

IBM InfoSphere DataStage y QualityStage
Versión 8 Release 7

*Guía de conectividad para
procedimientos almacenados*



IBM InfoSphere DataStage y QualityStage
Versión 8 Release 7

*Guía de conectividad para
procedimientos almacenados*



Nota

Antes de utilizar esta información y el producto al que da soporte, lea la información de la sección "Avisos y marcas registradas" en la página 47.

Contenido

Capítulo 1. Introducción 1

Introducción	1
Requisitos de configuración	1
Requisitos para las variables de entorno	2
La etapa Stored Procedure y el lienzo Paralelo	2
Correlación de datos de serie	2

Capítulo 2. Etapas Stored Procedure . . . 5

Funcionalidad	5
Acerca de la etapa Stored Procedure	5
Tipos de procedimientos	6
Parámetros	7
Utilización de la etapa Stored Procedure	8
Acerca de la página Etapa	9
Definición de la correlación de juego de caracteres	16
Definición de datos de entrada	16
Acerca de la página Entrada	16
Definición de datos de salida	19
Acerca de la página Salida	19
Enlaces	22
Uso de media secuencia	22
Emulación de enlaces de referencia	22

Capítulo 3. Información específica sobre bases de datos adicionales . . . 27

Oracle	27
Paquetes	27
Manejo de errores de PL/SQL	27
Gestión de cursores.	28
SELECT INTO	29
Sentencia de llamada definida por el usuario	29
DB2	29
Manejo de errores de SQL	29
Gestión de cursores.	30

SELECT INTO	31
Sentencia de llamada definida por el usuario	31
Sybase	31
Tamaño de transacción	31
Teradata	32
Importación de procedimientos almacenados	33
Importación de macros	33
Opciones para el juego de caracteres del cliente	34
Códigos de error	35
Recuento de actividades para enlace	35
Devolución de varias filas	36
Gestión de cursores.	36
SQL Server	36
Funciones escalares definidas por el usuario	36
Devolución de conjuntos de resultados y parámetros de salida	37
Valor de retorno	37
Parámetros de salida y códigos de retorno	37
Sintaxis de las llamadas a procedimientos y códigos de retorno	37

Cómo ponerse en contacto con IBM . . . 39

Acceso a la documentación de productos 41

Cómo leer los diagramas de sintaxis . . . 43

Accesibilidad de los productos 45

Avisos y marcas registradas 47

Índice 51

Capítulo 1. Introducción

Los procedimientos almacenados amplían la diversidad de InfoSphere DataStage. La siguiente sección proporciona una introducción a la etapa Stored Procedure (STP). La segunda mitad del capítulo trata de las instrucciones de instalación y ofrece información de configuración.

Introducción

Utilice la etapa Stored Procedure como:

- Un origen, devolviendo un conjunto de filas
- Un destino, pasando una fila a un procedimiento almacenado para grabación
- Una transformación, invocando a la lógica de dentro de la base de datos

La etapa Stored Procedure soporta parámetros de entrada y salida o argumentos para que pueda proporcionar un valor y reciba un valor a cambio. Puede procesar el valor devuelto después de que se haya ejecutado el procedimiento almacenado. La etapa Stored Procedure proporciona códigos de estado que indican si el procedimiento almacenado se ha completado satisfactoriamente o no. En caso negativo, permite manejar errores.

Actualmente la etapa Stored Procedure soporta Oracle, DB2, Sybase, Teradata y SQL Server. El diseño de la etapa Stored Procedure permite configurarla en tiempo de ejecución para ejecutar procedimientos almacenados en diferentes sistemas de gestión de bases de datos relacionales utilizando interfaces nativas.

Nota: Debe utilizar la interfaz gráfica de usuario personalizada para entrar datos en la etapa Stored Procedure. IBM® no da soporte al uso del editor de estilos de cuadrícula. No debería utilizarlo porque puede corromper los datos de trabajo de parámetros de la etapa Stored Procedure. El editor de la cuadrícula se inhabilitará en un release futuro del Diseñador de DataStage.

Requisitos de configuración

Oracle

Oracle requiere la siguiente configuración:

- Versión 10g del software de cliente Oracle en la máquina del servidor de InfoSphere DataStage, que necesita lo siguiente:
 - **Versión 10g:** Cliente de Oracle 10g (tiempo de ejecución)
- Configuración de SQL*Net utilizando un programa de configuración, como por ejemplo SQL*Net Easy Configuration, para configurar y añadir alias de base de datos.

DB2

DB2 utiliza el protocolo DRDA (TCP/IP) si los datos de DB2 residen en un sistema AS/400.

Requisitos para las variables de entorno

La etapa Stored Procedure requiere el establecimiento de variables de entorno para funcionar correctamente en un entorno UNIX. La Tabla 1 identifica las variables de entorno específicas que se necesitan.

Tabla 1. Variables de entorno necesarias para UNIX

Para...	Establezca la siguiente variable de entorno en la máquina del servidor...
Oracle	ORACLE_HOME TWO_TASK ORACLE_SID LD_LIBRARY_PATH ¹
DB2	DB2INSTANCE INSTHOME LD_LIBRARY_PATH
Sybase	SYBASE SYBASE_OCS LD_LIBRARY_PATH ^{2, 3}
Teradata	LD_LIBRARY_PATH PATH

¹El nombre de una variable de entorno determinada, denominada LD_LIBRARY_PATH más arriba, difiere dependiendo de la plataforma. Para determinar el nombre correcto que debe utilizar en su entorno:

- ¹Si la plataforma es AIX, utilice LIBPATH.
- ¹Si la plataforma es HP_UX, utilice SHLIB_PATH.
- ¹Si la plataforma es LINUX o Solaris, utilice LD_LIBRARY_PATH.

²Para funcionar correctamente en Linux, Sybase OC necesita la variable de entorno de idioma LANG = en.

³En las aplicaciones Sybase, para la variable de entorno LD_LIBRARY_PATH, las entradas de biblioteca de InfoSphere DataStage deben referenciarse antes que las entradas de la biblioteca de Sybase Open Client durante el tiempo de ejecución.

La etapa Stored Procedure y el lienzo Paralelo

Algunas etapas de conectividad pueden ejecutarse en el lienzo Paralelo. El valor predeterminado para *todas* las etapas es Secuencia. "En paralelo" significa que se puede establecer para que se ejecute en paralelo pero *no* es el valor predeterminado. Las que están disponibles tienen instrucciones adicionales para correlacionar datos de serie cuando se ha activado el soporte multilingüístico (consulte "**Correlación de datos de serie**"). DRS está disponible en el lienzo paralelo en UNIX. Puede utilizarlo como origen o destino o para procesar. Como origen, se ejecuta secuencialmente; como destino, se ejecuta en paralelo.

Correlación de datos de serie

El propósito de la correlación NONE en el lienzo del servidor consiste en desactivar la correlación de datos de serie en cualquier etapa en que se establezca la correlación, es decir, pasar los datos literalmente. Esta característica se maneja de modo distinto en el lienzo Paralelo. Cuando se definen datos de serie (char, varchar, etc.), hay un campo adicional en la cuadrícula Columnas de la etapa denominado Ampliado. Puede establecerse en blanco o en Unicode. Si se deja en blanco esta opción, no se producirá ninguna correlación (por ejemplo "NONE"); se hace caso omiso de la correlación especificada en el soporte multilingüístico. Esta

opción se establece en Unicode, se aplica la correlación de soporte multilingüístico. Para leer o grabar datos en japonés, por ejemplo, establezca Ampliado en Unicode. Cuando el compilador de trabajos detecta esta combinación (char, varchar, etc. y Unicode), genera el código de tiempo de ejecución apropiado.

Capítulo 2. Etapas Stored Procedure

Las etapas Stored Procedure amplían las posibilidades de InfoSphere DataStage proporcionando un recurso para ejecutar procedimientos almacenados. Un procedimiento almacenado es un bloque de construcciones de procedimiento y sentencias SQL incluidas que se almacena en una base de datos y que puede ser llamado por su nombre. Este capítulo describe los temas siguientes:

- "Funcionalidad"
- "Acerca de la etapa STP"
- "Utilización de la etapa STP"
- "Definición de datos de entrada"
- "Definición de datos de salida"
- "Enlaces"

Funcionalidad

La etapa Stored Procedure presenta la funcionalidad y las ventajas siguientes:

- Ejecuta procedimientos almacenados y funciones.
- Soporta bases de datos relacionales a través de interfaces nativas.
- Utiliza terminología de enlace estándar para una etapa activa.
- Soporta la correlación de columnas de entrada y salida con parámetros de entrada y salida en un procedimiento.
- Soporta la identificación de columnas adicionales para contener valores de retorno e información de estado.
- Soporta el paso a través de información de columnas si la etapa se utiliza a media secuencia en un trabajo.
- Acepta parámetros de trabajo para identificar determinadas propiedades en tiempo de ejecución.
- Soporta la grabación de información de ubicador.
- Soporte para soporte multilingüístico.

La siguiente funcionalidad no está soportada:

- Varios procedimientos en una sola etapa.
- Ejecución de procedimientos almacenados a nivel de columna. Si se necesita esta función, utilice una etapa de transformador para dirigir cada columna hacia una etapa Stored Procedure independiente.
- Ejecución de sentencias SQL.
- Enlaces de referencia.

Acerca de la etapa Stored Procedure

Las secciones siguientes describen los tipos de procedimientos que soporta la etapa Stored Procedure y el rol de la cuadrícula de parámetros.

Tipos de procedimientos

La etapa Stored Procedure soporta tres tipos de procedimientos:

- Procedimientos de transformación
- Procedimientos de origen
- Procedimientos de destino

Los procedimientos de origen y destino utilizan los metadatos de la columna como parámetros para pasarlos al procedimiento. No tiene que entrar la información de parámetros; si la entra, la etapa la ignorará. Los procedimientos de origen utilizan los metadatos de la columna como parámetros de salida (para leerlos desde la base de datos). Los procedimientos de destino utilizan los metadatos de la columna como parámetros de entrada (para grabarlos en la base de datos).

Nota: Los procedimientos de origen y destino no pueden tener enlaces de entrada y salida a la vez. Debe utilizar un procedimiento de transformación.

Un procedimiento de origen puede ejecutar un procedimiento almacenado que contenga una sentencia SQL SELECT. Puede:

- Llamarse una vez para el trabajo
- Utilizarse con un enlace de salida
- Devolver parámetros de salida
- Devolver más de una fila

Un procedimiento de destino puede ejecutar un procedimiento almacenado que contenga una sentencia SQL INSERT. Puede:

- Llamarse una vez para el trabajo
- Llamarse una vez para cada fila (predeterminado)
- Utilizarse con un enlace de entrada
- Pasar parámetros de entrada

Si desea mezclar y hacer coincidir parámetros con metadatos de columna, utilice un procedimiento de transformación y especifique los Parámetros de Stored Procedure en la cuadrícula de parámetros.

El procedimiento de transformación ejecuta varias decisiones lógicas en la base de datos. Puede:

- Llamarse una vez para el trabajo
- Llamarse una vez para cada fila (predeterminado)
- Pasar parámetros de entrada utilizando la cuadrícula de parámetros
- Devolver parámetros de salida utilizando la cuadrícula de parámetros
- Pasar a través de columnas si se utiliza media secuencia (enlaces de entrada y salida)
- Utilizarse sólo con un enlace de entrada
- Utilizarse sólo con un enlace de salida
- Utilizarse con un enlace de entrada y un enlace de salida
- Devolver más de una fila

Todos los procedimientos pueden:

- Devolver un código de error (definido por el usuario o desde la base de datos) que pueda correlacionarse con el flujo de datos
- Devolver un mensaje (definido por el usuario o desde la base de datos) que pueda correlacionarse con el flujo de datos
- Permitir decisiones de recuperación de código de aviso y muy graves definidas por el usuario
- Definir niveles de aislamiento de transacción
- Permitir la especificación de un tamaño de confirmación al ejecutar un procedimiento que inserta filas
- Permitir una sentencia de ejecución de procedimiento definida por el usuario (no se utiliza normalmente)
- Permitir la devolución de más de una fila al leer

Nota: La etapa Stored Procedure no ejecuta ninguna sentencia SQL. Utilícela exclusivamente para la ejecución de procedimientos almacenados. Utilice las etapas nativas para la ejecución de sentencias SQL.

Teradata

Utilizando la etapa Stored Procedure, puede invocar procedimientos almacenados, macros y funciones de Teradata desde trabajos de InfoSphere DataStage. No puede crear procedimientos almacenados de Teradata, macros o funciones desde InfoSphere DataStage. Dichos objetos deberían crearse con las propias interfaces de usuario de Teradata, como BTEQ o Teradata SQL Assistant (anteriormente conocida como Queryman antes de Teradata Tools and Utilities 7.0).

SQL Server

Un procedimiento almacenado es un grupo de sentencias Transact-SQL compiladas en un único plan de ejecución, almacenadas bajo un nombre y procesadas como una unidad.

La etapa Stored Procedure soporta las llamadas a procedimientos almacenados de Microsoft SQL Server (internos y externos) y funciones definidas por el usuario que devuelven un solo valor.

Nota: Es preferible invocar a las funciones definidas por el usuario que devuelven una tabla desde la etapa ODBC, porque la sentencia SELECT generada debe consultar una tabla.

