

Caso 1:

“Caso de estudio en SGBD”

En este caso de estudio se presentan los conceptos esenciales para el diseño de índices de tres estrellas. El objetivo de este tipo de índices es mejorar el desempeño de las operaciones de manipulación de datos como consultas, actualizaciones y borrados. Con el fin de analizar su beneficio, se analizó su comportamiento en dos Sistemas de Gestión de Bases de Datos (SGBD): Oracle 11g y Microsoft SQL Server 2008 R2.

Introducción.

En muchas organizaciones existen actualmente muchas aplicaciones desarrolladas en Sistemas de Gestión de Bases de Datos (SGBD) que presentan problemas de desempeño, es decir, los tiempos de respuesta no satisfacen las expectativas de los usuarios. Una de las posibles causas es que muchas operaciones, como las consultas, no están optimizadas. Por ejemplo, la falta de índices adecuados o la presencia de índices inadecuados pueden llevar a un pobre desempeño de las consultas.

Diseño de índices de tres estrellas.

Los índices se clasifican por el número de estrellas. Los índices de tres estrellas son los ideales para mejorar el desempeño de una operación DML. Un índice podría tener, por ejemplo, la tercera estrella, pero carecer de la primera o de la segunda. Para el diseño de los índices de tres estrellas, se consideran los dos siguientes casos:

A) Caso 1

Este caso se puede aplicar cuando:

- La cláusula WHERE no tiene predicados de rango (BETWEEN, >, >=, etc.)
- No hay reuniones (joins).
- Los predicados no presentan «dificultad» para el optimizador.

Primera estrella:

Para que un índice satisfaga la primera estrella, se deben elegir todas las columnas de la consulta que estén referenciadas en predicados de igualdad (WHERE col1 =:col1 AND col2 = :col2 ...), y estas serán las primeras columnas del índice.

Segunda estrella:

Para satisfacer la segunda estrella, el índice debe contener las columnas referenciadas en la cláusula ORDER BY (si la hay).

Tercera estrella:

Para satisfacer la tercera estrella se deben incluir en el índice todas las columnas referenciadas en la consulta. El orden de estas columnas en el índice no afecta el rendimiento de la consulta.

Este caso se puede aplicar cuando:

- No hay reuniones (joins).
- Ninguno de los predicados presenta dificultad para el optimizador.

Primera estrella:

Para que un índice satisfaga la primera estrella cuando hay predicados de rango, como BETWEEN, >, >=, etc. ; las primeras columnas del índice deberán ser las columnas que se encuentran referenciadas en predicados de igualdad (Si lo hay) seguidas de las columnas de predicado de rango.

Segunda estrella.

Para satisfacer la segunda estrella, la columna NOMBRE (referenciada en la cláusula ORDER BY.) debe ir antes de la columna APELLIDO y de esta forma se evita un proceso adicional para ordenar los resultados.

Tercera estrella.

Para satisfacer la tercera estrella se deberán incluir en el índice todas las columnas referenciadas en la consulta.

Consideraciones adicionales.

Con respecto a una operación de DML, siempre se puede diseñar un índice que tenga la tercera estrella, pero podría carecer de la primera o de la segunda estrella. Por lo tanto, la presencia de un predicado de rango implica que no siempre se puede tener un índice de tres estrellas. De esta forma se tiene 2 índices candidatos para el caso 2.

A). Índice candidato A

- Elegir las columnas del predicado de igualdad. Estas serán las primeras columnas del índice (su orden no importa en el índice)
- La siguiente columna del índice es la columna del predicado de rango más selectivo posible.
- Incluir en el índice las columnas referenciadas en la cláusula ORDER BY, además de las columnas ya incluidas en los pasos 1 y 2.
- Incluir en el índice todas las columnas restantes referenciadas por la consulta. (su orden no importa en el índice).

B) dice candidato B

- Elegir las columnas de predicado de igualdad. Estas serán las primeras columnas del índice (su orden no importa en el índice).
- Incluir en el índice las columnas referenciadas en la cláusula ORDER BY, además de las cláusulas ya incluidas en el paso 1.
- Incluir en el índice todas las columnas restantes referenciadas por la consulta, incluyendo la de los predicados de rango, (si las hay). Su orden no importa en el índice.

Desarrollo.

Para los experimentos se creó la tabla CLIENTE (CNO NOMBRE, APELLIDO y CIUDAD). La tabla se creó sin clave primaria, ya que estos SGBD se define automáticamente un índice sobre esta clave. La tabla se pobló con un millón de filas, las columnas. NOMBRE Y APELLIDO cada con 60 valores diferentes y la columna CIUDAD con 30 valores diferentes

Se crearon dos consultas para probar el caso de estudio:

A) Consulta 1

B) Consulta 2

```
SELECT  CNO, NOMBRE
FROM    CLIENTE
WHERE   APELLIDO = :APELLIDO
AND     CIUDAD  = :CIUDAD
ORDER BY NOMBRE;
```

Consulta 1

```
SELECT  CNO, NOMBRE
FROM    CLIENTE
WHERE   APELLIDO BETWEEN :APELLIDO1 AND :APELLIDO2
AND     CIUDAD  = :CIUDAD
ORDER BY NOMBRE;
```

Consulta 2 con un predicado de rango (BETWEEN).

1) Caso de estudio Oracle.

Se probó la consulta a uno con los siguientes pasos

- Índice adecuado.
- Acceso completo a la tabla.
- Índice de tres estrellas.

