

Semana 12: Respaldo y Recuperación

🔧 0. Preparación del Entorno (Prerrequisitos)

Antes de iniciar con la teoría y práctica, necesitamos una base de datos de prueba configurada correctamente.

Requisito previo: Crea una carpeta en tu disco C: llamada SQLBackups (ruta: C:\SQLBackups) para que los scripts funcionen sin errores de permisos.

Script de Configuración Inicial

```
SQL
USE master;
GO

-- 1. Crear la base de datos
IF EXISTS (SELECT name FROM sys.databases WHERE name =
'DB_Semana12')
    DROP DATABASE DB_Semana12;
GO

CREATE DATABASE DB_Semana12;
GO

USE DB_Semana12;
GO

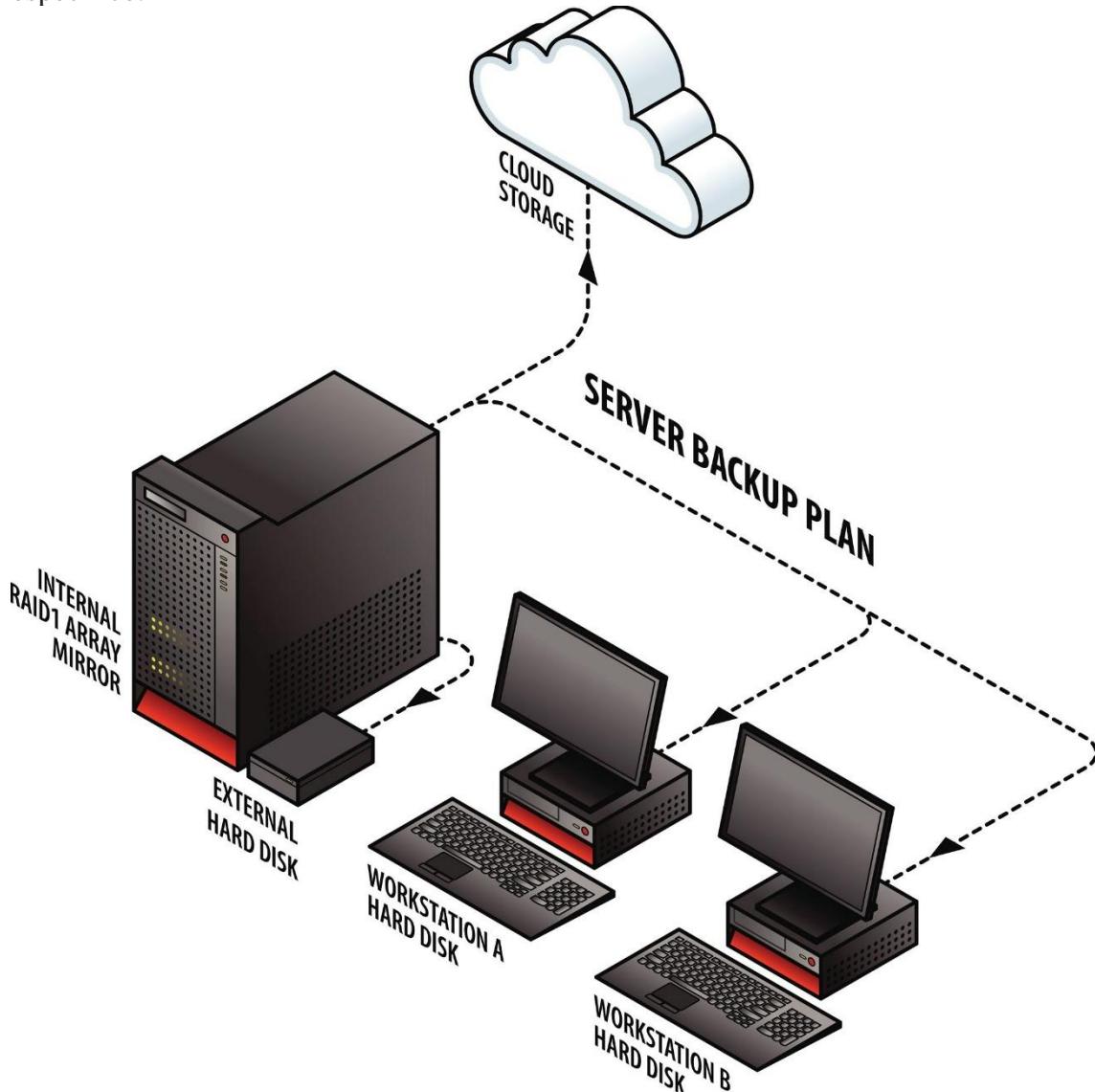
-- 2. Configurar Modelo de Recuperación COMPLETO (FULL)
-- Indispensable para realizar backups de logs y recuperación a un
punto en el tiempo.
ALTER DATABASE DB_Semana12 SET RECOVERY FULL;
GO

-- 3. Crear una tabla de ejemplo y poblar datos
CREATE TABLE Ventas (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    Producto VARCHAR(50),
    Monto DECIMAL(10,2),
    FechaRegistro DATETIME DEFAULT GETDATE()
);
INSERT INTO Ventas (Producto, Monto) VALUES
('Laptop', 1500.00), ('Mouse', 25.50), ('Teclado', 45.00);
GO
```

