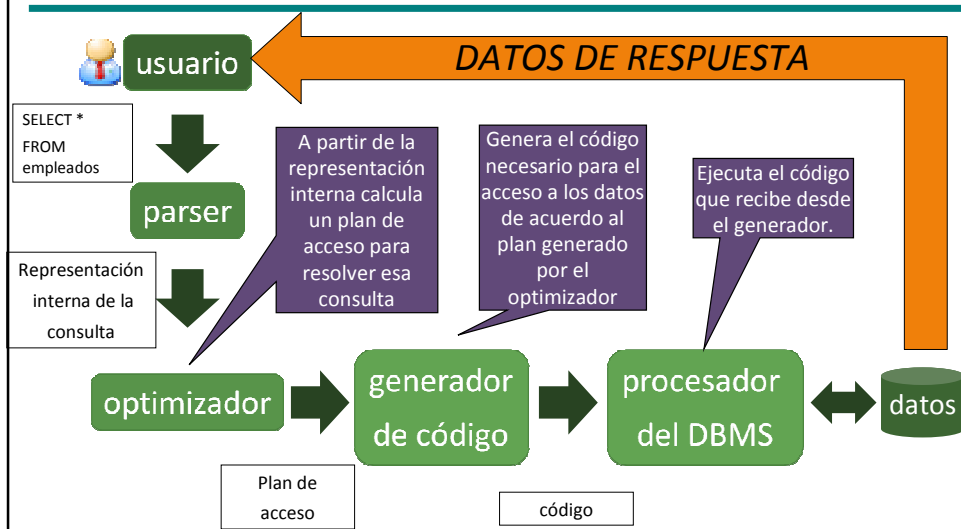


# Procesamiento y Optimización de Consultas

[EN 18]

[Ramakrishnan - Gehrke 12]

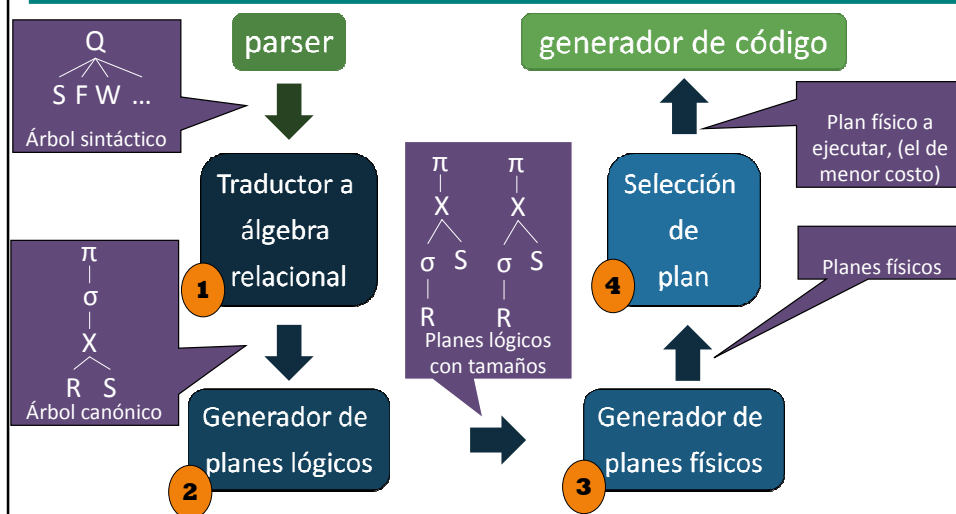
## ¿Cómo se resuelven las consultas?



## Estrategias usuales de los optimizadores

- Proceso detallado de Optimización.
- Optimización Heurística
  - Basada en *equivalencia* de las expresión del álgebra y ciertas estrategias básicas para limitar el tamaño de los resultados
- Optimización por Costos
  - Basada en *estimaciones* y datos del catálogo que permiten seleccionar un mejor plan de acceso.

