



**ASIGNATURA:** SISTEMAS GESTORES DE BASES DE DATOS **PROFESOR:** JUAN RAMÓN ALEMANY

## EJERCICIO

Una empresa utiliza la base de datos de AdventureWorks2008R2 en su gestión diaria. Un directivo de la empresa necesita poder realizar estudios sobre algunas de las tablas de dicha base de datos, concretamente las tablas de PRODUCT y PRODUCTSUBCATEGORY, que se encuentran en el servidor de PRODUCCIÓN.

Por ello, se ha decidido que la mejor solución es la de crear 2 scripts de SQL, de tal modo, que al ejecutarlos en el servidor de minería de datos (es un servidor diferente al de PRODUCCIÓN), cree las 2 tablas con todos los registros que contienen, nuestro objetivo es por lo tanto realizar una exportación de datos.

El primer script, tendrá 2 instrucciones de tipo CREATE TABLE para crear las tablas de PRODUCT y PRODUCTSUBCATEGORY.

En segundo script, tendrá una línea INSERT para cada uno de los registros que hay que insertar en las tablas PRODUCT y PRODUCTSUBCATEGORY.

Los comandos de CREATE TABLE del primer script, se escriben de forma NO PROGRAMADA, pero las "x" líneas del segundo script que hacen los INSERTS en las 2 tablas, evidentemente no se tienen que escribir, sino que hay que realizar uno o varios cursores que generen (escriban) las líneas con el comando INSERT completo.

Al ejecutar el segundo script, hay que decirle que imprima el resultado en un fichero, en lugar de imprimirlo por la pantalla, de este modo, ya nos quedan todas las instrucciones guardadas en un fichero.

Las tablas de PRODUCT y PRODUCTSUBCATEGORY tienen muchos campos, realizar solamente para aquellos campos que son de tipos de datos más comunes, como NVARCHAR, INTEGER, BOOLEAN, MONEY...

## OBJETIVO DEL EJERCICIO

El objetivo del ejercicio es ver cómo podemos realizar un sistema propio de exportación de datos y que también, nos puede servir como generador de copias de seguridad.



**ASIGNATURA:** SISTEMAS GESTORES DE BASES DE DATOS **PROFESOR:** JUAN RAMÓN ALEMANY

## PARTE DEL CÓDIGO DE LA SOLUCIÓN

### CREAR TABLAS

```
CREATE TABLE [Production].[ProductCopy](
    [ProductID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [Name] [dbo].[Name] NOT NULL,
    [ProductNumber] [nvarchar](25) NOT NULL,
    [MakeFlag] [dbo].[Flag] NOT NULL,
    [FinishedGoodsFlag] [dbo].[Flag] NOT NULL,
    [Color] [nvarchar](15) NULL,
    [StandardCost] [money] NOT NULL,
    [ListPrice] [money] NOT NULL,
    [Size] [nvarchar](5) NULL,
    [ProductSubcategoryID] [int]
REFERENCES
    [Production].[ProductSubcategoryCopy]([ProductSubcategoryID])
ON DELETE NO ACTION ON UPDATE NO ACTION NULL)
```

```
CREATE TABLE [Production].[ProductSubcategoryCopy](
    [ProductSubcategoryID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [ProductCategoryID] [int] NOT NULL,
    [Name] [dbo].[Name] NOT NULL,
    [ModifiedDate] [datetime] NOT NULL)
```

### CÓDIGO SQL PARA GENERAR INSTRUCCIONES INSERT PARA LA TABLA DE PRODUCTCOPY

```
DECLARE @ProductID INT
DECLARE @Name NVARCHAR(100)
DECLARE @ProductNumber NVARCHAR(25)
DECLARE @MakeFlag BIT
DECLARE @FinishedGoodsFlag BIT
DECLARE @Color NVARCHAR(15)
DECLARE @StandardCost MONEY
DECLARE @ListPrice MONEY
DECLARE @Size NVARCHAR(5)
DECLARE @ProductSubcategoryID INT
DECLARE Cursor_ProductCopy
CURSOR FOR
SELECT
    ProductID,
    Name,
    ProductNumber,
    MakeFlag,
    FinishedGoodsFlag,
    Color,
    StandardCost,
    ListPrice,
    Size,
    ProductSubcategoryID
FROM Production.Product
OPEN Cursor_ProductCopy
FETCH NEXT FROM Cursor_ProductCopy
INTO
    @ProductID,
    @Name,
```



