linux, UNIX et Windows version 9.5 et ultérieure et DB2 for z/OS version 9 et ultérieure.</p>

Options de colonne

Tableau 55. Options de colonne pour les sources de données DB2

Option	Description
NUMERIC_STRING	Indique comment gérer les chaînes numériques. La valeur par défaut est N. Si la colonne de chaîne de source de données ne contient que des chaînes numériques et aucun autre caractère, y compris les espaces vides, définissez l'option NUMERIC_STRING sur Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient aucun espace vide susceptible d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.
NO_EMPTY_STRING	Indique si le serveur source de données distant peut contenir des chaînes vides. Les valeurs valides sont Y et N. La valeur par défaut varie selon votre source de données distante. Pour les sources de données distantes Oracle, la valeur par défaut est Y ; toutes les valeurs de chaîne vides sont converties en valeurs NULL. Pour toutes les autres sources de données distantes, la valeur par défaut est N ; la source de données peut contenir des chaînes vides.
XML_ROOT	Spécifie l'élément racine XML à ajouter aux valeurs d'une colonne XML faisant référence à une séquence XML. Cette option veille à ce que les valeurs de la colonne XML forment un document XML syntaxiquement correct.

Référence des options Excel

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, et des options de pseudonyme.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 56. Options d'encapsuleur pour Excel

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.

Options de serveur

Tableau 57. Options de serveur pour Excel

Nom	Description
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.

Options de pseudonyme

Tableau 58. Options de pseudonyme pour Excel

Option	Description
FILE_PATH	Obligatoire. Spécifie le nom de fichier et le chemin d'accès au répertoire qualifié complet du tableur Excel auquel vous souhaitez accéder.
RANGE	Spécifie la plage de cellules à utiliser, par exemple, A1:C100. La valeur avant les deux-points correspond à la cellule supérieure gauche de la plage de cellules. La valeur située après les deux-points correspond à la cellule inférieure droite de la plage de cellules.

Référence des options Informix

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 59. Options d'encapsuleur pour Informix

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Options de serveur

Tableau 60. Options de serveur pour Informix

Nom	Description
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.
COMM_RATE	Spécifie le débit de communication, en mégaoctets par seconde, entre le serveur fédéré et le serveur source de données. Les valeurs valides sont supérieures à 0 et inférieures à 2147483648. La valeur par défaut est 2.

Tableau 60. Options de serveur pour Informix (suite)

Nom	Description
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.
DBNAME	Obligatoire. Spécifie le nom de la base de données Informix à laquelle vous souhaitez accéder.
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données. Si plusieurs plans d'accès remplissent les critères, c'est le plan dont le coût est le moins élevé qui est choisi.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 spécifie que l'optimiseur de requêtes fédéré détermine le nombre de demandes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
DB2_TWO_PHASE_COMMIT	Spécifie si le serveur fédéré se connecte à la source de données dans un protocole de validation à deux phases ou un protocole de validation à une phase. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur fédéré utilise le protocole de validation à une phase pour se connecter. Y spécifie que le serveur fédéré utilise un protocole de validation à deux phases pour se connecter.

Tableau 60. Options de serveur pour Informix (suite)

Nom	Description
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
FOLD_ID	Spécifie la casse de l'ID utilisateur envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie l'ID utilisateur en majuscule ; si l'ID utilisateur en majuscule échoue, le serveur envoie alors l'ID utilisateur en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
FOLD_PW	Spécifie la casse du mot de passe envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie le mot de passe en majuscule ; si le mot de passe en majuscule échoue, le serveur envoie alors le mot de passe en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
INFORMIX_CLIENT_LOCALE	Spécifie la valeur CLIENT_LOCALE à utiliser pour la connexion entre le serveur fédéré et le serveur de source de données. La valeur correspond à tout environnement local Informix valide. Si vous ne spécifiez pas cette option, la variable d'environnement CLIENT_LOCALE est définie sur la valeur qui est spécifiée dans le fichier db2dj.ini. Si la variable db2dj.ini ne spécifie pas la variable d'environnement CLIENT_LOCALE, l'option INFORMIX_CLIENT_LOCALE est définie sur l'environnement local Informix qui correspond le plus à la page de codes et au territoire de la base de données fédérée.

Tableau 60. Options de serveur pour Informix (suite)

Nom	Description
INFORMIX_DB_LOCALE	Spécifie la variable d'environnement DB_LOCALE Informix à utiliser pour la connexion entre le serveur fédéré et le serveur de source de données. Si l'option INFORMIX_DB_LOCALE n'est pas spécifiée, la variable d'environnement DB_LOCALE Informix est définie sur la valeur spécifiée dans le fichier db2dj.ini. Si le fichier db2dj.ini ne spécifie aucune valeur, la variable d'environnement DB_LOCALEInformix DB_LOCALE n'est pas définie.
INFORMIX_LOCK_MODE	Spécifie le mode de verrouillage à définir pour une source de données Informix. L'encapsuleur Informix émet la commande SET LOCK MODE immédiatement après s'être connecté à une source de données Informix. Les valeurs valides sont W, N et un nombre. La valeur par défaut est W ; l'encapsuleur attend que le verrou soit relâché pendant une durée illimitée. N indique de ne pas attendre ; une erreur est immédiatement renvoyée. Utilisez un nombre pour spécifier la durée d'attente maximale, en secondes. Si un interblocage ou un délai d'attente se produit, utilisez l'instruction ALTER SERVER pour modifier la valeur de l'option INFORMIX_LOCK_MODE. Par exemple : ALTER SERVER TYPE informix VERSION 9 WRAPPER informix OPTIONS (ADD informix_lock_mode '60')
IO_RATIO	Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.

Tableau 60. Options de serveur pour Informix (suite)

Nom	Description
IUD_APP_SVPT_ENFORCE	Indique si le serveur fédéré applique l'utilisation des instructions de point de sauvegarde des applications. Les valeurs valides sont Y et N. La valeur par défaut est Y ; si la source de données n'applique pas les instructions de point de sauvegarde des applications et qu'une erreur se produit lors d'une opération d'insertion, de mise à jour ou de suppression, le serveur fédéré annule la transaction et le code d'erreur SQL SQL1476N est renvoyé. Il est recommandé d'utiliser la valeur par défaut.
NODE	Obligatoire. Spécifie le nom selon lequel la source de données est définie en tant qu'instance dans son système de gestion de base de données relationnelle.
OLD_NAME_GEN	Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.
PUSHDOWN	Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y ; la source de données évalue les opérations. N spécifie que le serveur fédéré envoie les instructions SQL qui ne contiennent que SELECT avec les noms de colonne. Les prédicats, comme WHERE=, les fonctions de colonne et fonctions scalaires, comme MAX et MIN, les tris, comme ORDER BY OR GROUP BY, et les jointures ne sont pas inclus dans les instructions SQL que le serveur fédéré envoie à la source de données.

Options de mappage utilisateur

Tableau 61. Options de mappage utilisateur pour Informix

Nom	Description
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.

Tableau 61. Options de mappage utilisateur pour Informix (suite)

Nom	Description
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.

Options de colonne

Tableau 62. Options de colonne pour Informix

Nom	Description
NUMERIC_STRING	Indique comment gérer les chaînes numériques. La valeur par défaut est N. Si la colonne de chaîne de source de données ne contient que des chaînes numériques et aucun autre caractère, y compris les espaces vides, définissez l'option NUMERIC_STRING sur Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient aucun espace vide susceptible d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.

Références des options JDBC

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 63. Options d'encapsuleur pour JDBC

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. La seule valeur valide est Y car le serveur DB2 ne prend en charge que le chargement de la machine virtuelle Java en mode protégé. La valeur par défaut est Y. L'encapsuleur est exécuté en mode protégé.

Tableau 63. Options d'encapsuleur pour JDBC (suite)

Nom	Description
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Options de serveur

Tableau 64. Options de serveur pour JDBC

Nom	Description
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.
COMM_RATE	Spécifie le débit de communication, en mégaoctets par seconde, entre le serveur fédéré et le serveur source de données. Les valeurs valides sont supérieures à 0 et inférieures à 2147483648. La valeur par défaut est 2.
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.

Tableau 64. Options de serveur pour JDBC (suite)

Nom	Description
DATEFORMAT	Spécifie le format de date que la source de données utilise. Utilisez 'DD', 'MM' et 'YY' ou 'YYYY' pour spécifier le format de date. Vous pouvez spécifier un délimiteur comme un espace, un trait d'union ou une virgule. Par exemple, le format 'YYYY-MM-DD' spécifie une date comme 1958-10-01. La valeur peut contenir des valeurs nulles.
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 0. -1 spécifie que l'optimiseur de requêtes fédéré détermine le nombre de demandes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Tableau 64. Options de serveur pour JDBC (suite)

Nom	Description
DRIVER_CLASS	<p>Indique la bibliothèque des pilotes JDBC. Vous pouvez enregistrer le serveur pour plusieurs sources de données JDBC si le pilote JDBC est compatible avec la version de spécification JDBC 3.0 ou une version ultérieure. Consultez la documentation de votre pilote JDBC pour obtenir des informations sur les spécifications JDBC et la configuration de l'option de serveur DRIVER_CLASS.</p> <p>Exemple</p> <p>Dans cet exemple, la bibliothèque de pilotes JDBC com.ibm.db2.jcc.DB2Driver est indiquée :</p> <pre>DRIVER_CLASS 'com.ibm.db2.jcc.DB2Driver'</pre> <p>Important : Si vous définissez cette option, vous devez également définir l'option de serveur URL.</p>
DRIVER_PACKAGE	<p>Spécifie les modules des pilotes JDBC. Utilisez un séparateur de chemin d'accès pour spécifier plusieurs modules de classe de pilote. Utilisez un point-virgule sur les systèmes d'exploitation Windows et deux points sur les systèmes d'exploitation Linux et UNIX.</p> <p>Exemple</p> <p>Dans cet exemple, vous spécifiez plusieurs modules de pilotes avec deux points sur des systèmes d'exploitation Linux :</p> <pre>DRIVER_PACKAGE '/path1/file1.jar; /path2/file2.jar'</pre>
FOLD_ID	<p>Spécifie la casse de l'ID utilisateur envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie l'ID utilisateur en majuscule ; si l'ID utilisateur en majuscule échoue, le serveur envoie alors l'ID utilisateur en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.</p>
FOLD_PW	<p>Spécifie la casse du mot de passe envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie le mot de passe en majuscule ; si le mot de passe en majuscule échoue, le serveur envoie alors le mot de passe en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.</p>

Tableau 64. Options de serveur pour JDBC (suite)

Nom	Description
IO_RATIO	<p>Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23}. La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.</p>
JDBC_LOG	<p>Indique si l'encapsuleur JDBC crée des fichiers journaux pour la fonction de trace des erreurs. Les valeurs valides sont Y et N. La valeur par défaut est N. Il n'y pas de fichier journal créé. Si cette option de serveur est définie sur Y, l'encapsuleur JDBC enregistre les fichiers journaux JDBC dans le fichier <code>jdbc_wrapper_ID_produit.log</code> où <code>ID_produit</code> correspond à l'ID du produit. Le fichier journal est stocké dans le répertoire indiqué par le paramètre de configuration du gestionnaire de base de données DB2 DIAGPATH. Le répertoire par défaut sur les systèmes UNIX est <code>inst_home/sqllib/db2dump</code>.</p> <p>Recommandation : Définir cette option de serveur sur YES a un impact sur les performances de votre système. Il n'est pas conseillé d'activer la consignment sur des systèmes de production.</p>
OLD_NAME_GEN	<p>Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.</p>

Tableau 64. Options de serveur pour JDBC (suite)

Nom	Description
PUSHDOWN	Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y ; la source de données évalue les opérations. N spécifie que le serveur fédéré envoie les instructions SQL qui ne contiennent que SELECT avec les noms de colonne. Les prédicats, comme WHERE= ; les fonctions de colonne et fonctions scalaires, comme MAX et MIN ; les tris, comme ORDER BY OR GROUP BY ; et les jointures ne sont pas inclus dans les instructions SQL que le serveur fédéré envoie à la source de données.
TIMEFORMAT	Spécifie le format horaire que la source de données utilise. Utilisez 'hh12', 'hh24', 'mm', 'ss', 'AM' et 'A.M.' pour spécifier le format horaire. Par exemple, le format 'hh24:mm:22' spécifie une heure telle que 16:00:00. Le format 'hh12:mm:ss AM' spécifie une heure telle que 8:00:00 AM. La valeur peut contenir des valeurs nulles.
TIMESTAMPFORMAT	Spécifie le format d'horodatage que la source de données utilise. Les valeurs valides sont au format utilisé par les options DATEFORMAT et TIMEFORMAT. Spécifiez 'n' pour un dixième de seconde, 'nn' pour un centième de seconde, 'nnn' pour des millisecondes, et ainsi de suite, jusqu'à 'nnnnnn' pour des microsecondes. Par exemple, le format 'YYY-MM-DD-hh24:mm:ss.nnnnnn' spécifie un horodatage de type 1994-01-01-24:00:00.000000. La valeur peut contenir des valeurs nulles.

Tableau 64. Options de serveur pour JDBC (suite)

Nom	Description
URL	<p>Spécifie la chaîne de connexion JDBC du serveur distant.</p> <p>La chaîne de connexion JDBC est constituée de trois parties, séparées par deux points :</p> <ul style="list-style-type: none"> • Protocole de base de données • Nom du type de base de données ou nom du driver de connectivité • Identité de la base de données, via un alias ou un sous-nom <p>Exemple</p> <p>Dans cet exemple, la chaîne de connexion JDBC est</p> <pre>jdbc:db2://cn.ibm.com:50471/testdb :</pre> <p>URL</p> <pre>'jdbc:db2://cn.ibm.com:50471/testdb'</pre> <p>Vous pouvez enregistrer le serveur pour plusieurs sources de données JDBC si le pilote JDBC est compatible avec la version de spécification JDBC 3.0 ou une version supérieure. Consultez la documentation de votre pilote JDBC pour obtenir des informations sur les spécifications JDBC et la configuration de l'option de serveur URL.</p> <p>Important : Si vous définissez cette option, vous devez également définir l'option de serveur DRIVER_CLASS.</p>
VARCHAR_NO_TRAILING_BLANKS	<p>Spécifie si la source de données contient des colonnes VARCHAR ayant au moins un caractère blanc de fin. La valeur par défaut est N ; les colonnes VARCHAR contiennent au moins un caractère blanc de fin.</p>

Options de mappage utilisateur

Tableau 65. Options de mappage utilisateur pour JDBC

Nom	Description
REMOTE_AUTHID	<p>Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.</p>
REMOTE_PASSWORD	<p>Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.</p>

Options de colonne

Tableau 66. Options de colonne pour JDBC

Nom	Description
NUMERIC_STRING	Spécifie si la colonne contient des chaînes de caractères numériques incluant des blancs. Les valeurs valides sont Y et N. La valeur par défaut est N ; la colonne ne contient pas de chaînes numériques incluant des blancs. Si la colonne contient uniquement des chaînes numériques suivies par des blancs de fin, ne spécifiez pas Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient pas de blancs susceptibles d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.
VARCHAR_NO_TRAILING_BLANKS	Indique s'il existe au moins un blanc de fin dans la colonne VARCHAR.

Référence des options Microsoft SQL Server

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 67. Options d'encapsuleur pour Microsoft SQL Server

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.

Tableau 67. Options d'encapsuleur pour Microsoft SQL Server (suite)

Nom	Description
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Options de serveur

Tableau 68. Options de serveur pour Microsoft SQL Server

Nom	Description
CODEPAGE	Spécifie la page de codes correspondant au jeu de caractères codés de la configuration client de la source de données. Sous les systèmes UNIX et Microsoft Windows qui utilisent une base de données fédérée non Unicode, la valeur par défaut est la page de codes de la base de données fédérée. Sous les systèmes UNIX qui utilisent une base de données fédérée Unicode, la valeur par défaut est 1208. Sous les systèmes Windows qui utilisent une base de données fédérée Unicode, la valeur par défaut est 1202.
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.

Tableau 68. Options de serveur pour Microsoft SQL Server (suite)

Nom	Description
COMM_RATE	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4.
DBNAME	Obligatoire. Spécifie les alias de la base de données à laquelle vous souhaitez accéder. La valeur est sensible à la casse.
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données. Si plusieurs plans d'accès remplissent les critères, c'est le plan dont le coût est le moins élevé qui est choisi.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.

Tableau 68. Options de serveur pour Microsoft SQL Server (suite)

Nom	Description
DB2_TWO_PHASE_COMMIT	Spécifie si le serveur fédéré se connecte à la source de données dans un protocole de validation à deux phases ou un protocole de validation à une phase. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur fédéré utilise le protocole de validation à une phase pour se connecter. Y spécifie que le serveur fédéré utilise un protocole de validation à deux phases pour se connecter. Important : Si vous définissez cette option sur Y, vous devez également spécifier XA_OPEN_STRING_OPTION.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
FOLD_ID	Spécifie la casse de l'ID utilisateur envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie l'ID utilisateur en majuscule ; si l'ID utilisateur en majuscule échoue, le serveur envoie alors l'ID utilisateur en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
FOLD_PW	Spécifie la casse du mot de passe envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie le mot de passe en majuscule ; si le mot de passe en majuscule échoue, le serveur envoie alors le mot de passe en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.

Tableau 68. Options de serveur pour Microsoft SQL Server (suite)

Nom	Description
IO_RATIO	<p>Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1x10²³. La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.</p>
NODE	<p>Obligatoire. Si le serveur fédéré utilise Microsoft Windows, la valeur de NODE est le nom de système DSN que vous spécifiez pour Microsoft SQL Server. Si le serveur fédéré utilise UNIX ou Linux, la valeur de NODE est définie dans le fichier odbc.ini. La valeur est sensible à la casse.</p>
OLD_NAME_GEN	<p>Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.</p>
PASSWORD	<p>Spécifie si les mots de passe sont envoyés ou non à la source de données. La valeur par défaut est Y.</p>
PUSHDOWN	<p>Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y ; la source de données évalue les opérations. N spécifie que le serveur fédéré envoie les instructions SQL qui ne contiennent que SELECT avec les noms de colonne. Les prédicats, comme WHERE= ; les fonctions de colonne et fonctions scalaires, comme MAX et MIN ; les tris, comme ORDER BY OR GROUP BY ; et les jointures ne sont pas inclus dans les instructions SQL que le serveur fédéré envoie à la source de données.</p>

Tableau 68. Options de serveur pour Microsoft SQL Server (suite)

Nom	Description
XA_OPEN_STRING_OPTIONS	Obligatoire lorsque DB2_TWO_PHASE_COMMIT est défini sur Y. Spécifie l'ID du gestionnaire de ressources du registre Microsoft SQL Server.

Options de mappage utilisateur

Tableau 69. Options de mappage utilisateur pour Microsoft SQL Server

Nom	Description
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.

Options de colonne

Tableau 70. Options de colonne pour Microsoft SQL Server

Nom	Description
NUMERIC_STRING	Indique comment gérer les chaînes numériques. La valeur par défaut est N. Si la colonne de chaîne de source de données ne contient que des chaînes numériques et aucun autre caractère, y compris les espaces vides, définissez l'option NUMERIC_STRING sur Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient aucun espace vide susceptible d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.

Référence des options ODBC

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 71. Options d'encapsuleur pour ODBC

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité. Important : Si vous définissez cette option sur Y pour un système UNIX, vous devez également définir l'option d'encapsuleur DB2_SOURCE_CLIENT_MODE.
DB2_SOURCE_CLIENT_MODE	Spécifie que le client de la source de données est en 32 bits et que l'instance de base de données sur le serveur fédéré est en 64 bits. La seule valeur valide est 32 bits. Cette option n'est valide que pour UNIX. Important : Si vous définissez cette option, vous devez également définir l'option d'encapsuleur DB2_FENCED sur Y.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
MODULE	Obligatoire pour les serveurs fédérés qui sont exécutés sur un système UNIX. Spécifie le chemin complet de la bibliothèque qui contient l'implémentation du gestionnaire de pilote ODBC ou l'implémentation SQL/CLI. Il n'existe pas de valeur par défaut pour UNIX. Sur un système Microsoft Windows, la valeur par défaut est odb32.dll.

Options de serveur

Tableau 72. Options de serveur pour ODBC

Nom	Description
CODEPAGE	Spécifie la page de codes correspondant au jeu de caractères codés de la configuration client de la source de données. Sous les systèmes UNIX et Windows qui utilisent une base de données fédérée non Unicode, la valeur par défaut est la page de codes utilisée par la base de données fédérée. Sous les systèmes UNIX qui utilisent une base de données fédérée Unicode, la valeur par défaut est 1208. Sous les systèmes Windows qui utilisent une base de données fédérée Unicode, la valeur par défaut est 1202.
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.
COMM_RATE	Spécifie le débit de communication, en mégaoctets par seconde, entre le serveur fédéré et le serveur source de données. Les valeurs valides sont supérieures à 0 et inférieures à 2147483648. La valeur par défaut est 2.
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.
DATEFORMAT	Spécifie le format de date que la source de données utilise. Utilisez 'DD', 'MM' et 'YY' ou 'YYYY' pour spécifier le format de date. Vous pouvez spécifier un délimiteur comme un espace, un trait d'union ou une virgule. Par exemple, le format 'YYYY-MM-DD' spécifie une date comme 1958-10-01. La valeur peut contenir des valeurs nulles.

Tableau 72. Options de serveur pour ODBC (suite)

Nom	Description
DBNAME	Spécifie le nom de la base de données de la source de données à laquelle vous souhaitez accéder.
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 0. -1 spécifie que l'optimiseur de requêtes fédéré détermine le nombre de demandes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
FOLD_ID	Spécifie la casse de l'ID utilisateur envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie l'ID utilisateur en majuscule ; si l'ID utilisateur en majuscule échoue, le serveur envoie alors l'ID utilisateur en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
FOLD_PW	Spécifie la casse du mot de passe envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie le mot de passe en majuscule ; si le mot de passe en majuscule échoue, le serveur envoie alors le mot de passe en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.