## Proceso de Optimización



## Ejemplo de Optimización (1)

- Resolvamos esta consulta sobre las tablas:

- empleados(nombre, edad, salario, depto)
- departamentos(nroD, nombreD, piso, gerente)

```
select e.nombre, d.piso
from departamentos d,
empleados e
where e.depto = d.nroD
and e.salario > 30000
```

Traductor a  
álgebra  
relacional

1

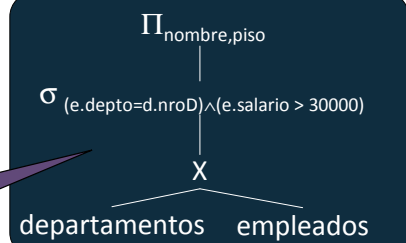
$\Pi_{\text{nombre, piso}}(\sigma_{\text{Salario} > 30000}(\text{empleados} * \text{departamentos}))$

Se escribe la  
consulta en  
álgebra relacional

1.1

Se genera el  
árbol canónico

1.2

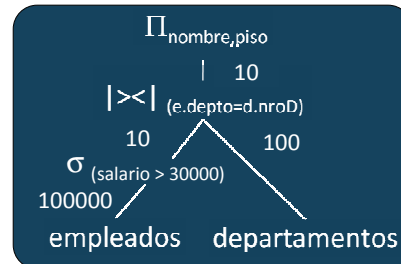
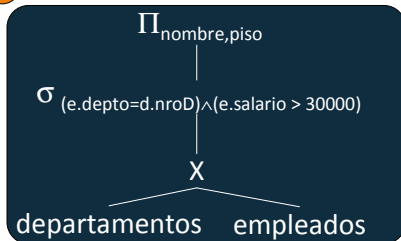


## Ejemplo de Optimización (2)

Generador de  
planes lógicos

2

- A partir del árbol canónico se generan planes lógicos.
- Se usan heurísticas y se agregan datos de tamaño.

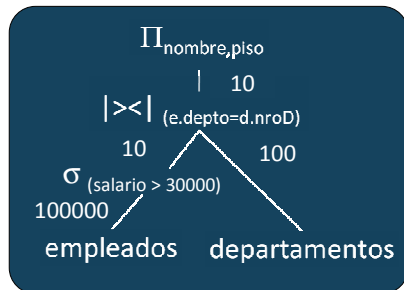


Parámetro	Valor
Tamaño de EMPLEADOS (tuplas)	100.000
Tamaño de DEPARTAMENTOS (tuplas)	100
Selectividad de $\sigma_{\{\text{salario} > 3000\}}$	1/10.000

## Ejemplo de Optimización (3)

3  
Generador de  
planes físicos

- Para cada plan lógico
- Se consideran diferentes implementaciones



Parámetro	Valor
Tamaño de EMPLEADOS (tuplas)	100.000
Tamaño de DEPARTAMENTOS (tuplas)	100
Selectividad de $\sigma_{(\text{salario} > 3000)}$	1/10.000
Cantidad de bloques para EMPLEADOS	2000
Cantidad de bloques para DEPTOS.	10
Índices sobre EMPLEADOS	B+ en salario
Índices sobre DEPARTAMENTOS	Hash en nroD

Operación	implementaciones		
$\sigma$	Busqueda lineal	Busqueda Binaria	Usar Indice
$ X $	Loop anidado	Loop único	SortMerge

## Ejemplo de Optimización (4)

4  
Selección  
de  
plan

- Calculo los costos (cant. de accesos a disco)

Operador	Implementación	Costo
$\sigma_c(R)$	Busqueda Lineal	$b_R$
	Busqueda Binaria	$\log_2 b_R + S_c$
	Uso de Indice	$\log_k  R  + S_c$
$R >< _c T$	Loop Anidado	$b_R + (b_R * b_T) + (j_s *  R  *  T ) / bfr_{RS}$

