

## Práctica A: Creación de desencadenadores

### Objetivos

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

- Crear desencadenadores para mantener la integridad de los datos.
- Crear desencadenadores para exigir reglas complejas de empresa.

### Requisitos previos

- El archivo lab073.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

### Creación de desencadenadores

En este ejercicio va a ejecutar una secuencia de comandos que crea un desencadenador. Una vez creado, lo probará para asegurarse de que funciona.

#### ✍ Para crear un desencadenador

En este procedimiento va a ejecutar un archivo de comandos para crear un desencadenador.

1. Abra el Analizador de consultas SQL y, si así se le pide, conéctese con el servidor local con autenticación de Microsoft Windows®.
2. En la lista **BD**, haga clic en **ClassNorthwind**.
3. Abra el archivo OrdDetInsert.sql y repáselo.

Esta secuencia de comandos crea un desencadenador en la tabla **Order Details**. Este desencadenador actualiza la columna **UnitsInStock** de la tabla **Products** cuando se inserta una fila en la tabla **Order Details** (cuando se recibe un pedido).

4. Ejecute el archivo OrdDetInsert.sql.
5. Ejecute el procedimiento almacenado del sistema **sp\_helptrigger** en la tabla **Order Details** de la base de datos **ClassNorthwind** para comprobar si se ha creado el desencadenador.

### ✍ Para probar el desencadenador

En este procedimiento va a probar el desencadenador recién creado. Para ello, insertará una fila en la tabla **Order Details**.

1. Ejecute la instrucción SELECT siguiente para seleccionar una fila de la tabla **Products** y determinar las unidades de un producto que hay en almacén:

```
SELECT * FROM Products WHERE ProductID = 22
```

La columna **UnitsInStock** debe contener el valor de 104. Si la columna **UnitsInStock** muestra un número diferente, anótelo.

2. Inserte una fila en la tabla **Order Details** que pida 50 unidades del producto 22. La instrucción INSERT será similar a la siguiente:

```
INSERT [Order Details]  
(OrderID, ProductID, UnitPrice, Quantity, Discount)  
VALUES (11077, 22, 21.00, 50, 0.0)  
GO
```

3. Haga una consulta en la tabla **Products** para comprobar que el valor de la columna **UnitsInStock** para el **ProductID** determinado ha cambiado a 54.

## Ejercicio 2

### Creación de un desencadenador para actualizar datos derivados

En este ejercicio creará dos tablas nuevas (sin restricciones PRIMARY y FOREIGN KEY) y, a continuación, un desencadenador en la tabla **NewCategories** para exigir integridad en la columna **NewProducts.Discontinued**. El archivo Soluciones\CategoryDelete.sql contiene la secuencia de comandos completa de este procedimiento.

#### ✍ Para crear un desencadenador que actualice datos derivados

En este procedimiento va a crear dos tablas nuevas denominadas **NewCategories** y **NewProducts**. Ninguna de las dos tiene las restricciones PRIMARY y FOREIGN KEY de las tablas **Categories** y **Products**. Creará un desencadenador en la tabla **NewCategories**. Este desencadenador actualiza la columna **Discontinued** de la tabla **NewProducts** cuando se elimina una categoría (cuando se elimina un registro de la tabla **NewCategories**). Todos los productos afectados se marcan con 1, lo que indica que ya no se suministran. Utilice el archivo CategoryDelete.sql y haga los cambios apropiados.

