



Guía de configuración para orígenes de datos federados



Guía de configuración para orígenes de datos federados

Nota

Antes de utilizar esta información y el producto al que da soporte, lea la información de la sección "Avisos" en la página 399.

Contenido

Capítulo 1. Configuración del servidor federado 1

Emisión de mandatos en un sistema federado	1
Verificación de la configuración del servidor federado	1
Habilitación del servidor federado para acceder a orígenes de datos	1
Establecimiento de variables de entorno necesarias para clientes de origen de datos	2
Verificar que los archivos de biblioteca están enlazados con el software cliente de origen de datos	4
Gestión de la clave y la política de licencia del producto	6
Creación de una base de datos federada	7
Juegos de códigos, secuencias de clasificación y soporte de idioma nacional	7
Configuración del servidor federado para acceder a orígenes de datos	12
Distinción entre mayúsculas y minúsculas y uso correcto de las comillas	15
Configuración de varios servidores federados para acceder a orígenes de datos	18

Capítulo 2. Configurar orígenes de datos 21

Configuración del acceso a orígenes de datos BioRS	21
Derivador BioRS	21
Adición de orígenes de datos BioRS a un servidor federado	22
Funciones personalizadas y consultas BioRS	38
Optimizar el rendimiento del derivador de BioRS	50
Configuración del acceso a orígenes de datos DB2	55
Catalogación de una entrada de nodo de DB2	55
Catalogación de la base de datos de DB2 remota	56
Registro del derivador de DB2	56
Registro de definiciones de servidor para orígenes de datos DB2	58
Creación de correlaciones de usuario para orígenes de datos DB2	60
Prueba de la conexión al servidor de orígenes de datos DB2	61
Registro de apodos para vistas y tablas de DB2	62
Configuración del acceso a orígenes de datos Excel	63
Derivador Excel	63
Adición de orígenes de datos Excel a un servidor federado	68
Configuración del acceso a orígenes de datos Informix	76
Configuración y prueba del archivo de configuración del cliente Informix	76
Establecimiento de variables de entorno de Informix	78
Registro del derivador de Informix	81
Registro de las definiciones de servidor para un origen de datos Informix	83

Creación de las correlaciones de usuario para un origen de datos Informix	85
Prueba de la conexión al servidor Informix	87
Registro de apodos para tablas, vistas y sinónimos de Informix	89
Configuración del acceso a orígenes de datos JDBC	90
Preparación del servidor federado para acceder a orígenes de datos a través de JDBC	91
Registro del derivador de JDBC	92
Registro de definiciones de servidor para orígenes de datos JDBC	94
Creación de correlaciones de usuario para orígenes de datos JDBC	97
Prueba de las conexiones a los servidores de orígenes de datos JDBC	98
Registro de apodos para vistas y tablas de orígenes de datos JDBC	99
Configuración del acceso a orígenes de datos Microsoft SQL Server	101
Preparación del servidor federado para acceder a orígenes de datos Microsoft SQL Server (Windows)	101
Preparación del servidor federado para acceder a orígenes de datos de Microsoft SQL Server (Linux, UNIX)	102
Establecimiento de variables de entorno de Microsoft SQL Server	103
Registro del derivador de Microsoft SQL Server	105
Registro de las definiciones de servidor para un origen de datos Microsoft SQL Server	107
Creación de las correlaciones de usuario para un origen de datos Microsoft SQL Server	109
Prueba de la conexión con el servidor remoto Microsoft SQL Server	110
Registro de apodos para vistas y tablas de Microsoft SQL Server	112
Utilización de la información de rastreo de ODBC para resolver los problemas de conexión a orígenes de datos Microsoft SQL Server	113
Configuración del acceso a orígenes de datos ODBC	114
Preparación del servidor federado para acceder a orígenes de datos a través de ODBC (Windows)	116
Preparación del servidor federado para acceder a orígenes de datos a través de ODBC (Linux, UNIX)	116
Registro del derivador de ODBC	117
Registro de las definiciones del servidor para un origen de datos ODBC	120
Creación de una correlación de usuario para un origen de datos ODBC	122
Prueba de la conexión al servidor de orígenes de datos ODBC	124
Registro de apodos para vistas y tablas de orígenes de datos ODBC	125

Optimización del rendimiento del derivador de ODBC con el programa de utilidad de ajuste de ODBC (db2fedsvrctg).	126	Registro del derivador de Sybase	188
Acceso a datos Excel mediante el derivador de ODBC.	130	Registro de las definiciones de servidor para un origen de datos Sybase	189
Configuración del acceso de ODBC a orígenes de datos IBM InfoSphere Classic Federation Server for z/OS	134	Creación de las correlaciones de usuarios para un origen de datos Sybase	192
Registro del derivador de ODBC	135	Prueba de la conexión al servidor Sybase	193
Registro de las definiciones del servidor para un origen de datos ODBC	138	Registro de apodos para vistas y tablas de Sybase.	194
Creación de una correlación de usuario para un origen de datos ODBC	140	Resolución de problemas de la configuración del derivador de Sybase	195
Prueba de la conexión al servidor de orígenes de datos ODBC.	142	Configuración del acceso a orígenes de datos de archivos con estructura de tabla	197
Registro de apodos para vistas y tablas de orígenes de datos ODBC	143	Visión general de los archivos estructurados en tablas	197
Optimización del rendimiento del derivador de ODBC con el programa de utilidad de ajuste de ODBC (db2fedsvrctg).	144	Atributos de archivos estructurados en tablas	198
Configuración del acceso a orígenes de datos OLE DB	148	Derivador de archivos estructurados en tablas	198
Registro del derivador de OLE DB	149	Adición de orígenes de datos de archivos con estructura de tabla a un servidor federado.	199
Registro de las definiciones de servidor para un origen de datos OLE DB.	150	Modelo de control de accesos de archivo para el derivador de archivos con estructura de tabla	208
Creación de las correlaciones de usuarios para un origen de datos OLE DB	151	Directrices para optimizar el rendimiento de las consultas para el derivador de archivos con estructura de tabla.	209
Configuración del acceso a orígenes de datos Oracle.	153	Configuración del acceso a orígenes de datos Teradata	209
Establecimiento de las variables de entorno de Oracle	153	Prueba de la conexión al servidor Teradata	210
Configuración y prueba del archivo de configuración del cliente Oracle	157	Comprobación de que la biblioteca de Teradata esté habilitada para los enlaces de tiempo de ejecución (AIX).	210
Registro del derivador de Oracle	158	Establecimiento de las variables de entorno de Teradata	211
Registro de las definiciones de servidor para un origen de datos Oracle	160	Registro del derivador de Teradata	215
Creación de las correlaciones de usuarios para un origen de datos Oracle	161	Registro de las definiciones de servidor para un origen de datos Teradata	217
Prueba de la conexión al servidor Oracle	163	Creación de la correlación de usuarios para un origen de datos Teradata	219
Registro de apodos para vistas y tablas de Oracle.	164	Prueba de la conexión al servidor Teradata	220
Configuración del acceso a los scripts como orígenes de datos	166	Registro de apodos para vistas y tablas de Teradata	222
Visión general del derivador de scripts	166	Resolución de problemas de la configuración de orígenes de datos Teradata	223
Adición de scripts como orígenes de datos a un sistema federado	168	Configuración del acceso a los orígenes de datos de servicios Web	226
Registro de apodos para los scripts (línea de mandatos de DB2).	176	Servicios web y derivador de servicios web	226
Consultas SQL con el derivador de scripts.	179	Registro del derivador de servicios web	233
Optimización del rendimiento del derivador de scripts.	181	Registro de la definición de servidor para orígenes de datos de servicios web	234
Configuración del acceso a orígenes de datos Sybase.	181	Habilitación de la seguridad mediante el derivador de servicios web.	236
Soporte de derivadores de Sybase para Adaptive Server Enterprise (ASE)	182	Registro de apodos para los orígenes de datos de servicios web	237
Establecimiento de las variables de entorno de Sybase.	183	Restricciones de consultas de los derivadores de servicios web	249
Instalación y prueba del archivo de configuración del cliente Sybase (Windows)	186	Orígenes de datos de servicios web - consultas de ejemplos	252
Configuración y prueba del archivo de configuración del cliente Sybase (UNIX)	187	Configuración del acceso a orígenes de datos XML	255
		Derivador XML	255
		Adición de XML a un sistema federado	258
		Consultas para orígenes de datos XML.	275

Capítulo 3. Soporte de origen de datos para opciones federadas. . . . 281

Capítulo 4. Guía de consulta de opciones para orígenes de datos. . . . 283

Información de consulta sobre opciones de BioRS 283
Información de consulta sobre opciones de bases de datos de DB2 287
Información de consulta sobre opciones de Excel 294
Información de consulta sobre opciones de Informix 295
Información de consulta sobre opciones de JDBC 301
Información de consulta sobre opciones de Microsoft SQL Server. 308
Información de consulta sobre opciones de ODBC 313
Información de consulta sobre opciones de Oracle 319
Información de consulta sobre opciones de Script 324
Información de consulta sobre opciones de Sybase 330
Información de consulta sobre opciones de Teradata 335
Información de consulta sobre opciones de archivos con estructura de tabla 339
Información de consulta sobre opciones de servicios web 342
Información de consulta sobre opciones de XML 349

Capítulo 5. Vistas de la tabla de catálogo global que contienen información federada 357

Capítulo 6. Opciones de correlaciones de funciones para sistemas federados 361

Capítulo 7. Tipos de servidor válidos en las sentencias de SQL 363

Capítulo 8. Correlaciones de tipos de datos 365

Correlaciones de tipos de datos directas por omisión 365
Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 Database para Linux, UNIX y Windows 365
Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para System i 366
Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para VM y VSE 367
Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para z/OS 367
Correlaciones de tipos de datos directas por omisión para orígenes de datos Informix 368
Correlaciones de tipos de datos directas por omisión para orígenes de datos JDBC 369
Correlaciones de tipos de datos directas por omisión para orígenes de datos Microsoft SQL Server 371

Correlaciones de tipos de datos directas por omisión para orígenes de datos ODBC 372
Correlaciones de tipos de datos directas por omisión para orígenes de datos Oracle NET8 373
Correlaciones de tipos de datos directas por omisión para orígenes de datos Sybase 374
Correlaciones de tipos de datos directas por omisión para orígenes de datos Teradata 375
Correlaciones de tipos de datos directas de ejemplo 376
Correlaciones de tipos de datos inversas por omisión 378
Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 Database para Linux, UNIX y Windows 379
Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para System i 380
Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para VM y VSE 381
Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para z/OS 381
Correlaciones de tipos de datos inversas por omisión para orígenes de datos Informix 382
Correlaciones de tipos de datos inversas por omisión para orígenes de datos JDBC 383
Correlaciones de tipos de datos inversas por omisión para orígenes de datos Microsoft SQL Server 384
Correlaciones de tipos de datos inversas por omisión para orígenes de datos ODBC 384
Correlaciones de tipos de datos inversas por omisión para orígenes de datos Oracle NET8 385
Correlaciones de tipos de datos inversas por omisión para orígenes de datos Sybase 386
Correlaciones de tipos de datos inversas por omisión para orígenes de datos Teradata 387
Correlaciones de tipos de datos por omisión Unicode 387
Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos JDBC 387
Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos JDBC 388
Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos Microsoft SQL Server. 388
Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos Microsoft SQL Server. 389
Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos NET8 389
Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos NET8 389
Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos ODBC 390
Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos ODBC 390
Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos Sybase 391

Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos Sybase . 391
Tipos de datos admitidos para orígenes de datos no relacionales 392

Documentación del producto 395

Documentación accesible 397

Avisos 399
Marcas registradas. 401

Índice. 403

Capítulo 1. Configuración del servidor federado

Antes de configurar orígenes de datos, configure el servidor federado y verifique que esté configurado debidamente.

Procedimiento

Para configurar el servidor federado, siga realice estas tareas:

Emisión de mandatos en un sistema federado

Si es la primera vez que utiliza Federation, repase cómo acceder al Centro de control de DB2 y cómo iniciar la línea de mandatos de DB2. Utilizará estas interfaces para ejecutar muchas tareas en un sistema federado.

El Centro de control proporciona cuadros de diálogo y asistentes para guiarle en la ejecución de una tarea, sin que sea necesario que conozca los mandatos de DB2 necesarios para realizar la tarea. Si es la primera vez que utiliza DB2 y Federation, puede ser conveniente utilizar el Centro de control, que es intuitivo y de fácil manejo. Para utilizar la línea de mandatos de DB2, escriba los mandatos de DB2 necesarios para ejecutar una tarea. Si ya conoce DB2, puede que prefiera utilizar la línea de mandatos. Cuando se pueden utilizar ambas interfaces para realizar una tarea, la documentación describe los pasos específicos que se deben seguir y los mandatos que se deben entrar.

Para acceder a una interfaz, realice lo siguiente:

- Para iniciar la línea de mandatos de DB2, escriba db2 en el indicador del sistema operativo. Puede entonces entrar mandatos, uno después de otro. Cuando termine de utilizar la línea de mandatos de DB2, escriba quit.
- Para utilizar el Centro de control de DB2, expanda la carpeta **Objetos federados** para mostrar la lista de objetos con los que puede trabajar. Luego seleccione un objeto y utilice los paneles y asistentes para realizar la tarea.

Verificación de la configuración del servidor federado

Cuando instala Federation, el software intenta configurar automáticamente el servidor federado. Para evitar problemas cuando configure orígenes de datos, compruebe que el servidor federado esté configurado debidamente.

Para verificar la configuración del servidor federado, realice estas tareas:

Habilitación del servidor federado para acceder a orígenes de datos

El parámetro FEDERATED debe estar establecido en YES para permitir que el servidor federado acceda a los orígenes de datos.

El programa de instalación intenta establecer el parámetro FEDERATED, pero es mejor verificar que esté establecido debidamente en YES. Si el parámetro FEDERATED no está establecido debidamente, todos los intentos para acceder a orígenes de datos fallarán y se devolverá el mensaje SQL20076, código de razón 1.

Procedimiento

Para verificar que el parámetro FEDERATED está establecido debidamente y habilitarlo si es necesario:

1. Desde la línea de mandatos de DB2, emita este mandato para mostrar los parámetros y sus valores actuales:
GET DATABASE MANAGER CONFIGURATION
2. Compruebe el valor del parámetro MAX_CONNECTIONS para determinar si el concentrador está activo:
 - Si MAX_CONNECTIONS es igual a MAX_COORDAGENTS, el concentrador está inactivo.
 - Si MAX_CONNECTIONS es mayor que MAX_COORDAGENTS, el concentrador está activo.

El parámetro MAX_CONNECTIONS no puede estar activo cuando el parámetro FEDERATED está establecido en YES. Si el concentrador está activo, cambie el valor de MAX_CONNECTIONS para que se igual a MAX_COORDAGENTS.

3. Compruebe el valor del parámetro FEDERATED. Si el parámetro FEDERATED está establecido en NO, cambie el valor a YES. Emita este mandato para cambiar el valor:

```
UPDATE DATABASE MANAGER CONFIGURATION USING FEDERATED YES
```

Establecimiento de variables de entorno necesarias para clientes de origen de datos

Compruebe que las variables de entorno necesarias estén establecidas en el archivo `db2dj.ini`. Esta tarea es necesaria para los orígenes de datos Informix, Microsoft® SQL Server, Oracle, Sybase y Teradata, así como para las aplicaciones que utilizan los derivadores JDBC y ODBC.

Antes de empezar

El administrador del sistema realiza esta tarea.

Si instala el software cliente antes de instalar el servidor federado, el programa de instalación establece automáticamente las variables de entorno necesarias para los orígenes de datos que el usuario seleccione. Si instala el software cliente después de instalar el servidor federado, actualizar a una nueva versión del software cliente o migrar a un nuevo hardware, el usuario debe establecer manualmente las variables de entorno necesarias.

La tabla siguiente lista las variables de entorno necesarias para cada origen de datos.

Tabla 1. Variables de entorno necesarias para cada origen de datos

Origen de datos	Variable necesaria	Valor
Informix	INFORMIXDIR INFORMIXSERVER	Para INFORMIXDIR, el directorio que contiene el software del cliente Informix. Para INFORMIXSERVER, el nombre del servidor Informix predeterminado.

Tabla 1. Variables de entorno necesarias para cada origen de datos (continuación)

Origen de datos	Variable necesaria	Valor
JDBC	Varía en función del controlador JDBC instalado. Si no especifica el paquete del controlador JDBC en el parámetro del servidor DRIVER_PACKAGE de la sentencia CREATE SERVER, debe especificar el paquete del controlador JDBC en la variable de entorno "CLASSPATH" del sistema. Consulte la documentación del controlador JDBC para obtener información específica acerca del establecimiento de variables de entorno del sistema.	Nombre de archivo y directorio de la biblioteca del controlador JDBC.
Microsoft SQL Server	DJX_ODBC_LIBRARY_PATH	Directorio donde reside la biblioteca de ODBC.
ODBC	Varía según la aplicación ODBC.	Varía según la aplicación ODBC.
Oracle	ORACLE_HOME	Directorio donde reside el software cliente de Oracle.
Sybase	SYBASE, SYBASE_OCS	Para SYBASE, el directorio que contiene el software del cliente de Sybase. Para SYBASE_OCS, el directorio, versión y release del software de cliente.
Teradata	COPYLIB	Directorio donde reside el software cliente de Teradata.

Procedimiento

Para establecer las variables de entorno manualmente:

1. Abra el archivo db2dj.ini.

El archivo db2dj.ini se añade al servidor federado durante la instalación. La ubicación por omisión de este archivo es la ubicación especificada por la variable DB2_DJ_INI del registro de la base de datos. Si la variable del registro DB2_DJ_INI no está definida, el archivo reside en estas ubicaciones:

- En UNIX®, el archivo reside en *dir_inic_instancia*/sql1lib/cfg/db2dj.ini, donde *dir_inic_instancia* es el directorio inicial del propietario de la instancia.
- En Microsoft Windows®, el archivo reside en %DB2PATH%\cfg\db2dj.ini, donde %DB2PATH% es el directorio donde está instalado el sistema de bases de datos DB2. Por ejemplo, C:\Archivos de programa\IBM\sql1lib.

2. Utilice la sintaxis *nombre_variable=valor_variable* para establecer el valor de la variable de entorno. Si *valor_variable* es un nombre de archivo o directorio, debe especificar la vía de acceso totalmente calificada. Por ejemplo, para establecer el directorio informix del directorio inicial /home/user1 como valor de la variable INFORMIXDIR, incluya esta entrada en el archivo db2dj.ini:

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

3. Desde la línea de mandatos de DB2, emita estos mandatos:

```
db2stop  
db2start
```

Estos mandatos detienen y reinician la instancia de la base de datos para asegurar que las variables de entorno sean efectivas.

4. Si está configurando en un sistema de varias particiones, copie el archivo `db2dj.ini` en cada partición.

Restricciones para el archivo `db2dj.ini`

El archivo `db2dj.ini` contiene las variables de entorno obligatorias y opcionales para un origen de datos.

Cuando modifique el archivo `db2dj.ini`, tenga en cuenta estas restricciones.

Para especificar cada variable, utilice el formato *nombre_variable=valor_variable*

donde

- *nombre_variable* es el nombre de la variable de entorno.
- *valor_variable* es el valor.

Para especificar un nombre de archivo o nombre de directorio como nombre de la variable, especifique el nombre totalmente calificado. El nombre no puede contener metacaracteres de nombre de archivo, tales como `~` (tilde) ni variables de entorno, tales como `$HOME`. Por ejemplo, para establecer el directorio `informix` y el directorio inicial llamado `/home/user1` como valor de la variable de entorno `INFORMIXDIR`, especifique este valor en el archivo `db2dj.ini`:

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

Tenga en cuenta estos límites:

- La longitud máxima para el nombre de una variable de entorno es 255 bytes
- La longitud máxima para el valor de una variable de entorno es 756 bytes
- La longitud máxima de una línea en el archivo `db2dj.ini` es 1021 bytes; los datos situados más allá de esta longitud no se tienen en cuenta.

Verificar que los archivos de biblioteca están enlazados con el software cliente de origen de datos

Para obtener los mejores resultados, verifique siempre que los archivos de biblioteca estén enlazados con el software cliente de origen de datos antes de utilizar el mandato `CREATE WRAPPER` para registrar el derivador. Esta tarea es aplicable solamente cuando el servidor federado utiliza UNIX y el cliente de origen de datos es Informix, Microsoft SQL Server, Oracle, Sybase o Teradata.

Cuando instala Federation, puede elegir configurar derivadores automáticamente. Si instala el software cliente de origen de datos en el servidor federado antes de instalar Federation en el servidor y selecciona configurar el derivador automáticamente durante el programa de instalación, los archivos de biblioteca del derivador se enlazan con el correspondiente software cliente de origen de datos. Si instala el software cliente de origen de datos después de instalar Federation, por ejemplo para actualizar a una nueva versión del cliente, debe enlazar manualmente los archivos de biblioteca del derivador con el software cliente de origen de datos.

Estas son las causas habituales de un error de enlace:

- Falta una variable de entorno necesaria durante la instalación.

- El cliente de origen de datos no está instalado en el servidor federado.
- El cliente de origen de datos que está instalado en el servidor federado no está a un nivel válido.

Procedimiento

Para verificar que los archivos de biblioteca están enlazados debidamente:

1. Abra el archivo de mensajes del origen de datos. El archivo de mensajes reside en el directorio donde está instalado el servidor federado, dentro del subdirectorio lib32 o lib64. Esta tabla lista los orígenes de datos, archivos de mensajes y archivos de biblioteca.

Tabla 2. Orígenes de datos, archivos de mensajes y archivos de biblioteca

Origen de datos	Archivo de mensajes	Archivos de biblioteca
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. Evalúe el contenido del archivo de mensajes:
 - Si el enlace se ha realizado satisfactoriamente, el archivo de mensajes aparece en el directorio donde está instalado Federation, y el archivo de mensajes muestra un mensaje de ejecución satisfactoria.
 - Si el enlace ha fallado, el archivo de mensajes muestra un mensaje de error. Si ni el archivo de biblioteca ni el archivo de mensajes aparece en el directorio donde está instalado Federation, debe enlazar manualmente el archivo de biblioteca con el software cliente de origen de datos.

Enlace manual de archivos de biblioteca con software cliente de origen de datos

Si los archivos de biblioteca no se muestran en la vía de directorios, debe ejecutar un script para enlazarlos al software cliente del origen de datos.

Antes de empezar

- El software cliente del origen de datos debe estar instalado y configurado en el servidor federado.

- Por omisión, los scripts emiten todos los mensajes en el idioma inglés. Para emitir mensajes en otro idioma, debe configurar una base de datos federada para que utilice una página de códigos para un idioma distinto del inglés.
- Debe definir las variables de entorno necesarias del origen de datos.

Tabla 3. Orígenes de datos, scripts y variables de entorno necesarias

Origen de datos	Script	Variable de entorno necesaria
Informix	djxlinkInformix	INFORMIXDIR
Microsoft SQL Server	djxlinkMssql	DJX_ODBC_LIBRARY_PATH
Oracle	djxlinkOracle	ORACLE_HOME
Sybase	djxlinkSybase	SYBASE, SYBASE_OCS
Teradata	djxlinkTeradata	COPLIB

Procedimiento

Para enlazar los archivos de biblioteca con el software cliente del origen de datos:

1. Abra un indicador de mandatos de UNIX y emita este mandato:


```
cd /opt/IBM/db2/V9.5/bin
```
2. Desde la línea de mandatos de DB2, emita estos mandatos:


```
db2 disconnect all
db2stop
```
3. Ejecute el script para cada origen de datos al que desee acceder. Consulte la tabla mostrada más arriba para conocer los nombres de los scripts.
4. Si realiza la instalación en un entorno de varias particiones, repita los pasos 1 a 3 en cada instancia de la base de datos.
5. Para cada instancia de la base de datos, vaya al directorio *dir_instal/instance*, donde *dir_instal* es el directorio de instalación de DB2, y emita este mandato:


```
./db2iupdt nombre_instancia
```

 Este mandato actualiza la configuración de la instancia y le proporciona acceso a los orígenes de datos.
6. Para cada instancia de la base de datos, visualice los permisos de archivo para los archivos de biblioteca. Compruebe que el propietario de la instancia de base de datos tiene permiso para leer y ejecutar los archivos.

Gestión de la clave y la política de licencia del producto

Cada sistema en el que se instala IBM® InfoSphere Federation Server debe almacenar un archivo de licencia que especifique la clave de licencia del producto. La política de licencia controla y supervisa el número de usuarios que tienen permiso para conectar con el servidor federado.

La clave de licencia reside en el directorio *license* del software de instalación. Si la clave de licencia no está registrada, debe registrarla manualmente.

1. Localice su archivo de licencia en el directorio *license* del software de instalación del producto:

isfs.lic

Archivo de la licencia completa del producto

isfs_t.lic

Archivo de la licencia para prueba y compra del producto

isfs_d.lic

Archivo de licencia de la edición de desarrollador

2. Si el archivo de licencia no se encuentra en el directorio, debe registrar manualmente una clave de licencia; consulte la información sobre el .
3. Especifique la política de licencia para controlar y supervisar los usuarios que tienen autorización para conectarse al servidor federado.

Creación de una base de datos federada

Antes de configurar el servidor federado para acceder a orígenes de datos, debe crear una base de datos federada.

Antes de empezar

- Debe tener autorización SYSADM o SYSCTRL para crear una base de datos.
- Federation debe estar instalado en el servidor que actuará como servidor federado.

Si la instancia de base de datos utiliza una configuración de varias particiones, todas las particiones listadas en el archivo db2nodes.cfg se ven afectadas cuando crea la base de datos. La partición de base de datos desde la que emite el mandato CREATE DATABASE pasa a ser la partición de catálogo de la nueva base de datos.

Procedimiento

Para crear la base de datos federada:

1. Determine el juego de códigos y secuencia de clasificación que desee especificar al crear la base de datos federada.
2. Desde la línea de mandatos de DB2, emita el mandato CREATE DATABASE.

Por ejemplo, para crear una base de datos denominada federated cuyo juego de códigos es ISO8859-15 y su territorio es Brasil, emita este mandato:

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

Juegos de códigos, secuencias de clasificación y soporte de idioma nacional

Cuando crea una base de datos federada, especifica el juego de códigos, el territorio y la secuencia de clasificación. Esta información determina el idioma en el que se almacenan los datos y la secuencia de clasificación de los datos.

Un *juego de códigos* es un conjunto de patrones exclusivos de bits que están asociados a los caracteres de un determinado idioma natural. Los productos IBM utilizan el término *página de códigos* como sinónimo de juego de códigos. Un *territorio* identifica un entorno local y especifica información específica del área geográfica correspondiente al juego de códigos especificado. Si no especifica estas opciones, la base de datos utiliza el idioma y secuencia de clasificación del cliente DB2 utilizado para crear la base de datos.

Antes de crear la base de datos federada, determine qué valores debe especificar para el juego de códigos y el territorio. Después de crear la base de datos, no puede cambiar estos valores. Para elegir un juego de códigos para la base de datos federada, evalúe el juego de códigos especificado por el origen de datos remoto al que accederá la base de datos federada. Seleccione un juego de códigos para la

base de datos federada que corresponda al juego de códigos utilizado por los orígenes de datos remotos. Si la base de datos federada accederá a varios orígenes de datos, evalúe los juegos de códigos especificados por todos los orígenes de datos remotos. Si los orígenes de datos utilizan juegos de códigos diferentes o incompatibles, especifique Unicode como juego de códigos para el servidor federado.

Para muchos orígenes de datos, la primera vez que un derivador se conecta a un origen de datos, el derivador realiza estas tareas:

1. Determina la página de códigos y el territorio de la base de datos federada.
2. Asocia el juego de códigos y el territorio a un entorno local de cliente de origen de datos, si el origen de datos admite uno.
3. Establece una variable de entorno, invoca una API de origen de datos para indicar al origen de datos cuál es el entorno local del cliente, o realiza la preparación para realizar una conversión de juego de códigos.

La conversión de página de códigos comprende la conversión de datos de tipo carácter entre la página de códigos de la base de datos del origen de datos y la página de códigos de la base de datos federada. Algunos orígenes de datos realizan la conversión de página de códigos. Para algunos orígenes de datos que no realizan la conversión de página de códigos, el derivador realiza la conversión. Por ejemplo, si la base de datos federada utiliza la página de códigos 819 y el territorio Estados Unidos, el entorno local equivalente del cliente Oracle es `American_America.WE8ISO8859P1`. El derivador Oracle establece automáticamente la variable de entorno `NLS_LANG` en el valor del entorno local del cliente Oracle. Luego, cuando los datos se envían desde la base de datos Oracle al derivador, la base de datos Oracle convierte los datos desde el juego de códigos `American_America.WE8ISO8859P1` a la página de códigos 819. Cuando los datos se envían desde el derivador a la base de datos Oracle, el servidor o cliente Oracle convierte los datos desde la página de códigos 819 al juego de códigos utilizado por la base de datos Oracle.

La secuencia de clasificación está relacionada con el idioma utilizado por el servidor federado y el servidor del origen de datos. Para especificar la secuencia de clasificación para la base de datos federada, incluya la opción `COLLATE USING` en el mandato `CREATE DATABASE`. Si la base de datos federada y el origen de datos utilizan la misma secuencia de clasificación, establezca la opción de servidor `COLLATING_SEQUENCE` en 'Y' cuando emita la sentencia `CREATE SERVER`. La secuencia de clasificación que especifique para la base de datos federada tiene efectos cuando se realizan consultas en las que intervienen clasificaciones o comparaciones de caracteres. De forma predeterminada, la base de datos federada utiliza una secuencia de clasificación con distinción entre mayúsculas y minúsculas. Pero algunos orígenes de datos utilizan de forma predeterminada una secuencia de clasificación sin distinción entre mayúsculas y minúsculas. Un origen de datos puede también ofrecer la posibilidad de personalizar la secuencia de clasificación o puede proporcionar opciones para establecer la página de códigos predeterminada. Si la secuencia de clasificación de la base de datos federada y el origen de datos son diferentes, los resultados devueltos por una consulta pueden ser distintos de los esperados. Por ejemplo, si la consulta comprende una clasificación de caracteres, se devuelven los resultados correctos, pero no en el orden esperado. Si la consulta comprende comparaciones de caracteres, puede que se obtengan resultados incorrectos.

Cuando se realizan operaciones basadas en caracteres, ello afecta al rendimiento de las consultas. Cuando las secuencias de clasificación son diferentes, el servidor

federado realiza clasificaciones y comparaciones de caracteres localmente para asegurar que se devuelvan resultados ordenados de forma coherente. Para obtener los mejores resultados, establezca la secuencia de clasificación de la base de datos federada en la misma secuencia utilizada por el origen de datos. Entonces, cuando sea posible, el optimizador de consultas transfiere las operaciones basadas en caracteres al origen de datos, por lo que el origen de datos, y no la base de datos federada, realiza las operaciones.

Configuración de Unicode para sistemas federados

Puede ejecutar derivadores relacionales y no relacionales y funciones definidas por el usuario en una base de datos en la página de códigos Unicode (UTF-8). Una base de datos en la página de códigos Unicode proporciona un entorno de servidor federado independiente de la plataforma.

Soporte Unicode para sistemas federados

Todos los derivadores relacionales y no relacionales y las funciones definidas por el usuario pueden ejecutarse en una base de datos con la página de códigos Unicode (UTF-8).

La base de datos en la página de códigos Unicode proporciona entornos de servidor federado que son independientes de la plataforma. La base de datos puede manipular datos almacenados en distintas páginas de códigos y en distintos orígenes de datos.

En Figura 1 en la página 10 una empresa tiene sucursales en varios países. Cada sucursal almacena datos de cliente con sus propias bases de datos y sus propias páginas de códigos. La base de datos de Microsoft SQL Server almacena datos en la página de códigos A. La base de datos Oracle almacena datos en la página de códigos B. La página de códigos A y la página de códigos B se encuentran en distintos territorios. Para integrar los datos de los distintos territorios, la empresa puede establecer la página de código de las bases de datos federadas en Unicode. Luego la empresa puede unir las tablas para ver el número total de pedidos de compra, independientemente del territorio.

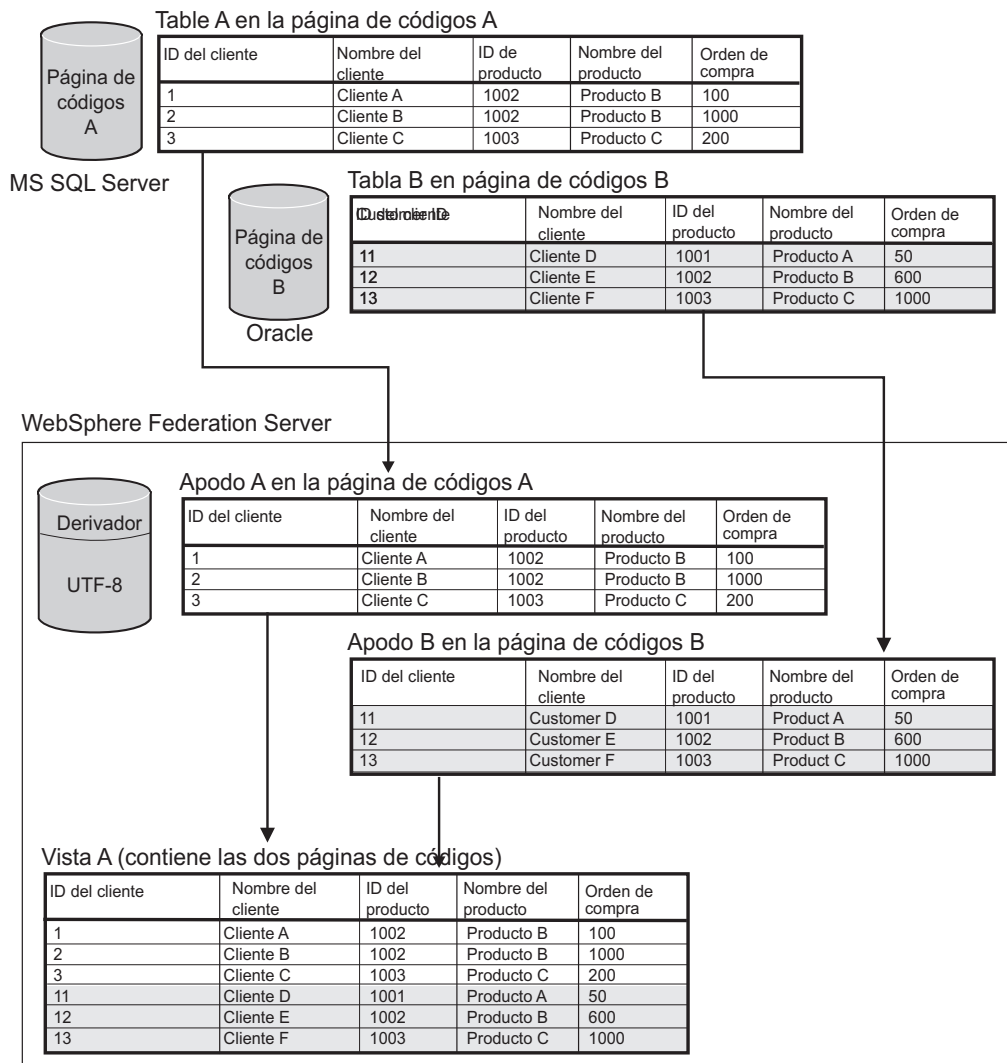


Figura 1. Ejemplo Unicode

Especificación de la página de códigos del cliente para el soporte Unicode de orígenes de datos Microsoft SQL Server y ODBC

Para garantizar una conversión correcta de la página de códigos para orígenes de datos Microsoft SQL Server y ODBC, debe especificar la página de códigos del cliente si la página de códigos difiere de la página de códigos de la base de datos federada.

Procedimiento

Para especificar la página de códigos del cliente, emita una sentencia CREATE SERVER con la opción CODEPAGE para establecer el valor del número de página de códigos del cliente. La página de códigos del cliente es la página de códigos del cliente de origen de datos.

Ejemplo: si el origen de datos es Microsoft SQL Server, el servidor federado se encuentra en Windows y el entorno local del sistema predeterminado del sistema operativo está establecido en Japonés (Shift-JIS), la opción CODEPAGE del servidor

debe establecerse en 943 (Shift-JIS) o 1202 (UTF-16LE). Para especificar la página de códigos 1202 para el origen de datos del servidor Microsoft SQL denominado FEDSERVERW, emita la sentencia siguiente:

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

Ejemplo: si el origen de datos es Microsoft SQL Server, el servidor federado se ejecuta en UNIX y el valor IANAAppCodePage del cliente DataDirect Connect es 6 (Shift-JIS), la opción CODEPAGE del servidor debe establecerse en 943 (Shift-JIS) o 1208 (UTF-8). Para especificar la página de códigos 1208 para el origen de datos Microsoft SQL Server denominado FEDSERVERU, emita la sentencia siguiente:

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

Páginas de códigos Unicode soportadas para la opción CODEPAGE del derivador MSSQL y ODBC

Los valores válidos de página de códigos son los soportados por DB2 Database for Linux®, UNIX, and Windows, además de los que se muestran en la tabla siguiente.

Tabla 4. Páginas de códigos Unicode soportadas para la opción CODEPAGE del derivador MSSQL y ODBC

Valor de la opción CODEPAGE	Descripción
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)

Especificación de la página de códigos del archivo para el soporte Unicode de orígenes de datos de archivo con estructura de tabla

Para garantizar una conversión correcta de la página de códigos para orígenes de datos de archivo con estructura de tabla, debe especificar la página de códigos del archivo si la página de códigos difiere de la página de códigos de la base de datos federada.

Restricciones

Sólo puede utilizar la opción CODEPAGE en una base de datos federada Unicode.

Acerca de esta tarea

Los valores válidos son los soportados por DB2 Database for Linux, UNIX, and Windows. El valor predeterminado es la página de códigos de la base de datos federada.

Procedimiento

Para especificar la página de códigos de un archivo con estructura de tabla, emita la sentencia CREATE NICKNAME con la opción CODEPAGE establecida en el número de página de códigos del archivo con estructura de tabla.

Ejemplo: la página de códigos de los datos de un archivo denominado DRUGDATA1.TXT es 943. Para especificar la página de códigos de un archivo con estructura de tabla como 943, emita la siguiente sentencia CREATE NICKNAME:

```
CREATE NICKNAME DRUGDATA1(Dcode Integer NOT NULL, Drug CHAR(20),
    Manufacutuer 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');
```

Especificación de la página de códigos del archivo para el soporte Unicode de orígenes de datos de archivo con estructura de tabla - ejemplo

El ejemplo siguiente muestra cómo especificar la página de códigos de un archivo con estructura de tabla.

La página de códigos de los datos de un archivo denominado DRUGDATA1.TXT es 943. Para especificar la página de códigos de un archivo con estructura de tabla como 943, emita la siguiente sentencia CREATE NICKNAME:

```
CREATE NICKNAME DRUGDATA1(Dcode Integer NOT NULL, Drug CHAR(20),
    Manufacutuer 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');
```

Errores que se producen cuando los tamaños de punto de código remoto y federado son diferentes

Cuando el tamaño del punto de código difiere entre la base de datos federada y el origen de datos remoto, es posible que se devuelvan datos truncados o que se produzcan anomalías de inserción o actualización.

Cuando se seleccionan datos del origen de datos remoto, dichos datos se truncan si la conversión de series de caracteres da como resultado un número de bytes superior al tamaño de la columna de apodo. Si los datos truncados terminan con un carácter pendiente, los bytes restantes se rellenan con blancos. También puede insertar o actualizar datos cuyo tamaño sea superior al de la columna de apodo si el tamaño de los datos convertidos es igual o inferior al tamaño de la columna remota.

Si la base de datos federada tiene un tamaño de punto de código inferior al del origen de datos remoto, la inserción o actualización de datos puede fallar. Las inserciones o actualizaciones fallarán si la conversión de series de caracteres da como resultado un número de bytes superior al tamaño de la columna del origen de datos remoto.

Ajuste el tamaño de la columna de apodo o de la columna de la tabla remota para evitar el truncamiento de datos o un error de truncamiento.

Configuración del servidor federado para acceder a orígenes de datos

Después de instalar y configurar el servidor federado y haber creado una base de datos federada, planifique y luego configure el acceso a los orígenes de datos.

El proceso para configurar el acceso a orígenes de datos es el mismo para cualquier origen de datos. Las diferencias del proceso residen en los valores determinados que aplica cuando finaliza cada tarea de configuración para cada

origen de datos. El proceso descrito aquí es genérico; para obtener información completa sobre la configuración de un origen de datos determinado, consulte la información de configuración detallada correspondiente al origen de datos.

Repasar cómo designar y especificar objetos

Durante el proceso de configuración utiliza sentencias de SQL de DB2 para registrar objetos. Antes de realizar cualquier tarea de configuración, compruebe que comprende las normas de denominación para objetos federados y comprende cómo el uso de comillas en las sentencias de SQL afecta a la distinción entre mayúsculas y minúsculas en los nombres de los objetos que especifica.

Registrar el derivador

Después de utilizar el programa de instalación de Federation para trabajar con un origen de datos, debe registrar el correspondiente derivador. Un derivador es un conjunto de archivos de biblioteca que el servidor federado utiliza para comunicar con el origen de datos y recuperar datos de él. Para cada tipo de origen de datos al que desee acceder, debe registrar un derivador. Por ejemplo, para acceder a una tabla en DB2 para Linux, UNIX y Windows, a una tabla en DB2 para iSeries y a una tabla en Teradata, debe registrar dos derivadores: el derivador DRDA para orígenes de datos DB2 y el derivador Teradata para orígenes de datos Teradata.

Como parte de la planificación, decida si desea utilizar el nombre de derivador predeterminado o asignar un nombre diferente al derivador, y repase las opciones de derivador existentes para cada origen de datos que está configurando. Cada origen de datos tiene una o más opciones de derivador que debe definir obligatoriamente.

Registrar las definiciones de servidor

Antes de poder acceder a determinados objetos de origen de datos, registre una o más definiciones de servidor. Para un origen de datos relacional, una definición de servidor representa una base de datos remota, una partición de base de datos o un nodo. Para un origen de datos no relacional, una definición de servidor a menudo está asociada a otros tipos de objetos de datos externos. Cada origen de datos tiene parámetros obligatorios y opcionales que debe especificar al registrar la definición de servidor.

Como parte de la planificación, repase las opciones de servidor existentes para el origen de datos determinado que está configurando. Cada origen de datos tiene una o más opciones de servidor que debe definir obligatoriamente.

Registrar correlaciones de usuarios

Si un origen de datos remoto requiere la autenticación del usuario y el ID de usuario y contraseña de un usuario remoto son diferentes de los utilizados por el usuario para conectar con la base de datos federada, es necesario definir una correlación de usuarios. Una correlación de usuarios es una asociación entre un ID de autorización del servidor federado y un ID de usuario y contraseña de un origen de datos. De forma predeterminada, las correlaciones de usuarios se almacenan en el catálogo del servidor federado.

Como parte de la planificación, determine si desea almacenar la información de la correlación de usuarios en un depósito externo, tal como un servidor LDAP, o en un archivo. Para utilizar un depósito externo, debe crear un plug-in que

proporcione una interfaz al servidor federado para poder interactuar con el depósito.

Actualizar estadísticas del origen de datos

Para cada origen de datos relacional al que desee acceder, utilice un mandato que sea equivalente al mandato RUNSTATS de DB2 para actualizar las estadísticas en el origen de datos remoto. Luego, cuando cree apodos, la información estadística más actual se añade al catálogo del sistema en la base de datos federada. Más tarde, cuando ejecute una consulta para el origen de datos, el optimizador de consultas utiliza esta información para determinar la forma más eficiente de ejecutar la consulta.

Después de crear apodos, las estadísticas contenidas en el origen de datos pueden cambiar. Cuando cambien las estadísticas para un origen de datos relacional, utilice el procedimiento almacenado SYSPROC.NNSTAT para actualizar la información estadística contenida en el catálogo del sistema. Cuando cambien las estadísticas para un origen de datos no relacional, utilice la herramienta proporcionada por el origen de datos no relacional, o actualice manualmente las estadísticas en la vista de catálogo SYSTAT.

Registrar apodos

Debe crear un apodo para cada objeto de origen de datos relacional al que desee acceder. Para algunos orígenes de datos no relacionales, debe definir una lista fija de columnas de entrada y salida cuando registra el apodo. Cada columna que especifique se correlaciona con un campo, columna o elemento determinados del objeto de origen de datos.

Como parte de la planificación, repase las opciones de apodo y columna existentes para el origen de datos que está configurando. Algunos orígenes de datos tienen opciones de apodo y columna necesarias que debe especificar.

Realizar tareas de configuración adicionales

Dependiendo de la forma en que desee trabajar con el origen de datos, puede ser conveniente realizar estas tareas de configuración adicionales.

Crear especificaciones de índice

Puede definir una especificación de índice para objetos que no tienen un índice. Por ejemplo, cree una especificación de índice cuando una tabla adquiera un nuevo índice o si el objeto de origen de datos, tal como una vista, carece de índice.

Definir correlaciones de tipos de datos alternativas

En el sistema federado, existen correlaciones definidas de forma predeterminada entre los tipos de datos del origen de datos y los tipos de datos de la base de datos federada. Para los orígenes de datos relacionales, puede definir correlaciones de tipos de datos alternativas. Por ejemplo, puede cambiar una correlación de tipos para todos los objetos de origen de datos situados en un servidor determinado o cambiar una correlación de tipos para un determinado objeto de origen de datos, tipo de origen de datos, u objeto y tipo de origen de datos.

Definir correlaciones de funciones alternativas

En el sistema federado, existen correlaciones de funciones definidas de forma predeterminada entre las funciones incorporadas del origen de datos

y las funciones incorporadas de la base de datos federada. Para los orígenes de datos relacionales, puede definir correlaciones de funciones alternativas. Por ejemplo, puede definir una correlación de funciones alternativa cuando desee utilizar una nueva función incorporada o función definida por el usuario que existe en el origen de datos, pero para la cual la base de datos carece de una función correlacionada.

Distinción entre mayúsculas y minúsculas y uso correcto de las comillas

Cuando especifica valores de opción y objetos en sentencia de SQL de DB2, es necesario que conozca cuándo es necesario usar comillas, qué tipo de comillas utilizar y cómo afectan a la distinción entre mayúsculas y minúsculas.

La forma de designar objetos cuando los crea por primera vez afecta al tipo de caracteres (mayúsculas o minúsculas) utilizados en el nombre del objeto y determina la forma de especificar nombres de objetos y valores de opciones en los mandatos. Por ejemplo, al crear un apodo, si no encierra el nombre entre comillas dobles, el catálogo del sistema almacena el apodo utilizando letras mayúsculas, sin importar el tipo de caracteres (mayúsculas o minúsculas) utilizado para designar el objeto. Si encierra el nombre entre comillas dobles, el catálogo almacena los caracteres del nombre de objeto utilizando el mismo tipo de caracteres (mayúsculas o minúsculas) que los especificados por el usuario al crear el nombre. Luego, cuando utilice el nombre de objeto como valor de opción, debe especificarlo respetando el mismo uso de mayúsculas y minúsculas. Por ejemplo, la opción de columna FOREIGN_KEY utilizada por el script, los servicios Web y los derivadores XML necesita que el usuario especifique el apodo como valor de opción para la columna de clave foránea. Cuando especifique el valor de opción, debe utilizar el mismo tipo de caracteres (mayúsculas o minúsculas) que el utilizado por el catálogo del servidor federado para almacenar el apodo.

La tabla siguiente describe el uso correcto de mayúsculas y minúsculas y de las comillas cuando se especifican valores de opción y objetos en sentencias de SQL de DB2.

Tabla 5. Uso correcto de mayúsculas y minúsculas y de las comillas

Identificador	Uso de mayúsculas/minúsculas y de comillas	Ejemplos
Valor de opción	Utilice el tipo de carácter (mayúsculas o minúsculas) necesario para el valor de opción y encierre el valor entre comillas simples.	<p>Esta sentencia crea una tabla de origen de datos llamada remote_schema.remote_table (todo en minúsculas):</p> <pre>CREATE TABLE newton.my_nick (c1 int) OPTIONS (remote_server 'MY_SERVER' remote_schema 'remote_schema', remote_tabname 'remote_table');</pre> <p>Esta sentencia crea una tabla de origen de datos llamada REMOTE_SCHEMA.REMOTE_TABLE (todo en mayúsculas):</p> <pre>CREATE TABLE newton.my_nick (c1 int) OPTIONS (remote_server'MY_SERVER' remote_schema 'REMOTE_SCHEMA', remote_tabname 'REMOTE_TABLE');</pre>
Objeto que solamente contiene caracteres en minúsculas	Utilice solamente caracteres en minúsculas y encierre el identificador entre comillas dobles.	<p>Esta sentencia crea un apodo para una tabla de origen de datos llamada infx_user.remote_table (todo en minúsculas):</p> <pre>CREATE NICKNAME my_nick FOR infx_server. "infx_user"."remote_table";</pre> <p>Nota: Algunos orígenes de datos, tales como Informix y Teradata, utilizan nombres en minúsculas por omisión.</p>
Objeto que solamente contiene caracteres en mayúsculas, números y caracteres de subrayado (_)	Existen dos posibilidades: <ul style="list-style-type: none"> • Utilice solamente caracteres en mayúsculas y encierre el identificador entre comillas dobles. • Utilice mayúsculas o minúsculas según desee y no encierre el identificador entre comillas dobles. 	<p>Cada una de estas sentencias crea el apodo MY_NICK (todo en mayúsculas):</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>

Para los ID de autorización y contraseñas del origen de datos, puede también utilizar las opciones de servidor FOLD_ID y FOLD_PW para convertir el ID y la contraseña al tipo de carácter correcto (mayúscula o minúscula).

Desde un indicador de mandatos del sistema operativo UNIX

Si encierra entre comillas un valor sensible a las mayúsculas y minúsculas en el indicador de mandatos de UNIX del servidor federado, debe asegurarse de que las comillas se analicen correctamente:

Sentencias de SQL que contienen comillas dobles, pero no contienen comillas simples

Si la sentencia de SQL contiene comillas dobles, pero no contiene comillas simples, encierre la sentencia completa entre comillas simples.

Por ejemplo, si desea emitir esta sentencia de SQL:


```
CREATE NICKNAME mi_apodo FOR mi_servidor."propietario"."mi_tabla"
```

Escriba el texto siguiente en el indicador de mandatos de UNIX:

```
db2 'CREATE NICKNAME mi_apodo FOR mi_servidor."propietario"."mi_tabla"'
```

Sentencias de SQL que contienen comillas simples, pero no contienen comillas dobles

Si la sentencia de SQL contiene comillas simples, pero no contiene comillas dobles, encierre la sentencia completa entre comillas dobles.

Por ejemplo, si desea emitir esta sentencia de SQL:

```
CREATE USER MAPPING FOR USER SERVER mi_servidor  
  OPTIONS(REMOTE_AUTHID 'mi_id', REMOTE_PASSWORD 'mi_contraseña')
```

Escriba el texto siguiente en el indicador de mandatos de UNIX:

```
db2 "CREATE USER MAPPING FOR USER SERVER mi_servidor  
  OPTIONS(REMOTE_AUTHID 'mi_id', REMOTE_PASSWORD 'mi_contraseña')"
```

Sentencias de SQL que contienen tanto comillas simples como dobles

Si la sentencia de SQL contiene comillas simples y comillas dobles:

- Encierre toda la sentencia entre comillas dobles
- Preceda cada comilla doble de la sentencia con una barra inclinada.

Por ejemplo, para emitir la siguiente sentencia de SQL:

```
CREATE USER MAPPING FOR "id_local" SERVER mi_servidor  
  OPTIONS(REMOTE_AUTHID 'mi_id', REMOTE_PASSWORD 'mi_contraseña')
```

Escriba el texto siguiente en el indicador de mandatos de UNIX:

```
db2 "CREATE USER MAPPING FOR \"id_local\" SERVER mi_servidor  
  OPTIONS(REMOTE_AUTHID 'mi_id', REMOTE_PASSWORD 'mi_contraseña')"
```

En los ejemplos anteriores se supone que las sentencias de SQL se entran desde el indicador de mandatos de UNIX y la sentencia se pasa al mandato de DB2 sin utilizar la opción `-f`. Para utilizar el mandato de DB2 con la opción `-f` para entrar las sentencias de SQL desde un archivo, escriba las sentencias tal como se muestra en la primera aparición de cada ejemplo.

Desde un indicador de mandatos del sistema operativo Windows

Para preservar los valores sensibles a mayúsculas y minúsculas cuando entra mandatos desde un indicador de mandatos de Microsoft Windows del servidor federado, anteponga una barra inclinada invertida a cada comilla doble. Por ejemplo, suponga que desea crear el apodo *apodo1* para la tabla *salario_semanal* de Microsoft SQL Server. La tabla reside en la base de datos *NORBASE*. El esquema local es *mi_esquema*.

En el indicador de mandatos de Windows del servidor federado, escriba:

```
db2 CREATE NICKNAME apodo1  
  FOR NORBASE."mi_esquema\"."salario_semanal\"
```

Desde la línea de mandatos de DB2 o desde una aplicación

Cuando especifica un valor desde la línea de mandatos de DB2 o desde una aplicación, puede preservar los valores sensibles a mayúsculas y minúsculas encerrando esos valores con las comillas apropiadas.

Por ejemplo, suponga que desea crear una correlación de usuarios para el ID de usuario *id_local*. El ID de usuario remoto es *mi_id* y la contraseña es *mi_contraseña*. Desea que los tres valores se mantengan en minúsculas. En el indicador de mandatos de DB2, escriba:

```
CREATE USER MAPPING FOR "id_local" SERVER mi_servidor
  OPTIONS(REMOTE_AUTHID 'mi_id', REMOTE_PASSWORD 'mi_contraseña')
```

Configuración de varios servidores federados para acceder a orígenes de datos

Un sistema federado puede constar de varios servidores federados. En lugar de configurar cada servidor federado por separado, puede ahorrar tiempo utilizando el Centro de control para configurar los servidores federados.

Antes de empezar

- Federation debe estar instalado en un servidor que actuará como servidor federado.
- Debe existir una base de datos federada en el servidor federado.

Cuando configure el primer servidor, la ventana **Resultado de acciones** captura las sentencias DDL que se emitieron al crear los objetos federados. Puede reutilizar o modificar estas sentencias, y aplicar las sentencias para configurar rápidamente otros servidores federados.

La ventana **Resultado de acciones** permanece activa durante la sesión actual. Si cierra la ventana **Resultado de acciones**, las sentencias de DDL de la sesión actual se siguen almacenando en la ventana **Resultado de acciones**. Sin embargo, si cierra el Centro de control de DB2, todas las sentencias de DLL de la sesión actual se eliminan de la ventana **Resultado de acciones**.

Procedimiento

Para configurar varios servidores federados para acceder a orígenes de datos:

1. Utilice el Centro de control para configurar el primer servidor federado para los orígenes de datos a los que desee acceder. Esta acción captura cada sentencia de DDL.
2. Visualice la página **Resultado de acciones** en la ventana **Resultado de acciones**. Si ha cerrado la ventana **Resultado de acciones**, pulse con el botón derecho en la carpeta **Objetos de base de datos federada** y pulse en **Mostrar acciones** para abrir la ventana **Resultado de acciones**.
3. Suprima cualquier sentencia DDL que no desee utilizar en los otros servidores federados. Para suprimir una sentencia, pulse con el botón derecho del ratón en la sentencia y seleccione **Eliminar**. Por ejemplo, puede desear eliminar las sentencias que muestran Failed (Fallida) en la columna de estado en la página **Resultado de las acciones**.
4. Copie las sentencias que desea utilizar en los otros servidores federados en la página **Editor de mandatos**.
 - a. Seleccione las sentencias que desea copiar. Para seleccionar varias sentencias, utilice la tecla Control.
 - b. Pulse con el botón derecho en las sentencias seleccionadas y pulse **Copiar en el Editor de mandatos**. Se abrirá la página **Editor de mandatos**.
5. Modifique las sentencias DDL en la página **Editor de mandatos** que desee utilizar en los otros servidores federados. Por ejemplo, podría cambiar cualquier sentencia que especifique un esquema local.

Debe modificar las correlaciones de usuario para especificar las contraseñas apropiadas para el servidor federado. Cuando el DDL para las sentencias CREATE USER MAPPING es capturado en la ventana **Resultado de acciones**, las contraseñas están enmascaradas por asteriscos. Debe sustituir los asteriscos por las contraseñas apropiadas.

6. Después de cambiar las sentencias DDL que se generaron en la ventana **Resultado de acciones**, ejecute las sentencias DDL en el siguiente servidor federado.

Capítulo 2. Configurar orígenes de datos

Puede configurar el servidor federado para que acceda a orígenes de datos relacionales y no relacionales.

Configuración del acceso a orígenes de datos BioRS

Puede integrar los datos que se encuentran en los bancos de datos de BioRS con la información procedente de otros orígenes utilizando un sistema federado.

Procedimiento

Para configurar un servidor federado para acceder a los orígenes de datos BioRS, debe proporcionar al servidor federado información acerca de los orígenes de datos y objetos a los que desea acceder. Después de configurar el servidor federado, puede crear consultas y utilizar funciones personalizadas para acceder a los orígenes de datos BioRS.

Derivador BioRS

BioRS es un sistema de consulta y recuperación que puede utilizarse para recuperar información de varios orígenes de datos. Para ejecutar consultas, el derivador BioRS utiliza las mismas API que la interfaz BioRS basada en web.

BioRS es un sistema de consulta y recuperación desarrollado por Biomax Informatics. Puede utilizar BioRS para recuperar información de varios orígenes de datos, incluyendo archivos sin formato y bases de datos relacionales. Generalmente, en el sistema BioRS los datos públicos, como SwissProt y GenBan, se descargan como archivos sin formato. BioRS puede integrar orígenes de datos públicos y orígenes de datos de propiedad (por ejemplo, bases de datos privadas mantenidas por la organización) en un entorno común.

Una vez que un origen de datos se ha integrado en el sistema BioRS, se hace referencia al mismo como *banco de datos*. Los elementos incluidos en cada entrada del banco de datos se conocen colectivamente como *esquema*. Los elementos de un banco de datos que están indexados pueden utilizarse en las funciones BIOR.S.CONTAINS, BIOR.S.CONTAINS_GE y BIOR.S.CONTAINS_LE. Las funciones BioRS se especifican en la cláusula WHERE de la sentencia SELECT. Se puede hacer referencia a los elementos que no están indexados en la lista SELECT y en otros predicados de la cláusula WHERE. Los elementos que no están indexados se procesan en el servidor federado.

Es posible establecer relaciones entre las entradas de los bancos de datos, a fin de poder enlazar bancos de datos entre sí en el sistema BioRS.

Los bancos de datos BioRS pueden tener una relación padre-hijo (los bancos de datos pueden anidarse). En una relación de este tipo, el banco de datos hijo contiene un elemento de tipo de datos de referencia denominado PARENT. El elemento PARENT hace referencia al elemento `_ID_ element` del banco de datos padre. A parte de la presencia de este elemento PARENT predefinido, los bancos de datos anidados contienen los mismos datos que los bancos de datos no anidados.

BioRS proporciona una interfaz basada en web que permite a los usuarios ejecutar consultas en los datos de los bancos de datos BioRS. El derivador BioRS utiliza las mismas interfaces de programación de aplicaciones (API) que la interfaz BioRS basada en web para ejecutar consultas.

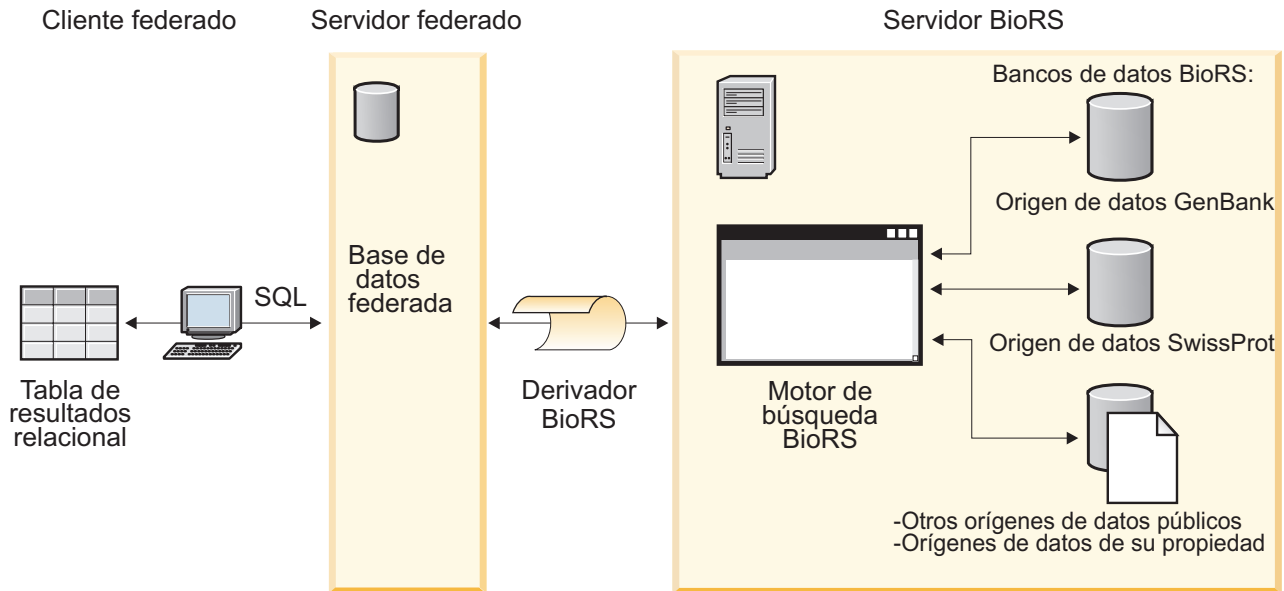


Figura 2. Funcionamiento de derivador BioRS

Desde el cliente, los usuarios o las aplicaciones emiten una consulta mediante sentencias SQL. A continuación, la consulta se envía al sistema federado en el que se halla instalado el derivador BioRS. En función de cómo se construya la consulta, puede utilizarse tanto el servidor federado como el servidor BioRS para procesar la consulta. El servidor BioRS puede encontrarse en un sistema distinto del sistema federado. La información de autenticación debe proporcionarla el sistema federado al servidor BioRS para cada consulta. Esta información puede ser una combinación de ID de usuario y contraseña, o una indicación no autenticada (generalmente una consulta de invitado).

Para obtener información detallada sobre el producto BioRS, consulte el sitio web de Biomax en: <http://www.biomax.com>.

Adición de orígenes de datos BioRS a un servidor federado

Para configurar un servidor federado para acceder a orígenes de datos BioRS, debe proporcionar al servidor federado información acerca de los orígenes de datos y objetos a los que desea acceder.

Antes de empezar

- Federation debe estar instalado en el servidor que actuará como servidor federado.
- Debe existir una base de datos en el servidor federado.

Acerca de esta tarea

Puede configurar un servidor federado para acceder a datos almacenados en orígenes de datos BioRS utilizando el Centro de control o emitiendo sentencias

SQL en la línea de mandatos. El Centro de control incluye un asistente que le guiará en los pasos necesarios para configurar los objetos federados necesarios.

Procedimiento

Para añadir orígenes de datos BioRS a un servidor federado:

1. Registre las funciones personalizadas para el derivador de BioRS.
2. Registre el derivador de BioRS.
3. Registre las definiciones de servidor BioRS.
4. Opcional: Cree las correlaciones de usuario de BioRS.
5. Registre los apodos para los bancos de datos de BioRS.

Registro de las funciones personalizadas para el derivador de BioRS

Debe registrar las funciones personalizadas de BioRS antes de registrar el derivador de BioRS. Las funciones personalizadas BioRS se utilizan con el derivador de BioRS para enviar predicados al motor de consultas de BioRS.

Acerca de esta tarea

Debe registrar todas las funciones personalizadas en cada instancia de base de datos federada en la que esté instalado el derivador de BioRS.

Todas las funciones personalizadas del derivador de BioRS deben registrarse con el nombre de esquema `biors`.

Procedimiento

Para registrar las funciones personalizadas de BioRS:

Para cada una de las funciones personalizadas de BioRS, emita la sentencia `CREATE FUNCTION`. Debe especificar dos tipos de datos en cada función personalizada. El primer tipo de datos que debe especificar es la columna indexada. El segundo tipo de datos que debe especificar es el término de búsqueda. Debe incluir las palabras clave `AS TEMPLATE`, `DETERMINISTIC` y `NO EXTERNAL ACTION` en la sentencia `CREATE FUNCTION`.

El ejemplo siguiente muestra la sintaxis de la función `BIORS.CONTAINS`:

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

Consejo: Para registrar las funciones personalizadas de BioRS, utilice el archivo de ejemplo `create_function_mappings.ddl`. El archivo de ejemplo se encuentra en el directorio `sqllib/samples/lifesci/biors` del servidor federado. El archivo de ejemplo contiene las sentencias `CREATE FUNCTION` para cada una de las posibles combinaciones de tipos de datos. Para registrar las funciones personalizadas, edite el archivo `create_function_mappings.ddl` para especificar los tipos de datos para las columnas de índice y los términos de búsqueda para cada función personalizada. A continuación, debe ejecutar el archivo `create_function_mappings.ddl` en cada instancia de base de datos federada en la que esté instalado el derivador de BioRS.

Funciones personalizadas para el derivador de BioRS:

Descripciones y ejemplos de las funciones personalizadas utilizadas con el derivador de BioRS.

El entorno federado utiliza dos motores de consultas. Para el derivador de BioRS, estos motores de consultas son el motor de consultas de la base de datos federada y el motor de consultas de BioRS. Puede especificar que los predicados se envíen al motor de BioRS mediante las funciones personalizadas de BioRS.

Las funciones personalizadas para el derivador de BioRS son:

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

Para registrar las funciones personalizadas de BioRS, se utiliza la sentencia CREATE FUNCTION.

La tabla siguiente indica las cuatro funciones personalizadas de BioRS con ejemplos de los tipos de datos válidos que puede especificar al registrar las funciones.

Tabla 6. Funciones personalizadas para el derivador de BioRS

Función	Descripción
BIRS.CONTAINS (VARCHAR(), VARCHAR()) BIRS.CONTAINS (VARCHAR(), CHAR()) BIRS.CONTAINS (VARCHAR(), DATE) BIRS.CONTAINS (VARCHAR(), TIMESTAMP)	Busca en una columna indexada los valores iguales (según la semántica de consultas de BioRS) al valor especificado por el usuario. El primer argumento debe ser una referencia a la columna indexada y el segundo argumento es el valor de búsqueda especificado por el usuario.
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)	Busca en una columna indexada los valores iguales o inferiores (según la semántica de consultas de BioRS) al valor especificado por el usuario. El primer argumento debe ser una referencia a la columna indexada y el segundo argumento es el valor de búsqueda especificado por el usuario.
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)	Busca en una columna indexada los valores iguales o superiores (según la semántica de consultas de BioRS) al valor especificado por el usuario. El primer argumento debe ser una referencia a la columna indexada y el segundo argumento es el valor de búsqueda especificado por el usuario.
BIRS.SEARCH_TERM (VARCHAR(), VARCHAR()) BIRS.SEARCH_TERM (VARCHAR(), CHAR()) BIRS.SEARCH_TERM (CHAR(), VARCHAR()) BIRS.SEARCH_TERM (CHAR(), CHAR())	Pasa una expresión de consulta de BioRS al motor de búsqueda de BioRS.

Registro del derivador de BioRS

Debe registrar un derivador para acceder a los orígenes de datos BioRS. Los servidores federados utilizan los derivadores para comunicarse con los orígenes de datos y para recuperar datos de éstos. Los derivadores se implementan como un conjunto de archivos de biblioteca.

Si utiliza un servidor proxy para acceder a archivos de BioRS, debe especificar la información del proxy en forma de opciones al registrar el derivador. Por omisión, las opciones del derivador se utilizan al consultar cualquier archivo de BioRS. Sin

embargo, si también especifica opciones de proxy al registrar una definición de servidor, las opciones de servidor tienen preferencia.

Procedimiento

Para registrar el derivador de BioRS:

Elija el método que desee utilizar para registrar el derivador de BioRS:

Método	Procedimiento
Utilizar el Centro de control	Inicie el asistente Objetos federados. Pulse la carpeta Objetos de base de datos federada con el botón derecho del ratón y pulse Crear objetos federados .
Desde la línea de mandatos	Emita la sentencia CREATE WRAPPER. <pre>CREATE WRAPPER nombre_derivador LIBRARY nombre_biblioteca;</pre> <p>Si utiliza un servidor proxy para acceder a documentos de BioRS, la sentencia que debe emitir es la siguiente:</p> <pre>CREATE WRAPPER nombre_derivador LIBRARY nombre_biblioteca OPTIONS (PROXY_TYPE 'tipo', PROXY_SERVER_NAME 'nombre_servidor', PROXY_SERVER_PORT 'número_puerto');</pre>

Debe especificar el parámetro LIBRARY en la sentencia CREATE WRAPPER. El nombre del archivo de biblioteca del derivador especificado dependerá del sistema operativo del servidor federado. Consulte la lista de archivos de biblioteca de derivador de BioRS para conocer el nombre de biblioteca correcto que debe especificar en la sentencia CREATE WRAPPER.

Archivos de biblioteca del derivador de BioRS:

Los archivos de biblioteca de derivador de BioRS se añaden al servidor federado cuando éste se instala.

Cuando instala el servidor federado, se añaden tres archivos de biblioteca a la vía de acceso del directorio predeterminado. Por ejemplo, si el servidor federado se ejecuta en AIX, los archivos de biblioteca del derivador que se añaden a la vía de acceso del directorio son libdb2lsbiors.a, libdb2lsbiorsF.a y libdb2lsbiorsU.a. El archivo de biblioteca de derivador predeterminado es libdb2lsbiors.a. Los demás archivos de biblioteca de derivador se utilizan con opciones de derivador específicas.

Debe incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER y especificar el nombre de archivo de biblioteca del derivador predeterminado.

En la tabla siguiente se indican las vías de acceso al directorio predeterminado y los nombres de archivo de biblioteca del derivador predeterminado.

Tabla 7. Ubicaciones de biblioteca y nombres de archivo del derivador de BioRS

Sistema operativo	Vía de acceso al directorio	Nombre de archivo de biblioteca de derivador
AIX	/usr/opt/<vía_acceso_instalación>/lib32/ /usr/opt/<vía_acceso_instalación>/lib64/	libdb2lsbiors.a

Tabla 7. Ubicaciones de biblioteca y nombres de archivo del derivador de BioRS (continuación)

Sistema operativo	Vía de acceso al directorio	Nombre de archivo de biblioteca de derivador
Linux	/opt/IBM/db2/<vía_acceso_instalación>/lib32libdb2lsbiors.so /opt/IBM/db2/<vía_acceso_instalación>/lib64	
Solaris	/opt/IBM/db2/<vía_acceso_instalación>/lib32libdb2lsbiors.so /opt/IBM/db2/<vía_acceso_instalación>/lib64	
Windows	%DB2PATH%\bin	db2lsbiors.dll

<vía_acceso_instalación> es la vía de acceso al directorio en el que está instalado el servidor federado en UNIX o Linux.

%DB2PATH% es la variable de entorno que especifica la vía de acceso al directorio donde se ha instalado el servidor federado en Windows. La vía de acceso al directorio de Windows predeterminado es C:\Program Files\IBM\SQLLIB.

Sentencia CREATE WRAPPER - Ejemplos para el derivador BioRS:

Utilice la sentencia CREATE WRAPPER para registrar el derivador BioRS. Los ejemplos muestran las opciones necesarias para acceder a los documentos BioRS con y sin un servidor proxy.

Registro de un derivador

Si no utiliza ningún servidor proxy para acceder a los documentos BioRS, debe emitir esta sentencia para registrar el derivador:

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

derivador_biors

Nombre que se asigna al derivador BioRS. No se permite la duplicación de nombres de derivador.

LIBRARY '*libdb2lsbiors.a*'

Nombre del archivo de biblioteca para los servidores federados que utilizan sistemas operativos AIX.

Registro de un derivador para un servidor proxy HTTP

Para registrar un derivador y especificar un servidor proxy HTTP, utilice esta sentencia:

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

PROXY_TYPE '*HTTP*'

Especifica el tipo de proxy que se utiliza para acceder a Internet cuando el servidor federado se encuentra detrás de un cortafuegos. Los valores válidos son 'NONE', 'HTTP' o 'SOCKS'.

PROXY_SERVER_NAME '*proxy.mysite.com*'

Especifica el nombre o la dirección IP del servidor proxy. Esta opción es necesaria si el valor para la opción de servidor PROXY_TYPE es 'HTTP' o 'SOCKS'.

PROXY_SERVER_PORT '81'

Especifica el número de puerto del servidor proxy. Esta opción es necesaria si el valor para la opción de servidor PROXY_TYPE es 'HTTP' o 'SOCKS'.

Registro de un derivador para un servidor SOCKS

Para registrar un derivador y especificar un servidor proxy SOCKS sin información de autenticación, utilice esta sentencia:

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

LIBRARY 'libdb2lsbiors.so'

Nombre del archivo de biblioteca de derivador para los servidores federados que utilizan los sistemas operativos Linux y Solaris.

PROXY_TYPE 'SOCKS'

Especifica el tipo de proxy que se utiliza para acceder a Internet cuando el servidor federado se encuentra detrás de un cortafuegos. Los valores válidos son 'NONE', 'HTTP' o 'SOCKS'.

PROXY_SERVER_NAME 'proxy_socks'

Especifica el nombre o la dirección IP del servidor proxy. Esta opción es necesaria si el valor para la opción de servidor PROXY_TYPE es 'HTTP' o 'SOCKS'.

PROXY_SERVER_PORT '1081'

Especifica el número de puerto del servidor proxy. Esta opción es necesaria si el valor para la opción de servidor PROXY_TYPE es 'HTTP' o 'SOCKS'.

Registro de la definición del servidor para un origen de datos BioRS

Debe registrar cada servidor BioRS al que desee acceder en la base de datos federada.

Si utiliza un servidor proxy para acceder a archivos de BioRS, puede especificar la información del proxy en forma de opciones al registrar la definición del servidor. Si especifica la información de proxy en la definición del servidor, las opciones del servidor tendrán preferencia sobre las opciones que haya especificado al registrar el derivador.

Procedimiento

Para registrar una definición de servidor para un origen de datos BioRS:

Elija el método que desee utilizar para registrar la definición del servidor:

Método	Procedimiento
Utilizar el Centro de control	Inicie el asistente Objetos federados. Pulse la carpeta Objetos de base de datos federada con el botón derecho del ratón y pulse Crear objetos federados .

Método	Procedimiento
Desde la línea de mandatos	<p>Emita la sentencia CREATE SERVER.</p> <pre>CREATE SERVER <i>nombre_definición_servidor</i> VERSION <i>númeroversión</i> WRAPPER <i>nombre_derivador</i> OPTIONS (NODE '<i>nombre_nodo</i>');</pre> <p>Si utiliza un servidor proxy para acceder a archivos de BioRS, debe especificar varias opciones al registrar el derivador de BioRS en la definición del servidor. Para especificar la información del servidor proxy al registrar el derivador de BioRS, emita esta sentencia:</p> <pre>CREATE SERVER <i>nombre_definición_servidor</i> VERSION <i>númeroversión</i> WRAPPER <i>nombre_derivador</i> OPTIONS (PROXY_TYPE '<i>tipo</i>', PROXY_SERVER_NAME '<i>nombre_servidor</i>', PROXY_SERVER_PORT '<i>número puerto</i>');</pre>

Cuando registre la definición de servidor, especificará las opciones de servidor en la sentencia CREATE SERVER. Existen opciones de servidor obligatorias y opcionales. La opción de servidor NODE es obligatoria.

Una vez registrada la definición del servidor, utilice la sentencia ALTER SERVER para añadir o eliminar opciones de servidor.

Sentencia CREATE SERVER - Ejemplos para el derivador de BioRS:

Utilice la sentencia CREATE SERVER para registrar definiciones del servidor para el derivador de BioRS. Los ejemplos muestran los parámetros obligatorios, los parámetros opcionales y las opciones adicionales del servidor que puede especificar.

Registro de una definición de servidor con los parámetros obligatorios

El ejemplo siguiente muestra cómo registrar una definición de servidor para un derivador de BioRS emitiendo la sentencia CREATE SERVER:

```
CREATE SERVER servidor_biors VERSION '5.2' WRAPPER derivador_biors
OPTIONS(NODE 'biors_myco.com');
```

servidor_biors

Nombre asignado por el usuario al servidor BioRS. No se permiten los nombres de definición de servidor duplicados.

VERSION '*5.2*'

La versión del servidor BioRS al que desea acceder. Las versiones de BioRS soportadas son 5.0.14 y 5.2. Si accede a un servidor BioRS de la versión 5.2, debe especificar '*5.2*' como valor del parámetro VERSION. No es necesario especificar esta opción si utiliza la versión 5.0.14. Si no especifica el valor, se utilizará para este parámetro el valor predeterminado '*1.0*', que es equivalente a la versión 5.0.14.

WRAPPER *derivador_biors*

El nombre del derivador que ha especificado en la sentencia CREATE WRAPPER.

NODE '*biors_myco.com*'

Especifica el nombre de host del sistema donde está disponible la

herramienta de consulta de BioRS. El valor predeterminado es *localhost*. Este valor es sensible a las mayúsculas y minúsculas.

Aunque el nombre del nodo se especifica como opción en la sentencia CREATE SERVER, es necesario para los orígenes de datos BioRS.

Registro de una definición de servidor utilizando parámetros opcionales y opciones de servidor

El ejemplo siguiente muestra parámetros adicionales y opciones de servidor que puede especificar al registrar una definición de servidor para un derivador de BioRS:

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

TYPE *BioRS*

Especifica el tipo de servidor de orígenes de datos para el que está configurando el acceso. Para el derivador de BioRS, el tipo de servidor debe ser BioRS. Este parámetro es opcional.

PORT '*5555*'

Especifica el número del puerto que debe utilizarse para conectarse al servidor BioRS. El valor predeterminado es '*5014*'.

TIMEOUT *30*

Especifica el tiempo en minutos durante el que el derivador de BioRS espera una respuesta del servidor BioRS. El valor predeterminado es 10. Este parámetro es opcional.

CASE_SENSITIVE '*N*'

Especifica si el servidor BioRS trata los nombres de forma sensible a las mayúsculas y minúsculas. Los valores válidos son '*Y*' o '*N*'. El valor predeterminado es '*Y*'.

En el producto BioRS, un parámetro de configuración controla la sensibilidad a las mayúsculas y minúsculas de los datos que se almacenan en la máquina del servidor BioRS. La opción CASE_SENSITIVE es el equivalente del servidor federado al parámetro de configuración del sistema BioRS. Debe sincronizar los valores de configuración de sensibilidad a las mayúsculas y minúsculas del servidor BioRS en el sistema BioRS y en el servidor federado. Si no mantiene sincronizados los valores de configuración de la sensibilidad a las mayúsculas y minúsculas entre el servidor BioRS y el servidor federado, se producirán errores cuando intente acceder a los datos de BioRS a través del servidor federado.

Importante: no puede cambiar ni suprimir la opción CASE_SENSITIVE después de crear una definición de BioRS. Si necesita cambiar la opción CASE_SENSITIVE, debe eliminar la definición del servidor y volver a crearla. Si elimina la definición del servidor BioRS, también deberá crear todos los apodos de BioRS que hacían referencia a ella. El servidor federado elimina automáticamente todos los apodos correspondientes a un servidor eliminado.

Registro de una definición de servidor que incluye un servidor proxy

```
CREATE SERVER servidor_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*'

Especifica el tipo de proxy utilizado para acceder a Internet cuando se encuentra detrás de un cortafuegos. Los valores válidos son 'NONE', 'HTTP' o 'SOCKS'.

PROXY_SERVER_NAME '*proxy.mysite.com*'

Especifica el nombre o la dirección IP del servidor proxy. Esta opción es necesaria si el valor de la opción de servidor PROXY_TYPE es 'HTTP' o 'SOCKS'.

PROXY_SERVER_PORT '*81*'

Especifica el número de puerto del servidor proxy. Esta opción es necesaria si el valor de la opción de servidor PROXY_TYPE es 'HTTP' o 'SOCKS'.

Registro de una definición de servidor que incluye un servidor proxy con información de autenticación

Para registrar una definición de servidor y especificar un servidor proxy SOCKS con información de autenticación, utilice la sentencia siguiente:

```
CREATE SERVER servidor_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*'

Especifica el ID de usuario del servidor proxy. Esta opción de servidor es necesaria si el valor de la opción de servidor PROXY_TYPE es 'SOCKS'.

PROXY_PASSWORD '*bargle*'

Especifica la contraseña del servidor proxy asociada con el nombre de usuario '*argle*'. Esta opción de servidor es necesaria si el valor de la opción de servidor PROXY_TYPE es 'SOCKS'.

Creación de las correlaciones de usuario para un origen de datos BioRS

Cuando se intenta acceder a un servidor BioRS, el servidor federado establece una conexión con el servidor BioRS. Dependiendo de los métodos de acceso a cuentas utilizados en el sistema BioRS, puede que no sea necesario crear correlaciones de usuario.

Una correlación de usuario es una asociación entre cada ID de usuario y contraseña del servidor federado y el ID de usuario y contraseña correspondientes del origen de datos.

Existen dos métodos para especificar correlaciones de usuario con sistemas federados. Puede utilizar un repositorio externo, por ejemplo LDAP, para almacenar las correlaciones de usuario, o crear las correlaciones de usuario en el catálogo de bases de datos federadas.

Antes de empezar

Los métodos de acceso a cuentas utilizados en el sistema BioRS determinarán si es necesario crear correlaciones de usuario:

- Si el servidor BioRS está configurado para el acceso de invitado para todas las cuentas de usuario, no es necesario crear correlaciones de usuario. El servidor federado utiliza una cuenta de invitado para acceder al servidor BioRS.
- Si dispone de un repositorio externo, por ejemplo LDAP, para almacenar las correlaciones de usuario, no es necesario crear correlaciones de usuario. Debe especificar la opción DB2_UM_PLUGIN en el derivador de BioRS. Puede especificar esta opción al registrar o modificar el derivador. El esquema del repositorio externo debe incluir el acceso de invitado.
- Si el servidor BioRS está configurado para autenticar cuentas de usuario con identificadores y contraseñas, debe crear correlaciones de usuario en la base de datos federada para las cuentas de usuario que vayan a utilizar el derivador de BioRS para acceder a orígenes de datos BioRS.
- Si el servidor BioRS está configurado para utilizar una mezcla de cuentas de usuario de invitado y autenticadas, debe crear correlaciones de usuario para las cuentas de usuario autenticadas que vayan a utilizar el derivador de BioRS para acceder a orígenes de datos BioRS.

Acerca de esta tarea

Las correlaciones de usuario ofrecen un modo de autenticar el acceso de los usuarios o las aplicaciones que consultan un origen de datos BioRS con el derivador de BioRS. Si no hay una correlación de usuario definida para un usuario o una aplicación, el servidor federado utiliza una cuenta de invitado. Si un banco de datos que se está consultando requiere autenticación, puede que se devuelva un mensaje de error.

Para asegurarse de que se pasan el ID de usuario y la contraseña correctos al servidor BioRS, cree correlaciones de usuario en la base de datos federada para los usuarios que tengan autorización para realizar búsquedas en los orígenes de datos BioRS. Al crear una correlación de usuario, la contraseña remota se almacena en formato cifrado en una tabla de catálogo del sistema de la base de datos federada.

Procedimiento

Para correlacionar un ID de usuario local con el ID de usuario y la contraseña del servidor BioRS:

Elija el método que desee utilizar para crear las correlaciones de usuario:

Método	Procedimiento
Utilizar el Centro de control	Inicie el asistente Objetos federados. Pulse la carpeta Objetos de base de datos federada con el botón derecho del ratón y pulse Crear objetos federados .

Método	Procedimiento
Utilizar la línea de mandatos	Emita la sentencia CREATE USER MAPPING. Por ejemplo: <pre>CREATE USER MAPPING FOR <i>IDusuario_local</i> SERVER <i>nombre_definición_servidor</i> OPTIONS (REMOTE_AUTHID '<i>IDusuario_remoto</i>', REMOTE_PASSWORD '<i>contraseña_remota</i>') PROXY_AUTHID '<i>IDusuario_servidor_proxy</i>', PROXY_PASSWORD '<i>contraseña_servidor_proxy</i>');</pre>

Si especifica información de autenticación para el servidor proxy al registrar una definición de servidor y una correlación de usuario, los valores que especifique en la sentencia CREATE USER MAPPING tendrán preferencia sobre los valores que especifique en la sentencia CREATE SERVER.

Por ejemplo, en su organización hay diez personas y especifica información de autenticación al registrar la definición del servidor. Crea correlaciones de usuario para tres de las diez personas. Cuando esas tres personas accedan al sistema federado, se utilizará la información de autenticación que ha especificado al crear las correlaciones de usuario. Para las siete personas restantes, se utilizará la información de autenticación que ha especificado al registrar la definición del servidor.

Sentencia CREATE USER MAPPING - Ejemplos para el derivador de BioRS:

Utilice la sentencia CREATE USER MAPPING para correlacionar un ID de usuario de servidor federado con un ID de usuario y una contraseña de servidor BioRS.

Puede crear una correlación de usuario especificando una cuenta de usuario invitado, una cuenta de usuario autenticado, el registro especial USER o un servidor proxy.

Creación de una correlación de usuario para una cuenta de usuario invitado

La opción de usuario GUEST especifica si el derivador de BioRS debe utilizar una cuenta de invitado para acceder al servidor BioRS.

El ejemplo siguiente muestra cómo especificar que se utiliza una cuenta de usuario invitado para acceder al servidor BioRS:

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

charlie Especifica el ID de autorización local que se correlaciona con un ID de usuario remoto definido en el servidor BioRS.

SERVER *servidor_biors*

Especifica el nombre de la definición del servidor que ha registrado en la sentencia CREATE SERVER para el servidor BioRS.

GUEST 'Y'

Especifica que el derivador de BioRS utiliza una cuenta de usuario invitado para autenticar a este usuario.

Creación de una correlación de usuario para una cuenta de usuario autenticado

El ejemplo siguiente muestra cómo correlacionar un ID de usuario de servidor federado con un ID de usuario y una contraseña de servidor BioRS:

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

charlie Especifica el ID de autorización local que se correlaciona con un ID de usuario y una contraseña remotos definidos en el servidor BioRS.

SERVER *servidor_biors*

Especifica el nombre de la definición del servidor que ha registrado en la sentencia CREATE SERVER para el servidor BioRS.

REMOTE_AUTHID '*charlene*'

Especifica el ID de usuario del servidor BioRS con el que está correlacionando *charlie*. Este ID remoto debe tener el formato esperado por el servidor BioRS.

REMOTE_PASSWORD '*all4one*'

Especifica la contraseña asociada a '*charlene*'.

Creación de una correlación de usuario mediante un registro especial

Puede utilizar el registro especial USER de base de datos federada para correlacionar el ID de autorización de la persona que emite la sentencia CREATE USER MAPPING con el ID de autorización del origen de datos especificado en la opción de usuario REMOTE_AUTHID.

A continuación se ofrece un ejemplo de la sentencia CREATE USER MAPPING, que incluye el registro especial USER:

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

Creación de una correlación de usuario para un servidor proxy

El ejemplo siguiente muestra cómo correlacionar un ID de usuario de servidor federado con un ID de usuario y una contraseña de servidor BioRS:

Para registrar una definición de servidor y especificar un servidor proxy SOCKS con información de autenticación, utilice la sentencia siguiente:

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

PROXY_AUTHID '*chuck*'

Especifica el ID de usuario del servidor proxy. Esta opción de correlación de usuario es necesaria cuando el servidor proxy requiere autenticación.

PROXY_PASSWORD '*them2us*'

Especifica la contraseña del servidor proxy asociada con el nombre de usuario '*chuck*'. Esta opción de correlación de usuario es necesaria cuando el servidor proxy requiere autenticación.

Registro de apodos para orígenes de datos BioRS

Para cada definición de servidor BioRS que registre, debe registrar un apodo para cada banco de datos al que desee acceder. Utilice estos apodos en lugar de los nombres de los bancos de datos cuando consulte los servidores BioRS.

Antes de empezar

- Si un banco de datos de BioRS no se ajusta a la sintaxis necesaria para la sentencia CREATE NICKNAME, debe utilizar la opción de apodo REMOTE_OBJECT al registrar el apodo.
- Si un nombre de elemento de BioRS no se ajusta a la sintaxis necesaria para la sentencia CREATE NICKNAME, debe utilizar la opción de columna ELEMENT_NAME al registrar el apodo.

Restricciones

No utilice el elemento AllText de BioRS como primera columna de un apodo. Puede utilizar el elemento AllText de BioRS en cualquier otra posición de columna (por ejemplo, como segunda o tercera columna).

Acerca de esta tarea

Después de integrar un origen de datos en el sistema BioRS, se hace referencia a él como *banco de datos* en BioRS. Los bancos de datos de BioRS equivalen a los apodos de un sistema federado.

Los nombres que dé a los apodos pueden tener una longitud de hasta 128 caracteres.

Puede registrar un apodo mediante el Centro de control o desde la línea de mandatos. El Centro de control incluye un asistente que le guiará en los pasos necesarios para registrar el apodo.

Procedimiento

Para registrar un apodo para un banco de datos de BioRS:

Elija el método que desee utilizar para registrar el apodo:

Método	Procedimiento
Utilizar el Centro de control	Inicie el asistente Objetos federados. Pulse la carpeta Objetos de base de datos federada con el botón derecho del ratón y pulse Crear objetos federados .
Utilizar la línea de mandatos	Emita la sentencia CREATE NICKNAME. Por ejemplo: <pre>CREATE NICKNAME apodo (column_name tipos_datos OPTIONS (opciones_columna_apodo), column_name tipos_datos OPTIONS (opciones_columna_apodo), column_name tipos_datos OPTIONS (opciones_columna_apodo)) FOR SERVER nombre_definición_servidor OPTIONS (opciones_apodo);</pre>

Al crear un apodo de BioRS, se define una lista de columnas de apodo. Las columnas de apodo especificadas deben corresponder a elementos de un formato de banco de datos específico de BioRS. BioRS define cinco tipos de datos posibles para los elementos: Text, Number, Date, Author y Reference. Los tipos de datos de BioRS sólo pueden correlacionarse con los tipos de datos CHAR, CLOB o VARCHAR utilizados por la base de datos federada.

Repita este paso para cada banco de datos de BioRS para el que desee crear un apodo.

Sentencia CREATE NICKNAME - Ejemplos para el derivador de BioRS:

Utilice la sentencia CREATE NICKNAME para registrar un apodo para un banco de datos de BioRS al que desea acceder.

Creación de apodos sencillos pero limitados para bancos de datos

La forma más sencilla de registrar un apodo para un banco de datos de BioRS es dar al apodo el mismo nombre que el banco de datos de BioRS.

Por ejemplo:

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

El nombre del apodo es *SwissProt*, que es el mismo que el nombre del banco de datos de BioRS correspondiente.

La utilización de esta sintaxis simple de CREATE NICKNAME limita al usuario en dos aspectos:

1. Está limitado a una familia de apodos para cada esquema de base de datos federada. Por ejemplo, tiene dos bancos de datos que tienen una relación padre-hijo. Los bancos de datos son SWISSPROT y SPFEAT. Estos bancos de datos forman una familia. Si utiliza la sintaxis predeterminada para la sentencia CREATE NICKNAME, tendrá el apodo SWISSPROT para el banco de datos SWISSPROT y el apodo SPFEAT para el banco de datos SPFEAT. Para tener más de un apodo para SWISSPROT en el esquema, debe utilizar la opción de apodo REMOTE_OBJECT al registrar el apodo.
2. Está limitado a los bancos de datos cuyos nombres puedan utilizarse como apodos. El nombre del banco de datos debe ajustarse a la sintaxis soportada por el servidor federado. Por ejemplo, si el nombre del banco de datos incluye un punto o un espacio, debe utilizar la opción de apodo REMOTE_OBJECT al registrar el apodo.

Creación de varios apodos para el mismo banco de datos

La opción de apodo REMOTE_OBJECT especifica el nombre del banco de datos de BioRS asociado con el apodo. El nombre que especifique en la opción de apodo REMOTE_OBJECT determinará el esquema y el banco de datos de BioRS para el apodo. La opción de apodo REMOTE_OBJECT también especifica la relación del apodo con otros apodos.

El ejemplo siguiente muestra el mismo conjunto de características de apodo que en el ejemplo anterior, pero cambia el nombre del apodo. El ejemplo utiliza la opción de apodo `REMOTE_OBJECT` para especificar el banco de datos de BioRS para el que se define el apodo:

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

Creación de apodos para bancos de datos que no se ajustan a la sintaxis del servidor federado

El ejemplo siguiente muestra cómo crear un apodo para un banco de datos de BioRS remoto que no se ajusta a la sintaxis requerida por el servidor federado:

```
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

Apodo exclusivo utilizado para identificar el banco de datos de BioRS.

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

Nombre y tipo de datos para una columna de tabla. Se especifica la opción de columna `ELEMENT_NAME` para la columna `ID`.

La opción de columna `ELEMENT_NAME` especifica el nombre del elemento BioRS. La sensibilidad de mayúsculas y minúsculas de este nombre depende de la sensibilidad a mayúsculas y minúsculas del servidor BioRS y del valor de la opción `CASE_SENSITIVE` del servidor. Sólo es necesario especificar el nombre del elemento BioRS si es diferente del nombre de columna. Los valores de opción de columna deben especificarse entre comillas simples.

En general, utilice la opción de columna `ELEMENT_NAME` en las siguientes circunstancias:

- Cuando un nombre de elemento BioRS contiene caracteres, tales como puntos y espacios, que no se ajustan a la sintaxis de federación válida. Por ejemplo, si el banco de datos contiene un elemento denominado `Pub.Date`, no puede utilizar el nombre de elemento como nombre de columna. Los caracteres como los puntos y los espacios no están soportados. Debe correlacionar el nombre del elemento con un nombre de columna válido.
- Cuando la sintaxis de un nombre de elemento BioRS no se ajusta a los estándares establecidos por su organización para el sistema federado. Por ejemplo, si la organización ha establecido que los convenios para esquemas, apodos y columnas deben incluir un prefijo, es posible que no pueda utilizarse un nombre de elemento BioRS como nombre de columna.
- Cuando es posible que el nombre de elemento BioRS no resulte obvio para los usuarios federados.

ALLTEXT VARCHAR(128)

Nombre y tipo de datos para una columna de tabla.

ENTRYDATE *VARCHAR(64)*

Nombre y tipo de datos para una columna de tabla.

FtLength *VARCHAR(16)*

Nombre y tipo de datos para una columna de tabla.

SERVER *biors1*

Nombre asignado al servidor BioRS en la sentencia CREATE SERVER.

OPTIONS (REMOTE_OBJECT 'SwissProt.Features')

Especifica el nombre del banco de datos de BioRS asociado con el apodo. Este nombre determina el esquema y el banco de datos de BioRS para el apodo. Este nombre también especifica la relación del apodo con otros apodos.

La sensibilidad de mayúsculas y minúsculas de este nombre depende de la sensibilidad a mayúsculas y minúsculas del servidor BioRS y del valor de la opción CASE_SENSITIVE del servidor.

Importante: No puede cambiar ni suprimir este nombre con la sentencia ALTER NICKNAME. Si el nombre del banco de datos de BioRS cambia, debe suprimir el apodo y crearlo de nuevo.

Debe especificar la opción de apodo REMOTE_OBJECT cuando el nombre de un banco de datos BioRS no se ajuste a la sintaxis de federación válida. En este ejemplo, el nombre de banco de datos "SwissProt.Features" no se ajusta a ella por varias razones. El nombre del banco de datos contiene un carácter (un punto) que no se admite en la sintaxis de federación válida, y contiene una mezcla de letras mayúsculas y minúsculas.

En general, utilice la opción de apodo REMOTE_OBJECT en las siguientes circunstancias:

- Cuando un nombre de banco de datos BioRS contiene caracteres, tales como puntos y espacios, que no se ajustan a la sintaxis de federación válida. Debe correlacionar el nombre del banco de datos con un nombre federado válido.
- Cuando la sensibilidad a mayúsculas y minúsculas de un banco de datos BioRS no se ajusta a los estándares establecidos por el usuario o por su organización para el sistema federado. Por ejemplo, si la organización ha establecido que los convenios para esquemas, apodos y columnas deben incluir un prefijo, es posible que no pueda utilizarse un nombre de banco de datos BioRS como nombre.
- Cuando es posible que el nombre del banco de datos BioRS no resulte obvio para los usuarios federados.

Creación de apodos para un banco de datos enlazado a otro banco de datos BioRS

El ejemplo siguiente muestra cómo crear un apodo para una tabla que utiliza un banco de datos de BioRS que está enlazado a otro banco de datos de 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');
```

El nombre de este apodo es SwissFT2. Las columnas de la tabla son ID, ALLTEXT, FtKey, FtLength, FtDescription y Parent. Se especifica la opción de columna ELEMENT_NAME para la columna ID. Se utiliza la opción REMOTE_OBJECT para especificar el nombre del banco de datos BioRS al que corresponde el apodo.

Además, la columna Parent utiliza la opción REFERENCED_OBJECT. Debe especificar esta opción para las columnas correspondientes a elementos de tipo de datos Reference de BioRS. La opción REFERENCED_OBJECT especifica el nombre del banco de datos BioRS al que hace referencia la columna. En este caso, el elemento Parent hace referencia al banco de datos SwissProt de BioRS.

Funciones personalizadas y consultas BioRS

Las funciones personalizadas BioRS se utilizan con el derivador de BioRS para enviar predicados al motor de consultas de BioRS.

El entorno federado utiliza dos motores de consultas. Para el derivador de BioRS, estos motores de consultas son la base de datos federada y BioRS. Puede especificar que los predicados se envíen al motor de BioRS mediante las cuatro funciones personalizadas de BioRS.

Todas las funciones personalizadas del derivador de BioRS deben registrarse con el nombre de esquema BIORs. Debe incluir el esquema BIORs siempre que utilice las funciones.

Las funciones personalizadas para el derivador de BIORs son:

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

Las funciones CONTAINS de BioRS

Las funciones personalizadas BIORs.CONTAINS, BIORs.CONTAINS_LE y BIORs.CONTAINS_GE requieren un argumento de columna de término de búsqueda y un argumento de texto de consulta. El ejemplo siguiente muestra una sentencia BIORs.CONTAINS:

```
BIORs.CONTAINS  
(columna_término_búsqueda,término_consulta)
```

El valor del argumento de columna de término de búsqueda debe hacer referencia a una columna BioRS indexada. La utilización de una columna no indexada producirá el mensaje de error SQL30090N ("Operación no válida para entorno de ejecución de aplicación").

El valor del argumento de término de consulta es el valor utilizado para buscar en el elemento indexado especificado en el argumento de columna de término de búsqueda.

El valor del argumento de término de consulta sólo puede ser un literal, una variable host o una referencia de columna. No puede utilizarse la concatenación aritmética ni de series. El valor del argumento de término de consulta tampoco puede ser NULL, aunque la columna de término de búsqueda utilizada esté definida para permitir valores nulos.

Las mayúsculas y minúsculas no son significativas en el argumento de término de consulta.

Tipos de datos válidos

Los tipos y formatos de datos válidos del argumento de término de consulta dependen del tipo de datos BioRS de la columna de término de búsqueda que se utilice. BioRS define cinco tipos de datos posibles: Text, Author, Date, Number y Reference.

En la tabla siguiente se indican los tipos de datos BioRS y los términos de consulta de función válidos para cada tipo de datos.

Tabla 8. Tipos de datos BioRS y términos de consulta válidos de función personalizada

Tipo de datos de columna de término de búsqueda	Término de consulta válido	Formato
Text	VARCHAR() o CHAR()	Término de texto de BioRS, incluidos los comodines.
Author	VARCHAR() o CHAR()	referencia al autor de BioRS en el formato "<apellido>, <inic>". "<apellido>" es el apellido del autor. "<inic>" son las iniciales del autor, sin puntos. Se acepta un espacio en blanco entre la coma y las iniciales. Como alternativa, puede especificarse sólo <apellido>, sin comas ni iniciales.
Date	VARCHAR(), CHAR(), DATE o TIMESTAMP	Si es una serie de caracteres, el formato de fecha válido para la base de datos federada, aaaa/mm/dd.
Number	VARCHAR() o CHAR(), INTEGER, SMALLINT, BIGINT REAL, DOUBLE, DECIMAL	Formatos de número válidos para la base de datos federada.
Reference	VARCHAR() o CHAR()	Término de texto de BioRS.

Todas las demás combinaciones de argumentos de columnas de término de búsqueda y término de consulta producirán el mensaje de error SQL30090N ("Operación no válida para entorno de ejecución de aplicación").

Utilización de caracteres comodín

El argumento de término de consulta para las columnas de término de búsqueda de los tipos de datos Text, Author y Reference debe coincidir con un patrón del lenguaje de consulta de BioRS. En BioRS, los argumentos de término de consulta pueden constar de series alfanuméricas y comodines. La función BIoRS.CONTAINS admite dos comodines: ? (signo de interrogación) y * (asterisco).

El carácter ? compara un único carácter. Por ejemplo, el predicado BIoRS.CONTAINS (description, 'bacteri?')=1 coincide con el término bacteria, pero no con el término bacterial.

