

IBM InfoSphere Information Server
Versión 11 Release 3

*Guía de conectividad para acceder a
orígenes de datos JDBC*



IBM InfoSphere Information Server
Versión 11 Release 3

*Guía de conectividad para acceder a
orígenes de datos JDBC*



Nota

Antes de utilizar esta información y el producto al que da soporte, lea la información del apartado "Avisos y marcas registradas" en la página 59.

Contenido

Orígenes de datos JDBC	1
Configurar el archivo de configuración de controlador	1
Diseñar trabajos con etapas JDBC Connector.	2
Importar metadatos JDBC	3
Definir un trabajo que incluya una etapa JDBC Connector	4
Conectarse a un origen de datos JDBC.	4
Leer datos utilizando una etapa JDBC Connector	5
Grabar datos utilizando la etapa JDBC Connector	8
Buscar datos en un origen de datos JDBC	12
Referencia	16
Propiedades para el conector JDBC	16
Propagación de columnas en tiempo de ejecución	24
Derivaciones de columna	26
Condiciones de rechazo	28
Lecturas particionadas.	29
Modalidades de grabación soportadas	32
Correlaciones de tipos de datos.	34
Optimización del rendimiento	39
Resolución de problemas de la etapa JDBC Connector	44
No se puede acceder al archivo de configuración	44
CREATE TABLE no se ha ejecutado satisfactoriamente	45
El trabajo falla e informa de errores de recorte de datos o de conversión de juego de caracteres	46

El trabajo falla y se han detectado columnas no utilizadas	47
El conector no ha detectado e informado de todas las discrepancias de esquema	49
No se ha encontrado un controlador adecuado al utilizar la etapa JDBC Connector	50

Accesibilidad de los productos 51

Acceso a la documentación del producto 53

Cómo aportar comentarios sobre la documentación del producto 55

Enlaces a sitios web que no son de IBM 57

Avisos y marcas registradas 59

Cómo ponerse en contacto con IBM 65

Índice 67

Orígenes de datos JDBC

Utilice el conector JDBC para conectarse a orígenes de datos JDBC soportados y realizar operaciones de acceso a datos y de importación de metadatos en ellos.

Utilice el conector JDBC para realizar las siguientes operaciones:

- Leer datos de orígenes de datos JDBC o grabar datos en orígenes de datos JDBC.
- Importar metadatos de orígenes de datos JDBC mediante InfoSphere Metadata Asset Manager (IMAM).

Referencia relacionada:

“Resolución de problemas de la etapa JDBC Connector” en la página 44
Hay varios errores comunes que son específicos de la etapa JDBC Connector.

Configurar el archivo de configuración de controlador

Antes de poder utilizar el conector JDBC, debe configurar el archivo de configuración de controlador. El conector utiliza este archivo para obtener información sobre los controladores JDBC disponibles en el sistema.

Procedimiento

1. Cree un archivo de configuración de controlador llamado `isjdbc.config` con el permiso de lectura habilitado para todos los usuarios. El nombre del archivo de configuración de controlador es sensible a las mayúsculas y minúsculas.
2. Abra un editor de texto e incluya las dos líneas siguientes que especifican la vía de acceso de clases y las clases Java de controlador:

```
CLASSPATH=vía_acceso_clases_controlador  
CLASS_NAMES=nombres_clases_controlador
```

El valor `vía_acceso_clases_controlador` es la vía de acceso de clases Java acumulativa para los controladores JDBC que va a utilizar a través del conector. El valor se especifica como una lista separada por signos de punto y coma de vías de acceso de directorio, vías de acceso de archivos `.jar` y vías de acceso de archivos `.zip` totalmente calificadas. Por ejemplo, si el controlador que tiene pensado utilizar se implementa como un archivo `.jar`, incluya la vía de acceso completa a ese archivo `.jar` en el valor `vía_acceso_clases_controlador`. Para obtener más información sobre los requisitos de vía de acceso de clases para el controlador, consulte la documentación del controlador.

El valor `nombres_clases_controlador` es una lista separada por signos de punto y coma de nombres de clases de controlador totalmente calificados implementados por los controladores JDBC que tiene previsto utilizar a través del conector. Las clases de controlador son las clases Java en los controladores que implementan la interfaz de API JDBC `java.sql.Driver`. Para obtener más información sobre la clase de controlador implementada por el controlador, consulte la documentación del controlador JDBC. Tenga en cuenta que no es necesario proporcionar esta información para los controladores que se implementan como archivos JAR y están basados en JDBC 4.0 o en una especificación JDBC posterior ya que, en esos casos, el conector puede determinar automáticamente el nombre de la clase de controlador a partir del archivo JAR del controlador.

3. Guarde el archivo `isjdbc.config` en el host de la capa de motor de InfoSphere Information Server bajo el directorio `DIR_INICIO_IS/Server/DSEngine`, donde `DIR_INICIO_IS` es el directorio de inicio de InfoSphere Information Server. Por

ejemplo, el directorio de inicio puede ser C:\IBM\InformationServer en un sistema Windows o /opt/IBM/InformationServer en un sistema Linux o UNIX. Si la capa de motor de su instalación de InfoSphere Information Server consta de varios hosts, este archivo debe estar disponible en la misma ubicación en todos los hosts. Para hacer que este archivo esté disponible en la misma ubicación en todos los hosts, puede configurar el directorio DSEngine como un directorio de red compartido.

Ejemplo

Este ejemplo muestra cómo puede configurar el archivo de configuración de controlador.

Suponga lo siguiente:

- Desea utilizar el conector JDBC con los dos siguientes controladores JDBC: controlador JDBC 4.0, Controlador A y controlador JDBC 3.0, Controlador B.
- El Controlador A se implementa como el archivo /opt/productA/driverA.jar, y el Controlador B se implementa como el archivo /app/productB/driverBimpl.jar.
- Para el controlador A, determina que el nombre de la clase Java de controlador es com.example.A.Driver y, para el Controlador B, determina que el nombre de la clase Java de controlador es com.example.DriverB.

Para configurar el archivo de configuración de controlador, realice las siguientes acciones:

1. Cree el archivo isjdbc.config y especifique las dos líneas siguientes:

```
CLASSPATH=/opt/productA/driverA.jar;/app/productB/driverBimpl.jar  
CLASS_NAMES=com.example.A.Driver;com.example.DriverB
```

Nota: El valor com.example.A.Driver se puede omitir de la entrada CLASS_NAMES porque el Controlador A es un controlador JDBC 4.0 y, para este controlador, el conector puede recuperar automáticamente el nombre de clase de controlador del archivo JAR del controlador.

2. Guarde el archivo en el host de la capa de motor de InfoSphere Information Server bajo el directorio DIR_INICIO_IS/Server/DSEngine, donde DIR_INICIO_IS es el directorio de inicio de InfoSphere Information Server.

Después de realizar cambios en el archivo de configuración de controlador, no es necesario reiniciar el motor de DataStage, los agentes ISF ni WebSphere Application Server. El conector JDBC reconocerá los cambios realizados en este archivo la próxima vez que se utilice para acceder a orígenes de datos JDBC.

Diseñar trabajos con etapas JDBC Connector

Puede utilizar etapas JDBC Connector en los trabajos para leer datos de los orígenes de datos JDBC o grabar datos en orígenes de datos JDBC en los contextos de esos trabajos.

Antes de empezar

Configure el archivo de configuración de controlador.

Procedimiento

1. Defina un trabajo que incluya una etapa JDBC Connector.
2. Defina una conexión con un origen de datos JDBC.

3. Para configurar la etapa JDBC Connector como una etapa de origen para leer datos del origen de datos JDBC, realice los pasos siguientes:
 - a. Configure la etapa JDBC Connector como origen.
 - b. Configure definiciones de columna en el enlace de salida.
4. Para configurar la etapa JDBC Connector como una etapa de destino para grabar datos en el origen de datos JDBC, realice los pasos siguientes:
 - a. Configure definiciones de columna en el enlace de entrada.
 - b. Configure la etapa JDBC como destino.
 - c. Opcional: Cree enlaces de rechazo para gestionar datos de entrada con errores.
5. Para configurar la etapa JDBC Connector para buscar datos en un origen de datos JDBC, realice el paso siguiente:
 - a. Configure operaciones de búsqueda normal o configure operaciones de búsqueda dispersa.

Importar metadatos JDBC

Antes de diseñar trabajos que utilicen el conector JDBC para leer, grabar o buscar datos, puede utilizar InfoSphere Metadata Asset Manager para importar los metadatos que representan tablas y vistas en el origen de datos JDBC. Los metadatos importados se guardan en el repositorio de metadatos.

Acerca de esta tarea

Con el conector JDBC, puede importar metadatos sobre los siguientes tipos de activos:

- El sistema host que contiene el origen de datos JDBC.
- El origen de datos.
- Esquemas de base de datos.
- Tablas de base de datos, tablas del sistema y vistas. Todas las tablas importadas se almacenan en el repositorio de metadatos como tablas de base de datos.
- Columnas de base de datos.

Cuando se importan metadatos JDBC utilizando InfoSphere Metadata Asset Manager,

- Cuando se selecciona **Incluir objeto del sistema**, el conector JDBC solicita que el controlador JDBC incluya los objetos que tengan el atributo de tipo de tabla JDBC establecido en SYSTEM TABLE. El controlador JDBC y el origen de datos determinan qué objetos en el origen de datos cumplen este criterio.
- Cuando se especifica el sistema host del repositorio de metadatos y los objetos de base de datos a los que importar los metadatos, si no se especifica un valor para **Nombre de la base de datos**, el conector utiliza el valor que se encuentra entre el primer y el segundo signo de dos puntos en el URL de conexión JDBC especificado. Para impedir que el conector importe los metadatos bajo el objeto de base de datos predeterminado generado automáticamente, especifique el **Nombre de la base de datos** bajo el que desee importar los metadatos.

Procedimiento

Importe metadatos utilizando InfoSphere Metadata Asset Manager. Para obtener más información sobre cómo importar metadatos utilizando InfoSphere Metadata Asset Manager, consulte la documentación en línea del producto en el IBM

Knowledge Center o la publicación IBM® InfoSphere Information Server - Guía para la gestión de metadatos operativos.

Tareas relacionadas:

“Conectarse a un origen de datos JDBC”

Para acceder a orígenes de datos JDBC, debe definir una conexión utilizando las propiedades de la sección **Conexión** de la página Propiedades.

Referencia relacionada:

“Importación de metadatos” en la página 40

Cuando utilice IBM InfoSphere Metadata Asset Manager para importar metadatos, puede mejorar el rendimiento importando todo el esquema en lugar de objetos de datos individuales. Los métodos de importación disponibles dependen de la implementación del controlador JDBC y del número de objetos que se van a importar.

Definir un trabajo que incluya una etapa JDBC Connector

Antes de poder leer, grabar o buscar datos en un origen de datos JDBC, debe crear un trabajo que incluya una etapa JDBC Connector. Después, debe añadir las etapas adicionales que sean necesarias y crear los enlaces necesarios.

Procedimiento

1. En el Cliente del Diseñador de InfoSphere DataStage and QualityStage, seleccione **Archivo > Nuevo** en el menú.
2. En la ventana **Nuevo**, seleccione el icono **Trabajo paralelo** y luego pulse **Aceptar**.
3. Añada la etapa JDBC Connector al trabajo:
 - a. En la paleta, seleccione la categoría **Base de datos**.
 - b. Localice **JDBC Connector** en la lista de tipos de etapas disponibles.
 - c. Arrastre el icono de la etapa JDBC Connector al lienzo.
 - d. Opcional: Cambie el nombre de la etapa JDBC Connector. Elija un nombre que indique la función de la etapa en el trabajo.
4. Cree los enlaces necesarios y las etapas adicionales para el trabajo:
 - Para un trabajo que lea datos de un origen de datos JDBC, cree la siguiente etapa del trabajo y luego cree un enlace de salida desde la etapa JDBC Connector a la siguiente etapa.
 - Para un trabajo que grabe datos en un origen de datos JDBC, cree uno o más enlaces desde otras etapas del trabajo a la etapa JDBC Connector. Si la etapa JDBC Connector tiene varios enlaces de entrada, puede utilizar la página Orden de los enlaces del editor de etapas para especificar el orden lógico de los enlaces. Si desea gestionar los registros rechazados, añada una etapa para contener los registros rechazados y luego añada un enlace de rechazo desde el conector JDBC a esa etapa. Si la etapa tiene varios enlaces de entrada, puede tener varios enlaces de rechazo, y cada enlace de rechazo puede estar asociado a un enlace de entrada.
 - Para un trabajo que busque datos en un origen de datos JDBC, cree un trabajo que incluya una etapa Lookup y luego cree un enlace de referencia desde la etapa JDBC Connector a la etapa Lookup.
5. Guarde el trabajo.

Conectarse a un origen de datos JDBC

Para acceder a orígenes de datos JDBC, debe definir una conexión utilizando las propiedades de la sección **Conexión** de la página Propiedades.

Antes de empezar

- Cree el archivo de configuración de controlador `isjdbc.config` y especifique información sobre los controladores en este archivo.
- Defina un trabajo que incluya una etapa JDBC Connector.

Procedimiento

1. Para abrir el editor de etapas, en el lienzo de diseño de trabajos, efectúe una doble pulsación en el icono de la etapa **JDBC Connector**.
2. En la página Propiedades, especifique valores para las propiedades de conexión. En la propiedad **URL**, especifique la serie de URL para el origen de datos JDBC en el formato específico del controlador.
3. En las propiedades de conexión **Nombre de usuario** y **Contraseña**, especifique las credenciales del usuario que se va a autenticar y autorizar para la conexión. Tenga en cuenta que algunos controladores permiten incluir valores de nombre de usuario y contraseña directamente en la serie de conexión URL. Para obtener más información, consulte la documentación del controlador.
4. En la propiedad **Atributos**, especifique cualquier propiedad de conexión específica del controlador adicional para el controlador. Por ejemplo, si el controlador da soporte a conexiones SSL, proporcione las propiedades de conexión SSL aquí. Cada propiedad de conexión específica del controlador debe especificarse como una línea separada en el formato *nombre_propiedad=valor_propiedad*. Tenga en cuenta que algunos controladores permiten incluir propiedades de conexión específicas del controlador directamente en la serie de conexión URL. Para obtener más información, consulte la documentación del controlador.
5. Pulse **Aceptar** para guardar los detalles.

Tareas relacionadas:

“Configurar el archivo de configuración de controlador” en la página 1
Antes de poder utilizar el conector JDBC, debe configurar el archivo de configuración de controlador. El conector utiliza este archivo para obtener información sobre los controladores JDBC disponibles en el sistema.

Leer datos utilizando una etapa JDBC Connector

Puede configurar una etapa JDBC Connector para conectarse a un origen de datos JDBC y leer datos del mismo.

Antes de empezar

- Defina un trabajo que contenga una etapa JDBC Connector.
- Defina una conexión con un origen de datos JDBC.

Acerca de esta tarea

La siguiente figura muestra un ejemplo del uso de la etapa JDBC Connector para leer datos. En este ejemplo, la etapa JDBC Connector lee datos de un origen de datos JDBC, y la etapa Sequential File graba los datos en un archivo. Cuando se configura la etapa JDBC Connector para leer datos, se crea un enlace de salida, que se muestra en la figura siguiente transfiriendo filas desde la etapa JDBC Connector a la etapa Sequential File.



Figura 1. Ejemplo del uso de la etapa JDBC Connector para leer datos de un origen de datos

Configurar el conector JDBC como origen

Para configurar una etapa JDBC Connector para leer o buscar filas en una tabla o vista JDBC, debe especificar la tabla o vista de origen o definir una sentencia SELECT completa.

Procedimiento

1. En el lienzo de diseño de trabajos, efectúe una doble pulsación en la etapa JDBC Connector.
2. Pulse el separador **Salida**.
3. Pulse el separador **Propiedades** y, en la sección **Uso**, especifique los valores para la operación de lectura.
4. Utilice uno de estos métodos para especificar el origen de los datos:
 - Establezca **Generar SQL en tiempo de ejecución** en **Sí**, y luego especifique el nombre de la tabla o vista en la propiedad **Nombre de tabla**. Utilice la sintaxis *nombre_tabla* o *propietario.nombre_tabla*, donde *propietario* es el propietario de la tabla. Si no especifica un propietario, el conector utiliza el esquema que pertenece al usuario que está conectado actualmente.
 - Establezca **Generar SQL en tiempo de ejecución** en **No** y luego especifique la sentencia SELECT en la propiedad **Sentencia Select**.
 - Establezca **Generar SQL en tiempo de ejecución** en **No** y luego especifique el nombre de archivo totalmente calificado del archivo que contiene la sentencia SQL en la propiedad **Sentencia Select**. Tenga en cuenta que este archivo debe residir en el host del motor. Si especifica un nombre de archivo, también debe establecer **Leer la sentencia de selección del archivo** en **Sí**.
5. Pulse **Aceptar** y luego guarde el trabajo.

Configurar definiciones de columna en un enlace

Las definiciones de columna, que se establecen en un enlace, especifican el formato de los registros de datos que el conector lee de una base de datos o graba en una base de datos.

Procedimiento

1. En el lienzo de diseño de trabajo, efectúe una doble pulsación en el icono de conector.

2. Utilice uno de los siguientes métodos para configurar las definiciones de columna:
 - Arrastre una definición de tabla desde la vista de repositorio al enlace en el lienzo de trabajo. A continuación, utilice los botones de flecha para mover las columnas entre las listas **Columnas disponibles** y **Columnas seleccionadas**.
 - En la página **Columnas**, pulse **Cargar** y seleccione una definición de tabla del repositorio de metadatos. A continuación, para elegir las columnas de la definición de tabla que desea aplicar al enlace, mueva las columnas de la lista **Columnas disponibles** a la lista **Columnas seleccionadas**.
3. Configure las propiedades para las columnas:
 - a. Pulse el botón derecho del ratón en la cuadrícula de columnas y seleccione **Propiedades** en el menú.
 - b. Seleccione las propiedades para mostrar, especifique el orden en el que mostrarlas y luego pulse **Aceptar**.
4. Opcional: Modifique las definiciones de columna. Puede cambiar los nombres de columna, los tipos de datos y otros atributos. Además, puede añadir, insertar o eliminar columnas.
5. Opcional: Guarde la nueva definición de tabla en el repositorio de metadatos:
 - a. En la página **Columnas**, pulse **Guardar** y luego pulse **Aceptar** para mostrar la vista de repositorio.
 - b. Navegue hasta una carpeta existente, o cree una nueva carpeta en la que guardar la definición de tabla.
 - c. Seleccione la carpeta y luego pulse **Guardar**.

Configurar el conector JDBC para lecturas particionadas

Si la etapa JDBC Connector está configurada para ejecutarse en varios nodos de proceso, cada uno de los nodos de proceso lee datos del origen de datos simultáneamente con otros nodos de proceso. Las particiones de registros de todos los nodos de proceso se combinan para producir el conjunto de datos de registro completo para el enlace de salida de la etapa.

Antes de empezar

- Añada la etapa JDBC Connector a un trabajo paralelo.
- Si el conector está configurado para buscar datos, asegúrese de que la propiedad **Tipo de búsqueda** está establecida en **Normal**.

