

Cursos SQL Server 2008 R2

Triggers

Cursos SQL Server 2008 R2

Índice

Introducción

Existen dos planteamientos en la estrategia de las copias de seguridad teniendo en cuenta la tipología de las bases de datos.

Distinguimos entre bases de datos de sistema y bases de datos de producción.

Bases de datos de sistema

- Se deben realizar copias no programadas. Solo al ser modificadas.
- No se hacen copias de la *tempdb*.

Bases de datos de producción

- Las copias normalmente son programadas

Existen dos formas de llevar a cabo las copias de seguridad:

- Copias de seguridad en frío
- Copias de seguridad en caliente

Las copias en frío se realizan copiando los ficheros de datos (.mdf, .ndf's) y si lo deseas del log. Para ello, deberíamos detener el servicio del motor SQL Server, ya que en caso contrario, los archivos están bloqueados. Una restauración supondría adjuntar la base de datos.

Lo habitual es hacer las copias en caliente, ya que permite llevar a cabo estrategias de recuperación más avanzadas (concepto de disponibilidad).

La copia de seguridad mantiene una copia duplicada de los datos que se puede recuperar en el momento en el que la pérdida es crítica.

Tipos de copias de Seguridad:

- Copia de seguridad completa
- Copia de seguridad diferencial
- Copia de seguridad del log de transacciones
- Copia de seguridad del grupo de archivos
- Copia de seguridad de archivos

Dependiendo del modelo de recuperación que tengamos en la base de datos, nuestras opciones a la hora de hacer una copia de seguridad serán diferentes.

El modelo de recuperación completa contará con las mismas opciones de la simple, y la opción de copia de seguridad del log.

Full Backup

El propósito de una copia de seguridad completa es la captura de todos los datos almacenados y poder utilizarla para volver a crear la base de datos.

Todo modelo de recuperación parte de una copia completa válida.

Para evitar una inconsistencia lógica realiza los siguientes pasos:

1. Cierra la base de datos, bloquea las actualizaciones en las páginas de la RAM.
2. Sitúa una marca en el log
3. Realiza una sincronización en el disco duro de todas las páginas de la RAM que están con el bit de cambio activado (CheckPoint).
4. Durante el tiempo que se emplea en el proceso anterior, las transacciones activas se siguen anotando en el log.
5. Una vez finalizado el checkpoint se procede a hacer una copia de todas las páginas del disco duro (diccionario también) y del “trozo del log” que ha recogido las actualizaciones durante el proceso de copia.
6. Se trunca el Log.

Realiza una copia de seguridad completa de la base de datos o de uno o varios archivos o grupos de archivos (BACKUP DATABASE).

Además, con el modelo de recuperación completa o el modelo de recuperación de cargas masivas de registros, realiza la copia de seguridad del registro de transacciones (BACKUP LOG).

El comando para realizar la copia de la base de datos es el siguiente:

```
BACKUP DATABASE <database name>
```

```
TO DISK='<directory>\<filename>' WITH INIT
```

Mediante la clausula TO se especifica el dispositivo, o el archivo al que enviaremos la copia de seguridad.

Mediante la clausula WITH se pueden especificar diferentes características.

Mediante INIT especificaremos que se reescriba sobre el dispositivo / archivo.

DIFERENTIAL BACKUP

Una copia de seguridad diferencial captura los cambios realizados desde la última FULL BACKUP.

En caso de no existir una FULL BACKUP no se podrá realizar una DIFERENTIAL BACKUP.

Cada DIFERENTIAL BACKUP acumula los cambios de la diferencial anterior.

Supongamos que se realiza una copia diferencial cada 4 horas, la que sucede a las 4 de la mañana contiene todo lo que contiene la anterior de las 12.

Para determinar qué extensiones deben ser copiadas, SQL Server mantiene un mapa de extensiones. En el momento que se realiza un FULL BACKUP este mapa se pone a cero.