El carácter comodín * compara cero o más caracteres. Por ejemplo, el predicado `BioRS.CONTAINS (description, 'bacteri*')`=1 coincide con los términos `bacteri`, `bacteria` y `bacterial`.

Para obtener información detallada acerca de los patrones del lenguaje de consulta de BioRS, consulte la documentación de BioRS.

Especificación de funciones BioRS CONTAINS en consultas

La función `BIORS.CONTAINS` puede especificarse para todos los tipos de columnas BioRS.

Las funciones personalizadas `BIORS.CONTAINS_GE` y `BIORS.CONTAINS_LE` sólo pueden especificarse para columnas cuyo tipo de datos BioRS subyacente sea `Number` o `Date`. La función `BIORS.CONTAINS_GE` selecciona filas en las que la columna contiene un valor igual o superior al valor representado por el argumento de término de consulta. La función `BIORS.CONTAINS_LE` selecciona filas en las que la columna contiene un valor igual o inferior al valor representado por el argumento de término de consulta.

Las funciones `BIORS.CONTAINS`, `BIORS.CONTAINS_GE` y `BIORS.CONTAINS_LE` devuelven un resultado entero. Si se utiliza alguna de las tres funciones `CONTAINS` en un predicado, el valor de retorno debe compararse con el valor 1 utilizando los operadores `=` o `<>`. Por ejemplo:

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

La expresión `NOT (BioRS.Contains (col,value) = 1)` es equivalente a la expresión `BioRS.CONTAINS (col,value) <> 1`.

La función SEARCH_TERM de BioRS

La función personalizada `BIORS.SEARCH_TERM` requiere un argumento de columna de término de búsqueda y un término de consulta. El ejemplo siguiente muestra la sintaxis de la función personalizada `SEARCH_TERM`:

```
BIORS.SEARCH_TERM (columna_término_búsqueda,término_consulta)
```

El valor del argumento de columna de término de búsqueda debe hacer referencia a la columna que representa el elemento `_ID_`.

El valor del argumento de término de consulta es una expresión que puede hacer referencia a varios elementos.

El valor del argumento de término de consulta sólo puede ser un literal, una variable host o una referencia de columna. No puede utilizarse la concatenación aritmética ni de series. El valor del argumento de término de consulta tampoco puede ser `NULL`, aunque la columna de término de búsqueda utilizada esté definida para permitir valores nulos.

Las mayúsculas y minúsculas no son significativas en el argumento de término de consulta.

Especificación de la función BioRS SEARCH_TERM en consultas

Emitiendo la función `BIORS.SEARCH_TERM`, puede ejecutar consultas que de otro modo no podrían realizarse. Puede utilizar esta función para especificar un término de búsqueda mediante el formato de BioRS. La función

BIORS.SEARCH_TERM requiere dos argumentos. El primer argumento es una referencia a la columna `_ID_` del apodo al que debe aplicarse el término. El segundo argumento es una serie de caracteres que contiene el término sin un nombre de banco de datos.

El ejemplo siguiente selecciona todas las columnas correspondientes a las entradas del banco de datos MyEMBL en las que el elemento SeqLength contiene un valor igual o superior a 100.

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

El ejemplo siguiente selecciona la columna MolWeight del apodo Swiss en la que el valor del elemento MolWeight es igual o superior a 100368.

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

Operaciones de igualdad en consultas BioRS

Puede utilizar un operador de igualdad (=) en expresiones literales o en consultas de unión, con determinadas limitaciones.

Si utiliza el operador de igualdad en una expresión literal o en una consulta de unión, el operador de igualdad debe hacer referencia al elemento `_ID_` de un banco de datos BioRS para que la consulta se envíe al servidor BioRS. Las consultas que incluyen un operador de igualdad, pero que no hacen referencia al elemento `_ID_` no se envían para que las procese el servidor BioRS.

Puede utilizar el operador de igualdad en una expresión literal. Por ejemplo:

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

Puede utilizar un predicado de igualdad en una unión entre un banco de datos BioRS y otra tabla local o apodos nonBioRS. Por ejemplo:

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

Una unión entre bancos de datos BioRS debe hacer referencia al elemento `_ID_` de un banco de datos y a un elemento de tipo de referencia para el otro banco de datos.

Sin embargo, el uso de un predicado de igualdad puede devolver resultados diferentes de los esperados en estas condiciones:

Comparación insensible a las mayúsculas y minúsculas

La operación no es sensible a las mayúsculas y minúsculas. Por ejemplo, `ID='100k_rat'` coincide con las dos series siguientes:

- '100k_rat'
- '100K_RAT'

Comparación de comodines

La sentencia `ID='100K_R*'` coincide con '100K_RAT' y con '100K_RODENT.'

Prefijos de banco de datos

La operación devuelve un prefijo que indica el banco de datos origen. Por ejemplo, `ID='100K_RAT'` en una unión del banco de datos SwissProt puede devolver el valor 'swissprot:100K_RAT.'

Nota: No cree aplicaciones que dependan de alguno de los comportamientos descritos.

El ejemplo siguiente ilustra el comportamiento de un predicado de igualdad en una unión.

La tabla local, w46851_t1, contiene los valores siguientes:

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

Puede unir la tabla w46851_t1 con un apodo w46851_n1 basado en el banco de datos SwissProt. La sentencia siguiente muestra la consulta de unión con una operación de igualdad:

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

Se devuelven los resultados siguientes:

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 registros seleccionados.

Sin embargo, el comportamiento esperado es que sólo se devuelva la fila 0.

Predicados equijoin para el derivador de BioRS

Debe especificar predicados para el motor de BioRS al utilizar las funciones personalizadas de BioRS, con una excepción. La excepción se produce al realizar operaciones equijoin durante una consulta.

Una operación *join* (unión) implica la recuperación de datos de dos o más tablas en función de valores de columna coincidentes. Una *equijoin* es una operación de unión en la que la condición de unión tiene el formato *expresión = expresión*. Para consultas BioRS, los términos de equijoin deben contener el elemento *_ID_* de un banco de datos y a un elemento de tipo Reference de otro banco de datos.

Ejemplo de definiciones de apodo y una consulta equijoin

Este ejemplo muestra definiciones de apodos de ejemplo y una consulta equijoin que utiliza los apodos de ejemplo.

El usuario desea consultar dos bancos de datos BioRS, SwissProt y SwissProt.features. El banco de datos SwissProt.features es hijo del banco de datos SwissProt y contiene un elemento denominado Parent. El elemento Parent contiene referencias a entradas identificadas por el elemento *_ID_* de SwissProt. El usuario registra dos definiciones de apodos para los dos bancos de datos.

Definición de apodo para el banco de datos SwissProt

El banco de datos SwissProt es el banco de datos padre.

```
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');
```

Definición de apodo para el banco de datos SwissProt.features

El banco de datos SwissProt.features es hijo del banco de datos SwissProt. Este apodo contiene el elemento **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 consulta que hace referencia a ambos apodos en una equijoin

En esta consulta, se aplican dos predicados al apodo tc600sprot (banco de datos SwissProt). Estos dos predicados filtran las filas que contienen el término anopheles cuyo año de publicación es 1997. Un predicado se aplica al apodo tc600feat (banco de datos SwissProt.features) que filtra las filas cuyo elemento FtKey contiene el término signal. Los dos apodos se unen mediante el término *f.Parent = s.ID*.

```

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

```

El conjunto de resultados final sólo contiene las filas que cumplen estos criterios, y las entradas del banco de datos SwissProt.features que hacen referencia a una entrada coincidente del banco de datos SwissProt.

El elemento AllText de BioRS

Todos los bancos de datos del sistema BioRS contienen un elemento denominado AllText. El elemento AllText es un elemento indexado que BioRS crea automáticamente para todos los bancos de datos.

Mediante el elemento AllText, puede realizar búsquedas en todo el texto de una entrada, no sólo en elementos indexados específicos. Por ejemplo, la búsqueda del término muscus puede devolver entradas en las que la palabra muscus aparece en el título, el resumen, la descripción o el organismo.

Para utilizar el elemento AllText en una consulta federada, debe correlacionar el elemento AllText con una columna de apodo. El elemento AllText se correlaciona con una columna de apodo al especificar columnas en la sentencia CREATE NICKNAME. Una columna de apodo correlacionada con el elemento AllText devolverá un valor NULL en sentencias SELECT. Cuando especifique una columna como elemento AllText, la columna no debe ser la primera declarada en una sentencia CREATE NICKNAME.

Una vez que el elemento AllText se ha correlacionado correctamente con una columna de apodo, puede utilizar esa columna de apodo en una función personalizada BIRS.CONTAINS.

Origen de datos BioRS - Consultas de ejemplo

Estos ejemplos incluyen un conjunto exhaustivo de consultas de ejemplo que pueden utilizarse para acceder a orígenes de datos BioRS y muestran las sentencias necesarias para crear los apodos utilizados en los ejemplos.

Estos ejemplos muestran cómo realizar las acciones siguientes:

- Estructurar las consultas para optimizar el rendimiento del sistema
- Utilizar las funciones personalizadas y comodines en las consultas
- Utilizar consultas para acceder a columnas de tipos de datos BioRS específicos
- Utilizar predicados relacionales para formar una expresión equijoin entre apodos padre e hijo

Las consultas de ejemplo utilizan los apodos swiss y swissft.

Sentencia CREATE NICKNAME para el apodo swiss

El apodo padre swiss se ha registrado para el banco de datos SwissProt utilizando la siguiente sentencia CREATE NICKNAME:

```

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 biors_server
  OPTIONS (REMOTE_OBJECT 'swissprot');

```

Sentencia CREATE NICKNAME para el apodo swissft

El apodo hijo swissft se ha registrado para el banco de datos SwissProt.Features utilizando la siguiente sentencia CREATE NICKNAME:

```

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 biors_server
  OPTIONS (REMOTE_OBJECT 'swissprot.features');

```

Impacto de las estructuras de consulta sobre el rendimiento del servidor federado

Las consultas y resultados de la tabla siguiente muestran cómo puede estructurar las consultas para optimizar la carga de trabajo entre el sistema federado y el servidor BioRS.

Tabla 9. Ejemplos de consultas diferentes que producen idénticos resultados

Consulta	Resultado
SELECT s.id FROM Swiss s WHERE BIORS.CONTAINS(s.id, '100K_RAT') = 1 FETCH FIRST 3 ROWS ONLY;	ID ----- 100K_RAT 1 registro(s) seleccionado(s).
SELECT s.id FROM Swiss s WHERE s.id LIKE '%100K_RAT%' FETCH FIRST 3 ROWS ONLY;	ID ----- 100K_RAT 1 registro(s) seleccionado(s).

Ambas consultas de esta tabla producen el mismo resultado. Sin embargo, la primera consulta se ejecuta mucho más rápido que la segunda. La primera consulta utiliza la función BIORS.CONTAINS para especificar el predicado de entrada. Como resultado, el servidor BioRS selecciona los datos del banco de datos SwissProt y luego pasa los datos seleccionados al servidor federado. En la segunda consulta, el predicado de entrada LIKE se especifica directamente en el apodo `swiss`. Como resultado, el servidor BioRS transfiere todo el banco de datos SwissProt al servidor federado. Una vez transferido el contenido del banco de datos, el servidor federado selecciona los datos.

Generalmente, el rendimiento de las consultas es mucho mejor cuando los predicados se envían al origen de datos para el proceso.

Consultas que utilizan comodines en la función personalizada BIORS.CONTAINS

Las consultas y resultados de la tabla siguiente muestran la utilización de caracteres comodín en la función personalizada BIORS.CONTAINS. Todos los resultados de las consultas son idénticos, aunque se utilizan caracteres comodín diferentes.

Tabla 10. Consultas de ejemplo que utilizan comodines en la función personalizada BIORS.CONTAINS

Consulta	Resultado
SELECT s.crossreference FROM Swiss s WHERE BIORS.CONTAINS(s.crossreference, 'MEDLINE') = 1 FETCH FIRST 3 ROWS ONLY;	CROSSREFERENCE ----- NCBI_TaxID=1011 NCBI_TaxID=5875 NCBI_TaxID=4081 3 registros seleccionados.
SELECT s.crossreference FROM Swiss s WHERE BIORS.CONTAINS(s.crossreference, '?ED?IN?') = 1 FETCH FIRST 3 ROWS ONLY;	CROSSREFERENCE ----- NCBI_TaxID=1011 NCBI_TaxID=5875 NCBI_TaxID=4081 3 registros seleccionados.

Tabla 10. Consultas de ejemplo que utilizan comodines en la función personalizada BIOR.S.CONTAINS (continuación)

Consulta	Resultado
SELECT s.crossreference FROM Swiss s WHERE BIORS.CONTAINS(s.crossreference, '*D*N*') = 1 FETCH FIRST 3 ROWS ONLY;	CROSSREFERENCE ----- NCBI_TaxID=1011 NCBI_TaxID=5875 NCBI_TaxID=4081 3 registros seleccionados.

Consultas que acceden a columnas del tipo de datos Author de BioRS

Las consultas y resultados de la tabla siguiente muestran cómo acceder a información de elementos de tipo de datos Author BioRS con la función personalizada BIOR.S.CONTAINS. La sintaxis de todas las consultas es prácticamente idéntica. La única diferencia es la presencia o ausencia de la primera inicial en el término de la consulta y la cantidad de espacio entre el nombre de pila y la última inicial.

Tabla 11. Consultas de ejemplo que acceden a columnas del tipo de datos Author de BioRS

Consulta	Resultado
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 registros seleccionados.
SELECT s.authors FROM Swiss s WHERE BIORS.CONTAINS(s.authors, 'Mueller,D') = 1 FETCH FIRST 3 ROWS ONLY;	AUTHORS ----- 0 registro(s) seleccionado(s).
SELECT s.authors FROM Swiss s WHERE BIORS.CONTAINS(s.authors, 'Mueller ,D') = 1 FETCH FIRST 3 ROWS ONLY;	AUTHORS ----- 0 registro(s) seleccionado(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 registros seleccionados.

Consultas que acceden a columnas del tipo de datos Date de BioRS

Las consultas y resultados de la tabla siguiente muestran cómo acceder a información de elementos de tipo Date BioRS con la función personalizada BIOR.S.CONTAINS.

Cuando un campo de tipo Date de BioRS contiene una secuencia de fechas, los resultados pueden contener información adicional, como se muestra en el segundo ejemplo de la tabla. Los elementos de tipo de datos Numeric de BioRS (Date y Number) pueden contener varios valores. Por tanto, los resultados de las consultas

que se ejecutan en elementos BioRS Date o Number también pueden contener varios valores. Los diversos valores están siempre separados por espacios.

Tabla 12. Consultas de ejemplo que acceden a columnas del tipo de datos Date de BioRS

Consulta	Resultado
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 registros seleccionados.
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 registros seleccionados.

Consultas que utilizan las funciones personalizadas BIORS.CONTAINS_LE y BIOR.S.CONTAINS_GE

Las consultas y resultados de la tabla siguiente muestran cómo utilizar las funciones personalizadas BIOR.S.CONTAINS_LE y BIOR.S.CONTAINS_GE.

Tabla 13. Consultas de ejemplo que utilizan las funciones personalizadas BIOR.S.CONTAINS_LE y BIOR.S.CONTAINS_GE

Consulta	Resultado
SELECT s.molweight FROM Swiss s WHERE BIORS.CONTAINS_LE(s.molweight, 100368) = 1 FETCH FIRST 3 ROWS ONLY;	MOLWEIGHT ----- 100368 10576 8523 3 registros seleccionados.
SELECT s.molweight FROM Swiss s WHERE BIORS.CONTAINS_GE(s.molweight, 100368) = 1 FETCH FIRST 3 ROWS ONLY;	MOLWEIGHT ----- 100368 103625 132801 3 registros seleccionados.
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 registros seleccionados.

Consultas que utilizan la función personalizada BIOR.S.SEARCH_TERM

Las consultas y resultados de la tabla siguiente muestran cómo utilizar la función personalizada BIOR.S.SEARCH_TERM para especificar un término de búsqueda utilizando el formato BioRS.

Tabla 14. Consultas de ejemplo que utilizan la función personalizada B.IORS.SEARCH_TERM

Consulta	Resultado
SELECT s.publicationyear FROM Swiss s WHERE B.IORS.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 registros seleccionados.
SELECT s.molweight FROM Swiss s WHERE B.IORS.SEARCH_TERM (s.id, '[MolWeight EQ number:100368;]') = 1 FETCH FIRST 10 ROWS ONLY;	MOLWEIGHT ----- 100368 100368 2 registros seleccionados.
SELECT s.molweight FROM Swiss s WHERE B.IORS.SEARCH_TERM (s.id, '[MolWeight GREATER number:100368;]') = 1 FETCH FIRST 10 ROWS ONLY;	MOLWEIGHT ----- 100368 103625 132801 194328 130277 287022 289130 135502 112715 112599 10 registros seleccionados.

Utilización de predicados relacionales para formar una expresión equijoin entre dos bancos de datos que tienen una relación padre-hijo

La consulta siguiente muestra cómo utilizar predicados relacionales para formar una expresión equijoin entre dos bancos de datos que tienen una relación padre-hijo:

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

En los resultados de consulta siguientes, el registro 100K_RAT es el padre de nueve registros hijos (100K_RAT.1 a 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 registros seleccionados.

Optimizar el rendimiento del derivador de BioRS

Puede mejorar el rendimiento de las consultas a los orígenes de datos BioRS optimizando el rendimiento del derivador de BioRS.

Directrices para optimizar el rendimiento del derivador BioRS

La estructura de las consultas y la información estadística acerca de los bancos de datos BioRS repercuten en el rendimiento de las consultas.

Minimice la cantidad de datos que se transfieren entre los motores de búsqueda.

El entorno federado utiliza dos motores de búsqueda. Para el derivador BioRS, estos motores de búsqueda son la base de datos federada y BioRS. Algunas configuraciones incluyen más de un motor de base de datos federada. El motor de base de datos federada procesa predicados (operadores relacionales, como =, BETWEEN, LIKE y <>) especificados en las columnas de apodo. El motor BioRS procesa predicados que se especifican utilizando cuatro funciones personalizadas para el derivador BioRS.

Para minimizar la cantidad de datos que se transfieren entre dos motores de búsqueda, estructure las consultas de modo que los datos se procesen en el sistema BioRS siempre que sea posible.

Si necesita realizar operaciones de unión en una consulta, aproveche alguna de las relaciones padre-hijo que ya existen en los bancos de datos BioRS y realice operaciones equijoin siempre que sea posible. Las operaciones equijoin se procesan en BioRS, que también minimiza la cantidad de datos que se transfieren entre los motores de búsqueda de base de datos federada y BioRS.

Importante: No interrumpa las consultas federadas a BioRS, por ejemplo mediante Control-D o Control-Z en el procesador de línea de mandatos, ni deteniendo el programa de aplicación. Cuando se interrumpen consultas, quedan procesos "muertos" en ejecución en el servidor BioRS. Estos procesos "muertos" degradarán rápidamente el rendimiento del servidor BioRS y del servidor federado. Si el número de procesos "muertos" en ejecución es elevado, pueden producirse errores inesperados durante el proceso de consultas federadas. Por ejemplo, una consulta válida puede devolver 0 filas, aun cuando se esperen filas. En situaciones extremas, el servidor BioRS, el servidor federado o ambos servidores pueden detenerse o finalizar de forma anómala.

Mantenga la información estadística de BioRS en el entorno federado.

En un sistema federado, la base de datos federada se basa en estadísticas de catálogo para objetos con apodo para optimizar el proceso de las consultas. Es muy importante mantener las estadísticas sobre los orígenes de datos BioRS actualizadas para optimizar el rendimiento del derivador BioRS. Si los datos estadísticos o las características estructurales para un objeto remoto en el que se define un apodo cambian, deberá actualizar las estadísticas de cardinalidad de la columna de apodo correspondiente en el sistema federado.

Para optimizar el rendimiento del derivador BioRS, realice estas actualizaciones periódicamente en el servidor federado.

Información estadística de BioRS

La información estadística de BioRS actual es esencial para optimizar el rendimiento del derivador BioRS.

En un sistema federado, para optimizar el proceso de las consultas la base de datos federada se basa en las estadísticas de catálogo para los objetos con apodo. Estas estadísticas se recuperan de los orígenes de datos BioRS cuando se registra un apodo mediante la sentencia CREATE NICKNAME. La base de datos federada verifica la presencia del objeto en el origen de datos y, a continuación, intenta recopilar los datos estadísticos existentes de el origen de datos. La información se lee de los catálogos de orígenes de datos y se coloca en el catálogo del sistema de la base de datos federada.

Para orígenes de datos BioRS, la información estadística de datos críticos incluye:

- La cardinalidad de un apodo. Para orígenes de datos BioRS, la cardinalidad de un apodo equivale al número de entradas del banco de datos BioRS correspondiente.
- La cardinalidad de la columna que corresponde al elemento BioRS_ID_. La cardinalidad de esta columna debe coincidir con la cardinalidad del apodo en el que se hace referencia a la columna.
- La cardinalidad de todas las columnas que el derivador BioRS pueda necesitar.

Consejo: Para optimizar el rendimiento del derivador BioRS, debe mantener estadísticas actualizadas acerca de los orígenes de datos BioRS. Si los datos estadísticos o las características estructurales cambian para un objeto remoto en el que se ha definido un apodo, debe actualizar las estadísticas de cardinalidad correspondientes en el sistema federado. Las estadísticas de cardinalidad se almacenan en las vistas SYSSTAT.TABLES y SYSSTAT.COLUMNS del catálogo del sistema de bases de datos federado.

Para mantener las estadísticas de cardinalidad BioRS en el sistema federado:

1. Determine las estadísticas de cardinalidad para el apodo, si es necesario.
2. Actualice las estadísticas de cardinalidad para el apodo en las vistas del catálogo.
3. Actualice las estadísticas de cardinalidad para las columnas en las vistas del catálogo.

Determinación de las estadísticas de cardinalidad de un banco de datos de BioRS

Debe determinar las estadísticas de cardinalidad de un banco de datos de BioRS para poder actualizar las estadísticas de apodos o actualizar la cardinalidad de la columna correspondiente al elemento _ID_ de BioRS.

Procedimiento

Para determinar las estadísticas de cardinalidad de un banco de datos específico de BioRS:

Utilice el programa de utilidad admin_find de BioRS o el programa de utilidad www_find.cgi y especifique la opción -c, destinada a la cardinalidad. Para obtener más información acerca de estos dos programas de utilidad de BioRS, consulte su documentación de BioRS.

Actualización de las estadísticas de cardinalidad de apodos de BioRS

Debe actualizar las estadísticas de cardinalidad de apodos cuando el contenido de un banco de datos de BioRS cambia de forma significativa.

Antes de empezar

Debe determinar el número de cardinalidad del banco de datos de BioRS correspondiente al apodo cuyas estadísticas desea actualizar.

Acerca de esta tarea

Para actualizar las estadísticas de cardinalidad de apodos de BioRS en el sistema federado, debe modificar la vista del catálogo SYSSTAT.TABLES.

El mantenimiento de estadísticas de cardinalidad correctas para los apodos permite al optimizador y al derivador de BioRS elegir el mejor plan de acceso a datos de rendimiento.

Procedimiento

Para actualizar las estadísticas de cardinalidad de apodos de BioRS:

Emita la sentencia UPDATE para modificar la vista del catálogo SYSSTAT.TABLES y especifique el número de cardinalidad correcto. La sintaxis de la sentencia UPDATE es la siguiente:

```
UPDATE SYSSTAT.TABLES SET CARD=número_cardinalidad
    WHERE TABSCHEMA=esquema_apodo
    AND TABNAME=nombre_apodo;
```

Por ejemplo, si el apodo es JONES.SWISS, utilizará la siguiente sentencia UPDATE para actualizar las estadísticas:

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

SYSSTAT.TABLES

La vista de catálogo del sistema en la base de datos federada donde se almacenan las estadísticas de apodos.

SET CARD=15312191

El número de cardinalidad del banco de datos de BioRS que corresponde al apodo para el que actualiza las estadísticas.

TABSCHEMA= 'JONES'

El nombre del esquema para el apodo que desea actualizar.

TABNAME='SWISS'

El nombre del apodo que desea actualizar.

Actualización de las estadísticas de cardinalidad de columnas de BioRS

Para actualizar las estadísticas de cardinalidad de columnas de BioRS en el sistema federado, debe modificar la vista del catálogo SYSSTAT.COLUMNS.

Puede actualizar las estadísticas de cardinalidad de columnas de BioRS antes de crear los apodos de BioRS o actualizar las estadísticas de cardinalidad de columnas de BioRS cuando desee mejorar el rendimiento de las consultas para orígenes de datos BioRS.

Restricciones

No utilice este procedimiento para actualizar las estadísticas de cardinalidad para columnas que correspondan al elemento `_ID_` de BioRS.

Debe asegurarse de que las estadísticas de cardinalidad para columnas de BioRS estén actualizadas para que el optimizador y el derivador de BioRS puedan elegir el mejor plan de acceso a datos durante el proceso de las consultas.

Acerca de esta tarea

Procedimiento

Para actualizar las estadísticas de cardinalidad de columnas de BioRS:

Emita la sentencia UPDATE para modificar la vista de catálogo SYSSTAT.COLUMNS. La sintaxis de la sentencia UPDATE es la siguiente:

```
UPDATE SYSSTAT.COLUMNS SET COLCARD=(SELECT COUNT(DISTINCT nombre_columna)
    FROM esquema_apodo.nombre_apodo)
    WHERE
    TABSCHEMA=esquema_apodo
    AND TABNAME=nombre_apodo
    AND COLNAME=nombre_columna;
```

SYSSTAT.COLUMNS

La vista de catálogo del sistema en la base de datos federada donde se almacenan las estadísticas de columna.

COLCARD=(SELECT COUNT(DISTINCT *nombre_columna*

El nombre de la columna del apodo para el que actualiza las estadísticas.

TABSCHEMA= *esquema_apodo*

El nombre del esquema para el apodo que desea actualizar.

TABNAME=*nombre_apodo*

El nombre del apodo que desea actualizar.

COLNAME=*nombre_columna*

El nombre de la columna cuyas estadísticas de cardinalidad desea actualizar.

La consulta puede tardar varios minutos en ejecutarse, ya que deben recuperarse todas las entradas del banco de datos asociado con el apodo.

Si una columna contiene varios valores (por ejemplo, el elemento `PublicationYear` del formato de la base de datos SwissProt), el cálculo es demasiado complejo para utilizar una consulta SQL. Para columnas de este tipo, debe calcular manualmente el valor de cardinalidad y luego actualizar la vista de catálogo

SYSSTAT.COLUMNS. Para calcular el valor de cardinalidad, divida el número de valores diferentes de la columna por el promedio de valores por fila. El valor de cardinalidad calculado no puede ser superior a la cardinalidad de la tabla.

Por ejemplo, si el apodo tiene las tres filas siguientes de valores para la columna `PublicationYear`, existen nueve valores diferentes y el promedio de valores de una fila es de cuatro.

- 1997 1992 1985

- 1997 1992 1982
- 1992 1991 1990 1976 1974 1971

La cardinalidad de esta columna PublicationYear es de 9 dividido por 4, o 3 (2,25 redondeado en el entero mayor). Puede actualizar la vista de catálogo SYSSTAT.COLUMNS mediante la siguiente sentencia UPDATE:

```
UPDATE SYSSTAT.COLUMNS SET CARDCOL=3
WHERE
  TABSCHEMA=esquema_apodo
  AND TABNAME=nombre_apodo
  AND COLNAME=nombre_columna;
```

Actualización de la cardinalidad de la columna `_ID_` de BioRS

Para actualizar las estadísticas de cardinalidad de columnas de BioRS para la columna que se correlaciona con el elemento `_ID_` de BioRS, debe modificar la vista del catálogo SYSSTAT.COLUMNS.

Antes de empezar

Debe determinar el número de cardinalidad del banco de datos de BioRS correspondiente al apodo en el que se hace referencia a la columna. El número de cardinalidad de la columna que se correlaciona con el elemento `_ID_` de BioRS debe coincidir con la cardinalidad del apodo en el que se hace referencia a la columna.

Debe asegurarse de que las estadísticas de cardinalidad de la columna que se correlaciona con el elemento `_ID_` de BioRS están actualizadas. El optimizador y el derivador de BioRS utilizan estas estadísticas para elegir el mejor plan de acceso a datos para procesar las consultas.

Acerca de esta tarea

Para actualizar la cardinalidad de la columna `_ID_` de BioRS, debe seleccionar entradas de la vista SYSCAT.COLOPTIONS que contengan la opción `ELEMENT_NAME`. Este derivador de BioRS utiliza esta opción para la correlación entre los nombres de columna de apodo de la base de datos federada y los nombres de elemento del servidor BioRS.

Procedimiento

Para actualizar las estadísticas de cardinalidad de la columna `_ID_` de BioRS:

Emita la sentencia UPDATE para modificar la vista de catálogo.

Por ejemplo:

```
UPDATE SYSSTAT.COLUMNS SET COLCARD=número_cardinalidad
WHERE
  TABSCHEMA=esquema_apodo
  AND TABNAME=nombre_apodo
  AND COLNAME=nombre_columna
  IN (SELECT nombre_columna FROM SYSCAT.COLOPTIONS
      WHERE
        TABSCHEMA=esquema_apodo
        AND TABNAME=nombre_apodo
        AND OPTION='ELEMENT_NAME';
      AND SETTING='_ID_')
```

SYSSTAT.COLUMNS

La vista de catálogo del sistema en la base de datos federada donde se almacenan las estadísticas de columna.

SET COLCARD=*número_cardinalidad*

El número de cardinalidad del banco de datos de BioRS que corresponde al apodo de la columna para la que actualiza las estadísticas.

TABSCHEMA= *esquema_apodo*

El nombre del esquema para el apodo que desea actualizar.

TABNAME=*nombre_apodo*

El nombre del apodo que desea actualizar.

COLNAME=*nombre_columna*

El nombre de la columna cuyas estadísticas de cardinalidad desea actualizar.

IN (SELECT *nombre_columna* FROM SYSCAT.COLOPTIONS

Esta sentencia SELECT determina el nombre de la columna que se correlaciona con el elemento *_ID_* de BioRS. SYSCAT.COLOPTIONS es la vista de catálogo del sistema en la base de datos federada donde se almacenan las opciones de columna.

OPTION='ELEMENT_NAME'

El valor de las filas de la vista SYSCAT.COLOPTIONS que indica que un nombre de columna de apodo está correlacionado con un nombre de elemento de BioRS.

SETTING='_ID_'

Especifica que la columna para la opción ELEMENT_NAME es '_ID_'.

Configuración del acceso a orígenes de datos DB2

Para configurar un servidor federado para acceder a orígenes de datos de la familia DB2, debe proporcionar al servidor federado información acerca de los orígenes de datos y objetos a los que desea acceder.

Antes de empezar

- Compruebe la configuración del servidor federado.

Procedimiento

Para configurar el servidor federado para acceder a orígenes de datos DB2, realice las tareas siguientes:

1. Catalogue una entrada de nodo DB2.
2. Catalogue la base de datos DB2 remota.
3. Registre el derivador DB2.
4. Registre el origen de datos DB2 del servidor.
5. Cree el origen de datos DB2 del usuario.
6. Pruebe la conexión con el servidor de orígenes de datos DB2.
7. Registre los apodos para las tablas y vistas de DB2.

Catalogación de una entrada de nodo de DB2

Debe catalogar una entrada de nodo que especifique el protocolo que el servidor federado utiliza para conectarse con el origen de datos DB2.

Procedimiento

Para catalogar una entrada de nodo:

Emita el mandato CATALOG TCPIP NODE.

Por ejemplo:

```
CATALOG TCPIP NODE nodo_db2 REMOTE system42 SERVER db2tcp42
```

donde:

- *nodo_db2* es el nombre que se asigna al nodo
- *system42* es el nombre de sistema principal donde reside el origen de datos
- *db2tcp42* es el nombre de servicio o el número de puerto primario de la instancia de gestor de bases de datos de servidor

Catalogación de la base de datos de DB2 remota

Para identificar la base de datos con la que se conecta el servidor federado, debe catalogar la base de datos de DB2 remota en el directorio de bases de datos de sistemas servidores federados.

Procedimiento

Para catalogar la base de datos remota:

1. Utilice el Asistente de configuración para catalogar la base de datos o emita el mandato CATALOG DATABASE.

Por ejemplo:

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

donde:

- *DB2DB390* es el nombre de la base de datos remota que debe catalogarse
 - *CLIENTS390* es el alias para la base de datos remota que se está catalogando. Si no especifica ningún alias, el gestor de bases de datos utiliza el nombre de la base de datos remota como alias.
 - *DB2NODE* es el nombre del nodo que ha catalogado previamente
 - AUTHENTICATION SERVER especifica que la autenticación tiene lugar en el nodo del origen de datos DB2
2. Si el nombre de la base de datos remota tiene más de ocho caracteres, emita el mandato CATALOG DCS DATABASE para crear una entrada de directorio DCS.

Por ejemplo:

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

donde:

- *SALES400* es el alias de la base de datos remota que se va a catalogar. El alias debe coincidir con el nombre de una entrada del directorio de bases de datos de sistemas servidores federados que esté asociada con el nodo remoto. El alias es el mismo nombre que ha especificado en el mandato CATALOG DATABASE.
- *SALES_DB2DB400* es el nombre de la base de datos de sistema principal de destino que desea catalogar.

Registro del derivador de DB2

Debe registrar un derivador para acceder a los orígenes de datos de la familia DB2. El servidor federado utiliza el derivador para comunicarse con los orígenes de datos y para recuperar datos de éstos. Un derivador se implementa como un conjunto de archivos de biblioteca.

Procedimiento

Para registrar un derivador:

Emita la sentencia CREATE WRAPPER y especifique el nombre para el derivador. El nombre del derivador predeterminado para los orígenes de datos de la familia DB2 es DRDA.

Por ejemplo:

```
CREATE WRAPPER DRDA
```

Cuando utiliza el nombre por omisión para registrar el derivador, no necesita especificar el nombre de biblioteca porque el servidor federado automáticamente utiliza el nombre de biblioteca por omisión que está asociado con el derivador. Si el nombre del derivador entra en conflicto con un nombre de derivador existente en la base de datos federada, puede sustituir el nombre del derivador por omisión por otro de su elección. Sin embargo, si lo hace, deberá incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER.

Por ejemplo, para registrar un derivador con el nombre derivador_db2 en un servidor federado que utiliza el sistema operativo AIX, emita esta sentencia:

```
CREATE WRAPPER derivador_db2 LIBRARY 'libdb2drda.a'
```

El nombre de biblioteca por omisión es específico del sistema operativo del servidor federado. Para obtener más información, vea la lista de archivos de biblioteca del derivador de DB2.

Archivos de biblioteca de derivador de DB2

Cuando instala el servidor federado, se añaden archivos de biblioteca de derivador a la vía de acceso al directorio por omisión.

Si no utiliza el nombre de derivador por omisión cuando registra un derivador, debe incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER y especificar el nombre de archivo de biblioteca del derivador por omisión.

Esta tabla lista las vías de acceso al directorio por omisión y los nombres de archivo de biblioteca del derivador por omisión. En la tabla, *install_path* es la vía de acceso al directorio donde se instala el servidor en UNIX o Linux y *%DB2PATH%* es la variable de entorno que especifica la vía de acceso al directorio donde se instala el servidor federado en Microsoft Windows. La vía de acceso al directorio de Windows por omisión es C:\Archivos de programa\IBM\sqliib.

Tabla 15. Vías de acceso al directorio y nombres de archivo de biblioteca de DB2 por sistema operativo

Sistema operativo	Vía de acceso al directorio	Nombre de archivo de biblioteca
AIX	/usr/opt/ <i>install_path</i> /lib32/ /usr/opt/ <i>install_path</i> /lib64/	libdb2drda.a
HP-UX	/opt/IBM/db2/ <i>install_path</i> /lib32 /opt/IBM/db2/ <i>install_path</i> /lib64	libdb2drda.so
Linux	/opt/IBM/db2/ <i>install_path</i> /lib32 /opt/IBM/db2/ <i>install_path</i> /lib64	libdb2drda.so
Solaris	/opt/IBM/db2/ <i>install_path</i> /lib32 /opt/IBM/db2/ <i>install_path</i> /lib64	libdb2drda.so
Windows	%DB2PATH%\bin	db2drda.dll

Para cada sistema operativo, se instalan tres archivos de biblioteca; un archivo de biblioteca por omisión y dos archivos adicionales que sólo se utilizan con opciones de derivador específicas.

Registro de definiciones de servidor para orígenes de datos DB2

El servidor federado necesita la información de autorización y de contraseña para conectar con cada servidor DB2. Dado que esta información de autorización y de contraseña no se almacena en el catálogo global, debe incluirla en cada definición de servidor.

Procedimiento

Para registrar una definición de servidor, utilice uno de los métodos siguientes:

- Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database. Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados.
- Emita la sentencia CREATE SERVER.

Cuando registra el servidor, debe incluir determinadas opciones de servidor necesarias. Este ejemplo sólo incluye las opciones de servidor que son necesarias para registrar un servidor DB2:

```
CREATE SERVER nombre_definición_servidor TYPE tipo_servidor
VERSION númeroversión WRAPPER DRDA
AUTHORIZATION "id_usuario" PASSWORD "contraseña"
OPTIONS (DBNAME 'nombre_base de datos')
```

DBNAME es una opción de servidor necesaria. El valor de DBNAME es el alias para la base de datos de DB2 a la que desea acceder. El alias se define cuando se cataloga la base de datos.

Nota: Para VERSION, si ha utilizado DB2 para z/OS Versión 8 para crear la base de datos en modalidad de compatibilidad, debe especificar la Versión 7. Cuando registra el servidor, puede especificar opciones de servidor adicionales en la sentencia CREATE SERVER. Estas opciones incluyen opciones de servidor generales y opciones de servidor específicas del origen de datos. Para obtener más información, vea la información de consulta sobre opciones de DB2.

Después de registrar el servidor, utilice la sentencia ALTER SERVER para añadir opciones de servidor adicionales o eliminar opciones de servidor existentes

Sentencia CREATE SERVER - Ejemplos para el derivador de DB2

Utilice la sentencia CREATE SERVER para registrar definiciones del servidor DB2. Este tema incluye un ejemplo completo con las opciones necesarias y un ejemplo que muestra el uso de opciones adicionales del servidor.

Ejemplo completo

El ejemplo siguiente muestra cómo registrar una definición de servidor para un derivador DRDA mediante la utilización de la sentencia CREATE SERVER:

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

DB2SERVER

Nombre que asigna al servidor de bases de datos DB2. No se permiten los nombres de definición de servidor duplicados. Esta es una opción de servidor obligatoria.

TYPE *DB2/ZOS*

Especifica el tipo de servidor de orígenes de datos para el que está configurando el acceso.

VERSIÓN 7

Es la versión del servidor de bases de datos de DB2 al que desea acceder.

Nota: Si ha utilizado DB2 para z/OS Versión 8 para crear la base de datos en modalidad de compatibilidad, debe especificar la Versión 7.

WRAPPER *DRDA*

Es el nombre que ha especificado en la sentencia CREATE WRAPPER.

AUTHORIZATION *"spalten"*

Es el ID de autorización en el origen de datos. Este ID debe tener autorización BINDADD en el origen de datos. Este valor es sensible a las mayúsculas y minúsculas.

PASSWORD *"db2guru"*

Es la contraseña asociada al ID de autorización en el origen de datos. Este valor es sensible a las mayúsculas y minúsculas.

DBNAME *'CLNTS390'*

Es el alias para la base de datos DB2 a la que desea acceder. Este alias se definió al catalogar la base de datos mediante el mandato CATALOG DATABASE. Este valor es sensible a las mayúsculas y minúsculas.

Aunque la variable de nombre de la base de datos se especifica como una opción en la sentencia CREATE SERVER, se necesita para los orígenes de datos de DB2.

Ejemplo de opción de servidor

Cuando registre la definición de servidor, podrá especificar opciones de servidor adicionales en la sentencia CREATE SERVER. Estas opciones incluyen las opciones de servidor generales y las opciones de servidor específicas del origen de datos de DB2. Para obtener más información, lea la información de consulta sobre opciones.

La opción CPU_RATIO indica la rapidez o lentitud con que la CPU del origen de datos ejecuta la CPU federada. Si establece la opción CPU_RATIO en '0,001', indica que la CPU del origen de datos remoto tiene 1000 veces más capacidad disponible que la CPU del servidor federado.

SAME_DECFLT_ROUNDING especifica si la modalidad de redondeo tanto del servidor federado como del servidor remoto utilizan el mismo valor de modalidad de redondeo DECFLOAT. Si establece la opción SAME_DECFLT_ROUNDING en 'Y', especifica que los valores de modalidad de redondeo DECFLOAT son los mismos en ambos servidores.

Por ejemplo:

```
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')
```

Creación de correlaciones de usuario para orígenes de datos DB2

Una correlación de usuarios define una asociación entre un ID de usuario y una contraseña en el servidor federado y el ID de usuario y la contraseña correspondientes en el servidor de orígenes de datos.

El hecho de que las correlaciones de usuario sean necesarias para orígenes de datos DB2 depende de la configuración del entorno federado. Si el entorno utiliza contextos fiables federados y autenticación de proxy, es posible que no se requiera ninguna correlación de usuarios o sólo muy pocas. Para obtener los mejores resultados, planifique y configure los contextos fiables federados antes de crear correlaciones de usuario.

Procedimiento

Para correlacionar el ID de usuario local al ID de usuario y contraseña del servidor de DB2:

Emita una sentencia CREATE USER MAPPING.

Por ejemplo:

```
CREATE USER MAPPING FOR IDusuario_local SERVER nombre_definición_servidor
    OPTIONS (REMOTE_AUTHID 'IDusuario_remoto', REMOTE_PASSWORD 'contraseña_remota')
```

REMOTE_AUTHID es el ID de autorización de conexión, no el ID de autorización de vinculación.

Sentencia CREATE USER MAPPING - Ejemplos para el derivador de DB2

Utilice la sentencia CREATE USER MAPPING para correlacionar un ID de autorización de servidor federado con un ID de usuario y una contraseña de servidor DB2.

Ejemplo completo

El ejemplo siguiente muestra cómo correlacionar un ID de autorización de servidor federado con un ID de usuario y una contraseña de DB2 remotos:

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

ALONZO

Correlaciona el ID de autorización local con el ID de usuario y la contraseña remotos.

SERVER DB2SERVER

Especifica el nombre del servidor de orígenes de datos DB2 que ha definido en la sentencia CREATE SERVER.

REMOTE_AUTHID 'al'

Especifica el ID de usuario de conexión en el servidor de orígenes de datos de la familia DB2 con el que está correlacionando *ALONZO*. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_ID en 'U' o 'L' en la sentencia CREATE SERVER.

REMOTE_PASSWORD 'day2night'

Especifica la contraseña asociada a '*al*'. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_PW en 'U' o 'L' en la sentencia CREATE SERVER.

Ejemplo de registro especial

A continuación se ofrece un ejemplo de la sentencia CREATE USER MAPPING, que incluye el registro especial USER:

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

Puede utilizar el registro especial USER de DB2 para correlacionar el ID de autorización de la persona que emite la sentencia CREATE USER MAPPING con el ID de usuario de origen de datos especificado en la opción de usuario REMOTE_AUTHID.

Prueba de la conexión al servidor de orígenes de datos DB2

Pruebe la conexión con el servidor de orígenes de datos DB2 para determinar si el servidor federado se ha configurado correctamente para acceder al servidor de orígenes de datos DB2.

Para probar la conexión con el servidor DB2, abra una sesión de paso a través y emita una sentencia SELECT de SQL en las tablas del sistema DB2.

Origen de datos DB2	Ejemplo
DB2 for z/OS	SET PASSTHRU <i>nombre_definición_servidor</i> SELECT count(*) FROM sysibm.systables SET PASSTHRU RESET
DB2 para Linux, UNIX y Windows	SET PASSTHRU <i>nombre_definición_servidor</i> SELECT count(*) FROM syscat.systables SET PASSTHRU RESET
DB2 para System i	SET PASSTHRU <i>nombre_definición_servidor</i> SELECT count(*) FROM qsys2.systables SET PASSTHRU RESET

Si la sentencia SELECT de SQL devuelve un recuento, el acceso a el origen de datos se ha configurado correctamente.

Resolución de errores de conexión de origen de datos

Una conexión de prueba con el servidor de origen de datos puede devolver un error por varios motivos. Existen acciones que permiten determinar el motivo del error.

Síntoma

Se devuelve un error al intentar establecer conexión con el origen de datos.

Causa

Hay varias causas posibles que explican un problema de conexión.

Resolución del problema

Para solucionar los problemas de errores de conexión con el origen de datos, consulte los siguientes elementos:

- Verifique que el origen de datos está disponible.
- Si procede, asegúrese de que el servidor de origen de datos esté configurado para conexiones entrantes.

- Asegúrese de que los valores de correlación de usuarios para las opciones REMOTE_AUTHID y REMOTE_PASSWORD son válidos para las conexiones con el origen de datos. Modifique la correlación de usuarios o cree otra correlación de usuarios según sea necesario.
- Si procede, asegúrese de que el software de cliente del servidor federado esté instalado y configurado correctamente para conectarse al origen de datos.
- Para orígenes de datos ODBC, asegúrese de que el controlador ODBC del servidor federado esté instalado y configurado correctamente para establecer conexión con el servidor de origen de datos ODBC. En servidores federados que ejecutan Windows, utilice la herramienta ODBC Data Source Administrator para verificar el controlador. En servidores federados que ejecutan UNIX, consulte la documentación del proveedor del cliente ODBC.
- Verifique que los valores para las variables establecidas en el servidor federado son correctas para el origen de datos. Estas variables incluyen las variables de entorno del sistema, las variables de la archivo db2dj.ini y las variables del registro de perfiles de DB2 (db2set).
- Compruebe la definición del servidor. Si es necesario, elimine la definición del servidor y créela de nuevo.

Registro de apodos para vistas y tablas de DB2

Para cada definición de servidor DB2, registre un apodo para cada tabla y vista a la que desee acceder. A continuación, utilice los apodos, no los nombres de los objetos de origen de datos, cuando consulte la base de datos DB2.

Antes de empezar

Antes de registrar un apodo, utilice el mandato DB2 RUNSTATS para actualizar las estadísticas en el origen de datos DB2. El servidor federado utiliza las estadísticas de origen de datos para optimizar el proceso de las consultas.

Restricciones

No puede crear un apodo en un alias de base de datos DB2.

Procedimiento

Para registrar un apodo:

Emita la sentencia CREATE NICKNAME.

Por ejemplo:

```
CREATE NICKNAME apodo FOR nombre_definición_servidor."esquema_remoto".
"tabla.remota"
```

Cuando se crea el apodo, el servidor federado consulta el catálogo de orígenes de datos utilizando el apodo. Esta consulta prueba la conexión al origen de datos. Si la conexión no funciona, se visualizará un mensaje de error.

Sentencia CREATE NICKNAME - Ejemplos para orígenes de datos DB2

Utilice la sentencia CREATE NICKNAME y las opciones de apodo necesarias para registrar un apodo para una tabla o vista de DB2 a la que desea acceder.

Este ejemplo muestra el uso de las opciones necesarias para la sentencia CREATE NICKNAME. También puede incluir opciones de apodo y de columna opcionales.

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

DB2SALES

Un apodo exclusivo que identifica la tabla o vista de DB2. El apodo puede incluir un esquema y el apodo. Si omite el esquema, se utiliza el ID de autorización del usuario que registra el apodo.

DB2SERVER.VINNIE.EUROPE

Es un identificador de tres partes para el objeto remoto:

- *DB2SERVER* es el nombre que ha asignado al servidor de bases de datos DB2 en la sentencia CREATE SERVER.
- *VINNIE* es el ID de usuario del propietario de la tabla o vista. Este valor es sensible a las mayúsculas y minúsculas.
- *EUROPE* es el nombre de la tabla o la vista remota a la que desea acceder.

Configuración del acceso a orígenes de datos Excel

Puede integrar los datos que se encuentran en orígenes de datos Excel con la información procedente de otros orígenes utilizando un sistema federado.

Procedimiento

Para configurar un servidor federado para acceder a orígenes de datos Excel, debe proporcionar al servidor federado información acerca de los orígenes de datos y objetos a los que desea acceder. Después de configurar el servidor federado, puede crear consultas para acceder a los orígenes de datos Excel.

Derivador Excel

Eu archivo de libro de trabajo Excel es un archivo creado mediante Microsoft Excel y tiene la extensión xls. El derivador Excel se utiliza para realizar búsquedas en archivos Excel.

Los archivos Excel se utilizan para almacenar información cuya visualización óptima es en una tabla, con las filas y columnas correspondientes. Los libros de trabajo Excel están formados por una o más páginas de hoja de cálculo u *hojas de trabajo*. Las hojas de trabajo se utilizan a menudo para realizar cálculos.

En la figura siguiente se muestra cómo conecta el derivador Excel las hojas de trabajo con el sistema federado.

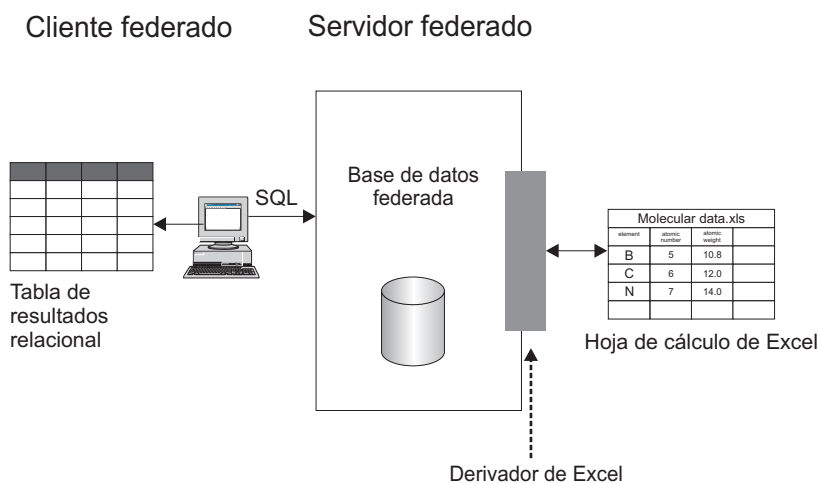


Figura 3. Funcionamiento de derivador Excel

El derivador Excel utiliza la sentencia CREATE NICKNAME para correlacionar las columnas de las hojas de trabajo Excel con las columnas del sistema federado. En la tabla siguiente se muestra un ejemplo de datos de hoja de trabajo en un archivo denominado Compound_Master.xls.

Tabla 16. Hoja de trabajo de ejemplo para Compound_Master.xls

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

La información de una hoja de trabajo Excel no suele estar disponible a través de mandatos SQL estándar. Cuando se instala y registra el derivador Excel en el servidor federado, es posible acceder a esta información como si fuera un origen de datos relacionales típico. Por ejemplo, si quisiera conocer todos los datos compuestos cuyo recuento molecular es superior a 100, ejecutaría la siguiente consulta SQL:

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

Los resultados de la consulta se muestran en la tabla siguiente.

Tabla 17. Resultados de la consulta

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

Métodos de acceso a datos Excel

Puede acceder a los datos de las hojas de trabajo de Microsoft Excel utilizando un derivador Excel o el derivador ODBC.

Para consultar datos Excel, ambos derivadores requieren un servidor federado que pueda abrir y leer las hojas de trabajo del libro de trabajo Excel. Por consiguiente, el libro de trabajo Excel debe encontrarse en el mismo sistema que el servidor federado o en una unidad accesible a través de la red.

Si utiliza el derivador Excel, la aplicación Excel debe estar instalada en el servidor federado.

Si utiliza el derivador ODBC, el controlador Excel ODBC debe estar en el servidor federado. Este controlador se instala automáticamente con Microsoft Windows®. No es necesario que la aplicación Excel esté instalada en el servidor federado.

Cada derivador impone algunos requisitos en la ubicación y el diseño de los datos de los libros de trabajo Excel. Con el derivador Excel, sólo es posible acceder a los datos de la primera hoja de trabajo del libro de trabajo. Con el derivador ODBC, puede acceder a datos de cualquier hoja de trabajo del libro de trabajo.

En los ejemplos siguientes se muestran los requisitos de diseño para estos dos derivadores.

Ejemplo de hoja de trabajo que contiene filas de etiquetas y una fórmula

Este ejemplo muestra una hoja de trabajo que contiene varias filas de etiquetas en la parte superior de la hoja de trabajo, filas en blanco y una fórmula en la fila 13. Para acceder a los datos de la hoja de trabajo, debe identificar el rango de células a las que desea acceder.

	A	B	C	D
1	Análisis de compuestos			
2				
3	Nombre del compuesto	Peso	Recuento molecular	¿Probado?
4	compuesto_A	1,23	367	probado
5	compuesto_G		210	
6	compuesto_F	0,000425536	174	probado
7	compuesto_Y	1,000256		probado
8	compuesto_Q		1024	
9	compuesto_B	33,5362		
10	compuesto_S	0,96723	67	probado
11	compuesto_O	1,2		probado
12				
13	Total de compuestos probados			5

Figura 4. Hoja de trabajo que contiene varias filas de etiquetas y una fórmula

Si utiliza el derivador Excel

Debe especificar el rango de células en la sentencia CREATE mediante la opción RANGE. Incluya sólo los datos del rango que especifique. No incluya ninguna etiqueta de columna en el rango. Las células que contienen fórmulas, como SUM, devuelven el resultado de la fórmula, no la fórmula. A menos que desee que se devuelvan los resultados de la fórmula, no incluya en el rango las celdas que contienen fórmulas. En este ejemplo, el rango de células que se incluye en la opción RANGE es A4:D11.

Si utiliza el derivador ODBC

Debe crear un nombre para el rango de celdas para designar explícitamente la ubicación de los datos en la hoja de trabajo. Excel hace referencia a este rango de celdas como *rango con nombre*. El controlador Excel ODBC reconoce sólo una fila de etiquetas, la primera del rango. No se permite ninguna fila en blanco entre las etiquetas y los datos. El rango con nombre debe incluir una fila de etiquetas de columna. Debe especificar el rango con nombre en la sentencia CREATE NICKNAME. En el rango que indique, debe incluir una fila de etiquetas de columna. Si no incluye una fila de etiquetas de columna en el rango con nombre, la primera fila de datos se tratará como etiquetas de columna. Las células que contienen fórmulas, como SUM, devuelven el resultado de la fórmula, no la fórmula. A menos que desee que se devuelvan los resultados de la fórmula, no incluya en el rango las celdas que contienen fórmulas. En este ejemplo, el rango de celdas que se indica es A3:D11.

Ejemplo de hoja de trabajo que contiene una fila de etiquetas

Este ejemplo muestra una hoja de trabajo que sólo contiene una fila de etiquetas de columna en la parte superior de la hoja de trabajo. El diseño no incluye filas adicionales con etiquetas, filas en blanco ni celdas con fórmulas.

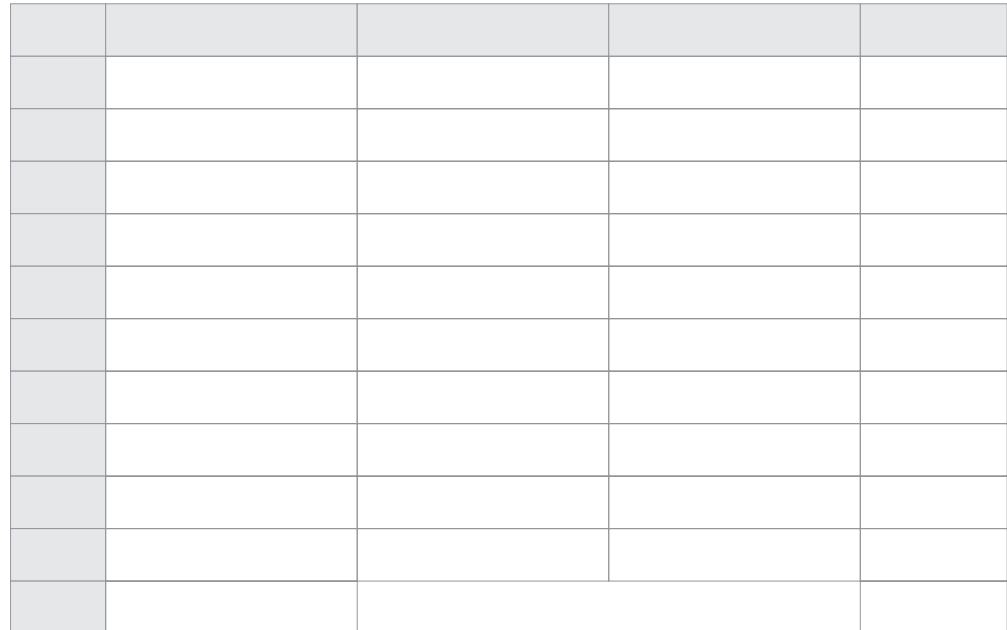


Figura 5. Hoja de trabajo que contiene una fila de etiquetas de columna en la fila 1

Si utiliza el derivador Excel

Debe especificar el rango de células en la sentencia CREATE mediante la opción RANGE. El rango no puede incluir las etiquetas de columna de la fila 1. El rango de celdas que especificaría es A2:D9.

Si utiliza el derivador ODBC

Puede acceder a estos datos sin crear un rango con nombre. El nombre de la hoja de trabajo se especifica en la sentencia CREATE NICKNAME. El derivador lee la primera fila que no está en blanco como etiquetas y utiliza la información como nombres de columna para el apodo. Las filas subsiguientes se leen como datos.

Ejemplo de hoja de trabajo que sólo contiene datos

Este ejemplo muestra una hoja de trabajo que sólo contiene datos. No hay ninguna fila para las etiquetas de columna, ninguna fila en blanco y ninguna celda con fórmulas.

	compuesto_A	1,23	367	probado
	compuesto_G		210	
	compuesto_F	0,000425536	174	probado
	compuesto_Y	1,000256		probado
	compuesto_Q		1024	
	compuesto_B	33,5362		
	compuesto_S	0,96723	67	probado
	compuesto_O	1,2		probado

Figura 6. Hoja de trabajo que sólo contiene datos

Si utiliza el derivador Excel

Si los datos se encuentran en la primera hoja de trabajo del libro de trabajo, el derivador accederá a los datos sin utilizar la opción RANGE. Si los datos se encuentran en otra hoja de trabajo del libro de trabajo, debe especificar la opción RANGE en la sentencia CREATE NICKNAME.

Si utiliza el derivador ODBC

Si utiliza el derivador ODBC para acceder a datos Excel, el derivador está limitado por aquello soportado por el controlador Excel ODBC. El controlador Excel ODBC requiere un formato específico para la hoja de trabajo. El controlador asume que la primera fila que no está en blanco contiene las etiquetas de columna. Si la primera fila que no está en blanco contiene datos, los datos de dicha fila se leen como etiquetas de columna para los demás datos. Si la hoja de trabajo no contiene ninguna fila de etiquetas de columna, la primera fila se utiliza como etiquetas y no como datos. De hecho, se pierden los datos de la primera columna. Para superar este requisito, debe modificar la hoja de trabajo. Inserte una fila antes de los datos y añada etiquetas para cada columna de datos, de modo que parezca el ejemplo que contiene una fila de etiquetas.

Adición de orígenes de datos Excel a un servidor federado

Para configurar un servidor federado para acceder a orígenes de datos Excel, debe proporcionar al servidor federado información acerca de los orígenes de datos y objetos a los que desea acceder.

Antes de empezar

- Los datos de las hojas Excel deben estar correctamente estructurados para que el derivador de Excel pueda acceder a ellos.
- Federation debe estar instalado en el servidor que actuará como servidor federado.
- Debe existir una base de datos en el servidor federado.

Restricciones

- El derivador de Excel sólo está disponible para versiones del sistema operativo Microsoft Windows soportadas por el servidor federado.
- La aplicación Excel debe estar instalada en el servidor federado.
- El libro Excel debe encontrarse en el mismo sistema que el servidor federado o en una unidad accesible de la red.
- El derivador de Excel sólo puede acceder a los datos de la primera hoja del libro Excel.
- El conjunto de páginas de códigos de la base de datos federada debe coincidir con el juego de caracteres de archivo de Excel; de lo contrario, las consultas pueden generar resultados inesperados.
- Las sesiones de paso a través no están permitidas.

Acerca de esta tarea

Puede configurar un servidor federado para acceder a datos almacenados en orígenes de datos Excel utilizando el Centro de control o emitiendo sentencias SQL en la línea de mandatos. El Centro de control incluye un asistente que le guiará en los pasos necesarios para configurar los objetos federados necesarios.

Procedimiento

Para añadir los orígenes de datos Excel a un servidor federado:

1. “Registro del derivador de Excel”.
2. “Registro de la definición del servidor para un origen de datos Excel” en la página 70.
3. “Registro de apodos para orígenes de datos Excel” en la página 71.

Registro del derivador de Excel

Debe registrar un derivador para acceder a los orígenes de datos Excel.

Acerca de esta tarea

Los servidores federados utilizan los derivadores para comunicarse con los orígenes de datos y para recuperar datos de ellos. Los derivadores se implementan como un conjunto de archivos de biblioteca.

Puede registrar un derivador mediante el Centro de control o desde la línea de mandatos. El Centro de control incluye un asistente que le guiará en los pasos necesarios para registrar el derivador.

Procedimiento

Para registrar el derivador de Excel:

Elija el método que desee utilizar para registrar el derivador de Excel:

Método	Procedimiento
Utilizar el Centro de control	Inicie el asistente Objetos federados. Pulse la carpeta Objetos de base de datos federada con el botón derecho del ratón y pulse Crear objetos federados .

Método	Procedimiento
Utilizar la línea de mandatos	<p>Emita la sentencia CREATE WRAPPER. Por ejemplo:</p> <pre>CREATE WRAPPER derivador_excel LIBRARY 'db2lsxls.dll';</pre> <p>Debe especificar el parámetro LIBRARY en la sentencia CREATE WRAPPER.</p>

Archivos de biblioteca del derivador de Excel:

Los archivos de biblioteca de derivador de Excel se añaden al servidor federado cuando éste se instala.

Cuando instala el servidor federado, se añaden tres archivos de biblioteca a la vía de acceso del directorio predeterminado. Por ejemplo, si el servidor federado se ejecuta en Windows, los archivos de biblioteca del derivador que se añaden a la vía de acceso del directorio son db2lsxls.dll, db2lsxlsF.dll y db2lsxlsU.dll. El archivo de biblioteca del derivador predeterminado es db2lsxls.dll. Los demás archivos de biblioteca del derivador se utilizan con opciones específicas del derivador.

Al registrar el derivador de Excel, debe incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER y especificar el nombre de archivo de biblioteca del derivador predeterminado.

En la tabla siguiente se indican la vías de acceso al directorio predeterminado y el nombre de archivo de biblioteca del derivador predeterminado.

Tabla 18. Ubicación de biblioteca y nombre de archivo del derivador de Excel

Sistema operativo	Vía de acceso al directorio	Nombre de archivo de biblioteca de derivador
Windows	%DB2PATH%\bin	db2lsxls.dll

%DB2PATH% es la variable de entorno que especifica la vía de acceso al directorio donde se ha instalado el servidor federado en Windows. La vía de acceso al directorio de Windows predeterminado es C:\Program Files\IBM\SQLLIB.

Registro de la definición del servidor para un origen de datos Excel

Debe registrar una definición de servidor debido a que la jerarquía de los objetos federados requiere que los archivos de libros Excel, identificados por apodos, estén asociados con un objeto de definición de servidor específico.

Acerca de esta tarea

Puede registrar una definición de servidor mediante el Centro de control o desde la línea de mandatos. El Centro de control incluye un asistente que le guiará en los pasos necesarios para registrar la definición de servidor.

Procedimiento

Para registrar una definición de servidor para un origen de datos Excel:

Elija el método que desee utilizar para registrar la definición del servidor:

Método	Procedimiento
Utilizar el Centro de control	Inicie el asistente Objetos federados. Pulse la carpeta Objetos de base de datos federada con el botón derecho del ratón y pulse Crear objetos federados .
Utilizar la línea de mandatos	Emita la sentencia CREATE SERVER. Por ejemplo: CREATE SERVER <i>nombre_definición_servidor</i> WRAPPER <i>derivador_excel</i> ;

Sentencia CREATE SERVER - Ejemplos para el derivador de Excel:

Utilice la sentencia CREATE SERVER para registrar definiciones de servidor para el derivador de Excel.

El ejemplo siguiente muestra cómo registrar una definición de servidor denominada *biochem_lab* para un libro que contiene datos bioquímicos. La sentencia CREATE SERVER que se emite es la siguiente:

```
CREATE SERVER biochem_lab WRAPPER derivador_excel;
```

biochem_lab

Nombre asignado por el usuario a la definición del servidor Excel. No se permiten los nombres de definición de servidor duplicados.

WRAPPER *derivador_Excel*

El nombre del derivador que ha especificado en la sentencia CREATE WRAPPER.

Registro de apodos para orígenes de datos Excel

Para cada definición de servidor Excel que registre, debe registrar un apodo para cada hoja Excel a la que desee acceder. Utilice estos apodos en lugar de las hojas cuando consulte los orígenes de datos Excel.

Acerca de esta tarea

Cuando se crea un apodo para una hoja Excel, la información de los datos de la hoja se correlaciona con una tabla relacional.

Las celdas en blanco de la hoja se interpretan como nulas (NULL).

Pueden existir hasta 10 filas en blanco consecutivas en la hoja, que se incluirán en el conjunto de datos. Más de 10 filas en blanco consecutivas se interpretan como el final del conjunto de datos.

Pueden existir columnas en blanco en la hoja. Sin embargo, estas columnas deben estar registradas y descritas como campos válidos aunque no se utilicen.

Procedimiento

Para registrar un apodo para una hoja Excel:

Elija el método que desee utilizar para registrar el apodo. Los apodos pueden tener una longitud de hasta 128 caracteres.

Método	Procedimiento
Utilizar el Centro de control	Inicie el asistente Objetos federados. Pulse la carpeta Objetos de base de datos federada con el botón derecho del ratón y pulse Crear objetos federados . Siga los pasos del asistente.
Desde la línea de mandatos	Emita la sentencia CREATE NICKNAME. Por ejemplo: <pre>CREATE NICKNAME apodo (column_name tipos_datos OPTIONS (opciones_columna_apodo), column_name tipos_datos OPTIONS (opciones_columna_apodo), column_name tipos_datos OPTIONS (opciones_columna_apodo)) FOR SERVER nombre_definición_servidor OPTIONS (opciones_apodo);</pre>

Repita este paso para cada hoja Excel para la que desee crear un apodo.

Sentencia CREATE NICKNAME - Ejemplos para el derivador de Excel:

Utilice la sentencia CREATE NICKNAME para registrar un apodo para una hoja Excel a la que desee acceder. Estos ejemplos muestran los parámetros obligatorios y las opciones de apodo opcionales.

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

Compounds

Apodo exclusivo utilizado para identificar la hoja Excel.

Importante: el apodo es un nombre de dos componentes que consta del esquema y del nombre del apodo. Si omite el esquema al registrar el apodo, se utiliza el ID de autorización del usuario que registra el apodo para el esquema del apodo.

Compound_ID INTEGER

Nombre y tipo de datos para una columna de la hoja que contiene los identificadores de los compuestos.

CompoundName VARCHAR(50)

Nombre y tipo de datos para una columna de la hoja que contiene los nombres de los compuestos.

MolWeight FLOAT

Nombre y tipo de datos para una columna de la hoja que contiene el peso molecular de los compuestos.

FOR SERVER biochem_lab

Nombre asignado a la definición del servidor Excel en la sentencia CREATE SERVER.

FILE_PATH 'C:\Mis Documentos\CompoundMaster.xls')

Especifica la vía de acceso completa al directorio y el nombre de archivo del libro de Excel que contiene los datos a los que desea acceder. Los datos deben encontrarse en la primera hoja del libro.

OPTIONS (RANGE 'B2:D25')

Especifica el rango de celdas a las que desea acceder en el libro que ha especificado en la opción de apodo FILE_PATH.

Cualquier error sintáctico o semántico en el valor de la opción de rango provocará el mensaje de error SQL1882E. Los errores pueden incluir:

- El rango no es válido. Por ejemplo, si la celda superior izquierda especificada en el rango está debajo o a la derecha de la celda inferior derecha.
- El número de columnas designado por el valor de rango no corresponde al número de columnas especificado en la sentencia CREATE NICKNAME.
- Se ha encontrado un carácter no válido u otro error de sintaxis.

Orígenes de datos Excel - consultas de ejemplo

Para acceder a datos Excel, se utiliza el apodo y las columnas de apodo definidas en las sentencias SQL del mismo modo que se utilizaría un nombre de tabla regular y columnas de tabla.

Estos ejemplos muestran cómo estructurar las consultas para acceder a datos Excel mediante el apodo compounds.

Selección de una columna de información específica

La consulta siguiente visualiza todos los elementos compound_ID cuyo peso molecular es superior a 2000:

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

Utilización de una condición OR en las sentencias SELECT

La consulta siguiente visualiza todos los registros en los que el nombre del compuesto o el peso molecular son nulos:

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

Utilización de condiciones LIKE y AND en las sentencias SELECT

La consulta siguiente visualiza todos los registros en los que el nombre del compuesto contiene la serie ase y el peso molecular es igual o superior a 300:

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

Origen de datos Excel - caso práctico de ejemplo

Este caso práctico muestra las sentencias SQL necesarias para registrar los objetos federados utilizados para acceder a una hoja Excel. En este caso práctico se incluyen varias consultas que puede ejecutar utilizando los apodos que cree.

Información de la hoja Excel

Este caso práctico se inicia con una hoja que contiene información relativa a diversos compuestos. El nombre del libro en el que se encuentra la hoja es *Compound_Master.xls* y el libro se ha creado en Excel. El nombre completo de la vía de acceso al libro es *C:\Data\Compound_Master.xls*.

La primera hoja del libro contiene cuatro columnas y nueve filas de datos. Las columnas listan los nombres de los compuestos, el peso de los compuestos, el recuento molecular del compuesto y si el compuesto se ha probado.

En la tabla siguiente se muestra el contenido de la hoja.

Tabla 19. Hoja de ejemplo *Compound_Master.xls*

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

Registro de los objetos federados

Para acceder a la hoja mediante el derivador de Excel, debe registrar los objetos en el servidor federado:

1. Registre el derivador de Excel.

Por ejemplo:

```
CREATE WRAPPER Excel LIBRARY 'db2lxs.xls.dll';
```

2. Registre la definición del servidor.

Por ejemplo:

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

3. Registre un apodo que haga referencia a la hoja Excel.

Por ejemplo:

```
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');
```

El proceso de registro ha finalizado. Ahora, la hoja Excel forma parte del sistema federado y puede utilizarse en consultas SQL.

Los ejemplos siguientes muestran las consultas SQL y los resultados devueltos desde el apodo *Compound_Master*.

Consulta que devuelve todos los datos coincidentes con una condición de cláusula WHERE específica

Para devolver todos los datos para los compuestos cuyo recuento molecular sea superior a 100, emita esta consulta:

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

Consulta que devuelve columnas específicas de la hoja

Para devolver los nombres y recuentos moleculares de todos los compuestos en los que el recuento molecular aún no se haya determinado, emita esta consulta:

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

Se devolverán todas las columnas de las filas 2, 3, 4, 6 y 8.

Se devolverán las columnas *compound_name* y *mol_count* de las filas 5, 7 y 10.

Consulta que cuenta el número de filas que coinciden con condiciones de cláusula WHERE específicas

Para devolver el número de compuestos cuyo peso es superior a 1 y que no se han probado, emita esta consulta:

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

Se devolverá el recuento de registros 1. El compuesto de la fila 7 coincide con los criterios de la consulta.

Consulta que devuelve columnas específicas de la hoja e incluye una sentencia SUBSELECT

Para devolver los nombres y recuentos moleculares de todos los compuestos en los que el recuento molecular se ha determinado y es inferior al promedio de recuentos moleculares, emita esta consulta:

```
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 subconsulta devolverá 368 para el promedio de recuentos moleculares. La consulta principal utiliza el promedio para devolver los resultados de consulta que se muestran en la tabla siguiente:

Tabla 20. Resultados de consulta

COMPOUND_NAME	MOL_COUNT
compound_A	367
compound_G	210
compound_F	174
compound_S	67

Modelo de control de acceso a archivos para el derivador de Excel

Para acceder a un archivo Excel, el derivador necesita una identidad de usuario a efectos de seguridad. El derivador de Excel utiliza la identidad de usuario asociada con el servicio de base de datos federada. El nombre del servicio de base de datos federada depende del nombre de la instancia de la base de datos. Por ejemplo, si el nombre de la instancia de la base de datos es DB2, el nombre del servicio será DB2 - DB2. Para determinar la identidad de usuario asociada con el servicio de base de datos federada, utilice el Panel de control de Windows para visualizar los servicios. Efectúe una doble pulsación sobre el nombre del servicio y visualice la página de propiedades de inicio de sesión.

Configuración del acceso a orígenes de datos Informix

Para configurar un servidor federado para acceder a orígenes de datos Informix, debe proporcionar al servidor federado información acerca de los orígenes de datos y objetos a los que desea acceder.

Antes de empezar

- El software del SDK del cliente Informix debe estar instalado y configurado en el servidor que actuará como servidor federado.
- Compruebe la configuración del servidor federado.
- Compruebe el parámetro FEDERATED para asegurarse de que Federation está habilitado.
- En servidores federados AIX, debe estar instalada la biblioteca AIX Base Application Development Math Library. Puede determinar si la biblioteca está instalada emitiendo el mandato AIX `lspp -l bos.adt.libm`.

Puede configurar un servidor federado para acceder a datos almacenados en orígenes de datos Informix utilizando el Centro de control de DB2 o emitiendo sentencias SQL en la línea de mandatos de DB2. El Centro de control de DB2 incluye un asistente que le guiará en los pasos necesarios para configurar los objetos federados necesarios.

Procedimiento

Para añadir orígenes de datos Informix a un servidor federado:

1. Configure y pruebe el archivo de configuración del cliente Informix.
2. Establezca las variables de entorno de Informix.
3. Registre el derivador.
4. Registre la definición del servidor.
5. Cree las correlaciones de usuario.
6. Pruebe la conexión con el servidor.
7. Registre los apodos para las vistas, tablas y sinónimos de Informix.

Configuración y prueba del archivo de configuración del cliente Informix

El archivo de configuración del cliente Informix se utiliza para conectarse a bases de datos Informix mediante las bibliotecas de cliente instaladas en el servidor federado.

Antes de empezar

El software del SDK del cliente Informix debe estar instalado en el servidor federado.

El archivo de configuración del cliente especifica la ubicación de cada servidor de bases de datos Informix y el tipo de conexión (protocolo) para el servidor de bases de datos.

La ubicación predeterminada del archivo de configuración del cliente depende del sistema operativo utilizado por el servidor federado.

- En servidores federados que ejecutan UNIX, la ubicación predeterminada y el nombre del archivo son `$INFORMIXDIR/etc/sqlhosts`. El archivo `sqlhosts` se instala con el SDK del cliente Informix.
- En servidores federados que ejecutan Windows, la ubicación predeterminada del registro `sqlhosts` es el sistema local.

El formato de `sqlhosts` se describe en la *Guía del administrador para Informix Dynamic Server*.

Procedimiento

Para configurar y probar el archivo de configuración del cliente Informix:

1. Configure el SDK del cliente Informix.
 - En servidores federados que ejecutan UNIX, puede configurar el SDK del cliente Informix editando el archivo `sqlhosts`. También puede copiar el archivo `sqlhosts` de otro sistema que tenga instalado Informix Connect o el SDK del cliente Informix.
 - En servidores federados que ejecutan Windows, puede configurar el SDK del cliente Informix con el programa de utilidad `Setnet32` de Informix. El programa de utilidad `Setnet32` configura el registro `sqlhosts`.
2. Compruebe la ubicación del archivo o registro `sqlhosts`.
 - En servidores federados que ejecutan UNIX, el archivo `sqlhosts` se encuentra en el directorio `$INFORMIXDIR/etc/`.
 - En servidores federados que ejecutan Windows, la información de `sqlhosts` se conserva en la clave siguiente del registro de Windows:
`HKEY_LOCAL_MACHINE\SOFTWARE\INFORMIX\SQLHOSTS`
3. Si desea colocar el archivo o registro `sqlhosts` en una vía de acceso que no sea la vía de acceso de búsqueda predeterminada, establezca la variable de entorno `INFORMIXSQLHOSTS` para especificar la ubicación del archivo. Utilice una de las opciones siguientes para establecer la variable de entorno `INFORMIXSQLHOSTS`.
 - En servidores federados que ejecutan UNIX, establezca la variable de entorno `INFORMIXSQLHOSTS` en el nombre completo del archivo `sqlhosts`.
 - En servidores federados que ejecutan Windows, utilice el programa de utilidad `Setnet32` para establecer la variable de entorno `INFORMIXSQLHOSTS` en el nombre del sistema Windows que almacena el registro.
4. Pruebe la conexión para asegurarse de que el software del cliente puede conectarse al servidor Informix. Si el programa de utilidad `dbaccess` de Informix se encuentra en el servidor federado, utilice esta herramienta para probar la conexión. De lo contrario, ejecute el programa de demostración de Informix para probar la configuración del cliente.

Establecimiento de variables de entorno de Informix

Las variables de entorno de Informix deben establecerse en el archivo `db2dj.ini` del servidor federado.

Existen variables de entorno obligatorias y opcionales para los orígenes de datos Informix. Si ha instalado el software del cliente Informix antes de instalar el derivador de Informix, las variables de entorno obligatorias de Informix se establecen en el archivo `db2dj.ini`.

Debe establecer las variables de entorno siguiendo los pasos de esta tarea si no ha instalado el software del cliente Informix antes de instalar el derivador de Informix o si desea establecer alguna de las variables de entorno opcionales.

Para establecer las variables de entorno de Informix:

1. Utilice uno de los métodos siguientes para establecer las variables de entorno de Informix que desee utilizar:

Método	Paso
Establecer las variables de entorno mediante el Centro de control de DB2	Utilice el Asistente para objetos federados del Centro de control de DB2. Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados
Establecer las variables de entorno automáticamente	Ejecute de nuevo el programa de instalación de DB2 y especifique la opción de instalación Personalizada . Siga las instrucciones del asistente. Importante: Al ejecutar de nuevo el programa de instalación, sólo se establecerán las variables de entorno obligatorias. Las variables de entorno opcionales deben establecerse manualmente.
Establecer las variables de entorno manualmente	Edite el archivo <code>db2dj.ini</code> . <ul style="list-style-type: none">• En servidores federados que ejecutan UNIX, el archivo <code>db2dj.ini</code> se encuentra en el directorio <code>sql1lib/cfg</code>.• En servidores federados que ejecutan Windows, el archivo <code>db2dj.ini</code> se encuentra en el directorio <code>%DB2PATH%\cfg</code>.

El archivo `db2dj.ini` contiene información de configuración relativa al software del cliente Informix que está instalado en el servidor federado. Si el archivo no existe, puede crear un archivo con el nombre `db2dj.ini` mediante cualquier editor de texto. En el archivo `db2dj.ini`, debe especificar la vía de acceso completa de las variables de entorno; de lo contrario, se producirán errores. Las variables de entorno siguientes muestran cuál puede ser la apariencia de una entrada del archivo `db2dj.ini` en UNIX:

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

2. Establezca las variables de entorno de conversión de páginas de códigos Informix (según sea necesario).
3. Para asegurarse de que las variables de entorno están establecidas en el servidor federado, reinicie la instancia de la base de datos federada.
Emita los mandatos siguientes para reiniciar la instancia de la base de datos federada:

```
db2stop
db2start
```

Variables de entorno de Informix

Existen variables de entorno necesarias y variables de entorno opcionales para los orígenes de datos Informix. Estas variables se definen en el archivo `db2dj.ini`.

Las variables de entorno válidas para Informix son:

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

Las variables de entorno `CLIENT_LOCALE`, `DB_LOCALE` y `DBNLS` son variables de entorno de página de códigos.

Descripción de las variables

INFORMIXDIR

Especifica la vía de acceso de directorio en la que se instala el software del SDK de Informix Client.

Por ejemplo:

- En servidores federados que ejecutan UNIX, establezca esta vía de acceso:

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

- En servidores federados que ejecutan Windows, establezca esta vía de acceso:

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

INFORMIXSERVER

Identifica el nombre del servidor Informix predeterminado. Este valor debe ser una entrada válida en el archivo `sqlhosts` (UNIX) o la clave de registro `SQLHOSTS` (Windows). Para obtener un valor para `INFORMIXSERVER`, consulte el archivo `sqlhosts`. Seleccione uno de los valores de `dbservername`. El valor `dbservername` es el primer valor de cada entrada del archivo `sqlhosts`.

Por ejemplo:

```
INFORMIXSERVER=inf10
```

Requisito: Si bien el derivador Informix no utiliza el valor de esta variable de entorno, el cliente Informix necesita que esté definida. El derivador utiliza el valor de la opción de servidor `NODE`, que especifica el servidor de bases de datos Informix al que se desea acceder.

INFORMIXSQLHOSTS

Si utiliza la vía de acceso predeterminada para el archivo `sqlhosts` de Informix, no es necesario que defina esta variable de entorno. No obstante, sí debe definir esta variable de entorno si utiliza una vía de acceso distinta para el archivo `sqlhosts` de Informix. Establezca la variable de entorno `INFORMIXSQLHOSTS` en el nombre de la vía de acceso completa en la que reside el archivo `sqlhosts` de Informix.

- En servidores federados que ejecutan UNIX, la vía de acceso predeterminada es \$INFORMIXDIR/etc/sqlhosts.
- En servidores federados que ejecutan Windows, si la clave de registro SQLHOSTS no reside en el sistema local, el valor para la variable de entorno INFORMIXSQLHOSTS es el nombre del sistema Windows que almacena el registro.

Un ejemplo UNIX para definir esta variable de entorno en otra vía de acceso sería:

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

Conversión de la página de códigos de Informix:

Cada vez que el derivador Informix se conecta con un origen de datos Informix, el derivador determina qué valor de página de códigos debe utilizar para dicha conexión. Puede hacer que el derivador Informix establezca el valor de la página de códigos o definir una página de códigos mediante la variable de entorno CLIENT_LOCALE.

Las variables de entorno que especifican la conversión de página de códigos de Informix se establecen en el archivo db2dj.ini del servidor federado.

Para la conversión de la página de códigos Informix, puede establecer las siguientes variables de entorno opcionales:

- CLIENT_LOCALE
- DB_LOCALE
- DBNLS

Las variables de entorno de la página de códigos de Informix son:

CLIENT_LOCALE

Especifica el entorno local de Informix que se desea utilizar. Utilice esta variable si no desea que el derivador Informix determine automáticamente el valor de la variable.

Por ejemplo:

```
CLIENT_LOCALE=valor_entorno_local_cliente_Informix
```

- Si la variable CLIENT_LOCALE se establece en el archivo db2dj.ini del servidor federado, el derivador utiliza el valor de la página de códigos del archivo db2dj.ini.
- Si la variable CLIENT_LOCALE no está establecida en el servidor federado, el derivador determina el territorio y la página de códigos de la base de datos federada. El derivador establece la variable CLIENT_LOCALE en el entorno local de Informix más cercano. Si no existe ningún entorno local de Informix coincidente, el derivador establece la variable CLIENT_LOCALE en el entorno local en_us.8859-1 para sistemas UNIX y en el entorno local en_us.CP1252 para sistemas Windows.

Para ver la lista de entornos locales de Informix, emita el mandato glfiles en el servidor Informix.

Consulte la publicación *Informix Guide to GLS Functionality* para obtener más información sobre las conversiones de páginas de códigos.

DB_LOCALE

Especifica que la base de datos Informix utiliza una página de códigos

distinta de la del entorno del cliente. Utilice esta variable si desea que Informix realice conversiones entre las dos páginas de códigos. Establezca la variable de entorno DB_LOCALE en el nombre del entorno de la base de datos Informix.

Por ejemplo:

```
DB_LOCALE=valor_entorno_local_bd_Informix
```

DBNLS

Especifica que Informix verifica si el valor de DB_LOCALE coincide con el entorno local actual de la base de datos Informix. Establezca esta variable de entorno en 1.

Por ejemplo:

```
DBNLS=1
```

Forzar que Informix realice la conversión de la página de códigos

La base de datos Informix utiliza una página de códigos distinta de la del entorno local del cliente y desea que Informix realice conversiones entre las dos páginas de códigos. Debe hacer lo siguiente:

1. Establezca la variable de entorno DB_LOCALE de Informix en el nombre del entorno local de la base de datos Informix. Esta variable se define en el archivo `db2dj.ini` del servidor federado.
2. Para verificar que el valor de DB_LOCALE coincide con el entorno local real de la base de datos Informix, establezca la variable de entorno DBNLS de Informix en 1. Esta variable se establece en el archivo `db2dj.ini` del servidor federado.

Datos de Informix que utilizan la página de códigos para chino GB 18030

Para acceder a datos que utilizan la página de códigos para chino GB 18030, utilice la página de códigos UTF-8 de la base de datos federada y añada el siguiente valor al archivo `db2dj.ini`, de modo que Informix convierta correctamente los datos GB 18030 a unicode.

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

Registro del derivador de Informix

Debe registrar un derivador para acceder a los orígenes de datos de Informix. Los servidores federados utilizan los derivadores para comunicarse con los orígenes de datos y para recuperar datos de éstos. Los derivadores se implementan como un conjunto de archivos de biblioteca.

Procedimiento

Para registrar el derivador de Informix:

Emita la sentencia `CREATE WRAPPER` y especifique el nombre predeterminado para el derivador de Informix.

Por ejemplo:

```
CREATE WRAPPER INFORMIX;
```

Recomendación: Utilice el nombre del derivador predeterminado. El nombre del derivador predeterminado para Informix es `INFORMIX`. Al registrar el derivador mediante el nombre predeterminado, el servidor federado utiliza automáticamente la biblioteca de derivador de Informix adecuada al sistema operativo en el que se ejecuta el servidor federado.

Si el nombre del derivador predeterminado entra en conflicto con un nombre de derivador existente en la base de datos federada, puede sustituir el nombre del derivador predeterminado por otro de su elección. Si utiliza un nombre que no es el predeterminado, deberá incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER.

Por ejemplo, para registrar un derivador con el nombre derivador_informix en un servidor federado que utiliza el sistema operativo AIX, emita esta sentencia:

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

El nombre del archivo de biblioteca del derivador especificado dependerá del sistema operativo del servidor federado. Consulte la lista de archivos de biblioteca de derivador de Informix para conocer el nombre de biblioteca correcto que debe especificar en la sentencia CREATE WRAPPER.

Archivos de biblioteca de derivador de Informix

Los archivos de biblioteca de derivador Informix se añaden al servidor federado cuando se instala Federation.

Cuando instala Federation, se añaden tres archivos de biblioteca de derivador a la vía de acceso del directorio predeterminado. Por ejemplo, si el servidor federado se ejecuta en AIX, los archivos de biblioteca del derivador que se añaden a la vía de acceso del directorio son libdb2informix.a, libdb2informixF.a y libdb2informixU.a. El archivo de biblioteca de derivador predeterminado es libdb2informix.a. Los demás archivos de biblioteca de derivador los utiliza internamente la biblioteca de derivador predeterminada.

Si decide no utilizar el nombre del derivador predeterminado cuando registra el derivador, debe incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER y especificar el nombre de archivo de biblioteca del derivador predeterminado.

En la tabla siguiente se indican las vías de acceso al directorio predeterminado y los nombres de archivo de biblioteca del derivador predeterminado.

Tabla 21. Ubicaciones de biblioteca y nombres de archivo del derivador Informix

Sistema operativo	Vía de acceso al directorio	Nombre de archivo de biblioteca
AIX	/usr/opt/dir_instalación/lib32/ /usr/opt/dir_instalación/lib64/	libdb2informix.a
HP-UX	/opt/IBM/db2/dir_instalación/lib32 /opt/IBM/db2/dir_instalación/lib64	libdb2informix.so
Linux	/opt/IBM/db2/dir_instalación/lib32 /opt/IBM/db2/dir_instalación/lib64	libdb2informix.so
Solaris	/opt/IBM/db2/dir_instalación/lib32 /opt/IBM/db2/dir_instalación/lib64	libdb2informix.so
Windows	%DB2PATH%\bin	db2informix.dll

- *vía_acceso_instalación* es la vía de acceso al directorio en el que está instalado Federation en UNIX o Linux.
- %DB2PATH% es la variable de entorno que especifica la vía de acceso al directorio donde se ha instalado Federation en Windows. La vía de acceso al directorio de Windows predeterminado es C:\Program Files\IBM\SQLLIB.

Registro de las definiciones de servidor para un origen de datos Informix

Debe registrar cada servidor Informix al que desee acceder en la base de datos federada.

Procedimiento

Para registrar una definición de servidor para un origen de datos Informix:

1. Localice el nombre de nodo en el archivo o registro `sqlhosts` de Informix.

Archivo `sqlhosts` de ejemplo:

```
inf10an onsoctcp anaconda inmx10
inf10bo onsoctcp boa ifmx10
inf10py onsoctcp python ifmx10
```

- El primer valor de cada línea es el *nombre_nodo*, como por ejemplo `inf10an`.
- El segundo valor de cada línea es el *nettype*, o tipo de conexión. En este ejemplo, `onsoctcp` indica que se trata de una conexión TCP/IP.
- El tercer valor de cada línea es el nombre de host, como por ejemplo `anaconda`, `boa` y `python`.
- El cuarto valor de cada línea es el nombre del servicio, como por ejemplo `inmx10`. El campo de nombre de servicio depende del *nettype* indicado en el segundo valor.

Para obtener más información acerca del formato del archivo `sqlhosts` y el significado de estos campos, consulte el manual de Informix *Administrators Guide for Informix Dynamic Server*.

2. Utilice uno de los métodos siguientes para crear la definición del servidor:

- Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database. Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados.
- Emita la sentencia `CREATE SERVER`.

Por ejemplo:

```
CREATE SERVER nombre_definición_servidor TYPE informix
  VERSION número_versión WRAPPER INFORMIX
  OPTIONS (NODE 'nombre_nodo', DBNAME 'nombre_base_datos');
```

Aunque las variables *'nombre_nodo'* y *'nombre_base_datos'* se especifican como opciones en la sentencia `CREATE SERVER`, estas opciones son necesarias para los orígenes de datos Informix.

Una vez registrada la definición del servidor, utilice la sentencia `ALTER SERVER` para añadir o eliminar opciones de servidor.

Sentencia `CREATE SERVER` - Ejemplos para el derivador de Informix

Utilice la sentencia `CREATE SERVER` para registrar definiciones del servidor para el derivador Informix. Este tema incluye un ejemplo completo con los parámetros necesarios y ejemplos que muestran el uso de opciones adicionales del servidor.

Ejemplo completo

El ejemplo siguiente muestra cómo registrar una definición de servidor para un derivador Informix emitiendo la sentencia `CREATE SERVER`:

```
CREATE SERVER asia TYPE informix VERSION 10 WRAPPER INFORMIX
  OPTIONS (NODE 'abc', DBNAME 'ventas');
```

asia Nombre que asigna al servidor de bases de datos Informix. No se permiten los nombres de definición de servidor duplicados.

TYPE *informix*

Especifica el tipo de servidor de orígenes de datos para el que está configurando el acceso. Para el derivador Informix, el tipo de servidor debe ser *informix*.

VERSION *10*

La versión del servidor de bases de datos Informix al que desea acceder.

WRAPPER *INFORMIX*

El nombre del derivador que ha especificado en la sentencia CREATE WRAPPER.

NODE *'abc'*

Nombre del nodo en el que reside el servidor de bases de datos Informix. Obtenga el nombre de nodo del archivo `sqlhosts`. Este valor es sensible a las mayúsculas y minúsculas.

Aunque el nombre del nodo se especifica como opción en la sentencia CREATE SERVER, es necesario para los orígenes de datos de Informix.

DBNAME *'ventas'*

Nombre de la base de datos Informix a la que desea acceder. Este valor es sensible a las mayúsculas y minúsculas.

Aunque el nombre de la base de datos se especifica como opción en la sentencia CREATE SERVER, es necesario para los orígenes de datos de Informix.

Opciones de servidor adicionales

Al crear una definición de servidor, puede especificar opciones de servidor adicionales en la sentencia CREATE SERVER. Las opciones de servidor pueden ser genéricas y específicas de Informix.

Opciones de servidor FOLD_ID y FOLD_PW

Cuando el servidor federado se conecta a un origen de datos, intenta hacerlo utilizando todas las combinaciones posibles de mayúsculas y minúsculas para el ID de usuario y la contraseña, así como en las mayúsculas y minúsculas actuales. El servidor federado puede llegar hasta nueve intentos antes de conectarse satisfactoriamente al servidor de orígenes de datos. Estos intentos pueden ralentizar los tiempos de conexión y provocar el bloqueo del ID de usuario. Puede evitar lo bloqueos especificando valores para las opciones de servidor FOLD_ID y FOLD_PW. Puede establecer las opciones de servidor FOLD_ID y FOLD_PW en 'N' (no convertir el ID de usuario o la contraseña).

Si establece las opciones de servidor FOLD_ID y FOLD_PW en 'N', debe especificar el ID de usuario y la contraseña con las mayúsculas y minúsculas correctas. La ventaja de establecer estas opciones de servidor en 'N' es que, si se especifica un ID de usuario o una contraseña incorrectos, el derivador no seguirá intentando las diversas combinaciones de mayúsculas y minúsculas. Estas dos opciones de servidor pueden reducir la posibilidad de sobrepasar el número máximo de intentos de inicio de sesión fallidos y el bloqueo del ID.

El ejemplo siguiente muestra una definición de servidor Informix con estas opciones de servidor:

```
CREATE SERVER asia TYPE informix VERSION 10 WRAPPER INFORMIX
  OPTIONS (NODE 'abc', DBNAME 'ventas', FOLD_ID 'N', FOLD_PW 'N');
```

Ejemplo de opción de servidor IUD_APP_SVPT_ENFORCE

La opción IUD_APP_SVPT_ENFORCE especifica si el servidor federado debe aplicar la detección o creación de sentencias de punto de recuperación de aplicación. Informix no da soporte a las sentencias de punto de recuperación de aplicación. Si se establece en 'N', el servidor federado no retrotraerá las transacciones cuando se encuentre un error. La aplicación deberá manejar la recuperación de errores.

La opción de servidor IUD_APP_SVPT_ENFORCE debe establecerse en 'N' para permitir la réplica a orígenes de datos Informix o desde ellos. El ejemplo siguiente muestra una definición de servidor Informix con la opción de servidor IUD_APP_SVPT_ENFORCE.

```
CREATE SERVER asia TYPE informix VERSION 10 WRAPPER INFORMIX
  OPTIONS (NODE 'abc', DBNAME 'ventas', IUD_APP_SVPT_ENFORCE 'N');
```

Opciones INFORMIX_DB_LOCALE e INFORMIX_CLIENT_LOCALE

La opción INFORMIX_DB_LOCALE establece la variable de entorno local de la base de datos (DB_LOCALE) utilizada para la conexión entre el servidor federado y el servidor de orígenes de datos. Si no se especifica la opción INFORMIX_DB_LOCALE, la variable de entorno local DB_LOCALE de Informix se establece en el valor especificado en el archivo db2dj.ini. Si el archivo db2dj.ini no especifica la variable de entorno DB_LOCALE, la variable de entorno DB_LOCALE de Informix no se establece. El valor válido es cualquier entorno local de Informix. Esta opción es opcional. El valor predeterminado es None (ninguno).

La opción INFORMIX_CLIENT_LOCALE establece la variable de entorno local del cliente (CLIENT_LOCALE) que debe utilizarse para la conexión entre el servidor federado y el servidor de orígenes de datos. Si no se especifica la opción INFORMIX_CLIENT_LOCALE, la variable de entorno local CLIENT_LOCALE de Informix se establece en el valor especificado en el archivo db2dj.ini. Si db2dj.ini no especifica CLIENT_LOCALE, la variable de entorno CLIENT_LOCALE de Informix se establece en el entorno local de Informix más parecido a la página de códigos y al territorio de la base de datos federada. El valor válido es cualquier entorno local Informix válido. Esta opción es opcional. El valor predeterminado es None (ninguno).

El ejemplo siguiente muestra una definición de servidor Informix con las opciones de servidor INFORMIX_DB_LOCALE_CLIENT y INFORMIX_CLIENT_LOCALE.

