

Optimización del Rendimiento de las Consultas

Contenido

Introducción	1
Introducción al optimizador de consultas	2
Obtención de información del plan de ejecución	13
Uso de un índice para abarcar una consulta	25
Estrategias de indización	36
Suplantación del optimizador de consultas	43

Notas para el instructor

Este módulo proporciona a los alumnos conocimientos avanzados acerca del funcionamiento del optimizador de consultas y la obtención de información del plan de ejecución. Se describe cómo crear índices que abarcan consultas, qué estrategias de índices se pueden implementar para reducir las operaciones de entrada y salida (E/S) y si es conveniente suplantar el optimizador de consultas.

En la práctica los alumnos utilizarán el plan de ejecución gráfico, recopilarán información de consultas y verán el resultado del optimizador de consultas. También verán información de índices y la utilizarán para observar cómo el optimizador de consultas optimiza una consulta y aplica las sugerencias del optimizador.

Después de completar este módulo, los alumnos serán capaces de:

- Explicar la función del optimizador de consultas y cómo asegura la optimización de las consultas.
- Utilizar diversos métodos para obtener información del plan de ejecución de forma que puedan determinar cómo se procesó una consulta en el optimizador de consultas y comprobar que se generó el plan de ejecución más efectivo.
- Crear índices que abarcan consultas.
- Identificar estrategias de indización que reduzcan el número de lecturas de página.
- Evaluar cuándo se debe suplantar el optimizador de consultas.

Introducción

Objetivo del tema

Proporcionar una introducción a los temas y objetivos del módulo.

Explicación previa

En este módulo aprenderá cómo el optimizador de consultas utiliza los índices y demás información para determinar cuál es el método más eficaz para tener acceso a los datos.

- Introducción al optimizador de consultas
- Obtención de información del plan de ejecución
- Uso de un índice para abarcar una consulta
- Estrategias de indización
- Suplantación del optimizador de consultas

En este módulo se describe cómo el optimizador de consultas utiliza los índices y demás información para determinar cuál es el método más eficaz para tener acceso a los datos.

Después de completar este módulo, el alumno será capaz de:

- Explicar la función del optimizador de consultas y cómo asegura la optimización de las consultas.
- Utilizar diversos métodos para obtener información del plan de ejecución de forma que pueda determinar cómo se procesó una consulta en el optimizador de consultas y comprobar que se generó el plan de ejecución más efectivo.
- Crear índices que abarcan consultas.
- Identificar estrategias de indización que reduzcan las lecturas de páginas.
- Evaluar cuándo se debe suplantar el optimizador de consultas.

◆ Introducción al optimizador de consultas

Objetivo del tema

Presentar los temas de esta sección.

Explicación previa

En esta sección trataremos el optimizador de consultas y su funcionamiento.

- Función del optimizador de consultas
- Cómo utiliza el optimizador de consultas la optimización basada en el costo
- Cómo funciona el optimizador de consultas
- Fases de la optimización de consultas
- Almacenamiento en caché del plan de ejecución
- Establecimiento de un límite de costo

Conocer el funcionamiento del optimizador de consultas le prepara para crear índices útiles, escribir consultas eficaces y optimizar las consultas que presentan un rendimiento insuficiente.

Función del optimizador de consultas

Objetivo del tema

Presentar el optimizador de consultas

Explicación previa

El optimizador de consultas es el componente encargado de generar el plan de ejecución óptimo para una consulta.

- **Determina el plan de ejecución más eficaz**
 - Determinar la existencia de índices y evaluar su utilidad
 - Determinar los índices o las columnas que se pueden utilizar
 - Determinar cómo procesar las combinaciones
 - Uso de evaluación basada en costos de alternativas
 - Creación de estadísticas de columnas
- **Utiliza información adicional**
- **Produce un plan de ejecución**

El optimizador de consultas es el componente encargado de generar el plan de ejecución óptimo para una consulta.

Determina el plan de ejecución más eficaz

El optimizador de consultas evalúa cada una de las instrucciones de Transact-SQL y decide cuál es el plan de ejecución más eficaz.

Calcula la entrada y salida (E/S) que se necesita para procesar una consulta mediante:

- La determinación de la existencia de índices y la evaluación de su utilidad para una consulta.
- La determinación de los índices o las columnas que se pueden utilizar para reducir el número de filas examinadas por la consulta. Mediante la reducción del número de filas examinadas, se reduce la cantidad de E/S, lo que aumenta el rendimiento de las consultas.
- La determinación de la estrategia más eficaz para procesar las operaciones de combinación, como el orden de combinación de las tablas y la estrategia de combinación que se va a utilizar.
- El uso de evaluación basada en costos de alternativas para seleccionar el plan más eficaz para una consulta determinada.
- La creación de estadísticas de columnas para mejorar el rendimiento de la consulta.

Utiliza información adicional

El optimizador de consultas utiliza información adicional acerca de los datos subyacentes y las estructuras de almacenamiento, el tamaño de los archivos y los tipos de estructuras de los archivos. También utiliza varias de sus propias operaciones internas, como la creación de índices o tablas temporales en la memoria, para mejorar el rendimiento de las consultas.

Produce un plan de ejecución

El optimizador de consultas produce un plan de ejecución que describe la secuencia de pasos necesarios para llevar a cabo una consulta. Optimiza el proceso de búsqueda, combinación, agrupación y ordenación de las filas.

Cómo utiliza el optimizador de consultas la optimización basada en el costo

Objetivo del tema

Describir la optimización basada en el costo.

Explicación previa

El optimizador de consultas se basa en el costo, lo que significa que calcula el costo de ejecutar un plan de ejecución para evaluarlo.

- **Limita el número de planes de optimización**
 - El costo se estima en términos de E/S y costo de CPU
- **Determina el tiempo de procesamiento de las consultas**
 - Utilice operadores físicos y secuencia de operaciones
 - Utilice procesos paralelos y en serie

El optimizador de consultas se basa en el costo, lo que significa que calcula el costo de ejecutar un plan de ejecución para evaluarlo.

Nota El cálculo de los costos sólo puede ser tan preciso como permitan los datos estadísticos disponibles acerca de las columnas, índices y tablas.

Limita el número de planes de optimización

Para llevar a cabo la ejecución en una cantidad de tiempo razonable, el optimizador de consultas limita el número de planes de optimización que tiene en cuenta. Mediante la evaluación de las secuencias de las operaciones relacionales necesarias para generar el conjunto de resultados, el optimizador de consultas crea un plan de ejecución que tiene el costo más bajo calculado en lo que se refiere a la E/S y a la pérdida de recursos de la CPU.

Determina el tiempo de procesamiento de las consultas

El rendimiento de las consultas está determinado por los operadores físicos que utiliza el optimizador de consultas y la secuencia de procesamiento de las operaciones. El objetivo es reducir:

- El número de filas devueltas.
- El número de páginas leídas.
- El tiempo de procesamiento global mediante la reducción del uso de recursos de E/S y CPU para un plan de ejecución.

Cuando el optimizador optimiza las consultas, no inicia el plan de ejecución con la menor pérdida de recursos sino que elige el plan de ejecución que devuelve los resultados al usuario de la forma más rápida con una reducción de recursos razonable.

Nota Si Microsoft® SQL Server™ 2000 tiene varios procesadores disponibles, el optimizador de consultas puede dividir la consulta entre ellos. Las consultas de duración prolongada suelen beneficiarse de planes de ejecución paralelos, pero una consulta en paralelo puede utilizar, en general, más recursos que el procesamiento de una consulta en serie.

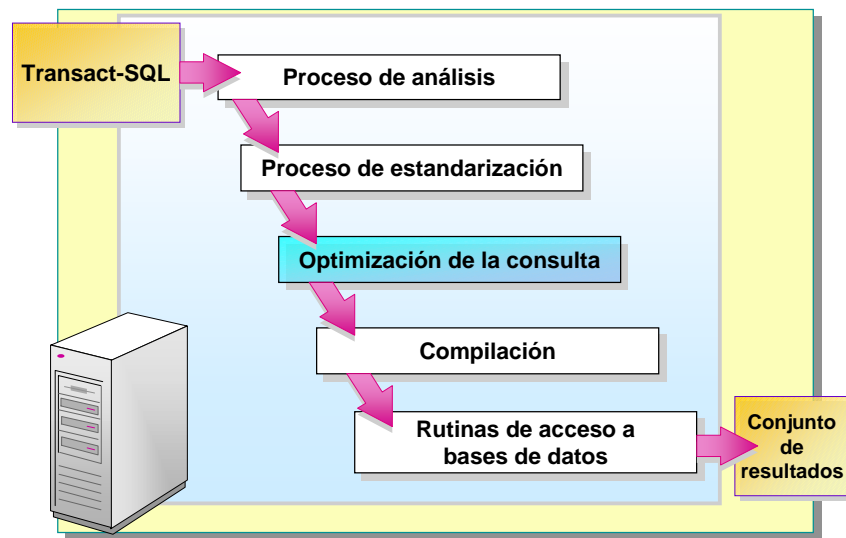
Cómo funciona el optimizador de consultas

Objetivo del tema

Presentar cómo procesa el optimizador de consultas una consulta después de que se emite a SQL Server.

