

Dpto. Ciencias e Ingeniería de la Computación
 Universidad Nacional del Sur

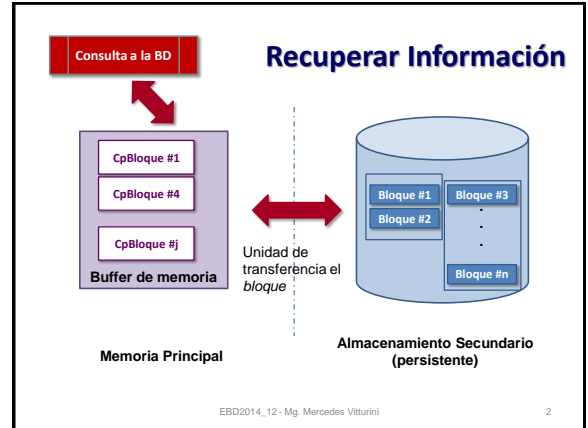
ELEMENTOS DE BASES DE DATOS

Segundo Cuatrimestre 2014

Clase 12:
Nivel Físico (Parte II)
Procesamiento y Optimización de Consultas




Mg. María Mercedes Vitturini
 [mvitturi@uns.edu.ar]



Leer (Q) / [input (B_q)]

Si bloque B_q con el dato Q *no está* en el Buffer de Memoria **ENTONCES**

Si no hay espacio en el Buffer de Memoria para B_q **ENTONCES**

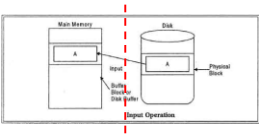
Liberar espacio para B_q

FIN-SI

Input (B_q)

FIN SI

Devolver Q



EBD2014_12 - Mg. Mercedes Vitturini 3

Escribir (Q) / [input (B_q) output (B_q)]

Si bloque B_q con el dato Q *no está* en el Buffer de Memoria **ENTONCES**

Si no hay espacio en el Buffer de Memoria para B_q **ENTONCES**

Liberar espacio para B_q

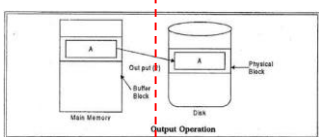
FIN-SI

Input (B_q)

FIN-SI

Escribir (Q)

Output (B_q)



EBD2014_12 - Mg. Mercedes Vitturini 4

Índices - Repaso

- Los índices aceleran el proceso de búsqueda de la información siguiendo una *clave de búsqueda*.
- Clave de búsqueda: no se requiere que sea una llave única

Tipos de Índices

Clave de búsqueda **Puntero**

- Índices ordenados.** Basados en una disposición ordenada de los valores de clave de búsqueda.
- Índices hash.** Basados en la distribución uniforme de los valores clave de búsqueda en *buckets*.
- Índice primario (clustering):** cuando el archivo de datos está ordenado secuencialmente por la clave de búsqueda del índice.
- Índice secundario (no clustering):** el orden del archivo de datos es distinto al orden de la clave de búsqueda del índice.
- Índice Denso:** contiene todos los valores de clave de búsqueda.
- Índice Ralo:** contiene algunos de los valores de la clave de búsqueda.

EBD2014_12 - Mg. Mercedes Vitturini 5

Índices - Repaso

- Los índices aceleran el proceso de búsqueda de la información según una *clave de búsqueda*.
 - Clave de búsqueda:** no se requiere que sea una llave única.
- El comportamiento de los índices y su organización varía en algún sentido según la clave de búsqueda sea:
 - Por uno o varios atributos.
 - Sea un índice único (un único registro asociado a cada clave de búsqueda) o no.
 - Si se admiten valores nulos en alguno/s de los atributos que forman la clave de búsqueda.
 - Es una superllave.

