

1^{er} Parcial Diseño y Gestión de Bases de Datos (11/12/2024)

Apellidos, Nombre:

1. (0,4 puntos) Indica si las siguientes afirmaciones son ciertas (C) o falsas (F).

En los privilegios fijos un usuario U sólo puede acceder en modo escritura a los objetos O que cumplan: $\text{clase}(U) \geq \text{clase}(O)$	F
La CPU puede operar directamente sobre el almacenamiento secundario, aunque en este caso el acceso será más lento	F
Un fichero es una secuencia de registros	V
Un fichero es una secuencia de bloques	V
La memoria principal asignada al SGBD está dividida en <i>búfers</i> cuyo tamaño suele ser menor que el del bloque.	F
La asignación continua de los bloques de un fichero facilita la expansión del fichero.	F
La asignación indexada de los bloques de un fichero utiliza un directorio en la cabecera del fichero que contiene las direcciones de los bloques del fichero.	V
La asignación de segmentos enlazados en un fichero utiliza grupos de bloques contiguos que se enlazan mediante punteros.	V

2. (0,4 puntos) Sea F un fichero con tres organizaciones posibles:

- 1) F es un fichero ordenado siendo A el campo de ordenación.
- 2) F es un fichero desordenado.
- 3) F es un fichero disperso, siendo A el campo de dispersión.

Indica en cuál de los tres supuestos anteriores serían más eficientes las siguientes operaciones justificando la respuesta en cada caso:

- a) Se quiere realizar una operación de búsqueda de registros con condición $A < 7$ (siendo A un campo clave).

En los tres casos sería necesario hacer un recorrido completo del fichero para comprobar la condición de desigualdad. En caso de que el fichero disperso ocupara más bloques que las otras organizaciones, el recorrido sería aún más costoso.

- b) Se quiere realizar una operación de borrado de registros con condición $A \geq 300 \wedge A < 1000$ (siendo A un campo no clave).

En el fichero disperso y en el desordenado sería necesario recorrer todos los registros para comprobar la condición, en el fichero ordenado por A se podría usar la búsqueda binaria para encontrar el primer registro con $A \geq 300$ y luego recuperar los siguientes registros mientras $A < 1000$, por ello esta organización sería la más adecuada.

3. (0,3 puntos) Se dice que una tabla es *volátil* si sufre muchas operaciones de inserción, borrado o modificación y que es *estática* en caso contrario.

Sea el siguiente esquema relacional:

T(A:entero, B:char(3), J: booleano)

CP: {A} VNN: {B}

S(C:entero, D:entero, E:char(100), F:char(500), G:char(100))

CP: {C} CAj: {D} → T(A) VNN: {E,F,G} Único: {D}

Y sean F_T y F_S los ficheros que almacenan las filas de T y S respectivamente, indica cuál sería la mejor organización para los supuestos y operaciones que se plantean sin considerar la definición de índices:

Supuestos	Organización para F_T y F_S
<ul style="list-style-type: none"> T es una tabla grande y estática. Muy frecuentemente se va a recuperar toda la información almacenada en T ordenada por B. S es una tabla de la que se conoce la cantidad de filas máxima que puede llegar a tener y la operación UPDATE S SET E='aaa' WHERE C= 8 es muy frecuente. 	<p>F_T: Ordenada por B</p> <p>F_S: Dispersa por C</p>
<ul style="list-style-type: none"> T es una tabla muy volátil y se va a recuperar muy frecuentemente la información con condición $A > 7$. S es una tabla de la que se conoce con bastante exactitud la cantidad de filas que puede llegar a tener. La operación de modificar el valor del campo C se realiza con mucha frecuencia. 	<p>F_T: Organizada como índice</p> <p>F_S: Desordenado</p>

4. (0,9 puntos) Dado un fichero F desordenado con 500.000 registros de tamaño 128 bytes, A es un campo clave de F de 24 bytes. Considerando que el tamaño de bloque es de 2Kb y el tamaño de un puntero es de 8 bytes, contesta a las siguientes preguntas (los valores que des deben estar justificados):

- a) ¿Cuál es el factor de bloque del fichero de datos y el factor de bloque del índice?

$$Fb_{\text{fichero}} = 2048/128 = 16$$

$$Fb_{\text{idx}} = 2048/32 = 64$$

- b) Se quiere construir un índice ordenado multinivel sobre A para F :

- ¿Cuántos niveles tendrá el índice ordenado multinivel? y ¿cuántas entradas tendrá el índice en cada uno de esos niveles?

Entradas en el 1^{er} nivel: 500.000

Bloque en el 1er nivel: $\lceil 500000/64 \rceil = 7.813$

Entradas 2º nivel: 7.813

Bloque en el 2º nivel: $\lceil 7813/64 \rceil = 123$

Entradas 3^{er} nivel: 123

Bloque en el 3^{er} nivel: $\lceil 123/64 \rceil = 2$

Entradas 4º nivel: 2

Bloque en el 4º nivel: $\lceil 2/64 \rceil = 1$

- ¿Cuántas entradas tendría el primer nivel si el campo A no fuera un campo clave y tuviera 150.000 valores diferentes?

En este caso el índice en su 1^{er} nivel tendría una entrada por valor, es decir: 150.000

- c) Se quiere construir un índice B+ sobre A para F :

- ¿Cuántas entradas caben en cada nodo hoja del índice?

