



© JMA 2009. All rights reserved

Contenidos

1. Reportes SQL de valor (SSRS) Depende de sus bases de datos
2. Optimización de la infraestructura
 - a) Consultas T-SQL Depende de sus bases de datos
 - b) Query Store
 - c) Extended Events
 - d) DMVs
 - e) Planes de Ejecución
3. Mantenimiento SQL SERVER
 - a) Planes de Mantenimiento
 - b) Tareas programadas
 - c) Procesos SQL Server
4. Mantenimiento de índices SQL Server
 - a) Fragmentación índices
 - b) Mantenimiento
 - c) Optimización
5. Conocer todos los servicios que presta SQL

© JMA 2009. All rights reserved

ARQUITECTURA FÍSICA

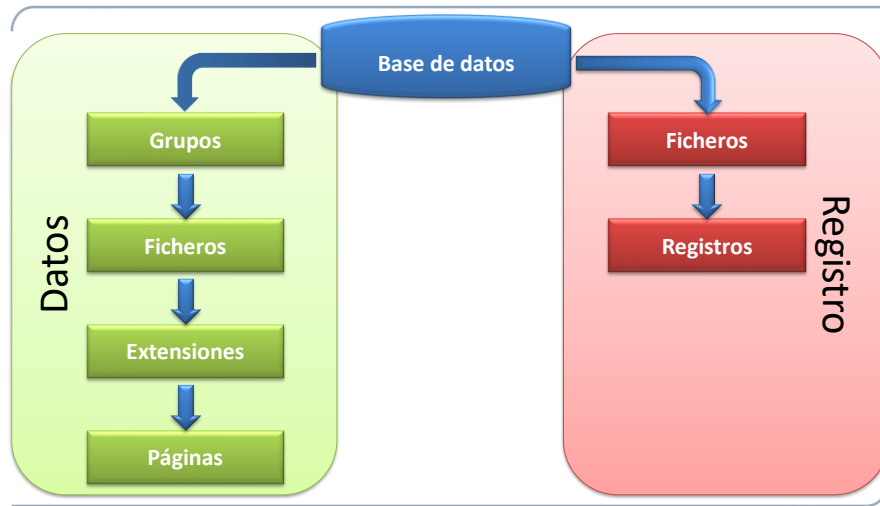
© JMA 2016. All rights reserved

Composición de un BBDD

- Desde el punto de vista lógico
 - Compuesta por objetos
 - Desde el punto de vista físico
 - Páginas, extensiones y ficheros
 - Dos tipos de ficheros
 - Datos
 - Contiene los objetos de la Base de Datos
 - Registro
 - Histórico de modificaciones realizadas en la Base de Datos
 - Sirve para garantizar la integridad de la Base de Datos
-

© JMA 2016. All rights reserved

Estructura Física



© JMA 2016. All rights reserved

Páginas

- La página es la unidad fundamental de asignación de almacenamiento de datos en SQL Server.
- El tamaño de página es de 8 KB (8192 bytes) y no se puede cambiar. Esto significa que las bases de datos de SQL Server tienen 128 páginas por megabyte.
- Cada página empieza con un encabezado de 96 bytes, que se utiliza para almacenar la información del sistema acerca de la página.
- Esta información incluye el número de página, el tipo de página, el espacio disponible en la página y el Id. de unidad de asignación del objeto propietario de la página.

© JMA 2016. All rights reserved

Tipos de Páginas

- **Datos:** Filas de datos con todos los datos excepto los datos text, ntext e image
- **Índice:** Entradas de índices
- **Texto o imagen** (LOB_DATA, ROW_OVERFLOW_DATA): Tipos de datos de objetos grandes (LOB): Datos text, ntext e image, opción text in row
- **Espacio libre en páginas** (PFS): Información acerca del espacio libre disponible en las páginas
- **Mapa de asignación de índices** (IAM): Información acerca de las extensiones utilizadas por una tabla o un índice
- **Mapa de asignación global** (GAM), **Mapa de asignación global secundaria** (SGAM): Información acerca de las extensiones asignadas
- **Mapa cambiado masivamente** (BCM): Información acerca de las extensiones modificadas por operaciones masivas desde la última instrucción BACKUP LOG
- **Mapa cambiado diferencial** (DCM): Información acerca de las extensiones que han cambiado desde la última instrucción BACKUP DATABASE

© JMA 2016. All rights reserved

Tipos de longitudes de datos

- Longitud predefinida
 - El dato siempre ocupa lo mismo en función al tipo de dato elegido.
- Longitud fija
 - El dato siempre ocupa lo mismo en función al tipo de dato elegido y la longitud definida.
- Longitud variable
 - La ocupación viene determinada por la longitud del dato y el tipo, se almacena en paginas de datos.
- Longitud variable – LOB
 - La ocupación viene determinada por la longitud del dato y el tipo, se almacena en paginas LOB.
- Longitud variable – MAX
 - La ocupación viene determinada por la longitud del dato y el tipo, se almacena en paginas de datos o paginas LOB en función del espacio libre.

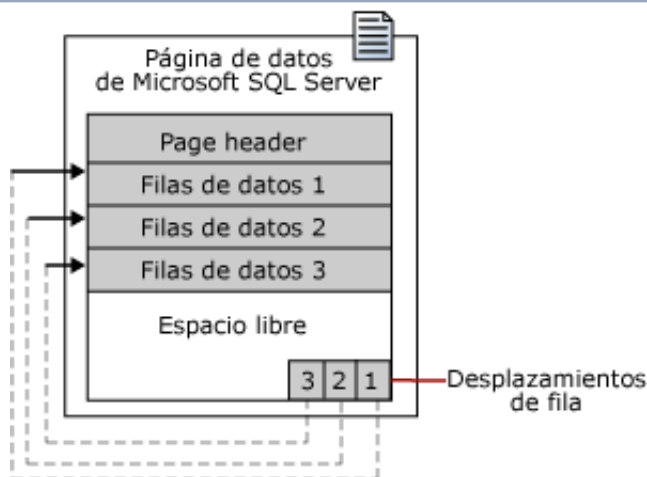
© JMA 2016. All rights reserved

Paginas de Datos

- La página está vinculada a un único objeto.
- La página puede contener varias filas.
- Una fila tiene que estar contenida en una página (más las páginas adicionales para las columnas LOB con un puntero de texto de 16 bytes).
- Las filas de datos son colocadas en posiciones consecutivas, de tal forma que una nueva fila pueda ocupar todo el espacio libre de la página.
- De cada página se pueden aprovechar 8060 bytes para almacenar datos.
- Al final de la página, comienza una tabla de desplazamiento de fila que contiene una entrada para cada fila de la página y registra la distancia del primer byte de la fila desde el inicio de la página. Las entradas en la tabla ocupan 2 bytes y están en orden inverso a la secuencia de las filas de la página.

© JMA 2016. All rights reserved

Paginas de Datos



© JMA 2016. All rights reserved

Paginas LOB

- Los datos LOB se guardan en una cadena enlazada de páginas LOB: divide el dato en bloques de 8 Kb y asigna cada bloque a una página LOB_DATA.
- El enlazamiento determina un máximo de 2^{18} paginas por lo el tamaño máximo de un dato LOB es de 2 Gb.
- Se requiere una cadena LOB para cada celda, columna de la fila, no nula.
- Una página LOB solo contiene datos de una única celda.
- Los datos LOB no nulos ocupan la menos una página LOB, mas el puntero de 16 bytes almacenado a nivel de fila en la página de datos.

© JMA 2016. All rights reserved

Optimización del espacio

- text in row:
 - Al activarlo, los datos LOB se guardan en la página de datos si tienen espacio.
 - Se puede fijar el limite entre 24 hasta 7000.
 - Puede provocar problemas de rendimiento.
- large value types out of row:
 - Activa el desbordamiento de página, cuando la fila no cabe en la página trata la columnas de longitud variable como si fueran LOB.
 - Mueve dinámicamente una o más columnas de longitud variable a páginas de la unidad de asignación ROW_OVERFLOW_DATA, empezando por la columna con el mayor ancho.
 - La longitud máxima de columna sigue siendo 8000, es la combinación de columnas la que la puede superar.
 - Puede empeorar el rendimiento de consultas y ordenaciones, suele ser preferible partición vertical.
- Estas opciones se configuran con el procedimiento almacenado: `sp_tableoption`

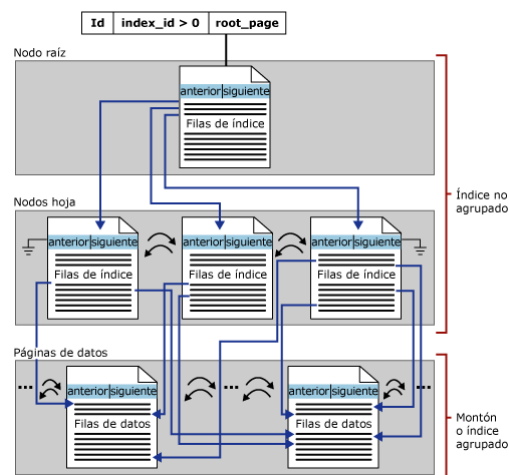
© JMA 2016. All rights reserved

Paginas de Índices

- Un índice es una estructura de disco asociada con una tabla o una vista que acelera la recuperación de filas de la tabla o de la vista.
- Un índice contiene claves generadas a partir de una o varias columnas ordenadas ascendente o descendentemente de la tabla o la vista.
- Dichas claves están almacenadas en una estructura (árbol b) que permite la búsqueda de forma rápida y eficiente de la fila o filas asociadas a los valores de cada clave.
- Es una estructura que maximiza la anchura y minimiza la profundidad.

© JMA 2016. All rights reserved

Paginas de Índices



© JMA 2016. All rights reserved

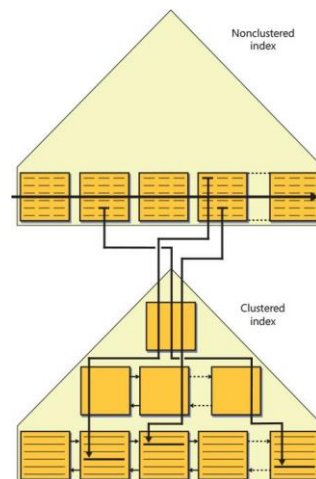
Paginas de Índices

- Están compuestas por registros de indexación, uno por fila de la tabla.
- Cada Registro de indexación contiene:
 - Clave de indexación:
 - compuesta por los valores de las columnas para la fila
 - como máximo de 900 bytes y 16 columnas
 - Localizador de fila:
 - Sin índice agrupado: RowID, el localizador es un puntero que se genera a partir del identificador (Id.) de archivo, el número de página y el número de la fila dentro de la página (8 bytes).
 - Con índice agrupado: es el valor de la clave del índice agrupado para la fila.
 - Columnas Incluidas.

© JMA 2016. All rights reserved

Tipos de índices: Agrupado

- Los índices agrupados ordenan y almacenan las filas de los datos de la tabla o vista de acuerdo con los valores de la clave del índice.
- Son columnas incluidas en la definición del índice.
- Sólo puede haber un índice clúster por cada tabla, porque las filas de datos sólo pueden estar ordenadas de una forma.
- La única ocasión en la que las filas de datos de una tabla están ordenadas es cuando la tabla contiene un índice clúster.
- Cuando una tabla tiene un índice clúster, la tabla se denomina tabla agrupada.
- Si una tabla no tiene un índice clúster, sus filas de datos están almacenadas en una estructura sin ordenar denominada montón.
- Puede ser único, garantiza que la clave de índice no contenga valores duplicados y, por tanto, cada fila de la tabla o vista es en cierta forma única.



© JMA 2016. All rights reserved

Tipos de índices: No agrupado

- Los índices no agrupados tienen una estructura separada de las filas de datos. Se pueden establecer como únicos.
- Un índice no agrupado contiene los valores de clave de índice no agrupado y cada entrada de valor de clave tiene un puntero a la fila de datos que contiene el valor clave.
- El puntero de una fila de índice no agrupado hacia una fila de datos se denomina localizador de fila.
- La estructura del localizador de filas depende de si las páginas de datos están almacenadas en un montón o en una tabla agrupada.
- Si están en un montón, el localizador de filas es un puntero hacia la fila.
- Si están en una tabla agrupada, el localizador de fila es la clave de índice agrupada.
- Se puede incluir columnas sin clave al nivel hoja de un índice no agrupado con el fin de eludir los límites existentes para las claves de índice, 900 bytes y columnas de 16 claves, así como para ejecutar consultas indizadas y totalmente cubiertas.

© JMA 2016. All rights reserved

Filtrado de índices

- Un índice filtrado es un índice no clúster optimizado, especialmente indicado para atender consultas que realizan selecciones a partir un subconjunto bien definido de datos.
- Utiliza un predicado de filtro para indizar una parte de las filas de la tabla.
- Un índice filtrado bien diseñado puede, en relación con los índices de tabla completa:
 - mejorar el rendimiento de las consultas y mayor calidad del plan
 - reducir los costos de mantenimiento
 - reducir el coste de almacenamiento del índice.
- Para diseñar índices filtrados efectivos, es importante entender qué consultas utiliza la aplicación y cómo se relacionan con los subconjuntos de datos.

© JMA 2016. All rights reserved

Otros tipos de índices

- Espacial
 - Un índice espacial proporciona la capacidad de realizar de forma más eficaz determinadas operaciones en objetos espaciales (datos espaciales) en una columna del tipo de datos geometry.
 - El índice espacial reduce el número de objetos a los que es necesario aplicar las operaciones espaciales, que son relativamente costosas.
- XML
 - Representación dividida y permanente de los objetos XML binarios grandes (BLOB) de la columna de tipo de datos xml.
- Texto completo
 - Tipo especial de índice funcional basado en símbolos (token) que crea y mantiene el motor de texto completo de Microsoft para SQL Server.
 - Proporciona la compatibilidad adecuada para búsquedas de texto complejas en datos de cadenas de caracteres.
 - Se almacenan fuera de los ficheros de la base de datos.

© JMA 2016. All rights reserved

Factor de relleno y fragmentación

- El factor de relleno determina la densidad de ocupación de páginas por parte del índice y fija de llenado de las paginas al crear el índice.
- La fragmentación de índices ocurre cuando los índices tienen páginas en las que la ordenación lógica, basada en el valor de clave, no coincide con la ordenación física dentro del archivo de datos.
- La fragmentación de índices provoca un desequilibrio en el llenado de las paginas de índice.
- Los índices muy fragmentados pueden reducir el rendimiento de la consulta y ralentizar la respuesta de la aplicación.
- Con la función del sistema **sys.dm_db_index_physical_stats** se podrá detectar la fragmentación de un índice específico, de todos los índices de una tabla o vista indizada, de todos los índices de una base de datos o de todos los índices de todas las bases de datos.
- Se puede solucionar la fragmentación del índice reorganizándolo o volviéndolo a generar.

© JMA 2016. All rights reserved

Número de Índices

- El número de índices afecta al rendimiento y ocupación de la base de datos.
- Es necesario mantener el número de índices al mínimo estrictamente necesario.
- Las restricciones PRIMARY KEY crean índices de forma automática de tipo UNIQUE. Por defecto como agrupados.
- Las restricciones UNIQUE crean índices UNIQUE de forma automática.
- Como máximo se puede tener:
 - Un índice agrupado por tabla.
 - 249 índices no agrupados (incluidos los creados por restricciones PRIMARY KEY o UNIQUE).

© JMA 2016. All rights reserved

Recomendaciones

- Indexar:
 - Tablas con muchas páginas (no necesariamente filas)
 - Columnas con un amplio rango de valores
 - Columnas que se usan: en consultas, agrupaciones, funciones de agregación y ordenaciones.
 - Columnas usadas en uniones
- NO Indexar:
 - Tablas que ocupen poco (pocas páginas)
 - Tablas con muchas modificaciones pero pocas consultas.
 - Columnas de gran tamaño
- El Asistente para optimización de índices suministra recomendaciones en función a la monitorización del sistema.

© JMA 2016. All rights reserved

Realizar operaciones de índice en línea

- Por defecto, las operaciones generar o volver a generar un índice de mantienen bloqueos exclusivos de los datos subyacentes y los índices asociados.
- Gracias a la opción ONLINE, es posible que usuarios simultáneos obtengan acceso a los datos de la tabla subyacente o del índice, así como a los índices no clúster asociados durante las operaciones de desfragmentación del índice.
- Para volver a generar un índice en línea:
 - Seleccionar Índice>Propiedades>Opciones:
 - Permitir procesamiento DML en línea = True
 - ALTER INDEX ... REBUILD WITH (ONLINE = ON);

© JMA 2016. All rights reserved

Directrices para operaciones en línea

- Se recomienda realizar operaciones de índices en línea en entornos empresariales que funcionan 24 horas al día, siete días a la semana, y en los que resulta fundamental la actividad simultánea de los usuarios durante las operaciones de índices.
- La tabla subyacente no se puede modificar (DDL), truncar o quitar mientras se está llevando a cabo una operación de índice en línea.
- Normalmente, las operaciones de índice en línea son más lentas que las operaciones de índice sin conexión equivalentes, independientemente del nivel de actividad de actualización simultánea.
- Como las estructuras de origen y de destino se mantienen durante la operación de índice en línea, el uso de recursos para insertar, actualizar y eliminar transacciones aumenta, potencialmente hasta el doble.
- Las operaciones de índice a gran escala pueden generar grandes cargas de datos que pueden hacer que el registro de transacciones se llene rápidamente.
- La recompilación de índices en línea puede provocar más fragmentación.