EBD2014_12 - Mg. Mercedes Vitturini 6

Punteros

“An index record, or index entry, consists of a search-key value and **pointers** to one or more records with that value as their search-key value. The pointer to a record consist of the identifier of a disk block and an offset within the disk block to identify the record within the block – Database System Concepts 6th Ed. A. Silberschatz”

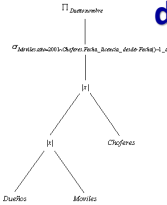
Pointers

“All Files are organized by using two basic constructs to link one piece of data with another piece of data: sequential storage and pointers. With sequential storage, one field or record is stored right after another field or record. Although simple to implement and use, sometimes sequential storage is not the most efficient way to organize data. A pointer is a field of data that can be used to locate a related field or record of data. In most cases, a pointer contains the address, or locations of the associated data. Pointers are used in a wide variety of data storage structures ... We define pointer here only because you need to know what a pointer is for understanding file organizations. You will likely never work directly with pointers because the DBMS will handle all pointer use and maintenance automatically” – Modern Database Management 9th Ed. Jeffrey A. Hoffer (2009)

EBD2014_12 - Mg. Mercedes Vitturini

7

Procesamiento y optimización de Consultas



Procesamiento de Consultas

Actividades involucradas:

1. Análisis y traducción:

- Se comprueba sintaxis,
- Se traduce la consulta a una representación interna basada en el *Álgebra Relacional (AR)*.

2. Optimización

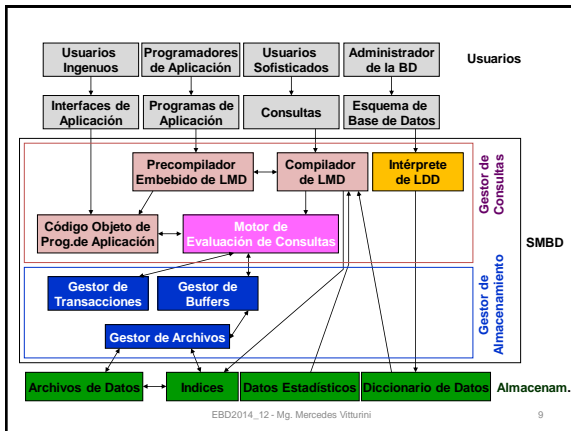
- En base a una *expresión en AR optimizada, índices existentes y datos estadísticos*, especifica cómo evaluar cada operación.

3. Evaluar

- Ejecuta el plan de ejecución más adecuado.

EBD2014_12 - Mg. Mercedes Vitturini

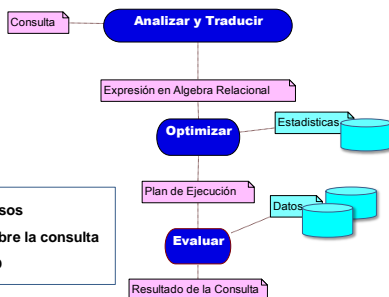
10



EBD2014_12 - Mg. Mercedes Vitturini

9

Procesamiento de Consultas



- Procesos
- I/O sobre la consulta
- I/O BD

EBD2014_12 - Mg. Mercedes Vitturini

11

Traducción a AR

- Una misma consulta puede tener distintas expresiones en el álgebra relacional.

```

SELECT saldo
FROM cuentas
WHERE saldo >= 50
    
```

$$\sigma_{\text{saldo} \geq 50}(\pi_{\text{saldo}}(\text{cuentas}))$$

ó

$$\pi_{\text{saldo}}(\sigma_{\text{saldo} \geq 50}(\text{cuentas}))$$

Estrategia general:

1. Producto cartesiano entre las relaciones del **FROM**
2. Selección con las condiciones del **WHERE**.
3. Realizar una **proyección** de las columnas del **SELECT**

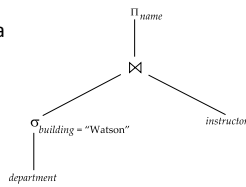
árbol canónico

EBD2014_12 - Mg. Mercedes Vitturini

12

Árbol y Plan de Ejecución

- Dada una consulta, se construye un **árbol** que consta de expresiones del AR.
- Las hojas contienen a las relaciones y los nodos interiores las operaciones algebraicas.
- A partir de la expresión del AR se arma un **plan de ejecución** agregando **anotaciones**



CONSULTA SQL ⇔ Vs. ÁRBOLES ⇔ Vs. PLANES DE EJECUCIÓN

EBD2014_12 - Mg. Mercedes Vitturini

13

Optimizador de Consultas

¿Por qué existen varios planes de ejecución para una consulta SQL?

