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

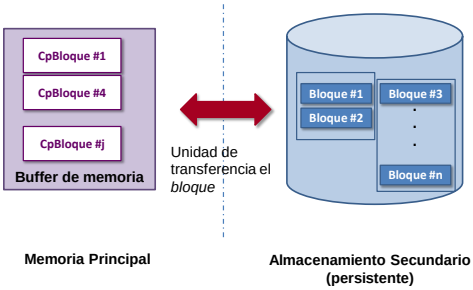
ELEMENTOS DE BASES DE DATOS

Segundo Cuatrimestre 2013

Clase 12: Nivel Físico – Procesamiento y Optimización de Consultas

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

Transferencia Memoria Disco



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

Clave de búsqueda	Puntero

Tipos de Índices

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

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 5th Ed. A. Silberschatz" (2006)**

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)**

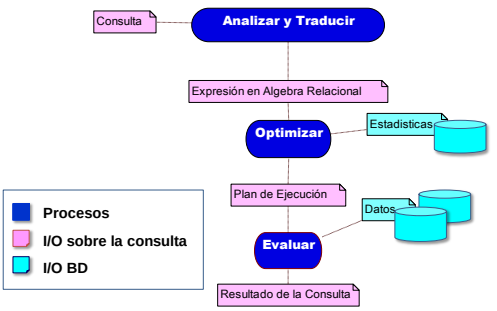
Procesamiento y optimización de Consultas

Procesamiento de Consultas

Pasos involucrados:

1. **Análisis y traducción**:
   - comprueba sintaxis,
   - 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.

Procesamiento de Consultas



Traducción a AR

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

SELECT saldo
FROM *cuentas*
WHERE saldo>= 50

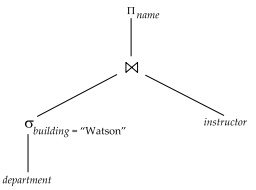
$\sigma_{saldo >= 50}(\pi_{saldo}(cuentas))$
ó
$\pi_{saldo}(\sigma_{saldo >= 50}(cuentas))$

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 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

Optimizador de Consultas

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

- Porque *algebraicamente se puede escribir de distintas maneras* 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 y elegir un plan de ejecución eficiente.

Optimización

Objetivo de la optimización: "*minimizar los accesos a disco*", para ello se considera:

- 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.

Optimización

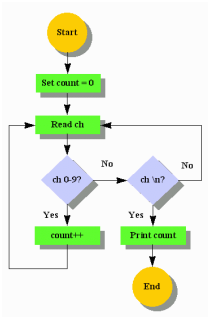
- Un **plan de ejecución** se construye especificando además estrategias o **anotaciones**:

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 se elige el que se estima más económico.

Estas transparencias proveen sólo una referencia a los temas. Para su estudio debe remitirse a la bibliografía.



Algoritmos para resolver Expresiones Algebraicas

Medidas de Costo

Dado que típicamente el acceso a disco predomina en el costo, si consideramos que:

- **b** número de bloques a transferir.
- **S** número de búsquedas
- t_T tiempo para transferir un bloque
- t_s tiempo para una búsqueda

Así, el costo de transferir **b** bloques con sus **S** búsquedas

$$b * t_T + S * t_s$$

Notación

- **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 que ocupa la relación r.
- h_i peso del índice.

Estimación

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.

Selección – File Scan

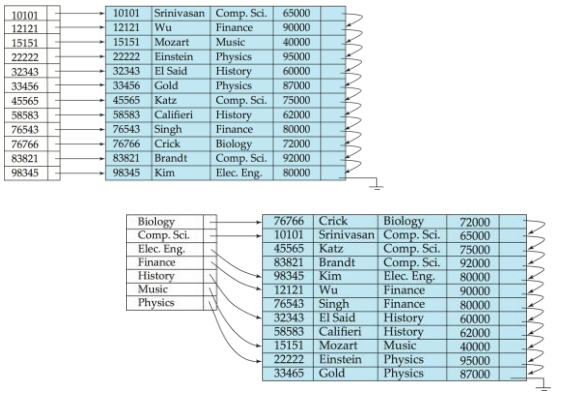
File scan

- **Algoritmo A1** (búsqueda lineal): recorrer secuencialmente cada bloque del archivo para verificar si sus registros 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 loa registros en el archivo y disponibilidad de índices*.