Tableau 72. Options de serveur pour ODBC (suite)

Nom	Description
IO_RATIO	<p>Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23}. La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.</p>
NODE	<p>Obligatoire. Spécifie le nom du noeud ou le nom DNS du système affecté à la source de données ODBC lorsque le DSN est défini. La valeur est sensible à la casse.</p>
OLD_NAME_GEN	<p>Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.</p>
PUSHDOWN	<p>Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y ; la source de données évalue les opérations. N spécifie que le serveur fédéré envoie les instructions SQL qui ne contiennent que SELECT avec les noms de colonne. Les prédicats, comme WHERE= ; les fonctions de colonne et fonctions scalaires, comme MAX et MIN ; les tris, comme ORDER BY OR GROUP BY ; et les jointures ne sont pas inclus dans les instructions SQL que le serveur fédéré envoie à la source de données.</p>

Tableau 72. Options de serveur pour ODBC (suite)

Nom	Description
TIMEFORMAT	Spécifie le format horaire que la source de données utilise. Utilisez 'hh12', 'hh24', 'mm', 'ss', 'AM' et 'A.M.' pour spécifier le format horaire. Par exemple, le format 'hh24:mm:22' spécifie une heure telle que 16:00:00. Le format 'hh12:mm:ss AM' spécifie une heure telle que 8:00:00 AM. La valeur peut contenir des valeurs nulles.
TIMESTAMPFORMAT	Spécifie le format d'horodatage que la source de données utilise. Les valeurs valides sont au format utilisé par les options DATEFORMAT et TIMEFORMAT. Spécifiez 'n' pour un dixième de seconde, 'nn' pour un centième de seconde, 'nnn' pour des millisecondes, et ainsi de suite, jusqu'à 'nnnnnn' pour des microsecondes. Par exemple, le format 'YYY-MM-DD-hh24:mm:ss.nnnnnn' spécifie un horodatage de type 1994-01-01-24:00:00.000000. La valeur peut contenir des valeurs nulles.
VARCHAR_NO_TRAILING_BLANKS	Spécifie si la source de données contient des colonnes VARCHAR ayant au moins un caractère blanc de fin. La valeur par défaut est N ; les colonnes VARCHAR contiennent au moins un caractère blanc de fin.

Options de mappage utilisateur

Tableau 73. Options de mappage utilisateur pour ODBC

Nom	Description
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.

Options de colonne

Tableau 74. Options de colonne pour ODBC

Nom	Description
NUMERIC_STRING	Spécifie si la colonne contient des chaînes de caractères numériques incluant des blancs. Les valeurs valides sont Y et N. La valeur par défaut est N ; la colonne ne contient pas de chaînes numériques incluant des blancs. Si la colonne contient uniquement des chaînes numériques suivies par des blancs de fin, ne spécifiez pas Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient pas de blancs susceptibles d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.
VARCHAR_NO_TRAILING_BLANKS	Indique s'il existe au moins un blanc de fin dans la colonne VARCHAR.

Référence des options Oracle

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 75. Options d'encapsuleur pour Oracle

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.

Tableau 75. Options d'encapsuleur pour Oracle (suite)

Nom	Description
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Options de serveur

Tableau 76. Options de serveur pour Oracle

Nom	Description
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.
COMM_RATE	Spécifie le débit de communication, en mégaoctets par seconde, entre le serveur fédéré et le serveur source de données. Les valeurs valides sont supérieures à 0 et inférieures à 2147483648. La valeur par défaut est 2.
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.

Tableau 76. Options de serveur pour Oracle (suite)

Nom	Description
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
DB2_TWO_PHASE_COMMIT	Spécifie si le serveur fédéré se connecte à la source de données dans un protocole de validation à deux phases ou un protocole de validation à une phase. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur fédéré utilise le protocole de validation à une phase pour se connecter. Y spécifie que le serveur fédéré utilise un protocole de validation à deux phases pour se connecter.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
FED_PROXY_USER	Spécifie l'ID d'autorisation à utiliser pour établir toutes les connexions accréditées sortantes lorsque la connexion entrante est non accréditée. Important : L'utilisateur dont l'ID est spécifié avec cette option doit avoir un mappage utilisateur spécifiant un REMOTE_AUTHID et un REMOTE_PASSWORD.

Tableau 76. Options de serveur pour Oracle (suite)

Nom	Description
FOLD_ID	Spécifie la casse de l'ID utilisateur envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie l'ID utilisateur en majuscule ; si l'ID utilisateur en majuscule échoue, le serveur envoie alors l'ID utilisateur en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
FOLD_PW	Spécifie la casse du mot de passe envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie le mot de passe en majuscule ; si le mot de passe en majuscule échoue, le serveur envoie alors le mot de passe en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
IO_RATIO	Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.
NODE	Obligatoire. Spécifie le nom du noeud où le serveur de base de données Oracle réside. Obtenez le nom du noeud à partir du fichier tnsnames.ora.
OLD_NAME_GEN	Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.

Tableau 76. Options de serveur pour Oracle (suite)

Nom	Description
PLAN_HINTS	Spécifie si des algorithmes d'optimisation de plan doivent être activés. Les algorithmes d'optimisation de plan sont des fragments d'instruction qui fournissent des informations supplémentaires que l'optimiseur de source de données utilise pour améliorer les performances de la requête. L'optimiseur de source de données utilise les algorithmes d'optimisation de plan pour décider s'il doit utiliser un index ou le type d'index ou de séquence de jointure de table à utiliser. Les valeurs valides sont Y et N. La valeur par défaut est N. Les algorithmes d'optimisation de plan doivent être activés. Y spécifie que les algorithmes d'optimisation de plan doivent être activés au niveau de la source de données s'ils sont pris en charge.
PUSHDOWN	Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y. Le serveur fédéré autorise la source de données à évaluer les opérations. N spécifie que le serveur fédéré récupère les colonnes à partir de la source de données distante.
VARCHAR_NO_TRAILING_BLANKS	Indique pour un serveur particulier si des colonnes VARCHAR contiennent des blancs de fin. Pour appliquer l'option à une seule colonne, utilisez l'option de colonne VARCHAR_NO_TRAILING_BLANKS.
XA_OPEN_STRING_OPTIONS	Spécifie des informations supplémentaires à ajouter à la chaîne. Par exemple, il peut s'agir du répertoire pour les fichiers de trace.

Options de mappage utilisateur

Tableau 77. Options de mappage utilisateur pour Oracle

Nom	Description
FED_PROXY_USER	Spécifie l'ID d'autorisation à utiliser pour établir toutes les connexions accréditées sortantes lorsque la connexion entrante est non accréditée. Important : L'utilisateur dont l'ID est spécifié avec cette option doit avoir un mappage utilisateur spécifiant un REMOTE_AUTHID et un REMOTE_PASSWORD. Si vous spécifiez l'option de mappage utilisateur FED_PROXY_USER, vous devez également spécifier l'option de serveur FED_PROXY_USER.

Tableau 77. Options de mappage utilisateur pour Oracle (suite)

Nom	Description
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.
USE_TRUSTED_CONTEXT	Indique si le mappage utilisateur est accrédité. Les valeurs valides sont Y et N. La valeur par défaut est N ; le mappage utilisateur n'est pas accrédité et ne peut être utilisé que dans des connexions sortantes non accréditées. Y spécifie que le mappage utilisateur est accrédité et peut être utilisé dans les connexions sortantes accréditées et non accréditées.

Options de colonne

Tableau 78. Options de colonne pour Oracle

Nom	Description
NUMERIC_STRING	Spécifie si la colonne contient des chaînes de caractères numériques incluant des blancs. Les valeurs valides sont Y et N. La valeur par défaut est N ; la colonne ne contient pas de chaînes numériques incluant des blancs. Si la colonne contient uniquement des chaînes numériques suivies par des blancs de fin, ne spécifiez pas Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient pas de blancs susceptibles d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.
VARCHAR_NO_TRAILING_BLANKS	Indique si la colonne contient des blancs de fin.

Référence des options de script

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur, un pseudonyme et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 79. Options d'encapsuleur pour les scripts

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE et SOCKS. La valeur par défaut est NONE.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.

Options de serveur

Tableau 80. Options de serveur pour les scripts

Nom	Description
DAEMON_PORT	Spécifie le numéro de port sur lequel le démon Script écoute les demandes de travaux Script. La valeur par défaut est 4099. Le numéro de port doit être le même que pour l'option DAEMON_PORT dans le fichier de configuration du démon. Si un nom de service est configuré pour le démon Script, utilisez un nom de service TCP/IP.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.

Tableau 80. Options de serveur pour les scripts (suite)

Nom	Description
NODE	Obligatoire. Spécifie le nom d'hôte DNS ou l'adresse IP du système sur lequel le démon Script est exécuté. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le port ou le nom de service du service proxy sur le serveur proxy. Les valeurs valides sont un numéro de port décimal allant de 1 à 32760 ou un nom de service.
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE et SOCKS. La valeur par défaut est NONE.

Options de mappage utilisateur

Tableau 81. Options de mappage utilisateur pour les scripts

Nom	Description
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy. Le mot de passe est chiffré lorsqu'il est stocké dans le catalogue de la base de données fédérée.

Options de pseudonyme

Tableau 82. Options de pseudonyme pour les scripts

Nom	Description
DATASOURCE	Obligatoire pour le pseudonyme racine. Spécifie le nom du script à appeler. Le script que vous spécifiez comme valeur de cette option doit également être spécifié dans le fichier de configuration du démon Script. Important : Cette option n'est valide que pour le pseudonyme racine.

Tableau 82. Options de pseudonyme pour les scripts (suite)

Nom	Description
NAMESPACES	<p>Spécifie les espaces de nom associés aux préfixes d'espace de nom utilisés dans les options XPATH et TEMPLATE pour chaque colonne. Utilisez cette syntaxe :</p> <pre>NAMESPACES'prefix1="actual_namespace1", prefix2="actual_namespace2"'</pre> <p>Utilisez une virgule pour séparer plusieurs espaces de nom. Par exemple :</p> <pre>NAMESPACES='http://www.myweb.com/cust', i='http://www.myweb.com/cust/id', n='http://www.myweb.com/cust/name''</pre>
STREAMING	<p>Spécifie si le document source doit être séparé en fragments logiques pour le traitement. Les fragments concordent avec le noeud correspondant à l'expression XPath du pseudonyme. L'encapsuleur procède ensuite à une analyse syntaxique et traite les données source fragment par fragment. Ce type d'analyse syntaxique minimise l'utilisation de la mémoire. Les valeurs valides sont Y et N. La valeur par défaut est N ; les documents ne sont pas soumis à une analyse syntaxique. Cette option n'est valide que sur le pseudonyme racine. Ne définissez pas les deux options STREAMING et VALIDATE sur Y.</p>
TIMEOUT	<p>Spécifie la durée d'attente maximale, en minutes, des résultats provenant du démon. La valeur par défaut est 60. Cette option n'est valide que pour le pseudonyme racine.</p>
VALIDATE	<p>Spécifie si le document source est validé pour garantir sa conformité au schéma XML ou à une définition de type de document (DTD) avant d'en extraire des données. La valeur par défaut est N ; la validation n'a pas lieu. Avant de définir la valeur sur Y, le fichier schéma ou le fichier DTD se trouve à l'emplacement spécifié par le document source. Cette option n'est valide que pour le pseudonyme racine. Ne définissez pas les deux options STREAMING et VALIDATE sur Y.</p>
XPATH	<p>Spécifie l'expression XPath qui identifie les éléments XML représentant des tuples individuels. L'option de pseudonyme XPATH pour un pseudonyme enfant est évaluée dans le contexte du chemin spécifié par l'option de pseudonyme XPATH de son parent. Cette expression XPath est utilisée comme un contexte permettant d'évaluer les valeurs de colonne identifiées par la présence de l'option de colonne XPATH.</p>

Options de colonne

Tableau 83. Options de colonne pour les scripts

Nom	Description
DEFAULT	Spécifie une valeur par défaut d'une colonne d'entrée de script. Si la requête SQL ne fournit pas de valeur, la valeur par défaut est utilisée.
FOREIGN_KEY	Indique que ce pseudonyme est un pseudonyme enfant et spécifie le nom du pseudonyme parent correspondant. Un pseudonyme peut avoir au maximum une option de colonne FOREIGN_KEY. La valeur de l'option est sensible à la casse. Ne spécifiez pas l'option XPATH pour cette colonne. La colonne peut être utilisée pour associer un pseudonyme parent et un pseudonyme enfant. Une instruction CREATE NICKNAME qui comprend une option FOREIGN_KEY échoue si le pseudonyme parent a un nom de schéma différent. A moins que le pseudonyme auquel il est fait référence dans une clause FOREIGN_KEY ait été explicitement défini en minuscule ou dans une casse mixte dans l'instruction CREATE NICKNAME, vous devez spécifier le pseudonyme en majuscule lorsque vous faites référence à ce pseudonyme dans la clause FOREIGN_KEY. Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne.
INPUT_MODE	Indique le mode entrée de la colonne. Les valeurs valides sont CONFIG et FILE_INPUT. CONFIG traite la valeur comme le mode entrée d'une colonne. FILE_INPUT spécifie un fichier qui stocke la valeur. L'encapsuleur transmet la valeur spécifiée au démon Script.
POSITION	Spécifie un entier pour les paramètres positionnels. Si cette valeur positionnelle est définie sur un entier, alors cette entrée doit être dans cette position dans la ligne de commande. Si cette option est définie, le commutateur est inséré à l'emplacement approprié lorsque la requête est exécutée. Si POSITION est défini sur -1, l'option est ajoutée comme la dernière option de ligne de commande. Un entier POSITION ne peut pas être utilisé deux fois dans le même pseudonyme. Cette option ne s'applique qu'aux colonnes d'entrée.

Tableau 83. Options de colonne pour les scripts (suite)

Nom	Description
PRIMARY_KEY	Obligatoire pour un pseudonyme parent ayant un ou plusieurs pseudonymes enfant. Spécifie que ce pseudonyme est un pseudonyme parent. Le type de données de colonne doit être VARCHAR(16). Un pseudonyme ne peut avoir qu'une seule option de colonne PRIMARY_KEY. Yes est la seule valeur valide. Ne spécifiez pas l'option XPATH pour cette colonne. La colonne ne peut être utilisée que pour associer des pseudonymes parent et des pseudonymes enfant. Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne.
SWITCH	Spécifie un indicateur pour le script sur la ligne de commande. La valeur de cette option précédait la valeur de la colonne fournie par WSSCRIPT.ARGS ou par la valeur par défaut, le cas échéant. Si vous ne spécifiez pas de valeur pour cette option et qu'une valeur par défaut existe pour la colonne, la valeur par défaut est ajoutée sans information de commutation. Cette option est obligatoire pour les colonnes d'entrée.
SWITCH_ONLY	Permet l'utilisation de commutateurs sans argument de ligne de commande. Les valeurs valides sont Y et N. Définissez cette option sur Y et les valeurs d'entrée valides sont Y et N. Pour une valeur d'entrée Y, seul le commutateur est ajouté à la ligne de commande. Pour une valeur d'entrée N, aucune valeur n'est ajoutée à la ligne de commande.
VALID_VALUES	Spécifie un ensemble de valeurs valides pour une colonne. Utilisez un point-virgule pour séparer plusieurs valeurs.
XPATH	Spécifie l'expression Xpath du document XML qui contient les données correspondant à cette colonne. L'encapsuleur évalue cette expression XPath après l'application par l'instruction CREATE NICKNAME de l'expression XPath à partir de l'option de pseudonyme XPATH.

Référence des options Sybase

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 84. Options d'encapsuleur pour Sybase

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Sous Microsoft Windows, les valeurs valides sont Y et N. La valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité. Sous UNIX, la valeur par défaut et la seule valeur valide est Y ; l'encapsuleur doit fonctionner en mode protégé.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Options de serveur

Tableau 85. Options de serveur pour Sybase

Nom	Description
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.
COMM_RATE	Spécifie le débit de communication, en mégaoctets par seconde, entre le serveur fédéré et le serveur source de données. Les valeurs valides sont supérieures à 0 et inférieures à 2147483648. La valeur par défaut est 2.

Tableau 85. Options de serveur pour Sybase (suite)

Nom	Description
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.
CONV_EMPTY_STRING	Spécifie si le serveur fédéré convertit une chaîne vide en un espace lors de tâches de réplication. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur fédéré ne convertit pas les chaînes vides. Définissez cette option sur Y lorsque la source de données présente une colonne avec un caractère ne pouvant avoir la valeur NULL qui stocke une chaîne vide.
DBNAME	Obligatoire. Spécifie le nom de la base de données à laquelle vous souhaitez accéder. Obtenez le nom de la base de données à partir du serveur Sybase.
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.

Tableau 85. Options de serveur pour Sybase (suite)

Nom	Description
DB2_TWO_PHASE_COMMIT	Spécifie si le serveur fédéré se connecte à la source de données dans un protocole de validation à deux phases ou un protocole de validation à une phase. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur fédéré utilise le protocole de validation à une phase pour se connecter. Y spécifie que le serveur fédéré utilise un protocole de validation à deux phases pour se connecter.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
FOLD_ID	Spécifie la casse de l'ID utilisateur envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie l'ID utilisateur en majuscule ; si l'ID utilisateur en majuscule échoue, le serveur envoie alors l'ID utilisateur en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
FOLD_PW	Spécifie la casse du mot de passe envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie le mot de passe en majuscule ; si le mot de passe en majuscule échoue, le serveur envoie alors le mot de passe en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.

Tableau 85. Options de serveur pour Sybase (suite)

Nom	Description
IFILE	<p>Spécifie le chemin d'accès et le nom du fichier d'interface Sybase à utiliser à la place du fichier d'interface par défaut.</p> <p>L'encapsuleur Sybase recherche le fichier d'interface dans les emplacements suivants, dans l'ordre spécifié : sous Microsoft Windows, dans l'option de serveur IFILE, puis dans le répertoire %DB2PATH%\interfaces et enfin dans le répertoire %SYBASE%\ini\sql.ini. Sous UNIX, dans l'option de serveur IFILE, puis dans le répertoire sqllib/interfaces et enfin dans le répertoire \$SYBASE/interfaces.</p>
IO_RATIO	<p>Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23}. La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4.</p> <p>Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1.</p> <p>Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.</p>
LOGIN_TIMEOUT	<p>Spécifie la durée d'attente, en secondes, du serveur fédéré avant d'abandonner une demande de connexion. La valeur par défaut est 0 ; le serveur fédéré attend pendant une durée illimitée.</p>
NODE	<p>Obligatoire. Spécifie le nom du noeud où le serveur Sybase réside. Le nom de noeud est dans le fichier des interfaces Sybase.</p>
OLD_NAME_GEN	<p>Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.</p>

Tableau 85. Options de serveur pour Sybase (suite)

Nom	Description
PACKET_SIZE	Spécifie la taille de paquet, en octets, que la bibliothèque client utilise pour envoyer des paquets TDS (Tabular Data Stream). Si l'encapsuleur Sybase a besoin d'envoyer ou de recevoir de grandes quantités de données image et texte, augmentez la valeur de PACKET_SIZE.
PLAN_HINTS	Spécifie si des algorithmes d'optimisation de plan doivent être activés. Les algorithmes d'optimisation de plan sont des fragments d'instruction qui fournissent des informations supplémentaires que l'optimiseur de source de données utilise pour améliorer les performances de la requête. L'optimiseur de source de données utilise les algorithmes d'optimisation de plan pour décider s'il doit utiliser un index ou le type d'index ou de séquence de jointure de table à utiliser. Les valeurs valides sont Y et N. La valeur par défaut est N. Les algorithmes d'optimisation de plan doivent être activés. Y spécifie que les algorithmes d'optimisation de plan doivent être activés au niveau de la source de données s'ils sont pris en charge.
PUSHDOWN	Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y ; la source de données évalue les opérations. N spécifie que le serveur fédéré envoie les instructions SQL qui ne contiennent que SELECT avec les noms de colonne. Les prédicats, comme WHERE= ; les fonctions de colonne et fonctions scalaires, comme MAX et MIN ; les tris, comme ORDER BY OR GROUP BY ; et les jointures ne sont pas inclus dans les instructions SQL que le serveur fédéré envoie à la source de données.
TIMEOUT	Spécifie la durée maximale, en secondes, au cours de laquelle le serveur fédéré attend que le serveur distant réponde à une commande. La valeur par défaut est 0, correspondant à une durée illimitée.
XA_OPEN_STRING_OPTIONS	Spécifie des chaînes ouvertes pour l'interface Sybase DTM XA. Ces chaînes s'ajoutent au nom LRM, au nom d'utilisateur et au mot de passe.

Options de mappage utilisateur

Tableau 86. Options de mappage utilisateur pour Sybase

Option	Description
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.

Options de colonne

Tableau 87. Options de colonne pour Sybase

Option	Description
NUMERIC_STRING	Indique comment gérer les chaînes numériques. La valeur par défaut est N. Si la colonne de chaîne de source de données ne contient que des chaînes numériques et aucun autre caractère, y compris les espaces vides, définissez l'option NUMERIC_STRING sur Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient aucun espace vide susceptible d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.

Référence des options Teradata

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 88. Options d'encapsuleur pour Teradata

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Sous Microsoft Windows, les valeurs valides sont Y et N. La valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité. Sous UNIX, la valeur par défaut Y ; l'encapsuleur doit fonctionner en mode protégé.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Options de serveur

Tableau 89. Options de serveur pour Teradata

Nom	Description
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.
COMM_RATE	Spécifie le débit de communication, en mégaoctets par seconde, entre le serveur fédéré et le serveur source de données. Les valeurs valides sont supérieures à 0 et inférieures à 2147483648. La valeur par défaut est 2.

Tableau 89. Options de serveur pour Teradata (suite)

Nom	Description
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Tableau 89. Options de serveur pour Teradata (suite)

Nom	Description
IO_RATIO	<p>Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23}. La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.</p>
NODE	<p>Obligatoire. Spécifie le nom d'alias ou l'adresse IP du serveur Teradata.</p>
OLD_NAME_GEN	<p>Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.</p>
PUSHDOWN	<p>Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y ; la source de données évalue les opérations. N spécifie que le serveur fédéré envoie les instructions SQL qui ne contiennent que SELECT avec les noms de colonne. Les prédicats, comme WHERE= ; les fonctions de colonne et fonctions scalaires, comme MAX et MIN ; les tris, comme ORDER BY OR GROUP BY ; et les jointures ne sont pas inclus dans les instructions SQL que le serveur fédéré envoie à la source de données.</p>

Options de mappage utilisateur

Tableau 90. Options de mappage utilisateur pour Teradata

Nom	Description
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.

Options de colonne

Tableau 91. Options de colonne pour Teradata

Nom	Description
NUMERIC_STRING	Indique comment gérer les chaînes numériques. La valeur par défaut est N. Si la colonne de chaîne de source de données ne contient que des chaînes numériques et aucun autre caractère, y compris les espaces vides, définissez l'option NUMERIC_STRING sur Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient aucun espace vide susceptible d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.