■ Tema 1: Estrategias de Backup y Comandos

Definiciones

- **Backup Completo (Full):** Copia la totalidad de la base de datos (datos, estructura, objetos). Es la base de cualquier estrategia.
- **Backup Diferencial:** Copia solo los datos que han cambiado desde el último backup *Completo*. Es acumulativo y más rápido que hacer otro Full.
- **Backup de Log de Transacciones (Transactional):** Copia el registro de transacciones (LDF). Permite vaciar el log activo y recuperar la base de datos a un segundo específico.



Getty Images

Script Teórico y Práctico

SQL

```
-- --- 1. BACKUP COMPLETO ---
-- Es el punto de partida. Sin esto, no puedes hacer diferenciales ni de logs.
BACKUP DATABASE DB_Semana12
TO DISK = 'C:\SQLBackups\DB_Semana12_Full.bak'
WITH FORMAT, -- Formatea el archivo para sobrescribir si existe
MEDIANAME = 'SQLServerBackups',
NAME = 'Full Backup de DB_Semana12';
GO

-- Simulemos actividad (Paso del tiempo)
INSERT INTO Ventas (Producto, Monto) VALUES ('Monitor', 200.00);
GO

-- --- 2. BACKUP DIFERENCIAL ---
-- Respalda solo el cambio (El registro del Monitor)
BACKUP DATABASE DB_Semana12
TO DISK = 'C:\SQLBackups\DB_Semana12_Diff.bak'
WITH DIFFERENTIAL,
NAME = 'Diff Backup de DB_Semana12';
GO

-- Simulemos más actividad crítica
INSERT INTO Ventas (Producto, Monto) VALUES ('Impresora', 300.00);
WAITFOR DELAY '00:00:02'; -- Esperamos 2 segundos
INSERT INTO Ventas (Producto, Monto) VALUES ('Error Humano', 0.00);
-- Dato erróneo
GO

-- --- 3. BACKUP DE LOG ---
-- Respalda las transacciones recientes
BACKUP LOG DB_Semana12
TO DISK = 'C:\SQLBackups\DB_Semana12_Log.trn'
WITH NAME = 'Log Backup de DB_Semana12';
GO
```

Verificación vía Interfaz(SSMS)

1. Clic derecho sobre la base de datos DB_Semana12.
 2. Ve a **Tasks (Tareas) > Back Up...**
 3. En "Backup type", despliega la lista. Verás las opciones: *Full, Differential, Transaction Log*.
 4. Si seleccionas una y das clic en "Contents" o "Media options", verás que es el equivalente gráfico a las opciones WITH FORMAT O NAME.
-

Tema 2 y 3: Verificación y Restauración (Mantenimiento)

Definiciones

- **Verificación (RESTORE VERIFYONLY):** Comprueba que el archivo de respaldo esté completo y legible, pero no restaura los datos.
- **Restauración (RESTORE DATABASE):** Proceso de volcar los datos del archivo al servidor.

Script Teórico y Práctico

SQL

```
-- --- VERIFICACIÓN DE INTEGRIDAD ---
RESTORE VERIFYONLY
FROM DISK = 'C:\SQLBackups\DB_Semana12_Full.bak';
-- Si el mensaje es "The backup set on file... is valid", el
archivo está sano.
```

Tema 4, 5 y 6: Restauración Avanzada y Punto en el Tiempo

Definiciones

- **Cadena de Restauración:** El orden estricto debe ser: Full \$\rightarrow\$ Último Diferencial \$\rightarrow\$ Todos los Logs secuenciales posteriores.
- **NORECOVERY:** Deja la base de datos en estado "Restoring" (esperando más archivos).
- **RECOVERY:** Finaliza la restauración y hace la base de datos accesible.
- **Point in Time (STOPAT):** Instrucción para detener la restauración de un Log en una fecha/hora exacta.

Práctica: Restauración Point-in-Time (Script)

Imagina que el dato "Error Humano" (insertado arriba) queremos evitarlo.
Restauraremos todo *antes* de ese error.

