MANUAL SQL SERVER MANAGEMENT STUDIO – CARLOS JOSÉ TORRE GARCÍA

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.

1. 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.

1. 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 |

1. Identificacion 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.
6. Optimizacion 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.

1. 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.

1. Indices

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:

#### 

#### 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

1. Estadisticas 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:



También puedes usar esta opción al recompilar:



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

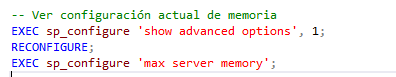
1. Optimizacion de Consultas

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

1. Configuracion de la memoria

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

Comandos útiles:





#### Buenas prácticas:

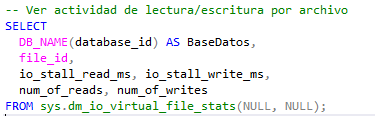
* No dejar que SQL Server use toda la RAM (deja al menos 1-2 GB para el sistema operativo).
* En servidores compartidos, limitar el uso de memoria por instancia.

1. Configuracion 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:

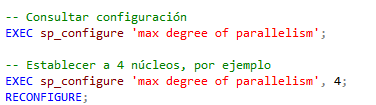


1. Configuracion del procesador

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.



#### Buenas prácticas:

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