- Porque *algebraicamente se puede escribir de maneras distintas* lógicamente equivalentes.
 - Y porque, dada una expresión algebraica *existen distintos algoritmos "físicos" para resolverla.*
- El **optimizador de consultas** es el componente del DBMS responsable de generar distintos planes y elegir un plan de ejecución eficiente.

EBD2014_12 - Mg. Mercedes Vitturini

14

Costo de una consulta

- En el costo de evaluar una consulta intervienen varios factores: tiempo acceso a disco, tiempo de transferencia de datos, tiempo de CPU, capacidad de ejecución en paralelo o distribuida, etc.

"In large database systems, **the cost to access data from disk is usually the most important cost**, since disk accesses are slow compared to in-memory operations. Moreover, CPU speeds have been improving much faster than have disk speeds. Thus, it is likely that **the time spent in disk activity will continue to dominate the total time to execute a query**. The CPU time taken for a task is harder to estimate since it depends on low-level details of the execution code. Although real-life query optimizers do take CPU costs into account, for simplicity in this book we ignore CPU costs and use only disk-access costs to measure the cost of a query-evaluation plan."

EBD2014_12 - Mg. Mercedes Vitturini

15

Costo de una consulta

We use the **number of block transfers** from disk and the **number of disk seeks** to estimate the cost of a query-evaluation plan. If the disk subsystem takes an average of t_r seconds to transfer a block of data, and has **an average block-access time** (disk seek time plus rotational latency) of t_s seconds, then **an operation that transfers b blocks and performs S seeks would take $b * t_r + S * t_s$ seconds**. The values of t_r and t_s must be calibrated for the disk system used, but typical values for high-end disks today would be $t_s = 4$ milliseconds and $t_r = 0.1$ milliseconds, assuming a 4-kilobyte block size and a transfer rate of 40 megabytes per second. We can refine our cost estimates further by distinguishing block reads from block writes, since block writes are typically about twice as expensive as reads (this is because disk systems read sectors back after they are written to verify that the write was successful).

Database System Concepts 6th Ed. A. Silberschatz"

EBD2014_12 - Mg. Mercedes Vitturini

16

Optimización

Objetivo de la optimización: "minimizar los accesos y transferencias desde disco", para ello se debe considerar:

- Adecuada **expresión algebraica**.
- Diversos **algoritmos disponibles para procesar consultas algebraicas** que implementan las operaciones del AR (selección, proyección, unión, intersección, join, diferencia)
- **Distribución y organización física de los datos**, hash, clustering, sin orden, etc.
- **Índices existentes**.
- **Información estadística existente**, tamaño de los archivos, cantidad de registros, cantidad de registros por bloque, cantidad de bloques, etc.

EBD2014_12 - Mg. Mercedes Vitturini

17

Optimización

- Un **plan de ejecución** se construye especificando además estrategias o **anotaciones**:

Por ejemplo:

- Evaluar la consulta usando el índice secundario i_j sobre saldo.
 - Evaluar la consulta haciendo recorrido secuencial sobre la relación
- Entre los distintos planes de ejecución posibles el optimizador elige el que se estima más económico.

EBD2014_12 - Mg. Mercedes Vitturini

18

Medidas de Costo

Típicamente, el acceso a disco predomina en el costo para resolver una consulta. Considerando:

- ⇨ **b** número de bloques a transferir.
- ⇨ **S** número de bloques a buscar.
- ⇨ **t_T** tiempo para transferir un bloque.
- ⇨ **t_S** tiempo para la búsqueda de un bloque (localización del bloque en disco).