```
CREATE SERVER asia TYPE informix VERSION 10 WRAPPER INFORMIX
  OPTIONS (NODE 'abc', DBNAME 'ventas', INFORMIX_DB_LOCALE 'en_us.8859-1',
  INFORMIX_CLIENT_LOCALE 'en_us.CP1252');
```

Creación de las correlaciones de usuario para un origen de datos Informix

Cuando se intenta acceder a un servidor Informix, el servidor federado establece una conexión con el servidor Informix utilizando un ID de usuario y una contraseña válidos para ese origen de datos. Debe definir una asociación (una correlación de usuario) entre cada ID de usuario y contraseña del servidor federado y un ID de usuario y contraseña correspondientes del origen de datos.

Cree una correlación de usuario para cada ID de usuario que vaya a acceder al sistema federado para enviar solicitudes distribuidas al origen de datos Informix.

Procedimiento

Para correlacionar un ID de usuario local con el ID de usuario y la contraseña del servidor Informix:

Emita una sentencia CREATE USER MAPPING.

Por ejemplo:

```
CREATE USER MAPPING FOR IDusuario_local SERVER nombre_definición_servidor
    OPTIONS (REMOTE_AUTHID 'IDusuario_remoto', REMOTE_PASSWORD 'contraseña_remota');
```

Aunque las variables REMOTE_AUTHID y REMOTE_PASSWORD se especifican como opciones en la sentencia CREATE USER MAPPING, estas opciones son necesarias para acceder a orígenes de datos Informix.

Sentencia CREATE USER MAPPING - Ejemplos para el derivador de Informix

Utilice la sentencia CREATE USER MAPPING para correlacionar un ID de usuario de servidor federado con un ID de usuario y una contraseña de servidor Informix. Este tema incluye un ejemplo completo con los parámetros necesarios y un ejemplo que muestra cómo utilizar el registro especial de DB2 USER con la sentencia CREATE USER MAPPING.

Ejemplo completo

El ejemplo siguiente muestra cómo correlacionar un ID de usuario de servidor federado con un ID de usuario y una contraseña de servidor Informix:

```
CREATE USER MAPPING FOR VINCENT SERVER asia
    OPTIONS (REMOTE_AUTHID 'vinnie', REMOTE_PASSWORD 'close2call');
```

VINCENT

Especifica el ID de usuario local que se correlaciona con un ID de usuario definido en el servidor Informix.

SERVER asia

Especifica el nombre de la definición del servidor que ha registrado en la sentencia CREATE SERVER para el servidor Informix.

REMOTE_AUTHID 'vinnie'

Especifica el ID de usuario del servidor de bases de datos Informix con el que está correlacionando *VINCENT*. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_ID en 'U' o 'L' en la sentencia CREATE SERVER.

Aunque el ID de usuario remoto se especifica como opción en la sentencia CREATE USER MAPPING, es necesario para los orígenes de datos de Informix.

REMOTE_PASSWORD 'close2call'

Especifica la contraseña asociada a '*vinnie*'. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_PW en 'U' o 'L' en la sentencia CREATE SERVER.

Aunque la contraseña remota se especifica como opción en la sentencia CREATE USER MAPPING, es necesaria para los orígenes de datos de Informix.

Ejemplo de registro especial

Puede utilizar el registro especial USER de DB2 para correlacionar el ID de autorización de la persona que emite la sentencia CREATE USER MAPPING con el ID de autorización del origen de datos especificado en la opción de usuario REMOTE_AUTHID.

A continuación se ofrece un ejemplo de la sentencia CREATE USER MAPPING, que incluye el registro especial USER:

```
CREATE USER MAPPING FOR USER SERVER asia
      OPTIONS (REMOTE_AUTHID 'vinnie', REMOTE_PASSWORD 'close2call');
```

Prueba de la conexión al servidor Informix

Pruebe la conexión con el servidor de orígenes de datos Informix para determinar si el servidor federado se ha configurado correctamente para acceder a orígenes de datos Informix.

Puede probar la conexión con el servidor Informix mediante la definición de servidor y las correlaciones de usuario que ha definido.

Procedimiento

Para probar la conexión al servidor Informix:

Abra una sesión de paso a través y emita una sentencia SELECT en las tablas del sistema Informix. Si la sentencia SELECT devuelve una cuenta, significa que la definición del servidor y las correlaciones de usuario están correctamente configuradas.

Por ejemplo:

```
SET PASSTHRU nombre_definición_servidor
SELECT count(*) FROM informix.systables
SET PASSTHRU RESET
```

Si la sentencia SELECT devuelve un error, debe resolver los errores de conexión.

Resolución de errores de conexión de origen de datos

Una conexión de prueba con el servidor de origen de datos puede devolver un error por varios motivos. Existen acciones que permiten determinar el motivo del error.

Síntoma

Se devuelve un error al intentar establecer conexión con el origen de datos.

Causa

Hay varias causas posibles que explican un problema de conexión.

Resolución del problema

Para solucionar los problemas de errores de conexión con el origen de datos, consulte los siguientes elementos:

- Verifique que el origen de datos está disponible.
- Si procede, asegúrese de que el servidor de origen de datos esté configurado para conexiones entrantes.

- Asegúrese de que los valores de correlación de usuarios para las opciones REMOTE_AUTHID y REMOTE_PASSWORD son válidos para las conexiones con el origen de datos. Modifique la correlación de usuarios o cree otra correlación de usuarios según sea necesario.
- Si procede, asegúrese de que el software de cliente del servidor federado esté instalado y configurado correctamente para conectarse al origen de datos.
- Para orígenes de datos ODBC, asegúrese de que el controlador ODBC del servidor federado esté instalado y configurado correctamente para establecer conexión con el servidor de origen de datos ODBC. En servidores federados que ejecutan Windows, utilice la herramienta ODBC Data Source Administrator para verificar el controlador. En servidores federados que ejecutan UNIX, consulte la documentación del proveedor del cliente ODBC.
- Verifique que los valores para las variables establecidas en el servidor federado son correctas para el origen de datos. Estas variables incluyen las variables de entorno del sistema, las variables de la archivo db2dj.ini y las variables del registro de perfiles de DB2 (db2set).
- Compruebe la definición del servidor. Si es necesario, elimine la definición del servidor y créela de nuevo.

Ajuste del rendimiento para el derivador Informix

Puede utilizar las opciones de servidor FOLD_ID y FOLD_PW para mejorar la conectividad entre el servidor federado y los orígenes de datos Informix.

Cuando el servidor federado se conecta a un origen de datos, el servidor intenta conectarse utilizando todas las combinaciones posibles de mayúsculas y minúsculas para el ID de usuario y la contraseña. Es posible que el servidor realice hasta nueve intentos antes de conectarse correctamente al servidor de origen de datos. Estos intentos pueden ralentizar los tiempos de conexión y es posible que ello dé lugar al bloqueo del ID de usuario.

El rendimiento puede mejorarse especificando los valores para las opciones de servidor FOLD_ID y FOLD_PW.

- Si todos los ID de usuario y las contraseñas de Informix son en minúsculas, establecer las opciones de servidor FOLD_ID y FOLD_PW con el valor 'L' puede mejorar el tiempo de conexión.

Por ejemplo:

```
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_ID 'L');
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_PW 'L');
```

- El servidor federado intenta todas las combinaciones de valores en mayúsculas y minúsculas para el ID de usuario y la contraseña. Puede reducir la probabilidad de que se supere el número máximo de intentos de inicio de sesión anómalos estableciendo estas opciones en 'N' (no convertir el ID de usuario y la contraseña). Si establece estos valores, siempre deberá especificar el ID de usuario y la contraseña con el uso correcto de mayúsculas y minúsculas. Si se especifica un ID de usuario y una contraseña que no son válidos, el derivador no seguirá intentando las distintas conexiones.

Por ejemplo:

```
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_ID 'N');
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_PW 'N');
```


Registro de apodos para tablas, vistas y sinónimos de Informix

Para cada definición de servidor Informix que registre, debe registrar un apodo para cada tabla, vista o sinónimo a los que desee acceder. Utilice estos apodos en lugar de los nombres de los objetos de origen de datos cuando consulte los servidores Informix.

Antes de empezar

Actualice las estadísticas del origen de datos Informix antes de registrar un apodo. La base de datos federada se basa en las estadísticas del catálogo de orígenes de datos para optimizar el proceso de las consultas. Puede utilizar el mandato de Informix UPDATE STATISTICS, equivalente al mandato de DB2 RUNSTATS, para actualizar las estadísticas del origen de datos.

Procedimiento

Para registrar un apodo para una tabla, vista o sinónimo de Informix:

Emita la sentencia CREATE NICKNAME. Los apodos pueden tener una longitud de hasta 128 bytes.

Por ejemplo:

```
CREATE NICKNAME apodo FOR nombre_definición_servidor."esquema_remoto".  
"tabla.remota"  
;
```

Cuando se crea el apodo, el servidor federado consulta el catálogo de orígenes de datos utilizando el apodo. Esta consulta prueba la conexión a la tabla, vista o sinónimo del origen de datos. Si la conexión no funciona, recibirá un mensaje de error.

Repita este paso para cada tabla, vista o sinónimo de Informix para los que desee crear un apodo.

Sentencia CREATE NICKNAME - Ejemplos para el derivador de Informix

Utilice la sentencia CREATE NICKNAME para registrar un apodo para una tabla, vista o sinónimo de Informix a los que desea acceder. Este tema incluye un ejemplo completo con los parámetros necesarios.

Ejemplo completo

```
CREATE NICKNAME JPSALES FOR asia."vinnie"."japan" ;
```

JPSALES

Apodo exclusivo utilizado para identificar la tabla, vista o sinónimo Informix.

Importante: El nombre de apodo consta de dos partes: el esquema y el apodo. Si omite el esquema al registrar el apodo, el esquema del apodo será el ID de autorización del usuario que registra el apodo.

asia."vinnie"."japan"

Es un identificador de tres partes para el objeto remoto:

- *asia* es el nombre de definición de servidor que ha asignado al servidor de bases de datos Informix en la sentencia CREATE SERVER.

- *vinnie* es el nombre del propietario al que pertenece la tabla, vista o sinónimo, a menos que la base de datos sea compatible con ANSI. En una base de datos compatible con ANSI, es el nombre del esquema.
- *japan* es el nombre de la tabla, vista o sinónimo remoto al que desea acceder.

El servidor federado convierte a mayúsculas los nombres de los esquemas y tablas Informix, a menos que especifique los nombres entre comillas.

Configuración del acceso a orígenes de datos JDBC

Para configurar el servidor federado para acceder a orígenes de datos JDBC, debe proporcionar al servidor federado información acerca de los orígenes de datos y objetos a los que desea acceder.

En este documento, los orígenes de datos a los que se accede mediante la API de JDBC se denominan orígenes de datos JDBC.

Antes de empezar

- El controlador JDBC debe estar instalado y configurado en el sistema que actúa como servidor federado.
- Compruebe la configuración del servidor federado.
- Compruebe el parámetro FEDERATED para asegurarse de que Federation está habilitado.
- Establezca las variables de entorno del sistema, las variables del archivo `db2dj.ini` y las variables del registro de perfiles de DB2 (`db2set`). Consulte la documentación suministrada por el origen de datos JDBC para conocer las variables del cliente JDBC. Es posible que sea necesaria la variable de entorno `LIBPATH`.

Restricciones

- El derivador de JDBC sólo está soportado en modalidad limitada.
- El derivador de JDBC no da soporte a las funciones y sentencias siguientes:
 - Sentencias `LOCK TABLE` en apodos
 - Aislamiento a nivel de sentencia
- Los orígenes de datos JDBC no dan soporte a operaciones de actualización y supresión posicionadas.
- El derivador de JDBC no da soporte a sentencias `INSERT`, `UPDATE` o `DELETE` en orígenes de datos que restringen el número de sentencias activas para cada conexión. Consulte la documentación de su origen de datos para determinar si éste restringe el número de sentencias activas para cada conexión.
- El derivador de JDBC no da soporte a operaciones en tablas que contienen columnas con tipos de datos que utilizan indicadores de tipo de datos SQL específicos del controlador. Los tipos de operaciones no soportadas incluyen las sentencias `CREATE NICKNAME` y `SELECT` en sesiones de paso a través. El derivador de JDBC sólo da soporte a los indicadores de tipos de datos SQL definidos por la especificación JDBC 3.0 y versiones posteriores. Consulte la documentación del controlador JDBC para conocer las especificaciones de JDBC.
- El derivador de JDBC no da soporte a LOB en sesiones de paso a través.
- Restricciones en correlación de tipos y conversión de tipos de datos:
 - Tipos de datos no soportados: `ARRAY`, `DATALINK`, `DISTINCT`, `JAVA_OBJECT`, `REF`, `STRUCT` y `OTHER`

- Tipos de datos con soporte limitado:
 - El soporte para el tipo de datos XML es limitado. El servidor federado procesa el tipo de datos CLOB sólo si el tipo de datos JDBC relacionado es CLOB o SQLXML (JDBC 4.0). De lo contrario, no existe soporte para el tipo de datos XML.
 - El derivador de JDBC almacena los tipos de datos DBCS y UNICODE como UCS-2.
 - El tipo de datos DECFLOAT en bases de datos DB2 y el tipo de datos NUMBER en bases de datos Oracle tienen un ámbito más amplio, y sus formatos pueden ser diferentes de los tipos de datos correspondientes del derivador de JDBC. La correlación con un tipo de datos DECFLOAT o NUMBER puede generar resultados inexactos.

Acerca de esta tarea

Puede configurar el servidor federado para acceder a orígenes de datos JDBC utilizando el Centro de control de DB2 o emitiendo sentencias SQL en la línea de mandatos de DB2. El Centro de control de DB2 incluye un asistente que le guiará en los pasos necesarios para configurar los objetos federados necesarios.

Procedimiento

Para añadir orígenes de datos JDBC a un servidor federado:

1. Prepare el servidor federado para acceder a orígenes de datos a través de JDBC.
2. Registre el derivador de JDBC.
3. Registre las definiciones de servidor para un origen de datos JDBC.
4. Cree una correlación de usuario para un origen de datos JDBC.
5. Pruebe la conexión con el servidor de orígenes de datos JDBC.
6. Registre apodos para las tablas y vistas del origen de datos JDBC.

Preparación del servidor federado para acceder a orígenes de datos a través de JDBC

El servidor federado debe poder acceder a orígenes de datos JDBC. Para preparar el servidor federado, debe determinar si es necesario definir la variable de entorno CLASSPATH.

Antes de empezar

Si utiliza un controlador JDBC que no sea el controlador JDBC predeterminado de DB2 en el archivo db2jcc.jar, debe añadir la información del controlador JDBC en la variable de entorno CLASSPATH. Opcionalmente, puede especificar los paquetes de controlador JDBC con el parámetro DRIVER_PACKAGE de la sentencia CREATE SERVER al registrar las definiciones de servidor.

Procedimiento

Para definir la variable de entorno CLASSPATH:

Registre el archivo .jar Java que contiene el controlador JDBC en la variable de entorno CLASSPATH.

Opción	Descripción
Para Linux y UNIX	<p>Ejecute el mandato export para registrar el controlador JDBC. Por ejemplo, si especifica el controlador JDBC de DB2, ejecute el siguiente mandato:</p> <pre>export CLASSPATH=\$CLASSPATH:dir_instancia_db2 /sqllib/java/db2jcc.jar</pre> <p>donde <i>dir_instancia_db2</i> es la vía de acceso de archivo completa en la que se halla instalada la instancia del sistema de bases de datos DB2.</p>
Para Windows	<p>Establezca la variable de entorno CLASSPATH del sistema en el controlador JDBC:</p> <ol style="list-style-type: none"> 1. Inicie sesión como administrador. 2. Abra el Panel de control y seleccione Sistema → Variables de entorno. 3. Añada el nombre y el directorio del archivo del controlador JDBC en la variable de entorno CLASSPATH del sistema. Utilice un punto y coma para separar la nueva entrada de las entradas existentes. <p>Consulte la documentación del controlador JDBC para obtener información específica acerca del registro de variables de entorno del sistema.</p>

Una vez que haya finalizado esta tarea, deberá registrar el derivador.

Registro del derivador de JDBC

Debe registrar un derivador para acceder a los orígenes de datos JDBC. Los servidores federados utilizan los derivadores para comunicarse con los orígenes de datos y para recuperar datos de éstos. Los derivadores se implementan como un conjunto de archivos de biblioteca.

Procedimiento

Para registrar el derivador de JDBC:

Utilice uno de los métodos siguientes:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .

Método	Descripción
Ejecute la sentencia CREATE WRAPPER y especifique el nombre predeterminado del derivador JDBC.	<p>Por ejemplo: CREATE WRAPPER JDBC;</p> <p>Al registrar el derivador mediante el nombre predeterminado, JDBC, el servidor federado utiliza automáticamente la biblioteca de derivador de JDBC adecuada al sistema operativo en el que se ejecuta el servidor federado.</p>
Ejecute la sentencia CREATE WRAPPER y especifique un nombre alternativo para el derivador de JDBC.	<p>Si el nombre del derivador predeterminado entra en conflicto con un nombre de derivador existente en la base de datos federada, puede sustituir el nombre del derivador predeterminado por otro de su elección. Si utiliza un nombre que no es el predeterminado, deberá incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER.</p> <p>Por ejemplo, para registrar un derivador con el nombre <i>jdbc_Wrapper</i> en un servidor federado que utiliza AIX, ejecute esta sentencia: CREATE WRAPPER <i>jdbc_Wrapper</i> LIBRARY 'libdb2rcjdbc.a';</p> <p>El archivo de biblioteca del derivador especificado dependerá del sistema operativo del servidor federado.</p>

Una vez finalizada esta tarea, debe registrar las definiciones de servidor.

Archivos de biblioteca del derivador de JDBC

Los archivos de biblioteca de derivador de JDBC se añaden al servidor federado cuando se instala el derivador.

Cuando instala el derivador de JDBC, se añaden archivos de biblioteca a la vía de acceso del directorio predeterminado. Por ejemplo, si el servidor federado se ejecuta en AIX, los archivos de biblioteca del derivador que se añaden a la vía de acceso del directorio son *libdb2rcjdbc.a*, *libdb2rcjdbcF.a*, *libdb2rcjdbcU.a* y *db2qgjdbc.jar*. El archivo de biblioteca de derivador predeterminado es *libdb2rcjdbc.a*. Los demás archivos de biblioteca de derivador los utiliza internamente el derivador de JDBC.

Si no utiliza el nombre de derivador predeterminado al registrar un derivador, debe especificar el parámetro **LIBRARY** en la sentencia CREATE WRAPPER.

Utilice las siguientes vías de acceso de directorio y nombres de archivo de biblioteca de derivador predeterminados para especificar el parámetro **LIBRARY** en la sentencia CREATE WRAPPER:

Tabla 22. Vías de acceso de directorio y nombres de archivo de biblioteca de cliente JDBC

Sistema operativo	Vía de acceso al directorio	Archivo de biblioteca de derivador
AIX	<i>/usr/opt/vía_acceso_instalación/lib32/</i> <i>/usr/opt/vía_acceso_instalación/lib64/</i>	<i>libdb2rcjdbc.a</i>
Linux	<i>/opt/IBM/db2/vía_acceso_instalación/lib32</i> <i>/opt/IBM/db2/vía_acceso_instalación/lib64</i>	<i>libdb2rcjdbc.so</i>
Solaris	<i>/opt/IBM/db2/vía_acceso_instalación/lib32</i> <i>/opt/IBM/db2/vía_acceso_instalación/lib64</i>	<i>libdb2rcjdbc.so</i>
Windows	%DB2PATH%\bin	<i>db2rcjdbc.dll</i>

donde *vía_acceso_instalación* es la vía de acceso al directorio en el que está instalado Federation en UNIX o Linux.

Sentencia CREATE WRAPPER - Ejemplos para el derivador de JDBC

Utilice la sentencia CREATE WRAPPER para registrar el derivador de JDBC.

Para registrar un derivador con el nombre predeterminado en un servidor federado, ejecute la sentencia CREATE WRAPPER con el nombre del derivador de JDBC, por ejemplo:

```
CREATE WRAPPER JDBC;
```

En los ejemplos siguientes, *jdbc_Wrapper* es el nombre alternativo que se asigna al derivador que se registra en la base de datos federada.

Servidor federado Linux y Solaris

El ejemplo siguiente muestra cómo registrar un derivador con un nombre alternativo:

```
CREATE WRAPPER jdbc_Wrapper LIBRARY 'libdb2rcjdbc.so';
```

Servidor federado AIX

El ejemplo siguiente muestra cómo registrar un derivador con un nombre alternativo:

```
CREATE WRAPPER jdbc_Wrapper LIBRARY 'libdb2rcjdbc.a';
```

Servidor federado Windows

El ejemplo siguiente muestra cómo registrar un derivador con un nombre alternativo:

```
CREATE WRAPPER jdbc_Wrapper LIBRARY 'db2rcjdbc.dll';
```

Registro de definiciones de servidor para orígenes de datos JDBC

Debe registrar cada servidor JDBC al que desee acceder en la base de datos federada.

Antes de empezar

Si utiliza un controlador JDBC que no es el predeterminado del servidor DB2, puede que necesite establecer la variable de entorno CLASSPATH para especificar los archivos de paquete del controlador JDBC. Opcionalmente, puede especificar los paquetes del controlador JDBC en la sentencia CREATE SERVER con la opción de servidor DRIVER_PACKAGE.

Procedimiento

Para registrar el servidor JDBC, debe especificar el nombre de paquete del controlador JDBC de la biblioteca del controlador JDBC y la serie de conexión JDBC del servidor remoto.

Para registrar una definición de servidor para un origen de datos JDBC:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Ejecute la sentencia CREATE SERVER .	<p>Ejecute la sentencia CREATE SERVER para cada servidor JDBC que desee registrar; por ejemplo:</p> <pre>CREATE SERVER nombre_definición_servidor TYPE tipo_origen-datos_jdbc VERSION número_version WRAPPER nombre_derivador_jdbc OPTIONS (DRIVER_CLASS 'vía-acceso_clases_controlador_jdbc', URL 'serie_conexión_url_jdbc');</pre> <p>Importante: Al ejecutar la sentencia CREATE SERVER, ésta no crea realmente una conexión con el origen de datos hasta que se ejecuta la sentencia CREATE NICKNAME. Si especifica información de conexión incorrecta en el parámetro OPTIONS, se le notificará el error hasta que ejecute las sentencias CREATE NICKNAME o de paso a través.</p>

Una vez finalizada esta tarea, debe crear una correlación de usuario.

Sentencia CREATE SERVER - Ejemplos del derivador de JDBC

Utilice la sentencia CREATE SERVER para registrar definiciones del servidor para el derivador de JDBC. Este ejemplo proporciona los parámetros obligatorios y un ejemplo con parámetros de servidor adicionales.

El ejemplo siguiente muestra cómo registrar una definición de servidor para un origen de datos DB2 emitiendo la sentencia 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');
```

Valores de los parámetros

jdbc_server1

Especifica un nombre asignado por el usuario al servidor de orígenes de datos JDBC. No se permiten los nombres de definición de servidor duplicados.

TYPE JDBC

Especifica el tipo de servidor de orígenes de datos al que se desea acceder. Este parámetro es opcional.

VERSION 3.0

Especifica la versión del origen de datos JDBC al que se desea acceder. Este parámetro es opcional.

WRAPPER jdbc_wrapper1

Especifica el nombre del derivador que ha especificado en la sentencia CREATE WRAPPER.

DRIVER_PACKAGE *'/home/My_LIB/JDBC_driver/derbyclient.jar'*

Especifica los paquetes del controlador JDBC.

DRIVER_CLASS *'com.ibm.db2.jcc.DB2Driver'*

Especifica la biblioteca del controlador JDBC.

URL *'jdbc:db2://matthaus.cn.ibm.com:50471/testdb'*

Especifica la serie de conexión JDBC del servidor remoto.

Parámetros del servidor

Al crear la definición de servidor, puede especificar opciones de servidor adicionales en la sentencia CREATE SERVER. Las opciones de servidor pueden incluir opciones genéricas del servidor y parámetros de servidor específicos de JDBC.

En general, los valores predeterminados de los parámetros del servidor tienen funciones limitadas. Puede utilizar los parámetros del servidor para optimizar la configuración.

Para acceder a orígenes de datos JDBC, debe especificar los parámetros de servidor DRIVER_CLASS y URL en la sentencia CREATE SERVER. Los parámetros de servidor DRIVER_PACKAGE y JDBC_LOG son opcionales. La sintaxis del parámetro OPTIONS que figura a continuación especifica todos los parámetros de servidor específicos de 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');
```

Parámetros

DRIVER_PACKAGE *'/path1/file1.jar: /path2/file2.jar'*

Especifica los paquetes del controlador JDBC y establece la variable de entorno CLASSPATH.

DRIVER_CLASS *'com.ibm.db2.jcc.DB2Driver'*

Especifica la biblioteca del controlador JDBC de DB2.

URL *'jdbc:db2://server.example.com:50471/testdb'*

Especifica la serie de conexión JDBC, que consta de tres partes separadas por signos de dos puntos:

- El protocolo de la base de datos
- El nombre del tipo de base de datos o el nombre del controlador de conectividad
- La identidad de base de datos mediante un alias o un subnombre

Ejemplos

Para bases de datos DB2

```
jdbc:db2://server.example.com:50471/testdb
```

donde jdbc es el protocolo, db2 es el tipo de base de datos y //server.example.com:50471/testdb es el alias de base de datos que hace referencia a una entrada de catálogo de bases de datos DB2 del cliente DB2.

Para Oracle

```
jdbc:oracle:thin:@//myhost:1521/orcl
```


donde jdbc es el protocolo, oracle es el tipo de base de datos y thin:@//myhost:1521/orcl es la información de cliente JDBC Oracle para acceder al servidor Oracle.

JDBC_LOG 'Y'

Especifica si deben crearse archivos de registro para el rastreo de errores. El valor predeterminado de esta opción de servidor es N.

Ejemplo

```
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');
```

Creación de correlaciones de usuario para orígenes de datos JDBC

Debe definir una asociación (una correlación de usuario) entre cada ID de usuario del servidor federado y el ID de usuario correspondiente del origen de datos.

Acerca de esta tarea

Cuando se intenta acceder a un servidor JDBC, el servidor federado establece una conexión con el servidor JDBC utilizando un ID de usuario y una contraseña para ese origen de datos.

Cree una correlación de usuario para cada ID de usuario que acceda al sistema federado para enviar solicitudes distribuidas al origen de datos JDBC.

Procedimiento

Para crear correlaciones de usuario para orígenes de datos JDBC:

Ejecute la sentencia CREATE USER MAPPING para correlacionar un ID de usuario local con el ID de usuario y la contraseña del origen de datos JDBC:

```
CREATE USER MAPPING FOR IDusuario_local
  SERVER nombre_definición_servidor
  OPTIONS (
    REMOTE_AUTHID 'IDusuario_remoto',
    REMOTE_PASSWORD 'contraseña_remota');
```

Los parámetros de correlación de usuario REMOTE_AUTHID y REMOTE_PASSWORD son obligatorios.

Una vez finalizada esta tarea, puede probar la conexión al origen de datos JDBC.

Sentencia CREATE USER MAPPING - Ejemplos para el derivador de JDBC

Este ejemplo muestra cómo utilizar el registro especial de DB2 USER con la sentencia CREATE USER MAPPING.

El ejemplo siguiente muestra cómo correlacionar un ID de autorización federado con un ID de usuario y una contraseña de origen de datos JDBC:

```
CREATE USER MAPPING FOR arturo
  SERVER jdbc_server1
  OPTIONS (
    REMOTE_AUTHID 'art',
    REMOTE_PASSWORD 'red4blue');
```

Parámetros

SERVER arturo

Especifica el ID de autorización local que se correlaciona con el ID de usuario y la contraseña remotos definidos en el origen de datos JDBC.

OPTIONS jdbc_server1

Especifica el nombre de la definición del servidor que ha definido en la sentencia CREATE SERVER para el origen de datos JDBC.

REMOTE_AUTHID 'art'

Especifica el ID de usuario remoto con el que se correlaciona arturo. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca el parámetro de servidor FOLD_ID en 'U' o 'L' en la sentencia CREATE SERVER.

REMOTE_PASSWORD 'red4blue'

Especifica la contraseña remota asociada a 'art'. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_PW en 'U' o 'L' en la sentencia CREATE SERVER.

Registro especial USER de DB2

Puede utilizar el registro especial USER de DB2 para correlacionar el ID de autorización de la persona que emite la sentencia CREATE USER MAPPING con el ID de autorización del origen de datos especificado en la opción de usuario REMOTE_AUTHID.

A continuación se ofrece un ejemplo de la sentencia CREATE USER MAPPING, que incluye el registro especial USER:

```
CREATE USER MAPPING FOR USER
  SERVER jdbc_server1
  OPTIONS (
    REMOTE_AUTHID 'art',
    REMOTE_PASSWORD 'red4blue');
```

Prueba de las conexiones a los servidores de orígenes de datos JDBC

Puede probar la conexión al el servidor de orígenes de datos JDBC mediante la definición de servidor y las correlaciones de usuario que ha definido.

Procedimiento

Para probar la conexión al servidor de orígenes de datos JDBC:

Abra una sesión de paso a través y emita una sentencia SELECT en las tablas del sistema de orígenes de datos JDBC. Si la sentencia SELECT devuelve una cuenta, significa que la definición del servidor y las correlaciones de usuario están correctamente configuradas.

```
SET PASSTHRU nombre_definición_servidor
SELECT count(*) FROM nombre.esquema.nombre.tabla
SET PASSTHRU RESET
```

Si la sentencia SELECT devuelve un error, debe resolver los errores de conexión.

Una vez finalizada esta tarea, debe registrar apodos para las tablas y vistas de orígenes de datos JDBC.

Resolución de errores de conexión de origen de datos

Una conexión de prueba con el servidor de origen de datos puede devolver un error por varios motivos. Existen acciones que permiten determinar el motivo del error.

Síntoma

Se devuelve un error al intentar establecer conexión con el origen de datos.

Causa

Hay varias causas posibles que explican un problema de conexión.

Resolución del problema

Para solucionar los problemas de errores de conexión con el origen de datos, consulte los siguientes elementos:

- Verifique que el origen de datos está disponible.
- Si procede, asegúrese de que el servidor de origen de datos esté configurado para conexiones entrantes.
- Asegúrese de que los valores de correlación de usuarios para las opciones REMOTE_AUTHID y REMOTE_PASSWORD son válidos para las conexiones con el origen de datos. Modifique la correlación de usuarios o cree otra correlación de usuarios según sea necesario.
- Si procede, asegúrese de que el software de cliente del servidor federado esté instalado y configurado correctamente para conectarse al origen de datos.
- Para orígenes de datos ODBC, asegúrese de que el controlador ODBC del servidor federado esté instalado y configurado correctamente para establecer conexión con el servidor de origen de datos ODBC. En servidores federados que ejecutan Windows, utilice la herramienta ODBC Data Source Administrator para verificar el controlador. En servidores federados que ejecutan UNIX, consulte la documentación del proveedor del cliente ODBC.
- Verifique que los valores para las variables establecidas en el servidor federado son correctas para el origen de datos. Estas variables incluyen las variables de entorno del sistema, las variables de la archivo db2dj.ini y las variables del registro de perfiles de DB2 (db2set).
- Compruebe la definición del servidor. Si es necesario, elimine la definición del servidor y créela de nuevo.

Registro de apodos para vistas y tablas de orígenes de datos JDBC

Para cada definición de servidor JDBC que registre, debe registrar un apodo para cada tabla o vista a la que desee acceder. Utilice estos apodos en lugar de los nombres de los objetos de origen de datos cuando consulte los orígenes de datos JDBC.

Antes de empezar

Actualice las estadísticas del origen de datos JDBC antes de registrar un apodo. La base de datos federada se basa en las estadísticas del catálogo de orígenes de datos para optimizar el proceso de las consultas. Utilice el mandato de origen de datos equivalente al mandato de DB2 RUNSTATS para actualizar las estadísticas del origen de datos.

Procedimiento

Para registrar un apodo para una tabla o vista de origen de datos JDBC:

Utilice uno de los métodos siguientes:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Ejecute la sentencia CREATE NICKNAME.	<p>Por ejemplo:</p> <pre>CREATE NICKNAME apodo FOR nombre_definición_servidor."esquema_remoto". "tabla.remota";</pre> <p>Los apodos pueden tener una longitud de hasta 128 caracteres.</p>

Cuando se crea el apodo, el servidor federado consulta el catálogo de orígenes de datos. Esta consulta prueba la conexión a la tabla o vista del origen de datos. Si la conexión no funciona, recibirá un mensaje de error.

Repita este paso para cada tabla o vista JDBC para la que desee crear un apodo.

Sentencia CREATE NICKNAME - Ejemplos para el derivador de JDBC

El ejemplo muestra cómo registrar un apodo para una tabla o vista JDBC mediante la sentencia CREATE NICKNAME.

Esta sentencia especifica la definición de servidor y el esquema y tabla remotos:

```
CREATE NICKNAME cust_europe FOR jdbc_server."vinnie"."italy"
```

cust_europe

Apodo exclusivo utilizado para identificar la tabla o vista JDBC. El apodo debe ser exclusivo dentro del esquema.

Importante: El nombre de apodo consta de dos partes: el esquema y el apodo. Si omite el esquema al registrar el apodo, el esquema del apodo se establecerá en el ID de autorización del usuario que registra el apodo.

Si el origen de datos JDBC no da soporte a la utilización de esquemas, omita el esquema en la sentencia CREATE NICKNAME; por ejemplo:

```
CREATE NICKNAME cust_europe FOR jdbc_server."italy"
```

jdbc_server."vinnie"."italy"

Identificador de tres partes para el objeto remoto:

jdbc_server

Nombre de la definición de servidor asignado por el usuario al servidor de orígenes de datos JDBC en la sentencia CREATE SERVER.

vinnie ID de usuario del propietario al que pertenece la tabla o vista.

`italy` Nombre de la tabla o la vista remota a la que desea acceder.

El servidor federado convierte a mayúsculas los nombres de los esquemas y tablas JDBC, a menos que especifique los nombres entre comillas dobles.

Configuración del acceso a orígenes de datos Microsoft SQL Server

Para configurar el servidor federado para acceder a orígenes de datos Microsoft SQL Server, debe proporcionar al servidor federado información acerca de los orígenes de datos y objetos a los que desea acceder.

Antes de empezar

- El controlador ODBC debe estar instalado y configurado en el servidor federado.
- Federation debe estar instalado en el servidor que actúa como servidor federado.
- Compruebe la configuración del servidor federado.
- Compruebe el parámetro FEDERATED para asegurarse de que Federation está habilitado.

Puede configurar el servidor federado para acceder a orígenes de datos Microsoft SQL Server utilizando el Centro de control de DB2 o emitiendo sentencias SQL en la línea de mandatos de DB2.

Procedimiento

Para configurar el acceso a orígenes de datos de Microsoft SQL Server:

1. Utilice uno de los métodos siguientes para preparar el servidor federado y la base de datos federada dependiendo del sistema operativo.
 - Preparar el servidor federado y la base de datos federada (Windows)
 - Preparar el servidor federado y la base de datos federada (UNIX)
2. Establecer las variables de entorno para el derivador de Microsoft SQL Server
3. Registrar el derivador
4. Registrar la definición del servidor
5. Crear las correlaciones de usuario
6. Probar la conexión con el servidor remoto Microsoft SQL Server
7. Registrar los apodos para las tablas y vistas de Microsoft SQL Server

Preparación del servidor federado para acceder a orígenes de datos Microsoft SQL Server (Windows)

En servidores federados que ejecutan Windows, el servidor federado debe poder acceder a orígenes de datos Microsoft SQL Server. Para preparar el servidor federado, debe comprobar los valores de DSN de sistema ODBC y probar la conexión a los orígenes de datos Microsoft SQL Server.

Procedimiento

Para preparar el servidor federado para acceder a orígenes de datos de Microsoft SQL Server:

1. Verifique que el DSN de sistema ODBC esté establecido para conectarse al origen de datos Microsoft SQL Server. En el Panel de control, localice la entrada de DSN existente para el servidor remoto Microsoft SQL Server o cree una entrada DSN.

La entrada de DSN para el servidor remoto Microsoft SQL Server es el valor que utilizará para la opción de servidor NODE al registrar la definición de servidor en la base de datos federada.

2. Utilice uno de los métodos siguientes para probar la conexión al origen de datos Microsoft SQL Server:
 - Seleccione **Configurar** en la ventana Administrador de orígenes de datos ODBC.
 - Utilice la herramienta de consultas de Microsoft SQL Server.

Una vez finalizada esta tarea, puede establecer las variables de entorno.

Preparación del servidor federado para acceder a orígenes de datos de Microsoft SQL Server (Linux, UNIX)

En servidores federados que ejecutan Linux o UNIX, el servidor federado debe poder acceder a los orígenes de datos de Microsoft SQL Server. Para preparar el servidor federado, debe verificar los valores en el archivo `odbc.ini`, crear enlaces simbólicos y probar la conexión con los orígenes de datos de Microsoft SQL Server.

Procedimiento

Para preparar el servidor federado para acceder a orígenes de datos de Microsoft SQL Server:

1. Verifique que el archivo `odbc.ini` se ha actualizado en el servidor federado. Si el archivo `odbc.ini` no existe en el servidor federado, puede crearlo en un editor de texto. Consulte la documentación del proveedor del cliente ODBC para obtener información sobre el archivo `odbc.ini`.

Recuerde: Coloque el archivo `odbc.ini` o una copia del mismo en el directorio padre del propietario de la instancia de DB2 para garantizar que es posible acceder al archivo si el propietario de la instancia no es el propietario raíz.

2. Verifique que la vía de acceso al archivo `odbc.ini` se encuentra en la variable de entorno `ODBCINI`.

Desde un indicador de mandatos del sistema operativo, emita el siguiente mandato:

```
export ODBCINI=$HOME/.odbc.ini
```

3. Cree los enlaces simbólicos apropiados:

Sistema operativo de servidor federado	Paso
Linux	<p>Cree los siguientes enlaces simbólicos:</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 utiliza DataDirect Technologies Connect para el controlador ODBC, verifique el nombre de la biblioteca y cree el enlace simbólico. El nombre de la biblioteca varía en función de la versión del controlador y de si se utiliza un controlador de 32 bits o de 64 bits.</p> <p>Por ejemplo, si utiliza DataDirect versión 4.2, deberá crear este enlace:</p> <pre>ln -s \$DJX_ODBC_LIBRARY_PATH/libivicu19.so/ usr/lib/libivicu19.so</pre> <p>Si utiliza DataDirect versión 5.0, deberá crear este enlace:</p> <pre>ln -s \$DJX_ODBC_LIBRARY_PATH/libivicu20.so/ usr/lib/libivicu20.so</pre> <p>Si utiliza DataDirect y no incluye el enlace simbólico, es posible que CREATE WRAPPER MSSQLODBC3 falle con el siguiente mensaje de error:</p> <p>SQL10013N The specified library name could not be loaded.</p>
Solaris	<p>Cree el siguiente enlace simbólico:</p> <pre>ln -s \$DJX_ODBC_LIBRARY_PATH/./ locale \$HOME/sql1lib/locale</pre> <p>\$HOME el el directorio padre del propietario de la instancia de DB2.</p>

4. Ejecute el script `/opt/odbc/odbc.sh`. Este script configura varias variables de entorno específicas del sistema operativo.
5. Pruebe la conexión desde el servidor federado al origen de datos de Microsoft SQL Server mediante el programa de utilidad `demojdbc` de DataDirect Connect ODBC. El programa de utilidad `demojdbc` se encuentra en el subdirectorio `/demo` de las bibliotecas de DataDirect Connect ODBC.

Tras completar esta tarea, podrá definir las variables de entorno.

Establecimiento de variables de entorno de Microsoft SQL Server

Las variables de entorno de Microsoft SQL Server deben establecerse en el archivo `db2dj.ini` del servidor federado.

Restricciones

Revise las restricciones concernientes al archivo db2dj.ini antes de iniciar esta tarea.

El archivo db2dj.ini contiene información de configuración relativa al controlador ODBC de Microsoft SQL Server que está instalado en el servidor federado.

Existen variables de entorno obligatorias y opcionales para los orígenes de datos Microsoft SQL Server.

Si ha instalado el software del cliente Microsoft SQL Server antes de instalar el derivador de Microsoft SQL Server, las variables de entorno obligatorias de Microsoft SQL Server se establecen en el archivo db2dj.ini.

Debe establecer las variables de entorno siguiendo los pasos de esta tarea si no ha instalado el software del cliente Microsoft SQL Server antes de instalar el derivador de Microsoft SQL Server o si desea establecer alguna de las variables de entorno opcionales.

Procedimiento

Para establecer las variables de entorno de Microsoft SQL Server:

1. Utilice uno de los métodos siguientes:

Método	Paso
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Establecer automáticamente las variables de entorno.	Ejecute el asistente de instalación de IBM InfoSphere Federation Server. Siga las instrucciones del asistente. Importante: Establezca las variables de entorno obligatorias ejecutando el asistente de instalación. Las variables de entorno opcionales deben establecerse manualmente.
Establecer manualmente las variables de entorno.	Edite el archivo db2dj.ini. <ul style="list-style-type: none">• En servidores federados que ejecutan UNIX, este archivo se encuentra en el directorio sql1lib/cfg.• En servidores federados que ejecutan Windows, este archivo se encuentra en el directorio %DB2PATH%\cfg. Si el archivo no existe, puede crear un archivo con el nombre db2dj.ini mediante cualquier editor de texto. En el archivo db2dj.ini, debe especificar la vía de acceso completa en el valor de las variables de entorno; de lo contrario, se producirán errores. Por ejemplo: DJX_ODBC_LIBRARY_PATH=/opt/odbc/lib ODBCINI=/opt/odbc/.odbc.ini

2. Para asegurarse de que las variables de entorno están establecidas en el servidor federado, reinicie la instancia de DB2 con estos mandatos:

```
db2stop  
db2start
```

Una vez finalizada esta tarea, puede registrar el derivador.

Variables de entorno de Microsoft SQL Server

Existen variables de entorno necesarias y variables de entorno opcionales para los orígenes de datos Microsoft SQL Server. Estas variables se definen en el archivo `db2dj.ini`.

Las variables de entorno siguientes son válidas para Microsoft SQL Server:

- `DJX_ODBC_LIBRARY_PATH`
- `ODBCINI`
- `LD_LIBRARY_PATH` (sólo Solaris)

Descripción de las variables

`DJX_ODBC_LIBRARY_PATH`

Especifica la vía de acceso de directorio de los archivos de biblioteca ODBC. Esta variable también debe especificarse en los servidores federados que ejecutan Solaris.

Por ejemplo:

```
DJX_ODBC_LIBRARY_PATH=directorio_controlador_ODBC/lib
```

directorio_controlador_ODBC es la vía de acceso de directorio en la que se ha instalado el controlador ODBC.

`ODBCINI`

Especifica la vía de acceso de directorio en la que se encuentra el archivo de configuración de ODBC (`odbc.ini`).

Por ejemplo:

```
ODBCINI=/home/db2inst1/.odbc.ini
```

No establezca la variable de entorno `ODBCINI` como variable del sistema.

`LD_LIBRARY_PATH` (sólo Solaris)

En servidores federados que ejecutan Solaris, especifica la vía de acceso de directorio de los archivos de biblioteca ODBC.

Por ejemplo:

```
LD_LIBRARY_PATH=directorio_controlador_ODBC/lib
```

Registro del derivador de Microsoft SQL Server

Debe registrar un derivador para acceder a los orígenes de datos de Microsoft SQL Server. Los servidores federados utilizan los derivadores para comunicarse con los orígenes de datos y para recuperar datos de éstos. Los derivadores se implementan como un conjunto de archivos de biblioteca.

Procedimiento

Para registrar el derivador de Microsoft SQL Server:

Utilice uno de los métodos siguientes:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .

Método	Descripción
<p>Emita la sentencia CREATE WRAPPER y especifique el nombre predeterminado para el derivador de Microsoft SQL Server.</p>	<p>Por ejemplo: <code>CREATE WRAPPER MSSQLODBC3;</code></p> <p>Recuerde: Al registrar el derivador mediante el nombre predeterminado, MSSQLODBC3, el servidor federado utiliza automáticamente la biblioteca de derivador de Microsoft SQL Server adecuada al sistema operativo en el que se ejecuta el servidor federado.</p> <p>Si el nombre del derivador predeterminado entra en conflicto con un nombre de derivador existente en la base de datos federada, puede sustituir el nombre del derivador predeterminado por otro de su elección. Si utiliza un nombre que no es el predeterminado, deberá incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER.</p> <p>Por ejemplo, para registrar un derivador con el nombre <code>derivador_sqlserver</code> en un servidor federado que utiliza AIX, emita esta sentencia: <code>CREATE WRAPPER derivador_sqlserver LIBRARY 'libdb2mssql13.a';</code></p> <p>El archivo de biblioteca del derivador especificado dependerá del sistema operativo del servidor federado.</p>

Una vez finalizada esta tarea, puede registrar la definición de servidor.

Archivos de biblioteca del derivador de Microsoft SQL Server

Los archivos de biblioteca de derivador de Microsoft SQL Server se añaden al servidor federado cuando se instala el derivador.

Cuando instala el derivador de Microsoft SQL Server, se añaden tres archivos de biblioteca a la vía de acceso del directorio predeterminado. Por ejemplo, si el servidor federado se ejecuta en AIX, los archivos de biblioteca del derivador que se añaden a la vía de acceso del directorio son `libdb2mssql13.a`, `libdb2mssql13F.a` y `libdb2mssql13U.a`. El archivo de biblioteca de derivador predeterminado es `libdb2mssql13.a`. Los demás archivos de biblioteca de derivador los utiliza internamente el derivador de Microsoft SQL Server.

Si no utiliza el nombre de derivador por omisión cuando registra un derivador, debe incluir el parámetro **LIBRARY** en la sentencia **CREATE WRAPPER** y especificar el nombre de archivo de biblioteca del derivador por omisión.

En la tabla siguiente se indican las vías de acceso al directorio predeterminado y los nombres de archivo de biblioteca del derivador predeterminado.

Tabla 23. Ubicaciones de biblioteca y nombres de archivo del cliente Microsoft SQL Server

Sistema operativo	Vía de acceso al directorio	Nombre de archivo de biblioteca
AIX	/usr/opt/vía_acceso_instalación/lib32/ /usr/opt/vía_acceso_instalación/lib64/	libdb2mssql3.a
Linux	/opt/IBM/db2/vía_acceso_instalación/lib32 /opt/IBM/db2/vía_acceso_instalación/lib64	libdb2mssql3.so
Solaris	/opt/IBM/db2/vía_acceso_instalación/lib32 /opt/IBM/db2/vía_acceso_instalación/lib64	libdb2mssql3.so
Windows	%DB2PATH%\bin	db2mssql3.dll

vía_acceso_instalación es la vía de acceso al directorio en el que está instalado el servidor federado en UNIX o Linux.

Registro de las definiciones de servidor para un origen de datos Microsoft SQL Server

Debe registrar cada servidor remoto Microsoft SQL Server al que desee acceder en la base de datos federada.

Procedimiento

Para registrar una definición de servidor para un origen de datos Microsoft SQL Server:

1. Localice el nombre de nodo de Microsoft SQL Server.
 - En servidores federados que ejecutan Windows, el nombre de nodo es el nombre DSN de sistema que ha especificado para el servidor remoto Microsoft SQL Server al que accede.
 - En servidores federados que ejecutan UNIX, el nombre de nodo se define en el archivo `.odbc.ini`.

Al principio del archivo `.odbc.ini`, hay una sección denominada ODBC Data Sources, que lista los nodos. Cada uno de los nodos contiene una sección en el archivo `.odbc.ini` que lo describe.

El ejemplo siguiente es un archivo `.odbc.ini` de AIX. Los nombres de nodo son `[rawilson]` y `[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. Utilice uno de los métodos siguientes para crear la definición del servidor:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .

Método	Descripción
Emita la sentencia CREATE SERVER .	<p>Por ejemplo:</p> <pre>CREATE SERVER nombre_definición_servidor TYPE MSSQLSERVER VERSION númeroversión WRAPPER nombre_derivador OPTIONS (NODE 'nombre_nodo', DBNAME 'nombre_base_datos');</pre> <p>Aunque las variables '<i>nombre_nodo</i>' y '<i>nombre_base_datos</i>' se especifican como opciones en la sentencia CREATE SERVER, estas opciones son necesarias para los orígenes de datos Microsoft SQL Server.</p> <p>Una vez registrada la definición del servidor, utilice la sentencia ALTER SERVER para añadir o eliminar opciones de servidor.</p>

Una vez finalizada esta tarea, puede crear correlaciones de usuario.

Sentencia **CREATE SERVER** - Ejemplos para el derivador de Microsoft SQL Server

Utilice la sentencia **CREATE SERVER** para registrar definiciones de servidor para el derivador de Microsoft SQL Server. Este tema ofrece un ejemplo completo con los parámetros obligatorios y un ejemplo con opciones adicionales del servidor.

El ejemplo siguiente muestra cómo registrar una definición de servidor para un derivador de Microsoft SQL Server emitiendo la sentencia **CREATE SERVER**:

```
CREATE SERVER sqlserver TYPE MSSQLSERVER VERSION 2000 WRAPPER nombre_derivador
OPTIONS (NODE 'sqlnode', DBNAME 'africa');
```

sqlserver

Nombre asignado por el usuario al servidor remoto Microsoft SQL Server.
No se permiten los nombres de definición de servidor duplicados.

TYPE MSSQLSERVER

Especifica el tipo de origen de datos para el que está configurando el acceso. Para el derivador de Microsoft SQL Server, el tipo de servidor debe ser **MSSQLSERVER**.

VERSION 2000

La versión del servidor de bases de datos de Microsoft SQL Server al que desea acceder.

WRAPPER nombre_derivador

El nombre del derivador que ha especificado en la sentencia **CREATE WRAPPER**.

NODE 'sqlnode'

Nombre del nodo en el que reside el servidor remoto Microsoft SQL Server. En servidores federados que ejecutan Windows, el nombre DSN de sistema para el servidor remoto Microsoft SQL Server al que accede. En servidores federados que ejecutan UNIX, el nodo definido en el archivo `.odbc.ini`.

Este valor es sensible a las mayúsculas y minúsculas.

Aunque el nombre del nodo se especifica como opción en la sentencia CREATE SERVER, es necesario para los orígenes de datos de Microsoft SQL Server.

DBNAME 'africa'

El nombre de la base de datos de Microsoft SQL Server a la que desea acceder. Este valor es sensible a las mayúsculas y minúsculas.

Aunque el nombre de la base de datos se especifica como opción en la sentencia CREATE SERVER, es necesario para los orígenes de datos de Microsoft SQL Server.

Opciones de servidor

Al crear una definición de servidor, puede especificar opciones de servidor adicionales en la sentencia CREATE SERVER. Las opciones de servidor pueden ser genéricas y específicas de Microsoft SQL Server.

La opción de servidor COLLATING_SEQUENCE especifica si el origen de datos utiliza la misma secuencia de clasificación que el servidor federado u otra secuencia de clasificación. En un servidor de bases de datos Microsoft SQL Server que ejecute Windows NT® o Windows 2000, la secuencia de clasificación predeterminada no es sensible a las mayúsculas y minúsculas (por ejemplo, 'STEWART' y 'Stewart' se consideran iguales). Para garantizar resultados correctos del servidor federado, establezca la opción de servidor COLLATING_SEQUENCE en 'I'. Este valor indica que el origen de datos Microsoft SQL Server es insensible a las mayúsculas y minúsculas.

El servidor federado no envía consultas si los resultados devueltos desde los orígenes de datos son diferentes de los resultados devueltos al procesar la consulta en el servidor federado. Al establecer la opción de servidor COLLATING_SEQUENCE en 'I', el servidor federado no envía las consultas que contienen datos o expresiones tipo serie y que también contienen alguna de las siguientes cláusulas, predicados o funciones:

- Cláusulas GROUP BY
- Cláusulas DISTINCT
- Predicados básicos, tales como igual a (=)
- Funciones agregadas, tales como MIN o MAX

El ejemplo siguiente muestra cómo especificar la opción de servidor COLLATING_SEQUENCE:

```
CREATE SERVER sqlserver TYPE MSSQLSERVER VERSION 2000 WRAPPER mssqlodbc3
  OPTIONS (NODE 'sqlnode', DBNAME 'africa', COLLATING_SEQUENCE 'I');
```

Creación de las correlaciones de usuario para un origen de datos Microsoft SQL Server

Cuando se intenta acceder a un servidor remoto Microsoft SQL Server, el servidor federado establece una conexión con el servidor remoto Microsoft SQL Server utilizando un ID de usuario y una contraseña válidos para ese origen de datos.

Procedimiento

Para correlacionar un ID de autorización local con un ID de usuario y una contraseña remotos de Microsoft SQL Server:

Emita una sentencia CREATE USER MAPPING.

Por ejemplo:

```
CREATE USER MAPPING FOR IDusuario_local SERVER nombre_definición_servidor
    OPTIONS (REMOTE_AUTHID 'IDusuario_remoto', REMOTE_PASSWORD 'contraseña_remota');
```

Una vez finalizada esta tarea, puede probar la conexión a las tablas y vistas de Microsoft SQL Server.

Sentencia CREATE USER MAPPING - Ejemplos para el derivador de Microsoft SQL Server

Utilice la sentencia CREATE USER MAPPING para correlacionar un ID de autorización federado con un ID de usuario y una contraseña remotos de Microsoft SQL Server. Este tema ofrece un ejemplo completo con los parámetros obligatorios y un ejemplo que muestra cómo utilizar el registro especial USER de DB2 con la sentencia CREATE USER MAPPING.

El ejemplo siguiente muestra cómo correlacionar un ID de autorización federado con un ID de usuario y una contraseña de servidor remoto Microsoft SQL Server:

```
CREATE USER MAPPING FOR elizabeth SERVER sqlserver
    OPTIONS (REMOTE_AUTHID 'liz', REMOTE_PASSWORD 'abc123');
```

elizabeth

Especifica el ID de autorización que se correlaciona con un ID de usuario y una contraseña remotos definidos en el servidor remoto Microsoft SQL Server.

SERVER *sqlserver*

Especifica el nombre de la definición del servidor que ha registrado en la sentencia CREATE SERVER para el servidor remoto Microsoft SQL Server.

REMOTE_AUTHID '*liz*'

Especifica el ID de usuario remoto con el que se correlaciona *elizabeth*. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_ID en 'U' o 'L' en la sentencia CREATE SERVER.

REMOTE_PASSWORD '*abc123*'

Especifica la contraseña remota asociada a '*liz*'. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_PW en 'U' o 'L' en la sentencia CREATE SERVER.

Registro especial USER de DB2

Puede utilizar el registro especial USER de DB2 para correlacionar el ID de autorización de la persona que emite la sentencia CREATE USER MAPPING con el ID de usuario remoto especificado en la opción REMOTE_AUTHID.

A continuación se ofrece un ejemplo de la sentencia CREATE USER MAPPING, que incluye el registro especial USER:

```
CREATE USER MAPPING FOR USER SERVER sqlserver
    OPTIONS (REMOTE_AUTHID 'liz', REMOTE_PASSWORD 'abc123');
```

Prueba de la conexión con el servidor remoto Microsoft SQL Server

Pruebe la conexión con el servidor remoto Microsoft SQL Server para determinar si el servidor federado se ha configurado correctamente para acceder a orígenes de datos Microsoft SQL Server.

Puede probar la conexión con el servidor remoto Microsoft SQL Server mediante la definición de servidor y las correlaciones de usuario que ha definido.

Procedimiento

Para probar la conexión con el servidor remoto Microsoft SQL Server:

Abra una sesión de paso a través y emita una sentencia SELECT en las tablas del sistema Microsoft SQL Server. Si la sentencia SELECT devuelve una cuenta, significa que la definición del servidor y las correlaciones de usuario están correctamente configuradas.

Por ejemplo:

```
SET PASSTHRU nombre_definición_servidor
SELECT count(*) FROM dbo.sysobjects
SET PASSTHRU RESET
```

Una vez finalizada esta tarea, puede registrar apodos.

Resolución de errores de conexión de origen de datos

Una conexión de prueba con el servidor de origen de datos puede devolver un error por varios motivos. Existen acciones que permiten determinar el motivo del error.

Síntoma

Se devuelve un error al intentar establecer conexión con el origen de datos.

Causa

Hay varias causas posibles que explican un problema de conexión.

Resolución del problema

Para solucionar los problemas de errores de conexión con el origen de datos, consulte los siguientes elementos:

- Verifique que el origen de datos está disponible.
- Si procede, asegúrese de que el servidor de origen de datos esté configurado para conexiones entrantes.
- Asegúrese de que los valores de correlación de usuarios para las opciones REMOTE_AUTHID y REMOTE_PASSWORD son válidos para las conexiones con el origen de datos. Modifique la correlación de usuarios o cree otra correlación de usuarios según sea necesario.
- Si procede, asegúrese de que el software de cliente del servidor federado esté instalado y configurado correctamente para conectarse al origen de datos.
- Para orígenes de datos ODBC, asegúrese de que el controlador ODBC del servidor federado esté instalado y configurado correctamente para establecer conexión con el servidor de origen de datos ODBC. En servidores federados que ejecutan Windows, utilice la herramienta ODBC Data Source Administrator para verificar el controlador. En servidores federados que ejecutan UNIX, consulte la documentación del proveedor del cliente ODBC.
- Verifique que los valores para las variables establecidas en el servidor federado son correctas para el origen de datos. Estas variables incluyen las variables de entorno del sistema, las variables de la archivo db2dj.ini y las variables del registro de perfiles de DB2 (db2set).

- Compruebe la definición del servidor. Si es necesario, elimine la definición del servidor y créela de nuevo.

Registro de apodos para vistas y tablas de Microsoft SQL Server

Para cada definición de servidor Microsoft SQL Server que registre, debe registrar un apodo para cada tabla o vista a la que desee acceder. Utilice estos apodos en lugar de los nombres de los objetos de origen de datos cuando consulte los servidores remotos Microsoft SQL Server.

Antes de empezar

Para asegurarse de que la base de datos federada contiene estadísticas actualizadas y completas, ejecute el procedimiento almacenado `sp_createstats` de Microsoft SQL Server y el mandato `CREATE STATISTICS` de Microsoft SQL Server desde la base de datos de Microsoft SQL Server antes de crear el apodo.

El procedimiento almacenado `sp_createstats` reúne estadísticas de todas las columnas predeterminadas de una tabla de origen de datos Microsoft SQL Server, pero no reúne estadísticas de las columnas que aparecen en primer lugar en un índice. Para asegurarse de que la base de datos federada contiene estadísticas completas de la tabla de Microsoft SQL Server, también debe utilizar el mandato `CREATE STATISTICS` de Microsoft SQL Server para reunir estadísticas para cada columna que aparezca en primer lugar en un índice.

Al utilizar el mandato `CREATE STATISTICS` desde la base de datos de Microsoft SQL Server, debe dar a la estadística el mismo nombre que la columna en la que se recogen las estadísticas. Al dar a la estadística el mismo nombre que la columna, se asegura de que, al registrar el apodo con la sentencia `CREATE NICKNAME`, la base de datos federada lea las estadísticas recogidas por el mandato `CREATE STATISTICS` de Microsoft SQL Server.

Procedimiento

Para registrar un apodo para una tabla o vista de Microsoft SQL Server:

Utilice uno de los métodos siguientes:

Método	Descripción
Utilizar el Centro de control.	Inicie el asistente Objetos federados. Pulse la carpeta Objetos de base de datos federada con el botón derecho del ratón y pulse Crear objetos federados . Siga los pasos del asistente.
Emita la sentencia <code>CREATE NICKNAME</code> .	Por ejemplo: <pre>CREATE NICKNAME apodo FOR nombre_definición_servidor."esquema_remoto". "tabla.remota";</pre>

Cuando se crea el apodo, el servidor federado consulta el catálogo de orígenes de datos utilizando el apodo. Esta consulta prueba la conexión a la tabla, vista o sinónimo del origen de datos. Si la conexión no funciona, recibirá un mensaje de error.

Repita este paso para cada tabla o vista de Microsoft SQL Server para la que desee crear un apodo.

Sentencia CREATE NICKNAME - Ejemplos para el derivador de Microsoft SQL Server

Utilice la sentencia CREATE NICKNAME para registrar un apodo para una tabla o vista de Microsoft a la que desea acceder. Este tema ofrece un ejemplo completo con los parámetros obligatorios.

El ejemplo siguiente muestra cómo registrar un apodo para una tabla o vista de Microsoft SQL Server mediante la sentencia CREATE NICKNAME.

```
CREATE NICKNAME cust_africa FOR sqlserver."vinnie"."egypt"
```

cust_africa

Apodo exclusivo utilizado para identificar la tabla o vista de Microsoft SQL Server.

Importante: El nombre de apodo consta de dos partes: el esquema y el apodo. Si omite el esquema al registrar el apodo, el esquema del apodo será el ID de autorización del usuario que registra el apodo.

sqlserver."vinnie"."egypt"

Es un identificador de tres partes para el objeto remoto:

- *sqlserver* es el nombre de definición de servidor que ha asignado al servidor remoto Microsoft SQL Server en la sentencia CREATE SERVER.
- *vinnie* es el ID de usuario del propietario al que pertenece la tabla o vista.
- *egypt* es el nombre de la tabla o la vista remota a la que desea acceder.

El servidor federado convierte a mayúsculas los nombres de los esquemas y tablas Microsoft SQL Server, a menos que especifique los nombres entre comillas.

Utilización de la información de rastreo de ODBC para resolver los problemas de conexión a orígenes de datos Microsoft SQL Server

Si experimenta problemas de conexión con el origen de datos, puede obtener información de rastreo de ODBC para analizar y resolver los problemas.

Síntoma

Sin embargo, la activación de un rastreo influye sobre el rendimiento del sistema. Debe desactivar el rastreo una vez resueltos los problemas de conectividad.

Si no puede conectarse al origen de datos con el derivador de Microsoft SQL Server, la ejecución de un rastreo puede facilitar el diagnóstico del problema.

Causa

La causa del problema puede ser un error en la configuración del derivador.

Diagnóstico del problema

Para diagnosticar el problema en un servidor federado que ejecuta Windows:

1. En el Panel de control, abra la carpeta **Herramientas administrativas**.
2. Pulse **Orígenes de datos (ODBC)** para abrir la ventana Administrador de orígenes de datos ODBC.

3. Pulse el separador Rastreo.
4. Pulse **Iniciar rastreo ahora** para iniciar el programa de utilidad de rastreo.

En un servidor federado que ejecuta UNIX:

1. cambie el archivo `odbc.ini`.

Por ejemplo, si utiliza el controlador DataDirect ODBC 3.x, busque el ejemplo del archivo `odbc.ini` en el directorio del cliente. El archivo `odbc.ini` contiene un ejemplo de los valores necesarios para activar los archivos de rastreo:

```
[ODBC]
Trace=1
TraceFile=/home/user1/trace_dir/filename.xxx
TraceDll==ODBC_driver_directory/odbc32.so
InstallDir=/opt/odbc
```

Para activar el rastreo, establezca la primera línea en `Trace=1`. Para desactivarlo, establezca la primera línea en `Trace=0`. El valor del `TraceFile` es la vía de acceso y el nombre del archivo sobre el que la instancia de la base de datos federada tiene acceso de escritura.

Resolución del problema

Compruebe si figuran problemas en el archivo de registro de rastreo.

En Windows, abra el Administrador de orígenes de datos ODBC y pulse el separador Rastreo. La vía de acceso al archivo de registro de rastreo se muestra en el campo Vía de acceso de archivo de registro.

En UNIX, abra el archivo `odbc.ini`. La vía de acceso al archivo de registro de rastreo se indica en el valor `TraceFile`.

Configuración del acceso a orígenes de datos ODBC

Para configurar el servidor federado para acceder a orígenes de datos ODBC, debe proporcionar al servidor federado información acerca de los orígenes de datos y objetos a los que desea acceder.

Antes de empezar

- El controlador ODBC debe estar instalado y configurado en el sistema que actúa como servidor federado.
- Federation debe estar instalado en el servidor que actúa como servidor federado.
- Compruebe la configuración del servidor federado.
- Compruebe el parámetro `FEDERATED` para asegurarse de que Federation está habilitado.
- Configuración adecuada de las variables de entorno del sistema, las variables del archivo `db2dj.ini` y las variables del registro de perfiles de DB2 (`db2set`). Consulte la documentación suministrada por el origen de datos ODBC para conocer las variables necesarias para el cliente ODBC. Es posible que sea necesaria la variable de entorno `LIBPATH`.

Restricciones

- El derivador de ODBC no puede utilizarse para acceder a orígenes de datos de la familia DB2. Utilice el derivador de DRDA para acceder a orígenes de datos de la familia DB2.
- El derivador de ODBC no da soporte a las funciones y sentencias siguientes:

- Sentencias LOCK TABLE en apodos
- Características obsoletas en ODBC 3.x
- Controladores X/Open o SQL/CLI
- Apodos de procedimientos almacenados
- Aplicación de atomicidad a nivel de sentencia mediante sentencias de punto de recuperación remoto
- Cursores WITH HOLD
- En orígenes de datos que no admitan operaciones de actualización y supresión posicionadas, las sentencias UPDATE y DELETE posicionadas y determinadas sentencias UPDATE y DELETE buscadas en un apodo fallarán si no existe un índice exclusivo en columnas sin capacidad para nulos en el apodo o en su tabla remota correspondiente. Se devolverá el error SQL30090 con el código de razón 21 cuando fallen estas sentencias.
- El derivador de ODBC no da soporte a sentencias INSERT, UPDATE o DELETE en orígenes de datos que restringen el número de sentencias activas para cada conexión. Consulte la documentación de su origen de datos para determinar si éste restringe el número de sentencias activas para cada conexión. Uno de los orígenes de datos ODBC a los que se aplica esta restricción es IBM Red Brick Warehouse.
- El derivador de ODBC no da soporte a operaciones en tablas que contienen columnas con tipos de datos que utilizan indicadores de tipo de datos SQL específicos del controlador. Los tipos de operaciones no soportadas incluyen las sentencias CREATE NICKNAME y SELECT en la modalidad de paso a través. El derivador de ODBC sólo da soporte a los indicadores de tipos de datos SQL definidos por el estándar ODBC en la publicación *Microsoft ODBC Programmer's Reference*.

Acerca de esta tarea

Puede configurar el servidor federado para acceder a orígenes de datos ODBC utilizando el Centro de control de DB2 o emitiendo sentencias SQL en la línea de mandatos de DB2. El Centro de control de DB2 incluye un asistente que le guiará en los pasos necesarios para configurar los objetos federados necesarios.

En este documento, los orígenes de datos a los que se accede mediante la API de ODBC se denominan orígenes de datos ODBC.

En función de sus necesidades, puede acceder a datos Excel mediante el derivador de ODBC en lugar de utilizar el derivador de Excel. Para configurar el derivador de ODBC para el acceso a datos Excel, consulte la sección Acceso a datos Excel mediante el derivador de ODBC.

Recomendación: En Microsoft SQL Server, debe utilizar el derivador de Microsoft SQL Server para acceder a orígenes de datos Microsoft SQL Server en lugar de utilizar el derivador de ODBC. El derivador de Microsoft SQL Server ofrece un mejor rendimiento de consultas y más funciones para Microsoft SQL Server. Para obtener más información acerca de la configuración del derivador de Microsoft SQL Server, consulte la sección Configuración del acceso a orígenes de datos Microsoft SQL Server.

Procedimiento

Para añadir orígenes de datos ODBC a un servidor federado:

1. Utilice uno de los métodos siguientes para preparar el servidor federado y la base de datos federada, dependiendo del sistema operativo:
 - Preparar el servidor federado para acceder a orígenes de datos a través de ODBC (Windows).
 - Preparar el servidor federado para acceder a orígenes de datos a través de ODBC (Linux, UNIX).
2. Registrar el derivador de ODBC.
3. Registrar las definiciones de servidor para un origen de datos ODBC.
4. Crear una correlación de usuario para un origen de datos ODBC.
5. Probar la conexión con el servidor de orígenes de datos ODBC.
6. Registrar apodos para las tablas y vistas del origen de datos ODBC.

Preparación del servidor federado para acceder a orígenes de datos a través de ODBC (Windows)

En servidores federados que ejecutan Windows, el servidor federado debe poder acceder a orígenes de datos ODBC. Para preparar el servidor federado, debe comprobar los valores de DSN de sistema ODBC.

Procedimiento

Para preparar el servidor federado para acceder a orígenes de datos a través de ODBC:

Compruebe que el controlador ODBC 3.x está instalado y configurado en el servidor federado.

El nombre de nodo del origen de datos ODBC debe definirse en el DSN de sistema. Consulte la documentación del controlador ODBC para conocer los procedimientos de instalación y configuración.

Si ha utilizado el Administrador de orígenes de datos ODBC de Microsoft para configurar el DSN, puede comprobar este valor en el Panel de control. Asegúrese de que el origen de datos ODBC esté registrado como DSN de sistema. De lo contrario, es posible que el servidor federado no pueda encontrar el DSN.

Una vez finalizada esta tarea, puede registrar el derivador.

Preparación del servidor federado para acceder a orígenes de datos a través de ODBC (Linux, UNIX)

En servidores federados que ejecutan Linux o UNIX, el servidor federado debe poder acceder a orígenes de datos ODBC. Para preparar el servidor federado, debe verificar los valores del archivo `odbc.ini`, crear enlaces simbólicos y probar la conexión con orígenes de datos ODBC.

Antes de empezar

Para el sistema operativo Linux para System z, debe establecer el parámetro **EnableDescribeParam** con el valor `1` en el archivo de configuración `odbc.ini` del DSN cuando utilice el derivador ODBC para conectarse a orígenes de datos Sybase u Oracle con el protocolo DataDirect Sybase Wire Protocol, el protocolo Oracle Wire Protocol, controladores de cliente Oracle:

```
EnableDescribeParam=1
```

Si no establece el parámetro **EnableDescribeParam** en 1, se genera el mensaje de error 1822N por vincular variables de host de tipo non-char y varchar cuando el objeto ODBC SERVER se establece en modo PUSHDOWN.

Para obtener más información sobre la conexión a orígenes de datos con un controlador DataDirect, consulte el tema sobre la <http://media.datadirect.com/download/docs/odbc/allodbc/userguide/ase6.html>.

Procedimiento

Para preparar el servidor federado para acceder a orígenes de datos a través de ODBC:

1. Verifique que el archivo `odbc.ini` se ha actualizado en el servidor federado. Si el archivo no existe, puede crearlo en un editor de texto. Consulte la documentación del proveedor del cliente ODBC para obtener información sobre el archivo `odbc.ini`.
2. Configure el cliente ODBC.
Consulte la documentación del proveedor del cliente ODBC para obtener instrucciones sobre cómo configurar el cliente ODBC.
3. Si el cliente es DataDirect ODBC o RedBrick, verifique que se han creado los enlaces simbólicos adecuados. En los enlaces simbólicos siguientes, `DIR_CLIENTE_ODBC` es el directorio en el que se ha instalado el cliente ODBC.

Sistema operativo de servidor federado	Paso
Linux	<p>Cree los siguientes enlaces simbólicos:</p> <pre>ln -s \$DIR_CLIENTE_ODBC/./locale /usr/local/locale ln -s \$DIR_CLIENTE_ODBC/libodbcinst.so /usr/lib/libodbcinst.so</pre> <p>Si utiliza el controlador DataDirect Technologies Connect for ODBC 4.2, también debe crear el siguiente enlace simbólico:</p> <pre>ln -s \$DIR_CLIENTE_ODBC/libivicu19.so /usr/lib/libivicu19.so</pre>
Solaris	<p>Cree el siguiente enlace simbólico:</p> <pre>ln -s \$DIR_CLIENTE_ODBC/./locale \$HOME/sql11b/locale</pre> <p><code>\$HOME</code> es el directorio padre del propietario de la instancia de DB2.</p>

4. Si el cliente es DataDirect ODBC, puede probar la conexión del servidor federado al origen de datos utilizando la herramienta `demoodbc` de DataDirect Connect ODBC.
 - a. Ejecute el script `/opt/odbc/odbc.sh`. Este script configura varias variables de entorno específicas del sistema operativo.
 - b. Pruebe la conexión con el origen de datos ODBC mediante la herramienta `demoodbc` de DataDirect Connect ODBC. La herramienta `demoodbc` se encuentra en el subdirectorio `/demo` de las bibliotecas de Connect ODBC.

Una vez que haya completado esta tarea, podrá registrar el derivador.

Registro del derivador de ODBC

Debe registrar un derivador para acceder a los orígenes de datos ODBC. Los servidores federados utilizan los derivadores para comunicarse con los orígenes de datos y para recuperar datos de éstos. Los derivadores se implementan como un conjunto de archivos de biblioteca.

Procedimiento

Para registrar el derivador de ODBC:

Utilice uno de los métodos siguientes:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Emita la sentencia CREATE WRAPPER y especifique el nombre predeterminado para el derivador de ODBC.	<p>Por ejemplo:</p> <pre>CREATE WRAPPER ODBC;</pre> <p>Recuerde: Al registrar el derivador mediante el nombre predeterminado, ODBC, el servidor federado utiliza automáticamente la biblioteca de derivador de ODBC adecuada al sistema operativo en el que se ejecuta el servidor federado.</p> <p>Debe especificar la opción de derivador MODULE en servidores federados que ejecuten Linux o UNIX. La opción de derivador MODULE especifica la vía de acceso completa de la biblioteca que contiene el gestor de controladores ODBC.</p> <p>En AIX y Solaris, si está configurando el acceso a orígenes de datos IBM InfoSphere Classic Federation Server for z/OS, deben especificarse las opciones DB2_FENCED y DB2_SOURCE_CLIENT_MODE, como se muestra en el ejemplo siguiente.</p> <pre>CREATE WRAPPER ODBC LIBRARY 'libdb2rcodbc.a' OPTIONS (DB2_FENCED 'Y', DB2_SOURCE_CLIENT_MODE '32BIT', MODULE '/opt/IBM/DB2IIClassic82/cli/lib/cacsqlcli.so')</pre>
Emita la sentencia CREATE WRAPPER y especifique un nombre alternativo para el derivador de ODBC.	<p>Si el nombre del derivador predeterminado entra en conflicto con un nombre de derivador existente en la base de datos federada, puede sustituir el nombre del derivador predeterminado por otro de su elección. Si utiliza un nombre que no es el predeterminado, deberá incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER.</p> <p>Por ejemplo, para registrar un derivador con el nombre derivador_odbc en un servidor federado que utiliza AIX, emita esta sentencia:</p> <pre>CREATE WRAPPER derivador_odbc LIBRARY 'libdb2rcodbc.a';</pre> <p>El archivo de biblioteca del derivador especificado dependerá del sistema operativo del servidor federado.</p>

Una vez finalizada esta tarea, puede registrar las definiciones de servidor.

Archivos de biblioteca del derivador de ODBC

Los archivos de biblioteca de derivador de ODBC se añaden al servidor federado cuando se instala el derivador.

Cuando instala el derivador de ODBC, se añaden tres archivos de biblioteca a la vía de acceso del directorio predeterminado. Por ejemplo, si el servidor federado se ejecuta en AIX, los archivos de biblioteca del derivador que se añaden a la vía de acceso del directorio son `libdb2rcodbc.a`, `libdb2rcodbcF.a` y `libdb2rcodbcU.a`. El archivo de biblioteca de derivador predeterminado es `libdb2rcodbc.a`. Los demás archivos de biblioteca de derivador los utiliza internamente el derivador de ODBC.

Si no utiliza el nombre de derivador por omisión cuando registra un derivador, debe incluir el parámetro `LIBRARY` en la sentencia `CREATE WRAPPER` y especificar el nombre de archivo de biblioteca del derivador por omisión.

En la tabla siguiente se indican las vías de acceso al directorio predeterminado y los nombres de archivo de biblioteca del derivador predeterminado.

Tabla 24. Ubicaciones de biblioteca y nombres de archivo del cliente ODBC

Sistema operativo	Vía de acceso al directorio	Archivo de biblioteca de derivador
AIX	<code>/usr/opt/vía_acceso_instalación/lib32/</code> <code>/usr/opt/vía_acceso_instalación/lib64/</code>	<code>libdb2rcodbc.a</code>
Linux	<code>/opt/IBM/db2/vía_acceso_instalación/lib32</code> <code>/opt/IBM/db2/vía_acceso_instalación/lib64</code>	<code>libdb2rcodbc.so</code>
Solaris	<code>/opt/IBM/db2/vía_acceso_instalación/lib32</code> <code>/opt/IBM/db2/vía_acceso_instalación/lib64</code>	<code>libdb2rcodbc.so</code>
Windows	<code>%DB2PATH%\bin</code>	<code>db2rcodbc.dll</code>

vía_acceso_instalación es la vía de acceso al directorio en el que está instalado Federation en UNIX o Linux.

Sentencia CREATE WRAPPER - Ejemplos para el derivador de ODBC

Utilice la sentencia `CREATE WRAPPER` para registrar el derivador de ODBC. Este tema ofrece ejemplos para Linux, UNIX y Windows.

En los ejemplos siguientes, *derivador_odbc* es el nombre que se asigna al derivador que se registra en la base de datos federada.

Servidor federado Linux y Solaris

El ejemplo siguiente muestra cómo registrar un derivador con el nombre predeterminado en un servidor federado que ejecuta Linux o Solaris:

```
CREATE WRAPPER odbc OPTIONS (MODULE '/opt/lib/odbc.so');
```

Debe especificar la opción de derivador `MODULE` en servidores federados que ejecuten Linux o UNIX. La opción de derivador `MODULE` especifica la vía de acceso completa de la biblioteca que contiene el gestor de controladores ODBC.

El ejemplo siguiente muestra cómo registrar un derivador con un nombre alternativo en un servidor federado que ejecuta Linux o Solaris:

```
CREATE WRAPPER derivador_odbc LIBRARY 'libdb2rcodbc.so'
  OPTIONS (MODULE '/opt/lib/odbc.so');
```

Servidor federado AIX

El ejemplo siguiente muestra cómo registrar un derivador con el nombre predeterminado en un servidor federado que ejecuta AIX:

```
CREATE WRAPPER odbc
  OPTIONS (MODULE '/usr/lib/odbc.a');
```

Debe especificar la opción de derivador MODULE en servidores federados que ejecuten UNIX. La opción de derivador MODULE especifica la vía de acceso completa de la biblioteca que contiene el gestor de controladores ODBC.

El ejemplo siguiente muestra cómo registrar un derivador con un nombre alternativo en un servidor federado que ejecuta AIX:

```
CREATE WRAPPER derivador_odbc LIBRARY 'libdb2rcodbc.a'
  OPTIONS (MODULE '/usr/lib/odbc.a');
```

IBM InfoSphere Classic Federation Server for z/OS (AIX)

El ejemplo siguiente muestra cómo registrar un derivador de ODBC emitiendo la sentencia CREATE WRAPPER en un sistema operativo AIX. En AIX y Solaris, deben especificarse las opciones DB2_FENCED y DB2_SOURCE_CLIENT_MODE, como se muestra en el ejemplo siguiente.

```
CREATE WRAPPER odbc
  OPTIONS (DB2_FENCED 'Y', DB2_SOURCE_CLIENT_MODE '32BIT',
  MODULE '/opt/IBM/DB2IIClassic82/cli/lib/cacsq1cli.so');
```

Debe especificar la opción de derivador MODULE en servidores federados que ejecuten UNIX. La opción de derivador MODULE especifica la vía de acceso completa de la biblioteca que contiene el gestor de controladores ODBC.

El ejemplo siguiente muestra cómo registrar un derivador para acceder a orígenes de datos IBM InfoSphere Classic Federation Server for z/OS con un nombre alternativo:

```
CREATE WRAPPER derivador_odbc LIBRARY 'libdb2rcodbc.a'
  OPTIONS (DB2_FENCED 'Y', DB2_SOURCE_CLIENT_MODE '32BIT',
  MODULE '/opt/IBM/DB2IIClassic82/cli/lib/cacsq1cli.so');
```

Servidor federado Windows

El ejemplo siguiente muestra cómo registrar un derivador con el nombre predeterminado en un servidor federado que ejecuta Windows:

```
CREATE WRAPPER odbc;
```

El ejemplo siguiente muestra cómo registrar un derivador con un nombre alternativo en un servidor federado que ejecuta Windows:

```
CREATE WRAPPER derivador_odbc LIBRARY 'db2rcodbc.dll';
```

Registro de las definiciones del servidor para un origen de datos ODBC

Debe registrar cada servidor ODBC al que desee acceder en la base de datos federada.

Procedimiento

Para registrar una definición de servidor para un origen de datos ODBC:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Emita la sentencia CREATE SERVER.	<p>1. Emita la sentencia CREATE SERVER y especifique las opciones de servidor obligatorias:</p> <pre>CREATE SERVER nombre_definición_servidor TYPE tipo_origen_datos VERSION número_versión WRAPPER nombre_derivador OPTIONS (NODE 'nombre_nodo', CODEPAGE 'nombre_página-códigos', DBNAME 'nombre_base-datos');</pre> <p>Aunque NODE, CODEPAGE y DBNAME son opciones de servidor en la sentencia CREATE SERVER, son obligatorias para orígenes de datos ODBC:</p> <ul style="list-style-type: none"> • La opción de servidor NODE debe establecerse en el origen de datos especificado en la palabra clave DATASOURCE del archivo cac.ini. <p>Ejemplo</p> <p>Si el archivo cac.ini especifica DATASOURCE = CACSAMP tcp/150.45.37.49/5000, la opción NODE debe establecerse en CACSAMP.</p> <ul style="list-style-type: none"> • La opción de servidor CODEPAGE debe establecerse en el número de página de códigos de la página de códigos de cliente especificada en el archivo cac.ini. <p>Ejemplo</p> <p>Si el archivo cac.ini especifica CLIENT CODEPAGE = IBM-850, la opción CODEPAGE debe establecerse en 850.</p> <ul style="list-style-type: none"> • La opción de servidor DBNAME debe establecerse en el nombre de la base de datos de origen de datos a la que se desea acceder. <p>Ejemplo</p> <p>Si el nombre del origen de datos ODBC es venice, la opción DBNAME debe establecerse en <i>venice</i>.</p> <p>2. Una vez creada la definición del servidor, utilice la sentencia ALTER SERVER para añadir o eliminar opciones de servidor.</p>

Una vez finalizada esta tarea, puede crear una correlación de usuario.

Sentencia CREATE SERVER - Ejemplos del derivador de ODBC

Utilice la sentencia CREATE SERVER para registrar definiciones de servidor para el derivador de ODBC. Este tema ofrece un ejemplo completo con los parámetros obligatorios y un ejemplo con opciones adicionales del servidor.

El ejemplo siguiente muestra cómo registrar una definición de servidor para un origen de datos MySQL emitiendo la sentencia CREATE SERVER:

```
CREATE SERVER servidor_mysql TYPE mysql
VERSION 4.0 WRAPPER nombre_derivador
OPTIONS (NODE 'nodo_odbc', DBNAME 'venice')
```

servidor_mysql

Nombre asignado por el usuario al servidor de orígenes de datos ODBC. No se permiten los nombres de definición de servidor duplicados.

TYPE *mysql*

Especifica el tipo de servidor de orígenes de datos para el que está configurando el acceso. Este parámetro es opcional.

VERSION *4.0*

Versión del origen de datos ODBC al que se desea acceder. Este parámetro es opcional.

WRAPPER *nombre_derivador*

El nombre del derivador que ha especificado en la sentencia CREATE WRAPPER.

NODE '*nodo_odbc*'

El nombre del nodo (el nombre DSN de sistema) asignado al origen de datos ODBC al definir el DSN. En servidores federados que ejecutan Windows, este valor debe ser el nombre de un DSN de sistema en la ventana Administrador de orígenes de datos ODBC. En servidores federados que ejecutan UNIX, el nombre del nodo es el DSN definido en el archivo de configuración de ODBC. Generalmente, el archivo de configuración de ODBC se denomina *odbc.ini*.

Aunque el nombre del nodo se especifica como opción en la sentencia CREATE SERVER, es necesario para los orígenes de datos ODBC.

DBNAME '*venice*'

Opcional. Nombre del origen de datos ODBC al que se desea acceder.

Opciones de servidor

Al crear la definición de servidor, puede especificar opciones de servidor adicionales en la sentencia CREATE SERVER. Las opciones de servidor pueden ser genéricas y específicas de ODBC.

Algunos orígenes de datos ODBC (por ejemplo, MySQL) no pueden procesar comillas simples en los nombres de tablas y columnas de sentencias SQL. Para acceder a estos orígenes de datos, debe incluir las siguientes opciones de servidor en la sentencia CREATE SERVER:

- `DB2_TABLE_QUOTE_CHAR ' ` '`
- `DB2_ID_QUOTE_CHAR ' ` '`
- `DB2_AUTHID_QUOTE_CHAR ' ` '`

El carácter ` es el delimitador de los identificadores como, por ejemplo, nombres de esquema, de tabla y de columna.

Por ejemplo:

```
CREATE SERVER servidor_mysql TYPE mysql
  VERSION 4.0 WRAPPER nombre_derivador
  OPTIONS (NODE 'nodo_mysql', DB2_TABLE_QUOTE_CHAR ' ` ',
  DB2_ID_QUOTE_CHAR ' ` ', DB2_AUTHID_QUOTE_CHAR ' ` ')
```

Creación de una correlación de usuario para un origen de datos ODBC

Cuando se intenta acceder a un servidor ODBC, el servidor federado establece una conexión con el servidor ODBC utilizando un ID de usuario y una contraseña

válidos para ese origen de datos. Para los orígenes de datos que requieren una correlación de usuario, debe definir una asociación (una correlación de usuario) entre cada ID de usuario y contraseña del servidor federado y el ID de usuario y contraseña correspondientes del origen de datos.

Acerca de esta tarea

Cree una correlación de usuario para cada ID de usuario que vaya a acceder al sistema federado para enviar solicitudes distribuidas al origen de datos ODBC.

Procedimiento

Para correlacionar un ID de usuario local con el ID de usuario y la contraseña del origen de datos ODBC:

Emita una sentencia CREATE USER MAPPING.

Por ejemplo:

```
CREATE USER MAPPING FOR IDusuario_local SERVER nombre_definición_servidor
    OPTIONS (REMOTE_AUTHID 'IDusuario_remoto', REMOTE_PASSWORD 'contraseña_remota');
```

Aunque REMOTE_AUTHID y REMOTE_PASSWORD son opciones de correlación de usuario en la sentencia CREATE USER MAPPING, estas opciones son necesarias para acceder a orígenes de datos ODBC:

La correlación de usuario debe correlacionar el ID de autorización de DB2 con el ID de usuario y la contraseña especificados en el archivo cac.ini.

Ejemplo

Si el archivo cac.ini especifica USERID = *MY_USERID* y USERPASSWORD = *MY_PASSWORD*, las opciones de la sentencia CREATE USER MAPPING deben especificarse del siguiente modo:

```
REMOTE_AUTHID = MY_USERID
REMOTE_PASSWORD = MY_PASSWORD
```

Una vez finalizada esta tarea, puede probar la conexión al origen de datos ODBC.

Sentencia CREATE USER MAPPING - Ejemplos para el derivador de ODBC

Utilice la sentencia CREATE USER MAPPING para correlacionar un ID de usuario de servidor federado con un ID de usuario y una contraseña de origen de datos ODBC. Este tema ofrece un ejemplo completo con los parámetros obligatorios y un ejemplo que muestra cómo utilizar el registro especial USER de DB2 con la sentencia CREATE USER MAPPING.

El ejemplo siguiente muestra cómo correlacionar un ID de autorización federado con un ID de usuario y una contraseña de origen de datos ODBC:

```
CREATE USER MAPPING FOR arturo SERVER servidor_mysql
    OPTIONS (REMOTE_AUTHID 'art', REMOTE_PASSWORD 'red4blue')
```

arturo Especifica el ID de autorización local que se correlaciona con el ID de usuario y la contraseña remotos definidos en el origen de datos ODBC.

servidor_mysql

Especifica el nombre de la definición del servidor que ha definido en la sentencia CREATE SERVER para el origen de datos ODBC.

'*art*'

Especifica el ID de usuario remoto con el que se correlaciona *arturo*. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_ID en 'U' o 'L' en la sentencia CREATE SERVER.

'red4blue'

Especifica la contraseña remota asociada a *'art'*. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_PW en 'U' o 'L' en la sentencia CREATE SERVER.

Registro especial USER de DB2

Puede utilizar el registro especial USER de DB2 para correlacionar el ID de autorización de la persona que emite la sentencia CREATE USER MAPPING con el ID de autorización del origen de datos especificado en la opción de usuario REMOTE_AUTHID.

A continuación se ofrece un ejemplo de la sentencia CREATE USER MAPPING, que incluye el registro especial USER:

```
CREATE USER MAPPING FOR USER SERVER servidor_mysql
  OPTIONS (REMOTE_AUTHID 'art', REMOTE_PASSWORD 'red4blue');
```

Prueba de la conexión al servidor de orígenes de datos ODBC

Pruebe la conexión con el servidor de orígenes de datos ODBC para determinar si el servidor federado se ha configurado correctamente para acceder a orígenes de datos ODBC.

Acerca de esta tarea

Puede probar la conexión con el servidor de orígenes de datos ODBC mediante la definición de servidor y las correlaciones de usuario que ha definido.

Procedimiento

Para probar la conexión al servidor de orígenes de datos ODBC:

Abra una sesión de paso a través y emita una sentencia SELECT en las tablas del sistema de orígenes de datos ODBC. Si la sentencia SELECT devuelve una cuenta, significa que la definición del servidor y las correlaciones de usuario están correctamente configuradas.

```
SET PASSTHRU nombre_definición_servidor
SELECT count(*) FROM nombre.esquema.nombre.tabla
SET PASSTHRU RESET
```

Si la sentencia SELECT devuelve un error, debe resolver los errores de conexión.

Una vez finalizada esta tarea, podrá registrar apodos para las tablas y vistas de orígenes de datos ODBC.

Resolución de errores de conexión de origen de datos

Una conexión de prueba con el servidor de origen de datos puede devolver un error por varios motivos. Existen acciones que permiten determinar el motivo del error.

Síntoma

Se devuelve un error al intentar establecer conexión con el origen de datos.

Causa

Hay varias causas posibles que explican un problema de conexión.

Resolución del problema

Para solucionar los problemas de errores de conexión con el origen de datos, consulte los siguientes elementos:

- Verifique que el origen de datos está disponible.
- Si procede, asegúrese de que el servidor de origen de datos esté configurado para conexiones entrantes.
- Asegúrese de que los valores de correlación de usuarios para las opciones REMOTE_AUTHID y REMOTE_PASSWORD son válidos para las conexiones con el origen de datos. Modifique la correlación de usuarios o cree otra correlación de usuarios según sea necesario.
- Si procede, asegúrese de que el software de cliente del servidor federado esté instalado y configurado correctamente para conectarse al origen de datos.
- Para orígenes de datos ODBC, asegúrese de que el controlador ODBC del servidor federado esté instalado y configurado correctamente para establecer conexión con el servidor de origen de datos ODBC. En servidores federados que ejecutan Windows, utilice la herramienta ODBC Data Source Administrator para verificar el controlador. En servidores federados que ejecutan UNIX, consulte la documentación del proveedor del cliente ODBC.
- Verifique que los valores para las variables establecidas en el servidor federado son correctas para el origen de datos. Estas variables incluyen las variables de entorno del sistema, las variables de la archivo db2dj.ini y las variables del registro de perfiles de DB2 (db2set).
- Compruebe la definición del servidor. Si es necesario, elimine la definición del servidor y créela de nuevo.

Registro de apodos para vistas y tablas de orígenes de datos ODBC

Para cada definición de servidor ODBC que registre, debe registrar un apodo para cada tabla o vista a la que desee acceder. Utilice estos apodos en lugar de los nombres de los objetos de origen de datos cuando consulte los orígenes de datos ODBC.

Antes de empezar

Actualice las estadísticas del origen de datos ODBC antes de registrar un apodo. La base de datos federada se basa en las estadísticas del catálogo de orígenes de datos para optimizar el proceso de las consultas. Utilice el mandato de origen de datos equivalente al mandato de DB2 RUNSTATS para actualizar las estadísticas del origen de datos.

Procedimiento

Para registrar un apodo para una tabla o vista de origen de datos ODBC, utilice uno de los métodos siguientes:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .

Método	Descripción
Emita la sentencia CREATE NICKNAME. Los apodos pueden tener una longitud de hasta 128 caracteres.	Por ejemplo: <pre>CREATE NICKNAME apodo FOR nombre_definición_servidor."esquema_remoto". "tabla.remota" ;</pre>

Cuando se crea el apodo, el servidor federado consulta el catálogo de orígenes de datos utilizando el apodo. Esta consulta prueba la conexión a la tabla o vista del origen de datos. Si la conexión no funciona, recibirá un mensaje de error.

Repita este paso para cada tabla o vista ODBC para la que desee crear un apodo.

Sentencia CREATE NICKNAME - Ejemplos para el derivador de ODBC

Utilice la sentencia CREATE NICKNAME para registrar un apodo para una tabla o vista ODBC a la que desea acceder. Este tema ofrece un ejemplo completo con los parámetros obligatorios.

El ejemplo siguiente muestra cómo registrar un apodo para una tabla o vista de ODBC mediante la sentencia CREATE NICKNAME.

```
CREATE NICKNAME cust_europe FOR servidor_mysql."vinnie"."italy"
```

cust_europe

Apodo exclusivo utilizado para identificar la tabla o vista ODBC. El apodo debe ser exclusivo dentro del esquema.

Importante: El nombre de apodo consta de dos partes: el esquema y el apodo. Si omite el esquema al registrar el apodo, el esquema del apodo será el ID de autorización del usuario que registra el apodo.

Si el origen de datos ODBC no admite esquemas, omita el esquema en la sentencia CREATE NICKNAME.

```
servidor_mysql."vinnie"."italy"
```

Identificador de tres partes para el objeto remoto:

- *servidor_mysql* es el nombre de definición de servidor que ha asignado al servidor de orígenes de datos ODBC en la sentencia CREATE SERVER.
- *vinnie* es el ID de usuario del propietario al que pertenece la tabla o vista.
- *italy* es el nombre de la tabla o la vista remota a la que desea acceder.

El servidor federado convierte a mayúsculas los nombres de los esquemas y tablas ODBC, a menos que especifique los nombres entre comillas.

Optimización del rendimiento del derivador de ODBC con el programa de utilidad de ajuste de ODBC (db2fedsvrfg)

Puede optimizar el rendimiento del derivador de ODBC con el programa de utilidad de ajuste de ODBC, db2fedsvrfg. Este programa de utilidad ejecuta un conjunto de consultas predefinidas en los orígenes de datos y comprueba la exactitud de los resultados. El programa de utilidad crea un conjunto de sentencias ALTER SERVER que pueden ejecutarse en el servidor para establecer las opciones de servidor a fin de obtener un rendimiento óptimo.

Antes de empezar

Deben configurarse los elementos siguientes en el servidor federado:

- Federation
- El derivador de ODBC
- El software de cliente ODBC
- Variables para el entorno del sistema. Es posible que sean necesarias las variables de entorno ODBCINI y LIBPATH.

Para obtener información acerca de la instalación, configuración y requisitos de ODBC, consulte la documentación suministrada con cada uno de estos productos.

Procedimiento

Para optimizar el rendimiento del derivador de ODBC con el programa de utilidad de ajuste de ODBC:

1. Si el servidor ODBC no se ha definido, utilice el Centro de control de DB2 o el Procesador de línea de mandatos de DB2 para conectarse al servidor federado y crear un servidor y un derivador de ODBC.
2. Opcional: si las tablas de prueba ya existen en el origen de datos ODBC, conéctese al origen de datos ODBC y elimínelas.
3. Ejecute el programa de utilidad de ajuste de ODBC desde la línea de mandatos de DB2 con las opciones que desee utilizar. El programa de utilidad puede tardar algún tiempo en finalizar.
4. Compruebe que el programa de utilidad se ha ejecutado correctamente. Si el programa de utilidad se ha ejecutado correctamente, verá el mensaje siguiente en la ventana de mandatos o en el archivo de salida especificado:

```
ALTER SERVER "DS1" OPTIONS (ADD opción1, 'valor1')
ALTER SERVER "DS1" OPTIONS (ADD opción2, 'valor2')
ALTER SERVER "DS1" OPTIONS (ADD opción3, 'valor3')
.....
```

El mandato db2fedsvr cfg ha finalizado satisfactoriamente.

Si el mandato no se ha ejecutado correctamente, verá un mensaje de error que indicará la razón del error. Corrija el problema y vuelva a ejecutar el mandato.

Es necesario crear las tablas de prueba manualmente en las siguientes circunstancias:

- Si desea utilizar nombres de tabla diferentes del predeterminado
 - Si el origen de datos al que accede mediante ODBC es sólo de lectura
 - Si el programa de utilidad de ajuste de ODBC no puede crear las tablas de prueba necesarias para el programa de utilidad
5. Conéctese al servidor federado en el que se ha definido el origen de datos.
 6. Utilice el programa de utilidad db2look para guardar los valores de servidor existentes antes de ejecutar el archivo creado por el programa de utilidad de ajuste de ODBC. Consulte la documentación del programa de utilidad db2look para obtener información acerca de cómo guardar los valores de servidor existentes.
 7. Opcional: si el servidor ODBC está definido, puede conectarse al servidor federado y eliminar las opciones del servidor. El programa de utilidad crea sentencias ALTER SERVER en el formato descrito en el paso 4. Si estas opciones de servidor ya se han añadido, las sentencias ALTER SERVER fallarán.
 8. Utilice el mandato siguiente para ejecutar las sentencias ALTER SERVER generadas por el programa de utilidad en la base de datos federada. Las

sentencias ALTER SERVER creadas por el programa de utilidad de ajuste de ODBC se encuentran en el archivo db2fedsvrcfg.sql.

```
db2 -tvf db2fedsvrcfg.sql
```

9. Compruebe los resultados de las sentencias ALTER SERVER. Si alguna de las sentencias ha fallado, puede modificarla en el archivo db2fedsvrcfg.sql y ejecutarla de nuevo hasta que sea satisfactoria.
10. Una vez finalizado el ajuste del servidor con el programa de utilidad de ajuste de ODBC, establezca la opción de servidor PUSHDOWN en 'Y' para completar el proceso de optimización.

Sintaxis del mandato db2fedsvrcfg - programa de utilidad de ajuste de ODBC

Utilice el mandato db2fedsvrcfg para mejorar el rendimiento del derivador de ODBC.

Sintaxis

```
db2fedsvrcfg -s nombreServidor [-m bibliotecaGestorControladoresODBC] -dsn nombreDSNodbc [-dbname nombreBDod] [-u idUsuario] [-p contraseña] [-noprep] [-prefix prefijoNombreTabla] [-suffix sufijoNombreTabla] [-dscp páginaCódigos] [-v] [-o archivoSalida] [-h]
```

Parámetros

Utilice el mandato db2fedsvrcfg32 cuando utilice controladores ODBC de 32 bits en AIX o Solaris. De lo contrario, utilice el mandato db2fedsvrcfg.

-dbname nombreBDod

El nombre de base de datos del origen de datos.

-dscp páginaCódigos

El identificador de página de códigos del origen de datos. Si no se especifica esta opción, el programa de utilidad utiliza la página de códigos del entorno del usuario. Este parámetro es opcional.

-dsn nombreDSNodbc

El nombre de origen de datos (DSN) del sistema para el origen de datos.

-h Provoca la visualización de ayuda detallada. Este parámetro es opcional.

-m bibliotecaGestorControladoresODBC

Nombre de archivo completo de la biblioteca del gestor de controladores ODBC. El nombre de archivo de biblioteca del gestor de controladores ODBC es opcional para Windows.

-noprep

Impide la creación de las tablas de prueba en el origen de datos antes de las pruebas. Este parámetro es opcional.

-o archivoSalida

Nombre de archivo completo del archivo de salida del programa de utilidad de ajuste de ODBC. El archivo de salida contiene las sentencias ALTER SERVER utilizadas para ajustar el rendimiento del derivador de ODBC. Este parámetro es opcional. Si no se especifica este parámetro, la salida se visualiza en la ventana de mandatos.

-p contraseña

La contraseña de conexión al origen de datos. Este parámetro es opcional.

-prefix prefijoNombreTabla

El prefijo de los nombres de tablas de origen de datos ODBC que el programa de utilidad utiliza para el análisis. Si no se especifica el prefijo, se utiliza el prefijo predeterminado, IITEST. Este parámetro es opcional.

-s nombreServidor

El nombre del servidor federado.

-suffix sufijoNombreTabla

El sufijo de los nombres de tablas de origen de datos ODBC que el programa de utilidad utiliza para el análisis. Si no se especifica un sufijo, se utiliza una serie vacía.

-u idUsuario

El nombre de usuario de conexión al origen de datos. Este parámetro es opcional.

-v Especifica que la salida del programa de utilidad es verbosa. Este parámetro es opcional.

Ejemplo

El ejemplo siguiente muestra el mandato que se ejecuta en datastore, el origen de datos ODBC. En este ejemplo, las tablas de prueba se denominan ABC n XYZ, siendo n un número del 1 al 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"
```

Definiciones de tablas de prueba para el programa de utilidad de ajuste de ODBC (db2fedsvr cfg)

En algunos casos, es necesario crear manualmente las tablas de prueba para el programa de utilidad de ajuste de ODBC. En este tema se describen las definiciones de tablas de prueba.

Es necesario crear las tablas de prueba para el programa de utilidad de ajuste de ODBC en las siguientes circunstancias:

- Si desea utilizar nombres de tabla diferentes del predeterminado
- Si el origen de datos al que accede mediante ODBC es sólo de lectura
- Si el programa de utilidad de ajuste de ODBC no puede crear las tablas de prueba necesarias para el programa de utilidad

La definición de tabla que sigue es aplicable a la totalidad de las siete tablas de prueba necesarias (IITEST1 a IITEST7). El prefijo del nombre de tabla predeterminado es IITEST, y el sufijo predeterminado es una serie vacía. Si especifica un prefijo y un sufijo diferentes, debe especificar las opciones *-prefix* y *-suffix* al ejecutar el programa de utilidad de ajuste de ODBC.

Tabla 25. Definición de tabla de prueba para el programa de utilidad de ajuste de ODBC para la tabla IITEST1

Nombre de columna	Tipo de datos SQL	Identificador de tipo de datos SQL	Longitud
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

Tabla 25. Definición de tabla de prueba para el programa de utilidad de ajuste de ODBC para la tabla IITEST1 (continuación)

Nombre de columna	Tipo de datos SQL	Identificador de tipo de datos SQL	Longitud
IT1C6	varchar(10)	SQL_VARCHAR	10
IT1C7	char(100)	SQL_CHAR	100

Tabla 26. Definición de tabla de prueba para el programa de utilidad de ajuste de ODBC para la tabla IITEST2

Nombre de columna	Tipo de datos SQL	Identificador de tipo de datos SQL	Longitud
IT2C1	integer	SQL_INTEGER	
IT2C2	integer	SQL_INTEGER	
IT2C3	char(30)	SQL_CHAR	30

Tabla 27. Definición de tabla de prueba para el programa de utilidad de ajuste de ODBC para las tablas IITEST3 a IITEST7

Número de columnas de cada tabla	Nombre de columna	Tipo de datos SQL	Identificador de tipo de datos SQL	Longitud
66	ITxCn	char(100)	SQL_CHAR	100

x Número correspondiente a la tabla que se define.

n Número de columna.

Por ejemplo, IT3C1 es el nombre de columna de la primera columna de la tabla IITEST3.

Acceso a datos Excel mediante el derivador de ODBC

Puede acceder a libros de Microsoft Excel con el derivador de ODBC mediante el controlador ODBC de Excel.

Antes de empezar

- El controlador ODBC de Excel debe encontrarse en el servidor federado.
- El servidor federado debe poder abrir y leer las hojas del libro Excel para recuperar los datos. Por tanto, los libros Excel debe encontrarse en el mismo sistema que el servidor federado o en una unidad de red correlacionada accesible.
- El formato de las columnas debe ajustarse al tipo de datos que se espera que tenga la columna.
- Los datos insertados en las columnas deben ajustarse al tipo de formato especificado para la columna.
- Si las ocho primeras filas de la hoja de cálculo no contienen datos, asegúrese de que estén vacías. Para asegurarse de que una celda está vacía, abra la hoja de cálculo en Microsoft Excel y seleccione **Edición** → **Borrar todo**.
- Asegúrese de que los datos insertados en las columnas de la hoja de cálculo se ajustan al tipo especificado.

Restricciones

- El derivador de ODBC no puede acceder a una hoja si un usuario o aplicación ya han abierto el libro en modalidad de lectura/escritura. Sin embargo, si el

derivador de ODBC abre el libro antes de que lo haga un usuario o aplicación, éstos pueden abrir el libro en modalidad de sólo lectura.

- El controlador ODBC de Excel espera que la primera fila que no esté en blanco contenga las etiquetas de las columnas de la hoja. Debe insertar una fila de etiquetas de columna en la hoja si ésta no las contiene.
- Dado que el controlador ODBC de Excel sólo está disponible para sistemas operativos Windows, sólo puede utilizar el derivador de ODBC para acceder a datos Excel en servidores federados que ejecuten Windows.
- Puede realizar operaciones de inserción y actualización en hojas Excel, pero no operaciones de supresión. El controlador ODBC de Excel no da soporte a operaciones de supresión. Para suprimir datos de una hoja, debe abrirla en Excel para realizar los cambios.

Acerca de esta tarea

No es necesario que la aplicación Excel esté instalada en el servidor federado. El controlador ODBC de Excel se instala automáticamente con Windows.

Con el derivador de ODBC y el controlador ODBC de Excel, puede acceder a datos de cualquiera de las hojas de un libro. El controlador ODBC de Excel interpreta un libro como una base de datos y cada una de las hojas del libro como una tabla.

El controlador ODBC de Excel da soporte a versiones anteriores de libros Excel aunque la versión de la aplicación Excel que los ha generado ya no esté soportada. Por ejemplo, Microsoft ya no da soporte a hojas creadas en Excel Versión 4.0, pero el controlador da soporte a hojas Excel creadas en esa versión.

