

MS SQL Server

- [Backup de MS SQL Server](#)
- [Buscar texto en una BBDD de MS SQLSERVER](#)

Backup de MS SQL Server

Backups de MS SQLSERVER

Vamos a ver un script sencillo que nos va a permitir realizar backups de SQL Server de Microsoft aunque tengamos la versión Express

El script es bastante sencillo, y debemos de arrancar el Windows Powershell ISE con permisos de Administrador

Una vez arrancado, creamos un archivo nuevo

Escribimos la siguiente sentencia:

```
Backup-SqlDatabase -ServerInstance "NOMBRE_DEL_SERVIDOR\INSTANCIA_MS_SQL" -Database " BASE_DE_DATOS" -BackupFile "C:\CARPETA_DE_COPIAS\Backup_$(Get-Date -Format 'ddMMyy_HHMMss').bak"
```

Debes de sustituir los textos (NOMBRE_DEL_SERVIDOR, INSTANCIA_MS_SQL, BASE_DE_DATOS, C:\CARPETA_DE_COPIAS) por los datos correspondientes a la instalación. Si tenemos que crear copias de varias bases de datos, generaremos una línea por cada Base de Datos a respaldar

Borrar copias anteriores

Ahora si no tenemos cuidado, llenaremos el disco de copias, para ello es conveniente borrar las copias anteriores a una fecha.

En el mismo script (O en otro nuevo), escribimos lo siguiente:

```
Get-ChildItem -Path "c:\CARPETA_DE_COPIAS" -Recurse | Where-Object {($_.LastWriteTime -lt (Get-Date).AddDays(-31))}
```

Esta línea, nos permite borrar los archivos anteriores a los últimos 31 días. (Get-Date).AddDays(-31))

Copia completa con anclajes mensuales

A continuación vamos a realizar un ejemplo completo con copias diarias y guardaremos una copia mensual durante un año.

Crearemos dos scripts, uno para las copias mensuales y otro para las copias diarias

Script para las copias diarias

```
Backup-SqlDatabase -ServerInstance "NOMBRE_DEL_SERVIDOR\INSTANCIA_MS_SQL" -Database "BASE_DE_DATOS" -File "c:\CARPETA_DE_COPIAS\DIARIAS\$(get-date -f yyyyMMdd).BAK"
```

Como puedes ver, hemos quitado los guiones de la fecha, puedes usar la versión con o sin guiones ((get-date -f yyyyMMdd).BAK o (get-date -f yyyy-MM-dd).BAK)

Ahora el script de las copias mensuales

```
Backup-SqlDatabase -ServerInstance "NOMBRE_DEL_SERVIDOR\INSTANCIA_MS_SQL" -Database "BASE_DE_DATOS" -File "c:\CARPETA_DE_COPIAS\MENSUALES\$(get-date -f yyyy-MM-dd).BAK"
```

Ahora ejecutaremos el script que gestiona el borrado de las copias antiguas

```
Get-ChildItem -Path "c:\CARPETA_DE_COPIAS\DIARIAS" -Recurse | Where-Object {($_.LastWriteTime -lt (get-date -f yyyy-MM-dd).AddDays(-30))} | Remove-Item  
Get-ChildItem -Path "c:\CARPETA_DE_COPIAS\MENSUALES" -Recurse | Where-Object {($_.LastWriteTime -lt (get-date -f yyyy-MM-dd).AddMonths(-1))} | Remove-Item
```

Script completo

El script se puede ver a continuación

Diario

```
Backup-SqlDatabase -ServerInstance "NOMBRE_DEL_SERVIDOR\INSTANCIA_MS_SQL" -Database "BASE_DE_DATOS" -File "c:\CARPETA_DE_COPIAS\DIARIAS\$(get-date -f yyyyMMdd).BAK"  
Get-ChildItem -Path "c:\CARPETA_DE_COPIAS\DIARIAS" -Recurse | Where-Object {($_.LastWriteTime -lt (get-date -f yyyy-MM-dd).AddDays(-30))} | Remove-Item  
Get-ChildItem -Path "c:\CARPETA_DE_COPIAS\MENSUALES" -Recurse | Where-Object {($_.LastWriteTime -lt (get-date -f yyyy-MM-dd).AddMonths(-1))} | Remove-Item
```

Planificador de tareas

Ahora, deberemos de crear una tarea, que ejecute la copia diaria y la limpieza, todos los días, y una tarea que se ejecutará mensualmente, que realiza la copia mensual de las Bases de Datos

Pendiente:

Veremos la posibilidad de ponerlo todo en un script que ejecute la copia diaria, y si es el último día de mes, o el primer día de mes (Dependiendo de la hora), ejecute la copia mensual.

Buscar texto en una BBDD de MS SQLSERVER

Para buscar un texto en una BBDD de MS SQLSERVER, crearemos un procedimiento almacenado que contenga

```
CREATE PROC BuscaValorEnBBDD
(
@StrValorBusqueda nvarchar(100)
)
AS
BEGIN

CREATE TABLE #Resultado (NombreColumna nvarchar(370), ValorColumna nvarchar(3630))
SET NOCOUNT ON

DECLARE @NombreTabla nvarchar(256),
@NombreColumna nvarchar(128),
@StrValorBusqueda2 nvarchar(110)

SET @NombreTabla = ''
SET @StrValorBusqueda2 = QUOTENAME('%' + @StrValorBusqueda + '%','''')

WHILE @NombreTabla IS NOT NULL
    BEGIN
        SET @NombreColumna = ''
        SET @NombreTabla =
        ( SELECT MIN( QUOTENAME( TABLE_SCHEMA) + '.' + QUOTENAME( TABLE_NAME) )
        FROM INFORMATION_SCHEMA. TABLES
        WHERE TABLE_TYPE = 'BASE TABLE'
        AND QUOTENAME( TABLE_SCHEMA) + '.' + QUOTENAME( TABLE_NAME) > @NombreTabla
        AND OBJECTPROPERTY(
        OBJECT_ID( QUOTENAME( TABLE_SCHEMA) + '.' + QUOTENAME( TABLE_NAME)), 'IsMSShipped' ) = 0)

        WHILE ( @NombreTabla IS NOT NULL) AND ( @NombreColumna IS NOT NULL)
```

```

BEGIN

SET @NombreColumna =

( SELECT MIN( QUOTENAME( COLUMN_NAME) )

FROM INFORMATION_SCHEMA.COLUMNS

WHERE TABLE_SCHEMA = PARSENAME( @NombreTabla, 2)

AND TABLE_NAME = PARSENAME( @NombreTabla, 1)

AND DATA_TYPE IN ( ' char' , ' varchar' , ' nchar' , ' nvarchar' )

AND QUOTENAME( COLUMN_NAME) > @NombreColumna)


IF @NombreColumna IS NOT NULL

    BEGIN

        INSERT INTO #Resultado

        EXEC

            ( 'SELECT ''' + @NombreTabla + '.' + @NombreColumna + ''', LEFT(' +

@NombreColumna + ', 3630)

            FROM ' + @NombreTabla + ' ( NOLOCK) ' + ' WHERE ' + @NombreColumna + ' LIKE ' +

@StrValorBusqueda2)

        END

    END

END

SELECT NombreColumna, ValorColumna FROM #Resultado

END

```

Para buscar, sólo tendremos que ejecutar en el servidor MSSQL

```
EXEC dbo.BuscaValorEnBBDD ' texto_a-buscar'
```