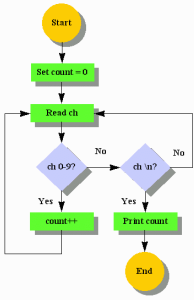
El costo de transferir **b** bloques con sus **S** búsquedas

$$b * t_T + S * t_S$$

EBD2014_12 - Mg. Mercedes Vitturini

20

Algunos Algoritmos para resolver Expresiones Algebraicas



Simplificación

Una estimación más acertada debería considerar otros factores, que no son tenidos en cuenta en una análisis de propósito general.

- Otros factores no considerados:
 - Velocidad del procesador.
 - Tamaño de buffer.
 - Si los datos han sido usados recientemente.
- Las estimaciones son sobre el peor caso.

EBD2014_12 - Mg. Mercedes Vitturini

21

Referencias usadas

- **b** número de bloques a transferir.
- **S** número de búsquedas
- **t_T** tiempo para transferir un bloque
- **t_S** tiempo para una búsqueda
- **b_r** número de bloques ocupados por la relación *r*.
- **h_i** peso del índice B+-tree.

EBD2014_12 - Mg. Mercedes Vitturini

22

Selección + File Scan

Estimación para File scan: se considera una operación de selección sobre una relación cuya tuplas están almacenadas consecutivamente.

- **Algoritmo A1 (búsqueda lineal):** recorrer secuencialmente toda la relación y verificar registro a registro si satisfacen la condición de selección.
 - Costo estimado: $t_S + b_r * t_T$
 - Si la selección es igualdad sobre un atributo clave candidata: $(b_r / 2) * t_T + t_S$
- Características: File scan se puede utilizar para cualquier condición de selección, orden de registros de datos en el archivo y disponibilidad de índices.

EBD2014_12 - Mg. Mercedes Vitturini

23

Selección + Índice B+Tree



Index Scan – Algoritmo de búsqueda con índice primario

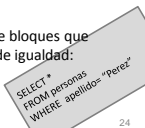
⇨ **Algoritmo A2** – índice primario (clustering), condición: igualdad sobre atributo clave candidata – ie. se espera retornar un único registro como respuesta.

– Costo estimado: $(h_i + 1) * (t_T + t_S)$

⇨ **Algoritmo A3** – índice primario (clustering), condición: igual sobre atributo no clave candidata – ie. se espera recuperar varios registros.

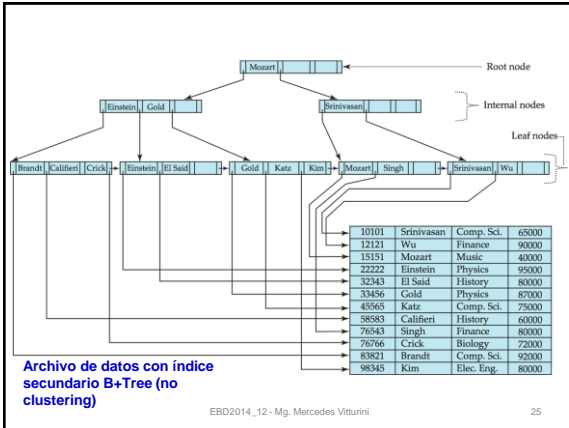
– Los registros están consecutivos, sea *b* el número de bloques que contienen los registros que satisfacen la condición de igualdad:

– Costo estimado: $h_i * (t_T + t_S) + t_S + t_T * b$



EBD2014_12 - Mg. Mercedes Vitturini

24



EBD2014_12 - Mg. Mercedes Vitturini

25

Selección + Índice Secundario B+Tree

Index Scan – Con índice secundario.