Explicación previa

Después de emitir una consulta, tienen lugar varios pasos que transforman la consulta original en un formato que el optimizador de consultas puede interpretar.



Sugerencia

Explique cómo contribuye cada paso al proceso de transformación de la consulta original en un formato que puede interpretar el optimizador de consultas.

Después de emitir una consulta, tienen lugar varios pasos que transforman la consulta original en un formato que el optimizador de consultas puede interpretar.

Proceso de análisis

El proceso de análisis comprueba la sintaxis de la consulta entrante y divide la sintaxis en partes a las que puede responder el motor de base de datos relacional. El resultado de este paso es un árbol de la consulta analizada.

Proceso de estandarización

El proceso de estandarización convierte una consulta a un formato útil para la optimización. Las cláusulas de sintaxis redundantes que se detectan se quitan. Se estandarizan las subconsultas, si es posible. El resultado de este paso es un árbol de la consulta estandarizada.

Optimización de la consulta

El proceso de selección de un plan de ejecución entre varios posibles se denomina *optimización*. En esta fase están implicados numerosos pasos. Sin embargo, los pasos siguientes son los que tienen mayor efecto en el costo del plan de ejecución: análisis de la consulta, selección de índices y selección de la combinación.

Compilación

La consulta se compila en código ejecutable.

Rutinas de acceso a bases de datos

El optimizador de consultas determina el mejor método para tener acceso a los datos, mediante la ejecución de un recorrido de tabla o mediante el uso de un índice disponible. Después se aplica el mejor método.

Fases de la optimización de consultas

Objetivo del tema

Presentar las fases por las que atraviesa la optimización de consultas.

Explicación previa

El proceso de optimización de consultas consta de tres fases.

■ Análisis de la consulta

- Identifica los criterios de búsqueda y combinación de la consulta

■ Selección de índices

- Determina si existe algún índice
- Produce una valoración de la utilidad del índice o índices

■ Selección de la combinación

- Evalúa qué estrategia de combinación se va a utilizar

El proceso de optimización de consultas consta de tres fases. Estas fases no son pasos de procesamiento distintos y sólo se utilizan para representar conceptualmente la actividad interna del optimizador de consultas.

Análisis de la consulta

La primera fase de optimización de las consultas se denomina *análisis de la consulta*. En esta fase, el optimizador identifica los criterios de búsqueda y combinación de la consulta. Mediante la limitación de la búsqueda, el optimizador de consultas reduce al mínimo el número de filas que se procesan. La reducción del número de filas procesadas reduce el número de páginas de índices y datos leídas.

Selección de índices

La selección de índices es la segunda fase de la optimización de las consultas. Durante esta fase, el optimizador de consultas detecta si existe un índice para las cláusulas identificadas. Después, se produce una valoración de la utilidad del índice o índices. La utilidad de un índice está determinada por la cantidad de filas que se devolverán. Esta información se obtiene de las estadísticas del índice o las estadísticas de la columna. Se realiza un cálculo del costo de diversos métodos de acceso mediante el cálculo de las lecturas de páginas físicas y lógicas necesarias para buscar las filas correctas.

Selección de la combinación

La selección de la combinación es la tercera fase de la optimización de las consultas. Si hay una consulta de varias tablas o autocombinación, se evalúa la estrategia de combinación que se va a utilizar. La determinación de la estrategia de combinación que se va a utilizar consiste en considerar una serie de factores: selectividad, densidad y memoria requerida para procesar la consulta.

Almacenamiento en caché del plan de ejecución

Objetivo del tema

Describir cómo se administran los planes de ejecución en la caché y cómo la reducción de compilaciones del plan puede mejorar el rendimiento.

Explicación previa

SQL Server tiene un bloque de memoria que se utiliza para almacenar los planes de ejecución y los búferes de datos.

- **Almacenamiento de un plan de ejecución en la memoria**
 - Una copia para todas las ejecuciones en serie
 - Otra copia para todas las ejecuciones en paralelo
- **Uso de un contexto de ejecución**
 - Se vuelve a utilizar un plan de ejecución existente, si lo hay
 - Se genera un plan de ejecución nuevo, si no existe uno
- **Recompilación de planes de ejecución**
 - Los cambios en la base de datos pueden hacer que el plan de ejecución sea ineficaz o deje de ser válido

SQL Server tiene un bloque de memoria que se utiliza para almacenar los planes de ejecución y los búferes de datos. El porcentaje del bloque de memoria asignado a los planes de ejecución o los búferes de datos varía dinámicamente, en función del estado del sistema. La parte del bloque de memoria que se utiliza para almacenar los planes de ejecución se denomina *caché de procedimientos*.

Almacenamiento de un plan de ejecución en la memoria

La mayor parte del plan de ejecución es una estructura de datos reutilizable y de sólo lectura que puede ser utilizada por un número cualquiera de usuarios. No se almacena ningún contexto de usuario en el plan de ejecución. Nunca hay más de dos copias del plan de ejecución en la memoria:

- Una copia para todas las ejecuciones en serie.
- Otra copia para todas las ejecuciones en paralelo.

La copia en paralelo cubre todas las ejecuciones en paralelo, con independencia de su grado de paralelismo.

Uso de un contexto de ejecución

Cada usuario que ejecuta una consulta tiene una estructura de datos que contiene los datos específicos de una ejecución como son los valores de los parámetros. Esta estructura de datos se denomina *contexto de ejecución*. Cuando se ejecuta una instrucción de Transact-SQL, SQL Server explora la caché de procedimientos para determinar si hay un plan de ejecución para esa instrucción.

- Si lo hay, SQL Server lo vuelve a utilizar.
Así se ahorra el trabajo de volver a compilar la instrucción de Transact-SQL.
- Si no hay un plan de ejecución, SQL Server genera uno nuevo para la consulta.

Recompilación de planes de ejecución

Ciertos cambios en una base de datos pueden hacer que un plan de ejecución sea ineficaz o deje de ser válido. Cuando SQL Server detecta cambios que invalidan un plan de ejecución, lo marca como no válido. Se compila un nuevo plan de ejecución para la siguiente conexión que ejecute la consulta.

Importante Se puede mejorar el rendimiento si se reduce el número de veces que se vuelve a compilar un plan.

Algunas condiciones que invalidan un plan de ejecución son:

- Los cambios estructurales realizados en una tabla o vista a la que hace referencia la consulta (instrucciones ALTER TABLE y ALTER VIEW).
- Generación de nuevas estadísticas de distribución, bien de forma explícita a partir de una instrucción, como en UPDATE STATISTICS, o automáticamente.
- Eliminación de un índice usado por el plan de ejecución.
- Una llamada explícita al procedimiento almacenado del sistema **sp_recompile**.
- Gran cantidad de cambios en las claves, o instrucciones INSERT o DELETE, en una tabla a la que hace referencia la consulta.
- En las tablas con desencadenadores, un aumento considerable del número de filas en las tablas inserted o deleted.

Nota SQL Server utiliza un algoritmo de caducidad para administrar con eficacia los planes de ejecución de la caché y evalúa el costo y el uso del plan de ejecución.

Establecimiento de un límite de costo

Objetivo del tema

Presentar el regulador de consultas.

Explicación previa

Puede ser conveniente establecer un límite para controlar el costo de ejecutar una consulta.

■ Especificación de un límite máximo

- Utilice el regulador de consultas para evitar que se ejecuten consultas que tarden mucho tiempo y usen muchos recursos del sistema

■ Especificación de límites de conexión

- Use el procedimiento almacenado `sp_configure`
- Ejecute la instrucción `SET QUERY_GOVERNOR_COST_LIMIT`
- Especifique 0 para desactivar el regulador de consultas

Puede ser conveniente establecer un límite para controlar el costo de ejecutar una consulta. El término *costo de la consulta* se refiere al cálculo del tiempo transcurrido, en segundos, necesario para ejecutar una consulta en una configuración de hardware específica.

Especificación de un límite máximo

Puede utilizar la opción **query governor cost limit** para evitar que se ejecuten consultas que tarden mucho tiempo y usen muchos recursos del sistema.

De forma predeterminada, las consultas tienen permiso para ejecutarse, sin importar el tiempo que tarden. El regulador de consultas utiliza un cálculo del costo para evitar que se ejecuten consultas con un costo elevado.

Aunque el valor de configuración se especifica en segundos, no se corresponde realmente con el tiempo, sino con el costo real calculado de la consulta. Puede especificar un límite máximo para el costo de la consulta que se va a ejecutar.

Como el regulador de consultas se basa en el cálculo del costo de la consulta, en lugar del tiempo real transcurrido, no se procesa en tiempo de ejecución. Si el costo calculado de una consulta es mayor que el límite especificado, la instrucción del regulador de consultas impide que se ejecute la consulta. Esto es más eficaz que permitir la ejecución de una consulta hasta que se alcanza un límite definido previamente y, después, detenerla.