Procedimiento

Para acceder a hojas Excel con el derivador de ODBC:

1. Asegúrese de que el libro Excel al que desea acceder se encuentre en el servidor federado o en una unidad de red correlacionada accesible.
2. Si los datos Excel se comparten a través de una red Windows de transición que utiliza grupos de trabajo, debe configurar los permisos de acceso en el origen de datos Excel.
3. Si es necesario, cambie el diseño de los datos en las hojas Excel para que se ajusten a los requisitos del controlador ODBC de Excel. Repita este paso para cada hoja o rango definido a los que desee acceder.
4. Si es necesario, cree los rangos definidos a los que desee acceder.
5. Cree un DSN de sistema para el libro al que desee acceder. Puede utilizar el Administrador de orígenes de datos ODBC para configurar el DSN de sistema. El nombre especificado al crear el DSN de sistema se asignará como valor de la opción `NODE` en la sentencia `CREATE SERVER`.

Si el origen de datos Excel se comparte a través de una red Windows que utiliza grupos de trabajo, debe especificar el nombre de base de datos del DSN de sistema con la sintaxis siguiente:

```
\\nombre_Sistema\subdirectorio_nombre-archivo
```

donde *nombre_Sistema* es el nombre del sistema del origen de datos Excel y *subdirectorio_nombre-archivo* es el subdirectorio y el nombre de archivo del archivo Excel.

Ejemplo

Si el nombre del sistema del origen de datos Excel es XLSQLS y el

directorio de red del archivo Excel es E:\share\test.xls, especificará el siguiente nombre de base de datos DSN:

```
\\XLSQLS\share\test.xls
```

donde el directorio raíz del directorio de red E: se sustituye por \\ y el nombre de sistema XLSQLS.

6. Emita la sentencia CREATE WRAPPER.
7. Especifique la ubicación del libro registrando un objeto de servidor en el catálogo del sistema de bases de datos federadas. Para el derivador de ODBC, necesita un objeto de servidor para cada DSN. El DSN se asocia con el libro cuando se utiliza el controlador ODBC de Excel. La opción NODE *compounds_workbook_dsn* corresponde al DSN de sistema que ha creado. La opción NODE es necesaria para que el derivador de ODBC acceda a las hojas Excel.

Para especificar la ubicación del libro, emita la sentencia CREATE SERVER y utilice el DSN como DSN de sistema para la opción NODE.

Por ejemplo:

```
CREATE SERVER compounds_workbook WRAPPER odbc  
  OPTIONS (NODE 'compounds_workbook_dsn', PASSWORD 'n')
```

Repita este paso para cada libro al que desea acceder.

8. Emita la sentencia CREATE NICKNAME para crear un apodo para la hoja a la que desea acceder. La sintaxis es la siguiente:

```
CREATE NICKNAME apodo FOR nombre_servidor.tabla_remota
```

9. Si ha creado un rango definido para acceder a los datos, especifique el nombre del rango en la parte *tabla_remota* de la sentencia CREATE NICKNAME.

Por ejemplo, si el nombre del rango es *testing*, la sentencia CREATE NICKNAME será:

```
CREATE NICKNAME compounds_nickname FOR compounds_workbook.testing
```

Para acceder a los datos de toda la hoja en lugar de un rango, especifique el nombre de la hoja seguido del símbolo \$.

Por ejemplo, si el nombre de la hoja es *Sheet1*, la sentencia CREATE NICKNAME será:

```
CREATE NICKNAME compounds_nick FOR compounds_workbook.Sheet1$
```

Configuración de los permisos de acceso para datos Excel en un grupo de trabajo al utilizar el derivador de ODBC

Puede configurar los permisos de acceso para acceder a datos Excel remotos compartidos por medio de una red Windows que utiliza grupos de trabajo.

Acerca de esta tarea

Todos los ejemplos de esta tarea están destinados a Windows XP Professional; consulte la documentación del sistema operativo en el que se encuentre el origen de datos para obtener información específica acerca de cómo establecer los permisos de sus cuentas de usuario.

Procedimiento

Para configurar los permisos de acceso al origen de datos Excel:

1. Establezca el modelo de compartimiento y seguridad en el panel **Acceso de red: Modelo de seguridad y para compartir para cuentas locales**. Puede seleccionar uno de los siguientes modelos de compartimiento y seguridad:

Sólo invitado: usuarios locales autenticados como invitados

Se conectará al origen de datos sólo a través de la cuenta de usuario de invitado. Recibirá el nivel de acceso que configure para la cuenta de usuario invitado y debe utilizar su contraseña de usuario de DB2.

Clásico: usuarios locales autenticados como ellos mismos

Puede utilizar su ID de usuario y contraseña de DB2 para conectarse al origen de datos o utilizar la cuenta de usuario invitado. Para el ID de usuario de DB2, el usuario recibe el nivel de acceso al origen de datos de acuerdo con el nivel de acceso del ID de usuario de DB2. Para la cuenta de usuario invitado, recibirá el nivel de acceso que configure para la cuenta de usuario invitado y debe utilizar su contraseña de usuario de DB2.

Ejemplo

Para seleccionar el modelo de compartimiento y seguridad **Clásico: usuarios locales autenticados como ellos mismos**, abra las **Herramientas administrativas** y desplácese a **Directiva de seguridad local** → **Directivas locales** → **Opciones de seguridad** → **Acceso de red: Modelo de seguridad y para compartir para cuentas locales**.

2. Cree la cuenta de usuario en el panel **Acceso a este equipo desde propiedades de red**:

Opción	Descripción
Para el modelo de seguridad Sólo invitado: usuarios locales autenticados como invitados	Especifique el grupo de trabajo de Windows y cree la cuenta de usuario invitado. Debe especificar la contraseña utilizando su contraseña de DB2.
Para el modelo de seguridad Clásico: usuarios locales autenticados como ellos mismo	<ul style="list-style-type: none">• Especifique el grupo de trabajo de Windows y cree la cuenta de usuario utilizando el ID de usuario y la contraseña de DB2.• Si desea utilizar la cuenta de usuario invitado, especifique el grupo de trabajo de Windows y cree la cuenta de usuario invitado. Debe especificar la contraseña de la cuenta de usuario invitado utilizando su contraseña de DB2.

Ejemplo

Para crear la cuenta de usuario invitado:

- a. Abra las **Herramientas administrativas** y desplácese a **Directiva de seguridad local** → **Directivas locales** → **Asignación de derechos de usuario** → **Tener acceso a este equipo desde la red**
 - b. Especifique el grupo de trabajo de Windows en el campo **Desde esta ubicación**.
 - c. Especifique el ID de usuario de DB2 en el campo **Escriba los nombres de objeto que desea seleccionar**.
 - d. Pulse **Comprobar nombres**.
 - e. En el panel **Escribir contraseña de red**, especifique el ID de usuario y la contraseña de DB2.
3. Establezca los permisos de compartimiento en el panel **Propiedades de compartir** estableciendo los permisos **Cambiar** y **Leer** en **Permitir**.

Ejemplo

Para establecer los permisos de compartimiento del directorio de red E:\share\test.xls:

- Pulse la carpeta del directorio E:\share con el botón derecho del ratón y pulse **Compartir y seguridad**.
 - En la pestaña **Compartir**, seleccione **Compartir esta carpeta** y pulse **Permisos**.
 - Seleccione el usuario y luego seleccione **Permitir** para los permisos **Cambiar y Leer**.
4. Establezca los permisos de NTFS en el panel **Propiedades** estableciendo los permisos **Leer & Ejecutar** y **Leer** en **Permitir**.

Ejemplo

para establecer los permisos NTFS del archivo E:\share\test.xls:

- Pulse el archivo test.xls con el botón derecho del ratón y pulse **Propiedades**.
- En la pestaña **Seguridad**, seleccione el usuario y luego seleccione **Permitir** para los permisos **Leer & Ejecutar** y **Leer**.

Complete el procedimiento para acceder a datos Excel mediante el derivador de ODBC.

Configuración del acceso de ODBC a orígenes de datos IBM InfoSphere Classic Federation Server for z/OS

El derivador de ODBC se ha optimizado para acceder a orígenes de datos IBM InfoSphere Classic Federation Server for z/OS. El configurar de ODBC detecta el controlador ODBC y configura automáticamente las opciones de rendimiento.

Antes de empezar

Deben configurarse los elementos siguientes en el servidor federado:

- Servidor federado
- Cliente IBM InfoSphere Classic Federation Server for z/OS
- Base de datos federada
- Variables del archivo db2dj.ini de entorno del sistema del registro de perfiles de DB2(db2set):
 - Para sistemas Linux y UNIX, la variable CAC_CONFIG debe especificar el directorio del archivo cac.ini, por ejemplo:
`CAC_CONFIG=/home/db2inst1/cac.ini`
 - Puede que sea necesaria la variable de entorno DB2LIBPATH, dependiendo de los directorios de instalación de la biblioteca del gestor de controladores ODBC y de la biblioteca del controlador ODBC.
- Para sistemas AIX y Solaris, la opción de derivador DB2_FENCED debe establecerse en Y y la opción de derivador DB2_SOURCE_CLIENT_MODE debe establecerse en 32BIT.

Para obtener información acerca de la instalación, configuración y requisitos de ODBC, consulte la documentación suministrada con cada uno de estos productos.

Restricciones

El derivador de ODBC no da soporte a la sentencia siguiente con orígenes de datos IBM InfoSphere Classic Federation Server for z/OS:

- CREATE TABLE

Los tipos de datos siguientes no están soportados en orígenes de datos IBM InfoSphere Classic Federation Server for z/OS:

- BLOB
- CLOB
- DBCLOB
- CHAR FOR BIT DATA
- VARCHAR FOR BIT DATA

Procedimiento

Para configurar el acceso de ODBC a orígenes de datos de IBM InfoSphere Classic Federation Server for z/OS:

1. Realice una de las tareas siguientes, en función del sistema operativo:
 - En Windows, asegúrese de que el nombre de nodo correspondiente al origen de datos esté definido como DSN de sistema. Si ha utilizado el Administrador de orígenes de datos ODBC de Microsoft para definir el DSN, puede utilizar el **Panel de control** para comprobar que esté registrado como DSN de sistema.
 - En UNIX, configure el controlador ODBC.
2. Puede que sea necesario añadir el directorio de bibliotecas de cliente del origen de datos a la variable de entorno DB2LIBPATH para que las bibliotecas de cliente se carguen correctamente. Ejecute el mandato db2set para establecer la variable de entorno DB2LIBPATH y especificar el directorio de instalación de la biblioteca del cliente de origen de datos.

```
db2set DB2LIBPATH="directorio_biblioteca_cliente"
```

Donde *directorio_biblioteca_cliente* es el directorio de la biblioteca de cliente de origen de datos.

Ejemplo

```
db2set DB2LIBPATH="/opt/IBM/DB2IIClassic82/cli/lib"
```

3. Registre el derivador de ODBC.
4. Registre las definiciones de servidor para un origen de datos ODBC.
5. Cree una correlación de usuario para un origen de datos ODBC.
6. Pruebe la conexión con el servidor de orígenes de datos ODBC.
7. Registre apodos para las tablas y vistas del origen de datos ODBC.

Registro del derivador de ODBC

Debe registrar un derivador para acceder a los orígenes de datos ODBC. Los servidores federados utilizan los derivadores para comunicarse con los orígenes de datos y para recuperar datos de éstos. Los derivadores se implementan como un conjunto de archivos de biblioteca.

Procedimiento

Para registrar el derivador de ODBC:

Utilice uno de los métodos siguientes:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Emita la sentencia CREATE WRAPPER y especifique el nombre predeterminado para el derivador de ODBC.	<p>Por ejemplo:</p> <pre>CREATE WRAPPER ODBC;</pre> <p>Recuerde: Al registrar el derivador mediante el nombre predeterminado, ODBC, el servidor federado utiliza automáticamente la biblioteca de derivador de ODBC adecuada al sistema operativo en el que se ejecuta el servidor federado.</p> <p>Debe especificar la opción de derivador MODULE en servidores federados que ejecuten Linux o UNIX. La opción de derivador MODULE especifica la vía de acceso completa de la biblioteca que contiene el gestor de controladores ODBC.</p> <p>En AIX y Solaris, si está configurando el acceso a orígenes de datos IBM InfoSphere Classic Federation Server for z/OS, deben especificarse las opciones DB2_FENCED y DB2_SOURCE_CLIENT_MODE, como se muestra en el ejemplo siguiente.</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>
Emita la sentencia CREATE WRAPPER y especifique un nombre alternativo para el derivador de ODBC.	<p>Si el nombre del derivador predeterminado entra en conflicto con un nombre de derivador existente en la base de datos federada, puede sustituir el nombre del derivador predeterminado por otro de su elección. Si utiliza un nombre que no es el predeterminado, deberá incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER.</p> <p>Por ejemplo, para registrar un derivador con el nombre derivador_odbc en un servidor federado que utiliza AIX, emita esta sentencia:</p> <pre>CREATE WRAPPER derivador_odbc LIBRARY 'libdb2rcodbc.a';</pre> <p>El archivo de biblioteca del derivador especificado dependerá del sistema operativo del servidor federado.</p>

Una vez finalizada esta tarea, puede registrar las definiciones de servidor.

Archivos de biblioteca del derivador de ODBC

Los archivos de biblioteca de derivador de ODBC se añaden al servidor federado cuando se instala el derivador.

Cuando instala el derivador de ODBC, se añaden tres archivos de biblioteca a la vía de acceso del directorio predeterminado. Por ejemplo, si el servidor federado se ejecuta en AIX, los archivos de biblioteca del derivador que se añaden a la vía de acceso del directorio son **libdb2rcodbc.a**, **libdb2rcodbcF.a** y **libdb2rcodbcU.a**. El

archivo de biblioteca de derivador predeterminado es libdb2rcodbc.a. Los demás archivos de biblioteca de derivador los utiliza internamente el derivador de ODBC.

Si no utiliza el nombre de derivador por omisión cuando registra un derivador, debe incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER y especificar el nombre de archivo de biblioteca del derivador por omisión.

En la tabla siguiente se indican las vías de acceso al directorio predeterminado y los nombres de archivo de biblioteca del derivador predeterminado.

Tabla 28. Ubicaciones de biblioteca y nombres de archivo del cliente ODBC

Sistema operativo	Vía de acceso al directorio	Archivo de biblioteca de derivador
AIX	/usr/opt/vía_acceso_instalación/lib32/ /usr/opt/vía_acceso_instalación/lib64/	libdb2rcodbc.a
Linux	/opt/IBM/db2/vía_acceso_instalación/lib32 /opt/IBM/db2/vía_acceso_instalación/lib64	libdb2rcodbc.so
Solaris	/opt/IBM/db2/vía_acceso_instalación/lib32 /opt/IBM/db2/vía_acceso_instalación/lib64	libdb2rcodbc.so
Windows	%DB2PATH%\bin	db2rcodbc.dll

vía_acceso_instalación es la vía de acceso al directorio en el que está instalado Federation en UNIX o Linux.

Sentencia CREATE WRAPPER - Ejemplos para el derivador de ODBC

Utilice la sentencia CREATE WRAPPER para registrar el derivador de ODBC. Este tema ofrece ejemplos para Linux, UNIX y Windows.

En los ejemplos siguientes, *derivador_odbc* es el nombre que se asigna al derivador que se registra en la base de datos federada.

Servidor federado Linux y Solaris

El ejemplo siguiente muestra cómo registrar un derivador con el nombre predeterminado en un servidor federado que ejecuta Linux o Solaris:

```
CREATE WRAPPER odbc OPTIONS (MODULE '/opt/lib/odbc.so');
```

Debe especificar la opción de derivador MODULE en servidores federados que ejecuten Linux o UNIX. La opción de derivador MODULE especifica la vía de acceso completa de la biblioteca que contiene el gestor de controladores ODBC.

El ejemplo siguiente muestra cómo registrar un derivador con un nombre alternativo en un servidor federado que ejecuta Linux o Solaris:

```
CREATE WRAPPER derivador_odbc LIBRARY 'libdb2rcodbc.so'  
OPTIONS (MODULE '/opt/lib/odbc.so');
```

Servidor federado AIX

El ejemplo siguiente muestra cómo registrar un derivador con el nombre predeterminado en un servidor federado que ejecuta AIX:

```
CREATE WRAPPER odbc  
OPTIONS (MODULE '/usr/lib/odbc.a');
```

Debe especificar la opción de derivador MODULE en servidores federados que ejecuten UNIX. La opción de derivador MODULE especifica la vía de acceso completa de la biblioteca que contiene el gestor de controladores ODBC.

El ejemplo siguiente muestra cómo registrar un derivador con un nombre alternativo en un servidor federado que ejecuta AIX:

```
CREATE WRAPPER derivador_odbc LIBRARY 'libdb2rcodbc.a'  
  OPTIONS (MODULE '/usr/lib/odbc.a');
```

IBM InfoSphere Classic Federation Server for z/OS (AIX)

El ejemplo siguiente muestra cómo registrar un derivador de ODBC emitiendo la sentencia CREATE WRAPPER en un sistema operativo AIX. En AIX y Solaris, deben especificarse las opciones DB2_FENCED y DB2_SOURCE_CLIENT_MODE, como se muestra en el ejemplo siguiente.

```
CREATE WRAPPER odbc  
  OPTIONS (DB2_FENCED 'Y', DB2_SOURCE_CLIENT_MODE '32BIT',  
  MODULE '/opt/IBM/DB2IIClassic82/cli/lib/cacsq1cli.so');
```

Debe especificar la opción de derivador MODULE en servidores federados que ejecuten UNIX. La opción de derivador MODULE especifica la vía de acceso completa de la biblioteca que contiene el gestor de controladores ODBC.

El ejemplo siguiente muestra cómo registrar un derivador para acceder a orígenes de datos IBM InfoSphere Classic Federation Server for z/OS con un nombre alternativo:

```
CREATE WRAPPER derivador_odbc LIBRARY 'libdb2rcodbc.a'  
  OPTIONS (DB2_FENCED 'Y', DB2_SOURCE_CLIENT_MODE '32BIT',  
  MODULE '/opt/IBM/DB2IIClassic82/cli/lib/cacsq1cli.so');
```

Servidor federado Windows

El ejemplo siguiente muestra cómo registrar un derivador con el nombre predeterminado en un servidor federado que ejecuta Windows:

```
CREATE WRAPPER odbc;
```

El ejemplo siguiente muestra cómo registrar un derivador con un nombre alternativo en un servidor federado que ejecuta Windows:

```
CREATE WRAPPER derivador_odbc LIBRARY 'db2rcodbc.d11';
```

Registro de las definiciones del servidor para un origen de datos ODBC

Debe registrar cada servidor ODBC al que desee acceder en la base de datos federada.

Procedimiento

Para registrar una definición de servidor para un origen de datos ODBC:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .

Método	Descripción
Emita la sentencia CREATE SERVER.	<p>1. Emita la sentencia CREATE SERVER y especifique las opciones de servidor obligatorias:</p> <pre>CREATE SERVER nombre_definición_servidor TYPE tipo_origen_datos VERSION número_versión WRAPPER nombre_derivador OPTIONS (NODE 'nombre_nodo', CODEPAGE 'nombre_página-códigos', DBNAME 'nombre_base-datos');</pre> <p>Aunque NODE, CODEPAGE y DBNAME son opciones de servidor en la sentencia CREATE SERVER, son obligatorias para orígenes de datos ODBC:</p> <ul style="list-style-type: none"> • La opción de servidor NODE debe establecerse en el origen de datos especificado en la palabra clave DATASOURCE del archivo cac.ini. <p>Ejemplo</p> <p>Si el archivo cac.ini especifica DATASOURCE = CACSAMP tcp/150.45.37.49/5000, la opción NODE debe establecerse en CACSAMP.</p> <ul style="list-style-type: none"> • La opción de servidor CODEPAGE debe establecerse en el número de página de códigos de la página de códigos de cliente especificada en el archivo cac.ini. <p>Ejemplo</p> <p>Si el archivo cac.ini especifica CLIENT CODEPAGE = IBM-850, la opción CODEPAGE debe establecerse en 850.</p> <ul style="list-style-type: none"> • La opción de servidor DBNAME debe establecerse en el nombre de la base de datos de origen de datos a la que se desea acceder. <p>Ejemplo</p> <p>Si el nombre del origen de datos ODBC es venice, la opción DBNAME debe establecerse en venice.</p> <p>2. Una vez creada la definición del servidor, utilice la sentencia ALTER SERVER para añadir o eliminar opciones de servidor.</p>

Una vez finalizada esta tarea, puede crear una correlación de usuario.

Sentencia CREATE SERVER - Ejemplos del derivador de ODBC

Utilice la sentencia CREATE SERVER para registrar definiciones de servidor para el derivador de ODBC. Este tema ofrece un ejemplo completo con los parámetros obligatorios y un ejemplo con opciones adicionales del servidor.

El ejemplo siguiente muestra cómo registrar una definición de servidor para un origen de datos MySQL emitiendo la sentencia CREATE SERVER:

```
CREATE SERVER servidor_mysql TYPE mysql
VERSION 4.0 WRAPPER nombre_derivador
OPTIONS (NODE 'nodo_odbc', DBNAME 'venice')
```

servidor_mysql

Nombre asignado por el usuario al servidor de orígenes de datos ODBC.
No se permiten los nombres de definición de servidor duplicados.

TYPE *mysql*

Especifica el tipo de servidor de orígenes de datos para el que está configurando el acceso. Este parámetro es opcional.

VERSION 4.0

Versión del origen de datos ODBC al que se desea acceder. Este parámetro es opcional.

WRAPPER *nombre_derivador*

El nombre del derivador que ha especificado en la sentencia CREATE WRAPPER.

NODE *'nodo_odbc'*

El nombre del nodo (el nombre DSN de sistema) asignado al origen de datos ODBC al definir el DSN. En servidores federados que ejecutan Windows, este valor debe ser el nombre de un DSN de sistema en la ventana Administrador de orígenes de datos ODBC. En servidores federados que ejecutan UNIX, el nombre del nodo es el DSN definido en el archivo de configuración de ODBC. Generalmente, el archivo de configuración de ODBC se denomina `odbc.ini`.

Aunque el nombre del nodo se especifica como opción en la sentencia CREATE SERVER, es necesario para los orígenes de datos ODBC.

DBNAME *'venice'*

Opcional. Nombre del origen de datos ODBC al que se desea acceder.

Opciones de servidor

Al crear la definición de servidor, puede especificar opciones de servidor adicionales en la sentencia CREATE SERVER. Las opciones de servidor pueden ser genéricas y específicas de ODBC.

Algunos orígenes de datos ODBC (por ejemplo, MySQL) no pueden procesar comillas simples en los nombres de tablas y columnas de sentencias SQL. Para acceder a estos orígenes de datos, debe incluir las siguientes opciones de servidor en la sentencia CREATE SERVER:

- `DB2_TABLE_QUOTE_CHAR '``'`
- `DB2_ID_QUOTE_CHAR '``'`
- `DB2_AUTHID_QUOTE_CHAR '``'`

El carácter ``` es el delimitador de los identificadores como, por ejemplo, nombres de esquema, de tabla y de columna.

Por ejemplo:

```
CREATE SERVER servidor_mysql TYPE mysql
  VERSION 4.0 WRAPPER nombre_derivador
  OPTIONS (NODE 'nodo_mysql', DB2_TABLE_QUOTE_CHAR '``',
          DB2_ID_QUOTE_CHAR '``', DB2_AUTHID_QUOTE_CHAR '``')
```

Creación de una correlación de usuario para un origen de datos ODBC

Cuando se intenta acceder a un servidor ODBC, el servidor federado establece una conexión con el servidor ODBC utilizando un ID de usuario y una contraseña válidos para ese origen de datos. Para los orígenes de datos que requieren una correlación de usuario, debe definir una asociación (una correlación de usuario) entre cada ID de usuario y contraseña del servidor federado y el ID de usuario y contraseña correspondientes del origen de datos.

Acerca de esta tarea

Cree una correlación de usuario para cada ID de usuario que vaya a acceder al sistema federado para enviar solicitudes distribuidas al origen de datos ODBC.

Procedimiento

Para correlacionar un ID de usuario local con el ID de usuario y la contraseña del origen de datos ODBC:

Emita una sentencia CREATE USER MAPPING.

Por ejemplo:

```
CREATE USER MAPPING FOR IDusuario_local SERVER nombre_definición_servidor
    OPTIONS (REMOTE_AUTHID 'IDusuario_remoto', REMOTE_PASSWORD 'contraseña_remota');
```

Aunque REMOTE_AUTHID y REMOTE_PASSWORD son opciones de correlación de usuario en la sentencia CREATE USER MAPPING, estas opciones son necesarias para acceder a orígenes de datos ODBC:

La correlación de usuario debe correlacionar el ID de autorización de DB2 con el ID de usuario y la contraseña especificados en el archivo cac.ini.

Ejemplo

Si el archivo cac.ini especifica USERID = MY_USERID y USERPASSWORD = MY_PASSWORD, las opciones de la sentencia CREATE USER MAPPING deben especificarse del siguiente modo:

```
REMOTE_AUTHID = MY_USERID
REMOTE_PASSWORD = MY_PASSWORD
```

Una vez finalizada esta tarea, puede probar la conexión al origen de datos ODBC.

Sentencia CREATE USER MAPPING - Ejemplos para el derivador de ODBC

Utilice la sentencia CREATE USER MAPPING para correlacionar un ID de usuario de servidor federado con un ID de usuario y una contraseña de origen de datos ODBC. Este tema ofrece un ejemplo completo con los parámetros obligatorios y un ejemplo que muestra cómo utilizar el registro especial USER de DB2 con la sentencia CREATE USER MAPPING.

El ejemplo siguiente muestra cómo correlacionar un ID de autorización federado con un ID de usuario y una contraseña de origen de datos ODBC:

```
CREATE USER MAPPING FOR arturo SERVER servidor_mysql
    OPTIONS (REMOTE_AUTHID 'art', REMOTE_PASSWORD 'red4blue')
```

arturo Especifica el ID de autorización local que se correlaciona con el ID de usuario y la contraseña remotos definidos en el origen de datos ODBC.

servidor_mysql

Especifica el nombre de la definición del servidor que ha definido en la sentencia CREATE SERVER para el origen de datos ODBC.

'art' Especifica el ID de usuario remoto con el que se correlaciona *arturo*. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_ID en 'U' o 'L' en la sentencia CREATE SERVER.

'red4blue'

Especifica la contraseña remota asociada a *'art'*. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_PW en 'U' o 'L' en la sentencia CREATE SERVER.

Registro especial USER de DB2

Puede utilizar el registro especial USER de DB2 para correlacionar el ID de autorización de la persona que emite la sentencia CREATE USER MAPPING con el ID de autorización del origen de datos especificado en la opción de usuario REMOTE_AUTHID.

A continuación se ofrece un ejemplo de la sentencia CREATE USER MAPPING, que incluye el registro especial USER:

```
CREATE USER MAPPING FOR USER SERVER servidor_mysql
      OPTIONS (REMOTE_AUTHID 'art', REMOTE_PASSWORD 'red4blue');
```

Prueba de la conexión al servidor de orígenes de datos ODBC

Pruebe la conexión con el servidor de orígenes de datos ODBC para determinar si el servidor federado se ha configurado correctamente para acceder a orígenes de datos ODBC.

Acerca de esta tarea

Puede probar la conexión con el servidor de orígenes de datos ODBC mediante la definición de servidor y las correlaciones de usuario que ha definido.

Procedimiento

Para probar la conexión al servidor de orígenes de datos ODBC:

Abra una sesión de paso a través y emita una sentencia SELECT en las tablas del sistema de orígenes de datos ODBC. Si la sentencia SELECT devuelve una cuenta, significa que la definición del servidor y las correlaciones de usuario están correctamente configuradas.

```
SET PASSTHRU nombre_definición_servidor
SELECT count(*) FROM nombre.esquema.nombre.tabla
SET PASSTHRU RESET
```

Si la sentencia SELECT devuelve un error, debe resolver los errores de conexión.

Una vez finalizada esta tarea, podrá registrar apodos para las tablas y vistas de orígenes de datos ODBC.

Resolución de errores de conexión de origen de datos

Una conexión de prueba con el servidor de origen de datos puede devolver un error por varios motivos. Existen acciones que permiten determinar el motivo del error.

Síntoma

Se devuelve un error al intentar establecer conexión con el origen de datos.

Causa

Hay varias causas posibles que explican un problema de conexión.

Resolución del problema

Para solucionar los problemas de errores de conexión con el origen de datos, consulte los siguientes elementos:

- Verifique que el origen de datos está disponible.
- Si procede, asegúrese de que el servidor de origen de datos esté configurado para conexiones entrantes.
- Asegúrese de que los valores de correlación de usuarios para las opciones REMOTE_AUTHID y REMOTE_PASSWORD son válidos para las conexiones con el origen de datos. Modifique la correlación de usuarios o cree otra correlación de usuarios según sea necesario.
- Si procede, asegúrese de que el software de cliente del servidor federado esté instalado y configurado correctamente para conectarse al origen de datos.
- Para orígenes de datos ODBC, asegúrese de que el controlador ODBC del servidor federado esté instalado y configurado correctamente para establecer conexión con el servidor de origen de datos ODBC. En servidores federados que ejecutan Windows, utilice la herramienta ODBC Data Source Administrator para verificar el controlador. En servidores federados que ejecutan UNIX, consulte la documentación del proveedor del cliente ODBC.
- Verifique que los valores para las variables establecidas en el servidor federado son correctas para el origen de datos. Estas variables incluyen las variables de entorno del sistema, las variables de la archivo db2dj.ini y las variables del registro de perfiles de DB2 (db2set).
- Compruebe la definición del servidor. Si es necesario, elimine la definición del servidor y créela de nuevo.

Registro de apodos para vistas y tablas de orígenes de datos ODBC

Para cada definición de servidor ODBC que registre, debe registrar un apodo para cada tabla o vista a la que desee acceder. Utilice estos apodos en lugar de los nombres de los objetos de origen de datos cuando consulte los orígenes de datos ODBC.

Antes de empezar

Actualice las estadísticas del origen de datos ODBC antes de registrar un apodo. La base de datos federada se basa en las estadísticas del catálogo de orígenes de datos para optimizar el proceso de las consultas. Utilice el mandato de origen de datos equivalente al mandato de DB2 RUNSTATS para actualizar las estadísticas del origen de datos.

Procedimiento

Para registrar un apodo para una tabla o vista de origen de datos ODBC, utilice uno de los métodos siguientes:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Emita la sentencia CREATE NICKNAME. Los apodos pueden tener una longitud de hasta 128 caracteres.	Por ejemplo: <pre>CREATE NICKNAME apodo FOR nombre_definición_servidor."esquema_remoto". "tabla.remota" ;</pre>

Cuando se crea el apodo, el servidor federado consulta el catálogo de orígenes de datos utilizando el apodo. Esta consulta prueba la conexión a la tabla o vista del origen de datos. Si la conexión no funciona, recibirá un mensaje de error.

Repita este paso para cada tabla o vista ODBC para la que desee crear un apodo.

Sentencia CREATE NICKNAME - Ejemplos para el derivador de ODBC

Utilice la sentencia CREATE NICKNAME para registrar un apodo para una tabla o vista ODBC a la que desea acceder. Este tema ofrece un ejemplo completo con los parámetros obligatorios.

El ejemplo siguiente muestra cómo registrar un apodo para una tabla o vista de ODBC mediante la sentencia CREATE NICKNAME.

```
CREATE NICKNAME cust_europe FOR servidor_mysql."vinnie"."italy"
```

cust_europe

Apodo exclusivo utilizado para identificar la tabla o vista ODBC. El apodo debe ser exclusivo dentro del esquema.

Importante: El nombre de apodo consta de dos partes: el esquema y el apodo. Si omite el esquema al registrar el apodo, el esquema del apodo será el ID de autorización del usuario que registra el apodo.

Si el origen de datos ODBC no admite esquemas, omita el esquema en la sentencia CREATE NICKNAME.

```
servidor_mysql."vinnie"."italy"
```

Identificador de tres partes para el objeto remoto:

- *servidor_mysql* es el nombre de definición de servidor que ha asignado al servidor de orígenes de datos ODBC en la sentencia CREATE SERVER.
- *vinnie* es el ID de usuario del propietario al que pertenece la tabla o vista.
- *italy* es el nombre de la tabla o la vista remota a la que desea acceder.

El servidor federado convierte a mayúsculas los nombres de los esquemas y tablas ODBC, a menos que especifique los nombres entre comillas.

Optimización del rendimiento del derivador de ODBC con el programa de utilidad de ajuste de ODBC (db2fedsvr cfg)

Puede optimizar el rendimiento del derivador de ODBC con el programa de utilidad de ajuste de ODBC, db2fedsvr cfg. Este programa de utilidad ejecuta un conjunto de consultas predefinidas en los orígenes de datos y comprueba la exactitud de los resultados. El programa de utilidad crea un conjunto de sentencias ALTER SERVER que pueden ejecutarse en el servidor para establecer las opciones de servidor a fin de obtener un rendimiento óptimo.

Antes de empezar

Deben configurarse los elementos siguientes en el servidor federado:

- Federation
- El derivador de ODBC
- El software de cliente ODBC
- Variables para el entorno del sistema. Es posible que sean necesarias las variables de entorno ODBCINI y LIBPATH.

Para obtener información acerca de la instalación, configuración y requisitos de ODBC, consulte la documentación suministrada con cada uno de estos productos.

Procedimiento

Para optimizar el rendimiento del derivador de ODBC con el programa de utilidad de ajuste de ODBC:

1. Si el servidor ODBC no se ha definido, utilice el Centro de control de DB2 o el Procesador de línea de mandatos de DB2 para conectarse al servidor federado y crear un servidor y un derivador de ODBC.
2. Opcional: si las tablas de prueba ya existen en el origen de datos ODBC, conéctese al origen de datos ODBC y elimínelas.
3. Ejecute el programa de utilidad de ajuste de ODBC desde la línea de mandatos de DB2 con las opciones que desee utilizar. El programa de utilidad puede tardar algún tiempo en finalizar.
4. Compruebe que el programa de utilidad se ha ejecutado correctamente. Si el programa de utilidad se ha ejecutado correctamente, verá el mensaje siguiente en la ventana de mandatos o en el archivo de salida especificado:

```
ALTER SERVER "DS1" OPTIONS (ADD opción1, 'valor1')
ALTER SERVER "DS1" OPTIONS (ADD opción2, 'valor2')
ALTER SERVER "DS1" OPTIONS (ADD opción3, 'valor3')
.....
```

El mandato db2fedsvr cfg ha finalizado satisfactoriamente.

Si el mandato no se ha ejecutado correctamente, verá un mensaje de error que indicará la razón del error. Corrija el problema y vuelva a ejecutar el mandato.

Es necesario crear las tablas de prueba manualmente en las siguientes circunstancias:

- Si desea utilizar nombres de tabla diferentes del predeterminado
 - Si el origen de datos al que accede mediante ODBC es sólo de lectura
 - Si el programa de utilidad de ajuste de ODBC no puede crear las tablas de prueba necesarias para el programa de utilidad
5. Conéctese al servidor federado en el que se ha definido el origen de datos.
 6. Utilice el programa de utilidad db2look para guardar los valores de servidor existentes antes de ejecutar el archivo creado por el programa de utilidad de ajuste de ODBC. Consulte la documentación del programa de utilidad db2look para obtener información acerca de cómo guardar los valores de servidor existentes.
 7. Opcional: si el servidor ODBC está definido, puede conectarse al servidor federado y eliminar las opciones del servidor. El programa de utilidad crea sentencias ALTER SERVER en el formato descrito en el paso 4. Si estas opciones de servidor ya se han añadido, las sentencias ALTER SERVER fallarán.
 8. Utilice el mandato siguiente para ejecutar las sentencias ALTER SERVER generadas por el programa de utilidad en la base de datos federada. Las sentencias ALTER SERVER creadas por el programa de utilidad de ajuste de ODBC se encuentran en el archivo db2fedsvr cfg.sql.
db2 -tvf db2fedsvr cfg.sql
 9. Compruebe los resultados de las sentencias ALTER SERVER. Si alguna de las sentencias ha fallado, puede modificarla en el archivo db2fedsvr cfg.sql y ejecutarla de nuevo hasta que sea satisfactoria.

10. Una vez finalizado el ajuste del servidor con el programa de utilidad de ajuste de ODBC, establezca la opción de servidor PUSHDOWN en 'Y' para completar el proceso de optimización.

Sintaxis del mandato db2fedsvrcfg - programa de utilidad de ajuste de ODBC

Utilice el mandato db2fedsvrcfg para mejorar el rendimiento del derivador de ODBC.

Sintaxis

```
db2fedsvrcfg -s nombreServidor [-m bibliotecaGestorControladoresODBC] -dsn
nombreDSNodbc [-dbname nombreBDod] [-u idUsuario] [-p contraseña] [-noprep] [-prefix
prefijoNombreTabla] [-suffix sufijoNombreTabla] [-dscp páginaCódigos] [-v] [-o
archivoSalida] [-h]
```

Parámetros

Utilice el mandato db2fedsvrcfg32 cuando utilice controladores ODBC de 32 bits en AIX o Solaris. De lo contrario, utilice el mandato db2fedsvrcfg.

-dbname nombreBDod

El nombre de base de datos del origen de datos.

-dscp páginaCódigos

El identificador de página de códigos del origen de datos. Si no se especifica esta opción, el programa de utilidad utiliza la página de códigos del entorno del usuario. Este parámetro es opcional.

-dsn nombreDSNodbc

El nombre de origen de datos (DSN) del sistema para el origen de datos.

-h Provoca la visualización de ayuda detallada. Este parámetro es opcional.

-m bibliotecaGestorControladoresODBC

Nombre de archivo completo de la biblioteca del gestor de controladores ODBC. El nombre de archivo de biblioteca del gestor de controladores ODBC es opcional para Windows.

-noprep

Impide la creación de las tablas de prueba en el origen de datos antes de las pruebas. Este parámetro es opcional.

-o archivoSalida

Nombre de archivo completo del archivo de salida del programa de utilidad de ajuste de ODBC. El archivo de salida contiene las sentencias ALTER SERVER utilizadas para ajustar el rendimiento del derivador de ODBC. Este parámetro es opcional. Si no se especifica este parámetro, la salida se visualiza en la ventana de mandatos.

-p contraseña

La contraseña de conexión al origen de datos. Este parámetro es opcional.

-prefix prefijoNombreTabla

El prefijo de los nombres de tablas de origen de datos ODBC que el programa de utilidad utiliza para el análisis. Si no se especifica el prefijo, se utiliza el prefijo predeterminado, IITEST. Este parámetro es opcional.

-s nombreServidor

El nombre del servidor federado.

-suffix sufijoNombreTabla

El sufijo de los nombres de tablas de origen de datos ODBC que el programa de utilidad utiliza para el análisis. Si no se especifica un sufijo, se utiliza una serie vacía.

-u idUsuario

El nombre de usuario de conexión al origen de datos. Este parámetro es opcional.

-v Especifica que la salida del programa de utilidad es verbosa. Este parámetro es opcional.

Ejemplo

El ejemplo siguiente muestra el mandato que se ejecuta en datastore, el origen de datos ODBC. En este ejemplo, las tablas de prueba se denominan ABC n XYZ, siendo n un número del 1 al 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"
```

Definiciones de tablas de prueba para el programa de utilidad de ajuste de ODBC (db2fedsvr cfg)

En algunos casos, es necesario crear manualmente las tablas de prueba para el programa de utilidad de ajuste de ODBC. En este tema se describen las definiciones de tablas de prueba.

Es necesario crear las tablas de prueba para el programa de utilidad de ajuste de ODBC en las siguientes circunstancias:

- Si desea utilizar nombres de tabla diferentes del predeterminado
- Si el origen de datos al que accede mediante ODBC es sólo de lectura
- Si el programa de utilidad de ajuste de ODBC no puede crear las tablas de prueba necesarias para el programa de utilidad

La definición de tabla que sigue es aplicable a la totalidad de las siete tablas de prueba necesarias (IITEST1 a IITEST7). El prefijo del nombre de tabla predeterminado es IITEST, y el sufijo predeterminado es una serie vacía. Si especifica un prefijo y un sufijo diferentes, debe especificar las opciones *-prefix* y *-suffix* al ejecutar el programa de utilidad de ajuste de ODBC.

Tabla 29. Definición de tabla de prueba para el programa de utilidad de ajuste de ODBC para la tabla IITEST1

Nombre de columna	Tipo de datos SQL	Identificador de tipo de datos SQL	Longitud
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

Tabla 30. Definición de tabla de prueba para el programa de utilidad de ajuste de ODBC para la tabla IITEST2

Nombre de columna	Tipo de datos SQL	Identificador de tipo de datos SQL	Longitud
IT2C1	integer	SQL_INTEGER	
IT2C2	integer	SQL_INTEGER	
IT2C3	char(30)	SQL_CHAR	30

Tabla 31. Definición de tabla de prueba para el programa de utilidad de ajuste de ODBC para las tablas IITEST3 a IITEST7

Número de columnas de cada tabla	Nombre de columna	Tipo de datos SQL	Identificador de tipo de datos SQL	Longitud
66	ITxCn	char(100)	SQL_CHAR	100

x Número correspondiente a la tabla que se define.

n Número de columna.

Por ejemplo, IT3C1 es el nombre de columna de la primera columna de la tabla IITEST3.

Configuración del acceso a orígenes de datos OLE DB

Para configurar el servidor federado para acceder a orígenes de datos OLE DB, debe proporcionar al servidor federado información acerca de los proveedores de OLE DB.

Antes de empezar

- Federation debe estar instalado en el servidor que actúa como servidor federado.
- Compruebe la configuración del servidor federado.
- Compruebe el parámetro federado para asegurarse de que la federación está habilitada.

Microsoft OLE DB es un conjunto de interfaces de modelo de objetos componentes (COM) que permiten a las aplicaciones acceder a información que no está almacenada normalmente en bases de datos. Un consumidor de OLE DB es una parte de un código de aplicación o sistema que consume una interfaz de OLE DB. Un proveedor de OLE DB es un componente de software que expone una interfaz de OLE DB.

Puede configurar un servidor federado para acceder a los datos almacenados en los orígenes de datos OLE DB emitiendo sentencias SQL en la línea de mandatos de DB2.

Después de configurar el acceso al origen de datos OLE DB, utilice la sentencia CREATE FUNCTION para registrar una función de tabla externa de OLE DB definida por el usuario en la base de datos federada.

Procedimiento

Para configurar el acceso a los orígenes de datos OLE DB en un servidor federado:

1. Registre el derivador.
2. Registre la definición de servidor.

3. Cree las correlaciones de usuarios.

Registro del derivador de OLE DB

Debe registrar un derivador para acceder a los orígenes de datos OLE DB. Los servidores federados utilizan derivadores para comunicarse con los orígenes de datos y recuperar datos de éstos. Los derivadores se implementan como un conjunto de archivos de biblioteca.

Acerca de esta tarea

El derivador de OLE DB permite acceder a los proveedores de OLE DB que son compatibles con Microsoft OLE DB versión 2.0 (o posterior).

El derivador de OLE DB sólo se utiliza como ayuda en el registro de funciones de tabla externas de OLE DB definidas por el usuario en la base de datos federada. A diferencia de otros derivadores, el derivador de OLE DB no utiliza apodos para acceder a los datos almacenados en los orígenes de datos.

Procedimiento

Para registrar el derivador de OLE DB:

Utilice uno de los métodos siguientes para registrar el derivador de OLE DB:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Emita la sentencia CREATE WRAPPER y especifique el nombre predeterminado del derivador de OLE DB.	<p>Por ejemplo:</p> <pre>CREATE WRAPPER OLEDB</pre> <p>Recuerde: Cuando registra el derivador utilizando el nombre predeterminado, OLEDB, el servidor federado utiliza automáticamente la biblioteca de derivador de OLE DB correspondiente al sistema operativo donde se ejecuta el servidor federado.</p> <p>Si el nombre del derivador predeterminado entra en conflicto con un nombre de derivador existente en la base de datos federada, puede sustituir el nombre del derivador predeterminado por otro de su elección. Si utiliza un nombre distinto al nombre predeterminado, deberá incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER.</p> <p>Por ejemplo, para registrar un derivador con el nombre oledb_wrapper en un servidor federado que utiliza Windows, emita la siguiente sentencia:</p> <pre>CREATE WRAPPER oledb_wrapper LIBRARY 'db2oledb.dll'</pre> <p>El archivo de biblioteca de derivador que especifica depende del sistema operativo del servidor federado.</p>

Cuando finalice esta tarea, puede registrar la definición de servidor.

Archivos de biblioteca de derivador de OLE DB

Los archivos de biblioteca de derivador de OLE DB se añaden al servidor federado cuando se instala el derivador.

Cuando instala el derivador de OLE DB, se añade el archivo de biblioteca a la vía de acceso de directorios predeterminada.

Si no utiliza el nombre de derivador predeterminado cuando registra el derivador, debe incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER y especificar el nombre de archivo de biblioteca.

La vía de acceso de directorios predeterminada y el nombre de archivo de biblioteca del derivador predeterminado se especifican en la tabla siguiente.

Tabla 32. Nombres de archivo y ubicaciones de la biblioteca de cliente de OLE DB

Sistema operativo	Vía de acceso de directorios	Nombre de archivo de biblioteca
Windows	%DB2PATH%\bin	db2oledb.dll

Registro de las definiciones de servidor para un origen de datos OLE DB

Debe registrar cada servidor OLE DB al que desee acceder en la base de datos federada.

Acerca de esta tarea

Puede registrar una definición de servidor para un origen de datos OLE DB desde la línea de mandatos de DB2.

Procedimiento

Para registrar una definición de servidor para un origen de datos OLE DB:

Emita la sentencia CREATE SERVER.

Por ejemplo:

```
CREATE SERVER nombre_definición_servidor WRAPPER
nombre_derivador
OPTIONS (CONNECTSTRING
'palabra_clave=valor;palabra_clave=valor');
```

Aunque la variable CONNECTSTRING se especifica como una opción en la sentencia CREATE SERVER, esta opción se necesita para los orígenes de datos OLE DB.

Cuando finalice esta tarea, puede crear una correlación de usuarios.

Sentencia CREATE SERVER - Ejemplos para el derivador de OLE DB

Utilice la sentencia CREATE SERVER para registrar definiciones de servidor para el derivador de OLE DB. Este tema proporciona un ejemplo completo con los parámetros necesarios y un ejemplo con opciones adicionales del servidor.

El ejemplo siguiente muestra cómo registrar una definición de servidor para un derivador de OLE DB emitiendo la sentencia CREATE SERVER:

```
CREATE SERVER Nwind WRAPPER OLEDB
  OPTIONS (CONNECTSTRING 'Provider=Microsoft.Jet.OLEDB.4.0;
    Data Source=c:\msdasdk\bin\oledb\nwind.mdb');
```

Nwind Nombre que asigna al origen de datos OLE DB. No se permiten los nombres de definición de servidor duplicados.

WRAPPER OLEDB

Nombre de derivador que ha especificado en la sentencia CREATE WRAPPER.

CONNECTSTRING 'Provider=Microsoft.Jet.OLEDB.4.0; Data Source=c:\msdasdk\bin\oledb\nwind.mdb'

Proporciona las propiedades de inicialización que son necesarias para conectarse a un proveedor de OLE DB.

El valor de la opción CONNECTSTRING contiene una serie de pares de palabra clave y valor separados por punto y coma. El signo igual (=) separa cada palabra clave y su valor. Las palabras clave son las descripciones de las propiedades de inicialización de OLE DB (conjunto de propiedades DBPROPSET_DBINT) o las palabras clave específicas del proveedor.

Para ver la sintaxis y la semántica completa de la opción CONNECTSTRING, consulte *Microsoft OLE DB 2.0 Programmer's Reference and Data Access SDK*, Microsoft Press, 1998.

Opciones de servidor

Cuando crea la definición de servidor, puede especificar opciones de servidor adicionales en la sentencia CREATE SERVER. Las opciones de servidor pueden ser opciones de servidor generales y opciones de servidor específicas de OLE DB.

La opción de servidor COLLATING_SEQUENCE especifica si el origen de datos utiliza la misma secuencia de clasificación que el servidor federado. El valor predeterminado para la opción de servidor COLLATING_SEQUENCE es 'N'. Cuando el origen de datos OLE DB utiliza la misma secuencia de clasificación que el servidor federado, establezca la opción de servidor COLLATING_SEQUENCE como 'Y'.

El siguiente ejemplo es una definición de servidor de OLE DB que incluye la opción de servidor 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')
```

Creación de las correlaciones de usuarios para un origen de datos OLE DB

Cuanto intenta acceder a un servidor OLE DB, el servidor federado establece una conexión con el servidor OLE DB utilizando un ID de usuario y una contraseña que son válidos para dicho origen de datos. Debe definir una asociación (una correlación de usuarios) entre cada ID y contraseña de autorización federada y el ID de usuario y la contraseña de origen de datos correspondiente.

Procedimiento

Para correlacionar un ID de autorización local con el ID de usuario y la contraseña remotos de OLE DB:

Emita una sentencia CREATE USER MAPPING.

Por ejemplo:

```
CREATE USER MAPPING FOR IDusuario_local
SERVER nombre_definición_servidor
    OPTIONS (REMOTE_AUTHID 'IDusuario_remoto', REMOTE_PASSWORD
'contraseña_remota') ;
```

Si la longitud de la contraseña en el origen de datos OLE DB o la contraseña en el servidor federado tiene menos de ocho caracteres, las sentencias SQL que acceden al origen de datos OLE DB fallarán. Aparece el siguiente mensaje de error:

```
SQL30082N
El intento de establecer una conexión ha fallado con la razón de seguridad "15"
("PROCESSING FAILURE"). SQLSTATE=08001
```

Para evitar este problema, cambie la contraseña del origen de datos OLE DB o cambie la contraseña en el servidor federado a ocho o más caracteres.

Sentencia CREATE USER MAPPING - Ejemplos para el derivador de OLE DB

Utilice la sentencia CREATE USER MAPPING para correlacionar un ID de autorización federado con un ID de usuario y una contraseña remotos de OLE DB. En este tema se proporciona un ejemplo completo con los parámetros necesarios y un ejemplo que muestra cómo utilizar el registro especial USER de DB2 con la sentencia CREATE USER MAPPING.

El ejemplo siguiente muestra cómo correlacionar un ID de autorización federado con un ID de usuario y una contraseña remotos de origen de datos OLE:

```
CREATE USER MAPPING FOR laura SERVER Nwind
    OPTIONS (REMOTE_AUTHID 'lulu', REMOTE_PASSWORD 'raiders')
```

laura Especifica el ID de autorización local que está correlacionando con un ID de usuario y una contraseña remotos definidos en el origen de datos OLE DB.

SERVER *Nwind*

Especifica el nombre de definición de servidor que ha registrado en la sentencia CREATE SERVER del servidor OLE DB.

REMOTE_AUTHID '*lulu*'

Especifica el ID de usuario en el servidor de bases de datos OLE DB con el que está correlacionando *lulu*. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_ID en 'U' o 'L' en la sentencia CREATE SERVER.

REMOTE_PASSWORD '*raiders*'

Especifica la contraseña asociada a '*lulu*'. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_PW en 'U' o 'L' en la sentencia CREATE SERVER.

Registro especial USER de DB2

Puede utilizar el registro especial USER de DB2 para correlacionar el ID de autorización de la persona que emite la sentencia CREATE USER MAPPING con el ID de autorización de origen de datos especificado en la opción de usuario REMOTE_AUTHID.

El siguiente ejemplo muestra una sentencia CREATE USER MAPPING que incluye el registro especial USER:

```
CREATE USER MAPPING FOR USER SERVER Nwind
    OPTIONS (REMOTE_AUTHID 'lulu', REMOTE_PASSWORD 'raiders');
```

Configuración del acceso a orígenes de datos Oracle

Para configurar un sistema federado para acceder a orígenes de datos Oracle, debe proporcionar al sistema federado información acerca de los orígenes de datos y los objetos a los que desea acceder.

Antes de empezar

- El software de cliente Oracle debe estar instalado y configurado en el servidor que actúa como el servidor federado.
- Federation debe estar instalado en el servidor que actúa como servidor federado.
- Compruebe la configuración del servidor federado.
- Compruebe el parámetro federado para asegurarse de que la federación está habilitada.

Procedimiento

Para añadir orígenes de datos Oracle a un sistema federado:

1. Establezca las variables de entorno de Oracle.
2. Configure y pruebe el archivo de configuración del cliente Oracle.
3. Registre el derivador de Oracle.
4. Registre las definiciones de servidor para un origen de datos Oracle.
5. Cree las correlaciones de usuarios.
6. Pruebe la conexión con el servidor Oracle.
7. Registre los apodos.

Establecimiento de las variables de entorno de Oracle

Las variables de entorno de Oracle deben establecerse en el archivo db2dj.ini en el servidor federado.

Restricciones

Revise las restricciones del archivo db2dj.ini.

El archivo db2dj.ini contiene información de configuración sobre el software de cliente Oracle que hay instalado en el servidor federado.

Existen variables de entorno necesarias y opcionales para los orígenes de datos Oracle.

Si ha instalado el software de cliente Oracle antes de instalar el derivador de Oracle, las variables de entorno de Oracle necesarias se establecen en el archivo db2dj.ini.

Si no ha instalado el software de cliente Oracle antes de instalar el derivador de Oracle, o si desea establecer algunas de las variables de entorno opcionales, debe establecer las variables de entorno utilizando los pasos de esta tarea.

Procedimiento

Para establecer las variables de entorno de Oracle:

1. Utilice uno de los métodos siguientes:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Establezca automáticamente las variables de entorno.	Ejecute el asistente de IBM InfoSphere Federation Server. Siga las instrucciones del asistente. Importante: Establezca las variables de entorno necesarias ejecutando el asistente de instalación. Las variables de entorno opcionales deben establecerse manualmente.
Establezca manualmente las variables de entorno.	Edite el archivo <code>db2dj.ini</code> : El archivo <code>db2dj.ini</code> se encuentra en el directorio especificado por la variable de registro <code>DB2_DJ_INI</code> de DB2. Si la variable <code>DB2_DJ_INI</code> no se establece, el archivo <code>db2dj.ini</code> reside en una de las siguientes vías de acceso predeterminadas, dependiendo del sistema operativo: <ul style="list-style-type: none"> • En Linux y UNIX: <code>padre_instancia/sql1lib/cfg/db2dj.ini</code>. <i>padre_instancia</i> El directorio padre del propietario de la instancia • En Windows: <code>%DB2PATH%\cfg\db2dj.ini</code> %DB2PATH% El directorio donde está instalado el sistema de base de datos de DB2, por ejemplo, <code>C:\Archivos de programa\IBM\sql1ib</code>. <p>Si el archivo no existe, puede crear un archivo con el nombre <code>db2dj.ini</code> utilizando un editor de texto. En el archivo <code>db2dj.ini</code>, debe especificar la vía de acceso completa en el valor de las variables de entorno; de lo contrario, aparecerán errores.</p>

2. En los servidores federados que ejecutan UNIX, añada la variable de entorno Oracle al archivo `.profile` de la instancia de DB2. Por ejemplo:

```
export
ORACLE_HOME=directorio_padre_oracle
export PATH=$ORACLE_HOME/bin:$PATH
```

Donde *directorio_padre_oracle* es el directorio donde está instalado el software de cliente Oracle.

3. En los servidores federados que ejecutan Linux o UNIX, ejecute el archivo `.profile` de la instancia de DB2 especificando:

```
.$HOME/.profile
```

4. Para asegurarse de que las variables de entorno están establecidas en el servidor federado, recicle la instancia de DB2 con estos mandatos:

```
db2stop
db2start
```

Cuando finalice esta tarea, puede configurar y probar el cliente Oracle.

Variables de entorno de Oracle

Existen variables de entorno necesarias y variables de entorno opcionales para orígenes de datos Oracle. Estas variables se definen en el archivo `db2dj.ini`.

Las variables de entorno válidas para Oracle son:

- ORACLE_HOME
- ORACLE_BASE (opcional)
- ORA_NLS (opcional)
- NLS_LANG (opcional)
- TNS_ADMIN (opcional)

Descripción de las variables

ORACLE_HOME

Establezca la variable de entorno ORACLE_HOME en la vía de acceso de directorio en la que se ha instalado el software de cliente Oracle.

Especifique la vía de acceso completa para la variable de entorno:

ORACLE_HOME=*directorio_padre_oracle*. Por ejemplo, si el directorio padre de Oracle es `\usr\oracle\8.1.7`, la entrada en el archivo `db2dj.ini` es `ORACLE_HOME=\usr\oracle\8.1.7`.

Si un usuario individual de la instancia federada establece la variable de entorno ORACLE_HOME localmente, la instancia federada no utiliza dicha serie. La instancia federada sólo utiliza el valor de ORACLE_HOME definido en el archivo `db2dj.ini`.

ORACLE_BASE

ORACLE_BASE representa la raíz del árbol de directorios del cliente Oracle. Si ha establecido la variable de entorno ORACLE_BASE al instalar el software de cliente Oracle, establezca la variable de entorno ORACLE_BASE en el servidor federado.

Por ejemplo:

```
ORACLE_BASE=directorio_raíz_oracle
```

ORA_NLS

Si ejecuta varias versiones de Oracle en el sistema, debe asegurarse de que:

- La variable de entorno ORA_NLS adecuada esté definida.
- Los archivos de datos NLS correspondientes para las versiones que utiliza estén disponibles.

Los datos específicos de la ubicación están almacenados en un directorio especificado por la variable de entorno ORA_NLS. Cada versión de Oracle tiene un directorio de datos ORA_NLS distinto.

Tabla 33. Variable de entorno ORA_NLS por versión

Versión de Oracle	Variable de entorno
8.x, 9.x	ORA_NLS33
10.x	ORA_NLS10

Por ejemplo, en los servidores federados que ejecutan UNIX que acceden a orígenes de datos Oracle 8.1, el valor de la variable de entorno ORA_NLS33 es:

```
ORA_NLS33=directorio_padre_oracle/ocommon/nls/admin/<data>
```

NLS_LANG

La variable de entorno NLS_LANG es una variable de entorno de página de códigos. Consulte la documentación NLS de Oracle para obtener información sobre cómo definir esta variable.

TNS_ADMIN

En servidores federados que ejecutan Windows

El cliente Oracle busca el archivo `tnsnames.ora` en el directorio `%ORACLE_HOME%\NETWORK\ADMIN` (`%ORACLE_HOME%` se define en el archivo `db2dj.ini`). Si el archivo `tnsnames.ora` no se encuentra en el directorio `%ORACLE_HOME%\NETWORK\ADMIN`, debe establecer la variable de entorno `TNS_ADMIN` en el archivo `db2dj.ini` en el servidor federado. La variable de entorno del archivo `db2dj.ini` se establece en la vía de acceso en la que se encuentra el archivo `tnsnames.ora`.

En servidores federados que ejecutan AIX o Linux

El cliente Oracle busca el archivo `tnsnames.ora` en el directorio `/etc`. Si el archivo `tnsnames.ora` no se encuentra en el directorio `/etc`, el cliente Oracle busca el archivo `tnsnames.ora` en el directorio `$ORACLE_HOME/network/admin` (`$ORACLE_HOME` se define en el archivo `db2dj.ini`). Si el archivo `tnsnames.ora` no se encuentra en el directorio `$ORACLE_HOME/network/admin`, debe establecer la variable de entorno `TNS_ADMIN` en el servidor federado. La variable de entorno del archivo `db2dj.ini` se establece en la vía de acceso en la que se encuentra el archivo `tnsnames.ora`.

Por ejemplo, si el archivo `tnsnames.ora` se encuentra en el directorio `/home/oracle`, debe establecer la variable de entorno en:

```
TNS_ADMIN=/home/oracle
```

En servidores federados que ejecutan Solaris

El cliente Oracle busca el archivo `tnsnames.ora` en el directorio `/var/opt/oracle`. Si el archivo `tnsnames.ora` no se encuentra en el directorio `/var/opt/oracle`, el cliente Oracle busca el archivo `tnsnames.ora` en el directorio `$ORACLE_HOME/network/admin` (`$ORACLE_HOME` se define en el archivo `db2dj.ini`). Si el archivo `tnsnames.ora` no se encuentra en el directorio `$ORACLE_HOME/network/admin`, debe establecer la variable de entorno `TNS_ADMIN`. La variable de entorno se establece en el archivo `db2dj.ini` del directorio en el que se encuentra el archivo `tnsnames.ora`.

Por ejemplo, si el archivo `tnsnames.ora` se encuentra en el directorio `/home/oracle`, debe establecer la variable de entorno en:

```
TNS_ADMIN=/home/oracle
```

Conversión de la página de códigos de Oracle:

Cada vez que el derivador Oracle se conecta a un origen de datos Oracle, determina qué valor de página de códigos debe utilizar para la conexión. Puede especificar que el derivador Oracle establezca el valor de página de códigos o puede designar una página de códigos mediante la variable de entorno `NLS_LANG`.

Las variables de entorno que especifican la conversión de página de códigos de Oracle se establecen en el archivo `db2dj.ini` del servidor federado.

Si la variable de entorno `NLS_LANG` se establece en el archivo `db2dj.ini` del servidor federado, el derivador utiliza el valor de página de códigos del archivo `db2dj.ini`.

Si la variable de entorno NLS_LANG no se establece en el archivo db2dj.ini del servidor federado, el derivador determina el territorio y la página de códigos de la base de datos federada. El derivador establece la variable de entorno NLS_LANG en el entorno local de Oracle coincidente más cercano. Si no existe ningún entorno local coincidente cercano, la variable de entorno NLS_LANG se establece en American_America.US7ASCII.

Consulte la documentación que se incluye con el software Oracle para obtener una lista de los entornos locales válidos.

Ejemplo de página de códigos para chino GB 18030:

Si accede a un origen de datos Oracle que contiene datos codificados con la página de códigos para chino GB 18030, y la base de datos federada utiliza la página de códigos UTF-8, el derivador Oracle establece la variable de entorno NLS_LANG de Oracle de este modo:

```
NLS_LANG=Simplified Chinese_China.UTF8
```

Este valor es correcto si utiliza un cliente Oracle 8; no obstante, si utiliza un cliente Oracle versión 9i (o posterior), debe establecer la variable de entorno NLS_LANG para sobrescribir el valor predeterminado del derivador Oracle. Establezca la variable de entorno NLS_LANG en Simplified Chinese_China.AL32UTF8, de modo que el cliente Oracle 9i convierta correctamente los datos GB 18030 en Unicode.

Por ejemplo:

```
NLS_LANG=Simplified Chinese_China.AL32UTF8
```

Configuración y prueba del archivo de configuración del cliente Oracle

El archivo de configuración del cliente Oracle se utiliza para conectarse a las bases de datos Oracle utilizando las bibliotecas de cliente que hay instaladas en el sistema federado.

Acerca de esta tarea

El archivo de configuración del cliente especifica la ubicación de cada servidor de bases de datos Oracle y el tipo de conexión (protocolo) del servidor de bases de datos.

El nombre predeterminado del archivo de configuración del cliente Oracle es tnsnames.ora.

La ubicación predeterminada del archivo de configuración del cliente depende del sistema operativo utilizado por el sistema federado:

- En los sistemas Linux y UNIX, la ubicación predeterminada del archivo es \$ORACLE_HOME/network/admin.
- En los sistemas Windows, la ubicación predeterminada del archivo es %ORACLE_HOME%\NETWORK\ADMIN.

Procedimiento

Para configurar y probar el archivo de configuración del cliente Oracle:

1. Cree el archivo `tnsnames.ora` utilizando el programa de utilidad de configuración NET8/NET de Oracle que se proporciona con el software de cliente Oracle.
En el archivo `tnsnames.ora`, SID (o `SERVICE_NAME`) es el nombre de la instancia Oracle y `HOST` es el nombre de host donde se encuentra el servidor Oracle.
2. Si desea ubicar el archivo `tnsnames.ora` en una vía de acceso distinta a la vía de acceso de búsqueda predeterminada, establezca la variable de entorno `TNS_ADMIN` para especificar la ubicación del archivo:
 - a. Edite el archivo `db2dj.ini` en el directorio `sql1lib/cfg` y establezca la variable de entorno `TNS_ADMIN`. Por ejemplo:
`TNS_ADMIN=x:/path/`
 - b. Emita los siguientes mandatos para reciclar la instancia de DB2 y comprobar que la variable de entorno está establecida en el programa:
`db2stop`
`db2start`
3. Pruebe la conexión para asegurarse de que el software de cliente puede conectarse con el servidor Oracle. Utilice el programa de utilidad `sqlplus` de Oracle para probar la conexión.

Cuando finalice esta tarea, puede registrar el derivador de Oracle.

Registro del derivador de Oracle

Debe registrar un derivador para acceder a los orígenes de datos Oracle. Los servidores federados utilizan derivadores para comunicarse con los orígenes de datos y recuperar datos de éstos. Los derivadores se implementan como un conjunto de archivos de biblioteca.

Procedimiento

Para registrar un derivador de Oracle:

Utilice uno de los métodos siguientes:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .

Método	Descripción
<p>Emita la sentencia CREATE WRAPPER y especifique el nombre predeterminado del derivador de Oracle.</p>	<p>Por ejemplo: <code>CREATE WRAPPER NET8 ;</code></p> <p>Recuerde: Cuando registra el derivador utilizando el nombre predeterminado, NET8, el servidor federado utiliza automáticamente la biblioteca de derivador de Oracle correspondiente al sistema operativo donde se ejecuta el servidor federado.</p> <p>Si no utiliza el nombre de derivador predeterminado cuando registra un derivador, debe incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER y especificar el nombre de archivo de biblioteca del derivador predeterminado.</p> <p>Por ejemplo, para registrar un derivador con el nombre oracle_wrapper en el servidor federado que utiliza AIX, emita la siguiente sentencia: <code>CREATE WRAPPER oracle_wrapper LIBRARY 'libdb2net8.a'</code></p> <p>El archivo de biblioteca de derivador que especifica depende del sistema operativo del servidor federado.</p>

Cuando finalice esta tarea, puede registrar las definiciones de servidor.

Archivos de biblioteca de derivador de Oracle

Los archivos de biblioteca de derivador de Oracle se añaden al servidor federado cuando se instala el derivador.

Cuando instala el derivador de Oracle, se añaden tres archivos de biblioteca a la vía de acceso de directorios predeterminada. Por ejemplo, si el servidor federado se ejecuta en AIX, los archivos de biblioteca que se añaden a la siguiente vía de acceso de directorios son libdb2net8.a, libdb2net8F.a y libdb2net8U.a. El archivo de biblioteca del derivador predeterminado es libdb2net8.a. El derivador de Oracle utiliza internamente los otros archivos de biblioteca de derivador.

Las vías de acceso de directorios predeterminadas y los nombres de archivo de biblioteca del derivador predeterminados se especifican en la tabla siguiente.

Tabla 34. Nombres de archivo y ubicaciones de la biblioteca de derivador de Oracle

Sistema operativo	Vía de acceso de directorios	Nombres de archivo de biblioteca
AIX	/usr/opt/vía_acceso_instalación/lib32/ /usr/opt/vía_acceso_instalación/lib64/	libdb2net8.a
Linux	/opt/IBM/db2/vía_acceso_instalación/lib32 /opt/IBM/db2/vía_acceso_instalación/lib64	libdb2net8.so
Solaris	/opt/IBM/db2/vía_acceso_instalación/lib32 /opt/IBM/db2/vía_acceso_instalación/lib64	libdb2net8.so

Tabla 34. Nombres de archivo y ubicaciones de la biblioteca de derivador de Oracle (continuación)

Sistema operativo	Vía de acceso de directorios	Nombres de archivo de biblioteca
Windows	%DB2PATH%\bin	db2net8.dll

vía_acceso_instalación es la vía de acceso de directorios donde se instala el servidor federado en UNIX o Linux.

Registro de las definiciones de servidor para un origen de datos Oracle

Debe registrar cada servidor Oracle al que desee acceder en la base de datos federada.

Procedimiento

Para registrar una definición de servidor para un origen de datos Oracle:

1. Ubique el nombre de nodo en el archivo `tnsnames.ora` de Oracle.

Archivo `tnsnames` de ejemplo:

```

paris_node =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = somehost)(PORT = 1521)))
    (CONNECT_DATA = (SERVICE_NAME = ora9i.seel)))
    
```

En este ejemplo, el nombre de nodo que se utiliza en la sentencia `CREATE SERVER` es `paris_node`.

2. Para crear la definición de servidor, utilice uno de los métodos siguientes:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Emita la sentencia <code>CREATE SERVER</code>	<p>Por ejemplo:</p> <pre> CREATE SERVER nombre_definición_servidor TYPE oracle VERSION número_versión WRAPPER nombre_derivador OPTIONS (NODE 'nombre_nodo') ; </pre> <p>Aunque la variable <code>nombre_nodo</code> se especifica como una opción en la sentencia <code>CREATE SERVER</code>, se necesita para los orígenes de datos Oracle.</p> <p>Después de registrar la definición de servidor, utilice la sentencia <code>ALTER SERVER</code> para añadir o eliminar opciones de servidor.</p>

Cuando finalice esta tarea, puede crear las correlaciones de usuarios.

Sentencia `CREATE SERVER` - Ejemplos para el derivador de Oracle

Utilice la sentencia `CREATE SERVER` para registrar definiciones de servidor para el derivador de Oracle. Este tema proporciona un ejemplo completo con los parámetros necesarios y un ejemplo con opciones adicionales del servidor.

El ejemplo siguiente muestra cómo registrar una definición de servidor para un derivador de Oracle emitiendo la sentencia CREATE SERVER:

```
CREATE SERVER
servidor_ora TYPE oracle VERSION
8.1.7 WRAPPER nombre_derivador
OPTIONS (NODE 'paris_node') ;
```

servidor_ora

Nombre que asigna al servidor de bases de datos Oracle. No se permiten los nombres de definición de servidor duplicados.

TYPE *oracle*

Especifica el tipo de servidor de origen de datos para el que está configurando el acceso. Para el derivador de Oracle, el tipo de servidor debe ser *oracle*.

VERSION *8.1.7*

Versión del servidor de bases de datos Oracle al que desea acceder.

WRAPPER *nombre_derivador*

Nombre de derivador que ha especificado en la sentencia CREATE WRAPPER.

NODE *'paris_node'*

Nombre del nodo donde reside el servidor de bases de datos Oracle. Obtenga el nombre de nodo del archivo tnsnames.ora. Este valor es sensible a las mayúsculas y minúsculas.

Aunque el nombre del nodo se especifica como una opción en la sentencia CREATE SERVER, se necesita para los orígenes de datos Oracle.

Opciones de servidor

Cuando crea la definición de servidor, puede especificar opciones de servidor adicionales en la sentencia CREATE SERVER. Estas opciones de servidor pueden ser opciones de servidor generales y opciones de servidor específicas de Oracle.

El servidor federado supone que todas las columnas VARCHAR de Oracle contienen blancos de cola. Si está seguro de que todas las columnas VARCHAR de la base de datos Oracle no contienen blancos de cola, puede establecer una opción de servidor para especificar que el origen de datos utiliza una semántica de comparación VARCHAR sin relleno de espacios en blanco.

El siguiente ejemplo muestra una definición de servidor Oracle con la opción de servidor VARCHAR_NO_TRAILING_BLANKS:

```
CREATE SERVER servidor_ora TYPE oracle VERSION
8.1.7 WRAPPER nombre_derivador
OPTIONS (NODE 'paris_node', VARCHAR_NO_TRAILING_BLANKS 'Y') ;
```

Utilice la opción de servidor VARCHAR_NO_TRAILING_BLANKS cuando ninguna de las columnas contenga blancos de cola. Si sólo algunas de las columnas VARCHAR contienen blancos de cola, puede establecer una opción en dichas columnas con la sentencia ALTER NICKNAME.

Creación de las correlaciones de usuarios para un origen de datos Oracle

Para acceder a un servidor Oracle, el servidor federado establece una conexión con el servidor Oracle utilizando un ID de usuario y una contraseña que son válidos para dicho origen de datos.

Restricciones

El ID de usuario en el origen de datos Oracle debe crearse utilizando el mandato `create user` con la cláusula `identified by`, en lugar de la cláusula `identified externally`.

Procedimiento

Para crear las correlaciones de usuario para un origen de datos Oracle:

Emita una sentencia `CREATE USER MAPPING`. Por ejemplo:

```
CREATE USER MAPPING FOR IDusuario_local SERVER nombre_definición_servidor
  OPTIONS (REMOTE_AUTHID 'IDusuario_remoto', REMOTE_PASSWORD
'contraseña_remota') ;
```

Cuando finalice esta tarea, pruebe la conexión con el servidor Oracle.

Sentencia `CREATE USER MAPPING` - Ejemplos para el derivador de Oracle

Utilice la sentencia `CREATE USER MAPPING` para correlacionar un ID de autorización federado con un ID de usuario y una contraseña remotos de Oracle. En este tema se proporciona un ejemplo completo con los parámetros necesarios y un ejemplo que muestra cómo utilizar el registro especial `USER` de DB2 con la sentencia `CREATE USER MAPPING`.

El ejemplo siguiente muestra cómo correlacionar un ID de autorización federado con un ID de usuario y una contraseña de Oracle:

```
CREATE USER MAPPING
FOR robert SERVER servidor_ora
  OPTIONS (REMOTE_AUTHID 'rob', REMOTE_PASSWORD 'then4now') ;
```

robert Especifica el ID de autorización que está correlacionando con un ID de usuario y una contraseña remotos definidos en el servidor Oracle.

SERVER *servidor_ora*

Especifica el nombre de definición de servidor que ha registrado en la sentencia `CREATE SERVER` del servidor Oracle.

REMOTE_AUTHID '*rob*'

Especifica el ID de usuario remoto con el que está correlacionando *robert*. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor `FOLD_ID` en 'U' o 'L' en la sentencia `CREATE SERVER`.

REMOTE_PASSWORD '*then4now*'

Especifica la contraseña remota asociada con *rob*. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor `FOLD_PW` en 'U' o 'L' en la sentencia `CREATE SERVER`.

Registro especial `USER` de DB2

Puede utilizar el registro especial `USER` de DB2 para correlacionar el ID de autorización de la persona que emite la sentencia `CREATE USER MAPPING` con el ID de autorización de origen de datos especificado en la opción de usuario `REMOTE_AUTHID`.

El siguiente ejemplo muestra una sentencia `CREATE USER MAPPING` que incluye el registro especial `USER`:

```
CREATE USER MAPPING FOR USER SERVER servidor_ora
  OPTIONS (REMOTE_AUTHID 'rob', REMOTE_PASSWORD 'then4now') ;
```

Prueba de la conexión al servidor Oracle

Pruebe la conexión con el servidor Oracle para determinar si el servidor federado se ha configurado correctamente para acceder a los orígenes de datos Oracle.

Acerca de esta tarea

Puede probar la conexión con el servidor Oracle utilizando la definición de servidor y las correlaciones de usuario que ha definido.

Procedimiento

Para probar la conexión al servidor Oracle:

Abra una sesión de paso a través y emita una sentencia SELECT en las tablas del sistema Oracle. Si la sentencia SELECT devuelve un recuento, la definición de servidor y la correlación de usuarios se han configurado correctamente.

Por ejemplo:

```
SET
PASSTHRU nombre_servidor_remoto
SELECT count(*) FROM sys.all_tables
SET PASSTHRU RESET
```

Si la sentencia SELECT devuelve un error, debe resolver los errores de conexión.

Cuando finalice esta tarea, puede registrar apodos para las tablas y vistas de Oracle.

Resolución de problemas de conectividad con los orígenes de datos Oracle

El problema más común que puede encontrar cuando configura el servidor federado para acceder a los orígenes de datos Oracle es la conectividad.

Síntoma

Si no puede conectarse a un origen de datos Oracle desde un servidor federado, es posible que deba actualizar el archivo hosts de TCP/IP.

Causa

Este problema puede deberse a un archivo hosts de TCP/IP caducado.

Resolución del problema

Para cada host en la sección DESCRIPTION del archivo tnsnames.ora, es posible que deba actualizar el archivo hosts de TCP/IP. La actualización o no de este archivo dependerá de cómo esté configurado TCP/IP en la red. Parte de la red debe convertir el nombre de host remoto especificado en la sección DESCRIPTION del archivo tnsnames.ora en una dirección.

Si la red tiene un servidor de nombres que reconoce el nombre de host, no es necesario actualizar el archivo hosts de TCP/IP. Si la red no tiene un servidor de nombres que reconoce el nombre de host, deberá añadir una entrada en el archivo hosts de TCP/IP para el host remoto.

La ubicación del archivo hosts de TCP/IP depende del sistema operativo que se ejecuta en el servidor federado:

En los servidores federados que ejecutan Linux o UNIX

El archivo hosts está en el directorio `/etc/hosts`.

En los servidores federados que ejecutan Windows

El archivo hosts está en el directorio `x:\winnt\system32\drivers\etc\hosts`.

Resolución de errores de conexión de origen de datos

Una conexión de prueba con el servidor de origen de datos puede devolver un error por varios motivos. Existen acciones que permiten determinar el motivo del error.

Síntoma

Se devuelve un error al intentar establecer conexión con el origen de datos.

Causa

Hay varias causas posibles que explican un problema de conexión.

Resolución del problema

Para solucionar los problemas de errores de conexión con el origen de datos, consulte los siguientes elementos:

- Verifique que el origen de datos está disponible.
- Si procede, asegúrese de que el servidor de origen de datos esté configurado para conexiones entrantes.
- Asegúrese de que los valores de correlación de usuarios para las opciones `REMOTE_AUTHID` y `REMOTE_PASSWORD` son válidos para las conexiones con el origen de datos. Modifique la correlación de usuarios o cree otra correlación de usuarios según sea necesario.
- Si procede, asegúrese de que el software de cliente del servidor federado esté instalado y configurado correctamente para conectarse al origen de datos.
- Para orígenes de datos ODBC, asegúrese de que el controlador ODBC del servidor federado esté instalado y configurado correctamente para establecer conexión con el servidor de origen de datos ODBC. En servidores federados que ejecutan Windows, utilice la herramienta ODBC Data Source Administrator para verificar el controlador. En servidores federados que ejecutan UNIX, consulte la documentación del proveedor del cliente ODBC.
- Verifique que los valores para las variables establecidas en el servidor federado son correctas para el origen de datos. Estas variables incluyen las variables de entorno del sistema, las variables de la archivo `db2dj.ini` y las variables del registro de perfiles de DB2 (`db2set`).
- Compruebe la definición del servidor. Si es necesario, elimine la definición del servidor y créela de nuevo.

Registro de apodos para vistas y tablas de Oracle

Para cada definición de servidor Oracle, debe registrar un apodo para cada tabla o vista a la que desee acceder. Utilice estos apodos en lugar de los nombres de los objetos de origen de datos cuando consulte los servidores Oracle.

Antes de empezar

Actualice las estadísticas en el origen de datos Oracle antes de registrar un apodo. La base de datos federada se basa en las estadísticas de catálogo de origen de datos para optimizar el proceso de las consultas. Utilice el mandato de origen de datos que es equivalente al mandato RUNSTATS de DB2 para actualizar las estadísticas de origen de datos.

Para los objetos de origen de datos que utilizan la seguridad de etiquetas de Oracle, los datos de apodo no se pueden almacenar en la memoria caché. Puede activar o desactivar la memoria caché utilizando la sentencia ALTER NICKNAME.

Restricción: La creación de un apodo para un sinónimo Oracle de un sinónimo no está soportada y generará el mensaje de error SQL0204.

Procedimiento

Para registrar un apodo para una tabla o vista de Oracle, utilice uno de los métodos siguientes:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Emita la sentencia CREATE NICKNAME.	Por ejemplo: CREATE NICKNAME apodo FOR nombre_definición_servidor."esquema_remoto". "tabla.remota" ;

Cuando se crea el apodo, el servidor federado consulta el catálogo de orígenes de datos utilizando el apodo. Esta consulta prueba la conexión con la tabla o la vista del origen de datos. Si la conexión no funciona, recibirá un mensaje de error. Repita este paso para cada tabla o vista de Oracle para la que desee crear un apodo.

Sentencia CREATE NICKNAME - Ejemplos para el derivador de Oracle

Utilice la sentencia CREATE NICKNAME para registrar un apodo para una tabla o vista de Oracle a la que desea acceder. Este tema proporciona un ejemplo completo con los parámetros necesarios.

El ejemplo siguiente muestra cómo registrar un apodo para una tabla o una vista de Oracle utilizando la sentencia CREATE NICKNAME.

```
CREATE NICKNAME
PARISINV FOR servidor_ora."vinnie"."inventory" ;
```

PARISINV

Apodo exclusivo que se utiliza para identificar la tabla o la vista de Oracle. El apodo está formado por el esquema y el apodo. Si omite el esquema al registrar el apodo, el esquema del apodo será el ID de autorización del usuario que registra el apodo.

servidor_ora."vinnie"."inventory"

Es un identificador de tres partes para el objeto remoto:

- *servidor_ora* es el nombre de definición de servidor que ha asignado al servidor de bases de datos Oracle en la sentencia CREATE SERVER.

- *vinnie* es el ID de usuario del propietario al que pertenece la tabla o la vista.
- *inventory* es el nombre de la tabla o la vista remota a la que desea acceder.

El servidor federado cambia los nombres de los esquemas y las tablas de Oracle a mayúsculas, a menos que los escriba entre comillas.

Configuración del acceso a los scripts como orígenes de datos

Para configurar un sistema federado para acceder a los scripts como orígenes de datos, debe registrar funciones personalizadas, un derivador, una definición de servidor y apodos para los scripts.

Antes de empezar

- Federation debe estar instalado en el servidor que actúa como servidor federado.
- Debe existir una base de datos federada en el servidor federado.

Acerca de esta tarea

Para configurar un servidor federado para acceder a los datos mediante scripts, emita sentencias SQL en la línea de mandatos de DB2.

Procedimiento

Para añadir scripts como orígenes de datos a un servidor federado:

1. Identifique o escriba un script.
2. Registre la función personalizada.
3. Configure el daemon del script.
4. Inicie el daemon del script.
5. Registre el derivador de scripts.
6. Registre la definición de servidor para un origen de datos al que accede un script.
7. Registre los apodos de los orígenes de datos.

Visión general del derivador de scripts

Puede utilizar scripts, por ejemplo, scripts Perl, para acceder a información en las bases de datos o para generar datos. Puede integrar los datos con datos de otros orígenes de datos federados utilizando el derivador de scripts.

Puede tener scripts existentes que devuelven datos de orígenes de datos como, por ejemplo, bancos de datos de biología, o que generan datos por su cuenta. El derivador de scripts permite utilizar scripts como si fueran orígenes de datos federados. El derivador de scripts permite el acceso a datos que, a continuación, pueden integrarse utilizando un sistema federado. Los scripts deben devolver resultados en XML.

Los apodos del derivador de scripts pueden incluir columnas de entrada y salida. Estos apodos utilizan plantillas de funciones en predicados para pasar valores de entrada al script. Los datos de salida del script se representan como XML con un formato jerárquico que, a continuación, puede correlacionarse con los apodos utilizando claves primarias y foráneas.

El derivador de scripts es un derivador de sólo lectura. El derivador de scripts no puede grabar datos en un origen de datos.

En el siguiente diagrama, los datos fluyen desde un script a través del derivador de scripts y el daemon de scripts a la base de datos federada, donde los datos pueden integrarse con datos de otros orígenes y verse con el cliente federado. De manera opcional, el sistema federado puede acceder al script mediante un servidor proxy y un cortafuegos.

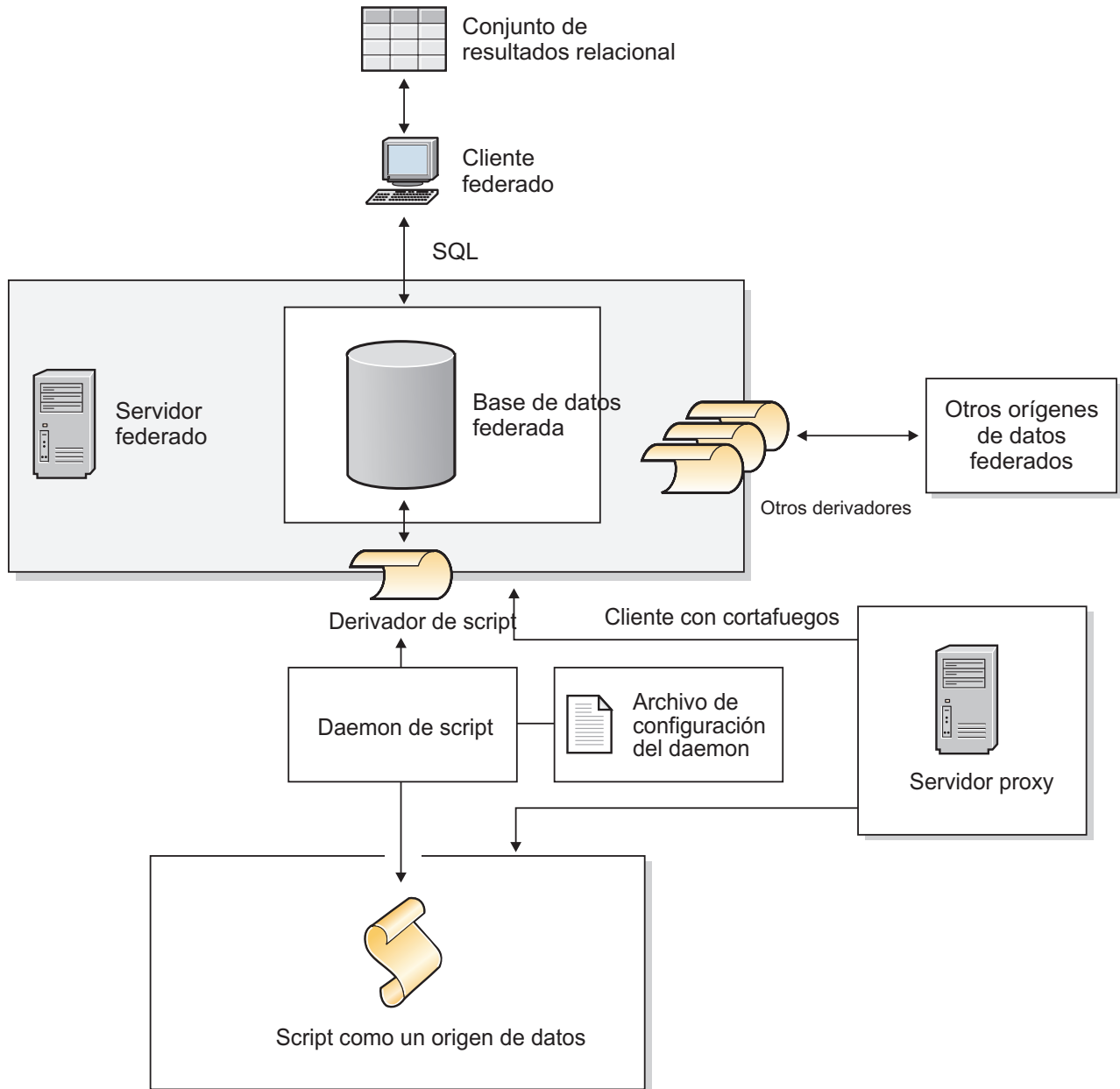


Figura 7. El derivador de scripts en un sistema federado

Se invoca un script desde el directorio que contiene el daemon de scripts. Si el script recupera datos de su propio archivo de datos y no puede ubicar el archivo de datos, puede que el script incluya vías de acceso relativas. Utilice vías de acceso absolutas en los scripts.

Adición de scripts como orígenes de datos a un sistema federado

Para configurar un servidor federado para acceder a los scripts como orígenes de datos, debe configurar el daemon de scripts, un derivador, una definición de servidor y apodos para los scripts.

Registro de la función personalizada del script

Debe registrar la función personalizada del script WSSCRIPT.ARGS antes de registrar el derivador de scripts.

Acerca de esta tarea

Debe registrar la función personalizada en cada instancia de base de datos federada donde esté instalado el derivador de scripts.

La función personalizada del derivador de scripts debe registrarse con el nombre de esquema WSSCRIPT.

Debe incluir palabras clave específicas cuando registre la función personalizada del derivador de scripts. Incluya las palabras clave AS TEMPLATE, DETERMINISTIC y NO EXTERNAL ACTION en la sentencia CREATE FUNCTION.

El entorno federado utiliza dos motores de consulta. Para el derivador de scripts, estos motores de consulta son el motor de consulta de la base de datos federada y el motor de consulta del derivador de scripts. Puede especificar que los predicados se envíen al motor del derivador de scripts utilizando las funciones personalizadas del derivador de scripts en la cláusula WHERE de la sentencia SELECT.

El archivo create_function_mappings.ddl en el directorio sqllib/samples/lifesci/script en el servidor federado especifica los tipos de datos de la función personalizada.

Procedimiento

Para registrar la función personalizada del script:

Ejecute el archivo create_function_mappings.ddl en cada instancia de base de datos federada donde esté instalado el derivador de scripts.

El siguiente ejemplo muestra la sintaxis de la función WSSCRIPT.ARGS:

```
CREATE
FUNCTION WSSCRIPT.ARGS (tipo_datos_columna_entrada()),
tipo_datos_columna_entrada())
  RETURNS INTEGER AS TEMPLATE
  DETERMINISTIC NO EXTERNAL ACTION;
```

Tipos de datos para la función personalizada del derivador de scripts:

Debe especificar el tipo de datos de la columna de entrada dos veces en cada función personalizada.

Registre una función personalizada WSSCRIPT aparte para cada tipo de datos válido del argumento column_name. La función personalizada WSSCRIPT tiene los siguientes tipos de datos:

- VARCHAR

- INTEGER
- CLOB
- DOUBLE
- DATE

Ambos parámetros en la función personalizada deben tener el mismo tipo de datos, que es el tipo de datos de la columna de entrada correspondiente. Cuando se utiliza en un predicado de consulta, el primer parámetro es el nombre de la columna de entrada de conmutador. La segunda columna es el valor que se pasa al script para este conmutador.

Configuración del daemon de scripts

El derivador de scripts requiere un daemon de scripts que escuche las solicitudes de trabajo de script del derivador de scripts. El daemon de scripts debe estar configurado antes de registrar el derivador de scripts.

Antes de empezar

El daemon de scripts tiene los siguientes requisitos previos:

- Tiene acceso de grabación a un directorio en el que el daemon puede grabar archivos temporales.
- Se ejecuta en un servidor al que puede acceder mediante TCP/IP desde el sistema federado. Este servidor puede ser el mismo servidor que opera como servidor federado o un servidor de scripts aparte.
- Requiere un archivo de configuración que debe estar en el mismo servidor que el daemon de scripts.
-
- Se ejecuta aparte del derivador de scripts y la base de datos federada.

Procedimiento

Para configurar el daemon de scripts:

1. Asegúrese de que los archivos ejecutables del daemon de scripts estén en el servidor correcto. Es posible que deba copiar los archivos ejecutables del daemon de scripts en otro servidor.

Durante la instalación de IBM InfoSphere Federation Server, los archivos ejecutables del daemon de scripts se instalan en el servidor federado. El nombre y la ubicación del archivo es la siguiente:

UNIX db2script_daemon se instala en el directorio \$DB2PATH/bin. \$DB2PATH es el directorio donde está instalado el servidor federado.

Windows

db2script_daemon.exe se instala en el directorio %DB2PATH%\bin. %DB2PATH% es el directorio donde está instalado el servidor federado, normalmente C:\SQLLIB\bin.

Si utiliza un servidor de scripts aparte, copie los archivos ejecutables y de configuración del daemon de scripts del servidor federado en el servidor de scripts. Los archivos ejecutables del daemon de scripts pueden ejecutarse en cualquier directorio del servidor de scripts que no contenga espacios en los nombres de la vía de acceso de directorios.

2. Asegúrese de que el archivo de configuración del daemon de scripts esté en el servidor correcto.

Durante la instalación del sistema federado, se instala un archivo de configuración de ejemplo del daemon de scripts en el servidor federado. El nombre del archivo de configuración de ejemplo es `SCRIPT_DAEMON.config`. La ubicación del archivo es la siguiente:

UNIX El archivo de configuración del daemon se instala en el directorio `$DB2PATH/bin`.

Windows

El archivo de configuración del daemon se instala en el directorio `%DB2PATH%\bin`.

De forma predeterminada, el daemon busca el archivo de configuración en el directorio de trabajo desde el que se inicia el daemon. Puede copiar el archivo de configuración en otra ubicación. Si utiliza un servidor de scripts, copie el archivo de configuración del daemon del directorio en el servidor federado en un directorio en el servidor de scripts. Puede copiar el archivo de configuración del daemon en cualquier directorio del servidor de scripts al que tenga acceso el daemon.

3. Edite el archivo de configuración de ejemplo del daemon de scripts.
 - a. Renombre el archivo de configuración para que pueda utilizar otra vez el archivo de ejemplo.
 - b. Asegúrese de que la primera línea del archivo de configuración sea un signo igual (=). Si falta el signo igual, el daemon no se inicia. Un mensaje de error indicará que no se ha especificado `DAEMON_PORT`.
 - c. Asegúrese de que la última línea del archivo de configuración termine con una nueva línea.

El archivo de configuración de ejemplo que se proporciona con el sistema federado termina con un carácter de nueva línea. Si la última línea no termina con un carácter de nueva línea, recibirá un mensaje de error cuando intente ejecutar la primera consulta de script que utiliza el origen de datos que aparece en la última línea.

- d. Asegúrese de que no existan espacios adicionales después de las vías de acceso de directorios o al final del archivo de configuración.
- e. Especifique las siguientes opciones en el archivo de configuración. Para las opciones que requieran vías de acceso, puede especificar vías de acceso relativas. Las vías de acceso relativas son relativas al directorio desde donde se inicia el proceso del daemon.

DAEMON_PORT=*número_puerto*

Puerto de red donde escucha el daemon las solicitudes de trabajo de script sometidas por el derivador. El valor predeterminado es 4099.

MAX_PENDING_REQUESTS=*número_de_solicitudes*

Número máximo de solicitudes de trabajo de script que pueden bloquearse en el daemon en un momento dado. Este número no representa el número de trabajos de script que se ejecutan simultáneamente, sólo el número de solicitudes de trabajo que pueden bloquearse a la vez. Establezca este valor en un número mayor que cinco. El daemon de scripts no restringe el número de trabajos de script que pueden ejecutarse simultáneamente.

DAEMON_LOGFILE_DIR=*dir*

Directorio donde el daemon crea el archivo de registro. Este archivo contiene información de estado y error generada por el daemon de scripts.

SCRIPT_OUT_DIR_PATH=*vía acceso*

Directorio donde el daemon crea el archivo temporal para almacenar los datos de salida de script. El daemon lee los datos de este archivo y los pasa al derivador a través de la conexión de red. Una vez pasados los datos al derivador, el derivador limpia el archivo temporal.

script specification entry=*entrada*

Lista de entradas que especifica el nombre y la ubicación de los scripts que puede invocar el derivador de scripts. La entrada tiene el formato:

nombre_script=vía acceso_completa_script

Son aplicables los siguientes ejemplos al sistema operativo designado:

UNIX Por ejemplo, para especificar un script que tenga acceso a un origen de datos Oracle, añada la siguiente línea al archivo de configuración del daemon:

```
oracle=/dsk/1/data/oracle
```

Windows

Por ejemplo, para especificar un script que tenga acceso a un origen de datos Oracle, añada la siguiente línea al archivo de configuración del daemon:

```
oracle=c:\data\oracle.a
```

El siguiente ejemplo muestra un archivo SCRIPT_DAEMON.cfg para cuatro 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
```

El archivo de configuración de ejemplo del daemon de scripts proporciona un ejemplo de cómo configurar el daemon de scripts.

Inicio del daemon de scripts

Antes de acceder a los orígenes de datos con scripts, debe iniciar el daemon de scripts.

Antes de empezar

Debe tener acceso de grabación en todas las vías de acceso que aparecen en el archivo de configuración del daemon para las opciones DAEMON_LOGFILE_DIR y SCRIPT_OUT_DIR_PATH.

Acerca de esta tarea

El archivo ejecutable del daemon de scripts inicia un nuevo proceso en el que se ejecuta el daemon de scripts.

Procedimiento

Para iniciar el daemon de scripts:

1. Abra el directorio donde se encuentra el archivo ejecutable del daemon.
2. Emita el mandato `db2script_daemon` para ejecutar los archivos ejecutables, incluidas las opciones correspondientes.

El siguiente ejemplo incluye las opciones:

```
db2script_daemon -a
acción -c archivo_config -d
nivel_depur -u ID_usuario -p
contraseña
```

Mandato `db2script_daemon` - opciones y ejemplos:

El mandato `db2script_daemon` inicia el daemon de scripts. Puede iniciar el daemon de scripts con varias opciones.

El mandato `db2script_daemon` puede utilizarse en servidores UNIX o Windows. Algunas de las opciones incluidas en la sintaxis sólo pueden utilizarse en servidores Windows.

Las opciones especificadas con la acción de inicio sólo afectan a la instancia actual del daemon y alteran temporalmente los valores especificados con la acción de instalación.

Opciones: mandato `db2script_daemon`

El mandato `db2script_daemon` tiene las siguientes opciones:

-a acción (sólo para Windows)

Ejecuta la actividad especificada. Las acciones válidas son estado, instalar, iniciar, detener y eliminar.

-c archivo_config

Indica al servicio del daemon que utilice el archivo de configuración especificado. Si no especifica el archivo de configuración, el daemon busca el archivo `SCRIPT_DAEMON.config` en el directorio donde están instalados los archivos ejecutables del daemon. Puede utilizar esta opción con las acciones de instalar e iniciar.

-d [1 | 2 | 3]

Establece el nivel de depuración del servicio del daemon en el valor especificado. Un valor 1 activa el registro cronológico, un valor 2 realiza un rastreo de todos los mandatos y un valor 3 activa el registro cronológico, realiza un rastreo de todos los mandatos y guarda todos los archivos temporales para capturar la salida XML. Puede utilizar esta opción con las acciones de instalar e iniciar.

-u ID_usuario (sólo para Windows)

Establece el servicio del daemon para ejecutarse con el ID de usuario especificado. Puede utilizar esta opción con la acción de instalar.

-p contraseña (sólo para Windows)

Especifica la contraseña del ID de usuario especificado. La contraseña sólo es válida y necesaria cuando especifica la opción `-u`. Si no se especifica la opción `-p` al establecer la opción `-u`, el programa solicita la contraseña. Puede utilizar esta opción con la acción de instalar.

Ejemplos: mandato db2script_daemon

Los siguientes ejemplos muestran cómo utilizar las opciones del daemon de scripts.

Iniciar el daemon

Para iniciar el daemon en UNIX, emita el siguiente mandato:

```
db2script_daemon
```

Este mandato supone que el archivo de configuración del daemon está en el mismo directorio que el archivo ejecutable.

Para instalar e iniciar el daemon en Windows, emita los siguientes mandatos:

```
db2script_daemon -a install  
db2script_daemon -a start
```

Especificar el archivo de configuración del daemon

Si ha cambiado el nombre del archivo de configuración del daemon o si el archivo de configuración no está en el mismo directorio que el archivo ejecutable del daemon, debe utilizar la opción `-c` cuando ejecute el archivo ejecutable. Esta opción especifica el nombre y la vía de acceso de directorios del archivo de configuración del daemon.

En este ejemplo, la información de configuración del daemon está en un archivo denominado `SCRIPT_D.config` en el subdirectorio `cfg` en un servidor UNIX. Emita el siguiente mandato:

```
db2script_daemon -c cfg/SCRIPT_D.config
```

Especificar el nivel de depuración

Si desea iniciar el daemon con la depuración activada y un nivel de depuración 2, emita los siguientes mandatos:

```
db2blast_daemon -a install -d 2  
db2blast_daemon -a start
```

Comprobar el estado del daemon (Windows)

Para comprobar el estado del daemon en un servidor Windows, emita el siguiente mandato:

```
db2blast_daemon -a status
```

Este mandato supone que el archivo de configuración del daemon está en el mismo directorio que el archivo ejecutable.

Detener el daemon

Para detener el daemon en UNIX, liste el ID de proceso del daemon emitiendo el siguiente mandato:

```
ps -ef | grep db2script
```

A continuación, utilice el ID de proceso para detener el daemon emitiendo el siguiente mandato:

```
kill ID_proceso
```

Este mandato supone que el archivo de configuración del daemon está en el mismo directorio que el archivo ejecutable.

Para detener el daemon en Windows, emita el siguiente mandato:

```
db2script_daemon -a stop
```

Eliminar el daemon

Puede eliminar el daemon de scripts cuando ya no desee utilizar el derivador de scripts.

Para eliminar el daemon, emita el siguiente mandato:

```
db2script_daemon -a remove
```

Registro del derivador de scripts

Debe registrar el derivador de scripts para acceder a los orígenes de datos con scripts. El derivador de scripts se implementa como un archivo de biblioteca.

Procedimiento

Para registrar el derivador de scripts:

Emita la sentencia CREATE WRAPPER, especificando un nombre para el derivador de scripts y el nombre del archivo de biblioteca de derivador.

Por ejemplo, para registrar un derivador con el nombre script_wrapper en un servidor federado que utiliza AIX, emita la siguiente sentencia:

```
CREATE WRAPPER script_wrapper LIBRARY 'libdb2lsscript.a';
```

El nombre del archivo de biblioteca de derivador que especifica depende del sistema operativo del servidor federado.

No hay opciones específicas del derivador de scripts para la sentencia CREATE WRAPPER del derivador de scripts. El derivador se ejecuta sin delimitar de forma predeterminada.

Archivo de biblioteca de derivador de scripts:

Para registrar el derivador de scripts, especifique el archivo de biblioteca de derivador de scripts del sistema operativo del servidor federado.

Cuando instala la federación, se añade el archivo de biblioteca de derivador de scripts a la vía de acceso de directorios predeterminada.

Las vías de acceso de directorios predeterminadas y los nombres de archivo de biblioteca del derivador predeterminados se especifican en la tabla siguiente.

Tabla 35. Nombres de archivo y ubicaciones de la biblioteca de derivador de scripts

Sistema operativo	Vía de acceso de directorios	Nombre de archivo de biblioteca de derivador
AIX	/usr/opt/vía_acceso_instalación/lib	libdb2lsscript.a
Linux	/opt/IBM/db2/ vía_acceso_instalación/lib	libdb2lsscript.so
Solaris	/opt/IBM/db2/ vía_acceso_instalación/lib	libdb2lsscript.so
Windows	%DB2PATH%\bin	db2lsscript.dll

- *vía_acceso_instalación* es la vía de acceso de directorios donde se instala la federación en UNIX o Linux.
- %DB2PATH% es la variable de entorno que se utiliza para especificar la vía de acceso de directorios donde se instala la federación en Windows. La vía de acceso de directorios predeterminada de Windows es C:\Archivos de programa\IBM\SQLLIB.

Registro de la definición de servidor para un script como un origen de datos (línea de mandatos de DB2)

Debe registrar cada servidor al que desee acceder en la base de datos federada.

Procedimiento

Para registrar una definición de servidor para un script:

Emita la sentencia CREATE SERVER.

Por ejemplo:

```
CREATE SERVER script_server WRAPPER script_wrapper
  OPTIONS (NODE 'myserver.example.com', DAEMON_PORT '4099');
```

Las opciones de servidor NODE y DAEMON_PORT son necesarias para los scripts como orígenes de datos.

Después de registrar la definición de servidor, utilice la sentencia ALTER SERVER para añadir o eliminar opciones de servidor.

Sentencia CREATE SERVER - Ejemplos para el derivador de scripts:

Los ejemplos muestran el uso de las opciones necesarias y las opciones adicionales de servidor.

Ejemplo de las opciones necesarias

El ejemplo siguiente muestra cómo registrar una definición de servidor para un derivador de scripts emitiendo la sentencia CREATE SERVER:

```
CREATE SERVER servidor1_scriptn
  WRAPPER derivador_scripts
  OPTIONS(NODE 'big_rs.empresa.com');
```

servidor1_scriptn

Nombre que asigna al servidor de scripts. No se permiten los nombres de definición de servidor duplicados.

WRAPPER *derivador_scripts*

Nombre del derivador.

NODE '*big_rs.empresa.com*'

Nombre de host del sistema en el que se ejecuta el proceso del daemon de script. Este valor es sensible a las mayúsculas y minúsculas.

Aunque el nombre del nodo se especifica como una opción en la sentencia CREATE SERVER, se necesita para el derivador de scripts.

Ejemplo de opción adicional

El ejemplo siguiente muestra una opción de servidor adicional que puede especificar cuando registra una definición de servidor para un derivador de scripts:

```
CREATE SERVER
servidor1_scriptn WRAPPER derivador_scripts
  OPTIONS(NODE 'big_rs.empresa.com', DAEMON_PORT '4088');
```

DAEMON_PORT '*4088*'

Especifica el número del puerto en el que el daemon escucha las solicitudes de trabajo de scripts. El número de puerto debe ser el mismo que ha especificado en la opción DAEMON_PORT del archivo de configuración del daemon. El número de puerto predeterminado es 4099.

Registro de apodos para los scripts (línea de mandatos de DB2)

Debe registrar un apodo aparte para cada script. Utilice estos apodos cuando consulte el origen de datos al que accede un script.

Acerca de esta tarea

El derivador de scripts asocia datos XML con los apodos. Los apodos padre e hijo se corresponden con los elementos raíz y anidados de un documento XML. Los apodos padre e hijo se conectan mediante claves primarias y foráneas que se especifican en la sentencia CREATE NICKNAME. Cada apodo está definido por expresiones XPath que identifican los elementos de datos XML y especifican cómo extraer valores de columna de cada elemento.

El origen de datos se especifica mediante la sentencia CREATE NICKNAME y se asocia con un nombre de script con la opción de apodo DATASOURCE. Debe crear una columna para cada argumento de entrada que se va a pasar al script. Utilice las opciones de columna de entrada para controlar la sintaxis de los conmutadores en la línea de mandatos. El valor de cada conmutador debe incluirse en el predicado de la consulta utilizando la función personalizada ARGS en el tiempo de ejecución.

Los scripts sencillos que no requieren argumentos de línea de mandato no necesitan columnas de entrada.

Procedimiento

Para registrar un apodo para un script:

Emita la sentencia CREATE NICKNAME. Los apodos pueden tener una longitud de hasta 128 caracteres.

Por ejemplo:

```
CREATE NICKNAME apodo
(
  nombre_columna tipo_datos OPTIONS
  ('opciones_columna_apodo'),
  nombre_columna tipo_datos OPTIONS
  ('opciones_columna_apodo'),
  nombre_columna tipo_datos OPTIONS
  ('opciones_columna_apodo')
)
FOR SERVER nombre_definición_servidor
OPTIONS (opciones_apodo);
```

Emita la sentencia para cada script para el que desee crear un apodo.

Sentencia CREATE NICKNAME - Ejemplos para el derivador de scripts

El ejemplo muestra cómo utilizar la sentencia CREATE NICKNAME para registrar apodos para el derivador de scripts.

El siguiente ejemplo crea un apodo padre para los datos XML que se devuelven de un script denominado 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 script_server
OPTIONS(DATASOURCE 'fee',
        XPATH '/doc/customer', STREAMING 'YES');

```

El siguiente ejemplo crea un apodo denominado orders. El apodo orders es hijo del apodo denominado customers, que se ha creado en el ejemplo anterior:

```

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 script_server
OPTIONS( XPATH './order');

```

Opciones de apodo de derivador de scripts

Puede especificar opciones cuando crea un apodo para un script. Sólo el apodo raíz puede incluir columnas de entrada.

