

## Práctica B: Creación de procedimientos almacenados con parámetros

### Objetivos

Después de realizar esta práctica, el alumno será capaz de:

- Crear un procedimiento almacenado con el Asistente para creación de procedimientos almacenados.
- Probar un procedimiento almacenado que incluya técnicas de control de errores.
- Crear mensajes personalizados de error.
- Crear procedimientos almacenados que devuelvan códigos.

### Requisitos previos

- El archivo lab071.zip contiene los archivos de comandos necesarios para la realización de la práctica, así como las correspondientes soluciones.
- Para la realización de las prácticas es necesario ejecutar el script Restore.cmd del directorio Restore.

## Ejercicio 1

### Utilización del Asistente para creación de procedimientos almacenados

En este ejercicio usará el Asistente para creación de procedimientos almacenados con el fin de crear un procedimiento almacenado en la base de datos **ClassNorthwind** que actualiza el número de teléfono de un empleado.

#### 🔗 Para usar el Asistente para creación de procedimientos almacenados

En este procedimiento usará el Asistente para creación de procedimientos almacenados con el fin de crear un procedimiento almacenado que actualiza el número de teléfono de un empleado.

1. Abra el Administrador corporativo de SQL Server.
2. En el árbol de la consola, haga clic en el servidor.
3. En el menú **Herramientas**, haga clic en **Asistentes**.
4. Expanda **Base de datos** y haga doble clic en Asistente para creación de procedimientos almacenados.
5. Seleccione la base de datos **ClassNorthwind**.
6. Cree un procedimiento almacenado que actualice el número de teléfono de un empleado. El número de teléfono está almacenado en la tabla **Employees**. Seleccione la acción **Actualizar** para la tabla **Employees**.
7. Haga clic en el botón **Modificar** para modificar las propiedades del procedimiento almacenado.
8. Denomine **UpdateEmployeePhone** al procedimiento almacenado.
9. Incluya sólo la columna **HomePhone** en la cláusula SET y sólo **EmployeeID** en la cláusula WHERE.
10. En el árbol de la consola, expanda la base de datos **ClassNorthwind** y, a continuación, haga clic en **Procedimientos almacenados**.
11. Compruebe que el procedimiento almacenado **UpdateEmployeePhone** aparece en el panel de detalles.
12. Repase las propiedades del procedimiento almacenado **UpdateEmployeePhone**.

¿Qué parámetros se definieron en el procedimiento almacenado?

**@EmployeeID\_1** y **@HomePhone\_2**

13. Abra el Analizador de consultas SQL y, si así se le pide, conéctese con el servidor local con Autenticación de Windows.
14. Ejecute el procedimiento almacenado **UpdateEmployeePhone** para comprobar que funciona como se esperaba. Actualice el número de teléfono a (503) 555-1212 para el empleado Nancy Davolio, cuyo EmployeeID es 1.

```
EXEC UpdateEmployeePhone  
@EmployeeID_1 = 1,  
@HomePhone_2 = '(503) 555-1212'
```

#### ✍ Para generar una secuencia de comandos

En este procedimiento generará y guardará una secuencia de comandos para el procedimiento almacenado que creó con el Asistente para creación de procedimientos almacenados.

1. Cambie al Administrador corporativo de SQL Server.
2. En el árbol de la consola, expanda **Bases de datos**, expanda **ClassNorthwind** y, a continuación, haga clic en **Procedimientos almacenados**.
3. En el panel de detalles, haga clic con el botón secundario del *mouse* en **UpdateEmployeePhone**, seleccione **Todas las tareas** y, a continuación, haga clic en **Generar secuencia de comandos SQL**.
4. Haga clic en **Aceptar** para generar la secuencia de comandos.
5. Guarde la secuencia de comandos como **UpdateEmployeePhone.sql**.
6. Abra la secuencia de comandos guardada y repásela.

## Ejercicio 2

### Utilización del control de errores en procedimientos almacenados

En este ejercicio ejecutará una secuencia de comandos que crea un procedimiento almacenado para agregar un proveedor y un producto a la base de datos **ClassNorthwind**. Después, probará el control de errores contenido en esta secuencia de comandos.