© JMA 2016. All rights reserved

Extensiones

- Unidad básica de almacenamiento de datos (E/S) que consta de 8 páginas contiguas físicamente. El SQL Server lee y escribe extensiones completas.
- Las extensiones están compuestas por 64 KB contiguos de espacio en disco, es decir, 16 extensiones por megabyte.
- Para hacer que la asignación de espacio sea eficaz, no se asigna extensiones completas a tablas con pequeñas cantidades de datos.
- Dos tipos de extensiones:
 - Mixtas
 - Pueden contener datos de hasta 8 objetos de la BBDD, uno por página
 - Uniformes
 - Contienen datos de un único objeto de la BBDD
- Cuando se crea un objeto se sitúa en una extensión mixta (<= v.14)
- Según se va llenando de datos, pasan a ocupar una extensión uniforme

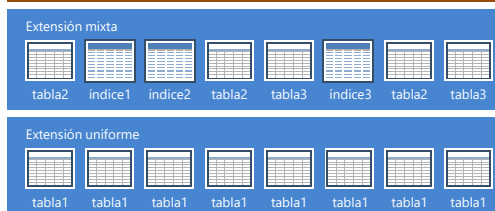
© JMA 2016. All rights reserved

Extensiones y páginas

Extensión de cabecera
de fichero



Extensiones



GAMs y SGAMs

Current use of extent	GAM bit setting	SGAM bit setting
Free, not being used	1	0
Uniform, extent, or full mixed extent	0	0
Current use of extent	0	1

© JMA 2016. All rights reserved

Administrar las asignaciones

- Las páginas GAM registran las extensiones que han sido asignadas. Las páginas SGAM registran las extensiones que actualmente se están utilizando como extensiones mixtas y además tienen al menos una página sin utilizar. Utilizan un bit por cada extensión y cada una cubre 64.000 extensiones o 4 GB de datos.
- Las páginas Espacio disponible en páginas (PFS) registran el estado de asignación de cada página, si una página concreta está asignada y la cantidad de espacio libre en cada página. Cada PFS cubre 8088 páginas o 64 Mb de datos.
- Una página del Mapa de asignación de índices (IAM) gestiona el espacio utilizado por los objetos y asigna las extensiones en una parte de 4 GB de un archivo de base de datos usado por una unidad de asignación (IN_ROW_DATA, ROW_OVERFLOW_DATA, LOB_DATA). Cada una cubre 64.000 extensiones o 4 GB de datos.

© JMA 2016. All rights reserved

Ficheros de Datos

- De dos tipos
 - Primario (.mdf): Un fichero de datos primario por BBDD (uno y solo uno). Almacena la información del catálogo y las tablas del sistema de la Base de Datos
 - Secundario (.ndf): Ficheros adicionales que pueden contener datos. Utilizados para balancear cargas y mejorar rendimiento
- Los archivos tienen dos nombres:
 - logical_file_name es el nombre que se utiliza para hacer referencia al archivo en las instrucciones Transact-SQL.
 - os_file_name es el nombre del archivo físico en disco.
- Se dimensionan en Megabytes (16 extensiones por Megabyte, 128 páginas por Megabyte)
 - Los ficheros pueden crecer automáticamente: porcentualmente o valor absoluto
 - Se puede impedir o limitar el crecimiento a un número máximo de megas (Máximo 16 TB)
 - Balancean la ocupación

© JMA 2016. All rights reserved

Grupos de Ficheros

- Permiten una gestión más óptima del espacio de una Base de Datos
- Como mínimo cuentan con uno, el PRIMARY
 - No puede ser borrado de la Base de Datos
 - Contiene el fichero Primario (.mdf):.
- El sistema balancea la carga de los datos proporcionalmente a los ficheros de datos de cada grupo
 - Permite un mejor rendimiento en la E/S
- Cada tabla se puede almacenar en grupos distintos, un grupo para las filas, otro para los LOB y cada índice en un grupo.
- Uno de los grupos de ficheros debe ser especificado como soporte de datos por defecto, se le asignaran los elemento que no indiquen lo contrario.
- Los grupos pueden ser marcados de solo lectura, con lo que se impide escribir en los ficheros del grupo.
- Permiten balancean la carga de rendimientos

© JMA 2016. All rights reserved

Particiones

- La creación de particiones en una base de datos mejora el rendimiento y simplifica el mantenimiento. Al dividir una tabla grande en tablas individuales más pequeñas, las consultas que tengan acceso únicamente a una parte de los datos pueden ejecutarse con mayor rapidez, ya que deben recorrer menos datos. Las tareas de mantenimiento (por ejemplo, volver a generar los índices o hacer copias de seguridad de una tabla), pueden ejecutarse con mayor rapidez.
- El particionamiento horizontal divide una tabla en varias particiones que contiene el mismo número de columnas pero menos filas. Se implementa con:
 - Funciones y Esquemas de partición
 - Tablas individuales (se reúnen en Vistas con consultas UNION)
- El particionamiento vertical divide una tabla en varias tablas que contienen menos columnas. La división de fila agrupa columnas de la tabla original y las reparte en varias tablas con menos columnas. Cada fila lógica de una tabla dividida coincide con la misma fila lógica en las demás tablas, con la misma PRIMARY KEY como FOREIGN KEY (requiere JOINS).

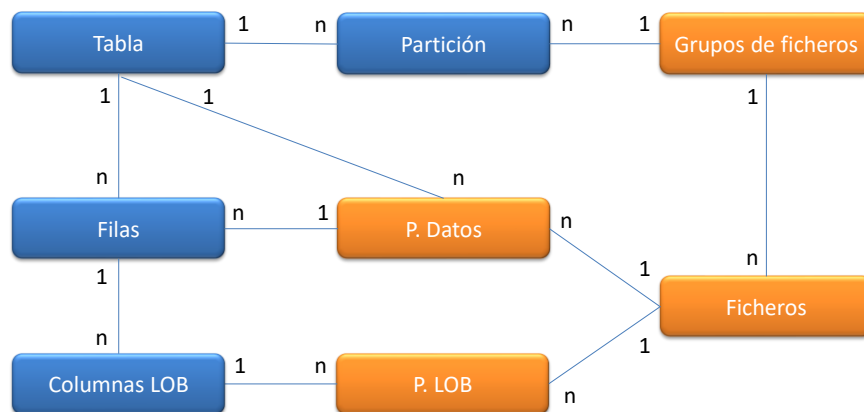
© JMA 2016. All rights reserved

Tablas e índices con particiones

- Sólo se admiten en las ediciones Enterprise.
- La partición facilita el uso de tablas e índices grandes, ya que permite administrar y tener acceso a subconjuntos de datos de forma rápida y eficaz, a la vez que mantiene la integridad de la recopilación de datos.
- Los datos se dividen en sentido horizontal y se reparten por más de un grupo de archivos dentro de la misma base de datos, puede tener un máximo de 1.000 particiones.
- La tabla o el índice se tratarán como una sola entidad lógica cuando se realicen consultas o actualizaciones en los datos.
- Útiles para tablas muy grandes con una división lógica establecida y mejorar el aprovechamiento de recursos.
- Una **función de partición** define la forma de asignar las filas de una tabla o un índice a un conjunto de particiones a partir de los valores de determinadas columnas, denominadas columnas de partición.
- Un **esquema de particiones** asigna cada partición especificada con la función de partición a un grupo de archivos.

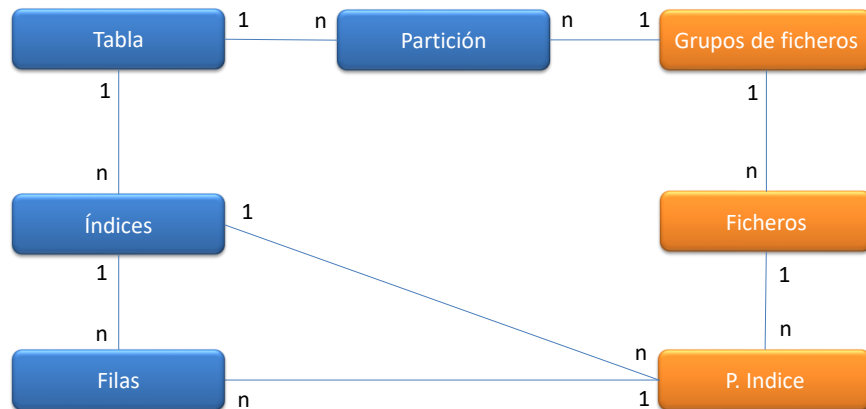
© JMA 2016. All rights reserved

Tabla: Almacenamiento de Datos



© JMA 2016. All rights reserved

Tabla: Almacenamiento de Índices



© JMA 2016. All rights reserved

Operaciones de Partición

- Crear función de partición
`CREATE PARTITION FUNCTION PF (int) AS RANGE RIGHT FOR VALUES (2014, 2015)`
- Crear esquema de partición
`CREATE PARTITION SCHEME PS AS PARTITION PF TO (FG_HIST, FG_ACT, FG_FUT);`
- Quitar una partición y mezclar sus valores en la partición anterior
`ALTER PARTITION FUNCTION PF() MERGE RANGE(2015);`
- Añadir una nueva partición
`ALTER PARTITION SCHEME PS NEXT USED FG2016;`
`ALTER PARTITION FUNCTION PF() SPLIT RANGE(2016);`

© JMA 2016. All rights reserved

Compresión de datos

- SQL Server admite la compresión de fila y de página para las tablas e índices.
- Ayuda para reducir el tamaño de la base de datos.
- Puede contribuir a mejorar el rendimiento de las cargas de trabajo que hacen un uso intensivo de las operaciones de E/S porque los datos se almacenan en menos páginas y las consultas deben leer menos páginas del disco.
- Requieren recursos de CPU adicionales en el servidor de base de datos para comprimir y descomprimir los datos, mientras los datos se intercambian con la aplicación.
- Para las tablas e índices con particiones, la opción de compresión se puede configurar para cada partición una configuración de compresión diferente.

© JMA 2016. All rights reserved

Nivel de compresión

- El tipo de compresión se puede establecer en ROW, PAGE o NONE
 - ROW: Se realiza a nivel de fila. Todas las columnas de longitud fija se tratan como si fueran de longitud variable. Ocupan lo que el dato contenido. No afecta a las columnas LOB.
 - PAGE: Se realiza a nivel de página completa, empieza con la compresión ROW, a la que se le aplica los algoritmos de compresión por prefijos y patrones (diccionarios).
- La compresión conlleva una sobrecarga de datos que puede superar la ocupación sin comprimir.

```
ALTER TABLE Tabla REBUILD WITH (DATA_COMPRESSION = PAGE);  
ALTER TABLE TablaParticionada REBUILD PARTITION = 2 WITH  
(DATA_COMPRESSION = ROW) ;
```

© JMA 2016. All rights reserved

Estimación de la ocupación

- Para calcular el espacio necesario para almacenar datos en una tabla:
 1. Calcular el espacio necesario para las filas:
 1. Calcular el numero de paginas de datos.
 2. Calcular el numero de páginas LOB por cada columna, sumar todas las columnas y multiplicar por el número de filas.
 2. Calcular el espacio necesario para los índices:
 1. Por cada índice, calcular el numero de paginas de índice.
 2. Sumar el numero de páginas de cada índice.
 3. Sumar el espacio necesario para las filas con el espacio necesario para los índices.
- El resultado viene expresado en páginas, para obtenerlo en megas, dividirlo entre 128.

© JMA 2016. All rights reserved

Estimar tamaños

- Tabla no agrupada (montón)
 - IN_ROW_DATA
 - $LF = \sum(TMC) + 24 * NCL$
 - $FxP = FLOOR(8096 / LF)$
 - $NPD = CEILING(NF / FxP)$
 - Por columna LOB
 - $NPL = CEILING(TML / 8000) * NF$
 - Por índice
 - $LRI = \sum(TMC) + 8$
 - $FxN = FLOOR(8096 / LRI)$
 - $NPI = CEILING(NF / FxN)$
 - Total
 - $NP = NPD + \sum(NPL) + \sum(NPI)$
- Tabla no agrupada (montón)
 - Por índice
 - $LRI = \sum(TMC) + \sum(TMCA)$
 - $FxN = FLOOR(8096 / LRI)$
 - $NPI = CEILING(NF / FxN)$
- En mb: $NP / 128$

Abreviaturas:

NF: Número de filas (inicial, previsto, periodo temporal, ...)
 LF: Longitud de fila (bytes)
 TMC: Tamaño medio de columna (bytes)
 FxP: Filas por páginas
 NPD: Número de páginas de datos
 TML: Tamaño medio del dato LOB (bytes)
 NPL: Número de páginas por columna LOB
 LRI: Longitud de registro de índice (bytes)
 FxN: Filas por páginas nodo de índice
 NPI: Número de páginas de índices
 TMCA: Tamaño medio de columna del índice agrupado

© JMA 2016. All rights reserved

Gestión automatizada

- Se puede automatizar la gestión del espacio asignado a una base de datos, fundamentalmente para sistemas no administrados.
- Crecimiento (solo a nivel de ficheros):
 - Se puede permitir que los ficheros crezcan automáticamente.
 - El crecimiento de ficheros provoca fragmentación en el disco duro con la consecuente reducción de rendimiento.
- Decrecimiento (solo a nivel de base de datos):
 - La opción de base de datos AUTO_SHRINK marca que cuando la base de datos tenga un 75% libre, reduzca todos los ficheros dejando un 25% de espacio libre a cada uno.
- Los automatismos degradan el rendimiento al entrar en periodos de máxima actividad de la base de datos.
- Los cambios de tamaño conllevan bloqueos a nivel de fichero y de base de datos.

© JMA 2016. All rights reserved

Registro de Transacciones

- Usados por el motor para grabar los cambios generados durante el uso de la Base de Datos
 - Registra tanto transacciones de usuarios como transacciones implícitas del sistema
 - El registro de transacciones es un archivo de registro circular.
- Utilizados para mantener la integridad de la Base de Datos y repararla en caso de fallo del sistema
- Toda base de datos debe tener al menos uno como mínimo.
- La extensión del archivo transaction log es .ldf
- Al igual que los ficheros de datos:
 - Se dimensionan en megas.
 - Pueden crecer automáticamente: porcentualmente o valor absoluto
 - Se puede impedir o limitar el crecimiento a un número máximo de megas (Máximo 12 TB)
- La falta de espacio puede impedir operar con la base de datos.

© JMA 2016. All rights reserved

Transacciones

- Una transacción es una secuencia de operaciones realizadas como una sola unidad lógica de trabajo.
- Una unidad lógica de trabajo debe exhibir cuatro propiedades, conocidas como propiedades ACID (atomicidad, coherencia, aislamiento y durabilidad), para ser calificada como transacción:
 - Atomicidad: Una transacción debe ser una unidad atómica de trabajo, tanto si se realizan todas sus modificaciones en los datos, como si no se realiza ninguna de ellas.
 - Coherencia: Cuando finaliza, una transacción debe dejar todos los datos en un estado coherente, es decir, se deben cumplir todas las reglas de integridad de todos los datos.
 - aislamiento: Las modificaciones realizadas por transacciones simultáneas se deben aislar de las modificaciones llevadas a cabo por otras transacciones simultáneas.
 - Durabilidad: Una vez concluida una transacción, sus efectos son permanentes en el sistema. Las modificaciones persisten aún en el caso de producirse un error del sistema.

© JMA 2016. All rights reserved

Especificar y exigir transacciones

- Los programadores de SQL son los responsables de iniciar y finalizar las transacciones en los puntos que exijan la coherencia lógica de los datos.
- El programador debe asegurar la atomicidad de las transacciones.
- SQL Server proporciona:
 - Servicios de bloqueo que preservan el aislamiento de la transacción.
 - Servicios de registro que aseguran la durabilidad de la transacción:
 - Aún en el caso de que falle el hardware del servidor, el sistema operativo o el propio SQL Server, SQL Server utiliza registros de transacciones, al reinicio, para deshacer automáticamente las transacciones incompletas en el momento en que se produjo el error en el sistema.
 - Características de administración de transacciones que exigen la atomicidad y coherencia de la transacción:
 - Una vez iniciada una transacción, debe concluirse correctamente o SQL Server deshacerá todas las modificaciones de datos realizadas desde que se inició la transacción.

© JMA 2016. All rights reserved

Tipos de transacciones

- Transacciones de confirmación automática
 - Éste es el modo predeterminado de SQL Server. Cada instrucción individual de Transact-SQL se confirma cuando termina. No tiene que especificar instrucciones para controlar las transacciones.
- Transacciones explícitas
 - Se inicia con BEGIN TRANSACTION, se confirma con COMMIT y se descarta con ROLLBACK.
- Transacciones implícitas
 - Establezca el modo de transacción implícita a través de una función de la API o la instrucción SET IMPLICIT_TRANSACTIONS ON de Transact-SQL. La siguiente instrucción inicia automáticamente una nueva transacción. Cuando se concluye la transacción, la instrucción de Transact-SQL siguiente inicia una nueva transacción.
- Transacciones distribuidas
 - Implican a varias instancias y se dividen en Fases de preparación y de confirmación.

© JMA 2016. All rights reserved

Información registrada

- Registro de la operación lógica (SQL).
 - Para confirmar la operación lógica, se vuelve a ejecutar.
 - Para deshacer la operación lógica, se ejecuta la operación lógica inversa.
- Registro de las imágenes (páginas) anterior y posterior.
 - Para confirmar la operación, se aplica la imagen posterior.
 - Para deshacer la operación, se aplica la imagen anterior.
- En el registro de transacciones se registran muchos tipos de operaciones, que incluyen:
 - El inicio y el final de cada transacción.
 - Todas las modificaciones de los datos (inserción, actualización y eliminación). Esto incluye las modificaciones de las tablas del sistema hechas por procedimientos almacenados del sistema o por instrucciones del lenguaje de definición de datos (DDL).
 - Las asignaciones o cancelaciones de asignación de extensiones.
 - La creación o eliminación de una tabla o un índice.

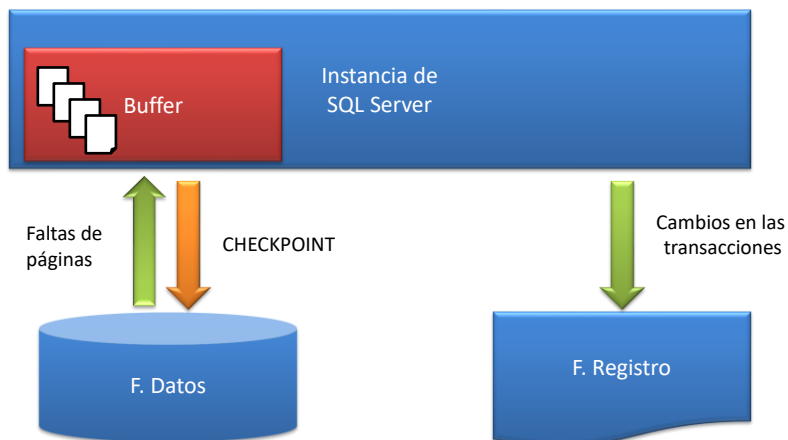
© JMA 2016. All rights reserved

Operativa

- Toda operación se registra en el registro de transacciones en el momento de producirse.
- Los datos se cargan en memoria, en caso de no estar ya, y modifican en memoria, pero no se graban en los ficheros de datos una vez terminadas las modificaciones.
- Los datos se graban periódicamente en los ficheros de datos en los denominados puntos de comprobación (CHECKPOINT).
 - Se marca el inicio del punto de comprobación en el registro de transacciones.
 - Se graban las páginas desfasadas en los ficheros de datos.
 - Se marca el final del punto de comprobación en el registro de transacciones.
- Se reserva el cauce de lectura para la carga de páginas de datos y el de escritura para la grabación en el fichero de transacciones.

