

Introducción a Optimización de Consultas

Tema 15



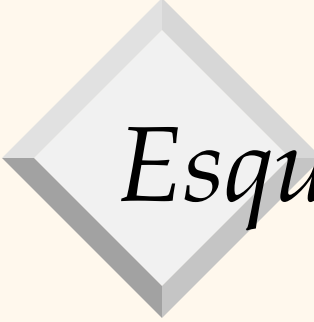
Descripción de optimización de Consultas

- ❖ Plan: *Árbol de operadores de álgebra relacional, con la opción de un algoritmo para cada operador.*
 - Cada operador es implementado usando una interface que toma las tuplas de salida y las coloca como tuplas de entrada y las computa.
- ❖ Dos puntos principales:
 - Dada una consulta, ¿qué planes son considerados?
 - ◆ Algoritmo para buscar el plan con el menor costo.
 - ¿Cómo se estima el costo del plan?
- ❖ **Idealmente**: Se desea el mejor plan. **En la practica**: Se evita el peor plan!



Puntos principales de la optimización

- ❖ **Estimación del Costo:** Aproximado al mejor de los casos.
 - Mantiene estadísticas en el catálogo del sistema que usa para estimar el costo de la operación y el tamaño del resultado.
 - Considera combinación de costo de CPU y I/O.
- ❖ En algunos casos es posible que la salida de cada operador sea canalizada (*pipelined*) al siguiente operador sin almacenarla en una relación temporal.



Esquemas de ejemplo

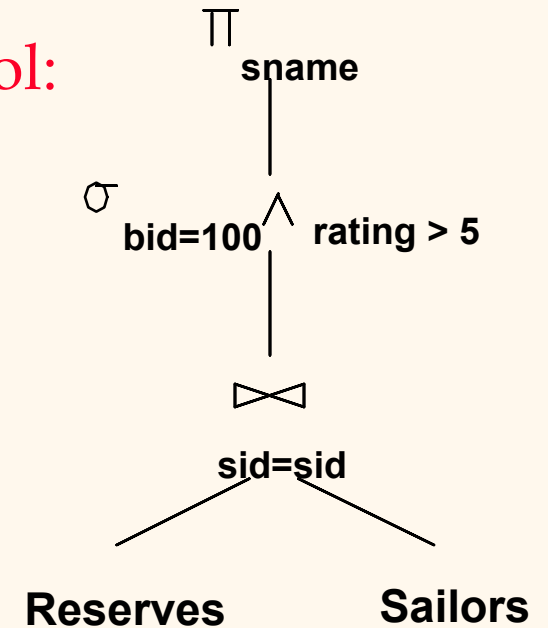
Sailors (*sid*: integer, *sname*: string, *rating*: integer, *age*: real)
Reserves (*sid*: integer, *bid*: integer, *day*: dates, *rname*: string)

- ❖ Similar al esquema anterior; se agrego *rname*.
- ❖ Reserves:
 - Cada tupla 40 bytes, 100 tuplas por página, 1000 páginas.
- ❖ Sailors:
 - Cada tupla 50 bytes, 80 tuplas por página, 500 páginas.

Ejemplo de motivación

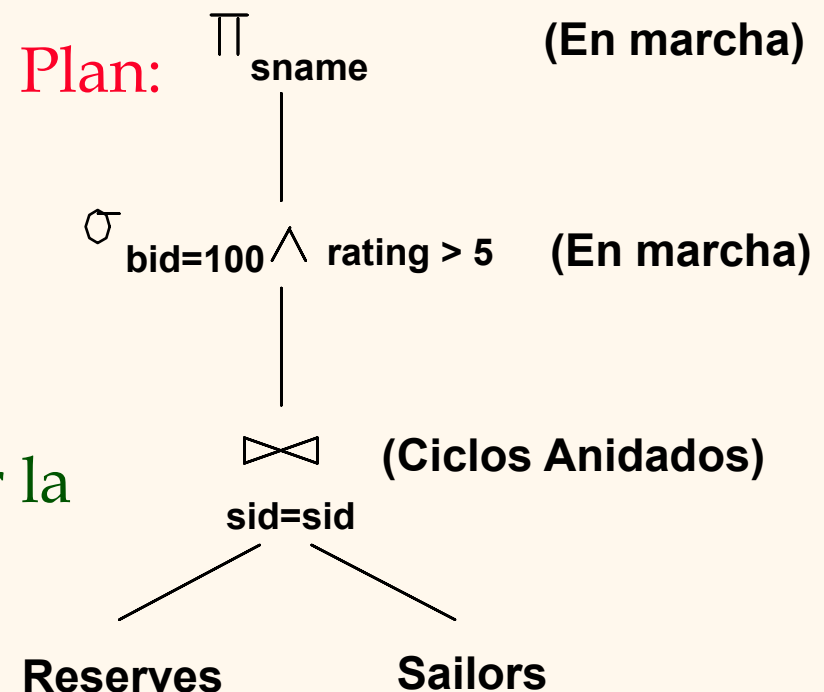
```
SELECT S.sname
FROM Reserves R, Sailors S
WHERE R.sid=S.sid AND
      R.bid=100 AND S.rating>5
```