Costo del plan A = 58600

Costo del plan B = 700

Costo del plan C = 10



Plan C al Generador  
de Código

## Resumen del Proceso de Optimización

- Generación del Álgebra (Árbol Canónico)
- Generación de planes lógicos (Optimización Heurística)
  - Implica la aplicación de determinadas estrategias (heurísticas) y consultas al catálogo para tamaños de las relaciones para transformar el árbol original.
- Generación de planes físicos (Optimización por Costos)
  - Implica asociar a cada operación de los planes lógicos generados una o más implementaciones.
  - Qué implementación depende de las estructuras de datos disponibles.
- Selección del Plan final (Optimización por Costos)
  - Implica la evaluación de los planes físicos generados en base a las cantidades de operaciones de I/O que realiza cada algoritmo

Generador de  
planes lógicos

2

## Optimización por Heurísticas

- Cambiar la consulta original por otra equivalente de forma de **minimizar** los resultados intermedios.
- Pueden existir varias alternativas.
- Se basa en aplicar equivalencias de los operadores del álgebra de forma que:
  1. Las selecciones se apliquen **lo antes posible**.
  2. Las ramas **izquierdas** de los join sean **más chicas** que las derechas.

## Reglas de equivalencia de expresiones

- $\sigma_{p_1 \wedge p_2}(R) = \sigma_{p_1}(\sigma_{p_2}(R))$  Cascada de selecciones
- $\sigma_{p_1}(\sigma_{p_2}(R)) = \sigma_{p_2}(\sigma_{p_1}(R))$  Conmutativa de la selección
- $\pi_{a_n}(\pi_{a_k \dots a_n}(R)) = \pi_{a_n}(R)$
- $\pi_{a_1 \dots a_n}(\sigma_p(R)) = \sigma_p(\pi_{a_1 \dots a_n}(R))$  [si p sólo contiene  $a_1 \dots a_n$ ]
- $\sigma_p(R \times E) = R \mid X \mid_p E$  Equivalencia join – producto y selección
- $R \mid X \mid_p E = E \mid X \mid_p R$  Conmutativa del join
- $(R \mid X \mid_p E) \mid X \mid_p S = R \mid X \mid_p (E \mid X \mid_p S)$  Asociativa del join
- $\sigma_{\theta_1}(R \times E) = (\sigma_{\theta_1}(R)) \times E$  [si  $\theta_1$  sólo contiene atributos de R]
- $\pi_{a_n \cup a_k}(R \times E) = \pi_{a_n}(R) \times \pi_{a_k}(E)$  [si  $a_n$  es de R y  $a_k$  es de E]

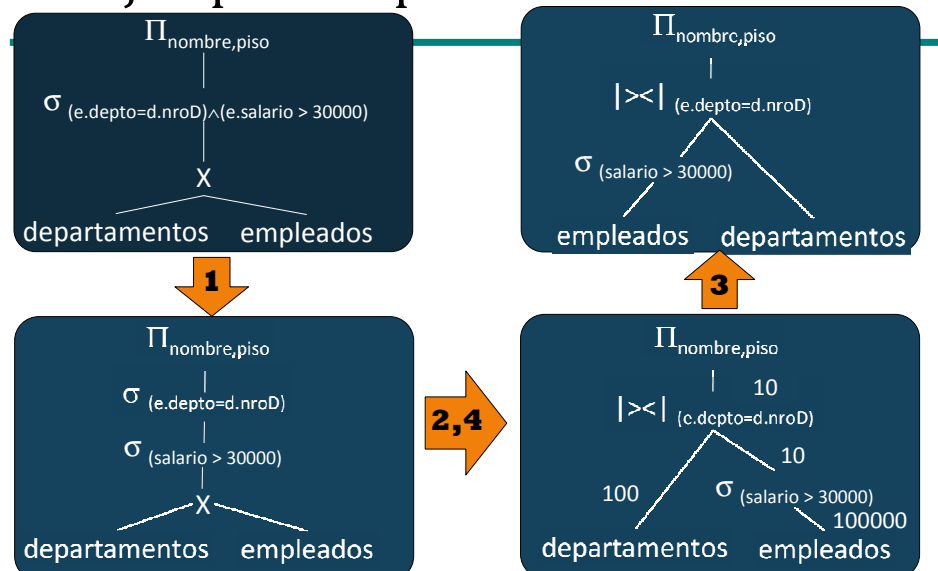
## Reglas de equivalencia de expresiones (2)