- **Algoritmo A4** - índice secundario, condición por igualdad. Los registros de datos una distribución distinta a la de la clave del índice.
 - Para un índice secundario único, recupera un único registro.
 - Costo estimado: $(h_i + 1) * (t_r + t_s)$
- Índice secundario no único. La igualdad espera recuperar n registros. En el peor caso, cada uno de los n registros que coinciden con la búsqueda están en bloques distintos.
- Costo estimado: $(h_i + n) * (t_r + t_s)$

EBD2014_12 - Mg. Mercedes Vitturini

26

Selección por rango + Índice

Selección y comparación: $a \leq v$ (ó $<$, $>$, \geq) usando índices.

- **Algoritmo A5** – índice primario, selección por comparación.
 - \geq , usar el índice para encontrar la primera tupla. Seguir recorriendo el archivo de datos en forma secuencial.
 - \leq recorrer secuencialmente el archivo de datos hasta alcanzar la primera tupla que es mayor a valor.
 - Costo estimado: $h_i * (t_r + t_s) + b * t_r$
- **Algoritmo A6** – índice secundario, comparación.
 - Recorrer secuencialmente el índice, por cada entrada de índice que cumple la condición y seguir su puntero.
 - Costo estimado: $(h_i + n) * (t_r + t_s)$
 - Conviene usar A6 si el número de registros a recuperar es bajo, sino es mejor recorrer secuencialmente el archivo de datos (A1).

EBD2014_12 - Mg. Mercedes Vitturini

27

Join – Loop anidado

Algoritmo $r \bowtie s$ – Loop anidado

```

PARA cada  $t_r$  in  $r$  HACER
  PARA cada  $t_s$  in  $s$  HACER
    SI  $(t_r, t_s)$  satisface condición de join ENTONCES
      Agregar  $(t_r, t_s)$  al resultado
    FIN-SI
  FIN-PARA
FIN-PARA
    
```

EBD2014_12 - Mg. Mercedes Vitturini

28

student 5000 registros en 100 bloques
 takes 10000 registros en 400 bloques

student \bowtie takes

No hay índices para *student* ni *takes*. Análisis del algoritmo join con loop anidado.

Se deben examinar $5000 * 10000 = 50 * 10^6$ pares de tuplas.

Mejor caso: "cargar" en memoria ambas relaciones una sola vez. Se requieren a lo sumo $100+400 = 500$ transferencias de bloque y 2 accesos a disco.

Con *student* como relación externa y *takes* como relación interna

Peor caso: bloques a transferir (t_r) $5000 * 400 + 100 = 2.000.100$, más $5000+100 = 5100$ accesos a disco (t_s).

Con *takes* como relación externa y *student* como relación interna

Peor caso: t_r $10000 * 100 + 400 = 1.000.400$, con t_s $10000 + 400 = 10400$. t_r es significativamente menor, y creció t_s .

Asumiendo $t_s = 4$ milisegundos and $t_r = 0.1$ milisegundos *takes* como relación externa tiene mejor desempeño

EBD2014_12 - Mg. Mercedes Vitturini

29

Otros algoritmos

- **Conjunción:** $\sigma_{\theta_1 \wedge \theta_2 \wedge \dots \wedge \theta_n}(r)$
- **Disyunción:** $\sigma_{\theta_1 \vee \theta_2 \vee \dots \vee \theta_n}(r)$
- **Negación:** $\sigma_{\neg \theta}(r)$
- **Ordenamiento,**
- **Join**

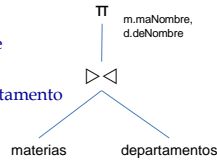
EBD2014_12 - Mg. Mercedes Vitturini

30

Plan de ejecución

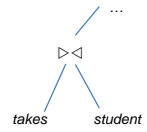
- Para la resolución de consultas primero se construyen representaciones algebraicas, en la forma de árbol.
- Por ejemplo:

```
SELECT m.maNombre, d.deNombre
FROM materias m, departamentos d
WHERE m.departamento = m.departamento
```



Árbol Algebraico

- En las *hojas* están las *relaciones*.
- Los *nodos interiores* representan *operaciones algebraicas* ($\pi, \sigma, \bowtie, \dots$).
- Un *arco hacia los sucesores* representa los *parámetros de entrada del sucesor*.
- Un *arco hacia su antecesor* representa el *resultado o output de la operación*.
- La salida de *la raíz* es el *resultado final de la expresión*.