Acerca de esta tarea

Cuando una etapa JDBC Connector está configurada para realizar lecturas particionadas, cada uno de los nodos de proceso de la etapa lee una parte de los datos del origen de datos y los registros recuperados por todos los nodos de proceso se combinan para producir el conjunto de resultados para el enlace de salida.

Procedimiento

1. En el lienzo de diseño de trabajos, efectúe una doble pulsación en la etapa JDBC Connector y luego pulse el separador **Etapa**.
2. En la página **Avanzado**, establezca la **Modalidad de ejecución** en **Paralela** o **Predeterminada (Paralela)** y luego pulse el separador **Salida**.
3. Defina la sentencia SELECT que el conector utiliza en tiempo de ejecución:
 - Establezca **Generar SQL** en **No** y luego especifique la sentencia SELECT en la propiedad **Sentencia Select**, o establezca la propiedad **Leer la sentencia**

- de selección del archivo en **Sí**, especifique el nombre del archivo en la propiedad **Sentencia Select** e incluya la sentencia **SELECT** en ese archivo.
4. Establezca **Habilitar lecturas particionadas** en **Sí**. Utilice los marcadores **[[número-nodo]]**, **[[número-nodo-base-uno]]** y **[[recuento-nodos]]** en la sentencia **SELECT** para controlar que partes de los datos son captados en los nodos de proceso individuales.
 5. Pulse **Aceptar** y luego guarde el trabajo.

Referencia relacionada:

“Lecturas particionadas” en la página 29

La etapa **JDBC Connector** se puede configurar para que se ejecute en varios nodos de proceso y para leer datos del origen de datos. En este escenario, cada uno de los nodos de proceso para la etapa recupera un conjunto (partición) de registros del origen de datos. Las particiones luego se combinan para producir el conjunto de resultados completo para el enlace de salida de la etapa.

Grabar datos utilizando la etapa **JDBC Connector**

Puede configurar una etapa **JDBC Connector** para conectarse a un origen de datos **JDBC** y grabar datos en él.

Antes de empezar

- Defina un trabajo que contenga una etapa **JDBC Connector**.
- Defina una conexión con un origen de datos **JDBC**.

Acerca de esta tarea

La siguiente figura muestra un ejemplo del uso de la etapa **JDBC Connector** para grabar datos. En este ejemplo, la etapa **Sequential File** lee datos de un archivo y luego la etapa **JDBC Connector** graba datos en el origen de datos **JDBC**.



Figura 2. Ejemplo del uso de la etapa **JDBC Connector** para grabar datos en un origen de datos

Referencia relacionada:

“Modalidades de grabación soportadas” en la página 32

Al configurar el conector **JDBC** como destino, puede utilizar la propiedad **Modalidad de grabación** para especificar la modalidad que desea utilizar para grabar filas en el origen de datos **JDBC**.

Configurar el conector JDBC como destino

Para configurar una etapa JDBC Connector para grabar filas en una tabla JDBC, debe especificar la tabla o vista de destino o definir las sentencias SQL.

Procedimiento

1. En el lienzo de diseño de trabajos, efectúe una doble pulsación en el icono de la etapa JDBC Connector.
2. Seleccione el enlace de entrada para editar.
3. Especifique la forma en que la etapa JDBC Connector graba datos en una tabla JDBC. La siguiente tabla muestra las maneras en que puede configurar el conector para grabar datos.

Tabla 1. Métodos para grabar datos en una tabla JDBC

Método	Procedimiento
Generar automáticamente el SQL en tiempo de ejecución	<ol style="list-style-type: none">1. Establezca Generar SQL en tiempo de ejecución en Sí.2. Establezca Modalidad de grabación en Insertar, Actualizar, Suprimir, Insertar y luego actualizar, Actualizar y luego insertar, Suprimir y luego insertar o Insertar sólo las filas nuevas.3. Especifique el nombre de la tabla de destino en el campo Nombre de tabla. Utilice la sintaxis <i>nombre_tabla</i> o <i>nombre_esquema.nombre_tabla</i>, donde <i>nombre_esquema</i> es el esquema propietario de la tabla. Cuando no se especifica <i>nombre_esquema</i>, el conector utiliza el esquema predeterminado del usuario conectado actualmente.
Especificar el SQL manualmente	<ol style="list-style-type: none">1. Establezca Generar SQL en tiempo de ejecución en No.2. Establezca Modalidad de grabación en Insertar, Actualizar, Suprimir, Insertar y luego actualizar, Actualizar y luego insertar, Insertar sólo las filas nuevas o Personalizada.3. Especifique sentencias SQL en los campos que correspondan a la modalidad de grabación que ha seleccionado.

Tabla 1. Métodos para grabar datos en una tabla JDBC (continuación)

Método	Procedimiento
Leer la sentencia SQL de un archivo	<ol style="list-style-type: none"> 1. Establezca Generar SQL en tiempo de ejecución en No. 2. Establezca Modalidad de grabación en Insertar, Actualizar, Suprimir, Insertar y luego actualizar, Actualizar y luego insertar, Suprimir y luego insertar, Insertar sólo las filas nuevas o Personalizada. 3. Especifique los nombres completos de los archivos que contienen las sentencias SQL en los campos que correspondan a la modalidad de grabación que ha seleccionado. 4. Establezca Leer la sentencia de inserción del archivo, Leer la sentencia de actualización del archivo, Leer la sentencia de supresión del archivo o Leer sentencias personalizadas del archivo en Sí.

4. Pulse **Aceptar** y luego guarde el trabajo.

Configurar definiciones de columna en un enlace

Las definiciones de columna, que se establecen en un enlace, especifican el formato de los registros de datos que el conector lee de una base de datos o graba en una base de datos.

Procedimiento

1. En el lienzo de diseño de trabajo, efectúe una doble pulsación en el icono de conector.
2. Utilice uno de los siguientes métodos para configurar las definiciones de columna:
 - Arrastre una definición de tabla desde la vista de repositorio al enlace en el lienzo de trabajo. A continuación, utilice los botones de flecha para mover las columnas entre las listas **Columnas disponibles** y **Columnas seleccionadas**.
 - En la página **Columnas**, pulse **Cargar** y seleccione una definición de tabla del repositorio de metadatos. A continuación, para elegir las columnas de la definición de tabla que desea aplicar al enlace, mueva las columnas de la lista **Columnas disponibles** a la lista **Columnas seleccionadas**.
3. Configure las propiedades para las columnas:
 - a. Pulse el botón derecho del ratón en la cuadrícula de columnas y seleccione **Propiedades** en el menú.
 - b. Seleccione las propiedades para mostrar, especifique el orden en el que mostrarlas y luego pulse **Aceptar**.
4. Opcional: Modifique las definiciones de columna. Puede cambiar los nombres de columna, los tipos de datos y otros atributos. Además, puede añadir, insertar o eliminar columnas.
5. Opcional: Guarde la nueva definición de tabla en el repositorio de metadatos:
 - a. En la página **Columnas**, pulse **Guardar** y luego pulse **Aceptar** para mostrar la vista de repositorio.

- b. Navegue hasta una carpeta existente, o cree una nueva carpeta en la que guardar la definición de tabla.
- c. Seleccione la carpeta y luego pulse **Guardar**.

Rechazar registros que contienen errores

Cuando una etapa JDBC Connector incluye un enlace de rechazo, los registros que cumplen los criterios especificados se dirigen automáticamente a la etapa de destino en el enlace de rechazo. El proceso continúa para los registros restantes.

Acerca de esta tarea

Cuando configure un enlace de rechazo, debe seleccionar una o más condiciones que controlen cuándo rechazar un registro y enviarlo a la etapa de destino que recibe los registros rechazados. También puede elegir incluir el código de error y el mensaje de error JDBC asociados en el registro rechazado. Si no define un enlace de rechazo o si define un enlace de rechazo pero un registro anómalo no coincide con ninguna de las condiciones de rechazo especificadas, el conector informa de un error y detiene el trabajo.

Después de ejecutar el trabajo, puede evaluar los registros rechazados y ajustar el trabajo y los datos en consecuencia.

Procedimiento

1. En el lienzo de diseño de trabajos, añada y configure una etapa de destino para recibir los registros rechazados.
2. Pulse el botón derecho del ratón en el icono de la etapa **JDBC Connector** y arrástrelo para crear enlace desde el conector JDBC hasta la etapa.
3. Si el enlace es el primer enlace para la etapa JDBC Connector, pulse el botón derecho del ratón en el enlace y seleccione **Convertir a rechazo**. Si la etapa JDBC Connector ya tiene un enlace de entrada, el nuevo enlace aparece automáticamente como un enlace de rechazo.
4. Efectúe una doble pulsación en el conector para abrir el editor de etapas.
5. En la página Salida, seleccione el enlace a la etapa de destino para registros rechazados en la lista **Nombre de salida (etapa en sentido descendente)**.
6. Pulse el separador **Rechazar**.
7. En la lista **Rechazar filas basándose en las condiciones seleccionadas**, seleccione las condiciones en las que desea que los registros del enlace de entrada se envíen al enlace de rechazo.
8. Utilice uno de los métodos de la tabla siguiente para especificar cuándo se debe detener un trabajo debido a que hay demasiadas filas rechazadas.

Método	Procedimiento
Detener un trabajo basándose en el porcentaje de filas que fallan.	<ol style="list-style-type: none"> 1. En la lista Terminar anormalmente si, seleccione Porcentaje. 2. En el campo Terminar anormalmente después de (%), especifique el porcentaje de filas rechazadas que hace que el trabajo se detenga. 3. En el campo Iniciar recuento después de (filas), especifique el número de filas de entrada que se procesarán antes de calcular el porcentaje de las filas rechazadas.

Método	Procedimiento
Detener un trabajo basándose en el número de filas que fallan.	<ol style="list-style-type: none"> 1. En la lista Terminar anormalmente si, seleccione Filas. 2. En el campo Terminar anormalmente después de (filas), especifique el número máximo de filas rechazadas que se permitirán antes de que se detenga el trabajo.

9. Opcional: En la lista **Añadir a fila de rechazo**, seleccione **ERRORCODE**, **ERRORTTEXT** o ambos. Los campos **ERRORCODE** y **ERRORTTEXT** de cada registro rechazado se establecen en los valores que dependen de la condición de rechazo bajo la que se ha rechazado el registro:

Condición de rechazo	Valor de ERRORCODE	Valor de ERRORTTEXT
Error de SQL	Establecido en el código de error proporcionado por el controlador.	Establecido en un mensaje de texto, que incluye información sobre el error proporcionada por el controlador.
Fila no insertada	1	Establecido en un mensaje de texto que indica que se insertaron cero filas para el registro actual.
Fila no actualizada	2	Establecido en un mensaje de texto que indica que se insertaron cero filas para el registro actual.
Fila no suprimida	3	Establecido en un mensaje de texto que indica que se insertaron cero filas para el registro actual.

10. Pulse **Aceptar** y luego guarde el trabajo.

Referencia relacionada:

“Condiciones de rechazo” en la página 28

Cuando utilice el conector JDBC para grabar datos en un origen de datos JDBC, puede añadir un enlace de rechazo a la etapa JDBC Connector y enviar los registros rechazados a la etapa conectada al otro extremo del enlace. Las condiciones de rechazo determinan cuándo se rechaza un registro.

“Modalidades de grabación soportadas” en la página 32

Al configurar el conector JDBC como destino, puede utilizar la propiedad **Modalidad de grabación** para especificar la modalidad que desea utilizar para grabar filas en el origen de datos JDBC.

Buscar datos en un origen de datos JDBC

Puede configurar el conector para realizar una búsqueda normal o una búsqueda dispersa en un origen de datos JDBC.

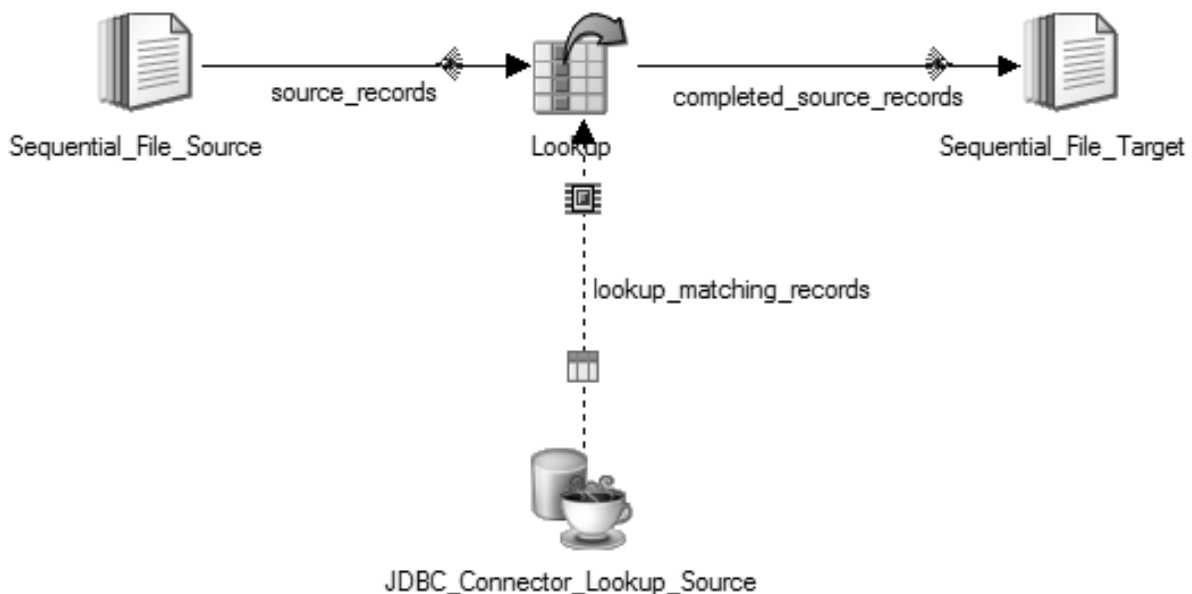
Antes de empezar

- Defina un trabajo que contenga la etapa JDBC Connector.
- Defina una conexión con un origen de datos JDBC.

Acerca de esta tarea

En la siguiente figura, una etapa Lookup extrae datos de un origen de datos JDBC basándose en los valores de parámetros de entrada que la etapa Lookup proporciona. Aunque el enlace de referencia parece ir desde la etapa JDBC Connector a la etapa Lookup, el enlace transfiere datos a y desde la etapa JDBC Connector. Los parámetros de entrada se transfieren desde el enlace de entrada de la etapa Lookup al enlace de referencia, y los valores de salida que la etapa JDBC Connector proporciona se transfieren desde la etapa JDBC Connector a la etapa Lookup. Los valores de salida se dirigen a las columnas en el enlace de salida de la etapa Lookup de acuerdo con las correlaciones de columnas definidas para la etapa Lookup.

Figura 3. Ejemplo del uso de la etapa JDBC Connector con una etapa Lookup.



Configurar operaciones de búsqueda normal

Puede configurar una etapa JDBC Connector para recuperar un conjunto de registros de un origen de datos JDBC y proporcionarlos a una etapa Lookup del mismo trabajo, que luego realiza una operación de búsqueda normal (en memoria) en dichos registros.

Antes de empezar

- Para especificar el formato de los registros que la etapa JDBC Connector lee de un origen de datos JDBC, configure definiciones de columna en un enlace.
- Configure la etapa JDBC Connector como origen para los datos de referencia.

Acerca de esta tarea

En una búsqueda normal, el conector ejecuta la sentencia SELECT especificada solamente una vez; por lo tanto, la sentencia SELECT no puede incluir ningún parámetro de entrada. La etapa Lookup busca en los datos del conjunto de

resultados que proporciona el conector y busca coincidencias para los conjuntos de parámetros que llegan en forma de registros en el enlace de entrada a la etapa Lookup. Una búsqueda normal también se denomina búsqueda en memoria porque la búsqueda se realiza en los datos almacenados en la memoria caché.

Normalmente, se utiliza una búsqueda normal cuando la tabla de destino es lo suficientemente pequeña para que todas las filas de la tabla puedan caber en la memoria.

Procedimiento

1. Añada una etapa Lookup al lienzo de diseño de trabajos y luego cree un enlace de referencia de la etapa JDBC Connector a la etapa Lookup.
2. Efectúe una doble pulsación en la etapa JDBC Connector.
3. En la lista **Tipo de búsqueda**, seleccione **Normal**.
4. Para guardar los cambios, pulse **Aceptar**.
5. Efectúe una doble pulsación en la etapa Lookup.
6. Para especificar las columnas de clave, arrastre las columnas necesarias del enlace de entrada al enlace de referencia. Las columnas del enlace de entrada contienen valores que se utilizan como valores de entrada para la operación de búsqueda.
7. Correlacione las columnas de enlace de entrada y de enlace de referencia con las columnas de enlace de salida y especifique condiciones para una anomalía en la búsqueda:
 - a. Arrastre o copie las columnas del enlace de entrada y del enlace de referencia al enlace de salida.
 - b. Para definir condiciones para una anomalía en la búsqueda, pulse el icono **Restricciones** en el menú.
 - c. En la columna **Anomalía en la búsqueda**, seleccione un valor y luego pulse **Aceptar**. Si selecciona **Rechazar**, debe tener un enlace de rechazo procedente de la etapa Lookup y una etapa de destino en la configuración del trabajo para capturar los registros rechazados.
 - d. Pulse **Aceptar**.
8. Guarde, compile y ejecute el trabajo.

Tareas relacionadas:

“Configurar definiciones de columna en un enlace” en la página 10

Las definiciones de columna, que se establecen en un enlace, especifican el formato de los registros de datos que el conector lee de una base de datos o graba en una base de datos.

“Configurar el conector JDBC como origen” en la página 6

Para configurar una etapa JDBC Connector para leer o buscar filas en una tabla o vista JDBC, debe especificar la tabla o vista de origen o definir una sentencia SELECT completa.

Configurar operaciones de búsqueda dispersa

Puede configurar una etapa JDBC Connector para realizar una operación de búsqueda dispersa (directa) en un origen de datos JDBC.

Antes de empezar

- Para especificar el formato de los registros que la etapa JDBC Connector lee de un origen de datos JDBC, configure definiciones de columna en un enlace.
- Configure la etapa JDBC Connector como origen para los datos de referencia.

Acerca de esta tarea

En una búsqueda dispersa, el conector ejecuta la sentencia SELECT especificada una vez para cada conjunto de parámetros que llega en forma de registro en el enlace de entrada a la etapa Lookup. Los parámetros de entrada especificados en la sentencia deben tener columnas correspondientes definidas en el enlace de referencia. Cada registro de entrada incluye un conjunto de valores de parámetro que se representan mediante columnas de clave. La etapa JDBC Connector establece los valores de parámetro en las variables de enlace en la sentencia SELECT, y luego la etapa JDBC Connector ejecuta la sentencia. El resultado de la búsqueda se direcciona como uno o más registros a través del enlace de referencia desde la etapa JDBC Connector a la etapa Lookup y desde la etapa Lookup al enlace de salida de la etapa Lookup. Una búsqueda dispersa también se denomina búsqueda directa porque la búsqueda se realiza directamente en el origen de datos.

Normalmente, se utiliza una búsqueda dispersa cuando la tabla de destino es demasiado grande para caber en la memoria.