Se soportan funciones escalares definidas por el usuario. Un parámetro de salida y cursor global que utilizan variedad de cursos no se soportan como métodos de devolver datos.

Los procedimientos almacenados de Microsoft SQL Server soportan parámetros de entrada y salida.

Parámetros

STP utiliza una cuadrícula de parámetros para correlacionar las columnas de entrada y salida con los parámetros de entrada y salida del procedimiento almacenado. La cuadrícula de parámetros está ubicada en el separador **Parámetros** de la página Etapa. Cuando se proporciona la información necesaria, la etapa analiza estos valores y utiliza la información para crear la llamada a procedimiento almacenado y enlazar las variables internas. Para los procedimientos de origen y

destino no se necesitan especificaciones de parámetro. Estas se ignoran porque los metadatos de columna se utilizan para correlacionar la información del parámetro.

En el caso del procedimiento de transformación, se utilizan las especificaciones de parámetros para correlacionar los parámetros con los metadatos de columna. La cuadrícula de parámetros debe rellenarse con la información adecuada para llamar al procedimiento almacenado.

Reglas para especificar valores de parámetros

- Los valores literales y los parámetros de trabajo de InfoSphere DataStage sólo se soportan para los procedimientos de transformación.
- El trabajo debe contener un enlace de entrada para que el **Tipo de parámetro** se defina como **Entrada/Salida** o **Entrada** si el valor del parámetro es ?. InfoSphere DataStage necesita una fila activa a partir de la cual se pueden leer los valores.
- El trabajo debe contener un enlace de salida para **Tipo de parámetro** para que se defina como **Entrada/Salida**, **Salida** o **Función**. (**CursorOutput** es una opción adicional en DB2.) InfoSphere DataStage necesita crear una fila activa para producir los valores en forma de salida.

Ejemplos de valores de parámetros

El ejemplo 1 muestra una especificación de cuadrícula de parámetros que envía datos desde un enlace de entrada hacia el procedimiento almacenado:

Tabla 2. Ejemplo 1

Nombre del parámetro	Correlaciones para columna	Marcador de parámetros/Literal	Tipo de parámetro
EMPNOP	EMPNO	?	Entrada
JOBP	JOB	?	Entrada/Salida
DEPTNOP	DEPTNO	99	Entrada
ENAMEP	ENAME	#ename#	Entrada

En el ejemplo 1, 99 es un literal, y #ename# es un parámetro de trabajo.

El ejemplo 2 muestra una especificación de cuadrícula de parámetros que envía y devuelve datos desde un procedimiento almacenado e inicia la ejecución de una función. Funciona de este modo porque sólo un **Tipo de parámetro** es Función:

Tabla 3. Ejemplo 2

Nombre del parámetro	Correlaciones para columna	Marcador de parámetros/Literal	Tipo de parámetro
FUNCSTATP	FUNCSTAT	?	Función
DNAMEP	DNAME	?	Salida
LOCP	LOC	#city#	Entrada

Utilización de la etapa Stored Procedure

Cuando utilice la interfaz gráfica de usuario para editar una etapa Stored Procedure, se abrirá el recuadro de diálogo Etapa STP.

Este recuadro de diálogo puede tener hasta tres páginas (en función de si existen entradas o salidas de la etapa):

- **Etapa.** Esta página muestra el nombre de la etapa que se está editando. El separador **General** define el proveedor de la base de datos, el origen de la base de datos e información de inicio de sesión para conectarse a la base de datos. También establece el nivel ISO de la transacción. Para obtener detalles, consulte "**Acerca de la página Etapa**".

El separador **Soporte multilingüístico** define una correlación de juego de caracteres que se debe utilizar con la etapa. (El separador **Soporte multilingüístico** sólo aparece si se ha instalado el soporte multilingüístico). Para obtener detalles, consulte "**Definición de la correlación de juego de caracteres**".

- **Entrada.** Esta página sólo se muestra si dispone de un enlace de entrada de esta página. Especifica las definiciones de columna asociadas para cada enlace de entrada de datos. Esta página también especifica instrucciones para el procedimiento almacenado.
- **Salida.** Esta página sólo se muestra si dispone de un enlace de salida de esta etapa. Especifica las definiciones de columna asociadas para cada enlace de salida de datos. Esta página también especifica información de códigos de retorno de error e instrucciones para el procedimiento almacenado.

Acerca de la página Etapa

Efectúe una doble pulsación del icono **STP**, pulse con el botón derecho del ratón el icono **STP** y seleccione **Propiedades** o seleccione **Propiedades** desde el menú **Editar**. Se abrirá el recuadro de diálogo Etapa STP.

La página Etapa consta de los separadores **General**, **Sintaxis**, **Parámetros**, **Códigos de error**, **Soporte multilingüístico** (opcional) y **Avanzado** (sólo para trabajos paralelos). El separador **General** de la página Etapa aparece de forma predeterminada.

El separador General

Utilice el separador **General** para establecer parámetros de conexión.

El separador contiene los campos siguientes:

- **Proveedor de la base de datos.** Tipo de base de datos relacional. En este release, las opciones son **Oracle**, **DB2**, **Sybase**, **Teradata** y **MSSQL Server**. El **Proveedor de base de datos** es necesario.
- **Origen de base de datos.** Nombre del alias de la base de datos.

Oracle

- Para Oracle, entre el nombre que ha creado mediante el asistente de configuración de Oracle.

DB2

- Para DB2, entre el nombre de la conexión, el nombre de la fuente de datos o el nombre de la base de datos.

SQL Server

- Para SQL Server, entre un nombre de ODBC DSN que describa la información sobre el controlador y el servidor.
 - **Nombre de usuario.** Nombre de usuario que debe utilizarse para conectar con la base de datos. Este usuario debe disponer de privilegios suficientes

para acceder a la base de datos y tablas de origen y de destino especificadas. Este campo es necesario. No existe ningún valor predeterminado.

- **Contraseña.** Contraseña asociada con el nombre de usuario especificado. Este campo es necesario. No existe ningún valor predeterminado.
- **Nombre de la base de datos.** Base de datos del servidor especificado.

DB2

- Este campo no está disponible para DB2.

Nota: Si selecciona **DB2** como el **Proveedor de la base de datos** y pulsa **Aceptar** al volver a la página Etapa, el **Nombre de la base de datos** no estará activo.

SQL Server

- Este campo no está disponible para SQL Server.
- **ISO de transacción.** Nivel de aislamiento de la transacción (nivel ISO). El nivel ISO proporciona la coherencia y el control de concurrencia necesarios entre las transacciones del trabajo y otras transacciones para un rendimiento óptimo. Utilice **ISO de transacción** para establecer el nivel de aislamiento de la transacción para los enlaces de entrada y de salida, seleccionando un valor en la lista proporcionada. El nivel de ISO es específico de la base de datos y podría ser que cada base de datos no pudiera dar soporte a todas las opciones genéricas presentadas. Para obtener más información sobre la utilización de estos niveles, consulte la documentación de la base de datos. Utilice uno de los siguientes niveles de aislamiento de transacción:
 - **Ninguna. El valor predeterminado.**
 - **Sólo lectura.** No fija bloqueos de grabación. Este es el nivel de ISO permitido más eficaz para un rendimiento óptimo.
 - **Lectura no confirmada.** Fija bloqueos exclusivos para los datos modificados. Estos bloqueos se mantienen hasta que se ejecuta una confirmación o una retroacción. No obstante, otras transacciones pueden leer pero no modificar los cambios no confirmados. No se fijan otros bloqueos.
 - **Lectura confirmada.** Fija bloqueos exclusivos para los datos modificados y bloqueos que se pueden compartir para todos los demás datos. Este es el valor predeterminado. Los bloqueos exclusivos se mantienen hasta que se ejecuta una confirmación o una retroacción. Los cambios no confirmados no pueden ser leídos por otras transacciones. Los bloqueos compartidos se liberan inmediatamente después de procesar los datos, permitiendo que otras transacciones puedan modificarlos.
 - **Lectura repetible.** Idéntico a Serializable con la excepción de que podrían visualizarse las filas fantasma.
 - **Serializable.** Fija bloqueos exclusivos para los datos modificados y bloqueos que se pueden compartir para todos los datos. Todos los bloqueos se mantienen hasta que se ejecuta una confirmación o una retroacción, evitando que otras transacciones modifiquen datos a los que se haya hecho referencia durante la transacción.

Oracle

Elija entre **Ninguno**, **Sólo lectura**, **Lectura confirmada** y **Serializable** para Oracle.

DB2

Elija entre **Ninguno**, **Lectura no confirmada**, **Lectura confirmada**, **Lectura repetible** y **Serializable** para DB2.

Sybase

Elija entre **Ninguno**, **Lectura no confirmada**, **Lectura confirmada** y **Serializable** para Sybase.

Teradata

ISO de transacción está inactivo si el **Proveedor de la base de datos** es **Teradata**. El RDBMS de Teradata da soporte a un modificador LOCKING en algunas sentencias SQL para permitir el control de los niveles de aislamiento de la transacción, pero las sentencias CALL y EXECUTE no dan soporte al modificador LOCKING. Para utilizar un método de bloqueo determinado, especifique el modificador LOCKING de las sentencias SQL en el cuerpo de la macro o procedimiento.

SQL Server

Elija entre **Ninguno**, **Lectura no confirmada**, **Lectura confirmada**, **Lectura repetible** y **Serializable** para SQL Server.

- **Modalidad de transacción.** Modalidad de conexión con el servidor de Teradata. Este campo sólo está activo si el **Proveedor de la base de datos** seleccionado es **Teradata** o un parámetro de trabajo.
 - Si el **Proveedor de la base de datos** es **Teradata**, las opciones para la **Modalidad de transacción** son **ANSI**, el valor predeterminado y **Teradata**. Este campo controla si la etapa Stored Procedure conecta con el servidor de Teradata en la modalidad de transacción de ANSI o Teradata. Una conexión en la modalidad de transacción de ANSI no puede llamar procedimientos almacenados que se han compilado en la modalidad de transacción de Teradata y una conexión en la modalidad de transacción de Teradata no puede llamar procedimientos almacenados que se han compilado en la modalidad de transacción de ANSI.
 - Si el **Proveedor de la base de datos** es un parámetro de trabajo, las opciones para la **Modalidad de transacción** serán **ANSI** y **Nativa**. La modalidad de transacción Nativa es equivalente a la modalidad de transacción de Teradata si el proveedor de la base de datos durante el tiempo de ejecución es Teradata. El campo de la modalidad de transacción no afecta a los demás proveedores.
- **Nota:** Si se produce un error en la modalidad de transacción de Teradata, Teradata retrotraerá automáticamente la transacción actual. En la modalidad de transacción de ANSI, un error no afecta a la transacción actual.
- **Juego de caracteres de cliente.** Juego de caracteres del cliente de Teradata que debe utilizarse al conectar con el servidor de Teradata. Este campo sólo está activo si el **Proveedor de la base de datos** es **Teradata** o un parámetro de trabajo. El valor predeterminado es **Predeterminado**.
- **Descripción.** Descripción opcional de la etapa Stored Procedure.

El separador Sintaxis

Utilice el separador **Sintaxis** para identificar el procedimiento almacenado que debe ejecutarse.

El separador **Sintaxis** contiene las siguientes propiedades:

- **Nombre del procedimiento.** Nombre del procedimiento almacenado que debe ejecutarse en el servidor. Debe proporcionar el nombre específico del procedimiento que desea ejecutar. También puede pulsar el botón ... situado a la derecha del **Nombre del procedimiento** para examinar el repositorio para seleccionar el procedimiento almacenado. Para ello, la definición del procedimiento almacenado debe importarse en el repositorio mediante el cliente de IBM InfoSphere DataStage and QualityStage Designer utilizando las funciones **Importar Definiciones de tablas Definiciones de procedimientos almacenados**.

Teradata

El recurso de importación Definiciones de procedimientos almacenados utiliza ODBC, por lo que debe configurar primero un nombre de fuente de datos ODBC (DSN). Recomendamos utilizar el último controlador ODBC de Teradata, versión 3.04, suministrado con el cliente Teradata Tools and Utilities 8.0.

- **Tipo de procedimiento.** Tipo de procedimiento que debe ejecutarse. Las opciones son:
 - **Origen.** Los procedimientos de origen emulan una sentencia SQL SELECT y utilizan los metadatos de la columna como parámetros de salida para que se lean desde la base de datos. Sólo disponen de un enlace de salida.
 - **Destino.** Los procedimientos de destino emulan una sentencia SQL INSERT y utilizan los metadatos de la columna como parámetros de entrada para que se graben en la base de datos. Sólo disponen de un enlace de entrada.
 - **Transformación.** Los procedimientos de transformación utilizan propiedades de procedimiento almacenado para ejecutar varias decisiones lógicas en la base de datos. Puede utilizarse sólo con un enlace de entrada, sólo con un enlace de salida o con un enlace de entrada y de salida. Especifique los parámetros de entrada y de salida en la cuadrícula de parámetros.
El valor predeterminado es **Transformación**.
- **Tipo de procedimiento de base de datos.** Detalles adicionales sobre el tipo de procedimiento que debe ejecutarse. **Tipo de procedimiento de base de datos** afecta a la sintaxis generada de llamada de procedimiento. Está activo sólo si el **Proveedor de la base de datos** es un **MSSQL Server**, **Teradata** o un parámetro de trabajo.