Référence des options des fichiers structurés sous forme de tables

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un pseudonyme et des options de colonne.

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Options d'encapsuleur

Tableau 92. Options d'encapsuleur pour les fichiers structurés sous forme de tables

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.

Options de serveur

Tableau 93. Options de serveur pour les fichiers structurés sous forme de tables

Nom	Description
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.

Options de pseudonyme

Tableau 94. Options de pseudonyme pour les fichiers structurés sous forme de tables

Nom	Description
COLUMN_DELIMITER	Spécifie un caractère unique à utiliser comme délimiteur pour séparer les colonnes dans le fichier structuré sous forme de tables. La valeur par défaut est une virgule (.). Un guillemet simple ne peut pas être utilisé comme délimiteur. Le délimiteur de colonne doit être cohérent sur l'ensemble du fichier. Une valeur NULL est représentée par deux délimiteurs l'un à côté de l'autre ou par un délimiteur suivi d'un caractère de fin de ligne, si la zone NULL est la dernière de la ligne. Le délimiteur de colonne ne peut pas exister en tant que donnée valide d'une colonne.
CODEPAGE	Spécifie la page de codes du fichier au niveau de la source de données. Cette option n'est valide que pour les bases de données fédérées qui utilisent Unicode. Les données source sont converties à partir de la page de codes spécifiée en Unicode.

Tableau 94. Options de pseudonyme pour les fichiers structurés sous forme de tables (suite)

Nom	Description
FILE_PATH	Indique le chemin qualifié complet vers le fichier structuré sous forme de table. Encadrez le nom de fichier par des guillemets simples. Le fichier de données doit être un fichier standard ou un lien symbolique, plutôt qu'un canal ou un autre type de fichier non standard. Important : Si vous spécifiez l'option FILE_PATH, ne spécifiez pas une colonne DOCUMENT.
KEY_COLUMN	Spécifie le nom de la colonne sur laquelle le fichier est stocké. Une colonne ayant l'option de colonne DOCUMENT ne peut pas être la colonne clé. Seules les clés à colonne unique sont prises en charge. La valeur doit être le nom d'une colonne définie dans l'instruction CREATE NICKNAME. La colonne doit être triée dans l'ordre croissant. La colonne clé doit être désignée comme ne pouvant avoir la valeur NULL en ajoutant l'option NOT NULL à ses définitions dans l'instruction de pseudonyme. Cette valeur fait la distinction majuscules/minuscules.
SORTED	Spécifie si le fichier au niveau de la source de données est ou n'est pas trié dans l'ordre croissant. Les valeurs valides sont Y et N. La valeur par défaut est N. Le fichier au niveau de la source de données n'est pas trié dans l'ordre croissant. Les sources de données triées doivent être triées dans l'ordre croissant conformément à la séquence de classement de l'environnement local en cours, comme défini par les paramètres de la catégorie Prise en charge de la langue nationale LC_COLLATE. Si vous spécifiez que la source de données est triée, définissez l'option VALIDATE_DATA_FILE sur Y.
VALIDATE_DATA_FILE	Pour les fichiers triés, cette option indique si l'encapsuleur vérifie que la colonne de clé est triée dans l'ordre croissant et contrôle la présence de clés ayant la valeur NULL. Cette validation n'a lieu qu'une seule fois lorsque le pseudonyme est créé pour la première fois. La valeur par défaut est N ; l'ordre de tri n'est pas vérifié. Cette option n'est valide que lorsque l'option SORTED est définie sur Y et que l'option DOCUMENT n'est pas spécifiée.

Options de colonne

Tableau 95. Options de colonne pour les fichiers structurés sous forme de tables

Option	Description
DOCUMENT	Vous permet de spécifier le chemin du fichier lors de l'exécution de la requête, plutôt que lorsque vous créez le pseudonyme. La seule valeur valide est FILE. Seule une colonne de chaque pseudonyme peut être spécifiée avec l'option DOCUMENT. La colonne associée à l'option DOCUMENT doit être un type de données VARCHAR ou CHAR. L'utilisation d'une option de colonne de pseudonyme DOCUMENT plutôt qu'une option de pseudonyme FILE_PATH implique que le fichier qui correspond à ce pseudonyme sera fourni lors de l'exécution de la requête. Si l'option DOCUMENT a la valeur FILE, la valeur fournie lors de l'exécution de la requête est le chemin complet du fichier dont le schéma correspond à la définition de pseudonyme pour ce pseudonyme.

Référence des options des services Web

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur, un pseudonyme et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 96. Options d'encapsuleur pour les services Web

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.

Tableau 96. Options d'encapsuleur pour les services Web (suite)

Nom	Description
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE, HTTP et SOCKS. La valeur par défaut est NONE.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le port ou le nom de service du service proxy sur le serveur proxy. Les valeurs valides sont un numéro de port décimal allant de 1 à 32760 ou un nom de service.
SSL_KEYSTORE_FILE	Spécifie le fichier de stockage pour les communications qui utilisent SSL ou TSL. Une valeur valide est un nom de chemin qualifié complet accessible par l'agent de base de données fédérée ou par un processus en mode protégé. La valeur par défaut est <i>install path/cfg/WSWrapperKeystore.kdb</i> .
SSL_KEYSTORE_PASSWORD	Spécifie le mot de passe à utiliser pour accéder au fichier dans l'option SSL_KEYSTORE_FILE. Les valeurs valides sont un mot de passe, qui est chiffré lorsqu'il est stocké dans le catalogue de la base de données fédérée, et <i>file:file_name</i> , où <i>file_name</i> est le chemin qualifié complet vers un fichier de dissimulation.
SSL_VERIFY_SERVER_CERTIFICATE	Spécifie si le certificat du serveur est vérifié lors de l'authentification SSL. Les valeurs valides sont Y et N. La valeur par défaut est N ; le certificat n'est pas vérifié.

Options de serveur

Tableau 97. Options de serveur pour les services Web

Nom	Description
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.

Tableau 97. Options de serveur pour les services Web (suite)

Nom	Description
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le port ou le nom de service du service proxy sur le serveur proxy. Les valeurs valides sont un numéro de port décimal allant de 1 à 32760 ou un nom de service.
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE, HTTP et SOCKS. La valeur par défaut est NONE.
SSL_CLIENT_CERTIFICATE_LABEL	Spécifie le certificat client à envoyer lors de l'authentification SSL. Si vous ne spécifiez pas de valeur, l'ID d'autorisation en cours de la base de données fédérée est utilisé pour déterminer l'emplacement du certificat.
SSL_KEYSTORE_FILE	Spécifie le fichier de stockage pour les communications qui utilisent SSL ou TSL. Une valeur valide est un nom de chemin qualifié complet accessible par l'agent de base de données fédérée ou par un processus en mode protégé. La valeur par défaut est <i>install path/cfg/WSWrapperKeystore.kdb</i> .

Tableau 97. Options de serveur pour les services Web (suite)

Nom	Description
SSL_KEYSTORE_PASSWORD	Spécifie le mot de passe à utiliser pour accéder au fichier dans l'option SSL_KEYSTORE_FILE. Les valeurs valides sont un mot de passe, qui est chiffré lorsqu'il est stocké dans le catalogue de la base de données fédérée, et <code>file:file_name</code> , où <code>file_name</code> est le chemin qualifié complet vers un fichier de dissimulation.
SSL_VERIFY_SERVER_CERTIFICATE	Spécifie si le certificat du serveur est vérifié lors de l'authentification SSL. Les valeurs valides sont Y et N. La valeur par défaut est N ; le certificat n'est pas vérifié.

Options de mappage utilisateur

Tableau 98. Options de mappage utilisateur pour les services Web

Nom	Description
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy. Le mot de passe est chiffré lorsqu'il est stocké dans le catalogue de la base de données fédérée.
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.
SSL_CLIENT_CERTIFICATE_LABEL	Spécifie le certificat client à envoyer lors de l'authentification SSL. Si vous ne spécifiez pas de valeur, l'ID d'autorisation en cours de la base de données fédérée est utilisé pour déterminer l'emplacement du certificat.

Options de pseudonyme

Tableau 99. Options de pseudonyme pour les services Web

Nom	Description
NAMESPACES	<p>Spécifie les espaces de nom associés aux préfixes d'espace de nom utilisés dans les options XPATH et TEMPLATE pour chaque colonne. Utilisez cette syntaxe :</p> <pre>NAMESPACES'prefix1="actual_namespace1", prefix2="actual_namespace2"'</pre> <p>Utilisez une virgule pour séparer plusieurs espaces de nom. Par exemple :</p> <pre>NAMESPACES='http://www.myweb.com/cust", i='http://www.myweb.com/cust/id", n='http://www.myweb.com/cust/name"'</pre>
SOAPACTION	<p>Obligatoire pour le pseudonyme racine. Spécifie l'attribut URI SOAPACTION à partir du format WSDL (Web Services Description Language). L'URL peut contenir une adresse IPv6 séparée par deux-points si elle est encadrée par des crochets. Par exemple :<code>http://[1080:0:0:0:8:800:200C:417A]</code></p> <p>Remarque : Cette option n'est pas valide pour les pseudonymes non racine.</p>
STREAMING	<p>Spécifie si le document source doit être séparé en fragments logiques pour le traitement. Les fragments concordent avec le noeud correspondant à l'expression XPath du pseudonyme. L'encapsuleur procède ensuite à une analyse syntaxique et traite les données source fragment par fragment. Ce type d'analyse syntaxique minimise l'utilisation de la mémoire. Les valeurs valides sont Y et N. La valeur par défaut est N ; les documents ne sont pas soumis à une analyse syntaxique. Cette option n'est valide que sur le pseudonyme racine.</p>
TEMPLATE	<p>Spécifie le fragment de modèle du pseudonyme à utiliser pour établir une requête SOAP. Le fragment doit être conforme à la syntaxe du modèle spécifié. Cette option n'est valide que pour le pseudonyme racine.</p>
URL	<p>Obligatoire pour le pseudonyme racine. Spécifie l'URL du noeud final du service Web. Les protocoles pris en charge sont HTTP et HTTPS. L'URL peut contenir une adresse IPv6 séparée par deux-points si elle est encadrée par des crochets. Par exemple :<code>http://[1080:0:0:0:8:800:200C:417A]</code></p>

Tableau 99. Options de pseudonyme pour les services Web (suite)

Nom	Description
XML_CODESET	Spécifie le codage à utiliser pour envoyer et recevoir des données XML. Cette option écrase le codage interne.
XPATH	Obligatoire. Spécifie l'expression Xpath qui identifie les éléments de réponse SOAP représentant des tuples individuels. Cette expression Xpath est utilisée comme un contexte permettant d'évaluer les valeurs de colonne identifiées par les options de colonne de pseudonyme XPATH.

Options de colonne

Tableau 100. Options de colonne pour les services Web

Nom	Description
ESCAPE_INPUT	Spécifie si les caractères spéciaux XML sont remplacés dans les valeurs d'entrée XML ou non. Utilisez cette option pour inclure des fragments XML comme entrée, par exemple, des fragments XML avec des éléments de répétition. Les valeurs valides sont Y et N. La valeur par défaut est N ; les valeurs d'entrée XML sont conservées. Le type de données de colonne doit être VARCHAR ou CHAR. Si ESCAPE_INPUT est défini sur Y, vous devez également spécifier l'option de colonne TEMPLATE.

Tableau 100. Options de colonne pour les services Web (suite)

Nom	Description
FOREIGN_KEY	<p>Indique que ce pseudonyme est un pseudonyme enfant et spécifie le nom du pseudonyme parent correspondant. Un pseudonyme peut avoir au maximum une option de colonne FOREIGN_KEY. La valeur de l'option est sensible à la casse. Ne spécifiez pas l'option XPATH pour cette colonne. La colonne ne peut être utilisée que pour associer un pseudonyme parent et un pseudonyme enfant. Une instruction CREATE NICKNAME qui comprend une option FOREIGN_KEY échoue si le pseudonyme parent a un nom de schéma différent. A moins que le pseudonyme auquel il est fait référence dans une clause FOREIGN_KEY ait été explicitement défini en minuscule ou dans une casse mixte dans l'instruction CREATE NICKNAME, vous devez spécifier le pseudonyme en majuscule lorsque vous faites référence à ce pseudonyme dans la clause FOREIGN_KEY.</p> <p>Remarque :</p> <ul style="list-style-type: none"> • Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne. • Si vous utilisez cette option de colonne, vous ne pourrez pas utiliser ultérieurement l'instruction ALTER NICKNAME pour supprimer l'option. En effet, vous devrez supprimer le pseudonyme puis le créer à nouveau sans cette option de colonne.
PRIMARY_KEY	<p>Obligatoire pour un pseudonyme parent ayant un ou plusieurs pseudonymes enfant. Spécifie que ce pseudonyme est un pseudonyme parent. Le type de données de colonne doit être VARCHAR(16). Un pseudonyme ne peut avoir qu'une seule option de colonne PRIMARY_KEY. Yes est la seule valeur valide. Ne spécifiez pas l'option XPATH pour cette colonne. La colonne ne peut être utilisée que pour associer des pseudonymes parent et des pseudonymes enfant.</p> <p>Remarque :</p> <ul style="list-style-type: none"> • Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne. • Si vous utilisez cette option de colonne, vous ne pourrez pas utiliser ultérieurement l'instruction ALTER NICKNAME pour supprimer l'option. En effet, vous devrez supprimer le pseudonyme puis le créer à nouveau sans cette option de colonne.

Tableau 100. Options de colonne pour les services Web (suite)

Nom	Description
SOAPACTIONCOLUMN	<p>Indique la colonne qui spécifie de manière dynamique l'action SOAP pour le noeud final des services Web lors de l'exécution d'une requête. Cette option n'est valide que pour le pseudonyme racine. Si le nom d'hôte est une adresse IPv6 (séparée par deux-points), encadrez le nom d'hôte par des crochets. Par exemple : <code>'http://[1080:0:0:0:8:800:200C:417A]:99/soap'</code> Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne.</p>
TEMPLATE	<p>Spécifie le fragment de modèle de colonne à utiliser pour établir un document d'entrée XML. Le fragment doit être conforme à la syntaxe du modèle spécifié.</p> <p>Remarque : Si vous utilisez cette option de colonne, vous ne pourrez pas utiliser ultérieurement l'instruction ALTER NICKNAME pour supprimer l'option. En effet, vous devrez supprimer le pseudonyme puis le créer à nouveau sans cette option de colonne.</p>
URLCOLUMN	<p>Indique la colonne qui spécifie de manière dynamique l'action SOAP pour le noeud final des services Web lors de l'exécution d'une requête. Cette option n'est valide que pour le pseudonyme racine. Si le nom d'hôte est une adresse IPv6 (séparée par deux-points), encadrez le nom d'hôte par des crochets. Par exemple : <code>'http://[1080:0:0:0:8:800:200C:417A]:99/soap'</code> Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne.</p>
XPATH	<p>Spécifie l'expression Xpath du document XML qui contient les données correspondant à cette colonne. L'encapsuleur évalue cette expression XPath après l'application par l'instruction CREATE NICKNAME de l'expression XPath à partir de l'option de pseudonyme XPATH.</p> <p>Remarque : Si vous utilisez cette option de colonne, vous ne pourrez pas utiliser ultérieurement l'instruction ALTER NICKNAME pour supprimer l'option. En effet, vous devrez supprimer le pseudonyme puis le créer à nouveau sans cette option de colonne.</p>

Référence des options XML

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur, un pseudonyme et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 101. Options d'encapsuleur pour XML

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE, HTTP et SOCKS. La valeur par défaut est NONE.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le port ou le nom de service du service proxy sur le serveur proxy. Les valeurs valides sont un numéro de port décimal allant de 1 à 32760 ou un nom de service.
SSL_KEYSTORE_FILE	Spécifie le fichier de stockage du certificat pour les communications qui utilisent SSL ou TSL. Une valeur valide est un nom de chemin qualifié complet accessible par l'agent de base de données fédérée ou par un processus en mode protégé.
SSL_KEYSTORE_PASSWORD	Spécifie le mot de passe à utiliser pour accéder au fichier dans l'option SSL_KEYSTORE_FILE. Les valeurs valides sont un mot de passe, qui est chiffré lorsqu'il est stocké dans le catalogue de la base de données fédérée, et <code>file:file_name</code> , où <code>file_name</code> est le chemin qualifié complet vers un fichier de dissimulation.
SSL_VERIFY_SERVER_CERTIFICATE	Spécifie si le certificat du serveur est vérifié lors de l'authentification SSL. Les valeurs valides sont Y et N. La valeur par défaut est N ; le certificat n'est pas vérifié.

Options de serveur

Tableau 102. Options de serveur pour XML

Nom	Description
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le port ou le nom de service du service proxy sur le serveur proxy. Les valeurs valides sont un numéro de port décimal allant de 1 à 32760 ou un nom de service.
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE, HTTP et SOCKS. La valeur par défaut est NONE.
SOCKET_TIMEOUT	Spécifie la durée maximale, en minutes, au cours de laquelle le serveur fédéré attend les résultats provenant du serveur proxy. Les valeurs valides sont des nombres supérieurs ou égaux à 0. La valeur par défaut est 0 ; le serveur attend pendant une durée illimitée.
SSL_CLIENT_CERTIFICATE_LABEL	Spécifie le certificat client à envoyer lors de l'authentification SSL. Si vous ne spécifiez pas de valeur, l'ID d'autorisation en cours de la base de données fédérée est utilisé pour déterminer l'emplacement du certificat.
SSL_KEYSTORE_FILE	Spécifie le fichier de stockage du certificat pour les communications qui utilisent SSL ou TSL. Une valeur valide est un nom de chemin qualifié complet accessible par l'agent de base de données fédérée ou par un processus en mode protégé.

Tableau 102. Options de serveur pour XML (suite)

Nom	Description
SSL_KEYSTORE_PASSWORD	Spécifie le mot de passe à utiliser pour accéder au fichier dans l'option SSL_KEYSTORE_FILE. Les valeurs valides sont un mot de passe, qui est chiffré lorsqu'il est stocké dans le catalogue de la base de données fédérée, et file:file_name, où file_name est le chemin qualifié complet vers un fichier de dissimulation.
SSL_VERIFY_SERVER_CERTIFICATE	Spécifie si le certificat du serveur est vérifié lors de l'authentification SSL. La valeur par défaut est N ; le certificat n'est pas vérifié.

Options de mappage utilisateur

Tableau 103. Options de mappage utilisateur pour XML

Nom	Description
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy.
SSL_CLIENT_CERTIFICATE_LABEL	Spécifie le certificat client à envoyer lors de l'authentification SSL. Si vous ne spécifiez pas de valeur, l'ID d'autorisation en cours de la base de données fédérée est utilisé pour déterminer l'emplacement du certificat.

Options de pseudonyme

Tableau 104. Options de pseudonyme pour XML

Nom	Description
DIRECTORY_PATH	Spécifie le nom de chemin d'un répertoire contenant un ou plusieurs fichiers XML. Utilisez cette option pour créer un pseudonyme unique sur de multiples fichiers source XML. L'encapsuleur XML n'utilise que des fichiers avec une extension .xml se trouvant dans le répertoire spécifié. L'encapsuleur XML ignore tous les autres fichiers du répertoire. Si vous spécifiez cette option de pseudonyme, ne spécifiez pas une colonne DOCUMENT. Cette option n'est valide que pour le pseudonyme racine.
FILE_PATH	Spécifie le chemin de fichier du document XML. Si vous spécifiez FILE_PATH, ne spécifiez pas une colonne DOCUMENT. Cette option n'est valide que pour le pseudonyme racine.

Tableau 104. Options de pseudonyme pour XML (suite)

Nom	Description
INSTANCE_PARSE_TIME	<p>Spécifie la durée nécessaire, en millisecondes, pour procéder à l'analyse syntaxique d'une ligne du document source XML. La valeur valide peut être un entier ou une valeur décimale. La valeur par défaut est 7. Cette option n'est valide que pour les colonnes du pseudonyme racine. Pour optimiser les requêtes de structures de source XML complexes ou volumineuses, modifiez les options INSTANCE_PARSE_TIME, XPATH_EVAL_TIME et NEXT_TIME.</p>
NAMESPACES	<p>Spécifie les espaces de nom associés aux préfixes d'espace de nom utilisés dans les options XPATH et TEMPLATE pour chaque colonne. Utilisez cette syntaxe :</p> <pre data-bbox="964 743 1450 825">NAMESPACES 'prefix1= "actual_namespace1", prefix2="actual_namespace2"'</pre> <p>Utilisez une virgule pour séparer plusieurs espaces de nom. Par exemple :</p> <pre data-bbox="964 926 1450 1003">NAMESPACES='http://www.myweb.com/cust", i='http://www.myweb.com/cust/id", n='http://www.myweb.com/cust/name"'</pre>
NEXT_TIME	<p>Spécifie la durée nécessaire, en millisecondes, pour déterminer l'emplacement des éléments source consécutifs à partir de l'expression XPath. La valeur par défaut est 1. Cette option est valide pour les pseudonymes racine et non racine. Pour optimiser les requêtes de structures de source XML complexes ou volumineuses, modifiez les options INSTANCE_PARSE_TIME, XPATH_EVAL_TIME et NEXT_TIME.</p>
STREAMING	<p>Spécifie si le document source doit être séparé en fragments logiques pour le traitement. Les fragments concordent avec le noeud correspondant à l'expression XPath du pseudonyme. L'encapsuleur procède ensuite à une analyse syntaxique et traite les données source fragment par fragment. Ce type d'analyse syntaxique minimise l'utilisation de la mémoire. Les valeurs valides sont Y et N. La valeur par défaut est N ; les documents ne sont pas soumis à une analyse syntaxique. Cette option n'est valide que sur le pseudonyme racine.</p>

Tableau 104. Options de pseudonyme pour XML (suite)

Nom	Description
VALIDATE	Spécifie si le document source est validé pour garantir sa conformité au schéma XML ou à une définition de type de document (DTD) avant d'en extraire des données. La valeur par défaut est N ; la validation n'a pas lieu. Avant de définir la valeur sur Y, le fichier schéma ou le fichier DTD se trouve à l'emplacement spécifié par le document source. Cette option n'est valide que pour le pseudonyme racine. Ne définissez pas les deux options STREAMING et VALIDATE sur Y.
XPATH	Obligatoire. Spécifie l'expression Xpath qui identifie les éléments représentant des tuples individuels. Cette expression Xpath est utilisée comme un contexte permettant d'évaluer les valeurs de colonne identifiées par l'option de colonne XPATH.
XPATH_EVAL_TIME	Spécifie la durée nécessaire, en millisecondes, pour évaluer l'expression XPath du pseudonyme et déterminer l'emplacement du premier élément. La valeur peut être un entier ou une valeur décimale. La valeur par défaut est 1. Cette option est valide pour les pseudonymes racine et non racine. Pour optimiser les requêtes de structures de source XML complexes ou volumineuses, modifiez les options INSTANCE_PARSE_TIME, XPATH_EVAL_TIME et NEXT_TIME.

