

Práctica B: Análisis de consultas que utilizan distintas estrategias de combinación

Objetivos

Después de realizar esta práctica, podrá evaluar cómo el optimizador de consultas procesa una consulta mediante estrategias de combinación de bucle anidado, mezcla y hash.

Requisitos previos

- El archivo lab073.zip contiene los archivos de comandos necesarios para la realización de la práctica, así como las correspondientes soluciones.

Para la realización de las prácticas es necesario ejecutar el script Restore.cmd del directorio Restore.

Ejercicio 1

Procesamiento de combinaciones de bucle anidado

En este ejercicio creará índices en las tablas **member** y **charge**, y observará cómo el optimizador de consultas procesa la consulta mediante una estrategia de combinación de bucle anidado.

Puede abrir, revisar y ejecutar secciones del archivo de comandos `NestedLoopJoin.sql` o escribir y ejecutar las instrucciones Transact-SQL proporcionadas.

✍ Para crear índices

En este procedimiento, va a crear tres índices en las tablas **member** y **charge**.

1. Abra el Analizador de consultas SQL y, si así se le pide, conéctese al servidor (local) con autenticación de Microsoft Windows.
2. Con el Analizador de consultas SQL, escriba y ejecute esta instrucción para eliminar los índices existentes en las tablas **member** y **charge**:

```
USE credit
EXEC index_cleanup member
EXEC index_cleanup charge
```

3. Escriba y ejecute esta instrucción para crear un índice compuesto no agrupado y único en la columna **lastname** y **firstname** de la tabla **member**:

```
USE credit
CREATE UNIQUE nonclustered INDEX mbr_name
    ON member(lastname, firstname)
```

4. Escriba y ejecute esta instrucción para crear un índice no agrupado en la columna **member_no** de la tabla **charge**:

```
USE credit
CREATE nonclustered INDEX chg_mem_no
    ON charge(member_no)
```

✍ Para observar cómo se procesa una consulta mediante una estrategia de combinación de bucle anidado

En este procedimiento, activará la opción de estadísticas, ejecutará una consulta y registrará la información estadística.

1. Escriba y ejecute esta instrucción para activar la opción de estadísticas:

```
SET STATISTICS IO ON
```

2. En la ventana Consulta, en el menú **Consulta**, haga clic en **Mostrar plan de ejecución**.

3. Escriba y ejecute esta instrucción SELECT para recuperar el número de miembro, el apellido y el número de cargo de los miembros que tengan el apellido Barr y su nombre sea Bos.

```
USE credit
SELECT member.member_no, lastname, charge_no
FROM member JOIN charge
ON member.member_no = charge.member_no
WHERE member.lastname = 'BARR'
AND firstname = 'BOS'
```

4. Registre la información estadística de la tabla **member**.

Información	Resultado
Número de exploraciones	1
Número de lecturas lógicas	3
Número y nombre de los índices utilizados para procesar la consulta	1 (mbr_name)

5. Registre la información estadística de la tabla **charge**.

Información	Resultado
Número de exploraciones	1
Número de lecturas lógicas	26
Número y nombre de los índices utilizados para procesar la consulta	1 (chg_mem_no)

¿Por qué la tabla **member** tiene tres operaciones de E/S mientras que la tabla **charge** requiere 26?

Para una fila de la tabla member hay varias filas en la tabla charge. Esto muestra que tiene una relación uno a varios estándar.

6. Haga clic en el botón **Mostrar plan de ejecución estimado** para mostrar el plan de ejecución de forma gráfica.
7. Examine el plan de ejecución.

¿Qué estrategia utilizó el optimizador de consultas para buscar las filas en ambas tablas?

Combinación de bucle anidado/interna. El optimizador de consultas utilizó el índice no agrupado para buscar el nombre BOS BARR en la tabla member. Una vez encontrado el miembro correcto, utilizó la columna member_no para buscar las filas coincidentes en el índice chg_mem_no de la tabla charge. Se trata de la combinación de bucle anidado/interna.

Ejercicio 2

Procesamiento de combinaciones de mezcla

En este ejercicio eliminará todos los índices de las tablas **member** y **charge**, ejecutará una consulta y evaluará el plan de ejecución. A continuación, creará índices en las tablas **member** y **charge**, volverá a ejecutar una consulta y comparará las operaciones de E/S de página.

Puede abrir, revisar y ejecutar secciones del archivo de comandos MergeJoin.sql o escribir y ejecutar las instrucciones Transact-SQL proporcionadas.

✍ Para observar cómo el optimizador de consultas procesa una consulta en una tabla sin índices

En este procedimiento eliminará todos los índices de las tablas **member** y **charge**, ejecutará una consulta y registrará la información estadística.

1. Escriba y ejecute esta instrucción para eliminar los índices existentes en las tablas **member** y **charge**:

```
USE credit
EXEC index_cleanup member
EXEC index_cleanup charge
```

2. Con la opción STATISTICS IO activada, escriba y ejecute esta instrucción SELECT para recuperar el número de miembro, apellido y número de cargo de los miembros cuyo apellido sea Hahn:

```
USE credit
SELECT member.member_no, lastname, charge_no
FROM member JOIN charge
ON member.member_no = charge.member_no
WHERE member.lastname = 'HAHN'
```

Nota A lo largo de este ejercicio, esta consulta se denominará consulta original.

