

Tema 5.1: Espejo (Mirroring) en SQL Server

Introducción al Mirroring en SQL Server

El mirroring (espejado) de bases de datos es una solución de alta disponibilidad y protección de datos ofrecida por SQL Server que permite mantener dos copias idénticas de una misma base de datos en servidores diferentes.

Conceptos fundamentales del Mirroring

Componentes principales:

1. **Base de datos principal (Principal):** La base de datos original que recibe todas las transacciones.
2. **Base de datos reflejada (Mirror):** La copia exacta de la base de datos principal.
3. **Testigo (Witness):** Opcional, un tercer servidor que permite el failover automático.

Modos de operación:

1. **Alto rendimiento (Asynchronous):** Las transacciones se envían al espejo sin esperar confirmación.
2. **Alta seguridad (Synchronous):** El principal espera confirmación del espejo antes de confirmar transacciones.
3. **Alta seguridad con failover automático:** Requiere un testigo para monitorear y activar failover automático.

Configuración del Mirroring en SQL Server

Requisitos previos:

- Ediciones Enterprise, Developer o Standard de SQL Server
- Mismo nivel de compatibilidad en ambas bases de datos

- Modelo de recuperación FULL
- Copia de seguridad inicial de la base de datos principal

Pasos para configurar:

```
-- En el servidor principal
BACKUP DATABASE MiBaseDeDatos TO DISK = 'C:\Backups\MiBaseDeDatos.bak'
WITH FORMAT;
GO

BACKUP LOG MiBaseDeDatos TO DISK = 'C:\Backups\MiBaseDeDatos.trn'
WITH FORMAT;
GO

-- En el servidor espejo
RESTORE DATABASE MiBaseDeDatos FROM DISK = 'C:\Backups\MiBaseDeDatos.bak'
WITH NORECOVERY, REPLACE;
GO

RESTORE LOG MiBaseDeDatos FROM DISK = 'C:\Backups\MiBaseDeDatos.trn'
WITH NORECOVERY;
GO

-- Configurar el mirroring (en el principal)
ALTER DATABASE MiBaseDeDatos
SET PARTNER = 'TCP://MirrorServerName:5022';
GO
```

Ventajas del Mirroring en SQL Server

1. **Alta disponibilidad:** Rápido failover en caso de fallo (especialmente en modo síncrono con testigo).
2. **Protección de datos:** Mantiene una copia exacta y actualizada de la base de datos.
3. **Mejor rendimiento:** El espejo puede usarse para operaciones de solo lectura (con snapshots).
4. **Configuración relativamente simple:** Comparado con otras soluciones como clustering.

Limitaciones y consideraciones

1. Solo protege una base de datos a la vez (no todo el servidor).
2. Requiere recursos adicionales (hardware, licencias).
3. El mirroring está en desuso desde SQL Server 2012 (Microsoft recomienda usar AlwaysOn Availability Groups).
4. Solo funciona dentro de la misma versión de SQL Server.

Comparación con otras tecnologías

Característica	Mirroring	AlwaysOn AG	Log Shipping
Failover automático	Sí (con testigo)	Sí	No
Múltiples bases datos	No	Sí	Sí (individualmente)
Réplicas de lectura	Solo con snapshots	Sí	No
Requiere edición Enterprise	No	Sí	No

Buenas prácticas

1. Usar hardware similar para principal y espejo.
2. Monitorear el estado del mirroring regularmente.
3. Configurar alertas para detectar problemas.
4. Considerar la latencia de red entre servidores.
5. Planear pruebas de failover periódicas.

Comandos útiles para monitoreo

```
-- Ver estado del mirroring
SELECT DB_NAME(database_id) AS DatabaseName,
       mirroring_state_desc,
       mirroring_role_desc,
       mirroring_partner_name
FROM sys.database_mirroring
WHERE mirroring_guid IS NOT NULL;
```

```
-- Forzar el servicio (en caso de fallo)  
ALTER DATABASE MiBaseDeDatos SET PARTNER FORCE_SERVICE_ALLOW_DATA_LOSS;
```

El mirroring en SQL Server es una tecnología efectiva para garantizar la alta disponibilidad y protección de datos, aunque actualmente está siendo reemplazada por AlwaysOn Availability Groups en versiones recientes. Su implementación sigue siendo relevante en entornos donde no se dispone de la edición Enterprise de SQL Server o para soluciones específicas que requieren una configuración más simple.