Teradata

Las opciones de Teradata son:

- **Procedimiento almacenado**
- **Macro**
- **Función escalar definida por el usuario**
- **Función de tabla definida por el usuario**

SQL Server

Las opciones de SQL Server son:

- **Procedimiento almacenado**
- **Función escalar definida por el usuario**

El valor predeterminado es **Procedimiento almacenado**.

- **Sintaxis de llamada de procedimiento.** Sintaxis utilizada para ejecutar la sentencia de llamada del procedimiento. La sintaxis se puede modificar desmarcando **Generar llamada de procedimiento**.
- **Generar llamada de procedimiento.** Opción para que la etapa genere la sentencia de llamada del procedimiento mediante los metadatos del trabajo. El nombre del procedimiento, la información sobre el parámetro o la información sobre la columna se utilizan para generar la sintaxis de llamada. Si se selecciona esta opción, la etapa generará la sentencia. Si se desmarca, la etapa no generará la sentencia. Para editar la sintaxis del procedimiento, desmarque **Generar llamada de procedimiento**. El valor predeterminado seleccionado es **Generar llamada de procedimiento**.

Nota: Al entrar sentencias de llamada definidas por el usuario (**Generar llamada de procedimiento** está desmarcada) para ejecutar procedimientos, debe respetar la sintaxis específica de la base de datos para la etapa a fin de ejecutar el procedimiento satisfactoriamente.

El separador Parámetros

El separador **Parámetros** identifica y describe los parámetros utilizados si el **Tipo de procedimiento** es **Transformar**.

La etapa Stored Procedure permite correlacionar las columnas de entrada y salida con los parámetros de entrada y salida del procedimiento mediante las especificaciones de los parámetros. La etapa utiliza la información suministrada en el separador **Parámetros** para compilar la llamada del procedimiento almacenado y enlazar las variables internas. El separador **Parámetros** se ignorará si el tipo de procedimiento se establece en **Origen** o **Destino**, ya que los metadatos de la columna se utilizan para correlacionar la información sobre los parámetros.

Las siguientes propiedades se incluyen en el separador **Parámetros**:

- **Nombre de parámetro.** Nombre del parámetro.
- **Correlaciones para columna.** Nombre de la columna en la que el parámetro realiza las correlaciones.
- **Marcador de parámetros/literal.** Valor literal, el parámetro de trabajo de IBM InfoSphere DataStage o un marcador de parámetros (?). El valor predeterminado es ?.
- **Tipo de parámetro.** Tipo del parámetro. Las opciones son:
 - **Entrada.** La etapa envía datos del enlace o del valor literal al procedimiento almacenado. Este es el valor predeterminado. La etapa debe contener un enlace de entrada. Seleccione **Entrada** para cumplir las cláusulas WHILE o WHERE para la operación seleccionada y para insertar valores en la tabla.
 - **Salida.** La etapa devuelve datos o un conjunto de resultados del procedimiento almacenado. La etapa debe contener un enlace de salida. Seleccione **Salida** para cumplir SELECT @param=10. El procedimiento almacenado devuelve filas individuales o parámetros de salida específicos.
 - **CursorOutput.** La etapa devuelve datos o un conjunto de resultados del procedimiento almacenado. La etapa debe contener un enlace de salida. **CursorOutput** sólo está disponible para DB2 y SQL Server. Seleccione **CursorOutput** para cumplir SELECT one from TABLE_NAME. El procedimiento almacenado devuelve un registro o un conjunto de resultados.
 - **Entrada/Salida.** La etapa envía datos del enlace y devuelve datos o un conjunto de resultados. La base de datos debe dar soporte al uso de parámetros que pueden enviar y devolver información. La etapa debe contener un enlace de entrada y de salida.

- **Función.** Algunas bases de datos dan soporte a funciones almacenadas. **Función** desencadena la sintaxis generada para ejecutar una función en lugar de un procedimiento almacenado. La correlación del conjunto de parámetros con F contiene el valor de retorno de la función almacenada. Sólo un parámetro se puede definir como un tipo de función. La etapa debe contener un enlace de salida.

Oracle

Los valores válidos para el **Tipo de parámetro** de Oracle son **Entrada, Salida, Entrada/Salida y Función.**

DB2

Los valores válidos para el **Tipo de parámetro** de DB2 son **Entrada, Salida, Salida de cursor y Entrada/Salida.**

Nota: A pesar de que la etapa Stored Procedure no da soporte a las funciones almacenadas con DB2, puede ejecutar las funciones almacenadas de una de estas dos formas:

- Llamar un procedimiento almacenado que ejecute una función definida por el usuario
- Codificar una llamada a una función almacenada dentro de una sentencia SQL

Sybase

Los valores válidos para el **Tipo de parámetro** de Sybase son **Entrada y Salida.** Sybase no da soporte a las funciones almacenadas.

Teradata

Los valores válidos para el **Tipo de parámetro** de Teradata son **Entrada, Salida, Entrada/Salida y Función.**

SQL Server

Los valores válidos para el **Tipo de parámetro** de SQL Server son **Entrada, Salida, Salida de cursor, Entrada/Salida y Función.**

- **Cargar...** Pulse el botón **Cargar...** ubicado bajo la cuadrícula para examinar el repositorio a fin de seleccionar los parámetros de procedimiento. Para ello, los parámetros de procedimiento deben importarse en el repositorio.

Teradata

Generalmente, el botón **Cargar** sólo carga desde el separador **Parámetros** del registro **Definiciones de tabla.** Pero si el **Proveedor de la base de datos** es **Teradata**, el botón **Cargar** también cargará parámetros desde el separador **Columnas** del registro **Definiciones de tabla.** Estos parámetros tienen el **Tipo de parámetro** establecido como

- **Entrada** si se selecciona **Clave** en el separador **Columnas** de la **Definición de tabla**
- **Salida** si no se selecciona **Clave** en el separador **Columnas** de la **Definición de tabla**

Esto es necesario, ya que el recurso de importación de Definiciones de procedimientos almacenados importa las columnas de salida de una macro en el separador **Columnas** y el conector de API de Teradata importa todos los parámetros en el separador **Columnas**.

El separador Códigos de error

El separador **Códigos de error** identifica errores definidos por el usuario.

El separador **Códigos de error** contiene las siguientes propiedades:

- **Errores definidos por el usuario.** Códigos que identifican errores. Las listas deben contener errores específicos de base datos expresados como la parte entera y errores definidos por el usuario que el procedimiento puede devolver. Existen dos categorías de error:

- **Errores muy graves.** Se especifican los errores que deben tratarse como muy graves. La etapa comprueba primero la lista de errores muy graves. Si encuentra uno de los códigos identificados como muy graves, el trabajo terminará anormalmente.

- **Avisos.** Se especifican errores que deben tratarse como avisos. La etapa comprueba la lista de avisos sólo después de comprobar la lista de errores muy graves. Si la etapa encuentra alguno de los códigos identificados como un aviso, la etapa grabará una entrada de registro y el proceso continuará.

El manejo de errores de la etapa Stored Procedure es similar al manejo de errores de la etapa ODBC. La etapa Stored Procedure proporciona las propiedades **Errores muy graves** y **Avisos** en las cuales puede entrar una lista de valores, separados por un espacio, para manejar los valores de retorno dentro del procedimiento almacenado.

La etapa comprueba estas listas de códigos y procesa los códigos de error tal y como se ha solicitado.

- Si el código se encuentra en la lista de errores muy graves, el trabajo terminará anormalmente.

- Si el código se encuentra en la lista de avisos, se grabará una entrada de registro y el proceso continuará.

La etapa comprueba primero la lista de errores muy graves. Si especifica el mismo código en ambas listas, el trabajo terminará anormalmente. Las listas deben contener errores específicos de base de datos que se han reducido para ajustarse a la parte del entero del error. También puede proporcionar códigos de error definidos por el usuario que el procedimiento podría devolver. Si deja las listas vacías (valor predeterminado), la etapa utilizará el estándar de la base de datos y procederá de dicha forma.

Al grabar datos en la base de datos, si los datos se truncan durante la transferencia de IBM InfoSphere DataStage, dispone de la opción de continuar el proceso. Entre el código de aviso especial -99 en la lista de avisos para desencadenar que la etapa transfiera los datos truncados a la base de datos. La etapa graba una entrada de aviso en el registro y el trabajo no termina anormalmente.

Nota: Tenga precaución al tratar los errores muy graves como avisos. Esta acción puede provocar que la etapa siga vías de acceso de código desconocidas y que el proceso falle. La "Violación de clave exclusiva" es un ejemplo de error que podría tratar como aviso, saltando la fila y continuando con el proceso. Pero si trata "No es posible conectar con la base de datos" como un aviso, la etapa no podrá recuperarlo y el proceso fallará.

Teradata

- **Cargar...** Pulse el botón **Cargar...** para examinar el repositorio en busca de determinados códigos de error muy grave o de aviso que deben importarse. Para ello, los códigos de procedimiento almacenado deben importarse en el repositorio.

El separador Avanzado

El separador **Avanzado** se utiliza sólo en trabajos paralelos. El separador **Avanzado** permite establecer algunas funciones particulares sobre el modo en que se comporta la etapa. Generalmente puede ignorar este separador y dejar que la etapa tome los valores predeterminados. Está pensado para que los usuarios avanzados ajusten las operaciones.

Definición de la correlación de juego de caracteres

Puede definir una correlación de juego de caracteres para una etapa. Realice esta tarea desde el separador **Soporte multilingüístico** en la página Etapa. El separador **Soporte multilingüístico** sólo aparece si se ha instalado el soporte multilingüístico.

Especifique la información mediante los siguientes campos:

- **Nombre de correlación que debe utilizarse con la etapa.** Define la correlación de juego de caracteres predeterminada para el proyecto o el trabajo. Puede cambiar la correlación seleccionando un nombre de correlación de la lista.
- **Mostrar todas las correlaciones.** Lista todas las correlaciones que se envían con InfoSphere DataStage.
- **Sólo correlaciones cargadas.** Muestra una lista de las correlaciones que están cargadas en ese momento.
- **Utilizar parámetro de trabajo...** Especifica los valores de los parámetros para el trabajo. Utilice el formato *#Param#*, donde *Param* es el nombre del parámetro de trabajo. La serie *#Parám#* se sustituye por el parámetro de trabajo cuando se ejecuta el trabajo.

Definición de datos de entrada

Al grabar datos en una tabla de una base de datos, la etapa Stored Procedure dispone de un enlace de entrada. Las propiedades de este enlace y las definiciones de columna de los datos se definen en la página Entrada del recuadro de diálogo Etapa STP de la interfaz gráfica de usuario.

Acerca de la página Entrada

La página Entrada contiene el campo **Nombre de entrada** y los separadores **General** y **Columnas**:

- **Nombre de entrada.** Nombre del enlace de entrada. La etapa Stored Procedure sólo puede contener un enlace de entrada.

El separador General

Utilice el separador **General** para establecer parámetros de conexión.

El separador contiene los campos siguientes:

- **Proveedor de la base de datos.** Tipo de base de datos relacional. En este release, las opciones son **Oracle**, **DB2**, **Sybase**, **Teradata** y **MSSQL Server**. El **Proveedor de base de datos** es necesario.
- **Origen de base de datos.** Nombre del alias de la base de datos.

Oracle

- Para Oracle, entre el nombre que ha creado mediante el asistente de configuración de Oracle.

DB2

- Para DB2, entre el nombre de la conexión, el nombre de la fuente de datos o el nombre de la base de datos.

SQL Server

- Para SQL Server, entre un nombre de ODBC DSN que describa la información sobre el controlador y el servidor.
 - **Nombre de usuario.** Nombre de usuario que debe utilizarse para conectar con la base de datos. Este usuario debe disponer de privilegios suficientes para acceder a la base de datos y tablas de origen y de destino especificadas. Este campo es necesario. No existe ningún valor predeterminado.
 - **Contraseña.** Contraseña asociada con el nombre de usuario especificado. Este campo es necesario. No existe ningún valor predeterminado.
 - **Nombre de la base de datos.** Base de datos del servidor especificado.

DB2

- Este campo no está disponible para DB2.

Nota: Si selecciona **DB2** como el **Proveedor de la base de datos** y pulsa **Aceptar** al volver a la página Etapa, el **Nombre de la base de datos** no estará activo.

SQL Server

- Este campo no está disponible para SQL Server.
- **ISO de transacción.** Nivel de aislamiento de la transacción (nivel ISO). El nivel ISO proporciona la coherencia y el control de concurrencia necesarios entre las transacciones del trabajo y otras transacciones para un rendimiento óptimo. Utilice **ISO de transacción** para establecer el nivel de aislamiento de la transacción para los enlaces de entrada y de salida, seleccionando un valor en la lista proporcionada. El nivel de ISO es específico de la base de datos y podría ser que cada base de datos no pudiera dar soporte a todas las opciones genéricas presentadas. Para obtener más información sobre la utilización de estos niveles, consulte la documentación de la base de datos. Utilice uno de los siguientes niveles de aislamiento de transacción:
 - **Ninguna. El valor predeterminado.**
 - **Sólo lectura.** No fija bloqueos de grabación. Este es el nivel de ISO permitido más eficaz para un rendimiento óptimo.
 - **Lectura no confirmada.** Fija bloqueos exclusivos para los datos modificados. Estos bloqueos se mantienen hasta que se ejecuta una confirmación o una retroacción. No obstante, otras transacciones pueden leer pero no modificar los cambios no confirmados. No se fijan otros bloqueos.
 - **Lectura confirmada.** Fija bloqueos exclusivos para los datos modificados y bloqueos que se pueden compartir para todos los demás datos. Este es el valor predeterminado. Los bloqueos exclusivos se mantienen hasta que se

ejecuta una confirmación o una retroacción. Los cambios no confirmados no pueden ser leídos por otras transacciones. Los bloqueos compartidos se liberan inmediatamente después de procesar los datos, permitiendo que otras transacciones puedan modificarlos.