Selección con índices

Index Scan – Algoritmos de búsqueda con índices primario.

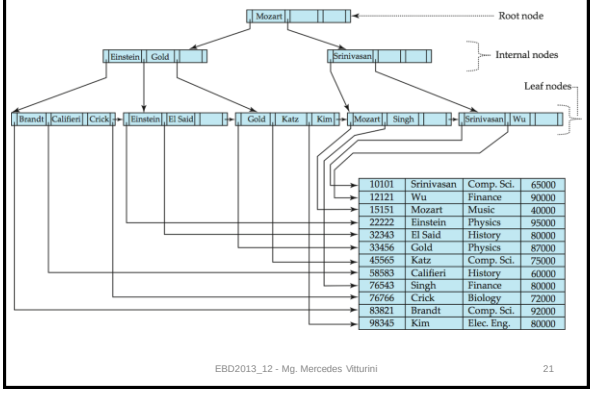
- **Algoritmo A2 –** índice primario (clustering), condición: igual sobre atributo clave candidata – ie, se espera retornar un único registro como respuesta.
  - **Costo estimado:** $(h_i + 1) * (t_T + t_S)$
- **Algoritmo A3 –** índice primario (clustering), condición: igual sobre atributo *no clave candidata* – ie. se espera recuperar varios registros.
  - Los registros están consecutivos, sea b de bloques que contienen los registros que coinciden
  - **Costo:** $h_i * (t_T + t_S) + t_S + t_T * b$



Selección usando índices

Index Scan – Con índice secundario.

- **Algoritmo A4** - índice secundario, condición por igualdad.
  - Para **una clave candidata**, recupera un único registro.
  - **Costo estimado: $(h_i + 1) * (t_T + t_S)$**
  - Si la **igualdad es de atributos no únicos**, se espera recuperar varios registros. En el peor de los casos, cada uno de los *n* registros que coinciden con la búsqueda están en bloques distintos.
  - **Costo estimado: $(h_i + n) * (t_T + t_S)$**



Selección + Comparación

Selección y comparación: a <= v (ó <, >, >=). Usando índices.

- **Algoritmo A5 –** índice primario, comparación.
  - >=, usar el índice para encontrar la primera tupla. Seguir recorriendo el archivo de datos en forma secuencial.
  - <= recorrer secuencialmente el archivo de datos hasta alcanzar la primera tupla que es mayor a v.
- **Algoritmo A6 –** índice secundario, comparación.
  - Recorrer secuencialmente el índice, por cada entrada del índice que cumple la condición seguir su puntero para obtener el registro de datos.
  - **Usar A6 conviene si el número de registros a recuperar es bajo, sino es preferible recorrer secuencialmente el archivo de datos (A1).**

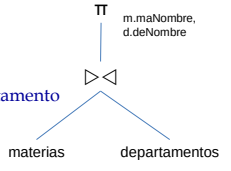
Otros algoritmos

- **Conjunción**: $\sigma_{\theta 1 \wedge \theta 2 \wedge \ldots \wedge \theta n}(r)$
- **Disyunción:** $\sigma_{\theta 1 \vee \theta 2 \vee \ldots \vee \theta n}(r)$
- **Negación:** $\sigma_{\neg \theta}(r)$
- **Ordenamiento,**
- **Join**

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



Estas transparencias proveen sólo una referencia a los temas. Para su estudio debe remitirse a la bibliografía.

Árbol Algebraico

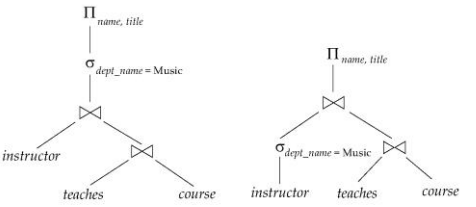
- En las hojas están las *relaciones*.
- Los nodos interiores representan *operaciones algebraicas* (π, σ, ▷◁, etc.)
- Un *arco hacia los sucesores* representa los parámetros de entrada del sucesor.
- Un *arco hacia su antecesor* representa el resultado o output de la operación.
- La salida de la raíz es el resultado final de la expresión.

