

Capítulo 4

Procesamiento de consultas

Evaluación de una consulta del álgebra de tablas



Visión general

- Para procesar una consulta, una consulta de SQL se puede **traducir** a una expresión del álgebra de tablas.
- Luego se puede **evaluar** la consulta del álgebra de tablas.
- La evaluación de una consulta del álgebra de tablas va a estar en términos de operadores físicos.

Visión general

- Una expresión de álgebra de tablas puede tener varias expresiones equivalentes.
- **Ejemplo:** $\sigma_{salary < 75000}(\Pi_{salary}(instructor))$ es equivalente a $\Pi_{salary}(\sigma_{salary < 75000}(instructor))$
- **¿Qué consecuencias tiene esto con respecto al procesamiento de consultas?**
- Que para procesar una consulta en álgebra de tablas puede convenir evaluar una consulta equivalente
 - con determinados operadores físicos.
 - O sea, que se puede **planear** cómo se va a procesar una consulta.

Visión general

- Dada una consulta C del álgebra de tablas un **plan de evaluación** consiste de una expresión E equivalente a C y operadores físicos para los operadores lógicos de E .
- La **máquina de ejecución de consultas** toma el plan de evaluación de consulta, ejecuta ese plan y retorna las respuestas de la consulta.

Visión general

- Los diferentes planes de evaluación para una consulta dada pueden tener diferentes costos.
- El SGBD debería construir un plan de evaluación que minimiza el costo de evaluación de consultas.
 - Una vez que un plan de evaluación es elegido, la consulta es evaluada con este plan.
- **Optimización de consultas:** entre todos aquellos planes de evaluación equivalentes encontrar aquel con menor costo.
 - El costo es estimado usando información estadística de la base de datos.
 - **P.ej:** número de registros en cada tabla, tamaño de los registros, etc.
- Con el fin de optimizar una consulta un optimizador de consultas debe conocer el **costo de cada operación**.

Asuntos de este capítulo

- En este capítulo estudiamos primero cómo ejecutar un plan de evaluación; y luego estudiaremos un poco sobre cómo elegir un plan de evaluación.
- Primero, dada una expresión del álgebra de tablas, hay que **elegir bien los operadores físicos**.
- Luego se estimará el **costo de procesamiento** de la consulta.

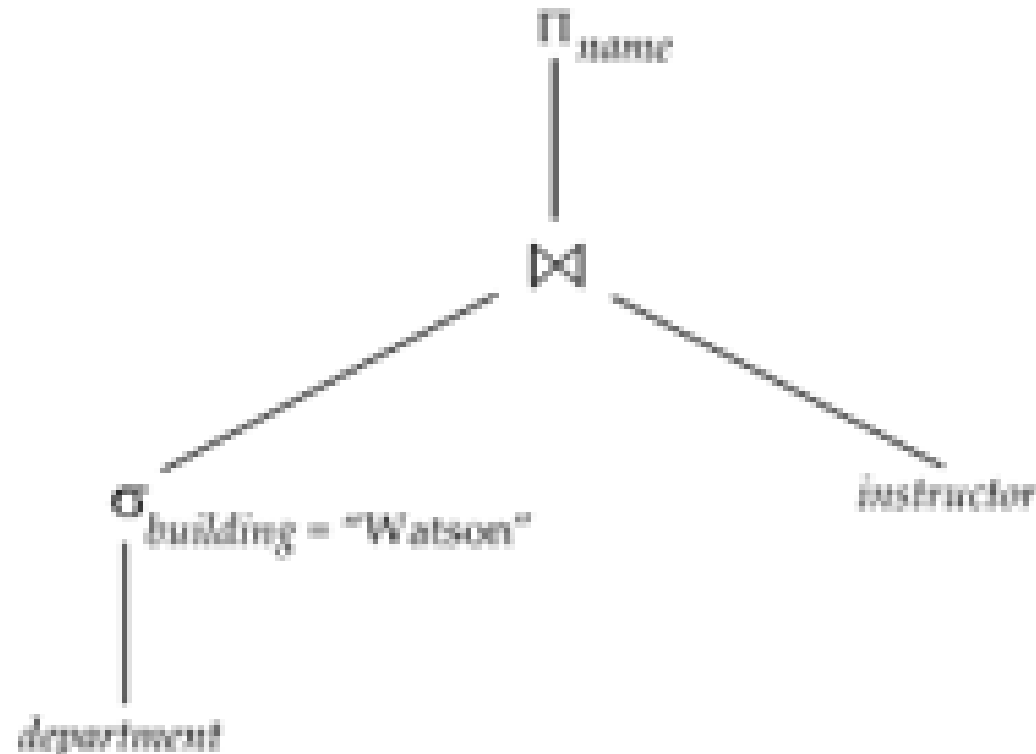
Árbol binario de ejecución

- El **árbol binario de ejecución** de una consulta es el árbol binario de una expresión de consulta.
 - Los nodos hoja son tablas de la base de datos
 - Los nodos internos son operadores del álgebra de tablas
- Para una consulta del álgebra de tablas puedo definir varias expresiones equivalentes y cada una tiene su árbol binario de ejecución.
- La evaluación de un árbol binario de ejecución va a estar en términos de operadores físicos.