#### ✍ Para crear y probar un procedimiento almacenado

En este procedimiento abrirá y repasará una secuencia de comandos que crea un procedimiento almacenado para agregar un proveedor y un producto nuevos. A continuación, probará que el procedimiento almacenado se ejecuta como se esperaba al usarlo para insertar un proveedor y un producto nuevos. Finalmente, probará el control de errores del procedimiento almacenado.

1. Cambie al Analizador de consultas SQL.
2. Abra el archivo SupplierProduct.sql y revise su contenido.

¿Cuál es la ventaja de usar la función del sistema @@error mientras inserta valores en las tablas **Suppliers** y **Products**?

**Si la inserción falla debido a un tipo de datos o a la infracción de una restricción, se deshace la transacción.**

3. Ejecute la secuencia de comandos para crear el procedimiento almacenado **SupplierProductInsert**.
4. Abra el archivo SupplierProductInsert.sql. Modifique la secuencia de comandos y escriba los valores apropiados para agregar un proveedor y un producto nuevos. (Puede usar los valores que desee.)
5. Ejecute la secuencia de comandos modificada.
6. Pruebe el control de errores del procedimiento almacenado **SupplierProductInsert**; para ello, modifique los valores y coloque un comentario en la línea del parámetro @contactname. Ejecute la secuencia modificada para asegurar que el valor se pasará por alto.

¿Qué mensaje de error recibe?

**Debe proporcionar el nombre de la compañía y del contacto, dirección, ciudad región, código postal, país, teléfono, nombre de producto y discontinuidad.**

**(Título del contacto, fax, página principal, precio de la unidad, unidades de almacén: 'Units on Order and Reorder Level can be null'.)**

## Ejercicio 3

### Personalización de mensajes de error

En este ejercicio creará un mensaje de error personalizado, que se guardará en el registro de aplicación del Visor de sucesos de Windows 2000, que indica el identificador del proveedor que se insertó junto con el usuario de SQL Server que realizó la inserción.

#### ✍ Para crear un mensaje de error personalizado

En este procedimiento modificará el procedimiento almacenado **SupplierProduct** para llamar a los mensajes de error personalizados. El archivo Soluciones\CustomErrorAnswer.sql contiene la secuencia de comandos completa de este procedimiento.

1. Abra la secuencia de comandos CustomError.sql, revise su contenido y ejecútela.
2. Busque el comentario `/* #1 Escriba aquí el código del alumno */` y agregue una variable al procedimiento almacenado **CustomError** que almacene el valor del nombre de usuario que inserta el proveedor.

```
/* #1 Escriba aquí el código del alumno. */
```

```
DECLARE @UserName nvarchar (60)
```

```
SELECT @UserName = suser_sname()
```

---

Sugerencia Use la función del sistema SUSER\_SNAME.

---

3. Busque el siguiente comentario `/* #2 Escriba aquí el código del alumno */`. Agregue una instrucción RAISERROR que indique que se ha agregado un nuevo proveedor.

La instrucción RAISERROR debe llamar al error #50018 y pasar los parámetros del número de proveedor y del usuario que está ejecutando el procedimiento almacenado.

Consulte los Libros en pantalla de SQL Server para obtener información adicional acerca de la instrucción RAISERROR.

```
RAISERROR (50018, 16, 1, @InsertSupplierID, @UserName)
```

4. Busque el siguiente comentario `/* #3 Escriba aquí el código del alumno */` para crear el mensaje de error número 50018 mediante el procedimiento almacenado del sistema **sp\_addmessage**. Incluya los valores Supplier y UserName en el mensaje de error.

```
EXEC sp_addmessage 50018, 16, 'Supplier %d was inserted by %s', 'us_english','true'
```

5. Ejecute la secuencia de comandos para crear el procedimiento almacenado **CustomError**.
6. Abra el archivo SupplierProductInsert.sql. Modifique la secuencia de comandos y escriba los valores apropiados para agregar un proveedor y un producto nuevos. (Puede usar los valores que desee.)
7. Ejecute la secuencia de comandos modificada.
8. Revise los resultados y, a continuación, abra el Visor de sucesos y examine el registro de aplicación para comprobar que se registró el mensaje de información.

