

Creación de scripts de inserción en SQL Server a partir de una consulta de selección

Luis Miguel Blanco Ancos

En determinadas ocasiones, necesitamos insertar filas en una tabla cuyos datos tomamos de una o varias tablas ya existentes en la misma u otra base de datos.

Podemos solucionar el problema utilizando una sentencia INSERT INTO combinada con una SELECT de la manera mostrada en el siguiente código fuente.

```
INSERT INTO NombreTablaDestino  
(CampoDestino1,CampoDestino2)  
SELECT CampoOrigen1,CampoOrigen2  
FROM NombreTablaOrigen  
WHERE Condición
```

Esta solución funciona bien si la ejecución de esta sentencia la llevamos a cabo desde una base de datos en la que tenemos tanto la tabla de origen de los datos como la de destino, sobre la que vamos a añadir las filas.

Sin embargo si la tabla de origen y la de destino pertenecen a dos bases de datos separadas físicamente, debemos llevar a cabo el traspaso de datos en dos fases independientes.

Entre las posibles soluciones con las que podemos abordar este problema se encontraría el hacer una copia de seguridad de la base de datos origen, o separarla del servidor para copiar los archivos de datos al servidor de destino. Estas técnicas, en el caso de requerir la copia de un conjunto pequeño de datos, pueden suponer un medio muy costoso en cuanto al consumo de recursos utilizado.

El truco planteado a continuación no es la panacea para resolver el problema, pero se trata de una solución sencilla, adecuada para casos en los que necesitemos copiar una cantidad no muy elevada de filas, o bien para crear scripts de carga de datos en nuevas bases de datos a partir de información ya existente.

Supongamos que tenemos una base de datos con información de clientes, y queremos tomar de la base de datos Northwind varios campos de las filas de la tabla Customers, cuyo valor en el campo Country es Germany, y llevar estos datos a la tabla MisClientes de nuestra base de datos.

Creación de scripts de inserción en SQL Server a partir de una consulta de selección

En primer lugar abriremos el Analizador de consultas de SQL Server, y nos conectaremos a la base de datos Northwind; seleccionaremos la opción de menú *Consulta + Resultados a archivo*, y en el panel de consulta escribiremos la siguiente sentencia. Observe el lector, que cuando necesitamos obtener datos de tipo cadena para después insertarlos en un campo destino, debemos hacer un uso especial de múltiples comillas simples para componer correctamente la sentencia.

```
SELECT 'INSERT INTO MisClientes VALUES ('' +  
CustomerID + ''','' + CompanyName + ''','' + ContactName + ''','' + City  
+ ''') AS SentencialInsercion  
FROM Customers  
WHERE Country = 'Germany'
```

Tras ejecutar esta sentencia, obtendremos un archivo con tantas instrucciones de inserción como filas de la tabla Customers cumplieran la condición. Un fragmento de dicho archivo sería el siguiente.

```
INSERT INTO MisClientes VALUES ('ALFKI','Alfreds Futterkiste','Maria Anders','Berlin')  
INSERT INTO MisClientes VALUES ('BLAUS','Blauer See Delikatessen','Hanna  
Moos','Mannheim')  
INSERT INTO MisClientes VALUES ('DRACD','Drachenblut Delikatessen','Sven  
Ottlieb','Aachen')  
INSERT INTO MisClientes VALUES ('FRANK','Frankenversand','Peter Franken','München')
```

Para insertar estos datos en nuestra base de datos destino, nos llevaremos este archivo y lo abriremos en el Analizador de consultas del servidor SQL Server destino, seleccionando todas estas sentencias de inserción y ejecutándolas.

Vamos a dar otra vuelta de tuerca a la creación de scripts usando esta técnica. Si lo que necesitamos es obtener datos de campos de diferentes tipos de datos, será preciso, para los tipos distintos de carácter, usar una función de conversión como CAST, para componer correctamente el conjunto de instrucciones resultantes.

En el siguiente código fuente de ejemplo, creamos un script que extrae datos de varios campos de las tablas Customers y Orders, para crear un conjunto de sentencias de inserción para una hipotética tabla de destino llamada DatosFacturas.

```
SELECT 'INSERT INTO DatosFacturas VALUES ('' +  
Customers.CustomerID + ''','' +  
Customers.CompanyName + ''','' +  
CAST(Orders.OrderID AS varchar(10)) + ',' +  
CAST(Orders.OrderDate AS varchar(25)) + ''','' +  
CAST(Orders.Freight AS varchar(10))  
+ ')' AS SentencialInsercion  
FROM Customers, Orders  
WHERE Customers.CustomerID = 'BLAUS'
```

```
AND Customers.CustomerID = Orders.CustomerID
```

Veamos seguidamente algunas de las sentencias de inserción resultantes de ejecutar el anterior código.

```
INSERT INTO DatosFacturas VALUES ('BLAUS','Blauer See Delikatessen',10501,'Abr 9 1997  
12:00AM',8.85)  
INSERT INTO DatosFacturas VALUES ('BLAUS','Blauer See Delikatessen',10509,'Abr 17 1997  
12:00AM',0.15)  
INSERT INTO DatosFacturas VALUES ('BLAUS','Blauer See Delikatessen',10582,'Jun 27 1997  
12:00AM',27.71)
```