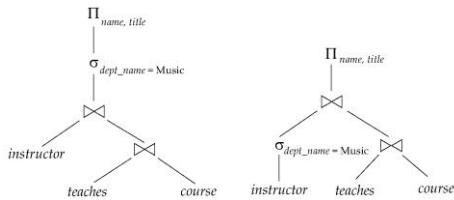


Evaluaciones:

- Cada *operación algebraica* tiene un ó más algoritmos que la *resuelven*.
- Cada *relación* tiene una *organización y métodos de acceso*.

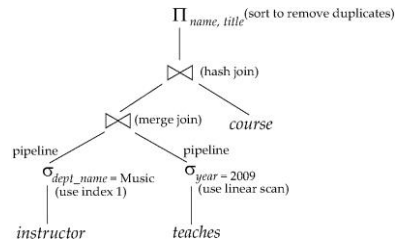
Árbol algebraico

- Expresiones equivalentes *implican diferentes algoritmos y planes y costos distintos*.
- Ejemplo:



Plan de ejecución

- En el *plan de ejecución* se define además exactamente los algoritmos a usar en cada operación.



Observaciones generales

- La *diferencia de costo de distintos planes de ejecución puede ser muy grande*.
- Pasos para optimización de consultas basada en costos:
 1. Generar expresiones lógicamente equivalentes, usando reglas de equivalencia.
 2. Agregar anotaciones con las alternativas.
 3. Elegir el plan más económico basado en el costo estimado.
- Herramientas para la estimación de costos:
 - Información estadística de las relaciones.
 - Estimación estadística para resultados intermedios.
 - Costo estimado de los algoritmos

Reglas de equivalencias

1. **Cascada de σ :** $\sigma_{\theta_1 \wedge \theta_2}(E) = \sigma_{\theta_1}(\sigma_{\theta_2}(E))$
2. **Commutativa de σ :** $\sigma_{\theta_1}(\sigma_{\theta_2}(E)) = \sigma_{\theta_2}(\sigma_{\theta_1}(E))$
3. **Cascada de π :** $\pi_{L_1}(\pi_{L_2}(E)) = \pi_{L_1}(E) = \pi_{L_1 \cap L_2}(E)$
4. **Commutativa de σ con respecto a π :** $\pi_X(\sigma_C(E)) = \sigma_C(\pi_X(E))$, si C referencia a atributos de X
5. **Commutativa del Producto Cartesiano (y Join):**
 $E_1 \times E_2 = E_2 \times E_1$
6. **Distributiva σ y Producto Cartesiano (y Join):** $(\sigma_C(E_1 \times E_2)) = \sigma_{C_1}(E_1) \times \sigma_{C_2}(E_2)$, con $C = C_1 \cup C_2$.
7. **Distributiva π y Producto Cartesiano:** $(\pi_L(E_1 \times E_2)) = \pi_{L_1}(E_1) \times \pi_{L_2}(E_2)$, con $L = L_1 \cup L_2$.
8. ...

Heurísticas

- Elegir árboles sesgados a izquierda.
- Descomponer las selecciones conjuntivas en selecciones simples.
- Llevar la selección lo más cercano a la hoja del árbol.
- Reemplazar productos cartesianos seguidos selección por joins. Evitar los productos cartesianos.
- Descomponer la lista de atributos de proyección y llevarlas lo más cercano posible a las hojas.
- Realizar primero los joins más selectivos (resultado menor cantidad de tuplas).
- En cada nodo, optar por los planes menos costosos.
- Considerar los índices.
- Mantener siempre que sea posible los resultados en memoria (pipeline)

EBD2014_12 - Mg. Mercedes Vitturini

37

Pasos para la optimización

1. Construir el árbol canónico.
2. Construir árboles equivalentes, utilizando alguna heurística.
No sobredimensionar la búsqueda del mejor plan.
3. Por cada árbol, hacer tantos planes de ejecución como combinaciones interesantes existan.
4. Hacer las anotaciones.