Arbol:

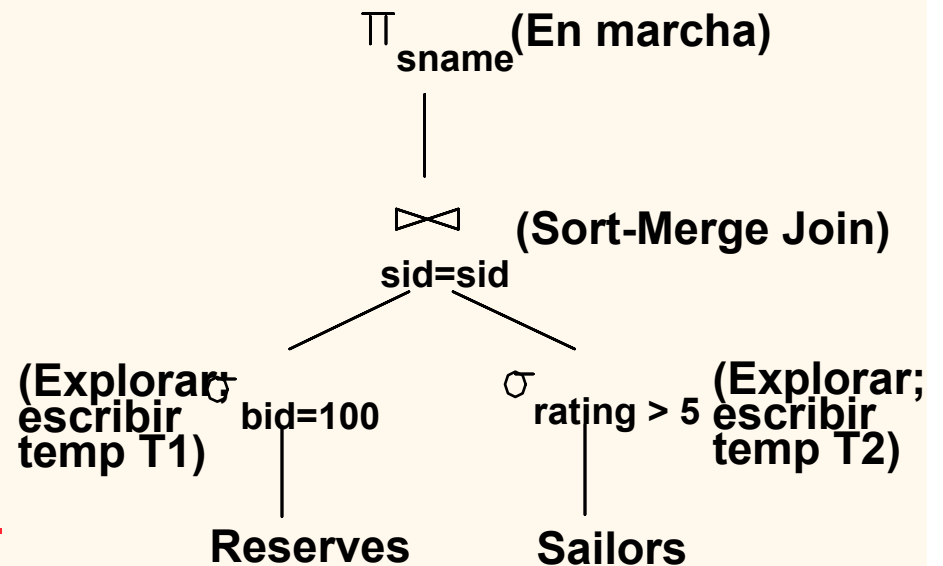


- ❖ Costo: 500+500*1000 I/Os
- ❖ No es el peor plan!
- ❖ Hay otras opciones: las selecciones pudieron colocarse antes, no se usan índices.
- ❖ Meta de optimización: Buscar el plan mas eficiente para obtener la misma respuesta.

Plan:



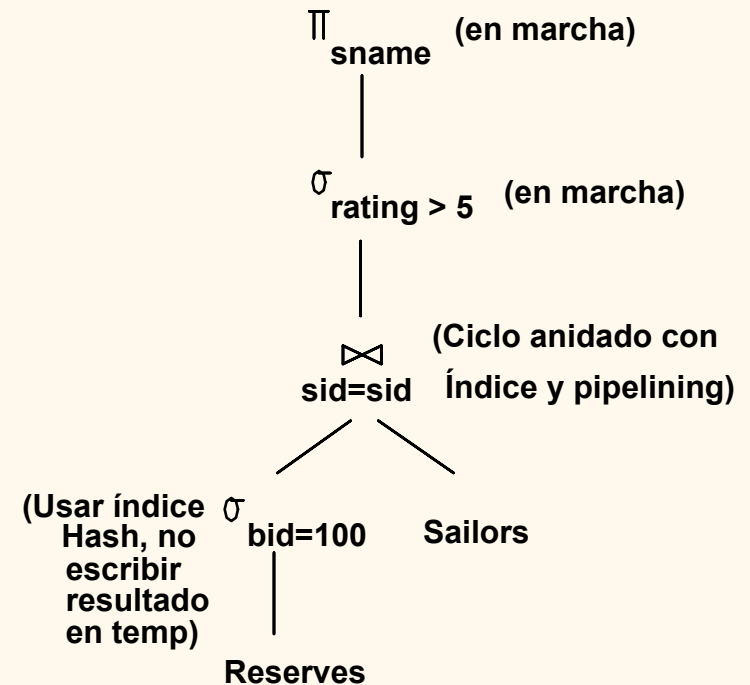
Plan Alternativa 1 (Sin índices)