- **Lectura repetible.** Idéntico a Serializable con la excepción de que podrían visualizarse las filas fantasma.
- **Serializable.** Fija bloqueos exclusivos para los datos modificados y bloqueos que se pueden compartir para todos los datos. Todos los bloqueos se mantienen hasta que se ejecuta una confirmación o una retroacción, evitando que otras transacciones modifiquen datos a los que se haya hecho referencia durante la transacción.

Oracle

Elija entre **Ninguno**, **Sólo lectura**, **Lectura confirmada** y **Serializable** para Oracle.

DB2

Elija entre **Ninguno**, **Lectura no confirmada**, **Lectura confirmada**, **Lectura repetible** y **Serializable** para DB2.

Sybase

Elija entre **Ninguno**, **Lectura no confirmada**, **Lectura confirmada** y **Serializable** para Sybase.

Teradata

ISO de transacción está inactivo si el **Proveedor de la base de datos** es **Teradata**. El RDBMS de Teradata da soporte a un modificador LOCKING en algunas sentencias SQL para permitir el control de los niveles de aislamiento de la transacción, pero las sentencias CALL y EXECUTE no dan soporte al modificador LOCKING. Para utilizar un método de bloqueo determinado, especifique el modificador LOCKING de las sentencias SQL en el cuerpo de la macro o procedimiento.

SQL Server

Elija entre **Ninguno**, **Lectura no confirmada**, **Lectura confirmada**, **Lectura repetible** y **Serializable** para SQL Server.

- **Modalidad de transacción.** Modalidad de conexión con el servidor de Teradata. Este campo sólo está activo si el **Proveedor de la base de datos** seleccionado es **Teradata** o un parámetro de trabajo.
 - Si el **Proveedor de la base de datos** es **Teradata**, las opciones para la **Modalidad de transacción** son **ANSI**, el valor predeterminado y **Teradata**. Este campo controla si la etapa Stored Procedure conecta con el servidor de Teradata en la modalidad de transacción de ANSI o Teradata. Una conexión en la modalidad de transacción de ANSI no puede llamar procedimientos almacenados que se han compilado en la modalidad de transacción de Teradata y una conexión en la modalidad de transacción de Teradata no puede llamar procedimientos almacenados que se han compilado en la modalidad de transacción de ANSI.
 - Si el **Proveedor de la base de datos** es un parámetro de trabajo, las opciones para la **Modalidad de transacción** serán **ANSI** y **Nativa**. La modalidad de

transacción Nativa es equivalente a la modalidad de transacción de Teradata si el proveedor de la base de datos durante el tiempo de ejecución es Teradata. El campo de la modalidad de transacción no afecta a los demás proveedores.

Nota: Si se produce un error en la modalidad de transacción de Teradata, Teradata retrotraerá automáticamente la transacción actual. En la modalidad de transacción de ANSI, un error no afecta a la transacción actual.

- **Juego de caracteres de cliente.** Juego de caracteres del cliente de Teradata que debe utilizarse al conectar con el servidor de Teradata. Este campo sólo está activo si el **Proveedor de la base de datos** es **Teradata** o un parámetro de trabajo. El valor predeterminado es **Predeterminado**.
- **Descripción.** Descripción opcional de la etapa Stored Procedure.

Separador Columnas

Este separador contiene las definiciones de columna para los datos grabados en la tabla. Las definiciones de columna aparecen en el mismo orden que en la cuadrícula Columnas. El separador **Columnas** se comporta de la misma forma que el separador **Columnas** de la etapa ODBC.

Teradata

En el separador **Columnas** de la página Entrada, el botón **Cargar** sólo carga datos del separador **Columnas** del registro Definiciones de columna. Si necesita cargar los parámetros de entrada de una macro, de un procedimiento almacenado o de una función definida por el usuario, puede importar el objeto con la etapa Teradata API. La etapa Teradata API importa los parámetros de entrada en el separador **Columnas**. El recurso de importación de Definiciones de procedimientos almacenados sólo importa parámetros de entrada en el separador **Parámetros**.

Definición de datos de salida

Cuando se utiliza la etapa Stored Procedure para ejecutar un procedimiento al principio de una vía de acceso de trabajo, la etapa ejecuta el procedimiento almacenado especificado una vez y envía una sola fila o varias filas de bajada hacia el enlace de salida. Los enlaces de salida especifican los datos que está extrayendo de una base de datos.

Acerca de la página Salida

La página Salida contiene un campo y los separadores **General** y **Columnas**.

- **Nombre de salida.** Nombre del enlace de salida.

Separador General

Este separador contiene los parámetros siguientes:

- **Estado de procedimiento para enlace.** Disposición de la información de retorno a partir de la ejecución del procedimiento almacenado. Si se selecciona **Estado de procedimiento para enlace** (el valor predeterminado), se dará salida al código de retorno del procedimiento almacenado y al mensaje de error mediante el enlace de salida como datos de columna. Las primeras dos columnas del enlace de salida mantienen la salida del código de retorno del procedimiento almacenado y la información del mensaje. Estas columnas tienen el campo de descripción establecido como PROC.RTN.CODE y PROC.RTN.MESS y deben ser

las primeras dos columnas definidas en el enlace de salida. La etapa crea o elimina estas columnas en el enlace de salida en función de si la opción **Estado de procedimiento para enlace** se ha seleccionado.

SQL Server

SQL Server opera de forma ligeramente distinta. Si se selecciona **Estado de procedimiento para enlace** (el valor predeterminado), se dará salida al código de retorno del procedimiento almacenado mediante el enlace de salida como datos de columna. La primera columna del enlace de salida mantiene la salida del código de retorno del procedimiento almacenado. Esta columna tiene el campo de descripción establecido como PROC.RTN.CODE y debe ser la primera columna definida en el enlace de salida. La etapa crea o elimina estas columnas en el enlace de salida en función de si la opción **Estado de procedimiento para enlace** se ha seleccionado.

Los metadatos de la columna se crean tal y como se muestra en la tabla siguiente.

Tabla 4. Metadatos de columna con SQL Server

Nombre	Tipo de datos	Descripción
ProcCode	Integer	PROC.RTN.CODE
ProcMess	Char(128)	PROC.RTN.MESS

Sybase

Sybase no puede devolver información sobre el estado para cada fila si el procedimiento devuelve un conjunto de resultados. El estado no se devuelve hasta que se han captado todas las filas, por lo que el código de retorno y el mensaje de error no se pueden correlacionar en el flujo de datos.

- **Recuento de actividades para enlace.** Si selecciona esta opción, se dará salida al número de filas afectadas por la macro o el procedimiento mediante el enlace de salida.

Teradata

El **Recuento de actividades para enlace** sólo está activo para Teradata o para un parámetro de trabajo.

- **Código de retorno de EOD de procedimiento.** Código que devuelve el procedimiento cuando alcanza el fin de los datos. Si selecciona la opción **El procedimiento devuelve varias filas**, debe proporcionar el código que debe devolver el procedimiento cuando finaliza la captación de filas. La etapa da salida a varias filas mediante el enlace de salida hasta que el conjunto de resultados del procedimiento se ha agotado y el procedimiento devuelve este código a la etapa Stored Procedure. El código desencadena un bucle de ejecución en la etapa que continúa dando salida a filas hasta que el procedimiento llamado provoca que la etapa Stored Procedure se detenga.

Oracle

El **Código de retorno de EOD de procedimiento** sólo está activo para Oracle. .

- **El procedimiento devuelve varias filas.** Indica que el procedimiento devuelve más de una fila. Seleccione la opción **El procedimiento devuelve varias filas** si el procedimiento devuelve varias filas. Desmarque la opción **El procedimiento**

devuelve varias filas si el procedimiento devuelve sólo una fila. De forma predeterminada, la opción **El procedimiento devuelve varias filas** está desmarcada.

Oracle

Para obtener más información sobre Oracle, consulte Gestión de cursores.

Sybase

Sybase utiliza parámetros de salida para devolver datos de columna del resultado establecido hasta que se captan todas las filas.

Teradata

La opción **El procedimiento devuelve varias filas** se comporta como en el caso de DB2. De forma predeterminada, la opción **El procedimiento devuelve varias filas** está desmarcada.

SQL Server

De forma predeterminada, la opción **El procedimiento devuelve varias filas** no está seleccionada. Si no se selecciona, sólo se devolverá la primera fila de los procedimientos almacenados que devuelven varias filas. Este proceso es similar a una búsqueda única. Si se selecciona, cada llamada de un procedimiento almacenado podrá dar salida a varias filas mediante el enlace de salida. Este proceso es equivalente a una búsqueda de cursor. Si un procedimiento almacenado devuelve varios conjuntos de resultados, el número de columnas de cada conjunto de resultados debe coincidir con el número de parámetros de salida del separador **Parámetros** y el tipo de datos de las columnas debe ser compatible con el tipo de datos de los parámetros de salida.

- **Descripción.** Descripción opcional del enlace de salida.

La utilización de **Código de retorno de EOD de procedimiento**, la grabación de procedimientos que captan datos de los cursores y la devolución de varias filas, son específicos de la base de datos.

Separador Columnas

La página del separador Columnas se comporta de la misma forma que el separador **Columnas** de la etapa ODBC y especifica qué columnas se agregan.

Teradata

En el separador **Columnas** de la página Salida, aparece la columna **ProcCount** si se selecciona **Recuento de actividades para enlace**.

El botón **Cargar** sólo se carga desde el separador **Columnas** del registro **Definiciones de tabla**. Si necesita cargar los parámetros de salida de un procedimiento almacenado o de una función definida por el usuario, puede importar el objeto con la etapa Teradata API. La etapa Teradata API importa todos los parámetros en el separador **Columnas**. El recurso de importación de Definiciones de procedimientos almacenados sólo importa parámetros en el separador **Parámetros**.

Enlaces

La etapa Stored Procedure sólo puede tener un enlace de entrada y/o un enlace de salida. Los enlaces de referencia se consideran enlaces de entrada y no están soportados por; esto se debe a una convención de etapa activa.

Sin embargo, es posible emular un enlace de referencia utilizando la media secuencia de la etapa y definiendo parámetros de entrada como claves de la cláusula WHERE en el procedimiento almacenado y definiendo parámetros de salida para mantener la fila resultante seleccionada. Consulte "**Emulación de enlaces de referencia**".

Uso de media secuencia

Si la etapa Stored Procedure se coloca a media secuencia (el trabajo contiene un enlace de entrada y de salida), los datos de columna de los parámetros de salida llegan de la ejecución del procedimiento. Si se ha seleccionado **Reenviar datos de fila**, los datos de otras columnas se transfieren al enlace de salida.

Metadatos de columna

La etapa Stored Procedure crea metadatos de columna en el enlace de salida copiando las columnas de entrada en las columnas de salida cuando se selecciona **Reenviar datos de fila**. Si el **Tipo de parámetro** es **Salida, Entrada/Salida o Función**, los metadatos de columna asociados deben existir en el enlace de salida. De forma predeterminada, cualquier procedimiento que defina parámetros que devuelvan datos necesita almacenamiento al que devolver los datos, lo que explica la necesidad de que existan dichos tipos de parámetro en el enlace de salida.

Datos de columna

Se aplican las siguientes reglas:

- Si **Tipo de parámetro** para la columna se establece en **Salida, Entrada/Salida o Función**, los datos de columna vienen de la ejecución del procedimiento.
- Cuando se selecciona **Reenviar datos de columna**:
 - Si **Tipo de parámetro** para la columna se establece en **Entrada**, los datos de columna se pasan del enlace de entrada al enlace de salida.
 - Si la columna no aparece en la cuadrícula de parámetros (y, por consiguiente, no se utiliza como un parámetro), los datos de columna se pasan del enlace de entrada al enlace de salida.

Emulación de enlaces de referencia

Aunque el escenario de enlace de referencia tradicional no se aplica a las etapas activas, STP puede manejar búsquedas de referencia utilizando un paradigma diferente. El ejemplo siguiente ilustra el uso de búsquedas de referencia con la etapa Stored Procedure.

En este ejemplo que utiliza Oracle, el objetivo consiste en unir tabla de empleados con la tabla de departamentos y proyectar el nombre del empleado junto con el nombre del departamento en el que trabaja este empleado. En un trabajo tradicional, `Sequential_File_0` introduce registros de empleado en la etapa `Transfer_1`, y la etapa `Oracle_OCI_3` selecciona el nombre del departamento en una lectura con clave de la tabla DEPT utilizando el número de departamento del registro de empleado. `Transformer_1` introduce el par nombre de empleado/departamento en la etapa `Sequential_File_2`. Esto se refleja en la

siguiente figura.