Options de colonne

Tableau 105. Options de colonne pour XML

Nom	Description
DOCUMENT	Indique que cette colonne est une colonne DOCUMENT. La valeur de la colonne DOCUMENT indique le type de données source XML fourni au pseudonyme lors de l'exécution de la requête. Cette option n'est valide que pour les colonnes du pseudonyme racine (le pseudonyme qui identifie les éléments au niveau supérieur du document XML). Seule une colonne pour chaque pseudonyme peut être spécifiée avec l'option DOCUMENT. Si vous utilisez une option de colonne DOCUMENT plutôt qu'une option de pseudonyme FILE_PATH ou DIRECTORY_PATH, le document qui correspond à ce pseudonyme sera fourni lors de l'exécution de la requête. Les valeurs valides sont FILE, DIRECTORY, URI et COLUMN. La valeur par défaut est FILE. La colonne doit être un type de données VARCHAR.

Tableau 105. Options de colonne pour XML (suite)

Nom	Description
FOREIGN_KEY	<p>Indique que ce pseudonyme est un pseudonyme enfant et spécifie le nom du pseudonyme parent correspondant. Un pseudonyme peut avoir au maximum une option de colonne FOREIGN_KEY. La valeur de l'option est sensible à la casse. Ne spécifiez pas l'option XPATH pour cette colonne. La colonne ne peut être utilisée que pour associer un pseudonyme parent et un pseudonyme enfant. Une instruction CREATE NICKNAME qui comprend une option FOREIGN_KEY échoue si le pseudonyme parent a un nom de schéma différent. A moins que le pseudonyme auquel il est fait référence dans une clause FOREIGN_KEY ait été explicitement défini en minuscule ou dans une casse mixte dans l'instruction CREATE NICKNAME, vous devez spécifier le pseudonyme en majuscule lorsque vous faites référence à ce pseudonyme dans la clause FOREIGN_KEY.</p> <p>Remarque :</p> <ul style="list-style-type: none"> • Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne. • Si vous utilisez cette option de colonne, vous ne pourrez pas utiliser ultérieurement l'instruction ALTER NICKNAME pour supprimer l'option. En effet, vous devrez supprimer le pseudonyme puis le créer à nouveau sans cette option de colonne.
PRIMARY_KEY	<p>Obligatoire pour un pseudonyme parent ayant un ou plusieurs pseudonymes enfant. Spécifie que ce pseudonyme est un pseudonyme parent. Le type de données de colonne doit être VARCHAR(16). Un pseudonyme ne peut avoir qu'une seule option de colonne PRIMARY_KEY. Yes est la seule valeur valide. Ne spécifiez pas l'option XPATH pour cette colonne. La colonne ne peut être utilisée que pour associer des pseudonymes parent et des pseudonymes enfant.</p> <p>Remarque :</p> <ul style="list-style-type: none"> • Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne. • Si vous utilisez cette option de colonne, vous ne pourrez pas utiliser ultérieurement l'instruction ALTER NICKNAME pour supprimer l'option. En effet, vous devrez supprimer le pseudonyme puis le créer à nouveau sans cette option de colonne.

Tableau 105. Options de colonne pour XML (suite)

Nom	Description
XPATH	<p>Spécifie l'expression Xpath du document XML qui contient les données correspondant à cette colonne. L'encapsuleur évalue cette expression XPath après l'application par l'instruction CREATE NICKNAME de l'expression XPath à partir de l'option de pseudonyme XPATH.</p> <p>Remarque : Si vous utilisez cette option de colonne, vous ne pourrez pas utiliser ultérieurement l'instruction ALTER NICKNAME pour supprimer l'option. En effet, vous devrez supprimer le pseudonyme puis le créer à nouveau sans cette option de colonne.</p>

Chapitre 5. Vues de la table du catalogue global contenant des informations fédérées

La plupart des vues de catalogue dans une base de données fédérée sont identiques à celles de toute autre base de données DB2 for Linux, UNIX et Windows.

Il existe plusieurs vues uniques contenant des informations importantes pour un système fédéré, telle que la vue SYSCAT.WRAPPERS.

Les vues SYSCAT sont en lecture seule. Vous ne pouvez pas effectuer une opération de mise à jour ou de suppression sur une vue du système SYSCAT. Il est recommandé d'utiliser les vues SYSSTAT pour mettre le catalogue système à jour. Modifiez les applications qui font référence à la vue SYSCAT pour référencer à la place la vue SYSSTAT pouvant être mise à jour.

Le tableau suivant répertorie les vues SYSCAT qui contiennent des informations fédérées. Ces vues sont en lecture seule.

Tableau 106. Vues de catalogue généralement utilisées avec un système fédéré

Vues de catalogue	Description
SYSCAT.CHECKS	Contient les informations de contrainte que vous avez définies.
SYSCAT.COLCHECKS	Contient des colonnes référencées par une contrainte de vérification.
SYSCAT.COLUMNS	Contient des informations de colonne sur les objets de source de données (tables et vues) pour lesquels vous avez créé des pseudonymes.
SYSCAT.COLOPTIONS	Contient des informations sur les valeurs d'option de colonne que vous avez définies pour un pseudonyme.
SYSCAT.CONSTDEP	Contient la dépendance d'une contrainte informationnelle que vous avez définie.
SYSCAT.DATATYPES	Contient des informations de type de données sur les types de données DB2 locaux intégrés et définis par l'utilisateur.
SYSCAT.DBAUTH	Contient les droits de base de données détenus par des utilisateurs individuels et des groupes.
SYSCAT.FUNCMAPOPTIONS	Contient des informations sur les valeurs d'option que vous avez définies pour un mappage de fonctions.
SYSCAT.FUNCMAPPINGS	Contient les mappages de fonctions entre la base de données fédérée et les objets de source de données.
SYSCAT.INDEXCOLUSE	Contient les colonnes qui participent à un index.
SYSCAT.INDEXES	Contient les spécifications d'index pour les objets de source de données.

Tableau 106. Vues de catalogue généralement utilisées avec un système fédéré (suite)

Vues de catalogue	Description
SYSCAT.INDEXOPTIONS	Contient des informations sur les options d'index.
SYSCAT.KEYCOLUSE	Contient les colonnes qui participent à une clé définie par une contrainte de clé unique, de clé primaire ou de clé externe.
SYSCAT.NICKNAMES	Contient des informations sur les pseudonymes que vous avez créés.
SYSCAT.REFERENCES	Contient des informations sur les contraintes référentielles que vous avez définies.
SYSCAT.ROUTINES	Contient des fonctions DB2 locales définies par l'utilisateur ou des modèles de fonction. Les modèles de fonction sont utilisés pour se mapper à une fonction de source de données.
SYSCAT.REVTYPEMAPPINGS	Cette vue n'est pas utilisée. Tous les mappages de types de données sont enregistrés dans la vue SYSCAT.TYPEMAPPINGS.
SYSCAT.ROUTINEOPTIONS	Contient des informations sur les valeurs d'option de routine fédérées.
SYSCAT.ROUTINEPARMOPTIONS	Contient des informations sur les valeurs d'option des paramètres de routine fédérés.
SYSCAT.ROUTINEPARMS	Contient un paramètre ou le résultat d'une routine définie dans SYSCAT.ROUTINES.
SYSCAT.ROUTINESFEDERATED	Contient des informations sur les routines fédérées que vous avez définies.
SYSCAT.SERVERS	Contient les définitions de serveur que vous créez pour les serveurs de source de données.
SYSCAT.TABCONST	Chaque ligne représente une contrainte de table et de pseudonyme de type CHECK, UNIQUE, PRIMARY KEY ou FOREIGN KEY.
SYSCAT.TABLES	Contient des informations sur chaque table, vue fédérée et pseudonyme DB2 local que vous créez.
SYSCAT.TYPEMAPPINGS	Contient des mappages de types de données aval et des mappages de types de données inversés. Le mappage s'effectue vers les types de données DB2 locaux à partir des types de données de la source de données. Ces mappages sont utilisés lorsque vous créez un pseudonyme sur un objet de source de données.
SYSCAT.USEROPTIONS	Contient des informations sur les autorisations des utilisateurs que vous définissez lors de la création de mappages utilisateur entre la base de données fédérée et les serveurs de source de données.
SYSCAT.VIEWS	Contient des informations sur les vues fédérées locales que vous créez.

Tableau 106. Vues de catalogue généralement utilisées avec un système fédéré (suite)

Vues de catalogue	Description
SYSCAT.WRAPOPTIONS	Contient des informations sur les valeurs d'option que vous avez définies pour un encapsuleur.
SYSCAT.WRAPPERS	Contient le nom de l'encapsuleur et du fichier de bibliothèque pour chaque source de données pour laquelle vous créez un encapsuleur.

Le tableau suivant répertorie les vues SYSSTAT qui contiennent des informations fédérées. Ces vues, accessibles en lecture seule, contiennent des statistiques que vous pouvez mettre à jour.

Tableau 107. Vues de catalogue global fédérées pouvant être mises à jour

Vues de catalogue	Description
SYSSTAT.COLUMNS	Contient des informations statistiques sur chaque colonne des objets de source de données (tables et vues) pour lesquels vous avez créé des pseudonymes. Les statistiques ne sont pas enregistrées pour les colonnes héritées de tables basées sur un type structuré.
SYSSTAT.INDEXES	Contient des informations statistiques sur chaque spécification d'index pour des objets de source de données.
SYSSTAT.ROUTINES	Contient des informations statistiques sur chaque fonction définie par l'utilisateur. N'inclut pas de fonctions intégrées. Les statistiques ne sont pas enregistrées pour les colonnes héritées de tables basées sur un type structuré.
SYSSTAT.TABLES	Contient des informations sur chaque table de base. Les informations sur les vues, les synonymes et les alias ne sont pas incluses dans cette vue. Pour les tables basées sur un type structuré, seule la table racine d'une hiérarchie de tables est incluse dans la vue. Les statistiques ne sont pas enregistrées pour les colonnes héritées de tables basées sur un type structuré.

Chapitre 6. Options de mappage de fonctions pour les systèmes fédérés

Le serveur fédéré fournit des mappages par défaut entre les fonctions de DB2 et les fonctions de source de données. Pour la plupart des sources de données, les mappages de fonctions par défaut figurent dans les encapsuleurs. Pour utiliser une fonction de source de données non reconnue par le serveur fédéré ou pour modifier le mappage par défaut, vous créez un mappage de fonctions.

Lorsque vous créez un mappage de fonctions, vous spécifiez le nom de la fonction de source de données et devez activer la fonction mappée. Ensuite, lorsque vous utilisez la fonction mappée, l'optimiseur de requêtes compare le coût d'exécution de la fonction au niveau de la source de données au coût d'exécution de la fonction au niveau du serveur fédéré.

Tableau 108. Options pour les mappages de fonctions

Nom	Description
DISABLE	Permet d'activer ou de désactiver un mappage de fonctions par défaut. Les valeurs valides sont Y et N. La valeur par défaut est N.
REMOTE_NAME	Nom de la fonction de source de données. La valeur par défaut est le nom local.

Chapitre 7. Types de serveur valides dans les instructions SQL

Les types de serveur indiquent le type de source de données représenté par la définition de serveur.

Les types de serveur varient en fonction du fournisseur, de la finalité et du système d'exploitation. Les valeurs prises en charge dépendent de la source de données.

Pour la plupart des sources de données, vous devez spécifier un type de serveur valide dans l'instruction CREATE SERVER.

Tableau 109. Sources de données et types de serveur

Source de données	Type de serveur
BioRS	Un type de serveur n'est pas requis dans l'instruction CREATE SERVER.
Excel	Un type de serveur n'est pas requis dans l'instruction CREATE SERVER.
IBM DB2 Universal Database for Linux, UNIX et Windows	DB2/UDB
IBM DB2 Universal Database for System i et AS/400	DB2/ISERIES
IBM DB2 Universal Database for z/OS	DB2/ZOS
IBM DB2 for VM	DB2/VM
Informix	INFORMIX
JDBC	JDBC (Obligatoire pour les sources de données JDBC prises en charge par les pilotes JDBC version 3.0 et ultérieure.)
Microsoft SQL Server	MSSQLSERVER (Obligatoire pour les sources de données prises en charge par le pilote DataDirect Connect ODBC 4.2 (ou version ultérieure) ou le pilote Microsoft SQL Server ODBC 3.0 (ou version ultérieure).)
ODBC	ODBC (Obligatoire pour les sources de données ODBC qui sont prises en charge par le pilote ODBC 3.x.)
OLE DB	Un type de serveur n'est pas requis dans l'instruction CREATE SERVER.
Oracle	ORACLE (Obligatoire pour les sources de données Oracle prises en charge par le logiciel client NET8.)
Sybase (CTLIB)	SYBASE
Fichiers structurés sous forme de table	Un type de serveur n'est pas requis dans l'instruction CREATE SERVER.
Teradata	TERADATA

Tableau 109. Sources de données et types de serveur (suite)

Source de données	Type de serveur
Services Web	Un type de serveur n'est pas requis dans l'instruction CREATE SERVER.
XML	Un type de serveur n'est pas requis dans l'instruction CREATE SERVER.

Chapitre 8. Mappages de type de données

Les mappages de types de données pour les sources de données non relationnelles incluent les mappages de types aval, les mappages de types inversés et les mappages de types propres à Unicode. Chaque source de données non relationnelle prend en charge des types de données spécifiques.

Mappages de types de données aval par défaut

Les mappages de type aval et inversé sont les deux types de mappage entre les types de données de sources de données et les types de données de bases de données fédérées. Un mappage de type aval est un mappage depuis un type de données distant vers un type de données local équivalent.

Vous pouvez remplacer un mappage de type de données par défaut, ou en créer un avec l'instruction CREATE TYPE MAPPING.

Ces mappages sont valides avec toutes les versions prises en charge, sauf mention contraire.

Pour tous les mappages de types de données aval par défaut d'une source de données vers une base de données fédérée, le schéma fédéré est SYSIBM.

Les tableaux suivants indiquent des mappages aval par défaut entre des types de données de bases de données fédérées et des types de données de sources de données.

Mappages de types de données aval par défaut pour les sources de données d'une base de données DB2 pour Linux, UNIX et Windows

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données d'une base de données DB2 pour Linux, UNIX et Windows.

Tableau 110. Mappages de types de données aval par défaut pour une base de données DB2 pour Linux, UNIX, et Windows (les colonnes ne sont pas toutes représentées)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BIGINT	-	-	-	-	-	-	BIGINT	-	0	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	-	-	-	-	-	-	CHAR	-	0	N
CHAR	-	-	-	-	Y	-	CHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DECFLOAT ²	-	-	-	-	-	-	DECFLOAT	-	0	-

Tableau 110. Mappages de types de données aval par défaut pour une base de données DB2 pour Linux, UNIX, et Windows (les colonnes ne sont pas toutes représentées) (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
DOUBLE	-	-	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
LONGVAR	-	-	-	-	N	-	CLOB	-	-	-
LONGVAR	-	-	-	-	Y	-	BLOB	-	-	-
LONGVARG	-	-	-	-	-	-	DBCLOB	-	-	-
REAL	-	-	-	-	-	-	REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP(p)	-	p	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	0	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	0	Y
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	0	N

Remarque :

1. Le type fédéré est TIMESTAMP(0) si le paramètre de configuration date_compat est défini sur ON.
2. L'option de serveur SAME_DECFLT_ROUNDING est définie par défaut sur N et les opérations ne seront pas transférées sur la source de données distante à moins que l'option ne soit définie sur Y. Pour plus d'informations sur l'option de serveur SAME_DECFLT_ROUNDING, voir Référence des options de base de données DB2.

Mappages de types de données aval par défaut pour les sources de données DB2 for System i

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données DB2 for System i.

Tableau 111. Mappages de types de données aval par défaut pour DB2 for System i (les colonnes ne sont pas toutes représentées)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	255	32672	-	-	-	-	VARCHAR	-	0	N
CHAR	1	254	-	-	Y	-	CHAR	-	0	Y
CHAR	255	32672	-	-	Y	-	VARCHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N

Tableau 111. Mappages de types de données aval par défaut pour DB2 for System i (les colonnes ne sont pas toutes représentées) (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
GRAPHIC	128	16336	-	-	-	-	VARGRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
NUMERIC	-	-	-	-	-	-	DECIMAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	-	6	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	Y	-	VARCHAR	-	0	Y
VARG	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

Mappages de types de données aval par défaut pour les sources de données DB2 pour VM et VSE

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données DB2 pour VM et VSE.

Tableau 112. Mappages de types de données aval par défaut pour DB2 Server pour VM et VSE (les colonnes ne sont pas toutes représentées)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	1	254	-	-	Y	-	CHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBAHW	-	-	-	-	-	-	SMALLINT	-	0	-
DBAINT	-	-	-	-	-	-	INTEGER	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	-	6	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	Y	-	VARCHAR	-	0	Y
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPH	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

Mappages de types de données aval par défaut pour les sources de données DB2 for z/OS

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données DB2 for z/OS.

Tableau 113. Mappages de types de données aval par défaut de DB2 for z/OS (les colonnes ne sont pas toutes représentées)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires de données distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	255	32672	-	-	-	-	VARCHAR	-	0	N
CHAR	1	254	-	-	Y	-	CHAR	-	0	Y
CHAR	255	32672	-	-	Y	-	VARCHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
ROWID	-	-	-	-	Y	-	VARCHAR	40	-	Y
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	-	6	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	Y	-	VARCHAR	-	0	Y
VARG	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

Mappages de types de données aval par défaut pour les sources de données Informix

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données Informix.

Tableau 114. Mappages de types de données aval par défaut pour Informix (les colonnes ne sont pas toutes représentées)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires de données distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BLOB	-	-	-	-	-	-	BLOB	2147483647	-	-
BOOLEAN	-	-	-	-	-	-	CHARACTER	1	-	-
BYTE	-	-	-	-	-	-	BLOB	2147483647	-	-
CHAR	1	254	-	-	-	-	CHARACTER	-	-	-
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CLOB	-	-	-	-	-	-	CLOB	2147483647	-	-

Tableau 114. Mappages de types de données aval par défaut pour Informix (les colonnes ne sont pas toutes représentées) (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
DATE	-	-	-	-	-	-	DATE	4	-	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	0	-
DATETIME ²	0	4	0	4	-	-	DATE	4	-	-
DATETIME	6	10	6	10	-	-	TIME	3	-	-
DATETIME	0	4	6	15	-	-	TIMESTAMP(6)	10	6	-
DATETIME	6	10	11	15	-	-	TIMESTAMP(6)	10	6	-
DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
DECIMAL	32	130	-	-	-	-	DOUBLE	8	-	-
DECIMAL	1	32	255	255	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	8	-	-
INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
INTERVAL	-	-	-	-	-	-	VARCHAR	25	-	-
INT8	-	-	-	-	-	-	BIGINT	19	0	-
LVARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
MONEY	1	31	0	31	-	-	DECIMAL	-	-	-
MONEY	32	32	-	-	-	-	DOUBLE	8	-	-
NCHAR	1	254	-	-	-	-	CHARACTER	-	-	-
NCHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
NVARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
REAL	-	-	-	-	-	-	REAL	4	-	-
SERIAL	-	-	-	-	-	-	INTEGER	4	-	-
SERIAL8	-	-	-	-	-	-	BIGINT	-	-	-
SMALLFLOAT	-	-	-	-	-	-	REAL	4	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
TEXT	-	-	-	-	-	-	CLOB	2147483647	-	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-

Remarques :

1. Le type fédéré est TIMESTAMP(0) si le paramètre de configuration date_compat est défini sur ON.
2. Pour le type de données Informix DATETIME, le serveur fédéré de DB2 UNIX et Windows utilise le qualificatif de haut niveau Informix pour REMOTE_LENGTH et le qualificatif de bas niveau Informix pour REMOTE_SCALE.

Les qualificatifs Informix sont les constantes "TU_" définies dans le fichier datatime.h d'Informix Client SDK. Les constantes sont les suivantes :

0 = YEAR	8 = MINUTE	13 = FRACTION(3)
2 = MONTH	10 = SECOND	14 = FRACTION(4)
4 = DAY	11 = FRACTION(1)	15 = FRACTION(5)
6 = HOUR	12 = FRACTION(2)	

Mappages de types de données aval par défaut pour les sources de données JDBC

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données JDBC.

Tableau 115. Mappages de types de données aval par défaut pour JDBC

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires de données distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BIGINT	-	-	-	-	-	-	BIGINT	8	-	-
BINARY	-	254	-	-	-	-	CHAR	-	-	Y
BINARY	255	32672	-	-	-	-	VARCHAR	-	-	Y
BINARY	32673	2147483647	-	-	-	-	BLOB	2147483647	-	-
BIT	-	-	-	-	-	-	SMALLINT	2	-	-
BLOB	-	-	-	-	-	-	BLOB	2147483647	-	-
BOOLEAN-	-	-	-	-	-	-	SMALLINT	2	-	-
CHAR	-	254	-	-	-	-	CHAR	-	-	-
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CHAR	32673	2147483647	-	-	-	-	CLOB	2147483647	-	-
CLOB	-	-	-	-	-	-	CLOB	2147483647	-	-
DATE	-	-	-	-	-	-	DATE	-	-	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-
DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
DECIMAL	32	38	0	38	-	-	DOUBLE	8	-	-
DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-
FLOAT	-	-	-	-	-	-	FLOAT	4	-	-
INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
LONGVARCHAR	-	32672	-	-	-	-	VARCHAR	-	-	-
LONGVARCHAR	32673	2147483647	-	-	-	-	CLOB	2147483647	-	-
LONGVARBINARY	-	32672	-	-	-	-	VARCHAR	-	-	Y
LONGVARBINARY	32673	2147483647	-	-	-	-	BLOB	2147483647	-	-
LONGNVARCHAR	-	16336	-	-	-	-	VARGRAPHIC	-	-	-
LONGNVARCHAR ²	16337	1073741823	-	-	-	-	DBCLOB	-	-	-
NCHAR ²	-	127	-	-	-	-	GRAPHIC	-	-	-
NCHAR ²	128	16336	-	-	-	-	VARGRAPHIC	-	-	-
NCHAR ²	16337	1073741823	-	-	-	-	DBCLOB	-	-	-
NCLOB ²	-	-	-	-	-	-	DBCLOB	-	-	-
NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-
NUMERIC	32	38	0	38	-	-	DOUBLE	8	-	-
NVARCHAR ²	-	16336	-	-	-	-	VARGRAPHIC	-	-	-
NVARCHAR ²	16337	1073741823	-	-	-	-	DBCLOB	-	-	-
REAL							REAL	4		
SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
TIME	-	-	-	-	-	-	TIME	3	-	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
TIMESTAMP(p)	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-
VARBINARY	-	32672	-	-	-	-	VARCHAR	-	-	Y
VARBINARY	32673	2147483647	-	-	-	-	BLOB	2147483647	-	-
VARCHAR	-	32672	-	-	-	-	VARCHAR	-	-	-
VARCHAR	32673	2147483647	-	-	-	-	CLOB	2147483647	-	-

Tableau 115. Mappages de types de données aval par défaut pour JDBC (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires de données distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
---------------------	------------------------------	------------------------------	-----------------------------	-----------------------------	---------------------------------------	---------------------------------	--------------------	------------------	-----------------	---------------------------

Remarque :

1. Le type fédéré est TIMESTAMP(0) si le paramètre de configuration date_compat est défini sur ON.
2. Les types de données suivants sont uniquement pris en charge par le pilote de périphérique JDBC 4.0 : NCHAR, NVARCHAR, LONGVARCHAR et NCLOB.