- ❖ **Principal diferencia: selecciones.**
- ❖ **Con 5 buffers, costo del plan:**
 - Explorar Reserves (1000) + Escribir relación temporal T1 (10 páginas, si se tienen 100 barcos, distribuidos uniformemente).
 - Explorar Sailors (500) + Escribir T2 (250 páginas, si se tienen 10 ratings).
 - Ordenar T1 ($2 \times 2 \times 10$), Ordenar T2 ($2 \times 4 \times 250$), unir ($10 + 250$)
 - **Total: 4060 I/Os.**
- ❖ **Si se usa join por ciclo anidado por bloques (BNL), costo del join = $10 + 4 \times 250$, costo total = 2770.**
- ❖ **Al realizar la proyección, T1 solo tiene *sid*, T2 solo *sid* y *sname*:**
 - T1 ocupa 3 páginas, costo de BNL reduce 250 páginas, **total < 2000.**

Plan Alternativa 2 Con índice

- ❖ Con un índice cluster sobre *bid* de Reserves, se obtiene $100,000/100 = 1000$ tuplas en $1000/100 = 10$ páginas.
- ❖ Join con ciclos anidados e índice (INL) y pipelining.
 - No ayuda proyectar los campos innecesarios de la salida
- ❖ La columna de Join *sid* es llave de Sailors.
 - Al menos una tupla coincide , índice unclustered sober *sid* OK.
- ❖ La decisión de no colocar *rating*>5 antes del join esta basada en que se posee un índice sobre *sid* en Sailors.
- ❖ **Costo:** La selección de las tuplas de Reserves (10 I/Os); cada una debe coincidir Sailors ($1000*1.2$); total **1210 I/Os**.





Estimación del costo

- ❖ Para cada plan considerado se debe estimar el costo:
 - Se debe **estimar el costo** de cada operación en el árbol.
 - ◆ Depende de la cardinalidad de las entradas.
 - ◆ Ya se analizo como estimar el costo de las operaciones (explorar secuencialmente y con índice, joins, etc.)
 - Se debe **estimar el tamaño del resultado** para cada operación del árbol!
 - ◆ Usar información de cada relación de entrada.
 - ◆ Para las selecciones y joins, asumir independencia de predicados.
- ❖ Estas estimación del costo son muy inexacto, pero trabaja bien en la práctica. Ahora se conocen técnicas mas sofisticadas.



Estadísticas Catálogos

- ❖ Es necesaria la información de las relaciones y los índices. *Lo catálogos* comúnmente contienen:
 - Cantidad de tuplas (NTuplas) y páginas (NPáginas) de cada relación.
 - Cantidad de valores de llaves distintos (NKeys) y NPages para cada índice.
 - Altura del índice, mínimo/máximo valor de llave (Low/High) para cada índice.
- ❖ Los catálogos se modifican periódicamente.

Estimación de tamaño y factor de reducción

```
SELECT attribute list  
FROM relation list  
WHERE term1 AND ... AND termk
```

- ❖ Consideremos por bloques una consulta:
- ❖ El máximo número de tuplas es el resultado del producto de las cardinalidades de las relaciones de la cláusula FROM.
- ❖ *Factor de reducción (RF)* asociado con cada *término* refleja el impacto de cada *término* en la reducción del tamaño del resultado. *Cardinalidad del resultado* = Max # tuplas * producto de todos los RF's.
 - Esta implícito la *asunción* que los *términos son independientes!*
 - Término *col=valor* tiene $RF = 1/NKeys(I)$, dado un índice I sobre *col*
 - Término *col1=col2* tiene $RF = 1/MAX(NKeys(I1), NKeys(I2))$
 - Término *col>valor* tiene $RF = (High(I)-value)/(High(I)-Low(I))$