**ASIGNATURA:** SISTEMAS GESTORES DE BASES DE DATOS **PROFESOR:** JUAN RAMÓN ALEMANY

```
@ProductNumber,  
@MakeFlag,  
@FinishedGoodsFlag,  
@Color,  
@StandardCost,  
@ListPrice,  
@Size,  
@ProductSubcategoryID  
WHILE (@@FETCH_STATUS=0)  
BEGIN  
PRINT 'INSERT INTO [Production].[ProductCopy]  
(ProductID,  
Name,  
ProductNumber,  
MakeFlag,  
FinishedGoodsFlag,  
Color,  
StandardCost,  
ListPrice,  
Size,  
ProductSubcategoryID)  
VALUES ('  
+  
CAST(@ProductID AS NVARCHAR(100)) + ', ' +  
@Name + ', ' +  
@ProductNumber + ', ' +  
CAST(@MakeFlag AS NVARCHAR(100)) + ', ' +  
CAST(@FinishedGoodsFlag AS NVARCHAR(100)) + ', ' +  
CASE WHEN @Color IS NULL THEN '*' ELSE @Color END + ', ' +  
CAST(@StandardCost AS NVARCHAR(100)) + ', ' +  
CAST(@ListPrice AS NVARCHAR(100)) + ', ' +  
CASE WHEN @Size IS NULL THEN '*' ELSE @Size END + ', ' +  
CASE WHEN @ProductSubcategoryID IS NULL THEN 'NULL' ELSE  
CAST(@ProductSubcategoryID AS NVARCHAR(100))END  
+ ')'  
FETCH NEXT FROM Cursor_ProductCopy  
INTO  
@ProductID,  
@Name,  
@ProductNumber,  
@MakeFlag,  
@FinishedGoodsFlag,  
@Color,  
@StandardCost,  
@ListPrice,  
@Size,  
@ProductSubcategoryID  
END  
CLOSE Cursor_ProductCopy  
DEALLOCATE Cursor_ProductCopy
```



**ASIGNATURA:** SISTEMAS GESTORES DE BASES DE DATOS **PROFESOR:** JUAN RAMÓN ALEMANY

DOCUMENTACIÓN APORTADA POR ALUMNOS PARA LA REALIZACIÓN DEL EJERCICIO

Para poder realizar los PRINTs en un fichero en lugar de la pantalla, publicamos los enlaces y el código necesario, por cortesía del alumno XAVIER SEGUÍ BELTRÁN.

URLS CON INFORMACIÓN

<http://devwithdave.com/SQLServer/stored-procedures/SQLServer-StoredProcedures--HOW-TO-WRITE-TO-A-TEXT-FILE.asp>

<https://www.youtube.com/watch?v=KWnnEhXCzPk>

EJECUTAR UNA VEZ LAS SIGUIENTES INSTRUCCIONES PARA CONFIGURAR EL SERVIDOR DE TAL MODO QUE PERMITA: "OLE AUTOMATION".

```
sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
sp_configure 'Ole Automation Procedures', 1;
GO
RECONFIGURE;
GO
```

CREACIÓN DEL PROCEDIMIENTO ALMACENADO

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[WriteToFile]
@File VARCHAR(2000),
@Text VARCHAR(2000)
AS
BEGIN
DECLARE @OLE INT
DECLARE @FileID INT
EXECUTE sp_OACreate 'Scripting.FileSystemObject', @OLE OUT
EXECUTE sp_OAMethod @OLE, 'OpenTextFile', @FileID OUT,
@File, 8, 1
EXECUTE sp_OAMethod @FileID, 'WriteLine', Null, @Text
EXECUTE sp_OADestroy @FileID
EXECUTE sp_OADestroy @OLE
END
```

**ASIGNATURA:** SISTEMAS GESTORES DE BASES DE DATOS **PROFESOR:** JUAN RAMÓN ALEMANY

#### EJECUCIÓN DEL PROCEDIMIENTO ALMACENADO

```
DECLARE @Path VARCHAR(2000)
DECLARE @Txt VARCHAR(2000)
SET @Path = 'C:\Users\Mertman\Documents\ACME\Log
Files\Log.txt'
SET @Txt = 'Did it work the next time?'
EXEC WriteToFile @Path, @Txt
```