Opciones de apodo

La lista siguiente describe las opciones de apodo:

DATASOURCE

Nombre del script que se invocará. El script debe estar incluido en el archivo de configuración del daemon de scripts. Esta opción es necesaria para los orígenes de datos en el apodo padre. Esta opción sólo se aplica al apodo raíz.

NAMESPACES

Lista separada por comas de pares nombre=valor que el derivador utiliza para resolver prefijos de espacios de nombres en la expresión XPath del apodo.

TIMEOUT

Tiempo máximo, en minutos, que espera el derivador de scripts los resultados del daemon de scripts. El valor predeterminado es 60 minutos. Esta opción sólo se aplica al apodo raíz.

VALIDATE

Especifica si el documento de origen XML debe validarse antes de extraer los datos XML. Si esta opción se establece en YES, el sistema de base de datos DB2 comprueba que la estructura del documento de origen cumple un esquema XML o una definición de tipo de documento (DTD). Esta opción sólo se acepta para las columnas del apodo raíz (el apodo que identifica los elementos en el nivel superior del documento XML). El valor predeterminado es NO.

El documento de origen XML no se valida si el derivador de scripts no puede ubicar el archivo de esquema XML o el archivo DTD (.xsd o .dtd). El sistema de base de datos DB2 no emite un mensaje de error si no se realiza la validación. Asegúrese de que el archivo de esquema XML o el archivo DTD esté en la ubicación que se especifica en el documento de origen XML. No establezca el parámetro VALIDATE en YES si establece el parámetro STREAMING en YES.

STREAMING

Especifica si el documento de origen XML está separado en fragmentos lógicos que se corresponden con el nodo que coincide con la expresión XPath del

apodo. A continuación, el derivador de scripts procesa el origen de datos XML fragmento a fragmento, lo que reduce el uso de memoria total. Esta opción sólo se acepta para las columnas del apodo raíz (el apodo que identifica los elementos en el nivel superior del documento XML). El valor de predeterminado de modalidad continua es NO. No establezca el parámetro STREAMING en YES si establece el parámetro VALIDATE en YES.

XPATH

Expresión XPath que identifica el elemento XML que representa las tuplas individuales en el origen de datos. El derivador evalúa la opción de apodo XPATH para un apodo hijo en el contexto de la vía de acceso que especifica la opción de apodo XPATH del apodo padre. Esta expresión XPath se utiliza como contexto para evaluar los valores de columna identificados por las opciones de columna de apodo XPATH.

Opciones de columna de apodo

Las opciones de columna de apodo se describen a continuación:

DEFAULT

Valor predeterminado de una columna de entrada. Esta opción sólo se aplica a las columnas de entrada.

El valor predeterminado se utiliza si la consulta SQL no proporciona ningún valor. Esta opción no es necesaria.

FOREIGN_KEY

Indica que el apodo es un apodo hijo y especifica el apodo padre correspondiente.

Un apodo puede tener como máximo una opción de columna FOREIGN_KEY. El valor de esta opción es sensible a las mayúsculas y minúsculas. La columna designada por la opción FOREIGN_KEY contiene una clave generada por el derivador. El valor de la columna no puede recuperarse en una consulta SELECT y la opción XPATH no puede especificarse. La columna sólo puede utilizarse para unir apodos padre y apodos hijo. Una sentencia CREATE NICKNAME con una opción FOREIGN_KEY falla si el apodo padre tiene un nombre de esquema diferente. A menos que el apodo referido en una cláusula FOREIGN_KEY se defina explícitamente en minúsculas, o en minúsculas y mayúsculas, entre comillas en la sentencia CREATE NICKNAME correspondiente, deberá especificar el apodo en mayúsculas cuando haga referencia a él en la cláusula FOREIGN_KEY.

Las columnas de clave foránea deben designarse como FOR BIT DATA y NOT NULL.

INPUT_MODE

Especifica la modalidad de entrada de una columna. Los valores válidos son CONFIG o FILE_INPUT. El derivador pasa el valor especificado al daemon de scripts.

CONFIG

El valor se trata como un parámetro configurable.

FILE_INPUT

Se crea un archivo que almacena el valor y el nombre de archivo se pasa como el argumento de la línea de mandatos.

POSITION

Valor entero para los parámetros posicionales. El orden de secuencia de posición empieza en 1. Esta opción sólo se aplica a las columnas de entrada.

Si el valor posicional se establece en un entero, esta entrada debe estar en esta posición en la línea de mandatos. Si se establece esta opción, el conmutador se inserta en la ubicación correspondiente cuando se ejecuta la consulta. Si POSITION se establece en -1, la opción se añade como la última opción de línea de mandatos. Por ejemplo, si un valor de columna debe ir al final de la línea de mandatos y no hay ninguna opción SWITCH, el establecimiento del valor de POSITION en -1 añade el valor al final de la línea de mandatos. Los valores enteros de POSITION no pueden duplicarse en un apodo. Esta opción no es necesaria.

PRIMARY_KEY

Indica que el apodo es un apodo padre. El tipo de datos de columna debe ser VARCHAR(16). Un apodo puede tener como máximo una opción de columna PRIMARY_KEY. El único valor válido es Y.

La columna designada como PRIMARY_KEY contiene una clave generada por el derivador. El valor de la columna no puede recuperarse en una consulta SELECT y la opción XPATH no puede especificarse. La columna sólo puede utilizarse para unir apodos padre y apodos hijo. Las columnas de clave primaria deben designarse como FOR BIT DATA y NOT NULL.

SWITCH

Serie de caracteres que especifica un parámetro para el script en la línea de mandatos. Esta opción sólo se aplica a las columnas de entrada.

En la línea de mandatos, el valor de esta opción precede el valor de columna proporcionado por WSSCRIPT.ARGS o el valor predeterminado, si existe. Si el valor del conmutador es una serie vacía y existe un valor predeterminado para la columna, el valor predeterminado se añade sin ninguna información de SWITCH cuando se genera la línea de mandatos. Si no existe ningún valor predeterminado y no se proporciona un valor para la columna en la consulta SQL, esta columna de entrada se ignora cuando se genera la línea de mandatos. Esta opción es necesaria para una columna de entrada.

SWITCH_ONLY

Permite el uso de conmutadores sin un argumento de línea de mandatos.

Si se especifica la opción SWITCH_ONLY con un valor Y, los valores de entrada válidos son Y o N. Para un valor de entrada Y, sólo se añade el conmutador a la línea de mandatos. Para un valor de entrada N, no se añade ningún valor a la línea de mandatos.

VALID_VALUES

Conjunto separado por puntos y comas de valores válidos para una columna.

XPATH

Especifica la expresión XPath en el documento XML que contiene los datos correspondientes a esta columna. El derivador de scripts evalúa la expresión XPath después de que la sentencia CREATE NICKNAME aplique esta expresión XPath desde la opción de apodo XPATH. Si ejecuta una consulta en un nombre de columna que tiene una referencia de código XPATH configurada incorrectamente como, por ejemplo, unas mayúsculas o minúsculas incorrectas, la consulta devuelve valores nulos en esta columna para todas las filas devueltas.

Consultas SQL con el derivador de scripts

Las consultas SQL que se realizan mediante el derivador de scripts utilizan la función personalizada para transportar los valores de entrada de los parámetros del script.

Una sentencia SELECT que pasa valores de parámetros a un script mediante el derivador de scripts debe contener al menos un predicado con una función personalizada para obtener los valores de entrada de los parámetros del script.

Apodos raíz

Por ejemplo, la siguiente sentencia crea un apodo raíz para un script denominado 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 script_server  
  OPTIONS(DATASOURCE 'myscript', XPATH 'doc/customer',  
    TIMEOUT '300', VALIDATE 'YES');
```

Las sentencias de la función personalizada son las siguientes:

```
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;
```

El archivo de configuración especifica los siguientes parámetros de configuración:

```
SCRIPT_OUT_DIR_PATH=C:\temp  
myscript=C:\perl\bin\perl myscript.pl -model
```

Para ejecutar una consulta y enviar el contenido de la tabla t1.bigdata al daemon y al archivo local C:\temp\f12345, emita la siguiente consulta y el mandato wsscript.args:

```
SELECT id, name FROM customers, t1  
  WHERE wsscript.args (customers.argfile, t1.bigdata) = 1
```

La consulta anterior da como resultado la siguientes línea de mandatos:

```
C:\perl\bin\perl myscript.pl -model -argle 1.0 -file C:\temp\f12345
```

Para ejecutar una consulta que utilice los valores predeterminados de todos los parámetros, ejecute una consulta en el apodo sin predicados. Para evitar problemas con el exceso de salida, incluya la opción STREAMING en el apodo.

Apodos hijo

El derivador de scripts correlaciona el conjunto de resultados XML del script con apodos que tienen una relación padre-hijo. Para recuperar datos de un apodo hijo,

una el apodo hijo con el apodo padre hasta la raíz. Las sentencias SELECT que hacen referencia a un apodo hijo deben unirse con el apodo padre del apodo hijo utilizando columnas de clave primaria y foránea.

La siguiente consulta muestra los nombres de clientes y las cantidades de cada pedido de cada cliente:

```
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
```

Especifique la sentencia c.cid=o.cid de unión para indicar la relación padre-hijo entre el apodo customers y el apodo orders. Si une un apodo hijo con sigo mismo, se devuelve un mensaje de error.

Optimización del rendimiento del derivador de scripts

La ubicación del daemon de scripts puede afectar al rendimiento de las consultas.

Para mejorar el rendimiento de la comunicación de red, utilice un servidor de scripts aparte para el daemon de scripts. Coloque el servidor federado y el servidor de scripts en servidores distintos. Asimismo, coloque el daemon de scripts en el servidor de scripts.

Configuración del acceso a orígenes de datos Sybase

Para configurar un servidor federado para acceder a orígenes de datos Sybase, debe proporcionar al servidor federado información acerca de los orígenes de datos y los objetos a los que desea acceder.

Antes de empezar

- El software Sybase Client SDK debe estar instalado en el servidor que actuará como servidor federado. Cuando instala el cliente Sybase en Windows, debe especificar la opción **Completa** o **Personalizada**. Si especifica la opción personalizada, debe especificar la opción **Biblioteca de interfaz XA del Administrador de transacciones distribuidas ASE**.
- IBM InfoSphere Federation Server debe estar instalado en el servidor que actúa como servidor federado.
- Compruebe la configuración del servidor federado.
- Compruebe el parámetro federado para asegurarse de que la federación está habilitada.

Puede configurar un servidor federado para acceder a los datos almacenados en los orígenes de datos Sybase utilizando el Centro de control de DB2 o emitiendo sentencias SQL en la línea de mandatos de DB2. El Centro de control de DB2 incluye un asistente que sirve de guía en los pasos necesarios para configurar los objetos federados que se necesitan.

Procedimiento

Para configurar el acceso a los orígenes de datos Sybase:

1. Establezca las variables de entorno de Sybase.
2. Configure y pruebe el archivo de configuración del cliente Sybase utilizando uno de los métodos siguientes, dependiendo del sistema operativo:

- Configure y pruebe el archivo de configuración del cliente Sybase (Linux, UNIX).
 - Configure y pruebe el archivo de configuración del cliente (Windows).
3. Registre el derivador.
 4. Registre la definición de servidor.
 5. Cree las correlaciones de usuarios.
 6. Pruebe la conexión con el servidor Sybase.
 7. Registre los apodos de las vistas y tablas de Sybase

Soporte de derivadores de Sybase para Adaptive Server Enterprise (ASE)

El derivador de Sybase da soporte a Sybase Adaptive Server Enterprise (ASE) 15.0, además de a ASE 12.5 y ASE 12.0.

Clientes soportados para Sybase ASE 15.0

Puede conectarse a ASE 15.0 utilizando el derivador de Sybase con el cliente Sybase, versión 12.5.1 o posterior.

Si utiliza Software Developer Kit (SDK), versión 12.5.1 como cliente Sybase, IBM recomienda instalar Electronic Software Distribution (ESD) #12 o posterior para el SDK en el servidor federado asociado con el derivador de Sybase. Si utiliza SDK, versión 15.0 como cliente Sybase, IBM recomienda instalar ESD #3 o posterior. Si no instala ESD, puede obtener un error inesperado cuando utilice el derivador de Sybase.

Actualización de las bibliotecas de Sybase para el cliente Sybase, versión 15.0

UNIX: si utiliza el cliente Sybase, versión 15.0, puede ejecutar el script Insyblibs de Sybase para actualizar los nombres de biblioteca de Sybase y conservar la coherencia entre las versiones soportadas de los archivos de biblioteca de Sybase. El script Insyblibs crea enlaces simbólicos desde los nombres de biblioteca nuevos a los nombres de biblioteca antiguos, lo que permite a las aplicaciones anteriores a la versión 15.0 trabajar con las bibliotecas renombradas.

Windows: si utiliza el cliente Sybase, versión 15.0, puede ejecutar el archivo copylibs.bat para copiar los archivos *.dll necesarios, lo que permite a las aplicaciones anteriores a la versión 15.0 trabajar con las bibliotecas renombradas.

Error del script Insyblibs

El script Insyblibs actual contiene un problema. Cuando ejecuta 'Insyblibs create', se produce el siguiente mensaje de error:

```
"libsyb*.s[o:
No existe ningún archivo o directorio de este tipo"
```

Como método alternativo a este problema, puede suprimir el | (carácter de barra) de los [] (corchetes) en la línea 34 del script. Sybase reconoce este problema. Para obtener más información, consulte Soporte de Sybase.

Tipos de datos no soportados

No puede crear apodos para los objetos de origen de datos que contienen tipos de datos no soportados. El derivador de Sybase no da soporte a los siguientes tipos de datos que se han introducido en ASE, versión 12.5.1:

- DATE
- TIME

El derivador de Sybase no da soporte a los siguientes tipos de datos que se han introducido en ASE, versión 15.0:

- BIGINT
- LONGSYSNAME
- UNITEXT
- UNSIGNED BIGINT
- UNSIGNED INT
- UNSIGNED SMALLINT

Establecimiento de las variables de entorno de Sybase

Las variables de entorno de Sybase deben establecerse en el archivo `db2dj.ini` en el servidor federado.

Restricciones

Revise las restricciones del archivo `db2dj.ini`.

El archivo `db2dj.ini` contiene información de configuración sobre el software de Sybase Open Client que hay instalado en el servidor federado.

Existen variables de entorno necesarias y opcionales para los orígenes de datos Sybase.

Si ha instalado el software de Sybase Open Client antes de instalar el derivador de Sybase, las variables de entorno de Sybase necesarias se establecen en el archivo `db2dj.ini`.

Si no ha instalado el software de Sybase Open Client antes de instalar el derivador de Sybase, o si desea establecer algunas de las variables de entorno opcionales, debe establecer las variables de entorno utilizando los pasos de esta tarea.

Procedimiento

Para establecer las variables de entorno de Sybase:

1. Utilice uno de los métodos siguientes:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Establezca automáticamente las variables de entorno.	Ejecute el asistente de IBM InfoSphere Federation Server. Siga las instrucciones del asistente. Importante: Establezca las variables de entorno necesarias ejecutando el asistente de instalación. Las variables de entorno opcionales deben establecerse manualmente.

Método	Descripción
Establezca manualmente las variables de entorno.	<p>Edite el archivo db2dj.ini:</p> <p>El archivo db2dj.ini se encuentra en el directorio especificado por la variable de registro DB2_DJ_INI de DB2. Si la variable DB2_DJ_INI no se establece, el archivo db2dj.ini reside en una de las siguientes vías de acceso predeterminadas, dependiendo del sistema operativo:</p> <ul style="list-style-type: none"> En Linux y UNIX: <i>padre_instancia</i>/sql1lib/cfg/db2dj.ini. <i>padre_instancia</i> El directorio padre del propietario de la instancia En Windows: %DB2PATH%\cfg\db2dj.ini %DB2PATH% El directorio donde está instalado el sistema de base de datos de DB2, por ejemplo, C:\Archivos de programa\IBM\sql1lib. <p>Si el archivo no existe, puede crear un archivo con el nombre db2dj.ini utilizando un editor de texto.</p> <p>En el archivo db2dj.ini, debe especificar la vía de acceso completa en el valor de la variable de entorno de SYBASE; de lo contrario, aparecerán errores. Por ejemplo:</p> <pre>SYBASE=/opt/sybase SYBASE_OCS=OCS-12_5</pre>

- En Linux y UNIX, actualice el archivo .profile que hay en la instancia de base de datos federada. Especifique información sobre las variables de entorno de Sybase que ha añadido al archivo db2dj.ini. En Windows, el archivo se actualiza cuando instala el software de Sybase Open Client.

Emita los siguientes mandatos para actualizar el archivo .profile:

```
export SYBASE=directorio_padre_sybase
export SYBASE_OCS=OCS-versión_release
export PATH=${SYBASE}/bin:$PATH
```

- En el directorio padre, ejecute el archivo .profile que hay en la instancia de base de datos federada.

Emita el siguiente mandato para ejecutar el archivo .profile:

```
. .profile
```

- En algunos sistemas operativos como, por ejemplo, en Linux, debe añadir la vía de acceso de la biblioteca del cliente Sybase a la variable DB2LIBPATH db2set, por ejemplo:

```
db2set DB2LIBPATH=/opt/sybase125/OCS-12_5/lib
```

- Para asegurarse de que las variables de entorno están establecidas en el servidor federado, recicle la instancia de la base de datos federada con estos mandatos:

```
db2stop
db2start
```

Cuando finalice esta tarea, debe registrar el derivador.

Variables de entorno de Sybase

Existen variables de entorno necesarias y variables de entorno opcionales para orígenes de datos Sybase. Estas variables se definen en el archivo db2dj.ini.

Las variables de entorno válidas para Sybase son:

- SYBASE
- SYBASE_OCS
- SYBASE_CHARSET (opcional)

Descripción de las variables

SYBASE

Especifica la vía de acceso de directorio en la que se ha instalado el software Sybase Open Client. Especifique la vía de acceso completa para esta variable de entorno.

Por ejemplo, si Sybase Open Client, versión 12.5, está instalado en el directorio D:\djxc\lient\sybase\V125, especifique la siguiente variable de entorno SYBASE:

```
SYBASE=D:\djxc\lient\sybase\V125
```

Si Sybase Open Client, versión 12.0, está instalado en el directorio D:\djxc\lient\sybase\V12, especifique la siguiente variable de entorno SYBASE:

```
SYBASE=D:\djxc\lient\sybase\V12
```

SYBASE_OCS

Especifica el directorio, la versión y el release del software Sybase Open Client instalado. No especifique la vía de acceso completa si especifica esta variable de entorno.

```
SYBASE_OCS=OCS-versión_release
```

Por ejemplo, si Sybase Open Client, versión 12.0, está instalado en el directorio D:\djxc\lient\sybase\V12\OCS-12_0, especifique el siguiente valor para la variable de entorno SYBASE_OCS:

```
SYBASE_OCS=OCS-12_0
```

Si Sybase Open Client, versión 12.5, está instalado en el directorio D:\djxc\lient\sybase\V125\OCS-12_5, especifique el siguiente valor para la variable de entorno SYBASE_OCS:

```
SYBASE_OCS=OCS-12_5
```

SYBASE_CHARSET

Especifica el nombre del juego de caracteres que se desea utilizar. Establezca la variable de entorno SYBASE_CHARSET en la página de códigos que especifique en el parámetro **CODESET** del servidor federado. En el directorio \$SYBASE\charsets encontrará una lista de los nombres de juegos de caracteres válidos.

Los juegos de caracteres se especifican de forma distinta entre el parámetro **CODESET** y la variable de entorno SYBASE_CHARSET. Por ejemplo, si establece el parámetro **CODESET** en el formato Unicode Transformation Format de 8 bits, UTF-8, especifique UTF8 en la variable de entorno SYBASE_CHARSET:

```
SYBASE_CHARSET=utf8
```

Si no establece la variable de entorno SYBASE_CHARSET, el derivador utiliza el juego de caracteres de Sybase que coincida con un juego de caracteres especificado en la página de códigos de la base de datos federada. Si no existe ningún juego de caracteres Sybase coincidente, el derivador utiliza el juego de caracteres iso_1.

Instalación y prueba del archivo de configuración del cliente Sybase (Windows)

El archivo de configuración del cliente Sybase se utiliza para establecer conexión con las bases de datos Sybase mediante las bibliotecas de cliente instaladas en el servidor federado.

Antes de empezar

El software de cliente Sybase debe estar instalado en el servidor federado.

Acerca de esta tarea

El archivo de configuración de cliente especifica la ubicación del servidor Sybase SQL Server y de la instancia de Adaptive Server Enterprise, así como el tipo de conexión (protocolo) para el servidor de bases de datos.

Debe instalar archivo de configuración de cliente en cada instancia del servidor federado que vaya a utilizarse para establecer conexión con Sybase.

Procedimiento

Para instalar y probar el archivo de configuración de cliente Sybase en los servidores federados que ejecutan Windows:

1. Instale el archivo de configuración de cliente mediante el programa de utilidad que se incluye con el software Sybase Open Client. Consulte la documentación de Sybase para obtener más información sobre el uso de este programa de utilidad.

El archivo de configuración de cliente se crea en el directorio %SYBASE%\ini. El nombre del archivo es sql.ini.

2. Pruebe la conexión para asegurarse de que el software Sybase Open Client puede conectarse con el servidor Sybase.

Utilice un programa de utilidad de consulta Sybase adecuado, como isql, para probar la conexión.

Por ejemplo, si el software Sybase Open Client está instalado en la vía de acceso de directorio D:\djxclient\sybase\V125, puede emitir los siguientes mandatos desde una línea de mandatos:

```
cd D:\djxclient\sybase\V125\OCS-12_5\bin
isql -Ssybnode -Umary
```

De forma alternativa, puede emitir el siguiente mandato desde un indicador de mandatos:

```
%SYBASE%\%SYBASE_OCS%\bin\isql -Ssybnode -Umary
```

Especificación de la vía de acceso para el archivo de interfaz

: Si desea utilizar un archivo de interfaz que no sea el archivo predeterminado, utilice la opción de servidor IFILE para especificar la vía de acceso. El derivador Sybase busca el archivo de interfaz en las siguientes ubicaciones, en el orden especificado:

- a. Opción de servidor IFILE
- b. %DB2PATH%\interfaces
- c. %SYBASE%\ini\sql.ini

Tras completar esta tarea, podrá definir las variables de entorno.

Configuración y prueba del archivo de configuración del cliente Sybase (UNIX)

El archivo de configuración del cliente Sybase se utiliza para conectarse a las bases de datos Sybase utilizando las bibliotecas de cliente que hay instaladas en el servidor federado.

Antes de empezar

El software Sybase Client SDK debe estar instalado en el servidor federado.

El archivo de configuración del cliente especifica la ubicación de cada Sybase SQL Server y cada instancia de Adaptive Server Enterprise, así como el tipo de conexión (protocolo) del servidor de bases de datos.

Debe configurar un archivo de configuración de cliente en cada instancia del servidor federado que se vaya a utilizar para conectarse a Sybase.

Procedimiento

Para configurar y probar el archivo de configuración del cliente Sybase en los servidores federados que ejecutan UNIX:

1. Configure el archivo de configuración del cliente utilizando el programa de utilidad que se proporciona con el software de Sybase Open Client.

El archivo de configuración del cliente se crea en el directorio `$$SYBASE`. El nombre predeterminado del archivo es `interfaces`. Consulte la documentación de Sybase para obtener más información sobre el uso de este programa de utilidad.

2. Pruebe la conexión para asegurarse de que el software de Sybase Open Client puede conectarse con el servidor Sybase.

Utilice el programa de utilidad de consultas de Sybase correspondiente, por ejemplo, `isql`, para probar la conexión.

Por ejemplo, si el software de Sybase Open Client está instalado en la vía de acceso de directorios `/opt/djxcclient/sybase/V125`, puede emitir el siguiente mandato desde un indicador de UNIX:

```
cd /opt/djxcclient/sybase/V125/OCS-12_5
isql -Ssybnode -Umary
```

O bien, puede emitir el siguiente mandato desde un indicador de UNIX:

```
$$SYBASE/$SYBASE_OCS/bin/isql -Ssybnode -Umary
```

Especificación de la vía de acceso del archivo `interfaces`

: Si desea utilizar un archivo `interfaces` distinto al archivo predeterminado, utilice la opción de servidor `IFILE` para especificar la vía de acceso. El derivador de Sybase busca el archivo `interfaces` en las siguientes ubicaciones, en el orden especificado:

- a. La opción de servidor `IFILE`
- b. `sqllib/interfaces`
- c. `$$SYBASE/interfaces`

Cuando finalice esta tarea, puede establecer las variables de entorno.

Registro del derivador de Sybase

Debe registrar un derivador para acceder a los orígenes de datos Sybase. Los servidores federados utilizan derivadores para comunicarse con los orígenes de datos y recuperar datos de éstos. Los derivadores se implementan como un conjunto de archivos de biblioteca.

Procedimiento

Para registrar el derivador de Sybase:

Utilice uno de los métodos siguientes:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database .	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Emita la sentencia CREATE WRAPPER y especifique el nombre predeterminado del derivador de Sybase.	<p>Por ejemplo:</p> <pre>CREATE WRAPPER CTLIB</pre> <p>Recuerde: Cuando registra el derivador utilizando el nombre predeterminado, CTLIB, el servidor federado utiliza automáticamente la biblioteca de derivador de Sybase correspondiente al sistema operativo donde se ejecuta el servidor federado.</p> <p>Si el nombre del derivador predeterminado entra en conflicto con un nombre de derivador existente en la base de datos federada, puede sustituir el nombre del derivador predeterminado por otro de su elección. Si utiliza un nombre distinto al nombre predeterminado, deberá incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER.</p> <p>Por ejemplo, para registrar un derivador con el nombre sybase_wrapper en un servidor federado que utiliza AIX, emita la siguiente sentencia:</p> <pre>CREATE WRAPPER sybase_wrapper LIBRARY 'libdb2ctlib.a';</pre> <p>El archivo de biblioteca de derivador que especifica depende del sistema operativo del servidor federado.</p>

Cuando finalice esta tarea, puede registrar la definición de servidor.

Archivos de biblioteca de derivador de Sybase

Los archivos de biblioteca de derivador de Sybase se añaden al servidor federado cuando se instala el derivador.

Cuando instala el derivador de Sybase, se añaden tres archivos de biblioteca a la vía de acceso de directorios predeterminada. Por ejemplo, si el servidor federado

se ejecuta en AIX, los archivos de biblioteca que se añaden a la vía de acceso de directorios libdb2ctlib.a, libdb2ctlibF.a y libdb2ctlibU.a. El derivador de Sybase utiliza internamente los otros archivos de biblioteca de derivador.

Si no utiliza el nombre de derivador predeterminado cuando registra el derivador, debe incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER y especificar el nombre de archivo de biblioteca predeterminado.

Las vías de acceso de directorios predeterminadas y los nombres de archivo de biblioteca del derivador predeterminados se especifican en la tabla siguiente.

Tabla 36. Nombres de archivo y ubicaciones de la biblioteca de derivador de Sybase

Sistema operativo	Vía de acceso de directorios	Nombre de archivo de biblioteca
AIX	/usr/opt/vía_acceso_instalación/lib32/ /usr/opt/vía_acceso_instalación/lib64/	libdb2ctlib.a
Linux	/opt/IBM/db2/vía_acceso_instalación/lib32libdb2ctlib.so /opt/IBM/db2/vía_acceso_instalación/lib64	
Solaris	/opt/IBM/db2/vía_acceso_instalación/lib32libdb2ctlib.so /opt/IBM/db2/vía_acceso_instalación/lib64	
Windows	%DB2PATH%\bin	db2ctlib.dll

vía_acceso_instalación es la vía de acceso de directorios donde se instala IBM InfoSphere Federation Server en UNIX o Linux.

Registro de las definiciones de servidor para un origen de datos Sybase

Debe registrar cada servidor Sybase al que desee acceder en la base de datos federada.

Procedimiento

Para registrar una definición de servidor para un origen de datos Sybase:

1. Ubique el nombre de nodo en el archivo interfaces de Sybase. El siguiente ejemplo muestra las entradas de los archivos de interfaces de los servidores federados que ejecutan UNIX o Windows:

UNIX:

```
sybase125
query tcp ether anaconda 4100
```

Windows:

```
[sybase125]
query=TCP,anaconda,4100
```

- La primera línea de cada ejemplo es el nombre de nodo, por ejemplo, sybase125.
- La segunda línea muestra el tipo de conexión, el nombre de host y el número de puerto. En este ejemplo, TCP indica que es una conexión TCP/IP, anaconda es el nombre de host y 4100 es el número de puerto. .

2. Para crear la definición de servidor, utilice uno de los métodos siguientes:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Emita la sentencia CREATE SERVER .	<p>Por ejemplo:</p> <pre>CREATE SERVER nombre_definición_servidor TYPE SYBASE VERSION número_versión_cliente_Sybase WRAPPER nombre_derivador OPTIONS (NODE 'nombre_nodo', DBNAME 'nombre_base_datos');</pre> <p>Aunque las variables '<i>nombre_nodo</i>' y '<i>nombre_base_datos</i>' se especifican como opciones en la sentencia CREATE SERVER, estas opciones son necesarias para los orígenes de datos Sybase.</p> <p>Importante: Si no ha cambiado el nombre del archivo <code>sql.ini</code> por interfaces al configurar el archivo de configuración del cliente Sybase, debe incluir la opción de servidor <code>IFILE</code> cuando registre la definición de servidor.</p> <p>Después de registrar la definición de servidor, utilice la sentencia ALTER SERVER para añadir o eliminar opciones de servidor.</p>

Cuando finalice esta tarea, puede crear correlaciones de usuarios.

Sentencia **CREATE SERVER** - Ejemplos para el derivador de Sybase

Utilice la sentencia **CREATE SERVER** para registrar definiciones de servidor para el derivador de Sybase. Este tema proporciona un ejemplo completo con los parámetros necesarios y un ejemplo con opciones adicionales del servidor.

El ejemplo siguiente muestra cómo registrar una definición de servidor para un derivador de Sybase emitiendo la sentencia **CREATE SERVER**:

```
CREATE SERVER SYBSERVER TYPE SYBASE VERSION 12.0 WRAPPER CTLIB
  OPTIONS (NODE 'nodo_syb', DBNAME 'sybdb');
```

SYBSERVER

Nombre que asigna al servidor Sybase. No se permiten los nombres de definición de servidor duplicados.

TYPE SYBASE

Especifica el tipo de servidor de origen de datos para el que está configurando el acceso. Para el derivador de CTLIB, el tipo de servidor debe ser **SYBASE**.

VERSION 12.0

Versión del software de cliente de base de datos Sybase que se utiliza para la conexión federada.

WRAPPER CTLIB

Nombre de derivador que ha especificado en la sentencia **CREATE WRAPPER**.

NODE *'nodo_syb'*

Nombre del nodo donde reside el servidor Sybase. Obtenga el nombre de nodo del archivo interfaces. Este valor es sensible a las mayúsculas y minúsculas.

Aunque el nombre de nodo se especifica como una opción en la sentencia CREATE SERVER, se necesita para los orígenes de datos Sybase.

DBNAME *'sybdb'*

Nombre de la base de datos Sybase a la que desea acceder. Este valor es sensible a las mayúsculas y minúsculas.

Aunque el nombre de la base de datos se especifica como una opción en la sentencia CREATE SERVER, se necesita para los orígenes de datos Sybase.

Opciones de servidor

Cuando crea una definición de servidor, puede especificar opciones de servidor adicionales en la sentencia CREATE SERVER. Las opciones de servidor pueden ser opciones de servidor generales y opciones de servidor específicas de Sybase.

Opción de servidor IFILE

Si no utiliza el archivo interfaces predeterminado, debe crear un archivo interfaces e incluir la opción de servidor IFILE en la sentencia CREATE SERVER. El archivo interfaces predeterminado es \$SYBASE/interfaces para Linux y UNIX, y %SYBASE%\ini\sql.ini para Windows.

El valor que especifica para la opción de servidor IFILE es el nombre y la vía de acceso completa del archivo sql.ini de Sybase Open Client.

En el siguiente ejemplo se muestra cómo utilizar la opción de servidor IFILE cuando registra una definición de servidor en un servidor federado que ejecuta Windows:

```
CREATE SERVER SYBSERVER TYPE SYBASE VERSION 12.0 WRAPPER CTLIB
  OPTIONS (NODE 'nodo_syb', DBNAME 'sybdb',
  IFILE 'C:\Sybase\ini\sql.ini');
```

Opción de servidor TIMEOUT

La opción de servidor TIMEOUT establece el número de segundos que espera el derivador una respuesta del servidor Sybase. Utilice la opción TIMEOUT para evitar puntos muertos en las transacciones.

En el siguiente ejemplo se muestra cómo especificar la opción de servidor TIMEOUT cuando registra una definición de servidor:

```
CREATE SERVER SYBSERVER TYPE SYBASE VERSION 12.0 WRAPPER CTLIB
  OPTIONS (NODE 'nodo_syb', DBNAME 'sybdb',
  TIMEOUT '60');
```

Las opciones de servidor adicionales específicas de Sybase son:

- LOGIN_TIMEOUT
- PACKET_SIZE

Creación de las correlaciones de usuarios para un origen de datos Sybase

Cuanto intenta acceder a un servidor Sybase, el servidor federado establece una conexión con el servidor Sybase utilizando un ID de usuario y una contraseña que son válidos para dicho origen de datos. Debe definir una asociación (una correlación de usuarios) entre cada ID de usuario y contraseña de servidor federado y el ID de usuario y la contraseña de origen de datos correspondiente.

Cree una correlación de usuarios para cada ID de usuario que vaya a acceder al sistema federado para enviar solicitudes distribuidas al origen de datos Sybase.

Procedimiento

Cree correlaciones de usuarios para un origen de datos Sybase:

Emita una sentencia CREATE USER MAPPING.

Por ejemplo:

```
CREATE USER MAPPING FOR IDusuario_local SERVER nombre_definición_servidor
    OPTIONS (REMOTE_AUTHID 'IDusuario_remoto', REMOTE_PASSWORD
'contraseña_remota');
```

Cuando finalice esta tarea, pruebe la conexión con el servidor Sybase.

Sentencia CREATE USER MAPPING - Ejemplos para el derivador de Sybase

Utilice la sentencia CREATE USER MAPPING para correlacionar un ID de usuario de servidor federado con un ID de usuario y una contraseña de servidor Sybase. En este tema se proporciona un ejemplo completo con los parámetros necesarios y un ejemplo que muestra cómo utilizar el registro especial USER de DB2 con la sentencia CREATE USER MAPPING.

El ejemplo siguiente muestra cómo correlacionar un ID de usuario de servidor federado con un ID de usuario y una contraseña de servidor Sybase:

```
CREATE USER MAPPING FOR maria SERVER SYBSERVER
    OPTIONS (REMOTE_AUTHID 'mary', REMOTE_PASSWORD 'day2night');
```

maria Especifica el ID de usuario local que está correlacionando con un ID de usuario definido en el servidor Sybase.

SERVER *SYBSERVER*

Especifica el nombre de definición de servidor que ha registrado en la sentencia CREATE SERVER del servidor Sybase.

REMOTE_AUTHID '*mary*'

Especifica el ID de usuario en el servidor Sybase con el que está correlacionando *maria*. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_ID en 'U' o 'L' en la sentencia CREATE SERVER.

Aunque el ID de usuario remoto se especifica como una opción en la sentencia CREATE SERVER, se necesita para los orígenes de datos Sybase.

REMOTE_PASSWORD '*day2night*'

Especifica la contraseña asociada con '*mary*'. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_PW en 'U' o 'L' en la sentencia CREATE SERVER.

Aunque la contraseña remota se especifica como una opción en la sentencia CREATE SERVER, se necesita para los orígenes de datos Sybase.

Registro especial USER de DB2

Puede utilizar el registro especial USER de DB2 para correlacionar el ID de autorización de la persona que emite la sentencia CREATE USER MAPPING con el ID de autorización de origen de datos especificado en la opción de usuario REMOTE_AUTHID.

El siguiente ejemplo muestra una sentencia CREATE USER MAPPING que incluye el registro especial USER:

```
CREATE USER MAPPING FOR USER SERVER SYBSERVER
    OPTIONS (REMOTE_AUTHID 'mary', REMOTE_PASSWORD 'day2night');
```

Prueba de la conexión al servidor Sybase

Pruebe la conexión con el servidor de origen de datos Sybase para determinar si el servidor federado se ha configurado correctamente para acceder a los orígenes de datos Sybase.

Puede probar la conexión con el servidor Sybase utilizando la definición de servidor y las correlaciones de usuario que ha definido.

Procedimiento

Para probar la conexión al servidor Sybase:

Abra una sesión de paso a través y emita una sentencia SELECT en las tablas del sistema Sybase. Si la sentencia SELECT devuelve un recuento, la definición de servidor y la correlación de usuarios se han configurado correctamente.

Por ejemplo:

```
SET PASSTHRU nombre_definición_servidor
SELECT count(*) FROM dbo.sysobjects
SET PASSTHRU RESET
```

Si la sentencia SELECT devuelve un error, debe resolver los errores de conexión.

Cuando finalice esta tarea, puede registrar apodos para las tablas y vistas de Sybase.

Resolución de errores de conexión de origen de datos

Una conexión de prueba con el servidor de origen de datos puede devolver un error por varios motivos. Existen acciones que permiten determinar el motivo del error.

Síntoma

Se devuelve un error al intentar establecer conexión con el origen de datos.

Causa

Hay varias causas posibles que explican un problema de conexión.

Resolución del problema

Para solucionar los problemas de errores de conexión con el origen de datos, consulte los siguientes elementos:

- Verifique que el origen de datos está disponible.
- Si procede, asegúrese de que el servidor de origen de datos esté configurado para conexiones entrantes.
- Asegúrese de que los valores de correlación de usuarios para las opciones REMOTE_AUTHID y REMOTE_PASSWORD son válidos para las conexiones con el origen de datos. Modifique la correlación de usuarios o cree otra correlación de usuarios según sea necesario.
- Si procede, asegúrese de que el software de cliente del servidor federado esté instalado y configurado correctamente para conectarse al origen de datos.
- Para orígenes de datos ODBC, asegúrese de que el controlador ODBC del servidor federado esté instalado y configurado correctamente para establecer conexión con el servidor de origen de datos ODBC. En servidores federados que ejecutan Windows, utilice la herramienta ODBC Data Source Administrator para verificar el controlador. En servidores federados que ejecutan UNIX, consulte la documentación del proveedor del cliente ODBC.
- Verifique que los valores para las variables establecidas en el servidor federado son correctas para el origen de datos. Estas variables incluyen las variables de entorno del sistema, las variables de la archivo db2dj.ini y las variables del registro de perfiles de DB2 (db2set).
- Compruebe la definición del servidor. Si es necesario, elimine la definición del servidor y créela de nuevo.

Registro de apodos para vistas y tablas de Sybase

Para cada definición de servidor Sybase, debe registrar un apodo para cada tabla o vista a la que desee acceder. Utilice estos apodos en lugar de los nombres de los objetos de origen de datos cuando consulte los servidores Sybase.

Antes de empezar

Actualice las estadísticas en el origen de datos Sybase antes de registrar un apodo. La base de datos federada se basa en las estadísticas de catálogo de origen de datos para optimizar el proceso de las consultas. Utilice el mandato de origen de datos que es equivalente al mandato RUNSTATS de DB2 para actualizar las estadísticas de origen de datos.

Procedimiento

Para registrar un apodo para tablas o vistas de Sybase, utilice uno de los métodos siguientes:

Métodos	Descripciones
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .

Métodos	Descripciones
Emita la sentencia CREATE NICKNAME. Los apodos pueden tener una longitud de hasta 128 caracteres.	Por ejemplo: CREATE NICKNAME <i>apodo</i> FOR <i>nombre_definición_servidor</i> . <i>"esquema_remoto"</i> . <i>"tabla.remota"</i> ;

Cuando se crea el apodo, el servidor federado consulta el catálogo de orígenes de datos utilizando el apodo. Esta consulta prueba la conexión con la tabla o la vista del origen de datos. Si la conexión no funciona, recibirá un mensaje de error.

Repita este paso para cada tabla o vista de Sybase para la que desee crear un apodo.

Sentencia CREATE NICKNAME - Ejemplos para el derivador de Sybase

Utilice la sentencia CREATE NICKNAME para registrar un apodo para una tabla o vista de Sybase a la que desea acceder. Este tema incluye un ejemplo completo con los parámetros necesarios.

El ejemplo siguiente muestra cómo registrar un apodo para una tabla o una vista de Sybase utilizando la sentencia CREATE NICKNAME.

```
CREATE NICKNAME SYBSALES FOR SYBSERVER."vinnie"."europe";
```

SYBSALES

Apodo exclusivo que se utiliza para identificar la tabla o la vista de Sybase.

Importante: El nombre de apodo consta de dos partes: el esquema y el apodo. Si omite el esquema al registrar el apodo, el esquema del apodo será el ID de autorización del usuario que registra el apodo.

*SYBSERVER."*vinnie*".*"europe"**

Es un identificador de tres partes para el objeto remoto:

- *SYBSERVER* es el nombre de definición de servidor que ha asignado al servidor de bases de datos Sybase en la sentencia CREATE SERVER.
- *vinnie* es el ID de usuario del propietario al que pertenece la tabla o la vista.
- *europe* es el nombre de la tabla o la vista remota a la que desea acceder.

El servidor federado convierte los nombres de los esquemas y las tablas de Sybase a mayúsculas, a menos que los escriba entre comillas.

Resolución de problemas de la configuración del derivador de Sybase

Problemas al cargar la biblioteca de derivador de Sybase

Cuando crea el derivador de Sybase, puede encontrar errores relacionados con la instalación del software de Sybase Open Client que impiden que se cargue la biblioteca de derivador de Sybase.

Síntoma

Cuando crea un derivador de Sybase, se emite el siguiente error de SQL:

SQL10013N No se ha podido cargar la biblioteca especificada "db2ctlibF.dll".

Causa

La biblioteca de la interfaz XA de Sybase del Administrador de transacciones distribuidas ASE no se ha instalado en el sistema Windows con el software de Sybase Open Client.

Resolución del problema

Vuelva a instalar el software de Sybase Open Client en Windows y seleccione la opción de instalación **Completa** o **Personalizada**. Si selecciona la opción de instalación **Personalizada**, especifique la opción **Biblioteca de interfaz XA del Administrador de transacciones distribuidas ASE**.

Falta la variable de entorno SYBASE

Si el archivo db2dj.ini no está en el directorio correcto o no se encuentra, aparecerá un error de SQL.

Síntoma

Se emite el siguiente error de SQL:

SQL1822N Se ha recibido un código de error inesperado "" del origen de datos "nombre de servidor". El texto y las señales asociadas son "Variable SYBASE no establecida".

Causa

El archivo db2dj.ini no se ha encontrado o no contiene la variable de entorno SYBASE. El archivo db2dj.ini se encuentra en el directorio especificado por la variable de registro DB2_DJ_INI de DB2. Si la variable DB2_DJ_INI no se establece, el archivo db2dj.ini reside en una de las siguientes vías de acceso predeterminadas, dependiendo del sistema operativo:

- En UNIX: *padre_instancia*/sqllib/cfg/db2dj.ini.

padre_instancia

El directorio padre del propietario de la instancia

- En Windows: %DB2PATH%\cfg\db2dj.ini

%DB2PATH%

El directorio donde está instalado el sistema de base de datos de DB2, por ejemplo, C:\Archivos de programa\IBM\sqllib.

Resolución del problema

Cree un archivo db2dj.ini con las variables de entorno Sybase necesarias y colóquelo en el directorio correcto para el sistema operativo. Puede utilizar el editor de texto para crear el archivo db2dj.ini.

Falta el nombre de nodo Sybase

Si el archivo de configuración del cliente Sybase no está configurado correctamente, el sistema federado no podrá ubicar el nombre de nodo Sybase.

Síntoma

Se emite el siguiente error de SQL:

SQL1097N

No se ha encontrado el nombre de nodo en el directorio del nodo.

Causa

No se ha encontrado el nombre de nodo Sybase.

Resolución del problema

Para resolver el problema:

- Si se ha especificado la opción de servidor IFILE, compruebe que el nombre de nodo esté declarado en el archivo que especifica.
- Si el archivo interfaces existe en el directorio sqllib, compruebe que el nombre de nodo esté declarado en él.
- Si no se ha especificado la opción de servidor IFILE y el archivo interfaces no existe en el directorio sqllib, compruebe que el nombre de nodo esté declarado en el archivo %SYBASE%\ini\sql.ini en Windows o el archivo \$SYBASE/interfaces en UNIX.

Configuración del acceso a orígenes de datos de archivos con estructura de tabla

Puede integrar los datos que hay en un archivo con estructura de tabla con información de otros orígenes utilizando un sistema federado.

Procedimiento

Para configurar un servidor federado para acceder a orígenes de datos de archivos con estructura de tabla, debe proporcionar al servidor federado información acerca de los orígenes de datos y los objetos a los que desea acceder. Una vez configurado el servidor federado, puede crear consultas y utilizar las funciones personalizadas para acceder a los orígenes de datos de archivos con estructura de tabla.

Visión general de los archivos estructurados en tablas

Un archivo estructurado en tablas es un archivo que tiene una estructura normal formada por una serie de registros o filas de información. Cada registro contiene el mismo número de campos. Los datos de los campos se separan mediante un delimitador, como por ejemplo, una coma.

El ejemplo siguiente muestra el contenido de un archivo denominado DRUGDATA1.TXT. Este archivo contiene tres registros, cada uno con tres campos, separados por comas:

```
234,DrugnameA,Manufacturer1  
332,DrugnameB,Manufacturer2  
333,DrugnameC,Manufacturer2
```

El primer campo es el número de ID exclusivo del fármaco. El segundo campo es el nombre del fármaco. El tercer campo es el nombre del fabricante que produce el fármaco.

El delimitador de campo puede tener más de un carácter de longitud. No puede utilizarse una comilla simple como carácter delimitador. El delimitador debe ser el mismo en todo el archivo. Un valor nulo se representa mediante dos

delimitadores juntos o un delimitador seguido por una línea de terminación, si el campo NULL es el último de la línea. El delimitador de columna no puede existir como datos válidos para una columna.

Por ejemplo:

```
234,DrugnameA,Manufacturer1
332,DrugnameB,Manufacturer2
333,DrugnameC,Manufacturer2
356,,Manufacturer1
```

Atributos de archivos estructurados en tablas

Los registros, o las filas, de los archivos estructurados en tablas pueden estar ordenados o no. El derivador de archivos estructurados en tablas busca archivos que están ordenados de forma más eficiente que los archivos que no están ordenados.

Si los datos de un archivo estructurado en tablas están ordenados, la ordenación debe ser en sentido ascendente en la columna de claves. Debe establecer la opción SORTED con el valor Y en la columna de claves cuando defina las columnas para el apodo. En caso contrario, el derivador procesa los datos del archivo estructurado en tablas como datos no ordenados.

Archivos ordenados

DRUGDATA1.TXT contiene registros ordenados. El archivo está ordenado por el primer campo, el número de ID exclusivo para el fármaco. Este campo es la clave principal porque es exclusivo para cada fármaco. Los archivos ordenados deben estar ordenados en sentido ascendente.

```
234,DrugnameA,Manufacturer1
332,DrugnameB,Manufacturer2
333,DrugnameC,Manufacturer2
```

Archivos no ordenados

DRUGDATA2.TXT contiene registros no ordenados. Los registros del archivo no aparecen en ningún orden determinado.

```
556,DrugnameB,Manufacturer2
234,DrugnameA,Manufacturer1
721,DrugnameC,Manufacturer2
```

Derivador de archivos estructurados en tablas

Los datos de un archivo estructurado en tablas pueden unirse con datos de otros archivos estructurados en tablas, datos relacionales o datos no relacionales y no estructurados.

Mediante un derivador, el servidor federado puede procesar sentencias SQL que consultan datos de un archivo estructurado en tablas como si los datos estuvieran en una tabla o vista relacional ordinaria.

En la figura siguiente se ilustra cómo trabaja el servidor federado con los archivos estructurados en tablas.

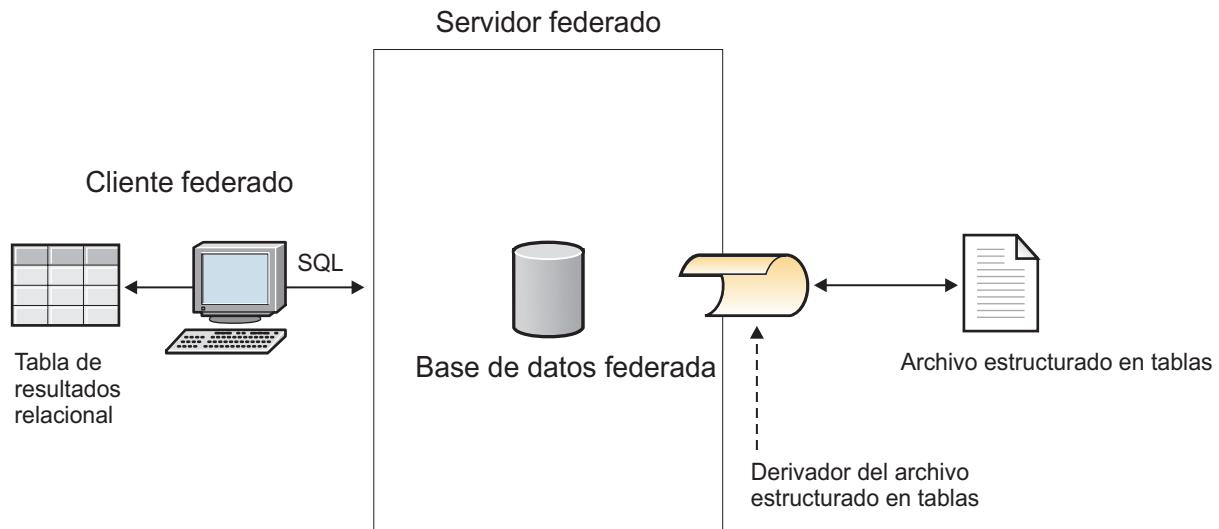


Figura 8. Funcionamiento del derivador de archivos estructurados en tablas

Por ejemplo, el archivo estructurado en tablas DRUGDATA2.TXT se encuentra en un sistema de su laboratorio. El archivo contiene estos datos:

```
556,DrugnameB,Manufacturer1
234,DrugnameA,Manufacturer2
721,DrugnameC,Manufacturer2
```

Resulta tedioso intentar consultar y establecer una correspondencia entre estos datos y otras tablas de otros orígenes de datos que utilice.

Una vez que haya registrado el archivo DRUGDATA2.TXT con el servidor federado, el derivador de archivos estructurados en tablas podrá acceder a los datos del archivo como si estuvieran en una tabla relacional.

Por ejemplo, puede ejecutar la siguiente consulta:

```
SELECT * FROM DRUGDATA2 ORDER BY DCODE
```

Esta consulta produce los siguientes resultados:

DCODE	DRUG	MANUFACTURER
234	DrugnameA	Manufacturer2
556	DrugnameB	Manufacturer1
721	DrugnameC	Manufacturer2

Puede unir los datos del archivo DRUGDATA2.TXT con los datos de otros orígenes de datos relacionales y no relacionales y analizar todos los datos conjuntamente.

Adición de orígenes de datos de archivos con estructura de tabla a un servidor federado

Para configurar un servidor federado para acceder a orígenes de datos de archivos con estructura de tabla, debe proporcionar al servidor federado información acerca de los orígenes de datos y los objetos a los que desea acceder.

Antes de empezar

- Federation debe estar instalado en el servidor que actuará como servidor federado.
- Debe existir una base de datos en el servidor federado.

Puede configurar un servidor federado para acceder a los datos almacenados en los orígenes de datos de archivos con estructura de tabla utilizando el Centro de control o emitiendo sentencias SQL en la línea de mandatos. El Centro de control incluye un asistente que sirve de guía en los pasos necesarios para configurar los objetos federados que se necesitan.

Procedimiento

Para añadir los orígenes de datos de archivos con estructura de tabla a un servidor federado:

1. Registre el derivador de archivos con estructura de tabla.
2. Registre la definición de servidor de archivos con estructura de tabla.
3. Registre los apodos de los archivos con estructura de tabla.

Registro del derivador de archivos con estructura de tabla

Debe registrar un derivador para acceder a los orígenes de datos de archivos con estructura de tabla. Los servidores federados utilizan derivadores para comunicarse con los orígenes de datos y recuperar datos de éstos. Los derivadores se implementan como un conjunto de archivos de biblioteca.

Puede registrar un derivador utilizando el Centro de control o la línea de mandatos. El Centro de control incluye un asistente que sirve de guía en los pasos necesarios para registrar el derivador.

Procedimiento

Para registrar el derivador de archivos con estructura de tabla:

Elija el método que desee utilizar para registrar el derivador de archivos con estructura de tabla:

Método	Procedimiento
Mediante el Centro de control	Inicie el Asistente para objetos federados. Pulse con el botón derecho en la carpeta Objetos de base de datos federada y pulse Crear objetos federados .
Desde la línea de mandatos	Emita la sentencia CREATE WRAPPER. <pre>CREATE WRAPPER nombre_derivador LIBRARY nombre_biblioteca;</pre> <p>Por ejemplo, para registrar un derivador con el nombre <code>derivador_archivos_planos</code> en el servidor federado que utiliza el sistema operativo AIX, emita la siguiente sentencia:</p> <pre>CREATE WRAPPER derivador_archivos_planos LIBRARY 'libdb2lsfile.a';</pre>

Debe especificar el parámetro LIBRARY en la sentencia CREATE WRAPPER. El nombre del archivo de biblioteca de derivador que especifica depende del sistema operativo del servidor federado. Consulte la lista de archivos de biblioteca de derivador de archivos con estructura de tabla para ver el nombre de biblioteca correcto que debe especificar en la sentencia CREATE WRAPPER.

Archivos de biblioteca de derivador de archivos con estructura de tabla:

Los archivos de biblioteca de derivador de archivos con estructura de tabla se añaden al servidor federado cuando se instala IBM InfoSphere Federation Server.

Cuando instala IBM InfoSphere Federation Server, se añaden tres archivos de biblioteca a la vía de acceso de directorios predeterminada. Por ejemplo, si el servidor federado se ejecuta en AIX, los archivos de biblioteca que se añaden a la vía de acceso de directorios son libdb2lsfile.a, libdb2lsfileF.a y libdb2lsfileU.a. El archivo de biblioteca del derivador predeterminado es libdb2lsfile.a. Los otros archivos de biblioteca de derivador se utilizan con opciones de derivador específicas.

Debe incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER y especificar el nombre de archivo de biblioteca del derivador predeterminado.

Las vías de acceso de directorios predeterminadas y los nombres de archivo de biblioteca del derivador predeterminados se especifican en la tabla siguiente.

Tabla 37. Nombres de archivo y ubicaciones de la biblioteca de cliente de archivos con estructura de tabla

Sistema operativo	Vía de acceso de directorios	Nombres de archivo de biblioteca de derivador
AIX	/usr/opt/<vía_acceso_instalación>/lib32/ /usr/opt/<vía_acceso_instalación>/lib64/	libdb2lsfile.a
Linux	/opt/IBM/db2/<vía_acceso_instalación>/lib32libdb2lsfile.so /opt/IBM/db2/<vía_acceso_instalación>/lib64	
Solaris	/opt/IBM/db2/<vía_acceso_instalación>/lib32libdb2lsfile.so /opt/IBM/db2/<vía_acceso_instalación>/lib64	
Windows	%DB2PATH%\bin	db2lsfile.dll

<vía_acceso_instalación> es la vía de acceso de directorios donde se instala IBM InfoSphere Federation Server en Linux o UNIX.

%DB2PATH% es la variable de entorno que se utiliza para especificar la vía de acceso de directorios donde se instala IBM InfoSphere Federation Server en Windows. La vía de acceso de directorios predeterminada de Windows es C:\Archivos de programa\IBM\SQLLIB.

Registro de la definición de servidor para archivos con estructura de tabla

Para los archivos con estructura de tabla, debe registrar una definición de servidor porque la jerarquía de los objetos federados requiere que los archivos con estructura de tabla, que se identifican mediante apodos, se asocien con un objeto de definición de servidor específico.

Puede registrar una definición de servidor utilizando el Centro de control o la línea de mandatos. El Centro de control incluye un asistente que sirve de guía en los pasos necesarios para registrar la definición de servidor.

Procedimiento

Para registrar una definición de servidor para un origen de datos de archivos con estructura de tabla:

Elija el método que desee utilizar para registrar la definición de servidor:

Método	Procedimiento
Mediante el Centro de control	Inicie el Asistente para objetos federados. Pulse con el botón derecho en la carpeta Objetos de base de datos federada y pulse Crear objetos federados .
Desde la línea de mandatos	Emita la sentencia CREATE SERVER. Por ejemplo: <pre>CREATE SERVER nombre_definición_servidor WRAPPER nombre_derivador;</pre>

Sentencia CREATE SERVER - Ejemplo para el derivador de archivos estructurados en tablas:

Utilice la sentencia CREATE SERVER para registrar la definición de servidor para el derivador de archivos estructurados en tablas. Este ejemplo muestra los parámetros necesarios.

El ejemplo siguiente muestra cómo registrar una definición de servidor denominada `biochem_lab` para un archivo de texto que contiene datos bioquímicos. La sentencia CREATE SERVER que debe emitir es la siguiente:

```
CREATE SERVER biochem_lab WRAPPER
derivador_archivos_planos;
```

biochem_lab

Nombre que asigna a la definición del servidor de archivos estructurados en tablas. No se permiten los nombres de definición de servidor duplicados.

WRAPPER *derivador_archivos_planos*

Nombre de derivador que ha especificado en la sentencia CREATE WRAPPER.

Registro de apodos para archivos con estructura de tabla

Debe registrar un apodo para cada archivo con estructura de tabla al que desee acceder. Utilice estos apodos en lugar de los nombres de los archivos cuando consulte los orígenes de datos de archivos con estructura de tabla.

Restricciones

- Si un campo no numérico es demasiado largo para su tipo de columna, los datos sobrantes se truncan.
- Si un campo decimal en el archivo tiene más dígitos después de la coma de lo permitido, los datos sobrantes se truncan. Por ejemplo, si el número es 10,123456 y el tipo de datos especifica que se permiten 3 dígitos después de la coma, el número se trunca a 10,123. El carácter que separa el entero de la fracción viene determinado por el elemento RADIXCHAR de la categoría de soporte multilingüístico LC_NUMERIC. El parámetro de escala del tipo de datos de columna especifica el número de dígitos permitidos después de la coma.
- Si intenta acceder a orígenes de datos de archivos con estructura de tabla que están en una unidad compartida desde un servidor federado que ejecuta Windows 2003, las consultas pueden fallar. Esta es una limitación de Windows 2003. Para evitar este problema, especifique la vía de acceso absoluta en la opción FILE_PATH de la sentencia CREATE NICKNAME.

- La longitud máxima para una línea de datos es de 10 megabytes (10485760 bytes).
-

Cuando crea un apodo para un archivo con estructura de tabla, la información en los datos del archivo se correlaciona con una tabla relacional. Puede crear apodos para el archivo con estructura de tabla de dos formas:

- Especifique el archivo con estructura de tabla cuando cree el apodo utilizando la opción de apodo FILE_PATH.
- Especifique el archivo con estructura de tabla cuando consulte el origen de datos utilizando la opción de columna de apodo DOCUMENT. Cuando se utiliza esta opción, puede utilizarse el apodo para representar datos de cualquier archivo con estructura de tabla cuyo esquema coincida con la definición de apodo.

Los nombres que da a los apodos pueden tener una longitud de hasta 128 caracteres.

Puede registrar un apodo utilizando el Centro de control o la línea de mandatos. El Centro de control incluye un asistente que sirve de guía en los pasos necesarios para registrar el apodo.

Procedimiento

Para registrar un apodo para un archivo con estructura de tabla:

Elija el método que desee utilizar para registrar el apodo:

Método	Procedimiento
Mediante el Centro de control	Inicie el Asistente para objetos federados. Pulse con el botón derecho en la carpeta Objetos de base de datos federada y pulse Crear objetos federados .
Utilizando la línea de mandatos	Emita la sentencia CREATE NICKNAME. Por ejemplo: <pre>CREATE NICKNAME apodo (column_name tipo_datos, column_name tipo_datos, column_name tipo_datos) FOR SERVER nombre_definición_servidor OPTIONS (opciones_apodo);</pre>

Repita este paso para cada archivo con estructura de tabla para el que desee crear un apodo.

Sentencia CREATE NICKNAME - Ejemplos para el derivador de archivos con estructura de tabla:

Utilice la sentencia CREATE NICKNAME para registrar un apodo para un archivo con estructura de tabla al que desea acceder.

Debe especificar la opción de apodo FILE_PATH o la opción de columna de apodo DOCUMENT cuando registra un apodo para un archivo con estructura de tabla.

Creación de un apodo con la opción de apodo FILE_PATH

El siguiente ejemplo muestra una sentencia CREATE NICKNAME para el archivo con estructura de tabla 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

Apodo exclusivo que se utiliza para identificar el archivo con estructura de tabla.

El nombre de apodo consta de dos partes: el esquema y el apodo. Si omite el esquema al registrar el apodo, el esquema del apodo es el ID de autorización del usuario que registra el apodo.

Dcode INTEGER NOT NULL

Nombre y tipo de datos de una columna de archivo con estructura de tabla que contiene un código de fármaco.

Drug CHAR(20)

Nombre y tipo de datos de una columna de archivo con estructura de tabla que contiene el nombre de un fármaco.

Manufacturer CHAR(20)

Nombre y tipo de datos de una columna de archivo con estructura de tabla que contiene el nombre del fabricante del fármaco.

FOR SERVER *biochem_lab*

Nombre que ha asignado a la definición de servidor de archivo con estructura de tabla en la sentencia CREATE SERVER.

FILE_PATH '/usr/pat/DRUGDATA1.TXT'

Especifica la vía de acceso de directorios completa y el nombre de archivo del archivo con estructura de tabla que contiene los datos a los que desea acceder. La vía de acceso debe escribirse entre comillas simples.

Creación de un apodo con la opción de columna de apodo DOCUMENT

Cuando crea un apodo utilizando la opción de columna de apodo DOCUMENT, especifica que se proporcionará el nombre del archivo con estructura de tabla cuando se ejecute una consulta que utilice el apodo. Sólo puede especificar la opción de columna de apodo DOCUMENT en una columna cuando registra el apodo. La columna asociada con la opción DOCUMENT debe tener un tipo de datos VARCHAR o CHAR. Deberá incluir la vía de acceso completa del archivo cuando ejecute una consulta que utilice el apodo.

El siguiente ejemplo muestra una sentencia CREATE NICKNAME que especifica la opción de columna de apodo DOCUMENT:

```
CREATE NICKNAME customers
(
  doc VARCHAR(100) OPTIONS(DOCUMENT 'FILE'),
  name VARCHAR(16),
  address VARCHAR(30),
  id VARCHAR(16)
)
FOR SERVER biochem_lab
```

customers

Nombre exclusivo del apodo.

El nombre de apodo consta de dos partes: el esquema y el apodo. Si omite el esquema al registrar el apodo, el esquema del apodo es el ID de autorización del usuario que registra el apodo.

doc *VARCHAR(100)* **OPTIONS(DOCUMENT 'FILE')**

Nombre y tipo de datos de una columna que se utiliza para especificar el nombre del archivo con estructura de tabla al que desea acceder. Debe especificar el nombre de archivo cuando ejecute la consulta.

name *VARCHAR(16)*

Nombre y tipo de datos de una columna de archivo con estructura de tabla que contiene el nombre del cliente.

address *VARCHAR(30)*

Nombre y tipo de datos de una columna de archivo con estructura de tabla que contiene la dirección del cliente.

id *VARCHAR(16)*

FOR SERVER *biochem_lab*

Nombre que ha asignado a la definición de servidor de archivo con estructura de tabla en la sentencia CREATE SERVER.

FILE_PATH *'/usr/pat/DRUGDATA1.TXT'*

Especifica la vía de acceso de directorios completa y el nombre de archivo del archivo con estructura de tabla que contiene los datos a los que desea acceder. Debe especificar la opción de apodo FILE_PATH o DOCUMENT en la sentencia CREATE NICKNAME. La vía de acceso debe escribirse entre comillas simples.

Creación de un apodo con parámetros opcionales

El siguiente ejemplo muestra una sentencia CREATE NICKNAME para el archivo con estructura de tabla 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

Apodo exclusivo que se utiliza para identificar el archivo con estructura de tabla.

El nombre de apodo consta de dos partes: el esquema y el apodo. Si omite el esquema al registrar el apodo, el esquema del apodo es el ID de autorización del usuario que registra el apodo.

Dcode *INTEGER NOT NULL*

Nombre y tipo de datos de una columna de archivo con estructura de tabla que contiene un código de fármaco.

Drug *CHAR(20)*

Nombre y tipo de datos de una columna de archivo con estructura de tabla que contiene el nombre de un fármaco.

Manufacturer *CHAR(20)*

Nombre y tipo de datos de una columna de archivo con estructura de tabla que contiene el nombre del fabricante del fármaco.

FOR SERVER *biochem_lab*

Nombre que ha asignado a la definición de servidor de archivo con estructura de tabla en la sentencia CREATE SERVER.

FILE_PATH *'/usr/pat/DRUGDATA1.TXT'*

Especifica la vía de acceso de directorios completa y el nombre de archivo del archivo con estructura de tabla que contiene los datos a los que desea acceder. Debe especificar la opción de apodo FILE_PATH o DOCUMENT en la sentencia CREATE NICKNAME. La vía de acceso debe escribirse entre comillas simples.

COLUMN_DELIMITER *','*

Especifica el delimitador que se utiliza para separar los campos en un archivo con estructura de tabla. El valor del delimitador debe escribirse entre comillas simples. El delimitador de columna puede tener más de un carácter de longitud. Si no especifica un delimitador de columna, el delimitador predeterminado es una coma. No se puede utilizar una comilla simple como delimitador. El delimitador de columna debe ser coherente en todo el archivo. Un valor nulo se representa mediante dos delimitadores juntos o mediante un delimitador seguido de un terminador de línea, si el campo NULL es el último en la línea. El delimitador de columna no puede existir como datos válidos para una columna. Por ejemplo, no puede utilizarse una coma como delimitador de columna si una de las columnas contiene datos con comas incorporadas.

SORTED *'Y'*

Especifica que el archivo de origen de datos está ordenado. Los orígenes de datos ordenados deben ir en orden ascendente según la secuencia de ordenación del entorno local actual, tal como se define en los valores de la categoría del soporte multilingüístico LC_COLLATE. Si especifica que el origen de datos está ordenado, establezca la opción VALIDATE_DATA_FILE como 'Y'. El valor predeterminado del parámetro SORTED es 'N'.

KEY_COLUMN *'DCODE'*

Nombre de la columna en el archivo que forma la clave con la que se ordena el archivo. El valor de la columna de clave debe escribirse entre comillas simples. Especifique esta opción sólo si ha especificado la opción de apodo SORTED. No debe especificar como columna de clave una columna designada con la opción de columna de apodo DOCUMENT. El valor debe ser el nombre de una columna definida en la sentencia CREATE NICKNAME. La columna de clave no puede contener valores nulos. Sólo se da soporte a las claves de columna individuales. No se permiten varias claves de columna. La columna debe ir en orden ascendente. Si no se especifica el valor para un apodo ordenado, de forma predeterminada toma el valor de la primera columna en el archivo de apodo.

VALIDATE_DATA_FILE *'Y'*

Especifica si el derivador comprueba que la columna de clave está en orden ascendente y comprueba la existencia de claves nulas. Los valores

válidos para esta opción son Y o N. La opción no está permitida si se utiliza la opción de columna de apodo DOCUMENT como vía de acceso de archivo.

Designación de columnas de clave cuando registra un apodo

Puede designar una columna de clave especificando la restricción NOT NULL en la sentencia de apodo:

```
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

Especifica que la columna no puede contener valores nulos o blancos.

El derivador no impone la restricción NOT NULL, pero la base de datos federada sí. Si crea un apodo y adjunta una restricción NOT NULL en una columna y selecciona una fila que contiene un valor nulo para la columna, la base de datos federada emite un error SQL0407N indicando que no puede asignar un valor NULL a una columna NOT NULL.

La excepción a esta regla son los apodos ordenados. La columna de clave de los apodos ordenados no puede ser NULL. Si se encuentra una columna de clave NULL para un apodo ordenado, se emite el error SQL1822N, indicando que falta la columna de clave.

Nombres de columna sensibles a las mayúsculas y minúsculas

La base de datos federada cambia los nombres de columna a mayúscula, a menos que la columna esté definida con comillas dobles. El siguiente ejemplo no funcionará correctamente porque el valor de la opción KEY_COLUMN está entre comillas simples. En el ejemplo, la base de datos federada convertirá el nombre de columna a EMPNO. Como resultado, cuando especifique empno en una consulta, la base de datos federada no reconocerá la columna.

```
CREATE NICKNAME depart (
  empno char(6) NOT NULL)
FOR SERVER DATASTORE
  OPTIONS(FILE_PATH'data.txt', SORTED 'Y', KEY_COLUMN 'empno');
```

Servidores federados de Windows 2003

Si intenta acceder a orígenes de datos de archivos con estructura de tabla que están en una unidad compartida desde un servidor federado que ejecuta Windows 2003, la consulta puede fallar con el siguiente mensaje de error:

```
SQL1822N Se
ha recibido un código de error inesperado "ERRNO = 2" del origen de datos "SERVERNAME1".
El texto y las señales asociadas son "No se puede leer el archivo".
SQLSTATE=560BD
```

Esta es una limitación de Windows 2003. Para evitar este problema, especifique la vía de acceso absoluta en la opción FILE_PATH de la sentencia CREATE NICKNAME.

El siguiente ejemplo muestra una sentencia CREATE NICKNAME con una vía de acceso abreviada especificada en la opción FILE_PATH:

```
CREATE NICKNAME apodo
(
  COL1 CHAR (10) NOT NULL
)
FOR SERVER nombreservidor1
OPTIONS (FILE_PATH 'X:\textfile1.txt');
```

donde X:\ es la unidad que se correlaciona con la máquina remota. Las consultas que utilizan este apodo pueden fallar porque ha especificado la vía de acceso abreviada.

Para el servidor federado que ejecuta Windows 2003, especifique la vía de acceso absoluta en la opción FILE_PATH de la sentencia CREATE NICKNAME.

Por ejemplo:

```
CREATE NICKNAME
apodo
(
  COL1 CHAR (10) NOT NULL
)
FOR SERVER nombreservidor1
OPTIONS (FILE_PATH '\\host.svl.ibm.com\D$\textfile1.txt') ;
```

Modelo de control de accesos de archivo para el derivador de archivos con estructura de tabla

El derivador accede a los archivos con estructura de tabla utilizando la información de autorización del propietario de la instancia de la base de datos federada. El derivador sólo puede acceder a los archivos que se pueden leer con este ID de usuario o ID de grupo. El ID de autorización que establece la conexión con la base de datos federada no se utiliza para acceder a los archivos con estructura de tabla.

En un servidor federado, un archivo con estructura de tabla para el que se ha creado un apodo debe ser accesible con la misma vía de acceso desde cada nodo. El archivo no tiene que estar en un nodo de la base de datos federada siempre que se pueda acceder a él desde cualquier nodo con una vía de acceso común.

Para acceder a un archivo con estructura de tabla, el derivador necesita una identidad de usuario a efectos de seguridad. El derivador de archivos con estructura de tabla utiliza la identidad de usuario asociada con el servicio de base de datos federada. El nombre del servicio de base de datos federada depende del nombre de la instancia de la base de datos. Por ejemplo, si el nombre de la instancia de la base de datos es DB2, el nombre del servicio es DB2 - DB2. Para determinar la identidad de usuario asociada con un servicio de base de datos federada, utilice el Panel de control de Windows para mostrar los servicios. Efectúe una doble pulsación en el nombre de servicio y muestre la página de propiedades de Iniciar sesión.

Archivos con estructura de tabla ubicados en unidades remotas

Los archivos con estructura de tabla a los que desea acceder deben estar en una unidad local o correlacionada.

Redes que tienen un dominio Windows configurado

La cuenta de inicio de sesión del servicio de base de datos federada debe ser una cuenta del dominio que tenga acceso a la carpeta compartida en la unidad correlacionada donde residen los archivos con estructura de tabla.

Redes que no tienen un dominio Windows configurado

La cuenta de inicio de sesión del servicio de base de datos federada debe tener el mismo nombre de usuario y la misma contraseña que un usuario válido en el sistema que comparte la carpeta. Dicho usuario debe estar en la lista de permisos de la carpeta compartida con al menos acceso de lectura.

Directrices para optimizar el rendimiento de las consultas para el derivador de archivos con estructura de tabla

Para aumentar el rendimiento de las consultas de archivos con estructura de tabla, tenga archivos que estén ordenados y cree estadísticas para los apodos.

Utilice las siguientes sugerencias para mejorar el rendimiento de las consultas:

- Ordene los datos en los archivos. El servidor federado puede realizar búsquedas en los archivos que están ordenados mejor que en los archivos que no lo están.
- Para los archivos ordenados, especifique un valor o un rango para la columna de clave cuando someta una consulta.
- Las estadísticas de los apodos de los archivos con estructura de tabla deben actualizarse manualmente actualizando las vistas SYSSTAT y SYSCAT. Utilice el recurso de actualización de las estadísticas de apodos para actualizar las estadísticas de los apodos de archivos con estructura de tabla.

Configuración del acceso a orígenes de datos Teradata

Para configurar un servidor federado para acceder a orígenes de datos Teradata, debe proporcionar al servidor federado información acerca de los orígenes de datos y los objetos a los que desea acceder.

Antes de empezar

- El software de cliente Teradata debe estar instalado y configurado en el servidor que actúa como el servidor federado.
- Federation debe estar instalado en el servidor que actúa como servidor federado.
- Compruebe la configuración del servidor federado.
- Compruebe el parámetro federado para asegurarse de que la federación está habilitada.

Acerca de esta tarea

Puede configurar un servidor federado para acceder a los datos almacenados en los orígenes de datos Teradata utilizando el Centro de control de DB2 o emitiendo sentencias SQL en la línea de mandatos de DB2. El Centro de control de DB2 incluye un asistente que sirve de guía en los pasos necesarios para configurar los objetos federados que se necesitan.

Procedimiento

Para configurar el acceso a los orígenes de datos Teradata:

1. Pruebe la conexión con el servidor Teradata.
2. Compruebe que la biblioteca de Teradata esté habilitada para los enlaces de tiempo de ejecución (AIX).
3. Establezca las variables de entorno para el derivador de Teradata.
4. Registre el derivador.

5. Registre la definición de servidor.
6. Cree las correlaciones de usuarios.
7. Pruebe la conexión con el servidor Teradata.
8. Registre los apodos de las vistas y tablas de Teradata.

Prueba de la conexión al servidor Teradata

Pruebe la conexión con el servidor Teradata para verificar que el software del cliente Teradata está configurado correctamente en el servidor federado.

Antes de empezar

El programa de utilidad Basic Teradata Query (BTEQ) y la Interfaz de programación de aplicaciones del conector de datos Teradata (PIOM) deben estar instalados en el servidor federado. El programa de utilidad BTEQ y la Interfaz de programación de aplicaciones del conector de datos Teradata se instalan en el servidor federado cuando instala el software del cliente Teradata.

Acerca de esta tarea

Utilice el programa de utilidad BTEQ para someter una consulta SQL para comprobar que el servidor federado puede conectarse con el servidor Teradata. Consulte la documentación de Teradata para obtener más información sobre el programa de utilidad BTEQ.

Procedimiento

Para probar la conexión al servidor Teradata:

1. Inicie una sesión en el programa de utilidad BTEQ y conéctese al servidor Teradata.
2. Emita un mandato SQL para comprobar que puede conectarse correctamente al servidor Teradata.

Por ejemplo:

```
select count(*) from dbc.tables;
```

Si la conexión es satisfactoria, podrá ver la salida de la consulta.

Si la conexión no es satisfactoria, recibirá un error. Compruebe el software del cliente Teradata para verificar que está instalado y configurado correctamente en el servidor federado.

3. Desconéctese del servidor Teradata y cierre la sesión del programa de utilidad BTEQ.

Cuando finalice esta tarea, compruebe que la biblioteca de Teradata esté habilitada para los enlaces de tiempo de ejecución.

Comprobación de que la biblioteca de Teradata esté habilitada para los enlaces de tiempo de ejecución (AIX)

Cuando añade un origen de datos Teradata al servidor federado en AIX, debe comprobar que los enlaces de tiempo de ejecución están habilitados antes de registrar derivadores o servidores.

Procedimiento

Para comprobar que la biblioteca de Teradata esté habilitada para los enlaces de tiempo de ejecución:

1. Vaya al directorio donde reside el archivo `libcliv2.so`.

El archivo `libcliv2.so` se instala con el software del cliente Teradata. De forma predeterminada, se instala en el directorio `/usr/lib`.

2. En un indicador de mandatos, emita el siguiente mandato UNIX para comprobar que los enlaces de tiempo de ejecución están habilitados:

```
dump -H libcliv2.so | grep libtli.a
```

3. Compruebe los nombres de archivo que se devuelven. Si se devuelve el nombre de archivo `libtli.a`, la biblioteca de Teradata está habilitada para los enlaces de tiempo de ejecución.

Si no se devuelve el nombre de archivo `libtli.a`, abra una ventana de mandatos y emita los siguientes mandatos UNIX para habilitar los enlaces de tiempo de ejecución para la biblioteca de 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
```

Cuando finalice esta tarea, puede establecer las variables de entorno.

Establecimiento de las variables de entorno de Teradata

Las variables de entorno de Teradata deben establecerse en el archivo `db2dj.ini` en el servidor federado.

Restricciones

Revise las restricciones del archivo `db2dj.ini`.

El archivo `db2dj.ini` contiene información de configuración sobre el software de cliente Teradata que hay instalado en el servidor federado.

Existen variables de entorno necesarias y opcionales para los orígenes de datos Teradata.

Si ha instalado el software de cliente Teradata antes de instalar el derivador de Teradata, las variables de entorno de Teradata necesarias se establecen en el archivo `db2dj.ini`.

Si no ha instalado el software de cliente Teradata antes de instalar el derivador de Teradata, o si desea establecer algunas de las variables de entorno opcionales, debe establecer las variables de entorno utilizando los pasos de esta tarea.

Procedimiento

Para establecer las variables de entorno de Teradata:

1. Utilice uno de los métodos siguientes:

Método	Paso
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .

Método	Paso
Establezca automáticamente las variables de entorno.	Ejecute el asistente de IBM InfoSphere Federation Server. Siga las instrucciones del asistente. Importante: Establezca las variables de entorno necesarias ejecutando el asistente de instalación. Las variables de entorno opcionales deben establecerse manualmente.
Establezca manualmente las variables de entorno.	<p>Edite el archivo db2dj.ini:</p> <p>El archivo db2dj.ini se encuentra en el directorio especificado por la variable de registro DB2_DJ_INI de DB2. Si la variable DB2_DJ_INI no se establece, el archivo db2dj.ini reside en una de las siguientes vías de acceso predeterminadas, dependiendo del sistema operativo:</p> <ul style="list-style-type: none"> En UNIX: <i>padre_instancia</i>/sql1lib/cfg/db2dj.ini. <i>padre_instancia</i> El directorio padre del propietario de la instancia En Windows: %DB2PATH%\cfg\db2dj.ini %DB2PATH% El directorio donde está instalado el sistema de base de datos de DB2, por ejemplo, C:\Archivos de programa\IBM\sql1lib. <p>Si el archivo no existe, puede crear un archivo con el nombre db2dj.ini utilizando un editor de texto. En el archivo db2dj.ini, debe especificar la vía de acceso completa en el valor de las variables de entorno; de lo contrario, aparecerán errores. Por ejemplo: COPLIB=/usr/lib</p>

- Establezca las variables de entorno de conversión de página de códigos de Teradata (según sea necesario).
- Para asegurarse de que las variables de entorno están establecidas en el servidor federado, recicle la instancia de la base de datos federada con estos mandatos:

```
db2stop
db2start
```

Cuando finalice esta tarea, puede registrar el derivador.

Variables de entorno de Teradata

Existen variables de entorno necesarias y variables de entorno opcionales para orígenes de datos Teradata. Estas variables se definen en el archivo db2dj.ini.

Las variables de entorno válidas para Teradata son:

- COPLIB
- COPERR
- TERADATA_CHARSET (opcional)
- NETRACE (opcional)
- COPANOMLOG (opcional)

Descripción de las variables

COPLIB

Especifica la vía de acceso de directorio en el servidor federado para el archivo `libcliv2.so`. Especifique la vía de acceso completa para la variable `COPLIB`.

Por ejemplo:

```
COPLIB=/usr/lib
```

Los archivos `libcliv2.so` y `errmsg.cat` residen generalmente en el mismo directorio.

COPERR

Especifica la vía de acceso de directorio en el servidor federado para el archivo `errmsg.cat`. Especifique la vía de acceso completa para la variable `COPERR`.

Por ejemplo:

```
COPERR=/usr/lib
```

TERADATA_CHARSET

Especifica el juego de caracteres de página de códigos que debe utilizarse con los orígenes de datos Teradata.

Cada vez que el servidor federado se conecta con un origen de datos Teradata, el derivador Teradata determina qué juego de caracteres de página de códigos debe utilizarse para dicha conexión. Puede hacer que el derivador Teradata establezca el juego de caracteres de página de códigos o puede designar una página de códigos mediante la variable de entorno `TERADATA_CHARSET`.

Si la variable de entorno `TERADATA_CHARSET` está establecida en el archivo `db2dj.ini` del servidor federado, el derivador utilizará el juego de caracteres de página de códigos del archivo `db2dj.ini`. El valor de la variable de entorno `TERADATA_CHARSET` no se valida; no obstante, si la variable de entorno se establece en un valor que no es válido, el origen de datos Teradata devuelve un error.

Si la variable de entorno `TERADATA_CHARSET` no se establece en el archivo `db2dj.ini` del servidor federado, el derivado detecta el juego de caracteres del cliente en función de la página de códigos de la base de datos.

En servidores federados que ejecutan UNIX, los valores siguientes son válidos para la variable de entorno `TERADATA_CHARSET`:

- `HANGULKSC5601_2R4`
- `KanjiEUC_0U`
- `LATIN1_0A`
- `LATIN9_0A`
- `LATIN1252_0A`
- `SCHGB2312_1T0`
- `TCHBIG5_1R0`
- `UTF8`
- `ASCII`

En servidores federados que ejecutan Windows, los valores siguientes son válidos para la variable de entorno `TERADATA_CHARSET`:

- `HANGULKSC5601_2R4`

- KanjiSJIS_0S
- LATIN1_0A
- LATIN1252_0A
- SCHGB2312_1T0
- TCHBIG5_1R0
- UTF8
- ASCII

NETRACE

Opcional. Habilita la función de rastreo del software de cliente Teradata. Esta variable sólo es necesaria para depuración.

COPANOMLOG

Opcional. Habilita la función de registro para el software de cliente Teradata. Esta variable sólo es necesaria para depuración.

Verificación del juego de caracteres en el servidor Teradata

Si el juego de caracteres correcto no está especificado en el servidor Teradata, es posible que reciba errores de conexión. Verifique que el juego de caracteres que desea utilizar está instalado en el servidor Teradata.

Procedimiento

Para verificar que el juego de caracteres que desea utilizar está instalado en el servidor Teradata:

1. Inicie sesión en el servidor Teradata mediante el programa de utilidad BTEQ o cualquier otro programa de utilidad de inicio de sesión válido.
2. Emita la siguiente sentencia para visualizar la tabla dbc.chartranslations: `select * from dbc.chartranslations;`
3. Compruebe el valor que se devuelve en la tercera columna, InstallFlag, de la tabla. El valor 'Y' en la tercera columna indica que el juego de caracteres está instalado y se encuentra en uso en el servidor Teradata.

Utilice la tabla siguiente para determinar si tiene el juego de caracteres correcto instalado:

Tabla 38. Juegos de caracteres para Teradata

Juego de caracteres de doble byte	Juego de caracteres de un byte	Juego de caracteres Teradata	Idioma	Juego de caracteres IBM DB2
941	897	"KanjiSJIS_0S"	Japonés	IBM-943
1362	1126	"HANGULKSC5601_2R4"	Coreano	1363
1385	1114	"SCHGB2312_1T0"	Chino simplificado	GBK
380	1115	"SCHGB2312_1T0"	Chino simplificado	IBM-1381
947	1114	"TCHBIG5_1R0"	Chino tradicional	big5
1200	1208	"UTF8"	Unicode	UTF-8
0	819	"Latin1_0A"	Inglés (Latin 1)	ISO8859-1
0	1252	"Latin1252_0A"	Inglés (Latin Win)	ISO8859-1/15
0	819	"ASCII"	Inglés (ASCII)	ISO8859-1

4. Si no tiene instalado el juego de caracteres necesario, instale el juego de caracteres para utilizar el derivador Teradata.

- ASCII está habilitado en el servidor Teradata, pero no está catalogado en la tabla `dbc.chartranslations`. Si todos los valores que se devuelven para `InstallFlag` son 'N', ASCII es el único juego de caracteres válido en el servidor Teradata y la variable de entorno `TERADATA_CHARSET` debe establecerse en ASCII en el archivo `db2dj.ini`.
 - Si el juego de caracteres que desea utilizar aparece en la tabla `dbc.chartranslations`, pero el valor `InstallFlag` está establecido en 'N', emita la siguiente sentencia para cambiar el valor de `InstallFlag` por 'Y':

```
update dbc.chartranslations
      set installflag='Y' where CharSetName= 'nombre_juego_caracteres';
```
 - Si el juego de caracteres que desea utilizar no aparece en la tabla `dbc.chartranslations`, póngase en contacto con el servicio de atención al cliente de Teradata.
5. Reinicie el servidor Teradata para actualizar la lista de juegos de caracteres. En una ventana de mandatos de Teradata, especifique lo siguiente: `tpareset -f reason_for_restart`

Resolución de problemas de juegos de caracteres para los orígenes de datos Teradata:

Cuando establece la variable de entorno `TERADATA_CHARSET` para un origen de datos Teradata, pueden aparecer errores si no se especifica el juego de caracteres correcto.

Síntoma

Si no se especifica el juego de caracteres correcto para el origen de datos Teradata, aparecerá el siguiente error:

```
SQL
1822N Se ha recibido un código de error inesperado "227"
del origen de datos "<string>". El texto y las señales
asociadas son "MTDP: EM_CHARNAME(227): nombre de juego de caracteres
no válido especific". SQLSTATE=560BD
```

Causa

El juego de caracteres especificado en la variable de entorno `TERADATA_CHARSET` no es correcto.

Resolución del problema

Compruebe que se haya instalado y especificado el juego de caracteres correcto en el servidor Teradata y en el archivo `db2dj.ini`.

Registro del derivador de Teradata

Debe registrar un derivador para acceder a los orígenes de datos Teradata. Los servidores federados utilizan derivadores para comunicarse con los orígenes de datos y recuperar datos de éstos. Los derivadores se implementan como un conjunto de archivos de biblioteca.

Procedimiento

Para registrar el derivador de Teradata:

Utilice uno de los métodos siguientes:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Emita la sentencia CREATE WRAPPER y especifique el nombre predeterminado del derivador de Teradata.	<p>Por ejemplo:</p> <pre>CREATE WRAPPER TERADATA;</pre> <p>Recuerde: Cuando registra el derivador utilizando el nombre predeterminado, TERADATA, el servidor federado utiliza automáticamente la biblioteca de derivador de Teradata correspondiente al sistema operativo donde se ejecuta el servidor federado.</p> <p>Si el nombre del derivador predeterminado entra en conflicto con un nombre de derivador existente en la base de datos federada, puede sustituir el nombre del derivador predeterminado por otro de su elección. Si utiliza un nombre distinto al nombre predeterminado, deberá incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER.</p> <p>Por ejemplo, para registrar un derivador con el nombre <code>tera_wrapper</code> en un servidor federado que utiliza AIX, emita la siguiente sentencia:</p> <pre>CREATE WRAPPER tera_wrapper LIBRARY 'libdb2teradata.a' OPTIONS (DB2_FENCED 'Y');</pre> <p>El archivo de biblioteca de derivador que especifica depende del sistema operativo del servidor federado.</p> <p>Opciones obligatorias para AIX y Solaris Nota: La opción de derivador DB2_FENCED en la sentencia CREATE WRAPPER es obligatoria para AIX y Solaris porque el derivador de Teradata sólo da soporte a clientes de 32 bits.</p> <p>Por ejemplo:</p> <pre>CREATE WRAPPER TERADATA OPTIONS (DB2_FENCED 'Y')</pre>

Cuando finalice esta tarea, puede registrar la definición de servidor.

Archivos de biblioteca de derivador de Teradata

Los archivos de biblioteca de derivador de Teradata se añaden al servidor federado cuando se instala el derivador.

Cuando instala el derivador de Teradata, se añaden tres archivos de biblioteca a la vía de acceso de directorios predeterminada. Por ejemplo, si el servidor federado se ejecuta en AIX, los archivos de biblioteca que se añaden a la vía de acceso de directorios son `libdb2teradata.a`, `libdb2teradataF.a` y `libdb2teradataU.a`. El archivo de biblioteca del derivador predeterminado es `libdb2teradata.a`. El derivador de Teradata utiliza internamente los otros archivos de biblioteca de derivador.

Si no utiliza el nombre de derivador predeterminado cuando registra el derivador, debe incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER y especificar el nombre de archivo de biblioteca del derivador predeterminado.

Las vías de acceso de directorios predeterminadas y los nombres de archivo de biblioteca del derivador predeterminados se especifican en la tabla siguiente.

Tabla 39. Nombres de archivo y ubicaciones de la biblioteca de derivador de Teradata

Sistema operativo	Vía de acceso de directorios	Nombres de archivo de biblioteca
AIX	/usr/opt/vía_acceso_instalación/lib32/ /usr/opt/vía_acceso_instalación/lib64/	libdb2teradata.a
Solaris	/opt/IBM/db2/vía_acceso_instalación/lib32 /opt/IBM/db2/vía_acceso_instalación/lib64	libdb2teradata.so
Windows	%DB2PATH%\bin	db2teradata.dll

vía_acceso_instalación es la vía de acceso de directorios donde se instala el servidor federado en UNIX.

Registro de las definiciones de servidor para un origen de datos Teradata

Debe registrar cada servidor Teradata al que desee acceder en la base de datos federada.

Procedimiento

Para registrar una definición de servidor para un origen de datos Teradata:

1. Ubique el archivo hosts.
 - En los servidores federados que ejecutan AIX, el archivo hosts se encuentra en el directorio /etc/hosts.
 - En los servidores federados que ejecutan Windows, el archivo hosts se encuentra en el directorio %WINDIR%\system32\drivers\etc\hosts.
2. Busque en el archivo hosts el alias del servidor remoto.
Este alias empieza con una serie alfabética y termina con el sufijo COP n . El valor n es el número del procesador de aplicaciones asociado con el procesador de comunicaciones de Teradata.
3. Busque el primer campo no numérico en la línea de hosts que contiene el alias.

Archivo hosts de ejemplo:

```
127.0.0.1      localhost

9.22.5.77     nodexyz      nodexyzCOP1   # servidor teradata

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
```

En este ejemplo, nodexyz es el nombre de nodo.

4. Para crear el servidor, utilice uno de los métodos siguientes:

Métodos	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Emita la sentencia CREATE SERVER.	<p>Por ejemplo:</p> <pre>CREATE SERVER nombre_definición_servidor TYPE TERADATA VERSION númeroversión WRAPPER nombre_derivador OPTIONS (NODE 'nombre_nodo');</pre> <p>Aunque la variable 'nombre_nodo' se especifica como una opción en la sentencia CREATE SERVER, esta opción se necesita para los orígenes de datos Teradata.</p>

Sentencia CREATE SERVER - Ejemplos para el derivador de Teradata

Utilice la sentencia CREATE SERVER para registrar definiciones de servidor para el derivador de Teradata. Este tema incluye un ejemplo completo con los parámetros necesarios y un ejemplo con opciones adicionales del servidor.

El ejemplo siguiente muestra cómo registrar una definición de servidor para un derivador de Teradata emitiendo la sentencia CREATE SERVER:

```
CREATE SERVER
servidor_tera TYPE TERADATA
  VERSION 2.5 WRAPPER mi_derivador
  OPTIONS (NODE 'nodo_tera');
```

servidor_tera

Nombre que asigna al servidor de bases de datos Teradata. No se permiten los nombres de definición de servidor duplicados.

TYPE TERADATA

Especifica el tipo de servidor de origen de datos para el que está configurando el acceso. Para el derivador de Teradata, el tipo de servidor debe ser TERADATA.

VERSION 2.5

Versión del servidor de bases de datos Teradata al que desea acceder.

WRAPPER TERADATA

Nombre de derivador que ha especificado en la sentencia CREATE WRAPPER.

NODE 'nodo_tera'

Nombre del nodo donde reside el servidor de bases de datos Teradata. Obtenga el nombre de nodo del archivo hosts. Este valor es sensible a las mayúsculas y minúsculas.

Aunque el nombre del nodo se especifica como una opción en la sentencia CREATE SERVER, se necesita para los orígenes de datos Teradata.

Opciones de servidor

Cuando crea una definición de servidor, puede especificar opciones de servidor adicionales en la sentencia CREATE SERVER. Las opciones de servidor pueden ser opciones de servidor generales y opciones de servidor específicas de Teradata.

Las opciones de servidor CPU_RATIO y IO_RATIO proporcionan información estadística sobre el servidor Teradata al optimizador de consultas. Para especificar que los recursos de la CPU del servidor federado tienen el doble de potencia que los recursos de la CPU del servidor Teradata, establezca el valor de la opción de servidor CPU_RATIO en 2.0. Para especificar que los dispositivos de E/S del servidor federado procesan datos tres veces más rápido que los dispositivos de E/S del servidor Teradata, establezca la opción de servidor IO_RATIO en 3.0.

El siguiente ejemplo muestra una definición de servidor Teradata con estas opciones:

```
CREATE SERVER servidor_tera TYPE TERADATA
  VERSION 2.5 WRAPPER mi_derivador
  OPTIONS (NODE 'nodo_tera', CPU_RATIO '2.0',
  IO_RATIO '3.0');
```

Creación de la correlación de usuarios para un origen de datos Teradata

Cuanto intenta acceder a un servidor Teradata, el servidor federado establece una conexión con el servidor Teradata utilizando un ID de usuario y una contraseña que son válidos para dicho origen de datos.

Acerca de esta tarea

Cree una correlación de usuarios para cada ID de usuario que vaya a acceder al sistema federado para enviar solicitudes distribuidas al origen de datos Teradata.

Procedimiento

Para crear las correlaciones de usuario para un origen de datos Teradata:

Emita una sentencia CREATE USER MAPPING.

Por ejemplo:

```
CREATE USER MAPPING FOR IDusuario_local SERVER nombre_definición_servidor
  OPTIONS (REMOTE_AUTHID 'IDusuario_remoto', REMOTE_PASSWORD 'contraseña_remota')
```

Aunque las variables REMOTE_AUTHID y REMOTE_PASSWORD se especifican como opciones en la sentencia CREATE USER MAPPING, estas opciones son necesarias para acceder a los orígenes de datos Teradata.

Cuando finalice esta tarea, pruebe la conexión desde el servidor federado al servidor Teradata.

Sentencia CREATE USER MAPPING - Ejemplos para el derivador de Teradata

Utilice la sentencia CREATE USER MAPPING para correlacionar un ID de autorización de servidor federado con un ID de usuario y una contraseña remotos de Teradata. En este tema se proporciona un ejemplo completo con los parámetros necesarios y un ejemplo que muestra cómo utilizar el registro especial USER de DB2 con la sentencia CREATE USER MAPPING.

El ejemplo siguiente muestra cómo correlacionar un ID de autorización local con un ID de usuario y una contraseña remotos de Teradata:

```
CREATE USER  
MAPPING FOR MICHAEL SERVER servidor_tera  
OPTIONS (REMOTE_AUTHID 'mike', REMOTE_PASSWORD 'passxyz123');
```

MICHAEL

Especifica el ID de autorización local que está correlacionando con el ID de usuario y la contraseña remotos definidos en el servidor Teradata.

SERVER *servidor_tera*

Especifica el nombre de definición de servidor que ha registrado en la sentencia CREATE SERVER del servidor Teradata.

REMOTE_AUTHID *'mike'*

Especifica el ID de usuario remoto de Teradata con el que está correlacionando *MICHAEL*. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_ID en 'U' o 'L' en la sentencia CREATE SERVER.

Aunque el ID de usuario remoto se especifica como una opción en la sentencia CREATE SERVER, se necesita para los orígenes de datos Teradata.

REMOTE_PASSWORD *'passxyz123'*

Especifica la contraseña remota de Teradata asociada a *'mike'*. El valor es sensible a las mayúsculas y minúsculas, a menos que establezca la opción de servidor FOLD_PW en 'U' o 'L' en la sentencia CREATE SERVER.

Aunque la contraseña remota se especifica como una opción en la sentencia CREATE SERVER, se necesita para los orígenes de datos Teradata.

Registro especial USER de DB2

Puede utilizar el registro especial USER de DB2 para correlacionar el ID de autorización de la persona que emite la sentencia CREATE USER MAPPING con el ID de autorización de origen de datos especificado en la opción de usuario REMOTE_AUTHID.

El siguiente ejemplo muestra una sentencia CREATE USER MAPPING que incluye el registro especial USER:

```
CREATE USER MAPPING  
FOR USER SERVER servidor_tera  
OPTIONS (REMOTE_AUTHID 'mike', REMOTE_PASSWORD  
'passxyz123');
```

Prueba de la conexión al servidor Teradata

Pruebe la conexión con el servidor de origen de datos Teradata para determinar si el servidor federado se ha configurado correctamente para acceder a los orígenes de datos Teradata.

Acerca de esta tarea

Puede probar la conexión con el servidor Teradata utilizando la definición de servidor y las correlaciones de usuario que ha definido.

Procedimiento

Para probar la conexión al servidor Teradata:

Abra una sesión de paso a través y emita una sentencia SELECT en las tablas del sistema Teradata. Si la sentencia SELECT devuelve un recuento, la definición de servidor y la correlación de usuarios se han configurado correctamente.

Por ejemplo:

```
SET PASSTHRU nombre_definición_servidor
SELECT count(*) FROM dbc.tables
SET PASSTHRU RESET
```

Si la sentencia SELECT devuelve un error, debe resolver los errores de conexión.

Cuando finalice esta tarea, puede registrar apodos para las tablas y vistas de Teradata.

Resolución de errores de conexión de origen de datos

Una conexión de prueba con el servidor de origen de datos puede devolver un error por varios motivos. Existen acciones que permiten determinar el motivo del error.

Síntoma

Se devuelve un error al intentar establecer conexión con el origen de datos.

Causa

Hay varias causas posibles que explican un problema de conexión.

Resolución del problema

Para solucionar los problemas de errores de conexión con el origen de datos, consulte los siguientes elementos:

- Verifique que el origen de datos está disponible.
- Si procede, asegúrese de que el servidor de origen de datos esté configurado para conexiones entrantes.
- Asegúrese de que los valores de correlación de usuarios para las opciones REMOTE_AUTHID y REMOTE_PASSWORD son válidos para las conexiones con el origen de datos. Modifique la correlación de usuarios o cree otra correlación de usuarios según sea necesario.
- Si procede, asegúrese de que el software de cliente del servidor federado esté instalado y configurado correctamente para conectarse al origen de datos.
- Para orígenes de datos ODBC, asegúrese de que el controlador ODBC del servidor federado esté instalado y configurado correctamente para establecer conexión con el servidor de origen de datos ODBC. En servidores federados que ejecutan Windows, utilice la herramienta ODBC Data Source Administrator para verificar el controlador. En servidores federados que ejecutan UNIX, consulte la documentación del proveedor del cliente ODBC.
- Verifique que los valores para las variables establecidas en el servidor federado son correctas para el origen de datos. Estas variables incluyen las variables de entorno del sistema, las variables de la archivo db2dj.ini y las variables del registro de perfiles de DB2 (db2set).
- Compruebe la definición del servidor. Si es necesario, elimine la definición del servidor y créela de nuevo.

Registro de apodos para vistas y tablas de Teradata

Para cada definición de servidor Teradata, debe registrar un apodo para cada tabla o vista a la que desee acceder. Utilice estos apodos en lugar de los nombres de los objetos de origen de datos cuando consulte los servidores Teradata.

Antes de empezar

La base de datos federada se basa en las estadísticas de catálogo de origen de datos para optimizar el proceso de las consultas. Para asegurarse de que la base de datos federada tiene estadísticas completas sobre las tablas de Teradata, utilice el mandato COLLECT STATISTICS de Teradata antes de registrar un apodo.

En el servidor Teradata, utilice el mandato COLLECT STATISTICS de Teradata para recopilar estadísticas sobre una o varias columnas o índices en una tabla.

Cuando registra el apodo con la sentencia CREATE NICKNAME, la base de datos federada lee las estadísticas en el catálogo del sistema Teradata y actualiza las estadísticas locales para el apodo.

Acerca de esta tarea

Cuando registra un apodo en una vista de Teradata, la base de datos federada reconoce todas las columnas de la vista como anulables, aunque las columnas de la vista de Teradata no permitan valores nulos. No existe ningún método alternativo para esta limitación.

Procedimiento

Para registrar un apodo para una tabla o vista de Teradata:

Método	Descripción
Utilice el asistente de Objetos federados en el centro de control de DB2 Universal Database.	Para iniciar el asistente, pulse con el botón derecho del ratón en la carpeta Objetos de base de datos federados y pulse Crear objetos federados .
Emita la sentencia CREATE NICKNAME. Los apodos pueden tener una longitud de hasta 128 caracteres.	Por ejemplo: <pre>CREATE NICKNAME apodo FOR nombre_definición_servidor.esquema_remoto.tabla_remota;</pre>

Cuando se crea el apodo, el servidor federado consulta el catálogo de orígenes de datos utilizando el apodo. Esta consulta prueba la conexión con la tabla o la vista del origen de datos. Si la conexión no funciona, recibirá un mensaje de error.

Repita este paso para cada tabla o vista de Teradata para la que desee crear un apodo.

Apodos Teradata en servidores federados

Al consultar un origen de datos Teradata desde un servidor federado, debe utilizar un apodo en la consulta para identificar la tabla y vista Teradata a las que desea acceder.

Al crear un apodo para una tabla o vista Teradata, el servidor federado se conecta al servidor Teradata asociado con la tabla o vista. El servidor federado utiliza el apodo para verificar la conexión con el servidor Teradata. La base de datos federada verifica la presencia de la tabla o vista en el origen de datos e intenta recopilar datos estadísticos sobre la tabla o vista Teradata del catálogo del servidor Teradata. Las estadísticas recopiladas sobre el objeto con apodo se almacenan en el catálogo global del servidor federado.

El servidor federado se basa en las estadísticas que recopila para los objetos con apodo para optimizar el proceso de las consultas. Dado que toda la información del catálogo Teradata o parte de ella puede ser utilizada por el optimizador de consultas, debe actualizar las estadísticas en el servidor Teradata antes de crear un apodo. Para actualizar las estadísticas en el servidor Teradata, utilice un mandato o un programa de utilidad que sea equivalente al mandato DB2 RUNSTATS.

No puede emitir una sentencia INSERT, UPDATE o DELETE para un apodo que hace referencia a una vista Teradata actualizable aménos que la sentencia SQL pueda enviarse por completo al origen de datos Teradata.

Sentencia CREATE NICKNAME - Ejemplos para el derivador de Teradata

Utilice la sentencia CREATE NICKNAME para registrar un apodo para una tabla o vista de Teradata a la que desea acceder. Este tema incluye un ejemplo completo con los parámetros necesarios.

Este ejemplo muestra cómo crear un apodo para una tabla o vista de Teradata en el servidor Teradata:

```
CREATE NICKNAME TERASALES FOR servidor_tera.vinnie.europe ;  
TERASALES
```

Apodo exclusivo que se utiliza para identificar la tabla o la vista de Teradata.

Importante: El nombre de apodo consta de dos partes: el esquema y el apodo. Si omite el esquema al registrar el apodo, el esquema del apodo será el ID de autorización del usuario que registra el apodo. El ID de autorización es para el servidor federado, no para el origen de datos Teradata remoto.

servidor_tera.vinnie.europe

Es un identificador de tres partes para el objeto remoto:

- *servidor_tera* es el nombre de definición de servidor que ha asignado al servidor de bases de datos Teradata en la sentencia CREATE SERVER.
- *vinnie* es el ID de usuario del propietario al que pertenece la tabla o la vista. Este valor es sensible a las mayúsculas y minúsculas.
- *europe* es el nombre de la tabla o la vista remota a la que desea acceder.

Resolución de problemas de la configuración de orígenes de datos Teradata

Habilitación de enlaces de tiempo de ejecución para libcliv2.so (AIX)

Si ejecuta el archivo `djxlinkTeradata.sh` para enlazar con la biblioteca compartida de Teradata denominada `libcliv2.so`, puede recibir un mensaje de error cuando emite una sentencia CREATE NICKNAME.

Síntoma

Un ejemplo de mensaje de error que puede recibir es:

```
DB21034E El mandato se ha procesado como una sentencia SQL porque no es un
mandato de procesador de línea de mandatos válido. Durante el proceso SQL, ha devuelto:
SQL30081N Se ha detectado un error de comunicación. Protocolo de comunicación que se
utiliza: "TCP/IP". API de comunicación que se utiliza: "SOCKETS". Ubicación donde se
ha detectado el error: "9.112.26.28". Función de comunicación que detecta
el error: "recv". Códigos de error específicos del protocolo: "*", "*", "0".
SQLSTATE=08001
```

Causa

La función `OsCall` ha hecho que se detenga el servidor federado.

Resolución del problema

Si recibe un mensaje de error, compruebe si existen archivos de captura en el directorio `/sqllib/db2dump`. Los nombres de los archivos de captura empiezan por la letra `t` y terminan con un sufijo `000`. Por ejemplo:

```
t123456.000
```

Consulte la información de rastreo en el archivo de captura y busque referencias de la función `OsCall` que indiquen que la función `OsCall` ha hecho que se detenga el servidor federado.

En el siguiente ejemplo se muestra información de rastreo con una referencia de función `OsCall` que puede encontrar en un archivo de captura:

```
*** Start stack traceback ***
```

```
0x239690E0 OsCall + 0x28C
0x23973FB0 mtdpassn + 0x8A4
0x239795A4 mtdp + 0x208
0x2395A928 MTDPIO + 0x28C
0x239609C4 CLICON + 0xD50
0x23962350 DBCHCL + 0xC4
```

Si encuentra una referencia de función `OsCall` en uno de los archivos de captura, debe habilitar los enlaces de tiempo de ejecución para la biblioteca compartida `libcliv2.so`.

Emita los siguientes mandatos UNIX para habilitar los enlaces de tiempo de ejecución:

```
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
```

Resolución de problemas de juegos de caracteres para los orígenes de datos Teradata

Cuando establece la variable de entorno `TERADATA_CHARSET` para un origen de datos Teradata, pueden aparecer errores si no se especifica el juego de caracteres correcto.

Síntoma

Si no se especifica el juego de caracteres correcto para el origen de datos Teradata, aparecerá el siguiente error:

```
SQL
1822N Se ha recibido un código de error inesperado "227"
del origen de datos "<string>". El texto y las señales
asociadas son "MTDP: EM_CHARNAME(227): nombre de juego de caracteres
no válido especific". SQLSTATE=560BD
```

Causa

El juego de caracteres especificado en la variable de entorno TERADATA_CHARSET no es correcto.

Resolución del problema

Compruebe que se haya instalado y especificado el juego de caracteres correcto en el servidor Teradata y en el archivo db2dj.ini.

Resolución de problemas de errores de operación de UPDATE o DELETE en los apodos

De forma predeterminada, las filas no se identifican de forma exclusiva en las tablas de los orígenes de datos Teradata. Puede recibir un mensaje de error SQL cuando intenta actualizar o suprimir un apodo asociado con una tabla o una vista de Teradata.

Síntoma

Se devuelve el error SQL30090N, RC="21" cuando intenta actualizar o suprimir un apodo asociado con una tabla o una vista de Teradata.

Causa

Este problema se debe a que las filas no se definen de forma exclusiva en la tabla del origen Teradata.

Resolución del problema

Para resolver el problema:

1. Elimine y vuelva a crear el apodo.
2. Cree al menos un índice exclusivo en la tabla de Teradata que se está actualizando o suprimiendo, y vuelva a intentar la operación.

Trabajo con el registro cronológico de acceso de Teradata

El producto Teradata proporciona una característica de registro cronológico de acceso que genera entradas de registro cuando Teradata comprueba los privilegios de seguridad específicos de varios usuarios en una o más bases de datos. Aunque el registro cronológico de acceso proporciona una gran cantidad información de seguridad significativa, esta característica aumenta considerablemente el uso del procesador y puede degradar el rendimiento del sistema.

Si necesita aumentar el rendimiento del sistema, evalúe las reglas de los privilegios de comprobación definidas para el registro cronológico de acceso. A continuación, termine las reglas que no sean necesarias mediante la definición de sentencias END LOGGING.

Para garantizar el máximo rendimiento, desactive todo el registro cronológico de acceso. Elimine la macro Teradata DBC.AccLogRules y, a continuación, fuerce un restablecimiento de la aplicación paralela de confianza (TPA) para detener el registro cronológico de acceso completamente.

Consulte la documentación de Teradata para obtener más información.

Configuración del acceso a los orígenes de datos de servicios Web

Para configurar el sistema federado para acceder a los orígenes de datos de servicios Web, debe proporcionar al servidor federado información acerca de los orígenes de datos y los objetos a los que desea acceder como, por ejemplo, un documento de lenguaje de descripción de servicios web (WSDL) válido.

Antes de empezar

- Federation debe estar instalado en el servidor que actuará como servidor federado.
- Debe existir una base de datos en el servidor federado.

Puede configurar el servidor federado para acceder a los orígenes de datos de servicios web desde el Centro de control de DB2 o utilizando la línea de mandatos de DB2. El Centro de control de DB2 incluye un asistente que sirve de guía en los pasos necesarios para configurar el servidor federado.

Procedimiento

Para configurar el acceso a los orígenes de datos de servicios Web:

1. Registre el derivador de servicios web.
2. Registre la definición de servidor para orígenes de datos de servicios web.
3. Registre las correlaciones de usuarios para habilitar la seguridad para la autenticación HTTP (opcional)
4. Registre los apodos de los orígenes de datos de servicios Web:
 - Registre los apodos de los orígenes de datos de servicios web utilizando la línea de mandatos de DB2.
 - Registre los apodos de los orígenes de datos de servicios web utilizando el Centro de control de DB2.
5. Cree vistas federadas para los apodos de servicios web.

Servicios web y derivador de servicios web

Los proveedores de servicios web se describen mediante documentos WSDL (lenguaje de descripción de servicios web). Puede utilizar el derivador de servicios web para acceder a los proveedores de servicios web.

El diagrama Figura 9 en la página 227 muestra la arquitectura de los servicios web.

1. Un proveedor de servicios web implementa un servicio y publica la información de WSDL en un intermediario de servicio, como UDDI.
2. El consumidor del servicio puede utilizar el intermediario del servicio para buscar un proveedor de servicios web.
3. Cuando el consumidor del servicio encuentra un proveedor de servicios web, el consumidor establece un enlace con el proveedor de servicios para poder utilizar el servicio web.

4. El consumidor invoca el servicio intercambiando mensajes SOAP (Protocolo de acceso a objetos simple) entre el solicitante y el proveedor.

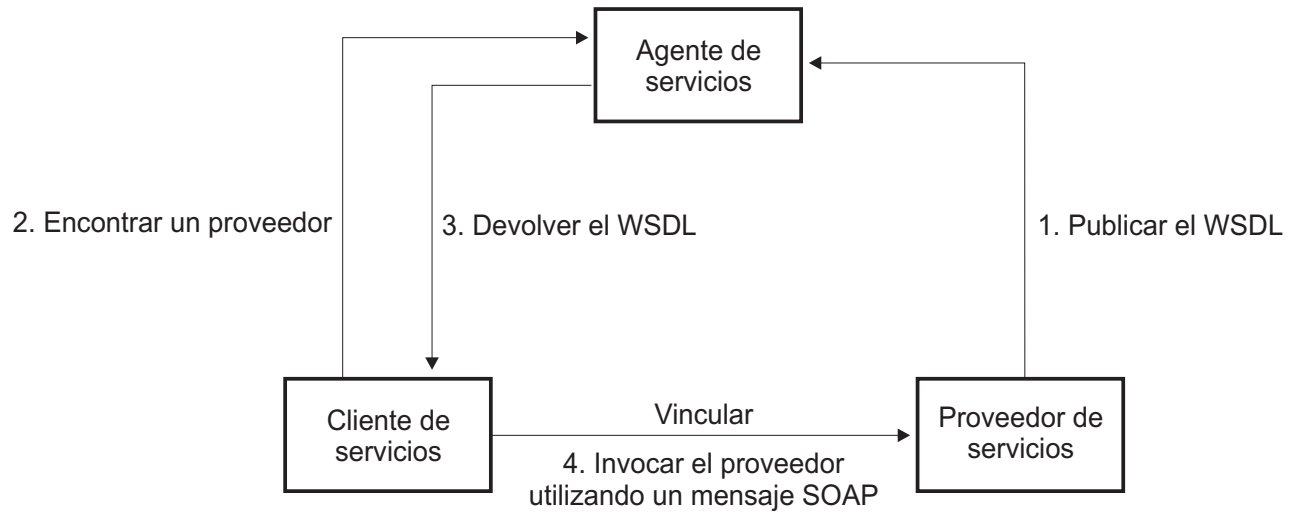


Figura 9. Servicios web: una arquitectura orientada a servicios

La especificación SOAP define el diseño de un mensaje basado en XML. Un mensaje SOAP se encuentra dentro de un sobre SOAP. El sobre consiste en una cabecera SOAP opcional y un cuerpo SOAP obligatorio. La cabecera SOAP puede contener información sobre el mensaje, como información de cifrado o información de autenticación. El cuerpo SOAP contiene el mensaje. La especificación SOAP también define una codificación predeterminada para los enlaces del lenguaje de programación, denominada codificación SOAP.

El documento WSDL y el servicio web

La clave para el servicio web es el documento WSDL. El documento WSDL es un documento XML que describe servicios web en términos de mensajes que envía y recibe. Los mensajes se describen utilizando un sistema de tipos, que suele ser el esquema XML. Una operación de servicio web asocia un patrón de intercambio de mensajes con uno o más mensajes. Un patrón de intercambio de mensajes identifica la secuencia y la cardinalidad de los mensajes que se envían o reciben, así como a quién se envían o quién recibe de forma lógica los mensajes. Una interfaz agrupa las operación sin ningún tipo de compromiso con el formato de transporte o de conexión. Un enlace WSDL especifica los detalles del formato de transporte y de conexión para una o más interfaces. Un punto final asocia una dirección de red con un enlace. Un servicio agrupa puntos finales que implementan una interfaz común. Los mensajes pueden contener información asociada a documentos o información asociada a procesos, lo que también se conoce como llamadas a procedimiento remoto (RPC). Un documento WSDL puede contener uno o más servicios web.

El ejemplo de la Figura 10 en la página 229 muestra la definición WSDL de un servicio simple que proporciona cotizaciones de acciones. El servicio web da soporte a una operación simple denominada `GetLastTradePrice`. Se puede acceder al servicio con el protocolo SOAP 1.1 a través de HTTP. La solicitud lee un símbolo del indicador de cotizaciones, que es un tipo de datos de serie, y devuelve un precio, que es un tipo de datos flotante. Los tipos de datos de serie y de coma flotante son tipos predefinidos en los estándares de esquemas XML. Un servicio

web también puede definir tipos de datos y utilizar dichos tipos de datos definidos por el usuario en los mensajes. Los tipos de datos XML predefinidos y los definidos por el usuario se correlacionan con columnas de apodos. El ejemplo completo y la especificación WSDL se encuentran en el sitio web W3C.

```

<?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>

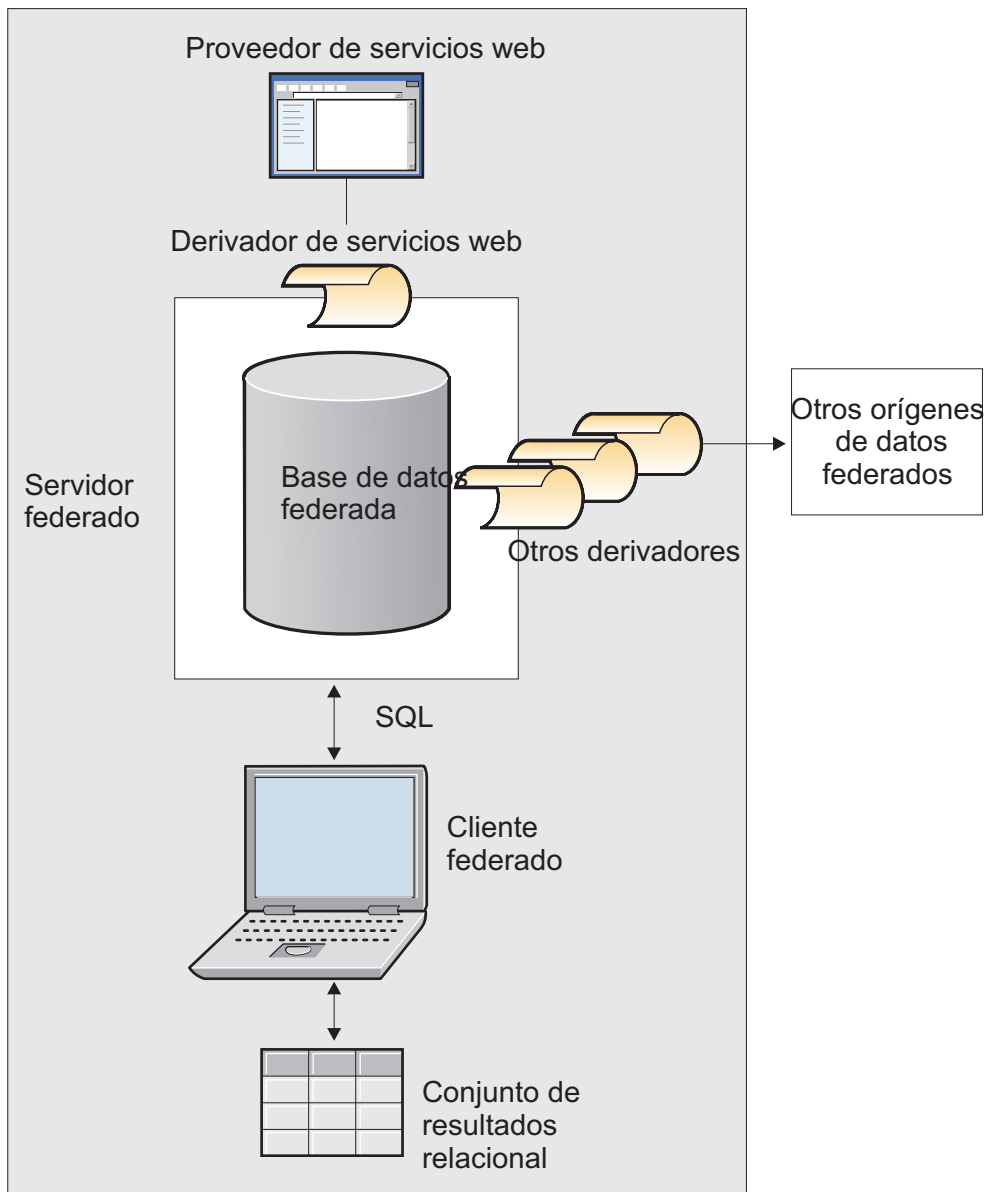
```

Figura 10. Ejemplo de documento WSDL

Documento WSDL, derivador de servicios web y apodos

El derivador de servicios web utiliza las operaciones en un tipo de puerto que tiene un enlace SOAP con un transporte HTTP. Los mensajes de entrada de la operación, y los tipos o elementos asociados, pasan a ser columnas en el apodo. Los mensajes de salida de la operación se extraen en la jerarquía de apodos. Puede crear una jerarquía distinta de apodos para cada operación en el documento WSDL.

Mediante el derivador de servicios web, puede utilizar funciones de los sistemas federados para unir datos de los servicios web con datos de otros orígenes de datos federados.



El ejemplo de la Figura 11 en la página 231 utiliza un documento WSDL que contiene un portType con el nombre de operación GETTEMP. Con este servicio web, debe especificar un código postal como entrada y recibir una temperatura

para dicho código 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>
      Returns current temperature in a given U.S. zipcode
    </documentation>
    <port name="TemperaturePort" binding="tns:TemperatureBinding">
      <soap:address
        location="http://services.xmethods.net:80/soap/servlet/rpcrouter" />
    </port>
  </service>
</definitions>
```

Figura 11. Servicio web GETTEMP

El valor de entrada se describe mediante la columna de código postal (zipcode) del apodo. El valor de salida se describe mediante la columna de retorno del apodo. En el documento WSDL, dichas columnas se identifican en el elemento messages. El elemento messages representa la definición lógica de los datos que se envían entre el proveedor de servicios web y el consumidor de servicios web. Si se necesita más información en el elemento messages, el documento WSDL también puede contener un elemento type. El elemento type puede hacer referencia a tipos predefinidos que se basan en las especificaciones de esquemas XML o tipos definidos por un usuario.

El ejemplo de la Figura 12 en la página 232 muestra el apodo que la herramienta DB2[®] Control Center Discovery produce a partir del documento WSDL del servicio web GETTEMP. La columna zipcode es una columna de entrada necesaria debido a

la sintaxis TEMPLATE del apodo:

```
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/*' ,
  NAMESPACE ' ns1="http://www.xmethods.net/sd/TemperatureService.wsdl",
    ns2="urn:xmethods-Temperature" ,
    soapenv="http://schemas.xmlsoap.org/soap/envelope/"');
```

Figura 12. Apodo GETTEMP

Las opciones de apodo en el derivador de servicios web, URL y SOAPACTION, proporcionan la capacidad de sobrescribir el punto final, o la dirección especificada al crear el apodo. Si utiliza las columnas habilitadas URLCOLUMN o SOAPACTIONCOLUMN en una consulta, puede utilizar direcciones dinámicas con los mismos apodos. Si define las opciones de apodo URL y SOAPACTION al crear un apodo y habilita URLCOLUMN y SOAPACTIONCOLUMN en la opción de columna, utiliza las últimas funciones de enlace de los derivadores de servicios web. El valor para la opción de apodo SOAPACTION pasa a ser un atributo en la cabecera HTTP. El valor para la opción de apodo URL es el URL HTTP al que se envía la solicitud.

Las opciones de apodo URL y SOAPACTION proporcionan asociaciones de apodos dinámicas. Estas direcciones dinámicas son útiles si varias empresas implementan un portType de servicio web. El derivador de servicios web requiere que las únicas diferencias entre los documentos WSDL sean los distintos valores de URL y SOAPACTIONS. Puede utilizar la última función de enlace para crear y utilizar el mismo apodo para distintos puntos finales de servicio que distintas empresas deseen utilizar. Los valores de URL y SOAPACTION provienen del documento WSDL.

En el ejemplo siguiente se muestra cómo utilizar las opciones de columna URLCOLUMN y 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',
    ...
  )

```

Figura 13. Apodo GetPartQuote

El ejemplo siguiente utiliza las columnas URLCOL y SOAPACTIONCOL definidas con las opciones de columna URLCOLUMN y SOAPACTIONCOLUMN habilitadas:

```

SELECT * FROM supplier_endpoints p,
  GetPartQuote q
WHERE partnumber=1234 AND
  p.url=q.urlcol AND
  p.soapaction=q.soapactioncol;

```

La aplicación SQL puede diferir la elección de los puntos finales que deben utilizarse hasta el momento de ejecutar la consulta, en lugar de definir un punto final específico en el momento de crear el apodo.

El derivador de servicios web puede separar grandes volúmenes de datos de documentos WSDL en fragmentos para reducir la memoria total utilizada. Especifique la opción **STREAMING** en la página de valores de la ventana de propiedades del Centro de control de DB2 cuando cree un apodo de servicios web. El derivador de servicios web procesa la secuencia resultante de datos XML y extrae la información solicitada por un fragmento de consulta. El derivador de servicios web analiza un fragmento cada vez. Utilice la opción **STREAMING** para analizar únicamente documentos XML de gran tamaño.

Registro del derivador de servicios web

Debe registrar un derivador para acceder a los orígenes de datos de servicios web. Los servidores federados utilizan derivadores para comunicarse con los orígenes de datos y recuperar datos de éstos. Los derivadores se implementan como un conjunto de archivos de biblioteca.

Antes de empezar

Consulte la lista de archivos de biblioteca de derivador de servicios web para ver el nombre correcto que debe especificar en la sentencia CREATE WRAPPER.

El nombre del archivo de biblioteca de derivador que especifica depende del sistema operativo del servidor federado.

Procedimiento

Para registrar un derivador:

Emita una sentencia CREATE WRAPPER con el nombre del derivador y el nombre del archivo de biblioteca de derivador. Por ejemplo, para registrar un derivador con el nombre websr_wrapper en un servidor federado que utiliza Windows, emita la siguiente sentencia:

```
CREATE WRAPPER websr_wrapper LIBRARY 'db2ws.dll';
```

Archivos de biblioteca de derivador de servicios web

Los archivos de biblioteca de derivador de servicios web se añaden al servidor federado cuando se instala el servidor federado.

Cuando instala IBM InfoSphere Federation Server, se añaden archivos de biblioteca a la vía de acceso de directorios predeterminada. Por ejemplo, si el servidor federado se ejecuta en AIX, los archivos de biblioteca que se añaden a la vía de acceso de directorios son libdb2ws.a, libdb2wsF.a y libdb2wsU.a. El archivo de biblioteca del derivador predeterminado es libdb2ws.a. Los otros archivos de biblioteca de derivador se utilizan con opciones de derivador específicas.

Debe incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER y especificar el nombre de archivo de biblioteca del derivador predeterminado.

Las vías de acceso de directorios predeterminadas y los nombres de archivo de biblioteca del derivador predeterminados se especifican en la tabla siguiente.

Tabla 40. Ubicaciones y nombres de archivo de bibliotecas para CREATE WRAPPER

Sistema operativo	Vía de acceso de directorios	Archivo de biblioteca de derivador
AIX	/usr/opt/<vía_acceso_instalación>/lib32/ /usr/opt/<vía_acceso_instalación>/lib64/	libdb2ws.a
Linux	/opt/IBM/db2/<vía_acceso_instalación>/lib32/ /opt/IBM/db2/<vía_acceso_instalación>/lib64	libdb2ws.so
Solaris	/opt/IBM/db2/<vía_acceso_instalación>/lib32/ /opt/IBM/db2/<vía_acceso_instalación>/lib64	libdb2ws.so
Windows	%DB2PATH%\bin	db2ws.dll

- <vía_acceso_instalación> es la vía de acceso de directorios donde se instala el servidor federado en Linux o UNIX.

%DB2PATH% es la variable de entorno que se utiliza para especificar la vía de acceso de directorios donde se instala el servidor federado en Windows. La vía de acceso de directorios predeterminada de Windows es C:\Archivos de programa\IBM\SQLLIB.

Registro de la definición de servidor para orígenes de datos de servicios web

Puede registrar una definición de servidor desde el Centro de control de DB2 o utilizando la línea de mandatos de DB2. Esta tarea describe cómo registrar una definición de servidor de servicios web desde la línea de mandatos de DB2.

Debe registrarse una definición de servidor para cada servicio web al que desee acceder.

Procedimiento

Para registrar una definición de servidor en el sistema federado para el derivador de servicios web, emita la sentencia CREATE SERVER.

Por ejemplo, para registrar una definición de servidor de servicios web denominada `ws_server` en Windows, emita la siguiente sentencia:

```
CREATE SERVER ws_server WRAPPER
derivador_serv_web;
```

Puede establecer parámetros de servidor de proxy y tiempo de espera opcionales para la sentencia `CREATE SERVER`.

Sentencia `CREATE SERVER` - Ejemplos para el derivador de servicios web

Utilice la sentencia `CREATE SERVER` para registrar definiciones de servidor para el derivador de servicios web con los valores de tiempo de espera y de servidor proxy.

Aun cuando no utilice un servidor proxy para acceder a documentos de servicios web, debe registrar una definición de servidor. La jerarquía de objetos federados requiere que los archivos de servicios web estén asociados con un objeto de definición de servidor específico. La sentencia que se emite para registrar una definición de servidor es la siguiente:

```
CREATE SERVER mi_servidor WRAPPER mi_derivador
OPTIONS (TIMEOUT '60');
```

`mi_servidor`

Nombre que se asigna a la definición del servidor de servicios web. No se permiten los nombres de definición de servidor duplicados.

`WRAPPER mi_derivador`

Nombre de derivador que ha especificado en la sentencia `CREATE WRAPPER`.

`TIMEOUT '60'`

Especifica el tiempo, en minutos, durante el que debe esperar el servidor federado para una transferencia de red y para el cálculo de un resultado.

Definiciones de servidor cuando se utiliza un servidor proxy

Debe utilizar las opciones de servidor proxy de la sentencia `CREATE SERVER` si se cumplen todas estas condiciones:

- Desea recuperar datos mediante un URI.
- El URI recupera datos de detrás de un cortafuegos, a través de un proxy.
- El cortafuegos o el proxy es HTTP o SOCKS.

Las opciones que especifique dependen del tipo de servidor proxy al que desee acceder.

Póngase en contacto con el administrador de la red para obtener información sobre el tipo de proxy que se utiliza y los valores que debe especificar para las opciones del proxy.

Registro de una definición de servidor para un servidor proxy HTTP

Para registrar una definición de servidor y especificar un servidor proxy HTTP, utilice esta sentencia:

```
CREATE SERVER servidor_ws_http
WRAPPER derivador_ws
OPTIONS (PROXY_TYPE 'HTTP', PROXY_SERVER_NAME 'proxy_http',
PROXY_SERVER_PORT '8080');
```

servidor_ws_http

Nombre que se asigna a la definición del servidor de servicios web. No se permiten los nombres de definición de servidor duplicados.

WRAPPER *derivador_ws*

Nombre de derivador que ha especificado en la sentencia CREATE WRAPPER.

PROXY_TYPE 'HTTP'

Especifica el tipo de proxy que se utiliza para acceder a Internet cuando el servidor se encuentra detrás de un cortafuegos.

PROXY_SERVER_NAME 'proxy_http'

Especifica el nombre o la dirección IP del servidor proxy.

PROXY_SERVER_PORT '8080'

Especifica el número de puerto del servidor proxy.

Registro de una definición de servidor para un servidor SOCKS con información sobre autenticación

Para registrar una definición de servidor y especificar un servidor proxy SOCKS con información de autenticación, emita esta sentencia:

```
CREATE SERVER servidor_ws_socks
  WRAPPER derivador_ws
  OPTIONS (PROXY_TYPE 'SOCKS', PROXY_SERVER_NAME 'proxy_socks',
    PROXY_SERVER_PORT '1081', PROXY_AUTHID 'Sampson',
    PROXY_PASSWORD 'sailing4me');
```

PROXY_AUTHID 'Sampson'

Especifica el nombre de usuario en el servidor proxy.

PROXY_PASSWORD 'sailing4me'

Especifica la contraseña del servidor proxy asociada con el nombre de usuario 'Sampson'.

Habilitación de la seguridad mediante el derivador de servicios web

El derivador de servicios web da soporte a la autenticación HTTP utilizando la sentencia CREATE USER MAPPING.

El derivador de servicios web da soporte a HTTPS como protocolo de transporte para los mensajes SOAP. El documento WSDL que genera el proveedor de servicios web contiene https:// en el URL. Por lo tanto, los mensajes SOAP en la solicitud HTTP o en la respuesta HTTP están cifrados.

Si el servicio web utiliza HTTPS como protocolo de transporte, puede configurar el derivador para validar los certificados SSL que el servidor envía para su identificación utilizando la opción de servidor o derivador SSL_VERIFY_SERVER_CERTIFICATE. El derivador de servicios web puede invocar servicios web con certificados autofirmados.

La correlación de usuarios es opcional. Si no especifica una correlación de usuarios, puede aparecer un error si el proveedor de servicios web espera información de autenticación. Algunos servidores pueden utilizar la autenticación para restringir el acceso a un servicio. La necesidad de autenticación no es evidente en la información del documento WSDL.

Procedimiento

Para correlacionar un ID de usuario de servidor federado con un ID de usuario y una contraseña de servicios web:

Emita la sentencia CREATE USER MAPPING.

Por ejemplo, con la siguiente sentencia CREATE USER MAPPING, cuando se accede al apodo de servicios web en el servidor S1, se envía la solicitud HTTP con SYSTEM como ID de usuario y MANAGER como contraseña.

```
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');
```

Registro de apodos para los orígenes de datos de servicios web

Para cada definición de servidor de servicios web, debe registrar un apodo para cada origen de datos al que desee acceder. Puede registrar un apodo utilizando la línea de mandatos o el Centro de control de DB2.

Para registrar los apodos de los orígenes de datos de servicios web:

Seleccione uno de los métodos siguientes:

- “Registro de apodos para los orígenes de datos de servicios web (línea de mandatos de DB2)”
- “Registro de apodos para orígenes de datos de servicios web (Centro de control de DB2)” en la página 238

Registro de apodos para los orígenes de datos de servicios web (línea de mandatos de DB2)

Para cada definición de servidor de servicios web, debe registrar un apodo para cada origen de datos al que desee acceder. Utilice estos apodos en lugar de los nombres de los orígenes de datos cuando consulte los orígenes de datos de servicios web.

Antes de empezar

Debe tener acceso a un documento WSDL válido que describe el servicio web con el que desea comunicarse.

Restricciones

- Sólo se da soporte a las operaciones de solicitud-respuesta.
- El único enlace que está soportado es un enlace SOAP con un transporte HTTP.
- Debe utilizar la opción TEMPLATE o la opción XPATH en cada columna, excepto las columnas especiales con las opciones SOAPACTIONCOLUMN, URLCOLUMN, PRIMARY_KEY o FOREIGN_KEY.

Debe crear una jerarquía de apodos para cada operación de servicio web definida en el documento de lenguaje de descripción de servicios web (WSDL). El apodo padre identifica el sobre SOAP; los apodos hijo identifican los elementos del cuerpo SOAP y la cabecera SOAP opcional.

Procedimiento

Para registrar apodos para los orígenes de datos de servicios web desde la línea de mandatos de DB2:

Emita una sentencia CREATE NICKNAME. Por ejemplo, para registrar los apodos en Windows para un servicio web denominado GETTEMP, emita la siguiente sentencia:

```
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/*' ,
  NAMESPACE ' ns1="http://www.xmethods.net/sd/TemperatureService.wsdl",
  ns2="urn:xmethods-Temperature" ,
  soapenv="http://schemas.xmlsoap.org/soap/envelope/" );
```

Registro de apodos para orígenes de datos de servicios web (Centro de control de DB2)

Para cada definición de servidor de servicios web que registre, debe registrar un apodo para cada origen de datos al que desee acceder. Utilice estos apodos en lugar de los nombres de los orígenes de datos cuando consulte los orígenes de datos de servicios web.

Antes de empezar

Debe tener acceso a un documento WSDL válido que describa el servicio web con el que desea comunicarse.

Restricciones

- Sólo se da soporte a operaciones de solicitud-respuesta.
- El enlace SOAP con transporte HTTP es el único enlace soportado.
- Debe utilizar la opción TEMPLATE o la opción XPATH en cada columna, excepto en las columnas especiales con las opciones SOAPACTIONCOLUMN, URLCOLUMN, PRIMARY_KEY o FOREIGN_KEY.

Debe crear una jerarquía de apodos para cada operación de servicio web que se defina en el documento de lenguaje de descripción de servicios web (WSDL). El apodo padre identifica el sobre SOAP; los apodos hijo identifican los elementos de cuerpo SOAP y la cabecera SOAP opcional.

En el Centro de control de DB2, puede utilizar la herramienta de descubrimiento para crear apodos. La entrada para la herramienta de descubrimiento es un URL de la ubicación de un documento WSDL. La herramienta de descubrimiento crea apodos como resultado procesar el documento WSDL. El Centro de control de DB2 genera apodos exclusivos agrupando nombres de partes o nombres de elementos con el nombre de columna del documento WSDL. El documento WSDL puede contener definiciones de esquema incrustadas en el archivo WSDL o en un archivo de esquema XML externo que se importa en el archivo WSDL. Estas definiciones de esquema se importan mediante una dirección URL.

Cuando el Centro de control de DB2 genera un apodo hijo que sólo se utiliza para entrada, la sentencia XPATH contiene un punto, como en el siguiente ejemplo:
XPATH '.'

Procedimiento

Para registrar apodos para orígenes de datos de servicios web desde el Centro de control de DB2:

1. Expanda la carpeta **Objetos de base de datos federada**.
2. Expanda la carpeta del derivador para el que desea registrar apodos.
3. Expanda la carpeta **Definiciones de servidor**.
4. Expanda la carpeta del servidor para el que desea registrar apodos.
5. Pulse con el botón derecho en la carpeta **Apodos** y seleccione **Crear**.
6. En la ventana Apodos, pulse **Descubrir** para definir los criterios de búsqueda que le ayudarán a seleccionar objetos en el origen de datos.
7. Especifique el documento WSDL que contiene la definición del servicio web al que desea que accedan los usuarios del sistema federado. El documento WSDL puede ser un documento local o puede especificar su ubicación mediante un URL.
8. Pulse **Aceptar** para crear el apodo de acuerdo con el documento WSDL seleccionado.

El Centro de control de DB2 extrae el documento WSDL en varias sentencias DLL de creación de apodos, con las definiciones de relaciones padre-hijo adecuadas. La sentencia del lenguaje de definición de datos (DDL) que genera el Centro de control de DB2 correlaciona todos los elementos de entrada con columnas del apodo raíz de la jerarquía de apodos.

Sentencia CREATE NICKNAME - Ejemplos para el derivador de servicios web

Cuando crea un apodo para acceder a un servicio web, crea una columna de entrada para cada valor en el mensaje de entrada de una operación de servicio web y una columna de salida para cada valor en el mensaje de salida de una operación de servicio web. Las definiciones de columna de entrada y salida se controlan con las definiciones de opción de columna de apodo.

La opción de columna TEMPLATE especifica que una columna es una columna de entrada. La opción de columna XPATH especifica que una columna es una columna de salida. Cuando la opción de apodo TEMPLATE contiene una notación entre delimitadores ([1,1]), la columna es una columna de entrada necesaria.

La opción de apodo NAMESPACES es una lista separada por comas de pares nombre-valor que un sistema federado utiliza para resolver los espacios de nombres de los elementos en los documentos XML de entrada y salida. Los espacios de nombres se utilizan en la solicitud de mensaje para definir los prefijos en la opción de apodo TEMPLATE. La opción de apodo NAMESPACES resuelve los prefijos en las expresiones XPath con los URI de espacio de nombre definidos en los esquemas WSDL o XML. Las expresiones XPath se aplican en el documento XML que se devuelve del servicio web.

Ejemplo 1: columnas de entrada necesarias

El siguiente ejemplo muestra un apodo para un servicio web denominado getQuote. El servicio web lee un símbolo de acciones y devuelve el precio de las acciones. La herramienta de descubrimiento en el Centro de control de DB2 crea la siguiente sentencia DDL:

```
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/*' ,  
  NAMESPACE 'ns2="urn:xmethods-delayed-quotes" ,  
    ns1="http://www.example.com/wsdl/  
net.xmethods.services.stockquote.StockQuote/" ,  
    soapenv="http://schemas.xmlsoap.org/soap/envelope/" ');
```

La opción de apodo TEMPLATE especifica la columna SYMBOL como una columna de entrada necesaria, ya que la columna contiene la designación [1,1]. En la opción de apodo TEMPLATE, se especifica el sobre SOAP completo para el servicio web. El valor de entrada getQuote está contenido en el sobre SOAP y los elementos del cuerpo. La opción de apodo XPATH contiene la información necesaria para encontrar el valor del precio de las acciones mediante el sobre SOAP y los códigos del cuerpo.

Utilice el apodo stockquote.stockquoteport_getquote_nn para acceder al servicio web, como en la siguiente consulta:

```
SELECT * FROM "stockquote.stockquoteport_getquote_nn"  
  WHERE symbol='IBM';
```

Debe utilizar el predicado symbol='IBM' en esta sentencia porque symbol es una columna de entrada necesaria. El predicado de igualdad es el único predicado válido en las columnas de entrada. Cada uno de los predicados de igualdad establece un valor en el mensaje de entrada. Si la columna de entrada es opcional, no es necesario un predicado de igualdad en la columna. Si la columna de entrada es necesaria, debe emitir la consulta con un predicado de igualdad. Puede utilizar un valor literal como IBM en una expresión de igualdad o un valor de un apodo o una tabla unida.

Ejemplo 2: elementos repetidos y apodos hijo

El siguiente ejemplo utiliza un servicio web denominado getZooReport que produce un informe de los zoológicos. El valor de entrada es un identificador de zoológico. El valor de salida es un informe que se describe mediante el siguiente esquema:

```

<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>
  ...

```

Figura 14. Servicio web *getZooReport*

La herramienta de descubrimiento en el Centro de control de DB2 genera la siguiente sentencia DDL basándose en el WSDL que contiene el esquema:

```

CREATE NICKNAME zooport_getzooreport_nn (
  zoid VARCHAR (48) OPTIONS(TEMPLATE '&column'),
  zoo_id VARCHAR (48) OPTIONS(XPATH './ns1:Zoo/@ns1:id'),
  report_zooname VARCHAR (48) OPTIONS(XPATH './ns1:Zoo/ns1:ZooName/text()'),
  report_count VARCHAR (48) OPTIONS(XPATH './ns1:Zoo/ns1:Count/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 " ');

```

Figura 15. Apodo padre $\hat{\Delta}$ 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" ');

```

Figura 16. Hijo de apodo 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" ');

```

Figura 17. Hijo de zooport_getzooreport_report_nn

El esquema incluye algunos elementos que se repiten, conocidos como elementos de secuencia. Estos elementos repetidos se convierten en apodos hijo del apodo padre, tal como se muestra en Figura 15, Figura 16 y Figura 17. Por ejemplo,

zooname, count, lastmodified y zookeeper son todos elementos de zoo. El elemento zoo contiene 0 o más elementos zookeeper. El apodo raíz, zoo, contiene las columnas zooname, count y lastmodified. La herramienta de descubrimiento del Centro de control de DB2 crea un apodo hijo, zookeeper, para describir los elementos repetidos de zookeeper. El tercer elemento en la columna zookeeper, animalcarelist, también contiene 0 o más elementos y, por lo tanto, se convierte en un apodo hijo, zooport_getzooreport_report_report_nn. La siguiente figura muestra la jerarquía de apodos:

Apodo raíz: zooport_getzooreport_nn

Zoo (padre):

- ZooName
- Count
- LastModified
- ZooKeeper (hay 0 o más elementos ZooKeeper)

Apodo hijo: zooport_getzooreport_report_nn

Elementos ZooKeeper

- Name
- NumberOfCages
- AnimalCareList (hay 0 o más elementos Animal)

Apodo hijo: zooport_getzooreport_report_report_nn

Animal

- Name
- Species
- Lot

Figura 18. Jerarquías de apodos padre y hijo

La siguiente sentencia es una consulta típica que puede emitir en los apodos para acceder al servicio web de informe de zoo. Cuando emite esta sentencia, recupera la información del informe de zoo basándose en un identificador específico y en donde coinciden las claves primaria y foránea de los informes de zoo de apodo hijo.

```
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;
```

Ejemplo 3: enlace posterior

El siguiente ejemplo muestra cómo puede utilizar la opción de enlace posterior. Puede utilizar esta opción desde el Centro de control de DB2 o desde una línea de mandatos de DB2. Si define las opciones de apodo URL y SOAPACTION, y habilita las opciones de columna URLCOLUMN y SOAPACTIONCOLUMN cuando crea un apodo, está utilizando las funciones de enlace posterior. El Centro de control de DB2 crea las opciones de columna URLCOLUMN y SOAPACTIONCOLUMN, y establece los valores de las columnas en yes.

El siguiente ejemplo es un servicio web que proporciona precios estimados para las piezas y está implementado por todos los proveedores de una empresa. Esta es la sentencia CREATE NICKNAME que incluye las definiciones de URLCOLUMN y 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',
    ...
  )

```

Para obtener los precios estimados de todos los proveedores con una sola consulta, son necesarios los valores que utiliza cada proveedor para las opciones de columna SOAPACTION y URL. La consulta será como la siguiente:

```

SELECT * FROM supplier_endpoints p,
       GetPartQuote q
WHERE partnumber=1234 AND
      p.url=q.urlcol AND
      p.soapaction=q.soapactioncol;

```

La tabla local `supplier_endpoints` contiene todos los URL y las direcciones SOAP con las que puede invocar el servicio web. Puede incluir una cláusula de precio `ORDER BY` para determinar el proveedor menos cargo para esta pieza.

Ejemplo 4: opción de columna `ESCAPE_INPUT`

Puede incluir fragmentos XML como valores de entrada en la consulta. Cuando registre un apodo, incluya la opción de columna `ESCAPE_INPUT=N`. Esta opción mantiene los caracteres especiales como, por ejemplo, los corchetes angulares (`<` y `>`) en fragmentos XML en los valores de entrada.

Cuando un esquema contiene valores de entrada repetidos que requieren que envíe XML como parte del mensaje SOAP, puede utilizar la opción de columna `ESCAPE_INPUT` para crear el mensaje de salida con el XML correcto.

Por ejemplo, el servicio web `zoo` incluye una operación para añadir un nuevo cuidador del zoo y los animales asociados con dicho cuidador. En el esquema de este ejemplo, `AnimalCareList` puede tener varios animales.

```

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')
  ...

```

Para añadir un nuevo cuidador del zoo con dos animales, emita una consulta como en el siguiente ejemplo:

```

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>;

```

El apodo `add_zookeeper` es una operación de servicio web que puede cambiar el estado del servicio web o actualizar información. Aunque los derivadores no relacionales no pueden actualizarse, la sentencia `SELECT` en este ejemplo actualiza la información del zoo para añadir un nuevo cuidador del zoo.

También puede utilizar la opción de columna `ESCAPE_INPUT` para un esquema con un elemento como `xsd:anyType`. En este caso, el tipo de elemento es desconocido. Puede utilizar la opción de columna `ESCAPE_INPUT` en la columna de entrada del elemento para especificar fragmentos XML arbitrarios para la entrada.

Acceso a cabeceras SOAP

Utilice el derivador de servicio web para acceder a valores de cabecera de Simple Object Access Protocol (SOAP) que están incluidos en un mensaje SOAP.

Un mensaje SOAP, ya sea una solicitud o una respuesta, está formado por un sobre que contiene los espacios de nombre que utiliza el resto del mensaje SOAP, una cabecera opcional y un cuerpo. Una cabecera SOAP normalmente contiene metadatos sobre el mensaje SOAP y otra información opcional. El mensaje SOAP viaja de un origen a un destino pasando por una serie de servicios intermedios que procesan el mensaje o sus efectos colaterales. Por ejemplo, un mensaje puede pasar por un servicio de transacciones que proporciona al cliente una invocación garantizada si falla la red, o un servicio de seguridad que se encuentra en un portal empresarial puede proporcionar información de autenticación. Estos intermediarios a menudo trabajan en los metadatos contenidos en la cabecera SOAP.

Las cabeceras SOAP a menudo contienen información opcional o soporte que produce interfaces de usuario. Por ejemplo, un banco permite a los clientes utilizar tarjetas para retirar dinero de sus cuentas corrientes en un cajero automático. No obstante, algunos clientes tienen más de una cuenta. Si un cliente tiene varias cuentas, el banco debe poder identificar la cuenta de la que se realiza la extracción.

En este ejemplo, el siguiente es el elemento de cabecera que especifica una cuenta bancaria:

```
<xsd:element name="AccountID" type="xsd:int"/>
```

El siguiente es un mensaje SOAP que utiliza este elemento de cabecera. En este caso, el cliente ha seleccionado retirar 1.000 dólares de la cuenta 1 y la operación tiene una comisión de 50 céntimos.

```

<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>

```

Las cabeceras SOAP tienen estos requisitos:

- `soapenv:Header` es un elemento opcional en `soapenv:Envelope`.

- Si se utiliza, la cabecera SOAP debe ser el primer elemento.
- Todos los hijos de soapenv:Header deben estar calificados para los espacios de nombres.
- soapenv:Header puede tener estos atributos: actor, mustUnderstand y estilo de codificación.

La jerarquía de apodos para acceder a las cabeceras SOAP requiere que el apodo raíz apunte al sobre SOAP. Los elementos de cabecera y de cuerpo son apodos hijo del apodo raíz. Puede crear apodos manualmente o utilizar el Centro de control para generarlos automáticamente. A continuación, puede crear consultas que accedan a la información almacenada en la cabecera SOAP.

Ejemplo

Estas sentencias CREATE NICKNAME muestran cómo crear apodos raíz, de cuerpo y de cabecera.

```
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 siguiente consulta devuelve el precio de las acciones de IBM para una fecha y hora concreta:

```

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';

```

Apodos y expresiones XPath

El derivador de servicios web utiliza apodos que corresponden a la estructura de árbol de datos XML. Cada apodo se define mediante expresiones XPath que representan valores de salida.

Los apodos corresponden a la estructura de árbol de los datos de documentos XML. Los apodos padre y los apodos hijo corresponden a la estructura raíz y a los elementos anidados de la estructura de árbol de datos. Estos apodos padre e hijo están conectados mediante claves primarias y foráneas que se especifican mediante la sentencia CREATE NICKNAME.

Cada apodo se define mediante expresiones XPath que representan valores de salida. El derivador de servicios web utiliza expresiones XPath para establecer una correspondencia entre los datos de un documento XML y las filas de la tabla de datos relacional. Estas expresiones XPath identifican los valores del documento XML y determinan la correspondencia entre estos valores y las columnas de cada filfa. El derivador sólo lee los datos XML y no actualiza el documento. La opción XPath contiene la información para buscar mensajes SOAP a través del sobre SOAP y las etiquetas de cuerpo SOAP. El mensaje getQuote está contenido en los elementos de sobre y cuerpo SOAP.

La expresión XPath para la opción NICKNAME apunta a etiquetas repetitivas que se encuentran en el elemento de salida. La expresión XPath determina cuántas filas habrá en el apodo. La expresión XPath de opción de columna es relativa a la opción XPath NICKNAME. La expresión XPath de opción de columna identifica los valores de una fila. La expresión XPath de opción NICKNAME en un apodo hijo es relativa a la expresión XPath de opción NICKNAME de un apodo padre.

Al crear un apodo, debe seleccionar opciones de un documento de lenguaje de descripción de servicios web (WSDL) de servicios web que especifican la asociación entre el apodo y el documento XML.

Opción TEMPLATE en los niveles de apodo y de columna

Al crear un apodo, el derivador de servicios web requiere los fragmentos de plantilla de nivel de apodo y nivel de columna, que es la opción TEMPLATE de la sentencia CREATE NICKNAME.

El derivador de servicios web crea documentos XML que son necesarios en el entorno de servicios web. El derivador utiliza esta información durante la planificación de las consultas y durante las fases de ejecución de las consultas.

La opción TEMPLATE en el derivador de servicios web:

El derivador de servicios web necesita la opción TEMPLATE en la sentencia CREATE NICKNAME cuando se crea el apodo.

Derivador de servicios web

Para el derivador de servicios web, los atributos necesarios y opcionales varían dependiendo de las definiciones en el documento WSDL y de cómo se derive una columna. Una columna puede derivarse de un elemento o un atributo de un elemento.

- Si la columna se deriva de un elemento, el valor minOccurs determina si una columna es opcional.
- Si el valor de minOccurs es igual a 0, la columna es opcional.
- Si el valor de minOccurs es igual a 1, la columna es necesaria.
- Si la columna se deriva de un atributo de un elemento, el valor de uso en el atributo determina si una columna es opcional.
- Si un atributo contiene el valor use=optional, la columna es opcional.
- Si un atributo contiene el valor use=required, la columna es necesaria.

El siguiente ejemplo es un atributo en una definición de esquema que está asociado con una columna.

```
<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>
```

Creación de vistas federadas para los apodos de servicios web

Cree vistas federadas para los apodos de servicios web para asegurarse de que las consultas que unen las partes de una jerarquía de apodos de servicios web se ejecutan correctamente.

Puede definir vistas federadas para la jerarquía de apodos que describen un documento de servicios web. Una vista federada garantiza que las consultas que unen las partes de una jerarquía de apodos de servicios web se ejecutan correctamente.

El derivador de servicios web requiere uniones de todos los apodos hijo para incluir todos los apodos en el apodo padre. Una sentencia SELECT sólo falla en un apodo hijo si el apodo padre no se incluye en la sentencia. El método preferido para incluir todos los apodos necesarios en una consulta es definir una vista federada.

Procedimiento

Para crear vistas federadas para los apodos de servicios web:

1. Defina una vista que incluya todos los apodos relacionados con una operación en el servicio web si desea unir los apodos.
2. En la cláusula WHERE de la vista, utilice predicados de unión para todas las columnas relacionadas mediante las opciones de columna PRIMARY_KEY y FOREIGN_KEY.

En el siguiente ejemplo, la clave primaria está en la columna ooport_getzooreport_pk en el apodo zooport_getzooreport_report_nn. La clave foránea está en la columna ooport_getzooreport_fkey en el apodo zooport_getzooreport_report_report_nn.

```
CREATE VIEW zooreport
(zoooid, zooname, number_of_zookeeper,
lastmodified, zookeeper_id, zookeeper_name,
cages, animal_name, animal_species, animal_lot)
AS ( SELECT zoooid, 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);
```

Puede obtener información de todos los apodos con la siguiente sentencia SELECT:
SELECT * FROM zooreport WHERE zoooid='1';

Restricciones de consultas de los derivadores de servicios web

El uso de predicados de consulta y uniones está restringido para los servicios web, tal como se muestra en los ejemplos proporcionados.

Predicados iguales

Los únicos predicados válidos en las columnas de entradas son los predicados iguales. Para las columnas de salida, son válidos todos los predicados.

El siguiente ejemplo devuelve un error con un mensaje que indica que el predicado no está soportado en la columna. En este ejemplo, la columna zipcode es una columna de entrada:

```
SELECT return FROM gettemp WHERE zipcode<'95141'
```

El siguiente ejemplo muestra una consulta válida que utiliza un predicado igual en las columnas de entrada. El apodo de los clientes se une con una tabla local que contiene ID de cliente. La consulta contiene un predicado adicional en la columna Sales, que es una columna sólo de salida.

```
SELECT a.name, a.address
FROM customers a, local_table b
WHERE
a.customer_id=b.custid AND
a.Sales > 300000;
```

Predicados para las columnas de entrada necesarias

Debe proporcionar valores de predicado de igualdad para todas las columnas de entrada necesarias en las consultas SQL para la jerarquía de apodos a la que hace referencia. El derivador devuelve un SQLCODE -901 para todas las consultas que incumplan esta restricción.

Predicados IN u OR

Para los derivadores de servicios web, no se permiten predicados IN u OR para las columnas de entrada.

Los siguientes ejemplos muestran consultas no válidas. El apodo de los clientes tiene una columna de entrada necesaria, customer_id:

```
SELECT * FROM customers
  WHERE customer_id IN (12345, 67890, 11223);
SELECT * FROM customers
  WHERE customer_id IN (SELECT custid FROM local_table); )
```

Uniones en las columnas de entrada opcionales

Los ejemplos siguientes muestran una restricción en la unión de columnas de entrada opcionales y en el uso de variables de host. No puede unir columnas de entrada opcionales de un apodo o una tabla local. Si WSDL genera una columna de apodo de entrada como opcional, deberá utilizarla en una unión y, a continuación, editar la sentencia DDL para cambiar la columna por una columna de entrada necesaria.

En este ejemplo, se crea un apodo de derivador de servicios web denominado order con shipping_method como columna de entrada opcional. La siguiente sentencia es una consulta válida porque utiliza un literal en el predicado:

```
SELECT * FROM order
  WHERE part="hammer" AND shipping_method="FEDEX";
```

No obstante, si incluye una tabla local denominada orderparts que define los componentes y los métodos de envío en la consulta, y la tabla contiene una columna denominada shipping_method que es opcional, la sentencia no será válida:

```
SELECT * FROM
  order o, orderparts op
  WHERE
    o.part="hammer" AND
    o.shipping_method=op.shipping_method
```

Si utiliza variables de host en las consultas con predicados en las columnas de entrada opcionales, la sentencia tampoco será válida:

```
SELECT * FROM
  order o
  WHERE
    o.part="hammer" AND
    o.shipping_method=:ashipping_method
```

Para garantizar unos resultados válidos, las columnas de entrada unidas deben ser columnas necesarias para los derivadores de servicios web.

Uniones externas

Las uniones externas entre los apodos que utilizan la clave primaria de un apodo padre y la clave foránea de las columnas de apodo hijo no están soportadas.

Si un elemento padre en un documento XML no contiene elementos hijo y utiliza una unión interna entre el apodo padre y el apodo hijo, no se devuelven filas para el elemento. Por ejemplo, para un determinado cliente, si no hay información

detallada de detalle bancario en el sistema SAP, no se devuelven filas para el apodo sap_bapi_customer_getdetail2_sap_customerbankdetail_NN para el cliente.

Las siguientes sentencias CREATE NICKNAME definen las columnas que se utilizan en la consulta de ejemplo:

```
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>',
...

```

Figura 19. Extracto del apodo 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>',
...

```

Figura 20. Extracto del apodo de detalle bancario del cliente

En el siguiente ejemplo, la consulta no devuelve filas porque hay una condición de unión interna entre los dos apodos:

```

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 una definición de apodo de derivador de servicios web contiene columnas de entrada necesarias, no está soportada una unión externa izquierda entre este apodo y otros apodos u otra tabla local.

Orígenes de datos de servicios web - consultas de ejemplos

Se muestran ejemplos de consultas SQL para el derivador de servicios web.

Ejemplo 1: utilización de tablas de consultas materializadas

Puede utilizar tablas de consultas materializadas para guardar localmente en la memoria caché los resultados de una consulta y mejorar el rendimiento de las consultas. Puede utilizar apodos de orígenes de datos de servicios web para crear tablas de consultas materializadas. Para algunas consultas, la base de datos puede determinar automáticamente si la tabla de consulta materializada puede responder a una consulta sin acceder a las tablas base. El siguiente procedimiento muestra cómo crear y rellenar una tabla de consulta materializada:

1. Cree una tabla local o base:

```
CREATE TABLE mystocks(ticker VARCHAR(10));
```

Puede utilizar la tabla local para mantener todos los valores que desea guardar en la memoria caché.

2. Inserte en la tabla todos los valores que desea guardar en la memoria caché:

```

INSERT INTO mystocks VALUES('IBM');
INSERT INTO mystocks VALUES('MSFT');
...

```

3. Cree un apodo de servicios 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. Cree una vista formada por el apodo y la tabla local:

```

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. Cree una tabla de consulta materializada:

```

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;

```

Incluya todas las columnas VARCHAR en el predicado de unión (nn.ticker y s.ticker) en la lista de salida de la tabla de consulta materializada para maximizar las oportunidades de que la tabla de consulta materializada se utilice en la base de datos federada.

Para diferir la renovación de la tabla de consulta materializada, especifique la palabra clave REFRESH DEFERRED. Las tablas de consultas materializadas que se especifican con la palabra clave REFRESH DEFERRED no reflejan los cambios en la tabla base subyacente. Utilice la cláusula DATA INITIALLY DEFERRED para que los datos no se inserten en la tabla como parte de la sentencia CREATE TABLE.

6. Emita una sentencia REFRESH TABLE para rellenar la tabla. Los datos de la tabla reflejan el resultado de la consulta como una instantánea en el momento de emitir la sentencia REFRESH TABLE. El siguiente ejemplo rellena la tabla stockquote_MQT y establece un valor para el registro especial con la antigüedad de renovación actual.

```
REFRESH TABLE stockquote_MQT;

SET CURRENT REFRESH AGE any;
```

Las consultas que se ejecutan en los datos en la tabla de consulta materializada son más rápidas que las consultas que se ejecutan en los datos en una tabla base. Si desea utilizar la tabla de consulta materializada, hace referencia a la vista, no al apodo:

```
SELECT * FROM stock_quote_view
WHERE ticker='IBM';
```

Si emite una consulta para seleccionar un valor que no se ha guardado en la memoria caché, se devuelven 0 filas.

Ejemplo 2: emisión de trabajos utilizando las claves primaria y foránea

Las columnas PRIMARY_KEY y FOREIGN_KEY definen relaciones entre los apodos padre e hijo. Cada apodo padre debe tener una opción de columna de clave primaria. El hijo de un apodo padre se define con la opción de columna de clave foránea que hace referencia a la columna de clave primaria de un apodo padre. Un apodo puede tener varios hijos, pero sólo un padre.

Como estas columnas sólo contienen datos binarios, las columnas se definen con las palabras clave FOR BIT DATA NOT NULL. El Centro de control de DB2 genera esta definición cuando crea el apodo. Puede definir explícitamente las columnas PRIMARY_KEY y FOREIGN_KEY como FOR BIT DATA NOT NULL cuando crea el apodo.

El siguiente ejemplo muestra cómo utiliza el derivador de servicios web las columnas PRIMARY_KEY y FOREIGN_KEY para asociar apodos padre e hijo.

```
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' ,
NAMESPACES ' 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' ,
NAMESPACES ' soapenv="http://schemas.xmlsoap.org/soap/envelope/" ,
ns1="http://myzoo.com" ');

```

La clave foránea nn_fk en el apodo zooport_getzooreport_report_zookeeper_nn hace referencia al apodo padre zooport_getzooreport_nn en la opción de clave foránea. Las columnas designadas de apodo de clave primaria y foránea no se corresponden con los datos del documento WSDL porque estas columnas de apodo contienen claves generadas por el derivador. Estas claves identifican una relación entre los apodos padre e hijo que es exclusiva sólo en una consulta. Si el apodo hijo contiene una columna de entrada, la plantilla de la opción de apodo padre hace referencia a dicho apodo hijo en la estructura de plantillas con la opción de apodo.

La siguiente sentencia SQL une los apodos padre e hijo:

```

SELECT *
FROM zooport_getzooreport_nn a,
zooport_getzooreport_report_zookeeper_nn z,
WHERE a.nn_pk = z.nn_fk
AND a.zooId = 100
;

```

En la siguiente descripción se explica cómo utiliza el derivador de servicios web las opciones de apodo y columna TEMPLATE y XPATH durante la ejecución de la consulta. No está diseñado como un ejemplo de una implementación específica.

Cuando une las columnas de clave primaria y foránea, el derivador de servicios web envía un mensaje al proveedor de servicios web, y se devuelve un conjunto de filas del proveedor de servicios web. El derivador genera un mensaje para el apodo padre sustituyendo los valores de la columna de entrada (a.zooId = 100) de la consulta por la referencia en la plantilla de opción de columna (ZOOID VARCHAR (48) OPTIONS(TEMPLATE '&column')) y, a continuación, todas las referencias de columna en la opción de plantilla de apodo (<zooId>&zooId[1,1]</zooId>). La opción de plantilla de apodo puede incluir referencias de columna o referencias de apodo hijo. A continuación, el mensaje se envía al servicio web.

El derivador genera las filas para un apodo aplicando la opción de apodo XPATH en el documento que devuelve el servicio web. Si la opción de apodo XPATH devuelve varios fragmentos XML, el apodo contiene varias filas. La opción de columna XPATH se aplica en los fragmentos XML resultantes que representan las filas para obtener los valores de columna. Si un apodo tiene uno o varios padres

indirectos, todas las expresiones XPATH de apodo padre se aplican en orden desde la parte superior de la jerarquía hacia abajo antes de que la opción de apodo XPATH y la opción de columna XPATH se apliquen para este apodo.

Configuración del acceso a orígenes de datos XML

Puede integrar los datos que hay en los orígenes de datos XML con información de otros orígenes utilizando un sistema federado.

Para configurar un servidor federado para acceder a orígenes de datos XML, debe proporcionar al servidor federado información acerca de los orígenes de datos y los objetos a los que desea acceder. Una vez configurado el servidor federado, puede crear consultas para acceder a los orígenes de datos XML.

Derivador XML

XML (Extensible Markup Language) es un formato universal para documentos y datos estructurados. XML utiliza etiquetas para estructurar los datos de los documentos.

Los archivos XML utilizan la extensión de archivo .xml. XML utiliza, como HTML, palabras incluidas entre paréntesis angulares (< >) como etiquetas. Las etiquetas estructuran los datos contenidos en el documento.

El documento siguiente es un ejemplo de documento 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>
```

Figura 21. Documento XML de ejemplo

Funcionamiento de derivador XML

El derivador XML permite utilizar SQL para consultar los siguientes tipos de datos:

- Documentos XML externos almacenados en un único archivo

- Varios archivos XML en una vía de acceso de directorio
- Archivos XML remotos a los que se hace referencia con un identificador universal de recursos (URI)
- Documentos XML almacenados en columnas relacionales

En la figura siguiente se muestra el funcionamiento del derivador XML con el sistema federado.

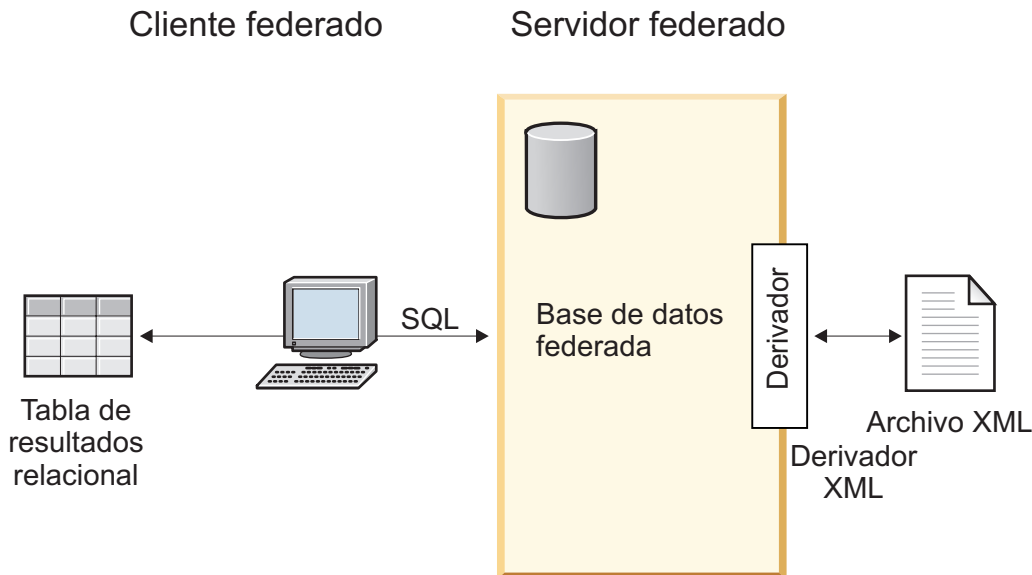


Figura 22. Funcionamiento de derivador XML

Con el derivador XML, puede correlacionar los datos XML de un origen de datos externo con un esquema relacional formado por un conjunto de apodos. La estructura de un documento XML es equivalente desde el punto de vista lógico a un esquema relacional en el que los elementos anidados y repetitivos se modelan de forma separada con claves foráneas.

Los apodos que corresponden a un documento XML se organizan en una estructura de árbol en la que los apodos hijo se correlacionan con elementos anidados en el elemento que corresponde al apodo padre.

Un apodo raíz es un apodo que se encuentra en el nivel más alto de una jerarquía de apodos. Un apodo no raíz es un apodo que tiene un apodo padre en una jerarquía de apodos. Puede tener apodos raíz que no se encuentren en el elemento de nivel superior de un documento XML.

Cuando los elementos anidados se repiten o tienen identidades distintas con estructuras complejas, pueden proporcionarse apodos distintos para cada elemento anidado.

Los apodos hijo y padre están conectados mediante claves primarias y foráneas generadas por el derivador.

Para correlacionar un documento XML con un esquema relacional formado por un conjunto de apodos, se utilizan expresiones XPath. XPath es un mecanismo de direccionamiento para identificar las partes de un archivo XML (por ejemplo, los grupos de nodos y atributos de un árbol de documentos XML). La sintaxis básica de XPath es parecida al direccionamiento del sistema de archivos.

Cada apodo se define mediante una expresión XPath que identifica los elementos XML que representan tuplas individuales, y un conjunto de expresiones XPath que especifica cómo extraer los valores de columna de cada elemento.

Ejemplo de correlación de documento XML

El siguiente ejemplo ilustra lo siguiente:

- Cómo se correlaciona el documento XML con un conjunto de apodos
- Cómo se establecen las relaciones padre e hijo utilizando claves primarias y foráneas
- Cómo se utilizan las expresiones XPath para definir tuplas y columnas individuales en cada elemento del documento
- Cómo puede ejecutarse una consulta en el documento XML tras registrar el documento en el sistema federado

El documento XML de ejemplo contiene un conjunto de elementos de cliente. Cada elemento incluye varios elementos de pedido (order) y pago (payment).

Los elementos de pedido incluyen varios elementos de artículo (item).

En la figura siguiente se muestra la relación que existe entre los elementos.

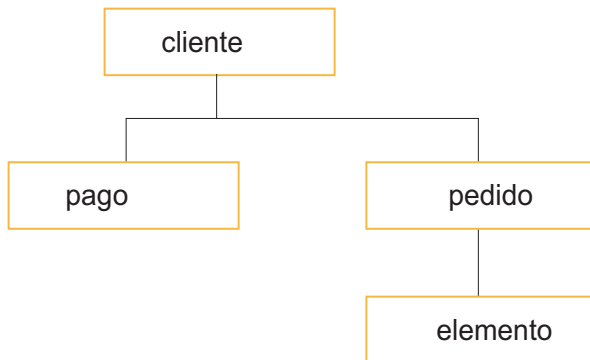


Figura 23. Estructura de árbol del documento XML de ejemplo

Desde esta estructura, puede utilizar la sentencia CREATE NICKNAME para correlacionar el documento XML en un esquema relacional que incluye cuatro apodos:

- customers
- orders
- payments
- items

Las relaciones entre los apodos se definen especificando cada apodo como apodo padre o apodo hijo utilizando opciones de columnas de apodo especiales que especifican claves primarias y foráneas. Cada apodo padre debe tener una columna especial que se designa con la opción de columna de clave primaria. El hijo de un apodo padre se define con la opción de columna de clave foránea que hace referencia a la columna de clave primaria de un apodo padre. Las columnas de apodo primarias y foráneas designadas no se corresponden con los datos del documento XML porque estas columnas de apodo incluirán claves generadas por el derivador. Un apodo puede tener varios hijos, pero sólo puede tener un padre. El apodo raíz no tiene padre.

Para el documento XML de ejemplo, el apodo customers tiene una clave primaria definida, y los apodos orders, payments e items tiene claves foráneas definidas que apuntan al apodo padre. Las claves foráneas de los apodos orders y payments apuntan al apodo customers, y la clave foránea del apodo items apunta al apodo orders.

Para identificar los elementos XML que representan tuplas individuales, debe crear una expresión XPath. En este ejemplo se hace referencia a todos los elementos customer mediante la expresión XPath '/doc/customer', y a todos los elementos order, mediante la expresión XPath './order'. El punto de la expresión XPath './order' indica que las tuplas de cada elemento order están anidadas en las tuplas del elemento customer correspondiente.

Debe crear un conjunto de expresiones XPath para especificar cómo deben extraerse los valores de cada elemento. En este ejemplo se hace referencia al atributo id de los elementos customer, que ahora es una columna definida en el apodo, mediante la expresión Xpath './@id'. Para hacer referencia al elemento name de los elementos customer, se utiliza la expresión XPath './name', y se utiliza la expresión XPath './address/@street' para hacer referencia al elemento address de los elementos customer.

Una vez que se ha correlacionado el documento XML con un conjunto de apodos mediante la sentencia CREATE NICKNAME, debe definir cada apodo como padre o hijo utilizando claves primarias y foráneas. Debe especificar expresiones XPath en estas claves primarias y foráneas que definan tuplas y columnas individuales en cada elemento del documento. Luego puede ejecutar consultas SQL en el documento XML.

Adición de XML a un sistema federado

Para configurar un servidor federado para acceder a orígenes de datos XML, debe proporcionar al servidor federado información acerca de los orígenes de datos y los objetos a los que desea acceder.

Antes de empezar

- Federation debe estar instalado en el servidor que actuará como servidor federado.
- Debe existir una base de datos en el servidor federado.

Acerca de esta tarea

Puede configurar un servidor federado para acceder a los datos almacenados en los orígenes de datos XML utilizando el Centro de control o emitiendo sentencias SQL en la línea de mandatos. El Centro de control incluye un asistente que sirve de guía en los pasos necesarios para configurar los objetos federados que se necesitan.

Procedimiento

Para añadir orígenes de datos XML a un servidor federado:

1. Registre el derivador de XML.
2. Registre la definición de servidor XML.
3. Registre los apodos de los orígenes de datos XML.
4. Cree vistas federadas para los apodos no raíz.

Registro del derivador de XML

Debe registrar un derivador para acceder a los orígenes de datos XML.

Acerca de esta tarea

Los servidores federados utilizan derivadores para comunicarse con los orígenes de datos y recuperar datos de éstos. Los derivadores se implementan como un conjunto de archivos de biblioteca.

Puede registrar un derivador utilizando el Centro de control o la línea de mandatos. El Centro de control incluye un asistente que sirve de guía en los pasos necesarios para registrar el derivador.

Si utiliza un servidor proxy y un almacén de claves para acceder a los archivos XML, puede especificar la información del servidor proxy como opciones cuando registre la definición de servidor o el derivador. Si especifica la información del servidor proxy cuando registra la definición de servidor, estos valores alteran temporalmente los valores del servidor proxy especificados al registrar el derivador.

Procedimiento

Para registrar el derivador de XML:

Elija el método que desee utilizar para registrar el derivador de XML:

Método	Procedimiento
Mediante el Centro de control	Inicie el Asistente para objetos federados. Pulse con el botón derecho en la carpeta Objetos de base de datos federada y pulse Crear objetos federados .
Desde la línea de mandatos	Emita la sentencia CREATE WRAPPER. <pre>CREATE WRAPPER nombre_derivador LIBRARY nombre_biblioteca;</pre> <p>Si utiliza un servidor proxy o un almacén de claves para acceder a los archivos XML, debe especificar opciones de servidor cuando registre la definición de servidor o el derivador de XML. Para especificar la información del servidor proxy cuando registra el derivador de XML, la sentencia que emite es:</p> <pre>CREATE WRAPPER nombre_derivador LIBRARY nombre_biblioteca OPTIONS (PROXY_TYPE 'tipo', PROXY_SERVER_NAME 'nombre_servidor', PROXY_SERVER_PORT 'numero_puerto');</pre>

Debe especificar el parámetro LIBRARY en la sentencia CREATE WRAPPER. El nombre del archivo de biblioteca de derivador que especifica depende del sistema operativo del servidor federado. Consulte la lista de archivos de biblioteca de derivador de XML para ver el nombre de biblioteca correcto que debe especificar

en la sentencia CREATE WRAPPER.

Si utiliza un servidor proxy o un almacén de claves para acceder a los archivos XML, debe especificar opciones de servidor cuando registre la definición de servidor o el derivador de XML.

Archivos de biblioteca de derivador de XML:

Los archivos de biblioteca de derivador de XML se añaden al servidor federado cuando se instala el servidor federado.

Cuando instala IBM InfoSphere Federation Server, se añaden tres archivos de biblioteca a la vía de acceso de directorios predeterminada. Por ejemplo, si el servidor federado se ejecuta en AIX, los archivos de biblioteca que se añaden a la vía de acceso de directorios son libdb2lsxml.a, libdb2lsxmlF.a y libdb2lsxmlU.a. El archivo de biblioteca del derivador predeterminado es libdb2lsxml.a. Los otros archivos de biblioteca de derivador se utilizan con opciones de derivador específicas.

Debe incluir el parámetro LIBRARY en la sentencia CREATE WRAPPER y especificar el nombre de archivo de biblioteca del derivador predeterminado.

Las vías de acceso de directorios predeterminadas y los nombres de archivo de biblioteca del derivador predeterminados se especifican en la tabla siguiente.

Tabla 41. Nombres de archivo y ubicaciones de la biblioteca de derivador de XML

Sistema operativo	Vía de acceso de directorios	Nombre de archivo de biblioteca de derivador
AIX	/usr/opt/<vía_acceso_instalación>/lib32/ /usr/opt/<vía_acceso_instalación>/lib64/	libdb2lsxml.a
Linux	/opt/IBM/db2/<vía_acceso_instalación>/lib32 /opt/IBM/db2/<vía_acceso_instalación>/lib64	libdb2lsxml.so
Solaris	/opt/IBM/db2/<vía_acceso_instalación>/lib32 /opt/IBM/db2/<vía_acceso_instalación>/lib64	libdb2lsxml.so
Windows	%DB2PATH%\bin	db2lsxml.dll

<vía_acceso_instalación> es la vía de acceso de directorios donde se instala el servidor federado en Linux o UNIX.

%DB2PATH% es la variable de entorno que se utiliza para especificar la vía de acceso de directorios donde se instala el servidor federado en Windows. La vía de acceso de directorios predeterminada de Windows es C:\Archivos de programa\IBM\SQLLIB.

Sentencia CREATE WRAPPER - Ejemplos para el derivador XML:

Utilice la sentencia CREATE WRAPPER para registrar el derivador XML. Los ejemplos muestran los parámetros necesarios para acceder a documentos XML con y sin servidor proxy.

Registro de un derivador

Si no utiliza un servidor proxy para acceder a documentos XML, la sentencia que se emite para registrar el derivador es la siguiente:

```
CREATE WRAPPER
derivador_xml LIBRARY 'libdb2lxml.a';
```

derivador_xml

Nombre que se asigna al derivador XML. No se permite la duplicación de nombres de derivador.

LIBRARY 'libdb2lxml.a'

Nombre del archivo de biblioteca para los servidores federados que utilizan sistemas operativos AIX.

Registro de un derivador para un servidor proxy HTTP

Para registrar un derivador y especificar un servidor proxy HTTP, utilice esta sentencia:

```
CREATE WRAPPER derivador_xml LIBRARY
'libdb2lxml.a'
  OPTIONS (PROXY_TYPE 'HTTP', PROXY_SERVER_NAME 'proxy_http',
          PROXY_SERVER_PORT '8080');
```

PROXY_TYPE 'HTTP'

Especifica el tipo de proxy que se utiliza para acceder a Internet cuando el servidor se encuentra detrás de un cortafuegos. Los valores válidos son 'NONE', 'HTTP' o 'SOCKS'.

PROXY_SERVER_NAME 'proxy_http'

Especifica el nombre o la dirección IP del servidor proxy. Esta opción es necesaria si el valor para la opción de servidor PROXY_TYPE es 'HTTP' o 'SOCKS'.

PROXY_SERVER_PORT '8080'

Especifica el número de puerto del servidor proxy. Esta opción es necesaria si el valor para la opción de servidor PROXY_TYPE es 'HTTP' o 'SOCKS'.

Registro de un derivador para un servidor SOCKS

Para registrar un derivador y especificar un servidor proxy SOCKS sin información de autenticación, utilice esta sentencia:

```
CREATE WRAPPER
derivador_xml LIBRARY 'libdb2lxml.so'
  OPTIONS (PROXY_TYPE 'SOCKS', PROXY_SERVER_NAME
'proxy_socks',
          PROXY_SERVER_PORT '1081');
```

LIBRARY 'libdb2lxml.so'

Especifica el nombre del archivo de biblioteca de derivador para los servidores que utilizan los sistemas operativos Linux y Solaris.

PROXY_TYPE 'SOCKS'

Especifica el tipo de proxy que se utiliza para acceder a Internet cuando el servidor se encuentra detrás de un cortafuegos. Los valores válidos son 'NONE', 'HTTP' o 'SOCKS'.

PROXY_SERVER_NAME 'proxy_socks'

Especifica el nombre o la dirección IP del servidor proxy. Esta opción es necesaria si el valor para la opción de servidor PROXY_TYPE es 'HTTP' o 'SOCKS'.

PROXY_SERVER_PORT '1081'

Especifica el número de puerto del servidor proxy. Esta opción es necesaria si el valor para la opción de servidor PROXY_TYPE es 'HTTP' o 'SOCKS'.

Registro de la definición de servidor para un origen de datos XML

Debe registrar una definición de servidor para recuperar documentos XML. La definición de servidor que registre dependerá de si utiliza un servidor proxy para acceder a los documentos XML.

Acerca de esta tarea

Para recuperar los documentos XML que hay detrás de un cortafuegos utilizando un proxy y un identificador universal de recursos (URI), debe incluir las opciones del servidor proxy en la sentencia CREATE SERVER.

Si no utiliza un servidor proxy, también debe registrar una definición de servidor, porque la jerarquía de los objetos federados requiere que los archivos XML estén asociados con un objeto de definición de servidor específico.

Puede registrar una definición de servidor utilizando el Centro de control o la línea de mandatos. El Centro de control incluye un asistente que sirve de guía en los pasos necesarios para registrar la definición de servidor.

Procedimiento

Para registrar una definición de servidor para un origen de datos XML:

Elija el método que desee utilizar para registrar la definición de servidor:

Método	Procedimiento
Mediante el Centro de control	Utilice el Asistente para objetos federados. Pulse con el botón derecho en la carpeta Objetos de base de datos federada y pulse Crear objetos federados . Siga los pasos del asistente.
Desde la línea de mandatos	Emita la sentencia CREATE SERVER. Si no utiliza un servidor proxy para acceder a los documentos XML, la sentencia que emite es: <pre>CREATE SERVER nombre_definición_servidor WRAPPER nombre_derivador;</pre> Si utiliza un servidor proxy para acceder a los documentos XML, la sentencia que emite es: <pre>CREATE SERVER nombre_definición_servidor WRAPPER nombre_derivador OPTIONS (PROXY_TYPE 'tipo', PROXY_SERVER_NAME 'nombre_servidor', PROXY_SERVER_PORT 'número_puerto');</pre>

Si utiliza un servidor proxy para acceder a los documentos XML, el protocolo que utiliza puede necesitar que especifique un ID de autorización y una contraseña para el servidor proxy. Especifique la información de autenticación como opciones cuando registre la definición de servidor.

Sentencia CREATE SERVER - Ejemplos para el derivador XML:

Utilice la sentencia CREATE SERVER para registrar definiciones de servidor para el derivador XML. Los ejemplos muestran los parámetros necesarios para acceder a documentos XML con y sin servidor proxy.

Registro de una definición de servidor

Aun cuando no utilice un servidor proxy para acceder a documentos XML, debe registrar una definición de servidor. La jerarquía de objetos federados requiere que los archivos XML estén asociados con un objeto de definición de servidor específico. La sentencia que se emite para registrar una definición de servidor es la siguiente:

```
CREATE SERVER
servidor_xml WRAPPER derivador_xml;
```

servidor_xml

Nombre que se asigna a la definición del servidor XML. No se permiten los nombres de definición de servidor duplicados.

WRAPPER *derivador_xml*

Nombre de derivador que ha especificado en la sentencia CREATE WRAPPER.

Definiciones de servidor cuando se utiliza un servidor proxy

Debe utilizar las opciones de servidor proxy de la sentencia CREATE SERVER si se cumplen todas estas condiciones:

- Desea recuperar datos mediante un URI.
- El URI utilizado recuperará datos de detrás de un cortafuegos, a través de un proxy.
- El cortafuegos o el proxy utilizado es HTTP o SOCKS.

Las opciones que especifique dependen del tipo de servidor proxy al que desee acceder.

Póngase en contacto con el administrador de la red para obtener información sobre el tipo de proxy que se utiliza y los valores que debe especificar para las opciones del proxy.

Registro de una definición de servidor para un servidor proxy HTTP

Para registrar una definición de servidor y especificar un servidor proxy HTTP, utilice esta sentencia:

```
CREATE SERVER servidor_http_xml
  WRAPPER derivador_xml
  OPTIONS (PROXY_TYPE 'HTTP', PROXY_SERVER_NAME 'proxy_http',
          PROXY_SERVER_PORT '8080');
```

servidor_http_xml

Nombre que se asigna a la definición del servidor XML. No se permiten los nombres de definición de servidor duplicados.

WRAPPER *derivador_xml*

Nombre de derivador que ha especificado en la sentencia CREATE WRAPPER.

PROXY_TYPE 'HTTP'

Especifica el tipo de proxy que se utiliza para acceder a Internet cuando el servidor se encuentra detrás de un cortafuegos. Los valores válidos son 'NONE', 'HTTP' o 'SOCKS'.

PROXY_SERVER_NAME 'proxy_http'

Especifica el nombre o la dirección IP del servidor proxy. Esta opción es necesaria si el valor para la opción de servidor PROXY_TYPE es 'HTTP' o 'SOCKS'.

PROXY_SERVER_PORT '8080'

Especifica el número de puerto del servidor proxy. Esta opción es necesaria si el valor para la opción de servidor PROXY_TYPE es 'HTTP' o 'SOCKS'.

Registro de una definición de servidor para un servidor SOCKS

Para registrar una definición de servidor y especificar un servidor proxy SOCKS cuando la información de autenticación no es necesaria, utilice esta sentencia:

```
CREATE SERVER servidor_socks_xml
  WRAPPER derivador_xml
  OPTIONS (PROXY_TYPE 'SOCKS', PROXY_SERVER_NAME 'proxy_socks',
    PROXY_SERVER_PORT '1081');
```

servidor_socks_xml

Nombre que se asigna a la definición del servidor XML. No se permiten los nombres de definición de servidor duplicados.

WRAPPER *derivador_xml*

Nombre de derivador que ha especificado en la sentencia CREATE WRAPPER.

PROXY_TYPE 'SOCKS'

Especifica el tipo de proxy que se utiliza para acceder a Internet cuando el servidor se encuentra detrás de un cortafuegos. Los valores válidos son 'NONE', 'HTTP' o 'SOCKS'.

PROXY_SERVER_NAME 'proxy_socks'

Especifica el nombre o la dirección IP del servidor proxy. Esta opción es necesaria si el valor para la opción de servidor PROXY_TYPE es 'HTTP' o 'SOCKS'.

PROXY_SERVER_PORT '1081'

Especifica el número de puerto del servidor proxy. Esta opción es necesaria si el valor para la opción de servidor PROXY_TYPE es 'HTTP' o 'SOCKS'.

Registro de una definición de servidor para un servidor SOCKS con información sobre autenticación

Para registrar una definición de servidor y especificar un servidor proxy SOCKS con información de autenticación, utilice esta sentencia:

```
CREATE SERVER servidor_socks_xml
  WRAPPER derivador_xml
  OPTIONS (PROXY_TYPE 'SOCKS', PROXY_SERVER_NAME 'proxy_socks',
    PROXY_SERVER_PORT '1081', PROXY_AUTHID 'Martin',
    PROXY_PASSWORD 'not4me');
```

PROXY_AUTHID 'Martin'

Especifica el nombre de usuario en el servidor proxy. Esta opción es necesaria si el valor de la opción de servidor PROXY_TYPE es 'SOCKS'.

PROXY_PASSWORD 'not4me'

Especifica la contraseña del servidor proxy asociada con el nombre de usuario 'Martin'. Esta opción es necesaria si el valor de la opción de servidor PROXY_TYPE es 'SOCKS'.

Especificación de un límite de tiempo en respuestas de servidor proxy

Además de las opciones de servidor necesarias para los servidores proxy, también puede especificar la opción de servidor SOCKET_TIMEOUT cuando registre una definición de servidor. La opción de servidor SOCKET_TIMEOUT especifica el tiempo máximo, en minutos, durante el que el servidor esperará los resultados del servidor proxy. Si no se especifica la opción de servidor SOCKET_TIMEOUT, no existe límite de tiempo. El servidor federado esperará indefinidamente los resultados del servidor proxy.

Para registrar una definición de servidor y especificar el tiempo durante el que el servidor federado espera una respuesta del servidor proxy, utilice esta sentencia:

```
CREATE SERVER servidor_http_xml
  WRAPPER derivador_xml
  OPTIONS (PROXY_TYPE 'HTTP', PROXY_SERVER_NAME 'proxy_http',
    PROXY_SERVER_PORT '8080', SOCKET_TIMEOUT '12');
```

SOCKET_TIMEOUT 12

Especifica que el servidor federado esperará 12 minutos para obtener una respuesta del servidor proxy.

Acceso a archivos XML mediante un servidor proxy

Si la red utiliza un servidor proxy, debe especificar información acerca de dicho servidor al registrar el derivador o la definición del servidor para los orígenes de datos XML.

Las opciones que especifique dependen del tipo de servidor proxy al que desee acceder, y de si utiliza un protocolo SSL (Secure Socket Layer) o TLS (Transport Layer Security).

Puede especificar el proxy y las opciones SSL cuando registre un derivador o una definición de servidor:

- Si especifica estas opciones al registrar el derivador, los apodos asociados con el derivador utilizarán las opciones establecidas para el derivador.
- Si especifica estas opciones al registrar una definición de servidor, los apodos asociados con dicha definición de servidor utilizarán las opciones establecidas para la definición de servidor.
- Si especifica valores distintos para estas opciones al registrar el derivador y la definición de servidor, los valores establecidos para la definición de servidor prevalecerán sobre aquellos establecidos para el derivador.

El derivador XML tiene una característica de validación que puede presentar limitaciones si se utiliza con el servidor proxy. Las condiciones en las que verá estas limitaciones son:

- Utiliza la característica de proxy, en el nivel de servidor, y ha establecido distintas opciones de proxy.
- El documento de la instancia XML contiene una referencia aun esquema XML externo que se encuentra fuera del cortafuegos.

Si tiene una de estas condiciones, intente cambiar la ubicación del esquema XML por una ubicación dentro del cortafuegos. Si cambia la ubicación del esquema XML, debe actualizar el documento de la instancia XML con la nueva ubicación del esquema XML.

Opciones de servidor proxy

En la tabla siguiente se indican las opciones que debe especificar para cada tipo de servidor proxy:

Tabla 42. Opciones necesarias con los servidores proxy

Tipo de servidor proxy	Opciones de derivador o de servidor necesarias
HTTP o SOCKS sin autenticación	PROXY_TYPE PROXY_SERVER_NAME PROXY_SERVER_PORT
HTTP o SOCKS con autenticación	PROXY_TYPE PROXY_SERVER_NAME PROXY_SERVER_PORT PROXY_AUTHID PROXY_PASSWORD

Opciones de servidor SSL

En la tabla siguiente se indican las opciones que debe especificar si utiliza los protocolos SSL:

Tabla 43. Opciones necesarias con los protocolos SSL

Objeto federado	Opciones necesarias
Derivador	SSL_KEYSTORE_FILE SSL_KEYSTORE_PASSWORD SSL_VERIFY_SERVER_CERTIFICATE
Definición de servidor	SSL_KEYSTORE_FILE SSL_KEYSTORE_PASSWORD SSL_VERIFY_SERVER_CERTIFICATE SSL_CLIENT_CERTIFICATE_LABEL

Apodos para orígenes de datos XML

Debe registrar un apodo para cada documento XML al que desee acceder. Utilice estos apodos, en lugar de los nombres de documento, cuando consulte orígenes de datos XML.

Puede registrar apodos para orígenes de datos XML utilizando el Centro de control o la línea de mandatos. El Centro de control simplifica el proceso de creación de apodos XML. Antes de registrar los apodos, debe comprender lo siguiente:

Asociaciones de datos entre apodos y documentos XML:

Los apodos corresponden a la estructura de árbol de los datos de documentos XML. Los apodos padre y los apodos hijo corresponden a la estructura raíz y a los elementos anidados de la estructura de árbol de datos. Estos apodos padre e hijo están conectados mediante claves primarias y foráneas que se especifican mediante la sentencia CREATE NICKNAME.

Cada apodo se define mediante expresiones XPath que realizan las siguientes funciones:

- Identifica los elementos XML que representan tuplas individuales.
- Especifica cómo extraer los valores de columna de cada elemento.

El derivador XML utiliza expresiones XPath para establecer una correspondencia entre los datos del documento XML y las filas de la tabla de datos relacional. Estas expresiones XPath identifican los valores del documento XML y determinan la correspondencia entre estos valores y las columnas de cada fila. El derivador XML sólo lee los datos del documento XML. El derivador XML no actualiza estos datos.

Al crear un apodo, debe seleccionar opciones que especifican la asociación entre el apodo y el documento XML. Los apodos están asociados con los documentos XML de forma fija o con los nombres de origen que se especifiquen.

Con una asociación fija, el apodo representa datos de documentos XML específicos. Estos documentos XML incluyen:

Un archivo local

Debe especificar un archivo XML como documento XML.

Varios archivos locales en una vía de acceso de directorio

Debe especificar una vía de acceso de directorio en la que residan varios archivos XML. Los archivos XML de esta vía de acceso de directorio proporcionan los datos del documento XML al apodo. Todos los archivos XML deben tener la misma configuración. Si uno de los archivos XML del directorio tiene una configuración distinta de la configuración del apodo, el derivador XML devuelve valores nulos cuando procesa dicho archivo de datos XML. El directorio debe ser local para el servidor federado o debe ser posible acceder al directorio desde un sistema de archivos compartido.

Al explorar el directorio, el derivador XML retiene y analiza sólo los archivos cuya extensión es `.xml`. El derivador XML ignora todos los demás archivos, incluidos aquellos que tienen la extensión `.txt`, la extensión `.xsd` y los archivos que no tienen extensión alguna.

Utilice la opción `FILE_PATH` de la sentencia `CREATE NICKNAME` para especificar datos fijos de un archivo. Utilice la opción `DIRECTORY_PATH` para especificar datos fijos de un directorio.

Si especifica los datos de origen mientras se ejecuta la consulta, puede utilizar el apodo para representar los datos de cualquier fuente de documentos XML cuyo esquema coincida con la definición del apodo. Estos documentos XML incluyen:

Identificadores de referencia uniformes (URI)

Un archivo XML remoto al que hace referencia un URI suministra los datos del documento XML al apodo. Especifique este origen de documento mediante la opción de la columna de apodo `DOCUMENT 'URI'`.

Columnas relacionales

Las columnas de una tabla relacional, vista o apodo se utilizan como entrada para el documento XML. Especifique este origen de documento mediante la opción de la columna de apodo `DOCUMENT 'COLUMN'`.

Archivo

Un único archivo que contiene datos XML se suministra como entrada mientras se ejecuta la consulta. Especifique este origen de documento mediante la opción de la columna de apodo `DOCUMENT 'FILE'`.

Directorio

Varios archivos XML de una vía de acceso de directorio determinada suministran los datos mientras se ejecuta la consulta. Especifique este origen de documento mediante la opción de la columna de apodo DOCUMENT 'DIRECTORY'.

Debe especificar la opción de columna DOCUMENT para indicar que los datos de origen se suministran mientras se ejecuta la consulta. Especifique URI, COLUMN, FILE o DIRECTORY con la columna DOCUMENT para indicar el tipo de origen de documentos XML.

No es posible especificar una opción FILE_PATH o una opción DIRECTORY_PATH con la opción de columna DOCUMENT.

Utilice la opción STREAMING para separar los datos de documentos XML en fragmentos. Puede utilizar la opción STREAMING con datos que tienen un formato fijo o con datos que provienen de nombres de origen especificados al ejecutar una consulta. El derivador XML procesa la secuencia resultante de datos XML y extrae la información solicitada por un fragmento de consulta. El derivador XML analiza un fragmento cada vez. Dado que los fragmentos se analizan uno por uno, el uso total de memoria disminuye. El tiempo de proceso necesario para ejecutar toda la consulta aumenta en función de la capacidad de memoria del servidor. Por consiguiente, utilice la opción STREAMING sólo para analizar documentos XML cuyo tamaño sea de 50 MB o más.

También puede seleccionar valores de opciones de apodo que ayuden a optimizar las consultas que recuperan grandes cantidades de datos XML o datos que contienen varios elementos anidados. Estas opciones son:

- INSTANCE_PARSE_TIME
- XPATH_EVAL_TIME
- NEXT_TIME

Puede definir valores para estas opciones para probar y optimizar la consulta XML. Estos valores de opción controlan el tiempo de proceso necesario para localizar elementos y para analizar los datos de las filas del documento XML.

Recurso de modelos de coste para el derivador XML:

El derivador XML proporciona un recurso de modelos de coste para optimizar las consultas sobre apodos que corresponden a los documentos origen XML.

Cuando crea un apodo mediante la sentencia CREATE NICKNAME, puede especificar las siguientes opciones de apodo para dar soporte al recurso de modelos de coste:

- INSTANCE_PARSE_TIME
- XPATH_EVAL_TIME

Puede utilizar los valores predeterminados para estas opciones de apodo. También puede establecer los valores para estas opciones de apodo para optimizar las consultas sobre los apodos raíz y no raíz que cree.

La opción de apodo INSTANCE_PARSE_TIME indica la cantidad de tiempo, en milisegundos, necesaria para leer y analizar un elemento raíz que produce filas del apodo raíz. El tiempo de análisis incluye todos los elementos contenidos no raíz que producen filas. Por ejemplo, si el apodo raíz es clientes, los elementos no raíz

son todos los elementos que corresponden a los pedidos, pagos y artículos de cada cliente. El derivador XML crea una estructura en la memoria para representar estos elementos raíz y no raíz que producen filas.

La opción de apodo XPATH_EVAL_TIME indica la cantidad de tiempo, en milisegundos, necesaria para evaluar las expresiones XPath que localizan los datos correspondientes a una fila del apodo. Las expresiones XPath que se evalúan incluyen las expresiones XPath que localizan las filas reales y aquellas que localizan los valores de columna dentro de dichas filas.

Espacios de nombres para orígenes de datos XML:

Utilice la opción de columna de apodo NAMESPACES para identificar los elementos o atributos que forman parte de un espacio de nombres.

Puede especificar la opción de apodo NAMESPACES cuando registre apodos. El valor de la opción de columna de apodo NAMESPACES es una lista separada por comas de pares formados por nombre y valor. El derivador XML utiliza los pares de nombre y valor para resolver los prefijos de espacio de nombres que se encuentran en expresiones XPath de apodo. Los prefijos que se utilizan en expresiones XPath se procesan mediante el procesador XPath.

En el ejemplo siguiente, el documento XML incluye el nombre, el código y una descripción para los tres productos. El documento XML declara dos espacios de nombres, `http://www.one.com` y `http://www.two.com`, y tiene un espacio de nombres predeterminado `http://www.default.com`. El elemento `product` está asociado con el espacio de nombres `ns1`. El elemento `product` contiene los atributos `name` y `code` y el elemento `desc`. El atributo `name` no está asociado con ningún espacio de nombres. El atributo `code` está asociado con el espacio de nombres `ns2`. El elemento `desc` está asociado con el espacio de nombres `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="Computer" ns2:code="ABC123"
  <desc>"The Computer product description"</desc>
<ns1:product name="Keyboard" ns2:code="EFG456"
  <desc>"The Keyboard product description"</desc>
<ns1:product name="Mouse" ns2:code="HIJ789"
  <desc>"The Mouse product description"</desc>
</ns1:product>
</doc>
```

En la tabla siguiente se muestra el espacio de nombres asociado con cada elemento y atributo del documento XML.