Procedimiento

1. Añada una etapa Lookup al lienzo de diseño de trabajos y luego cree un enlace de referencia de la etapa JDBC Connector a la etapa Lookup.
2. Efectúe una doble pulsación en la etapa JDBC Connector.
3. En la lista **Tipo de búsqueda**, seleccione **Dispersa**.
4. Especifique las columnas de clave:
 - a. Si ha establecido **Generar SQL** en **Sí** al configurar el conector como origen, especifique el nombre de tabla, y luego especifique las columnas de clave en la página Columnas.
 - b. Si ha establecido **Generar SQL** en **No** al configurar el conector como origen, especifique un valor para la propiedad **Sentencia Select**. En la parte de selección de la sentencia SELECT, liste las columnas que desea devolver a la etapa Lookup. Asegúrese de que las columnas en la lista de selección tengan las columnas correspondientes en el enlace de referencia. Cada parámetro en la cláusula WHERE de la sentencia debe tener la palabra *ORCHESTRATE* y un punto añadidos al principio del nombre de columna. La palabra *ORCHESTRATE* puede estar toda en letras mayúsculas o toda en letras minúsculas, como por ejemplo *ORCHESTRATE.Field001*. La siguiente sentencia SELECT es un ejemplo de la sintaxis correcta de la cláusula WHERE: `select Field002,Field003 from MY_TABLE where Field001 = ORCHESTRATE.Field001`. Los nombres de columna que van a continuación de la palabra *ORCHESTRATE* deben tener las columnas de clave coincidentes en el enlace de referencia.
5. Para guardar los cambios, pulse **Aceptar**.
6. Efectúe una doble pulsación en la etapa Lookup.
7. Correlacione las columnas de enlace de entrada y de enlace de referencia con las columnas de enlace de salida y especifique condiciones para una anomalía en la búsqueda:
 - a. Arrastre o copie las columnas del enlace de entrada y del enlace de referencia al enlace de salida.
 - b. Para definir condiciones para una anomalía en la búsqueda, pulse el icono **Restricciones** en el menú.
 - c. En la columna **Anomalía en la búsqueda**, seleccione un valor y luego pulse **Aceptar**. Si selecciona **Rechazar**, debe tener un enlace de rechazo precedente

de la etapa Lookup y una etapa de destino en la configuración del trabajo para capturar los registros rechazados.

d. Pulse **Aceptar**.

8. Guarde, compile y ejecute el trabajo.

Tareas relacionadas:

“Configurar definiciones de columna en un enlace” en la página 6

Las definiciones de columna, que se establecen en un enlace, especifican el formato de los registros de datos que el conector lee de una base de datos o graba en una base de datos.

“Configurar el conector JDBC como origen” en la página 6

Para configurar una etapa JDBC Connector para leer o buscar filas en una tabla o vista JDBC, debe especificar la tabla o vista de origen o definir una sentencia SELECT completa.

Referencia

Las siguientes secciones tratan con más detalle características funcionales específicas y escenarios de uso del conector JDBC.

Propiedades para el conector JDBC

Utilice estas opciones para gestionar cómo lee y graba datos el conector.

Propiedad **Habilitar identificadores con comillas**

Para configurar el conector para entrecomillar los nombres de objeto en las sentencias SQL generadas, establezca la propiedad **Habilitar identificadores con comillas** en Sí.

Uso

Cuando el conector está configurado para generar automáticamente sentencias SQL, se puede configurar para encerrar los nombres de objeto de columna, esquema y tabla en las sentencias entre las comillas que estén soportadas por el controlador. Entrecomillar los nombres de objeto en las sentencias puede ser necesario para conservar las mayúsculas y minúsculas o para permitir el uso de espacios o caracteres especiales en los nombres.

El conector realiza una consulta al controlador para determinar las comillas que utiliza el origen de datos. Si el conector no puede obtener esta información, utiliza comillas dobles como el signo de comillas de forma predeterminada.

Ejemplo

Suponga que el origen de datos utiliza comillas dobles como el signo de comillas y que cambia automáticamente los nombres de objeto en las sentencias a mayúsculas cuando estos no están encerrados explícitamente entre comillas dobles. La siguiente tabla muestra ejemplos de la sentencia SELECT que se genera para combinaciones de nombres de columna, nombres de tabla y valores de la propiedad **Habilitar identificadores con comillas**.

Columnas de enlace de entrada	Nombre de tabla	Valor establecido para la propiedad Habilitar identificadores con comillas	Sentencia generada
c1, c2	t1	No	SELECT c1, c2 FROM t1
c1, c2	t1	Sí	SELECT "c1","c2" FROM "t1"
c1, c2	s1.t1	No	SELECT c1, c2 FROM s1.t1
c1, c2	s1.t1	Sí	SELECT "c1", "c2" FROM "s1"."t1"

Referencia relacionada:

“Derivaciones de columna” en la página 26

Cuando el conector utiliza una sentencia SELECT para leer filas del origen de datos, el conector debe correlacionar los nombres de columna devueltos por la sentencia SELECT con las columnas definidas en el enlace de salida.

Propiedades para la gestión de transacciones

Utilice la propiedad **Nivel de aislamiento** para especificar el nivel de aislamiento para las transacciones en la conexión que el conector establece con el origen de datos.

Uso

El valor predeterminado para esta propiedad es **Predeterminado**. Cuando se selecciona este valor, el conector no establece ningún nivel de aislamiento para las transacciones y se utiliza el nivel de aislamiento predeterminado para el controlador. Para obtener más información sobre los niveles de aislamiento a los que da soporte el controlador y cómo el controlador los correlaciona con los niveles de aislamiento soportados por el origen de datos del programa de fondo, consulte la documentación del producto de su controlador.

Si se selecciona un valor distinto de **Predeterminado** para la propiedad **Nivel de aislamiento** y el conector no puede establecer el nivel de aislamiento especificado, el conector detiene el trabajo y registra un mensaje de error. El mensaje de error incluye la información de error del controlador. Si el conector establece un nivel de aislamiento específico pero determina que el controlador ha cambiado automáticamente el nivel de aislamiento, el conector detiene el trabajo y registra un mensaje de error.

Si el controlador no da soporte a niveles de aislamiento, la propiedad **Nivel de aislamiento** debe establecerse en **Predeterminado**.

El conector también permite habilitar la **Modalidad de confirmación automática** para la conexión. Cuando se selecciona esta modalidad, las operaciones que realiza el conector en la conexión con el origen de datos se confirman automáticamente una vez completadas.

Propiedad Ordenación de registros

Utilice la propiedad **Ordenación de registros** para especificar el orden de procesamiento de registros de varios enlaces de entrada.

Uso

Puede establecer la propiedad **Ordenación de registros** en los valores que figuran en la siguiente tabla.

Tabla 2. Valores de la propiedad **Ordenación de registros**

Valor	Descripción
Todos los registros	Se procesan todos los registros del primer enlace, después se procesan todos los registros del segundo enlace, y así sucesivamente.
Primer registro	Se procesa un registro de cada enlace hasta que se procesan todos los registros de todos los enlaces.
Ordenado	Los registros se seleccionan de los enlaces de entrada según el orden que especifique para las propiedades Columna de clave , Nombre de columna , Orden de clasificación , Orden de los nulos y Distinguir entre mayúsculas y minúsculas .

Ejemplo

Este ejemplo describe cómo los cambios en los valores de la propiedad **Ordenación de registros** afectan al orden en el que la etapa procesa los registros.

Suponga que la etapa JDBC Connector tiene tres enlaces de entrada, Enlace1, Enlace2 y Enlace3. Cada uno de los enlaces tiene dos columnas definidas para él: C1 de tipo VarChar(20) y C2 de tipo Fecha.

Los registros llegan a estos enlaces en el orden que se muestra en la siguiente tabla:

Número de registro	Enlace1	Enlace2	Enlace3
1	value 2,2001-01-01	NULL,2002-02-01	Value 3,2003-03-01
2	NULL,2001-01-02	value 2,2002-02-02	value 2,2003-03-02
3	Value 1,2001-01-03	Value 3,2002-02-03	Value 2,2003-03-03
4	value 3,2001-01-04		value 4,2003-03-04

Suponga los siguientes valores:

- **Ordenación de registros** se establece en **Ordenado**.
- La columna C1 se utiliza como la columna de clave para la ordenación.
- **Orden de clasificación** se establece en **Ascendente**.
- **Orden de los nulos** se establece en **Antes**.
- **Distinguir entre mayúsculas y minúsculas** se establece en **No**.

En este escenario, el conector ordena los registros en los enlaces de entrada basándose en las opciones de ordenación que se han especificado y los registros se procesan en el orden en que se muestra en la siguiente tabla:

Orden	Enlace	Registro
1	Enlace1	NULL,2001-01-02
2	Enlace2	NULL,2002-02-01
3	Enlace1	Value1,2001-01-03
4	Enlace1	value2,2001-01-01
5	Enlace2	value2,2002-02-02
6	Enlace3	value2,2003-03-02
7	Enlace3	Value2,2003-03-03
8	Enlace1	value3,2001-01-04
9	Enlace2	Value3,2002-02-03
10	Enlace3	Value3,2003-03-01
11	Enlace3	value4,2003-03-04

Propiedad Acción en tabla

Utilice la propiedad **Acción en tabla** para configurar el conector para que realice acciones crear, sustituir y truncar en una tabla en tiempo de ejecución. Estas acciones se realizan antes de grabar datos en la tabla.

Uso

Puede establecer la propiedad **Acción en tabla** en los valores que se indican en la siguiente tabla.

Tabla 3. Valores de la propiedad **Acción en tabla**

Valor	Descripción
Anexar	No se realiza ninguna acción en la tabla. Esta opción es el valor predeterminado.
Crear	<p>Crear una tabla en tiempo de ejecución.</p> <p>Utilice uno de estos métodos para especificar la sentencia CREATE TABLE:</p> <ul style="list-style-type: none"> • Establezca Generar sentencia Create table en tiempo de ejecución en Sí y especifique el nombre de la tabla que desea crear en la propiedad Nombre de tabla. En este caso, el conector genera automáticamente la sentencia CREATE TABLE a partir de las definiciones de columna del enlace de entrada. Los nombres de columna de la nueva tabla coinciden con los nombres de columna del enlace. Los tipos de datos de las columnas de la nueva tabla se generan a partir de las definiciones de columna del enlace. • Establezca Generar sentencia Create table en tiempo de ejecución en No y especifique la sentencia CREATE TABLE en la propiedad Sentencia Create table.

Tabla 3. Valores de la propiedad **Acción en tabla** (continuación)

Valor	Descripción
Sustituir	<p>Sustituir una tabla en tiempo de ejecución.</p> <p>Utilice uno de estos métodos para especificar la sentencia DROP TABLE:</p> <ul style="list-style-type: none"> • Establezca Generar sentencia Drop table en tiempo de ejecución en Sí y especifique el nombre de la tabla que desea descartar en la propiedad Nombre de tabla. • Establezca Generar sentencia Drop table en tiempo de ejecución en No y especifique la sentencia DROP TABLE en la propiedad Sentencia Drop table. <p>Utilice uno de estos métodos para especificar la sentencia CREATE TABLE:</p> <ul style="list-style-type: none"> • Establezca Generar sentencia Create table en tiempo de ejecución en Sí y especifique el nombre de la tabla que desea crear en la propiedad Nombre de tabla. • Establezca Generar sentencia Create table en tiempo de ejecución en No y especifique la sentencia CREATE TABLE en la propiedad Sentencia Create table.
Truncar	<p>Truncar una tabla en tiempo de ejecución.</p> <p>Utilice uno de estos métodos para especificar la sentencia TRUNCATE TABLE:</p> <ul style="list-style-type: none"> • Establezca Generar sentencia Truncate table en tiempo de ejecución en Sí y especifique el nombre de la tabla que desea truncar en la propiedad Nombre de tabla. • Establezca Generar sentencia Truncate table en tiempo de ejecución en No y especifique la sentencia TRUNCATE TABLE en la propiedad Sentencia Truncate table.

Para configurar el trabajo para que finalice de forma anómala cuando falle la sentencia especificada mediante la acción de tabla, puede establecer la propiedad correspondiente en **Sí**:

- **Detener el trabajo cuando falle la sentencia de creación de tabla**
- **Detener el trabajo cuando falle la sentencia de descartar tabla**
- **Detener el trabajo cuando falle la sentencia de truncamiento de tabla**

De lo contrario, cuando la sentencia falle, el conector registrará un mensaje de aviso y el trabajo continuara.

Propiedad Informar de discrepancia de esquema

Utilice la propiedad **Informar de discrepancia de esquema** para comparar las definiciones de columna en el enlace con las definiciones de columna en el origen de datos.

Uso

Establezca la propiedad **Informar de discrepancia de esquema** en **Sí** para comparar las definiciones de columna en el enlace con las definiciones de columna en el origen de datos. El conector registra mensajes para cualquier discrepancia de tipo de datos que detecta.

Dependiendo del controlador y del escenario de uso, es posible que el conector no pueda detectar todas las discrepancias. El conector realiza los informes de discrepancia de esquema siempre que sea posible. Independientemente de la creación de informes de discrepancia de esquema, el conector registra un mensaje de error si detecta un error de conversión de tipo de datos o de recorte de datos.

Cuando el conector está configurado para grabar registros de enlace de entrada en el origen de datos, la propiedad **Informar de discrepancia de esquema** está habilitada para los enlaces de entrada sólo si el conector está configurado para generar sentencias SQL automáticamente. Si especifica las sentencias SQL manualmente, el conector no intenta realizar la detección anticipada de discrepancias de esquema y la propiedad **Informar de discrepancia de esquema** está inhabilitada.

Propiedad Descartar campos que no coincidan

Utilice la propiedad **Descartar campos que no coincidan** para especificar cómo manejar las columnas no utilizadas en el enlace de entrada.

Uso

Cuando se crea un trabajo que graba datos del enlace de entrada en el origen de datos, puede utilizar la propiedad **Descartar campos que no coincidan** para controlar cómo manejar las columnas no utilizadas (campos) en el enlace de entrada. Las columnas no utilizadas en el enlace de entrada son las columnas que el conector no ha emparejado con ningún parámetro de la sentencia SQL de destino.

Puede establecer la propiedad **Descartar campos que no coincidan** en los valores que se indican en la siguiente tabla.

Tabla 4. Valores de la propiedad **Descartar campos que no coincidan**

Valor	Descripción
Sí	El conector descarta todas las columnas no utilizadas en el enlace de entrada. Para cada columna descartada, el conector graba un mensaje informativo en el registro de trabajo para indicar que la columna y sus valores asociados no se tienen en cuenta.
No	Cuando el conector encuentra una columna no utilizada en el enlace de entrada, el conector registra un mensaje de error y detiene el trabajo.

Puede utilizar la propiedad **Habilitar identificadores con comillas** para especificar si la coincidencia de nombres entre las columnas del enlace de entrada y las columnas de tabla o los parámetros de la sentencia SQL de destino distingue entre mayúsculas y minúsculas.

Referencia relacionada:

“Propiedad Habilitar identificadores con comillas” en la página 16
Para configurar el conector para entrecomillar los nombres de objeto en las sentencias SQL generadas, establezca la propiedad **Habilitar identificadores con comillas** en Sí.

Propiedad Juego de caracteres para columnas no Unicode

Puede utilizar las propiedades de juego de caracteres para especificar el juego de caracteres que se utiliza para las columnas no Unicode en los enlaces conectados a la etapa.

Uso

Utilice la propiedad **Juego de caracteres para columnas no Unicode** para seleccionar el juego de caracteres que se utiliza para los valores de las columnas de enlace Char, VarChar y LongVarChar para las que la propiedad **Ampliado** no está establecida en Unicode.

Si establece la propiedad en **Predeterminado**, se utiliza el juego de caracteres del entorno local del sistema host del motor.

Si establece la propiedad en **Personalizado**, debe proporcionar el nombre del juego de caracteres que se va a utilizar. Establezca la propiedad **Nombre de juego de caracteres** en el nombre de la codificación de juego de caracteres para los valores de las columnas de enlace Char, VarChar y LongVarChar para las que la propiedad **Ampliado** no está establecida en Unicode.

Pulse el botón **Seleccionar** de esta propiedad para obtener la lista de todos los juegos de caracteres disponibles.

Propiedad Ejecutar sentencias SQL Before y After

Utilice la propiedad **Ejecutar sentencias SQL Before y After** para configurar el conector para que ejecute sentencias SQL separadas por signos de punto y coma antes o después de procesar datos. Puede configurar el conector para que ejecute sentencias SQL antes o después de procesar cualquier dato en un trabajo o para que ejecute sentencias SQL una vez antes o después de procesar los datos en cada nodo.

Uso

La ejecución de sentencias SQL antes o después de procesar datos es útil cuando necesita realizar operaciones que preparan objetos de origen de datos para el acceso a datos. Por ejemplo, puede utilizar una sentencia SQL para crear una tabla de destino y añadir un índice a la misma. Las sentencias SQL que especifique se ejecutan una vez para todo el trabajo, antes de procesar cualquier dato.

Una vez que el conector ha ejecutado las sentencias especificadas en la propiedad **Sentencia SQL Before** o la propiedad **Sentencia SQL After**, el conector confirma explícitamente la transacción actual. Por ejemplo, si especifica una sentencia DML, como INSERT, UPDATE o DELETE, en la propiedad **Sentencia SQL Before**, los resultados de la sentencia DML son visibles en los nodos individuales.

Para ejecutar sentencias SQL en cada nodo en el que el conector está configurado para ejecutarse, utilice la propiedad **Sentencia SQL Before (nodo)** o la propiedad **Sentencia SQL After (nodo)**. El conector ejecuta las sentencias SQL especificadas

una vez antes de procesar cualquier dato en cada nodo o una vez después de procesar cualquier dato en cada nodo. A continuación, el conector confirma explícitamente la transacción actual.

Cuando especifique la sentencia que desea ejecutar antes o después de procesar, especifique las sentencias SQL o especifique la vía de acceso completa al archivo que contiene la sentencia SQL. No incluya variables de enlace de entrada o variables de enlace de salida en la sentencia SQL. Si la sentencia contiene estos tipos de variables, el conector registra un mensaje de error y la operación se detiene. Si especifica un nombre de archivo, el archivo debe residir en el host de la capa de motor, y debe establecer la propiedad **Leer la sentencia SQL Before del archivo, Leer la sentencia SQL Before (nodo) del archivo, Leer la sentencia SQL After (nodo) del archivo** o **Leer la sentencia SQL After del archivo** en Sí.

Cuando el conector ejecuta un conjunto de sentencias que se han especificado en cualquiera de las propiedades **Sentencia SQL Before, Sentencia SQL Before (nodo), Sentencia SQL After (nodo)** o **Sentencia SQL After**, el conector informa de un error y detiene el trabajo si falla cualquiera de las sentencias del conjunto. Puede configurar el conector para que informe de un mensaje de aviso y siga procesando las sentencias restantes del conjunto. Para configurar el conector para que informe de un mensaje de aviso, establezca la propiedad **Detener el trabajo cuando falle la sentencia SQL Before, Detener el trabajo cuando falle la sentencia SQL Before (nodo), Detener el trabajo cuando falle la sentencia SQL After (nodo)** o **Detener el trabajo cuando falle la sentencia SQL After** en No.