Especificación de límites de conexión

Se pueden especificar límites para todas las conexiones o sólo las consultas de una conexión determinada. Para aplicar los límites de costo del regulador de consultas, puede:

- Utilizar el procedimiento almacenado **sp_configure** para aplicar límites a todas las conexiones.

Puede cambiar el límite de costo del regulador de consultas sólo cuando **show advanced options** está configurado a 1. La configuración tiene efecto inmediato. No tiene que detener y reiniciar el servidor.

- Ejecutar la instrucción **SET QUERY_GOVERNOR_COST_LIMIT** para aplicar límites para una conexión determinada.
- Especificar 0 (predeterminado) para desactivar el regulador de consultas. En este caso, todas las consultas se ejecutan sin límites.

◆ Obtención de información del plan de ejecución

Objetivo del tema

Presentar los temas de esta sección.

Explicación previa

Se puede obtener información acerca del plan de ejecución mediante tres métodos.

- Presentación de los resultados de las instrucciones STATISTICS
- Presentación de los resultados de SHOWPLAN_ALL y SHOWPLAN_TEXT
- Presentación gráfica del plan de ejecución

El optimizador de consultas responde a la información que tiene disponible al determinar el mejor plan de ejecución. Se puede obtener información acerca del plan de ejecución mediante la consulta de la tabla **sysindexes**. También se puede obtener información mediante las instrucciones STATISTICS, las instrucciones SHOWPLAN y la presentación gráfica del plan de ejecución.

Presentación de los resultados de las instrucciones STATISTICS

Objetivo del tema

Describir la presentación de los resultados estadísticos mediante las instrucciones STATISTICS.

Explicación previa

Se pueden utilizar las instrucciones STATISTICS IO, STATISTICS TIME y STATISTICS PROFILE para obtener información que ayude a diagnosticar consultas de ejecución prolongada.

Instrucción		Ejemplo de salida	
STATISTICS TIME		SQL Server Execution Times: CPU time = 0 ms, elapsed time = 2 ms.	
STATISTICS PROFILE		Rows Executes StmtText StmtId... ----- 47 1 SELECT * FROM [charge] 16 WHERE (([charge_amt]>=@1) . . .	
STATISTICS IO		Table 'member'. Scan count 1, logical reads 23, physical reads 0, read-ahead reads 0.	

Se pueden utilizar las instrucciones STATISTICS IO, STATISTICS TIME y STATISTICS PROFILE para obtener información que ayude a diagnosticar consultas de ejecución prolongada. Los resultados de las instrucciones STATISTICS proporcionan información acerca del plan de ejecución real.

STATISTICS TIME obtiene información acerca del número de milisegundos necesarios para analizar, compilar y ejecutar cada instrucción.

STATISTICS PROFILE muestra la información de perfil de una instrucción. Al ejecutar una consulta, el conjunto de resultados incluye los resultados de la instrucción SHOWPLAN_ALL y dos columnas adicionales. En la tabla siguiente se muestran las columnas adicionales.

Columna	Descripción
Rows	Número real de filas que produce cada operador
Executes	Número de veces que se ejecutó el operador

STATISTICS IO obtiene información acerca de la cantidad de lecturas de páginas generadas por las consultas. Los resultados de STATISTICS IO incluyen los valores de la tabla siguiente.

Valor	Descripción	Información adicional
Lecturas lógicas	Número de páginas leídas de la caché de datos	Se tiene acceso a todas las páginas en la caché de datos. Si una página no está disponible en la caché, se debe leer físicamente del disco.
Lecturas físicas	Número de páginas leídas del disco	Este valor siempre es menor o igual que el valor de lecturas lógicas. El método que se utiliza para calcular el valor de Proporción de aciertos de caché es el siguiente:
$\text{Proporción de aciertos de caché} = \frac{\text{Lecturas lógicas}}{\text{Lecturas lógicas} - \text{Lecturas físicas}}$		
Lecturas anticipadas	Número de páginas colocadas en la caché	Un número alto para este valor significa que el valor de lecturas físicas es menor y la proporción de aciertos de caché es mayor que si no estuviera habilitada la lectura anticipada.
Número de exploraciones	Número de veces que se tuvo acceso a la tabla	Las tablas externas de una combinación izquierda siempre deben tener un número de exploraciones igual a 1. Para las tablas internas, el número de lecturas lógicas está determinado por el número de exploraciones multiplicado por el número de páginas a las que se tiene acceso en cada exploración.

Nota Las instrucciones SET permanecen vigentes durante la sesión hasta que se especifica la opción OFF o hasta que se termina la sesión.

Presentación de los resultados de SHOWPLAN_ALL y SHOWPLAN_TEXT

Objetivo del tema

Describir el uso de las instrucciones SHOWPLAN.

Explicación previa

Se puede obtener información detallada acerca de cómo se ejecutan las consultas y cuántos recursos se requieren para procesarlas mediante las instrucciones SET SHOWPLAN_TEXT y SET SHOWPLAN_ALL.

■ Estructura de los resultados de la instrucción SHOWPLAN

- Devuelven información en forma de un conjunto de filas
- Forman un árbol jerárquico
- Representan los pasos realizados por el optimizador de consultas
- Muestran un cálculo de cómo se optimizó una consulta, no el plan de ejecución real

■ Detalles de los pasos de ejecución

■ Diferencia entre los resultados de SHOWPLAN_TEXT y SHOWPLAN_ALL

Se pueden utilizar las instrucciones SET SHOWPLAN_TEXT y SET SHOWPLAN_ALL para obtener información detallada acerca de cómo se ejecutan las consultas y cuántos recursos se requieren para procesarlas.

Estructura de los resultados de la instrucción SHOWPLAN

Los resultados de la instrucción SHOWPLAN:

- Devuelven información en forma de un conjunto de filas.
- Forman un árbol jerárquico.
- Representan los pasos realizados por el optimizador de consultas para ejecutar cada instrucción.
- Muestran un cálculo de cómo se optimizó una consulta, no el plan de ejecución real. El cálculo se basa en las estadísticas existentes.

Detalles de los pasos de ejecución

Cada instrucción que se refleja en los resultados contiene una única fila con el texto de la instrucción, seguido por varias filas con los detalles de los pasos de ejecución. Algunos detalles de los pasos de ejecución son:

- Los índices que se utilizan con cada tabla.
- El orden de combinación de las tablas.
- La selección del modo de actualización.
- Las tablas de trabajo y demás estrategias.

Diferencia entre los resultados de SHOWPLAN_TEXT y SHOWPLAN_ALL

La diferencia entre los resultados de SHOWPLAN_TEXT y SHOWPLAN_ALL es que SHOWPLAN_ALL devuelve información adicional, como el cálculo de filas, E/S, CPU y promedio de tamaño de las filas de la consulta.

Nota Las instrucciones SET permanecen vigentes durante la sesión hasta que se especifica la opción OFF o hasta que se termina la sesión.

◆ Presentación gráfica del plan de ejecución

Objetivo del tema

Presentar los temas de este apartado.

Explicación previa

Se puede utilizar el Analizador de consultas SQL para ver gráficamente un plan de ejecución codificado con colores.

- Elementos del plan de ejecución gráfico
- Lectura de los resultados del plan de ejecución gráfico
- Uso de la operación de consulta de marcadores

Se puede utilizar el Analizador de consultas SQL para ver gráficamente un plan de ejecución codificado con colores.

Elementos del plan de ejecución gráfico

Objetivo del tema

Presentar los elementos del plan de ejecución gráfico.

Explicación previa

En el plan de ejecución gráfico, que contiene los elementos siguientes, se utilizan iconos para representar la ejecución de partes específicas de las instrucciones y consultas.

- Los *pasos* son unidades de trabajo que se utilizan para procesar una consulta
- La *secuencia de pasos* es el orden en que se procesan los pasos
- Los *operadores lógicos* describen la operación algebraica relacional que se utiliza para procesar una instrucción
- Los *operadores físicos* describen el algoritmo de implementación física que se utiliza para procesar una instrucción










Sugerencia

Repase brevemente el icono, el operador físico y la descripción que utiliza el optimizador de consultas. A continuación se ofrece una lista de los operadores físicos.

En el plan de ejecución gráfico, que contiene los elementos siguientes, se utilizan iconos para representar la ejecución de partes específicas de las instrucciones y consultas:

- Los *pasos* son unidades de trabajo que se utilizan para procesar una consulta.
- La *secuencia de pasos* es el orden en que se procesan los pasos.
- Los *operadores lógicos* describen la operación algebraica relacional que se utiliza para procesar una instrucción; por ejemplo, llevar a cabo una agregación. En general, el operador lógico coincide con el operador físico. No todos los pasos que se realizan para procesar una consulta o en operaciones de actualización implican operaciones lógicas.
- Los *operadores físicos* describen el algoritmo de implementación física que se utiliza para procesar una instrucción; por ejemplo, explorar un índice agrupado. Cada paso de la ejecución de una consulta o una operación de actualización incluye un operador físico.

La tabla siguiente contiene una lista parcial de los operadores físicos que se utilizan para representar los algoritmos que utiliza el optimizador de consultas.

