Parcialito 6

Alumno: Alan Ezequiel Valdevenito.
Padrón: 107585.

Ejercicio 1

Item a

Esquema
profesores(<u>legajo</u>, nombre, apellido, genero, titulo)

Consulta
$\sigma_{titulo=\text{“}LICENCIADO\text{”} \wedge genero=\text{“}M\text{”}}(profesores)$

Información
▪ `n(profesores) = 500` y `F(profesores) = 25`. ▪ `V(titulo, profesores) = 10` y `V(genero, profesores) = 2`. ▪ La tabla profesores tiene un índice llamado `prtit` por título con `Height(prtit) = 2`. Este índice **no** es de clustering. ▪ Los valores de las columnas titulo y genero se almacenan siempre en mayúsculas.

Esta consulta tiene como resultado aquellas filas de la tabla profesores tales que se cumpla simultáneamente la condición de que su titulo sea LICENCIADO y su género sea M.

Notemos la tabla profesores tiene un índice (que no es de clustering) por el atributo título. Luego, como este atributo no es clave entonces es un índice secundario.

Notemos también que en la consulta tenemos una conjunción (AND) de dos condiciones.

¿Qué pasa si se aplica primero la primer condición?. Como tenemos un atributo que tiene un índice secundario asociado, se aplica primero esta condición y luego se selecciona del resultado a aquellas tuplas que cumplen con la segunda condición sin costo adicional.

Como tenemos un índice, podemos utilizar el método de Index Scan. Luego, el costo de este método considerando una búsqueda con índice secundario se calcula como

$$Costo(Seleccion) = Height\big(I(A_i, R)\big) + \lceil \frac{n(R)}{V(A_i, R)} \rceil$$

$$Costo(Seleccion) = Height\big(I(titulo, profesores)\big) + \lceil \frac{n(profesores)}{V(titulo, profesores)} \rceil$$

$$Costo(Seleccion) = 2 + \left\lceil \frac{500}{10} \right\rceil = 2 + 50 = 52$$

¿Qué pasa si se aplica primero la segunda condición?. Como tenemos un atributo simple (no tiene un índice asociado) podemos utilizar el método de File Scan. Luego, el costo de este método se calcula como

$$Costo(Seleccion) = B(R) = \frac{n(R)}{F(R)} = \frac{n(profesores)}{F(profesores)} = 20$$

Luego, el costo de aplicar primero la segunda condición es de 20 bloques accedidos.

Si ahora aplicamos la segunda condición tenemos que el costo será $2 + 20 = 22$ bloques accedidos.

Por lo tanto, el método de acceso más eficiente para resolver la consulta es primero aplicar la segunda condición (File Scan) y luego la primer condición (Index Scan) tal que el costo de la consulta es de 22 bloques accedidos.

Item b

Para estimar el tamaño de una selección de la forma $\sigma_{A_i=c}(R)$, utilizaremos la variabilidad de A_i en $R(V(A_i, R))$, que es la cantidad de valores distintos que puede tomar el atributo A_i en dicha relación. Luego, realizaremos la siguiente estimación

$$n\left(\sigma_{A_i=c}(R)\right) = \frac{n(R)}{V(A_i, R)}$$

$$n(Seleccion) = \frac{n(profesores)}{V(titulo, profesores)\ x\ V(genero, profesores)} = \frac{500}{10x2} = 25$$

Ejercicio 2

Item a

Esquema
megusta(<u>id_usuario, id_publicación</u>, fecha_hora)

Consulta
$megusta \bowtie_{\substack{id\_usuario \neq id\_usuario' \wedge \\ id\_publicacion = id\_publicacion'}} megusta'$

Información
▪ `n(megusta) = 100,000,000` y `F(megusta) = 1,000` ▪ `V(id_usuario, megusta) = 50,000` y `V(id_publicación, megusta) = 10,000,000` ▪ No se cuenta con índices y se dispone de `M = 1,001` bloques de memoria disponibles.

Esta consulta tiene como resultado los usuarios (sin incluir al dueño de la publicación) que le dieron me gusta a una publicación.

Notemos que no tenemos índices. Luego, podemos usar el método de loops anidados por bloque, el método de sort-merge y el método de junta hash (variante GRACE).

Notemos también que en la consulta tenemos una conjunción (AND) de dos condiciones.

Comencemos analizando el método de loops anidados por bloque.

¿Cuál es la cantidad de bloques de almacenamiento que ocupa la tabla?.

$$B(R) = \frac{n(R)}{F(R)} = \frac{n(megusta)}{F(megusta)} = \frac{100.000.000}{1.000} = 100.000$$

Notemos que tenemos más bloques en la relación, que bloques de memoria disponibles.