SQL

```
USE master;
GO
```

```
-- IMPORTANTE: Para restaurar, nadie debe estar usando la DB.
ALTER DATABASE DB_Semana12 SET SINGLE_USER WITH ROLLBACK IMMEDIATE;

-- 1. Restaurar el FULL con NORECOVERY (Esperando más datos)
RESTORE DATABASE DB_Semana12
FROM DISK = 'C:\SQLBackups\DB_Semana12_Full.bak'
WITH NORECOVERY, REPLACE;

-- 2. Restaurar el DIFERENCIAL con NORECOVERY
```

```

RESTORE DATABASE DB_Semana12
FROM DISK = 'C:\SQLBackups\DB_Semana12_Diff.bak'
WITH NORECOVERY;

-- 3. Restaurar el LOG con STOPAT (Punto en el tiempo) y RECOVERY
-- (Finalizar)
-- Nota: Ajusta la fecha/hora a un minuto antes de que ejecutaras
el insert de 'Error Humano'
-- Ejemplo: Si son las 10:30, pon 10:29.
DECLARE @TargetTime DATETIME = DATEADD(MINUTE, -1, GETDATE());

RESTORE LOG DB_Semana12
FROM DISK = 'C:\SQLBackups\DB_Semana12_Log.trn'
WITH STOPAT = @TargetTime,
RECOVERY;

ALTER DATABASE DB_Semana12 SET MULTI_USER;
GO

-- Verificación: No debería aparecer 'Error Humano'
SELECT * FROM DB_Semana12.dbo.Ventas;

```

Verificación vía Interfaz (SSMS)

1. Clic derecho sobre Databases > **Restore Database**.
 2. Selecciona Device y busca tu archivo .bak.
 3. En la pestaña izq. **Timeline**: Aquí puedes arrastrar una barra temporal para elegir el "Specific date and time". SQL Server seleccionará automáticamente qué Full, Diff y Logs necesita.
-

⚠ Práctica Final: Simulación de Desastre (Pérdida de .MDF)

Objetivo: Simular que el disco donde están los datos se corrompió o el archivo .mdf fue borrado, pero tenemos los backups y el log de transacciones (.ldf) intacto o salvable.

Pasos:

1. **Generar datos críticos no respaldados:**

SQL

```

USE DB_Semana12;
INSERT INTO Ventas (Producto, Monto) VALUES ('Datos Críticos Post-
Backup', 9999.99);
GO

```

2. **Simular el desastre (Borrar el archivo):**

- Ejecuta: ALTER DATABASE DB_Semana12 SET OFFLINE;
- Ve a la ruta de datos de SQL (Usualmente C:\Program Files\Microsoft SQL Server\MSSQL...\\MSSQL\\DATA) y **BORRA** o mueve el archivo DB_Semana12.mdf.

- Intenta ponerla online: `ALTER DATABASE DB_Semana12 SET ONLINE;`
 - *Resultado:* Error grave. La base de datos aparecerá como "Recovery Pending".
3. **La Solución (Tail Log Backup y Recuperación):**

Este es el truco de experto. Aunque el .mdf murió, el .ldf sigue vivo. Haremos un "Tail Log Backup" (Backup de la cola del log) para salvar esos 'Datos Críticos Post-Backup'.

SQL

```
USE master;
GO

-- 1. Intentar salvar la cola del log (Tail Log Backup)
-- Usamos NO_TRUNCATE porque la DB está caída.
BACKUP LOG DB_Semana12
TO DISK = 'C:\SQLBackups\DB_Semana12_Tail.trn'
WITH NO_TRUNCATE, INIT, NAME = 'Tail Log Backup';

-- 2. Comenzamos la restauración desde cero (FULL)
RESTORE DATABASE DB_Semana12
FROM DISK = 'C:\SQLBackups\DB_Semana12_Full.bak'
WITH NORECOVERY, REPLACE;

-- 3. Restauramos el Diferencial (si existiera y fuera válido en la cadena)
RESTORE DATABASE DB_Semana12
FROM DISK = 'C:\SQLBackups\DB_Semana12_Diff.bak'
WITH NORECOVERY;

-- 4. Restauramos el Log normal
RESTORE LOG DB_Semana12
FROM DISK = 'C:\SQLBackups\DB_Semana12_Log.trn'
WITH NORECOVERY;

-- 5. Restauramos el TAIL LOG que acabamos de salvar (Aquí está la magia)
RESTORE LOG DB_Semana12
FROM DISK = 'C:\SQLBackups\DB_Semana12_Tail.trn'
WITH RECOVERY; -- Finalizamos aquí

-- Verificar
SELECT * FROM DB_Semana12.dbo.Ventas;
```

Si realizaste esto correctamente, habrás recuperado la base de datos completa, incluso los datos que se insertaron después del último backup programado, gracias al Tail Log.
