

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**DISEÑO E IMPLEMENTACIÓN
DE BASE DE DATOS****Tema****TRABAJO CON EL ENTORNO SQL**

TRABAJO CON EL ENTORNO SQL

1. ¿QUÉ ES UNA BASE DE DATOS?

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite, una base de datos es un sistema de archivos electrónico. Las bases de datos tradicionales se organizan por campos, registros y archivos, un campo es una pieza única de información; un registro es un sistema completo de campos y un archivo es una colección de registros. Por ejemplo, una guía de teléfono es análoga a un archivo, contiene una lista de registros cada uno de los cuales consiste en tres campos: nombre, dirección y número de teléfono.

2. COMO CREAR UNA BASE DE DATOS

2.1. ARCHIVOS DE LA BASE DE DATOS

Existen tres tipos de archivos para agruparlos que son:

- Principal o Main Data File con extensión (.mdf) es el archivo de datos principal donde se encuentra la información inicial de la base de datos, en este archivo se almacenan los datos y los objetos del usuario, aunque también se puede realizar en archivos secundarios, toda base de datos cuenta con un archivo principal.
- Secundarios o Secondary Data File con extensión (.ndf) son los archivos de datos secundarios donde la base de datos puede contener uno o varios, se pueden usar opcional, están definidos por el usuario y almacenan los datos de los usuarios, se pueden utilizar para repartir la información en varios discos agregando cada archivo en una unidad de disco diferente, además si una base de datos supera el límite de almacenamiento para un archivo de Windows se podrán utilizar archivos de datos secundarios para el crecimiento de la misma.
- De registro o Log Data File con extensión (.log) son archivos del registro de transacciones, contiene la información de se utiliza para la

recuperación de la base de datos cada BD debe contar con al menos un archivo de registro.

2.2. NOMBRES DE ARCHIVO LÓGICO Y FÍSICO

Existen dos tipos de nombres de archivo que son:

logical_file_name es el nombre que le da para hacer referencia al archivo físico en todas las instrucciones Transact-SQL, el nombre del archivo lógico debe cumplir con las reglas de identificador de SQL Server, su nombre debe ser único.

Os_file_name: es el nombre que se le da para hacer referencia al archivo físico que incluye la ruta para ingresar al directorio, el nombre del archivo debe seguir las reglas para nombres de archivos del sistema operativo. Para crear una base de datos es necesario realizar el script que se muestra en la siguiente figura:

```
CREATE DATABASE TickedDB
on
(
    NAME='TickedDB',
    FILENAME='D:\DB\TickedDB.mdf',
    SIZE=10,
    MAXSIZE=50,
    FILEGROWTH=5
)
LOG ON
(
    NAME='TickedDB_log',
    FILENAME='D:\Log\TickedDB_log.ldf',
    SIZE=5MB,
    MAXSIZE=25MB,
    FILEGROWTH=5MB
)
```

Figura 1: Script para crear una base de datos

Se puede observar en el siguiente script la creación de una base de datos con un tipo de archivo lógico y físico.

NAME: hace referencia al nombre lógico de la base de datos

FILENAME: ruta donde se encuentra físicamente el archivo

SIZE: tamaño que soporta la base de datos

MAXSIZE: tamaño máximo de la base de datos

FILEGROWTH: crecimiento automático de los archivos de la base de datos

NOTA: recuerden se ingresan dos veces debido a que la primera hace referencia al archivo primario y la otra es del archivo de registro.

Para modificar una BD es necesario realizar el Script que se muestra en la siguiente figura:

```
ALTER DATABASE TickedDB
    ADD FILEGROUP GroupFile_1;
go
ALTER DATABASE TickedDB
    ADD FILE
    (
        NAME=Data_1,
        FILENAME='D:\DB\Data_1.ndf',
        SIZE=5MB,
        MAXSIZE=100MB,
        FILEGROWTH=5MB
    ),
    (
        NAME=Data_2,
        FILENAME='D:\DB\Data_2.ndf',
        SIZE=5MB,
        MAXSIZE=100MB,
        FILEGROWTH=5MB
    )
    TO FILEGROUP GroupFile_1;
```