Donde:

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

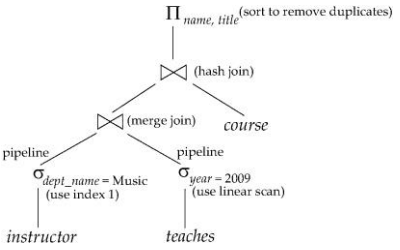
Árbol algebraico

- Expresiones equivalentes implicaran 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. **Conmutativa de σ:** $\sigma_{\theta 1}(\sigma_{\theta 2}(E)) = \sigma_{\theta 2}(\sigma_{\theta 1}(E))$
3. **Cascada de π:** $\pi_{L1}(\pi_{L2}(E)) = \pi_{L1}(E) = \pi_{L1 \cap L2}(E)$
4. **Conmutativa de σ con respecto a π:** $\pi_X(\sigma_C(E)) = \sigma_C(\pi_X(E))$, si C referencia a atributos de X
5. **Conmutativa del Producto Cartesiano (y Join):** $E_1 X E_2 = E_2 X E_1$
6. **Distributiva σ y Producto Cartesiano (y Join):** $(\sigma_C(E_1 X E_2)) = \sigma_{C1}(E_1) X \sigma_{C2}(E_2)$, con $C = C_1 U C_2$.
7. **Distributiva π y Producto Cartesiano:** $(\pi_L(E_1 X E_2)) = \pi_{L1}(E_1) X \pi_{L2}(E_2)$, con $L = L_1 U L_2$.
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.
- Remplazar 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)

Estas transparencias proveen sólo una referencia a los temas. Para su estudio debe remitirse a la bibliografía.

Pasos para la optimización

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

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+ secuendario sobre nacionalidad para autor

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"

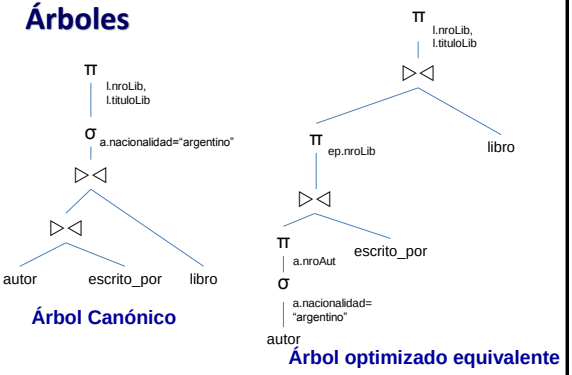
- Consulta canónica

$\pi_{l.nroLib,\ l.tituloLib}(\sigma_{a.nacionalidad="argentino"}(autor \rhd\lhd escrito\_por \rhd\lhd libro))$

- Consulta equivalente optimizada

$\pi_{l.nroLib,\ l.tituloLib}((libro) \rhd\lhd \pi_{l.nroLib}(\pi_{l.nroAut}(\sigma_{a.nacionalidad="argentino"} autor) \rhd\lhd escrito\_por))$

Árboles



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.**

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

Estas transparencias proveen sólo una referencia a los temas. Para su estudio debe remitirse a la bibliografía.

Costo estimado por índice

- índice B+ cluster sobre nroAut para *autor* – índice sobre atributo clave.
  - Tamaño registro índice 12
  - [4000 / 12] 334registros 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.

Análisis del plan ejecución

$\sigma_{nacionalidad= \text{“argentino”}}$

- Costo File scan = **68 bloques** ✓
- 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.

- π_{nroAut} pipeline en memoria

Análisis plan de ejecución

- nroAut ▷◁ escrito_por
- Resultado anterior en memoria. Costo join con FileScan = 260 bloques para 100 autores

- 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 ✓

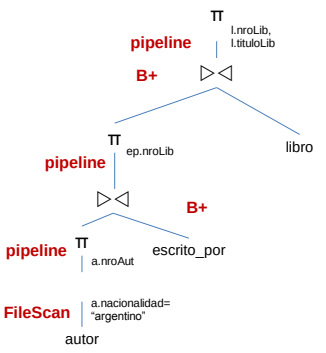
- π_{nroLib} pipeline en memoria

Análisis del plan de ejecución

- nroLib ▷◁ libro
- Resultado anterior en memoria. Costo join con FileScan = 2600 * 2440/2 bloques

- 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 arbol 1, (**1 + 1) * 2600 bloques** para el peor de los casos ✓

Plan de Ejecución propuesto



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 13 y 14 (ed. 2005)

Estas transparencias proveen sólo una referencia a los temas. Para su estudio debe remitirse a la bibliografía.