© JMA 2016. All rights reserved

Lectura/Escritura



© JMA 2016. All rights reserved

Puntos de comprobación

- Se ejecutan de forma fija:
 - Cuando se ejecuta una instrucción CHECKPOINT. Se insertan puntos de comprobación en la base de datos actual de la conexión.
 - Cuando se utiliza ALTER DATABASE para cambiar una opción de base de datos. ALTER DATABASE inserta puntos de comprobación en la base de datos cuando se cambian las opciones de la base de datos.
 - Cuando se detiene una instancia de SQL Server por: Ejecutar una instrucción SHUTDOWN o se para el servicio.
- Se genera periódicamente de manera automática en cada base de datos para reducir el tiempo que tardaría la instancia en recuperar la base de datos.
 - Se genera cuando el número de registros alcanza el número que estima que puede procesar durante el tiempo especificado en la opción intervalo de recuperación.
 - En el modelo de recuperación sencilla, se genera puntos adicionales cuando el registro está ocupado en un 70 por ciento.

© JMA 2016. All rights reserved

Modelos de recuperación

- Sencillo:
 - Uso mínimo del registro de transacciones, se eliminan las transacciones una vez superado su punto de comprobación.
- Completo:
 - Uso extensivo del registro de transacciones, se eliminan las transacciones cuando se aseguran mediante una copia de seguridad.
- Carga masiva:
 - Modelo completo pero con registro mínimo de las operaciones de carga masiva. Truncan la secuencia de restauración.
- Cuando se produce la recuperación:
 - Al arrancar la instancia
 - Al restaurar la base de datos

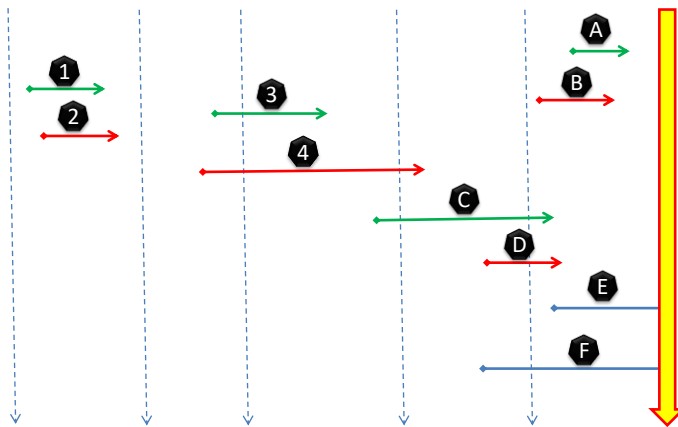
© JMA 2016. All rights reserved

Proceso de recuperación

- Relanza las transacciones iniciadas y confirmadas después del último punto de comprobación.
- Ignora las transacciones iniciadas y canceladas después del último punto de comprobación.
- En las transacciones iniciadas antes del último punto de comprobación y confirmadas después del último punto de comprobación, sobrescribe los ficheros de datos con la imagen posterior almacenada en el registro de transacciones.
- En las transacciones iniciadas antes del último punto de comprobación y descartadas después del último punto de comprobación, sobrescribe los ficheros de datos con la imagen anterior almacenada en el registro de transacciones.
- En las transacciones iniciadas antes del último punto de comprobación y no completadas después del último punto de comprobación, sobrescribe los ficheros de datos con la imagen anterior almacenada en el registro de transacciones.

© JMA 2016. All rights reserved

Recovery



© JMA 2016. All rights reserved

Bloqueos

- Balanceo: simultaneidad frente a coherencia
- Estrategias: optimista y pesimista
- Problemas de simultaneidad
- Grados de aislamiento
- Modos de interbloqueo
- Granularidad y jerarquías de bloqueo
- Interbloqueos

© JMA 2016. All rights reserved

Problemas de simultaneidad

- Actualizaciones perdidas:
 - Este problema surge cuando dos o más transacciones seleccionan la misma fila y, a continuación, la actualizan de acuerdo con el valor seleccionado originalmente.
 - Ninguna transacción es consciente de las otras transacciones.
 - La última actualización sobrescribe las actualizaciones realizadas por las otras transacciones y, en consecuencia, se pierden datos.
- Dependencia no confirmada (lectura no actualizada o desfasada):
 - Este problema se produce cuando una transacción selecciona una fila que está siendo actualizada por otra transacción.
 - La transacción que llega en segundo lugar lee datos que todavía no han sido confirmados y que la transacción que actualiza la fila puede modificar.

© JMA 2016. All rights reserved

Problemas de simultaneidad

- **Análisis incoherente (lectura irrepitable):**
 - Este problema se produce cuando una transacción obtiene acceso a la misma fila varias veces y en cada ocasión lee datos diferentes.
 - El análisis incoherente es similar a la dependencia no confirmada en tanto que una transacción está modificando los datos que está leyendo una segunda transacción.
 - Sin embargo, en el caso del análisis incoherente, los datos que lee la segunda transacción están confirmados por la transacción que realizó el cambio.
 - Además, el análisis incoherente comprende varias lecturas (dos o más) de la misma fila, y cada vez hay otra transacción que modifica la información; de ahí el término lectura irrepitable.

© JMA 2016. All rights reserved

Problemas de simultaneidad

- **Lecturas fantasmas o irrepitibles:**
 - Este problema se produce cuando se realiza una acción de insertar o eliminar en una fila y ésta pertenece a un intervalo de filas que está leyendo una transacción.
 - La primera lectura que hizo la transacción en el intervalo de filas muestra una fila que ya no existe en la segunda lectura o en lecturas sucesivas porque otra transacción la ha eliminado.
 - De forma similar, como consecuencia de una inserción realizada por otra transacción, la segunda lectura o las lecturas sucesivas de la transacción muestran una fila que no existía en la primera lectura.

© JMA 2016. All rights reserved

Niveles de aislamiento

- **READ UNCOMMITTED**

Implementa las lecturas no confirmadas o el bloqueo de nivel de aislamiento 0, lo que significa que no hay bloqueos compartidos y que los bloqueos exclusivos no están garantizados. Cuando se establece esta opción, es posible leer datos no confirmados, los valores pueden cambiar y pueden aparecer y desaparecer filas en el conjunto de datos antes del final de la transacción. Esta opción tiene el mismo efecto que establecer NOLOCK en todas las tablas y en todas las instrucciones SELECT de la transacción. Se trata del menos restrictivo de los cuatro niveles de aislamiento.

- **READ COMMITTED (predeterminada)**

Especifica que se mantengan los bloqueos compartidos mientras se leen datos para evitar lecturas no actualizadas, pero se pueden modificar los datos antes del final de la transacción, lo que provoca lecturas no repetibles o datos fantasmas. Esta opción es la predeterminada en SQL Server.

© JMA 2016. All rights reserved

Niveles de aislamiento

- **REPEATABLE READ**

Se establecen bloqueos para todos los datos utilizados en la consulta, lo que impide que otros usuarios los actualicen, aunque es posible insertar nuevas filas fantasmas en los datos que otro usuario establezca, de modo que se incluyan en lecturas posteriores de la misma transacción. Como la simultaneidad es inferior que el nivel de aislamiento predeterminado, sólo se debe usar esta opción cuando sea necesario.

- **SERIALIZABLE**

Se establece un bloqueo de intervalo en el conjunto de datos, lo que impide que otros usuarios actualicen o inserten filas en el conjunto de datos hasta que finalice la transacción. Es el más restrictivo de los cuatro niveles de aislamiento. Al ser menor la simultaneidad, sólo se debe utilizar esta opción cuando sea necesario. Esta opción tiene el mismo efecto que establecer HOLDLOCK en todas las tablas y en todas las instrucciones SELECT de la transacción.

© JMA 2016. All rights reserved

Niveles de aislamiento

- **SNAPSHOT**

Especifica que los datos leídos por cualquier instrucción de una transacción sean la versión coherente, desde el punto de vista transaccional, de los datos existentes al comienzo de la transacción.

La transacción únicamente puede reconocer las modificaciones de datos confirmadas antes del comienzo de la misma.

Las instrucciones que se ejecuten en la transacción actual no verán las modificaciones de datos efectuadas por otras transacciones después del inicio de la transacción actual.

El efecto es el mismo que se obtendría si las instrucciones de una transacción obtuviesen una instantánea de los datos confirmados tal como se encontraban al comienzo de la transacción.

© JMA 2016. All rights reserved

Niveles de aislamiento

Nivel de aislamiento	Actualización perdidas	Lectura no actualizada	Lectura no repetible	Lectura Fantasma
Lectura no confirmada (READ UNCOMMITTED)	Sí	Sí	Sí	Sí
Lectura confirmadas (READ COMMITTED)	Sí	No	Sí	Sí
Lectura repetible (REPEATABLE READ)	No	No	No	Sí
Instantánea (SNAPSHOT)	No	No	No	No
Serializable (SERIALIZABLE)	No	No	No	No

© JMA 2016. All rights reserved

Modos de bloqueo

- Los modos de bloqueo determinan el modo en que las transacciones simultáneas pueden tener acceso a los recursos.
 - **Compartido (S)**: Se utiliza para operaciones de lectura que no cambian ni actualizan datos, como la instrucción SELECT.
 - **Actualizar (U)**: Se utiliza en recursos que se pueden actualizar. Evita una forma común de interbloqueo que se produce cuando varias sesiones leen, bloquean y actualizan recursos.
 - **Exclusivo (X)**: Se utiliza para operaciones de modificación de datos, como INSERT, UPDATE o DELETE. Garantiza que no puedan realizarse varias actualizaciones simultáneamente en el mismo recurso.
 - **Intención**: Se utiliza para establecer una jerarquía de bloqueos. Los tipos de bloqueo de intención son: **intención compartido (IS)**, **intención exclusivo (IX)** y **compartido con intención exclusivo (SIX)**.
 - **Esquema**: Se utiliza cuando se ejecuta una operación que depende del esquema de una tabla. Hay dos tipos de bloqueo de esquema: **modificación del esquema (Sch-M)** y **modificación de estabilidad (Sch-S)**.
 - **Actualización masiva (BU)**: Se utiliza cuando se copian datos de forma masiva en una tabla y se especifica la sugerencia TABLOCK.
 - **Intervalo de claves**: Protege el intervalo de filas que lee una consulta cuando se utiliza el nivel de aislamiento de transacciones serializables. Garantiza que otras transacciones no puedan insertar filas que podrían incluirse como respuesta de las consultas de la transacción serializable si las consultas se volvieron a ejecutar.

© JMA 2016. All rights reserved

Granularidad y jerarquías de bloqueo

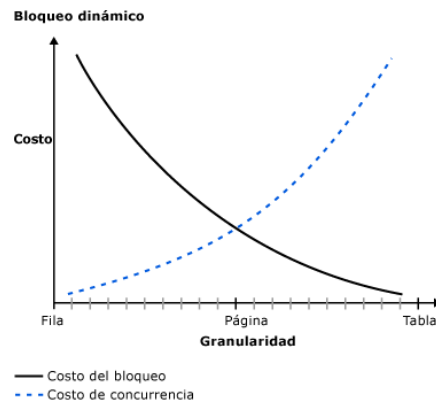
La extensión de bloqueo es el proceso de convertir muchos bloqueos concretos en menos bloqueos más generales, lo que reduce la sobrecarga del sistema al tiempo que aumenta la probabilidad de contención de simultaneidad.

- **RID**: Identificador de fila que se utiliza para bloquear una sola fila de un montón.
- **KEY**: Bloqueo de fila dentro de un índice que se utiliza para proteger intervalos de claves en transacciones serializables.
- **PAGE**: Página de 8 kilobytes (KB) de una base de datos, como páginas de datos o de índices.
- **EXTENT**: Grupo contiguo de ocho páginas, como páginas de datos o de índices.
- **HoBT**: Montón o árbol b. Bloqueo que protege un árbol B (índice) o las páginas de datos del montón en una tabla que no posee un índice agrupado.
- **TABLE**: Tabla completa, con todos los datos e índices.
- **FILE**: Archivos de la base de datos.
- **APPLICATION**: Recurso especificado por la aplicación.
- **METADATA**: Bloqueos de metadatos.
- **ALLOCATION_UNIT**: Unidad de asignación.
- **DATABASE**: Base de datos completa.

© JMA 2016. All rights reserved

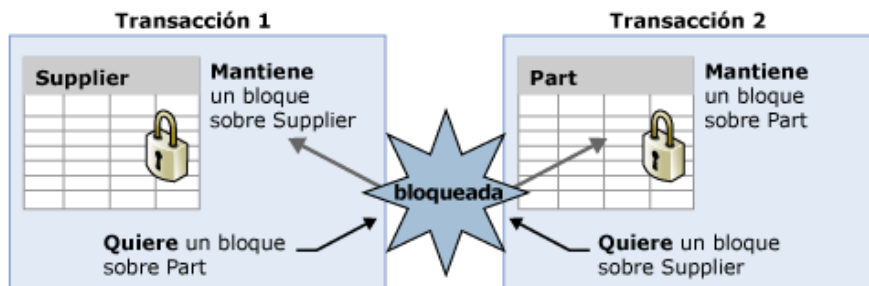
Bloqueo dinámico

- La utilización de bloqueos de bajo nivel, como los de fila, aumenta la simultaneidad reduciendo la probabilidad de que dos transacciones soliciten bloqueos de los mismos datos al mismo tiempo. También aumenta el número de bloqueos y los recursos necesarios para administrarlos.
- Los bloqueos de alto nivel de tabla o página producen una sobrecarga menor, pero a costa de reducir la simultaneidad.



© JMA 2016. All rights reserved

Interbloqueos



© JMA 2016. All rights reserved

Interbloqueos

- Un interbloqueo se produce cuando hay una dependencia cíclica entre dos o más subprocesos para algún conjunto de resultados.
- Detección de interbloqueos: un subproceso de supervisión de bloqueos que periódicamente inicia una búsqueda por todas las tareas de una instancia
 - El intervalo predeterminado es de 5 segundos
 - Baja hasta sólo 100 milisegundos
 - Segundo intento, solución de interbloqueo
- Solución de interbloqueo
 - Prevención mediante el Motor de consultas y plan de ejecución
 - Expiración por tiempos de espera
 - sesión, transacción, SET LOCK_TIMEOUT, ...
 - Finalización de subproceso (error 1205)
 - Menor nivel (SET DEADLOCK_PRIORITY)
 - reversión menos costosa (deshacer)
 - a igualdad: aleatoriamente

© JMA 2016. All rights reserved

Interbloqueos

- Para evitar los interbloqueos:
 1. No abra una transacción mientras examina los datos si es posible.
 2. Las transacciones deben ser atómicas.
 3. Haga la transacción lo más corta posible.
 4. La transacción no debería depender de interfaz de usuario.
 5. Haga un uso inteligente de los niveles más bajos de aislamiento de las transacciones.
 6. Maneje siempre que sea posible los conjuntos de datos en el mismo orden

© JMA 2016. All rights reserved

Información de bloqueo

- Los bloqueos pueden reducir el rendimiento de la instancia.
- SQL Server ofrece varias formas de obtener información acerca de la actividad de bloqueo actual en una instancia:
 - Se puede realizar una consulta de la vista de administración dinámica **sys.dm_tran_locks** para obtener información acerca del estado actual de bloqueo de una instancia del Database Engine (Motor de base de datos).
 - Mediante el SQL Server Profiler, se puede especificar la categoría de eventos de los bloqueos para capturar información acerca de los bloqueos de eventos en una traza.
 - En el Monitor de sistema, se pueden especificar contadores del objeto de bloqueos para supervisar el nivel de bloqueo de una instancia del Motor de base de datos.

© JMA 2016. All rights reserved

Ajustes de instancia

- Management Studio - Propiedades:
 - Memoria (máxima y mínima)
 - Procesador (afinidad)
 - Seguridad (modo y auditorias)
 - Conexiones (número y valores por defecto)
 - Configuración de base de datos (por defecto)
 - Opciones Avanzadas (paralelismo, red, ...)
- Transact SQL:
 - `sp_configure 'show advanced options', 1;`
 - `RECONFIGURE;`
- Especificaciones de producto para SQL Server

© JMA 2016. All rights reserved

Buffer Pool Extension

- A partir de SQL Server 2014 (12.x), en versiones Estándar o superior, la extensión del grupo de búferes proporciona una perfecta integración de una extensión de la memoria de acceso aleatorio no volátil (es decir, una unidad de estado sólido) con el grupo de búferes del Motor de base de datos para mejorar considerablemente el rendimiento de E/S.
- El almacenamiento SSD se utiliza como extensión del subsistema de memoria en lugar del subsistema de almacenamiento en disco.
- Establecer la opción de extensión del grupo de búferes:

```
ALTER SERVER CONFIGURATION
SET BUFFER POOL EXTENSION ON
(FILENAME = 'F:\SSDCACHE\Example.BPE', SIZE = 50 GB);
```

© JMA 2016. All rights reserved

Propiedades de la base de datos

- Management Studio :
 - Nombre, Propietario, Modelo de recuperación, Nivel de compatibilidad y Tipo de contención.
 - Grupos de archivos (Filas, FILESTREAM, Datos con optimización para memoria) y Archivos.
 - Automatismos, contención, estado, recuperación y valores por defecto.
- Transact SQL:
 - CREATE DATABASE y ALTER DATABASE
 - ALTER DATABASE SET
 - Función: DATABASEPROPERTYEX

© JMA 2016. All rights reserved

Automatismos

- Automatismos
 - Crear estadísticas automáticamente: Se crean automáticamente las estadísticas de optimización que faltan.
 - Creación automática de estadísticas incrementales: Si se desea utilizar la opción incremental cuando se crean estadísticas por partición.
 - Actualizar estadísticas automáticamente: Se actualizan automáticamente las estadísticas de optimización no actualizadas.
 - Actualizar estadísticas automática y asincrónicamente: Las consultas que inician una actualización automática de estadísticas obsoletas no esperan a que las estadísticas se actualicen antes de la compilación. Las consultas posteriores usan las estadísticas actualizadas si están disponibles.
 - Cerrar automáticamente: La base de datos se cierra y se liberan sus recursos después de que salga el último usuario.
 - Reducir automáticamente: Cuando la base de datos tenga un 75% libre, se reducen todos los ficheros dejando un 25% de espacio libre a cada uno.
- Motor de consultas:
 - Estimación de cardinalidad heredada, Examen de parámetros, Correcciones del optimizador de consultas, Max DOP (Paralelismo), Parametrización

© JMA 2016. All rights reserved

Optimizar la configuración de base de datos

- Colocar los datos y los archivos de registro en unidades independientes evita cuellos de botellas y balancea la carga entre los diferentes dispositivos.
- La compresión de datos puede contribuir a mejorar el rendimiento de las cargas de trabajo que hacen un uso intensivo de las operaciones de E/S porque los datos se almacenan en menos páginas y las consultas deben leer menos páginas del disco. No obstante, se requieren recursos de CPU adicionales en el servidor de base de datos para comprimir y descomprimir los datos, mientras los datos se intercambian con la aplicación.
- Las nuevas versiones suelen incorporar importantes mejoras de rendimiento, el nivel de compatibilidad de una base de datos puede limitar el acceso a dichas mejoras

© JMA 2016. All rights reserved

RAID

- RAID (matriz redundante de discos independientes) es un sistema de discos, denominado matriz, que contiene varias unidades de disco para proporcionar mayor rendimiento, confiabilidad y capacidad de almacenamiento a un costo menor. Las matrices con tolerancia a errores están divididas en seis niveles RAID: del 0 al 5. Cada nivel utiliza un algoritmo diferente para implementar la tolerancia a errores. Con SQL Server se utilizan normalmente los niveles RAID 0, 1 y 5.
- Una matriz de discos de hardware mejora el rendimiento de E/S debido a que las funciones de E/S, tales como la creación de bandas y reflejos, se controlan de forma eficaz mediante el firmware. Por el contrario, una RAID basada en el sistema operativo representa un menor costo pero consume ciclos del procesador. Cuando el costo representa un problema y se necesita redundancia y alto rendimiento, los volúmenes RAID-5 son una buena solución.