Cuando el conector se utilice para grabar registros en la base de datos y esté configurado para realizar una acción de tabla en la tabla de destino antes de grabar datos, puede utilizar la propiedad **Realizar primero la acción de tabla** para controlar lo que se ejecuta primero, la sentencia SQL Before o las sentencias de acción de tabla.

Propiedad Ejecutar sentencias SQL de inicio y finalización

Utilice la propiedad **Ejecutar sentencias SQL de inicio y finalización** para ejecutar sentencias SQL en cada transacción, inmediatamente después de que se inicie la transacción e inmediatamente antes de finalice satisfactoriamente la transacción.

Uso

Las sentencias especificadas en la propiedad **Sentencia SQL Begin** se ejecutan cada vez que el conector inicia una transacción y antes de procesar cualquier registro en dicha transacción.

Las sentencias especificadas en la propiedad **Sentencia SQL End** se ejecutan después de procesar todos los registros de una transacción e inmediatamente antes de que la transacción finalice satisfactoriamente.

Se graban mensajes informativos en el registro de trabajo al final del trabajo para indicar el número de veces que la **Sentencia SQL Begin** y la **Sentencia SQL End** se ejecutaron en nodos de proceso individuales para la etapa.

En el conector JDBC, se utilizan transacciones para agrupar registros en los enlaces de la etapa a fin de poder procesarlos como una sola unidad de trabajo. El número de registros en cada transacción se define mediante la propiedad de conector **Recuento de registros**, registros de marcador de fin de oleada o una combinación

de ambos. Las transacciones en el conector JDBC se corresponden con las transacciones en el origen de datos JDBC al que está conectada la etapa cuando se cumplen las siguientes condiciones:

- El origen de datos es un origen de datos transaccionales.
- En el conector JDBC, la propiedad **Modalidad de confirmación automática** está establecida en **Inhabilitar**.
- En el conector JDBC, la propiedad **Nivel de aislamiento** está establecida en un valor distinto de **Predeterminado**. O bien, el valor está establecido en **Predeterminado** y el nivel de aislamiento predeterminado en el origen de datos implica el uso de transacciones en el origen de datos.

Las propiedades **Sentencia SQL Begin** y **Sentencia SQL End** son aplicables independientemente de si las transacciones en la etapa de conector JDBC se corresponden o no con las transacciones del origen de datos JDBC.

Cuando el trabajo se ejecuta, todas las apariciones de las expresiones `[[número-nodo]]`, `[[número-nodo-base-uno]]` y `[[recuento-nodos]]` en las propiedades **Sentencia SQL Begin** y **Sentencia SQL End** se sustituyen por el índice de base cero del nodo de proceso actual, el índice de base uno del nodo de proceso actual y el número total de nodos de proceso para la etapa, respectivamente.

Por ejemplo, suponga que la etapa de conector JDBC está configurada para ejecutarse en tres nodos de proceso y que la propiedad **Sentencia SQL Begin** tiene el siguiente valor:

```
INSERT INTO T1 VALUES ([[número-nodo]], 'Nodo [[número-nodo-base-uno]] de [[recuento-nodos]]')
```

Cuando se ejecuta el trabajo, la etapa ejecuta la siguiente **Sentencia SQL Begin** en los nodos de proceso individuales:

- En el primer nodo: `INSERT INTO T1 VALUES (0, 'Nodo 1 de 3')`
- En el segundo nodo: `INSERT INTO T1 VALUES (1, 'Nodo 2 de 3')`
- En el tercer nodo: `INSERT INTO T1 VALUES (2, 'Nodo 3 de 3')`

Las sentencias que se especifican en las propiedades **Sentencia SQL Begin** y **Sentencia SQL End** se ejecutan mientras la sentencia principal para procesar registros de datos está abierta. Por ejemplo, cuando la etapa de conector tiene un enlace de salida, la sentencia `SELECT` que lee registros del origen de datos está abierta cuando se crean y ejecutan las sentencias en las propiedades **Sentencia SQL Begin** y **Sentencia SQL End**. Otro ejemplo, suponga que la etapa de conector tiene un enlace de entrada y está configurada para insertar registros en el origen de datos. La sentencia `INSERT` que se utiliza para insertar registros en el origen de datos está abierta cuando se crean y ejecutan las sentencias en las propiedades **Sentencia SQL Begin** y **Sentencia SQL End**.

Para algunos controladores JDBC, sólo una sentencia puede estar abierta a la vez en la conexión. Si utiliza las propiedades **Sentencia SQL Begin** y **Sentencia SQL End** con estos controladores, el controlador informa de un error, y el trabajo falla.

Propagación de columnas en tiempo de ejecución

Puede configurar una etapa JDBC para añadir automáticamente columnas al enlace de salida en tiempo de ejecución.

Uso

Antes de poder habilitar la propagación de columnas en tiempo de ejecución en una etapa, la propagación de columnas en tiempo de ejecución debe habilitarse para trabajos paralelos a nivel de proyecto desde el Administrador de IBM InfoSphere DataStage and QualityStage.

Cuando la propiedad de propagación de columnas en tiempo de ejecución está habilitada para la etapa JDBC Connector, el conector revisa las columnas para ver el conjunto de resultados de la sentencia que está configurado para ejecutar. A continuación, el conector compara este conjunto de columnas con el conjunto de columnas que están definidas en el enlace de salida. Si hay columnas en el conjunto de resultados que no están definidas en el enlace de salida, el conector las añade al enlace de salida.

Para visualizar y habilitar la propiedad **Propagación de columnas en tiempo de ejecución**, primero debe habilitar la opción **Habilitar la propagación de columnas de tiempo en ejecución para trabajos paralelos** para el proyecto de DataStage actual en el Administrador de DataStage and QualityStage. A continuación, marque el recuadro de selección **Propagación de columnas en tiempo de ejecución** en la página Columnas para el enlace de salida en el editor de etapas.

Para comprobar si la columna de enlace existe para una determinada columna del conjunto de resultados, el conector realiza los pasos siguientes:

1. Comprueba si la propiedad Derivación para cualquier columna de enlace coincide con el nombre de la columna del conjunto de resultados. Esta coincidencia es sensible a las mayúsculas y minúsculas. Si el valor de la propiedad Derivación empieza y termina con comillas, los identificadores entre comillas se eliminan antes de comprobar la propiedad. Si se encuentra una columna coincidente, se presupone que la columna representa la columna del conjunto de resultados y el conector no añade una columna al enlace.
2. Si no se encuentra una columna coincidente, el conector comprueba si la propiedad Nombre de cualquier columna de enlace coincide con la columna del conjunto de resultados. Esta coincidencia es sensible a las mayúsculas y minúsculas sólo si la propiedad **Habilitar identificadores con comillas** está establecida en **Sí** para el enlace. Si se encuentra una columna coincidente, se presupone que la columna representa la columna del conjunto de resultados y el conector no añade una columna al enlace. Si no se encuentra una columna coincidente, el conector añade una columna al enlace. El conector añade una columna al enlace de salida que tiene el mismo nombre que la columna del conjunto de resultados, con las siguientes modificaciones:
 - El signo de dólar (\$) se sustituye por el valor `__036__`.
 - El signo de almohadilla (#) se sustituye por el valor `__035__`.
 - Cada carácter no alfanumérico distinto del signo de subrayado (`_`), el signo de dólar (\$) y el signo de almohadilla (#) se sustituye por el valor de par de subrayados `__`.
 - Si se realiza cualquier sustitución por un par de subrayados, o si el primer carácter del nombre de columna no es un carácter de letra, un signo de subrayado (`_`), un signo de dólar (\$) o un signo de almohadilla (#), se añade el prefijo `CC_n` al nombre de columna, donde `n` es el índice de la columna de expresión SQL en la lista de sentencias SELECT.
 - Si el controlador ha informado que el nombre de la columna del conjunto de resultados era una serie vacía o un nulo, se utiliza el índice de columna (1, 2 y así sucesivamente) como el nombre de columna.

- Los caracteres de espacio en blanco se ignoran.
- Si el nombre de la columna resultante es el mismo que el de una columna que ya se había añadido al enlace, se le añade el sufijo *m*, donde *m* es el valor entero más pequeño superior o igual a 1 y cuyo resultado sea un nombre de columna exclusivo.

En las columnas que el conector añade, la propiedad Derivación se establece en un valor que coincida con el nombre de la columna del conjunto de resultados.

Ejemplo

Los siguientes ejemplos ilustran cómo funciona la propagación de columnas en tiempo de ejecución.

Suponga que la etapa JDBC Connector está configurada para captar datos del origen de datos y proporcionar registros en el enlace de salida. Además, los ejemplos presuponen que el recuadro **Propagación de columnas en tiempo de ejecución** está seleccionado para el enlace de salida.

Suponga que no hay ninguna columna definida para el enlace de salida, y el conector está configurado para generar automáticamente una sentencia SELECT para leer de la tabla TABLE1. La tabla TABLE1 contiene las columnas C1, C2 y C3. El conector genera y ejecuta la sentencia SELECT * FROM TABLE1 y añade las columnas C1, C2 y C3 al enlace de salida.

Suponga que no hay ninguna columna definida para el enlace de salida, y el conector está configurado para generar automáticamente una sentencia SELECT para leer de la tabla TABLE1. La tabla TABLE1 contiene las columnas C\$1, C 2 y C!3#. El conector genera y ejecuta la sentencia SELECT * FROM TABLE1 y añade las columnas C__036__1, C2 y CC_3_C__3__035__ al enlace de salida. El conector genera los nombres de columna de la siguiente manera:

- C\$1 El conector sustituye \$ por __036__.
- C 2 El conector elimina el carácter de espacio entre C y 2.
- C!3# El conector sustituye ! por __ y sustituye # por __035__ y añade el prefijo CC_3_ porque esta columna es la tercera columna en el conjunto de resultados.

Derivaciones de columna

Cuando el conector utiliza una sentencia SELECT para leer filas del origen de datos, el conector debe correlacionar los nombres de columna devueltos por la sentencia SELECT con las columnas definidas en el enlace de salida.

El conector utiliza el siguiente proceso para correlacionar columnas: El conector busca la columna de enlace de salida que tenga la propiedad Derivación establecida en un valor que coincida con el nombre de columna devuelto por la sentencia SELECT. Si el conector no encuentra dicha columna en el enlace, busca la columna de enlace de salida que tenga la propiedad Nombre establecida en un valor que coincida con el nombre de columna devuelto por la sentencia SELECT. Si el conector tampoco encuentra esa columna, añade la columna al enlace de salida si la propagación de columnas en tiempo de ejecución está habilitada para el enlace. De lo contrario, ignora los valores en la columna devuelta por la sentencia SELECT cuando lee filas del origen de datos.

Cuando el conector correlaciona columnas, compara el nombre de columna devuelto por la sentencia SELECT y el valor de la propiedad Derivación para las columnas de enlace de salida de manera sensible a las mayúsculas y minúsculas, independientemente del valor de la propiedad **Habilitar identificadores con comillas**. Si el valor de Derivación está encerrado entre comillas, se eliminan las comillas antes de realizar la comparación. Sin embargo, la forma en que compara los nombres de columna de la sentencia y los valores de Nombre de las columnas de enlace de salida se basa en el valor de la propiedad **Habilitar identificadores con comillas**. Si esta propiedad está establecida en **Sí**, la comparación es sensible a las mayúsculas y minúsculas, y si esta propiedad está establecida en **No**, la comparación no es sensible a las mayúsculas y minúsculas.

Cuando el conector está configurado para generar automáticamente la sentencia SELECT en tiempo de ejecución, éste ensambla la sentencia basándose en la propiedad **Nombre de tabla** especificada y las definiciones de columna en el enlace de salida. Si la propagación de columnas en tiempo de ejecución está habilitada para el enlace de salida, el conector incluye todas las columnas de tabla en la sentencia generando la siguiente sentencia:

```
SELECT * FROM nombre_tabla
```

Si la propagación de columnas en tiempo de ejecución no está habilitada para el enlace de salida, el conector incluye en la sentencia únicamente las columnas del enlace de salida. Para cada columna en el enlace de salida, incluye el valor que está establecido para la propiedad Derivación de la columna. Si no se ha establecido ningún valor para la propiedad Derivación, el conector incluye el valor que está establecido para la propiedad Nombre en la sentencia.

Cuando el conector incluye valores de Derivación en la sentencia SELECT que se genera, el conector siempre los incluye exactamente en la misma forma en que están especificados para las columnas de enlace. Esto incluye cualquier comilla especificada en los valores de Derivación. Por otra parte, cuando incluye los valores de Nombre de columna en el texto de la sentencia SELECT generada, si la propiedad **Habilitar identificadores con comillas** está establecida en **Sí**, el conector encierra los valores entre comillas en el texto de la sentencia. Si la propiedad **Habilitar identificadores con comillas** está establecida en **No**, el conector incluye los valores en la misma forma en que están especificados para las columnas de enlace.

Ejemplo

Suponga que la tabla T1 tiene las siguientes columnas:

C1 A

C1 B

C2

Para leer filas de esta tabla, puede definir las columnas en el enlace de salida con la propiedad Nombre establecida en C1A, C1B y C2 y la propiedad Derivación establecida en series vacías. Luego, puede configurar el conector para que ejecute la siguiente sentencia SELECT (suponga que el origen de datos da soporte a los alias de columna):

```
SELECT "C1 A" AS C1A, "C1 B" AS C1B, C2 FROM T1
```

En este caso, las columnas devueltas por la sentencia son C1A, C1B y C2. El conector no encuentra ninguna columna en el enlace de salida que tenga la propiedad Derivación establecida en uno de estos valores. No obstante, debido a que encuentra columnas que tienen la propiedad Nombre establecida en cada uno de estos valores, correlaciona las columnas de la sentencia con esas columnas del enlace de salida satisfactoriamente.

De forma alternativa, puede utilizar los valores de la propiedad Derivación de las columnas de enlace de salida. Puede definir las columnas C1A, C1B y C2 en el enlace de salida, establecer la propiedad Derivación de las columnas C1A en C1 A, establecer la propiedad Derivación de las columnas C1B en C1 B y dejar la propiedad Derivación de la columna C2 en blanco. A continuación, puede configurar la etapa para generar automáticamente la sentencia SELECT.

El conector utiliza la propiedad Derivación de las columnas C1A y C1B y genera la siguiente sentencia SELECT:

```
SELECT "C1 A", "C1 B", C2 FROM T1
```

En este caso, las columnas devueltas por la sentencia son C1 A, C1 B y C2. El conector correlaciona la columna C1 A de la sentencia con la columna C1A del enlace de salida porque la columna de enlace de salida tiene la propiedad Derivación establecida en C1 A, que coincide con el nombre de columna C1 A de la sentencia. Asimismo, el conector correlaciona la columna C1 B de la sentencia con la columna C1B del enlace de salida porque esta columna de enlace de salida tiene la propiedad Derivación establecida en C1 B, que coincide con la columna C1 B de la sentencia. Por último, el conector correlaciona la columna C2 de la sentencia con la columna C2 del enlace de salida.

Referencia relacionada:

“Propiedad Habilitar identificadores con comillas” en la página 16

Para configurar el conector para entrecomillar los nombres de objeto en las sentencias SQL generadas, establezca la propiedad **Habilitar identificadores con comillas** en Sí.

“Propagación de columnas en tiempo de ejecución” en la página 24

Puede configurar una etapa JDBC para añadir automáticamente columnas al enlace de salida en tiempo de ejecución.

Condiciones de rechazo

Cuando utilice el conector JDBC para grabar datos en un origen de datos JDBC, puede añadir un enlace de rechazo a la etapa JDBC Connector y enviar los registros rechazados a la etapa conectada al otro extremo del enlace. Las condiciones de rechazo determinan cuándo se rechaza un registro.

Puede establecer las siguientes condiciones de rechazo:

Error de SQL

El controlador de registros informa de un error cuando intenta realizar la operación de grabación especificada por el conector. Esta condición es aplicable a todas las modalidades de grabación.

Fila no insertada

La sentencia INSERT especificada se ha ejecutado en el registro y la sentencia no ha producido un error. Sin embargo, no se ha insertado ninguna fila en la tabla de destino. Esta condición es aplicable a las siguientes modalidades de grabación: Insertar, Insertar y luego actualizar, Actualizar y luego insertar, Suprimir y luego insertar, Insertar sólo las filas nuevas.

Fila no actualizada

La sentencia UPDATE especificada se ha ejecutado en el registro y la sentencia no ha producido un error. Sin embargo, no se ha actualizado ninguna fila en la tabla de destino. Esta condición es aplicable a las siguientes modalidades de grabación: Actualizar, Insertar y luego actualizar.

Fila no suprimida

La sentencia DELETE especificada se ha ejecutado en el registro y la sentencia no ha producido un error. Sin embargo, no se ha suprimido ninguna fila de la tabla de destino. Esta condición sólo es aplicable a la modalidad de grabación Suprimir.

Si se selecciona una condición de rechazo con una modalidad de grabación para la que no es aplicable, el conector pasa por alto el valor de la condición de rechazo.

Tareas relacionadas:

“Rechazar registros que contienen errores” en la página 11

Cuando una etapa JDBC Connector incluye un enlace de rechazo, los registros que cumplen los criterios especificados se dirigen automáticamente a la etapa de destino en el enlace de rechazo. El proceso continúa para los registros restantes.

Referencia relacionada:

“Modalidades de grabación soportadas” en la página 32

Al configurar el conector JDBC como destino, puede utilizar la propiedad

Modalidad de grabación para especificar la modalidad que desea utilizar para grabar filas en el origen de datos JDBC.

Lecturas particionadas

La etapa JDBC Connector se puede configurar para que se ejecute en varios nodos de proceso y para leer datos del origen de datos. En este escenario, cada uno de los nodos de proceso para la etapa recupera un conjunto (partición) de registros del origen de datos. Las particiones luego se combinan para producir el conjunto de resultados completo para el enlace de salida de la etapa.

Para habilitar las lecturas particionadas, establezca la propiedad **Habilitar lecturas particionadas** en Sí.

Cuando la etapa está configurada para lecturas particionadas, ejecuta la sentencia especificada en la propiedad **Sentencia Select** en cada nodo de proceso. Puede utilizar marcadores especiales en las sentencias para asegurar que cada uno de los nodos de proceso recupere una partición distinta de registros del origen de datos.

Los siguientes marcadores se pueden utilizar en las sentencias para especificar las distintas particiones de registros a recuperar en los nodos de proceso individuales:

- [[número-nodo]] - se sustituye en tiempo de ejecución por el índice del nodo de proceso actual. Los índices son de base cero. El marcador se sustituye por el valor 0 en el primer nodo de proceso, el valor 1 en el segundo nodo de proceso, el valor 2 en el tercer nodo de proceso, y así sucesivamente.
- [[número-nodo-base-uno]] - se sustituye en tiempo de ejecución por el índice del nodo de proceso actual. Los índices son de base uno. El marcador se sustituye por el valor 1 en el primer nodo de proceso, el valor 2 en el segundo nodo de proceso, el valor 3 en el tercer nodo de proceso, y así sucesivamente.
- [[recuento-nodos]] - el número total de nodos de proceso para la etapa. De forma predeterminada, este número es el número de nodos en el archivo de configuración paralelo. La ubicación del archivo de configuración paralelo se

especifica en la variable de entorno *APT_CONFIG_FILE*. La etapa se puede configurar para que se ejecute en un subconjunto de nodos del archivo de configuración paralelo, definiendo restricciones de nodo en la página Avanzado bajo la página Etapa en el editor de etapas.