EBD2014_12 - Mg. Mercedes Vitturini

38

Ejemplo

- Sean las relaciones
 - Libro (nroLib (4), tituloLib(84), pagLib(8)) – Tamaño registro 96 bytes
 - Autor (nroAut (4), nombreAut(40), nacionalidad (10)) – Tamaño reg 54 bytes
 - Escrito_por (nroLib, nroAut) – Tamaño registro 8 bytes
- Tamaño de bloque = 4000 bytes
- Cantidad de Tuplas en libro $T_{libro} = 100000$ registros
- Cantidad de Tuplas en escrito_por $T_{escrito_por} = 130000$ registros
- Cantidad de Tuplas en autor $T_{autor} = 5000$ registros
- Índices
 - índice B+ cluster sobre nroLib para libro – índice sobre atributo clave
 - índice B+ cluster sobre nroLib y nroAut para escrito_por – índice sobre atributo clave
 - índice B+ cluster sobre nroAut para autor – índice sobre atributo clave
 - índice B+ secuenciario sobre nacionalidad para autor

EBD2014_12 - Mg. Mercedes Vitturini

39

Consulta

```
SELECT l.nroLib, l.tituloLib
FROM autor a, escrito_por ep, libro l
WHERE a.nroAut = ep.nroAut
      AND ep.nroLib = l.nroLib
      AND a.nacionalidad = "argentino"
```

- Consulta canónica

$\pi_{l.nroLib, l.tituloLib}(\sigma_{a.nacionalidad="argentino"}(autor \bowtie escrito_por \bowtie libro))$

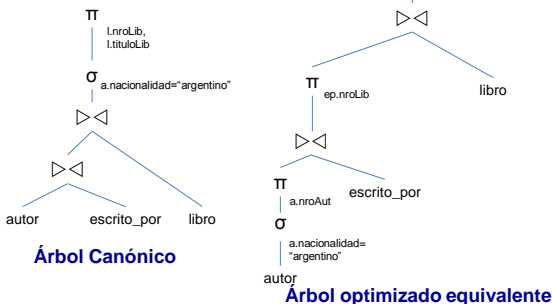
- Consulta equivalente optimizada

$\pi_{l.nroLib, l.tituloLib}((libro) \bowtie \pi_{l.nroLib}(\pi_{l.nroAut}(\sigma_{a.nacionalidad="argentino"} autor) \bowtie escrito_por))$

EBD2014_12 - Mg. Mercedes Vitturini

40

Árboles



EBD2014_12 - Mg. Mercedes Vitturini

41

Bloques ocupados

- Longitud de *autor* $L_a=54$ by
- Bloques que ocupa :
 - $[tamaño_bloque/L_a] = [4000/54] = 74$ registros por bloque
 - $[5000/74] = 68$ bloques.
- Longitud de *escrito_por* $L_{ep}=8$ by
- Bloques que ocupa escrito_por:
 - $[tamaño_bloque/L_{ep}] = [4000/8] = 500$ registros por bloque
 - $[130000/500] = 260$ bloques.
- Longitud de *libro* $L_l=96$ by
- Bloques que ocupa libro:
 - $[tamaño_bloque/L_l] = [4000/96] = 41$ registros por bloque
 - $[100000/41] = 2440$ bloques.