Tabla 44. Elementos, atributos y espacios de nombre del documento XML

Elemento o atributo	Espacio de nombres en el documento XML	Notas
product: elemento del documento XML.	ns1="http://www.one.com"	Ninguno
name: atributo del elemento product del documento XML.	Ninguno	El atributo name no está asociado con ningún espacio de nombres.
code: atributo del elemento product en el documento XML.	ns2="http://www.two.com"	Ninguno

Tabla 44. Elementos, atributos y espacios de nombre del documento XML (continuación)

Elemento o atributo	Espacio de nombres en el documento XML	Notas
desc: elemento del elemento product en el documento XML.	"http://www.default.com"	El elemento desc utiliza el espacio de nombres default, que no contiene ningún prefijo.

Al registrar el apodo para el documento XML, debe definir tres columnas que se correspondan con los elementos y atributos del documento XML. Debe especificar la información del espacio de nombre en la opción de apodo NAMESPACES. Por ejemplo:

```
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 xml_server
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");
```

En la tabla siguiente se muestra la correspondencia entre los espacios de nombres del documento XML y los valores que se especifican en la sentencia CREATE NICKNAME.

Tabla 45. Espacios de nombres del documento XML y valores correspondientes de la sentencia CREATE NICKNAME

Espacio de nombres en el documento XML	Nombre de columna en la sentencia CREATE NICKNAME	Valor en la opción de columna XPATH	Valor en la opción de apodo NAMESPACES
El espacio de nombres es ns1="http://www.one.com".	Ninguno	Ninguno	El valor es pre1="http://www.one.com". El valor es un prefijo que se define (pre1) y el identificador exclusivo para el espacio de nombres ("http://www.one.com").
Ninguno. El atributo no se asocia con ningún espacio de nombres.	Name	El valor es @name. El valor es un atributo del documento XML (name).	Ninguno
El espacio de nombres es ns2="http://www.two.com".	Code	El valor es @pre2:code. El valor es un prefijo que se define (pre2) y un atributo del documento XML (code).	El valor es pre2="http://www.two.com". El valor es un prefijo que se define (pre2) y el identificador exclusivo para el espacio de nombres ("http://www.two.com").
El espacio de nombres es "http://www.default.com". El espacio de nombres predeterminado no contiene ningún prefijo.	Description	El valor es ./default:desc. El valor es un prefijo que se define (default) y un elemento del documento XML (desc).	El valor es default="http://www.default.com". El valor es un prefijo que se define (default) y el identificador exclusivo para el espacio de nombres ("http://www.default.com").

La opción de apodo NAMESPACES utiliza descriptores empaquetados para dar soporte a series que tienen más de 256 caracteres.

Para obtener más información sobre los espacios de nombres XML, consulte la explicación sobre espacios de nombres del sitio web W3C.

Registro de apodos para los orígenes de datos XML:

Para cada definición de servidor XML, debe registrar un apodo para cada documento XML al que desee acceder. Utilice estos apodos en lugar de los nombres de los documentos XML cuando consulte los orígenes de datos XML.

Restricciones

Si intenta acceder a orígenes de datos XML que están en una unidad compartida desde un servidor federado que ejecuta Windows 2003, las consultas pueden fallar. Esta es una limitación de Windows 2003. Para evitar este problema, especifique la vía de acceso absoluta en la opción FILE_PATH de la sentencia CREATE NICKNAME.

Debe crear apodos correspondientes a la estructura de árbol del origen de datos XML. Los apodos padre corresponden a la estructura raíz del árbol. Los apodos hijo corresponden a los elementos anidados en el elemento del apodo padre.

Puede registrar apodos utilizando el Centro de control o la línea de mandatos. El Centro de control incluye un asistente que sirve de guía en los pasos necesarios para registrar la definición de servidor.

Procedimiento

Para registrar los apodos de los orígenes de datos XML:

Elija el método que desee utilizar para registrar la definición de servidor:

Método	Procedimiento
Mediante el Centro de control	Utilice el Asistente para objetos federados. Pulse con el botón derecho en la carpeta Objetos de base de datos federada y pulse Crear objetos federados . Siga los pasos del asistente.
Desde la línea de mandatos	Emita la sentencia CREATE NICKNAME. Los apodos pueden tener una longitud de hasta 128 caracteres.

Sentencia CREATE NICKNAME - Ejemplos para el derivador de XML:

Utilice la sentencia CREATE NICKNAME para registrar apodos para los documentos XML. Se proporciona un ejemplo completo que muestra cómo crear apodos padre e hijo, y ejemplos para las opciones de columna específicas.

Recomendación: No utilice el operador auto o descendiente // cuando especifique opciones de apodo y columnas XPATH en las consultas. El operador auto o descendiente es un operador XPath. El uso del operador auto o descendiente puede disminuir el rendimiento del servidor federado.

Ejemplo completo

El siguiente ejemplo muestra cómo crear apodos para los orígenes de datos XML utilizando el archivo XML de ejemplo, tal como se muestra en el siguiente archivo XML de ejemplo.

```
<?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>
```

Figura 24. Archivo XML de ejemplo

El apodo padre

El primer paso es crear el apodo padre, customers. Para crear el apodo, emita la siguiente sentencia:

```
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 xml_server
OPTIONS(DIRECTORY_PATH '/home/db2user',
        XPATH '/doc/customer', STREAMING 'YES');
```

Esta sentencia crea el apodo customers para varios archivos XML en la vía de acceso de directorios especificada, /home/db2user.

La opción de apodo STREAMING indica que el origen de datos XML está separado y procesado por un nodo, en este ejemplo, por un registro de cliente. Cuando se utiliza la opción de apodo STREAMING, el derivador no almacena el documento XML completo en la memoria. En su lugar, el derivador de XML divide el documento en varias secciones que se analizan individual y secuencialmente. La opción de apodo STREAMING sólo debe utilizarse con documentos XML de gran tamaño. El rendimiento de las consultas se ve afectado cuando utiliza esta opción.

Los apodos hijo

El siguiente paso es crear los apodos hijo para los elementos orders, payments e items.

Emita la siguiente sentencia para crear el apodo hijo 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 xml_server  
OPTIONS( XPATH './order');
```

Emita la siguiente sentencia para crear el apodo hijo 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 xml_server  
OPTIONS( XPATH './payment');
```

Emita la siguiente sentencia para crear el apodo hijo 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 xml_server  
OPTIONS( XPATH './item');
```

Ejemplos de la opción de columna de apodo

Los ejemplos siguientes muestran cómo incluir las opciones de columna de apodo DOCUMENT cuando crea apodos. Los ejemplos también muestran cómo se utilizan estas opciones en las consultas.

Ejemplo de DOCUMENT 'FILE'

Los ejemplos siguientes muestran cómo incluir las opciones de columna de apodo DOCUMENT cuando crea apodos. Los ejemplos también muestran cómo se utilizan estas opciones en las consultas.

El siguiente ejemplo CREATE NICKNAME muestra el uso de la opción de columna de apodo 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 xml_server  
OPTIONS(XPATH '/doc/customer');
```

A continuación, puede ejecutar la siguiente consulta en el apodo customers, especificando la ubicación del documento XML en la cláusula WHERE:

```
SELECT * FROM customers WHERE doc = '/home/db2user/Customers.xml';
```

Ejemplo de DOCUMENT 'DIRECTORY'

El siguiente ejemplo CREATE NICKNAME muestra el uso de la opción de columna de apodo 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 xml_server  
OPTIONS(XPATH '/doc/customer');
```

A continuación, puede ejecutar la siguiente consulta en el apodo customers:

```
SELECT name FROM customers WHERE doc = '/home/data/xml';
```

Esta consulta recupera los documentos XML que se encuentran en la vía de acceso de directorios /home/data/xml, que se especifica en la cláusula WHERE.

Ejemplo de DOCUMENT 'URI'

El siguiente ejemplo CREATE NICKNAME muestra el uso de la opción de columna de apodo 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 xml_server  
OPTIONS(XPATH '/doc/customer');
```

A continuación, puede ejecutar la siguiente consulta en el apodo customers para recuperar los datos XML de la ubicación remota:

```
SELECT * FROM customers WHERE doc = 'http://www.lg-mv.org/foo.xml';
```

Ejemplo de DOCUMENT 'COLUMN'

El siguiente ejemplo CREATE NICKNAME muestra el uso de la opción de columna de apodo 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 xml_server  
OPTIONS(XPATH '/doc/name');
```

A continuación, puede ejecutar una de las siguientes consultas en el apodo emp para recuperar los datos 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>';
```

o bien

```
SELECT * FROM emp WHERE doc = (SELECT * FROM xml_tab);
```

La tabla `xml_tab` contiene una columna que se rellena con los datos XML.

Consultas para orígenes de datos XML

Antes de crear consultas para acceder a orígenes de datos, debe llevar a cabo algunas acciones para optimizar el rendimiento de las consultas.

Vistas federadas

Puede utilizar las vistas federadas para asegurarse de que las consultas que unen los fragmentos de una jerarquía de apodos XML se ejecutan correctamente.

Cómo evitar el operador automático o descendiente

No utilice el operador automático o descendiente `//` cuando especifique opciones de apodos y columnas XPATH al crear apodos XML. El operador automático o descendiente es un operador XPath, y su uso puede repercutir negativamente en el rendimiento del servidor federado.

Servidores federados Windows 2003

Si intenta acceder a orígenes de datos XML que se encuentran en una unidad compartida desde un servidor federado que ejecuta Windows 2003, es posible que la consulta falle con el siguiente mensaje de error:

```
SQL1822N Unexpected error code "ERRNO = 2" received from data source
"XML_SERVER". Associated text and tokens are "Unable to read file".
SQLSTATE=560BD
```

Se trata de una limitación de Windows 2003. Para evitar este problema, especifique la vía de acceso absoluta en la opción `FILE_PATH` de la sentencia `CREATE NICKNAME`.

El ejemplo siguiente muestra una sentencia `CREATE NICKNAME` en la que se ha especificado una vía de acceso abreviada en la opción `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 xml_server
  OPTIONS(DIRECTORY_PATH '\home\db2user',
          XPATH '/doc/customer', STREAMING 'YES');
```

Las consultas que utilizan este apodo pueden fallar debido a la vía de acceso abreviada.

Para los servidores que ejecutan Windows 2003, especifique la vía de acceso absoluta en la opción `FILE_PATH` de la sentencia `CREATE NICKNAME`.

Por ejemplo:

```
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 xml_server
        OPTIONS(DIRECTORY_PATH '\\host.svl.ibm.com\D$\home\db2user',
                XPATH '/doc/customer', STREAMING 'YES');

```

Creación de vistas federadas para los apodos de derivador de XML

Puede crear vistas federadas para la jerarquía de apodos que describen un documento XML. La definición de vistas federadas garantiza que las consultas que unen las partes de una jerarquía de apodos XML se ejecutan correctamente.

Acerca de esta tarea

Una *vista federada* es una vista de la base de datos federada que hace referencia a un apodo, en lugar de a una tabla de origen de datos.

En la jerarquía de apodos XML, el apodo raíz y las consultas que unen columnas, excepto las columnas especiales PRIMARY_KEY y FOREIGN_KEY, no se ven afectadas por el uso de vistas federadas.

Cuando crea vistas federadas para los apodos XML, debe incluir todos los predicados necesarios y una vía de acceso completa al directorio raíz.

Procedimiento

Para crear vistas federadas para los apodos XML:

Utilice la sentencia CREATE VIEW para definir una vista para cada apodo no raíz. La vista debe ser una unión de todos los apodos en la vía de acceso al apodo raíz.

1. En la cláusula WHERE de la vista, defina las columnas PRIMARY_KEY y FOREIGN_KEY como los predicados de unión.
2. En la lista SELECT, incluya todas las columnas del apodo no raíz, excepto la columna designada con la opción de columna de apodo FOREIGN_KEY. En la lista SELECT, incluya la columna del apodo padre designada con la opción PRIMARY_KEY.

Sentencia CREATE VIEW - Ejemplos para el derivador XML

Utilice la sentencia CREATE VIEW para crear vistas federadas para apodos que no son raíz. Este ejemplo incluye un archivo XML de ejemplo, las sentencias que se utilizan para crear vistas y muestra cómo utilizar las vistas en una consulta.

Puede crear vistas federadas sobre la jerarquía de apodos que describen un documento XML para asegurarse de que las consultas que unen las partes de una jerarquía de apodos XML se ejecutan correctamente. Al especificar una vista federada en una consulta, los datos se recuperan del origen de datos remoto.

En los ejemplos siguientes se muestra cómo crear vistas federadas para apodos que no son raíz para describir documentos origen XML.

Archivo XML de ejemplo

Los ejemplos siguientes se basan en este archivo XML de ejemplo.

```

<?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>

```

Figura 25. Archivo XML de ejemplo

Sentencias CREATE VIEW para los apodos que no son raíz

El ejemplo siguiente muestra cómo crear una vista federada para el apodo order que no es raíz:

```

CREATE VIEW order_view AS
  SELECT o.amount, o.date, o.oid, c.cid
  FROM customers c, orders o
  WHERE c.cid = o.cid;

```

El ejemplo siguiente muestra cómo crear una vista federada para el apodo payment que no es raíz:

```

CREATE VIEW payment_view AS
  SELECT p.number, p.date, c.cid
  FROM customers c, payments p
  WHERE c.cid = p.cid;

```

El ejemplo siguiente muestra cómo crear una vista federada para el apodo item que no es raíz:

```

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;

```

Consulta que utiliza las vistas federadas

Las consultas que se emiten a las vistas federadas se procesan correctamente porque la vía de acceso de unión del directorio raíz se especifica en la cláusula WHERE.

La consulta siguiente, por ejemplo, utiliza el número de identificación del cliente y la fecha de emisión del pedido para devolver la cantidad pedida y el importe debido. En lugar de utilizar apodos en la consulta, las vistas se especifican en la cláusula FROM.

```
SELECT o.amount, p.amount
FROM order_view o, payment_view p
WHERE p.date = o.date AND
      p.cid = o.cid;
```

Sugerencias para la optimización de consultas para el recurso de modelos de costes XML

El recurso de modelos de coste para el derivador XML optimiza las consultas en los apodos que se creen.

El recurso de modelos de costes utiliza las siguientes opciones de apodo de la sentencia CREATE NICKNAME:

- INSTANCE_PARSE_TIME
- XPATH_EVAL_TIME

Puede especifica valores para estas opciones de apodo al emitir la sentencia CREATE NICKNAME para registrar un apodo para un origen de datos XML.

El recurso de modelos de costes utiliza estos valores de parámetro para determinar la cantidad de tiempo necesaria para analizar los datos de cada fila de un documento origen XML. Los valores de parámetro también se utilizan para evaluar la expresión XPath para el apodo.

Puede utilizar los valores predeterminados para estas opciones de apodo. No obstante, si desea optimizar las consultas en estructuras origen XML de gran tamaño o complejas para los apodos que cree, utilice el siguiente ejemplo como guía.

Ejemplo de optimización de una consulta de gran tamaño

El documento XML tiene un esquema relacional con cuatro apodos:

- customers
- orders
- payments
- items

El apodo raíz es customers.

Ejecute consultas en cada apodo. Ejecute cada consulta en una muestra de datos XML que sea típica para su entorno.

Por ejemplo:

```
SELECT * from customers;
SELECT * from orders;
SELECT * from payments;
SELECT * from items;
```

Anote del tiempo (en milisegundos) necesario para ejecutar cada consulta utilizando el mandato db2batch, o un mandato o programa de utilidad

equivalente. Puede utilizar el mandato db2batch para obtener un archivo de salida que contenga el tiempo necesario para ejecutar consultas. Anote el número de tuplas que se devuelven.

Para cada podo, utilice las fórmulas siguientes para determinar los valores óptimos para las opciones de apodo INSTANCE_PARSE_TIME y XPATH_EVAL_TIME:

```
INSTANCE_PARSE_TIME =  
  (75% X tiempo de ejecución de SELECT * consulta) ÷ número de tuplas devueltas  
XPATH_EVAL_TIME     =  
  (25% X tiempo de ejecución de SELECT * consulta) ÷ número de tuplas devueltas
```

Para el apodo raíz (en este ejemplo, customers), utilice los valores calculados para las opciones de apodo INSTANCE_PARSE_TIME y XPATH_EVAL_TIME.

Para los apodos que no son raíz (en este ejemplo, orders, payments e items), utilice sólo el valor calculado para el parámetro XPATH_EVAL_TIME. El valor del parámetro INSTANCE_PARSE_TIME no es aplicable a los apodos que no son raíz.

Puede utilizar estas fórmulas como guía para ajustar sus consultas. Los valores óptimos para estas opciones de apodo también dependen de la complejidad de los documentos origen XML y de la velocidad del procesador que se utilice.

Origen de datos XML - consultas de ejemplos

Ejemplos de consultas que utilizan apodos XML.

Estos ejemplos utilizan los apodos customers, orders e items.

Consulta que devuelve un valor específico de los documentos XML

Cuando se ejecuta la siguiente sentencia SELECT, el derivador devuelve los nombres de todos los clientes:

```
SELECT name FROM customers;
```

Consulta que devuelve todos los registros de un determinado cliente

Cuando se ejecuta la siguiente sentencia SELECT, el derivador devuelve todos los registros en los que el nombre de cliente es Chang:

```
SELECT * FROM customers  
  WHERE name='Chang';
```

Consulta que devuelve valores específicos basados en una condición de unión entre un apodo padre e hijo

Cuando se ejecuta la siguiente sentencia SELECT, el derivador devuelve los nombres de los clientes y las cantidades de cada pedido realizado por cada cliente. Debe especificar la unión, c.cid=o.cid, para indicar la relación padre-hijo entre el apodo customers y el apodo orders.

```
SELECT c.name, o.amount FROM customers c, orders o  
  WHERE c.cid=o.cid;
```

Consulta que muestra cómo especificar condiciones de unión entre un apodo padre y varios apodos hijo

Cuando se ejecuta la siguiente sentencia SELECT, el derivador devuelve las direcciones de los clientes, las cantidades de los pedidos y los nombres de los

artículos de cada pedido y artículo de cada cliente. Debe especificar las dos uniones para mantener las relaciones padre-hijo.

```
SELECT c.address, o.amount, i.name FROM customers c, orders o, items i
      WHERE c.cid=o.cid AND o.oid=i.oid;
```

Consultas que muestran cómo especificar un documento XML en una consulta

Los ejemplos siguientes muestran cómo grabar consultas utilizando un apodo que especifica una opción de columna de apodo DOCUMENT en lugar de una opción de apodo FILE_PATH.

La sentencia CREATE NICKNAME que se utiliza para crear el apodo customers es:

```
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 xml_server
OPTIONS(XPATH '/doc/customer');
```

La consulta siguiente selecciona todos los datos del archivo XML Customers.xml con la vía de acceso /home/db2user/Customers.xml:

```
SELECT * FROM customers
      WHERE doc='/home/db2user/Customers.xml';
```

La consulta siguiente selecciona los nombres de los clientes y las fechas de sus pedidos en el archivo Customers.xml para cada pedido de una cantidad superior a 1000. La vía de acceso /home/db2user/Customers.xml especifica la ubicación del archivo Customers.xml.

```
SELECT c.name, o.date FROM customers c, orders o
      WHERE c.doc='/home/db2user/Customers.xml' AND o.amount > 1000;
```


Capítulo 3. Soporte de origen de datos para opciones federadas

Consulte esta tabla cuando desee saber si un origen de datos soporta o no una característica federada determinada.

Antes de poder utilizar algunas de estas características, es posible que necesite establecer opciones de servidor o de derivador concretas, o realizar otras tareas para habilitar la función. Para obtener más información, consulte los temas concretos de cada característica.

Tabla 46. Características y orígenes de datos soportados

Característica	Orígenes de datos
Puntos de rescate de aplicación con operaciones WRITE en apodos	DB2 para Linux, UNIX y Windows
Optimización asíncrona	Todos los orígenes de datos
Tablas de memoria caché	Familia DB2 InformixMicrosoft SQL ServerOracleSybase
Importación de datos a apodos	Familia DB2 InformixMicrosoft SQL ServerOracleSybaseTeradata
Tolerancia a errores en expresiones de tablas anidadas	Familia DB2 InformixJDBCMicrosoft SQL ServerODBCOracleSybaseTeradata
Repositorio de correlación de usuarios externos	Todos los orígenes de datos
Indicadores de salud federados	Familia DB2 Excel InformixJDBCMicrosoft SQL ServerODBCOracleSybase Archivos con estructura de tablaTeradataXML (sólo apodos raíz)
Procedimientos federados	Familia DB2, en modalidad fiable Oracle, en modalidad fiable Microsoft SQL Server, en modalidad fiable Sybase, en modalidad delimitada, con el servidor federado instalado en UNIX Sybase, en modalidad delimitada o fiable, con el servidor federado instalado en Linux o Microsoft Windows
Contextos fiables federados	DB2 para Linux, UNIX y Windows Versión 9.5 DB2 para z/OS Versión 9 Oracle
Proxy HTTP	Servicios Web XML
Aislamiento a nivel de conexión	Familia DB2 InformixJDBCMicrosoft SQL ServerODBCOracleSybase
Control de acceso basado en etiquetas	DB2 para Linux, UNIX y Windows Versión 9.1 y 9.5 Oracle
Operaciones LOB de lectura y escritura	DB2 para z/OS DB2 para Linux, UNIX y Windows DB2 para System i Oracle

Tabla 46. Características y orígenes de datos soportados (continuación)

Característica	Orígenes de datos
Operaciones LOB de sólo lectura	BioRSInformixJDBCMicrosoft SQL ServerODBCScriptSybaseTeradataServicios Web XML
Tablas de consulta materializadas	Todos los orígenes de datos, con restricciones específicas
Recurso de actualización de estadísticas de apodo	BioRS Familia DB2 Excel InformixJDBCMicrosoft SQL ServerODBCOracleSybase Archivos con estructura de tablaTeradataXML (sólo apodos raíz)
Sesiones de paso a través	DRDA InformixOracleMicrosoft SQL ServerSybaseTeradata
Tipo de datos XML remotos	DB2 para Linux, UNIX y Windows Derivador XML
Proxy SOCKS	BioRSScriptServicios Web XML
Capa de sockets seguros (SSL)	Servicios Web XML
Aislamiento a nivel de sentencia	Familia DB2 Microsoft SQL Server
Transacciones de asignación de dos fases	DB2 para Linux, UNIX y Windows, en modalidad fiable DB2 para System i, en modalidad fiable DB2 para z/OS, en modalidad fiable Informix, en modalidad fiable Microsoft SQL Server, en modalidad fiable, con el servidor federado instalado en Microsoft Windows Oracle, en modalidad fiable Sybase, en modalidad fiable, con el servidor federado instalado en Microsoft Windows Sybase, en modalidad delimitada, con el servidor federado instalado en UNIX
Soporte para Unicode	Todos los orígenes de datos

Capítulo 4. Guía de consulta de opciones para orígenes de datos

Cada origen de datos soporta determinadas opciones de derivador, de servidor, de correlación de usuarios, de apodo y de columna.

Información de consulta sobre opciones de BioRS

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios, de apodo y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse en las sentencias CREATE WRAPPER y CREATE SERVER.

Tabla 47. Opciones de derivador para BioRS

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java™, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Esta opción es obligatoria si el valor de PROXY_TYPE es HTTP o SOCKS. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.

Tabla 47. Opciones de derivador para BioRS (continuación)

Nombre	Descripción
PROXY_SERVER_PORT	Especifica el puerto o el nombre de servicio del servicio proxy en el servidor proxy. Esta opción es obligatoria si el valor de PROXY_TYPE es HTTP o SOCKS. Los valores válidos son un número de puerto decimal comprendido entre 1 y 32760 o un nombre de servicio.
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE, HTTP y SOCKS. El valor predeterminado es NONE. Si establece esta opción en HTTP o SOCKS, también deberá especificar las opciones PROXY_SERVER_NAME y PROXY_SERVER_PORT.

Opciones de servidor

Tabla 48. Opciones de servidor para BioRS

Nombre	Descripción
CASE_SENSITIVE	Especifica si el servidor BioRS trata los nombres distinguiendo entre mayúsculas y minúsculas. Los valores válidos son Y y N. El valor predeterminado es Y (los nombres se tratan distinguiendo entre mayúsculas y minúsculas). En el producto BioRS, un parámetro de configuración controla si hay que distinguir entre mayúsculas y minúsculas en los datos almacenados en el servidor BioRS. La opción CASE_SENSITIVE y el parámetro de configuración deben especificar el mismo valor. Si tiene que cambiar el valor de la opción CASE_SENSITIVE después de crear la definición de servidor, deberá descartar la definición de servidor y crearla de nuevo, así como todos los apodos.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.

Tabla 48. Opciones de servidor para BioRS (continuación)

Nombre	Descripción
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
NODE	Obligatoria. Especifica el nombre de host DNS o la dirección IP del sistema en que está disponible la herramienta de consultas de BioRS. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6. El valor predeterminado es localhost.
PORT	Especifica el puerto de conexión del servidor BioRS. Los valores válidos son un puerto numérico o un nombre de servicio TCP/IP. El valor predeterminado es 5014.
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_SERVER_PORT	Especifica el puerto o el nombre de servicio del servicio proxy en el servidor proxy. Los valores válidos son un número de puerto decimal comprendido entre 1 y 32760 o un nombre de servicio.
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE, HTTP y SOCKS. El valor predeterminado es NONE.
TIMEOUT	Especifica el tiempo máximo, en minutos, que el servidor federado esperará una respuesta del servidor remoto. El valor predeterminado es 10.

Opciones de correlación de usuarios

Tabla 49. Opciones de correlación de usuarios para BioRS

Opción	Descripción
GUEST	Especifica que se utiliza el ID de autenticación GUEST de BioRS para realizar operaciones. Los valores válidos son Y y N. El valor predeterminado es N (el ID GUEST de BioRS no se utiliza). Esta opción no es válida si se especifican las opciones REMOTE_AUTHID y REMOTE_PASSWORD.
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy. La contraseña se cifra al almacenarla en el catálogo de la base de datos federada.
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.

Opciones de apodo

Tabla 50. Opciones de apodo para BioRS

Opción	Descripción
REMOTE_OBJECT	Especifica el nombre del banco de datos de BioRS asociado con el apodo. Este nombre determina el esquema y el banco de datos de BioRS del apodo. El nombre también especifica la relación del apodo con otros apodos. El que esta opción distinga entre mayúsculas y minúsculas depende de que lo haga el servidor BioRS y del valor de la opción de servidor CASE_SENSITIVE. No se puede utilizar la sentencia ALTER NICKNAME para cambiar o suprimir este nombre. Si el nombre del banco de datos de BioRS cambia, deberá suprimir el apodo y crearlo de nuevo.
TIMEOUT	Especifica el tiempo máximo, en minutos, que se esperará una respuesta del servidor de orígenes de datos. El valor predeterminado es 10.

Opciones de columna

Tabla 51. Opciones de columna para BioRS

Opción	Descripción
ELEMENT_NAME	Especifica el nombre de elemento de BioRS. El que este nombre distinga entre mayúsculas y minúsculas depende de que lo haga el servidor BioRS y del valor de la opción de servidor CASE_SENSITIVE. Debe especificar el nombre de elemento de BioRS solamente si es distinto del nombre de columna.
IS_INDEXED	Especifica si la columna correspondiente está indexada y, por lo tanto, si puede hacerse referencia a ella en un predicado. Los valores válidos son Y y N. El valor predeterminado es N (la columna no está indexada).
REFERENCED_OBJECT	Especifica el nombre del banco de datos de BioRS al que hace referencia la columna actual. El que este nombre distinga entre mayúsculas y minúsculas depende de que lo haga el servidor BioRS y del valor de la opción de servidor CASE_SENSITIVE. Esta opción sólo es válida para las columnas que tienen el tipo de datos Reference de BioRS.

Información de consulta sobre opciones de bases de datos de DB2

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios, de apodo y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a orígenes de datos DB2 y se identifican las opciones obligatorias que deben especificarse.

Tabla 52. Opciones de derivador para orígenes de datos DB2

Name	Description
DB2_FENCED	Necesario. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).

Tabla 52. Opciones de derivador para orígenes de datos DB2 (continuación)

Name	Description
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Opciones de servidor

Tabla 53. Opciones de servidor para orígenes de datos DB2

Name	Description
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N e I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.
COMM_RATE	Especifica la velocidad de comunicación, en megabytes por segundo, entre el servidor federado y el servidor de orígenes de datos. Los valores válidos son los números enteros mayores que 0 y menores que 2147483648. El valor predeterminado es 2.
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.

Tabla 53. Opciones de servidor para orígenes de datos DB2 (continuación)

Name	Description
DATE_COMPAT	Especifica si se aplica el parámetro <code>date_compat</code> a la base de datos. Los valores válidos son Y y N. El valor predeterminado es N. Esta opción de servidor sólo es válida para DB2 Database para Linux, UNIX y Windows, Versión 9.7 or posterior.
DBNAME	Necesario. Especifica la base de datos concreta a utilizar para la conexión de base de datos DB2 remota inicial. Esta base de datos concreta es el alias de base de datos para la base de datos DB2 remota catalogada en el servidor federado mediante el mandato <code>CATALOG DATABASE</code> o el asistente de configuración de DB2.
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
DB2_TWO_PHASE_COMMIT	Especifica si el servidor federado se conecta con el origen de datos mediante el protocolo de confirmación en dos fases o mediante el protocolo de confirmación en una fase. Los valores válidos son Y y N. El valor predeterminado es N (el servidor federado utiliza el protocolo de confirmación en una fase para conectarse). Y especifica que el servidor federado utiliza el protocolo de confirmación en dos fases para conectarse.
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, <code>"UserMappingRepositoryLDAP"</code> . En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.

Tabla 53. Opciones de servidor para orígenes de datos DB2 (continuación)

Name	Description
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
FED_PROXY_USER	Especifica el ID de autorización que hay que utilizar para establecer todas las conexiones de salida fiables cuando la conexión de entrada no es fiable. El usuario cuyo ID se especifica en esta opción debe tener una correlación de usuarios que especifique las opciones REMOTE_AUTHID y REMOTE_PASSWORD. Restricción: Esta opción de servidor sólo es válida para DB2 Database para Linux, UNIX y Windows, Versión 9.5 y posterior, y DB2 for z/OS Versión 9 y posterior.
FOLD_ID	Especifica si el ID de usuario se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía el ID de usuario en mayúsculas y, si éste falla, el servidor lo envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
FOLD_PW	Especifica si la contraseña se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía la contraseña en mayúsculas y, si ésta falla, el servidor la envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
IO_RATIO	Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.

Tabla 53. Opciones de servidor para orígenes de datos DB2 (continuación)

Name	Description
NO_EMPTY_STRING	<p>Especifica si el servidor de orígenes de datos remotos contiene series vacías. Los valores válidos son Y y N. El valor predeterminado varía en función del origen de datos remoto. Para orígenes de datos Oracle remotos, el valor predeterminado es Y; todos los valores de serie vacíos se convierten a valores NULL. Para el resto de orígenes de datos remotos, el valor predeterminado es N; el origen de datos puede contener series vacías.</p> <p>Puede mejorar el rendimiento de su sistema estableciendo esta opción a Y en las configuraciones del sistema en las que el servidor federado está en modalidad compatible con VARCHAR2, pero el origen de datos remoto no es compatible con VARCHAR2.</p>
NUMBER_COMPAT	<p>Especifica si el servidor de orígenes de datos admite el tipo de datos NUMBER. Los valores válidos son Y y N. El valor predeterminado es N; el servidor de orígenes de datos no admite el tipo de datos NUMBER. En los sistemas en los que el servidor federado no admite el tipo de datos NUMBER pero el servidor de orígenes de datos sí, debe establecer la opción NUMBER_COMPAT a Y porque el servidor de orígenes de datos puede devolver resultados DECFLOAT que están fuera del intervalo del tipo de datos DECIMAL y causan el error SQLSTATE 560BD.</p> <p>Restricción: Esta opción de servidor sólo es válida para DB2 Database para Linux, UNIX y Windows, Versión 9.7 y posterior.</p>
OLD_NAME_GEN	<p>Especifica cómo convertir los nombres de columna y los nombre de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.</p>

Tabla 53. Opciones de servidor para orígenes de datos DB2 (continuación)

Name	Description
PUSHDOWN	<p>Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el origen de datos evalúa operaciones). N especifica que el servidor federado envía sentencias de SQL que incluyen sólo SELECT con nombres de columna. En las sentencias de SQL que el servidor federado envía al origen de datos no se incluyen predicados, como WHERE=, funciones de columna y escalares, como MAX y MIN, clasificaciones, como ORDER BY o GROUP BY ni uniones.</p>
SAME_DECFLT_ROUNDING	<p>Especifica si la modalidad de redondeo del servidor federado y del servidor de orígenes de datos está utilizando la misma configuración de modalidad de redondeo DECFLOAT. Los valores válidos son Y y N. El valor predeterminado es N; el servidor federado y el servidor remoto tienen configuraciones de modalidad de redondeo DECFLOAT diferentes.</p> <p>Importante: Si establece esta opción a Y en los casos en que las modalidades de redondeo son distintas en el servidor federado y el servidor de orígenes de datos, es posible que reciba resultados de redondeo DECFLOAT incorrectos.</p> <p>Para configurar un servidor federado y un servidor de orígenes de datos existentes que utilizan la misma configuración de modalidad de redondeo DECFLOAT, utilice la sentencia ALTER SERVER.</p> <p>Restricción: Esta opción de servidor sólo es válida para DB2 Database para Linux, UNIX y Windows, Versión 9.5 y posterior.</p>
VARCHAR2_COMPAT	<p>Especifica si el origen de datos remoto es compatible con VARCHAR2. Los valores válidos son Y y N. El valor predeterminado varía en función del origen de datos remoto. Para los orígenes de datos Oracle remotos, el valor predeterminado es Y; el origen de datos es compatible con VARCHAR2. Para el resto de orígenes de datos remotos, el valor predeterminado es N; el origen de datos no está establecido en modalidad compatible VARCHAR2.</p> <p>Debe establecer esta opción de servidor a Y si su origen de datos DB2 Database para Linux, UNIX y Windows, ODBC o JDBC está configurado en modalidad compatible con VARCHAR2.</p>

Opciones de correlación de usuarios

Tabla 54. Opciones de de correlación de usuarios para orígenes de datos DB2

Opción	Description
FED_PROXY_USER	<p>Especifica el ID de autorización que hay que utilizar para establecer todas las conexiones de salida fiables cuando la conexión de entrada no es fiable. El usuario cuyo ID se especifica en esta opción debe tener una correlación de usuarios que especifique las opciones REMOTE_AUTHID y REMOTE_PASSWORD. Si especifica la opción de correlación de usuarios FED_PROXY_USER, también deberá especificar la opción de servidor FED_PROXY_USER.</p> <p>Restricción: Esta opción de servidor sólo es válida para DB2 Database para Linux, UNIX y Windows, Versión 9.5 y posterior, y DB2 for z/OS Versión 9 y posterior.</p>
ACCOUNTING_STRING	<p>Necesario en caso que haya que pasar información contable. Especifica una serie contable DRDA. Los valores válidos incluyen cualquier serie que tenga 255 o menos caracteres.</p>
REMOTE_AUTHID	<p>Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.</p>
REMOTE_PASSWORD	<p>Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.</p>
USE_TRUSTED_CONTEXT	<p>Especifica si la correlación de usuarios es fiable. Los valores válidos son Y y N. El valor predeterminado es N (la correlación de usuarios no es fiable y sólo puede utilizarse en conexiones federadas de salida que no son fiables). Y especifica que la correlación de usuarios es fiable y que puede utilizarse en conexiones federadas de salida fiables y no fiables.</p> <p>Restricción: Esta opción de servidor sólo es válida para DB2 Database para Linux, UNIX y Windows, Versión 9.5 y posterior, y DB2 for z/OS Versión 9 y posterior.</p>

Opciones de columna

Tabla 55. Opciones de columna para orígenes de datos DB2

Opción	Descripción
NUMERIC_STRING	Especifica cómo tratar las series numéricas. El valor predeterminado es N. Si la columna de tipo serie del origen de datos sólo contiene series numéricas y ningún otro tipo de caracteres, incluidos los blancos, establezca la opción NUMERIC_STRING en Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.
NO_EMPTY_STRING	Especifica si el servidor de orígenes de datos remotos contiene series vacías. Los valores válidos son Y y N. El valor predeterminado varía en función del origen de datos remoto. Para orígenes de datos Oracle remotos, el valor predeterminado es Y; todos los valores de serie vacíos se convierten a valores NULL. Para el resto de orígenes de datos remotos, el valor predeterminado es N; el origen de datos puede contener series vacías.
XML_ROOT	Especifica el elemento raíz XML que se debe añadir a los valores de una columna XML que hace referencia a una secuencia XML. Esta opción asegura que los valores de la columna XML son un documento XML con el formato correcto.

Información de consulta sobre opciones de Excel

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor y de apodo.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 56. Opciones de derivador para Excel

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).

Opciones de servidor

Tabla 57. Opciones de servidor para Excel

Nombre	Descripción
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.

Opciones de apodo

Tabla 58. Opciones de apodo para Excel

Opción	Descripción
FILE_PATH	Obligatoria. Especifica la vía de acceso totalmente calificada del directorio y el nombre de archivo de la hoja de cálculo Excel a la que desea acceder.
RANGE	Especifica el rango de celdas que se va a utilizar; por ejemplo, A1:C100. El valor que precede a los dos puntos especifica la celda superior izquierda del rango. El valor que sigue a los dos puntos especifica la celda inferior derecha del rango.

Información de consulta sobre opciones de Informix

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 59. Opciones de derivador para Informix

Nombre	Descripción
DB2_FENCED	Necesario. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Opciones de servidor

Tabla 60. Opciones de servidor para Informix

Nombre	Descripción
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N y I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.
COMM_RATE	Especifica la velocidad de comunicación, en megabytes por segundo, entre el servidor federado y el servidor de orígenes de datos. Los valores válidos son los números enteros mayores que 0 y menores que 2147483648. El valor predeterminado es 2.

Tabla 60. Opciones de servidor para Informix (continuación)

Nombre	Descripción
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.
DBNAME	Necesario. Especifica el nombre de la base de datos Informix a la que desea acceder.
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos. Si más de un plan de acceso cumple estos criterios, se elegirá el que tenga el coste menor.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
DB2_TWO_PHASE_COMMIT	Especifica si el servidor federado se conecta con el origen de datos mediante el protocolo de confirmación en dos fases o mediante el protocolo de confirmación en una fase. Los valores válidos son Y y N. El valor predeterminado es N (el servidor federado utiliza el protocolo de confirmación en una fase para conectarse). Y especifica que el servidor federado utiliza el protocolo de confirmación en dos fases para conectarse.

Tabla 60. Opciones de servidor para Informix (continuación)

Nombre	Descripción
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
FOLD_ID	Especifica si el ID de usuario se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía el ID de usuario en mayúsculas y, si éste falla, el servidor lo envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
FOLD_PW	Especifica si la contraseña se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía la contraseña en mayúsculas y, si ésta falla, el servidor la envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
INFORMIX_CLIENT_LOCALE	Especifica qué CLIENT_LOCALE utilizar para la conexión entre el servidor federado y el servidor de orígenes de datos. El valor es cualquier entorno local Informix válido. Si no especifica esta opción, la variable de entorno CLIENT_LOCALE se establece en el valor especificado en el archivo db2dj.ini. Si el archivo db2dj.ini no especifica la variable de entorno CLIENT_LOCALE, el INFORMIX_CLIENT_LOCALE se establece en el entorno local de Informix más parecido a la página de códigos y al territorio de la base de datos federada.

Tabla 60. Opciones de servidor para Informix (continuación)

Nombre	Descripción
INFORMIX_DB_LOCALE	Especifica qué DB_LOCALE de Informix utilizar para la conexión entre el servidor federado y el servidor de orígenes de datos. Si no se especifica la opción INFORMIX_DB_LOCALE, la variable de entorno local DB_LOCALE de Informix se establece en el valor especificado en el archivo db2dj.ini. Si el archivo db2dj.ini no especifica un valor, no se establece la variable de entorno DB_LOCALE de Informix.
INFORMIX_LOCK_MODE	Especifica qué modalidad de bloqueo establecer para un origen de datos de Informix. El derivador de Informix emite el mandato SET LOCK MODE inmediatamente después de conectarse a un origen de datos de Informix. Los valores válidos son W, N y un número. El valor predeterminado es W; el derivador espera de forma indefinida a que se libere el bloqueo. N especifica no esperar; se devuelve un error inmediatamente. Utilice un número para especificar el tiempo máximo, en segundos, que debe esperar. Si se produce un tiempo muerto o se excede el tiempo de espera, utilice la sentencia ALTER SERVER para cambiar el valor de la opción INFORMIX_LOCK_MODE. Por ejemplo: <pre>ALTER SERVER TYPE informix VERSION 9 WRAPPER informix OPTIONS (ADD informix_lock_mode '60')</pre>
IO_RATIO	Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.

Tabla 60. Opciones de servidor para Informix (continuación)

Nombre	Descripción
IUD_APP_SVPT_ENFORCE	Especifica si el servidor federado debe aplicar el uso de sentencias de punto de recuperación de aplicación. Los valores válidos son Y y N. El valor predeterminado es Y; en caso que el origen de datos no aplique sentencias de punto de recuperación de aplicación y que ocurra un error durante la operación de inserción, actualización o supresión, el servidor federado deshace la transacción y se devuelve el código de error SQL1476N. Es recomendable que utilice el valor predeterminado.
NODE	Necesario. Especifica el nombre por el que el origen de datos está definido como una instancia a su sistema de gestión de bases de datos relacionales.
OLD_NAME_GEN	Especifica cómo convertir los nombres de columna y los nombre de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.
PUSHDOWN	Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el origen de datos evalúa operaciones). N especifica que el servidor federado envía sentencias de SQL que incluyen sólo SELECT con nombres de columna. En las sentencias de SQL que el servidor federado envía al origen de datos no se incluyen predicados, como WHERE=, funciones de columna y escalares, como MAX y MIN, clasificaciones, como ORDER BY o GROUP BY ni uniones.

Opciones de correlación de usuarios

Tabla 61. Opciones de correlación de usuarios para Informix

Nombre	Descripción
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.

Tabla 61. Opciones de correlación de usuarios para Informix (continuación)

Nombre	Descripción
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.

Opciones de columna

Tabla 62. Opciones de columna para Informix

Nombre	Descripción
NUMERIC_STRING	Especifica cómo tratar las series numéricas. El valor predeterminado es N. Si la columna de tipo serie del origen de datos sólo contiene series numéricas y ningún otro tipo de caracteres, incluidos los blancos, establezca la opción NUMERIC_STRING en Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.

Información de consulta sobre opciones de JDBC

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 63. Opciones de derivador para JDBC

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. El único valor válido es Y porque el servidor DB2 solamente permite cargar la JVM en modalidad delimitada. El valor predeterminado es Y (el derivador se ejecuta en modalidad delimitada).

Tabla 63. Opciones de derivador para JDBC (continuación)

Nombre	Descripción
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Opciones de servidor

Tabla 64. Opciones de servidor para JDBC

Nombre	Descripción
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N e I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.
COMM_RATE	Especifica la velocidad de comunicación, en megabytes por segundo, entre el servidor federado y el servidor de orígenes de datos. Los valores válidos son los números enteros mayores que 0 y menores que 2147483648. El valor predeterminado es 2.
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.

Tabla 64. Opciones de servidor para JDBC (continuación)

Nombre	Descripción
DATEFORMAT	Especifica el formato de fecha que utiliza el origen de datos. Utilice 'DD', 'MM' y 'YY' o 'YYYY' para especificar el formato de fecha. Puede especificar un delimitador, como por ejemplo un espacio, un guión o una coma. Por ejemplo, el formato 'YYYY-MM-DD' especifica una fecha del tipo 1958-10-01. El valor puede contener valores nulos.
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 0. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Tabla 64. Opciones de servidor para JDBC (continuación)

Nombre	Descripción
DRIVER_CLASS	<p>Especifica la biblioteca de controladores JDBC. El servidor se puede registrar en varios orígenes de datos JDBC si el controlador JDBC se ajusta a la especificación JDBC versión 3.0 y posteriores. Consulte la documentación del controlador JDBC para obtener más información sobre las especificaciones JDBC y sobre cómo establecer la opción de servidor DRIVER_CLASS.</p> <p>Ejemplo</p> <p>En este ejemplo, se especifica la biblioteca de controladores JDBC com.ibm.db2.jcc.DB2Driver:</p> <pre>DRIVER_CLASS 'com.ibm.db2.jcc.DB2Driver'</pre> <p>Importante: Si especifica esta opción, también deberá especificar la opción de servidor URL.</p>
DRIVER_PACKAGE	<p>Especifica los paquetes de controladores JDBC. Utilice un separador de vías de acceso para especificar varios paquetes de clases de controlador. Utilice un punto y coma en los sistemas operativos Windows y dos puntos en los sistemas operativos Linux y Unix.</p> <p>Ejemplo</p> <p>En este ejemplo, se especifican varios paquetes de controladores separados por dos puntos en sistemas operativos Linux:</p> <pre>DRIVER_PACKAGE '/via1/archivo1.jar: /via2/archivo2.jar'</pre>
FOLD_ID	<p>Especifica si el ID de usuario se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía el ID de usuario en mayúsculas y, si éste falla, el servidor lo envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.</p>
FOLD_PW	<p>Especifica si la contraseña se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía la contraseña en mayúsculas y, si ésta falla, el servidor la envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.</p>

Tabla 64. Opciones de servidor para JDBC (continuación)

Nombre	Descripción
IO_RATIO	<p>Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23}. El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.</p>
JDBC_LOG	<p>Especifica si el derivador JDBC crea archivos de registro para el rastreo de errores. Los valores válidos son Y y N. El valor predeterminado es N (el archivo de registro no se crea). Si esta opción de servidor se establece en Y, el derivador JDBC graba archivos de registro JDBC en el archivo <code>jdbc_wrapper_id_prod.log</code>, donde <code>id_prod</code> es el ID de producto. El archivo de registro se almacena en el directorio especificado por el parámetro de configuración del gestor de bases de datos DB2 DIAGPATH. El directorio predeterminado en sistemas UNIX es <code>inst_home/sql/lib/db2dump</code>.</p> <p>Recomendación: Establecer esta opción de servidor en YES (SÍ) afectará al rendimiento del sistema y no debe habilitar el registro de anotaciones en sistemas de producción.</p>
OLD_NAME_GEN	<p>Especifica cómo convertir los nombres de columna y los nombre de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.</p>

Tabla 64. Opciones de servidor para JDBC (continuación)

Nombre	Descripción
PUSHDOWN	Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el origen de datos evalúa operaciones). N especifica que el servidor federado envía sentencias de SQL que incluyen sólo SELECT con nombres de columna. En las sentencias de SQL que el servidor federado envía al origen de datos no se incluyen predicados, como WHERE=, funciones de columna y escalares, como MAX y MIN, clasificaciones, como ORDER BY o GROUP BY ni uniones.
TIMEFORMAT	Especifica el formato de hora que utiliza el origen de datos. Utilice 'hh12', 'hh24', 'mm', 'ss', 'AM' y 'A.M.' para especificar el formato de hora. Por ejemplo, el formato 'hh24:mm:22' especifica una hora del tipo 16:00:00. El formato 'hh12:mm:ss AM' especifica una hora del tipo 8:00:00 AM. El valor puede contener valores nulos.
TIMESTAMPFORMAT	Especifica el formato de indicación de fecha y hora que utiliza el origen de datos. Los valores válidos son los que utilizan las opciones DATEFORMAT y TIMEFORMAT. Especifique 'n' para las décimas de segundo, 'nn' para las centésimas de segundo, 'nnn' para los milisegundos y así hasta 'nnnnnn' para los microsegundos. Por ejemplo, el formato 'YYY-MM-DD-hh24:mm:ss.nnnnnn' especifica una indicación de fecha y hora del tipo 1994-01-01-24:00:00.000000. El valor puede contener valores nulos.

Tabla 64. Opciones de servidor para JDBC (continuación)

Nombre	Descripción
URL	<p>Especifica la serie de conexión JDBC del servidor remoto.</p> <p>La serie de conexión JDBC consta de tres partes separadas por dos puntos:</p> <ul style="list-style-type: none"> • Protocolo de la base de datos • Nombre del tipo de la base de datos o nombre del controlador de conectividad • Identidad de la base de datos mediante un alias o un subnombre <p>Ejemplo</p> <p>En este ejemplo, la serie de conexión JDBC es</p> <pre>jdbc:db2://cn.ibm.com:50471/testdb:</pre> <p>URL</p> <pre>'jdbc:db2://cn.ibm.com:50471/testdb'</pre> <p>El servidor se puede registrar en varios orígenes de datos JDBC si el controlador JDBC se ajusta a la especificación JDBC versión 3.0 y posteriores. Consulte la documentación del controlador JDBC para obtener más información sobre las especificaciones JDBC y sobre cómo establecer la opción de servidor URL.</p> <p>Importante: Si especifica esta opción, también deberá especificar la opción de servidor DRIVER_CLASS.</p>
VARCHAR_NO_TRAILING_BLANKS	<p>Especifica si el origen de datos contiene columnas VARCHAR que contienen como mínimo un carácter en blanco de cola. El valor predeterminado es N (las columnas VARCHAR contienen como mínimo un carácter en blanco de cola).</p>

Opciones de correlación de usuarios

Tabla 65. Opciones de correlación de usuarios para JDBC

Nombre	Descripción
REMOTE_AUTHID	<p>Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.</p>
REMOTE_PASSWORD	<p>Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.</p>

Opciones de columna

Tabla 66. Opciones de columna para JDBC

Nombre	Descripción
NUMERIC_STRING	Especifica si la columna contiene series de caracteres numéricos que incluyen blancos. Los valores válidos son Y y N. El valor predeterminado es N (la columna no contiene series numéricas que incluyen blancos). Si la columna contiene solamente series numéricas seguidas por blancos de cola, no especifique Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.
VARCHAR_NO_TRAILING_BLANKS	Especifica si hay como mínimo un blanco de cola en la columna VARCHAR.

Información de consulta sobre opciones de Microsoft SQL Server

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 67. Opciones de derivador para Microsoft SQL Server

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).

Tabla 67. Opciones de derivador para Microsoft SQL Server (continuación)

Nombre	Descripción
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Opciones de servidor

Tabla 68. Opciones de servidor para Microsoft SQL Server

Nombre	Descripción
CODEPAGE	Especifica la página de códigos que corresponde al juego de caracteres codificado de la configuración de cliente del origen de datos. En sistemas UNIX y Microsoft Windows que utilizan una base de datos federada que no es Unicode, el valor predeterminado es la página de códigos de la base de datos federada. En sistemas UNIX que utilizan una base de datos federada Unicode, el valor predeterminado es 1208. En sistemas Windows que utilizan una base de datos federada Unicode, el valor predeterminado es 1202.
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N e I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.

Tabla 68. Opciones de servidor para Microsoft SQL Server (continuación)

Nombre	Descripción
COMM_RATE	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la velocidad de CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4.
DBNAME	Obligatoria. Especifica el alias de la base de datos a la que desea acceder. El valor es sensible a mayúsculas y minúsculas.
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos. Si más de un plan de acceso cumple estos criterios, se elegirá el que tenga el coste menor.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.

Tabla 68. Opciones de servidor para Microsoft SQL Server (continuación)

Nombre	Descripción
DB2_TWO_PHASE_COMMIT	<p>Especifica si el servidor federado se conecta con el origen de datos mediante el protocolo de confirmación en dos fases o mediante el protocolo de confirmación en una fase. Los valores válidos son Y y N. El valor predeterminado es N (el servidor federado utiliza el protocolo de confirmación en una fase para conectarse). Y especifica que el servidor federado utiliza el protocolo de confirmación en dos fases para conectarse.</p> <p>Importante: Si establece esta opción en Y, también deberá especificar la opción XA_OPEN_STRING_OPTION.</p>
DB2_UM_PLUGIN	<p>Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.</p>
DB2_UM_PLUGIN_LANG	<p>Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.</p>
FOLD_ID	<p>Especifica si el ID de usuario se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía el ID de usuario en mayúsculas y, si éste falla, el servidor lo envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.</p>
FOLD_PW	<p>Especifica si la contraseña se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía la contraseña en mayúsculas y, si ésta falla, el servidor la envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.</p>

Tabla 68. Opciones de servidor para Microsoft SQL Server (continuación)

Nombre	Descripción
IO_RATIO	Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.
NODE	Obligatoria. Si el servidor federado utiliza Microsoft Windows, el valor de NODE es el nombre DSN del sistema especificado para Microsoft SQL Server. Si el servidor federado utiliza UNIX o Linux, el valor de NODE se define en el archivo <code>odbc.ini</code> . El valor es sensible a mayúsculas y minúsculas.
OLD_NAME_GEN	Especifica cómo convertir los nombres de columna y los nombre de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.
PASSWORD	Especifica si hay que enviar contraseñas a un origen de datos o no. El valor predeterminado es Y.
PUSHDOWN	Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el origen de datos evalúa operaciones). N especifica que el servidor federado envía sentencias de SQL que incluyen sólo SELECT con nombres de columna. En las sentencias de SQL que el servidor federado envía al origen de datos no se incluyen predicados, como WHERE=, funciones de columna y escalares, como MAX y MIN, clasificaciones, como ORDER BY o GROUP BY ni uniones.

Tabla 68. Opciones de servidor para Microsoft SQL Server (continuación)

Nombre	Descripción
XA_OPEN_STRING_OPTIONS	Opción obligatoria si se establece DB2_TWO_PHASE_COMMIT en Y. Especifica el ID del gestor de recursos del Registro de Microsoft SQL Server.

Opciones de correlación de usuarios

Tabla 69. Opciones de correlación de usuarios para Microsoft SQL Server

Nombre	Descripción
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.

Opciones de columna

Tabla 70. Opciones de columna para Microsoft SQL Server

Nombre	Descripción
NUMERIC_STRING	Especifica cómo tratar las series numéricas. El valor predeterminado es N. Si la columna de tipo serie del origen de datos sólo contiene series numéricas y ningún otro tipo de caracteres, incluidos los blancos, establezca la opción NUMERIC_STRING en Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.

Información de consulta sobre opciones de ODBC

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 71. Opciones de derivador para ODBC

Nombre	Descripción
DB2_FENCED	<p>Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).</p> <p>Importante: Si establece esta opción en Y para un sistema UNIX, también deberá establecer la opción de derivador DB2_SOURCE_CLIENT_MODE.</p>
DB2_SOURCE_CLIENT_MODE	<p>Especifica que el cliente del origen de datos es de 32 bits y que la instancia de base de datos del servidor federado es de 64 bits. El único valor válido es 32 bits. Esta opción sólo es válida para UNIX.</p> <p>Importante: Si establece esta opción, también deberá establecer la opción de derivador DB2_FENCED en Y.</p>
DB2_UM_PLUGIN	<p>Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.</p>
DB2_UM_PLUGIN_LANG	<p>Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.</p>
MODULE	<p>Obligatoria para servidores federados que se ejecutan en un sistema UNIX. Especifica la vía de acceso completa de la biblioteca que contiene la implementación del gestor de controladores ODBC o la implementación de SQL/CLI. No hay valor predeterminado para UNIX. En un sistema Microsoft Windows, el valor predeterminado es odbc32.dll.</p>

Opciones de servidor

Tabla 72. Opciones de servidor para ODBC

Nombre	Descripción
CODEPAGE	Especifica la página de códigos que corresponde al juego de caracteres codificado de la configuración de cliente del origen de datos. En sistemas UNIX y Windows que utilizan una base de datos federada que no es Unicode, el valor predeterminado es la página de códigos que utiliza la base de datos federada. En sistemas UNIX que utilizan una base de datos federada Unicode, el valor predeterminado es 1208. En sistemas Windows que utilizan una base de datos federada Unicode, el valor predeterminado es 1202.
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N e I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.
COMM_RATE	Especifica la velocidad de comunicación, en megabytes por segundo, entre el servidor federado y el servidor de orígenes de datos. Los valores válidos son los números enteros mayores que 0 y menores que 2147483648. El valor predeterminado es 2.
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.
DATEFORMAT	Especifica el formato de fecha que utiliza el origen de datos. Utilice 'DD', 'MM' y 'YY' o 'YYYY' para especificar el formato de fecha. Puede especificar un delimitador, como por ejemplo un espacio, un guión o una coma. Por ejemplo, el formato 'YYYY-MM-DD' especifica una fecha como 1958-10-01. El valor puede contener valores nulos.

Tabla 72. Opciones de servidor para ODBC (continuación)

Nombre	Descripción
DBNAME	Especifica el nombre de la base de datos de origen de datos a la que desea acceder.
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 0. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
FOLD_ID	Especifica si el ID de usuario se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía el ID de usuario en mayúsculas y, si éste falla, el servidor lo envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
FOLD_PW	Especifica si la contraseña se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía la contraseña en mayúsculas y, si ésta falla, el servidor la envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.

Tabla 72. Opciones de servidor para ODBC (continuación)

Nombre	Descripción
IO_RATIO	Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.
NODE	Obligatoria. Especifica el nombre del nodo o el nombre DSN del sistema que se asigna al origen de datos ODBC cuando se define el DSN. El valor es sensible a mayúsculas y minúsculas.
OLD_NAME_GEN	Especifica cómo convertir los nombres de columna y los nombres de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.
PUSHDOWN	Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el origen de datos evalúa operaciones). N especifica que el servidor federado envía sentencias de SQL que incluyen sólo SELECT con nombres de columna. En las sentencias de SQL que el servidor federado envía al origen de datos no se incluyen predicados, como WHERE=, funciones de columna y escalares, como MAX y MIN, clasificaciones, como ORDER BY o GROUP BY ni uniones.

Tabla 72. Opciones de servidor para ODBC (continuación)

Nombre	Descripción
TIMEFORMAT	Especifica el formato de hora que utiliza el origen de datos. Utilice 'hh12', 'hh24', 'mm', 'ss', 'AM' y 'A.M.' para especificar el formato de hora. Por ejemplo, el formato 'hh24:mm:22' especifica una hora del tipo 16:00:00. El formato 'hh12:mm:ss AM' especifica una hora del tipo 8:00:00 AM. El valor puede contener valores nulos.
TIMESTAMPFORMAT	Especifica el formato de indicación de fecha y hora que utiliza el origen de datos. Los valores válidos son los que utilizan las opciones DATEFORMAT y TIMEFORMAT. Especifique 'n' para las décimas de segundo, 'nn' para las centésimas de segundo, 'nnn' para los milisegundos y así hasta 'nnnnnn' para los microsegundos. Por ejemplo, el formato 'YYY-MM-DD-hh24:mm:ss.nnnnnn' especifica una indicación de fecha y hora del tipo 1994-01-01-24:00:00.000000. El valor puede contener valores nulos.
VARCHAR_NO_TRAILING_BLANKS	Especifica si el origen de datos contiene columnas VARCHAR que contienen como mínimo un carácter en blanco de cola. El valor predeterminado es N (las columnas VARCHAR contienen como mínimo un carácter en blanco de cola).

Opciones de correlación de usuarios

Tabla 73. Opciones de correlación de usuarios para ODBC

Nombre	Descripción
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.

Opciones de columna

Tabla 74. Opciones de columna para ODBC

Nombre	Descripción
NUMERIC_STRING	Especifica si la columna contiene series de caracteres numéricos que incluyen blancos. Los valores válidos son Y y N. El valor predeterminado es N (la columna no contiene series numéricas que incluyen blancos). Si la columna contiene solamente series numéricas seguidas por blancos de cola, no especifique Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.
VARCHAR_NO_TRAILING_BLANKS	Especifica si hay como mínimo un blanco de cola en la columna VARCHAR.

Información de consulta sobre opciones de Oracle

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 75. Opciones de derivador para Oracle

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).

Tabla 75. Opciones de derivador para Oracle (continuación)

Nombre	Descripción
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Opciones de servidor

Tabla 76. Opciones de servidor para Oracle

Nombre	Descripción
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N e I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.
COMM_RATE	Especifica la velocidad de comunicación, en megabytes por segundo, entre el servidor federado y el servidor de orígenes de datos. Los valores válidos son los números enteros mayores que 0 y menores que 2147483648. El valor predeterminado es 2.
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.

Tabla 76. Opciones de servidor para Oracle (continuación)

Nombre	Descripción
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
DB2_TWO_PHASE_COMMIT	Especifica si el servidor federado se conecta con el origen de datos mediante el protocolo de confirmación en dos fases o mediante el protocolo de confirmación en una fase. Los valores válidos son Y y N. El valor predeterminado es N (el servidor federado utiliza el protocolo de confirmación en una fase para conectarse). Y especifica que el servidor federado utiliza el protocolo de confirmación en dos fases para conectarse.
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
FED_PROXY_USER	Especifica el ID de autorización que hay que utilizar para establecer todas las conexiones de salida fiables cuando la conexión de entrada no es fiable. Importante: El usuario cuyo ID se especifica en esta opción debe tener una correlación de usuarios que especifique las opciones REMOTE_AUTHID y REMOTE_PASSWORD.

Tabla 76. Opciones de servidor para Oracle (continuación)

Nombre	Descripción
FOLD_ID	Especifica si el ID de usuario se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía el ID de usuario en mayúsculas y, si éste falla, el servidor lo envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
FOLD_PW	Especifica si la contraseña se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía la contraseña en mayúsculas y, si ésta falla, el servidor la envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
IO_RATIO	Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.
NODE	Obligatoria. Especifica el nombre del nodo en que reside el servidor de bases de datos Oracle. Obtenga el nombre del nodo del archivo tnsnames.ora.
OLD_NAME_GEN	Especifica cómo convertir los nombres de columna y los nombre de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.

Tabla 76. Opciones de servidor para Oracle (continuación)

Nombre	Descripción
PLAN_HINTS	Especifica si se van a habilitar las sugerencias de plan o no. Las sugerencias de plan son fragmentos de sentencias que proporcionan información adicional que el optimizador de orígenes de datos utiliza para mejorar el rendimiento de la consulta. El optimizador de orígenes de datos utiliza las sugerencias de plan para decidir si utilizar un índice o no y qué índice o qué secuencia de unión de tabla utilizar. Los valores válidos son Y y N. El valor predeterminado es N (las sugerencias de plan no se habilitan). Y especifica que las sugerencias de plan se van a habilitar en el origen de datos si éste admite las sugerencias de plan.
PUSHDOWN	Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el servidor federado permite que el origen de datos evalúe operaciones). N especifica que el servidor federado recupera columnas del origen de datos remoto.
VARCHAR_NO_TRAILING_BLANKS	Especifica para un servidor determinado si las columnas VARCHAR contienen blancos de cola. Para aplicar la opción a una columna individual, utiliza la opción de columna VARCHAR_NO_TRAILING_BLANKS.
XA_OPEN_STRING_OPTIONS	Especifica información adicional que hay que añadir a la serie. Por ejemplo, la información podría ser el directorio de los archivos de rastreo.

Opciones de correlación de usuarios

Tabla 77. Opciones de correlación de usuarios para Oracle

Nombre	Descripción
FED_PROXY_USER	Especifica el ID de autorización que hay que utilizar para establecer todas las conexiones fiables de salida cuando la conexión de entrada no es fiable. Importante: El usuario cuyo ID se especifica en esta opción debe tener una correlación de usuarios que especifique las opciones REMOTE_AUTHID y REMOTE_PASSWORD. Si especifica la opción de correlación de usuarios FED_PROXY_USER, también deberá especificar la opción de servidor FED_PROXY_USER.

Tabla 77. Opciones de correlación de usuarios para Oracle (continuación)

Nombre	Descripción
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.
USE_TRUSTED_CONTEXT	Especifica si la correlación de usuarios es fiable. Los valores válidos son Y y N. El valor predeterminado es N (la correlación de usuarios no es fiable y sólo puede utilizarse en conexiones de salida que no son fiables). Y especifica que la correlación de usuarios es fiable y que puede utilizarse en conexiones de salida fiables y no fiables.

Opciones de columna

Tabla 78. Opciones de columna para Oracle

Nombre	Descripción
NUMERIC_STRING	Especifica si la columna contiene series de caracteres numéricos que incluyen blancos. Los valores válidos son Y y N. El valor predeterminado es N (la columna no contiene series numéricas que incluyen blancos). Si la columna contiene solamente series numéricas seguidas por blancos de cola, no especifique Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.
VARCHAR_NO_TRAILING_BLANKS	Especifica si la columna contiene blancos de cola.

Información de consulta sobre opciones de Script

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios, de apodo y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 79. Opciones de derivador para scripts

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE y SOCKS. El valor predeterminado es NONE.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_SERVER_PORT	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.

Opciones de servidor

Tabla 80. Opciones de servidor para scripts

Nombre	Descripción
DAEMON_PORT	Especifica el número de puerto en que el daemon Script está a la escucha de solicitudes de trabajos de scripts. El valor predeterminado es 4099. El número de puerto debe ser el mismo que el de la opción DAEMON_PORT en el archivo de configuración del daemon. Si se configura un nombre de servicio para el daemon Script, utilice un nombre de servicio TCP/IP.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.

Tabla 80. Opciones de servidor para scripts (continuación)

Nombre	Descripción
NODE	Obligatoria. Especifica el nombre de host DNS o la dirección IP del sistema en que se ejecuta el daemon Script. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_SERVER_PORT	Especifica el puerto o el nombre de servicio del servicio proxy en el servidor proxy. Los valores válidos son un número de puerto decimal comprendido entre 1 y 32760 o un nombre de servicio.
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE y SOCKS. El valor predeterminado es NONE.

Opciones de correlación de usuarios

Tabla 81. Opciones de correlación de usuarios para scripts

Nombre	Descripción
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy. La contraseña se cifra al almacenarla en el catálogo de la base de datos federada.

Opciones de apodo

Tabla 82. Opciones de apodo para scripts

Nombre	Descripción
DATASOURCE	Opción obligatoria para el apodo raíz. Especifica el nombre del script que hay que llamar. El script que especifique como el valor de esta opción también debe especificarse en el archivo de configuración del daemon Script. Importante: Esta opción sólo es válida para el apodo raíz.
NAMESPACES	Especifica los espacios de nombres asociados con los prefijos de espacio de nombres que se utilizan en las opciones XPATH y TEMPLATE de cada columna. Utilice esta sintaxis: <pre>NAMESPACES'prefijo1="espacio_de_nombres_real1", prefijo2="espacio_de_nombres_real2"'</pre> Utilice comas para separar varios espacios de nombres. Por ejemplo: <pre>NAMESPACES='http://www.miweb.com/cliie", i='http://www.miweb.com/cliie/id", n='http://www.miweb.com/cliie/nombre"'</pre>
STREAMING	Especifica si el documento fuente debe dividirse en fragmentos lógicos para procesarlo. Los fragmentos corresponden al nodo que coincide con la expresión XPath del apodo. A continuación, el derivador analiza y procesa los datos del documento fuente fragmento a fragmento. Este tipo de análisis minimiza la utilización de memoria. Los valores válidos son Y y N. El valor predeterminado es N (los documentos no se analizan). Esta opción sólo es válida en el apodo raíz. No establezca las opciones STREAMING y VALIDATE en Y a la vez.
TIMEOUT	Especifica el tiempo máximo, en minutos, que se esperará algún resultado del daemon. El valor predeterminado es 60. Esta opción sólo es válida para el apodo raíz.
VALIDATE	Especifica si el documento fuente se valida para asegurarse de que se ajusta a un esquema XML o a una definición de tipo de documento (DTD) antes de que los datos se extraigan de él. El valor predeterminado es N (la validación no se realiza). Antes de establecer el valor en Y, el archivo de esquema o el archivo DTD ha de estar en la ubicación especificada por el documento fuente. Esta opción sólo es válida para el apodo raíz. No establezca las opciones STREAMING y VALIDATE en Y a la vez.

Tabla 82. Opciones de apodo para scripts (continuación)

Nombre	Descripción
XPATH	Especifica la expresión XPath que identifica los elementos XML que representan tuplas individuales. La opción de apodo XPATH para un apodo hijo se evalúa en el contexto de la vía de acceso especificada por la opción de apodo XPATH de su padre. La expresión XPath se utiliza como contexto para evaluar valores de columnas identificadas por la presencia de la opción de columna XPATH.

Opciones de columna

Tabla 83. Opciones de columna para scripts

Nombre	Descripción
DEFAULT	Especifica un valor predeterminado para una columna de entrada de script. Si la consulta SQL no proporciona un valor, se utilizará este valor predeterminado.
FOREIGN_KEY	Indica que este apodo es un apodo hijo y especifica el nombre del apodo padre correspondiente. Un apodo puede tener como mucho una opción de columna FOREIGN_KEY. El valor de esta opción es sensible a mayúsculas y minúsculas. No especifique la opción XPATH para esta columna. La columna sólo puede utilizarse para unir un apodo padre y un apodo hijo. Una sentencia CREATE NICKNAME que incluya la opción FOREIGN_KEY fallará si el apodo padre tiene un nombre de esquema diferente. A menos que el apodo al que se hace referencia en la cláusula FOREIGN_KEY se hubiera definido en la sentencia CREATE NICKNAME explícitamente en minúsculas o en una combinación de mayúsculas y minúsculas, deberá especificar el apodo en mayúsculas cuando haga referencia a éste en la cláusula FOREIGN_KEY. Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna.
INPUT_MODE	Especifica la modalidad de entrada de la columna. Los valores válidos son CONFIG y FILE_INPUT. CONFIG trata el valor como la modalidad de entrada de una columna. FILE_INPUT especifica un archivo que almacena el valor. El derivador pasa el valor especificado al daemon Script.

Tabla 83. Opciones de columna para scripts (continuación)

Nombre	Descripción
POSITION	Especifica un valor entero para parámetros posicionales. Si el valor posicional se establece en un entero, esta entrada deberá estar en esta posición en la línea de mandatos. Si se establece esta opción, el conmutador se inserta en la ubicación adecuado cuando se ejecuta la consulta. Si se establece POSITION en -1, la opción se añade como la última opción de la línea de mandatos. Un valor entero POSITION no puede utilizarse dos veces en el mismo apodo. Esta opción sólo se aplica a columnas de entrada.
PRIMARY_KEY	Opción obligatoria para un apodo padre que tiene uno o varios apodos hijo. Especifica que este apodo es un apodo padre. El tipo de datos de columna debe ser VARCHAR(16). Un apodo puede tener solamente una opción de columna PRIMARY_KEY. El único valor válido es Yes (Sí). No especifique la opción XPATH para esta columna. La columna sólo puede utilizarse para unir apodos padre y apodos hijo. Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna.
SWITCH	Especifica un distintivo para el script en la línea de mandatos. El valor de esta opción precedía el valor de columna suministrado por WSSCRIPT.ARGS o por el valor predeterminado, si existe. Si no especifica un valor para esta opción y existe un valor predeterminado para la columna, éste se añade sin información de conmutador. Esta opción es obligatoria para las columnas de entrada.
SWITCH_ONLY	Permite utilizar conmutadores sin un argumento de línea de mandatos. Los valores válidos son Y y N. Si establece esta opción en Y y los valores de entrada válidos serán Y y N. Si el valor de entrada es Y, sólo se añade el conmutador a la línea de mandatos. Si el valor de entrada es N, no se añade ningún valor a la línea de mandatos.
VALID_VALUES	Especifica un conjunto de valores válidos para una columna. Utilice un punto y coma para separar varios valores.
XPATH	Especifica la expresión XPath en el documento XML que contiene los datos correspondientes a esta columna. El derivador evalúa esta expresión XPath después de que la sentencia CREATE NICKNAME aplique la expresión XPath de la opción de apodo XPATH.

Información de consulta sobre opciones de Sybase

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 84. Opciones de derivador para Sybase

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. En Microsoft Windows, los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable). En UNIX, el valor predeterminado y único valor válido es Y (el derivador debe ejecutarse en modalidad delimitada).
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Opciones de servidor

Tabla 85. Opciones de servidor para Sybase

Nombre	Descripción
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N e I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.

Tabla 85. Opciones de servidor para Sybase (continuación)

Nombre	Descripción
COMM_RATE	Especifica la velocidad de comunicación, en megabytes por segundo, entre el servidor federado y el servidor de orígenes de datos. Los valores válidos son los números enteros mayores que 0 y menores que 2147483648. El valor predeterminado es 2.
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.
CONV_EMPTY_STRING	Especifica si el servidor federado convierte una serie vacía en un espacio durante las tareas de duplicación. Los valores válidos son Y y N. El valor predeterminado es N (el servidor federado no convierte series vacías). Establezca esta opción en Y si el origen de datos tiene una columna de caracteres que no admite nulos que almacena una serie vacía.
DBNAME	Obligatoria. Especifica el nombre de la base de datos a la que desea acceder. Obtenga el nombre de la base de datos del servidor Sybase.
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos.

Tabla 85. Opciones de servidor para Sybase (continuación)

Nombre	Descripción
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
DB2_TWO_PHASE_COMMIT	Especifica si el servidor federado se conecta con el origen de datos mediante el protocolo de confirmación en dos fases o mediante el protocolo de confirmación en una fase. Los valores válidos son Y y N. El valor predeterminado es N (el servidor federado utiliza el protocolo de confirmación en una fase para conectarse). Y especifica que el servidor federado utiliza el protocolo de confirmación en dos fases para conectarse.
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
FOLD_ID	Especifica si el ID de usuario se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía el ID de usuario en mayúsculas y, si éste falla, el servidor lo envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
FOLD_PW	Especifica si la contraseña se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía la contraseña en mayúsculas y, si ésta falla, el servidor la envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.

Tabla 85. Opciones de servidor para Sybase (continuación)

Nombre	Descripción
IFILE	Especifica la vía de acceso y el nombre del archivo de interfaz de Sybase que se va a utilizar en lugar del archivo de interfaz predeterminado. El derivador Sybase busca el archivo de interfaz en los lugares siguientes, en el orden especificado: en Microsoft Windows, en la opción de servidor IFILE, después en el directorio %DB2PATH%\interfaces y, finalmente, en el directorio %SYBASE%\ini\sql.ini. En UNIX, en la opción de servidor IFILE, a continuación en el directorio sqllib/interfaces y, finalmente, en el directorio \$SYBASE/interfaces.
IO_RATIO	Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.
LOGIN_TIMEOUT	Especifica el tiempo, en segundos, que el servidor federado esperará antes de abandonar una solicitud de inicio de sesión. El valor predeterminado es 0 (el servidor federado espera de forma indefinida).
NODE	Obligatoria. Especifica el nombre del nodo en que reside el servidor Sybase. El nombre de nodo se encuentra en el archivo de interfaces de Sybase.
OLD_NAME_GEN	Especifica cómo convertir los nombres de columna y los nombre de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.

Tabla 85. Opciones de servidor para Sybase (continuación)

Nombre	Descripción
PACKET_SIZE	Especifica el tamaño de paquete, en bytes, que utiliza la biblioteca de cliente para enviar paquetes de secuencia de datos tabular (TDS). Si el derivador Sybase necesita enviar o recibir grandes cantidades de datos de tipo texto e imagen, aumente el valor de PACKET_SIZE.
PLAN_HINTS	Especifica si se van a habilitar las sugerencias de plan o no. Las sugerencias de plan son fragmentos de sentencias que proporcionan información adicional que el optimizador de orígenes de datos utiliza para mejorar el rendimiento de la consulta. El optimizador de orígenes de datos utiliza las sugerencias de plan para decidir si utilizar un índice o no y qué índice o qué secuencia de unión de tabla utilizar. Los valores válidos son Y y N. El valor predeterminado es N (las sugerencias de plan no se habilitan). Y especifica que las sugerencias de plan se van a habilitar en el origen de datos si éste admite las sugerencias de plan.
PUSHDOWN	Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el origen de datos evalúa operaciones). N especifica que el servidor federado envía sentencias de SQL que incluyen sólo SELECT con nombres de columna. En las sentencias de SQL que el servidor federado envía al origen de datos no se incluyen predicados, como WHERE=, funciones de columna y escalares, como MAX y MIN, clasificaciones, como ORDER BY o GROUP BY ni uniones.
TIMEOUT	Especifica el tiempo máximo, en segundos, que el servidor federado esperará una respuesta del servidor remoto a un mandato. El valor predeterminado es 0, que especifica que el tiempo de espera es indefinido.
XA_OPEN_STRING_OPTIONS	Especifica series abiertas para la interfaz Sybase DTM XA. Estas series se añaden al nombre del gestor de recursos lógicos, el nombre de usuario y la contraseña.

Opciones de correlación de usuarios

Tabla 86. Opciones de correlación de usuarios para Sybase

Opción	Descripción
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.

Opciones de columna

Tabla 87. Opciones de columna para Sybase

Opción	Descripción
NUMERIC_STRING	Especifica cómo tratar las series numéricas. El valor predeterminado es N. Si la columna de tipo serie del origen de datos sólo contiene series numéricas y ningún otro tipo de caracteres, incluidos los blancos, establezca la opción NUMERIC_STRING en Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.

Información de consulta sobre opciones de Teradata

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 88. Opciones de derivador para Teradata

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. En Microsoft Windows, los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable). En UNIX, el valor predeterminado es Y (el valor derivador debe ejecutarse en modalidad delimitada).
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Opciones de servidor

Tabla 89. Opciones de servidor para Teradata

Nombre	Descripción
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N e I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.
COMM_RATE	Especifica la velocidad de comunicación, en megabytes por segundo, entre el servidor federado y el servidor de orígenes de datos. Los valores válidos son los números enteros mayores que 0 y menores que 2147483648. El valor predeterminado es 2.

Tabla 89. Opciones de servidor para Teradata (continuación)

Nombre	Descripción
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Tabla 89. Opciones de servidor para Teradata (continuación)

Nombre	Descripción
IO_RATIO	Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.
NODE	Obligatoria. Especifica el nombre de alias o la dirección IP del servidor Teradata.
OLD_NAME_GEN	Especifica cómo convertir los nombres de columna y los nombre de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.
PUSHDOWN	Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el origen de datos evalúa operaciones). N especifica que el servidor federado envía sentencias de SQL que incluyen sólo SELECT con nombres de columna. En las sentencias de SQL que el servidor federado envía al origen de datos no se incluyen predicados, como WHERE=, funciones de columna y escalares, como MAX y MIN, clasificaciones, como ORDER BY o GROUP BY ni uniones.

Opciones de correlación de usuarios

Tabla 90. Opciones de correlación de usuarios para Teradata

Nombre	Descripción
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.

Opciones de columna

Tabla 91. Opciones de columna para Teradata

Nombre	Descripción
NUMERIC_STRING	Especifica cómo tratar las series numéricas. El valor predeterminado es N. Si la columna de tipo serie del origen de datos sólo contiene series numéricas y ningún otro tipo de caracteres, incluidos los blancos, establezca la opción NUMERIC_STRING en Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.

Información de consulta sobre opciones de archivos con estructura de tabla

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de apodo y de columna.

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Opciones de derivador

Tabla 92. Opciones de derivador para archivos con estructura de tabla

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).

Opciones de servidor

Tabla 93. Opciones de servidor para archivos con estructura de tabla

Nombre	Descripción
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.

Opciones de apodo

Tabla 94. Opciones de apodo para archivos con estructura de tabla

Nombre	Descripción
COLUMN_DELIMITER	Especifica un solo carácter que se utilizará como delimitador que separa las columnas del archivo con estructura de tabla. El valor predeterminado es la coma (.). Los apóstrofes no pueden utilizarse como delimitadores. El delimitador de columna debe ser el mismo en todo el archivo. Un valor nulo se representa mediante dos delimitadores juntos o por un delimitador seguido de un terminador de línea, si el campo NULL es el último de la línea. El delimitador de columna no puede existir como datos válidos de una columna.
CODEPAGE	Especifica la página de códigos del archivo del origen de datos. Esta opción sólo es válida para bases de datos federadas que utilizan Unicode. Los datos fuente se convierten a partir de la página de códigos especificada en Unicode.

Tabla 94. Opciones de apodo para archivos con estructura de tabla (continuación)

Nombre	Descripción
FILE_PATH	<p>Especifica la vía de acceso totalmente calificada del archivo con estructura de tabla. Escriba el nombre de archivo entre apóstrofos. El archivo de datos debe ser un archivo estándar o un enlace simbólico, en vez de un conducto u otro tipo de archivo que no sea estándar.</p> <p>Importante: Si especifica la opción FILE_PATH, no especifique una columna DOCUMENT.</p>
KEY_COLUMN	<p>Especifica el nombre de la columna que se utiliza para ordenar el archivo. Una columna que tenga la opción de columna DOCUMENT no puede utilizar la columna de clave. Sólo se admiten las claves de una sola columna. El valor debe ser el nombre de una columna definida en la sentencia CREATE NICKNAME. La columna debe ordenarse de forma ascendente. Debe especificarse que la columna de clave no admite nulos añadiendo la opción NOT NULL a su definición en la sentencia de apodo. Este valor es sensible a mayúsculas y minúsculas.</p>
SORTED	<p>Especifica si el archivo del origen de datos está ordenado de forma ascendente o no. Los valores válidos son Y y N. El valor predeterminado es N (el archivo del origen de datos no está ordenado de forma ascendente). Los orígenes de datos ordenados deben ordenarse de forma ascendente según la secuencia de clasificación del entorno local actual, tal como definen los valores de la categoría de soporte multilingüístico LC_COLLATE. Si especifica que el origen de datos está ordenado, establezca la opción VALIDATE_DATA_FILE en Y.</p>
VALIDATE_DATA_FILE	<p>En el caso de archivos ordenados, esta opción especifica si el derivador ha de verificar que la columna de clave está ordenada de forma ascendente y si ha de comprobar la existencia de claves nulas. Esta validación se realiza sólo una vez la primera vez que se crea el apodo. El valor predeterminado es N (no se verifica el orden de clasificación). Esta opción sólo es válida si la opción SORTED se establece en Y y si no se especifica la opción DOCUMENT.</p>

Opciones de columna

Tabla 95. Opciones de columna para archivos con estructura de tabla

Opción	Descripción
DOCUMENT	Permite especificar la vía de acceso del archivo cuando se ejecuta la consulta y no cuando se crea el apodo. El único valor válido es FILE. Con la opción DOCUMENT sólo puede especificarse una columna de cada apodo. El tipo de datos de la columna asociada con la opción DOCUMENT debe ser VARCHAR o CHAR. La utilización de la opción de columna de apodo DOCUMENT en lugar de la opción de apodo FILE_PATH implica que el archivo que corresponde a este apodo se suministra cuando se ejecuta la consulta. Si la opción DOCUMENT tiene el valor FILE, el valor que se suministra cuando se ejecuta la consulta es la vía de acceso completa del archivo cuyo esquema coincide con la definición de este apodo.

Información de consulta sobre opciones de servicios web

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios, de apodo y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 96. Opciones de derivador para servicios web

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Tabla 96. Opciones de derivador para servicios web (continuación)

Nombre	Descripción
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE, HTTP y SOCKS. El valor predeterminado es NONE.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_SERVER_PORT	Especifica el puerto o el nombre de servicio del servicio proxy en el servidor proxy. Los valores válidos son un número de puerto decimal comprendido entre 1 y 32760 o un nombre de servicio.
SSL_KEYSTORE_FILE	Especifica el archivo de almacenamiento de certificados para las comunicaciones que utilizan SSL o TSL. Un valor válido es un nombre de vía de acceso totalmente calificada al que pueda acceder el agente de la base de datos federada o un proceso de modalidad delimitada. El valor predeterminado es <i>vía de acceso de instalación/cfg/WSWrapperKeystore.kdb</i> .
SSL_KEYSTORE_PASSWORD	Especifica la contraseña que hay que utilizar para acceder al archivo de la opción SSL_KEYSTORE_FILE. Los valores válidos son una contraseña, que se cifra al almacenarla en el catálogo de la base de datos federada, y <i>file:nombre_archivo</i> , donde <i>nombre_archivo</i> es la vía de acceso totalmente calificada de un archivo stash.
SSL_VERIFY_SERVER_CERTIFICATE	Especifica si el certificado del servidor se verifica durante la autenticación SSL. Los valores válidos son Y y N. El valor predeterminado es N (el certificado no se verifica).

Opciones de servidor

Tabla 97. Opciones de servidor para servicios web

Nombre	Descripción
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.

Tabla 97. Opciones de servidor para servicios web (continuación)

Nombre	Descripción
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_SERVER_PORT	Especifica el puerto o el nombre de servicio del servicio proxy en el servidor proxy. Los valores válidos son un número de puerto decimal comprendido entre 1 y 32760 o un nombre de servicio.
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE, HTTP y SOCKS. El valor predeterminado es NONE.
SSL_CLIENT_CERTIFICATE_LABEL	Especifica el certificado del cliente que se ha de enviar durante la autenticación SSL. Si no especifica un valor, se utilizará el ID de autorización actual de la base de datos federada para encontrar el certificado.
SSL_KEYSTORE_FILE	Especifica el archivo de almacenamiento de certificados para las comunicaciones que utilizan SSL o TSL. Un valor válido es un nombre de vía de acceso totalmente calificada al que pueda acceder el agente de la base de datos federada o un proceso de modalidad delimitada. El valor predeterminado es <i>vía de acceso de instalación/cfg/WSWrapperKeystore.kdb</i> .

Tabla 97. Opciones de servidor para servicios web (continuación)

Nombre	Descripción
SSL_KEYSTORE_PASSWORD	Especifica la contraseña que hay que utilizar para acceder al archivo de la opción SSL_KEYSTORE_FILE. Los valores válidos son una contraseña, que se cifra al almacenarla en el catálogo de la base de datos federada, y <code>file:nombre_archivo</code> , donde <code>nombre_archivo</code> es la vía de acceso totalmente calificada de un archivo stash.
SSL_VERIFY_SERVER_CERTIFICATE	Especifica si hay que verificar el certificado del servidor durante la autenticación SSL. Los valores válidos son Y y N. El valor predeterminado es N (el certificado no se verifica).

Opciones de correlación de usuarios

Tabla 98. Opciones de correlación de usuarios para servicios web

Nombre	Descripción
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy. La contraseña se cifra al almacenarla en el catálogo de la base de datos federada.
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.
SSL_CLIENT_CERTIFICATE_LABEL	Especifica el certificado del cliente que se ha de enviar durante la autenticación SSL. Si no especifica un valor, se utilizará el ID de autorización actual de la base de datos federada para encontrar el certificado.

Opciones de apodo

Tabla 99. Opciones de apodo para servicios web

Nombre	Descripción
NAMESPACES	<p>Especifica los espacios de nombres asociados con los prefijos de espacio de nombres que se utilizan en las opciones XPATH y TEMPLATE de cada columna. Utilice esta sintaxis:</p> <pre>NAMESPACES'prefijo1="espacio_de_nombres_real1", prefijo2="espacio_de_nombres_real2"'</pre> <p>Utilice comas para separar varios espacios de nombres. Por ejemplo:</p> <pre>NAMESPACES='http://www.miweb.com/ctie', i='http://www.miweb.com/ctie/id', n='http://www.miweb.com/ctie/nombre''</pre>
SOAPACTION	<p>Opción obligatoria para el apodo raíz. Especifica el atributo URI SOAPACTION del formato del lenguaje de descripción de servicios web (WSDL). El URL puede contener una dirección IPv6 separada por dos puntos si se escribe entre corchetes. Por ejemplo:http:// [1080:0:0:0:8:800:200C:417A]</p> <p>Nota: Esta opción no es válida para apodos distintos del apodo raíz.</p>
STREAMING	<p>Especifica si el documento fuente debe dividirse en fragmentos lógicos para procesarlo. Los fragmentos corresponden al nodo que coincide con la expresión XPath del apodo. A continuación, el derivador analiza y procesa los datos del documento fuente fragmento a fragmento. Este tipo de análisis minimiza la utilización de memoria. Los valores válidos son Y y N. El valor predeterminado es N (los documentos no se analizan). Esta opción sólo es válida en el apodo raíz.</p>
TEMPLATE	<p>Especifica el fragmento de plantilla de apodo que se va a utilizar para crear una solicitud SOAP. El fragmento debe ajustarse a la sintaxis especificada para la plantilla. Esta opción sólo es válida para el apodo raíz.</p>
URL	<p>Opción obligatoria para el apodo raíz. Especifica el URL del punto final de servicio web. Los protocolos admitidos son HTTP y HTTPS. El URL puede contener una dirección IPv6 separada por dos puntos si se escribe entre corchetes. Por ejemplo:http:// [1080:0:0:0:8:800:200C:417A]</p>

Tabla 99. Opciones de apodo para servicios web (continuación)

Nombre	Descripción
XML_CODESET	Especifica la codificación que hay que utilizar para enviar y recibir datos XML. Esta opción altera temporalmente la codificación interna.
XPATH	Obligatoria. Especifica la expresión XPath que identifica los elementos de respuesta SOAP que representan tuplas individuales. La expresión XPath se utiliza como contexto para evaluar valores de columnas que identifican las opciones de columna de apodo XPATH.

Opciones de columna

Tabla 100. Opciones de columna para servicios web

Nombre	Descripción
ESCAPE_INPUT	Especifica si hay que sustituir caracteres XML especiales en los valores XML de entrada o no. Utilice esta opción para incluir fragmentos XML como entrada, por ejemplo, para incluir fragmentos XML que tienen elementos repetidos. Los valores válidos son Y y N. N es el valor predeterminado (los valores XML de entrada se mantienen). El tipo de datos de columna debe ser VARCHAR o CHAR. Si ESCAPE_INPUT se establece en Y, también deberá especificar la opción de columna TEMPLATE.

Tabla 100. Opciones de columna para servicios web (continuación)

Nombre	Descripción
FOREIGN_KEY	<p>Indica que este apodo es un apodo hijo y especifica el nombre del apodo padre correspondiente. Un apodo puede tener como mucho una opción de columna FOREIGN_KEY. El valor de esta opción es sensible a mayúsculas y minúsculas. No especifique la opción XPATH para esta columna. La columna sólo puede utilizarse para unir un apodo padre y un apodo hijo. Una sentencia CREATE NICKNAME que incluya la opción FOREIGN_KEY fallará si el apodo padre tiene un nombre de esquema diferente. A menos que el apodo al que se hace referencia en la cláusula FOREIGN_KEY se hubiera definido en la sentencia CREATE NICKNAME explícitamente en minúsculas o en una combinación de mayúsculas y minúsculas, deberá especificar el apodo en mayúsculas cuando haga referencia a éste en la cláusula FOREIGN_KEY.</p> <p>Nota:</p> <ul style="list-style-type: none"> • Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna. • Si se establece esta opción de columna, luego no podrá utilizar la sentencia ALTER NICKNAME para descartar la opción. Lo que deberá hacer es descartar el apodo y, a continuación, crearlo de nuevo sin esta opción de columna.
PRIMARY_KEY	<p>Opción obligatoria para un apodo padre que tiene uno o varios apodos hijo. Especifica que este apodo es un apodo padre. El tipo de datos de columna debe ser VARCHAR(16). Un apodo puede tener solamente una opción de columna PRIMARY_KEY. El único valor válido es Yes (Sí). No especifique la opción XPATH para esta columna. La columna sólo puede utilizarse para unir apodos padre y apodos hijo.</p> <p>Nota:</p> <ul style="list-style-type: none"> • Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna. • Si se establece esta opción de columna, luego no podrá utilizar la sentencia ALTER NICKNAME para descartar la opción. Lo que deberá hacer es descartar el apodo y, a continuación, crearlo de nuevo sin esta opción de columna.

Tabla 100. Opciones de columna para servicios web (continuación)

Nombre	Descripción
SOAPACTIONCOLUMN	<p>Especifica la columna que especifica dinámicamente la acción SOAP para el punto final de servicios web cuando se ejecuta una consulta. Esta opción sólo es válida para el apodo raíz. Si el nombre de host es una dirección IPv6 (separada por dos puntos), escriba el nombre de host entre corchetes. Por ejemplo: 'http://[1080:0:0:0:8:800:200C:417A]:99/soap'. Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna.</p>
TEMPLATE	<p>Especifica el fragmento de plantilla de columna que se va a utilizar para crear el documento XML de entrada. El fragmento debe ajustarse a la sintaxis especificada para la plantilla.</p> <p>Nota: Si se establece esta opción de columna, luego no podrá utilizar la sentencia ALTER NICKNAME para descartar la opción. Lo que deberá hacer es descartar el apodo y, a continuación, crearlo de nuevo sin esta opción de columna.</p>
URLCOLUMN	<p>Especifica la columna que especifica dinámicamente la acción SOAP para el punto final de servicios web cuando se ejecuta una consulta. Esta opción sólo es válida para el apodo raíz. Si el nombre de host es una dirección IPv6 (separada por dos puntos), escriba el nombre de host entre corchetes. Por ejemplo: 'http://[1080:0:0:0:8:800:200C:417A]:99/soap'. Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna.</p>
XPATH	<p>Especifica la expresión XPath en el documento XML que contiene los datos correspondientes a esta columna. El derivador evalúa esta expresión XPath después de que la sentencia CREATE NICKNAME aplique la expresión XPath de la opción de apodo XPATH.</p> <p>Nota: Si se establece esta opción de columna, luego no podrá utilizar la sentencia ALTER NICKNAME para descartar la opción. Lo que deberá hacer es descartar el apodo y, a continuación, crearlo de nuevo sin esta opción de columna.</p>

Información de consulta sobre opciones de XML

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios, de apodo y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 101. Opciones de derivador para XML

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. N es el valor predeterminado (el derivador se ejecuta en modalidad fiable).
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE, HTTP y SOCKS. El valor predeterminado es NONE.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_SERVER_PORT	Especifica el puerto o el nombre de servicio del servicio proxy en el servidor proxy. Los valores válidos son un número de puerto decimal comprendido entre 1 y 32760 o un nombre de servicio.
SSL_KEYSTORE_FILE	Especifica el archivo de almacenamiento de certificados para las comunicaciones que utilizan SSL o TSL. Un valor válido es un nombre de vía de acceso totalmente calificada al que pueda acceder el agente de la base de datos federada o un proceso de modalidad delimitada.
SSL_KEYSTORE_PASSWORD	Especifica la contraseña que hay que utilizar para acceder al archivo de la opción SSL_KEYSTORE_FILE. Los valores válidos son una contraseña, que se cifra al almacenarla en el catálogo de la base de datos federada, y <code>file:nombre_archivo</code> , donde <code>nombre_archivo</code> es la vía de acceso totalmente calificada de un archivo stash.
SSL_VERIFY_SERVER_CERTIFICATE	Especifica si el certificado del servidor se verifica durante la autenticación SSL. Los valores válidos son Y y N. El valor predeterminado es N (el certificado no se verifica).

Opciones de servidor

Tabla 102. Opciones de servidor para XML

Nombre	Descripción
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_SERVER_PORT	Especifica el puerto o el nombre de servicio del servicio proxy en el servidor proxy. Los valores válidos son un número de puerto decimal comprendido entre 1 y 32760 o un nombre de servicio.
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE, HTTP y SOCKS. El valor predeterminado es NONE.
SOCKET_TIMEOUT	Especifica el tiempo máximo, en minutos, que el servidor federado esperará algún resultado del servidor proxy. Un valor válido es cualquier número que sea mayor o igual que 0. El valor predeterminado es 0 (el servidor esperará de forma indefinida).
SSL_CLIENT_CERTIFICATE_LABEL	Especifica el certificado del cliente que se ha de enviar durante la autenticación SSL. Si no especifica un valor, se utilizará el ID de autorización actual de la base de datos federada para encontrar el certificado.
SSL_KEYSTORE_FILE	Especifica el archivo de almacenamiento de certificados para las comunicaciones que utilizan SSL o TSL. Un valor válido es un nombre de vía de acceso totalmente calificada al que pueda acceder el agente de la base de datos federada o un proceso de modalidad delimitada.

Tabla 102. Opciones de servidor para XML (continuación)

Nombre	Descripción
SSL_KEYSTORE_PASSWORD	Especifica la contraseña que hay que utilizar para acceder al archivo de la opción SSL_KEYSTORE_FILE. Los valores válidos son una contraseña, que se cifra al almacenarla en el catálogo de la base de datos federada, y <code>file:nombre_archivo</code> , donde <code>nombre_archivo</code> es la vía de acceso totalmente calificada de un archivo stash.
SSL_VERIFY_SERVER_CERTIFICATE	Especifica si hay que verificar el certificado del servidor durante la autenticación SSL. El valor predeterminado es N (el certificado no se verifica).

Opciones de correlación de usuarios

Tabla 103. Opciones de correlación de usuarios para XML

Nombre	Descripción
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy.
SSL_CLIENT_CERTIFICATE_LABEL	Especifica el certificado del cliente que se ha de enviar durante la autenticación SSL. Si no especifica un valor, se utilizará el ID de autorización actual de la base de datos federada para encontrar el certificado.

Opciones de apodo

Tabla 104. Opciones de apodo para XML

Nombre	Descripción
DIRECTORY_PATH	Especifica el nombre de la vía de acceso de un directorio que contiene uno o varios archivos XML. Utilice esta opción para crear un único apodo para varios archivos de origen XML. El derivador XML utiliza solamente los archivos con extensión <code>.xml</code> que se encuentran en el directorio especificado. El derivador XML no tiene en cuenta los otros archivos del directorio. Si especifica esta opción de apodo, no especifique una columna DOCUMENT. Esta opción sólo es válida para el apodo raíz.
FILE_PATH	Especifica la vía de acceso del archivo del documento XML. Si especifica FILE_PATH, no especifique una columna DOCUMENT. Esta opción sólo es válida para el apodo raíz.

Tabla 104. Opciones de apodo para XML (continuación)

Nombre	Descripción
INSTANCE_PARSE_TIME	<p>Especifica el tiempo, en milisegundos, necesario para analizar una fila del documento fuente XML. El valor válido puede ser un entero o un valor decimal. El valor predeterminado es 7. Esta opción sólo es válida para columnas del apodo raíz. Para optimizar consultas de estructuras fuente XML complejas o de gran tamaño, modifique las opciones INSTANCE_PARSE_TIME, XPATH_EVAL_TIME y NEXT_TIME.</p>
NAMESPACES	<p>Especifica los espacios de nombres asociados con los prefijos de espacio de nombres que se utilizan en las opciones XPATH y TEMPLATE de cada columna. Utilice esta sintaxis:</p> <pre data-bbox="963 743 1450 827">NAMESPACES'prefijo1="espacio_de_nombres_real1", prefijo2="espacio_de_nombres_real2"</pre> <p>Utilice comas para separar varios espacios de nombres. Por ejemplo:</p> <pre data-bbox="963 926 1450 999">NAMESPACES='http://www.miweb.com/clie", i='http://www.miweb.com/clie/id", n='http://www.miweb.com/clie/nombre"</pre>
NEXT_TIME	<p>Especifica el tiempo, en milisegundos, necesario para encontrar elementos fuente posteriores de la expresión XPath. El valor predeterminado es 1. Esta opción es válida para apodos raíz y distintos del apodo raíz. Para optimizar consultas de estructuras fuente XML complejas o de gran tamaño, modifique las opciones INSTANCE_PARSE_TIME, XPATH_EVAL_TIME y NEXT_TIME.</p>
STREAMING	<p>Especifica si el documento fuente debe dividirse en fragmentos lógicos para procesarlo. Los fragmentos corresponden al nodo que coincide con la expresión XPath del apodo. A continuación, el derivador analiza y procesa los datos del documento fuente fragmento a fragmento. Este tipo de análisis minimiza la utilización de memoria. Los valores válidos son Y y N. El valor predeterminado es N (los documentos no se analizan). Esta opción sólo es válida en el apodo raíz.</p>

Tabla 104. Opciones de apodo para XML (continuación)

Nombre	Descripción
VALIDATE	Especifica si el documento fuente se valida para asegurarse de que se ajusta a un esquema XML o a una definición de tipo de documento (DTD) antes de que los datos se extraigan de él. El valor predeterminado es N (la validación no se realiza). Antes de establecer el valor en Y, el archivo de esquema o el archivo DTD ha de estar en la ubicación especificada por el documento fuente. Esta opción sólo es válida para el apodo raíz. No establezca las opciones STREAMING y VALIDATE en Y a la vez.
XPATH	Obligatoria. Especifica la expresión XPath que identifica los elementos que representan las tuplas individual. La expresión XPath se utiliza como contexto para evaluar valores de columnas identificadas por la opción de columna XPATH.
XPATH_EVAL_TIME	Especifica el tiempo necesario, en milisegundos, para evaluar la expresión XPath del apodo y para encontrar el primer elemento. El valor puede ser un entero o un valor decimal. El valor predeterminado es 1. Esta opción es válida para apodos raíz y distintos del apodo raíz. Para optimizar consultas de estructuras fuente XML complejas o de gran tamaño, modifique las opciones INSTANCE_PARSE_TIME, XPATH_EVAL_TIME y NEXT_TIME.

Opciones de columna

Tabla 105. Opciones de columna para XML

Nombre	Descripción
DOCUMENT	Especifica que esta columna es una columna DOCUMENT. El valor de la columna DOCUMENT indica el tipo de datos fuente XML suministrado al apodo cuando se ejecuta la consulta. Esta opción sólo es válida para columnas del apodo raíz (el apodo que identifica los elementos en el nivel superior del documento XML). Con la opción DOCUMENT sólo puede especificarse una columna para cada apodo. Si utiliza una opción de columna DOCUMENT en vez de una opción de apodo FILE_PATH o DIRECTORY_PATH, el documento que corresponde a este apodo se suministra cuando se ejecuta la consulta. Los valores válidos son FILE, DIRECTORY, URI y COLUMN. El valor predeterminado es FILE. El tipo de datos de la columna debe ser VARCHAR.

Tabla 105. Opciones de columna para XML (continuación)

Nombre	Descripción
FOREIGN_KEY	<p>Indica que este apodo es un apodo hijo y especifica el nombre del apodo padre correspondiente. Un apodo puede tener como mucho una opción de columna FOREIGN_KEY. El valor de esta opción es sensible a mayúsculas y minúsculas. No especifique la opción XPATH para esta columna. La columna sólo puede utilizarse para unir un apodo padre y un apodo hijo. Una sentencia CREATE NICKNAME que incluya la opción FOREIGN_KEY fallará si el apodo padre tiene un nombre de esquema diferente. A menos que el apodo al que se hace referencia en la cláusula FOREIGN_KEY se hubiera definido en la sentencia CREATE NICKNAME explícitamente en minúsculas o en una combinación de mayúsculas y minúsculas, deberá especificar el apodo en mayúsculas cuando haga referencia a éste en la cláusula FOREIGN_KEY.</p> <p>Nota:</p> <ul style="list-style-type: none"> • Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna. • Si se establece esta opción de columna, luego no podrá utilizar la sentencia ALTER NICKNAME para descartar la opción. Lo que deberá hacer es descartar el apodo y, a continuación, crearlo de nuevo sin esta opción de columna.
PRIMARY_KEY	<p>Opción obligatoria para un apodo padre que tiene uno o varios apodos hijo. Especifica que este apodo es un apodo padre. El tipo de datos de columna debe ser VARCHAR(16). Un apodo puede tener solamente una opción de columna PRIMARY_KEY. El único valor válido es Yes (Sí). No especifique la opción XPATH para esta columna. La columna sólo puede utilizarse para unir apodos padre y apodos hijo.</p> <p>Nota:</p> <ul style="list-style-type: none"> • Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna. • Si se establece esta opción de columna, luego no podrá utilizar la sentencia ALTER NICKNAME para descartar la opción. Lo que deberá hacer es descartar el apodo y, a continuación, crearlo de nuevo sin esta opción de columna.

Tabla 105. Opciones de columna para XML (continuación)

Nombre	Descripción
XPATH	<p>Especifica la expresión XPath en el documento XML que contiene los datos correspondientes a esta columna. El derivador evalúa esta expresión XPath después de que la sentencia CREATE NICKNAME aplique la expresión XPath de la opción de apodo XPATH.</p> <p>Nota: Si se establece esta opción de columna, luego no podrá utilizar la sentencia ALTER NICKNAME para descartar la opción. Lo que deberá hacer es descartar el apodo y, a continuación, crearlo de nuevo sin esta opción de columna.</p>

Capítulo 5. Vistas de la tabla de catálogo global que contienen información federada

La mayoría de vistas de catálogo de una base de datos federada son iguales que las vistas de catálogo de cualquier otra base de datos DB2 para Linux, UNIX y Windows.

Hay algunas vistas exclusivas que contienen información relacionada con un sistema federado, por ejemplo, la vista SYSCAT.WRAPPERS.

Las vistas SYSCAT son de sólo lectura. No puede emitir una operación de actualización o inserción en una vista del esquema SYSCAT. Las vistas SYSSTAT son la forma recomendada de actualizar el catálogo de sistema. Cambie las aplicaciones que hacen referencia a la vista SYSCAT para hacer referencia a la vista SYSSTAT actualizable.

La tabla siguiente lista las vistas SYSCAT que contienen información federada. Estas vistas son de sólo lectura.

Tabla 106. Vistas de catálogo que suele utilizarse con un sistema federado

Vistas de catálogo	Descripción
SYSCAT.CHECKS	Contiene la información de restricciones de comprobación definida por el usuario.
SYSCAT.COLCHECKS	Contiene las columnas referenciadas por una restricción de comprobación.
SYSCAT.COLUMNS	Contiene información de columna sobre los objetos de orígenes de datos (tablas y vistas) para los que ha creado apodos.
SYSCAT.COLOPTIONS	Contiene la información sobre los valores de opciones de columna que ha establecido para un apodo.
SYSCAT.CONSTDEP	Contiene la dependencia de una restricción informativa definida por el usuario.
SYSCAT.DATATYPES	Contiene información de tipo de datos sobre tipos de datos DB2 locales incorporados y definidos por el usuario.
SYSCAT.DBAUTH	Contiene las autoridades de bases de datos mantenidas por grupos y usuarios individuales.
SYSCAT.FUNCMAPOPTIONS	Contiene información sobre los valores de opciones que ha establecido para una correlación de funciones.
SYSCAT.FUNCMAPPINGS	Contiene las correlaciones de funciones entre la base de datos federada y los objetos de origen de datos.
SYSCAT.INDEXCOLUSE	Contiene las columnas que forman parte de un índice.
SYSCAT.INDEXES	Contienen las especificaciones de índice para los objetos de origen de datos.

Tabla 106. Vistas de catálogo que suele utilizarse con un sistema federado (continuación)

Vistas de catálogo	Descripción
SYSCAT.INDEXOPTIONS	Contiene la información sobre las opciones de índice.
SYSCAT.KEYCOLUSE	Contiene las columnas que forman parte de una clave definida por una restricción de clave exclusiva, de clave primaria o de clave foránea.
SYSCAT.NICKNAMES	Contiene la información sobre los apodos que ha creado.
SYSCAT.REFERENCES	Contiene la información sobre las restricciones referenciales que ha definido.
SYSCAT.ROUTINES	Contiene funciones DB2 locales definidas por el usuario, o plantillas de función. Las plantillas de función se utilizan para correlacionar a una función de origen de datos.
SYSCAT.REVTYPEMAPPINGS	Esta vista no se utiliza. Todas las correlaciones de tipos de datos se registran en la vista SYSCAT.TYPEMAPPINGS.
SYSCAT.ROUTINEOPTIONS	Contiene la información sobre los valores de opciones de rutinas federadas.
SYSCAT.ROUTINEPARMOPTIONS	Contiene la información sobre los valores de opciones de parámetros de rutinas federadas.
SYSCAT.ROUTINEPARMS	Contiene un parámetro o el resultado de una rutina definida en SYSCAT.ROUTINES.
SYSCAT.ROUTINESFEDERATED	Contiene la información sobre las rutinas federadas que ha definido.
SYSCAT.SERVERS	Contiene definiciones de servidor creadas para los servidores de orígenes de datos.
SYSCAT.TABCONST	Cada fila representa una tabla y restricciones de apodo del tipo CHECK, UNIQUE, PRIMARY KEY o FOREIGN KEY.
SYSCAT.TABLES	Contiene información sobre cada apodo, vista federada y tabla DB2 local que cree.
SYSCAT.TYPEMAPPINGS	Contiene correlaciones de tipos de datos directas y correlaciones de tipos de datos inversas. La correlación es a tipos de datos DB2 locales de tipos de datos de orígenes de datos. Estas correlaciones se utilizan cuando crea un apodo en un objeto de orígenes de datos.
SYSCAT.USEROPTIONS	Contiene la información de autorización de usuario establecida al crear correlaciones de usuarios entre la base de datos federada y los servidores de orígenes de datos.
SYSCAT.VIEWS	Contiene la información sobre las vistas federadas locales que ha creado.
SYSCAT.WRAPOPTIONS	Contiene la información sobre los valores de opciones que ha establecido para un derivador.

Tabla 106. Vistas de catálogo que suele utilizarse con un sistema federado (continuación)

Vistas de catálogo	Descripción
SYSCAT.WRAPPERS	Contiene el nombre del derivador y el archivo de biblioteca para cada origen de datos para el que cree un derivador.

La tabla siguiente lista las vistas SYSSTAT que contienen información federada. Estas son vistas de sólo lectura que contienen estadísticas que puede actualizar.

Tabla 107. Vistas de catálogo global federadas y actualizables

Vistas de catálogo	Descripción
SYSSTAT.COLUMNS	Contiene información estadística sobre cada columna de los objetos de orígenes de datos (tablas y vistas) para los que ha creado apodos. Las estadísticas no se graban para las columnas heredadas de las tablas con tipos.
SYSSTAT.INDEXES	Contienen información estadística sobre cada especificación de índice para los objetos de origen de datos.
SYSSTAT.ROUTINES	Contiene información estadística sobre cada función definida por el usuario. No incluye funciones incorporadas. Las estadísticas no se graban para las columnas heredadas de las tablas con tipos.
SYSSTAT.TABLES	Contiene información sobre cada tabla base. la información de vista, sinónimo y alias no está incluida en esta vista. En las tablas con tipos solamente se incluye en la vista la tabla raíz de una jerarquía de tablas. Las estadísticas no se graban para las columnas heredadas de las tablas con tipos.

Capítulo 6. Opciones de correlaciones de funciones para sistemas federados

El servidor federado proporciona correlaciones predeterminadas entre las funciones de DB2 y las funciones de orígenes de datos. Para la mayor parte de orígenes de datos, las correlaciones de funciones predeterminadas se encuentran en los derivadores. Para utilizar una función de orígenes de datos que el servidor federado no reconoce o para cambiar la correlación predeterminada, cree una correlación de funciones.

Cuando crea una correlación de funciones, especifica el nombre de la función de orígenes de datos y debe habilitar la función correlacionada. Luego, cuando utiliza la función correlacionada, el optimizador de consultas compara el coste de ejecutar la función en el origen de datos con el coste de ejecutar la función en el servidor federado.

Tabla 108. Opciones para las correlaciones de funciones

Nombre	Descripción
DISABLE	Habilita o inhabilita una correlación de funciones predeterminada. Los valores válidos son Y y N. El valor predeterminado es N.
REMOTE_NAME	En nombre de la función de origen de datos. El valor predeterminado es el nombre local.

Capítulo 7. Tipos de servidor válidos en las sentencias de SQL

Los tipos de servidor indican el tipo de orígenes de datos que representa la definición de servidor.

Los tipos de servidor varían en función del proveedor, el objetivo y el sistema operativo. Los valores soportados dependen del origen de datos.

Para la mayoría de orígenes de datos, debe especificar un tipo de servidor válido en la sentencia CREATE SERVER.

Tabla 109. Orígenes de datos y tipos de servidores

Origen de datos	Tipo de servidor
BioRS	No es necesario un tipo de servidor en la sentencia CREATE SERVER.
Excel	No es necesario un tipo de servidor en la sentencia CREATE SERVER.
IBM DB2 Universal Database para Linux, UNIX y Windows	DB2/UDB
IBM DB2 Universal Database para System i y AS/400	DB2/ISERIES
IBM DB2 Universal Database para z/OS	DB2/ZOS
IBM DB2 para VM	DB2/VM
Informix	INFORMIX
JDBC	JDBC (necesario para orígenes de datos JDBC soportados por los controladores JDBC 3.0 y posterior.)
Microsoft SQL Server	MSSQLSERVER (necesario para los orígenes de datos soportados por el controlador de DataDirect Connect ODBC 4.2 (o posterior) o por el controlador de Microsoft SQL Server ODBC 3.0 (o posterior).)
ODBC	ODBC (necesario para orígenes de datos ODBC soportados por el controlador ODBC 3.x.)
OLE DB	No es necesario un tipo de servidor en la sentencia CREATE SERVER.
Oracle	ORACLE (necesario para los orígenes de datos Oracle soportados por el software cliente Oracle NET8.)
Sybase (CTLIB)	SYBASE
Archivos con estructura de tabla	No es necesario un tipo de servidor en la sentencia CREATE SERVER.
Teradata	TERADATA
Servicios web	No es necesario un tipo de servidor en la sentencia CREATE SERVER.

Tabla 109. Orígenes de datos y tipos de servidores (continuación)

Origen de datos	Tipo de servidor
XML	No es necesario un tipo de servidor en la sentencia CREATE SERVER.

Capítulo 8. Correlaciones de tipos de datos

Las correlaciones de tipos de datos para orígenes de datos relacionales incluyen correlaciones de tipos de datos directas, correlaciones de tipos de datos inversas y correlaciones de tipos de datos que son específicas de Unicode. Cada origen de datos no relacional es compatible con determinados tipos de datos.

Correlaciones de tipos de datos directas por omisión

Los dos tipos de correlaciones entre los tipos de datos de origen de datos y los tipos de datos de base de datos federada son las correlaciones de tipos de datos directas e inversas. Una correlación de tipos de datos directa es la correlación de un tipo de datos remoto con un tipo de datos local comparable.

Con la sentencia CREATE TYPE MAPPING puede alterar temporalmente la correlación de tipos de datos por omisión o crear una correlación de tipos de datos nueva.

Estas correlaciones son válidas con todas las versiones compatibles, a menos que se indique lo contrario.

Para todas las correlaciones de tipos de datos directas por omisión de un origen de datos con la base de datos federada, el esquema federado es SYSIBM.

En las tablas siguientes se muestran las correlaciones directas por omisión entre tipos de datos de base de datos federada y tipos de datos de origen de datos.

Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 Database para Linux, UNIX y Windows

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 Database para Linux, UNIX y Windows.

Tabla 110. Correlaciones de tipos de datos directas por omisión de DB2 Database para Linux, UNIX y Windows (no se muestran todas las columnas)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BIGINT	-	-	-	-	-	-	BIGINT	-	0	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	-	-	-	-	-	-	CHAR	-	0	N
CHAR	-	-	-	-	S	-	CHAR	-	0	S
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DECFLOAT ²	-	-	-	-	-	-	DECFLOAT	-	0	-
DOUBLE	-	-	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	-	-	-

Tabla 110. Correlaciones de tipos de datos directas por omisión de DB2 Database para Linux, UNIX y Windows (no se muestran todas las columnas) (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
LONGVAR	-	-	-	-	N	-	CLOB	-	-	-
LONGVAR	-	-	-	-	S	-	BLOB	-	-	-
LONGVARG	-	-	-	-	-	-	DBCLOB	-	-	-
REAL	-	-	-	-	-	-	REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP(p)	-		p	p	-	-	TIMESTAMP(p)	-	p	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	0	N
VARCHAR	-	-	-	-	S	-	VARCHAR	-	0	S
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	0	N

Nota:

1. El tipo federado es TIMESTAMP(0) si el parámetro de configuración date_compat se establece en ON.
2. La opción de servidor SAME_DECFLT_ROUNDING se establece en N por omisión y las operaciones no se enviarán al origen de datos remoto a menos que SAME_DECFLT_ROUNDING se establezca en Y. Hallará más información sobre la opción de servidor SAME_DECFLT_ROUNDING en Información de consulta sobre opciones de DB2 Database.

Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para System i

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para System i.

Tabla 111. Correlaciones de tipos de datos directas por omisión de DB2 para System i (no se muestran todas las columnas)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	255	32672	-	-	-	-	VARCHAR	-	0	N
CHAR	1	254	-	-	S	-	CHAR	-	0	S
CHAR	255	32672	-	-	S	-	VARCHAR	-	0	S
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
GRAPHIC	128	16336	-	-	-	-	VARGRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
NUMERIC	-	-	-	-	-	-	DECIMAL	-	-	-

Tabla 111. Correlaciones de tipos de datos directas por omisión de DB2 para System i (no se muestran todas las columnas) (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	-	6	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	S	-	VARCHAR	-	0	S
VARG	1	16336	-	-	-	-	VARGGRAPHIC	-	0	N
VARGGRAPHIC	1	16336	-	-	-	-	VARGGRAPHIC	-	0	N

Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para VM y VSE

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para VM y VSE

Tabla 112. Correlaciones de tipos de datos directas por omisión de DB2 Server para VM y VSE (no se muestran todas las columnas)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	1	254	-	-	S	-	CHAR	-	0	S
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	-	-	S	-	VARCHAR	-	0	S
VARGGRAPHIC	1	16336	-	-	-	-	VARGGRAPHIC	-	0	N
VARGGRAPH	1	16336	-	-	-	-	VARGGRAPHIC	-	0	N

Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para z/OS

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para z/OS

Tabla 113. Correlaciones de tipos de datos directas por omisión de DB2 para z/OS (no se muestran todas las columnas)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	255	32672	-	-	-	-	VARCHAR	-	0	N
CHAR	1	254	-	-	S	-	CHAR	-	0	S
CHAR	255	32672	-	-	S	-	VARCHAR	-	0	S
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	-	-	-	-	S	-	VARCHAR	40	-	S
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	-	6	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	S	-	VARCHAR	-	0	S
VARG	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

Correlaciones de tipos de datos directas por omisión para orígenes de datos Informix

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos Informix.

Tabla 114. Correlaciones de tipos de datos directas por omisión de Informix (no se muestran todas las columnas)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BLOB	-	-	-	-	-	-	BLOB	2147483647	-	-
BOOLEAN	-	-	-	-	-	-	CHARACTER	1	-	-
BYTE	-	-	-	-	-	-	BLOB	2147483647	-	-
CHAR	1	254	-	-	-	-	CHARACTER	-	-	-
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CLOB	-	-	-	-	-	-	CLOB	2147483647	-	-
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	-

Tabla 114. Correlaciones de tipos de datos directas por omisión de Informix (no se muestran todas las columnas) (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
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	-	-	-

Notas:

1. El tipo federado es `TIMESTAMP(0)` si el parámetro de configuración `date_compat` se establece en `ON`.
2. Para el tipo de datos `DATETIME` de Informix, el servidor federado DB2 UNIX y Windows utiliza el calificador de alto nivel de Informix como `REMOTE_LENGTH` (longitud remota) y el calificador de bajo nivel de Informix como `REMOTE_SCALE` (escala remota).

Los calificadores de Informix son las constantes "TU_" definidas en el archivo `datatime.h` de Informix Client SDK. Las constantes son:

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)	

Correlaciones de tipos de datos directas por omisión para orígenes de datos JDBC

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos JDBC.

Tabla 115. Correlaciones de tipos de datos directas por omisión de JDBC

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BIGINT	-	-	-	-	-	-	BIGINT	8	-	-
BINARY	-	254	-	-	-	-	CHAR	-	-	S

Tabla 115. Correlaciones de tipos de datos directas por omisión de JDBC (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BINARY	255	32672	-	-	-	-	VARCHAR	-	-	S
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	-	-	S
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	-	-	S
VARBINARY	32673	2147483647	-	-	-	-	BLOB	2147483647	-	-
VARCHAR	-	32672	-	-	-	-	VARCHAR	-	-	-
VARCHAR	32673	2147483647	-	-	-	-	CLOB	2147483647	-	-

Nota:

1. El tipo federado es TIMESTAMP(0) si el parámetro de configuración date_compat se establece en ON.
2. Tipos de datos que sólo admite el controlador JDBC 4.0: NCHAR, NVARCHAR, LONGVARCHAR y NCLOB.

El derivador JDBC no admite los tipos de datos siguientes: DATALINK, OTHER, JAVA_OBJECT, DISTINCT, STRUCT, ARRAY y REF.

Correlaciones de tipos de datos directas por omisión para orígenes de datos Microsoft SQL Server

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos Microsoft SQL Server.

Tabla 116. Correlaciones de tipos de datos directas por omisión de Microsoft SQL Server

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
bigint ¹	-	-	-	-	-	-	BIGINT	-	-	-
binary	1	254	-	-	-	-	CHARACTER	-	-	S
binary	255	8000	-	-	-	-	VARCHAR	-	-	S
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	-	S
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	-	S
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	-	-	S
SQL_BINARY	255	8000	-	-	-	-	VARCHAR	-	-	S
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	-	-	S

Tabla 116. Correlaciones de tipos de datos directas por omisión de Microsoft SQL Server (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
SQL_INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
SQL_LONGVARCHAR	-	-	-	-	-	-	CLOB	2147483647	-	N
SQL_LONGVARBINARY	-	-	-	-	-	-	BLOB	-	-	S
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	-	-	S
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	-	S
tinyint	-	-	-	-	-	-	SMALLINT	2	-	-
uniqueidentifier	1	4000	-	-	S	-	VARCHAR	16	-	S
varbinary	1	8000	-	-	-	-	VARCHAR	-	-	S
varchar	1	8000	-	-	-	-	VARCHAR	-	-	N

Nota:

1. Esta correlación de tipos de datos sólo es válida con Microsoft SQL Server Versión 2000.

Correlaciones de tipos de datos directas por omisión para orígenes de datos ODBC

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos ODBC.

Tabla 117. Correlaciones de tipos de datos directas por omisión de ODBC (no se muestran todas las columnas)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
SQL_BIGINT	-	-	-	-	-	-	BIGINT	8	-	-
SQL_BINARY	1	254	-	-	-	-	CHARACTER	-	-	S
SQL_BINARY	255	32672	-	-	-	-	VARCHAR	-	-	S
SQL_BIT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_CHAR	1	254	-	-	-	-	CHAR	-	-	N
SQL_CHAR	255	32672	-	-	-	-	VARCHAR	-	-	N
SQL_DATE	-	-	-	-	-	-	DATE	-	-	-
SQL_DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-

Tabla 117. Correlaciones de tipos de datos directas por omisión de ODBC (no se muestran todas las columnas) (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
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	-	S
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)1	10	6	-
SQL_TIMESTAMP(p)-	-	-	-	-	-	-	TIMESTAMP(6)1	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	-	-	S
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

Nota:

1. El tipo federado es TIMESTAMP(0) si el parámetro de configuración date_compat se establece en ON.

Correlaciones de tipos de datos directas por omisión para orígenes de datos Oracle NET8

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos Oracle NET8.

Tabla 118. Correlaciones de tipos de datos directas por omisión de Oracle NET8

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BLOB	0	0	0	0	-	\0	BLOB	2147483647	0	S
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	0	N

Tabla 118. Correlaciones de tipos de datos directas por omisión de Oracle NET8 (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
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	S
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	S
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

Nota:

1.

- TIMESTAMP(p) representa una indicación de fecha y hora con una escala variable que va de 0 a 9. La escala de la indicación de fecha y hora de Oracle está correlacionada con el tipo de datos TIMESTAMP(6) predeterminado. Puede cambiar la correlación de tipos de datos por omisión y correlacionar el tipo de datos TIMESTAMP de Oracle con un tipo de datos TIMESTAMP federado de la misma escala utilizando una correlación de tipos de datos definidos por el usuario.
- Esta correlación de tipos de datos sólo es válida para configuraciones de cliente y servidor Oracle 9i (o posterior).

Correlaciones de tipos de datos directas por omisión para orígenes de datos Sybase

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos Sybase.

Tabla 119. Correlaciones de tipos de datos directas por omisión de Sybase CTLIB

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
binary	1	254	-	-	-	-	CHAR	-	-	S
binary	255	32672	-	-	-	-	VARCHAR	-	-	S
bit	-	-	-	-	-	-	SMALLINT	-	-	-
char	1	254	-	-	-	-	CHAR	-	-	N
char	255	32672	-	-	-	-	VARCHAR	-	-	N
char null (véase varchar)										
date	-	-	-	-	-	-	DATE	-	-	-
date	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-
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	-	-	-

Tabla 119. Correlaciones de tipos de datos directas por omisión de Sybase CTLIB (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
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 (véase 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	-	S
tinyint	-	-	-	-	-	-	SMALLINT	-	-	-
unichar ²	1	254	-	-	-	-	CHAR	-	-	N
unichar ²	255	32672	-	-	-	-	VARCHAR	-	-	N
unichar null (véase univarchar)										
univarchar ²	1	32672	-	-	-	-	VARCHAR	-	-	N
varbinary	1	32672	-	-	-	-	VARCHAR	-	-	S
varchar	1	32672	-	-	-	-	VARCHAR	-	-	N

Nota:

1. El tipo federado es TIMESTAMP(0) si el parámetro de configuración date_compat se establece en ON.
2. Válido para bases de datos federadas que no son Unicode.

Correlaciones de tipos de datos directas por omisión para orígenes de datos Teradata

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos Teradata.

Tabla 120. Correlaciones de tipos de datos directas por omisión de Teradata (no se muestran todas las columnas)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BLOB	1	2097088000	-	-	-	-	BLOB	-	-	-
BYTE	1	254	-	-	-	-	CHAR	-	-	S
BYTE	255	32672	-	-	-	-	VARCHAR	-	-	S
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	-	-	S
VARBYTE	32763	64000	-	-	-	-	BLOB	-	-	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
VARCHAR	32673	64000	-	-	-	-	CLOB	-	-	-
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	-	-
VARGRAPHIC	16337	32000	-	-	-	-	DBCLOB	-	-	-

Nota:

1. El tipo federado es TIMESTAMP(0) si el parámetro de configuración date_compat se establece en ON.

Correlaciones de tipos de datos directas de ejemplo

Puede utilizar las correlaciones de tipos de datos directas de ejemplo para sacar partido del soporte del tipo de datos TIMESTAMP con la precisión.

Para orígenes de datos Informix, estas correlaciones de tipos de datos se utilizan para tipos de columna de apodo, parámetros de procedimientos federados, paso a través y conjuntos de resultados de procedimientos federados.

Para orígenes de datos distintos de Informix, estas correlaciones de tipos de datos solamente afectan a las correlaciones para tipos de columna de apodo y parámetros de procedimientos federados. Las correlaciones no afectan a los conjuntos de resultados a través de y de procedimientos federados.

Correlaciones de tipos de datos directas - ejemplo de Informix

Al crear objetos federados, se puede usar la correlación de tipos de datos directa de ejemplo proporcionada por Informix.

Es necesario crear estas correlaciones antes de crear un objeto federado.