Figura 2: Script para modificar una Base de Datos

3. RESPALDO Y RESTAURACIÓN DE UNA BASE DE DATOS

En SQL Server existen tres tipos de respaldos y restauración:

- **Respaldo Completo:** es una copia de toda la base de datos en un determinado momento, la restauración de este respaldo solo recupera la información del momento en que se realizó el respaldo, generalmente se usan para base de datos pocos críticas y pequeñas. El script que se utiliza para realizar un respaldo completo de BD es el que se muestra en la siguiente figura:

```
BACKUP DATABASE TickedDB
TO DISK = 'C:\RespaldosDB\TickedDB.bak'
WITH FORMAT;
```

Figura 3: Script para realizar un respaldo completo de BD

También se puede realizar por asistente colocándose en el explorador de objetos en la carpeta de base de datos (Databases) -> colocarte en la base de datos que requieres copiar -> botón derecho-> tareas (tasks) -> copia de seguridad (Back Up) como se muestra en la siguiente figura

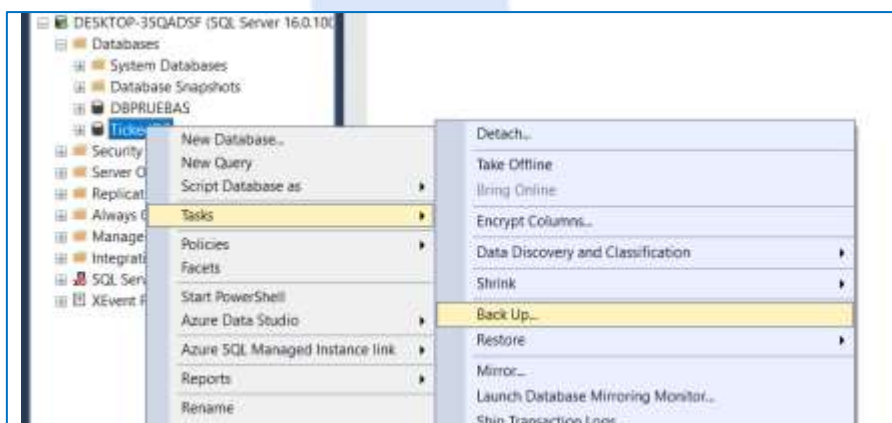


Figura 4: Back Up por asistente

Una vez seleccionando en copia de seguridad (Back Up) aparecerá la siguiente pantalla donde se deberá especificar que es una copia de seguridad completa (full) en backup type, después haciendo clic en el botón “Add”, debajo de “Destination” y especificando el nombre del archivo con la extensión (.bak) y el

destino donde se va a guardar el archivo como se muestra en la siguiente figura:

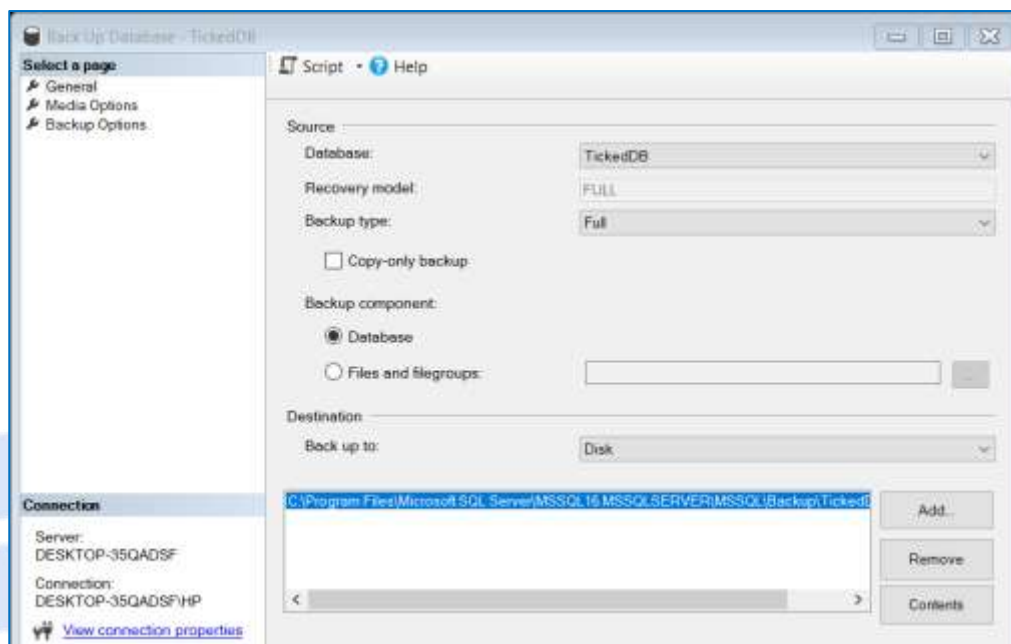


Figura 5: Destino donde se guardará el Back up

- Respaldo Diferencial:** solo realizan una copia de los datos nuevos y modificados a partir del último respaldo completo, ideal para grandes bases de datos o con poca densidad de transacciones, requiere menos espacio que un respaldo completo, para restaurar una base de datos con un mecanismo diferencial se debe de contar el respaldo completo. La información restaurada será la que existía en el momento que se realizó el respaldo diferencial. El script que se utiliza para realizar un respaldo diferencial de BD es como se muestra en la siguiente figura:



Figura 6: Representación de un respaldo diferencial.

```
BACKUP DATABASE TickedDB  
TO DISK = N'C:\RespalDOSDB\TickEDDB.bak  
WITH DIFFERENTIAL,  
NOFORMAT, NOINIT,  
NAME = N'TickedDB Backup',  
SKIP, NOREWIND,  
NOUNLOAD, STATS = 10
```

Figura 7: Script para realizar un respaldo diferencial.

BACKUP DATABASE: nombre de la copia de seguridad de la base de datos.

TO DISK: ubicación donde se guardará el archivo de la copia del BD.

WITH DIFFERENTIAL: agregamos que es de tipo diferencial.

NOFORMAT: el encabezado del medio no debe escribirse para esta operación.

NOINIT: No sobrescribe los archivos de la copia de seguridad.

NAME: nombre del archivo que se va a generar como .bak.

SKIP: verifica la fecha y hora de vencimiento de los archivos de la copia de seguridad antes de sobrescribirlos.

NOREWIND: indica que SQL Server mantendrá la cinta abierta después de la operación de copia de seguridad

NOUNLOAD: son configuraciones de sesión que persisten durante la vida de la sesión o hasta que se reinicia especificando la alternativa.

STATS: informa el porcentaje completado a partir del umbral para informar el siguiente intervalo. También se puede realizar por asistente al igual que el completo, lo diferente es seleccionar en tipo de copia de seguridad hay que seleccionar tipo diferencial, como se muestra en la figura a continuación.