© JMA 2016. All rights reserved

RAID

- La creación de bandas de datos (RAID 0) es la configuración RAID con mejor rendimiento, pero si un disco tiene errores, se deja de tener acceso a todos los datos del conjunto de bandas. Una técnica común de instalación de los sistemas de administración de bases de datos relacionales consiste en configurar la base de datos en una unidad RAID 0 y, a continuación, colocar el registro de transacciones en una unidad reflejada (RAID 1). Puede obtener el mejor rendimiento de E/S de disco para la base de datos y mantener la posibilidad de recuperar los datos a través de un registro de transacciones reflejado, siempre y cuando realice copias de seguridad de la base de datos con regularidad.
- Si es necesario recuperar rápidamente los datos, es posible reflejar el registro de transacciones y colocar la base de datos en un disco RAID 5, aunque con un rendimiento menor. RAID 5 proporciona redundancia de todos los datos de la matriz, permitiendo el cambio de unidades sin que se produzca tiempo de inactividad del sistema.

© JMA 2016. All rights reserved

Dimensionado de la B.D. tempdb

- La base de datos del sistema **tempdb** es un recurso global disponible para todos los usuarios de la instancia de SQL Server, utilizada para:
 - Objetos temporales de usuario que se hayan creado explícitamente, como: tablas y procedimientos temporales, variables de tabla o cursores.
 - Objetos internos creados por Motor de base de datos de SQL Server, como: tablas de trabajo para almacenar resultados intermedios para colas u ordenación.
 - Versiones de fila generadas por las transacciones de modificación de datos en una base de datos que utiliza transacciones de lectura confirmada que usan transacciones de aislamiento de versiones de fila o de aislamiento de instantáneas.
 - Versiones de fila que se hayan generado mediante transacciones de modificación de datos para características, como: operaciones de índice en línea, conjuntos de resultados activos múltiples (MARS) y desencadenadores AFTER.

© JMA 2016. All rights reserved

Consideraciones sobre tempdb

- tempdb se vuelve a crear cada vez que se inicia SQL Server, de forma que el sistema siempre se inicia con una copia limpia de la base de datos.
- El tamaño de tempdb puede afectar al rendimiento de un sistema, si el tamaño es demasiado pequeño, el procesamiento del sistema podría estar demasiado ocupado con el crecimiento automático de la base de datos y, por tanto, no podría satisfacer las necesidades de carga de trabajo cada vez que se inicia SQL Server.
- La ausencia de espacio en tempdb puede acarrear el bloqueo de la instancia, los ficheros deben crecer automáticamente.

© JMA 2016. All rights reserved

Operaciones no disponibles sobre tempdb

- Agregar grupos de archivos, quitar el grupo de archivos principal, el archivo de datos principal o el archivo de registro.
- Activar reducir o cerrar automáticamente.
- Cambiar el nombre de la base de datos o del grupo de archivos principal.
- Cambiar la intercalación o el propietario de la base de datos (es propiedad de **sa**).
- Establecer la base de datos en OFFLINE o READ_ONLY.
- Crear una instantánea, eliminar la base de datos o eliminar el usuario **guest**.
- Realizar una copia de seguridad o restaurar la base de datos.
- Habilitar el mecanismo de captura de cambios en los datos.
- Participar en el reflejo de la base de datos.
- Ejecutar DBCC CHECKALLOC o DBCC CHECKCATALOG.

© JMA 2016. All rights reserved

Optimizar el rendimiento de tempdb

- Establecer el tamaño de los archivos en un valor lo suficientemente alto para contener la carga de trabajo habitual del entorno, evitando que se expanda con demasiada frecuencia.
- Debe establecerse que crezca automáticamente para las situaciones no previstas, con un incremento suficiente que evite hacerlo con demasiada frecuencia.
- La división de tempdb en varios archivos de datos del mismo tamaño proporciona un alto grado de eficiencia paralela en las operaciones que usan tempdb.
- Colocar la base de datos tempdb en un subsistema de E/S rápido. Cree bandas en disco si hay muchos discos conectados directamente.
- Colocar la base de datos tempdb en discos diferentes de los que usan las bases de datos de usuario.

© JMA 2016. All rights reserved

AUTOMATIZACIÓN DE TAREAS ADMINISTRATIVAS

© JMA 2009. All rights reserved

Tareas habituales del administrador

- **Mantenimiento preventivo**
 - Operaciones rutinarias:
 - Copias de seguridad
 - Importación y exportación de datos
 - Replicaciones
 - Mantenimiento de datos (paso a históricos, consolidaciones, ...)
 - Mantenimiento de estructuras (reindexación, liberación de espacio, ...)
 - Supervisión del sistema y acciones correctivas:
 - Espacio en la base de datos y del registro
 - Rendimientos
- **Mantenimiento reactivo**
 - Resolución de incidencias

© JMA 2009. All rights reserved

Agente de SQL Server

- SQL Server Agent es el servicio responsable de automatizar las tareas administrativas de SQL Server
- Se debe configurar el servicio SQL Server Agent:
 - para inicio automático
 - para reiniciarlos de forma automática si se detienen de forma inesperada
 - que la cuenta de inicio de sesión esté asignada a la función sysadmin
- La cuenta de inicio de sesión DEBERÍA:
 - Usar el modo de autenticación de Windows
 - Utilizar una cuenta de usuario de dominio de Windows
 - Asignar esta cuenta al grupo local Administradores

© JMA 2009. All rights reserved

Base de Datos msdb

- La Base de Datos msdb se encarga de almacenar la información referente al servicio SQL Server Agent, y a la planificación de trabajos.
- Entre la información que almacena se encuentra la información relativa a:
 - Trabajos
 - Alertas
 - Operadores
 - Programaciones
 - Planes de Mantenimiento
 - Historial
- Para utilizar el Agente SQL Server, los usuarios deben ser miembros de una o más de las siguientes funciones fijas de base de datos:
 - SQLAgentUserRole
 - SQLAgentReaderRole
 - SQLAgentOperatorRole

© JMA 2009. All rights reserved

Correo electrónico de base de datos

- El Correo electrónico de base de datos permite a las aplicaciones de base de datos enviar mensajes de correo electrónico a los usuarios.
- Los mensajes enviados pueden incluir resultados de consultas y archivos de cualquier recurso de la red.
- Utiliza el protocolo estándar SMTP
- Está organizado en Perfiles: Públicos y Privados
- Cada perfil puede incluir varias cuentas de conmutación por error.
- Se encuentra desactivado de forma predeterminada.
- La base de datos msdb es la base de datos host del Correo electrónico de base de datos.
- Para enviar correo debe pertenecer a la función de BD DatabaseMailUserRole en msdb.
- Características
 - Seguridad de los perfiles, Regulador del tamaño y Extensiones de archivo prohibidas.
 - Configuración integrada, Registro, Auditoría y Compatibilidad con HTML
 - Procedimiento sp_send_dbmail, vistas y procedimientos sysmail_

© JMA 2009. All rights reserved

Operadores

- Un operador es la personas responsable del mantenimiento, total o parcial, de una o varias bases de datos o instancias de SQL Server
- Métodos de envío de Notificaciones:
 - Correo electrónico
 - Localizador (por correo electrónico) (SMS, "pager" o busca)
 - Envío de red (net send)
- Se puede fijar los horarios de uso del localizador (Programación del localizador)
- Un operador a prueba de errores recibe la notificación cuando:
 - no se pueden enviar mensajes al localizador de ninguno de los operadores responsables de la alerta
 - no puede tener acceso a las tablas del sistema en la base de datos msdb

© JMA 2009. All rights reserved

Trabajos

- Un trabajo es un conjunto de pasos secuenciales
- La definición de un paso puede usar:
 - Programas ejecutables y comandos del sistema operativo.
 - Instrucciones Transact-SQL, incluidos los procedimientos almacenados y los procedimientos almacenados extendidos.
 - Secuencias de comandos Microsoft ActiveX
 - Tareas de réplica.
 - Tareas de Analysis Services.
 - Paquetes de Integration Services.
- El flujo del trabajo se controla indicando el siguiente paso o deteniéndolo cuando se ejecute correctamente o cause un error, así como los reintentos después de dar error.
- El propietario del trabajo debe tener los permisos adecuados para ejecutar todos los pasos del trabajo.
- Se les pueden asignar Operadores a los que se notifican el resultado de las ejecuciones.
- Los trabajos se pueden organizar en categorías.

© JMA 2009. All rights reserved

Programaciones

- Un trabajo se puede ejecutar manualmente, según su programación, en respuesta a una alerta o dentro de un plan de mantenimiento.
- Se puede programar trabajos para que se inicien automáticamente:
 - Al iniciarse SQL Server Agent.
 - A una hora específica (sólo una vez).
 - De forma periódica (diaria, semanal o mensualmente).
 - Cuando la CPU esté inactiva (configurable a nivel de servicio).
- Cada trabajo acepta varias programaciones (programación múltiple).
- Los trabajos se pueden desactivar, en cuyo caso no se iniciarán según la programación

© JMA 2009. All rights reserved

Historial de trabajos

- Los trabajos son automáticos pero no deben quedar desatendidos
- Se recomienda el uso de listas de control (CHECKLIST) para su supervisión.
- Revisión del historial de un trabajo individual:
 - Resultado de cada paso del trabajo: correcto o erróneo
 - Duración de ejecución
 - Errores y mensajes
- Se puede supervisar la actividad actual mediante el Monitor de actividad de trabajo.
- Configuración del tamaño del historial de trabajos:
 - Conservar información acerca de cada trabajo
 - Sobrescribir historial cuando se alcance el tamaño máximo

© JMA 2009. All rights reserved

Alertas

- Una alerta indica que ha ocurrido un suceso. Una alerta responde a un tipo de evento específico, pueden ser de los siguientes tipos :
 - Eventos de SQL Server
 - un error específico (Número de error)
 - un error de la gravedad específica (Nivel de gravedad)
 - en una base de datos específica
 - con una la cadena de texto especifica en el mensaje del suceso
 - Condiciones de rendimiento de SQL Server, dependiente de un objeto, contador e instancia.
 - Eventos de WMI, debe definir el Espacio de nombres y la Consulta
- En respuesta a la alerta se puede:
 - Ejecutar un trabajo
 - Mandar una notificación a uno o varios operadores

© JMA 2009. All rights reserved

Planes de mantenimiento

- Permiten configurar y agrupar las principales tareas de mantenimiento necesarias para garantizar el buen funcionamiento de la base de datos.
- El plan puede contar con una o varias de las siguientes tareas:
 - Reorganizar los datos de las páginas de datos y de índices mediante una nueva generación de los índices con un nuevo factor de relleno
 - Comprimir archivos de datos mediante la eliminación de las páginas de la base de datos que estén vacías
 - Actualizar las estadísticas de los índices para el optimizador de consultas.
 - Realizar comprobaciones de coherencia interna de los datos y de las páginas de datos de la base de datos
 - Realizar copias de seguridad de la base de datos y de los archivos de registro de transacciones
 - Configurar el trasvase de registros (sincronizar bases de datos)

© JMA 2009. All rights reserved

Herramientas del administrador

- Vistas del sistema
- Procedimientos y funciones del sistema
- Informes del SSMS
- Utilidades del sistema operativo
- Directivas
- DBCC: Data Base Command Console

© JMA 2009. All rights reserved

Administración basada en directivas

- La administración basada en directivas es un sistema para administrar una o varias instancias de SQL Server similar a la existente en el Windows.
- Las Facetas son un conjunto de propiedades lógicas que modelan el comportamiento o las características de ciertos aspectos administrados. Las facetas vienen predefinidos.
- Una Condición es una expresión booleana que especifica un conjunto de estados permitidos para una o mas facetas.
- Una Directiva de cuando, como y sobre que se aplica una condición.
- Modos de evaluación
 - A petición. Este modo evalúa la directiva cuando lo especifica el usuario directamente.
 - Al cambiar: impedir. Este modo automatizado utiliza desencadenadores DDL para evitar las infracciones de las directivas.
 - Al cambiar: sólo registrar. Este modo automatizado utiliza la notificación de eventos para evaluar una directiva cuando se realiza un cambio relevante.
 - Al programar. Este modo automatizado utiliza un trabajo del Agente SQL Server para evaluar una directiva periódicamente.

© JMA 2009. All rights reserved

DBCC: Instrucciones de mantenimiento

Comando	Descripción
DBREINDEX	Regenera uno o varios índices de una tabla de la base de datos especificada.
DBREPAIR	Quita una base de datos dañada (obsoleta). utilizar DROP DATABASE
INDEXDEFRAG	Desfragmenta los índices agrupados y secundarios de la tabla o la vista especificada.
SHRINKDATABASE	Reduce el tamaño de los archivos de datos de la base de datos especificada.
SHRINKFILE	Reduce el tamaño del archivo de datos o de registro especificado de la base de datos relacionada.
CLEANTABLE	Recupera espacio correspondiente a columnas de longitud variable y a columnas de texto que se han quitado.
UPDATEUSAGE	Informa y corrige las imprecisiones de la tabla sysindexes, que pueden provocar que el procedimiento almacenado del sistema sp_spaceused informe incorrectamente del uso del espacio.

© JMA 2009. All rights reserved

DBCC: Instrucciones de validación

Comando	Descripción
CHECKALLOC	Comprueba la coherencia de las estructuras de asignación de espacio en disco para una base de datos determinada.
CHECKCATALOG	Comprueba la coherencia de las tablas del sistema y de éstas entre sí, en la base de datos especificada.
CHECKCONSTRAINTS	Comprueba la integridad de una restricción especificada o de todas las restricciones de una tabla determinada.
CHECKDB	Comprueba la asignación y la integridad estructural de todos los objetos de la base de datos especificada.
CHECKFILEGROUP	Comprueba la asignación y la integridad estructural de todas las tablas (de la base de datos actual) del grupo de archivos especificado.
CHECKIDENT	Comprueba el valor de identidad actual de la tabla especificada y, si fuera necesario, corrige el valor de identidad.
CHECKTABLE	Comprueba la integridad de las páginas de datos, de índices y de columnas text, ntext e image de la tabla especificada o la vista indexada.
NEWALLOC	Idéntica a DBCC CHECKALLOC

© JMA 2009. All rights reserved

MODELO LÓGICO

© JMA 2009. All rights reserved

Vistas

- Una vista es una tabla virtual cuyo contenido está definido por una consulta.
 - Combinar columnas de varias tablas de forma que parezcan una sola tabla.
 - Agregar información agrupada o calculada en lugar de presentar los detalles.
 - Unir información de varios resultados
 - Restringir el acceso del usuario a filas concretas de una tabla.
 - Restringir el acceso del usuario a columnas específicas.
- Tipos:
 - Vistas estándar
 - Vistas indexadas o materializada
 - Vistas con particiones o distribuidas
 - Vistas parametrizadas (funciones de tipo tabla)

© JMA 2009. All rights reserved

Vistas

- Se puede modificar los datos mediante una vista.
 - Son actualizables (UPDATE, DELETE o INSERT) mientras la modificación afecte sólo a una de las tablas base.
 - Las columnas a modificar en la vista deben hacer referencia directa a los datos subyacentes de las columnas de la tabla (no puede ser resultado de una función de agregado o cálculo).
 - Deben aparecer todas las columnas requeridas que no dispongan de valor predeterminado.
 - Las columnas que se modifican no pueden verse afectadas por cláusulas GROUP BY, HAVING o DISTINCT.
 - Permite el uso de desencadenadores INSTEAD OF
 - La cláusula WITH CHECK OPTION exige que todas las instrucciones de modificación de datos ejecutadas en la vista se ajusten a los criterios especificados en la instrucción SELECT que define la vista.

