

Resumen para el control:

Álgebra relacional, SQL, optimización

Nota: los problemas 1 y 2 tratan de los siguientes esquemas:

```
LIBRO(ID_LIBRO, TITULO, NOMBRE_EDITORIAL)
AUTORES_LIBRO(ID_LIBRO, NOMBRE_AUTOR)
EDITORIAL(NOMBRE, DIRECCION, TELEFONO)
COPIAS_LIBRO(ID_LIBRO, ID_SUCURSAL, NUM_DE_COPIAS)
PRESTAMOS(ID_LIBRO, ID_SUCURSAL, NUM_TARJETA, FECHA_SALE, FECHA_DEVOL)
SUCURSAL_BIBLIOTECA(ID_SUCURSAL, NOMBRE_SUCURSAL, DIRECCION)
LECTOR(NUM_TARJETA, NOMBRE, DIRECCION, TELEFONO)
```

Estos esquemas modelan una red de bibliotecas (de lectura) con información sobre sus libros.

Problema 1 (SQL)

¿En qué sucursales está “A FIRST COURSE IN DATABASE SYSTEMS”? ¿Cuántas copias hay por sucursal?

Consulta muuuuuy sencilla:

```
SELECT C.ID_SUCURSAL, C.NUM_COPIAS
FROM COPIAS_LIBRO C, LIBRO L
WHERE L.TITULO = "A FIRST COURSE IN DATABASE SYSTEMS"
AND L.ID_LIBRO = C.ID_LIBRO ;
```

¿Cuál es el tiempo promedio de devolución de libros según sucursal? Use orden descendente.

Agrupamos PRESTAMOS por sucursal, y sacamos el promedio de los tiempos de retorno. Evitaremos los NULL, claro está.

```
SELECT P.ID_SUCURSAL, AVG(P.FECHA_DEVOL - P.FECHA_SALE) AS TIEMPO_M
FROM PRESTAMOS P
WHERE P.FECHA_DEVOL NOT NULL
GROUP BY P.ID_SUCURSAL
ORDER BY TIEMPO_M DESC ;
```

¿Qué autor tiene el mayor número de coautores?

Esta consulta tiene varias soluciones, para el gusto del creativo. De forma simple, son los autores que tienen una cantidad de coautores **mayor o igual** a los del resto:

```
SELECT AL1.NOMBRE_AUTOR
FROM AUTORES_LIBRO AL1, AUTORES_LIBRO AL2
WHERE AL1.ID_LIBRO = AL2.ID_LIBRO
GROUP BY AL1.NOMBRE_AUTOR
HAVING COUNT(DISTINCT AL2.NOMBRE_AUTOR)
>= ( SELECT COUNT(DISTINCT AL4.NOMBRE_AUTOR)
      FROM AUTORES_LIBRO AL3, AUTORES_LIBRO AL4
      WHERE AL3.ID_LIBRO = AL4.ID_LIBRO
      GROUP BY AL3.NOMBRE_AUTOR ) ;
```

¿Qué editorial hay publicado la mayor cantidad de libros?

Problema propuesto y sencillo.

Problema 2 (Álgebra relacional y optimización)

¿Dónde viven los lectores que han sacado libros de 3 ó más sucursales? Use agregación. Optimice.

Relacionamos LECTOR a PRESTAMOS y después, con el operador de agregación, exigimos tres sucursales distintas.

$$\begin{aligned}
 A &\leftarrow \text{LECTOR} \\
 B &\leftarrow \text{PRESTAMOS} \\
 C &\leftarrow \text{NUM_TARJETA} \bowtie_{\text{NUM_TARJETA}, \text{count}(\text{distinct ID_SUCURSAL as NUM})} B \\
 \text{Resp} &\leftarrow \pi_{A.DIRECCION} [\sigma_{C.NUM \geq 3} (A \times C)]
 \end{aligned}$$

De esta consulta vale notar observar que sólo se puede optimizar al realizar proyecciones. Luego:

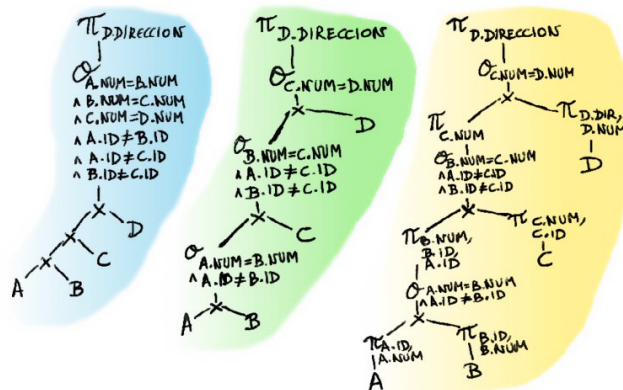
$$\begin{aligned}
 A &\leftarrow \pi_{\text{NUM_TARJETA}, \text{DIRECCION}} \text{LECTOR} \\
 B &\leftarrow \pi_{\text{NUM_TARJETA}, \text{ID_SUCURSAL}} \text{PRESTAMOS} \\
 C &\leftarrow \text{NUM_TARJETA} \bowtie_{\text{NUM_TARJETA}, \text{count}(\text{distinct ID_SUCURSAL as NUM})} B \\
 \text{Resp} &\leftarrow \pi_{A.DIRECCION} [\sigma_{C.NUM \geq 3} (A \times C)]
 \end{aligned}$$

¿Dónde viven los lectores que han sacado libros de 3 ó más sucursales? Sin agregación. Optimice.

Bueno, tomamos tres copias de PRESTAMO, hacemos producto cartesiano, y verificamos que los tríos tengan igual NUM_TARJETA pero distinto ID_SUCURSAL. Luego proyectamos por NUM_TARJETA para cruzar con LECTOR y recuperar DIRECCIÓN. Es algo engorroso de anotar, pero esta explicación debería ser suficiente.

$$\begin{aligned}
 A, B, C &\leftarrow \text{PRESTAMOS} \\
 D &\leftarrow \text{LECTOR} \\
 \text{Resp} &\leftarrow \pi_{D.dir} \sigma_{\substack{A.num_tarj=B.num_tarj \wedge B.num_tarj=C.num_tarj \wedge \\ C.num_tarj=D.num_tarj \wedge A.id_suc \neq B.id_suc \wedge \\ A.id_suc \neq C.id_suc \wedge B.id_suc \neq C.id_suc}} (A \times B \times C \times D)
 \end{aligned}$$

La optimización la hacemos de forma gráfica:



Recordemos los pasos esenciales de la optimización heurística de consultas: (1) romper una selección en varias operaciones (sólo válido con *and*), (2) se llevan estas selecciones lo más cerca posible de las hojas para reducir el número de tuplas y (3) se proyecta justo antes de los join, para reducir el tamaño de las tuplas.