3. Registre la información estadística de la tabla **member**.

Información	Resultado
Número de exploraciones	1
Número de lecturas lógicas	145
Número y nombre de los índices utilizados para procesar la consulta	Ninguno (recorrido de tabla)

4. Registre la información estadística de la tabla **charge**.

Información	Resultado
Número de exploraciones	1
Número de lecturas lógicas	582
Número y nombre de los índices utilizados para procesar la consulta	Ninguno (recorrido de tabla)

✍ Para crear índices y evaluar cómo el optimizador de consultas procesa una consulta en una tabla con índices

En este procedimiento creará índices en las tablas **member** y **charge**, volverá a ejecutar la consulta original y evaluará cómo el optimizador de consultas ha procesado la consulta mediante una estrategia de combinación de mezcla.

1. Escriba y ejecute esta instrucción para crear un índice compuesto no agrupado en las columnas **member_no** y **lastname** de la tabla **member**:

```
USE credit
CREATE nonclustered INDEX mbr_name
    ON member(member_no, lastname)
```

2. Escriba y ejecute esta instrucción para crear un índice compuesto no agrupado en las columnas **member_no** y **charge_no** de la tabla **charge**:

```
USE credit
CREATE nonclustered INDEX chg_charge_no
    ON charge(member_no, charge_no)
```

3. Vuelva a ejecutar la consulta original, que recupera el número de miembro, el apellido y el número de cargo de los miembros cuyo apellido sea Hahn.

Consulta original

```
USE credit
SELECT member.member_no, lastname, charge_no
    FROM member JOIN charge
    ON member.member_no = charge.member_no
    WHERE member.lastname = 'HAHN'
```

4. Registre la información estadística de la tabla **member**.

Información	Resultado
Número de exploraciones	1
Número de lecturas lógicas	32
Número y nombre de los índices utilizados para procesar la consulta	1 (mbr_name)

5. Registre la información estadística de la tabla **charge**.

Información	Resultado
Número de exploraciones	1
Número de lecturas lógicas	235
Número y nombre de los índices utilizados para procesar la consulta	1 (chg_charge_no)

6. Compare la información estadística de una consulta ejecutada en una tabla sin índices y la misma consulta ejecuta en una tabla con índices útiles.

¿Se han reducido las operaciones de E/S al agregar índices?

Sí. Mediante los índices, el optimizador de consulta sólo lee las páginas necesarias para devolver los resultados.

7. Haga clic en el botón **Mostrar plan de ejecución estimado** para mostrar el plan de ejecución de forma gráfica.

8. Examine el plan de ejecución.

¿Qué estrategia de combinación utilizó el optimizador de consultas para procesar la combinación?

Combinación de mezcla/combinación interna. El optimizador de consultas recorrió partes del índice de la tabla charge y parte del índice de la tabla member. Pudo combinar (mezclar) dicha información y devolver la información solicitada.

¿Por qué el optimizador de consultas seleccionó esta estrategia?

El optimizador de consultas seleccionó esta estrategia porque pudo tratar la consulta de cada tabla. Debido a que trató la consulta, sólo recorrió las páginas del nivel de hoja de los índices no agrupados, que se mantienen ordenados. Esto permitió que la combinación de mezcla aprovechara las entradas de cada tabla, que ya estaban ordenadas.

Ejercicio 3

Procesamiento de combinaciones hash

En este ejercicio eliminará todos los índices en las tablas **member** y **charge**, ejecutará una consulta y observará cómo el optimizador de consultas procesa la consulta mediante una estrategia de combinación hash.

Puede abrir, revisar y ejecutar secciones del archivo de comandos HashJoin.sql o escribir y ejecutar las instrucciones Transact-SQL proporcionadas.

✍ Para observar cómo se procesa una consulta mediante una estrategia de combinación hash

En este procedimiento eliminará todos los índices de las tablas **member** y **charge**, ejecutará una consulta y evaluará cómo el optimizador de consultas ha procesado la consulta.

1. Escriba y ejecute esta instrucción para eliminar los índices existentes en las tablas **member** y **charge**:

```
USE credit
EXEC index_cleanup member
EXEC index_cleanup charge
```

2. Escriba y ejecute esta instrucción SELECT para recuperar el número de miembro, el apellido y el número de cargo de los miembros que tengan el apellido Barr y su nombre sea Bos.

```
USE credit
SELECT m.member_no, lastname, charge_no
FROM member m JOIN charge c
ON m.member_no = c.member_no
WHERE m.lastname = 'BARR'
AND firstname = 'BOS'
```

3. Haga clic en el botón **Mostrar plan de ejecución estimado** para mostrar el plan de ejecución de forma gráfica.
4. Examine el plan de ejecución.

¿Qué estrategia de combinación utilizó el optimizador de consultas para buscar estas filas? ¿Por qué?

Combinación hash/combinación interna. Sin los índices, el optimizador de consultas debe recorrer ambas tablas para buscar las filas solicitadas. A continuación, el optimizador de consultas genera una tabla hash a partir de las filas que cumplen la condición de la tabla member, procesa cada fila de la tabla charge y devuelve las filas coincidentes.