- $R \cup E = E \cup R$  Conmutativa de la unión
- $R \cap E = E \cap R$  Conmutativa de la intersección
- $R \cup (E \cup D) = (R \cup E) \cup D$  Asociativa de la unión
- $R \cap (E \cap D) = (R \cap E) \cap D$  Asociativa de la intersección
- $\sigma_c(R \cup E) = \sigma_c(R) \cup \sigma_c(E)$
- $\sigma_c(R \cap E) = \sigma_c(R) \cap \sigma_c(E)$
- $\sigma_c(R - E) = \sigma_c(R) - \sigma_c(E)$
- $\pi_{a_n}(R \cup E) = \pi_{a_n}(R) \cup \pi_{a_n}(E)$  Distributivas

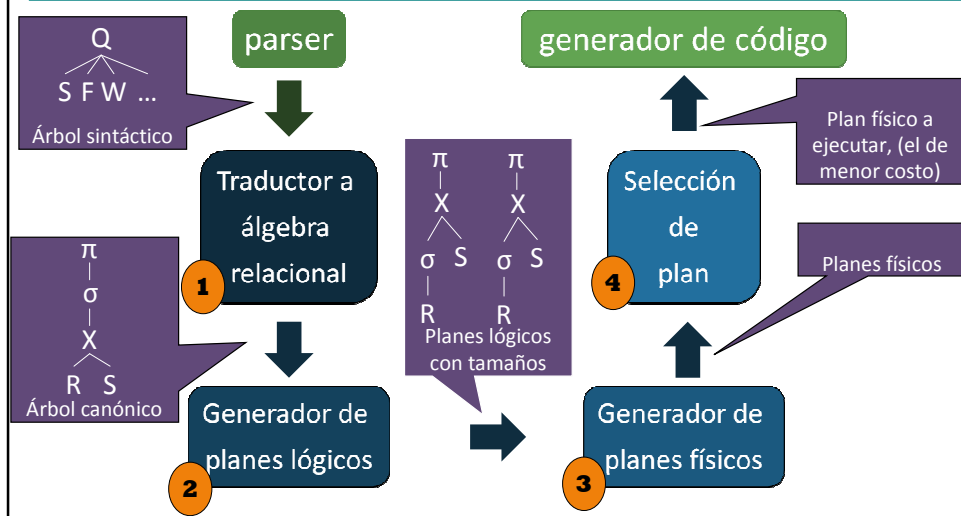
# Heurísticas

- Reglas para reducir los tamaños intermedios.
  - Cambiar las selecciones conjuntivas por una “**cascada**” de selecciones simples.
  - Mover las **selecciones lo más abajo** que se pueda en el árbol.
  - Poner a la **izquierda** de los productos las hojas que generen menos tuplas.
  - Cambiar** secuencias de selecciones y productos por join's.
  - Mover las **proyecciones lo más abajo** posible en el árbol, agregando las proyecciones que sean necesarias.

## Ejemplo de optimización heurística



## Proceso de Optimización



## Optimización por Costos

- **Plan Físico**
  - Le asocia a cada operador del álgebra que aparece en un plan lógico, una implementación.
  - Como se pueden considerar diferentes implementaciones para cada operador, entonces un mismo plan lógico puede originar diferentes planes físicos.
  - Es necesario estimar el costo (cantidad de operaciones de I/O) de los diferentes planes que se generen y elegir el de costo mínimo.
- Para evaluar el costo, es necesario considerar ciertos parámetros que tienen influencia en el cálculo de la cantidad de operaciones de I/O.