Les types de données suivants ne sont pas pris en charge par l'encapsuleur JDBC : DATALINK, OTHER, JAVA_OBJECT, DISTINCT, STRUCT, ARRAY et REF.

Mappages de types de données aval par défaut pour les sources de données Microsoft SQL Server

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données Microsoft SQL Server.

Tableau 116. Mappages de types de données aval par défaut pour Microsoft SQL Server

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires de données distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
bigint ¹	-	-	-	-	-	-	BIGINT	-	-	-
binary	1	254	-	-	-	-	CHARACTER	-	-	Y
binary	255	8000	-	-	-	-	VARCHAR	-	-	Y
bit	-	-	-	-	-	-	SMALLINT	2	-	-
char	1	254	-	-	-	-	CHAR	-	-	N
char	255	8000	-	-	-	-	VARCHAR	-	-	N
datetime	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
decimal	1	31	0	31	-	-	DECIMAL	-	-	-
decimal	32	38	0	38	-	-	DOUBLE	-	-	-
float	-	8	-	-	-	-	DOUBLE	8	-	-
float	-	4	-	-	-	-	REAL	4	-	-
image	-	-	-	-	-	-	BLOB	2147483647	-	Y
int	-	-	-	-	-	-	INTEGER	4	-	-
money	-	-	-	-	-	-	DECIMAL	19	4	-
nchar	1	127	-	-	-	-	CHAR	-	-	N
nchar	128	4000	-	-	-	-	VARCHAR	-	-	N
numeric	1	31	0	31	-	-	DECIMAL	-	-	-
numeric	32	38	0	38	-	-	DOUBLE	8	-	-
ntext	-	-	-	-	-	-	CLOB	2147483647	-	Y
nvarchar	1	4000	-	-	-	-	VARCHAR	-	-	N
real	-	-	-	-	-	-	REAL	4	-	-
smallint	-	-	-	-	-	-	SMALLINT	2	-	-
smalldatetime	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
smallmoney	-	-	-	-	-	-	DECIMAL	10	4	-
SQL_BIGINT	-	-	-	-	-	-	BIGINT	-	-	-
SQL_BINARY	1	254	-	-	-	-	CHARACTER	-	-	Y
SQL_BINARY	255	8000	-	-	-	-	VARCHAR	-	-	Y

Tableau 116. Mappages de types de données aval par défaut pour Microsoft SQL Server (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires de données distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
SQL_BIT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_CHAR	1	254	-	-	-	-	CHAR	-	-	N
SQL_CHAR	255	8000	-	-	-	-	VARCHAR	-	-	N
SQL_DATE	-	-	-	-	-	-	DATE	4	-	-
SQL_DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_DECIMAL	32	38	0	38	-	-	DOUBLE	8	-	-
SQL_DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_FLOAT	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_GUID	-	-	-	-	-	-	VARCHAR	-	-	Y
SQL_INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
SQL_LONGVARCHAR	-	-	-	-	-	-	CLOB	2147483647	-	N
SQL_LONGVARBINARY	-	-	-	-	-	-	BLOB	-	-	Y
SQL_NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_NUMERIC	32	38	0	38	-	-	DOUBLE	8	-	-
SQL_REAL	-	-	-	-	-	-	REAL	8	-	-
SQL_SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_TIME	-	-	-	-	-	-	TIME	3	-	-
SQL_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	10	6	-
SQL_TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_VARBINARY	1	8000	-	-	-	-	VARCHAR	-	-	Y
SQL_VARCHAR	1	8000	-	-	-	-	VARCHAR	-	-	N
SQL_WCHAR	1	254	-	-	-	-	CHARACTER	-	-	N
SQL_WCHAR	255	8800	-	-	-	-	VARCHAR	-	-	N
SQL_WLONGVARCHAR	-	1073741823	-	-	-	-	CLOB	2147483647	-	N
SQL_WVARCHAR	1	16336	-	-	-	-	VARCHAR	-	-	N
text	-	-	-	-	-	-	CLOB	-	-	N
timestamp	-	-	-	-	-	-	VARCHAR	8	-	Y
tinyint	-	-	-	-	-	-	SMALLINT	2	-	-
uniqueidentifier	1	4000	-	-	Y	-	VARCHAR	16	-	Y
varbinary	1	8000	-	-	-	-	VARCHAR	-	-	Y
varchar	1	8000	-	-	-	-	VARCHAR	-	-	N

Remarque :

1. Ce mappage est uniquement valide avec Microsoft SQL Server 2000.

Mappages de types de données aval par défaut pour les sources de données ODBC

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données ODBC.

Tableau 117. Mappages de types de données aval par défaut pour ODBC (les colonnes ne sont pas toutes représentées)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires de données distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
SQL_BIGINT	-	-	-	-	-	-	BIGINT	8	-	-
SQL_BINARY	1	254	-	-	-	-	CHARACTER	-	-	Y
SQL_BINARY	255	32672	-	-	-	-	VARCHAR	-	-	Y
SQL_BIT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_CHAR	1	254	-	-	-	-	CHAR	-	-	N
SQL_CHAR	255	32672	-	-	-	-	VARCHAR	-	-	N
SQL_DATE	-	-	-	-	-	-	DATE	-	-	-
SQL_DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-
SQL_DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_DECIMAL	32	38	0	38	-	-	DOUBLE	8	-	-
SQL_DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_FLOAT	-	8	-	-	-	-	FLOAT	8	-	-
SQL_FLOAT	-	4	-	-	-	-	FLOAT	4	-	-
SQL_INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
SQL_LONGVARCHAR	-	-	-	-	-	-	CLOB	2147483647	-	N
SQL_LONGVARBINARY	-	-	-	-	-	-	BLOB	2147483647	-	Y
SQL_NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_NUMERIC	32	32	0	31	-	-	DOUBLE	8	-	-
SQL_REAL	-	-	-	-	-	-	REAL	4	-	-
SQL_SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
SQL_TIMESTAMP(p)	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
SQL_TYPE_DATE	-	-	-	-	-	-	DATE	4	-	-
SQL_TYPE_TIME	-	-	-	-	-	-	TIME	3	-	-
SQL_TYPE_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	10	-	-
SQL_TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_VARBINARY	1	32672	-	-	-	-	VARCHAR	-	-	Y
SQL_VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	N
SQL_WCHAR	1	127	-	-	-	-	CHAR	-	-	N
SQL_WCHAR	128	16336	-	-	-	-	VARCHAR	-	-	N
SQL_WVARCHAR	1	16336	-	-	-	-	VARCHAR	-	-	N
SQL_WLONGVARCHAR	-	1073741823	-	-	-	-	CLOB	2147483647	-	N

Remarque :

1. Le type fédéré est TIMESTAMP(0) si le paramètre de configuration date_compat est défini sur ON.

Mappages de types de données aval par défaut pour les sources de données Oracle NET8

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données Oracle NET8.

Tableau 118. mappages de types de données aval par défaut Oracle NET8

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BLOB	0	0	0	0	-	\0	BLOB	2147483647	0	Y
CHAR	1	254	0	0	-	\0	CHAR	0	0	N
CHAR	255	2000	0	0	-	\0	VARCHAR	0	0	N
CLOB	0	0	0	0	-	\0	CLOB	2147483647	0	N
DATE	0	0	0	0	-	\0	TIMESTAMP(6)	0	0	N
FLOAT	1	126	0	0	-	\0	DOUBLE	0	0	N
LONG	0	0	0	0	-	\0	CLOB	2147483647	0	N
LONG RAW	0	0	0	0	-	\0	BLOB	2147483647	0	Y
NUMBER	10	18	0	0	-	\0	BIGINT	0	0	N
NUMBER	1	38	-84	127	-	\0	DOUBLE	0	0	N
NUMBER	1	31	0	31	-	>=	DECIMAL	0	0	N
NUMBER	1	4	0	0	-	\0	SMALLINT	0	0	N
NUMBER	5	9	0	0	-	\0	INTEGER	0	0	N
NUMBER	-	10	0	0	-	\0	DECIMAL	0	0	N
RAW	1	2000	0	0	-	\0	VARCHAR	0	0	Y
ROWID	0	0	0	NULL	-	\0	CHAR	18	0	N
TIMESTAMP(p) ¹	-	-	-	-	-	\0	TIMESTAMP(6)	10	6	N
VARCHAR2	1	4000	0	0	-	\0	VARCHAR	0	0	N

Remarque :

1.

- TIMESTAMP(p) représente un horodatage avec une échelle de variable comprise entre 0 et 9. L'échelle de l'horodatage Oracle est mappé vers TIMESTAMP(6) par défaut. Vous pouvez modifier ce mappage de type par défaut et mapper l'horodatage Oracle TIMESTAMP vers un horodatage TIMESTAMP fédéré de la même échelle en utilisant un mappage de type défini par l'utilisateur.
- Ce mappage est uniquement valide pour les configurations client et de serveurs Oracle 9i (ou ultérieur).

Mappages de types de données aval par défaut pour les sources de données Sybase

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données Sybase.

Tableau 119. Mappages de types de données aval par défaut pour Sybase CTLIB

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
binary	1	254	-	-	-	-	CHAR	-	-	Y
binary	255	32672	-	-	-	-	VARCHAR	-	-	Y
bit	-	-	-	-	-	-	SMALLINT	-	-	-
char	1	254	-	-	-	-	CHAR	-	-	N
char	255	32672	-	-	-	-	VARCHAR	-	-	N
char null (voir varchar)										
date	-	-	-	-	-	-	DATE	-	-	-
date	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-

Tableau 119. Mappages de types de données aval par défaut pour Sybase CTLIB (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
datetime	-	-	-	-	-	-	TIMESTAMP(6)	-	-	-
datetimn	-	-	-	-	-	-	TIMESTAMP	-	-	-
decimal	1	31	0	31	-	-	DECIMAL	-	-	-
decimal	32	38	0	38	-	-	DOUBLE	-	-	-
decimaln	1	31	0	31	-	-	DECIMAL	-	-	-
decimaln	32	38	0	38	-	-	DOUBLE	-	-	-
float	-	4	-	-	-	-	REAL	-	-	-
float	-	8	-	-	-	-	DOUBLE	-	-	-
floatn	-	4	-	-	-	-	REAL	-	-	-
floatn	-	8	-	-	-	-	DOUBLE	-	-	-
image	-	-	-	-	-	-	BLOB	-	-	-
int	-	-	-	-	-	-	INTEGER	-	-	-
intn	-	-	-	-	-	-	INTEGER	-	-	-
money	-	-	-	-	-	-	DECIMAL	19	4	-
moneyn	-	-	-	-	-	-	DECIMAL	19	4	-
nchar	1	254	-	-	-	-	CHAR	-	-	N
nchar	255	32672	-	-	-	-	VARCHAR	-	-	N
nchar null (voir nvarchar)										
numeric	1	31	0	31	-	-	DECIMAL	-	-	-
numeric	32	38	0	38	-	-	DOUBLE	-	-	-
numericn	1	31	0	31	-	-	DECIMAL	-	-	-
numericn	32	38	0	38	-	-	DOUBLE	-	-	-
nvarchar	1	32672	-	-	-	-	VARCHAR	-	-	N
real	-	-	-	-	-	-	REAL	-	-	-
smalldatetime	-	-	-	-	-	-	TIMESTAMP(6)	-	-	-
smallint	-	-	-	-	-	-	SMALLINT	-	-	-
smallmoney	-	-	-	-	-	-	DECIMAL	10	4	-
sysname	-	-	-	-	-	-	VARCHAR	30	-	N
text	-	-	-	-	-	-	CLOB	-	-	-
time	-	-	-	-	-	-	TIME	-	-	-
timestamp	-	-	-	-	-	-	VARCHAR	8	-	Y
tinyint	-	-	-	-	-	-	SMALLINT	-	-	-
unichar ²	1	254	-	-	-	-	CHAR	-	-	N
unichar ²	255	32672	-	-	-	-	VARCHAR	-	-	N
unichar null (voir univarchar)										
univarchar ²	1	32672	-	-	-	-	VARCHAR	-	-	N
varbinary	1	32672	-	-	-	-	VARCHAR	-	-	Y
varchar	1	32672	-	-	-	-	VARCHAR	-	-	N

Tableau 119. Mappages de types de données aval par défaut pour Sybase CTLIB (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
---------------------	------------------------------	------------------------------	-----------------------------	-----------------------------	----------------------------	---------------------------------	--------------------	------------------	-----------------	---------------------------

Remarque :

1. Le type fédéré est TIMESTAMP(0) si le paramètre de configuration date_compat est défini sur ON.
2. Valide pour les bases de données fédérées non Unicode.

Mappages de types de données aval par défaut pour les sources de données Teradata

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données Teradata.

Tableau 120. Mappages de types de données aval par défaut pour Teradata (les colonnes ne sont pas toutes représentées)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BLOB	1	2097088000	-	-	-	-	BLOB	-	-	-
BYTE	1	254	-	-	-	-	CHAR	-	-	Y
BYTE	255	32672	-	-	-	-	VARCHAR	-	-	Y
BYTE	32673	64000	-	-	-	-	BLOB	-	-	-
BYTEINT	-	-	-	-	-	-	SMALLINT	-	-	-
CHAR	1	254	-	-	-	-	CHARACTER	-	-	-
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CHAR	32673	64000	-	-	-	-	CLOB	-	-	-
CLOB	1	2097088000 (Latin)	-	-	-	-	CLOB	-	-	-
CLOB	1	1048544000 (Unicode)	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	-	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-
DECIMAL	1	18	0	18	-	-	DECIMAL	-	-	-
DOUBLE PRECISION	-	-	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	-	-
GRAPHIC	128	16336	-	-	-	-	VARGRAPHIC	-	-	-
GRAPHIC	16337	32000	-	-	-	-	DBCLOB	-	-	-
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
INTERVAL	-	-	-	-	-	-	CHAR	-	-	-
NUMERIC	1	18	0	18	-	-	DECIMAL	-	-	-
REAL	-	-	-	-	-	-	DOUBLE	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	0	21	0	21	-	-	TIME	-	-	-
TIMESTAMP(<i>p</i>)	-	-	<i>p</i>	<i>p</i>	-	-	TIMESTAMP(6)	10	6	-
VARBYTE	1	32762	-	-	-	-	VARCHAR	-	-	Y
VARBYTE	32763	64000	-	-	-	-	BLOB	-	-	-

Tableau 120. Mappages de types de données aval par défaut pour Teradata (les colonnes ne sont pas toutes représentées) (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
VARCHAR	32673	64000	-	-	-	-	CLOB	-	-	-
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	-	-
VARGRAPHIC	16337	32000	-	-	-	-	DBCLOB	-	-	-

Remarque :

1. Le type fédéré est TIMESTAMP(0) si le paramètre de configuration date_compat est défini sur ON.

Echantillons de mappages de types de données aval

Vous pouvez utiliser les exemples de mappages de types aval pour exploiter la prise en charge du type de données TIMESTAMP.

Pour les sources de données Informix, ces mappages sont utilisés pour les colonnes de pseudonymes, les paramètres des procédures fédérées, les passes-systèmes et les ensembles de résultats des procédures fédérées.

Pour les autres sources de données, ces mappages affectent uniquement les colonnes de pseudonymes et les paramètres des procédures fédérées. Les passe-systèmes et les ensembles de résultats des procédures fédérées ne sont pas concernés.

Mappages de types de données aval - exemple pour Informix

Lorsque vous créez des objets fédérés, vous pouvez utiliser l'exemple de mappage de types aval fourni pour Informix.

Vous devez créer ces mappages avant les objets fédérés.

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,10);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(1)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,11);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(2)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,12);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,13);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(4)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,14);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(5)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,15);
```

Mappages de types de données aval - exemple pour Microsoft SQL Server

Lorsque vous créez des objets fédérés, vous pouvez utiliser l'exemple de mappage de types aval fourni pour Microsoft SQL Server.

Vous devez créer ces mappages avant un pseudonyme ou une procédure fédérée.

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE mssqlserver REMOTE TYPE "datetime";
```

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE mssqlserver REMOTE TYPE "smalldatetime";
```

Mappages de types de données aval - exemple pour Oracle

Lorsque vous créez des objets fédérés, vous pouvez utiliser l'exemple de mappage de types aval fourni pour Oracle.

Vous devez créer ces mappages avant un pseudonyme ou une procédure fédérée.

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(0);
```

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(1)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(1);
```

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(2)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(2);
```

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(3);
```

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(4)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(4);
```

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(5)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(5);
```

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(7)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(7);
```

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(8)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(8);
```

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(9)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(9);
```

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE oracle REMOTE TYPE date;
```

Mappages de types de données aval - exemple pour Sybase

Lorsque vous créez des objets fédérés, vous pouvez utiliser l'exemple de mappage de types aval fourni pour Sybase.

Vous devez créer ces mappages avant un pseudonyme ou une procédure fédérée.

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE sybase REMOTE TYPE datetime);
```

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE sybase REMOTE TYPE smalldatetime);
```

Mappages de types de données aval - exemple pour Teradata

Lorsque vous créez des objets fédérés, vous pouvez utiliser l'exemple de mappage de types aval fourni pour Teradata.

Vous devez créer ces mappages avant un pseudonyme ou une procédure fédérée.

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(0);
```

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(1)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(1);
```