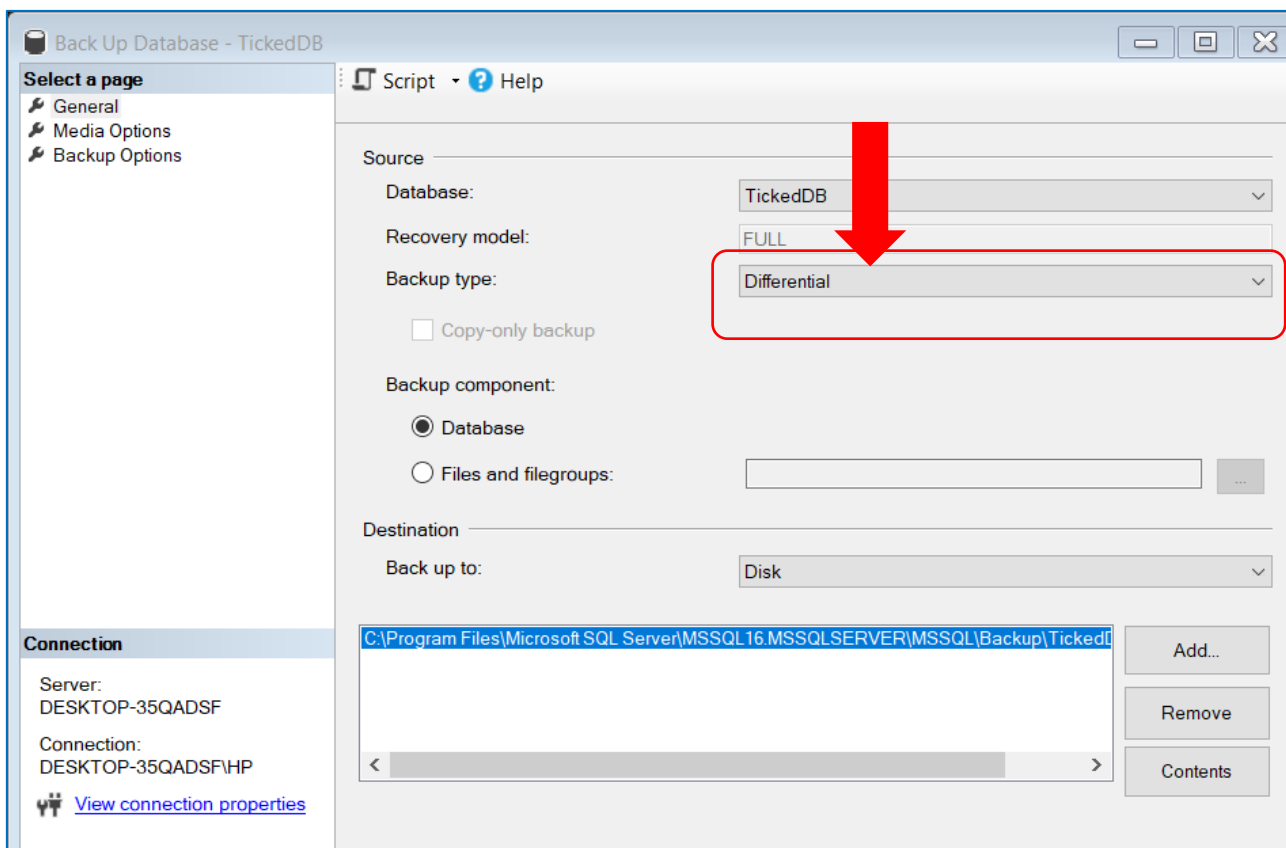


Figura 8: Representación de los tipos de respaldo.

- **Respaldo Diferencial + Log de transacciones**
 - Este mecanismo es ideal para base de datos con alta densidad de transacciones y no se permiten pérdidas de datos.
 - Si el log de transacciones no está dañado se puede recuperar hasta el último momento de la base de datos.
 - Se requieren de más ficheros y el espacio es un poco mayor con respecto al mecanismo diferencial.



Figura 9: Representación de los tipos de respaldo.

Este mecanismo consta de: 1. Un Respaldo completo. 2. RespalDOS diferenciales. 3. RespalDOS de log de transacciones (opcional). 4. Log de transacciones. Para restaurar usando este mecanismo:

1. Restaurar el respaldo completo con la opción NORECOVERY
2. Restaurar el último respaldo diferencial con la opción NORECOVERY

3. Restaurar cada uno de los respaldos de log de transacciones en el mismo orden en que fueron realizados utilizando la opción NORECOVERY
4. Restaurar la base de datos usando la opción RECOVERY



Figura 10: Representación de restauración de Back Up.

Los scripts que se utilizan para cada respaldo son:

<p>1. Respaldo Completo:</p> <pre>RESTORE DATABASE TickedDB FROM DISK = N'C:\DB\TickedDB.bak' WITH FILE = 1, NORECOVERY</pre>	<p>2. Respaldo Diferencial:</p> <pre>RESTORE DATABASE TickedDB FROM DISK = N'C:\DB\TickedDB.bak' WITH FILE = 4, NORECOVERY</pre>
<p>3. Respaldo de Log de transacciones:</p> <pre>RESTORE LOG TickedDB FROM DISK = N'C:\DB\TickedDB.log.bak' WITH FILE = 1, NORECOVERY</pre>	<p>4. Log de Transacciones:</p> <pre>RESTORE DATABASE TickedDB WITH RECOVERY</pre>

Figura 11: Scripts de los tipos de restauración de Back Up

Bibliografía:

- Manual SQLServer-Transact SQL Básico/Avanzado – Claudia Isabel Degollado



INSTITUTO
KHIPU