Cuando las lecturas particionadas están habilitadas y la propiedad **Leer la sentencia de selección del archivo** está establecida en **Sí**, el conector resuelve todos los marcadores [[número-nodo]], [[número-nodo-base-uno]] y [[recuento-nodos]] en el valor de la propiedad **Sentencia Select** antes de acceder a la ubicación de archivo especificada mediante esta propiedad. Una vez que recupera el texto de la sentencia SELECT del archivo, resuelve de nuevo todos los marcadores [[número-nodo]], [[número-nodo-base-uno]] y [[recuento-nodos]] en el texto de la sentencia antes de ejecutar la sentencia.

El soporte de marcadores también se aplica a las propiedades **SQL Before (nodo)** y **SQL After (nodo)**. Cuando las lecturas particionadas están habilitadas, todos los marcadores [[número-nodo]], [[número-nodo-base-uno]] y [[recuento-nodos]] en las sentencias SQL Before (nodo) y SQL After (nodo) se resuelven en cada nodo de proceso antes de ejecutar las sentencias. Cuando la propiedad **Leer la sentencia SQL Before (nodo) del archivo** o **Leer la sentencia SQL After (nodo) del archivo** está establecida en **Sí**, todos los marcadores [[número-nodo]], [[número-nodo-base-uno]] y [[recuento-nodos]] en los nombres de archivo especificados en las propiedades **SQL Before (nodo)** y **SQL After (nodo)** se resuelven antes de abrir los archivos. Asimismo, todos los marcadores [[número-nodo]], [[número-nodo-base-uno]] y [[recuento-nodos]] en las sentencias SQL Before (nodo) y SQL After (nodo) que se recuperan de los archivos se resuelven en cada nodo de proceso antes de ejecutar las sentencias en ese nodo.

Asegúrese de que los registros devueltos por las sentencias en los nodos de proceso individuales tienen definiciones de campo coincidentes. Por ejemplo, si la sentencia en uno de los nodos de proceso devuelve registros con las columnas C1 INTEGER, C2 VARCHAR(20) y C3 DATE, asegúrese de que las sentencias en todos los nodos de proceso restantes devuelven registros con las mismas definiciones de columna.

Ejemplo

Suponga que la etapa se ha configurado para leer registros de la tabla de origen de datos TABLE1, que tiene las siguientes columnas: C1 INTEGER, C2 VARCHAR(10) y C3 DATE. Asimismo, suponga que la etapa se ha configurado para ejecutarse en paralelo en cuatro nodos de proceso y realizar lecturas particionadas. La propiedad **Sentencia Select** se ha establecido en el siguiente valor:

```
SELECT C1, C2, C3 FROM TABLE1 WHERE MOD(C1, [[recuento-nodos]]) = [[número-nodo]]
```

El conector ejecuta las siguientes sentencias en cada uno de los nodos de proceso:

Tabla 5. Sentencias a ejecutar en cada nodo de proceso

Nodo	Sentencia
1	SELECT C1, C2, C3 FROM TABLE1 WHERE MOD(C1, 4) = 0
2	SELECT C1, C2, C3 FROM TABLE1 WHERE MOD(C1, 4) = 1
3	SELECT C1, C2, C3 FROM TABLE1 WHERE MOD(C1, 4) = 2

Tabla 5. Sentencias a ejecutar en cada nodo de proceso (continuación)

Nodo	Sentencia
4	SELECT C1, C2, C3 FROM TABLE1 WHERE MOD(C1, 4) = 3

Suponga que la tabla de origen contiene las siguientes filas:

Tabla 6. Filas en la tabla de origen

C1	C2	C3
1	Valor uno	2013-01-01
2	Valor dos	2013-01-02
3	Valor tres	2013-01-03
4	Valor cuatro	2013-01-04
5	Valor cinco	2013-01-05
6	Valor seis	2013-01-06
7	Valor siete	2013-01-07
8	Valor ocho	2013-01-08
9	Valor nueve	2013-01-09
10	Valor diez	2013-01-10

Las siguientes filas son leídas por el primer nodo de proceso (índice de nodo 0):

Tabla 7. Filas captadas por el primer nodo de proceso

C1	C2	C3
4	Valor cuatro	2013-01-04
8	Valor ocho	2013-01-08

Las siguientes filas son captadas por el segundo nodo de proceso (índice de nodo 1):

Tabla 8. Filas captadas por el segundo nodo de proceso

C1	C2	C3
1	Valor uno	2013-01-01
5	Valor cinco	2013-01-05
9	Valor nueve	2013-01-09

Las siguientes filas son captadas por el tercer nodo de proceso (índice de nodo 2):

Tabla 9. Filas captadas por el tercer nodo de proceso

C1	C2	C3
2	Valor dos	2013-01-02
6	Valor seis	2013-01-06
10	Valor diez	2013-01-10

Las siguientes filas son captadas por el cuarto nodo de proceso (índice de nodo 3):

Tabla 10. Filas captadas por el cuarto nodo de proceso

C1	C2	C3
3	Valor tres	2013-01-03
7	Valor siete	2013-01-07

Tareas relacionadas:

“Configurar el conector JDBC para lecturas particionadas” en la página 7
Si la etapa JDBC Connector está configurada para ejecutarse en varios nodos de proceso, cada uno de los nodos de proceso lee datos del origen de datos simultáneamente con otros nodos de proceso. Las particiones de registros de todos los nodos de proceso se combinan para producir el conjunto de datos de registro completo para el enlace de salida de la etapa.

Modalidades de grabación soportadas

Al configurar el conector JDBC como destino, puede utilizar la propiedad **Modalidad de grabación** para especificar la modalidad que desea utilizar para grabar filas en el origen de datos JDBC.

En la siguiente tabla se listan las modalidades de grabación y se describen las operaciones que el conector realiza en la tabla de destino para cada modalidad de grabación.

Tabla 11. Modalidades de grabación y descripciones

Modalidad de grabación	Descripción
Insertar	El conector intenta insertar registros del enlace de entrada como filas en la tabla de destino.
Actualizar	El conector intenta actualizar filas en la tabla de destino que corresponden a los registros que llegan al enlace de entrada.
Suprimir	El conector intenta suprimir las filas de la tabla de destino que corresponden a los registros que llegan al enlace de entrada.
Insertar sólo las filas nuevas	El comportamiento de esta modalidad de grabación es muy similar al de la modalidad de grabación Insertar. Sin embargo, cuando se selecciona esta modalidad de grabación, los registros que no se puede grabar en la base de datos debido a una violación de restricción de integridad se ignoran, y el conector procesa los registros restantes. Cuando se produce cualquier error que no sea una violación de restricción de integridad, el conector registra igualmente un error y detiene el trabajo.
Insertar y luego actualizar	Para cada registro de entrada, el conector primero intenta insertar el registro como una fila nueva en la tabla de destino. Si la operación de inserción falla debido a una violación de restricción de integridad, el conector actualiza la fila existente en la tabla de destino con los nuevos valores del registro de entrada.

Tabla 11. Modalidades de grabación y descripciones (continuación)

Modalidad de grabación	Descripción
Actualizar y luego insertar	Para cada registro de entrada, el conector primero intenta localizar las filas coincidentes en la tabla de destino y actualizarlas con los nuevos valores del registro de entrada. Si la operación de actualización tiene como resultado cero filas actualizadas, el conector inserta el registro como una fila nueva en la tabla de destino.
Suprimir y luego insertar	Para cada registro de entrada, el conector primero intenta suprimir las filas coincidentes en la tabla de destino. Independientemente de si se ha suprimido realmente alguna fila o no, el conector luego ejecuta la sentencia de inserción para insertar el registro como una fila nueva en la tabla de destino.
Personalizada	El conector ejecuta un conjunto de sentencias en los registros de entrada. El conjunto puede tener cero o más sentencias separadas por signos de punto y coma.

Si el conector genera la sentencia INSERT en tiempo de ejecución, utiliza todas las columnas del enlace de entrada en la cláusula VALUES de la sentencia. Si genera la sentencia UPDATE en tiempo de ejecución, utiliza todas las columnas de clave del enlace de entrada en la cláusula WHERE y todas las columnas no de clave del enlace de entrada en la cláusula SET de la sentencia. Si genera la sentencia DELETE en tiempo de ejecución, utiliza todas las columnas de clave del enlace de entrada en la cláusula WHERE de la sentencia.

El conector puede ejecutar las sentencias en modalidad de proceso por lotes. En esta modalidad, el conector ejecuta las sentencias en un lote de registros en vez de en registros individuales.

El conector ejecuta las sentencias en modalidad de proceso por lotes sólo si se cumplen todas las condiciones siguientes:

- La propiedad **Modalidad de grabación** está establecida en **Insertar**, **Actualizar**, **Suprimir** o **Personalizada**.
- No hay enlaces de rechazo definidos para la etapa.
- La propiedad **Tamaño de lote** está establecida en un valor superior a 1.
- El controlador da soporte a la modalidad de operación de proceso por lotes.

En todos los demás casos, el conector ejecuta las sentencias en registros individuales.

Ejemplo

El siguiente es un ejemplo de una sentencia INSERT definida por el usuario que utiliza la sintaxis de parámetros de enlace de DataStage (**ORCHESTRATE.nombre_parámetro**).

```
INSERT INTO TABLE1 VALUES (ORCHESTRATE.C1, ORCHESTRATE.C2)
```

En este caso, el conector enlaza los parámetros de sentencia a las columnas de enlace de entrada por nombre. El parámetro de sentencia **ORCHESTRATE.C1** se correlaciona con la columna de enlace de entrada C1 y el parámetro de sentencia **ORCHESTRATE.C2** se correlaciona con la columna de enlace de entrada C2.

El siguiente es un ejemplo de una sentencia UPDATE definida por el usuario que mezcla la sintaxis de parámetros de enlace de DataStage (**ORCHESTRATE.nombre_parámetro**) y de JDBC (signos de interrogación).

```
UPDATE TABLE_TEST SET C2 = ORCHESTRATE.C2, C3 = ?
WHERE C1 = ? + ORCHESTRATE.C4 AND C4 = ?
```

En este caso, el conector enlaza los parámetros de sentencia ORCHESTRATE a las columnas de enlace de entrada por nombre, y los parámetros de sentencia de signo de interrogación por posición. En el ejemplo anterior, los parámetros de enlace **ORCHESTRATE.C2** y **ORCHESTRATE.C4** se enlazan a las columnas de enlace de entrada C2 y C4, respectivamente, y el primero, segundo y tercer signos de interrogación se enlazan a la primera, segunda y tercera columnas de enlace de entrada, respectivamente.

Tareas relacionadas:

“Rechazar registros que contienen errores” en la página 11

Cuando una etapa JDBC Connector incluye un enlace de rechazo, los registros que cumplen los criterios especificados se dirigen automáticamente a la etapa de destino en el enlace de rechazo. El proceso continúa para los registros restantes.

“Grabar datos utilizando la etapa JDBC Connector” en la página 8

Puede configurar una etapa JDBC Connector para conectarse a un origen de datos JDBC y grabar datos en él.

Referencia relacionada:

“Condiciones de rechazo” en la página 28

Cuando utilice el conector JDBC para grabar datos en un origen de datos JDBC, puede añadir un enlace de rechazo a la etapa JDBC Connector y enviar los registros rechazados a la etapa conectada al otro extremo del enlace. Las condiciones de rechazo determinan cuándo se rechaza un registro.

Correlaciones de tipos de datos

InfoSphere DataStage da soporte a tipos de datos distintos de los tipos de datos Java y JDBC. Para realizar las operaciones, el conector debe correlacionar los tipos de datos en función del escenario en el que se utilice el conector.

Correlaciones de tipos de datos de InfoSphere DataStage con tipos de datos Java

El conector correlaciona los tipos de datos de InfoSphere DataStage con tipos de datos Java cuando se utiliza como una etapa en trabajos de DataStage para leer registros del origen de datos JDBC y proporcionarlos en el enlace de salida, o para leer registros del enlace de entrada y grabarlos en el origen de datos JDBC.

El conector correlaciona los tipos de datos DataStage con tipos de datos Java en los siguientes escenarios:

- Cuando el conector lee registros del origen de datos JDBC, el conector recupera los valores de campo del controlador JDBC como valores de tipo Java. Para determinar el tipo de datos Java a utilizar para el valor que recupera del controlador, el conector comprueba el tipo de datos DataStage de la columna correspondiente en el enlace.
- Cuando el conector graba registros en el origen de datos JDBC, el conector proporciona los valores de campo al controlador JDBC como valores de tipo

Java. Para determinar el tipo de datos Java a utilizar para el valor que proporciona al controlador, el conector comprueba el tipo de datos DataStage de la columna correspondiente en el enlace.

La siguiente tabla muestra las reglas de correlación entre los tipos de datos de InfoSphere DataStage y los tipos de datos Java.

Nota: Para algunos tipos de datos de InfoSphere DataStage, la tabla muestra más de un tipo de datos Java. En esos casos, el conector elige el tipo de datos Java que determina que describe con más exactitud la columna de origen de datos JDBC correspondiente.

Tabla 12. Tipos de datos de InfoSphere DataStage y sus correspondientes tipos de datos Java

Tipos de datos de InfoSphere DataStage	Tipos de datos Java
BigInt	long
BigInt (Sin signo)	java.lang.String
Binary	byte[]
Bit	boolean
Char	java.lang.String
Char (Unicode)	java.lang.String
Date	java.sql.Date
Decimal	java.math.BigDecimal
Double	double
Float	float
Integer	int
Integer (Sin signo)	long
LongNVarChar	java.sql.NClob, java.sql.Clob, java.lang.String
LongVarBinary	java.sql.Blob, byte[]
LongVarChar	java.sql.NClob, java.sql.Clob, java.lang.String
LongVarChar (Unicode)	java.sql.NClob, java.sql.Clob, java.lang.String
NChar	java.lang.String
Numeric	java.math.BigDecimal
NVarChar	java.lang.String
Real	float
SmallInt	short
SmallInt (Sin signo)	int
Time	java.sql.Time
Time (Microsegundos)	java.sql.Time
Timestamp	java.sql.Timestamp
Timestamp (Microsegundos)	java.sql.Timestamp
TinyInt	byte
TinyInt (Sin signo)	short
VarBinary	byte[]
VarChar	java.lang.String

Tabla 12. Tipos de datos de InfoSphere DataStage y sus correspondientes tipos de datos Java (continuación)

Tipos de datos de InfoSphere DataStage	Tipos de datos Java
VarChar (Unicode)	java.lang.String
Desconocido	java.lang.String

Correlaciones de tipos de datos de InfoSphere DataStage con tipos de datos nativos

Cuando el conector está configurado para crear la tabla de destino y generar la sentencia CREATE TABLE en tiempo de ejecución, correlaciona los tipos de datos de InfoSphere DataStage de las columnas de enlace de datos con los tipos de datos nativos del origen de datos. Luego incluye los tipos de datos nativos en las definiciones de columna de la sentencia generada.

Para determinar el tipo de datos nativo que especificar en la sentencia CREATE TABLE para cada columna de enlace de entrada, el conector primero correlaciona el tipo de datos DataStage de la columna de enlace con el tipo de datos JDBC. A continuación, realiza una consulta al controlador JDBC para determinar el tipo de datos nativo que corresponde a ese tipo de datos JDBC.

La siguiente tabla muestra las reglas de correlación entre los tipos de datos de InfoSphere DataStage y los tipos de datos JDBC.

Nota: Para algunos tipos de datos de InfoSphere DataStage, la tabla muestra más de un tipo de datos JDBC. En esos casos, el conector realiza una consulta al controlador JDBC para determinar el tipo de datos nativo, empezando por el primer tipo de datos JDBC de la lista y, en caso de que el controlador informe de que no hay ningún tipo de datos nativo que corresponda al mismo, el conector continúa realizando consultas al controlador para los restantes tipos de datos JDBC de la lista. Si el conector llega al último tipo de datos JDBC de la lista y el controlador sigue informando de que no hay ningún tipo de datos nativo que corresponda al mismo, el conector incluye el tipo de datos JDBC literal en la sentencia. En esos casos, es probable que sean necesarias modificaciones manuales de la sentencia CREATE TABLE generada.

Tabla 13. Tipos de datos JDBC y sus correspondientes tipos de datos de InfoSphere DataStage

Tipo de datos de InfoSphere DataStage	Tipo de datos JDBC
BigInt	BIGINT
BigInt (Sin signo)	BIGINT
Binary	BINARY, VARBINARY
Bit	BIT, BOOLEAN, TINYINT, SMALLINT, INTEGER
Char	CHAR, VARCHAR
Char (Unicode)	NCHAR, NVARCHAR, CHAR
Date	DATE
Decimal	DECIMAL, NUMERIC
Double	DOUBLE, FLOAT
Float	FLOAT, REAL, DOUBLE
Integer	INTEGER

Tabla 13. Tipos de datos JDBC y sus correspondientes tipos de datos de InfoSphere DataStage (continuación)

Tipo de datos de InfoSphere DataStage	Tipo de datos JDBC
Integer (Sin signo)	INTEGER
LongNVarChar	NCLOB, LONGNVARCHAR, CLOB, LONGVARCHAR, NVARCHAR, VARCHAR
LongVarBinary	BLOB, LONGVARBINARY, VARBINARY
LongVarChar	CLOB, LONGVARCHAR, VARCHAR
LongVarChar (Unicode)	NCLOB, LONGNVARCHAR, CLOB, LONGVARCHAR, NVARCHAR, VARCHAR
NChar	NCHAR, NVARCHAR, CHAR
Numeric	NUMERIC, DECIMAL
NVarChar	NVARCHAR, NCHAR, VARCHAR
Real	REAL, FLOAT, DOUBLE
SmallInt	SMALLINT, INTEGER
SmallInt (Sin signo)	SMALLINT, INTEGER
Time	TIME
Time (Microsegundos)	TIME
Timestamp	TIMESTAMP
Timestamp (Microsegundos)	TIMESTAMP
TinyInt	TINYINT, SMALLINT, INTEGER
TinyInt (Sin signo)	TINYINT, SMALLINT, INTEGER
VarBinary	VARBINARY, BINARY
VarChar	VARCHAR, CHAR
VarChar (Unicode)	NVARCHAR, NCHAR, VARCHAR
Desconocido	NCLOB, LONGNVARCHAR, CLOB, LONGVARCHAR, NVARCHAR, VARCHAR

Correlaciones de tipos de datos JDBC con tipos de datos de InfoSphere DataStage

En un enlace de salida en el que las columnas de DataStage se establecen a partir de los tipos de datos JDBC, la etapa Java Connector correlaciona los tipos de datos JDBC con los tipos de datos de InfoSphere DataStage.