El comando más simple para realizar una copia diferencial:

```
BACKUP DATABASE <database name>
```

```
TO DISK='<directory>\<filename>'
```

```
WITH DIFERENCIAL
```

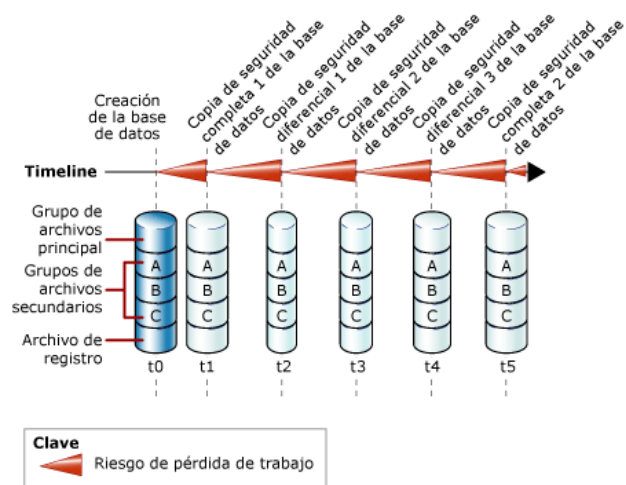
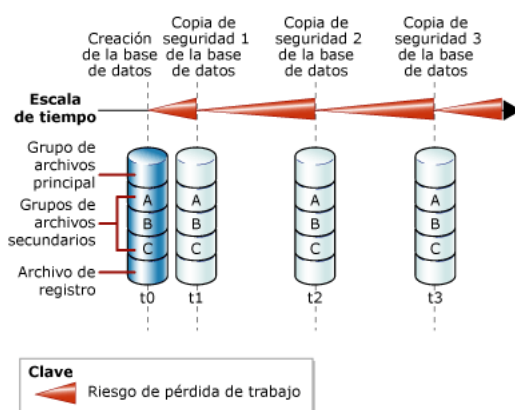
A medida que la base de datos se actualiza, la cantidad de datos que incluyen las copias de seguridad diferenciales aumenta.

La intención es que la incidencia de las copias en el rendimiento del servidor sea menor.



El proceso de restauración de la copia de seguridad es más lento. Si hiciésemos solo completas sería más rápido, pero se degradaría el rendimiento al realizar las copias.

La estrategia debe ser C .. D .. D ... C .. D ...



RESTAURACIÓN de copia de seguridad

La capacidad de restaurar lo antes posible una base de datos con la mínima pérdida de datos puede ser un factor importante en una instalación.

El concepto se denomina disponibilidad.

RESTAURACIÓN FULL BACKUP

Todos los modelos de recuperación de base de datos comienzan por una restauración completa de la base de datos a un momento concreto.

- La restauración de la base de datos es un proceso largo que puede costar un 30% más que la realización de la copia de seguridad de la base de datos.
- La restauración de una base de datos reescribe una base de datos con el mismo nombre (opcionalmente se puede cambiar). Si esta no existe, la operación de restauración crea los archivos y grupos de archivos para la base de datos.
- Como la creación de los archivos puede tardar tiempo es mejor no borrar la base de datos existente antes de restaurarla.

La sintaxis para una recuperación completa será la siguiente:

```
RESTORE DATABASE Adventureworks FROM DISK = 'C:\DEMO\BACKUP\AdvenFull.BAK'
```

```
WITH REPLACE
```

Este comando utilizará el contenido del fichero AdvenFull.BAK para la operación de restauración.

La opción REPLACE indica que se reescriba la base de datos existente *Adventureworks*.

RESTAURACIÓN DIFFERENTIAL BACKUP

Para restaurar una base de datos que utiliza el modelo Completa/Diferenciales se debe restaurar la copia de seguridad completa y después aplicar la última copia diferencial.

```
RESTORE DATABASE AdventureWorks
```

```
FROM DISK = 'Z:\SQLServerBackups\AdventureWorks.bak'
```

```
WITH FILE = 6
```

```
NORECOVERY;
```

```
RESTORE DATABASE AdventureWorks
```

```
FROM DISK = 'Z:\SQLServerBackups\AdventureWorks.bak'
```

```
WITH FILE = 9
```

RECOVERY;

NORECOVERY / RECOVERY

WITH NORECOVERY se pueden seguir restaurando otras copias de seguridad para poner al día la base de datos a un momento posterior.

WITH RECOVERY recupera la base de datos y no se pueden restaurar otras copias de seguridad. Es el valor predeterminado. Si no se finaliza con esta opción en la última restauración, la base de datos no será coherente (quedará inutilizable).

