

## Certamen 3

Nicolás Gómez Morgado

Administración y programación de base de datos

18 de julio de 2024

### 1. Ejercicios guía 1

- Supongamos que estamos utilizando un sistema de disco donde el tiempo para mover la cabeza de lectura/escritura a un bloque de 15ms, y el tiempo de transferencia de un bloque es de 0.4ms. Supongamos que queremos calcular el join de R con S, y tenemos que  $B(R)=1000$ ,  $B(S)=500$  y  $M=101$ . Para acelerar el join, queremos leer y escribir tantos bloques como podamos en posiciones consecutivas del disco, y usar buffers que puedan ser múltiplos de un bloque. Responda las siguientes preguntas.

Tiempo en mover cabeza a un bloque: 15ms.

Tiempo transferencia de un bloque: 0.4ms

$R \bowtie S$

$B(R) = 1000$

$B(S) = 500$

$M = 101 \rightarrow 100$  (Se deja uno para la salida)

- ¿Cuántas E/S de disco se requieren para realizar esta operación join?

1 <sup>er</sup> pasada	Leer M bloques de R en MP[Memoria principal] (ordenar, escribir contenido ordenando)	$2B(R)$ crear sublistas ordenadas
2 <sup>da</sup> pasada	El atributo de ordenación y unión	$B(R)$ leer cada sublista
	Total	$3B(R)$

Que para R son  $3B(R)$ , por lo tanto para S, es lo mismo  $3B(S)$ .

Se requiere  $B(R) + B(S)$ .

**Total:**

Lo mismo para S  $\rightarrow 3B(S)$  App. requerimientos  $\lfloor \sqrt{B(R) + B(S)} \rfloor$

$$\begin{aligned}
 \text{Disco E/S} &= 3(B(R) + B(S)) \\
 &= 3(1000 + 500) \\
 &= 4500
 \end{aligned}$$

Por lo tanto son necesarias 4500 E/S de disco para realizar la union.

- b) ¿Cuanto tiempo toma un join basado en ordenamiento (sort-merge join), suponiendo que escribimos sublistas ordenadas en bloques consecutivos del disco?

$$\text{Sublista de R} = \frac{B(R)}{100} = \frac{1000}{100} = 10$$

$$\text{Sublista de S} = \frac{B(S)}{100} = \frac{500}{100} = 5$$

1 <sup>er</sup> pasada	R (Lectura y escritura) y S (Lectura y escritura): misma pista de lectura/escritura secuenciales	
	$[15 + (10) \cdot [(100 \cdot 4) + (100 \cdot 4)]] + [15 + (5 \cdot [(100 \cdot 4) + (100 \cdot 4)])]$ $[15 + (10) \cdot [80]] + [15 + (5 \cdot [80])]$ $[15 + 800] + [15 + 400]$ $815 + 415$ 1230 ms.	
2 <sup>da</sup> pasada	Original 15 buffers c/u con 1 bloque	Nueva: 15 buffers c/u con 6 bloques
	15 buffers · 15 ms = 100 times · 15 buffers · 15 ms = 25500 ms = 22.5 seg	17 times · 15 buffers · 15 ms $\sim 3825$ ms $\sim 3.825$ seg

$$1 \text{ buffer} - 1 \text{ bloque} = 1230ms + 22500ms = 23730ms \approx 23,7seg.$$

$$1 \text{ buffer} - 6 \text{ bloques} = 1230ms + 3825ms = 5055ms \approx 5,05seg.$$

2. Supongamos que tenemos las relaciones  $R(a, b)$  ,  $S(b, c)$  ,  $T(c, d)$  ,  $U(d, e)$  con las siguientes características:

$$\begin{array}{ll}
T(R) = 100 & T(T) = 100 \\
V(R, b) = 100 & V(T, c) = 10 \\
T(S) = 100 & V(T, d) = 100 \\
V(S, b) = 100 & T(U) = 100 \\
V(S, c) = 10 & V(U, d) = 100
\end{array}$$

Computar un orden de Join de R sobre S Sobre T Sobre U  $[R \bowtie S \bowtie T \bowtie U]$ , utilizando:

- a) Dinámica (Dynamic Programming method)

R	S	T	U
100	100	100	100
0	0	0	0
R	S	T	U

Considerando los pares:

- 1)  $R \bowtie S = 100$
- 2)  $R \bowtie T = 10000$
- 3)  $R \bowtie U = 10000$
- 4)  $S \bowtie T = 1000$
- 5)  $S \bowtie U = 10000$
- 6)  $T \bowtie U = 100$

	R,S	R,T	R,U	S,T	S,U	T,U
Size	100	10000	10000	1000	10000	100
Cost.						
Best plan						

Ahora considerar la unión de 3 de las 4 relaciones. Elegir 2 para unir primero.