El conector correlaciona los tipos de datos JDBC con tipos de datos de InfoSphere DataStage en los siguientes escenarios:

- El conector tiene un enlace de salida, y el valor **Propagación de columnas en tiempo de ejecución** está seleccionad para el enlace. El conector añade definiciones de columna al enlace en tiempo de ejecución basándose en las definiciones de columna en el origen de datos JDBC. Para cada columna en el origen de datos incluida por la sentencia SELECT que está configurada para el enlace, el conector correlaciona el tipo de datos JDBC con el tipo de datos de InfoSphere DataStage.
- El conector lee, graba o busca datos en el origen de datos, y la propiedad **Informar de discrepancia de esquema** está establecida en **Sí**. El conector intenta determinar si las definiciones de columna en el origen de datos son compatibles

con las definiciones de columna en el enlace. Si el conector detecta una discrepancia, registra un mensaje de aviso.

- En InfoSphere Metadata Asset Manager (IMAM), el conector importa definiciones de tabla del origen de datos JDBC. Cuando el conector obtiene una definición de tabla del origen de datos JDBC, las columnas en esa definición de tabla son de tipos de datos JDBC, tal como ha informado el controlador JDBC. Antes de que el conector pueda guardar esta definición de tabla en el repositorio de metadatos, el conector debe correlacionar los tipos de datos de columna JDBC con tipos de datos de columna de DataStage.

La siguiente tabla muestra las reglas de correlación entre los tipos de datos JDBC y los tipos de datos de InfoSphere DataStage.

Tabla 14. Tipos de datos JDBC y sus correspondientes tipos de datos de InfoSphere DataStage

Tipo de datos JDBC	Tipo de datos de InfoSphere DataStage
ARRAY	VarChar
BIGINT	BigInt
BINARY	Binary
BIT	Bit
BLOB	LongVarBinary
BOOLEAN	Bit
CHAR	Char
CLOB	LongVarChar
DATALINK	VarChar
DATE	Date
DECIMAL	Decimal
DISTINCT	VarChar
DOUBLE	Double
FLOAT	Double
INTEGER	Integer
JAVA_OBJECT	VarChar
LONGNVARCHAR	LongNVarChar
LONGVARBINARY	LongVarBinary
LONGVARCHAR	LongVarChar
NCHAR	NChar
NCLOB	LongNVarChar
NULL	VarChar
NUMERIC	Numeric
NVARCHAR	NVarChar
OTHER	VarChar
REAL	Real
REF	VarChar
ROWID	VarChar
SMALLINT	SmallInt

Tabla 14. Tipos de datos JDBC y sus correspondientes tipos de datos de InfoSphere DataStage (continuación)

Tipo de datos JDBC	Tipo de datos de InfoSphere DataStage
SQLXML	VarChar
STRUCT	VarChar
TIME	Time
TIMESTAMP	Timestamp (Microsegundos)
TINYINT	TinyInt
VARBINARY	VarBinary
VARCHAR	VarChar
cualquier otro tipo de datos	VarChar

Cuando la propiedad **Generar todas las columnas como Unicode** está establecida en **Sí**, los valores Char, VarChar y LongVarChar en la segunda columna de la tabla cambian a NChar, NVarChar y LongNVarChar, respectivamente.

Tareas relacionadas:

“Importar metadatos JDBC” en la página 3

Antes de diseñar trabajos que utilicen el conector JDBC para leer, grabar o buscar datos, puede utilizar InfoSphere Metadata Asset Manager para importar los metadatos que representan tablas y vistas en el origen de datos JDBC. Los metadatos importados se guardan en el repositorio de metadatos.

Referencia relacionada:

“Propagación de columnas en tiempo de ejecución” en la página 24

Puede configurar una etapa JDBC para añadir automáticamente columnas al enlace de salida en tiempo de ejecución.

“Propiedad Informar de discrepancia de esquema” en la página 20

Utilice la propiedad **Informar de discrepancia de esquema** para comparar las definiciones de columna en el enlace con las definiciones de columna en el origen de datos.

Optimización del rendimiento

Puede configurar diversos valores en la etapa JDBC Connector que pueden optimizar su rendimiento.

Valores de controlador

Familiarícese con los requisitos, posibilidades y opciones de configuración del controlador y ajuste los valores para mejorar el rendimiento.

Algunos controladores JDBC permiten especificar propiedades de conexión personalizadas en la serie de conexión URL, que se especifica en la propiedad de conector **URL**.

Algunos controladores permiten especificar propiedades de conexión personalizadas, enumerándolas en formato nombre=valor en la propiedad **Atributos**. Si se especifican opciones de esta manera, cada opción debe especificarse en una línea aparte.

El controlador también puede requerir que la máquina virtual Java (JVM) esté configurada con un conjunto determinado de opciones para proporcionar ventajas de rendimiento. Puede especificar opciones de JVM personalizadas para utilizarlas

en trabajos de InfoSphere DataStage cuando la JVM se inicializa en cada nodo de proceso para la etapa. La propiedad **Valores de Java** se utiliza para esa finalidad.

Importación de metadatos

Cuando utilice IBM InfoSphere Metadata Asset Manager para importar metadatos, puede mejorar el rendimiento importando todo el esquema en lugar de objetos de datos individuales. Los métodos de importación disponibles dependen de la implementación del controlador JDBC y del número de objetos que se van a importar.

Cuando utilice IBM InfoSphere Metadata Asset Manager para importar metadatos, puede importar todos los objetos de datos que contiene el esquema o seleccionar objetos de datos individuales. Cuando el conector JDBC importa objetos de datos individuales de un esquema, solicita a los metadatos un objeto a la vez. Este es el método óptimo cuando desee importar un pequeño subconjunto del esquema global.

Sin embargo, si selecciona un gran subconjunto del esquema, el rendimiento es mejor si importa el esquema completo.

Por ejemplo, si un esquema contiene 50 tablas y selecciona 30 tablas del esquema para importarlas, puede que sea más rápido importar todo el esquema. Si importa 15 tablas, puede que sea más rápido importar solamente las 15 tablas en lugar de todo el esquema.

Tareas relacionadas:

“Importar metadatos JDBC” en la página 3

Antes de diseñar trabajos que utilicen el conector JDBC para leer, grabar o buscar datos, puede utilizar InfoSphere Metadata Asset Manager para importar los metadatos que representan tablas y vistas en el origen de datos JDBC. Los metadatos importados se guardan en el repositorio de metadatos.

Valores de la propiedad Longitud

Asegurarse de que todas las columnas en el enlace tengan la propiedad **Longitud** establecida en un valor concreto puede mejorar el rendimiento.

Para las columnas de enlace Char, VarChar, NChar, NVarChar, Binary y VarBinary que no tienen la propiedad **Longitud** establecida, el conector utiliza el valor especificado en la propiedad **Longitud predeterminada para columnas** como la longitud de columna. Para las columnas de enlace LongVarChar, LongNVarChar y LongVarBinary que no tienen la propiedad **Longitud** establecida, el conector utiliza el valor especificado en la propiedad **Longitud predeterminada para columnas largas** como la longitud de columna.

Para mejorar el rendimiento, especifique un valor para la propiedad **Longitud** para todas las columnas. Especifique el valor más pequeño que siga siendo suficientemente grande para acomodar los datos reales representados por la columna. Cuando se especifica un valor para la propiedad **Longitud**, el conector asigna de forma óptima los recursos, como el almacenamiento de memoria, para intercambiar valores de datos con el controlador JDBC y con la infraestructura de InfoSphere DataStage.

Conversiones de juego de caracteres

La eliminación de conversiones de juego de caracteres en la etapa puede mejorar el rendimiento.

Cuando los enlaces a la etapa JDBC Connector tienen columnas Char, VarChar y LongVarChar para las que la propiedad **Ampliado** no está establecida en Unicode, el conector realiza la conversión de juego de caracteres. Cuando el conector pasa los valores al controlador JDBC, el conector los convierte de la codificación de juego de caracteres especificada en la propiedad **Juego de caracteres para columnas no Unicode** a la codificación de juego de caracteres UTF-16.

Para impedir que el conector realice las conversiones de juego de caracteres en los datos de caracteres, utilice columnas NChar, NVarChar y LongNVarChar en los enlaces, en lugar de columnas Char, VarChar y LongVarChar. O bien, puede dejar las columnas Char, VarChar y LongVarChar en los enlaces, pero establecer la propiedad **Ampliado** en Unicode. El conector intercambia entonces los valores con el controlador como valores UTF-16.

Otras etapas del trabajo pueden requerir columnas Char, VarChar y LongVarChar en sus enlaces, con la propiedad **Ampliado** para las columnas que no están establecidas en Unicode. En este caso, puede seguir utilizando las columnas NChar, NVarChar y LongNVarChar en los enlaces de la etapa JDBC Connector y realizar la correlación con o desde columnas Char, VarChar y LongVarChar en una etapa situada en sentido descendente o ascendente de la etapa JDBC Connector.

Compare el rendimiento al convertir juegos de caracteres en una etapa distinta de la etapa JDBC Connector con el rendimiento al convertir juegos de caracteres en la etapa JDBC Connector, y elija la opción que mejor se adapte a su entorno.

Valores de tamaño de captación

El ajuste del valor de la propiedad **Tamaño de captación** puede mejorar el rendimiento cuando se leen registros del origen de datos.

El conector proporciona el valor de la propiedad **Tamaño de captación** al controlador. Este valor es una indicación al controlador de cuántos registros se han de recuperar del origen de datos en un único viaje de ida y vuelta al origen de datos. El conector siempre solicita los registros al controlador de uno en uno, y el controlador puede utilizar la opción de captación previa para poder proporcionar algunos de los registros al conector localmente.

El valor predeterminado para la propiedad **Tamaño de captación** es 0. El valor predeterminado permite obtener un rendimiento óptimo en muchos de los casos. También puede establecer la propiedad **Tamaño de captación** en un valor específico y comprobar los resultados para ver si son mejores que los resultados con el tamaño de captación predeterminado.

Cambiar el valor de la propiedad **Tamaño de captación** puede mejorar el rendimiento cuando la etapa lee muchos registros, por ejemplo, cientos de miles de registros o más. Ejecute el trabajo con los valores de tamaño de captación de 20, 200, 2000 y 20000 y observe las tendencias de rendimiento para determinar el valor óptimo. Si ninguno de los valores de tamaño de captación mejora el rendimiento, utilice el valor predeterminado.

Esperas de bloqueo

La elección del tipo de particionamiento para los registros en el enlace de entrada puede afectar el rendimiento del conector.

Cuando la etapa está configurada para grabar registros en el origen de datos en paralelo en varios nodos de proceso, puede configurar el particionamiento de registros en el enlace de entrada para reducir o evitar esperas de bloqueo en operaciones de grabación simultáneas.

Por ejemplo, si ha configurado la etapa para que se ejecute en modalidad de grabación Actualizar y luego insertar, el conector primero intenta ejecutar la sentencia UPDATE en cada registro de entrada. En este ejemplo, la cláusula WHERE hace referencia a las columnas de clave primaria en la tabla de destino. Si el resultado de esa sentencia es cero actualizaciones, el conector ejecuta la sentencia INSERT para el registro. Si el conjunto de datos de entrada contiene varios registros con los mismos valores para las columnas de clave, la sentencia UPDATE se ejecuta para cada uno de los registros. Si dos nodos de proceso intentan esta operación en dos registros diferentes con los mismos valores de clave, en las transacciones que están activas al mismo tiempo, el nodo de proceso que primero ha ejecutado la sentencia adquiere el bloqueo en la fila correspondiente en el origen de datos. El nodo de proceso retiene el bloqueo hasta que finaliza su transacción. El otro nodo de proceso debe esperar a que se libere el bloqueo.

Para asegurar que todos los registros de entrada que tengan los mismos valores de campo de clave sean procesados por el mismo nodo de proceso, configure el valor **Tipo de partición** para el enlace de manera que esos registros se direccionen al mismo nodo de proceso. El **Tipo de partición** se especifica en la página Particionamiento del enlace en el editor de etapas. Por ejemplo, puede establecer la propiedad **Tipo de partición en Hash**. Para las columnas de clave hash, elija las columnas de clave de enlace de entrada a las que haga referencia la cláusula WHERE. El particionamiento de registros en el enlace de entrada evita esperas de bloqueo y asegura que la operación UPDATE se ejecute en los registros con los valores de clave coincidentes en el mismo orden en que llegan al enlace de entrada.

Varios enlaces

Aunque el uso de varios enlaces de entrada es un práctico mecanismo para reducir el número de instancias de JVM y de conexiones con orígenes de datos, también puede afectar negativamente al rendimiento del conector.

El conector da soporte a varios enlace de entrada. Cada uno de los enlaces se puede configurar para ejecutar operaciones distintas en el origen de datos. El orden en el que se procesan los registros en los enlaces se controla mediante la propiedad **Ordenación de registros**. El proceso de los registros para todos los enlaces de entrada tiene lugar en una única instancia de JVM y todos los enlaces de entrada comparten una conexión con el origen de datos.

Cuando la etapa está configurada para ejecutarse en modalidad paralela, se crea una instancia de JVM y una conexión con el origen de datos en cada nodo de proceso. El inconveniente de utilizar varios enlaces de entrada es que podría dar lugar a un aumento en el consumo de los ciclos de procesador por parte del conector.

Debe sopesar este efecto negativo y la ventaja de compartir la instancia de JVM y la conexión de origen de datos entre los enlaces, y elegir la configuración que se mejor se adapte a sus necesidades.

Enlaces de rechazo

Los enlaces de rechazo son un práctico mecanismo para manejar registros de enlace de entrada erróneos, pero al mismo tiempo su uso puede afectar negativamente al rendimiento del conector.

Los enlaces de rechazo manejan los registros de enlace de entrada que han provocado un error en la operación de grabación. Cuando se utilizan enlaces de rechazo, en lugar de detener el trabajo, la etapa se puede configurar para enviar los registros erróneos a los enlaces de rechazo y seguir procesando los registros de entrada. Una vez completado el trabajo, puede comprobar los registros rechazados para determinar acciones adicionales.

Sin embargo, los enlaces de rechazo pueden afectar negativamente al rendimiento del conector. Cuando la etapa con enlaces de rechazo no satisfaga sus requisitos de rendimiento, considere la posibilidad de sustituir los enlaces de rechazo por un mecanismo alternativo para manejar los registros de entrada erróneos. Por ejemplo, si se utilizaron enlaces de rechazo para manejar registros que violaron restricciones NULL o CHECK de la tabla de destino, compruebe los registros para esas violaciones y manéjelos en las etapas de proceso en sentido ascendente de la etapa de conector.

De forma alternativa, compruebe si hay mecanismos proporcionados por el origen de datos de programa de fondo que permitan el manejo de registros erróneos en el lado del origen de datos. Por ejemplo, podría indicar al origen de datos que redireccione los registros a una tabla de errores dedicada.

Confirmaciones de transacción

Reducir la frecuencia de las confirmaciones de transacción al leer registros del origen de datos puede mejorar el rendimiento.

Cuando el conector está configurado para leer registros del origen de datos y enviar los registros al enlace de salida, confirma la transacción después de cada N registros, donde N es el valor que se especifica en la propiedad **Recuento de registros**.

Si la sentencia ejecutada por la etapa no produce cambios en el origen de datos y no es necesario enviar los registros al enlace de salida en oleadas, considere la posibilidad de establecer la propiedad **Recuento de registros** en 0. El valor 0 para esta propiedad indica que la transacción se confirma una sola vez después de que se hayan leído todos los registros del origen de datos.

Proceso por lotes

La utilización del proceso por lotes cuando el conector graba registros en el origen de datos puede mejorar el rendimiento.

El conector da soporte a la modalidad de operación de proceso por lotes cuando graba registros en el origen de datos. En este tipo de operación, el conector proporciona varios registros al controlador JDBC para incluirlos en un lote y luego solicita al controlador que envíe el lote al origen de datos. Este proceso puede mejorar el rendimiento. El tamaño de los lotes se especifica mediante la propiedad **Tamaño de lote**.

Cuando la propiedad **Recuento de registros** se establece en 0, la transacción se confirma después de que todos los registros se hayan grabado en el origen de datos. Establecer la propiedad **Recuento de registros** en 0 puede dar como resultado transacciones de larga ejecución, dependiendo del número de registros de

entrada. No obstante, si se establece la propiedad **Recuento de registros** en un valor pequeño, se producen confirmaciones de transacción frecuentes.

Si reduce el tamaño de lote a un valor pequeño, las ventajas que ofrece el proceso por lotes pueden reducirse o perderse. Tenga en cuenta que cuando el valor de recuento de registros no es 0, el conector ajusta automáticamente el valor de tamaño de lote especificado, si es necesario, para asegurar que el valor de recuento de registros sea un múltiplo del mismo.

Modalidades de grabación

Las modalidades de grabación Insertar y luego actualizar y Actualizar y luego insertar pueden dar como resultado un rendimiento diferente para los mismos registros de entrada.

Puede mejorar el rendimiento minimizando el número total de sentencias que se ejecutan para cada registro de entrada. Para reducir el número total de sentencias que se ejecutan en cada registro de entrada durante la ejecución del trabajo, seleccione la modalidad de grabación en función de su conocimiento sobre los datos de entrada.

Para un registro de entrada al ejecutar la modalidad de grabación Insertar y luego actualizar, el conector primero intenta ejecutar la sentencia de inserción. Si eso falla, el conector ejecuta la sentencia de actualización. Al ejecutar la modalidad de grabación Actualizar y luego insertar, el conector primero intenta ejecutar la sentencia de actualización. Si eso falla, el conector ejecuta la sentencia de inserción.

Puede mejorar el rendimiento seleccionando la modalidad de grabación en función de los datos de entrada.

Si se espera que la mayoría de los registros de entrada se inserten, y sólo se actualizará un pequeño porcentaje de ellos, la modalidad de grabación Insertar y luego actualizar puede tener mejor rendimiento.

Si la mayoría de los registros de entrada ya están en el origen de datos y deben actualizarse, y sólo se insertará un pequeño porcentaje de los registros de entrada, la modalidad de grabación Actualizar y luego insertar puede tener mejor rendimiento.

Resolución de problemas de la etapa JDBC Connector

Hay varios errores comunes que son específicos de la etapa JDBC Connector.

No se puede acceder al archivo de configuración

El archivo `isjdbc.config` está configurado pero un error indica que no se puede acceder al archivo de configuración.

Síntomas

Un error indica que no se puede acceder al archivo de configuración.

Resolución del problema

Realice los pasos siguientes:

- Asegúrese de que el archivo de configuración está guardado en el directorio DSEngine de la instalación de la capa de motor de InfoSphere Information Server.
 - En los sistemas operativos Windows, la ubicación predeterminada del directorio DSEngine es `C:\IBM\InformationServer\Server\DSEngine`.

