



**DEPARTAMENTO
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

Taller de Optimización

9 de Junio de 2017

Bases de Datos

Grupo 3

Integrante	LU	Correo electrónico
Ciruelos Rodríguez, Gonzalo	063/14	gonzalo.ciruelos@gmail.com
Ferrante, Alejandro	371/09	matapalabras@hotmail
Goldstein, Brian	027/14	brai.goldstein@gmail.com
Thibeault, Gabriel	114/13	gabriel.eric.thibeault@gmail.com

Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

Índice

1. Ejercicios	3
1.1. Ejercicio 1	3
1.1.1. Queries	3
1.1.2. Índice Unclustered	3
1.1.3. Índice Clustered	3
1.1.4. Discusión	3
1.2. Ejercicio 2	5
1.2.1. Queries	5
1.2.2. Índice sobre corp_no	5
1.2.3. Índice sobre region_no	5
1.2.4. Discusión	5
1.3. Ejercicio 3	6
1.3.1. Queries	6
1.3.2. Índice clustered sobre member	6
1.3.3. Índice unclustered sobre charge	6
1.3.4. Discusión	6
1.4. Ejercicio 4	8
1.4.1. Queries	8
1.4.2. Índice clustered sobre member	8
1.4.3. Discusión	8
2. Conclusiones	9

1. Ejercicios

1.1. Ejercicio 1

En este ejercicio tenemos 2 índices sobre la misma tabla (member) y la misma columna (member_no, que es clave primaria). La única diferencia entre los índices es que uno es clustered y el otro no.

1.1.1. Queries

- (a) `SELECT * FROM member`
- (b) `SELECT * FROM member WHERE member_no > 100`
- (c) `SELECT member_no FROM member WHERE member_no > 100`
- (d) `SELECT * FROM member WHERE member_no > 9990`

1.1.2. Índice Unclustered

- (a) Realiza un table scan (filas estimadas: 10000).
- (b) Realiza un table scan (filas estimadas: 9900.01).
- (c) Realiza un index seek (filas estimadas: 9900.01).
- (d) Realiza un index seek (filas estimadas 10.999) y un nested loop con el resultado del index seek (filas estimadas: 10.999) y de un RID Lookup (filas estimadas: 1). ¹

1.1.3. Índice Clustered

- (a) Realiza un clustered index scan (filas estimadas: 10000).
- (b) Realiza un clustered index seek (filas estimadas: 9900.01).
- (c) Realiza un clustered index seek (filas estimadas: 9900.01).
- (d) Realiza un clustered index seek (filas estimadas: 10.999).

1.1.4. Discusión

A priori, se esperaría que los planes de ejecución para cada índice sean similares, puesto que los índices son muy similares. Sin embargo, fueron bastante distintos.

Creemos que las diferencias se deben principalmente a que el motor de base de datos estima que hay pequeñas diferencias entre una y otra estrategia. Por ejemplo, en la tercera

¹Creemos que con filas estimadas en el RID Lookup se refiere a cantidad de matches en la tabla por entrada del índice.

query los planes de ejecución coinciden, porque la cantidad de filas son muchas. Sin embargo, en la cuarta query, como la cantidad de filas estimadas del resultado es muchísimo menor, se hace algo distinto: con el índice clustered se hace simplemente un index seek, pero con el unclustered se hace un index seek y luego eso se junta con las filas correspondientes de la tabla.

Estos resultados nos permiten saber que, en un índice clustered, los datos están mucho más acoplados con el índice, en cambio en el unclustered no (al menos en la implementación del SQL Server que utilizamos).

1.2. Ejercicio 2

En este ejercicio tenemos 2 índices sobre la misma tabla, uno sobre la columna corp_no y otro sobre la columna region_no.

1.2.1. Queries

- (a) `SELECT * FROM member WHERE region_no = 9 AND corp_no = 126`
- (b) `SELECT * FROM member WHERE region_no = 9 AND corp_no = 368`

1.2.2. Índice sobre corp_no

- (a) Realiza un inner join entre un index seek y un RID lookup (filas estimadas del join y el index seek: 1.5, filas estimadas del RID lookup: 1). Luego filtrará los resultados del join (filas estimadas: 1).
- (b) Realiza un inner join entre un index seek y un RID lookup (filas estimadas del join y el index seek: 12, filas estimadas del RID lookup: 1). Luego filtrará los resultados del join (filas estimadas: 1.3308).

1.2.3. Índice sobre region_no

- (a) Realiza un table scan (filas estimadas: 1).
- (b) Realiza un table scan (filas estimadas: 1.3308).

1.2.4. Discusión

En la tabla hay 1109 elementos con region_no = 9. Además, hay 2 con corp_no = 126 y 12 con corp_no = 368. Hay 10000 filas en total.