© JMA 2009. All rights reserved

Funciones definidas por el usuario

- Las funciones son subrutinas formadas por una o varias instrucciones Transact-SQL o CLR que se pueden utilizar para encapsular un código con el fin de utilizarlo de nuevo posteriormente.
- Las funciones definidas por el usuario pueden tener parámetros de entrada y devuelven un único valor (dato o tabla interna)
- SQL Server admite cuatro tipos de funciones definidas por el usuario:
 - Funciones escalares
 - Funciones de valores de tabla en línea
 - Funciones de valores de tabla de múltiples instrucciones
 - Funciones de agregado (solo CLR)

© JMA 2009. All rights reserved

Desencadenadores DML

- Un desencadenador es un tipo especial de procedimiento almacenado que entra en vigor cuando se modifican datos en una tabla especificada utilizando una o más operaciones de modificación de datos: UPDATE (actualización), INSERT (inserción) o DELETE (eliminación).
- Los desencadenadores pueden exigir restricciones más complejas que las definidas con restricciones CHECK y permiten implementar las reglas de negocio.
- Pueden realizar cambios en cascada a través de tablas relacionadas de la base de datos.
- Los desencadenadores son automáticos: se activan inmediatamente después de que se efectúen modificaciones en los datos de la tabla, como una entrada manual o una acción de la aplicación.
 - Se pueden habilitar y deshabilitar.
 - Se pueden definir varios por tabla, cada uno para una o varias operaciones.
 - Se puede fijar cuál es el primero y el último en ejecutarse: `sp_settriggerorder 'First|None|Last'`

© JMA 2009. All rights reserved

Optimización con vistas y funciones

- Externamente, las aplicaciones no distinguen entre tablas y vistas.
- Las vistas, al ser actualizables, complementadas por desencadenadores permiten la manipulación del modelo lógico sustituyendo tablas por vistas:
 - Particionamiento horizontal: consultas UNION
 - Particionamiento vertical: consultas JOIN
- Las funciones de valores de tabla pueden utilizar parámetros en sus consultas por lo permiten la implementación de “vistas parametrizadas”, que permiten:
 - las consultas preparadas
 - la propagación de parámetros desde las consultas exteriores a las interiores

© JMA 2009. All rights reserved

Índices

- Compuestos por una o varias columnas ordenadas ascendente o descendentemente
- Guardan clave indexación, clave en índice agrupado o marcador fila real y columnas incluidas.
- Se pueden crear sobre Tablas, Vistas y columnas calculadas.
- Los dos tipos:
 - Agrupado
 - No agrupado
- Un índice único garantiza que la columna indexada no contiene valores no nulos duplicados.
- Índices y Restricciones:
 - Las restricciones PRIMARY KEY crean índices agrupados de forma automática de tipo UNIQUE.
 - Las restricciones UNIQUE crean índices UNIQUE de forma automática.
 - Las restricciones FOREIGN KEY no crean índices, pero suelen ser convenientes.
- Los no agrupados permiten columnas INCLUIDAS

© JMA 2009. All rights reserved

Índices

- El factor de relleno determina la densidad de ocupación de páginas por parte del índice.
- El número de índices afecta al rendimiento y ocupación de la base de datos.
- El Asistente para optimización de índices suministra recomendaciones en función a la monitorización del sistema.
- Limitaciones
 - Sólo el propietario de la tabla puede crear índices en la misma tabla.
 - Un índice agrupado por tabla.
 - 249 índices como máximo (incluidos los creados por restricciones PRIMARY KEY o UNIQUE).
 - 16 columnas como máximo.
 - El tamaño máximo de la clave de indexación es 900 bytes.
- Índices especiales:
 - Texto completo, XML y Espaciales

© JMA 2009. All rights reserved

Índices

- Para crear un índice mediante el Explorador de objetos:
 - En el Explorador de objetos, expanda la tabla en la que desea crear un índice.
 - Haga clic con el botón derecho en la carpeta Índices, seleccione Nuevo índice ...
- Para crear un índice mediante el Diseñador de tablas:
 - En el Explorador de objetos, expanda la base de datos en la que desea crear o modificar una tabla con un índice.
 - Haga clic con el botón derecho en la carpeta Tablas y, luego, haga clic en Nueva tabla... o en Diseño.
 - En el menú Diseñador de tablas , haga clic en Índices o claves.
- Para crear un índice mediante Transact-SQL:


```
CREATE CLUSTERED INDEX IX_TestTable_TestCol1
ON dbo.TestTable (TestCol1);
```

© JMA 2009. All rights reserved

Estadísticas

- SQL Server (2005...) permite crear información estadística acerca de la distribución de valores en una columna.
- El optimizador de consultas utiliza esta información estadística para determinar el plan de consultas óptimo realizando una estimación del costo de usar un índice para evaluar la consulta.
- Se almacenan con la misma estructura que los índices, sustituyendo el localizador por un contador.
- Se pueden crear automáticamente o manualmente. Es necesario actualizarlas periódicamente:
`exec sp_updatestats`
- Se pueden consultar:
`DBCC SHOW_STATISTICS (N'Tabla', N'Estadística') WITH HISTOGRAM;`

© JMA 2009. All rights reserved

Estadísticas

- Para crear una estadística mediante el Explorador de objetos:
 - En el Explorador de objetos, expanda la tabla en la que desea crear la estadística.
 - Haga clic con el botón derecho en la carpeta Estadísticas, seleccione Nueva estadística ...
- Para crear una estadística mediante Transact-SQL:
`CREATE STATISTICS ContactMail1
 ON Person.Person (EmailPromotion);`

© JMA 2009. All rights reserved

OPTIMIZACIÓN DE SENTENCIAS

© JMA 2016. All rights reserved

Introducción

- El SQL, al ser un lenguaje de alto nivel, requiere ser compilado antes de poder ejecutarse.
 - El resultado de la compilación se denomina plan de ejecución y se almacenan en una parte de la memoria de SQL Server denominada caché del plan.
 - Fases de la ejecución:
 1. Búsqueda en la cache
 2. Validación sintáctica
 3. Conversión y homogeneización (parametrización)
 4. Optimización
 5. Almacenamiento en cache
 6. Establecer contexto
 7. Ejecutar plan
-

© JMA 2016. All rights reserved

Modos de ejecución

- La ejecución del **modo de fila** es un método de procesamiento de consultas que se usa con tablas RDMBS tradicionales, donde los datos se almacenan en formato de fila. Cuando se ejecuta una consulta y accede a los datos de tablas de almacén de filas, los operadores del árbol de ejecución y los operadores secundarios leen todas las filas necesarias, en todas las columnas especificadas en el esquema de tabla. De cada fila que se lee, SQL Server recupera las columnas que son necesarias para el conjunto de resultados, como se hace referencia mediante una instrucción SELECT, un predicado JOIN o un predicado de filtro.
- La ejecución del **modo por lotes** es un método de procesamiento de consultas en el que las consultas procesan varias filas a la vez. Cada columna dentro de un lote se almacena como un vector en un área de memoria independiente, por lo que el procesamiento del modo por lotes se basa en vectores. En el procesamiento del modo por lotes también se usan algoritmos que se optimizan para las CPU de varios núcleos y el rendimiento de aumento de memoria que se encuentran en el hardware moderno.

© JMA 2016. All rights reserved

Pasos básicos para procesar una instrucción

1. El analizador examina la instrucción y la divide en unidades lógicas como palabras clave, expresiones, operadores e identificadores.
2. Se genera un árbol de la consulta, a veces denominado árbol de secuencia, que describe los pasos lógicos que se requieren para transformar los datos de origen en el formato que necesita el conjunto de resultados.
3. El optimizador de consultas analiza diferentes formas de acceso a las tablas de origen y a continuación:
 - Selecciona la serie de pasos que devuelve los resultados de la forma más rápida utilizando el menor número posible de recursos.
 - El árbol de la consulta se actualiza para registrar esta serie exacta de pasos.
 - La versión final y optimizada del árbol de la consulta se denomina plan de ejecución.
4. El motor relacional comienza a ejecutar el plan de ejecución. A medida que se procesan los pasos que necesitan datos de las tablas base, el motor relacional solicita al motor de almacenamiento que pase los datos de los conjuntos de filas solicitados desde el motor relacional.
5. El motor relacional procesa los datos que devuelve el motor de almacenamiento en el formato definido para el conjunto de resultados y devuelve el conjunto de resultados al cliente.

© JMA 2016. All rights reserved

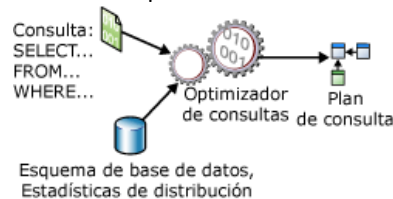
Optimizador de consultas

- El optimizador de consultas de SQL Server es un optimizador basado en el costo.
- Cada plan de ejecución posible tiene asociado un costo en términos de la cantidad de recursos del equipo que se utilizan.
- El optimizador de consultas debe analizar los planes posibles y elegir el de menor costo estimado.
- Algunas instrucciones complejas tienen miles de planes de ejecución posibles. En estos casos, el optimizador de consultas no analiza todas las combinaciones posibles. En lugar de esto, utiliza algoritmos complejos para encontrar un plan de ejecución que tenga un costo razonablemente cercano al mínimo posible.
- El optimizador elige, además de por el costo de recursos mínimo, el plan que devuelve resultados al usuario a la mayor brevedad posible con un costo razonable de recursos.

© JMA 2016. All rights reserved

Optimizador de consultas

- La entrada al optimizador consta de la consulta, el esquema de la base de datos (definiciones de tabla e índice) y las estadísticas de base de datos.
- La salida del optimizador es un plan de ejecución de la consulta, en ocasiones denominado plan de la consulta o simplemente plan.
- Un plan de ejecución de consulta es una definición de los siguientes elementos:
 - La secuencia en la que se tiene acceso a las tablas de origen.
 - Los métodos que se utilizan para extraer los datos de cada tabla.



© JMA 2016. All rights reserved

Optimizador de consultas

- El optimizador utilizará un plan de ejecución en paralelo para devolver resultados si esto no afecta negativamente a la carga del servidor.
- El optimizador confía en las estadísticas de distribución y en los índices disponibles cuando calcula los costos de recursos de métodos diferentes para extraer información de una tabla o un índice.
- Las estadísticas de distribución se mantienen para las columnas y los índices, indicando la posibilidad de seleccionar los valores de un índice o de una columna determinados.
- Si las estadísticas no están actualizadas, puede que el optimizador de consultas no realice la mejor elección para el estado actual de la tabla.

© JMA 2016. All rights reserved

Optimizador de consultas

- El optimizador de consultas es importante porque permite que el servidor de la base de datos se ajuste dinámicamente a las condiciones cambiantes de la base de datos, sin necesitar la entrada de un programador o de un administrador de base de datos.
- Esto permite a los programadores centrarse en la descripción del resultado final de la consulta.
- Pueden estar seguros de que el optimizador de consultas creará un plan de ejecución eficaz para el estado actual de la base de datos cada vez que se ejecuta la instrucción.

© JMA 2016. All rights reserved

Estimación de cardinalidad

- El optimizador de consultas determina el costo de ejecución de un plan de consulta en función de dos factores principales:
 - Una predicción del número total de filas procesadas en cada nivel de un plan de consulta, denominado cardinalidad del plan.
 - El modelo de costos del algoritmo determinado por los operadores que se utiliza en la consulta basándose en la cardinalidad.
- La estimación de cardinalidad se calcula principalmente a partir de histogramas que se crean al crear, manual o automáticamente, índices y estadísticas, también puede utilizar en ocasiones información de restricciones y nuevas versiones lógicas de.
- El optimizador de consultas utiliza las estimaciones de cardinalidad a la hora de elegir un plan para ejecutar la consulta.
- En algunos casos, SQL Server no puede calcular con precisión las cardinalidades. Esto deriva en cálculos de costos inexactos que pueden provocar planes de consulta de menor calidad.
- La calidad del plan de consulta tiene un impacto directo sobre la mejora del rendimiento de las consultas.

© JMA 2016. All rights reserved

Causas de imprecisiones en la estimación de cardinalidad

- Consultas con predicados que utilizan operadores de comparación entre distintas columnas de la misma tabla.
- Consultas con predicados que utilizan operadores y se cumple alguna de las siguientes condiciones:
 - No hay estadísticas en las columnas que se utilizan a uno u otro lado de los operadores.
 - La distribución de valores en las estadísticas no es uniforme, pero la consulta busca un conjunto de valores muy selectivos. Esta situación se cumple especialmente cuando el operador es distinto al operador de igualdad (=).
 - El predicado utiliza el operador de comparación No es igual a (!=) o el operador lógico NOT.
- Consultas que utilizan alguna de las funciones integradas o una función escalar definida por el usuario cuyo argumento no es un valor constante.
- Consultas que implican columnas de combinación por medio de operadores aritméticos o de concatenación de cadenas.
- Consultas que comparan variables cuyos valores no se conocen cuando la consulta se compila y optimiza.

© JMA 2016. All rights reserved

Estimación de cardinalidad (2014)

- La lógica de estimación de la cardinalidad, denominada estimador de cardinalidad, se ha rediseñado en SQL Server 2014 para mejorar la calidad de los planes de consulta y, por tanto, mejorar el rendimiento de las consultas.
- El nuevo estimador de cardinalidad incorpora suposiciones y algoritmos que funcionan bien en las cargas de trabajo OLTP y de almacenamiento de datos modernos. Se puede seguir utilizando el anterior:


```
ALTER DATABASE SCOPED CONFIGURATION
SET LEGACY_CARDINALITY_ESTIMATION = ON;
SELECT ...
OPTION (USE HINT ('FORCE_LEGACY_CARDINALITY_ESTIMATION'));
```
- El almacén de consultas (2016) es una herramienta muy útil para examinar el rendimiento de las consultas.

© JMA 2016. All rights reserved

Parametrización

- El uso de parámetros, incluidos los marcadores de parámetros de las aplicaciones ADO, OLE DB y ODBC, puede incrementar las posibilidades de volver a utilizar los planes de ejecución y evita las inyecciones de SQL.
- Si una instrucción Transact-SQL se ejecuta sin parámetros, SQL Server parametriza la instrucción internamente para aumentar las posibilidades de hacerla coincidir con un plan de ejecución existente. Este proceso se denomina parametrización simple. En las versiones de SQL Server anteriores a la 2005, el proceso se denominaba parametrización automática.
- SQL Server parametriza una clase relativamente pequeña de consultas en la parametrización simple. La parametrización forzada establece que, con algunas limitaciones, todas las consultas de una base de datos se parametrizan: cualquier valor literal que aparezca en una instrucción SELECT, INSERT, UPDATE o DELETE, enviado de cualquier forma, se convierte en un parámetro.

© JMA 2016. All rights reserved

Examen de parámetros

- Con la expresión sensibilidad de los parámetros, también conocida como "examen de parámetros", se hace referencia a un proceso mediante el cual SQL Server "examina" los valores de parámetros actuales durante la compilación o la recompilación y los pasa al optimizador de consultas para que se puedan usar para generar planes de ejecución de consultas potencialmente más eficaces.
- Los valores de parámetros se examinan durante la compilación o la recompilación de los siguientes tipos de lotes:
 - Procedimientos almacenados
 - Consultas enviadas mediante `sp_executesql`
 - Consultas preparadas

© JMA 2016. All rights reserved

Caché de procedimientos

- SQL Server tiene un bloque de memoria que se utiliza para almacenar planes de ejecución y búferes de datos. El porcentaje del conjunto que se asigna a los planes de ejecución o a los búferes de datos varía dinámicamente según el estado del sistema. La parte del bloque de memoria que se utiliza para almacenar los planes de ejecución se denomina caché de procedimientos.
- Los planes de ejecución de SQL Server tienen los siguientes componentes principales:
 - **Plan de consulta:** La mayor parte del plan de ejecución es una estructura de datos reentrante de solo lectura que varios usuarios pueden utilizar. No se almacena ningún contexto de usuario en el plan de consulta. Nunca hay más de dos copias del plan de consulta en la memoria: una copia para todas las ejecuciones en serie y otra para todas las ejecuciones en paralelo.
 - **Contexto de ejecución:** Cada usuario que ejecuta la consulta tiene una estructura de datos que alberga los datos específicos de su ejecución, como los valores de los parámetros. Las estructuras de datos del contexto de ejecución se vuelven a utilizar. Si un usuario ejecuta una consulta y una de las estructuras no está en uso, ésta se reinicializa con el contexto del nuevo usuario.

© JMA 2016. All rights reserved

Caché de procedimientos

- El uso de parámetros, incluidos los marcadores de parámetros de las aplicaciones ADO, OLE DB y ODBC, incrementa las posibilidades de volver a utilizar los planes de ejecución. Si no se incorporan explícitamente parámetros en el diseño de las consultas, el optimizador de consultas de SQL Server se encarga de parametrizar automáticamente determinadas consultas mediante el uso de la parametrización simple o forzada.
- Los planes de ejecución permanecen en la memoria caché de procedimientos en tanto en cuanto haya suficiente memoria para almacenarlos.
- Cuando existe presión de memoria, el motor de base de datos usa un enfoque basado en costos para determinar qué planes de ejecución hay que quitar de la memoria caché de procedimientos.
- Determinados cambios en la base de datos pueden causar que un plan de ejecución deje de ser eficaz o válido en función del nuevo estado de la base de datos.
- SQL Server detecta los cambios que hacen que un plan de ejecución no sea válido y lo marca como no válido.
- Después, debe volver a compilarse un nuevo plan para la próxima conexión que ejecute la consulta.

© JMA 2016. All rights reserved

Grado de paralelismo

- SQL Server detecta de forma automática el mejor grado de paralelismo para cada instancia de una ejecución de consulta en paralelo o de una operación de índice del lenguaje de definición de datos (DDL).
- Para ello utiliza los siguientes criterios:
 - Si SQL Server se ejecuta en un equipo que disponga de más de un microprocesador o CPU.
 - Si hay suficientes subprocesos de trabajo disponibles.
 - El coste asociado al tipo de operación de consulta o índice ejecutado.
 - Si hay un número suficiente de filas que se van a procesar.
 - Si las estadísticas de distribución actuales están disponibles.
- El grado de paralelismo se puede establecer a nivel de servidor, de base de datos, de instrucción de consulta o índice (con la sugerencia de consulta MAXDOP) y de carga de trabajo del grupo de cargas de trabajo del Resource Governor.

© JMA 2016. All rights reserved

Causas de ejecución lenta

- Comunicaciones de red lentas.
- Memoria inadecuada en el equipo servidor o falta de memoria disponible para SQL Server.
- Bloqueos e interbloqueos.
- Falta de estadísticas útiles
- Falta de índices útiles.
- Falta de vistas indizadas útiles.
- Falta de particiones útiles.
- Inadecuada redacción de la consulta.
- Inadecuada estructura lógica o física de la base de datos.