- En los sistemas operativos Linux y UNIX, la ubicación predeterminada es `/opt/IBM/InformationServer/Server/DSEngine`.
- Asegúrese de que el nombre del archivo es `isjdbc.config`, con todos los caracteres en minúsculas y sin ningún carácter de espacio en blanco inicial y final.
- Asegúrese de que el archivo está guardado en el sistema de la capa de motor.
- Asegúrese de que cuando la capa de motor conste de varios hosts, el archivo sea accesible desde la misma ubicación en todos los hosts. Puede hacer una copia del archivo en la misma ubicación DSEngine en cada uno de los hosts, o utilizar una ubicación de red DSEngine compartida a la que tengan acceso todos los hosts.
- Asegúrese de que el archivo tiene el acceso de lectura para el usuario de InfoSphere DataStage y el usuario Administrador local (Windows) o el usuario root del sistema (Linux y UNIX). En Linux y UNIX, compruebe los permisos del archivo `.odbc.ini` que está almacenado en la misma ubicación y establezca los mismos permisos en el archivo `isjdbc.config`.
- Asegúrese de que el contenido del archivo de configuración está en el formato correcto.

CREATE TABLE no se ha ejecutado satisfactoriamente

El conector está configurado para crear la tabla de destino automáticamente, pero la sentencia CREATE TABLE que se ha generado no se ha ejecutado satisfactoriamente.

Síntomas

Dependiendo de las definiciones de columna en el enlace de entrada, el controlador y el origen de datos, es posible que el conector no pueda generar una sentencia CREATE TABLE válida.

Causas

Para generar la sentencia CREATE TABLE, el conector primero correlaciona cada definición de columna en el enlace de entrada del tipo de datos de InfoSphere DataStage data con el tipo de datos JDBC más parecido. A continuación, el conector realiza una consulta al controlador para determinar el tipo nativo del origen de datos que corresponde a ese tipo de datos JDBC.

Dependiendo del tipo de datos del enlace de entrada, el conector puede probar varios tipos de datos JDBC diferentes. Sin embargo, en algunos casos, el controlador no indica un tipo de datos nativo para ninguno de los tipos de datos JDBC que el conector ha presentado. En esos casos, el conector utiliza el primer tipo de datos JDBC en la sentencia CREATE TABLE. Esto, a su vez, puede hacer que la sentencia no se ejecute. El conector graba el texto completo de la sentencia CREATE TABLE generada en el registro de trabajo.

Resolución del problema

Para corregir el error, realice uno o más de los pasos siguientes:

- Utilice tipos de datos diferentes para las columnas de enlace de entrada. Asegúrese de que las definiciones de columna del enlace de entrada sean válidas para los valores de datos reales representados por esas columnas cuando se ejecuta el trabajo.
- Especifique la sentencia CREATE TABLE manualmente. Puede empezar con la sentencia CREATE TABLE generada copiándola del registro de trabajo en la propiedad **Sentencia Create table** en el editor de etapas.

Para obtener más información, consulte la documentación para el origen de datos donde desea crear la tabla. Localice los tipos de datos nativos que se utilizarán para las columnas necesarias en la sentencia y luego especifique esos tipos de datos directamente en la sentencia.

Asegúrese de que los tipos de datos nativos que especifique para las columnas de tabla sean válidos para los valores de datos reales que se graban en esas columnas cuando se ejecuta el trabajo.

Para obtener más información sobre la correlación, consulte el siguiente tema: “Correlaciones de tipos de datos de InfoSphere DataStage con tipos de datos nativos” en la página 36

El trabajo falla e informa de errores de recorte de datos o de conversión de juego de caracteres

Cuando el conector lee valores de o graba valores en una columna basada en caracteres en el origen de datos utilizando una columna basada en caracteres en el enlace, el trabajo finaliza de forma anómala e informa de errores de recorte de datos o de conversión de juego de caracteres, aunque la propiedad Longitud de la columna en el enlace coincida con la longitud de la columna en el origen de datos.

Síntomas

El trabajo falla e informa de errores de recorte de datos o de conversión de juego de caracteres.

Causas

Para las columnas Char, VarChar, LongVarChar, NChar, NVarChar y LongNVarChar en el enlace, el conector intercambia valores con el controlador JDBC como valores Unicode Java™ de doble byte.

Para las columnas NChar, NVarChar y LongNVarChar y las columnas Char, VarChar y LongVarChar para las que la propiedad **Ampliado** está establecida en Unicode, el conector intercambia valores con la infraestructura de InfoSphere DataStage como valores Unicode de doble byte.

Para las columnas Char, VarChar y LongVarChar para las que la propiedad **Ampliado** no está establecida en Unicode, el conector intercambia valores con la infraestructura de InfoSphere DataStage según la codificación de juego de caracteres especificada para la etapa en la propiedad **Juego de caracteres para columnas no Unicode**. Cuando esta propiedad se establece en el valor **Predeterminado**, se utiliza la codificación de juego de caracteres del entorno local del sistema. Cuando esta propiedad se establece en el valor **Personalizado**, la codificación de juego de caracteres que se debe utilizar se especifica en la propiedad **Nombre de juego de caracteres**.

La propiedad Longitud para las columnas del enlace se mide en unidades de doble byte para las columnas NChar, NVarChar y LongNVarChar y las columnas Char, VarChar y LongVarChar para las que la propiedad **Ampliado** está establecida en Unicode, y se mide en bytes para las columnas Char, VarChar y LongVarChar para las que la propiedad **Ampliado** no está establecida en Unicode. Estas diferencias pueden dar lugar a definiciones de columna incompatibles incluso cuando se especifique la misma Longitud para la columna en el enlace y la columna en el origen de datos.

Por ejemplo, una columna en el origen de datos puede estar definida como columna VARCHAR(10) y ser capaz de almacenar 10 valores Unicode de doble

byte. Suponga que la columna correspondiente en el enlace está definida como columna VarChar(10) y que la propiedad **Ampliado** no está establecida en Unicode. Cuando el conector capta el valor de esta columna en el origen de datos, lo capta como una serie Unicode Java que consta de 10 caracteres Unicode de doble byte.

Suponga que el conector debe convertir este valor a la codificación UTF-8, ya sea porque la propiedad **Juego de caracteres para columnas no Unicode** estaba establecida en Predeterminado y UTF-8 es la codificación de juego de caracteres en el entorno local del sistema actual, o porque la propiedad **Juego de caracteres para columnas no Unicode** estaba establecida en el valor **Personalizado** y la codificación de juego de caracteres UTF-8 estaba especificada en la propiedad **Nombre de juego de caracteres**. Suponga además que algunos de los caracteres de doble byte en el valor captado dan como resultado caracteres UTF-8 de 2, 3 o 4 bytes. El valor convertido en ese caso puede requerir de 10 a 40 bytes de almacenamiento. Si requiere más de 10 bytes de almacenamiento, el conector detecta que la Longitud de la columna de enlace no es suficiente e informa de un error.

Resolución del problema

Para corregir el error mencionado en el ejemplo anterior, realice una de las siguientes acciones:

- Cambie el tipo de datos de la definición de columna en el enlace a NVarChar(10).
- Establezca la propiedad **Ampliado** para la columna en Unicode.
- Aumente la Longitud en la definición de columna a 40 para acomodar todos los valores.

El trabajo falla y se han detectado columnas no utilizadas

Cuando el conector detecta una columna en el enlace que no es utilizada por las sentencias que el conector ejecuta para los registros de ese enlace, el conector informa de un error para la columna desaparejada detectada.

Síntomas

El conector informa de un error para la columna desaparejada en el enlace y el trabajo finaliza de forma anómala.

Resolución del problema

La resolución de este error depende de si el conector está configurado para leer datos de un origen de datos JDBC o para grabar datos en un origen de datos JDBC

Si el conector está configurado para leer filas del origen de datos, asegúrese de que la sentencia SELECT que el conector utiliza para leer las filas incluya una columna en su lista de selección que corresponda a la columna en el enlace. O bien, elimine la columna desaparejada del enlace.

La siguiente tabla incluye ejemplos de escenarios y los problemas respectivos que puede encontrar cuando el conector está configurado para leer datos de un origen de datos JDBC y la resolución de los problemas.

Sentencia SELECT	Columnas de enlace de salida	Problema	Resoluciones
SELECT C1, C3 FROM TABLE1	C1, C2 y C3	La columna de enlace C2 se trata como una columna desaparejada y produce un error	Realice uno de los pasos siguientes: <ul style="list-style-type: none"> • Cambie la sentencia a SELECT C1, C2, C3 FROM TABLE1 • Elimine la columna C2 del enlace de salida.
La sentencia SELECT se genera automáticamente y la tabla de origen TABLE1 contiene las columnas C1, C3 y C4	C1, C2 y C3	El conector no incluye la columna de tabla C4 en la sentencia generada y no capta datos para ella porque determina que no hay una columna correspondiente definida en el enlace de salida. Sin embargo, el conector incluye la columna C2 en la sentencia generada porque está presente en el enlace de salida y eso produce un error, ya que la tabla no tiene la columna con ese mismo nombre.	Realice uno de los pasos siguientes: <ul style="list-style-type: none"> • Añada la columna C2 a la tabla. • Elimine la columna C2 del enlace de salida.

Si el conector está configurado para grabar registros del enlace de entrada en el origen de datos, a cada columna del enlace de entrada le debe hacer referencia una o más sentencias SQL que la etapa utiliza para grabar registros en el origen de datos. Si no se hace referencia a una columna del enlace de entrada, el comportamiento depende del valor de la propiedad **Descartar campos que no coincidan**. Cuando esta propiedad se establece en **No**, el conector informa de un error y el trabajo finaliza de forma anómala. Cuando la propiedad se establece en **Sí**, el conector emite un mensaje informativo para cada columna de enlace de entrada no utilizada para indicar que la columna se ignora, y el trabajo continúa.

La siguiente tabla incluye ejemplos de escenarios y los problemas respectivos que puede encontrar cuando el conector está configurado para grabar datos en un origen de datos JDBC y la resolución de los problemas.

Modalidades de grabación	Columnas de enlace de entrada	Problema	Resoluciones
<p>El conector está configurado para ejecutarse en modalidad de grabación Insertar y para ejecutar la sentencia</p> <pre>INSERT INTO TABLE1(C1, C2, C3) VALUES (ORCHESTRATE.C1, 'job 1', ORCHESTRATE.C3)</pre>	C1, C2 y C3	<p>El conector determina que a la columna de enlace de entrada C2 no le hace referencia la sentencia INSERT y actúa según el valor de la propiedad</p> <p>Descartar campos que no coincidan.</p>	<p>Realice uno de los pasos siguientes:</p> <ul style="list-style-type: none"> • Cambie la sentencia INSERT a <pre>INSERT INTO TABLE1(C1, C2, C3) VALUES (ORCHESTRATE.C1, ORCHESTRATE.C2, ORCHESTRATE.C3)</pre> • Elimine la columna C2 del enlace de entrada.
<p>El conector está configurado para ejecutarse en modalidad de grabación Insertar y generar automáticamente la sentencia INSERT que se utilizará para insertar registros en la tabla TABLE1 con las columnas C1, C3 y C4</p>	C1, C2 y C4	<p>El conector no incluye la columna C4 en la sentencia INSERT generada porque no hay una columna correspondiente en el enlace de entrada. Dado que la columna C2 está en el enlace de entrada y no tiene una columna correspondiente en la tabla de destino, el conector trata esta columna de enlace de entrada como una columna desaparejada.</p>	<p>Realice uno de los pasos siguientes:</p> <ul style="list-style-type: none"> • Añada la columna C2 a la tabla. • Elimine la columna C2 del enlace de entrada.

El conector no ha detectado e informado de todas las discrepancias de esquema

El conector está configurado para informar de las discrepancias entre las definiciones de columna en el enlace y las definiciones de columna en el origen de datos JDBC. Sin embargo, el conector no ha detectado e informado de todas las discrepancias.

Síntomas

El conector intenta determinar los tipos de datos JDBC de las columnas en el origen de datos que corresponden a las columnas en el enlace. Si el conector determina que los tipos de datos son incompatibles, emite el aviso de discrepancia de esquema.

El controlador, el origen de datos y el escenario de uso de la etapa pueden impedir que se produzcan comprobaciones de discrepancias de tipo de datos.

Causas

El conector comprueba si hay una discrepancia de esquema siempre que sea posible. Sin embargo, no siempre puede detectar todas las discrepancias.

Resolución del problema

Esta característica debe utilizarse solamente como ayuda para configurar las definiciones de columna en el enlace. La comprobación de discrepancia de esquema se realiza durante la fase de configuración del trabajo.

Independientemente de la comprobación de discrepancia de esquema, el conector comprueba si hay más discrepancia de esquema durante la fase de proceso del trabajo. La comprobación adicional ocurre cuando el conector proporciona valores al controlador JDBC o recibe valores del controlador JDBC. Si entonces el conector detecta problemas de recorte de datos o de conversión de juego de caracteres, el conector informa del error y el trabajo finaliza de forma anómala.

No se ha encontrado un controlador adecuado al utilizar la etapa JDBC Connector

Se producen errores cuando la vía de acceso de clases del controlador no se ha especificado correctamente en el archivo de configuración JDBC.

Síntomas

Se produce un error que indica que no se ha encontrado ningún conductor adecuado. El conector no ha podido localizar un controlador adecuado para el valor de URL especificado.

Resolución del problema

Asegúrese de que la información sobre la vía de acceso de clases del controlador y el nombre de la clase de controlador esté incluida en el archivo de configuración `isjdbc.config` como se indica a continuación:

- La entrada `CLASSPATH` en el archivo de configuración debe incluir la vía de acceso de clases totalmente calificada del controlador. Si el controlador consta de más de un archivo JAR o depende de varios archivos JAR, deben incluirse todas las ubicaciones de archivos JAR.

Si el controlador se ha implementado en uno o más archivos JAR, asegúrese de que los archivos JAR especificados están presentes en el sistema. Además, asegúrese de que los archivos JAR tiene los permisos de lectura y ejecución habilitados para el usuario de InfoSphere DataStage y el usuario Administrador local en Windows o el usuario root del sistema en Linux y UNIX.

- Si no está seguro de cuál es la versión JDBC del controlador, compruebe si el archivo `META-INF/services/java.sql.Driver` está presente en el archivador JAR del controlador. Si este archivo está presente y contiene un solo valor, que indica el nombre de clase JDBC Driver, no es necesario especificar este nombre de clase en la entrada `CLASS_NAMES`.

La entrada `CLASS_NAMES` en el archivo de configuración debe incluir el nombre completo de la clase Java del controlador que implementa la interfaz JDBC `java.sql.Driver`. Esta información sólo es necesaria si el controlador se basa en la versión de la API JDBC anterior a la versión 4.0. Para averiguar el nombre de esta clase, consulte la documentación del controlador.

Accesibilidad de los productos

Puede obtener información sobre el estado de accesibilidad de los productos de IBM.

Los módulos de producto y las interfaces de usuario de IBM InfoSphere Information Server no son totalmente accesibles.

Para obtener información sobre el estado de accesibilidad de los productos de IBM, consulte la información de accesibilidad de productos de IBM en http://www.ibm.com/able/product_accessibility/index.html.

Documentación sobre accesibilidad

Se proporciona documentación accesible para los productos en IBM Knowledge Center. IBM Knowledge Center presenta la documentación en formato XHTML 1.0, que se puede ver en la mayoría de navegadores web. Dado que IBM Knowledge Center utiliza XHTML, puede establecer preferencias de visualización en el navegador. Esto también le permite utilizar lectores de pantalla y otras tecnologías de asistencia para acceder a la documentación.

La documentación que está en IBM Knowledge Center se proporciona en archivos PDF, que no son totalmente accesibles.

IBM y la accesibilidad

Consulte el sitio web IBM Human Ability and Accessibility Center para obtener más información sobre el compromiso de IBM con la accesibilidad.

Acceso a la documentación del producto

La documentación se proporciona en diversos formatos: en el IBM Knowledge Center en línea, en un centro de información opcional instalado localmente y como manuales PDF. Puede acceder a la ayuda en línea o instalada localmente directamente desde las interfaces de cliente del producto.

IBM Knowledge Center es el mejor lugar para encontrar la información más actualizada de InfoSphere Information Server. IBM Knowledge Center contiene ayuda para la mayoría de las interfaces del producto, así como documentación completa para todos los módulos de producto de la suite. Puede abrir IBM Knowledge Center desde el producto instalado o desde un navegador web.

Cómo acceder a IBM Knowledge Center

Existen varias maneras de acceder a la documentación en línea:

- Pulse el enlace **Ayuda** en la parte superior derecha de la interfaz de cliente.
- Pulse la tecla F1. Normalmente, la tecla F1 abre el tema que describe el contexto actual de la interfaz de cliente.

Nota: La tecla F1 no funciona en clientes web.

- Escriba la dirección en un navegador web, por ejemplo, cuando no tenga iniciada una sesión en el producto.

Escriba la siguiente dirección para acceder a todas las versiones de la documentación de InfoSphere Information Server:

<http://www.ibm.com/support/knowledgecenter/SSZJPZ/>

Si desea acceder a un tema concreto, especifique el número de versión con el identificador de producto, el nombre del plug-in de documentación y la vía de acceso al tema en el URL. Por ejemplo, el URL para la versión 11.3 de este tema es el siguiente. (El símbolo \Rightarrow indica una continuación de línea):

http://www.ibm.com/support/knowledgecenter/SSZJPZ_11.3.0/=>com.ibm.swg.im.iis.common.doc/common/accessingiidoc.html

Consejo:

El Knowledge Center tiene también un URL corto:

<http://ibm.biz/knowctr>

Para especificar un URL corto a una página de producto, versión o tema específico, utilice un carácter de almohadilla (#) entre el URL corto y el identificador de producto. Por ejemplo, el URL corto a toda la documentación de InfoSphere Information Server es el siguiente URL:

<http://ibm.biz/knowctr#SSZJPZ/>

Y el URL corto al tema anterior para crear un URL ligeramente más corto es el siguiente URL (El símbolo \Rightarrow indica una continuación de línea):

http://ibm.biz/knowctr#SSZJPZ_11.3.0/com.ibm.swg.im.iis.common.doc/=>common/accessingiidoc.html

Cambiar los enlaces de ayuda para que hagan referencia a la documentación instalada localmente

IBM Knowledge Center contiene la versión más actualizada de la documentación. Sin embargo, puede instalar una versión local de la documentación como un centro de información y configurar los enlaces de ayuda para que apunten a él. Un centro de información local es útil si su empresa no proporciona acceso a Internet.

Siga las instrucciones de instalación que vienen con el paquete de instalación del centro de información para instalarlo en el sistema que elija. Después de instalar e iniciar el centro de información, puede utilizar el mandato **iisAdmin** en el sistema de la capa de servicios para cambiar la ubicación de la documentación a la que hacen referencia la tecla F1 y los enlaces de ayuda del producto. (El símbolo ⇒ indica una continuación de línea):

Windows

```
vía_instalación_IS\ASBServer\bin\iisAdmin.bat -set -key ⇒  
com.ibm.iis.infocenter.url -value http://<host>:<puerto>/help/topic/
```

AIX Linux

```
vía_instalación_IS/ASBServer/bin/iisAdmin.sh -set -key ⇒  
com.ibm.iis.infocenter.url -value http://<host>:<puerto>/help/topic/
```

Donde <host> es el nombre del sistema donde está instalado el centro de información y <puerto> es el número de puerto para el centro de información. El número de puerto predeterminado es 8888. Por ejemplo, en un sistema llamado server1.example.com que utilice el puerto predeterminado, el valor del URL sería <http://server1.example.com:8888/help/topic/>.