BACKUP - Opciones

TO DISK

La sintaxis básica BACKUP para especificar un archivo de copia de seguridad mediante su nombre de dispositivo físico es:

```
BACKUP DATABASE AdventureWorks
```

```
TO DISK = 'Z:\SQLServerBackups\AdventureWorks.bak';
```

```
GO
```

Un dispositivo lógico de copia de seguridad es un objeto de servidor definido por el usuario que apunta a un dispositivo físico de copia de seguridad.

```
BACKUP DATABASE AdventureWorks
```

```
TO AWcopiasseg;
```

```
GO
```

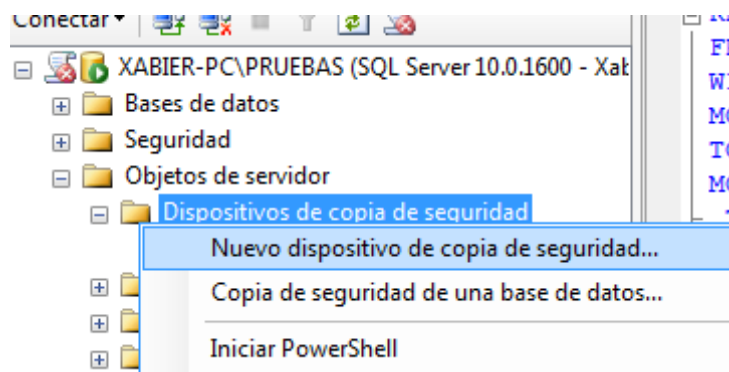
```
EXEC master.dbo.sp_addumpdevice
```

```
@devtype = N'disk',
```

```
@logicalname = N'AWcopiasseg',
```

```
@physicalname = 'D:\AWcopiasseg\copia_AW.bak'
```

```
GO
```



RESTAURACIÓN - Opciones

Restaurar una base de datos y mover archivos

En el ejemplo siguiente se restaura una base de datos completa y el registro de transacciones, y se mueve la base de datos restaurada.

Restauracion0.sql

ANTES

Nombre lógico	Tamaño inicial (MB)	Crecimiento automático	Ruta de acceso	Nombre de archivo
AdventureW...	1	En 16 MB, crecimiento sin límite	C:\Program Files\Microsoft SQL Server\MSSQL10.PRUEBAS\MSSQL\DATA	AdventureWorks_Data.mdf
AdventureW...	2	En 16 MB, crecimiento limitado a...	C:\Program Files\Microsoft SQL Server\MSSQL10.PRUEBAS\MSSQL\DATA	AdventureWorks_Log.ldf

Después

Archivos de la base de datos:					
Nombre lógico	os	Tamaño inicial (MB)	Crecimiento automático	Ruta de acceso	Nombre de archivo
AdventureW...		234	En 16 MB, crecimiento sin límite	C:\Program Files\Microsoft SQL Server\MSSQL10.PRUEBAS\MSSQL\DATA	NewAdvWorks.mdf
AdventureW...		242	En 16 MB, crecimiento limitado a ...	D:\	NewAdvWorks.ldf

La instrucción MOVE hace que se restauren los datos y el archivo de registro en las ubicaciones especificadas.

La instrucción RESTORE FILELISTONLY se usa para determinar el número y los nombres de los archivos de la base de datos que se están restaurando.

La nueva copia de la base de datos se llama *TestDB*.

Restauracion1.sql

FILE GROUP BACKUP

Es una estrategia alternativa al full backup. En lugar de hacer la copia de seguridad completa se puede realizar una copia de seguridad por grupo de archivo individual. El punto de partida de esta alternativa debe ser una copia de seguridad de todos los grupos de archivos.

Esta estrategia se utilizará en el caso de que el tamaño de la base de datos haga impracticable la recuperación de toda la base de datos.

BACKUP DATABASE <database name>

FILEGROUP = '<filegroup name>'

TO DISK = '<directory>\<filename>'

NOTA: Solo con forma de recuperación completa

Para realizar una copia de diferencial *grupal diferencial* utilizaremos la siguiente notación:

BACKUP DATABASE <database name>

FILEGROUP = '<filegroup name>'

TO DISK = '<directory>\<filename>'

WITH DIFFERENTIAL

A la base de datos *TestDB* le añadiremos dos grupos de archivos:

UNO y DOS.

Nos aseguramos de que su modelo de recuperación es completa.

Filegroup1.sql

Archivos de la base de datos:					
Nombre lógico	Tipo de archivo	Grupo de archivos	Tamaño inicial (MB)	Crecimiento automático	Ruta de acceso
AdventureW...	Datos de filas	PRIMARY	234	En 16 MB, crecimiento sin límite	C:\Users\Administrador\Documents\...
DOSDATA	Datos de filas	DOS	5	En 1 MB, crecimiento sin límite	C:\Users\Administrador\Documents\...
UNODATA	Datos de filas	UNO	5	En 1 MB, crecimiento sin límite	C:\Users\Administrador\Documents\...
AdventureW...	Registro	No aplicable	242	En 16 MB, crecimiento limitado a...	C:\Users\Administrador\Documents\...