```

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(2)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(2);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(3);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(4)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(4);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(5)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(5);

```

Mappages de types de données inversés par défaut

Pour la plupart des sources de données, les mappages de type par défaut figurent dans les encapsuleurs.

Les mappages de type aval et inversé sont les deux types de mappage entre les types de données de sources de données et les types de données de bases de données fédérées. Un mappage de type aval est un mappage depuis un type de données distant vers un type de données local équivalent. Un mappage de type inversé est un mappage utilisé avec le langage DDL transparent pour créer ou modifier des tables distantes.

Les mappages de type par défaut pour les sources de données DB2 se trouvent dans l'encapsuleur DRDA. Les mappages de type par défaut pour Informix se trouvent dans l'encapsuleur INFORMIX, et ainsi de suite.

Lorsque vous définissez une table ou vue distante d'une base de données fédérée, la définition inclut un mappage de type inversé. Le mappage s'effectue depuis un type de données de base de données fédérée locale pour chacune des colonnes, et le type de données distantes correspondantes. Par exemple, il y a un mappage de type inversé par défaut où le type de données local REAL pointe sur le type SMALLFLOAT d'Informix.

Les bases de données fédérées ne prennent pas en charge le mappage pour les types LONG VARCHAR, LONG VARGRAPHIC et les types définis par l'utilisateur.

Lorsque vous créez une table distante avec l'instruction CREATE TABLE, vous devez indiquer les types de données locales que vous voulez inclure dans la table distante. Le mappage de type inversé par défaut va attribuer à ces colonnes les types distants correspondants. Supposons que vous utilisiez l'instruction CREATE TABLE pour définir une table Informix avec une colonne C2. Vous indiquez BIGINT comme type de données pour C2 dans l'instruction. Le mappage de type inversé par défaut du type BIGINT dépend de la version d'Informix sur laquelle vous créez votre table. Le mappage de C2 dans la table Informix sera effectué vers le type DECIMAL si vous utilisez Informix version 8 et vers le type INT8 si vous utilisez Informix version 9.

Vous pouvez remplacer un mappage de type de données inversé par défaut, ou en créer un avec l'instruction CREATE TYPE MAPPING.

Les tableaux suivants indiquent des mappages inversés par défaut entre des types de données locales de bases de données fédérées et des types de données de sources de données distantes.

Ces mappages sont valides avec toutes les versions prises en charge, sauf mention contraire.

Mappages de types de données inversés par défaut pour les sources de données d'une base de données DB2 pour Linux, UNIX et Windows

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données d'une base de données DB2 pour Linux, UNIX et Windows.

Tableau 121. Mappages de types de données inversés par défaut pour une base de données DB2 pour Linux, UNIX, et Windows (les colonnes ne sont pas toutes représentées)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires fédérées
BIGINT	-	8	-	-	-	-	BIGINT	-	-	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	Y	-	CHAR	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE ¹	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DECFLOAT ²	-	8	-	-	-	-	DECFLOAT	-	0	-
DECFLOAT ²	-	16	-	-	-	-	DECFLOAT	-	0	-
DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	-	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP(<i>p</i>)	-	-	<i>p</i>	<i>p</i>	-	-	TIMESTAMP(<i>p</i>)	-	<i>p</i> ³	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	-	N
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	-

Remarque :

1. Lorsque le paramètre `date_compat` est défini sur OFF, le type fédéré DATE est mappé vers TIMESTAMP(0).
2. L'option de serveur `SAME_DECFLT_ROUNDING` est définie par défaut sur N et les opérations ne seront pas transférées sur la source de données distante à moins que l'option ne soit définie sur Y. Pour plus d'informations sur l'option de serveur `SAME_DECFLT_ROUNDING`, voir Référence des options de base de données DB2.
3. Pour la version 9.5, ou une version antérieure, l'échelle distante pour TIMESTAMP est 6.

Mappages de types de données inversés par défaut pour les sources de données DB2 for System i

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données DB2 for System i.

Tableau 122. Mappages de types de données inversé par défaut pour DB2 for System i (les colonnes ne sont pas toutes représentées)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données binaires	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHARACTER	-	-	N
CHARACTER	-	-	-	-	Y	-	CHARACTER	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	NUMERIC	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	FLOAT	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP(<i>p</i>)	-	-	<i>p</i>	<i>p</i>	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPHIC	-	-	-	-	-	-	VARG	-	-	N

Mappages de types de données inversés par défaut pour les sources de données DB2 pour VM et VSE

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données DB2 pour VM et VSE.

Tableau 123. Mappages de types de données inversés par défaut pour DB2 pour VM et VSE (les colonnes ne sont pas toutes représentées)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	-
CHARACTER	-	-	-	-	Y	-	CHAR	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-

Tableau 123. Mappages de types de données inversés par défaut pour DB2 pour VM et VSE (les colonnes ne sont pas toutes représentées) (suite)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPH	-	-	-	-	-	-	VARGRAPH	-	-	N

Mappages de types de données inversés par défaut pour les sources de données DB2 for z/OS

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données DB2 for z/OS.

Tableau 124. Mappages de types de données inversés par défaut pour DB2 for z/OS (les colonnes ne sont pas toutes représentées)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	Y	-	CHAR	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	N

Mappages de types de données inversés par défaut pour les sources de données Informix

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données Informix.

Tableau 125. Mappages de types de données inversés par défaut pour Informix

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BIGINT ¹	-	-	-	-	-	-	DECIMAL	19	-	-
BIGINT ²	-	-	-	-	-	-	INT8	-	-	-
BLOB	1	2147483647	-	-	-	-	BYTE	-	-	-
CHARACTER	-	-	-	-	N	-	CHAR	-	-	-
CHARACTER	-	-	-	-	Y	-	BYTE	-	-	-
CLOB	1	2147483647	-	-	-	-	TEXT	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	SMALLFLOAT	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	DATETIME	6	10	-
TIMESTAMP	-	10	-	-	-	-	DATETIME	0	15	-
VARCHAR	1	254	-	-	N	-	VARCHAR	-	-	-
VARCHAR ¹	255	32672	-	-	N	-	TEXT	-	-	-
VARCHAR	-	-	-	-	Y	-	BYTE	-	-	-
VARCHAR ²	255	2048	-	-	N	-	LVARCHAR	-	-	-
VARCHAR ²	2049	32672	-	-	N	-	TEXT	-	-	-

Remarque :

1. Ce mappage est uniquement valide avec la version 8 (ou une version antérieure) d'Informix.
2. Ce mappage est uniquement valide avec la version 9 (ou une version ultérieure) d'Informix.

Pour le type de données Informix DATETIME, le serveur fédéré utilise le qualificatif de haut niveau Informix pour REMOTE_LENGTH et le qualificatif de bas niveau Informix pour REMOTE_SCALE.

Les qualificatifs Informix sont les constantes "TU_" définies dans le fichier datatype.h d'Informix Client SDK. Les constantes sont les suivantes :

0 = YEAR	8 = MINUTE	13 = FRACTION(3)
2 = MONTH	10 = SECOND	14 = FRACTION(4)
4 = DAY	11 = FRACTION(1)	15 = FRACTION(5)
6 = HOUR	12 = FRACTION(2)	

Mappages de types de données inversés par défaut pour les sources de données JDBC

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données JDBC qui répondent aux mappages de types du pilote JDBC DB2.

Tableau 126. Mappages de types de données inversés par défaut pour JDBC

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BIGINT	-	-	-	-	-	-	BIGINT	-	-	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-

Tableau 126. Mappages de types de données inversés par défaut pour JDBC (suite)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
CHAR	-	-	-	-	Y	-	BINARY	-	-	-
CHAR	-	-	-	-	-	-	CHAR	-	-	-
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	NCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	NCHAR	-	-	-
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	-	-	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP(p)	-	min(9,p)	-
VARCHAR	-	-	-	-	Y	-	VARBINARY	-	-	-
VARCHAR	-	-	-	-	N	-	VARCHAR	-	-	-
VARGRAPHIC	-	-	-	-	-	-	NVARCHAR	-	-	-

Mappages de types de données inversés par défaut pour les sources de données Microsoft SQL Server

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données Microsoft SQL Server.

Tableau 127. Mappages de types de données inversés par défaut pour Microsoft SQL Server (les colonnes ne sont pas toutes représentées)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BIGINT ¹	-	-	-	-	-	-	bigint	-	-	-
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	Y	-	binary	-	-	-
CHARACTER	-	-	-	-	N	-	char	-	-	-
CLOB	-	-	-	-	-	-	text	-	-	-
DATE	-	4	-	-	-	-	datetime	-	-	-
DECIMAL	-	-	-	-	-	-	decimal	-	-	-
DOUBLE	-	8	-	-	-	-	float	-	-	-
INTEGER	-	-	-	-	-	-	int	-	-	-
SMALLINT	-	-	-	-	-	-	smallint	-	-	-
REAL	-	4	-	-	-	-	real	-	-	-
TIME	-	3	-	-	-	-	datetime	-	-	-
TIMESTAMP	-	10	-	-	-	-	datetime	-	-	-
VARCHAR	1	8000	-	-	N	-	varchar	-	-	-
VARCHAR	8001	32672	-	-	N	-	text	-	-	-

Tableau 127. Mappages de types de données inversés par défaut pour Microsoft SQL Server (les colonnes ne sont pas toutes représentées) (suite)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
VARCHAR	1	8000	-	-	Y	-	varbinary	-	-	-
VARCHAR	8001	32672	-	-	Y	-	image	-	-	-

Remarque :

1. Ce mappage est uniquement valide avec Microsoft SQL Server 2000.

Mappages de types de données inversés par défaut pour les sources de données ODBC

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données ODBC.

Tableau 128. Mappages de types de données inversés par défaut pour ODBC (les colonnes ne sont pas toutes représentées)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BIGINT	-	-	-	-	-	-	SQL_BIGINT	-	-	-
BLOB	-	-	-	-	-	-	SQL_LONGVARBINARY	-	-	-
CHAR	-	-	-	-	Y	-	SQL_BINARY	-	-	-
CHAR	-	-	-	-	N	-	SQL_CHAR	-	-	-
CLOB	-	-	-	-	-	-	SQL_LONGVARCHAR	-	-	-
DATE	-	4	-	-	-	-	SQL_TYPE_DATE	-	-	-
DBCLOB	-	-	-	-	-	-	SQL_WLONGVARCHAR	-	-	-
DECIMAL	-	-	-	-	-	-	SQL_DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	SQL_DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	SQL_FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	SQL_WCHAR	-	-	-
INTEGER	-	-	-	-	-	-	SQL_INTEGER	-	-	-
REAL	-	4	-	-	-	-	SQL_REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SQL_SMALLINT	-	-	-
TIME	-	3	-	-	-	-	SQL_TYPE_TIME	-	-	-
TIMESTAMP	-	10	-	-	-	-	SQL_TYPE_TIMESTAMP(p)	-	-	-
VARCHAR	-	-	-	-	Y	-	SQL_VARBINARY	-	-	-
VARCHAR	-	-	-	-	N	-	SQL_VARCHAR	-	-	-
VARGRAPHIC	-	-	-	-	Y	-	SQL_WVARCHAR	-	-	-

Mappages de types de données inversés par défaut pour les sources de données Oracle NET8

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données Oracle NET8.

Tableau 129. Mappages de types de données inversés par défaut pour Oracle NET8

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BIGINT	0	8	0	0	N	\0	NUMBER	19	0	N
BLOB	0	2147483647	0	0	Y	\0	BLOB	0	0	Y
CHARACTER	1	254	0	0	N	\0	CHAR	0	0	N
CHARACTER	1	254	0	0	Y	\0	RAW	0	0	Y
CLOB	0	2147483647	0	0	N	\0	CLOB	0	0	N
DATE ¹	0	4	0	0	N	\0	DATE	0	0	N
DECIMAL	0	0	0	0	N	\0	NUMBER	0	0	N
DECFLOAT	0	8	0	0	N	\0	NUMBER	0	0	N
DECFLOAT	0	16	0	0	N	\0	NUMBER	0	0	N
DOUBLE	0	8	0	0	N	\0	FLOAT	126	0	N
FLOAT	0	8	0	0	N	\0	FLOAT	126	0	N
INTEGER	0	4	0	0	N	\0	NUMBER	10	0	N
REAL	0	4	0	0	N	\0	FLOAT	63	0	N
SMALLINT	0	2	0	0	N	\0	NUMBER	5	0	N
TIME	0	3	0	0	N	\0	DATE	0	0	N
TIMESTAMP ²	0	10	0	0	N	\0	DATE	0	0	N
TIMESTAMP(p) ³	-	-	-	-	N	\0	TIMESTAMP(p)-	-	-	N
VARCHAR	1	4000	0	0	N	\0	VARCHAR2	0	0	N
VARCHAR	1	2000	0	0	Y	\0	RAW	0	0	Y

Remarque :

1. Lorsque le paramètre date_compat est défini sur OFF, le type fédéré DATE est mappé vers la date Oracle. Lorsque le paramètre date_compat est défini sur ON, le type fédéré DATE (équivalent à TIMESTAMP(0)) est mappé vers Oracle TIMESTAMP(0).
2. Ce mappage est uniquement valide avec Oracle Version 8.
3.
 - TIMESTAMP(p) représente un horodatage avec une échelle de variable comprise entre 0 et 9 pour Oracle et 0 et 12 pour la fédération. Lorsque l'échelle va de 0 à 9, le type TIMESTAMP distant d'Oracle a la même échelle que le type TIMESTAMP fédéré. Si l'échelle du type TIMESTAMP fédéré est supérieure à 9, l'échelle correspondante du type TIMESTAMP d'Oracle est 9, qui correspond à l'échelle d'Oracle la plus élevée.
 - Ce mappage est uniquement valide avec Oracle Version 9, 10 et 11.

Mappages de types de données inversés par défaut pour les sources de données Sybase

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données Sybase.

Tableau 130. Mappages de types de données inversés par défaut pour Sybase CTLIB

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BIGINT	-	-	-	-	-	-	decimal	19	0	-
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	N	-	char	-	-	-
CHARACTER	-	-	-	-	Y	-	binary	-	-	-
CLOB	-	-	-	-	-	-	text	-	-	-

Tableau 130. Mappages de types de données inversés par défaut pour Sybase CTLIB (suite)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
DATE	-	-	-	-	-	-	datetime	-	-	-
DECIMAL	-	-	-	-	-	-	decimal	-	-	-
DOUBLE	-	-	-	-	-	-	float	-	-	-
INTEGER	-	-	-	-	-	-	integer	-	-	-
REAL	-	-	-	-	-	-	real	-	-	-
SMALLINT	-	-	-	-	-	-	smallint	-	-	-
TIME	-	-	-	-	-	-	datetime	-	-	-
TIMESTAMP	-	-	-	-	-	-	datetime	-	-	-
VARCHAR ¹	1	255	-	-	N	-	varchar	-	-	-
VARCHAR ¹	256	32672	-	-	N	-	text	-	-	-
VARCHAR ²	1	16384	-	-	N	-	varchar	-	-	-
VARCHAR ²	16385	32672	-	-	N	-	text	-	-	-
VARCHAR ¹	1	255	-	-	Y	-	varbinary	-	-	-
VARCHAR ¹	256	32672	-	-	Y	-	image	-	-	-
VARCHAR ²	1	16384	-	-	Y	-	varbinary	-	-	-
VARCHAR ²	16385	32672	-	-	Y	-	image	-	-	-

Remarque :

1. Ce mappage est uniquement valide pour CTLIB avec la version 12.0 (ou une version antérieure) de Sybase.
2. Ce mappage est uniquement valide pour CTLIB avec la version 12.5 (ou une version ultérieure) de Sybase.

Mappages de types de données inversés par défaut pour les sources de données Teradata

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données Teradata.

Tableau 131. Mappages de types de données inversés par défaut pour Teradata (les colonnes ne sont pas toutes représentées)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BLOB	1	2097088000	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHARACTER	-	-	-
CHARACTER	-	-	-	-	Y	-	BYTE	-	-	-
CLOB	1	2097088000	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	-	-
DBCLOB ¹	1	64000	-	-	-	-	VARGRAPHIC	-	-	-
DECIMAL	1	18	0	18	-	-	DECIMAL	-	-	-
DECIMAL	19	31	0	31	-	-	FLOAT	8	-	-
DOUBLE	-	-	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	-
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
REAL	-	-	-	-	-	-	FLOAT	8	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-

Tableau 131. Mappages de types de données inversés par défaut pour Teradata (les colonnes ne sont pas toutes représentées) (suite)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
TIME	-	-	-	-	-	-	TIME	15	-	-
TIMESTAMP	10	10	6	6	-	-	TIMESTAMP	26	6	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	Y	-	VARBYTE	-	-	-
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	-

Remarque :

1. Le type de données Teradata VARGRAPHIC peut comprendre uniquement la longueur indiquée (1 à 32 000) d'un type d'objet DBCLOB.

Mappages de types de données par défaut Unicode

Certaines sources de données prennent en charge les mappages de types de données aval et les mappages de types de données inversés pour les bases de données Unicode.

Mappages de types de données aval par défaut Unicode pour les sources de données JDBC

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données JDBC avec une base de données fédérée Unicode.

Tableau 132. Mappages de types de données aval par défaut Unicode pour les sources de données JDBC

UTF-8	JDBC	
Type de données	Type de données	Longueur
CHAR	CHAR	1 à 254 octets
VARCHAR	VARCHAR	1 à 32 672 octets
CLOB	CLOB	-
GRAPHIC	NCHAR	1 à 127 caractères
VARGRAPHIC	NVARCHAR	1 à 16 336 caractères
DBCLOB	NCLOB	-

Mappages de types de données inversés par défaut Unicode pour les sources de données JDBC

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données JDBC avec une base de données fédérée Unicode.

Tableau 133. Mappages de types de données inversés par défaut Unicode pour les sources de données JDBC

UTF-8	JDBC	
Type de données	Longueur	Type de données
CHAR	1 à 254 octets	CHAR

Tableau 133. Mappages de types de données inversés par défaut Unicode pour les sources de données JDBC (suite)

UTF-8		JDBC
Type de données	Longueur	Type de données
VARCHAR	1 à 32 672 octets	VARCHAR
CLOB	1 à 2 147 483 647 octets	CLOB
GRAPHIC	1 à 127 caractères	NCHAR
VARGRAPHIC	1 à 16 336 caractères	NVARCHAR
DBCLOB	1 à 1 073 741 823 caractères	NCLOB

Mappages de types de données aval par défaut Unicode pour les sources de données Microsoft SQL Server

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données Microsoft SQL Server avec une base de données fédérée Unicode.

Tableau 134. Mappages de types de données aval par défaut Unicode pour les sources de données Microsoft SQL Server

UTF-8	Microsoft SQL Server	
Type de données	Type de données	Longueur
CHAR	CHAR	1 à 254 octets
VARCHAR	CHAR	255 à 8 000 octets
	VARCHAR	1 à 8 000 octets
CLOB	TEXT	-
GRAPHIC	NCHAR	1 à 127 caractères
VARGRAPHIC	NCHAR	128 à 16 336 caractères
	NVARCHAR	1 à 16 336 caractères
DBCLOB	NTEXT	-

Mappages de types de données inversés par défaut Unicode pour les sources de données Microsoft SQL Server

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données Microsoft SQL Server avec une base de données fédérée Unicode.

Tableau 135. Mappages de types de données inversés par défaut Unicode pour les sources de données Microsoft SQL Server

UTF-8	Microsoft SQL Server	
Type de données	Longueur	Type de données
CHAR	1 à 254 octets	CHAR
VARCHAR	1 à 32 672 octets	VARCHAR
CLOB	1 à 2 147 483 647 octets	TEXT
GRAPHIC	1 à 127 caractères	NCHAR
VARGRAPHIC	1 à 16 336 caractères	NVARCHAR
DBCLOB	1 à 1 073 741 823 caractères	NTEXT

Mappages de types de données aval par défaut Unicode pour les sources de données NET8

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données NET8 avec une base de données fédérée Unicode.

Tableau 136. Mappages de types de données aval par défaut Unicode pour les sources de données NET8

UTF-8		Oracle
Type de données	Type de données	Longueur
CHAR	CHAR	1 à 254 octets
VARCHAR	CHAR	255 à 2 000 octets
	VARCHAR2	1 à 4 000 octets
DBCLOB	NCLOB	
GRAPHIC	NCHAR	1 à 127 caractères
VARGRAPHIC	NCHAR	128 à 1 000 caractères
	NVARCHAR2	1 à 2 000 caractères

Mappages de types de données inversés par défaut Unicode pour les sources de données NET8

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données NET8 avec une base de données fédérée Unicode.

Tableau 137. Mappages de types de données inversés par défaut Unicode pour les sources de données NET8

UTF-8		Oracle
Type de données	Longueur	Type de données
CHAR	1 à 254 octets	CHAR
VARCHAR	1 à 4 000 octets	VARCHAR2
CLOB	1 à 2 147 483 647 octets	CLOB
GRAPHIC	1 à 127 caractères	NCHAR
VARGRAPHIC	1 à 2 000 caractères	NVARCHAR2
DBCLOB	1 à 1 073 741 823 caractères	NCLOB

Mappages de types de données aval par défaut Unicode pour les sources de données ODBC

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données ODBC avec une base de données fédérée Unicode.

Tableau 138. Mappages de types de données aval par défaut Unicode pour les sources de données ODBC

UTF-8		ODBC
Type de données	Type de données	Longueur
CHAR	SQL_CHAR	1 à 254 octets

Tableau 138. Mappages de types de données aval par défaut Unicode pour les sources de données ODBC (suite)

UTF-8	ODBC	
Type de données	Type de données	Longueur
VARCHAR	SQL_CHAR	255 à 32 672 octets
	SQL_VARCHAR	1 à 32 672 octets
CLOB	SQL_LONGVARCHAR	-
GRAPHIC	SQL_WCHAR	1 à 127 caractères
VARGRAPHIC	SQL_WCHAR	128 à 16 336 caractères
	SQL_WVARCHAR	1 à 16 336 caractères
DBCLOB	SQL_WLONGVARCHAR	-

Mappages de types de données inversés par défaut Unicode pour les sources de données ODBC

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données ODBC avec une base de données fédérée Unicode.

Tableau 139. Mappages de types de données inversés par défaut Unicode pour les sources de données ODBC

UTF-8	ODBC	
Type de données	Longueur	Type de données
CHAR	1 à 254 octets	SQL_CHAR
VARCHAR	1 à 32 672 octets	SQL_VARCHAR
CLOB	1 à 2 147 483 647 octets	SQL_LONGVARCHAR
GRAPHIC	1 à 127 caractères	SQL_WCHAR
VARGRAPHIC	1 à 16 336 caractères	SQL_WVARCHAR
DBCLOB	1 à 1 073 741 823 caractères	SQL_WLONGVARCHAR

Mappages de types de données aval par défaut Unicode pour les sources de données Sybase

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données Sybase CTLIB avec une base de données fédérée Unicode.

Tableau 140. Mappages de types de données aval par défaut Unicode pour les sources de données Sybase CTLIB

UTF-8	Sybase	
Type de données	Type de données	Longueur
CHAR	char	1 à 254 octets
	nchar	1 à 127 caractères
VARCHAR	char	255 à 32 672 octets
	varchar	1 à 32 672 octets
	nchar	128 à 16 336 caractères
	nvarchar	1 à 16 336 caractères

Tableau 140. Mappages de types de données aval par défaut Unicode pour les sources de données Sybase CTLIB (suite)

UTF-8	Sybase	
Type de données	Type de données	Longueur
CLOB	text	
GRAPHIC	unichar	1 à 127 caractères
VARGRAPHIC	unichar	128 à 16 336 caractères
	univarchar	1 à 16 336 caractères

Mappages de types de données inversés par défaut Unicode pour les sources de données Sybase

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données Sybase CTLIB avec une base de données fédérée Unicode.

Tableau 141. Mappages de types de données inversés par défaut Unicode pour les sources de données Sybase CTLIB

UTF-8	Sybase	
Type de données	Longueur	Type de données
CHAR	1 à 254 octets	char
VARCHAR	1 à 32 672 octets	varchar
CLOB	1 à 2 147 483 647 octets	text
GRAPHIC	1 à 127 caractères	unichar
VARGRAPHIC	1 à 16 336 caractères	univarchar

Types de données pris en charge pour les sources de données non relationnelles

Pour la plupart des sources de données non relationnelles, vous devez remplir les informations contenues dans les colonnes, y compris le type de données, lorsque vous créez les pseudonymes pour accéder à la source de données.