Icono	Operador físico	Descripción del operador
	Consulta de marcadores	Utiliza un marcador (Id. de fila o clave de agrupación) para consultar la fila correspondiente en la tabla o índice agrupado
	Filtro	Examina la entrada y sólo devuelve las filas que cumplen la expresión del filtro que aparece en la columna argument
	Coincidencia hash	Crea una tabla <i>hash</i> mediante el cálculo de un valor <i>hash</i> para cada fila de información de generación
	Recorrido de índice	Recupera todas las filas del índice no agrupado especificado en la columna argument
	Búsqueda en índice	Utiliza la capacidad de búsqueda de los índices para recuperar filas de un índice no agrupado
	Combinación de mezcla	Realiza todos los tipos de combinaciones (excepto autocombinación y combinación cruzada), incluidas las operaciones UNION
	Bucles anidados	Busca en la tabla interna cada una de las filas de la tabla externa, generalmente mediante un índice
	Orden	Ordena todas las filas entrantes
	Recorrido de tabla	Recupera todas las filas de la tabla especificada en la columna argument

Nota Para ver la lista completa de iconos y obtener más información, busque en “mostrar gráficamente el plan de ejecución mediante el Analizador de consultas SQL” en los Libros en pantalla de SQL Server.

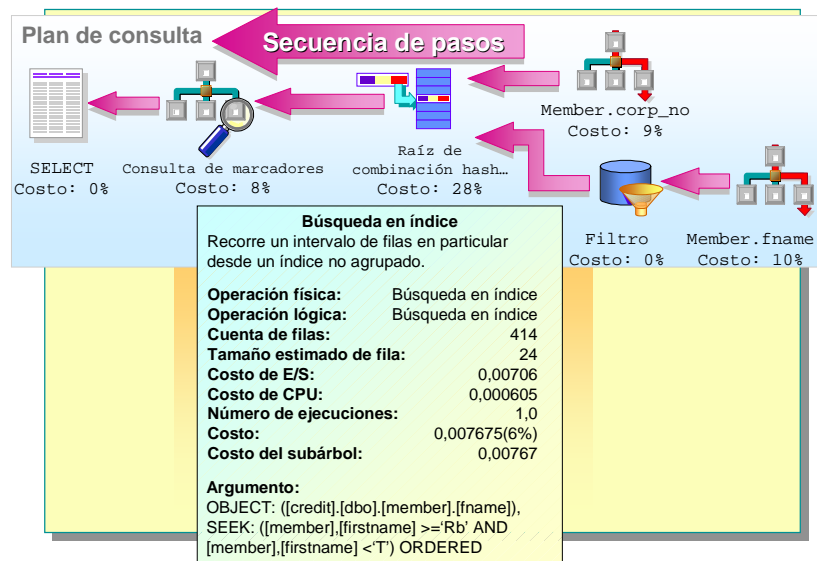
Lectura de los resultados del plan de ejecución gráfico

Objetivo del tema

Describir cómo se leen los resultados del plan de ejecución.

Explicación previa

Los resultados del plan de ejecución gráfico se leen de derecha a izquierda y de arriba a abajo. Se muestra cada consulta del lote que se analiza; el costo de cada consulta se incluye como porcentaje del costo total del lote.



Sugerencia

Mediante el Analizador de consultas SQL, active Mostrar plan de ejecución y, después, ejecute una consulta.

En los resultados del plan de ejecución, coloque el puntero en un icono para mostrar la información adicional acerca de esa operación específica.

Los resultados del plan de ejecución gráfico se leen de derecha a izquierda y de arriba a abajo. Se muestra cada consulta del lote que se analiza, lo que incluye el costo de cada consulta como porcentaje del costo total del lote.

Cada paso puede tener uno o varios nodos para procesar. El término *nodo* se refiere a una operación utilizada por el optimizador de consultas que se representa mediante un icono.

El plan de ejecución puede tener varios nodos para un paso determinado.

- Cada nodo está relacionado con un nodo principal.
- Todos los nodos con el mismo nodo principal se especifican en la misma columna.
- Cada nodo se conecta con su nodo principal mediante puntas de flecha.
- Las operaciones recursivas se muestran con un símbolo de iteración.
- Los operadores se muestran como símbolos relacionados con un principal específico.
- Si el lote contiene varias instrucciones, se crean varios planes de ejecución.

Sugerencia

Describe brevemente los tipos de información detallada que se puede ver.

Presentación de información adicional

Al colocar el puntero en cada nodo (representado mediante un icono), se puede ver información detallada acerca de los operadores físicos y lógicos, además de la información de la tabla siguiente.

Medidas	Descripción
Cuenta de filas	El número de filas devueltas por el operador.
Tamaño estimado de fila	El tamaño estimado de la fila devuelta por el operador.
Costo de E/S	El costo estimado de toda la actividad de E/S de la operación. Este valor debe ser lo más bajo posible.
Costo de CPU	El costo calculado de toda la actividad de CPU de la operación.
Número de ejecuciones	El número de veces que se ejecutó la operación durante la consulta.
Costo	El costo para el optimizador de consultas al ejecutar la operación, que incluye el costo de esta operación como porcentaje del costo total de la consulta.
Costo del subárbol	El costo total para el optimizador de consultas al ejecutar esta operación y todas las operaciones anteriores en el mismo subárbol.
Argumento	Los predicados y parámetros utilizados por la consulta.

Uso de la operación de consulta de marcadores

Objetivo del tema

Describir cómo el optimizador de consultas utiliza la operación de consulta de marcadores.

Explicación previa

La consulta de marcadores es un operador interno utilizado por el optimizador de consultas.

- **Análisis del plan de consultas**
 - Normalmente se utiliza después de haber procesado todos los pasos
- **Recuperación de filas**
 - Identificadores de fila
 - Claves de agrupación
- **Observación de los detalles**
 - Se utiliza una etiqueta de marcador para buscar la fila
- **Determinación de cuándo se utiliza el operador de consulta de marcadores**
 - Las consultas contienen la cláusula IN o el operador OR

La *consulta de marcadores* es un operador interno que utiliza con frecuencia el optimizador de consultas. Cuando el optimizador de consultas identifica registros que son candidatos posibles para el conjunto de resultados que se pretende, anota la información que identifica la ubicación de las filas (marcador) y continúa las operaciones que afinan la búsqueda.

Si se incluye una fila en la búsqueda, SQL Server utiliza la ubicación de la fila en el marcador para buscarla mediante el análisis del plan de consultas, recuperación de las filas, observación de los detalles y determinación de cuándo se utiliza el operador de consulta de marcadores.

Análisis del plan de consultas

En el plan de consultas, el optimizador de consultas utiliza, generalmente, el operador de consulta de marcadores después de que se hayan procesado los demás pasos.

Recuperación de filas

El operador de consulta de marcadores recupera todas las filas apropiadas mediante:

- Un identificador de fila (RID) para buscar la fila correspondiente en un montón.
- La clave de agrupación para buscar la fila correspondiente en un índice agrupado.

Observación de los detalles

En el plan de la consulta, los detalles del operador de consulta de marcadores contienen:

- Una etiqueta de marcador que se utiliza para buscar la fila en la tabla o índice agrupado.
- El nombre de la tabla o el nombre del índice agrupado donde se encuentra la fila.
- La cláusula WITH PREFETCH si el optimizador de consultas determina que la lectura anticipada es la mejor manera de buscar marcadores en la tabla o el índice agrupado.

Determinación de cuándo se utiliza el operador de consulta de marcadores

En general, el optimizador de consultas utiliza el operador de consulta de marcadores para procesar las consultas que contienen la cláusula IN y los operadores OR en la cláusula WHERE.

Ejemplo

En este ejemplo la tabla **member** tiene un índice no agrupado en la columna **member_no**. El optimizador de consultas utiliza un operador de consulta de marcadores para recuperar las filas correspondientes.

```
USE credit
SELECT *
FROM member
WHERE member_no
IN (4567,8765,4321)
```

◆ Uso de un índice para abarcar una consulta

Objetivo del tema

Presentar los temas de esta sección.

Explicación previa

Se pueden crear índices que satisfagan los criterios de la consulta sin tener acceso a las páginas de datos.

- Introducción a los índices que abarcan consultas
- Búsqueda de datos mediante índices que abarcan consultas
- Identificación de si se puede utilizar un índice para abarcar una consulta
- Determinación de si se utiliza un índice para abarcar una consulta
- Instrucciones para la creación de índices que abarcan consultas

Se pueden crear índices que satisfagan los criterios de la consulta sin tener acceso a las páginas de datos. Esta estrategia puede mejorar el rendimiento de las consultas.

Introducción a los índices que abarcan consultas

Objetivo del tema

Presentar el concepto de índices que abarcan consultas.

Explicación previa

Al crear índices, puede que desee crear un índice que abarque las consultas más comunes con el fin de reducir la cantidad de E/S.