Notemos, sabiendo cuántos datos hay, que cuando el índice es sobre region_no, como hay 1009 elementos que matchean, el planificador del motor evalúa que no le conviene recorrer el índice, y directamente recorre toda la tabla.

En el otro caso, como hay muy pocos elementos que matchean el corp_no de la query, el motor evalúa que es muy conveniente usar el índice, y por esa razón lo hace.

1.3. Ejercicio 3

En este ejercicio tenemos 2 índices sobre dos tablas distintas, pero ambos sobre la misma columna: member_no. El primer índice es un índice clustered sobre la tabla member (en este caso, el índice será sobre una clave primaria). El segundo índice es un índice unclustered sobre la tabla charge (o sea, el índice será sobre una clave foránea).

1.3.1. Queries

- (a) `SELECT c.*, m.* FROM charge c JOIN member m ON c.member_no = m.member_no`
- (b) `SELECT c.*, m.* FROM charge c JOIN member m ON c.member_no = m.member_no WHERE c.charge_no > 99990`
- (c) `SELECT c.*, m.* FROM charge c JOIN member m ON c.member_no = m.member_no WHERE m.member_no < 1000`

1.3.2. Índice clustered sobre member

- (a) Realiza un inner join (filas estimadas: 99862.7) de un clustered index scan (filas estimadas: 10000) y un table scan de charge (filas estimadas: 100000).
- (b) Realiza un inner join (filas estimadas: 10.9999) de un table scan de charge (filas estimadas: 10.999) y un clustered index seek (filas estimadas: 1).
- (c) Realiza un table scan sobre charge (filas estimadas: 632.116) y ordena ese resultado. A eso se le hace un inner join (632.116) con un clustered index seek de member (filas estimadas 999.9).

1.3.3. Índice unclustered sobre charge

- (a) Realiza un inner join (filas estimadas: 99862.7) de un table scan de member (filas estimadas: 10000) y un table scan de charge (filas estimadas: 100000).
- (b) Realiza un inner join (filas estimadas: 10.9999) de un table scan de charge (filas estimadas: 10.9999) y un table scan de member (filas estimadas: 10000).
- (c) Realiza un inner join (filas estimadas: 632.116) de un table scan de charge (filas estimadas: 632.116) y un table scan de member (filas estimadas: 999.9).

1.3.4. Discusión

Se puede ver una drástica diferencia entre el índice Clustered y el Unclustered: en el primer caso el motor emplea Index Scan/Seek mientras que en el segundo, Table Scan para todas las queries. Si bien esto se debe a que usar un índice Clustered es más eficiente en términos de tiempo de acceso a disco que uno Unclustered, el motor favorece Table Scan por sobre Index Seek incluso cuando se estiman sólo 11 filas de 100000.

Cabe destacar que para el primer índice en la segunda query el motor, al estimar pocas filas, simplemente realiza un Nested Loop Inner Join; en la tercera, al estimar un orden superior de filas, ordena el resultado del Table Scan y lleva a cabo un Merge Join.

1.4. Ejercicio 4

En este ejercicio tenemos sólo un índice. Éste es unclustered y es sobre la tabla member, sobre la columna firstname.

1.4.1. Queries

- (a) `SELECT * FROM member WHERE firstname = 'UVI'`
- (b) `SELECT * FROM member WHERE rtrim(ltrim(firstname)) = 'UVI'`

1.4.2. Índice clustered sobre member

- (a) Realiza un inner join (filas estimadas: 1) sobre el resultado de index seek (filas estimadas: 1) y un RID Lookup (filas estimadas: 1).
- (b) Realiza un table scsan (filas estimadas: 1000) sobre member.

1.4.3. Discusión

Se observa el uso del índice (en particular, se realiza un Index Seek) en la primera query pero no en la segunda. El índice no soporta la operación de búsqueda necesaria para ésta (búsqueda por igualdad sobre `rtrim(ltrim(firstname))`) en vez de sólo `firstname`), por lo que se debe emplear un Table Scan.

2. Conclusiones

Este taller nos permitió aprender varios aspectos distintos de las bases de datos:

- Cómo obtener el plan de ejecución de una cierta query. Además, como leer e interpretar toda esa información, y entender porqué el plan es ese y no otro.
- Además, como por ejemplo pasó en el ejercicio 1, viendo el plan de ejecución de una misma query, pero utilizando distintos índices, podemos intentar entender como funciona por dentro el motor de base de datos que estamos usando. Por ejemplo, como implementa los índices, las queries o las operaciones.
- Por último, como elegir índices para una tabla SQL, teniendo en cuenta las queries que vamos a realizar sobre ella. Esta es una decisión completamente no trivial, porque un índice es una estructura muy cara de mantener, y no es obvio que traiga mejoras inmediatas en la performance de las queries, a menos que se las elija con mucho cuidado.