1. Escriba y ejecute la consulta siguiente para crear dos tablas nuevas denominadas **NewCategories** y **NewProducts**.

```
USE ClassNorthwind
GO
--Crea una tabla NewCategories
SELECT * INTO NewCategories FROM Categories
--Crea una tabla NewProducts
SELECT * INTO NewProducts FROM Products
GO
```

2. Escriba y ejecute la consulta siguiente para crear un desencadenador en la tabla **NewCategories**. Este desencadenador actualiza la columna **Discontinued** de la tabla **NewProducts** a 1 cuando se elimina la categoría principal de un producto (cuando se elimina una fila de la tabla **NewCategories**).

```
CREATE TRIGGER Category_Delete
ON NewCategories
FOR DELETE
AS
UPDATE P SET Discontinued = 1
FROM NewProducts AS P INNER JOIN Deleted AS d
ON P.CategoryID = D.CategoryID
```

3. Escriba y ejecute la siguiente instrucción SELECT que consulta la tabla **NewProducts** para determinar si se siguen suministrando los productos con **CategoryID 7**.

```
SELECT ProductID, CategoryID, Discontinued  
FROM NewProducts WHERE CategoryID = 7
```

4. Escriba una instrucción DELETE que quite una fila de la tabla **NewCategories** y compruebe que el desencadenador se ejecuta correctamente.

¿Son necesarios estos desencadenadores para mantener la integridad de los datos en la base de datos **ClassNorthwind**? Razone la respuesta.

**Sí. La integridad referencial en cascada podría quitar los productos relacionados de la tabla NewProducts, pero un desencadenador es la mejor manera de implementar una acción más compleja, como conservar los registros en NewProducts mientras se actualiza la columna Discontinued.**

## Ejercicio 3

### Creación de un desencadenador que mantiene una regla compleja de empresa

En este ejercicio va a crear un desencadenador INSERT en la tabla **NewProducts**. Este desencadenador determina si hay un historial de pedidos en la tabla **Order Details** antes de permitir una eliminación en la tabla **NewProducts**.

#### ✍ Para crear un desencadenador en la tabla NewProducts

En este procedimiento utilizará la tabla **NewProducts** creada en el ejercicio anterior. Creará un desencadenador que determina si hay un historial de pedidos para un producto que se va a eliminar. Si nunca se ha pedido el producto, éste se puede eliminar. Si el producto tiene un historial de pedidos, se deshace la eliminación de la tabla de productos y el desencadenador devuelve un mensaje de error personalizado. El archivo Soluciones\BusinessRule.sql contiene la secuencia de comandos completa de este ejercicio.

1. Cree un desencadenador DELETE en la tabla **NewProducts** que determine si hay un historial de pedidos para un producto que se va a eliminar. Si hay registros en la tabla **Order Details** para ese producto, muestre un mensaje y deshaga el desencadenador.

2. Elimine el producto 6 de la tabla **NewProducts** para probar el desencadenador.

El archivo Soluciones\BusinessRule.sql es la secuencia de comandos completa de este paso.

¿Se ha activado el desencadenador? Razone la respuesta.

**Sí. El producto eliminado de la tabla NewProducts tenía pedidos en la tabla Order Details. El desencadenador evitó la eliminación y devolvió un mensaje de error.**

## Ejercicio 4

### Comprobación del orden de activación de las restricciones y los desencadenadores

En este ejercicio modificará la instrucción del ejercicio anterior para probar el orden de activación de las restricciones y los desencadenadores. El archivo Soluciones\BusinessRule2.sql contiene la secuencia de comandos completa de este ejercicio.

#### ✍ Para modificar el desencadenador del ejercicio anterior

1. Cree un desencadenador en la tabla **Products** similar al que utilizó en el ejercicio anterior y que se denomine **Product\_Delete2**.

Recuerde que en el ejercicio anterior se creó un desencadenador DELETE denominado **Product\_Delete** en la tabla **NewProducts**.

```
CREATE TRIGGER Product_Delete2
ON Products FOR DELETE
AS
IF (Select Count (*)
    FROM [Order Details] INNER JOIN deleted
    ON [Order Details].ProductID = Deleted.ProductID
    ) > 0
BEGIN
    RAISERROR('Transaction cannot be processed. This Product
still has a history of orders.', 16, 1)
    ROLLBACK TRANSACTION
END
```

2. Pruebe el desencadenador.

¿Se ha activado el desencadenador? Razone la respuesta.

**No. La restricción PRIMARY KEY impidió que se activara el desencadenador.**