- Sólo los índices no agrupados pueden abarcar consultas
- Los índices deben contener todas las columnas a las que se hace referencia en la consulta
- No es necesario el acceso a las páginas de datos
- Las vistas indizadas pueden agregar datos anteriores
- Los índices que abarcan consultas recuperan los datos rápidamente

Al crear índices, puede que desee crear un índice que abarque las consultas más comunes con el fin de reducir la cantidad de E/S.

Sólo los índices no agrupados pueden abarcar consultas

Los índices que abarcan consultas contienen todos los datos necesarios de una consulta en las hojas de un índice no agrupado.

Los índices deben contener todas las columnas a las que se hace referencia en la consulta

Un índice que abarca una consulta debe contener todas las columnas a las que se hace referencia en la instrucción `SELECT`. Si hay un índice agrupado, los campos de la clave de agrupación están en las hojas del índice no agrupado y contribuyen a abarcar la consulta.

No es necesario el acceso a las páginas de datos

Cuando un índice abarca una consulta, el optimizador de consultas no tiene acceso a las páginas de datos porque todos los datos necesarios están contenidos en el índice. La cantidad de E/S se reduce considerablemente.

Las vistas indizadas pueden agregar datos anteriores

Si una vista indizada suma, cuenta o crea un promedio de columnas, el optimizador de consultas puede utilizar esta vista para proporcionar valores almacenados al resolver una consulta. Las vistas indizadas que agregan datos anteriores pueden mejorar notablemente el rendimiento.

Los índices que abarcan consultas recuperan los datos rápidamente

La creación de índices que abarcan consultas es una de las maneras más rápidas de tener acceso a los datos, especialmente para consultas de baja selectividad.

Al comparar las hojas de los índices agrupados y de los no agrupados, la ventaja de tener índices que abarcan consultas es evidente.

Tipo de índice	Contenido del nivel de hoja
-----------------------	------------------------------------

Agrupado	Fila entera (páginas de datos reales)
----------	---------------------------------------

No agrupado	Valor de clave
-------------	----------------

Como los valores de clave tienen, generalmente, un tamaño más pequeño que las filas reales, una página de índice puede almacenar más valores de clave que filas completas. El almacenamiento de valores de clave requiere menos páginas, lo que reduce la cantidad de E/S.

◆ Búsqueda de datos mediante índices que abarcan consultas

Objetivo del tema

Presentar los temas de este apartado.

Explicación previa

El optimizador de consultas explora las hojas de diferentes maneras cuando se puede utilizar un índice para abarcar una consulta.

- Ejemplo de exploración de una sola página
- Ejemplo de exploración de recorrido parcial
- Ejemplo de exploración de recorrido completo

El optimizador de consultas explora las hojas de diferentes maneras cuando se puede utilizar un índice para abarcar una consulta. Abarcar una consulta puede consistir en leer una sola página, un intervalo de páginas o todas las páginas de las hojas. Nunca se tiene acceso a las páginas de datos.

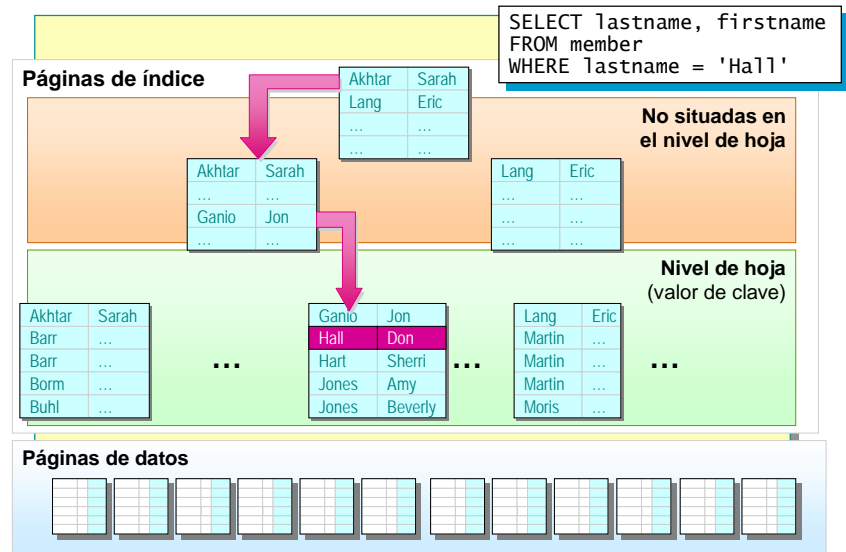
Ejemplo de exploración de una sola página

Objetivo del tema

Ilustrar la exploración de una sola página de un índice.

Explicación previa

En este ejemplo una consulta que abarca un índice requiere la lectura de una sola página de una hoja.



La exploración de una sola página se produce cuando se lee sólo una de las páginas de una hoja en el nivel que no es de hoja. Leer una página es similar a una consulta puntual, donde la información (una única fila o varias filas) se encuentra en una sola página.

Nota La exploración de una sola página no significa que la consulta sólo pueda devolver una fila. Una consulta puntual puede devolver una fila o todas las filas de una página. En cualquier caso, todos los datos se encuentran en una sola página.

Ejemplo

En este ejemplo, un índice compuesto y no agrupado en las columnas **lastname** y **firstname** abarca la consulta.

```
SELECT lastname, firstname
FROM member
WHERE lastname = 'Hall'
```

Más información

Para simplificar, en la diapositiva no se muestra el puntero del nivel de hoja del índice no agrupado a las páginas de datos (montón o índice agrupado).

SQL Server ejecuta los pasos siguientes para obtener la información:

1. Recorre el árbol del índice y compara el apellido Hall con los valores de clave.
2. Continúa recorriendo el índice hasta que alcanza la primera página del nivel de hoja que contiene el valor de clave Hall.
3. Devuelve las filas correspondientes sin tener acceso a las páginas de datos porque los valores de clave **lastname** y **firstname** están contenidos en el nivel de hoja.

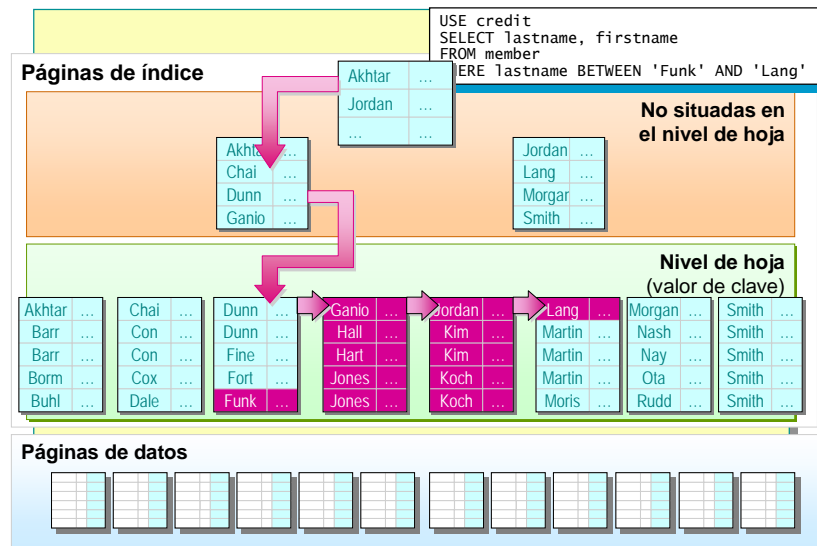
Ejemplo de exploración de recorrido parcial

Objetivo del tema

Ilustrar la exploración de recorrido parcial de un índice.

Explicación previa

En este ejemplo una consulta que abarca un índice requiere la lectura de una serie de páginas de nivel de hoja.



Un recorrido parcial se produce cuando se lee un intervalo de páginas del nivel de hoja.

Ejemplo

En este ejemplo un índice compuesto no agrupado en las columnas **lastname** y **firstname** abarca la consulta mediante un recorrido parcial de las páginas de nivel de hoja.

```
USE credit
SELECT lastname, firstname
FROM member
WHERE lastname BETWEEN 'Funk' AND 'Lang'
```

Para su información

La columna **firstname** se omite para simplificar la diapositiva. Consulte la diapositiva anterior en caso de confusión.

SQL Server ejecuta los pasos siguientes para obtener la información:

1. Recorre el árbol del índice.
2. Comienza a leer las páginas de nivel de hoja en la página que contiene la primera aparición del apellido Funk.
Los datos del nivel de hoja están ordenados de forma ascendente.
3. Lee el intervalo de páginas del nivel de hoja hasta el apellido Lang.
En este momento, el recorrido parcial se completa.
4. Devuelve las filas correspondientes sin tener acceso a las páginas de datos porque se recorre el nivel de hoja en busca de los apellidos entre Funk y Lang.