Certains encapsuleurs non relationnels créent toutes les colonnes nécessaires pour accéder à une source de données. On appelle ces colonnes les *colonnes fixes*. Avec d'autres encapsuleurs, vous pourrez indiquer certains types de données, voire tous, pour les colonnes dans l'instruction CREATE NICKNAME.

Les sections suivantes, répertorient les encapsuleurs pour lesquels vous pouvez indiquer les types de données, et les types de données pris en charge par les encapsuleurs.

Types de données pris en charge par l'encapsuleur BioRS

Le tableau suivant répertorie les types de données DB2 pris en charge par l'encapsuleur BioRS.

Tableau 142. Types de données BioRS mappés vers les types de données DB2

Types de données BioRS	Type de données DB2
AUTHOR	CHARACTER, CLOB, VARCHAR
DATE	CHARACTER, CLOB, VARCHAR
NUMBER	CHARACTER, CLOB, VARCHAR
REFERENCE	CHARACTER, CLOB, VARCHAR
TEXT	CHARACTER, CLOB, VARCHAR

La longueur maximum autorisée pour le type de données CLOB est de 5 mégaoctets.

Types de données pris en charge par l'encapsuleur Excel

Le tableau suivant répertorie les types de données DB2 pris en charge par l'encapsuleur Excel.

Tableau 143. Types de données Excel mappés vers les types de données DB2

Types de données Excel	Type de données DB2
character	CHARACTER
date	DATE
number	DOUBLE
number	FLOAT
integer	INTEGER
character	VARCHAR

Types de données pris en charge par l'encapsuleur Script

Le tableau suivant répertorie les types de données DB2 pris en charge par l'encapsuleur Script.

Tableau 144. Types de données Script mappés vers les types de données DB2

Types de données XML	Type de données DB2
character	BLOB
character	CHARACTER
character	CHARACTER FOR BIT DATA
character	CLOB (longueur maximum : 5 mégaoctets)
date	DATE
number	DECIMAL
number	DOUBLE
number	FLOAT
integer	INTEGER
number	REAL
integer	SMALLINT
character	VARCHAR
character	VARCHAR FOR BIT DATA

Types de données pris en charge par l'encapsuleur Fichiers structurés sous formes de tables

Le tableau suivant répertorie les types de données DB2 pris en charge par l'encapsuleur Fichiers structurés sous formes de tables.

Tableau 145. Types de données Fichiers structurés sous formes de tables mappés vers les types de données DB2

Types de données Fichiers structurés sous formes de tables	Type de données DB2
character	CHARACTER
character	CLOB (longueur maximum : 5 mégaoctets)
number	DECIMAL
number	DOUBLE
number	FLOAT
integer	INTEGER
number	REAL
integer	SMALLINT
character	VARCHAR

Types de données pris en charge par l'encapsuleur services Web

Le tableau suivant répertorie les types de données DB2 pris en charge par l'encapsuleur services Web. L'encapsuleur services Web utilise les types de données XML.

Tableau 146. Types de données XML mappés vers les types de données DB2 pour l'encapsuleur services Web

Types de données XML	Type de données DB2
character	BLOB
character	CHARACTER
character	CHARACTER FOR BIT DATA
character	CLOB (longueur maximum : 5 mégaoctets)
date	DATE
number	DECIMAL
number	DOUBLE
number	FLOAT
integer	INTEGER
number	REAL
integer	SMALLINT
character	VARCHAR
character	VARCHAR FOR BIT DATA

Types de données pris en charge par l'encapsuleur XML

Le tableau suivant répertorie les types de données DB2 pris en charge par l'encapsuleur XML.

Tableau 147. Types de données XML mappés vers les types de données DB2 pour l'encapsuleur XML

Types de données XML	Type de données DB2
character	BLOB
character	CHARACTER
character	CHARACTER FOR BIT DATA
character	CLOB (longueur maximum : 5 mégaoctets)
date	DATE
number	DECIMAL
number	DOUBLE
number	FLOAT
integer	INTEGER
number	REAL
integer	SMALLINT
character	VARCHAR
character	VARCHAR FOR BIT DATA
Document XML	XML

Documentation du produit

La documentation est disponible sur différents supports et dans plusieurs formats, notamment dans l'aide qui s'affiche directement dans l'interface du produit, dans le centre de documentation de la suite et dans des documents au format PDF.

Vous pouvez également commander en ligne des publications IBM au format papier ou via votre représentant IBM local.

Pour commander des publications en ligne, visitez le Centre de publications IBM à l'adresse www.ibm.com/shop/publications/order.

Vous pouvez envoyer vos commentaires sur la documentation en suivant les procédures ci-après :

- Formulaire de commentaire en ligne : www.ibm.com/software/data/rcf/
- Courriel : comments@us.ibm.com

Documentation accessible

La documentation est fournie au format XHTML. Vous pouvez la consulter dans la plupart des navigateurs Web.

XHTML vous permet de visualiser la documentation en fonction de préférences d'affichage définies dans votre navigateur. Cela vous permet d'utiliser des lecteurs d'écran ainsi que d'autres technologies d'assistance.

Les diagrammes de syntaxe sont livrés en notation décimale à points. Ce format est uniquement disponible si vous accédez à la documentation en ligne à l'aide d'un lecteur d'écran.

Remarques

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 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

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 Japan

Le paragraphe ci-dessous ne s'applique ni au Royaume-Uni, ni dans aucun 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 CONTREFAÇON 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
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 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 de l'ICA, des Conditions internationales d'utilisation des 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.

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.

LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Les exemples de programme sont fournis "EN L'ETAT", sans garantie aucune. IBM ne sera en aucun cas responsable des dommages liés à l'utilisation de ces exemples de programmes.

Toute copie totale ou partielle de ces exemples de programmes et des oeuvres qui en sont dérivées doit comprendre une notice de copyright, libellée comme suit :

© (nom de votre société) (année). Des segments de code sont dérivés des Programmes exemples d'IBM Corp. © Copyright IBM Corp. _entrez l'année ou les années_. All rights reserved.

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 marques IBM et certains marques non IBM sont repérées lors de leur première apparition dans ce document avec le symbole adéquat.

IBM, le logo IBM et ibm.com sont des marques ou des marques déposées d'International Business Machines Corp., enregistrées dans de nombreuses juridictions internationales. Les noms de produits ou de services autres peuvent être des marques d'IBM ou de sociétés autres. Une liste à jour des marques d'IBM est disponible sur le Web dans "Copyright and trademark information" à l'adresse suivante : www.ibm.com/legal/copytrade.shtml.

Les termes qui suivent sont des marques d'autres sociétés :

Adobe, le logo Adobe, PostScript, le logo PostScript sont des marques d'Adobe Systems Incorporated aux Etats-Unis et/ou dans certains autres pays.

IT Infrastructure Library est une marque de The Central Computer and Telecommunications Agency, qui fait désormais partie de The Office of Government Commerce.

Intel, le logo Intel, Intel Inside, le logo Intel Inside, Intel Centrino, le logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, et Pentium sont des marques d'Intel Corporation ou de ses filiales aux Etats-Unis et dans certains autres pays.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

ITIL est une marque déposée et une marque déposée communautaire de l'Office of Government Commerce et est déposée auprès du Patent and Trademark Office américain.

UNIX est une marque enregistrée de The Open Group aux Etats-Unis et/ou dans certains autres pays.

Cell Broadband Engine est une marque de Sony Computer Entertainment, Inc. aux Etats-Unis et/ou dans certains autres pays. Elle est utilisée sous licence.

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques de Sun Microsystems, Inc. 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

accessibilité 411
Accessibilité 409
alias
 CREATE NICKNAME
 exemples 200
 création 243, 244
 Services Web 246
 description 181, 254
 enregistrement 65
 enregistrement des tables, vues et
 synonymes Informix 91
 enregistrement pour les scripts 181
 enregistrement pour les sources de
 données Microsoft SQL Server 115
 enregistrement pour les sources de
 données ODBC 129, 148
 Langage de description des services
 Web (WSDL : Web Services
 Description Language) 233
 Oracle, sources de données 169
 sources de données de services
 Web 243, 244
 Sources de données JDBC 102
 Sybase, sources de données 199
 techniques de base de données 256
 Teradata, sources de données 228,
 229

B

banque de données
 description 21
Base de données DB2
 fonctions fédérées prises en
 charge 291
 options d'encapsuleur 299
 options de colonne 299
 options de mappage utilisateur 299
 options de pseudonyme 299
 options de serveur 299
bases de données
 catalogage distant 58
bases de données fédérées
 création 7
 jeux de codes 8
 séquence de classement 8
bibliothèque
 encapsuleur de script 179
bibliothèques
 encapsuleur BioRS 25
 encapsuleur de fichier structuré sous
 forme de tables 206
 Encapsuleur Excel 72
 encapsuleur Teradata 222
 encapsuleur XML 268
bibliothèques d'encapsuleur 240
 noms de fichier BioRS 25
 Noms de fichier Excel 72
 noms de fichier script 179

bibliothèques d'encapsuleur (*suite*)
 noms de fichier structuré sous forme
 de tables 206
 noms de fichier Teradata 222
 XML 268
Biomax Informatics 21
BioRS
 fonctions fédérées prises en
 charge 291
 options d'encapsuleur 295
 options de colonne 295
 options de mappage utilisateur 295
 options de pseudonyme 295
 options de serveur 295
 types de données, pris en charge 404

C

caractères majuscules
 utilisation dans les instructions
 SQL 15
caractères minuscules
 utilisation dans les instructions
 SQL 15
catalogue
 Voir catalogue global 369
catalogue global
 vues contenant des informations
 fédérées 369
Centre de contrôle
 accès 1
classeurs Microsoft Excel
 accès 134
 droits d'accès 137
clés de licence
 Fédération 7
commande db2fedsvrcfg
 création de tables de test 133, 151
 exemples 132, 150
 syntaxe 132, 150
commande db2script_daemon 177
commandes
 émission 1
configuration
 serveurs fédérés 18
connectivité
 identification et résolution des
 incidents pour les sources de
 données Oracle 168
 test des connexions au serveur
 Oracle 167
connexions
 identification et résolution des
 incidents 64, 90, 102, 114, 128, 147,
 168, 198, 227
 test 63
contextes accrédités fédérés
 sources de données prises en
 charge 291

contrôle d'accès basé sur les labels
 sources de données prises en
 charge 291
conversion de la page de codes
 Informix 82
conversion de pages de codes
 Sources de données Oracle 161
CREATE NICKNAME
 Sources de données JDBC 103
CREATE NICKNAME (instruction)
 exemples, Excel 74
 exemples BioRS 35
 exemples de fichiers structurés sous
 forme de tables 209
 exemples XML 280
 identification et résolution des
 incidents pour les sources de
 données Teradata 230
 Informix 92
 sources de données de script 181
 sources de données de services
 Web 243, 244
 syntaxe, Excel 73
 syntaxe BioRS 34
 syntaxe de fichier structuré sous
 forme de tables 207
 Teradata 229
CREATE SERVER
 source de données JDBC 98
CREATE USER MAPPING
 sources de données JDBC 100
CREATE WRAPPER
 exemples pour ODBC 123, 141
 Sources de données JDBC 96
CREATE WRAPPER (instruction)
 enregistrement de sources de données
 OLE DB 153
 Excel, exemple 71
 exemple de fichier structuré sous
 forme de tables 205
 exemples BioRS 26
 exemples XML 268
 syntaxe BioRS 25
 syntaxe de fichier structuré sous
 forme de tables 205
 syntaxe XML 267

D

DAEMON_LOGFILE_DIR
 démon Script 174
DAEMON_PORT
 démon Script 174
DB2 for Linux, UNIX et Windows
 mappages de types de données aval
 par défaut 377
 mappages de types de données
 inversés par défaut 391
DB2 for System i
 mappages de types de données aval
 par défaut 377

- DB2 for System i (*suite*)
 - mappages de types de données inversés par défaut 391
- DB2 for z/OS et OS/390
 - mappages de types de données aval par défaut 377
 - mappages de types de données inversés par défaut 391
- DB2 pour VM et VSE
 - mappages de types de données aval par défaut 377
 - mappages de types de données inversés par défaut 391
- définitions de serveur
 - enregistrement 60
 - enregistrement des sources de données Informix
 - ligne de commande DB2 85
 - enregistrement des sources de données OLE DB 155
 - enregistrement des sources de données Sybase 194
 - enregistrement des sources de données Teradata 223
 - enregistrement pour les sources de données Microsoft SQL Server 110
 - enregistrement pour les sources de données ODBC 124, 143
 - enregistrement pour les sources de données Oracle 164
 - Services Web 240
 - source de données JDBC 97
- démon
 - configuration pour le script 174
 - démarrage pour un script 176
 - fichier de configuration
 - exemples de script 174
- documentation
 - accessible 411
- Documentation
 - Accessible 409

E

- élément AllText
 - BioRS 45
- emplacement, par défaut 2
- en-têtes SOAP
 - accès fédéré 252
- encapsuleur DB2
 - enregistrement 59
- encapsuleur de script
 - création 179
 - fonctions personnalisées 173
 - généralités 171
- Encapsuleur de services Web
 - sources de données 232
- encapsuleur DRDA
 - noms de fichiers de bibliothèque 59
- encapsuleurs
 - bibliothèques de l'encapsuleur Microsoft SQL Server 109
 - configuration de l'accès aux sources de données Teradata 215
 - configuration des encapsuleurs JDBC 92

- encapsuleurs (*suite*)
 - configuration des encapsuleurs ODBC 118, 138
 - configuration des encapsuleurs OLE DB 152
 - enregistrement d'encapsuleurs Oracle 163
 - enregistrement d'encapsuleurs Sybase 193
 - enregistrement d'encapsuleurs Teradata 221
 - enregistrement des encapsuleurs Microsoft SQL Server 108
 - enregistrement des encapsuleurs ODBC 121, 140
 - enregistrements des encapsuleurs JDBC 95
 - fichiers de bibliothèque d'encapsuleur Oracle 164
 - Fichiers de bibliothèque de l'encapsuleur JDBC 96
 - Fichiers de bibliothèque de l'encapsuleur ODBC 122, 141
 - Fichiers de bibliothèque de l'encapsuleur OLE DB 154
 - script 170, 171
 - sources de données de services Web 232, 239
 - vérification de la liaison d'exécution pour les sources de données Teradata 216
- encapsuleurs de services Web
 - création 239
- encapsuleurs Informix
 - amélioration des performances 91
 - création de mappages utilisateur 88
 - enregistrement 84
 - variables d'environnement 80, 81
- enregistrement des erreurs
 - désactivation de la consignation d'accès Teradata 232
- erreurs des opérations UPDATE ou DELETE
 - identification et résolution des incidents pour les sources de données Teradata 231
- Excel
 - fonctions fédérées prises en charge 291
 - options d'encapsuleur 306
 - options de pseudonyme 306
 - options de serveur 306
- exemples
 - commande db2script_daemon 177
 - encapsuleurs de services Web 246, 256
 - exemples XML 288
 - Fichiers Excel 66, 67
 - instruction CREATE NICKNAME, BioRS 35
 - instruction CREATE NICKNAME, Excel 74
 - instruction CREATE NICKNAME, fichier structuré sous forme de tables 209
 - instruction CREATE SERVER, BioRS 28

- exemples (*suite*)
 - instruction CREATE SERVER, Excel 73
 - instruction CREATE SERVER, fichier structuré sous forme de tables 207
 - instruction CREATE SERVER, services Web 241
 - instruction CREATE SERVER, XML 270
 - instruction CREATE USER MAPPING, BioRS 32
 - instruction CREATE VIEW, XML 285
 - instruction CREATE WRAPPER, BioRS 26
 - instruction CREATE WRAPPER, Excel 71
 - instruction CREATE WRAPPER, fichier structuré sous forme de tables 205
 - instruction CREATE WRAPPER, XML 268
 - instruction UPDATE, BioRS 54
 - mappage de document XML 263
 - requêtes BioRS 43, 45
 - requêtes Excel 75
 - Serveur proxy
 - exemple de mappage utilisateur BioRS 32
 - sources de données BioRS
 - serveur proxy, exemple de mappage utilisateur 32
 - sources de données de script, CREATE NICKNAME 181
 - sources de données de script, CREATE SERVER 180
- exemples, CREATE SERVER
 - Informix 86
 - Teradata 224
- exemples, instruction CREATE USER MAPPING
 - Informix 89
 - Sybase, sources de données 197
- exemples, requêtes
 - Services Web 259
- exemples avec CREATE NICKNAME
 - Microsoft SQL Server, sources de données 116
 - ODBC, sources de données 130, 148
 - Oracle, sources de données 170
- exemples avec CREATE SERVER
 - Microsoft SQL Server, sources de données 111
 - Oracle, sources de données 165
 - Sources de données de bases de données OLE 155
 - Sybase, sources de données 195
- exemples avec CREATE USER MAPPING
 - Microsoft SQL Server, sources de données 113
 - ODBC, sources de données 127, 146
 - Oracle, sources de données 166
 - Sources de données de bases de données OLE 157
 - Teradata, sources de données 225

F

- fédération
 - installation 1
- Fédération
 - activation 1
 - identification et résolution des incidents 6
- fenêtre Sortie d'action 18
- fichier db2dj.ini 2
 - restrictions 4
- fichier de bibliothèque d'encapsuleur
 - noms de fichiers 4
- fichier de configuration client Informix
 - configuration et test 79
- fichier de configuration client Oracle
 - configuration et tests 162
- fichier de configuration de client Sybase
 - configuration et tests 191, 192
- fichiers de bibliothèque
 - liaison 4, 6
 - noms de fichiers 4
- fichiers de configuration client Informix
 - configuration et test 79
- Fichiers Excel
 - préparation de l'accès aux 67
 - types de données, pris en charge 404
- Fichiers structurés sous forme de tables
 - fonctions fédérées prises en charge 291
 - options d'encapsuleur 351
 - options de colonne 351
 - options de pseudonyme 351
 - options de serveur 351
 - prise en charge du format Unicode 11, 12
 - types de données, pris en charge 404
- fonction de mise à jour des statistiques de pseudonyme
 - sources de données prises en charge 291
- fonction WSSCRIPT.ARGS 173
- fonctions personnalisées
 - BioRS 38
 - BioRS, descriptions 24
 - BioRS, enregistrement 23
 - script 173

G

- guillemets
 - utilisation dans les instructions SQL 15

I

- ID utilisateur
 - sensibilité à la casse 15
- identification et résolution des incidents
 - accès à une source de données 1
 - configuration du serveur fédéré 1
 - connexions 64, 90, 102, 114, 128, 147, 168, 198, 227
 - installation de la fédération 4, 6
 - serveur fédéré 4

- importation de données
 - sources de données prises en charge 291
- indicateurs d'état de santé fédérés
 - sources de données prises en charge 291
- Informix
 - chemin du répertoire par défaut 84
 - conversion de la page de codes 82
 - exemple avec l'instruction CREATE NICKNAME 92
 - fichiers de bibliothèque 6
 - fonctions fédérées prises en charge 291
 - instruction CREATE SERVER 86
 - instruction CREATE USER MAPPING 89
 - mappages de types de données aval par défaut 377
 - mappages de types de données inversés par défaut 391
 - noms de fichier de bibliothèque 84
 - options d'encapsuleur 307
 - options de colonne 307
 - options de mappage utilisateur 307
 - options de serveur 307
 - script djsxlinkInformix 6
- instruction CREATE FUNCTION
 - syntaxe BioRS 23
- instruction CREATE SERVER
 - exemple, Excel 73
 - exemple de services Web 241
 - exemples BioRS 28
 - exemples de fichiers structurés sous forme de tables 207
 - exemples XML 270
 - Informix 86
 - sources de données de script 180
 - Sybase, sources de données 195
 - syntaxe, Excel 73
 - syntaxe BioRS 27
 - syntaxe de fichier structuré sous forme de tables 207
 - syntaxe XML 269
 - Teradata 224
- Instruction CREATE SERVER
 - exemples pour ODBC 125, 143
 - sources de données de la famille DB2 61
- instruction CREATE USER MAPPING
 - exemples BioRS 32
 - Informix 89
 - Sybase, sources de données 197
 - syntaxe BioRS 30
- Instruction CREATE VIEW
 - exemples XML 285
- instruction UPDATE
 - exemple, cardinalité de colonne BioRS 54
 - exemple, cardinalité de la colonne `_ID_` de BioRS 56
 - exemple, cardinalité de pseudonyme BioRS 54
 - exemples
 - instruction UPDATE, BioRS 54
- interpréteur de commandes (CLP)
 - démarrage 1

- isolement au niveau de la connexion
 - sources de données prises en charge 291
- isolement de niveau d'instruction
 - sources de données prises en charge 291

J

- JDBC
 - fonctions fédérées prises en charge 291
 - options d'encapsuleur 313
 - options de colonne 313
 - options de mappage utilisateur 313
 - options de serveur 313
- jeux de caractères
 - identification et résolution des incidents des configurations de source de données Teradata 221, 231
- jeux de codes
 - base de données fédérée 8

L

- Langage de description des services Web (WSDL : Web Services Description Language)
 - alias 233
 - mappage aux pseudonymes 254
- lecteurs d'écran 411
- Lecteurs d'écran 409
- liaison
 - fichiers de bibliothèque 4
- licences
 - Fédération 7
- LOB
 - sources de données prises en charge, lecture et écriture 291
 - sources de données prises en charge, lecture seule 291

M

- mappage
 - documents XML 254
- mappages de fonctions
 - options 373
- mappages de type aval
 - mappages par défaut 377
 - Unicode
 - Microsoft SQL Server 401
 - ODBC, sources de données 402
 - Sources de données JDBC 400
 - sources de données NET8 402
 - Sybase, sources de données 403
- mappages de type de données
 - aval 377
 - inversé 391
- mappages de type inversé
 - mappages par défaut 391
 - Unicode
 - Microsoft SQL Server, sources de données 401
 - ODBC, sources de données 403

- mappages de type inversé *(suite)*
 - Unicode *(suite)*
 - Sources de données JDBC 400
 - sources de données NET8 402
 - Sybase, sources de données 404
- mappages de types de données aval par défaut
 - exemple pour Informix 389
 - exemple pour Microsoft SQL Server 389
 - Exemple pour Oracle 390
 - exemple pour Sybase 390
 - Exemple pour Teradata 390
 - exemples 389
 - Microsoft SQL Server, sources de données 383, 396
 - ODBC, sources de données 385
 - sources de données d'une base de données DB2 pour Linux, UNIX et Windows 377
 - sources de données DB2 for System i 378
 - sources de données DB2 for z/OS 380
 - sources de données DB2 pour VM et VSE 379
 - sources de données Informix 380, 395
 - Sources de données JDBC 382
 - sources de données Oracle NET8 386
 - Sybase, sources de données 386
 - Teradata, sources de données 388, 399
- mappages de types de données inversés par défaut
 - ODBC, sources de données 397
 - sources de données d'une base de données DB2 pour Linux, UNIX et Windows 392
 - sources de données DB2 for System i 393
 - sources de données DB2 for z/OS 394
 - sources de données DB2 pour VM et VSE 393
 - Sources de données JDBC 395
 - sources de données Oracle NET8 398
 - Sybase, sources de données 398
- mappages utilisateur
 - création 62
 - création pour les sources de données Informix 88
 - création pour les sources de données Sybase 197