Obtener la documentación en PDF y en copia impresa

- Los manuales en archivos PDF están disponibles en línea y puede accederse a ellos desde este documento de soporte: <https://www.ibm.com/support/docview.wss?uid=swg27008803&wv=1>.
- También puede solicitar publicaciones de IBM en formato impreso en línea o a través de su representante local de IBM. Para solicitar publicaciones en línea, vaya al Centro de Publicaciones de IBM en <http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>.

Cómo aportar comentarios sobre la documentación del producto

Puede aportar valiosos comentarios en relación a la documentación de IBM.

Sus comentarios ayudarán a IBM a ofrecer información de calidad. Puede utilizar cualquiera de los métodos siguientes para enviar sus comentarios:

- Para proporcionar un comentario acerca de un tema del IBM Knowledge Center que está alojado en el sitio web de IBM, inicie la sesión y pulse el botón **Añadir comentario** en la parte inferior del tema. Los comentarios enviados de esta manera serán visibles para todos los usuarios.
- Para enviar un comentario acerca de un tema del IBM Knowledge Center a IBM y que ningún otro usuario pueda ver, inicie la sesión y pulse en el enlace **Comentarios** en la parte inferior del IBM Knowledge Center.
- Envíe sus comentarios utilizando el formulario de comentarios del lector que encontrará en www.ibm.com/software/awdtools/rcf/.
- Envíe sus comentarios por correo electrónico a comments@us.ibm.com. Incluya el nombre y el número de versión del producto, así como el nombre y el número de pieza de la información (si es pertinente). Si su comentario es sobre un texto específico, incluya la ubicación del texto (por ejemplo, un título, un número de tabla o un número de página).

Enlaces a sitios web que no son de IBM

Este Information Center puede suministrar enlaces o referencias a sitios web y recursos que no son de IBM.

IBM no asume representaciones, garantías, u otros compromisos de ningún tipo sobre cualquier incumplimiento de sitios web ajenos a IBM o recursos de terceros (incluyendo cualquier sitio web de Lenovo) que pueda estar referenciado, accesible desde, o vinculado a cualquier sitio de IBM. Un enlace con un sitio web que no sea de IBM no significa que IBM recomiende el contenido o el uso de dicho sitio web o su propietario. Además, IBM no es una parte o no es responsable de las transacciones en las que pueda entrar con terceros, aun cuando aprenda de dichas partes (o utilice un enlace a tales partes) desde un sitio de IBM. En consecuencia, el cliente tiene conocimiento y acepta que IBM no es responsable de la disponibilidad de tales sitios o recursos externos, ni tampoco del contenido, servicios, productos u otros materiales situados en dichos sitios o recursos o disponibles desde ellos.

Cuando el usuario accede a un sitio web que no es de IBM, incluso a uno que pueda contener el logotipo de IBM, debe comprender que es independiente de IBM y que IBM no controla el contenido de dicho sitio web. Es responsabilidad del usuario tomar las precauciones necesarias para protegerse frente a virus, gusanos, troyanos y otros programas potencialmente destructivos, así como proteger su información como estime oportuno.

Avisos y marcas registradas

Esta información ha sido desarrollada para productos y servicios ofrecidos en los Estados Unidos. Este material puede estar disponible en IBM en otros idiomas. Sin embargo, es posible que deba tener una copia del producto o de la versión del producto en ese idioma para poder acceder al mismo.

Avisos

Es posible que IBM no ofrezca en otros países los productos, servicios o características que se describen en este documento. Póngase en contacto con el representante local de IBM para obtener información acerca de los productos y servicios que actualmente están disponibles en su localidad. Cualquier referencia a un producto, programa o servicio de IBM no implica ni establece 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 infrinja 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 patente pendientes de aprobación que cubran temas tratados en este documento. La posesión de este documento no confiere ninguna licencia sobre dichas patentes. Puede enviar consultas sobre licencias, por escrito, a la siguiente dirección:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 EE.UU.

Para realizar consultas relativas a la información de juego de caracteres de doble byte (DBCS), póngase en contacto con el departamento de propiedad intelectual de IBM de su país o envíe las consultas, por escrito, a:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokio 103-8510, Japón

El párrafo siguiente no se aplica en el Reino Unido ni en ningún otro país en el que las disposiciones en él expuestas sean incompatibles con la legislación local: INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL", SIN GARANTÍAS DE NINGÚN TIPO, NI EXPLÍCITAS NI IMPLÍCITAS, INCLUYENDO, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO INFRACCIÓN, COMERCIALIZACIÓN E IDONEIDAD PARA UN PROPÓSITO DETERMINADO. Algunas legislaciones no contemplan la declaración de limitación de garantías, ni implícitas ni explícitas, en determinadas transacciones, por lo que cabe la posibilidad de que esta declaración no se aplique en su caso.

Esta información puede contener imprecisiones técnicas o errores tipográficos. La información incluida en este documento está sujeta a cambios periódicos, que se

incorporarán en nuevas ediciones de la publicación. IBM puede realizar en cualquier momento mejoras o cambios en los productos o programas descritos en esta publicación sin previo aviso.

Cualquier referencia hecha en esta información a sitios web que no sean de IBM se proporciona únicamente para su comodidad y no debe considerarse en modo alguno como una aprobación de dichos sitios web. Los materiales de estos sitios web no forman parte de los materiales de este producto de IBM y el uso que haga de estos sitios web es de la entera responsabilidad del usuario.

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 licenciatarios de este programa que deseen obtener información acerca del mismo con el fin de: (i) intercambiar la información entre los programas creados independientemente y otros programas (incluido éste) y (ii) utilizar mutuamente la información que se ha intercambiado, deben ponerse en contacto con:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 EE.UU.

Esta información puede estar disponible, sujeta a los términos y condiciones pertinentes, incluido en algunos casos el pago de una cantidad determinada.

IBM proporciona el programa bajo licencia descrito en este documento, y todo el material bajo licencia disponible para el mismo, bajo los términos del Acuerdo de cliente de IBM, el Acuerdo acuerdo internacional de licencia de programa de IBM o cualquier otro acuerdo equivalente entre las partes.

Los datos de rendimiento incluidos en este documento se determinaron en un entorno controlado. Por tanto, los resultados obtenidos en otros entornos operativos pueden variar significativamente. Algunas mediciones pueden haberse efectuado en sistemas a nivel de desarrollo, y no existe ninguna garantía de que dichas mediciones sean las mismas en sistemas de disponibilidad general. Además, es posible que algunas mediciones se hayan estimado mediante extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables a su entorno específico.

La información relacionada con productos no de IBM se ha obtenido de los suministradores de dichos productos, de sus anuncios publicados o de otras fuentes de información pública disponibles. IBM no ha probado dichos productos y no puede confirmar la precisión del rendimiento, la compatibilidad ni ninguna otra afirmación relacionada con productos que no son de IBM. Las consultas acerca de las prestaciones de los productos que no son de IBM deben dirigirse a los suministradores de tales productos.

Todas las declaraciones relativas a la dirección o intención futura de IBM están sujetas a cambios o anulación sin previo aviso y representan únicamente metas y objetivos.

Esta información se suministra sólo con fines de planificación. La presente información esta sujeta a cambios antes de que los productos que en ella se describen estén disponibles.

Esta información contiene ejemplos de datos e informes utilizados en las operaciones de negocios 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 los nombres y direcciones utilizados por una empresa real es totalmente casual.

LICENCIA DE COPYRIGHT:

Esta información contiene programas de aplicación de ejemplo en idioma de origen, que ilustra las técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo de cualquier forma, sin pagar a IBM, con la finalidad de desarrollar, utilizar, comercializar o distribuir programas de aplicación que se ajusten a la interfaz de programación de aplicaciones para la plataforma operativa para la que se han escrito los programas de ejemplo. Estos ejemplos no se han probado bajo todas las condiciones posibles. Por lo tanto, IBM no puede garantizar ni dar por sentada la fiabilidad, capacidad de servicio o funcionamiento de esos programas. Los programas de ejemplo se suministran "TAL CUAL", sin garantía de ninguna clase. IBM no se hará responsable de los daños que puedan derivarse del uso de los programas de ejemplo.

Cada copia, parcial o completa, de estos programas de ejemplo o cualquier trabajo derivado, debe incluir un aviso de copyright como el siguiente:

© (el nombre de su empresa) (año). Partes de este código provienen de programas de ejemplo de IBM Corp. © Copyright IBM Corp. _escriba el año o años_. Reservados todos los derechos.

Si está viendo esta información en copia software, es posible que las fotografías y las ilustraciones en color no aparezcan.

Consideraciones sobre la política de privacidad

Los productos de software de IBM, incluidas las soluciones de software como servicio, ("Ofertas de software"), pueden utilizar cookies u otras tecnologías para recopilar información sobre el uso de productos, para ayudar a mejorar la experiencia del usuario final, para personalizar las interacciones con el usuario final o para otros fines. En muchos casos, las Ofertas de software no recopilan información de identificación personal. Algunas de nuestras Ofertas de software pueden ayudarle a recopilar información de identificación personal. Si esta Oferta de software utiliza cookies para recopilar información de identificación personal, la información específica sobre el uso de cookies por parte de esta oferta se expone más abajo.

Dependiendo de las configuraciones desplegadas, esta Oferta de software puede utilizar cookies de sesión o persistentes. Si un producto o componente no está en la lista, ese producto o componente no utiliza cookies.

Tabla 15. Uso de cookies de los productos y componentes de InfoSphere Information Server

Módulo de producto	Componente o característica	Tipo de cookie que se utiliza	Recopilar estos datos	Finalidad de los datos	Inhabilitación de las cookies
Cualquiera (parte de la instalación de InfoSphere Information Server)	Consola web de InfoSphere Information Server	<ul style="list-style-type: none"> Sesión Persistente 	Nombre de usuario	<ul style="list-style-type: none"> Gestión de sesiones Autenticación 	No se pueden inhabilitar

Tabla 15. Uso de cookies de los productos y componentes de InfoSphere Information Server (continuación)

Módulo de producto	Componente o característica	Tipo de cookie que se utiliza	Recopilar estos datos	Finalidad de los datos	Inhabilitación de las cookies
Cualquiera (parte de la instalación de InfoSphere Information Server)	InfoSphere Metadata Asset Manager	<ul style="list-style-type: none"> • Sesión • Persistente 	Ninguna información de identificación personal	<ul style="list-style-type: none"> • Gestión de sesiones • Autenticación • Usabilidad de usuario mejorada • Configuración de inicio de sesión único 	No se pueden inhabilitar
InfoSphere DataStage	Etapas Big Data File	<ul style="list-style-type: none"> • Sesión • Persistente 	<ul style="list-style-type: none"> • Nombre de usuario • Firma digital • ID de sesión 	<ul style="list-style-type: none"> • Gestión de sesiones • Autenticación • Configuración de inicio de sesión único 	No se pueden inhabilitar
InfoSphere DataStage	Etapas XML	Sesión	Identificadores internos	<ul style="list-style-type: none"> • Gestión de sesiones • Autenticación 	No se pueden inhabilitar
InfoSphere DataStage	Consola de operaciones de IBM InfoSphere DataStage and QualityStage	Sesión	Ninguna información de identificación personal	<ul style="list-style-type: none"> • Gestión de sesiones • Autenticación 	No se pueden inhabilitar
InfoSphere Data Click	Consola web de InfoSphere Information Server	<ul style="list-style-type: none"> • Sesión • Persistente 	Nombre de usuario	<ul style="list-style-type: none"> • Gestión de sesiones • Autenticación 	No se pueden inhabilitar
InfoSphere Data Quality Console		Sesión	Ninguna información de identificación personal	<ul style="list-style-type: none"> • Gestión de sesiones • Autenticación • Configuración de inicio de sesión único 	No se pueden inhabilitar
InfoSphere QualityStage Standardization Rules Designer	Consola web de InfoSphere Information Server	<ul style="list-style-type: none"> • Sesión • Persistente 	Nombre de usuario	<ul style="list-style-type: none"> • Gestión de sesiones • Autenticación 	No se pueden inhabilitar
InfoSphere Information Governance Catalog		<ul style="list-style-type: none"> • Sesión • Persistente 	<ul style="list-style-type: none"> • Nombre de usuario • Identificadores internos • Estado del árbol 	<ul style="list-style-type: none"> • Gestión de sesiones • Autenticación • Configuración de inicio de sesión único 	No se pueden inhabilitar
InfoSphere Information Analyzer	Etapas Reglas de datos en el cliente del Diseñador de InfoSphere DataStage and QualityStage	Sesión	ID de sesión	Gestión de sesiones	No se pueden inhabilitar

Si las configuraciones desplegadas para esta Oferta de software le ofrecen como cliente la posibilidad de recopilar información de identificación personal de los usuarios finales mediante cookies y otras tecnologías, debe buscar asesoramiento jurídico sobre la legislación aplicable a dicha recopilación de datos, incluidos los requisitos de notificación y consentimiento.

Para obtener más información sobre el uso de diversas tecnologías, incluidas las cookies, para estos fines, consulte la Política de privacidad de IBM en <http://www.ibm.com/privacy>, la sección “Cookies, balizas web y otras tecnologías” de la Declaración de privacidad en línea de IBM en <http://www.ibm.com/privacy/details> y la “Declaración de privacidad de productos de software y software como servicio de IBM” (en inglés) en <http://www.ibm.com/software/info/product-privacy>.

Marcas registradas

IBM, el logotipo de IBM e [ibm.com](http://www.ibm.com) son marcas comerciales o marcas registradas de International Business Machines Corp., registradas en muchas jurisdicciones en todo el mundo. Otros nombres de productos y servicios pueden ser marcas registradas de IBM u otras empresas. Encontrará una lista actual de las marcas registradas de IBM en el sitio web www.ibm.com/legal/copytrade.shtml.

Los términos siguientes son marcas comerciales o marcas registradas de otras empresas:

Adobe es una marca registrada de Adobe Systems Incorporated en los Estados Unidos y/o en otros países.

Intel e Itanium son marcas comerciales o marcas registradas de Intel Corporation o sus filiales en los Estados Unidos y otros países.

Linux es una marca registrada de Linus Torvalds en los Estados Unidos y/ en otros países.

Microsoft, Windows y Windows NT son marcas registradas de Microsoft Corporation en los Estados Unidos y/o en otros países.

UNIX es una marca registrada de The Open Group en los Estados Unidos y en otros países.

Java y todas las marcas registradas y logotipos basados en Java son marcas comerciales o marcas registradas de Oracle y/o sus filiales.

El Servicio de correos de Estados Unidos (United States Postal Service) es propietario de las siguientes marcas registradas: CASS, CASS Certified, DPV, LACS^{Link}, ZIP, ZIP + 4, ZIP Code, Post Office, Postal Service, USPS y United States Postal Service. IBM Corporation es un licenciataria no exclusivo de DPV y LACS^{Link} del Servicio de correos de Estados Unidos.

Otros nombres de empresas, productos y servicios pueden ser marcas comerciales o marcas de servicio de terceros.

Cómo ponerse en contacto con IBM

Puede ponerse en contacto con IBM para obtener soporte al cliente, servicios de software, información sobre productos e información general. También puede facilitar comentarios a IBM sobre los productos y la documentación.

En la tabla siguiente se listan los recursos para soporte al cliente, servicios de software, formación e información sobre productos y soluciones.

Tabla 16. Recursos de IBM

Recurso	Descripción y ubicación
Portal de soporte de IBM	Puede personalizar la información de soporte eligiendo los productos y los temas que le interesen en www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server
Servicios de software	Puede encontrar información sobre servicios de software, de tecnologías de la información y de consultoría de negocio en el sitio de soluciones, en www.ibm.com/businesssolutions/
Mi IBM	Puede gestionar enlaces a sitios web de IBM y a información que satisfaga sus necesidades específicas de soporte técnico creando una cuenta en el sitio Mi IBM en www.ibm.com/account/
Formación y certificación	Puede obtener información sobre formación técnica y servicios de educación diseñados para personas, empresas y organizaciones públicas, a fin de adquirir, mantener y optimizar sus habilidades de TI en http://www.ibm.com/training
Representantes de IBM	Puede contactar con un representante de IBM para obtener información sobre soluciones en www.ibm.com/connect/ibm/us/en/

Índice

A

accesibilidad de los productos
 accesibilidad 51
avisos legales 59

C

conector
 definiciones de columna 6, 10
conector JDBC 1
 búsquedas
 configuración 6
 dispersas 14
 normales 13
 visión general 12
 columnas desaparejadas 21
 Conectarse a un origen de datos
 JDBC 5
 configuración
 configurar el conector JDBC como
 destino 9
 configurar el conector JDBC como
 origen 6
 definición de trabajo 4
 diseñar trabajos 2
 ejemplos
 buscar datos 12
 grabaciones
 acciones a realizar antes de
 grabar 19
 condiciones de rechazo 28
 configuración 9
 métodos soportados 32
 rechazar registros 11
 grabar datos 8
 habilitar identificadores con comillas
 derivaciones de columna 26
 lecturas
 configuración 6
 lecturas paralelas 7
 lecturas particionadas 7
 lecturas particionadas 29
 leer datos 5
 nivel de aislamiento 17
 propagación de columnas en tiempo
 de ejecución 25
 propiedades
 acción en tabla 19
 descartar campos que no
 coincidan 21
 Ejecutar sentencias SQL Before y
 After 22, 23
 habilitar identificadores con
 comillas 16
 nivel de aislamiento 17
 ordenación de registros 18
 propagación de columnas en
 tiempo de ejecución 25
 Propiedades para el conector
 JDBC 16

conector JDBC (*continuación*)
 rechazar registros
 condiciones de rechazo 28
 configuración 11
 Referencias 16
 sensibilidad a las mayúsculas y
 minúsculas 16
controlador JDBC
 archivo de configuración de
 controlador 1
correlacionar
 tipos de datos 34, 36, 37

D

Discrepancia de tipo de datos 21
documentación del producto
 acceder 53

E

etapa JDBC Connector
 Optimización del rendimiento 39

I

Importar metadatos
 InfoSphere Metadata Asset
 Manager 3
Informar de discrepancia de esquema 21

J

Juego de caracteres
 Juego de caracteres para columnas no
 Unicode
 Nombre de juego de caracteres 22

M

marcas registradas
 lista de 59

O

Optimización del rendimiento
 atributo Longitud 40
 columnas 40
 columnas sin vincular 44
 Configurar valores de controlador 39
 confirmaciones de transacción 43
 conversiones de juego de
 caracteres 41
 enlaces de rechazo 43
 esperas de bloqueo 42
 importar metadatos 40
 particionamiento de registros 42
 proceso por lotes 43

Optimización del rendimiento
(*continuación*)

 tamaño de captación diferente 41
 varios enlaces 42
orígenes JDBC
 conector JDBC 1

R

resolución de problemas
 conector JDBC 44

S

servicios de software
 contactar 65
sitios web
 no IBM 57
sitios web no de IBM
 enlaces a 57
soporte
 cliente 65
soporte al cliente
 contactar 65

T

tipos de datos 34, 36
 cargar datos 34
 DataStage 34, 36, 37
 grabar datos 34
 importar datos 37
 Java 34
 JDBC 37
 leer datos 37
 nativos 36



Impreso en España

SC43-1753-00