Para el primer caso se creó un índice llamado PRUEBA 1 con las columnas, APELLIDO Y NOMBRE. En la consulta se incluyó el *hint* INDEX para forzar el uso del índice de Oracle. (Un *hint* es una indicación para influir el plan de ejecución) Nótese que este índice es inadecuado porque no tiene ninguna estrella con respecto a esta consulta. Luego se ejecutó la misma consulta, pero sin forzar el uso del índice, es decir, sin el *hint*.

Se creó un índice de tres estrellas para la Consulta 1 llamado PRUEBA 2, el cual tiene las columnas, APELLIDO, CIUDAD, NOMBRE, CON en ese orden. Posteriormente se aprobó la Consulta 2, se analizaron los siguientes casos.

- Índice correspondiente al candidato A.

- Índice correspondiente al candidato. B (evita un proceso adicional para ordenar los resultados).

Índice correspondiente al candidato A.

Se creó un índice correspondiente al candidato A llamado PRUEBA 3 con las columnas, CIUDAD, APELLIDO, NOMBRE, y CNO. En ese orden se eligen estas columnas dado que la primera columna de este índice (CIUDAD) corresponde a una columna referenciada en un predicado de igualdad, seguida del predicado de rango más selectivo (APELLIDO) es decir, el predicado de rango con la menor proporción de filas que satisface el predicado.

Índice correspondiente al candidato B.

Se creó un índice correspondiente al candidato B llamado PRUEBA 4 con las columnas, CIUDAD, NOMBRE, APELLIDO, Y CON, en ese orden y de esta manera, los resultados estarán ordenados y se debería evitar un proceso adicional para ordenar los resultados.

2) Caso de estudio SQL Server.

Se probó la consulta uno con los siguientes casos.

- Índice inadecuado.
- Acceso completo a la tabla.
- Índice de tres estrellas.

Para el primer caso se creó un índice llamado PRUEBA 1, con las columnas, APELLIDO Y NOMBRE. Además, al igual que en Oracle, se requiere el hint INDEX para forzar el uso del índice en SQL Server. Posteriormente se ejecuta la consulta 1, pero para este caso no se usa el hint INDEX.

Se creó un índice de tres estrellas para la consulta 1 llamado PRUEBA 2, el cual tiene las columnas, APELLIDO, CIUDAD, NOMBRE Y CON, en ese orden. Ahora se considera la consulta 2 y se analizaron los siguientes casos:

- Índice correspondiente al candidato A.
- Índice correspondiente al candidato B (evita un proceso adicional para ordenar los resultados)

Índice correspondiente al candidato A.

Al igual que en el caso anterior, se creó un índice llamado PRUEBA. 3, con las columnas, CIUDAD, APELLIDO, NOMBRE, Y CON, en ese orden.

Índice correspondiente al candidato B.

Se creó un índice correspondiente al candidato B llamado PRUEBA 4, con las columnas, CIUDAD, NOMBRE, APELLIDO Y CON, en ese orden y de manera y de esta manera, los resultados estarán ordenados y se debería evitar un proceso adicional para ordenarlos.

Conclusiones.

A continuación, se presentan los resultados del caso de estudio, donde se analizó el comportamiento de índices en dos SGBD. Oracle 11g y SQL Server 2008 R 2; dichas pruebas se realizaron con una CPU de 2,00 GB de RAM y un procesador AMD Athlon™ 1.60 GHz.

En la siguiente tabla, se presentó un resumen de los resultados obtenidos.

MÉTODO USADO	ORACLE	SQL SERVER	CONSULTA	TIEMPO DE EJECUCIÓN	
				Oracle	SQL Server
Índice inadecuado (se fuerza su uso)	Se requiere un hint para forzar el uso del índice inadecuado.		1	90 s	526 ms (0,526 s)
Índice inadecuado (no se fuerza su uso)	El optimizador decide hacer un acceso completo a la tabla en vez de usar un índice inadecuado.		1	17 s	376 ms (0,376 s)
Índice de tres estrellas	Se evidencia una mejora significativa en cuanto al rendimiento (con respecto al índice inadecuado y al acceso completo a la tabla).		1	1 s	2 ms (0,002 s)
	El optimizador no evitó un proceso de ordenamiento adicional como se esperaba.	El optimizador evitó un proceso de ordenamiento adicional como se esperaba.			
Índice candidato A	El optimizador usó el índice sin necesidad de forzar su uso.		2	4 s	882 ms (0,882 s)
Índice candidato B	El optimizador usó el índice sin necesidad de forzar su uso. Se evitó un acceso completo a la tabla (se usó como camino de acceso sólo el índice).		2	5 s	797 ms (0,797 s)
	El optimizador no evitó un proceso de ordenamiento adicional como se esperaba.	El optimizador evitó un proceso de ordenamiento adicional como se esperaba.			

Resumen de resultados caso de estudio Oracle y SQL Server.

Se observó como el SGBD Oracle no evita un proceso adicional para ordenar los resultados cuando se usa un índice de tipo candidato B, proceso que se debería haber evitado. Sin embargo, en el SGBD. SQL Server sí se evitó dicho proceso.

También se mostró como Oracle decide no usar un índice si éste es inadecuado, ya que esto estaría resultaría más costoso que hacer un proceso completo a la tabla. Este mismo comportamiento se evidenció en SQL Server, al igual que cuando se usó un índice de tipo candidato A.

Fuente consultada.

Moreno, FJ. Y Olalde. JW. 2012. Índices de 3 estrellas. Un caso de estudio en Oracle. Y ese QL server. Ingenio, revista de la Facultad de Ingeniería. Volumen 13, número 26. Recuperado: Octubre 12 2020. Sitio web.

<https://revistas.usb.edu.co/index.php/Ingenium/article/view/1275>