 - définition pour les sources de données Microsoft SQL Server 113
 - définition pour les sources de données ODBC 127, 145
 - définition pour les sources de données Oracle 166
 - définition pour les sources de données Teradata 225
 - Sources de données de bases de données OLE 156
 - Sources de données JDBC 100
- marques 416

- Microsoft SQL Server
 - fichiers de bibliothèque 6
 - fonctions fédérées prises en charge 291
 - mappages de types de données aval par défaut 377
 - mappages de types de données aval par défaut Unicode 401
 - mappages de types de données inversés par défaut 391
 - options d'encapsuleur 320
 - options de colonne 320
 - options de mappage utilisateur 320
 - options de serveur 320
 - prise en charge du format Unicode 11
 - script djlMssql 6
- Microsoft SQL Server, sources de données
 - alias 115
 - configuration de l'accès 104
 - configuration des serveurs fédérés 104
 - configuration des variables d'environnement 107
 - enregistrement 108
 - enregistrement des définitions de serveur 110
 - exemple avec CREATE NICKNAME 116
 - exemples avec CREATE SERVER 111
 - exemples avec CREATE USER MAPPING 113
 - fichiers de bibliothèque 4
 - fichiers de bibliothèque de l'encapsuleur 109
 - identification et résolution des incidents de configuration 117
 - mappages de types de données aval par défaut 383, 396
 - mappages de types de données inversés par défaut Unicode 401
 - test des connexions à Microsoft SQL Server 114
 - variables d'environnement 2, 108
- mots de passe
- sensibilité à la casse 15

N

- noeuds
 - catalogage 57
 - DB2 57
- Notices légales 413

O

- ODBC
 - fonctions fédérées prises en charge 291
 - mappages de types de données aval par défaut 377
 - options d'encapsuleur 326
 - options de colonne 326
 - options de mappage utilisateur 326
 - options de serveur 326

- ODBC *(suite)*
 - prise en charge du format Unicode 11
 - variables d'environnement 2
- ODBC, sources de données
 - accès aux classeurs Microsoft Excel 134
 - configuration de l'accès aux sources de données IBM InfoSphere Classic Federation Server for z/OS 138
 - configuration de l'accès aux sources de données ODBC 118
 - configuration des serveurs fédérés 119
 - configuration du serveur fédéré 120
 - définition des mappages utilisateur 127, 145
 - droits d'accès 137
 - enregistrement 121, 140
 - enregistrement de pseudonymes 129, 148
 - enregistrement des définitions de serveur 124, 143
 - exemples avec CREATE NICKNAME 130, 148
 - exemples avec CREATE SERVER 125, 143
 - exemples avec CREATE USER MAPPING 127, 146
 - exemples avec CREATE WRAPPER 123, 141
 - fichiers de bibliothèque de l'encapsuleur 122, 141
 - mappages de types de données aval par défaut 385
 - mappages de types de données aval par défaut Unicode 402
 - mappages de types de données inversés par défaut 397
 - mappages de types de données inversés par défaut Unicode 403
 - optimisation avec l'utilitaire d'optimisation ODBC 131, 149
 - tables de test pour l'utilitaire d'optimisation ODBC 133, 151
 - test de l'accès aux 128, 146
- optimisation
 - sources de données de script 186
- optimisation de l'état asynchrone
 - sources de données prises en charge 291
- optimisation de requête
 - instructions pour les fichiers structurés sous forme de tables 214
 - instructions pour les requêtes BioRS 51
 - XML 287
- option CODEPAGE 11
- option de mappage de fonctions DISABLE
 - paramètres valides 373
- option de mappage de fonctions REMOTE_NAME
 - paramètres valides 373
- option de serveur FOLD_ID
 - sensibilité à la casse 15

- option de serveur FOLD_PW
 - sensibilité à la casse 15
- options d'encapsuleur
 - Base de données DB2 299
 - BioRS 295
 - Excel 306
 - Fichiers structurés sous forme de tables 351
 - Informix 307
 - JDBC 313
 - Microsoft SQL Server 320
 - ODBC 326
 - Oracle 331
 - Script 337
 - Services Web 354
 - Sybase 341
 - Teradata 347
 - XML 362
- options de colonne
 - Base de données DB2 299
 - BioRS 295
 - Fichiers structurés sous forme de tables 351
 - Informix 307
 - JDBC 313
 - Microsoft SQL Server 320
 - ODBC 326
 - Oracle 331
 - Script 337
 - Services Web 354
 - Sybase 341
 - Teradata 347
 - XML 362
- options de colonne de pseudonyme
 - encapsuleur de services Web 255
 - Encapsuleur de services Web 233
- options de mappage utilisateur
 - Base de données DB2 299
 - BioRS 295
 - Informix 307
 - JDBC 313
 - Microsoft SQL Server 320
 - ODBC 326
 - Oracle 331
 - Script 337
 - Services Web 354
 - Sybase 341
 - Teradata 347
 - XML 362
- options de pseudonyme
 - Base de données DB2 299
 - BioRS 295
 - Fichiers structurés sous forme de tables 351
 - Script 337
 - Services Web 354
 - XML 362
- options de serveur
 - Base de données DB2 299
 - BioRS 295
 - définition des options de serveur FOLD_ID et FOLD_PW 91
 - Excel 306
 - exemples pour OLE DB 155
 - Fichiers structurés sous forme de tables 351
 - Informix 307

- options de serveur (*suite*)
 - JDBC 313
 - Microsoft SQL Server 320
 - ODBC 326
 - Oracle 331
 - Script 337
 - Services Web 354
 - Sybase 341
 - Teradata 347
 - XML 362
- Oracle
 - fichiers de bibliothèque 6
 - fonctions fédérées prises en charge 291
 - mappages de types de données aval par défaut 377
 - mappages de types de données inversés par défaut 391
 - options d'encapsuleur 331
 - options de colonne 331
 - options de mappage utilisateur 331
 - options de serveur 331
 - script djlincOracle 6
- Oracle, sources de données
 - accès 157
 - ajout à des systèmes fédérés 157
 - configuration de la connectivité 162
 - définition de variables d'environnement 158
 - définition des mappages utilisateur 166
 - enregistrement 163
 - enregistrement de pseudonymes 169
 - enregistrement des définitions de serveur 164
 - exemples avec CREATE NICKNAME 170
 - exemples avec CREATE SERVER 165
 - exemples avec CREATE USER MAPPING 166
 - fichiers de bibliothèque 4
 - fichiers de bibliothèque de l'encapsuleur 164
 - variables d'environnement 2

P

- pages de codes 9, 11, 12
 - base de données fédérée 8
- paramètre CONCENTRATOR 1
- paramètre FEDERATED 1
- pilote ODBC Excel
 - accès aux classeurs Microsoft Excel 134
 - droits d'accès 137
- planification
 - accès aux données Excel 67
- points de sauvegarde d'une application
 - sources de données prises en charge 291
- problèmes de connexion
 - Microsoft SQL Server, sources de données 117
- procédures fédérées
 - sources de données prises en charge 291

- Proxy HTTP
 - sources de données prises en charge 291
- Proxy SOCKS
 - sources de données prises en charge 291
- pseudonymes XML
 - vues fédérées 285

R

- référéntiel de mappage utilisateur
 - sources de données prises en charge 291
- registre spécial USER DB2
 - exemples pour les sources de données OLE DB 157
- règles de licence
 - Fédération 7
- requêtes
 - accès aux sources de données Services Web 259
 - accès aux sources de données BioRS 38, 45
 - accès aux sources de données Excel 75
 - exemples
 - encapsuleurs de services Web 256
 - opérations d'égalité BioRS 42
 - requêtes XML 288
 - sources de données de script, accès aux sources de données 185
- restrictions
 - fichier db2dj.ini 4

S

- scénarios
 - accès aux fichiers Excel 76
- schémas XML
 - mappage aux pseudonymes 254
- Script
 - fonctions fédérées prises en charge 291
 - options d'encapsuleur 337
 - options de colonne 337
 - options de mappage utilisateur 337
 - options de pseudonyme 337
 - options de serveur 337
- scripts
 - configurer en tant que sources de données 170
- Secure Socket Layer (SSL)
 - sources de données prises en charge 291
- sécurité
 - Services Web 242
- sensibilité à la casse
 - conservation 15
- séquences de classement
 - base de données fédérée 8
- serveur fédéré
 - bases de données 7
 - commandes 1
 - configuration 1

- serveur fédéré (*suite*)
 - configuration de l'accès à Informix 78
 - identification et résolution des incidents 1
 - présentation de la configuration 12
- serveur Informix
 - test de la connexion 89
- serveur Oracle
 - test de la connexion 167
- serveur proxy
 - exemple d'encapsuleur BioRS 26
 - exemple de définition de serveur de services Web 241
- serveurs
 - test des connexions à Microsoft SQL Server 114
 - test des connexions aux serveurs Sybase 198
- serveurs fédérés
 - accès aux sources de données Microsoft SQL Server 105
 - catalogage de bases de données 58
 - configuration de l'accès à Microsoft SQL Server 104
 - configuration de l'accès aux sources de données Microsoft SQL Server 104
 - configuration de l'accès aux sources de données Sybase 186
 - configuration de plusieurs 18
 - configuration des sources de données 57
 - configuration pour JDBC 94
 - configuration pour ODBC 119, 120
 - test d'accès aux serveurs de sources de données JDBC 101
 - test de l'accès aux serveurs de source de données ODBC 128, 146
 - test de la connexion au serveur Teradata 216
- serveurs proxy
 - accès aux sources de données XML 273
- Services Web
 - alias 233, 246
 - définitions de serveur 240
 - exemple avec l'instruction CREATE SERVER 241
 - exemples 256
 - exemples de requêtes 259
 - fichiers de bibliothèque de l'encapsuleur 240
 - fonctions fédérées prises en charge 291
 - options d'encapsuleur 354
 - options de colonne 354
 - options de mappage utilisateur 354
 - options de pseudonyme 354
 - options de serveur 354
 - sécurité 242
 - serveur proxy, exemple de définition de serveur 241
 - types de données, pris en charge 404
 - vues fédérées 255
- sessions passe-système
 - sources de données prises en charge 291
- source de données JDBC
 - configuration des serveurs fédérés 94
 - enregistrement des définitions de serveur 97
 - exemples CREATE SERVER 98
 - exemples CREATE USER MAPPING 100
- sources de données
 - configuration 1, 57
 - configuration de l'accès à Informix 78
 - enregistrement 59
 - enregistrement des définitions de serveur Informix
 - ligne de commande DB2 85
 - enregistrement des encapsuleurs Informix 84
 - fichiers de messages 4
 - fonctions prises en charge 291
 - identification et résolution des incidents de connexion 64, 90, 102, 114, 128, 147, 168, 198, 227
 - options 295
 - présentation de la configuration 12
 - test de la connexion au serveur Informix 89
 - types de serveur valides 375
- sources de données BioRS 21
 - ajout de BioRS à un système fédéré 22
 - caractères génériques, dans les requêtes 38, 45
 - chemin du répertoire par défaut 25
 - conditions requises 22
 - définition de serveur,
 - enregistrement 27
 - description 21
 - élément AllText 45
 - encapsuleur, enregistrement 25
 - exemples avec l'instruction CREATE NICKNAME 35
 - exemples avec l'instruction CREATE SERVER 28
 - exemples avec l'instruction CREATE WRAPPER 26
 - exemples de requêtes 43, 45
 - fonctions personnalisées, descriptions 24
 - fonctions personnalisées, enregistrement 23
 - fonctions personnalisées, présentation 38
 - informations statistiques 52
 - mappages utilisateur, création 30
 - noms de bibliothèque de l'encapsuleur 25
 - optimisation des performances de l'encapsuleur 51
 - performances d'encapsuleur, optimisation 51
 - performances d'interrogation, optimisation 52
 - programme utilitaire admin_find 53
- sources de données BioRS (*suite*)
 - programme utilitaire
 - www_find.cgi 53
 - pseudonymes, enregistrement 34
 - requêtes, opérations d'égalité 42
 - requêtes, utilisation de prédicats de jointure 43
 - Serveur proxy
 - exemple de définition de serveur BioRS 28
 - serveur proxy, exemple d'encapsuleur 26
 - serveur proxy, exemple de définition de serveur 28
 - statistiques
 - cardinalité de colonne, mise à jour 54
 - cardinalité de la colonne _ID_, mise à jour 56
 - cardinalité de pseudonyme, mise à jour 54
 - statistiques de cardinalité, établissement 53
 - statistiques de cardinalité, pseudonymes 54
 - syntaxe de l'instruction CREATE FUNCTION 23
 - syntaxe de l'instruction CREATE NICKNAME 34
 - syntaxe de l'instruction CREATE SERVER 27
 - syntaxe de l'instruction CREATE USER MAPPING 30
 - syntaxe de l'instruction CREATE WRAPPER 25
- sources de données d'une base de données DB2 pour Linux, UNIX et Windows
 - mappages de types de données aval par défaut 377
 - mappages de types de données inversés par défaut 392
- sources de données DB2
 - alias 65
 - chemin de répertoire par défaut 59
 - configuration 57
 - création de pseudonymes 65
 - enregistrement 59
 - enregistrement de serveurs 60
 - fichiers de bibliothèque 59
 - instruction CREATE SERVER 61
 - mappages utilisateur 62
 - test des connexions 63
- Sources de données DB2 for System i
 - mappages de types de données aval par défaut 378
 - mappages de types de données inversés par défaut 393
- sources de données DB2 for z/OS
 - mappages de types de données aval par défaut 380
 - mappages de types de données inversés par défaut 394
- sources de données DB2 pour VM et VSE
 - mappages de types de données aval par défaut 379

- sources de données DB2 pour VM et VSE (*suite*)
 - mappages de types de données inversés par défaut 393
- Sources de données de bases de données OLE
 - configuration de l'accès 152
 - définition des mappages utilisateur 156
 - enregistrement 153
 - enregistrement des définitions de serveur 155
 - exemples avec CREATE SERVER 155
 - exemples avec CREATE USER MAPPING 157
 - noms de fichier de bibliothèque 154
 - options de serveur 155
- sources de données de fichiers structurés sous forme de tables 202
 - ajout à un système fédéré 204
 - attributs de type de fichier 203
 - autorisations pour l'accès aux fichiers 214
 - chemin du répertoire par défaut 206
 - conditions requises 204
 - définition de serveur, enregistrement 207
 - encapsuleur, enregistrement 205
 - exemple avec l'instruction CREATE WRAPPER 205
 - exemples avec l'instruction CREATE NICKNAME 209
 - exemples avec l'instruction CREATE SERVER 207
 - généralités 202
 - noms de bibliothèque de l'encapsuleur 206
 - optimisation de requête 214
 - présentation de l'encapsuleur 203
 - pseudonymes, enregistrement 207
 - syntaxe de l'instruction CREATE NICKNAME 207
 - syntaxe de l'instruction CREATE SERVER 207
 - syntaxe de l'instruction CREATE WRAPPER 205
- sources de données de script
 - définition de serveur 179
 - enregistrement de pseudonymes 181
 - exemple avec l'instruction CREATE NICKNAME 181
 - exemples de requêtes 185
 - instruction CREATE SERVER 180
 - noms de bibliothèque de l'encapsuleur 179
 - optimisation des performances 186
 - options de pseudonyme 182
- sources de données de services Web
 - enregistrement de pseudonymes 243
 - enregistrement de pseudonymes (centre de contrôle DB2) 244
 - enregistrement de pseudonymes (ligne de commande DB2) 243
- sources de données Excel 66
 - ajout d'Excel à un système fédéré 70
 - autorisations pour l'accès aux fichiers 78
- sources de données Excel (*suite*)
 - chemin du répertoire par défaut 72
 - conditions requises 70
 - définition de serveur, enregistrement 73
 - description 66
 - encapsuleur, enregistrement 71
 - exemple de scénario utilisateur 76
 - exemples de requêtes 75
 - instruction CREATE NICKNAME, exemples 74
 - instruction CREATE SERVER, exemples 73
 - instruction CREATE WRAPPER, exemples 71
 - noms de bibliothèque de l'encapsuleur 72
 - pseudonymes, enregistrement 73
- sources de données IBM InfoSphere Classic Federation Server for z/OS
 - configuration de l'accès 138
- sources de données Informix
 - fichiers de bibliothèque 4
 - mappages de types de données aval par défaut 380, 395
 - variables d'environnement 2
- Sources de données JDBC
 - alias 102
 - configuration de l'accès 92
 - définition des mappages utilisateur 100
 - enregistrement des définitions de serveur 95
 - exemples avec CREATE NICKNAME 103
 - exemples avec CREATE WRAPPER 96
 - fichiers de bibliothèque de l'encapsuleur 96
 - mappages de types de données aval par défaut 382
 - mappages de types de données aval par défaut Unicode 400
 - mappages de types de données inversés par défaut 395
 - mappages de types de données inversés par défaut Unicode 400
 - test de l'accès aux 101
 - variables d'environnement 2
- sources de données Microsoft SQL
 - définition des mappages utilisateur 113
- sources de données Microsoft SQL Server
 - accès 105
- sources de données NET8
 - mappages de types de données aval par défaut Unicode 402
 - mappages de types de données inversés par défaut Unicode 402
- sources de données non relationnelles
 - types de données pris en charge 404
- sources de données Oracle
 - identification et résolution des incidents de connectivité 168
- Sources de données Oracle
 - conversion de pages de codes 161
 - variables d'environnement 159
- Sources de données Oracle NET8
 - mappages de types de données aval par défaut 386
 - mappages de types de données inversés par défaut 398
- sources de données XML 262
 - ajout à un système fédéré 266
 - alias 276
 - blocs de données 274
 - chemin du répertoire par défaut 268
 - conditions requises 266
 - CREATE WRAPPER (instruction) 268
 - définition de serveur, enregistrement 269
 - encapsuleur, enregistrement 267
 - espaces de nom 277
 - exemples avec l'instruction CREATE NICKNAME 280
 - exemples avec l'instruction CREATE SERVER 270
 - exemples avec l'instruction CREATE VIEW 285
 - expressions XPath 274
 - fonction de modèle de coût 276
 - généralités 263
 - noms de bibliothèque de l'encapsuleur 268
 - optimisation de requête 276
 - optimisation de requête, fonction de modèle de coût 287
 - optimisation de requête, présentation 284
 - optimisation de requête, vues fédérées 285
 - options de pseudonyme
 - analyse syntaxique de documents XML 274
 - espaces de nom 277
 - options de pseudonyme, optimisation de requête 276
 - présentation, pseudonymes 274
 - pseudonymes, associations de données 274
 - pseudonymes, enregistrement 279
 - requêtes, exemples 288
 - serveur proxy, exemples d'encapsuleur 268
 - serveur proxy, exemples de définition de serveur 270
 - serveurs proxy 273
- spécification des valeurs
 - options 15
- SQL10013N
 - identification et résolution des incidents 201
- SQL1097N
 - identification et résolution des incidents 202
- SQL1822N
 - identification et résolution des incidents 201
- statistiques
 - pour les sources de données BioRS 52
- support de langue nationale
 - base de données fédérée 8

Sybase
 fichiers de bibliothèque 6
 fonctions fédérées prises en charge 291
 mappages de types de données aval par défaut 377
 mappages de types de données inversés par défaut 391
 options d'encapsuleur 341
 options de colonne 341
 options de mappage utilisateur 341
 options de serveur 341
 script djxlinkSybase 6
 Sybase, sources de données
 chemin du répertoire par défaut de l'encapsuleur 194
 configuration de l'accès 186
 configuration des variables d'environnement 188
 configure du fichier de configuration client 191, 192
 création de mappages utilisateur 197
 enregistrement 193
 enregistrement de pseudonymes 199
 enregistrement des définitions de serveur 194
 exemples avec CREATE NICKNAME 200
 exemples avec CREATE SERVER 195
 exemples avec CREATE USER MAPPING 197
 fichiers de bibliothèque 4
 identification et résolution des incidents des configurations de l'encapsuleur 201, 202
 instruction CREATE USER MAPPING 197
 mappages de types de données aval par défaut 386
 mappages de types de données aval par défaut Unicode 403
 mappages de types de données inversés par défaut 398
 mappages de types de données inversés par défaut Unicode 404
 noms de fichier de bibliothèque d'encapsuleur 194
 test des connexions au serveur Sybase 198
 variables d'environnement 2, 190
 synonymes Informix
 enregistrement de pseudonymes 91

T

tables de mémoire cache
 sources de données prises en charge 291
 tables de requêtes matérialisées
 sources de données prises en charge 291
 tables Informix
 enregistrement de pseudonymes 91
 Teradata

exemple avec l'instruction CREATE NICKNAME 229
 fichiers de bibliothèque 6

Teradata (*suite*)
 fonctions fédérées prises en charge 291
 instruction CREATE SERVER 224
 mappages de types de données aval par défaut 377
 mappages de types de données inversés par défaut 391
 noms de bibliothèque de l'encapsuleur 222
 options d'encapsuleur 347
 options de colonne 347
 options de mappage utilisateur 347
 options de serveur 347
 script djxlinkTeradata 6
 Teradata, sources de données
 alias 229
 configuration de l'accès 215
 définition de variables d'environnement 217
 définition des mappages utilisateur 225
 désactivation de la consignation d'accès 232
 enregistrement 221
 enregistrement de pseudonymes 228
 enregistrement des définitions de serveur 223
 exemples avec CREATE USER MAPPING 225
 fichiers de bibliothèque 4
 identification et résolution des incidents avec l'instruction CREATE NICKNAME 230
 identification et résolution des incidents avec les jeux de caractères 221, 231
 identification et résolution des incidents avec les pseudonymes 231
 identification et résolution des incidents de connectivité 226
 mappages de types de données aval par défaut 388, 399
 optimisation 232
 test de la configuration client 216
 test de la connexion 216
 test de la connexion au serveur fédéré 226
 variables d'environnement 2, 218
 vérification de la liaison d'exécution 216
 tolérance aux erreurs
 sources de données prises en charge 291
 traces ODBC 117
 types de données
 pour les sources de données non relationnelles 404
 types de serveur
 types fédérés valides 375

U

Unicode 9, 11, 12
 sources de données prises en charge 291

utilitaire d'optimisation ODBC
 création de tables de test 133, 151
 exemples avec la commande db2fedsrvrfg 132, 150
 optimisation des encapsuleurs ODBC 131, 149
 syntaxe de la commande db2fedsrvrfg 132, 150

V

validation en deux phases
 sources de données prises en charge 291
 variable de registre DB2_DJ_INI 2
 variables d'environnement 2
 configuration des variables d'environnement Sybase 188
 configuration des variables Microsoft SQL Server 107
 définition 2
 définition de variables Oracle 158
 définition des variables d'environnement Teradata 217
 encapsuleurs Informix 80
 identification et résolution des incidents avec la variable d'environnement Sybase 201
 Informix 81
 Microsoft SQL Server, sources de données 108
 requis 2
 restrictions 4
 sources de données 2
 Sources de données Oracle 159
 Sybase, sources de données 190
 Teradata, sources de données 218
 vues de catalogue SYSCAT 369
 vues de catalogue SYSSTAT 369
 vues fédérées
 Services Web 255
 vues Informix
 enregistrement de pseudonymes 91

W

Web Services
 Accès à un en-tête SOAP 252

X

XML
 options d'encapsuleur 362
 options de colonne 362
 options de mappage utilisateur 362
 options de pseudonyme 362
 options de serveur 362
 types de données, pris en charge 404
 XPath
 utilisation avec les pseudonymes 254



SC19-1034-02



Spine information:

IBM InfoSphere Federation Server

Version 9.7

IBM InfoSphere Federation Server - Guide de configuration pour des sources de données fédérées