Ejemplo:

Realizaremos una copia de seguridad completa de la base de datos.

Filegroup2.sql

Esta será la base para cada copia de seguridad diferencial que realicemos.

Realizaremos algunos cambios en cada uno de los grupos de archivos.

Filegroup3.sql

Realizaremos la copia de seguridad parcial de cada grupo de archivos.

Filegroup4.sql

Realizamos la restauración parcial de cada grupo de archivos.

Filegroup5.sql

Es una técnica que se puede utilizar en caso de tener grupos de archivos de solo lectura. Se podrá especificar que no se haga copia de seguridad de los grupos de archivos marcados como de solo lectura.

```
BACKUP DATABASE PUBS READ_WRITE_FILEGROUPS
```

```
TO DISK='C:\DEMO\BACKUP\PUBS1.BAK'
```

TRANSACTION LOG BACKUP

Para poder realizar copias de seguridad de este tipo será necesario tener configurada la base de datos con modelo de recuperación Completo o de registro masivo.

Como en todos los modelos de copias, Transaction LOG BACKUP se permite solamente después de haber realizado un FULL BACKUP. Ya que la copia transaccional únicamente contiene solamente un subconjunto de datos.

Cada LOG BACKUP comienza en el número de Secuencia de log donde terminó la anterior copia LOG BACKUP.

Observación

A veces, los imperativos legales de algunas empresas, o sus propias reglas, obligan al administrador a implementar una arquitectura que permita la implantación de la base de datos en un momento dado de un periodo anterior (días, meses, ...). Esto fuerza a introducir en el sistema las copias de los log's.

Toda transacción de la que se ha realizado una copia se puede borrar del registro de transacciones, lo que permite al sistema reutilizar espacio del log.

La sintaxis para realizar esta acción es:

```
BACKUP LOG <database name>
```

```
TO DISK=<directory\filename>
```

```
WITH INIT
```

- ✓ Permite restaurar una base de datos a cualquier momento de una copia de seguridad de registros (*recuperación a un momento dado*).
- ✓ Las desventajas de usar las copias de seguridad de registros son que requieren espacio de almacenamiento y aumentan la duración y la complejidad de las restauraciones.

Ejemplo copia de seguridad

Nos aseguramos que el modelo de recuperación de la base de datos AW es completa.

Realizamos la copia de seguridad completa en la base de datos AW.

Log1.sql

Insertamos una línea en la tabla dbo.t1

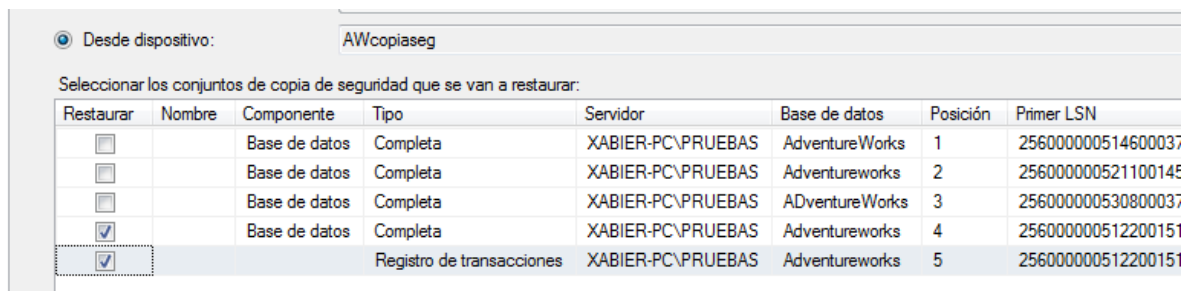
Realizamos una copia de seguridad del log

Insertamos una segunda línea en la tabla dbo.t1

Restauramos la copia de seguridad del log.

Restaurar la copia de seguridad del log hasta un momento concreto

Restauramos la copia de seguridad del log.