## Ejercicio 4

### Utilización de códigos de retorno

En este ejercicio creará un procedimiento con la palabra clave OUTPUT mediante la secuencia de comandos Return1.sql. A continuación, ejecutará ese procedimiento almacenado y lo probará con distintos códigos de retorno mediante las secuencias de comandos Return2.sql y Return3.sql.

#### ⚡ Para crear el procedimiento almacenado OrderCount

En este procedimiento creará un procedimiento almacenado denominado **OrderCount** que cuenta el número de pedidos pendientes de un cliente. Si el cliente tiene al menos un pedido pendiente, devuelve el valor de estado 1. Si el cliente no tiene pedidos pendientes, devuelve el valor de estado 0. Éste es un ejemplo de procedimiento almacenado anidado.

1. Mediante el Analizador de consultas SQL, abra la secuencia de comandos Return1.sql, revise su contenido y ejecútela.
2. Escriba y ejecute el siguiente procedimiento:

```
EXEC OrderCount 1,1
```

¿Cuál es el resultado?

**Este comando se completa con éxito pero no devuelve datos.**

### ✍ Para ejecutar el procedimiento almacenado **OrderCount** con la opción **OUTPUT**

En este procedimiento observará los efectos de utilizar la opción **OUTPUT** en el procedimiento almacenado **OrderCount**.

1. Mediante el Analizador de consultas **SQL**, abra la secuencia de comandos **Return2.sql**, revise su contenido y ejecútela.

Esta secuencia de comandos ejecuta el procedimiento almacenado **OrderCount** y pasa el valor de un identificador **CustomerID** que tiene pedidos pendientes.

¿Cuál es el resultado?

**El cliente RATTC tiene 18 pedidos pendientes.**

2. Abra la secuencia de comandos **Return3.sql**, revise su contenido y ejecútela.

Esta secuencia de comandos ejecuta el procedimiento almacenado **OrderCount** y pasa el valor de un identificador **CustomerID** que tiene pedidos pendientes.

¿Cuál es el resultado?

**El cliente WOLZA NO tiene pedidos pendientes.**



## Si dispone de tiempo

### Ejecución de procedimientos almacenados extendidos

En este ejercicio ejecutará un procedimiento almacenado extendido y verá el nombre de archivo de la biblioteca de vínculos dinámicos (DLL) donde se define la función.

#### ✍ Para ejecutar un procedimiento almacenado extendido

En este procedimiento ejecutará el procedimiento almacenado extendido **xp\_cmdshell** para enumerar todos los archivos y carpetas del directorio raíz de la unidad C:.

1. Mediante el Analizador de consultas SQL, compruebe que está usando la base de datos **master**.
2. Ejecute el procedimiento almacenado extendido **xp\_cmdshell** para ver la lista de todos los archivos de la carpeta C:\.

```
EXEC master..xp_cmdshell 'dir c:\'
```

¿Cuál fue el resultado?

**El contenido del directorio se devolvió como filas de texto.**

3. Ejecute el procedimiento almacenado del sistema **sp\_helptext** para ver la definición de **xp\_cmdshell**.

```
EXEC master..sp_helptext xp_cmdshell
```

¿Cuál fue el resultado?

**Xplog70.dll. Ésta es la biblioteca DLL que contiene la función del procedimiento almacenado extendido.**

## Si dispone de tiempo

### Trazas de procedimientos almacenados mediante el Analizador de SQL

En este procedimiento utilizará la herramienta gráfica Analizador de SQL para trazar procedimientos almacenados individuales.

#### ✍ Para realizar una traza de los eventos del procedimiento almacenado con el Analizador de SQL

En este procedimiento iniciará una traza del Analizador de SQL mediante la utilización de una plantilla de traza personalizada para supervisar procedimientos almacenados.

1. Abra el Analizador de SQL.
2. En la barra de herramientas, haga clic en **Nueva traza**.
3. Conecte con el servidor (local) con Autenticación de Windows.
4. En la ficha **Sucesos**, agregue todos los procedimientos almacenados y clases de sucesos de Transact-SQL.
5. Haga clic en **Ejecutar**.
6. Cambie al Analizador de consultas SQL, abra la secuencia de comandos `SupplierProductInsert.sql`, revise su contenido y ejecútela.
7. Cambie al Analizador de SQL.
8. Deténgase y revise la traza.