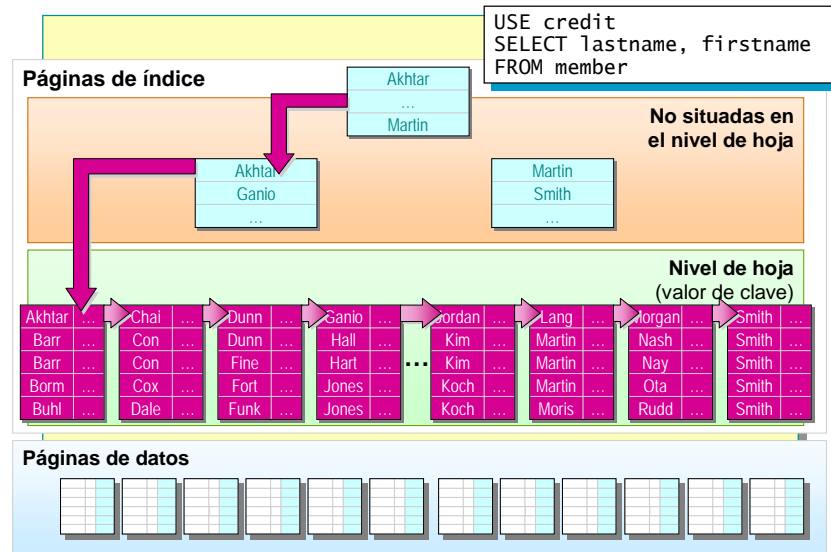
Ejemplo de exploración de recorrido completo

Objetivo del tema

Ilustrar la exploración de recorrido completo de un índice.

Explicación previa

En este ejemplo una consulta que abarca un índice requiere la lectura de todas las páginas de nivel de hoja.



Un recorrido completo se produce cuando se leen todas las páginas del nivel de hoja. De manera similar a un recorrido de tabla, un recorrido completo se produce cuando una consulta no incluye una cláusula **WHERE** o cuando la cláusula **WHERE** no es selectiva.

Ejemplo

En este ejemplo un índice compuesto no agrupado en las columnas **lastname** y **firstname** abarca la consulta mediante un recorrido completo de las páginas de nivel de hoja.

```
USE credit
SELECT lastname, firstname
FROM member
```

Para su información

Esta diapositiva ilustra un recorrido de los datos ordenados por **lastname**. Si se solicitan datos, sin ordenar u ordenados por **firstname**, SQL Server puede utilizar las páginas de asignación para identificar y recorrer todas las páginas del índice y, después, rechazar las que no estén situadas en las hojas. La columna **firstname** se omite para simplificar la diapositiva.

Para recuperar la información SQL Server hace lo siguiente:

1. Recorre el árbol del índice.
2. Lee las páginas de las hojas, comenzando por la primera página, y las recorre todas hasta que llega a la última.
3. Devuelve las filas correspondientes sin tener acceso a las páginas de datos porque se recorre el nivel de hoja.

Nota El recorrido del nivel de hoja de un índice también es un recorrido de datos paralelo. SQL Server utiliza el procesamiento de lectura anticipada para mejorar el rendimiento de la consulta.

Identificación de si se puede utilizar un índice para abarcar una consulta

Objetivo del tema

Indicar cuándo el optimizador de consultas puede utilizar un índice para abarcar una consulta.

Explicación previa

Estos factores afectan a la capacidad de un índice para abarcar una consulta.

- Todos los datos necesarios deben estar en el índice
- Un índice compuesto es útil aunque no se haga referencia a la primera columna
- No es necesaria una cláusula WHERE
- Se puede utilizar un índice no agrupado si requiere menos E/S que un índice agrupado que contenga una columna a la que se hace referencia en la cláusula WHERE
- Los índices se pueden combinar para abarcar consultas

Estos factores afectan a la capacidad de un índice para abarcar una consulta:

- Todos los datos necesarios deben estar en el índice. Estos datos incluyen todas las columnas a las que se hace referencia, con independencia de si se devuelven en el conjunto de resultados, se utilizan para la ordenación o agregación, o se suministran en la cláusula WHERE.
- Una columna de un índice puede contribuir a abarcar una consulta incluso cuando no sea la primera columna a la que se hace referencia en un índice compuesto.

Por ejemplo, un índice compuesto en **SalesRep, Region, Amount** (en este orden) puede abarcar una consulta que sólo haga referencia a **Region** y **SUM(Amount)**.

- No es necesaria una cláusula WHERE. El optimizador de consultas recorre el nivel de hoja completo.
- Un índice no agrupado se puede utilizar para abarcar una consulta si requiere menos E/S que un índice agrupado que contiene una columna a la que se hace referencia en la cláusula WHERE.
- Los índices se pueden combinar para abarcar consultas. Si algunas tablas o todas a las que se hace referencia en una operación de combinación tienen un índice que abarcar una consulta, los resultados se combinan mediante una operación de combinación especial y, después, se devuelven las filas.

Determinación de si se utiliza un índice para abarcar una consulta

Objetivo del tema

Explicar cómo se determina si una consulta está abarcada por un índice.

Explicación previa

Las consultas que están abarcadas por un índice no son aparentes explícitamente para los usuarios. Se puede observar el plan de ejecución gráfico o comparar la E/S para determinar si el optimizador de consultas utilizó un índice para abarcar una consulta.

■ Observación de los resultados del plan de ejecución

- Se muestra la frase “Comprobar un índice no agrupado, en su totalidad o sólo un intervalo”

■ Comparación de E/S

- Índice no agrupado
 - Número total de niveles no situados en el nivel de hoja
 - Número total de páginas que componen el nivel de hoja
 - Número total de filas por página del nivel de hoja
 - Número total de filas por página de datos
- Número total de páginas que componen la tabla

Las consultas que están abarcadas por un índice no son aparentes explícitamente para los usuarios. Se puede observar el plan de ejecución gráfico o comparar la E/S para determinar si el optimizador de consultas utilizó un índice para abarcar una consulta.

Observación de los resultados del plan de ejecución

Se puede ver el plan de ejecución gráficamente. Si los resultados de un plan de ejecución muestran la frase “Comprobar un índice no agrupado, en su totalidad o sólo un intervalo”, el optimizador de consultas pudo abarcar la consulta mediante un índice.

Comparación de E/S

También se pueden ver los resultados de STATISTICS IO. Al evaluar el costo de un índice que abarca una consulta, recuerde que el optimizador de consultas siempre intenta abarcar la consulta al evaluar un plan de ejecución.

Como ayuda para determinar si la consulta está abarcada por un índice, debe conocer la información siguiente acerca del índice no agrupado y la tabla:

- Índice no agrupado
 - Número total de niveles no situados en el nivel de hoja
 - Número total de páginas que componen el nivel de hoja
 - Número total de filas por página del nivel de hoja
 - Número total de filas por página de datos
- Número total de páginas que componen la tabla

Nota Si lo prefiere, también puede calcular el tamaño del nivel de hoja de un índice no agrupado en lugar de utilizar la instrucción STATISTICS IO. Como alternativa, puede consultar **sysindexes** y revisar la columna **dpages**, que mostrará el tamaño del nivel de hoja.

Instrucciones para la creación de índices que abarcan consultas

Objetivo del tema

Proporcionar algunas instrucciones para crear índices que abarquen consultas.

Explicación previa

Al crear índices que abarquen consultas, tenga en cuenta las instrucciones siguientes.

- **Agregar columnas a los índices**
- **Reduzca el tamaño de la clave del índice**
- **Mantenga una proporción de tamaño entre la fila y la clave**

Al crear índices que abarquen consultas, tenga en cuenta las instrucciones siguientes:

- **Agregar columnas a los índices.** Puede que desee agregar columnas a índices que:
 - Abarcan más de una consulta.
 - Contribuyen a resolver algunas de las consultas más comunes.
 - Se hace referencia a ellas con frecuencia.
 - No aumentan considerablemente el tamaño de la clave.
- **Reduzca el tamaño de la clave del índice.** Al definir la clave del índice (valores de clave), evite especificar valores de clave demasiado grandes. Las filas grandes aumentan el tamaño de la fila, el número de niveles de índice y el número total de páginas. Las ventajas de rendimiento obtenidas con la creación de un índice que abarca consultas se reducirían.
- **Mantenga una proporción de tamaño entre la fila y la clave.** El aumento del tamaño de la clave del índice en relación con el tamaño de las filas puede afectar al rendimiento de la consulta. Un ejemplo extremo es la creación de un índice no agrupado en todas las columnas de una tabla. Al hacer esto, se genera una copia virtual de la tabla y se almacena en el nivel de hoja del índice no agrupado de forma ordenada.

Sugerencia

Mencione que las columnas que pueden ser muy grandes en una ocasión pueden ser aceptables en otra.

Pregunte a los alumnos qué número de caracteres es demasiado grande para optimizar una tabla.

◆ Estrategias de indización

Objetivo del tema

Presentar los temas de esta sección.

Explicación previa

Se puede implementar estrategias de indización para mejorar el rendimiento de las consultas.

- Evaluación de la E/S para las consultas que tienen acceso a un intervalo de datos
- Indización para varias consultas
- Instrucciones para la creación de índices

Se puede implementar estrategias de indización para mejorar el rendimiento de las consultas.

Evaluación de la E/S para las consultas que tienen acceso a un intervalo de datos

Objetivo del tema

Ilustrar las diferencias en la E/S de páginas mediante distintos métodos de acceso.