EBD2014_12 - Mg. Mercedes Vitturini

42

Costo estimado por índices

- índice B+ cluster sobre nroLib para *libro* – índice sobre atributo clave
 - Tamaño registro índice 12
 - [4000 / 12] 334 registros índices por bloque.
 - 2440 bloques de datos, nivel del árbol 1
- índice B+ cluster sobre nroLib y nroAut para escrito_por (nroAut+nroLib)– índice sobre atributo clave
 - Tamaño registro índice 16
 - [4000 / 16] 250 registros índices por bloque.
 - 260 bloques de datos, nivel del árbol 2

EBD2014_12 - Mg. Mercedes Vitturini

43

Costo estimado por índice

- índice B+ cluster sobre nroAut para *autor* – índice sobre atributo clave.
 - Tamaño registro índice 12
 - [4000 / 12] 334 registros índices por bloque.
 - 68 bloques de datos, nivel del árbol 1.
- índice B+ secundario sobre nacionalidad para autor
 - Tamaño registro índice 18
 - [4000 / 18] 223 registros índices por bloque. 1 bloque para el índice y un bloque por indirección.
 - Suponiendo una distribución equitativa entre 50 nacionalidades, 100 autores por nacionalidad.

EBD2014_12 - Mg. Mercedes Vitturini

44

Análisis del plan ejecución

$\sigma_{\text{nacionalidad} = \text{"argentino"}}$

- Costo File scan = **68 bloques** ✓ $\rightarrow t_s + 68 * t_b$
- Costo usando índice secundario: 1 acceso al índice + 1 acceso indirección + en el peor de los casos cada autores exige traer el bloque lo aloja, 102 accesos a bloque. $\rightarrow 2t_s + 100 * (t_s + t_b)$
- π_{nroAut} pipeline en memoria

EBD2014_12 - Mg. Mercedes Vitturini

45

Análisis plan de ejecución

$\text{nroAut} \triangleright \triangleleft \text{escrito_por}$

- Resultado anterior en memoria. Costo join con FileScan = 260 bloques para 100 autores $\rightarrow 100 * (t_s + 260 * t_b)$
- Usando el índice B+ de escrito_por, los datos están en el índice y asumiendo distribución uniforme, a lo sumo 100 autores, por dos niveles, **200 bloques** para el peor de los casos ✓ $\rightarrow 200 * (t_s + t_b)$
- π_{nroLib} pipeline en memoria

EBD2014_12 - Mg. Mercedes Vitturini

46

Análisis del plan de ejecución

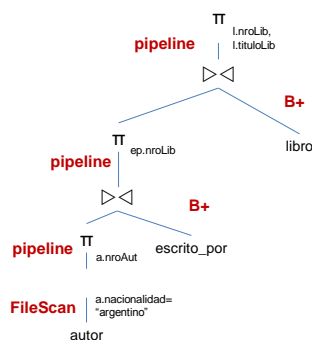
$\text{nroLib} \triangleright \triangleleft \text{libro}$

- Resultado anterior en memoria. Costo join con FileScan = 2600 * 2440/2 bloques $\rightarrow 2600 * (t_s + 2440/2 * t_b)$
- Usando el índice B+ de libros y asumiendo distribución uniforme, a lo sumo 5000 autores en 50 nacionalidades, 100 autores argentinos, 130000 escrito_por de 5000 autores, 26 libros por autor. Libros para la operación 2600, con nivel de árbol 1, **(1 + 1) * 2600 bloques** para el peor de los casos $\rightarrow 2600 * 2 * (t_s + t_b)$ ✓

EBD2014_12 - Mg. Mercedes Vitturini

47

Plan de Ejecución propuesto



EBD2014_12 - Mg. Mercedes Vitturini

48

Temas de la clase de hoy

- Procesamiento de consultas
 - Etapas
 - Algoritmos
- Optimización de consultas
 - Expresiones algebraicas equivalentes.ç
 - Plan de ejecución
 - Ejemplo
- **Bibliografía:**
 - *Database System Concepts* – Abraham Silberschatz – Capítulos 12 y 13 (ed. 2010)

EBD2014_12 - Mg. Mercedes Vitturini

49