Árbol binario de ejecución

- El árbol binario de ejecución asociado a la consulta

$\Pi_{\text{name}}(((\sigma_{\text{building}=\text{'watson'}}(\text{department})) \bowtie \text{instructor}))$



Evaluación del árbol binario de ejecución

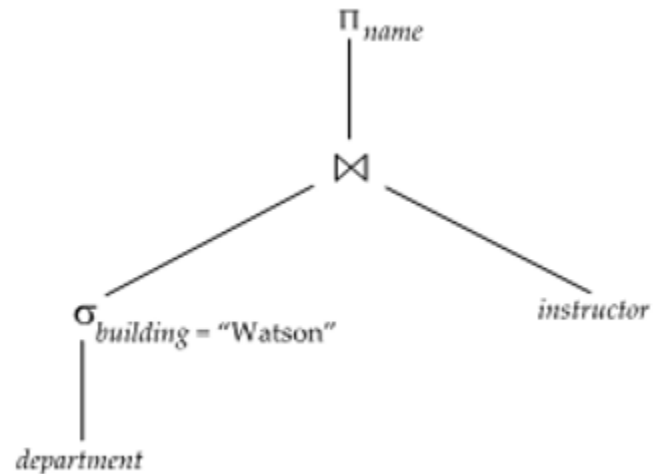
- El resultado de evaluar un operador de un nodo interno del árbol binario de ejecución que no es la raíz del árbol se llama **resultado intermedio**.
- Para la evaluación del árbol binario de ejecución hay dos enfoques:
 - **Materialización**: los resultados intermedios se guardan en disco en tablas temporales a las cuales tiene acceso el sistema gestor de BD (SGBD).
 - No hay índices sobre este tipo de tablas.
 - **Encausamiento**: a medida que se van generando los resultados intermedios se van pasando al siguiente operador.
 - Los resultados intermedios no se guardan en disco.

Evaluación del árbol binario de ejecución

- **Comparando ambos enfoques:** Materialización usa mucho menos memoria y bastante más espacio en disco y encausamiento usa mucho menos espacio en disco, y bastante más memoria.
- Vamos a estudiar solamente materialización.

Materialización

- **Ejemplo:** Supongamos que tenemos el árbol binario de ejecución y usamos materialización:



1. Computamos y almacenamos en disco primero $\sigma_{building = 'Watson'}$ (*department*)
2. Computamos y almacenamos en disco la reunión natural del resultado intermedio anterior con *instructor*.
3. Computamos la proyección según *nombre*.

Materialización

- Con **materialización**:
 - Cambiar nodos lógicos por físicos en el árbol binario de ejecución.
 - Si hay más de un operador físico posible, elegir el menos costoso.
 - Evaluamos un operador físico por vez comenzando en el nivel más bajo.
 - Usamos resultados intermedios en tablas temporales para evaluar los operadores físicos del siguiente nivel.

Materialización

- Para poder **optimizar una consulta**, vimos que es necesario elegir un buen **plan de ejecución**.
 - Este sería un árbol binario de ejecución, junto con los operadores físicos que se usan para cada operador de la consulta.
- Para poder **elegir entre varios planes de ejecución**, es necesario tener los costos de cada uno.
 - Así elegimos el menos costoso.
- Entonces hay que poder **evaluar el costo** de un plan de ejecución usando materialización.
- Dado un árbol binario de ejecución, estudiamos:
 - cómo se lo evalúa usando materialización y
 - cómo calcular el costo de evaluarlo usando materialización.
 - Para ello veremos un procedimiento que integra las dos tareas.

Cálculo de estimaciones

- **Para calcular el costo de evaluar un árbol binario de ejecución usando materialización vamos a necesitar comprender:**
 - Cómo estimar el tamaño de un registro de una tabla en bytes.
 - Cómo estimar el tamaño del resultado de un operador físico en cantidad de tuplas.
 - Cómo estimar el tamaño en bloques del resultado de un operador físico, si tenemos una estimación de la cantidad de tuplas del mismo.
 - Cómo estimar la cantidad de claves de búsqueda en un nodo de índice de árbol B+.

Cálculo de estimaciones

- **Estimación del tamaño de un registro de una tabla en bytes:**
 - Asumimos que tenemos el esquema de la tabla.
 - Primero estimamos el **costo de cada atributo** de la tabla en bytes.
 - Para una organización de archivo secuencial, en cada fila de registro hay que contar también el **tamaño en bytes de un puntero** a disco.
 - El tamaño de un puntero a disco en un sistema operativo con **direccionamiento de 64 bits** suele tener **8 bytes**.
 - Para calcular el tamaño de una fila en bytes de un archivo secuencial:
 - se suman los tamaños de los atributos (en bytes) de una tupla, más el tamaño del puntero a disco (en bytes).

