

MANUAL SQL SERVER MANAGEMENT STUDIO – RICSE POMALAZA SOFÍA

MANUAL, **MODULO 6**, ADMINISTRACION DE RENDIMIENTO:

1. Monitoreo del Rendimiento de SQL Server:

El monitoreo de rendimiento permite **detectar problemas, optimizar recursos y mantener la base de datos rápida y estable**. SQL Server ofrece herramientas integradas para supervisar y analizar su comportamiento en tiempo real o histórico.

A. Herramientas de monitoreo

SQL Server Management Studio (SSMS):

- Vista de **Actividad Actual** (Activity Monitor): muestra uso de CPU, consultas lentas, bloqueos, etc.

SQL Server Profiler:

- Rastrea eventos en tiempo real (consultas, bloqueos, errores).
- Útil para detectar problemas específicos.

Extended Events:

- Alternativa moderna a Profiler, más eficiente y flexible.
- Permite registrar eventos y analizarlos sin sobrecargar el sistema.

Dynamic Management Views (DMVs):

- Consultas como sys.dm_exec_requests, sys.dm_os_wait_stats, sys.dm_exec_query_stats para obtener detalles técnicos de uso y rendimiento.

Performance Monitor (Windows):

- Permite monitorear el uso de recursos del sistema (CPU, memoria, disco) relacionados con SQL Server.

B. Métricas clave de Rendimiento

Estas métricas ayudan a evaluar si SQL Server está funcionando correctamente:

Métrica	¿Qué indica?
CPU Usage	Carga de procesamiento
Buffer Cache Hit Ratio	% de lecturas desde memoria (debe ser > 90%)
Page Life Expectancy (PLE)	Tiempo que una página de datos permanece en caché (bajo = presión de memoria)
Disk I/O	Lectura/escritura en disco, clave para performance
Wait Stats	En qué está esperando el servidor (bloqueos, I/O, CPU)
Query Duration	Tiempo que toman las consultas
Blocking Sessions	Procesos que están bloqueando a otros

C. Identificación de cuellos de botella

Un **cuello de botella** es cualquier recurso que limita el rendimiento general. Puedes identificarlos observando:

1. **CPU alta:**
 - Consultas pesadas, sin índices.
 - Usa: sys.dm_exec_query_stats, Activity Monitor.
2. **Memoria insuficiente:**
 - Buffer Cache Hit Ratio bajo, PLE muy bajo.
 - Solución: más RAM o ajustar consultas.
3. **Disco lento:**
 - Tiempos de I/O altos.
 - Revisa con sys.dm_io_virtual_file_stats.
4. **Bloqueos y Deadlocks:**
 - Sesiones en espera prolongada.
 - Usa sp_who2, sys.dm_exec_requests.
5. **Consultas mal optimizadas:**
 - Usa planes de ejecución (Execution Plan) para ver si faltan índices o hay escaneos completos.

2. Optimización de Consultas

Optimizar consultas significa **mejorar su rendimiento** para que usen menos recursos (CPU, memoria, disco) y se ejecuten más rápido. Esto se logra analizando su comportamiento y afinando su estructura o el diseño de los datos.

A. Planes de ejecución de consultas

El **plan de ejecución** muestra cómo SQL Server procesa una consulta: qué índices usa, si escanea o busca, el orden de las operaciones, etc.

Cómo ver el plan de ejecución:

En SSMS, selecciona tu consulta y presiona:

- **Ctrl + M** → Muestra el plan estimado.
- **Ctrl + L** → Ejecuta y muestra el plan real.

Qué buscar:

- **Table Scan**: SQL revisa toda la tabla (lento).
- **Index Seek**: Usa índice para buscar más rápido.
- **Key Lookup**: Buscar datos adicionales fuera del índice.

Usa este análisis para identificar qué partes de la consulta necesitan índices o reescritura.

B. Índices

Los índices mejoran la velocidad de búsqueda, ordenamiento y filtros. Son como "índices de un libro".

Tipos principales:

- **Índice agrupado (Clustered)**: organiza físicamente los datos. Solo puede haber uno por tabla.
- **Índice no agrupado (Non-clustered)**: estructura separada para acelerar consultas.

Ejemplo:

```
-- Índice no agrupado para acelerar búsquedas por nombre  
CREATE INDEX idx_Nombre ON Productos(Nombre);
```

Buenas prácticas:

- Crea índices en columnas usadas en WHERE, JOIN, ORDER BY.
- Evita demasiados índices, ya que afectan el rendimiento de escritura.
- Revisa fragmentación con sys.dm_db_index_ph

C. Estadísticas de tablas

Las **estadísticas** ayudan a SQL Server a decidir el mejor plan de ejecución. Muestran información sobre la distribución de valores en una columna.

¿Qué pasa si están desactualizadas?

SQL Server podría elegir un mal plan de ejecución, lo que **ralentiza las consultas**.

Cómo actualizarlas:

```
-- Actualizar estadísticas de una tabla  
UPDATE STATISTICS Productos;
```

También puedes usar esta opción al recompilar:

```
-- Recompilar procedimiento con estadísticas nuevas  
EXEC sp_recompile 'NombreProcedimiento';
```

SQL Server normalmente actualiza estadísticas automáticamente, pero en tablas grandes o con muchos cambios, **es recomendable hacerlo manualmente** periódicamente.

3. Optimización de Consultas

SQL Server depende del **hardware** y su configuración para rendir al máximo. Ajustar cómo usa **memoria, disco y CPU** puede mejorar significativamente su rendimiento.

A. Configuración de la memoria

SQL Server administra la memoria dinámicamente, pero puedes configurar **límites máximos y mínimos** según los recursos del servidor.

Comandos útiles:

```
-- Ver configuración actual de memoria
EXEC sp_configure 'show advanced options', 1;
RECONFIGURE;
EXEC sp_configure 'max server memory';

-- Establecer límite de memoria (en MB)
EXEC sp_configure 'max server memory', 4096; -- 4 GB, por ejemplo
RECONFIGURE;
```

Buenas prácticas:

- No dejar que SQL Server use toda la RAM (deja al menos 1-2 GB para el sistema operativo).
- En servidores compartidos, limitar el uso de memoria por instancia.

B. Configuración del disco

El **acceso a disco** impacta directamente en el rendimiento de lectura/escritura de datos y archivos de log.

Recomendaciones:

- Separar los archivos:
 - Datos (.mdf) en un disco.
 - Logs (.ldf) en otro disco.
 - Backups en una ubicación distinta.
- Usar discos SSD si es posible.
- Ver estadísticas de I/O con:

```
-- Ver actividad de lectura/escritura por archivo
SELECT
    DB_NAME(database_id) AS BaseDatos,
    file_id,
    io_stall_read_ms, io_stall_write_ms,
    num_of_reads, num_of_writes
FROM sys.dm_io_virtual_file_stats(NULL, NULL);
```

C. Configuración del procesador

SQL Server usa múltiples núcleos para ejecutar consultas en paralelo, pero también puedes controlar su uso.

Configuraciones clave:

- **MAXDOP (Max Degree of Parallelism):** define cuántos núcleos usar por consulta.

```
-- Consultar configuración
EXEC sp_configure 'max degree of parallelism';

-- Establecer a 4 núcleos, por ejemplo
EXEC sp_configure 'max degree of parallelism', 4;
RECONFIGURE;
```

Buenas prácticas:

- Para servidores OLTP (muchas transacciones pequeñas): MAXDOP = 1 o bajo.
- Para servidores OLAP (consultas grandes): permitir más núcleos.