```
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);
```

Correlaciones de tipos de datos directas - ejemplo de Microsoft SQL Server

Al crear objetos federados, se puede usar la correlación de tipos de datos directa de ejemplo proporcionada por Microsoft SQL Server.

Es necesario crear estas correlaciones antes de crear un apodo o procedimiento federado.

```
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";
```

Correlaciones de tipos de datos directas - ejemplo de Oracle

Al crear objetos federados, se puede usar la correlación de tipos de datos directa de ejemplo proporcionada por Oracle.

Es necesario crear estas correlaciones antes de crear un apodo o procedimiento federado.

```
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;

```

Correlaciones de tipos de datos directas - ejemplo de Sybase

Al crear objetos federados, se puede usar la correlación de tipos de datos directa de ejemplo proporcionada por Sybase.

Es necesario crear estas correlaciones antes de crear un apodo o procedimiento federado.

```

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);

```

Correlaciones de tipos de datos directas - ejemplo de Teradata

Al crear objetos federados, se puede usar la correlación de tipos de datos directa de ejemplo proporcionada por Teradata.

Es necesario crear estas correlaciones antes de crear un apodo o procedimiento federado.

```

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);

```

Correlaciones de tipos de datos inversas por omisión

Para la mayor parte de orígenes de datos, las correlaciones de tipos por omisión se encuentran en los derivadores.

Los dos tipos de correlaciones entre los tipos de datos de origen de datos y los tipos de datos de base de datos federada son las correlaciones de tipos de datos directas e inversas. Una correlación de tipos de datos directa es la correlación de un tipo de datos remoto con un tipo de datos local comparable. El otro tipo de

correlación es una correlación de tipos de datos inversa, que se utiliza con DDL transparente para crear o modificar tablas remotas.

Las correlaciones de tipos de datos por omisión para orígenes de datos de la familia DB2 están en el derivador DRDA. Las correlaciones de tipos por omisión para Informix están en el derivador INFORMIX, etc.

Cuando se define una tabla o una vista remota para la base de datos federada, la definición incluye una correlación de tipos de datos inversa. La correlación es entre un tipo de datos de base de datos federada local para cada columna y el tipo de datos remoto correspondiente. Por ejemplo, existe una correlación de tipo de datos inversa por omisión en que el tipo de datos local REAL indica el tipo de datos SMALLFLOAT de Informix.

Las bases de datos federadas no admiten los tipos de datos LONG VARCHAR, LONG VARGRAPHIC ni los definidos por el usuario.

Cuando utilice la sentencia CREATE TABLE para crear una tabla remota, especifique los tipos de datos locales que desea incluir en la tabla remota. Estas correlaciones de tipos de datos inversas por omisión asignarán los tipos remotos correspondientes a estas columnas. Por ejemplo, suponga que utiliza la sentencia CREATE TABLE para definir una tabla Informix con una columna C2 y que en la sentencia especifica BIGINT como el tipo de datos de C2. La correlación de tipos de datos inversa por omisión de BIGINT depende de la versión de Informix con que se está creando la tabla. La correlación para C2 en la tabla Informix será con DECIMAL en Informix Versión 8 y con INT8 en Informix Versión 9.

Con la sentencia CREATE TYPE MAPPING puede alterar temporalmente la correlación de tipos de datos inversa por omisión o crear una correlación de tipos de datos inversa nueva.

En las tablas siguientes se muestran las correlaciones inversas por omisión entre tipos de datos locales de base de datos federada y tipos de datos de origen de datos remoto.

Estas correlaciones son válidas con todas las versiones compatibles, a menos que se indique lo contrario.

Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 Database para Linux, UNIX y Windows

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 Database para Linux, UNIX y Windows.

Tabla 121. Correlaciones de tipos de datos inversas por omisión de DB2 Database para Linux, UNIX y Windows (no se muestran todas las columnas)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits federados
BIGINT	-	8	-	-	-	-	BIGINT	-	-	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	S	-	CHAR	-	-	S
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE ¹	-	4	-	-	-	-	DATE	-	-	-

Tabla 121. Correlaciones de tipos de datos inversas por omisión de DB2 Database para Linux, UNIX y Windows (no se muestran todas las columnas) (continuación)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits federados
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(p)-	-	-	p	p	-	-	TIMESTAMP(p)-	-	p ³	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	S	-	VARCHAR	-	-	S
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	-	N
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	-

Nota:

1. Si el parámetro date_compat se establece en OFF, el tipo de datos DATE federado se correlaciona con el tipo de datos TIMESTAMP(0).
2. La opción de servidor SAME_DECFLT_ROUNDING se establece en N por omisión y las operaciones no se enviarán al origen de datos remoto a menos que SAME_DECFLT_ROUNDING se establezca en Y. Hallará más información sobre la opción de servidor SAME_DECFLT_ROUNDING en Información de consulta sobre opciones de DB2 Database.
3. Para la versión 9.5 (o anterior), la escala remota de TIMESTAMP es 6.

Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para System i

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para System i.

Tabla 122. Correlaciones de tipos de datos inversas por omisión de DB2 para System i (no se muestran todas las columnas)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operaciones de datos federadas	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHARACTER	-	-	N
CHARACTER	-	-	-	-	S	-	CHARACTER	-	-	S
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	NUMERIC	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-

Tabla 122. Correlaciones de tipos de datos inversas por omisión de DB2 para System i (no se muestran todas las columnas) (continuación)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operaciones de datos federadas	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
REAL	-	4	-	-	-	-	FLOAT	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	S	-	VARCHAR	-	-	S
VARGRAPHIC	-	-	-	-	-	-	VARG	-	-	N

Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para VM y VSE

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para VM y VSE

Tabla 123. Correlaciones de tipos de datos inversas por omisión de DB2 para VM y VSE (no se muestran todas las columnas)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	-
CHARACTER	-	-	-	-	S	-	CHAR	-	-	S
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	-	-	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	S	-	VARCHAR	-	-	S
VARGRAPH	-	-	-	-	-	-	VARGRAPH	-	-	N

Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para z/OS

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para z/OS

Tabla 124. Correlaciones de tipos de datos inversas por omisión de DB2 para z/OS (no se muestran todas las columnas)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	S	-	CHAR	-	-	S
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	-	-	-	-	S	-	VARCHAR	-	-	S
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	N

Correlaciones de tipos de datos inversas por omisión para orígenes de datos Informix

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos Informix.

Tabla 125. Correlaciones de tipos de datos inversas por omisión de Informix

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BIGINT ¹	-	-	-	-	-	-	DECIMAL	19	-	-
BIGINT ²	-	-	-	-	-	-	INT8	-	-	-
BLOB	1	2147483647	-	-	-	-	BYTE	-	-	-
CHARACTER	-	-	-	-	N	-	CHAR	-	-	-
CHARACTER	-	-	-	-	S	-	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	-	-	-

Tabla 125. Correlaciones de tipos de datos inversas por omisión de Informix (continuación)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
VARCHAR ¹	255	32672	-	-	N	-	TEXT	-	-	-
VARCHAR	-	-	-	-	S	-	BYTE	-	-	-
VARCHAR ²	255	2048	-	-	N	-	LVARCHAR	-	-	-
VARCHAR ²	2049	32672	-	-	N	-	TEXT	-	-	-

Nota:

1. Esta correlación de tipos de datos sólo es válida con el servidor Informix Versión 8 (o anterior).
2. Esta correlación de tipos de datos sólo es válida con el servidor Informix Versión 9 (o posterior).

Para el tipo de datos DATETIME de Informix, el servidor federado utiliza el calificador de alto nivel de Informix como REMOTE_LENGTH (longitud remota) y el calificador de bajo nivel de Informix como REMOTE_SCALE (escala remota).

Los calificadores de Informix son las constantes "TU_" definidas en el archivo datatime.h de Informix Client SDK. Las constantes son:

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)	

Correlaciones de tipos de datos inversas por omisión para orígenes de datos JDBC

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos JDBC que se ajustan a las correlaciones de tipos de datos del controlador JDBC de DB2.

Tabla 126. Correlaciones de tipos de datos inversas por omisión de JDBC

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BIGINT	-	-	-	-	-	-	BIGINT	-	-	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	-	-	-	-	S	-	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	-	-	-	-	S	-	VARBINARY	-	-	-
VARCHAR	-	-	-	-	N	-	VARCHAR	-	-	-

Tabla 126. Correlaciones de tipos de datos inversas por omisión de JDBC (continuación)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
VARGRAPHIC	-	-	-	-	-	-	NVARCHAR	-	-	-

Correlaciones de tipos de datos inversas por omisión para orígenes de datos Microsoft SQL Server

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos Microsoft SQL Server.

Tabla 127. Correlaciones de tipos de datos inversas por omisión de Microsoft SQL Server (no se muestran todas las columnas)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BIGINT ¹	-	-	-	-	-	-	bigint	-	-	-
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	S	-	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	-	-	-
VARCHAR	1	8000	-	-	S	-	varbinary	-	-	-
VARCHAR	8001	32672	-	-	S	-	image	-	-	-

Nota:

1. Esta correlación de tipos de datos sólo es válida con Microsoft SQL Server Versión 2000.

Correlaciones de tipos de datos inversas por omisión para orígenes de datos ODBC

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos ODBC.

Tabla 128. Correlaciones de tipos de datos inversas por omisión de ODBC (no se muestran todas las columnas)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BIGINT	-	-	-	-	-	-	SQL_BIGINT	-	-	-
BLOB	-	-	-	-	-	-	SQL_LONGVARBINARY	-	-	-
CHAR	-	-	-	-	S	-	SQL_BINARY	-	-	-
CHAR	-	-	-	-	N	-	SQL_CHAR	-	-	-

Tabla 128. Correlaciones de tipos de datos inversas por omisión de ODBC (no se muestran todas las columnas) (continuación)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
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	-	-	-	-	S	-	SQL_VARBINARY	-	-	-
VARCHAR	-	-	-	-	N	-	SQL_VARCHAR	-	-	-
VARGRAPHIC	-	-	-	-	S	-	SQL_WVARCHAR	-	-	-

Correlaciones de tipos de datos inversas por omisión para orígenes de datos Oracle NET8

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos Oracle NET8.

Tabla 129. Correlaciones de tipos de datos inversas por omisión de Oracle NET8

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BIGINT	0	8	0	0	N	\0	NUMBER	19	0	N
BLOB	0	2147483647	0	0	S	\0	BLOB	0	0	S
CHARACTER	1	254	0	0	N	\0	CHAR	0	0	N
CHARACTER	1	254	0	0	S	\0	RAW	0	0	S
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

Tabla 129. Correlaciones de tipos de datos inversas por omisión de Oracle NET8 (continuación)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
VARCHAR	1	2000	0	0	S	\0	RAW	0	0	S

Nota:

- Si el parámetro date_compat se establece en OFF, el tipo de datos DATE federado se correlaciona con el tipo de datos de tipo fecha de Oracle. Si el parámetro date_compat se establece en ON, el tipo de datos DATE federado (equivalente al tipo de datos TIMESTAMP(0)), se correlaciona con el tipo de datos TIMESTAMP(0) de Oracle.
- Esta correlación de tipos de datos sólo es válida con Oracle Versión 8.
- TIMESTAMP(p) representa una indicación de fecha y hora con una escala variable que va de 0 a 9 para Oracle y de 0 a 12 para la federación. Si la escala es de 0 a 9, el tipo de datos TIMESTAMP remoto de Oracle tiene la misma escala que el tipo de datos TIMESTAMP federado. Si la escala del tipo de datos TIMESTAMP federado es mayor que 9, la escala correspondiente del tipo de datos TIMESTAMP de Oracle es 9, que es la mayor escala posible de Oracle.
 - Esta correlación de tipos de datos sólo es válida con Oracle Versión 9, 10 y 11.

Correlaciones de tipos de datos inversas por omisión para orígenes de datos Sybase

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos Sybase.

Tabla 130. Correlaciones de tipos de datos inversas por omisión de Sybase CTLIB

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BIGINT	-	-	-	-	-	-	decimal	19	0	-
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	N	-	char	-	-	-
CHARACTER	-	-	-	-	S	-	binary	-	-	-
CLOB	-	-	-	-	-	-	text	-	-	-
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	-	-	S	-	varbinary	-	-	-
VARCHAR ¹	256	32672	-	-	S	-	image	-	-	-
VARCHAR ²	1	16384	-	-	S	-	varbinary	-	-	-
VARCHAR ²	16385	32672	-	-	S	-	image	-	-	-

Nota:

- Esta correlación de tipos de datos sólo es válida para CTLIB con la versión 12.0 (o anterior) del servidor Sybase.
- Esta correlación de tipos de datos sólo es válida para CTLIB con la versión 12.5 (o posterior) del servidor Sybase.

Correlaciones de tipos de datos inversas por omisión para orígenes de datos Teradata

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos Teradata.

Tabla 131. Correlaciones de tipos de datos inversas por omisión de Teradata (no se muestran todas las columnas)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BLOB	1	2097088000	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHARACTER	-	-	-
CHARACTER	-	-	-	-	S	-	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	-	-	-
TIME	-	-	-	-	-	-	TIME	15	-	-
TIMESTAMP	10	10	6	6	-	-	TIMESTAMP	26	6	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	S	-	VARBYTE	-	-	-
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	-

Nota:

1. El tipo de datos VARGRAPHIC de Teradata sólo puede contener la longitud especificada (de 1 a 32000) de un tipo de datos DBCLOB.

Correlaciones de tipos de datos por omisión Unicode

Determinados orígenes de datos son compatibles con correlaciones de tipos de datos directas y correlaciones de tipos de datos inversas para bases de datos Unicode.

Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos JDBC

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos JDBC cuando la base de datos federada es una base de datos Unicode.

Tabla 132. Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos JDBC

UTF-8	JDBC
Tipo de datos	Tipo de datos Longitud
CHAR	CHAR de 1 a 254 bytes
VARCHAR	VARCHAR de 1 a 32672 bytes

Tabla 132. Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos JDBC (continuación)

UTF-8	JDBC	
Tipo de datos	Tipo de datos	Longitud
CLOB	CLOB	-
GRAPHIC	NCHAR	de 1 a 127 caracteres
VARGRAPHIC	NVARCHAR	de 1 a 16336 caracteres
DBCLOB	NCLOB	-

Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos JDBC

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos JDBC cuando la base de datos federada es una base de datos Unicode.

Tabla 133. Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos JDBC

UTF-8	JDBC	
Tipo de datos	Longitud	Tipo de datos
CHAR	de 1 a 254 bytes	CHAR
VARCHAR	de 1 a 32672 bytes	VARCHAR
CLOB	de 1 a 2147483647 bytes	CLOB
GRAPHIC	de 1 a 127 caracteres	NCHAR
VARGRAPHIC	de 1 a 16336 caracteres	NVARCHAR
DBCLOB	de 1 a 1073741823 caracteres	NCLOB

Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos Microsoft SQL Server

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos Microsoft SQL Server cuando la base de datos federada es una base de datos Unicode.

Tabla 134. Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos Microsoft SQL Server

UTF-8	Microsoft SQL Server	
Tipo de datos	Tipo de datos	Longitud
CHAR	CHAR	de 1 a 254 bytes
VARCHAR	CHAR	de 255 a 8000 bytes
	VARCHAR	de 1 a 8000 bytes
CLOB	TEXT	-
GRAPHIC	NCHAR	de 1 a 127 caracteres
VARGRAPHIC	NCHAR	de 128 a 16336 caracteres
	NVARCHAR	de 1 a 16336 caracteres
DBCLOB	NTEXT	-

Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos Microsoft SQL Server

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos Microsoft SQL Server cuando la base de datos federada es una base de datos Unicode.

Tabla 135. Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos Microsoft SQL Server

UTF-8		Microsoft SQL Server
Tipo de datos	Longitud	Tipo de datos
CHAR	de 1 a 254 bytes	CHAR
VARCHAR	de 1 a 32672 bytes	VARCHAR
CLOB	de 1 a 2 147 483 647 bytes	TEXT
GRAPHIC	de 1 a 127 caracteres	NCHAR
VARGRAPHIC	de 1 a 16336 caracteres	NVARCHAR
DBCLOB	de 1 a 1 073 741 823 caracteres	NTEXT

Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos NET8

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos NET8 cuando la base de datos federada es una base de datos Unicode.

Tabla 136. Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos NET8

UTF-8	Oracle	
Tipo de datos	Tipo de datos	Longitud
CHAR	CHAR	de 1 a 254 bytes
VARCHAR	CHAR	de 255 a 2000 bytes
	VARCHAR2	de 1 a 4000 bytes
DBCLOB	NCLOB	
GRAPHIC	NCHAR	de 1 a 127 caracteres
VARGRAPHIC	NCHAR	de 128 a 1000 caracteres
	NVARCHAR2	de 1 a 2000 caracteres

Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos NET8

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos NET8 cuando la base de datos federada es una base de datos Unicode.

Tabla 137. Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos NET8

UTF-8		Oracle
Tipo de datos	Longitud	Tipo de datos
CHAR	de 1 a 254 bytes	CHAR
VARCHAR	de 1 a 4000 bytes	VARCHAR2
CLOB	de 1 a 2 147 483 647 bytes	CLOB
GRAPHIC	de 1 a 127 caracteres	NCHAR
VARGRAPHIC	de 1 a 2000 caracteres	NVARCHAR2
DBCLOB	de 1 a 1 073 741 823 caracteres	NCLOB

Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos ODBC

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos ODBC cuando la base de datos federada es una base de datos Unicode.

Tabla 138. Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos ODBC

UTF-8	ODBC	
Tipo de datos	Tipo de datos	Longitud
CHAR	SQL_CHAR	de 1 a 254 bytes
VARCHAR	SQL_CHAR	de 255 a 32672 bytes
	SQL_VARCHAR	de 1 a 32672 bytes
CLOB	SQL_LONGVARCHAR	-
GRAPHIC	SQL_WCHAR	de 1 a 127 caracteres
VARGRAPHIC	SQL_WCHAR	de 128 a 16336 caracteres
	SQL_WVARCHAR	de 1 a 16336 caracteres
DBCLOB	SQL_WLONGVARCHAR	-

Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos ODBC

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos ODBC cuando la base de datos federada es una base de datos Unicode.

Tabla 139. Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos ODBC

UTF-8	ODBC	
Tipo de datos	Longitud	Tipo de datos
CHAR	de 1 a 254 bytes	SQL_CHAR
VARCHAR	de 1 a 32672 bytes	SQL_VARCHAR
CLOB	de 1 a 2 147 483 647 bytes	SQL_LONGVARCHAR
GRAPHIC	de 1 a 127 caracteres	SQL_WCHAR

Tabla 139. Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos ODBC (continuación)

UTF-8		ODBC
Tipo de datos	Longitud	Tipo de datos
VARGRAPHIC	de 1 a 16336 caracteres	SQL_WVARCHAR
DBCLOB	de 1 a 1 073 741 823 caracteres	SQL_WLONGVARCHAR

Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos Sybase

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos Sybase CTLIB cuando la base de datos federada es una base de datos Unicode.

Tabla 140. Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos Sybase CTLIB

UTF-8	Sybase	
Tipo de datos	Tipo de datos	Longitud
CHAR	char	de 1 a 254 bytes
	nchar	de 1 a 127 caracteres
VARCHAR	char	de 255 a 32672 bytes
	varchar	de 1 a 32672 bytes
	nchar	de 128 a 16336 caracteres
	nvarchar	de 1 a 16336 caracteres
CLOB	text	
GRAPHIC	unichar	de 1 a 127 caracteres
VARGRAPHIC	unichar	de 128 a 16336 caracteres
	univarchar	de 1 a 16336 caracteres

Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos Sybase

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos Sybase CTLIB cuando la base de datos federada es una base de datos Unicode.

Tabla 141. Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos Sybase CTLIB

UTF-8	Sybase	
Tipo de datos	Longitud	Tipo de datos
CHAR	de 1 a 254 bytes	char
VARCHAR	de 1 a 32672 bytes	varchar
CLOB	de 1 a 2 147 483 647 bytes	text
GRAPHIC	de 1 a 127 caracteres	unichar
VARGRAPHIC	de 1 a 16336 caracteres	univarchar

Tipos de datos admitidos para orígenes de datos no relacionales

Para la mayoría de los orígenes de datos no relacionales, debe especificar la información de columna, incluido el tipo de datos, cuando se crean los apodos para acceder al origen de datos.

Algunos de los derivadores no relacionales crean todas las columnas necesarias para acceder a un origen de datos. Estas se conocen como *columnas fijas*. Otros derivadores permiten especificar algunos o todos los tipos de datos de las columnas en la sentencia CREATE NICKNAME.

En las secciones siguientes se indican los derivadores para los que se pueden especificar tipos de datos, y los tipos de datos que admite cada derivador.

Tipos de datos admitidos por el derivador BioRS

En la tabla siguiente se indican los tipos de datos DB2 que admite el derivador BioRS.

Tabla 142. Tipos de datos BioRS que se correlacionan con tipos de datos DB2

Tipos de datos BioRS	Tipo de datos DB2
AUTHOR	CHARACTER, CLOB, VARCHAR
DATE	CHARACTER, CLOB, VARCHAR
NUMBER	CHARACTER, CLOB, VARCHAR
REFERENCE	CHARACTER, CLOB, VARCHAR
TEXT	CHARACTER, CLOB, VARCHAR

La longitud máxima permitida para el tipo de datos CLOB es de 5 megabytes.

Tipos de datos admitidos por el derivador Excel

En la tabla siguiente se indican los tipos de datos DB2 que admite el derivador Excel.

Tabla 143. Tipos de datos Excel que se correlacionan con tipos de datos DB2

Tipos de datos Excel	Tipo de datos DB2
carácter	CHARACTER
fecha	DATE
número	DOUBLE
número	FLOAT
entero	INTEGER
carácter	VARCHAR

Tipos de datos admitidos por el derivador Script

En la tabla siguiente se indican los tipos de datos DB2 que admite el derivador Script.

Tabla 144. Tipos de datos Script que se correlacionan con tipos de datos DB2

Tipos de datos XML	Tipo de datos DB2
carácter	BLOB
carácter	CHARACTER
carácter	CHARACTER FOR BIT DATA
carácter	CLOB (la longitud máxima es de 5 megabytes)
fecha	DATE
número	DECIMAL
número	DOUBLE
número	FLOAT
entero	INTEGER
número	REAL
entero	SMALLINT
carácter	VARCHAR
carácter	VARCHAR FOR BIT DATA

Tipos de datos admitidos por el derivador de archivos con estructura de tabla

En la tabla siguiente se indican los tipos de datos DB2 que admite el derivador de archivos con estructura de tabla

Tabla 145. Tipos de datos archivo con estructura de tabla que se correlacionan con tipos de datos DB2

Tipos de datos archivo con estructura de tabla	Tipo de datos DB2
carácter	CHARACTER
carácter	CLOB (la longitud máxima es de 5 megabytes)
número	DECIMAL
número	DOUBLE
número	FLOAT
entero	INTEGER
número	REAL
entero	SMALLINT
carácter	VARCHAR

Tipos de datos admitidos por el derivador de servicios web

En la tabla siguiente se indican los tipos de datos DB2 que admite el derivador de servicios web. El derivador de servicios web utiliza tipos de datos XML.

Tabla 146. Tipos de datos XML que se correlacionan con tipos de datos DB2 para el derivador de servicios web

Tipos de datos XML	Tipo de datos DB2
carácter	BLOB
carácter	CHARACTER

Tabla 146. Tipos de datos XML que se correlacionan con tipos de datos DB2 para el derivador de servicios web (continuación)

Tipos de datos XML	Tipo de datos DB2
carácter	CHARACTER FOR BIT DATA
carácter	CLOB (la longitud máxima es de 5 megabytes)
fecha	DATE
número	DECIMAL
número	DOUBLE
número	FLOAT
entero	INTEGER
número	REAL
entero	SMALLINT
carácter	VARCHAR
carácter	VARCHAR FOR BIT DATA

Tipos de datos admitidos por el derivador XML

En la tabla siguiente se indican los tipos de datos DB2 que admite el derivador XML.

Tabla 147. Tipos de datos XML que se correlacionan con tipos de datos DB2 para el derivador XML

Tipos de datos XML	Tipo de datos DB2
carácter	BLOB
carácter	CHARACTER
carácter	CHARACTER FOR BIT DATA
carácter	CLOB (la longitud máxima es de 5 megabytes)
fecha	DATE
número	DECIMAL
número	DOUBLE
número	FLOAT
entero	INTEGER
número	REAL
entero	SMALLINT
carácter	VARCHAR
carácter	VARCHAR FOR BIT DATA
documento XML	XML

Documentación del producto

La documentación se proporciona en diversas ubicaciones y formatos, también en la ayuda que se abre directamente desde la interfaz del producto, en un Information Center (centro de información) para toda la suite y en manuales en archivos PDF.

También puede realizar pedidos de publicaciones de IBM en formato de copia en papel en línea o por medio de su representante local de IBM.

Para solicitar publicaciones en línea, vaya al Centro de publicaciones de IBM en www.ibm.com/shop/publications/order.

Puede enviar sus comentarios sobre la documentación de los modos siguientes:

- Formulario de comentarios del lector en línea: www.ibm.com/software/data/rcf/
- Correo electrónico: comments@us.ibm.com

Documentación accesible

La documentación se ofrece en formato XHTML, que se puede ver en la mayoría de navegadores web.

XHTML permite visualizar la documentación según las preferencias de visualización que ha establecido en el navegador. También permite utilizar lectores de pantalla y otras tecnologías de asistencia.

Los diagramas de sintaxis se proporcionan en formato decimal de puntos. Este formato solo está disponible si accede a la documentación en línea utilizando el lector de pantalla.

Avisos

La presente información se ha desarrollado para productos y servicios ofrecidos en Estados Unidos.

Es posible que IBM no comercialice en otros países los productos, servicios o características que se describen en este manual. Consulte al representante local de IBM para obtener información sobre los productos y servicios que actualmente pueden adquirirse en su zona. Cualquier referencia a un producto, programa o servicio de IBM no pretende afirmar ni implicar que sólo se pueda utilizar dicho producto, programa o servicio de IBM. En su lugar se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no vulnere ninguno de los derechos de propiedad intelectual de IBM. Sin embargo, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patentes en tramitación que afecten al tema tratado en este documento. La posesión de este documento no otorga ninguna licencia sobre dichas patentes. Puede realizar consultas sobre licencias escribiendo a:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 EE. UU.

Para formular consultas relacionadas con el juego de caracteres de doble byte (DBCS), póngase en contacto con el departamento de la propiedad intelectual de IBM de su país o envíe las consultas, por escrito, a la siguiente dirección:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 Japón

El párrafo siguiente no es aplicable al Reino Unido ni a ningún país en donde tales disposiciones sean incompatibles con la legislación local:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN TAL CUAL, SIN GARANTÍA DE NINGUNA CLASE, NI EXPLÍCITA NI IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN DE DERECHOS, COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN DETERMINADO. Algunos estados no permiten la exclusión de garantías expresas o implícitas en determinadas transacciones, por lo que es posible que esta declaración no sea aplicable en su caso.

Esta publicación puede contener inexactitudes técnicas o errores tipográficos. Periódicamente se efectúan cambios en la información aquí contenida; dichos cambios se incorporarán a las nuevas ediciones de la publicación. IBM puede efectuar, en cualquier momento y sin previo aviso, mejoras y cambios en los productos y programas descritos en esta publicación.

Las referencias hechas en esta publicación a sitios Web que no son de IBM se proporcionan sólo para la comodidad del usuario y no constituyen un aval de esos sitios Web. La información contenida en estos sitios Web no forma parte de la información del presente producto IBM, y el usuario es responsable de la utilización de dichos sitios.

IBM puede utilizar o distribuir cualquier información que se le facilite de la manera que considere adecuada, sin contraer por ello ninguna obligación con el remitente.

Los licenciarios de este programa que deseen obtener información sobre él con el fin de habilitar: (i) el intercambio de información entre programas creados de forma independiente y otros programas (incluido éste) y (ii) el uso mutuo de la información intercambiada, deben ponerse en contacto con:

IBM Corporation
J46A/G4
555 Bailey Avenue
San José, CA 95141-1003 EE.UU.

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluido en algunos casos el pago de una tarifa.

El programa bajo licencia descrito en este documento y todo el material bajo licencia asociado a él los proporciona IBM según los términos del Acuerdo de Cliente de IBM, el Acuerdo Internacional de Programas Bajo Licencia de IBM o cualquier acuerdo equivalente entre el usuario e IBM.

Los datos de rendimiento contenidos en este documento se obtuvieron en un entorno controlado. Por lo tanto, los resultados obtenidos en otros entornos operativos pueden variar significativamente. Algunas mediciones pueden haberse realizado en sistemas experimentales y no es seguro que estas mediciones sean las mismas en los sistemas disponibles comercialmente. Además, algunas mediciones pueden haberse calculado mediante extrapolación. Los resultados reales pueden variar. Los usuarios del presente manual deben verificar los datos aplicables para su entorno específico.

La información referente a productos que no son de IBM se ha obtenido de los proveedores de esos productos, de sus anuncios publicados o de otras fuentes disponibles públicamente. IBM no ha probado esos productos y no puede confirmar la exactitud del rendimiento, la compatibilidad ni ninguna otra afirmación referente a productos que no son de IBM. Las preguntas sobre las prestaciones de productos que no son de IBM deben dirigirse a los proveedores de esos productos.

Todas las declaraciones de intenciones de IBM están sujetas a cambio o cancelación sin previo aviso, y sólo representan objetivos.

Esta información sólo tiene como objeto la planificación. La información incluida en este documento puede cambiar antes de que los productos descritos se hagan disponibles.

Este manual contiene ejemplos de datos e informes que se utilizan en operaciones comerciales diarias. Para ilustrarlos de la forma más completa posible, los ejemplos incluyen nombres de personas, empresas, marcas y productos. Todos estos

nombres son ficticios y cualquier similitud con nombres y direcciones utilizados por una empresa real es totalmente fortuita.

LICENCIA DE COPYRIGHT:

Este manual contiene programas de aplicaciones de ejemplo escritos en lenguaje fuente, que muestran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo como desee, sin pago alguno a IBM con la intención de desarrollar, utilizar, comercializar o distribuir programas de aplicaciones de acuerdo con la interfaz de programación de aplicaciones correspondiente a la plataforma operativa para la que están escritos los programas de ejemplo. Estos ejemplos no se han probado exhaustivamente bajo todas las condiciones. Por lo tanto, IBM no puede asegurar ni implicar la fiabilidad, utilidad o función de estos programas. Los programas de ejemplo se proporcionan "tal como están", sin garantías de ningún tipo. IBM no se hace responsable de los daños que se hayan podido causar debido al uso de los programas de ejemplo.

Cada copia o parte de estos programas de ejemplo o cualquier trabajo derivado debe incluir una nota de copyright como la siguiente:

© (nombre de la empresa) (año). Partes de este código proceden de programas de ejemplo de IBM Corp. © Copyright IBM Corp. _entrar el año o los años_. Reservados todos los derechos.

Si examina esta información mediante una copia software, es posible que las fotografías y las ilustraciones en color no aparezcan.

Marcas registradas

Las marcas registradas de IBM y algunas marcas registradas ajenas a IBM se señalan la primera vez que aparecen en este documento con el símbolo correspondiente.

IBM, el logotipo de IBM e ibm.com son marcas registradas de International Business Machines Corp. en muchos países o regiones de alrededor del mundo. Otros nombres de servicios o de productos pueden ser marcas registradas de IBM u otras empresas. Encontrará una lista actualizada de las marcas registradas de IBM en el apartado "Copyright and trademark information" de www.ibm.com/legal/copytrade.shtml.

Los términos siguientes son marcas registradas de otras empresas:

Adobe, el logotipo de Adobe, PostScript y el logotipo de PostScript son marcas registradas de Adobe Systems Incorporated en los Estados Unidos y/o en otros países.

IT Infrastructure Library es una marca registrada de Central Computer and Telecommunications Agency, que ahora forma parte de la Office of Government Commerce.

Intel, el logotipo de Intel, Intel Inside, el logotipo de Intel Inside, Intel Centrino, el logotipo de Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium y Pentium son marcas registradas de Intel Corporation o de subsidiarias en los Estados Unidos y en otros países.

Linux es una marca registrada de Linus Torvalds en Estados Unidos o en otros países.

Microsoft, Windows, Windows NT y el logotipo de Windows son marcas registradas de Microsoft Corporation en los Estados Unidos y/o en otros países.

ITIL es una marca registrada y una marca registrada de comunidad registrada de la Office of Government Commerce y está registrada en la Oficina de Patentes y Marcas Registradas de los EE.UU.

UNIX es una marca registrada de The Open Group en Estados Unidos y en otros países.

Cell Broadband Engine es una marca registrada de Sony Computer Entertainment, Inc. en los Estados Unidos, otros países o en ambos, y se utiliza bajo licencia a partir de ello.

Java y todas las marcas registradas basadas en Java son marcas registradas de Sun Microsystems, Inc. en los Estados Unidos o en otros países

Otros nombres de empresas, productos o servicios pueden ser marcas registradas o marcas de servicio de terceros.

Índice

A

- accesibilidad 395, 397
- aislamiento a nivel de conexión
 - orígenes de datos soportados 281
- aislamiento a nivel de sentencia
 - orígenes de datos soportados 281
- apodo y column, opciones derivador de servicios web 226, 247
- apodos
 - creación 237
 - servicios web 239
 - crear 238
 - CREATE NICKNAME
 - ejemplos 195
 - descripción 176, 247
 - lenguaje de descripción de servicios web (WSDL) 226
 - orígenes de datos de servicios Web 237
 - orígenes de datos JDBC 99
 - orígenes de datos Oracle 164
 - orígenes de datos Sybase 194
 - orígenes de datos Teradata 222
 - registrar para orígenes de datos Microsoft SQL Server 112
 - registrar para orígenes de datos ODBC 125, 143
 - registro 62
 - registro para scripts 176
 - registro para tablas, vistas y sinónimos de Informix 89
 - servicios web, orígenes de datos 238
 - técnicas de base de datos 249
 - Teradata, orígenes de datos 223
- apodos XML
 - vistas federadas 276
- archivo de configuración de cliente de Sybase
 - configurar y probar 186
- archivo de configuración del cliente Informix
 - configuración y prueba 77
- archivo de configuración del cliente Oracle
 - configuración y prueba 157
- archivo de configuración del cliente Sybase
 - configuración y prueba 187
- archivos con estructura de tabla características federadas soportadas 281
- opciones de apodo 339
- opciones de columna 339
- opciones de derivador 339
- opciones de servidor 339
- soporte de Unicode 11, 12
- tipos de datos admitidos 392

archivos de biblioteca

- enlace 4, 5
- nombres de archivo 4

- archivos de biblioteca del derivador
 - nombres de archivo 4
- archivos de configuración
 - cliente Informix
 - configuración y prueba 77
- archivos estructurados en tablas, orígenes de datos
 - atributos de tipos de archivo 198
 - CREATE SERVER, ejemplos de la sentencia 202
 - derivador, visión general 198
 - visión general 197
- archivos Excel
 - preparar para el acceso 65
 - tipos de datos admitidos 392
- asignación de dos fases
 - orígenes de datos soportados 281
- aviso legales 399

B

- banco de datos
 - descripción 21
- base de datos DB2
 - características federadas soportadas 281
- Base de datos DB2
 - opciones de apodo 287
 - opciones de columna 287
 - opciones de correlación de usuarios 287
 - opciones de derivador 287
 - opciones de servidor 287
- bases de datos
 - catalogación, remotas 56
- bases de datos federadas
 - crear 7
 - juegos de códigos 7
 - secuencia de clasificación 7
- biblioteca
 - derivador de scripts 174
- bibliotecas
 - derivador de archivos con estructura de tabla 201
 - derivador de BioRS 25
 - derivador de Excel 70
 - derivador de Teradata 216
 - derivador XML 260
- bibliotecas de derivador 234
 - nombres de archivo de scripts 174
 - nombres de archivo de Teradata 216
 - nombres de archivos BioRS 25
 - nombres de archivos con estructura de tabla 201
 - nombres de archivos Excel 70
 - XML 260
- Biomax Informatics 21
- BioRS
 - características federadas soportadas 281
 - opciones de apodo 283

BioRS (*continuación*)

- opciones de columna 283
- opciones de correlación de usuarios 283
- opciones de derivador 283
- opciones de servidor 283
- tipos de datos admitidos 392

BioRS, orígenes de datos

- CREATE WRAPPER, ejemplos de sentencia 26
- descripción 21
- información estadística 51
- optimizar el rendimiento del derivador 50
- rendimiento de las consultas, optimizar 51
- servidor proxy, ejemplo de derivador 26

C

- cabeceras SOAP
 - acceso federado 245
- Capa de sockets seguros (SSL)
 - orígenes de datos soportados 281
- casos prácticos
 - acceso a archivos Excel 74
- catálogo
 - Consulte catálogo global 357
- catálogo global
 - vistas que contienen información federada 357
- Centro de control
 - acceder 1
- claves de licencia
 - Federación 6
- comillas
 - uso en sentencias de SQL 15
- CONCENTRATOR, parámetro 1
- conectividad
 - prueba de las conexiones del servidor Oracle 163
 - resolución de problemas para orígenes de datos Oracle 163
- conexiones
 - prueba 61
 - resolución de problemas 61, 87, 99, 111, 124, 142, 164, 193, 221
- configuración
 - servidores federados 18
- consultas
 - acceso a orígenes de datos servicios web 252
 - acceso a orígenes de datos BioRS 38, 44
 - acceso a orígenes de datos Excel 73
 - consultas XML 279
 - ejemplos
 - derivadores de servicios web 249
 - operaciones de igualdad BioRS 41

- consultas (*continuación*)
 - orígenes de datos de scripts, acceso a los orígenes de datos 180
 - contextos fiables federados
 - orígenes de datos soportados 281
 - contraseñas
 - distinción entre mayúsculas y minúsculas 15
 - control de acceso basado en etiquetas
 - orígenes de datos soportados 281
 - controlador ODBC de Excel
 - acceso a libros Microsoft Excel 130
 - permisos de acceso 132
 - conversión de página de códigos
 - Informix 80
 - Oracle, orígenes de datos 156
 - correlación de tipos de datos inversa
 - Unicode
 - orígenes de datos NET8 390
 - correlacionar
 - documentos XML 247
 - correlaciones de funciones
 - opciones 361
 - correlaciones de tipos de datos
 - directas 365
 - inversas 378
 - correlaciones de tipos de datos directas
 - correlaciones por omisión 365
 - Unicode
 - Microsoft SQL Server 388
 - orígenes de datos JDBC 387
 - orígenes de datos NET8 389
 - orígenes de datos ODBC 390
 - orígenes de datos Sybase 391
 - correlaciones de tipos de datos directas por omisión
 - ejemplo de Informix 377
 - ejemplo de Microsoft SQL Server 377
 - ejemplo de Oracle 377
 - ejemplo de Sybase 378
 - ejemplo de Teradata 378
 - ejemplos 376
 - orígenes de datos DB2 Database para Linux, UNIX y Windows 365
 - orígenes de datos DB2 para System i 366
 - orígenes de datos DB2 para VM y VSE 367
 - orígenes de datos DB2 para z/OS 368
 - orígenes de datos Informix 368, 382
 - orígenes de datos JDBC 369
 - orígenes de datos Microsoft SQL Server 371, 384
 - orígenes de datos ODBC 372
 - orígenes de datos Oracle NET8 373
 - orígenes de datos Sybase 374
 - orígenes de datos Teradata 376, 387
 - correlaciones de tipos de datos inversas
 - correlaciones por omisión 378
 - Unicode
 - orígenes de datos JDBC 388
 - orígenes de datos Microsoft SQL Server 389
 - orígenes de datos ODBC 390
 - orígenes de datos Sybase 391
 - correlaciones de tipos de datos inversas por omisión
 - orígenes de datos DB2 Database para Linux, UNIX y Windows 379
 - orígenes de datos DB2 para System i 380
 - orígenes de datos DB2 para VM y VSE 381
 - orígenes de datos DB2 para z/OS 382
 - orígenes de datos JDBC 383
 - orígenes de datos ODBC 384
 - orígenes de datos Oracle NET8 385
 - orígenes de datos Sybase 386
 - correlaciones de usuario
 - crear 60
 - crear para orígenes de datos
 - Informix 86
 - definir para orígenes de datos
 - Microsoft SQL Server 109
 - definir para orígenes de datos
 - ODBC 123, 140
 - orígenes de datos JDBC 97
 - correlaciones de usuarios
 - creación de orígenes de datos
 - Sybase 192
 - definición para orígenes de datos
 - Oracle 162
 - definición para orígenes de datos
 - Teradata 219
 - orígenes de datos OLE DB 151
 - CREATE NICKNAME
 - orígenes de datos JDBC 100
 - CREATE NICKNAME, sentencia
 - servicios web, orígenes de datos 238
 - CREATE SERVER
 - orígenes de datos JDBC 95
 - CREATE SERVER, ejemplos
 - Informix 83
 - CREATE SERVER, sentencia
 - archivos estructurados en tablas, ejemplos 202
 - ejemplo de servicios web 235
 - ejemplos XML 263
 - CREATE USER MAPPING
 - orígenes de datos JDBC 97
 - CREATE USER MAPPING, ejemplos
 - orígenes de datos OLE DB 152
 - orígenes de datos Oracle 162
 - orígenes de datos Teradata 220
 - CREATE VIEW, sentencia
 - ejemplos XML 276
 - CREATE WRAPPER
 - ejemplos para ODBC 119, 137
 - orígenes de datos JDBC 94
 - CREATE WRAPPER, sentencia
 - BioRS, ejemplos 26
 - ejemplos XML 260
- ## D
- daemon
 - archivo de configuración
 - ejemplos de script 169
 - configurar para el script 169
 - iniciar para el script 171
 - DAEMON_LOGFILE_DIR
 - daemon de scripts 169
 - DAEMON_PORT
 - daemon de scripts 169
 - DB2 para Linux, UNIX y Windows
 - correlaciones de tipos de datos directas por omisión 365
 - correlaciones de tipos de datos inversas por omisión 378
 - DB2 para System i
 - correlaciones de tipos de datos directas por omisión 365
 - correlaciones de tipos de datos inversas por omisión 378
 - DB2 para z/OS y OS/390
 - correlaciones de tipos de datos directas por omisión 365
 - correlaciones de tipos de datos inversas por omisión 378
 - db2dj.ini, archivo 2
 - restricciones 4
 - definiciones de servidor
 - orígenes de datos JDBC 94
 - registrar para orígenes de datos
 - Microsoft SQL Server 107
 - registrar para orígenes de datos
 - ODBC 120, 138
 - registro 58
 - registro de orígenes de datos Informix
 - línea de mandatos DB2 83
 - registro para los orígenes de datos
 - OLE DB 150
 - registro para los orígenes de datos
 - Oracle 160
 - registro para los orígenes de datos
 - Sybase 189
 - registro para los orígenes de datos
 - Teradata 217
 - servicios web 234
 - derivador de DB2
 - registro 57
 - derivador de scripts
 - creación 174
 - funciones personalizadas 168
 - visión general 166
 - derivador de servicios web
 - orígenes de datos 226
 - derivador DRDA
 - nombres de archivos de biblioteca 57
 - derivadores
 - archivos de biblioteca de derivador de
 - OLE DB 150
 - archivos de biblioteca de derivador de
 - Oracle 159
 - archivos de biblioteca del derivador
 - de JDBC 93
 - archivos de biblioteca del derivador
 - de ODBC 119, 136
 - bibliotecas de derivador de Microsoft
 - SQL Server 106

derivadores (*continuación*)
 comprobación de enlaces de tiempo
 de ejecución para orígenes de datos
 Teradata 210
 configuración de derivadores de OLE
 DB 148
 configuración del acceso a orígenes de
 datos Teradata 209
 configurar derivadores de JDBC 90
 configurar derivadores de
 ODBC 114, 134
 orígenes de datos de servicios
 Web 226, 233
 registrar derivadores de Microsoft
 SQL Server 105
 registro de derivadores de
 Oracle 158
 registro de derivadores de
 Sybase 188
 registro de derivadores de
 Teradata 215
 registro de derivadores JDBC 92
 registro de derivadores ODBC 118,
 135
 script 166
 derivadores de Informix
 creación de correlaciones de
 usuario 86
 variables de entorno 78
 derivadores de servicios web
 creación 233
 derivadores Informix
 mejorar el rendimiento 88
 registro 81
 variables de entorno 79
 distinción entre mayúsculas y minúsculas
 preservar 15
 documentación
 accesible 395, 397

E

ejemplos
 archivos Excel 63, 65
 consultas BioRS 42, 44
 consultas Excel 73
 correlación de documentos XML 255
 CREATE SERVER, sentencia, archivos
 estructurados en tablas 202
 CREATE SERVER, sentencia, servicios
 web 235
 CREATE SERVER, sentencia,
 XML 263
 CREATE VIEW, sentencia XML 276
 CREATE WRAPPER, sentencia,
 XML 260
 derivadores de servicios web 239,
 249
 ejemplos XML 279
 mandato db2script_daemon 172
 orígenes de datos BioRS
 ejemplo de correlación de usuarios,
 servidor proxy 32
 orígenes de datos de scripts, CREATE
 NICKNAME 176
 orígenes de datos de scripts, CREATE
 SERVER 175

ejemplos (*continuación*)
 sentencia CREATE NICKNAME,
 archivo con estructura de tabla 203
 sentencia CREATE NICKNAME,
 BioRS 35
 sentencia CREATE NICKNAME,
 Excel 72
 sentencia CREATE SERVER,
 BioRS 28
 sentencia CREATE SERVER, Excel 71
 sentencia CREATE USER
 MAPPING 32
 sentencia CREATE WRAPPER,
 archivo con estructura de tabla 200
 sentencia CREATE WRAPPER,
 BioRS 26
 sentencia CREATE WRAPPER,
 Excel 69
 sentencia UPDATE, BioRS 53
 servidor proxy
 ejemplo de correlación de usuarios
 BioRS 32
 ejemplos, consultas
 servicios web 252
 ejemplos, CREATE SERVER
 Teradata 218
 ejemplos, sentencia CREATE USER
 MAPPING
 Informix 86
 orígenes de datos Sybase 192
 ejemplos de CREATE NICKNAME
 orígenes de datos Microsoft SQL
 Server 113
 orígenes de datos ODBC 126, 144
 orígenes de datos Oracle 165
 ejemplos de CREATE SERVER
 orígenes de datos Microsoft SQL
 Server 108
 orígenes de datos OLE DB 150
 orígenes de datos Oracle 161
 orígenes de datos Sybase 190
 ejemplos de CREATE USER MAPPING
 orígenes de datos Microsoft SQL
 Server 110
 orígenes de datos ODBC 123, 141
 elemento AllText
 BioRS 44
 enlace
 archivos de biblioteca 4
 errores de operación de UPDATE o
 DELETE
 resolución de problemas para orígenes
 de datos Teradata 225
 especificar valores
 opciones 15
 esquemas XML
 correlacionar con apodos 247
 estadísticas
 orígenes de datos BioRS 51
 Excel
 características federadas
 soportadas 281
 opciones de apodo 294
 opciones de derivador 294
 opciones de servidor 294
 Excel, orígenes de datos
 descripción 63

F

federación
 instalación 1
 FEDERATED, parámetro 1
 Federation
 habilitar 1
 resolución de problemas 5
 función WSSCRIPT.ARGS 168
 funciones personalizadas
 BioRS 38
 BioRS, descripciones 24
 BioRS, registrar 23
 script 168

I

ID de usuario
 distinción entre mayúsculas y
 minúsculas 15
 importación de datos
 orígenes de datos soportados 281
 indicadores de salud federados
 orígenes de datos soportados 281
 Informix
 archivos de biblioteca 5
 características federadas
 soportadas 281
 conversión de página de códigos 80
 correlaciones de tipos de datos
 directas por omisión 365
 correlaciones de tipos de datos
 inversas por omisión 378
 djsxlinkInformix, script 5
 ejemplo de sentencia CREATE
 NICKNAME 89
 nombres de archivos de biblioteca 82
 opciones de columna 295
 opciones de correlación de
 usuarios 295
 opciones de derivador 295
 opciones de servidor 295
 sentencia CREATE SERVER 83
 Sentencia CREATE USER
 MAPPING 86
 vía de acceso al directorio por
 omisión 82

J

JDBC
 características federadas
 soportadas 281
 opciones de columna 301
 opciones de correlación de
 usuarios 301
 opciones de derivador 301
 opciones de servidor 301
 JDBC, orígenes de datos
 configurar servidores federados 91
 juegos de caracteres
 resolución de problemas de
 configuraciones de orígenes de datos
 Teradata 215, 225
 juegos de códigos
 base de datos federada 7

L

- lectores de pantalla 395, 397
- lenguaje de descripción de servicios web (WSDL)
 - apodos 226
 - correlacionar con apodos 247
- libros Microsoft Excel
 - acceder 130
 - permisos de acceso 132
- licencias
 - Federación 6
- LOB
 - orígenes de datos soportados, lectura y escritura 281
 - orígenes de datos soportados, sólo lectura 281

M

- mandato db2fedsvrcfg
 - crear tablas de prueba 129, 147
 - ejemplos 128, 146
 - sintaxis 128, 146
- mandato db2script_daemon 172
- mandatos
 - emitir 1
- marcas registradas 401
- mayúsculas
 - uso en sentencias de SQL 15
- Microsoft SQL Server
 - archivos de biblioteca 5
 - características federadas soportadas 281
 - correlaciones de tipos de datos directas por omisión 365
 - correlaciones de tipos de datos directas por omisión Unicode 388
 - correlaciones de tipos de datos inversas por omisión 378
 - djxlinkMssql, script 5
 - opciones de columna 308
 - opciones de correlación de usuarios 308
 - opciones de derivador 308
 - opciones de servidor 308
 - soporte de Unicode 11
- Microsoft SQL Server, orígenes de datos
 - acceder 102
 - variables de entorno 105
- minúsculas
 - uso en sentencias de SQL 15

N

- nodos
 - catalogación 55
 - DB2 55

O

- ODBC
 - características federadas soportadas 281
 - correlaciones de tipos de datos directas por omisión 365

- ODBC (*continuación*)
 - opciones de columna 313
 - opciones de correlación de usuarios 313
 - opciones de derivador 313
 - opciones de servidor 313
 - soporte de Unicode 11
 - variables de entorno 2
- ODBC, orígenes de datos
 - configurar el servidor federado 116
- opción CODEPAGE 11
- opción de correlaciones de funciones DISABLE
 - valores válidos 361
- opción de correlaciones de funciones REMOTE_NAME
 - valores válidos 361
- opción de servidor FOLD_ID
 - distinción entre mayúsculas y minúsculas 15
- opción de servidor FOLD_PW
 - distinción entre mayúsculas y minúsculas 15
- opciones de apodo
 - archivos con estructura de tabla 339
 - Base de datos DB2 287
 - BioRS 283
 - Script 325
 - servicios web 342
 - XML 350
- opciones de columna
 - archivos con estructura de tabla 339
 - Base de datos DB2 287
 - BioRS 283
 - Informix 295
 - JDBC 301
 - Microsoft SQL Server 308
 - ODBC 313
 - Oracle 319
 - Script 325
 - servicios web 342
 - Sybase 330
 - Teradata 335
 - XML 350
- opciones de correlación de usuarios
 - Base de datos DB2 287
 - BioRS 283
 - Informix 295
 - JDBC 301
 - Microsoft SQL Server 308
 - ODBC 313
 - Oracle 319
 - Script 325
 - servicios web 342
 - Sybase 330
 - Teradata 335
 - XML 350
- opciones de derivador
 - archivos con estructura de tabla 339
 - Base de datos DB2 287
 - BioRS 283
 - Excel 294
 - Informix 295
 - JDBC 301
 - Microsoft SQL Server 308
 - ODBC 313
 - Oracle 319

- opciones de derivador (*continuación*)
 - Script 325
 - servicios web 342
 - Sybase 330
 - Teradata 335
 - XML 350
- opciones de servidor
 - archivos con estructura de tabla 339
 - Base de datos DB2 287
 - BioRS 283
 - definir las opciones de servidor FOLD_ID y FOLD_PW 88
 - ejemplos de OLE DB 150
 - Excel 294
 - Informix 295
 - JDBC 301
 - Microsoft SQL Server 308
 - ODBC 313
 - Oracle 319
 - Script 325
 - servicios web 342
 - Sybase 330
 - Teradata 335
 - XML 350
- optimización
 - orígenes de datos de scripts 181
- optimización asíncrona
 - orígenes de datos soportados 281
- optimización de consultas
 - directrices para consultas BioRS 50
 - directrices para los archivos con estructura de tabla 209
 - XML 278
- Oracle
 - archivos de biblioteca 5
 - características federadas soportadas 281
 - correlaciones de tipos de datos directas por omisión 365
 - correlaciones de tipos de datos inversas por omisión 378
 - djxlinkOracle, script 5
 - opciones de columna 319
 - opciones de correlación de usuarios 319
 - opciones de derivador 319
 - opciones de servidor 319
- Oracle, orígenes de datos
 - conversión de página de códigos 156
 - variables de entorno 155
- orígenes de datos
 - archivos de mensajes 4
 - características soportadas 281
 - configuración 55
 - configuración del acceso a Informix 76
 - configurar 1
 - opciones 283
 - probar la conexión con el servidor Informix 87
 - registro 57
 - registro de derivadores Informix 81
 - registro de las definiciones de servidor Informix
 - línea de mandatos DB2 83

- orígenes de datos (*continuación*)
 - resolución de problemas de conexiones 61, 87, 99, 111, 124, 142, 164, 193, 221
 - tipos de servidor válidos 363
 - visión general de la configuración 12
- orígenes de datos BioRS 21
 - adición de BioRS a un sistema federado 22
 - apodos, registrar 34
 - comodines en consultas 38, 44
 - consultas de ejemplo 42, 44
 - correlaciones de usuario, crear 30
 - definición de servidor, registrar 27
 - derivador, registrar 24
 - ejemplo de definición de servidor, servidor proxy 28
 - ejemplos de sentencia CREATE SERVER 28
 - ejemplos de sentencias CREATE NICKNAME 35
 - elemento AllText 44
 - estadísticas
 - cardinalidad de apodos, actualizar 52
 - cardinalidad de columna `_ID_`, actualizar 54
 - cardinalidad de columnas, actualizar 53
 - estadísticas de cardinalidad, apodos 52
 - estadísticas de cardinalidad, determinación 51
 - funciones personalizadas, descripciones 24
 - funciones personalizadas, registrar 23
 - funciones personalizadas, visión general 38
 - nombres de biblioteca del derivador 25
 - operaciones de igualdad, consultas 41
 - programa de utilidad `admin_find` 51
 - programa de utilidad `www_find.cgi` 51
 - rendimiento del derivador, optimizar 50
 - requisitos previos 22
 - servidor proxy
 - ejemplo de definición de servidor BioRS 28
 - sintaxis de la sentencia CREATE FUNCTION 23
 - sintaxis de la sentencia CREATE NICKNAME 34
 - sintaxis de la sentencia CREATE SERVER 27
 - sintaxis de la sentencia CREATE USER MAPPING 30
 - sintaxis de la sentencia CREATE WRAPPER 24
 - utilizar predicados de unión, consultas 42
 - vía de acceso al directorio predeterminado 25
- orígenes de datos DB2
 - apodos 62
 - archivos de biblioteca 57
 - configuración 55
 - correlaciones de usuario 60
 - creación de apodos 62
 - prueba de las conexiones 61
 - registro 57
 - registro de servidores 58
 - sentencia CREATE SERVER 58
 - vía de acceso al directorio por omisión 57
- orígenes de datos DB2 Database para Linux, UNIX y Windows
 - correlaciones de tipos de datos directas por omisión 365
 - correlaciones de tipos de datos inversas por omisión 379
- orígenes de datos DB2 para System i
 - correlaciones de tipos de datos directas por omisión 366
 - correlaciones de tipos de datos inversas por omisión 380
- orígenes de datos DB2 para VM y VSE
 - correlaciones de tipos de datos directas por omisión 367
 - correlaciones de tipos de datos inversas por omisión 381
- orígenes de datos DB2 para z/OS
 - correlaciones de tipos de datos directas por omisión 368
 - correlaciones de tipos de datos inversas por omisión 382
- orígenes de datos de archivos con estructura de tabla 197
 - adición a un sistema federado 199
 - apodos, registro 202
 - autorizaciones para el acceso de archivo 208
 - definición de servidor, registro 201
 - derivador, registro 200
 - ejemplo de sentencia CREATE WRAPPER 200
 - ejemplos de sentencia CREATE NICKNAME 203
 - nombres de biblioteca de derivador 201
 - optimización de consultas 209
 - requisitos previos 199
 - sintaxis de sentencia CREATE NICKNAME 202
 - sintaxis de sentencia CREATE SERVER 201
 - sintaxis de sentencia CREATE WRAPPER 200
 - vía de acceso de directorios predeterminada 201
- orígenes de datos de scripts
 - consultas de ejemplo 180
 - definición de servidor 175
 - ejemplo de sentencia CREATE NICKNAME 176
 - nombres de biblioteca de derivador 174
 - opciones de apodo 177
 - optimización del rendimiento 181
 - registro de apodos 176
- orígenes de datos de scripts (*continuación*)
 - sentencia CREATE SERVER 175
- orígenes de datos de servicios Web
 - registro de apodos 237
 - registro de apodos (línea de mandatos de DB2) 237
- orígenes de datos Excel 63
 - adición de Excel a un sistema federado 68
 - apodos, registrar 71
 - autorizaciones para el acceso a archivos 76
 - caso práctico de usuario de ejemplo 74
 - consultas de ejemplo 73
 - definición de servidor, registrar 70
 - derivador, registrar 69
 - nombres de biblioteca del derivador 70
 - requisitos previos 68
 - sentencia CREATE NICKNAME, ejemplos 72
 - sentencia CREATE SERVER, ejemplos 71
 - sentencia CREATE WRAPPER, ejemplos 69
 - vía de acceso al directorio por omisión 70
- orígenes de datos IBM InfoSphere Classic Federation Server for z/OS
 - configurar acceso 134
- orígenes de datos Informix
 - archivos de biblioteca 4
 - correlaciones de tipos de datos directas por omisión 368, 382
- Orígenes de datos Informix variables de entorno 2
- orígenes de datos JDBC
 - apodos 99
 - archivos de biblioteca del derivador 93
 - configurar acceso 90
 - correlación de tipos de datos directa por omisión Unicode 387
 - correlación de tipos de datos inversa por omisión Unicode 388
 - correlaciones de tipos de datos directas por omisión 369
 - correlaciones de tipos de datos inversas por omisión 383
 - definir correlaciones de usuario 97
 - ejemplos de CREATE NICKNAME 100
 - ejemplos de CREATE SERVER 95
 - ejemplos de CREATE USER MAPPING 97
 - ejemplos de CREATE WRAPPER 94
 - probar el acceso 98
 - registro de definiciones de servidor 92, 94
 - variables de entorno 2
- orígenes de datos Microsoft SQL
 - definir correlaciones de usuario 109
- orígenes de datos Microsoft SQL Server
 - apodos 112
 - archivos de biblioteca 4

- orígenes de datos Microsoft SQL Server *(continuación)*
 - archivos de biblioteca del derivador 106
 - configurar acceso 101
 - configurar las variables de entorno 104
 - configurar servidores federados 101
 - correlación de tipos de datos inversa por omisión Unicode 389
 - correlaciones de tipos de datos directas por omisión 371, 384
 - ejemplo de CREATE NICKNAME 113
 - ejemplos de CREATE SERVER 108
 - ejemplos de CREATE USER MAPPING 110
 - probar conexiones a Microsoft SQL Server 111
 - registrar 105
 - registro de definiciones de servidor 107
 - resolución de problemas de configuración 113
 - variables de entorno 2
- orígenes de datos NET8
 - correlación de tipos de datos inversa por omisión Unicode 390
 - correlaciones de tipos de datos directas por omisión Unicode 389
- orígenes de datos no relacionales admitidos, tipos de datos 392
- orígenes de datos ODBC
 - acceso a libros Microsoft Excel 130
 - ajuste con el programa de utilidad de ajuste de ODBC 126, 144
 - archivos de biblioteca del derivador 119, 136
 - configuración del acceso a orígenes de datos ODBC 114
 - configurar el acceso a orígenes de datos IBM InfoSphere Classic Federation Server for z/OS 134
 - configurar servidores federados 116
 - correlación de tipos de datos inversa por omisión Unicode 390
 - correlaciones de tipos de datos directas por omisión 372
 - correlaciones de tipos de datos directas por omisión Unicode 390
 - correlaciones de tipos de datos inversas por omisión 384
 - definir correlaciones de usuario 123, 140
 - ejemplos de CREATE NICKNAME 126, 144
 - ejemplos de CREATE SERVER 121, 139
 - ejemplos de CREATE USER MAPPING 123, 141
 - ejemplos de CREATE WRAPPER 119, 137
 - permisos de acceso 132
 - probar el acceso 124, 142
 - registro 118, 135
 - registro de apodos 125, 143
- orígenes de datos ODBC *(continuación)*
 - registro de definiciones de servidor 120, 138
 - tablas de prueba para el programa de utilidad de ajuste de ODBC 129, 147
- orígenes de datos OLE DB
 - configuración del acceso 148
 - CREATE USER MAPPING, ejemplos 152
 - definición de correlaciones de usuario 151
 - ejemplos de CREATE SERVER 150
 - nombres de archivos de biblioteca 150
 - opciones de servidor 150
 - registro 149
 - registro de las definiciones de servidor 150
- orígenes de datos Oracle
 - acceder 153
 - adición a sistemas federados 153
 - archivos de biblioteca 4
 - archivos de biblioteca del derivador 159
 - configuración de conectividad 157
 - CREATE USER MAPPING, ejemplos 162
 - definición de correlaciones de usuario 162
 - ejemplos de CREATE NICKNAME 165
 - ejemplos de CREATE SERVER 161
 - establecimiento de las variables de entorno 153
 - registro 158
 - registro de apodos 164
 - registro de las definiciones de servidor 160
 - resolución de problemas de conectividad 163
 - variables de entorno 2
- orígenes de datos Oracle NET8
 - correlaciones de tipos de datos directas por omisión 373
 - correlaciones de tipos de datos inversas por omisión 385
- orígenes de datos Sybase
 - archivos de biblioteca 4
 - configuración de las variables de entorno 183
 - configuración del acceso 181
 - configuración del archivo de configuración del cliente 187
 - correlación de tipos de datos inversa por omisión Unicode 391
 - correlaciones de tipos de datos directas por omisión 374
 - correlaciones de tipos de datos directas por omisión Unicode 391
 - correlaciones de tipos de datos inversas por omisión 386
 - creación de correlaciones de usuario 192
 - CREATE USER MAPPING, ejemplos 192
- orígenes de datos Sybase *(continuación)*
 - ejemplos de CREATE NICKNAME 195
 - ejemplos de CREATE SERVER 190
 - nombres de archivos de biblioteca de derivador 188
 - prueba de conexiones con el servidor Sybase 193
 - registro 188
 - registro de apodos 194
 - registro de las definiciones de servidor 189
 - resolución de problemas de configuraciones de derivador 195, 196
 - sentencia CREATE USER MAPPING 192
 - variables de entorno 2
 - vía de acceso de directorios predeterminada de derivador 188
- orígenes de datos Teradata
 - ajuste 225
 - archivos de biblioteca 4
 - comprobación de enlaces de tiempo de ejecución 210
 - configuración del acceso 209
 - correlaciones de tipos de datos directas por omisión 376, 387
 - CREATE USER MAPPING, ejemplos 220
 - definición de correlaciones de usuario 219
 - establecimiento de las variables de entorno 211
 - inhabilitar el registro cronológico de acceso 225
 - prueba de la conexión 210
 - prueba de la conexión con el servidor federado 220
 - prueba de la configuración del cliente 210
 - registro 215
 - registro de apodos 222
 - registro de las definiciones de servidor 217
 - resolución de problemas de apodos 225
 - resolución de problemas de conectividad 220
 - resolución de problemas de juegos de caracteres 215, 225
 - resolución de problemas de la sentencia CREATE NICKNAME 224
 - variables de entorno 2
- orígenes de datos XML 255
 - adición a un sistema federado 258
 - apodos, registro 271
 - consultas, ejemplos 279
 - definición de servidor, registro 262
 - derivador, registro 259
 - ejemplos de sentencia CREATE NICKNAME 271
 - nombres de biblioteca de derivador 260
 - optimización de consultas, vistas federadas 276

orígenes de datos XML (*continuación*)
requisitos previos 258
vía de acceso de directorios
predeterminada 260

P

páginas de códigos 9, 11, 12
base de datos federada 7
planificación
acceder a datos Excel 65
políticas de licencia
Federación 6
problemas de conexión
orígenes de datos Microsoft SQL
Server 113
procedimientos federados
orígenes de datos soportados 281
procesador de línea de mandatos (CLP)
iniciar 1
programa de utilidad de ajuste de ODBC
ajuste de derivadores de ODBC 126,
144
crear tablas de prueba 129, 147
ejemplos del mandato
db2fedsrvrcfg 128, 146
sintaxis del mandato
db2fedsrvrcfg 128, 146
Proxy HTTP
orígenes de datos soportados 281
proxy SOCKS
orígenes de datos soportados 281
puntos de rescate de aplicación
orígenes de datos soportados 281

R

rastros ODBC 113
recurso de actualización de estadísticas
de apodo
orígenes de datos soportados 281
registro cronológico de errores
inhabilitar el registro cronológico de
acceso de Teradata 225
Registro especial USER de DB2
ejemplos para los orígenes de datos
OLE DB 152
repositorio de correlación de usuarios
orígenes de datos soportados 281
resolución de problemas
acceso a orígenes de datos 1
conexiones 61, 87, 99, 111, 124, 142,
164, 193, 221
configuración del servidor
federado 1
instalación de Federation 4, 5
servidor federado 4
restricciones
db2dj.ini, archivo 4

S

Script
características federadas
soportadas 281
opciones de apodo 325

Script (*continuación*)
opciones de columna 325
opciones de correlación de
usuarios 325
opciones de derivador 325
opciones de servidor 325
scripts
configuración como orígenes de
datos 166
secuencias de clasificación
base de datos federada 7
seguridad
servicios web 236
sentencia CREATE FUNCTION
sintaxis de BioRS 23
sentencia CREATE NICKNAME
ejemplos de archivos con estructura
de tabla 203
ejemplos de BioRS 35
ejemplos XML 271
Excel, ejemplos 72
Excel, sintaxis 71
Informix 89
orígenes de datos de scripts 176
orígenes de datos de servicios
Web 237
resolución de problemas para orígenes
de datos Teradata 224
sintaxis de archivos con estructura de
tabla 202
sintaxis de BioRS 34
Teradata 223
sentencia CREATE SERVER
ejemplos de BioRS 28
ejemplos para ODBC 121, 139
Excel, ejemplo 71
Excel, sintaxis 70
Informix 83
orígenes de datos de la familia
DB2 58
orígenes de datos de scripts 175
orígenes de datos Sybase 190
sintaxis de archivos con estructura de
tabla 201
sintaxis de BioRS 27
sintaxis XML 262
Teradata 218
sentencia CREATE USER MAPPING
orígenes de datos Sybase 192
sintaxis de BioRS 30
Sentencia CREATE USER MAPPING
ejemplos de BioRS 32
Informix 86
sentencia CREATE WRAPPER
ejemplo de archivos con estructura de
tabla 200
Excel, ejemplo 69
registro de los orígenes de datos OLE
DB 149
sintaxis de archivos con estructura de
tabla 200
sintaxis de BioRS 24
sintaxis XML 259
sentencia UPDATE
cardinalidad de apodos de BioRS,
ejemplo 52

sentencia UPDATE (*continuación*)
cardinalidad de columnas de BioRS,
ejemplo 53
cardinalidad de la columna _ID_ de
BioRS, ejemplo 54
ejemplos
sentencia UPDATE, BioRS 52
servicios web
acceso a cabeceras SOAP 245
apodos 226, 239
archivos de biblioteca del
derivador 234
características federadas
soportadas 281
consultas de ejemplo 252
CREATE SERVER, ejemplo de
sentencia 235
definiciones de servidor 234
ejemplos 249
opciones de apodo 342
opciones de columna 342
opciones de correlación de
usuarios 342
opciones de derivador 342
opciones de servidor 342
seguridad 236
servidor proxy, ejemplo de definición
de servidor 235
tipos de datos admitidos 392
vistas federadas 248
servicios web, orígenes de datos
registrar apodos (Centro de control de
DB2) 238
servidor federado
bases de datos 7
configuración 1
configuración del acceso a
informix 76
configurar 1
mandatos 1
resolución de problemas 1
visión general de la configuración 12
servidor Informix
probar la conexión 87
servidor Oracle
prueba de la conexión 163
servidor proxy
BioRS, ejemplo de derivador 26
servicios web, ejemplo de definición
de servidor 235
servidores
probar conexiones a Microsoft SQL
Server 111
prueba de conexiones con servidores
Sybase 193
servidores federados
acceder a orígenes de datos de
Microsoft SQL Server sources 102
catalogación de bases de datos 56
configuración de orígenes de
datos 55
configuración del acceso a orígenes de
datos Microsoft SQL Server 101
configuración del acceso a orígenes de
datos Sybase 181
configuración múltiple 18

servidores federados (*continuación*)
 configurar el acceso a Microsoft SQL Server 101
 configurar para JDBC 91
 configurar para ODBC 116
 probar el acceso a servidores de orígenes de datos JDBC 98
 probar el acceso a servidores de orígenes de datos ODBC 124, 142
 prueba de la conexión con el servidor Teradata 210

servidores proxy
 acceder a orígenes de datos XML 265

sesiones de paso a través
 orígenes de datos soportados 281

signo de comilla
 uso en sentencias de SQL 15

sinónimos de Informix
 registro de apodos 89

soporte de idioma nacional (NLS)
 base de datos federada 7

SQL10013N
 resolución de problemas 195

SQL1097N
 resolución de problemas 196

SQL1822N
 resolución de problemas 196

Sybase
 archivos de biblioteca 5
 características federadas soportadas 281
 correlaciones de tipos de datos directas por omisión 365
 correlaciones de tipos de datos inversas por omisión 378
 djsxlinkSybase, script 5
 opciones de columna 330
 opciones de correlación de usuarios 330
 opciones de derivador 330
 opciones de servidor 330

Sybase, orígenes de datos
 configurar el archivo de configuración del cliente 186
 variables de entorno 184

T

tablas de consulta materializadas
 orígenes de datos soportados 281

tablas de Informix
 registro de apodos 89

tablas de memoria caché
 orígenes de datos soportados 281

Teradata
 archivos de biblioteca 5
 características federadas soportadas 281
 correlaciones de tipos de datos directas por omisión 365
 correlaciones de tipos de datos inversas por omisión 378
 djsxlinkTeradata, script 5
 ejemplo de sentencia CREATE NICKNAME 223
 nombres de biblioteca de derivador 216

Teradata (*continuación*)
 opciones de columna 335
 opciones de correlación de usuarios 335
 opciones de derivador 335
 opciones de servidor 335
 sentencia CREATE SERVER 218

Teradata, orígenes de datos
 apodos 223
 variables de entorno 212

tipos de datos
 para orígenes de datos no relacionales 392

tipos de servidor
 tipos federados válidos 363

tolerancia de errores
 orígenes de datos soportados 281

U

ubicación por omisión 2

Unicode 9, 11, 12
 orígenes de datos soportados 281

V

variable de registro DB2_DJ_INI 2

variables de entorno 2
 configuración de las variables de entorno de Sybase 183
 configurar variables de Microsoft SQL Server 104
 derivadores de Informix 78
 establecer 2
 establecimiento de las variables de entorno de Teradata 211
 establecimiento de las variables de Oracle 153
 Informix 79
 Microsoft SQL Server, orígenes de datos 105
 necesarias 2
 Oracle, orígenes de datos 155
 orígenes de datos 2
 resolución de problemas de la variable de entorno Sybase 196
 restricciones 4
 Sybase, orígenes de datos 184
 Teradata, orígenes de datos 212

ventana Resultado de las acciones 18

vistas de catálogo SYSCAT 357
 vistas de catálogo SYSSTAT 357

vistas de Informix
 registro de apodos 89

vistas federadas
 servicios web 248

X

XML
 opciones de apodo 350
 opciones de columna 350
 opciones de correlación de usuarios 350
 opciones de derivador 350
 opciones de servidor 350

XML (*continuación*)
 tipos de datos admitidos 392

XML, orígenes de datos
 apodos 268
 apodos, asociaciones de datos 266
 CREATE SERVER, ejemplos de la sentencia 263
 CREATE VIEW, ejemplos de sentencia 276
 CREATE WRAPPER, sentencia 260
 espacios de nombres 269
 expresiones XPath 266
 opciones de apodo, optimización de consultas 268
 opciones de apodos
 analizar documentos XML 266
 espacios de nombres 269
 optimización de consultas 268
 optimización de consultas, recurso de modelos de costes 278
 optimización de las consultas, visión general 275
 recurso de modelo de coste 268
 servidor proxy, ejemplos de definición de servidor 263
 servidor proxy, ejemplos de derivador 260
 servidores proxy 265
 tuplas 266
 visión general 255
 visión general, apodos 266

XPath
 utilizar con apodos 247



Impreso en España

SC11-3955-02



Spine information:

IBM InfoSphere Federation Server

Versión 9.7

Guía de configuración para orígenes de datos federados