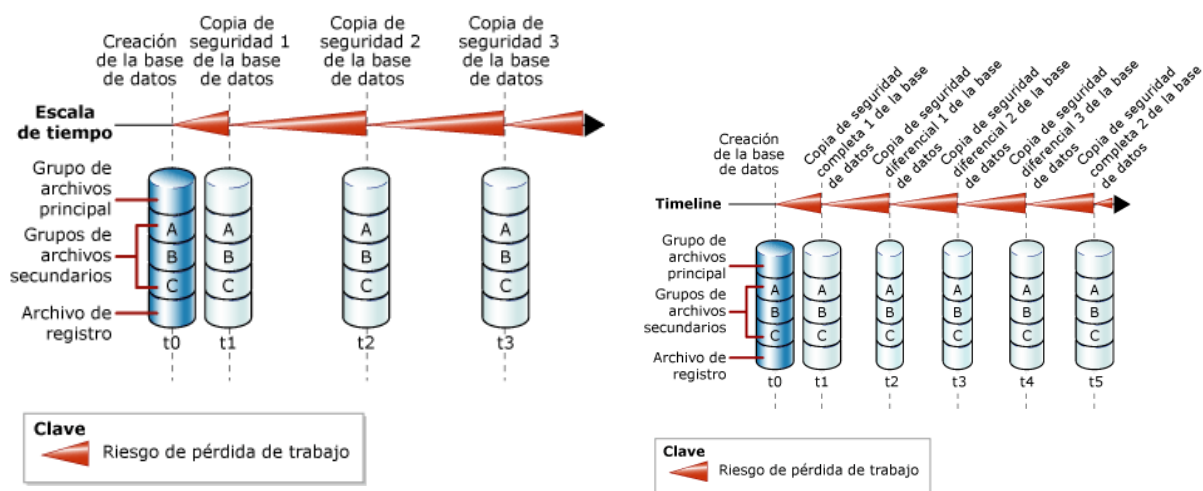
Restaurar	Nombre	Componente	Tipo	Servidor	Base de datos	Posición	Primer LSN
<input type="checkbox"/>		Base de datos	Completa	XABIER-PC\PRUEBAS	AdventureWorks	1	256000000514600037
<input type="checkbox"/>		Base de datos	Completa	XABIER-PC\PRUEBAS	Adventureworks	2	256000000521100145
<input type="checkbox"/>		Base de datos	Completa	XABIER-PC\PRUEBAS	ADventureWorks	3	256000000530800037
<input checked="" type="checkbox"/>		Base de datos	Completa	XABIER-PC\PRUEBAS	Adventureworks	4	256000000512200151
<input checked="" type="checkbox"/>		Registro de transacciones		XABIER-PC\PRUEBAS	Adventureworks	5	256000000512200151

Restaurar la copia de seguridad del log hasta un momento concreto

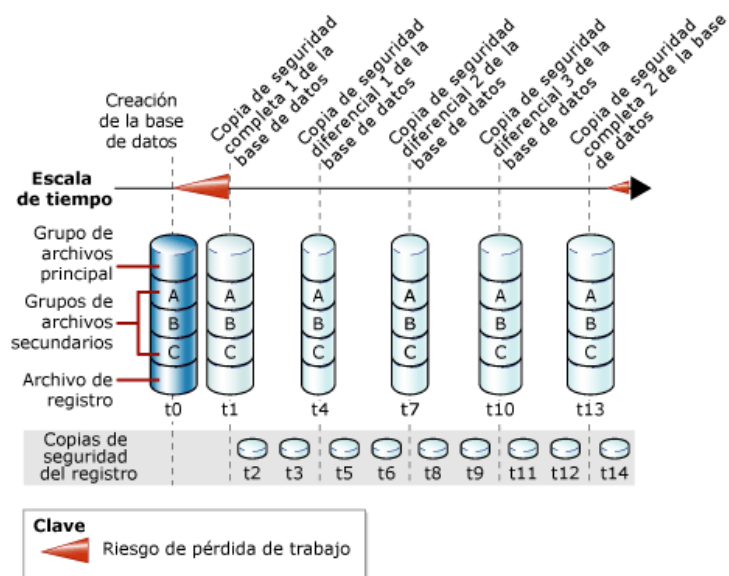
Estrategia de copia de seguridad

- Las estrategias de copia se implementan de forma programada.
- Dado que hay que programar, es necesario que el Agente del servicio del motor esté activo.
- Toda estrategia debe estar basada en un equilibrio entre seguridad, rendimiento y tiempo de no disponibilidad en el caso de “accidente”.

Modelo de recuperación Simple



Modelo de recuperación Completa



Más información

C/ Miracruz, 10 (Bº de Gros) 20001 Donostia

Telf.: 943 275819

email: seim@centroseim.com