Luego, como en nuestro caso nos entraría CASI la primer tabla entera (salvo por su ultimo bloque), el costo de este método es

$$Costo(R * S) = \min\big(B(R), B(S)\big) + \left\lceil \frac{\min(B(R), B(S))}{M-2} \right\rceil x\ B(S)$$

$$Costo(megusta * megusta') = B(megusta) + \left\lceil \frac{B(megusta)}{999} \right\rceil x\ B(megusta')$$

$$Costo(megusta * megusta') = B(megusta) + \left\lceil \frac{B(megusta)}{999} \right\rceil x\ B(megusta')$$

$$Costo(megusta * megusta') = 100.000 + \left\lceil \frac{100.000}{999} \right\rceil x\ 100.000$$

$$Costo(megusta * megusta') = 100.000 + 101\ x\ 100.000 = 10.200.000$$

Analizamos el método de sort-merge.

Notemos que los archivos no entran en memoria, entonces debe utilizarse un algoritmo de sort externo.

El costo de ordenar R y volverlo a guardar en disco ordenado es

$$2\ x\ B(R) x\ \lceil \log_{M-1}(B(R)) \rceil$$

Una vez ordenados, se hace un merge de ambos archivos que sólo selecciona aquellos pares de tuplas en que coinciden los atributos de junta. El merge recorre una única vez cada archivo, con un costo de B(R) + B(S). El costo total es entonces:

$$Costo(R * S) = B(R) + B(S) + 2\ x\ B(R)\ x\ \lceil \log_{M-1}(B(R)) \rceil + 2\ x\ B(S)\ x\ \lceil \log_{M-1}(B(S)) \rceil$$

Sin embargo, como en nuestro caso tenemos que $R = S$ el costo total es

$$Costo(R * R') = B(R) + 2\ x\ B(R)\ x\ \lceil \log_{M-1}(B(R)) \rceil$$

$$Costo(megusta * megusta') = B(megusta) + 2\ x\ B(megusta)\ x\ \lceil \log_{1.001-1}(B(megusta)) \rceil$$

$$Costo(megusta * megusta') = 100.000 + 2\ x\ 100.000\ x\ \lceil \log_{1.000}(100.000) \rceil$$

$$Costo(megusta * megusta') = 100.000 + 2\ x\ 100.000\ x\ 2 = 500.000$$

Analizamos el método de junta hash (variante GRACE).

Este método cuenta con tres reglas:

1) Tenemos que m (cantidad de particiones de hash) debe ser menor o igual a M-1 (donde M es la memoria disponible) por la etapa de particionamiento.
2) Debe cumplirse que $\frac{B(R)}{m} \leq M - 2$, siendo R la relación más pequeña, por la etapa de loops anidados.
3) $m \leq V(A,R),\ m \leq V(A,S)$ asumiendo A el atributo de junta.

En nuestro caso, se debe cumplir que:

1) $m \leq 1000$
2) $\frac{100.000}{m} \leq 999$
3) $m \leq 50.000$

Luego, el costo de este método es

$$Costo(R * S) = 3\ x\ (B(R) + B(S))$$

pero como el join se hace sobre la misma tabla el costo resulta ser

$$Costo(R * R') = 3\ x\ B(R)$$

$$Costo(megusta * megusta') = 3\ x\ B(megusta)$$

$$Costo(megusta * megusta') = 3\ x\ 100.000 = 300.000$$

Nota: El costo total resulta ser el mencionado ya que tenemos un $B(R)$ para leer bloque a bloque la tabla, un $B(R)$ para grabar las particiones que se hacen una única vez por el atributo de junta y un ultimo $B(R)$ para hacer la parte de loops anidados donde se levanta cada partición entera en memoria y con dos punteros distintos se prueban todas las combinaciones entre filas para ser si cumplen o no la condición de la junta.

Por lo tanto, el método de acceso más eficiente para resolver la consulta es utilizando el método de junta hash (variante GRACE) tal que el costo de la consulta es de 300.000 bloques accedidos.

Item b

Realizaremos la siguiente estimación

$$n(R \bowtie_{\text{R.A=S.A}} \text{S}) = \frac{n(R)\, n(S)}{\text{máx}\big(V(A,R),\ V(A,S)\big)}$$

$$n(megusta * \text{ megusta}') = \frac{n(megusta)\, n(megusta')}{\max\big(V(id\_usuario, megusta),\ V(id\_publicacion, megusta)\big)}$$

$$n(megusta * \text{ megusta}') = \frac{100.000.000\ x\ 100.000.000}{\max(50.000,\ 10.000.000)} = \frac{100.000.000\ x\ 100.000.000}{10.000.000}$$

$$n(megusta * \text{ megusta}') = 1.000.000.000$$