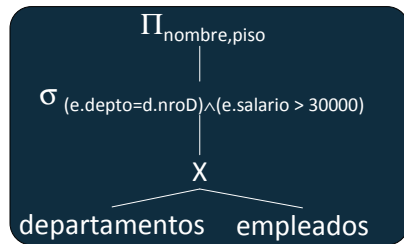
## Parámetros para la Estimación de Costos y Tamaños

Nombre	Definición	Notación	Fórmula
Tamaño o Cardinalidad de una Relación T	Cantidad de Registros	$r, n_T$	-
Tamaño del Registro de una Relación T	Cantidad de Bytes de un registro	$R, R_T$	-
Cantidad de Bloques para una Relación T	Cantidad de bloques necesarios para almacenar los registros de una relación	$b, b_T$	-
Factor de Bloqueo para una Relación o índice T	Cantidad de registros que entran en un bloque	$bfr, bfr_T$	$\lfloor \text{bytes del bloque} / \text{bytes del registro} \rfloor$
Cantidad de Niveles de un índice	Cantidad de niveles de un índice (la fórmula depende del tipo)	$x, x_T$	$\log_k(n_T) + 1$ (para un B+ con k punteros por nodo sobre clave)
Cantidad de valores distintos del atributo A en la tabla T	Cantidad de valores distintos que tiene un atributo en una tabla	$d, V(A, T)$	$n_T$ ( para un atributo clave )

## Parámetros para la Estimación de Costos y Tamaños (2)

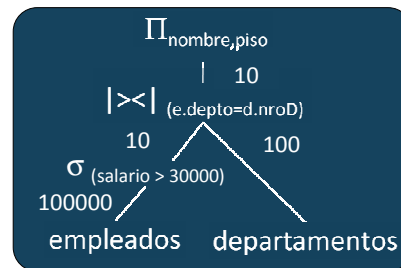
Nombre	Definición	Notación	Fórmula
Selectividad de una selección	Fracción que indica cuántos registros se deben seleccionar con respecto a la tabla original.	$sl, sl(\sigma_c(T))$	$1/V(A, T)$ (si la condición es una igualdad por el atributo A y se asume distribución uniforme)
Selectividad de un join	Fracción que indica cuántos registros se deben seleccionar con respecto al producto cartesiano original.	$js, js(R X _cS)$	$1/\text{Min}(V(A, R), V(A, S))$ (si es el join natural de R y S por el atributo A)
Cardinalidad de una selección	Cantidad de registros en el resultado de una selección	$s, T(\sigma_c(R))$	$n_R * sl(\sigma_c(R))$
Cardinalidad de un join	Cantidad de registros en el resultado de un join	$j, T(R X _cS)$	$n_R * n_S * js(R X _cS)$
-	Información del tipo de cada índice (si es primario, o arbol B+, etc.)	-	-

## Ejemplo de estimación de tamaños



Parámetro	Valor
Tamaño de EMPLEADOS (tuplas)	100.000
Tamaño de DEPARTAMENTOS (tuplas)	100
Selectividad de $\sigma_{(\text{salario} > 3000)}$	1/10.000

- Además por las tablas sabemos:
  - que c/empleador está relacionado con un sólo departamento.
  - que nroD es clave en departamentos



## Implementaciones de los operadores

- A cada operador de un plan lógico se le asigna una implementación.
- Luego hay que estimar el costo de todo el plan basándose en los costos de cada algoritmo.
- Es importante la estrategia de implementación:
  - Pipelined: hay operadores que se ejecutan simultáneamente y pueden pasarse los resultados a medida que se generan. No necesita grabar los resultados intermedios.
  - No Pipelined: los operadores se ejecutan secuencialmente y es necesario grabar resultados intermedios.
- Asumimos:
  - Selección y Join (No Pipelined): Se debe considerar el costo de grabar el resultado intermedio.
  - Proyección (Pipelined): no hay costo intermedio