Explicación previa

El optimizador de consultas tiene en cuenta automáticamente varios planes de ejecución y calcula la E/S necesaria para cada plan.

```
SELECT charge_no
FROM charge
WHERE charge_amt BETWEEN 20 AND 30
```

<i>Método de acceso</i>	<i>Página de E/S</i>
Recorrido de tabla	10.417
Índice agrupado en la columna charge_amt	1042
Índice no agrupado en la columna charge_amt	100.273
Índice compuesto en las columnas charge_amt y charge_no	273

Sugerencia

El ejemplo no se puede ejecutar en la base de datos credit.

Consulte los supuestos en el cuaderno de trabajo al comparar la E/S de páginas para los distintos índices.

El optimizador de consultas tiene en cuenta automáticamente varios planes de ejecución y calcula la E/S necesaria para cada plan. Después, inicia un plan de ejecución con la menor cantidad de E/S además de tener en cuenta otras consideraciones. Compare la E/S de páginas entre los distintos métodos de acceso que puede utilizar el optimizador de consultas.

Por ejemplo, considere la siguiente consulta que recupera un intervalo de datos y, a continuación, compare la E/S de esta consulta con otros métodos de acceso a los datos.

```
SELECT charge_no
FROM charge
WHERE charge_amt BETWEEN 20 AND 30
```

Sugerencia

Señale que los métodos de acceso ilustrados en la diapositiva utilizan esta información.

Al comparar los distintos métodos debe suponer lo siguiente:

- Hay 1 millón de filas y 96 filas por página.
- El número total de páginas es de 10.147.
- No hay ningún índice agrupado.
- Hay 100.000 filas en el intervalo de 20,00 a 30,00 dólares.
- En una página de hoja de índice no agrupado entran 367 filas de índice.

Recorrido de tabla

La realización de un recorrido de tabla es beneficiosa para las consultas en las que el conjunto de resultados incluye un porcentaje alto de una tabla (baja selectividad). Los recorridos de tabla son apropiados cuando el total de E/S de páginas de una consulta sobrepasaría el número de páginas de la tabla.

Al ejecutar la consulta que recorre una tabla, el número de operaciones de E/S de páginas es de 10.417. Compare la E/S de páginas en el recorrido de una tabla con un índice no agrupado en la columna **charge_amt**. Un recorrido de tabla es más eficaz.

Índice agrupado en la columna charge_amt

Para obtener la información, SQL Server ejecuta los pasos siguientes:

1. Busca en el índice agrupado el valor mínimo, en este caso 20,00.
2. Lee las filas comenzando en 20,00 y detiene la búsqueda en 30,00.

Como la columna **charge_amt** está agrupada, el orden físico de los datos se organiza según la cantidad que cobrar. Todos los datos que pertenecen a ese intervalo están en orden secuencial en las páginas siguientes, con lo que resulta fácil recuperarlos. El número de operaciones de E/S de páginas es de 1.042 (100.000/96 filas por página).

Índice no agrupado en la columna charge_amt

SQL Server ejecuta los pasos siguientes para obtener la información:

1. Busca el intervalo de valores en las hojas del índice no agrupado y recupera el RID de cada fila. En este caso, se tiene acceso a 273 páginas de nivel de hoja (100.000/367).
2. Los datos se recuperan de cada página mediante la consulta de marcadores para las filas correspondientes.

El número de operaciones de E/S de páginas es, aproximadamente, de 100.273. Recuperar los datos mediante un índice no agrupado en la columna **charge_amt** es el método menos eficaz porque SQL Server debe leer una página por cada fila y, *además*, se leen las hojas del índice para recuperar los valores RID. Cada página de datos se lee varias veces en la caché.

Índice compuesto en las columnas charge_amt y charge_no

El número de operaciones de E/S de páginas es de 273 (100.000/367 filas por página). El promedio de filas de índice por nivel de hoja es de 367. Como las columnas **charge_amt** y **charge_no** están en el índice, SQL Server no busca en las páginas de datos, lo que reduce la cantidad de E/S.

Indización para varias consultas

Objetivo del tema

Ilustrar los problemas de crear índices para resolver las consultas más importantes.

Explicación previa

Elegir el índice más apropiado en función de una consulta individual es más fácil que crear un índice para consultas con varias prioridades.

Ejemplo 1

```
USE credit
SELECT charge_no, charge_dt, charge_amt
FROM charge
WHERE statement_no = 19000 AND member_no = 3852
```

Ejemplo 2

```
USE credit
SELECT member_no, charge_no, charge_amt
FROM charge
WHERE charge_dt between '07/30/1999'
AND '07/31/1999' AND member_no = 9331
```

Sugerencia

Utilice los ejemplos de la diapositiva y consulte la tabla del cuaderno de trabajo al comparar el rendimiento de ambas consultas con diferentes estrategias de indización.

Elegir el índice más apropiado en función de una consulta individual es más fácil que crear un índice para consultas con varias prioridades. La creación de índices para consultas con varias prioridades es más compleja porque el mejor índice para una consulta puede no serlo para otra. El objetivo es lograr un rendimiento aceptable para todas las consultas con la prioridad más alta mediante la evaluación de la E/S.

Situación de ejemplo en una empresa

En los ejemplos siguientes, se supone que las consultas más comunes solicitadas por los usuarios son la búsqueda de cobros a clientes para un informe específico (ejemplo 1) y la búsqueda de cobros a clientes para un día específico (ejemplo 2). La primera consulta de ejemplo es el 15 por ciento de la tabla. La otra consulta es muy selectiva y sólo tiene acceso a unas pocas filas.

Ejemplo 1

```
USE credit
SELECT charge_no, charge_dt, charge_amt
FROM charge
WHERE statement_no = 19000 AND member_no = 3852
```

Ejemplo 2

```
USE credit
SELECT member_no, charge_no, charge_amt
FROM charge
WHERE charge_dt between '07/30/1999' AND '07/31/1999'
AND member_no = 9331
```

En la tabla siguiente se compara el rendimiento de las consultas de los ejemplos 1 y 2 en función de la posible estrategia de indización que se implemente. La mejor estrategia es usar un índice agrupado en la columna **member_no**.

Tipo de índice	Columna	Consulta del ejemplo 1	Consulta del ejemplo 2
Agrupado	member_no	Muy rápida.	Muy rápida.
No agrupado	charge_no	Utiliza el índice agrupado.	Utiliza el índice agrupado.
Agrupado	charge_no	Más lenta que si se creara un índice agrupado en la columna member_no .	Lenta. El índice no agrupado en member_no no es eficaz con intervalos de datos.
No agrupado	member_no		
Agrupado	member_no	Muy rápida.	Muy rápida.
No agrupado, compuesto	statement_no, member_no	Utiliza el índice agrupado.	Utiliza el índice agrupado.
Agrupado	charge_no	Más lenta que si se creara un índice agrupado en la columna member_no .	Rápida.
No agrupado, compuesto	member_no, charge_dt		Un índice compuesto aumenta considerablemente el rendimiento del índice no agrupado.

Instrucciones para la creación de índices

Objetivo del tema

Proporcionar instrucciones para la creación de índices útiles.

Explicación previa

Para asegurar que los índices que se crean son útiles para el optimizador de consultas, tenga en cuenta las instrucciones siguientes.

- Determine las prioridades de todas las consultas
- Determine la selectividad de cada parte de la cláusula WHERE de cada consulta
- Determine si es conveniente crear un índice
- Identifique las columnas que se deben indizar
- Determine el mejor orden para las columnas de los índices compuestos
- Determine los demás índices que sean necesarios
- Pruebe el rendimiento de las consultas

La decisión acerca de la cantidad de índices, el tipo y las columnas en las que crearlos debe basarse en el conocimiento completo de los datos y las necesidades de los usuarios.

Para asegurar que los índices que se crean son útiles para el optimizador de consultas, tenga en cuenta las instrucciones siguientes:

- Determine las prioridades de todas las consultas.
 - Examine y conozca totalmente los datos y cómo se utilizarán.
 - Determine las transacciones de prioridad para la base de datos.
- Determine la selectividad de cada parte de la cláusula WHERE de cada consulta.
- Determine si es conveniente crear un índice.

Hay situaciones en las que no se debe crear un índice. Algunas son las siguientes:

- Si el optimizador de consultas nunca utiliza el índice.
- Si los valores de las columnas son de baja selectividad.
- Si la columna que se va a indizar es demasiado ancha.
- Identifique las columnas que se deben indizar.
 - Cree un índice en una columna que se utilice como clave de combinación para mejorar el rendimiento de la combinación. Así, el optimizador de consultas tendrá la opción de utilizar un índice en lugar de realizar un recorrido de tabla.
 - Evalúe si se busca con frecuencia en la columna.
 - Asegúrese de que se indizan las columnas a las que se hace referencia en las cláusulas WHERE de las consultas con prioridad más alta.
- Determine el mejor orden para las columnas de los índices compuestos.