© JMA 2016. All rights reserved

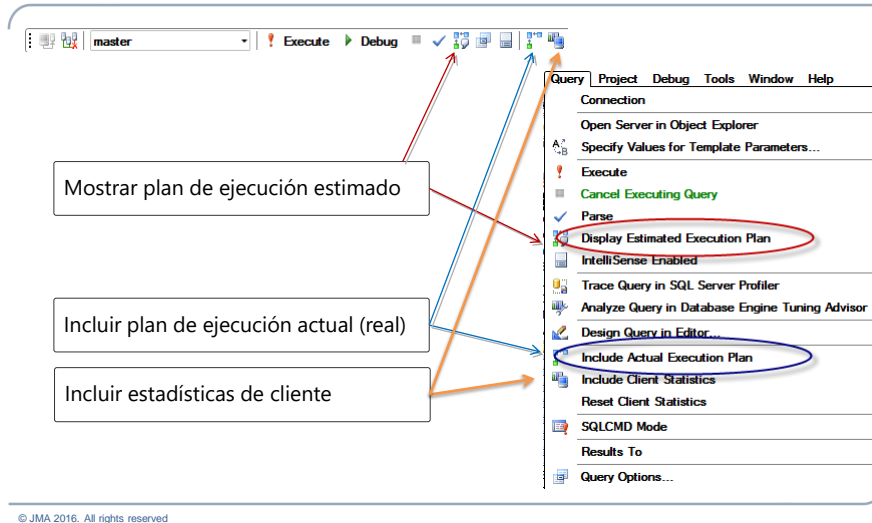
Acceso a los planes de ejecución

- Las herramientas de SQL Server proporcionan acceso a los planes de ejecución para mostrar cómo se ejecutó o ejecutaría una consulta.
- SSMS proporciona acceso a dos tipos de planes de ejecución:
 - Los planes de ejecución **estimados** no ejecutan la consulta, muestran el plan que probablemente se utilizaría si se ejecutase la consulta.
 - Los planes de ejecución **reales** se muestran una vez que se ejecuta la consulta. Muestran el plan que realmente ha utilizado por SQL Server.
- Los planes están disponibles en formato de texto (obsoleto), en formato XML y en representación gráfica.
- El Visor de planes es accesible en el panel de resultados de SSMS.
- Funciones y vistas de administración dinámica relacionadas con ejecuciones

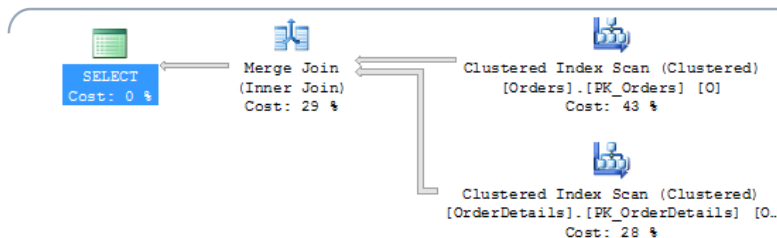
```
SELECT * FROM sys.dm_exec_cached_plans cp CROSS APPLY
sys.dm_exec_query_plan(cp.plan_handle);
```

© JMA 2016. All rights reserved

Mostrar los planes y estadísticas



Interpretar los planes de ejecución

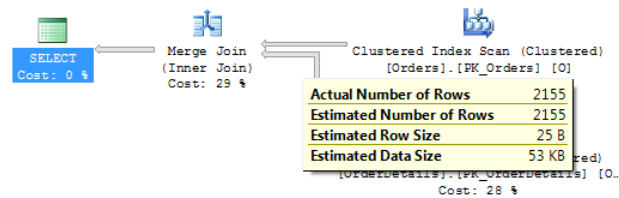


- Los planes de ejecución gráficos deben leerse de arriba a abajo y de derecha a izquierda.
- Cada nodo de la estructura en árbol se representa como un icono que especifica el operador lógico y físico utilizado para ejecutar esa parte de la consulta o instrucción.
- Cada nodo está relacionado con un nodo principal.
- Los nodos secundarios que tienen el mismo nodo principal se dibujan en la misma columna. Sin embargo, no todos los nodos de la misma columna tienen necesariamente el mismo nodo principal.
- Los operadores se muestran como símbolos relacionados con un nodo principal específico.
- Cada nodo se conecta a su nodo principal mediante reglas con puntas de flecha.

© JMA 2016. All rights reserved

Interpretar los planes de ejecución

- El ancho de la flecha es proporcional al número de filas.
- Se utiliza el número real de filas cuando está disponible. Si no, se utiliza el número estimado de filas.
- Cuando la consulta contiene varias instrucciones, se dibujan varios planes de ejecución de la consulta.
- Las partes de las estructuras en árbol se determinan por el tipo de instrucción ejecutada.
- Si se coloca el puntero sobre uno de los iconos o una de las flechas muestra la información ampliada sobre el mismo



© JMA 2016. All rights reserved

Mostrar estadísticas de ejecución

- Se pueden incluir y comparar las estadísticas de ejecución de cliente de diferentes redacciones de la consulta.
- STATISTICS TIME mostrará el tiempo dedicado a analizar y compilar una consulta
SET STATISTICS TIME ON;
- STATISTICS IO mostrará la cantidad de actividad de disco generada por una consulta
SET STATISTICS IO ON;
- STATISTICS PROFILE mostrará la información de perfil de una instrucción
SET STATISTICS PROFILE ON;
- Para obtener resultados fiables es conveniente forzar un punto de comprobación y borrar el buffer cache para que vuelva a cargar las páginas de disco:
CHECKPOINT
DBCC DROPCLEANBUFFERS
Reset Client Statistics (menú)

© JMA 2016. All rights reserved

Operadores

- Los operadores describen cómo SQL Server ejecuta una consulta o una instrucción DML (Lenguaje de manipulación de datos).
- El optimizador de consultas utiliza operadores para generar un plan de consulta con el fin de crear el resultado especificado en la consulta o para realizar la operación especificada en la instrucción DML.
- El plan de consulta es un árbol que consta de operadores físicos.
- Los operadores se clasifican como:
 - Operadores lógicos: describen una operación de procesamiento de consulta relacional a nivel conceptual.
 - Operadores físicos: implementan realmente la operación definida por un operador lógico utilizando un método o algoritmo concreto. Los operadores físicos se inicializan, recopilan datos y se cierran.
- Algunos operadores son a la vez lógicos y físicos.

© JMA 2016. All rights reserved

Recuperación de páginas

- **Scan:** recupera todas las páginas aunque solo se devuelven las filas que cumplan el predicado.



Table Scan



Clustered Index Scan



Index Scan

- **Seek:** usa la capacidad de búsqueda de los índices para recuperar solo las páginas que contengan las filas que cumplan el predicado.



Clustered Index Seek



Index Seek

- **Spool:** guarda un resultado de consulta intermedio en la base de datos tempdb.



Spool






Table Spool



Index Spool



© JMA 2016. All rights reserved

Búsquedas

-  **Bookmark Lookup:** usa un marcador (identificador de fila o clave de agrupación en clústeres) para buscar la fila correspondiente en la tabla o índice clúster.
Sustituido por los operadores Clustered Index Seek y RID Lookup.
-  **Key Lookup:** es una búsqueda de marcadores en un tabla con un índice clúster.
-  **RID Lookup:** es una búsqueda de marcadores en un montón que usa un identificador de fila suministrado (RID).

© JMA 2016. All rights reserved

Combinaciones

-  **Nested Loops:** realiza las operaciones lógicas de combinación interna, combinación externa izquierda, semicombinación izquierda y anti semicombinación. Las combinaciones de bucles anidados buscan en la tabla interna cada fila de la tabla externa, normalmente mediante un índice. El procesador de consultas decide, en función de los costos anticipados, si debe ordenar o no la entrada externa para mejorar la ubicación de las búsquedas en el índice de la entrada interna. Se devuelven las filas que cumplen el predicado.
-  **Merge Join:** requiere dos entradas ordenadas por sus respectivas columnas, que se pueden realizar mediante la inserción de operaciones de ordenación explícitas en el plan de consulta. Es especialmente eficaz si no se necesita un orden explícito, por ejemplo, si hay un índice idóneo de árbol b en la base de datos o si el orden se puede aprovechar en varias operaciones, como en una combinación de mezcla y una agrupación con acumulación.

© JMA 2016. All rights reserved

Combinaciones



Hash Match: genera una tabla hash y calcula un valor hash para cada fila de su entrada de compilación. A continuación, por cada fila de sondeo (como corresponda), calcula un valor hash (con la misma función hash) y busca las coincidencias en la tabla hash. El comportamiento depende de la operación lógica que se esté realizando:

- Para cualquier combinación, utiliza la primera entrada (superior) para generar la tabla hash y la segunda entrada (inferior) para sondear la tabla hash. Obtendrá como resultado las coincidencias (o las no coincidencias) que indique el tipo de combinación. Si varias combinaciones utilizan la misma columna de combinación, estas operaciones se agrupan en un equipo hash.
- Para los operadores Distinct o Aggregate, utiliza la entrada para generar la tabla hash (para ello, quite los duplicados y calcule las expresiones de agregado). Cuando se haya generado la tabla hash, recorre la tabla y presenta todas las entradas.
- En el caso del operador Union, utiliza la primera entrada para generar la tabla hash (para ello, quite los duplicados). Usa la segunda entrada (que no debe tener duplicados) para sondear la tabla hash, devolver todas las filas que no tengan coincidencias y, a continuación, recorrer la tabla hash para devolver todas las entradas.



© JMA 2016. All rights reserved

Tipos de combinación

- **Cross Join:** combina cada fila de la primera entrada (superior) con cada fila de la segunda entrada (inferior).
- **Inner Join:** devuelve todas las filas que cumplen la combinación de la primera entrada (superior) con la segunda entrada (inferior).
- **Full Outer Join:** devuelve cada fila que cumple el predicado de combinación de la primera entrada (superior) combinada con cada fila de la segunda entrada (inferior). También devuelve filas de la primera entrada que no tenga coincidencias en la segunda entrada y de la segunda entrada que no tenga coincidencias en la primera entrada. La entrada que no contiene valores coincidentes se devuelve como un valor nulo.
- **Left Outer Join:** devuelve cada fila que cumple la combinación de la primera entrada (superior) con la segunda entrada (inferior). También devuelve las filas de la primera entrada que no tienen filas coincidentes en la segunda entrada. Las filas que no coinciden en la segunda entrada se devuelven como valores NULL. Si no hay ningún predicado de combinación, cada una de las filas es coincidente.
- **Left Semi Join:** devuelve todas las filas de la primera entrada (superior) cuando hay una fila coincidente en la segunda entrada (inferior). Si no hay ningún predicado de combinación, cada fila es una fila coincidente.
- **Left Anti Semi Join:** devuelve todas las filas de la primera entrada (superior) cuando no hay ninguna fila coincidente en la segunda entrada (inferior). Si no hay ningún predicado de combinación, cada fila es una fila coincidente.
- **Right Outer Join, Right Semi Join, Right Anti Semi Join:** Similares a los Left empezando con la segunda entrada.
- **Union:** recorre varias entradas, obtiene cada fila recorrida y quita los duplicados.






© JMA 2016. All rights reserved

Filtrados

-  **Filter:** recorre la entrada y solo devuelve las filas que cumplen la expresión del filtro (predicado).
-  **Bitmap:** usado para implementar filtros de mapas de bits en planes de consulta paralelos. Los filtros de mapas de bits agiliza la ejecución de la consulta al eliminar las filas con valores de clave que no pueden generar ningún registro de combinación antes de pasar las filas a través de otro operador, por ejemplo, el operador Parallelism. Un filtro de mapas de bits usa una representación compacta de un conjunto de valores de una tabla en una parte del árbol de operadores para filtrar filas de una segunda tabla en otra parte del árbol. Al quitar las filas innecesarias en las primeras fases de la consulta, los operadores subsiguientes tienen que trabajar en menos filas y el rendimiento total de la consulta mejora. El optimizador determina cuándo un mapa de bits es suficientemente selectivo para resultar útil y en qué operadores se aplica el filtro. Bitmap es un operador físico.






© JMA 2016. All rights reserved

Otros operadores

-  **Sort:** ordena todas las filas entrantes. Las columnas llevan como prefijo el valor ASC, si el orden de las columnas es ascendente, o el valor DESC, si es descendente.
-  **Stream Aggregate:** agrupa las filas por una o varias columnas y, a continuación, calcula una o varias expresiones agregadas devueltas por la consulta. El resultado de este operador puede ser utilizado por operadores posteriores de la consulta, devuelto al cliente, o ambas cosas. Requiere una entrada ordenada por las columnas dentro de sus grupos.
-  **Compute Scalar:** evalúa una expresión para generar un valor escalar calculado que se puede devolver al usuario, hacer referencia a él en cualquier otra parte de la consulta o ambas cosas a la vez.
-  **UDX:** implementan una de las múltiples operaciones XQuery y XPath. Los operadores extendidos (UDX): FOR XML, XML SERIALIZER, XML FRAGMENT SERIALIZER, XQUERY STRING, XQUERY LIST DECOMPOSER, XQUERY DATA, XQUERY CONTAINS, UPDATE XML NODE.
-  **Top:** recorre la entrada y solo devuelve el primer número o porcentaje especificado de filas, basándose en un criterio de ordenación si es posible.

© JMA 2016. All rights reserved

Otros operadores

-  **Assert:** comprueba una condición y si la expresión da como resultado un valor distinto a NULL se generará el error apropiado (validaciones).
-  **Concatenation:** explora varias entradas y devuelve cada fila explorada, se utiliza para implementar la construcción UNION ALL, tiene dos o más entradas y una salida. Concatenation copia filas del primer flujo de entrada en el flujo de salida y, a continuación, repite esta operación con cada flujo de entrada adicional.
-  **Segment:** divide el conjunto de entrada en segmentos basados en el valor de una o varias columnas.
-  **Split:** se utiliza para optimizar el procesamiento de actualizaciones. Divide cada operación de actualización en una operación de eliminación y una operación de inserción.
-  **Table-valued Function:** evalúa una función con valores de tabla y almacena las filas resultantes en la base de datos tempdb. Cuando los iteradores principales solicitan las filas, la función con valores de tabla devuelve las filas desde tempdb.

© JMA 2016. All rights reserved

Detalles del nodo

- **Operación física:** Operador físico utilizado. Los operadores físicos presentados en color rojo indican que el optimizador de consultas ha emitido una advertencia, por ejemplo, la falta de estadísticas de columna o de predicados de combinación. Esto puede causar que el optimizador de consultas elija un plan de consulta menos eficaz que el esperado.
- **Operación lógica:** Operador lógico que coincide con el operador físico, como el operador Inner Join.
- **Tamaño de fila estimado:** Tamaño estimado de la fila creada por el operador (bytes).
- **Costo de E/S estimado:** Costo estimado de toda la actividad de E/S para la operación. Este valor *debe ser lo más pequeño posible*.
- **Costo de CPU estimado:** Costo estimado de toda la actividad de la CPU para la operación.
- **Costo de operador estimado:** Costo del optimizador de consultas para ejecutar esta operación. El costo de esta operación como porcentaje del costo total de la consulta se muestra entre paréntesis. Debido a que el motor de consultas selecciona la operación más eficaz para realizar la consulta o ejecutar la instrucción, este valor *debe ser el menor posible*.
- **Costo de subárbol estimado:** Costo total del optimizador de consultas para ejecutar esta operación y todas las operaciones del mismo subárbol anteriores a ésta.
- **Número de filas o Número de filas estimado:** Número de filas que produce el operador.
- En función del nodo: Objeto de salida, Lista de salida (columnas), Número de ejecuciones, Número de filas, Tamaño, Ordenación
- En algunos casos se mostraran advertencias o sugerencias sobre el nodo.

© JMA 2016. All rights reserved

Sugerencias de tabla

- Las sugerencias de tabla invalidan el comportamiento predeterminado del optimizador de consultas mientras dura la instrucción de lenguaje de manipulación de datos (DML), especificando un método de bloqueo, uno o varios índices, una operación de procesamiento de la consulta como, por ejemplo, un examen de tabla o una búsqueda de índice, u otras opciones.
- Las sugerencias de tabla se especifican en la cláusula FROM ... WITH de la instrucción DML y solo afectan a la tabla o a la vista a la que se hace referencia en esa cláusula.

© JMA 2016. All rights reserved

SELECT

- | | |
|--|---------------------|
| • INDEX (index_value [,...n]) INDEX = (index_value) | • READCOMMITTED |
| • FASTFIRSTROW | • READCOMMITTEDLOCK |
| • FORCESEEK [(index_value (index_column_name [,...]))] | • READPAST |
| • FORCESCAN | • READUNCOMMITTED |
| • HOLDLOCK | • REPEATABLEREAD |
| • NOLOCK | • ROWLOCK |
| • NOWAIT | • SERIALIZABLE |
| • PAGLOCK | • TABLOCK |
| | • TABLOCKX |
| | • UPDLOCK |
| | • XLOCK |

© JMA 2016. All rights reserved

INSERT, UPDATE, DELETE

- KEEPIDENTITY
- KEEPDEFAULTS
- FASTFIRSTROW
- HOLDLOCK
- IGNORE_CONSTRAINTS
- IGNORE_TRIGGERS
- NOLOCK
- NOWAIT
- PAGLOCK
- READCOMMITTED
- READCOMMITTEDLOCK
- READPAST
- REPEATABLEREAD
- ROWLOCK
- SERIALIZABLE
- TABLOCK
- TABLOCKX
- UPDLOCK
- XLOCK

© JMA 2016. All rights reserved

Sugerencias de consulta

- Las sugerencias de consulta (hint) invalidan el comportamiento predeterminado del optimizador de consultas mientras dura la instrucción de consulta.
- Son sugerencias por que el optimizador no está obligado a seguirlas.
- Se pueden usar para especificar un método de bloqueo en las tablas afectadas, uno o varios índices, una operación de procesamiento de la consulta como un recorrido de tabla o una búsqueda de índice, u otras opciones.
- Las sugerencias de consulta se especifican en la cláusula OPTION y se aplican a toda la consulta.
- Como el optimizador de consultas de SQL Server suele seleccionar el mejor plan de ejecución para las consultas, se recomienda que solo se hagan como último recurso.

© JMA 2016. All rights reserved

Sugerencias de consulta

- { HASH | ORDER } GROUP
- { CONCAT | HASH | MERGE } UNION
- { LOOP | MERGE | HASH } JOIN
- FAST number_rows
- FORCE ORDER
- MAXDOP
number_of_processors
- OPTIMIZE FOR (
@variable_name { UNKNOWN
| = literal_constant } [, ...n])
- OPTIMIZE FOR UNKNOWN
- PARAMETERIZATION { SIMPLE
| FORCED }
- RECOMPILE
- ROBUST PLAN
- KEEP PLAN
- KEEPFIXED PLAN
- EXPAND VIEWS
- MAXRECURSION number
- USE PLAN N'xml_plan'
- TABLE HINT (
exposed_object_name [,
<table_hint> [[,]...n]])

© JMA 2016. All rights reserved

Guías de plan

- Las guías de plan se pueden utilizar para optimizar el rendimiento de las consultas cuando no pueda o no desee cambiar directamente el texto de dichas consultas.
- Las guías de plan pueden ser de gran utilidad cuando el rendimiento de un pequeño subconjunto de consultas de una aplicación de base de datos implementada por otro proveedor no es el esperado.
- Las guías de plan influyen en la optimización de las consultas adjuntando sugerencias de consulta o un plan de consulta fijo para ellas.
- En la guía de plan, se especifica la instrucción de Transact-SQL que se desea optimizar y además una cláusula OPTION que incluye las sugerencias de consulta que se desean utilizar o un plan de consulta específico con el que desea optimizar la consulta.
- Cuando la consulta se ejecuta, el SQL Server hace coincidir la instrucción de Transact-SQL con la guía de plan y además asocia en tiempo de ejecución la cláusula OPTION a la consulta o utiliza el plan de consultas especificado.