## Implementación de los operadores: Estimaciones de Costos

- En el costo consideramos sólo los acceso a disco:
  - de lectura.
  - de grabación.
- Siempre se realizan las operaciones de a bloque (página) que pueden contener varios registros de índice o datos.
- Los costos de lectura dependen de la organización de los datos
- El costo de grabación **siempre** es el costo de grabar todo el resultado (R):
  - $\lceil n_R/bf_R \rceil$  donde  $bf_R = (\text{cant. bytes bloque}) / (\text{cant. bytes tupla})$
- Al ver los algoritmos consideramos las lecturas, pero en el costo debemos agregar la grabación.

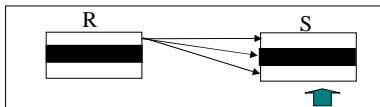
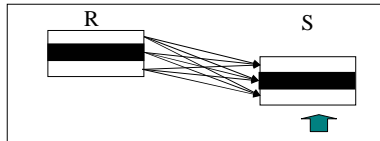
## Implementación de la selección ( $\sigma_c(R)$ )

- Búsqueda lineal.
  - **Restricciones de uso:** ninguna.
  - **Descripción:** leer cada registro y si cumple la condición se pone en el resultado.
  - **Costos de lectura:**
    - Peor caso:  $b_R$  (cantidad de bloques de la relación R)
    - Promedio:  $b_R/2$
- Búsqueda Binaria.
  - **Restricciones de uso:** registros ordenados.
  - **Descripción:** leer el bloque del medio y en función de la condición leer el del medio de la primera o segunda mitad y así hasta encontrarlo o no tener más bloques para leer.
  - **Costos de lectura:**  $\log_2 b_R + \lceil s/bf_R \rceil - 1$

## Implementación de la selección ( $\sigma_c(R)$ ) con Índices

- **Primario o Cluster:**
  - **Restricciones de uso:** registros ordenados.
  - **Costos de lectura:**
    - $x + \lceil s/bf_R \rceil$  ( $x$  es la cantidad de niveles del índice)
    - Si el índice es primario  $x+1$  (sólo 1 bloque tiene el valor buscado)
- **Hash:**
  - **Restricciones de uso:** sólo para condiciones por igualdad.
  - **Costos de lectura:** 1 o 2 dependiendo del tipo de hash
- **Secundario con B+:**
  - **Restricciones de uso:** ninguna
  - **Costos de lectura:**  $x + s$  (peor caso, asumiendo que cada registro está en un bloque distinto)

## Implementación de los Operadores: Join ( $R \bowtie_{A=B} S$ )

- **Loop Anidado por Registros**

  - **Restricciones de uso:** ninguna
  - **Descripción:** Para cada registro de R acceder a todos los bloques de S y combinar ese registro de R con todos los de S.
  - **Costos de lectura:**  $b_R + n_R * b_S$
- **Loop Anidado por Bloques**

  - **Restricciones de uso:** ninguna
  - **Descripción:** para cada bloque de R combinar todos los registros de ese bloque con los de S y luego pasar al siguiente bloque de S
  - **Costos de lectura:**  $b_R + \lceil b_R/(M-2) \rceil * b_S$  Sea M la cantidad de buffers

## Implementación del Join ( $R \bowtie_{A=B} S$ ) (2)

- Sort-Merge Join
  - **Restricciones de uso:** las dos tablas deben tener los registros ordenados. Si no es así hay que agregar los costos de ordenación.
  - **Descripción:** recorrer R y S en paralelo combinando los registros.
  - **Costo de lectura:**  $b_R + b_S$
  - **Costo de ordenación:**  $2 * b * (1 + \log_2 b)$
- Index Join (Single Loop)
  - **Restricciones de uso:** existencia de un índice para S
  - **Descripción:** recorrer R y acceder por el índice a S.
  - **Costo de lectura:**  $b_R + (n_R * Z)$  donde Z depende del tipo de índice.
    - secundario:  $Z = x + s_S$
    - cluster:  $Z = x + \lceil s_S / bf_S \rceil$
    - primario:  $Z = x + 1$
    - hash = h

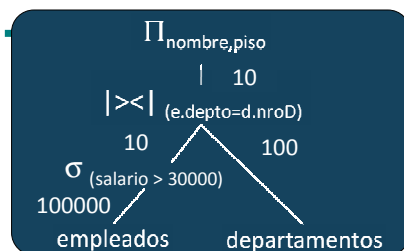
## Implementaciones de los Operadores.