- Determine los demás índices que sean necesarios.
 - Determine el número mínimo de índices que se pueden crear para cada tabla.
 - Compare el beneficio de rendimiento del índice con el mantenimiento de actualización.
 - Si una consulta se ejecuta con poca frecuencia, puede ser conveniente crear un índice para una actividad determinada (cuando puede proporcionar un beneficio de rendimiento significativo) y, después, eliminarlo.
- Pruebe el rendimiento de las consultas.

Después de crear los índices, pruebe el rendimiento de las consultas con prioridad más alta mediante la ejecución de las instrucciones siguientes para cada consulta:

- SET SHOWPLAN ON
- SET STATISTICS IO ON
- SET STATISTICS TIME ON

◆ Suplantación del optimizador de consultas

Objetivo del tema

Presentar los temas de esta sección.

Explicación previa

Esta sección describe las formas de suplantar el optimizador de consultas.

- Determinación de cuándo se debe suplantar el optimizador de consultas
- Uso de las sugerencias y la instrucción SET FORCEPLAN
- Comprobación del rendimiento de las consultas después de suplantar el optimizador de consultas

En esta sección se describen las formas de suplantar el optimizador de consultas y cómo determinar cuándo se debe hacer. Cuando se suplanta el optimizador de consultas, es importante probar y asegurar el rendimiento de las consultas.

Determinación de cuándo se debe suplantar el optimizador de consultas

Objetivo del tema

Señalar alternativas para suplantar el optimizador de consultas.

Explicación previa

En general, no es aconsejable suplantar el optimizador de consultas.

- Limitar las sugerencias de optimizador
- Antes de suplantar el optimizador de consultas debe explorar las demás alternativas mediante:
 - Actualización de estadísticas
 - Recompilación de procedimientos almacenados
 - Revisión de las consultas o argumentos de búsqueda
 - Evaluación de la posibilidad de crear índices diferentes

Si las consultas no se ejecutan con eficacia, tiene la posibilidad de suplantar el optimizador de consultas mediante *sugerencias de optimizador*. Las sugerencias de optimizador son palabras clave que se incluyen en la consulta para forzar una operación de optimización específica.

Se debe limitar el uso de sugerencias de optimizador porque convierten la optimización en estática. Las sugerencias de optimizador evitan que el optimizador de consultas se ajuste a un entorno dinámico. Después de utilizar las sugerencias de optimizador debe supervisar constantemente el rendimiento de las consultas para comprobar que la consulta se ejecuta de forma óptima.

Antes de considerar la suplantación del optimizador de consultas debe explorar las demás alternativas mediante:

- Actualización de estadísticas.
- Recompilación de procedimientos almacenados.
- Revisión de las consultas o argumentos de búsqueda para determinar si se deben volver a escribir.
- Evaluación de la posibilidad de crear índices diferentes.

Uso de las sugerencias y la instrucción SET FORCEPLAN

Objetivo del tema

Tratar la suplantación del optimizador de consultas.

Explicación previa

Se puede suplantarse el optimizador de consultas mediante las sugerencias de optimizador o la instrucción SET FORCEPLAN.

- Sugerencias de tabla
- Sugerencias de combinación
- Sugerencias de consulta
- Instrucción SET FORCEPLAN

Se puede suplantarse el optimizador de consultas mediante las sugerencias o la instrucción SET FORCEPLAN. Se puede especificar una sugerencia de optimizador de consultas en las instrucciones SELECT, INSERT, UPDATE o DELETE. Se pueden utilizar tres tipos de sugerencias para suplantarse el optimizador de consultas.

Sugerencias de tabla

Una *sugerencia de tabla* especifica un recorrido de tabla, uno o varios índices que debe utilizar el optimizador de consultas, o bien un método de bloqueo que debe utilizar el optimizador de consultas con esta tabla y para una instrucción. Al utilizar las sugerencias de tabla, tenga en cuenta lo siguiente:

- Cada sugerencia de tabla sólo se puede especificar una vez, aunque se pueden tener varias sugerencias de tabla
- La cláusula WITH se debe especificar junto al nombre de la tabla

Sintaxis

nombreTabla [[AS] *aliasTabla*] [WITH (< sugerenciaTabla > [,...n])]

```
WITH ( < sugerenciaTabla > ) ::=
{ INDEX ( valorÍndice [ ,...n ] )
| FASTFIRSTROW
| HOLDLOCK
| NOLOCK
| PAGLOCK
| READCOMMITTED
| READPAST
| READUNCOMMITTED
| REPEATABLEREAD
| ROWLOCK
| SERIALIZABLE
| TABLOCK
| TABLOCKX
| UPDLOCK
| XLOCK
}
```

Sugerencias de combinación

Las sugerencias de combinación exigen una estrategia de combinación entre dos tablas. Se especifican en la cláusula FROM de una consulta. Cuando se especifica una sugerencia de combinación, el optimizador de consultas exige automáticamente el orden de combinación para todas las tablas combinadas en la consulta, en función de la posición de las palabras clave ON.

Sintaxis

```
< sugerenciaCombinación > ::=
{ LOOP | HASH | MERGE | REMOTE }
```

Sugerencias de consulta

Las sugerencias de consulta controlan una amplia variedad de acciones. Se puede especificar que el optimizador de consultas utilice una sugerencia determinada para una consulta mediante la cláusula OPTION. Al utilizar la cláusula OPTION, tenga en cuenta los hechos siguientes:

- Cada sugerencia de consulta sólo se puede especificar una vez, aunque se pueden tener varias sugerencias de consulta.
- La cláusula OPTION se debe especificar con la consulta más externa de la instrucción.
- La sugerencia de consulta afecta a todos los operadores de la instrucción.
- Si se incluye el operador UNION en la consulta principal, sólo la última consulta que incluya el operador UNION puede tener la cláusula OPTION.

Sugerencia

Recuerde a los alumnos que el operador UNION aumenta el número de filas, mientras que las operaciones de combinación aumentan el número de columnas.

Sintaxis

```
[ OPTION ( < sugerenciaConsulta > [ ,...n ] )  
  < sugerenciaConsulta > ::=  
    { { HASH | ORDER } GROUP  
    | { CONCAT | HASH | MERGE } UNION  
    | { LOOP | MERGE | HASH } JOIN  
    | FAST númeroFilas  
    | FORCE ORDER  
    | MAXDOP número  
    | ROBUST PLAN  
    | KEEP PLAN  
    | KEEPFIXED PLAN  
    | EXPAND VIEWS  
    }
```

Instrucción SET FORCEPLAN

Mediante la cláusula FROM puede hacer que el optimizador de consultas combine las tablas en el orden en que se enumeran. Al utilizar la instrucción SET FORCEPLAN, el optimizador de consultas sólo utiliza combinaciones de bucle anidado.

La instrucción SET FORCEPLAN es una opción de sesión.

Sintaxis

```
SET FORCEPLAN {ON | OFF}
```

Nota Si una o varias sugerencias de consulta causan que el optimizador de consultas no genere un plan de ejecución válido, SQL Server cancela la ejecución y emite el mensaje de error 8622. Debe volver a emitir la consulta sin especificar ninguna sugerencia de optimizador ni utilizar la instrucción SET FORCEPLAN.

Comprobación del rendimiento de las consultas después de suplantar el optimizador de consultas

Objetivo del tema

Describir la importancia de probar y volver a evaluar el rendimiento de las consultas *después* de suplantar el optimizador de consultas.

Explicación previa

Si determina la necesidad de suplantar el optimizador de consultas, pruebe y vuelva a evaluar el rendimiento de las consultas.

- Compruebe que mejora el rendimiento
- Documente las razones para utilizar sugerencias de optimizador
- Vuelva a probar las consultas regularmente

Si determina que la suplantación del optimizador de consultas es necesaria, compruebe que el rendimiento ha mejorado, documente las razones para suplantar el optimizador de consultas y vuelva a probar las consultas periódicamente.

Compruebe que mejora el rendimiento

Para comprobar que las sugerencias del optimizador de consultas mejorarán el rendimiento, especifique la opción ON para las instrucciones STATISTICS IO y STATISTICS TIME y seleccione **Mostrar plan de ejecución** en el analizador de consultas. En la mayor parte de los casos, suplantar el optimizador de consultas no mejora el rendimiento.

Si pasa valores de entrada en un procedimiento almacenado, compruebe que no se compromete el rendimiento para *ninguno* de los valores de entrada. Las sugerencias de optimizador pueden mejorar el rendimiento para ciertos valores de entrada, pero pueden comprometerlo para otros.

Documente las razones para utilizar sugerencias de optimizador

Si la suplantación del optimizador de consultas mejora el rendimiento, asegúrese de documentar las razones. Esto le permite volver a evaluar periódicamente la validez de las sugerencias de optimizador. Si las circunstancias cambian, es posible que las sugerencias de optimizador ya no sean necesarias y puedan comprometer el rendimiento.

Vuelva a probar las consultas regularmente

El optimizador de consultas es dinámico y evalúa constantemente el mejor plan de ejecución cuando los datos cambian. Si utiliza sugerencias de optimizador, el plan de ejecución se vuelve estático. Por este motivo, debe volver a probar, de forma periódica, las consultas para las que suplanta el optimizador de consultas.