© JMA 2016. All rights reserved

Tipos de guías de plan

- **OBJECT:** compara las consultas que se ejecutan en el contexto de procedimientos almacenados de Transact-SQL, funciones escalares definidas por el usuario, funciones definidas por el usuario con valores de tabla de múltiples instrucciones y desencadenadores DML.
- **SQL:** compara las consultas que se ejecutan en el contexto de instrucciones independientes de Transact-SQL y lotes que no forman parte de un objeto de base de datos.
 - Las guías de plan basadas en SQL también se pueden usar para comparar consultas que se parametrizan en un formulario especificado.
 - Las guías de plan de SQL se aplican a las instrucciones y lotes independientes de Transact-SQL.
 - Con frecuencia, las aplicaciones envían esas instrucciones utilizando el procedimiento almacenado del sistema `sp_executesql`.
- **TEMPLATE:** compara consultas independientes que se parametrizan en un formulario especificado.
 - Estas guías de plan se usan para reemplazar la opción `PARAMETERIZATION` actual de una base de datos para una clase de consultas por medio de `SET`.

© JMA 2016. All rights reserved

Ejemplos de guías de plan

```
sp_create_plan_guide @name = N'Guide1',
    @type = N'OBJECT',
    @module_or_batch = N'Sales.GetSalesOrderByCountry',
    @stmt = N'SELECT * FROM Sales.SalesOrderHeader AS h,
        Sales.Customer AS c,
        Sales.SalesTerritory AS t
    WHERE h.CustomerID = c.CustomerID
        AND c.TerritoryID = t.TerritoryID
        AND CountryRegionCode = @Country_region',
    @params = NULL,
    @hints = N'OPTION (OPTIMIZE FOR (@Country_region = N"US"))';

sp_create_plan_guide @name = N'Guide2',
    @type = N'SQL',
    @module_or_batch = NULL,
    @stmt = N'SELECT TOP 1 * FROM Sales.SalesOrderHeader ORDER BY OrderDate DESC',
    @params = NULL,
    @hints = N'OPTION (MAXDOP 1)';

sp_control_plan_guide, sys.plan_guides, sys.fn_validate_plan_guide
```

© JMA 2016. All rights reserved

Limitaciones para las guías

- El número total de guías de plan que se pueden crear solo está limitado por los recursos de los que disponga el sistema.
- Las guías de plan deberían limitarse a aquellas consultas de gran importancia cuyo rendimiento se desea mejorar o estabilizar.
- No se deben usar las guías de plan para influenciar la mayor parte de la carga de la consulta de una aplicación implementada.
- Se debe volver a evaluar y probar las definiciones de guías de plan al actualizar la aplicación a una nueva versión de SQL Server.
 - Los requisitos de optimización del rendimiento y el comportamiento de la coincidencia de las guías de plan pueden cambiar.
 - Aunque una guía de plan no válida no hará que una consulta provoque un error, el plan se compilara sin utilizar la guía de plan y posiblemente no sea la mejor opción
- No están disponibles en las ediciones Express.

© JMA 2016. All rights reserved

Planes de ejecución forzados

- El `xml_plan` es un literal de cadena derivado del plan de consulta con formato XML que se genera para una consulta.
- Un `xml_plan` se puede redactar manualmente u obtenerse mediante “Guardar plan de ejecución”, consultando la columna `query_plan` de la función `sys.dm_exec_query_plan` o eventos de traza: Showplan XML, Showplan XML Statistics Profile y Showplan XML For Query Compile.
- No todos los elementos de los planes de consulta con formato XML se fuerzan con la sugerencia `USE PLAN`. Los elementos que calculan expresiones escalares se omiten, así como algunas expresiones relacionales.
- El plan de consulta con formato XML especificado en `xml_plan` debe validarse con el esquema XSD `Showplanxml.xsd`.
- La sugerencia de consulta `USE PLAN` se puede especificar como una sugerencia de consulta en una instrucción SQL independiente, o especificarse en el parámetro `@hints` de una guía de plan.
- Se debe indicar siempre `xml_plan` como literal Unicode especificando el prefijo `N`, como en `N'xml_plan'`, con lo que se garantiza que no se pierda ninguno de los caracteres del plan específico del estándar Unicode cuando SQL Server Database Engine (Motor de base de datos de SQL Server) interprete la cadena.

© JMA 2016. All rights reserved

Limitaciones para Planes forzados

- Los cambios que se produzcan en la base de datos, como la eliminación de índices, pueden invalidar un plan de consulta forzado.
- Un plan de consulta puede quedarse obsoleto aunque no se haga directamente referencia en el plan a un objeto eliminado.
- Instalar un Service Pack o una nueva versión de SQL Server puede impedir que fuerce un plan generado en una versión anterior, por lo que deberán probarse todos los planes forzados siempre que se actualice el servidor.
- El uso de planes forzados en una consulta reemplaza todas las sugerencias de combinación y de índice utilizadas en la misma consulta.
- Los únicos planes de consulta que pueden forzarse son los que pueden encontrarse mediante la estrategia de búsqueda típica del optimizador de consultas.

© JMA 2016. All rights reserved

Almacén de consultas (2016)

- Los planes de ejecución para cualquier consulta específica en SQL Server suelen evolucionar con el tiempo debido a una serie de motivos diferentes, como cambios en las estadísticas, cambios de esquema, creación o eliminación de los índices, etc.
- La caché de procedimientos (donde se almacenan los planes de consulta almacenados en caché) solo almacena el plan de ejecución más reciente. Los planes también se eliminan de la caché de planes debido a la presión de memoria.
- Como resultado, es posible que las regresiones de rendimiento de consultas provocadas por los cambios de planes de ejecución no sean triviales y que su resolución lleve mucho tiempo.
- La característica del Almacén de consultas de SQL Server ofrece datos detallados sobre el rendimiento y la elección del plan de consultas.

© JMA 2016. All rights reserved

Almacén de consultas (2016)

- Esta característica simplifica la solución de problemas de rendimiento al permitirle encontrar rápidamente las diferencias de rendimiento provocadas por cambios en los planes de consulta.
- El Almacén de consultas captura automáticamente un historial de consultas, planes y estadísticas en tiempo de ejecución y las conserva para su revisión.
- Además, separa los datos por ventanas de tiempo, lo que permite ver patrones de uso de la base de datos y comprender cuándo se produjeron cambios del plan de consultas en el servidor.

© JMA 2016. All rights reserved

Almacén de consultas (2016)

- El Almacén de consultas no está habilitado de forma predeterminada para las nuevas bases de datos de SQL Server. Se habilita en el cuadro de diálogo Propiedades de la base de datos, página Almacén de consultas y en el cuadro Modo de operación (solicitado) se selecciona Lectura y escritura.
 - SET QUERY_STORE = ON (OPERATION_MODE = READ_WRITE);
- El Almacén de consultas contiene tres almacenes:
 - un almacén de planes para conservar la información del plan de ejecución,
 - un almacén de estadísticas de runtime para conservar la información de las estadísticas de ejecución,
 - un almacén de estadísticas de espera para conservar la información de las estadísticas de espera.

© JMA 2016. All rights reserved

Ajuste automático (2017)

- El ajuste automático es una característica de base de datos que proporciona información de los posibles problemas de rendimiento de las consultas, recomienda soluciones y corrige automáticamente los problemas identificados.
 - Corrección automática de planes de ejecución
 - Administración automática de índices
- El ajuste automático es un proceso de supervisión y análisis continuo que aprende constantemente sobre las características de la carga de trabajo e identifica posibles problemas y mejoras.

© JMA 2016. All rights reserved

Ajuste automático (2017)

- Se puede habilitar el ajuste automático por base de datos y especificar que se debe forzar el último plan bueno cada vez que se detecta una regresión de cambios en el plan:

```
ALTER DATABASE <yourDatabase> SET AUTOMATIC_TUNING (
FORCE_LAST_GOOD_PLAN = ON );
```
- Una vez habilitada esta opción, se aplicará en Motor de base de datos automáticamente las recomendaciones en las que la ganancia de CPU estimada sea superior a 10 segundos, o el número de errores en el nuevo plan sea mayor que el número de errores del plan recomendado, y comprobará que el plan forzado es mejor que el actual.

© JMA 2016. All rights reserved

Procesamiento de consultas inteligente (2019)

- El Procesamiento inteligente de consultas (IQP) basado en el análisis de los almacenes de consultas (2016) incluye características con un gran impacto que mejoran el rendimiento de las cargas de trabajo existentes con un esfuerzo de implementación mínimo.
- **Compilación aplazada de la variable de tabla:** mejora la calidad del plan y el rendimiento general de las consultas que hacen referencia a variables de tabla.
- **Inserción de UDF escalar:** transforma las UDF escalares (funciones definidas por el usuario) en expresiones relacionales y las incrusta en la consulta SQL que realiza la llamada.

© JMA 2016. All rights reserved

IQP

- **Comentarios de concesión de memoria en el modo por lotes:** recalcula la memoria real necesaria para repetir consultas y, a continuación, actualiza el valor de concesión del plan almacenado en caché para optimizar el uso de recursos y reducir los costos (anteriormente solo estaba disponible en modo por lotes para tablas con índice de columnstore)
- **Modo por lotes en el almacén de filas:** permite la ejecución en modo por lotes para cargas de trabajo analíticas sin necesidad de índices de almacén de columnstore. Ideal para escenarios en los que la creación de un índice de columnstore agrega demasiada sobrecarga a una carga de trabajo transaccional (como HTAP) o no es una buena opción para la carga de trabajo.

© JMA 2016. All rights reserved

IQP

- **Procesamiento aproximado de consultas:** agrega grandes conjuntos de datos donde la capacidad de respuesta es más crítica que la precisión absoluta para reducir el costo y el espacio de memoria necesarios.
- **Generación de perfiles de consultas ligera:** recopila estadísticas de tiempo de ejecución e información sobre los planes de ejecución con una sobrecarga de rendimiento mínima.
- **Combinaciones adaptables:** en los procesos por lotes, seleccionan dinámicamente un tipo de combinación en tiempo de ejecución según las filas de entrada reales.
- **Ejecución intercalada:** Usa la cardinalidad real de la función con valores de tabla y múltiples instrucciones detectada en la primera compilación en lugar de una estimación fija.

© JMA 2016. All rights reserved

MANTENIMIENTO, SUPERVISIÓN Y OPTIMIZACIÓN DE RENDIMIENTOS

© JMA 2009. All rights reserved

Introducción

- El objetivo de supervisar bases de datos es evaluar el rendimiento de un servidor.
- Una supervisión eficaz implica tomar instantáneas periódicas del rendimiento actual para aislar procesos que causan problemas y recopilar datos de forma continua a lo largo del tiempo para realizar el seguimiento de las tendencias de rendimiento.
- SQL Server y el sistema operativo Windows proporcionan herramientas que permiten ver las condiciones actuales de la base de datos y realizar un seguimiento del rendimiento a medida que éstas cambian.
- La supervisión de SQL Server permite:
 - Determinar si el rendimiento se puede mejorar.
 - Evaluar la actividad de los usuarios.
 - Solucionar problemas o depurar componentes de aplicaciones.
- La supervisión continua es fundamental puesto que el SQL Server ofrece un servicio en un entorno dinámico. A lo largo del tiempo cambian:
 - Los datos de la aplicación y la distribución de los mismos.
 - Los modelos de negocio sustentados por las bases de datos.
 - La forma de conexión y el tipo de acceso que requieren los usuarios.
 - Los tipos de aplicaciones que tienen acceso a SQL Server.

© JMA 2009. All rights reserved

Conocimientos y habilidades

- Familiaridad con la arquitectura de base de datos
 - La arquitectura de base de datos determina el consumo de recursos.
 - Es necesario conocer como gestiona el almacenamiento, la memoria, los procesos, la concurrencia, las transacciones, ...
 - Así como las estructuras de datos relacionales básicas, las estructuras de almacenamiento y la arquitectura del motor de bases de datos.
- Conocimiento de SQL y Transact SQL
 - Debido a la existencia de herramientas basadas en GUI, es posible crear aplicaciones y administrar una base de datos sin saber SQL.
 - Sin embargo, es imposible ajustar una base de datos o aplicaciones sin saber SQL.
- Familiaridad con las herramientas proporcionados por la base de datos para SQL Tuning

© JMA 2009. All rights reserved

Metodologías

- Preventiva (supervisión)
 - Es una metodología TOP-DOWN
 - Primero se centra en una aplicación ya diseñada y en sentencias SQL
 - Después en la realización de optimización INTERNA (estructuras, I/O)
 - Posibilidad de efectos secundarios
- Reactiva
 - Dirigida a la Bases de Datos en producción
 - Centrada en problemas puntuales de rápida resolución
 - Se realiza después de la preventiva como norma general
- Ambas metodologías no son excluyentes.

© JMA 2009. All rights reserved

Pasos para la supervisión

1. Definir los objetivos concretos de supervisión.
2. Seleccionar la herramienta apropiada.
3. Identificar los componentes que se desea supervisar.
4. Seleccionar métricas para dichos componentes.
5. Tomar medidas.
6. Analizar las medidas obtenidas.
7. Solucionar los problemas detectados o implementar las posibles mejoras.

© JMA 2009. All rights reserved

Componentes a supervisar

- Las áreas siguientes afectan al rendimiento de SQL Server:
 - Recursos del sistema (hardware)
 - Sistema operativo
 - Arquitectura de red
 - Configuración del motor de la base de datos
 - Modelo físico/lógico de las bases de datos
 - Aplicaciones de bases de datos
 - Aplicaciones cliente
- Dentro de los Recursos del sistema, los componentes principales en los que se debe concentrar inicialmente son:
 - Actividad del disco
 - Uso de la memoria
 - Uso del procesador
 - Uso de la red

© JMA 2009. All rights reserved

Cuellos de botella

- Se denomina cuello de botella al fenómeno en donde el rendimiento o capacidad de un sistema completo es severamente limitado por componentes individuales.
- Entre las causas de estos cuellos de botella se incluyen:
 - Recursos que funcionan incorrectamente.
 - Recursos mal configurados.
 - Recursos del mismo tipo que no distribuyen de forma equilibrada las cargas de trabajo; por ejemplo, cuando un recurso monopoliza un disco.
 - Recursos insuficientes que requieren componentes adicionales o actualizados.
- Para detectar los cuellos de botella es necesario realizar un proceso de toma de medidas y su análisis ulterior.

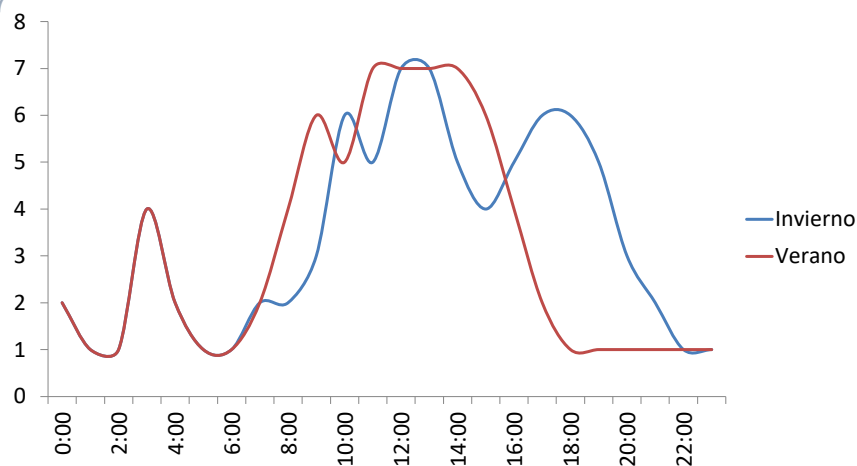
© JMA 2009. All rights reserved

Toma de medidas

- Se denomina toma de medidas al proceso de captura de valores cuantitativos sobre los elementos representativos en diferentes momentos.
- Teniendo en cuenta las diferentes idiosincrasias de los sistemas y de los recursos humanos, es necesario tomar medidas:
 - A diferentes horas del día.
 - Los diferentes días de la semana.
 - A lo largo de todo el mes
 - En diferentes meses
- La toma de medidas es un proceso continuo acumulativo que deben representar las cargas de trabajo habituales que se ejecutan en las bases de datos o sistemas.
- La calidad de las medidas tomadas determina la calidad del análisis.