Explicación Detallada de los Códigos de Configuración para Mirroring en SQL Server

A continuación, desgloso cada parte de los comandos SQL utilizados para configurar el mirroring:

1. Creación de la copia de seguridad inicial (en el servidor principal)

```
BACKUP DATABASE MiBaseDeDatos TO DISK = 'C:\Backups\MiBaseDeDatos.bak'  
WITH FORMAT;
```

- **BACKUP DATABASE:** Comando para realizar una copia de seguridad completa de la base de datos.
- **MiBaseDeDatos:** Nombre de la base de datos que se va a reflejar.
- **TO DISK:** Especifica que la copia se guardará en disco.
- **'C:\Backups\MiBaseDeDatos.bak':** Ruta y nombre del archivo de backup.
- **WITH FORMAT:** Sobrescribe cualquier archivo existente y crea un nuevo conjunto de medios.

2. Copia de seguridad del log de transacciones (en el servidor principal)

```
BACKUP LOG MiBaseDeDatos TO DISK = 'C:\Backups\MiBaseDeDatos.trn'  
WITH FORMAT;
```

- **BACKUP LOG:** Realiza una copia de seguridad del log de transacciones (necesario para poner el mirror en estado de recuperación).
- **.trn:** Extensión típica para archivos de log de transacciones.
- Este paso es crucial porque el mirroring depende del log de transacciones para sincronizar las bases de datos.

3. Restauración de la base de datos (en el servidor espejo)

```
RESTORE DATABASE MiBaseDeDatos FROM DISK = 'C:\Backups\MiBaseDeDatos.bak'  
WITH NORECOVERY, REPLACE;
```

- **RESTORE DATABASE:** Comando para restaurar una base de datos desde backup.
- **FROM DISK:** Especifica la ubicación del archivo de backup.
- **WITH NORECOVERY:** Deja la base de datos en estado de recuperación, permitiendo aplicar logs posteriores (esencial para mirroring).
- **REPLACE:** Sobrescribe cualquier base de datos existente con el mismo nombre.

4. Restauración del log de transacciones (en el servidor espejo)

```
RESTORE LOG MiBaseDeDatos FROM DISK = 'C:\Backups\MiBaseDeDatos.trn'
WITH NORECOVERY;
```

- **RESTORE LOG:** Aplica el log de transacciones a la base de datos restaurada.
- **WITH NORECOVERY:** Mantiene la base de datos en estado de recuperación para prepararla para el mirroring.

5. Configuración del mirroring (en el servidor principal)

```
ALTER DATABASE MiBaseDeDatos
SET PARTNER = 'TCP://MirrorServerName:5022';
```

- **ALTER DATABASE:** Modifica la configuración de la base de datos.
- **SET PARTNER:** Establece el servidor espejo como partner de mirroring.
- **'TCP://MirrorServerName:5022':** Endpoint de comunicación con el servidor espejo.
 - **MirrorServerName:** Nombre del servidor que actuará como espejo.
 - **5022:** Puerto típico usado para mirroring (puede configurarse otro).

Configuración Adicional Necesaria

Antes de estos pasos, se deben configurar los endpoints en ambos servidores:

```
-- En ambos servidores (principal y espejo)
CREATE ENDPOINT MirroringEndpoint
```

```
STATE = STARTED
AS TCP (LISTENER_PORT = 5022)
FOR DATABASE_MIRRORING (ROLE = PARTNER);
```

- **CREATE ENDPOINT:** Crea un punto de comunicación para el mirroring.
- **LISTENER_PORT = 5022:** Puerto donde escuchará las conexiones de mirroring.
- **ROLE = PARTNER:** Especifica que este servidor puede actuar como principal o espejo.

Comando para Configurar el Testigo (Opcional)

```
ALTER DATABASE MiBaseDeDatos
SET WITNESS = 'TCP://WitnessServerName:5022';
```

- **SET WITNESS:** Configura un tercer servidor como testigo para habilitar failover automático.

Importante sobre la Secuencia de Configuración

1. Primero configurar el endpoint en AMBOS servidores
2. Realizar backups en el principal
3. Restaurar backups en el espejo (con NORECOVERY)
4. Iniciar el mirroring desde el principal
5. Si se usa, configurar el testigo por último

Cada uno de estos comandos es esencial para establecer correctamente el mirroring y garantizar que la base de datos espejo esté perfectamente sincronizada y lista para asumir el rol principal en caso de fallo.