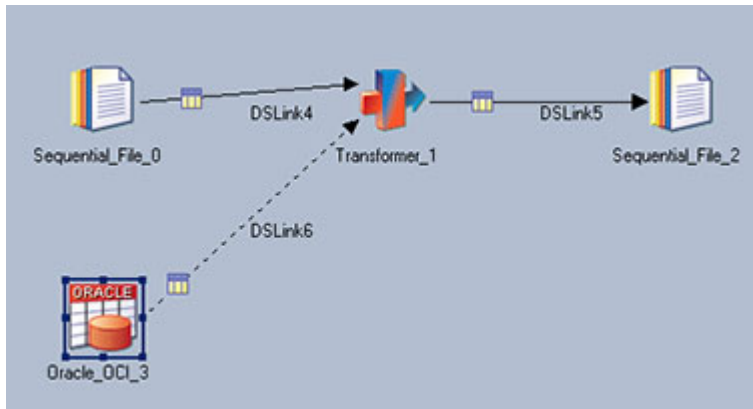


Figura 1. Trabajo de referencia tradicional

Para desarrollar este trabajo utilizando la etapa Stored Procedure, Sequential_File_0 introduce registros de empleado en la etapa Stored Procedure. La etapa Stored Procedure ejecuta el procedimiento DSSP para cada fila. La etapa Stored Procedure pasa el número de departamento del registro de empleado como un parámetro y devuelve el nombre de departamento como un parámetro de salida de la búsqueda con clave efectuada por el procedimiento. Al seleccionar **Reenviar datos de fila**, todos los datos de la columna de entrada se copian en los datos de la columna de salida (incluido el nombre del empleado). La fila que consta de las columnas de registro de empleados y el nombre de departamento del registro de departamentos se envía a Transformer_10. A continuación el transformador proyecta el par nombre de empleado/nombre de departamento hacia la etapa Sequential_File_2. La etapa Transformer se utiliza para limitar la selección de columnas en el enlace de salida, porque todas las columnas se copian del enlace de entrada en el enlace de salida cuando se selecciona **Reenviar datos de fila**.

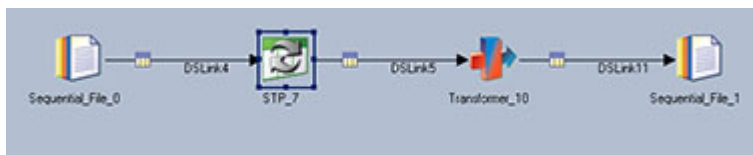


Figura 2. Trabajo de referencia de la etapa Stored Procedures de InfoSphere DataStage

Para este ejemplo, el contenido del separador **Columnas** de la página Entrada de la etapa Stored Procedure se muestra en la tabla siguiente.

Tabla 5. Ejemplo del separador **Columnas** de la página Entrada

Nombre de la columna	Clave	Tipo SQL	Longitud	Escala	Con posibilidades de nulos	Visualización
EMPNO	Sí	Decimal	4		No	6
ENAME	No	VarChar	10		Sí	10
JOB	No	VarChar	9		Sí	4
MGR	No	Decimal	4		Sí	6
SAL	No	Decimal	7		Sí	9

Tabla 5. Ejemplo del separador **Columnas** de la página Entrada (continuación)

Nombre de la columna	Clave	Tipo SQL	Longitud	Escala	Con posibilidades de nulos	Visualización
COMM	No	Decimal	7		Sí	9
DEPTNO	No	Decimal	2		No	4

El contenido del separador **Columnas** de la página Salida de la etapa Stored Procedure se muestra en la tabla siguiente.

Tabla 6. Ejemplo del separador **Columnas** de la página Salida

Nombre de la columna	Derivación	Clave	Tipo SQL	Longitud	Escala	Con posibilidades de nulos	Visualización
ProcCode		No	Integer	10		No	10
ProcMess		No	VarChar	128		No	32
EMPNO		Sí	Decimal	4		No	6
ENAME		No	VarChar	10		Sí	10
JOB		No	VarChar	9		Sí	9
MGR		No	Decimal	4		Sí	6
SAL		No	Decimal	7		Sí	9
COMM		No	Decimal	7		Sí	9
DEPTNP		No	Decimal	2		No	4
DNAME		No	VarChar	14		Sí	14

El contenido de las columnas contenidas en el enlace de salida de la etapa Transformer se muestra en la tabla siguiente.

Tabla 7. Ejemplo del contenido de columnas en el enlace de salida de la etapa Transformer

Nombre de la columna	Clave	Tipo SQL	Longitud	Escala	Con posibilidades de nulos	Visualización
ENAME		VarChar	10		Sí	10
DNAME		VarChar	14		Sí	14

Para lograr el objetivo, establezca las propiedades de la etapa Stored Procedure de la siguiente manera:

- Separador **General** de la página Entrada:
 - **Reenviar datos de fila** está seleccionado
 - **Ejecutar procedimiento para cada fila** está seleccionado
- Separador **Sintaxis** de la página Etapa:
 - **Tipo de procedimiento** se ha establecido en **Transformación**
- Separador **Parámetros** de la página Etapa

Tabla 8. Propiedades del separador **Parámetros**

Nombre de columna	Valor del parámetro	Tipo de parámetro
DEPTNO	?	Entrada
DNAME	?	Salida

El procedimiento almacenado DSSP21 está formado por el código siguiente:

```
PROCEDURE DSSP21 (DEPTNO_ARG IN DEPT.DEPTNO%TYPE,  
                 DNAME_ARG OUT DEPT.DNAME%TYPE)  
AS BEGIN  
    SELECT DNAME INTO DNAME_ARG  
    FROM DEPT WHERE DEPTNO = DEPTNO_ARG;  
END;
```

Capítulo 3. Información específica sobre bases de datos adicionales

Determinados temas, como la utilización del **Código de retorno de EOD de procedimiento**, la grabación de procedimientos que captan datos de los cursores y la devolución de varias filas, son específicos de la base de datos. Este apéndice contiene información específica sobre las siguientes bases de datos:

- "Oracle"
- "DB2"
- "Sybase"
- "Teradata"
- "SQL Server"

Oracle

Para obtener información específica sobre la grabación de procedimientos PL/SQL, consulte la documentación de la base de datos. Las siguientes características son específicas de Oracle.

Paquetes

Además de ejecutar procedimientos almacenados, la etapa Stored Procedure soporta y ejecuta funciones y procedimientos almacenados en paquetes. Establezca el **Nombre del procedimiento** del separador **Sintaxis** de la página Etapa en paquete.nombreprocedimiento para ejecutar un procedimiento de un paquete.

Manejo de errores de PL/SQL

Utilizando la posibilidad de manejo de errores de la etapa Stored Procedure puede comprobar si hay errores de base de datos específicos de Oracle, definir sus propios códigos de errores o ambas cosas. La llamada de PL/SQL `RAISE_APPLICATION_ERROR()` devuelve el estado de la base de datos de Oracle o definido por el usuario a la etapa Stored Procedure. El siguiente fragmento de PL/SQL captura varios estados de error:

```
EXCEPTION
WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR (-20001,'No data found - just warning');
WHEN ZERO_DIVIDE THEN
    /* return Oracle database error (ORA-01476 and message, treat as warning*/
WHEN OTHERS THEN
    RAISE_APPLICATION_ERROR (-20002,"SOMETHING BAD HAPPENED - ABORT');
```

Proporcione los valores siguientes para **Errores definidos por el usuario** en el separador **Códigos de error** de la página **Etapa**:

- Errores muy graves: 20002
- Avisos: 20001 01476

Puede entrar varios códigos de error en cada categoría. Sepárelos mediante espacios.

En este ejemplo, NO DATA FOUND y ZERO DIVIDE se tratan como avisos, el trabajo sigue ejecutándose y la etapa procesa las filas subsiguientes. El resto de los errores son muy graves y hacen que el trabajo termine anormalmente.

Si selecciona **Estado de procedimiento para enlace** en el separador **General** de la página Salida, RAISE_APPLICATION_ERROR() realiza una retrotracción sobre las filas pendientes. Los códigos y mensajes de error de RAISE_APPLICATION_ERROR() se pasan en sentido descendente al enlace de salida como las primeras dos columnas.

Gestión de cursores

Mediante una gestión de cursores correcta, puede seleccionar más de una fila o devolver conjuntos de resultados.

Si el procedimiento almacenado devuelve varias filas, debe codificar el procedimiento PL/SQL para que pueda interactuar con la etapa Stored Procedure. Debe declarar explícitamente el cursor como externo al procedimiento que está ejecutando la etapa Stored Procedure para procesar las filas.

La gestión del cursor consta de dos partes:

1. El diseñador debe construir el procedimiento para inicializar el cursor con la sentencia OPEN y captar sólo una fila. Si el procedimiento decide detener la captación de filas, debe aumentar el error "Código de retorno de EOD de procedimiento" para notificar que la etapa Stored Procedure detenga la salida de filas y libere el cursor con la sentencia CLOSE. El procedimiento almacenado no debería contener un bucle FETCH en el código PL/SQL, ya que InfoSphere DataStage necesita procesar una fila cada vez.

El ámbito del cursor debe ser externo al procedimiento, ya que la etapa Stored Procedure ejecuta el procedimiento hasta que el procedimiento devuelve el "Código de retorno de EOD de procedimiento." Si el cursor no es externo, cada ejecución del procedimiento devolverá la primera fila del conjunto de resultados, ya que cada vez que se invoca el procedimiento, el cursor se cierra y se vuelve a abrir. Para declarar el cursor como externo, ubíquelo en un paquete. Así se modifica el ámbito del cursor para que ya no esté limitado a un bloque PL/SQL determinado. Si abre el cursor empaquetado, éste permanecerá abierto en la parte superior de las ejecuciones del procedimiento que realiza la etapa Stored Procedure.

2. Debe proporcionar el **Código de retorno de EOD de procedimiento** en el enlace de salida. Indica que la etapa Stored Procedure debe ejecutar el procedimiento hasta que dicho procedimiento devuelva el código que se ha entrado mediante RAISE_APPLICATION_ERROR(). Este código se devuelve cuando el cursor del procedimiento devuelve %NotFound después de una llamada de captación.

En este ejemplo, el procedimiento almacenado explora la tabla completa y devuelve 20001 cuando se agotan todas las filas. El EOD 20001 puede devolverse en cualquier momento si el usuario no desea leer toda la tabla pero desea devolver, por ejemplo, las primeras diez líneas, proporcionando al usuario el control total sobre el proceso. El siguiente fragmento de PL/SQL gestiona el cursor de la forma adecuada:

```
PACKAGE BODY DSSP3 AS
  CURSOR GET_COL IS SELECT DEPTNO,DNAME,LOC FROM DEPT;
  PROCEDURE DSSP3 (DEPTNO_ARG OUT DEPT.DEPTNO%TYPE,
                  DNAME_ARG OUT DEPT.DNAME%TYPE,
```

```

        LOC-ARG OUT DELP.LOC%TYPE)
IS BEGIN
  IF NOT GET_COL%ISOPEN THEN
    OPEN GET_COL;
  END IF;
  FETCH GET_COL INTO DEPTNO_ARG,DNAME_ARG,LOC_ARG;
  IF GET_COL%NOTFOUND THEN
    CLOSE GET-COL
    RAISE_APPLICATION_ERROR(-20001,'EOD FOUND');
END;
```

En la etapa Stored Procedure, el **Código de retorno de EOD de procedimiento** se establece en 20001.

SELECT INTO

Acerca de esta tarea

Puede seleccionar una sola fila utilizando STP. Borre **El procedimiento devuelve varias filas**. Grabe la sentencia SELECT INTO con una cláusula WHERE en una KEY para seleccionar una o cero (NO_DATA_FOUND) filas y para asignar los valores seleccionados a variables de salida. BULK COLLECT y las variables de enlace de matriz no están soportadas; por consiguiente, SELECT INTO debe devolver sólo una fila.

```

PROCEDURE DSSP5 (DEPTNO_ARG OUT DEPT.DEPTNO%TYPE,
                DNAME_ARG OUT DEPT.DNAME%TYPE,
                LOC_ARG OUT DEPT.LOC%TYPE)
AS BEGIN
  SELECT DEPTNO, DNAME, LOC INTO
    DEPTNO_ARG, DNAME_ARG, LOC_ARG
  FROM DEPT WHERE DEPTNO=10;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR (-20001, 'NO DATA FOUND');
  WHEN OTHERS THEN
    RAISE_APPLICATION_ERROR (-200002, 'SOMETHING BAD HAPPENED');
END;
```

Sentencia de llamada definida por el usuario

Acerca de esta tarea

Si desea entrar su propia sentencia de llamada para el procedimiento, no seleccione **Generar llamada de procedimiento**. Entre un bloque PL/SQL anónimo que contenga la sintaxis de invocación del procedimiento que desea ejecutar.

DB2

Las siguientes características son específicas de DB2.