R,S,TR,S,UR,T,US,T,U

- 1) (R,S,T) =  
 $R \bowtie S = 100 \leftarrow$   
 $R \bowtie T = 10000$   
 $S \bowtie T = 1000$

$$T((R \bowtie S) \bowtie T) = \frac{T(R \bowtie S) \cdot T(T)}{\max\{V(R \bowtie S, c), V(T, c)\}} = \frac{100 \cdot 100}{\max\{10, 10\}} \\ = \frac{10000}{10} = 1000$$

- 2) (R,S,U) =  
 $R \bowtie S = 100 \leftarrow$   
 $R \bowtie U = 10000$   
 $S \bowtie U = 10000$

**NIIDEA**

$$T((R \bowtie S) \bowtie U) = \frac{T(R \bowtie S) \cdot T(U)}{\max\{V(R \bowtie S, c), V(U, c)\}} = \frac{1000 \cdot 100}{\max\{10, 10\}} \\ = \frac{100000}{10} = 10000$$

- 3) (R,T,U) =  
 $R \bowtie T = 10000$   
 $R \bowtie U = 10000$   
 $T \bowtie U = 100 \leftarrow$

$$T((R \bowtie T) \bowtie U) = \frac{T(R \bowtie T) \cdot T(U)}{\max\{V(R \bowtie T, d), V(U, d)\}} = \frac{10000 \cdot 100}{\max\{100, 100\}} \\ = \frac{1000000}{100} = 10000$$

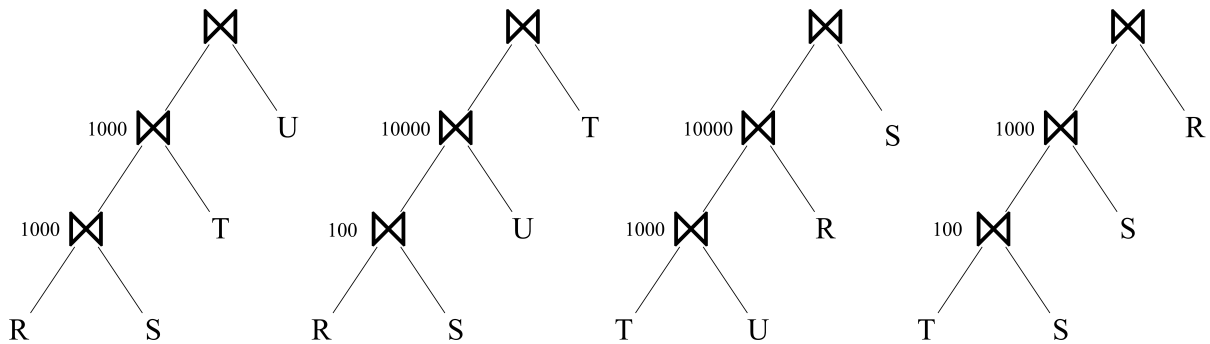
$$\begin{aligned}
 4) (S, T, U) = \\
 S \bowtie T &= 1000 \\
 S \bowtie U &= 10000 \\
 T \bowtie U &= 100 \leftarrow
 \end{aligned}$$

$$\begin{aligned}
 T((S \bowtie T) \bowtie U) &= \frac{T(S \bowtie T) \cdot T(U)}{\max\{V(S \bowtie T, d), V(U, d)\}} = \frac{1000 \cdot 100}{\max\{100, 100\}} \\
 &= \frac{100000}{100} = 1000
 \end{aligned}$$

	RST	RSU	RTU	STU
S	1000	10000	10000	1000
C	100	100	100	100
P	$(R \bowtie S) \bowtie T$	$(R \bowtie S) \bowtie U$	$(T \bowtie U) \bowtie R$	$(T \bowtie U) \bowtie S$

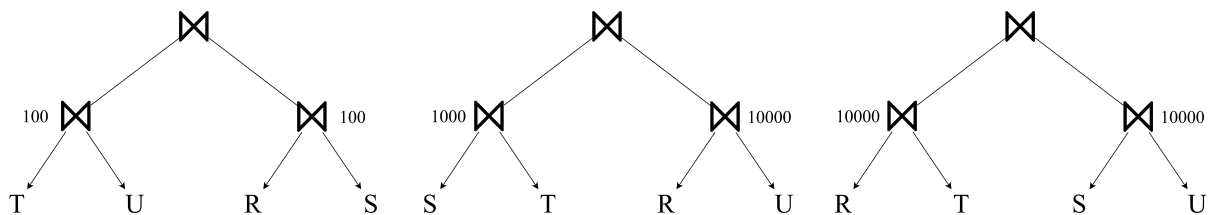
Tripletes.

Considerar arboles.