	Algoritmo	Costo	Condición	Organización
$\sigma_c(R)$	Búsqueda Lineal	$b_R$ (peor caso) $b_R/2$ (prom)	Todas	--
	Búsqueda Binaria	$\log_2 b_R + \lceil s/bf_R \rceil - 1$	Todas	Ordenado
	Índice Primario	$x + 1$	Igualdad	Ordenado
	Hash	1 o 2	Igualdad	--
	Índice Primario	$x + (b/2)$ (prom)	de orden	Ordenado
	Índice Cluster	$x + \lceil s/bf_R \rceil$	Todas	Ordenado
	Índice secundario B+	$x + s$ (peor caso)	Todas	--
	Grabación Intermedia	$s/bf_R$	Todas	--

## Implementaciones de los Operadores (2)

	Algoritmo	Costo	Condición	Organización
$R  > < _C S$	Loop Anidado (registros)	$b_R + (n_R * b_S)$	Todas	--
	Loop Anidado (bloque)	$b_R + \lceil b_R / (M-2) \rceil * b_S$	Todas	--
	Sort Merge	$b_R + b_S + \text{costo ordenación}$	Todas	Índice
	Index join	$b_R + n_R * Z$	Todas	Índice

## Ejemplo de optimización por costos



Parámetro	Valor
Tamaño de EMPLEADOS (tuplas)	100.000
Tamaño de DEPARTAMENTOS	100
Selectividad de $\sigma_{(\text{salario} > 30000)}$	1/10.000
$Bf_{\text{empleados}}, bf_{\text{departamentos}}$	10
$bf_{\text{empleados} \bowtie \text{departamentos}}$	5
Índices sobre EMPLEADOS	B+ en salario, x=5
Todas las tablas tienen índice primario con x=1	
Asumimos que hay 3 buffers (M=3)	

	implementación	Costo leer	Costo grabar
$\sigma_{(\text{salario} > 30000)}$ Empleados	Búsqueda lineal	10.000	1
	Búsqueda binaria	No es posible	
	Índice secundario	15	1
$ > < $	Loop anidado reg.	101	2
	Loop anidado bloq.	11	2
	Sort Merge	11 o 12	2
Costo total mínimo		15+11 = 26	2+1=3

## Relaciones

### **Temas ( id\_c, titulo, autor, idioma, creado)**

Tamaño de la tupla es 100 bytes. Una página puede contener 50 tuplas.

Tiene 10.000 tuplas. Existen 500 autores y 3 idiomas.

Hay un índice sobre los títulos y otro sobre los autores.

Expresa el título, autor, idioma y año de creación de un cierto tema musical.

### **Interprete ( id\_i, nombre, pais)**

Tamaño de la tupla es 40 bytes. Una página puede contener 100 tuplas.

Tiene 1 000 tuplas, y hay 20 países diferentes.

Hay un índice sobre nombre y otro sobre país.

Expresa el nombre y el país de origen de intérpretes de temas musicales.

### **Canta ( id\_c, id\_i, año)**

Tamaño de la tupla es 10 bytes. Una página puede contener 140 tuplas.

Tiene 25.000 tuplas.

Expresa el año de la primera vez que un intérprete cantó un cierto tema.

```
SELECT      nombre, año
FROM  Temas T, Canta C,Interprete I,
WHERE      C.id_c = T.id_c and C.id_c = I.id_c and
           I.pais = "Mexico" and T.idioma = "Ingles";
```