En cada hoja caben $p-1$ valores y (siendo p el orden del árbol), $p-1$ punteros a registros de datos, 1 puntero a la hoja anterior y otro puntero a la hoja siguiente, es decir $p+1$ punteros:

En cada hoja (bloque) cabe: $24(p-1) + 8(p+1) = 2048$

$$24p - 24 + 8p + 8 = 2048$$

$$p = \lceil (2048 + 16) / 32 \rceil = 64$$

El número de entradas que caben en cada nodo hoja es $p-1$: 63 valores en cada hoja

- ¿Cuántas entradas caben en cada nodo interno del índice?

Los nodos internos son distintos porque no tienen punteros al nodo siguiente y al anterior cada nodo interno se almacenan $p-1$ valores y p punteros a subárboles.

En cada nodo interno (bloque) cabe: $24(p-1) + 8p = 2048$

$$24p - 24 + 8p = 2048$$

$$p = \lceil (2048 + 24) / 32 \rceil = 64$$

El cada nodo interno hay 63 valores del campo de indexación y 64 punteros.

5. (0,9 puntos) Dado el siguiente esquema relacional:
- R(A:entero, B:boolean, C:entero, D:char(20))
 CP:{A}
 S(E:char(3), A:entero, G: char(50)).
 CP:{E}
 CAj: {A}→R
 T(A:entero, E:char(3), H:entero)
 CP:{A,E}
 CAj:{A}→R
 CAj:{E}→S

Suponiendo que existe un índice sobre el atributo *A* en el fichero que almacena la tabla *R* y un índice sobre el atributo *E* en el fichero que almacena la tabla *S*, si la tabla *R* tiene 1.000 filas y la tabla *T* tiene 200.000, con una distribución uniforme de los valores de *A* en *T*, resuelve las siguientes cuestiones:

- a) ¿Cuál sería la selectividad de condición de concatenación: $R \bowtie_{T.A=R.A} T$?

La selectividad de una concatenación es el cociente entre la cantidad de filas que hay en la tabla resultado de la concatenación y la cardinalidad del producto cartesiano:

$$js = |R \bowtie_{T.A=R.A} T| / |R \times T|$$

En este caso dado que *A* en *T* es una clave ajena a *R* que no puede ser nula (forma parte de la clave primaria), todas las filas de *T* aparecen en el resultado:

$$js = 200.000 / 200.000.000 = 1/1000$$

- b) Dada la consulta:

```
SELECT R.A, R.B, S.E, T.H
FROM R, S, T
WHERE T.H>0 AND T.A=R.A AND R.A=S.A;
```

- Escribe la expresión equivalente en Álgebra Relacional.

$((R \times S) \times T)$ DONDE $T.H > 0 \wedge T.A = R.A \wedge R.A = S.A$ [R.A, R.B, S.E, T.H] o bien

$$\pi_{R.A, R.B, S.E, T.H} \sigma_{T.H > 0 \wedge T.A = R.A \wedge R.A = S.A} ((R \times S) \times T)$$

- Escribe la expresión de Álgebra Relacional equivalente al árbol de consulta optimizado (obviamente resuelve este apartado después de haber resuelto el último de la página siguiente).

Expresión optimizada:

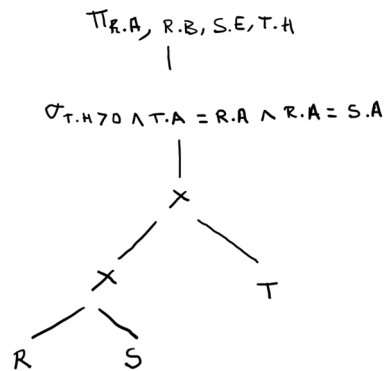
$$\left(R \bowtie_{R.A=S.A} S \bowtie_{T.A=R.A} T \text{ DONDE } T.H > 0 \right) [R.A, R.B, S.E, T.H]$$

o bien:

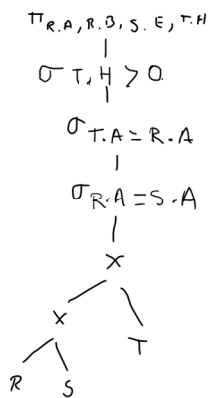
$$\pi_{R.A, R.B, S.E, T.H} \left(R \bowtie_{R.A=S.A} S \right) \bowtie_{T.A=R.A} \sigma_{T.H > 0} T$$

- Dibuja el árbol de consulta sin optimizar y optimízalo aplicando el algoritmo. Para cada paso de la transformación dibuja el árbol e indica la regla utilizada.

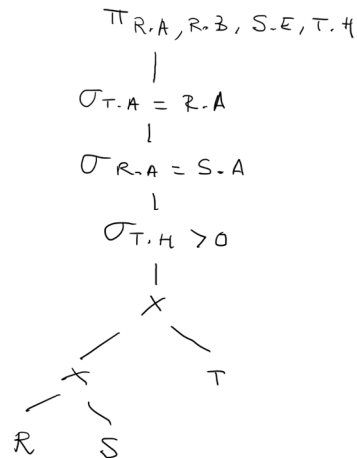
Árbol sin optimizar:



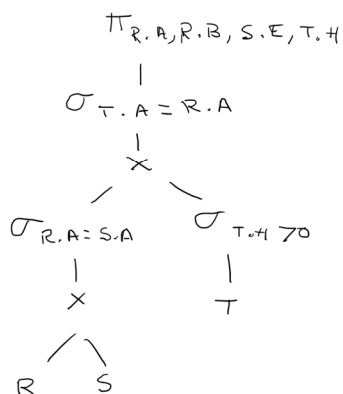
Regla 1.



Regla 2. (2 veces)



Regla 6 (2 veces)



Regla 12 (2 veces)