**Agrupando:**

$$\begin{aligned}
 (((R \bowtie S) \bowtie T) \bowtie U) &= 1000 + 100 = 1100 \\
 (((R \bowtie S) \bowtie U) \bowtie T) &= 10000 + 100 = 10100 \\
 (((R \bowtie T) \bowtie U) \bowtie S) &= 10000 + 100 = 10100 \\
 (((S \bowtie T) \bowtie U) \bowtie R) &= 1000 + 100 = 1100
 \end{aligned}$$



**Agrupando:**

$$\begin{aligned}
 (T \bowtie U) \bowtie (R \bowtie S) &= 100 + 100 = 200 \leftarrow \\
 (S \bowtie U) \bowtie (R \bowtie U) &= 10000 + 1000 = 11000 \\
 (R \bowtie T) \bowtie (S \bowtie U) &= 10000 + 10000 = 20000
 \end{aligned}$$

b) Greedy method

**Se toma una decision sin retroceder.**

Base: Pares de relaciones cuyo tamaño estimado es el mas pequeño (árbol actual).

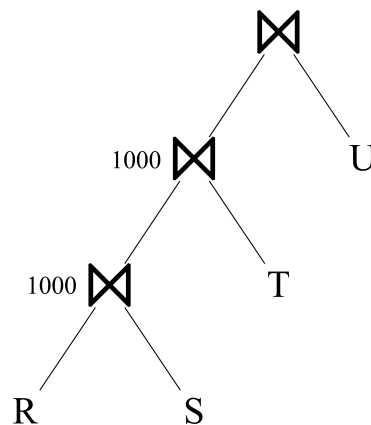
$$\begin{array}{ll} R, S = 100 & (1) \quad S, T = 10000 \\ R, T = 10000 & S, U = 10000 \\ R, U = 10000 & T, U = 100 \quad (2) \end{array}$$

Inducción: Encontrar todas las relaciones no includes, en este caso, T y U.

$$\begin{array}{l} R \bowtie S - ((R \bowtie S) \bowtie U) = 10000 \\ R \bowtie S - ((R \bowtie S) \bowtie T) = 1000 \end{array}$$

Se escoge T. Luego hay que unirse a U, no hay mas opciones.

$$(((R \bowtie S) \bowtie T) \bowtie U) = 1100$$

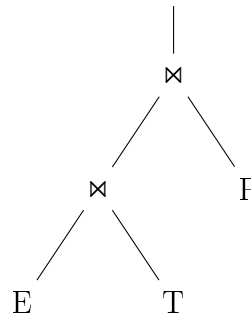


## 2. Ejercicios guía 2

1. Traspasar la consulta SQL a un árbol de consulta.

- SELECT apellido1
- FROM Empleado E, trabajaEn T, proyecto P
- WHERE nameProy='aquarius'
- AND T.idProy = P.idProy
- AND E.idEmpleado = T.idEmpleado
- AND E.FechaNacimiento > '1957-12-21'

- Obtener el árbol inicial (canónico) de la consulta

$$\pi_{\text{apellido1}} \wedge \sigma_{\text{idProy} = \text{P.idProy} \wedge \text{nameProy} = \text{'aquarius'} \wedge \text{idEmpleado} = \text{T.idEmpleado} \wedge \text{E.FechaNacimiento} > \text{'1957-12-21'}}$$


- Explique como se optimiza el árbol de consulta mediante la optimización vista en clases

2. Considere las siguientes relaciones:

Variedades(IdVar, Nombre, Prog2, Prog1)

Predios(IdPredio, NombrePredio, Comuna, Superficie)

Siembra(IdPredio, IdVar, HaSem, Rdto, añoA)

Sea la siguiente consulta: “Listar los nombres de las variedades sembradas en el predio idPredio = 10 y que el año 2015 tuvieron un rendimiento mayor a 60 qq/ha”.

- a) Escriba la consulta en SQL para la consulta anterior
- b) Escriba la consulta en A.Relacional para la consulta anterior
- c) Obtener el árbol inicial (canónico) de la consulta
- d) Explique como se optimiza el árbol de consulta mediante el algoritmo de optimización algebraica (visto en clases)

3. Suponer que tenemos las relaciones R(a, b), S(b, c), T(c, d), y U(d, e) con las siguientes características:

T(R) = 300

V(R, b) = 100

T(S) = 200

V(S, b) = 100

T(T) = 150

T(U) = 500

V(S, c) = 20

V(T, c) = 20

V(T, d) = 300

V(U, d) = 300



4. . Suponer que tenemos las relaciones  $R(a, b)$ ,  $S(b, c)$ ,  $T(c, d)$ , y  $U(d, e)$  con las siguientes características:

$$T(R) = 50$$

$$V(R, b) = 250$$

$$T(S) = 55$$

$$V(S, b) = 500$$

$$V(S, c) = 5$$

$$V(U, d) = 50$$

$$T(T) = 50$$

$$T(U) = 45$$

$$V(T, c) = 15$$

$$V(T, d) = 500$$

Para 3 y 4, calcular un orden/JOINS para  $R$ ,  $S$ ,  $T$ ,  $U$ , usando Programación dinámica (como visto en clases). Mostrar tabla inicial de costos, los cálculos de cada etapa y árboles.