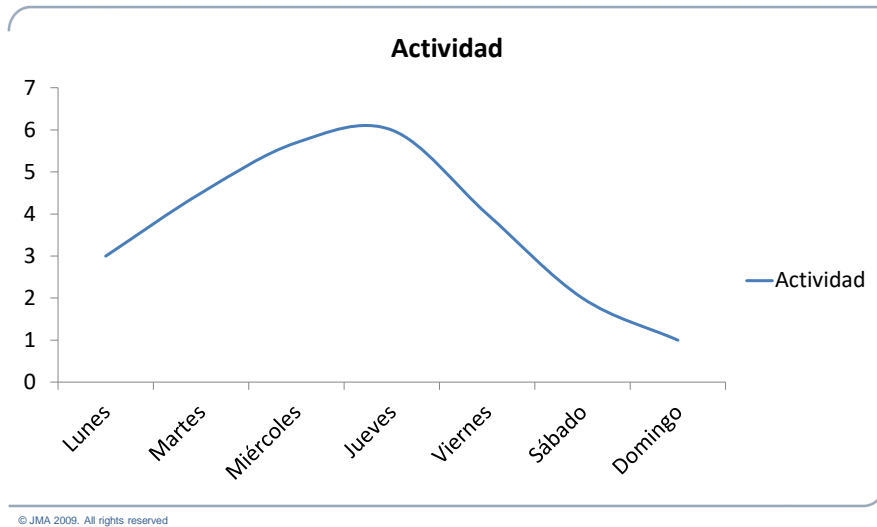
© JMA 2009. All rights reserved

Gráficas diaria

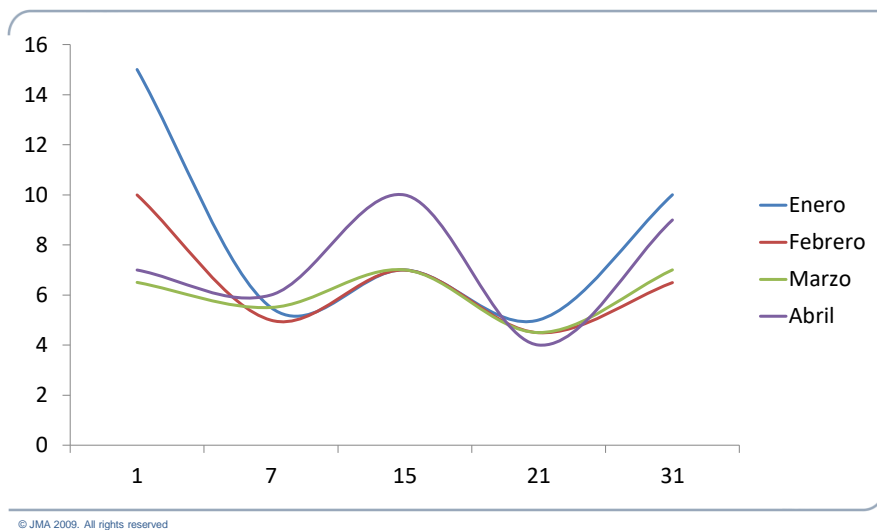


© JMA 2009. All rights reserved

Gráfica semanal



Graficas mensuales



Optimizaciones

- Ajustes en el Sistema Operativo.
- Ajustes en la configuración de SQL Server
 - Memoria, E/S, Conexiones ...
- Optimizar el diseño físico de las bases de datos
 - Ficheros, índices, mantenimiento de integridad, estadísticas, desfragmentación...
- Optimizar el diseño lógico sobre las bases de datos
 - Estructuras de datos, consultas, planes de ejecución ...

© JMA 2009. All rights reserved

Herramientas

- Visor del archivo de registros
- Monitor de sistema
- Monitor de actividad
- Vistas, procedimientos y funciones de sistema en Transact-SQL
- Recopilación de datos
- Utilidad de SQL Server
- Analizador de SQL Server
- Asistente para la optimización
- Regulador de recursos

© JMA 2009. All rights reserved

Visor del archivo de registros

- Se puede usar el Visor del archivo de registros de SQL Server Management Studio para obtener acceso a la información sobre los errores y eventos capturados en los registros siguientes:
 - SQL Server
 - Agente SQL Server
 - Historial de trabajos
 - Recopilación de datos
 - Correo electrónico de base de datos
 - Eventos de Windows (permite integrar los eventos del Visor de eventos de Windows).
 - Registros de auditoría.

© JMA 2009. All rights reserved

Monitor de sistema (S.O.)

- El Monitor de sistema es la utilidad gráfica del sistema operativo que permite supervisar al sistema operativo Windows, principalmente para buscar cuellos de botella.
- Se basa en contadores de rendimiento asociados a los diferentes objetos del sistema, como procesadores, memoria, caché, subprocesos y procesos.
- Cada uno de estos objetos tiene asociado un conjunto de contadores que miden el uso de los dispositivos, la longitud de las colas, las demoras y otros indicadores del rendimiento y la congestión interna.
- La instalación del SQL Server añade un conjunto completo de contadores que permite la supervisión combinada del sistema operativo con el SQL Server.

© JMA 2009. All rights reserved

Contadores

- Disco:
 - PhysicalDisk: Avg. Disk Queue Length
 - PhysicalDisk: Avg. Disk sec/Read, PhysicalDisk: Avg. Disk sec/Write
 - Processor:% Privileged Time
 - SQL Server:Buffer Manager: Page reads/sec.
 - SQL Server:Buffer Manager: Page writes/sec.
 - SQL Server:Buffer Manager: Checkpoint pages/sec.
 - SQL Server:Buffer Manager: Lazy writes/sec.
- Procesador
 - Processor: % Processor Time
 - System: Processor Queue Length
 - SQL Server: Plan Cache
 - SQL Server: SQL Statistics

© JMA 2009. All rights reserved

Contadores

- Memoria:
 - Memory: Page Faults/sec
 - Memory: Available Bytes
 - Memory: Pages/sec
 - SQL Server: Buffer Manager: Buffer Cache Hit Ratio
 - SQL Server: Memory Manager: Total Server Memory
- Red:
 - Network Interface: Bytes Total/sec
 - Network Interface: Current Bandwidth

© JMA 2009. All rights reserved

Monitor de actividad

- El Monitor de actividad muestra información acerca de los procesos de SQL Server y el modo en que estos afectan a la instancia actual de SQL Server.
- Organiza la información en las siguientes áreas:
 - Gráficas con la información general
 - Tareas de usuario activas
 - Esperas de recursos
 - E/S de archivo de datos
 - Consultas costosas recientes.

© JMA 2009. All rights reserved

Recopilación de datos

- SQL Server proporciona un recopilador de datos que se puede utilizar para obtener y guardar datos recopilados de varios orígenes.
- Los conjuntos de recopilación de datos del sistema constan de lo siguiente:
 - Uso de disco: Recopila datos sobre el uso del disco y del registro para todas las bases de datos instaladas en el sistema.
 - Actividad del servidor: Recopila estadísticas de uso de recursos y datos de rendimiento del servidor y SQL Server.
 - Estadísticas de consultas: Recopila estadísticas de consulta, texto de consultas individuales, planes de consulta y consultas concretas.
- El recopilador de datos proporciona un informe histórico para cada uno de los conjuntos de recopilación de datos del sistema:
 - Resumen de uso de disco
 - Historial de estadísticas de consultas
 - Historial de actividad del servidor

© JMA 2009. All rights reserved

Utilidad de SQL Server

- La Utilidad de SQL Server modela las entidades relacionadas con SQL Server de una organización en una vista unificada.
- Los puntos de vista del Explorador de Utilidad y de la Utilidad de SQL Server en SQL Server Management Studio (SSMS) proporcionan a los administradores una vista global del estado de los recursos de múltiples instancias SQL Server a través de una instancia de SQL Server que actúa como punto de control de la utilidad (UCP).
- Disponible a partir de la versión SQL Server 2008R2, el UCP debe ser una instancia Enterprise Edition.
- La combinación del resumen y los datos detallados presentados por el UCP sobre directivas de infrautilización o sobreutilización, y sobre diversidad de parámetros clave, habilita posibilidades de consolidación de recursos y de fácil identificación de sobreutilización.

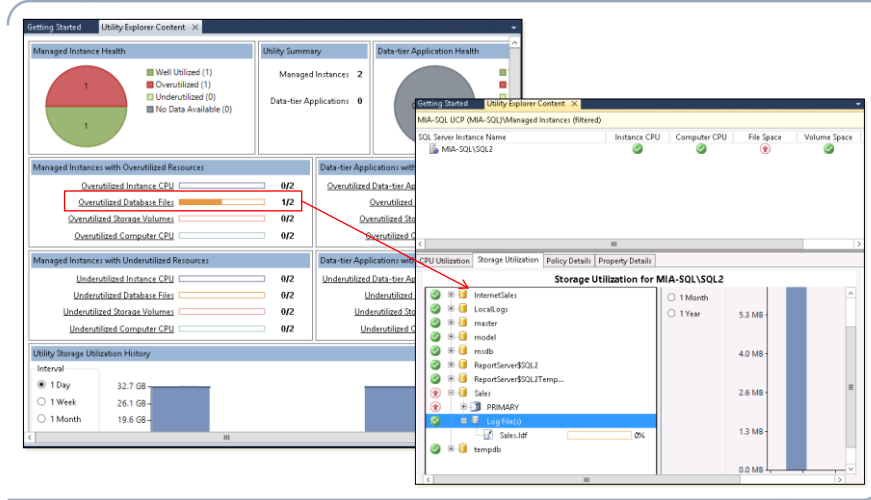
© JMA 2009. All rights reserved

Utilidad de SQL Server

- Las directivas de mantenimiento se pueden configurar y ajustarse para modificar umbrales de uso mayor o menor de los recursos.
- Es posible cambiar las directivas de supervisión globales o configurar directivas de supervisión individuales para cada entidad administrada en la Utilidad SQL Server.
- Se debe deshabilitar la funcionalidad de recopilación de datos de la utilidad de SQL Server mientras la instancia de SQL Server se esté inscribiendo en la utilidad de SQL Server.
- Una vez se haya inscrito la instancia con el UCP, puede reiniciar los conjuntos de recopilación de la utilidad que no sea de SQL Server.
- Se debe tener en cuenta que todos los conjuntos de recopilación en la instancia administrada cargarán sus datos en el almacén de administración de datos de la utilidad (UMDW); el nombre del archivo de datos de la utilidad es sysutility_mdw.
- Una instancia no se puede inscribir en mas de UCP a la vez.

© JMA 2009. All rights reserved

Utilidad de SQL Server



Analizador de SQL Server

- El SQL Server Profiler de Microsoft es una interfaz gráfica de usuario de Seguimiento SQL que se utiliza para supervisar una instancia de Database Engine (Motor de base de datos) o de Analysis Services.
- Puede capturar y guardar datos acerca de cada evento en un archivo o en una tabla para analizarlos posteriormente.
- El SQL Server Profiler muestra el modo en que SQL Server resuelve las consultas internamente.
- Esto permite a los administradores ver exactamente las instrucciones Transact-SQL o las Expresiones multidimensionales que se envían al servidor y cómo el servidor tiene acceso a la base de datos o al cubo para devolver los conjuntos de resultados.
- Mediante el SQL Server Profiler puede hacer lo siguiente:
 - Crear una traza que se base en una plantilla que se puede reutilizar
 - Observar el resultado de la traza a medida que se ejecuta la traza
 - Almacenar el resultado de una traza en una tabla
 - Iniciar, detener, pausar y modificar el resultado de la traza según sea necesario
 - Reproducir el resultado de la traza

Optimizar el diseño físico

- El rendimiento de los sistemas de bases de datos empresariales depende de una configuración eficaz de las estructuras de diseño físico de las bases de datos que componen dichos sistemas.
- Entre las estructuras de diseño físico se incluyen índices, clúster, vistas indizadas y particiones, cuyo objetivo es mejorar el rendimiento y la facilidad de uso de las bases de datos.
- SQL Server proporciona el Asistente para la optimización de motor de base de datos, una herramienta que analiza los efectos del rendimiento de las cargas de trabajo (un conjunto de instrucciones Transact-SQL que se ejecuta en las bases de datos que se desean optimizar) en una o más bases de datos.

© JMA 2009. All rights reserved

Asistente para la optimización

- El Asistente para la optimización de motor de base de datos es una herramienta que analiza el efecto en el rendimiento de las cargas de trabajo ejecutadas en una o varias bases de datos.
- Una carga de trabajo es un conjunto de instrucciones Transact-SQL que se ejecutan en bases de datos que se desean optimizar.
- Después de analizar los efectos de una carga de trabajo en la base de datos, el Asistente proporciona recomendaciones para agregar, eliminar o modificar estructuras de diseño físico de las bases de datos analizadas.
- Estas estructuras de rendimiento físico incluyen índices agrupados, índices no agrupados, vistas indizadas y particiones.
- La implementación de las recomendaciones permiten al procesador de consultas realizar las tareas de carga de trabajo en el menor tiempo posible.

© JMA 2009. All rights reserved

Regulador de recursos

- El Regulador de recursos permite administrar la carga de trabajo y los recursos de SQL Server especificando los límites del consumo de recurso por solicitudes entrantes.
- En el contexto del Regulador, la carga de trabajo es un conjunto de consultas o solicitudes de dimensiones similares que pueden (y deberían) tratarse como una entidad única.
- Cuanto más uniforme es el modelo del uso de recursos de una carga de trabajo, mayores son las ventajas que pueden obtenerse del Regulador de recursos.
- Los límites sobre los recursos pueden reconfigurarse en tiempo real con un impacto mínimo sobre las cargas de trabajo que se están ejecutando.
- En un entorno donde varias cargas de trabajo distintas están presentes en el mismo servidor, el Regulador permite diferenciar estas cargas de trabajo y asignar los recursos compartidos (CPU y memoria) a medida que se soliciten, en función de los límites que se especifiquen.
- El regulador de recursos sólo está disponible en las ediciones Enterprise.

© JMA 2009. All rights reserved

Tipos de problemas de los recursos

- EL Regulador está diseñado para solucionar los siguientes tipos de problemas sobre los recursos que pueden normalmente encontrarse en un entorno de base de datos:
 - Consultas fuera de control en el servidor. En este escenario, una consulta que requiere un uso intensivo de los recursos puede ocupar la mayoría o todos los recursos del servidor.
 - Ejecución imprevisible de la carga de trabajo. En este escenario, las aplicaciones simultáneas en el mismo servidor tienen cargas de trabajo de tamaño y tipo diferente.
 - Establecer la prioridad de la carga de trabajo. En este escenario, se permite que una carga de trabajo se ejecute más rápidamente que otra o se le garantiza la finalización si hay contención de recursos. El Regulador permite asignar una importancia relativa a las cargas de trabajo.
- Todos los escenarios anteriores requieren la capacidad de diferenciar las cargas de trabajo de alguna manera. El Regulador proporciona la capacidad de:
 - Clasificar las conexiones entrantes y enrutar sus cargas de trabajo a un grupo concreto.
 - Supervisar el uso de los recursos para cada carga de trabajo en un grupo.
 - Agrupar recursos y establecer límites específicos del grupo sobre el uso de la CPU y la asignación de memoria. Se evita o minimiza así la probabilidad de consultas fuera de control.
 - Asociar cargas de trabajo agrupadas a un grupo concreto de recursos.
 - Identificar y establecer las prioridades para las cargas de trabajo.

© JMA 2009. All rights reserved

Restricciones del Regulador

- La administración de recursos se limita a SQL Server Database Engine (Motor de base de datos de SQL Server). El Regulador de recursos no se puede utilizar para Analysis Services, Integration Services y Reporting Services.
- No hay ninguna supervisión o administración de las cargas de trabajo entre las instancias de SQL Server.
- La especificación de los límites se refiere al ancho banda de la CPU y a la memoria administradas por SQL Server.
- El Regulador de recursos puede administrar las cargas de trabajo de OLTP pero estos tipos de consultas, que son normalmente muy cortas en duración, no siempre ocupan la CPU el tiempo suficiente como para aplicar los controles de ancho de banda. Este hecho puede sesgar las estadísticas obtenidas para el porcentaje de uso de la CPU.
- Sólo está disponible en las ediciones Enterprise.

© JMA 2009. All rights reserved

Grupos de cargas de trabajo

- Un grupo de cargas de trabajo sirve como contenedor para las solicitudes de sesión que sean similares de acuerdo con los criterios de clasificación que se definen en la función de clasificación.
- Un grupo de cargas de trabajo permite la supervisión agregada del consumo de recursos y la aplicación de una directiva uniforme a todas las solicitudes en el grupo.
- Un grupo define las directivas para sus miembros: Importancia (Baja, Media, y Alta), Nº máximo de solicitudes simultáneas y los máximos por solicitud en: Tiempo de CPU (s) , % de concesión de memoria, Tiempo de espera de concesión (s) y Grado de paralelismo..
- El regulador de recursos predefine dos grupos de cargas de trabajo: el grupo interno (no se puede cambiar solo supervisar) y el grupo predeterminado.
- Las solicitudes se clasifican en el grupo predeterminado cuando se dan las condiciones siguientes:
 - No hay ningún criterio para clasificar una solicitud.
 - Hay un intento de clasificar la solicitud en un grupo inexistente.
 - Hay un error de clasificación general.

© JMA 2009. All rights reserved

Grupos de recursos de servidor

- Un grupo de recursos de servidor o grupo, representa los recursos físicos del servidor (se puede pensar en un grupo como en una instancia virtual de SQL Server dentro de una instancia de SQL Server) y agrupa a los grupos de carga.
- Un grupo esta compuesto por una parte que no se superpone con otros grupos, lo que permite una reserva mínima de recursos, y otra parte que se comparte con otros grupos, lo que permite consumir el consumo máximo de recursos. Por grupo, se puede especificar los siguientes valores porcentuales para cada recurso:
 - MIN y MAX para la CPU
 - MIN y MAX para la memoria
- Grupo interno: representa los recursos utilizados por el SQL Server, este grupo siempre contiene el grupo de carga interno únicamente y no se puede alterar de ninguna forma. El consumo de recurso por el grupo interno no está restringido dado que cualquier carga está considerada como crítica.
- Grupo predeterminado: es el primer grupo de usuario predefinido, es asignado siempre que no se puede clasificar en otro grupo y contiene siempre al grupo de carga predeterminado.
- Grupos de recursos de servidor definidos por el usuario: son asignados en función a la clasificación y contienen sus propios grupos de carga.

© JMA 2009. All rights reserved

Clasificación

- El regulador de recursos admite la clasificación de sesiones de entrada, basada en un conjunto de criterios escritos por el usuario y contenidos en una función.
- Los resultados de la lógica de la función permiten al regulador de recursos clasificar las sesiones en los grupos de cargas de trabajo existentes.
- La función definida por el usuario tiene las siguientes características y comportamientos:
 - Se evalúa para cada nueva sesión.
 - Devuelve NULL, predeterminado o el nombre de un grupo.
 - Se debería definir con ámbito del servidor (base de datos maestra).
 - Debe ser única y no se puede eliminar o modificar mientras se use de clasificador.
 - No recibe parámetros, obtiene el contexto a través de funciones (APP_NAME(), CONNECTIONPROPERTY(), HOST_NAME(), IS_MEMBER(), IS_SRVROLEMEMBER(), LOGINPROPERTY(), ORIGINAL_DB_NAME(), SUSER_NAME(), SUSER_SNAME()) y vistas (sys.dm_exec_connections, sys.dm_exec_sessions, sys.dm_exec_requests) del sistema.

© JMA 2009. All rights reserved

Flujo del Regulador

- Se autentica el inicio de sesión.
- Se ejecuta el desencadenador LOGON.
- Se crea una sesión para la conexión entrante.
- La sesión se clasifica siguiendo la función de clasificación.
- La carga de trabajo de la sesión se enruta a un grupo de cargas de trabajo.
- El grupo de cargas de trabajo utiliza el grupo de recursos de servidor al que está asociado.
- El grupo de recursos de servidor proporciona y limita los recursos requeridos sesión.