Manejo de errores de SQL

Utilizando la posibilidad de manejo de errores de la etapa Stored Procedure puede comprobar si hay errores de base de datos específicos de DB2, definir sus propios códigos de errores o ambas cosas. Declare variables de condición y un manejador para cada condición. Más que comunicarse directamente con los programadores, SQL devuelve códigos de error al programa de aplicación en forma de SQLCODE y SQLSTATE. La etapa Stored Procedure utiliza la sentencia de control RESIGNAL para devolver un SQLSTATE definido por el usuario y un mensaje de error

definido por el usuario al proceso que llama. El ejemplo siguiente define todos los elementos principales de este manejo de errores definido por el usuario:

```
DECLARE rec_cnt int;
DECLARE SQLSTATE CHAR(5);
DECLARE EXIT HANDLER FOR SQLSTATE '75000'
RESIGNAL SQLSTATE '75000' SET MESSAGE_TEXT = 'Invalid department number.';
SELECT count (*) INTO rec_cnt FROM DB2ADMIN.STPTEST WHERE STPTEST.ACHAR='XYZ';
IF rec_cnt = 0
THEN SIGNAL SQLSTATE '75000';
END IF;
```

Proporcione los valores siguientes en el separador **Códigos de error** de la página Etapa:

- Errores muy graves: 2002
- Avisos: 2001 1479

donde 2001 significa NO DATA FOUND (no se han encontrado datos) y 1476 significa ZERO DIVIDE (división cero).

En este ejemplo, NO DATA FOUND y ZERO DIVIDE se tratan como avisos, el trabajo sigue ejecutándose y la etapa procesa las filas subsiguientes. El resto de los errores son muy graves y hacen que el trabajo termine anormalmente.

Si selecciona **Estado de procedimiento para enlace** en el separador **General** de la página Salida, la sentencia de control RESIGNAL realiza una retrotracción sobre las filas pendientes. Los códigos y mensajes de error de la sentencia de control RESIGNAL se pasan en sentido descendente al enlace de salida como las primeras dos columnas.

Gestión de cursores

Mediante una gestión de cursores correcta, puede seleccionar más de una fila o devolver conjuntos de resultados.

Seleccione **CursorOutput** como el **Tipo de parámetro**. La etapa Stored Procedure no requiere ninguna especificación para el **Código de retorno de EOD de procedimiento**.

Codifique el procedimiento SQL para que interactúe con la etapa Stored Procedure de InfoSphere DataStage de la siguiente forma.

- La etapa Stored Procedure no requiere que la declaración del cursor sea externa con respecto al procedimiento que se ejecuta para procesar las filas. A diferencia de Oracle, no se capta una fila en cada ocasión, por lo que se da salida a todas las filas.
- El procedimiento almacenado no debería contener un bucle FETCH en el código SQL, ya que el código del procedimiento almacenado realiza la captación de forma interna y transfiere las filas al cliente de DataStage de una a una.
- No cierre el cursor dentro del procedimiento almacenado; STP será responsable. La etapa Stored Procedure capta todos los registros hasta SQL_NO_DATA_FOUND. A continuación, cierre el cursor.

De forma predeterminada, la etapa Stored Procedure sólo devuelve una fila del cliente a menos que seleccione la opción **El procedimiento devuelve varias filas**. Si

se selecciona este recuadro de selección, no se generará ningún mensaje como en el caso de Oracle, ya que EOD no es necesario para el conector de procedimiento almacenado de DB2.

En este ejemplo, el procedimiento almacenado obtiene todas las filas con columnas seleccionadas de la tabla DEPT en el cursor Cursor1. La etapa Stored Procedure de DataStage capta todas las filas y no se requiere ningún bucle FETCH. El siguiente fragmento de SQL gestiona el cursor de la forma adecuada:

```
CREATE PROCEDURE DB2ADMIN.SPPR2 ()
P1: BEGIN
    DECLARE cursor1 CURSOR WITH RETURN FOR
        SELECT DNO, DNAME, LOC FROM DEPT;
    OPEN cursor1;
END P1
```

SELECT INTO

Acerca de esta tarea

Puede seleccionar una sola fila o datos específicos utilizando STP. Grabe la sentencia SELECT INTO con una cláusula WHERE en una KEY para seleccionar una o cero (NO_DATA_FOUND) filas y para asignar los valores seleccionados a variables de salida.

Este ejemplo utiliza una sentencia SELECT INTO para devolver tres columnas distintas de una tabla a parámetros de salida.

```
CREATE PROCEDURE SAMPLE.SP11LED (OUT DEPTNO INT,
                                OUT DEPTNAME VARCHAR (30),
                                OUT LOC VARCHAR (30))
P1: BEGIN
    DECLARE EXIT HANDLER FOR NOT FOUND
    SELECT DNO, DNAME, LOC INTO
        DEPTNO, DEPTNAME, LOC
    FROM DEPT
    WHERE DNO = 101;
END P1
```

Sentencia de llamada definida por el usuario

Acerca de esta tarea

Si desea entrar su propia sentencia de llamada para el procedimiento, no seleccione **Generar llamada de procedimiento**. Entre un bloque PL/SQL anónimo que contenga la sintaxis de invocación del procedimiento que desea ejecutar.

Sybase

Las siguientes características son específicas de Sybase.

Tamaño de transacción

Si **Tamaño de transacción** es igual o mayor que 1, la etapa Stored Procedure establece la modalidad de transacción compatible con ANSI, también denominada modalidad encadenada, para que se inicie implícitamente una transacción

- Una vez establecida la conexión
- Antes de la primera sentencia de este tipo una vez confirmada o retrotraída una transacción.

En la modalidad encadenada, la etapa Stored Procedure controla la confirmación y la retrotracción, pero el procedimiento desde establecerse de modo que se ejecute en cualquier modalidad. Para obtener más información sobre cómo establecer esta modalidad en un procedimiento, consulte la documentación de Sybase

Consulte **Tamaño de transacción**

Teradata

La etapa Stored Procedure soporta llamadas a macros de Teradata, procedimientos almacenados (internos y externos), funciones escalares y funciones de tabla. Resulta más adecuado invocar a las funciones de totales desde la etapa Teradata API, porque la sentencia SELECT generada debe consultar una tabla.

Aunque la etapa Teradata API no se ha diseñado para invocar macros de Teradata, procedimientos almacenados ni funciones definidas por el usuario, es posible invocar a dichos objetos con la opción "SQL definido por el usuario". Sin embargo, existen algunas restricciones a la hora de invocar a dichos objetos desde la etapa Teradata API.

- La etapa Teradata API sólo puede importar parámetros de entrada de una macro en el separador Columnas.
- La etapa Teradata API sólo puede devolver una fila de una macro invocada en el en la ce de referencia de la etapa.
- La etapa Teradata API sólo puede devolver un conjunto de resultados procedentes de una macro invocada en el enlace de salida de la etapa.
- Si se invoca a una macro desde el enlace de referencia o de salida de la etapa Teradata API, la primera sentencia de la macro debe ser una sentencia SELECT.
- La etapa Teradata API sólo conecta en modalidad de transacción ANSI, por lo que sólo puede llamar a procedimientos que se han compilado en modalidad de transacción ANSI. Invocar a un procedimiento almacenado que se ha compilado en modalidad de transacción Teradata devolverá un error con el mensaje "Modalidad de sesión no válida para la ejecución del procedimiento".
- La etapa Teradata API sólo puede llamar a procedimientos almacenados que tenga todo parámetros IN o todo parámetros INOUT y OUT, y el número de parámetros debe coincidir con el número de columnas del separador Columnas. A un procedimiento almacenado que tenga todo parámetros IN se le debe invocar desde el enlace de entrada de la etapa. Es parecido a llamar a un procedimiento de destino en la etapa Stored Procedure. A un procedimiento almacenado que tenga todo parámetros OUT se le debe invocar desde el enlace de salida de la etapa. Es parecido a llamar a un procedimiento de origen en la etapa Stored Procedure. A un procedimiento almacenado que tenga todo parámetros INOUT y OUT se le debe invocar desde el enlace de referencia de la etapa. Los parámetros INOUT deben tener el atributo Key (clave) seleccionado en el separador Columnas, y los parámetros OUT no deben tener el atributo Key seleccionado. Para llamar a un procedimiento almacenado desde el enlace de referencia o desde el enlace de salida se necesita Teradata API versión 1.2.5 o posterior.

La etapa ODBC soporta llamadas a procedimientos almacenados y macros de Teradata, pero no soporta llamadas a los mismos en un enlace de referencia.

La etapa Stored Procedure no sufrirá las restricciones de las etapas Teradata API y ODBC, por lo que será el método preferido para llamar a macros, procedimientos almacenados, funciones escalares y funciones de tabla de Teradata. Además, puesto

que la etapa Stored Procedure permitirá llamadas a macros y funciones de tabla que devuelvan más de una fila por cada fila de entrada, el usuario tiene capacidad para realizar una búsqueda de cursor. Actualmente la capacidad de realizar una búsqueda de cursor en Teradata no está disponible en ningún otro componente de InfoSphere DataStage.

La tabla siguiente contiene una lista del componente de InfoSphere DataStage recomendado para cada situación.

Caso del usuario

Etapa de conectividad recomendada

Llamar a una macro de Teradata

Procedimiento almacenado

Llamar a un procedimiento almacenado de Teradata

Procedimiento almacenado

Llamar a una función escalar definida por el usuario

Procedimiento almacenado

Llamar a una función de tabla definida por el usuario

Procedimiento almacenado

Llamar a una función de totales definida por el usuario

Teradata API

Importar parámetros en el separador Parámetros de la etapa Stored Procedure

Importación de definiciones de procedimiento almacenado

Importar parámetros de salida de macro en el separador Columnas de Stored Procedure o Teradata API

Importación de definiciones de procedimiento almacenado

Importar parámetros de entrada de macro en el separador Columnas de Stored Procedure o Teradata API

Importación de la etapa Teradata API

Las siguientes características son específicas de Teradata.

Importación de procedimientos almacenados

Acerca de esta tarea

Si intenta importar un procedimiento almacenado en lugar de una macro, debe seleccionar **Inhabilitar conversión de CALL a EXEC** en la pantalla **Opción de controlador de ODBC de Teradata**. Además, debe establecer la **Modalidad de sesión** en la modalidad de transacción que se ha utilizado para compilar el procedimiento almacenado.

Los procedimientos almacenados y las funciones definidas por el usuario disponen de esquemas predefinidos. Por consiguiente, todos sus parámetros se importan en el separador **Parámetros** con el **Tipo de entrada/salida** establecido de forma adecuada. El tipo **RETURN** indica que el parámetro es el valor de retorno de una función escalar o de totales. Sólo un parámetro dispondrá del tipo **RETURN**.

Importación de macros

Al importar una macro, el recurso de importación muestra los parámetros de entrada de la macro. Sólo se puede determinar un esquema de salida de la macro ejecutando la macro de forma que el recurso de importación le ofrezca la opción de

suministrar una lista de argumentos. Si pulsa **Cancelar** no se ejecutará la macro, de forma que sólo se importarán los parámetros de entrada.

Si la macro se ejecuta durante la importación, el separador **Columnas** de la **Definición de tabla** mostrará las columnas de salida de la macro. El separador **Parámetros** muestra los parámetros de entrada de la macro.

Opciones para el juego de caracteres del cliente

Las opciones para el **Juego de caracteres de cliente** son:

- Valor predeterminado
- Automático
- ASCII (Unix y Windows)
- EBCDIC (IBM)
- EBCDIC037_0E (IBM EE.UU. y Canadá)
- EBCDIC273_0E (IBM Austria y Alemania)
- EBCDIC277_0E (IBM Dinamarca y Noruega)
- HANGULEBCDIC933_1II (IBM coreano)
- HANGULKSC5601_2R4 (Unix y Windows coreanos)
- KANJIEBCDIC5026_0I (IBM japonés)
- KANJIEUC_0U (Unix japonés)
- KANJISJIS_0S (Windows japonés)
- LATIN1_0A (ISO 8859-1 Unix Latin 1)
- LATIN9_0A (ISO 8859-15 Unix Latin 9)
- LATIN1252_0A (Windows Latin)
- SCHGB2312_1T0 (Unix y Windows en chino simplificado)
- CHBIG5_1R0 (Unix y Windows en chino tradicional)
- TCHEBCDIC937_3IB (IBM en chino tradicional)
- UTF8 (Universal Transformation Format)

Si el **Proveedor de base de datos** es un parámetro de trabajo, los juegos de caracteres de cliente específicos de Teradata no aparecerán en la lista de juegos de caracteres disponibles. En la lista sólo aparecerán **Proveedor** y **Automático**. No obstante, aún se puede especificar un juego de caracteres de cliente específico de Teradata o un parámetro de trabajo en el campo. Este campo no afecta a otros proveedores.

Valor predeterminado especifica que el juego de caracteres predeterminado del cliente de Teradata debería utilizarse para conectar con el servidor de Teradata.

Automático especifica que el juego de caracteres de cliente de Teradata debería establecerse automáticamente en función de la correlación de etapas del separador **Soporte multilingüístico**, tal y como se muestra en la siguiente tabla.

Correlación de etapas de soporte multilingüístico de trabajo de servidor	Correlación de etapas de soporte multilingüístico de trabajo paralelo	Juego de caracteres de cliente de Teradata
ASCII	ASCL_ASCII	ASCII
BIG5	ASCL_BIG5	TCHBIG5_1R0
EBCDIC	ASCL_EBCDIC	EBCDIC

Correlación de etapas de soporte multilingüístico de trabajo de servidor	Correlación de etapas de soporte multilingüístico de trabajo paralelo	Juego de caracteres de cliente de Teradata
EBCDIC-037	ASCL_EBCDIC-037	EBCDIC037_0E
EBCDIC-1026	ASCL_EBCDIC-1026	KANJIEBCDIC5026_0I
	IBM273	EBCDIC273_0E
	IBM277	EBCDIC277_0E
EBCDIC-IBM933	ASCL_EBCDIC-IBM933	HANGULEBCDIC933_1II
EBCDIC-IBM937	ASCL_EBCDIC-IBM937	TCHEBCDIC937_3IB
GB2312	ASCL_GB2312	SCHGB2312_1T0
ISO8859-1	ASCL_ISO8859-1	LATIN1_0A
ISO8859-15	ASCL_ISO8859-15	LATIN9_0A
JPN-EUC	ASCL_JPN-EUC	KANJIEUC_0U
KSC5601	ASCL_KSC5601	HANGULKSC5601_2R4
MS1252	ASCL_MS1252	LATIN1252_0A
SHIFT-JIS	ASCL_SHIFT-JIS	KANJISJIS_0S
UTF8	UTF-8	UTF8

ASCII, EBCDIC, y UTF8 siempre están instalados en los Juegos de caracteres del cliente de Teradata y siempre están disponibles. Los demás Juegos de caracteres del cliente deben instalarse en el servidor de Teradata para que puedan utilizarse.

Códigos de error

Puede especificar que los códigos de error de Teradata se traten como errores muy graves o avisos, tal y como se realiza para otros proveedores de bases de datos. Si trata un error como un aviso en la modalidad de transacción de Teradata, tenga en cuenta que las filas de entrada anteriores pueden resultar afectadas por el error. Si se produce un error en la modalidad de transacción de Teradata, Teradata retrotraerá automáticamente la transacción actual. En la modalidad de transacción ANSI, un error no afecta a la transacción actual. Si un error en la modalidad de transacción de Teradata provoca que las filas no confirmadas se retrotraigan, la etapa Stored Procedure emitirá un mensaje informativo en el registro Director que especificará el número de filas de entrada afectadas. La etapa Stored Procedure también emite un mensaje informativo para el mensaje de error de retrotracción de Teradata, "RDBMS code 3514: User-generated transaction ABORT." Este mensaje se puede convertir en un mensaje de aviso o de error muy grave si indica 3514 en la lista de avisos o de errores muy graves.

Recuento de actividades para enlace

Si selecciona esta opción, se dará salida al número de filas afectadas por la macro o el procedimiento mediante el enlace de salida. Una macro devolverá el número de filas afectadas por las sentencias de la macro, como el número de filas insertadas, actualizadas, suprimidas y seleccionadas. Un procedimiento almacenado que sólo disponga de parámetros IN siempre devolverá un recuento de 0. Una función escalar definida por el usuario o un procedimiento almacenado que disponga de parámetros INOUT u OUT devolverá un recuento de 1 o devolverá un recuento de 0 si se ha producido un error. Una función de tabla definida por el usuario devolverá el número de filas devueltas.

Devolución de varias filas

Si se desmarca **El procedimiento devuelve varias filas**, sólo se devolverá la primera fila de las funciones de macro y de tabla que devuelven varias filas. Este proceso es similar a una búsqueda única. Si se selecciona esta opción, cada llamada de una función de macro o de tabla puede dar salida a varias filas mediante el enlace de salida. Este proceso es equivalente a una búsqueda de cursor. Si un macro devuelve varios conjuntos de resultados, el número de columnas de cada conjunto de resultados debe coincidir con el número de parámetros de salida del separador **Parámetros** y el tipo de datos de las columnas debe ser compatible con el tipo de datos de los parámetros de salida.

Si el **Tipo de procedimiento de base de datos** es **Procedimiento almacenado** o **Función escalar definida por el usuario**, la opción **El procedimiento devuelve varias filas** no estará activa, ya que este tipo de objetos no pueden devolver varias filas.

Si el objeto ejecutado es una macro y la opción **El procedimiento devuelve varias filas** está seleccionada, se dará salida a una o varias filas mediante el enlace de salida para cada sentencia de la macro. Cada sentencia **SELECT** de la macro puede dar salida a varias filas mediante el enlace de salida. Otras sentencias pueden dar salida a una fila mediante el enlace de salida. En cada fila del enlace de salida, las columnas de estado del procedimiento y de recuento de actividades contienen el estado y el recuento para dicha sentencia.

Si se desmarca **El procedimiento devuelve varias filas**, las columnas de estado contendrán el estado de la primera sentencia de la macro y la columna de recuento de actividades contendrá el total de los recuentos de actividades de cada sentencia de la macro.

Gestión de cursores

Código de retorno de EOD de procedimiento no está activo si **Proveedor de la base de datos** es **Teradata**. Cuando un procedimiento almacenado de Teradata devuelve datos, cierra todos los cursores que haya abierto, por lo que este recurso no es útil a la hora de captar varias filas de procedimientos almacenados de Teradata. Para captar varias filas, utilice una macro o una función definida por el usuario.

SQL Server

Las siguientes características son específicas de SQL Server.

Funciones escalares definidas por el usuario

Cuando **Tipo de procedimiento de base de datos** en el separador **Sintaxis** se ha establecido en **Función escalar definida por el usuario** (consulte "**El separador Sintaxis**"), el valor de retorno debe ser el primer parámetro en la cuadrícula de parámetros del separador **Parámetros** con **Tipo de parámetro** establecido en "**Función**" (consulte "**El separador Parámetros**").

Devolución de conjuntos de resultados y parámetros de salida

Cuando la etapa tiene un enlace de salida y el **Tipo de procedimiento** es **Origen** (consulte "**El separador Sintaxis**"),

- Si el procedimiento almacenado ejecutado devuelve un conjunto de resultados, debe seleccionarse **El procedimiento devuelve varias filas** (consulte "**El separador General**").
- Si el procedimiento ejecutado devuelve parámetros de salida, debe borrarse **El procedimiento devuelve varias filas**.

En el caso de los procedimientos de origen, no hay forma de devolver un conjunto de resultados y parámetros de salida, porque no existe un tipo de parámetro. Utilice el **Tipo de procedimiento** de **Transformación** para llevarlo a cabo.

Valor de retorno

Puesto que la etapa Stored Procedure maneja los valores de retorno utilizando **Estado de procedimiento para enlace** (consulte "**El separador General**"), cuando pulse **Cargar** (consulte "**Separador Columnas**"), deberá eliminar o suprimir manualmente el primer parámetro denominado "Valor de retorno".

Parámetros de salida y códigos de retorno

Los parámetros de salida y los códigos de retorno no aparecen en la secuencia de datos hasta que se han captado todos los resultados. Por este motivo, una fila adicional de la salida contiene los valores de código de resultado y parámetro de salida, mientras que las otras columnas contienen valores nulos. El resto de las filas contienen el código de retorno y el parámetro de salida establecido en cero (0) y nulo respectivamente. Esta fila adicional puede filtrarse utilizando una transformación para comprobar el código de retorno del procedimiento.

Sintaxis de las llamadas a procedimientos y códigos de retorno

La sintaxis de las llamadas a procedimientos siempre utiliza un código de retorno {?=call...} aunque no seleccione **Estado de procedimiento para enlace** (consulte "**El separador General**").

Cómo ponerse en contacto con IBM

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

La siguiente tabla enumera los recursos para soporte al cliente, servicios de software, formación, e información de productos y soluciones.

Tabla 9. Recursos de IBM

Recurso	Descripción y ubicación
Portal de soporte de IBM	Puede personalizar la información de soporte seleccionando los productos y los temas que sean de su interés 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 empresarial 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: 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/software/sw-training/
Representantes de IBM	Puede contactar con un representante de IBM para obtener información sobre soluciones en www.ibm.com/connect/ibm/us/en/

Facilitar comentarios

La tabla siguiente describe la forma en que se facilitan comentarios a IBM acerca de productos y documentación de productos.

Tabla 10. Facilitar comentarios a IBM

Tipo de comentarios	Acción
Comentarios sobre el producto	Puede proporcionar comentarios generales sobre productos mediante la encuesta de consumo en el sitio web www.ibm.com/software/data/info/consumability-survey

Tabla 10. Facilitar comentarios a IBM (continuación)

Tipo de comentarios	Acción
Comentarios sobre la documentación	<p>Para realizar comentarios acerca del Information Center, pulse el enlace Comentarios situado en la parte superior derecha de cualquiera de los temas del Information Center. También puede enviar sus comentarios sobre los manuales en archivos PDF, el Information Center o cualquier otra documentación de los siguientes modos:</p> <ul style="list-style-type: none"><li data-bbox="935 527 1414 611">• Formulario de comentarios en línea de los lectores: www.ibm.com/software/data/rcf/<li data-bbox="935 621 1414 646">• Correo electrónico: comments@us.ibm.com

Acceso a la documentación de productos

La documentación se proporciona en diversas ubicaciones y formatos, también en la ayuda que se abre directamente desde la interfaz del producto, en un Information Center para toda la suite y en manuales en archivos PDF.

El Information Center se instala como un servicio común con IBM InfoSphere Information Server. El Information Center contiene una ayuda para la mayoría de interfaces del producto, así como documentación completa para todos los módulos de productos de la suite. Puede abrir el Information Center desde el producto instalado o bien desde un navegador Web.

Acceso a Information Center

Puede utilizar los métodos siguientes para abrir el Information Center instalado.

- Pulse el enlace **Ayuda** de la parte superior derecha de la interfaz de cliente.

Nota: Desde IBM InfoSphere FastTrack e IBM InfoSphere Information Server Manager, el elemento **Ayuda** principal abre un sistema de ayuda local. Seleccione **Ayuda > Abrir Information Center** para abrir el Information Center de toda la suite.

- Pulse la tecla F1. La tecla F1 abre generalmente el tema que describe el contexto actual de la interfaz de cliente.

Nota: La tecla F1 no funciona en clientes Web.

- Utilice un navegador Web para acceder al Information Center instalado, aunque no haya iniciado sesión en el producto. Especifique la siguiente dirección en un navegador Web: `http://host_name:port_number/infocenter/topic/com.ibm.swg.im.iis.productization.iisinfsv.home.doc/ic-homepage.html`. El nombre_host es el nombre del sistema de capa de servicios en el que está instalado en Information Center, y número_puerto es el número de puerto para InfoSphere Information Server. El número de puerto predeterminado es 9080. Por ejemplo, en un sistema Microsoft® Windows® Server denominado iisdocs2, la dirección Web tendrá este formato: `http://iisdocs2:9080/infocenter/topic/com.ibm.swg.im.iis.productization.iisinfsv.nav.doc/dochome/iisinfsv_home.html`.

También hay disponible un subconjunto del Information Center, que se renueva periódicamente, en el sitio web de IBM `http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r7/index.jsp`.

Obtener la documentación en PDF y en copia impresa

- También puede disponer de un subconjunto de manuales en archivos PDF mediante el instalador de software de InfoSphere Information Server y el soporte de distribución. El resto de manuales en archivos PDF está disponible en línea y pueden accederse 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, ya sea 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`.

Facilitar comentarios sobre la documentación

Puede enviar sus comentarios sobre la documentación de la siguiente manera:

- Formulario de comentarios en línea de los lectores: www.ibm.com/software/data/rcf/
- Correo electrónico: comments@us.ibm.com

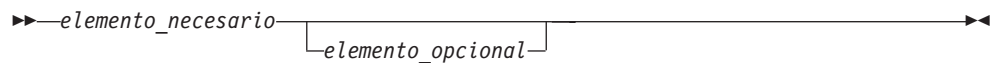
Cómo leer los diagramas de sintaxis

Las reglas siguientes se aplican a los diagramas de sintaxis que se utilizan en esta documentación:

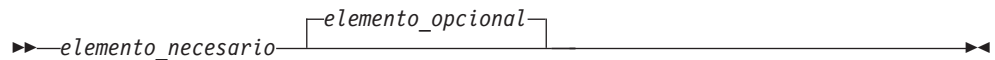
- Lea los diagramas de sintaxis de izquierda a derecha y de arriba abajo, siguiendo el recorrido de la línea. Se utilizan los convenios siguientes:
 - El símbolo >>--- indica el inicio de un diagrama de sintaxis.
 - El símbolo ---> indica que el diagrama de sintaxis continúa en la línea siguiente.
 - El símbolo >--- indica que el diagrama de sintaxis viene de la línea anterior.
 - El símbolo --->< indica el final de un diagrama de sintaxis.
- Los elementos necesarios aparecen en la línea horizontal (la línea principal).



- Los elementos opcionales aparecen debajo de la línea principal.

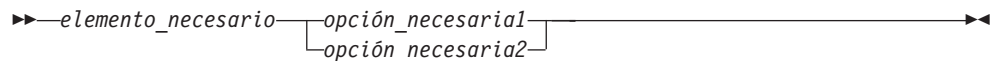


Si aparece un elemento opcional sobre la línea principal, dicho elemento no tendrá efecto sobre el elemento de sintaxis y sólo se utilizará para facilitar la lectura.

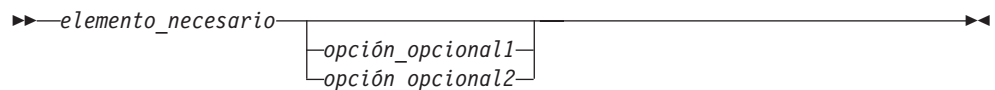


- Si se puede elegir entre dos o más elementos, éstos aparecerán apilados verticalmente.

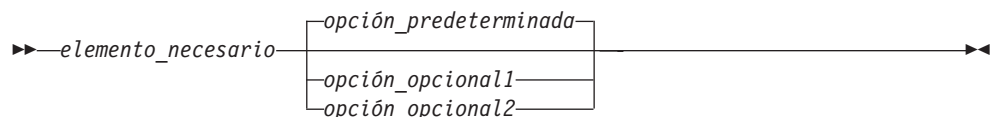
Si se debe elegir uno de los elementos, un elemento de la pila aparece en la línea principal.



Si la elección de uno de los elementos es opcional, toda la pila aparecerá por debajo de la línea principal.



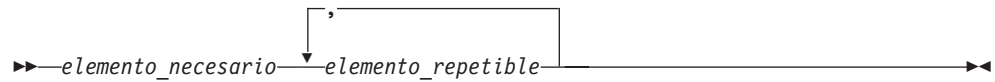
Si uno de los elementos es el predeterminado, aparecerá por encima de la línea principal y las opciones restantes se mostrarán por debajo.



- Una flecha que vuelve hacia la izquierda, sobre la línea principal, indica un elemento que se puede repetir.

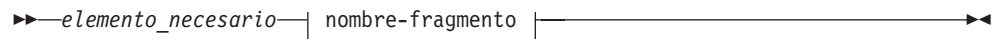


Si la flecha de repetición contiene una coma, los elementos repetidos se deben separar mediante una coma.

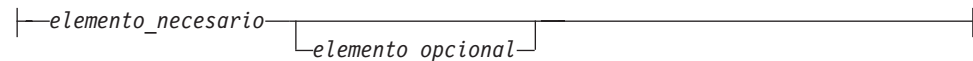


Una flecha de repetición sobre una pila indica que los elementos de la pila se pueden repetir.

- A veces, un diagrama se debe dividir en fragmentos. El fragmento de sintaxis se muestra por separado del diagrama de sintaxis principal, pero el contenido del fragmento se debe leer como si formara parte de la línea principal del diagrama.



Nombre-fragmento:



- Las palabras clave, y sus abreviaturas mínimas si las hay, aparecen en mayúsculas. Se deben escribir exactamente tal como se muestran.
- Las variables aparecen en letras minúsculas en cursiva (por ejemplo, **nombre-columna**). Representan nombres o valores proporcionados por el usuario.
- Separe las palabras clave y los parámetros con un espacio como mínimo si no se muestra ningún signo de puntuación en el diagrama.
- Entre los signos de puntuación, paréntesis, operadores aritméticos y otros símbolos exactamente como se muestran en el diagrama.
- Las notas a pie de página se muestran mediante un número entre paréntesis, por ejemplo (1).

Accesibilidad de los productos

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

Los módulos y las interfaces de usuario de los productos de IBM InfoSphere Information Server no son totalmente accesibles. El programa de instalación instala los siguientes módulos y componentes del producto:

- IBM InfoSphere Business Glossary
- IBM InfoSphere Business Glossary Anywhere
- IBM InfoSphere DataStage
- IBM InfoSphere FastTrack
- IBM InfoSphere Information Analyzer
- IBM InfoSphere Information Services Director
- IBM InfoSphere Metadata Workbench
- IBM InfoSphere QualityStage

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 accesible

Se proporciona documentación accesible sobre los productos de InfoSphere Information Server en un Information Center. El Information Center presenta la documentación en formato XHTML 1.0, que se puede ver en la mayoría de navegadores web. El formato XHTML permite establecer propiedades de visualización en el navegador. También permite utilizar lectores de pantalla y otras tecnologías de asistencia para acceder a la documentación.

IBM y la accesibilidad

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

Avisos y marcas registradas

La presente información se ha desarrollado para productos y servicios ofrecidos en Estados Unidos.

Avisos

Es posible que IBM no comercialice en otros países los productos, servicios o características que se describen en este manual. Consulte al representante local de IBM para obtener información sobre los productos y servicios que actualmente pueden adquirirse en su zona. Cualquier referencia a un producto, programa o servicio de IBM no pretende afirmar ni implicar que sólo se pueda utilizar dicho producto, programa o servicio de IBM. En su lugar se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no vulnere ninguno de los derechos de propiedad intelectual de IBM. Sin embargo, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patentes en tramitación que afecten al tema tratado en este documento. La posesión de este documento no otorga ninguna licencia sobre dichas patentes. Puede 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 formular consultas relacionadas con el juego de caracteres de doble byte (DBCS), póngase en contacto con el departamento de la propiedad intelectual de IBM de su país o envíe las consultas, por escrito, a la siguiente dirección:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japón

El párrafo siguiente no es aplicable al Reino Unido ni a ningún país en donde tales disposiciones sean incompatibles con la legislación local:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN TAL CUAL, SIN GARANTÍA DE NINGUNA CLASE, NI EXPLÍCITA NI IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN DE DERECHOS, COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN DETERMINADO. Algunos estados no permiten la declaración de limitación de responsabilidad de garantías expresas o implícitas en determinadas transacciones. Por consiguiente, es posible que esta declaración no se aplique en su caso.

Esta información podría contener imprecisiones técnicas o errores tipográficos. La información de este documento está sujeta a cambios periódicos; dichos cambios se incorporarán en nuevas ediciones de la publicación. Es posible que IBM realice

mejoras o efectúe cambios en el(los) producto(s) y/o el(los) programa(s) descritos en esta publicación en cualquier momento sin previo aviso.

Las referencias hechas en esta publicación a sitios Web que no son de IBM se proporcionan sólo para la comodidad del usuario y no constituyen un aval de esos sitios Web. La información contenida en estos sitios Web no forma parte de la información del presente producto IBM, y el usuario es responsable de la utilización de dichos sitios.

IBM puede utilizar o distribuir cualquier información que se le facilite de la manera que considere adecuada, sin contraer por ello ninguna obligación con el remitente.

Los licenciatarios de este programa que deseen obtener información sobre él con el fin de habilitar: (i) el intercambio de información entre programas creados de forma independiente y otros programas (incluido éste) y (ii) el uso mutuo de la información intercambiada, deben ponerse en contacto con:

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

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluido en algunos casos el pago de una tarifa.

El programa bajo licencia descrito en este documento y todo el material bajo licencia asociado a él los proporciona IBM según los términos del Acuerdo de Cliente de IBM, el Acuerdo Internacional de Programas Bajo Licencia de IBM o cualquier acuerdo equivalente entre el usuario e IBM.

Los datos de rendimiento contenidos en este documento se obtuvieron en un entorno controlado. Por consiguiente, es posible que los resultados obtenidos en otros entornos operativos varíen de forma significativa. 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 ajenos a IBM se ha obtenido a partir de los proveedores de dichos productos, los anuncios que han publicado u otras fuentes de dominio público. IBM no ha probado dichos productos ni puede confirmar la precisión del rendimiento, la compatibilidad ni ninguna otra afirmación sobre productos ajenos a IBM. Las preguntas sobre las prestaciones de productos que no son de IBM deben dirigirse a los proveedores de esos productos.

Todas las declaraciones de intenciones de IBM están sujetas a cambio o cancelación sin previo aviso, y sólo representan objetivos.

Esta información sólo tiene como objeto la planificación. La información de este documento está sujeta a cambios antes de que los productos descritos estén disponibles.

Este manual contiene ejemplos de datos e informes que se utilizan en operaciones comerciales diarias. Para ilustrarlos de la forma más completa posible, los ejemplos

incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier similitud con nombres y direcciones utilizados por una empresa real es totalmente fortuita.

LICENCIA DE COPYRIGHT:

Esta información contiene programas de aplicación de ejemplo en código fuente que ilustran técnicas de programación en diferentes plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo de cualquier forma, sin pagar a IBM, con la finalidad de desarrollar, utilizar 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. IBM, por lo tanto, no puede garantizar ni dar por sentada la fiabilidad, la capacidad de mantenimiento ni el funcionamiento de dichos programas. Los programas de ejemplo se suministran "TAL CUAL", sin garantía de ninguna clase. IBM no se hace responsable de los daños que se hayan podido causar debido al uso de los programas de ejemplo.

Todas las copias o partes de estos programas de ejemplo, o cualquier trabajo derivado, deberán incluir un aviso de copyright como el siguiente:

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

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

Marcas registradas

IBM, el logotipo de IBM e ibm.com son marcas registradas de International Business Machines Corp. en muchos países o regiones de alrededor del mundo. Otros nombres de productos y servicios pueden ser marcas registradas de IBM u otras empresas. Encontrará una lista actualizada de las marcas registradas de IBM en www.ibm.com/legal/copytrade.shtml.

Los términos siguientes son marcas registradas de otras compañías:

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

IT Infrastructure Library es una marca registrada de Agencia Central de Telecomunicaciones e informática and Telecommunications Agency que es ahora parte de la Oficina de Comercio Gubernamental.

Intel, el logotipo de Intel, Intel Inside, el logotipo de Intel Inside, Intel Centrino, el logotipo de Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium y Pentium son marcas comerciales o marcas registradas de Intel Corporation o de sus subsidiarias en los Estados Unidos y en otros países.

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

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

ITIL es un marca registrada y es una marca registrada comunitaria de la Oficina de Comercio Gubernamental, y está registrada en la Oficina de marcas registradas y patentes de los EE.UU.

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

Cell Broadband Engine es una marca registrada de Sony Computer Entertainment, Inc. en los Estados Unidos y/o en otros países y se utiliza bajo la licencia correspondiente.

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

El servicio postal de los 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 tiene titularidad no exclusiva de licencias de DPV y LACS^{Link} de United States Postal Service.

Otros nombres de empresas, productos o servicios pueden ser marcas registradas o marcas de servicio de otros.

Índice

A

accesibilidad de los productos
 accesibilidad 45
argumentos, véase parámetros 1
avisos legales 47

B

BTEQ 7

C

conjuntos de filas, devolución 1

D

datos de serie, correlación 2
DB2
 alias de base de datos 9, 16, 17
 gestión de cursores 30
 ISO de transacción 9, 16
 manejo de errores de SQL 29
 nombre de base de datos 9, 10, 16, 17
 nombres de base de datos 9, 16
 requisitos de configuración 1
 SELECT INTO, utilización 31
 sentencias de llamada definidas por el usuario 31
 tipos de parámetro 13
 variables de entorno 2
decisiones lógicas 6
documentación del producto
 acceder 41

E

editor de estilos de cuadrícula 1
enlaces 22
enlaces de entrada 16
enlaces de referencia 22
enlaces de referencia, emulación 22
enlaces de salida 19

F

filas
 grabación 1
 transformación 1
funcionalidad 5

I

introducción 1

L

lienzo en paralelo 2

M

marcas registradas
 lista 47

N

NLS 5

O

Oracle
 alias de base de datos 9, 16, 17
 Código de retorno de EOD de procedimiento 19
 gestión de cursores 19, 28
 ISO de transacción 9, 16
 manejo de errores de SQL 27
 paquetes 27
 requisitos de configuración 1
 SELECT INTO, utilización 29
 sentencias de llamada definidas por el usuario 29
 tipos de parámetro 13
 variables de entorno 2

P

página Entrada 9, 16
página Etapa
 separador General 9, 16
 separador Soporte multilingüístico 16
página Salida 9, 19
 separador General 19
parámetros 6, 7
 entrada 1
 salida 1
procedimientos de destino 6
procedimientos de origen 6
procedimientos de transformación 6, 8

R

reglas para valores de parámetros 8
requisitos de configuración 1, 5

S

sentencias Transact-SQL 7
separador General
 página Etapa 9, 16
 página Salida 19
separador Soporte multilingüístico 9, 16

separadores

 General
 página Etapa 9, 16
 página Salida 19
 NLS 9, 16
 Parámetros 7

servicios de software

 contactar 39

sitios web

 no IBM 43

soporte

 cliente 39

soporte al cliente

 contactar 39

SQL Server

 alias de base de datos 9, 16

 códigos de retorno 37

 conjuntos de resultados 37

 El procedimiento devuelve varias filas 19

 Estado de procedimiento para enlace 19

 Funciones escalares definidas por el usuario 36

 ISO de transacción 9, 16

 nombres de base de datos 9, 16

 parámetros de salida 37

 separador Sintaxis 12

 sintaxis de las llamadas a procedimientos 37

 tipos de parámetro 13

 valores de retorno 37

Sybase 9, 16

 información de estado 19

 parámetros, salida 19

 tamaño de transacción 31

 tipos de parámetro 13

 variables de entorno 2

T

Teradata

 cargar 19

 códigos de error 15, 35

 columna ProcCount 21

 comparación

 etapa Stored Procedure y etapa

 ODBC 32

 etapa Stored Procedure y Teradata

 API 32

 devolución de varias filas 36

 funciones de totales 32

 gestión de cursores 36

 importación

 macros 33

 procedimientos almacenados 33

 ISO de transacción 9, 16

 juego de caracteres del cliente,

 opciones 34

 origen de datos ODBC 12

Teradata (*continuación*)
 Recuento de actividades para
 enlace 19, 35
 separador Parámetros 13
 separador Sintaxis 12
 tipos de parámetro 13
 variables de entorno 2
Teradata SQL Assistant 7
tipos de procedimientos
 destino 6
 origen 6
 transformación 6, 8

U

uso de media secuencia 22

V

valores de parámetros, reglas 8
variables de entorno 2
visión general 1



Impreso en España

SC11-7987-00