Cálculo de estimaciones

- **Estimación del tamaño del resultado de un operador físico en cantidad de tuplas.**
 - La proyección tiene la misma cantidad de tuplas que la tabla a la que se aplica.
 - El producto cartesiano $r \times s$ tiene tamaño: $|r \times s| = |r| * |s|$ registros.
 - La ordenación de una tabla preserva la cantidad de tuplas en el resultado.
 - La concatenación de tablas $r ++ s$ tiene tamaño: $|r ++ s| = |r| + |s|$ registros.
 - Se proveen cotas superiores para las estimaciones de las cantidades de tuplas de los siguientes operadores (a menos que se tenga información adicional para su cálculo exacto):
 - Para $r \cap s$: $\min(|r|, |s|)$
 - Para $r - s$: $|r|$
 - Para $V(r)$: $|r|$
 - A continuación estudiaremos como estimar la cantidad de tuplas para selección y reunión selectiva.

Tamaño de resultado de operador físico de selección

- **Problema:** Hay que estimar el tamaño de los resultados en cantidad de bloques a escribir a disco para los operadores físicos de selección.
- **Solución:**
 1. Usar una función de probabilidad llamada **factor de selectividad** para calcular la cantidad de registros del resultado del operador físico de selección.
 2. A partir de cantidad de registros, calcular la cantidad de bloques del resultado intermedio.

Tamaño de resultado de operador físico de selección

- Si el operador de selección o reunión usa predicado P , y el input del operador es r , denotamos al factor de selectividad mediante: $fs(P, r)$.
- **Para selección:**
 - Cantidad de registros resultado intermedio = $|r| * fs(P, r)$
Donde $|r|$ es la cantidad de registros de r .
- Una forma de calcular el factor de selectividad es asumir **uniformidad** e **independencia**:
 - **Uniformidad**: todos los valores de un atributo son igualmente probables
 - **Independencia**: condiciones sobre diferentes atributos son independientes

Tamaño de resultado de operador físico de selección

- **Ejemplo:** atributo sexo de persona: todos los valores de sexo son igualmente probables (la proporción de hombres es igual a la proporción de mujeres).
 - $fs(\text{sexo} = 'F', \text{persona}) = 1/2$
- **Ejemplo:** Atributos sexo y edad en persona, con predicado:
 - $\text{edad} == 40 \ \&\& \ \text{sexo} == 'F'$.
 - La edad es independiente del sexo.
 - $fs(\text{edad} == 40 \ \&\& \ \text{sexo} == 'F', \text{persona}) =$
 $fs(\text{edad} == 40, \text{persona}) * fs(\text{sexo} == 'F', \text{persona})$
- Pero no es obligatorio usar uniformidad o independencia para calcular el factor de selectividad.

Factor de selectividad para selección

- Definimos $fs(P, r)$ para distintos tipos de propiedades P y tabla r .
- Desde ahora r tabla, A, A' atributos de r , c y c' constantes.
- **Regla 1:** Asumiendo uniformidad:
 - $fs(A == c, r) = 1/V(A, r)$, donde $V(A, r)$ número de distintos valores que aparecen en r para A .
- **Regla 2:** Asumiendo uniformidad, A con valor numérico (usando valores consecutivos):
 - $fs(A \geq c, r) = (\max(A, r) - c) / (\max(A, r) - \min(A, r))$
- **Regla 3:** Asumiendo uniformidad, A con valor numérico:
 - $fs(A < c, r) = (c - \min(A)) / (\max(A) - \min(A) + 1)$

Factor de selectividad para selección

- **Regla 4:** asumiendo uniformidad, A con valor numérico:
 - $fs(c \leq A < c', r) = (c' - c) / (\max(A, r) - \min(A, r))$
- **Regla 5:** asumiendo independencia:
 - $fs(P_1 \wedge P_2 \wedge \dots \wedge P_n, r) = fs(P_1, r) fs(P_2, r) \dots fs(P_n, r)$
- **Regla 6:** usando propiedad de probabilidades:
 - $fs(\neg P, r) = 1 - fs(P, r)$
- **Regla 7:** asumiendo independencia
 - $fs(P \vee Q, r) = fs(\neg (\neg P \wedge \neg Q), r) = 1 - fs(\neg P \wedge \neg Q, r) = 1 - (fs(\neg P, r) * fs(\neg Q, r))$
 $= 1 - ((1 - fs(P, r)) * (1 - fs(Q, r)))$

Tamaño de resultado de operador físico de reunión

- **Problema:** Hay que estimar el tamaño de los resultados en cantidad de bloques a escribir a disco para los operadores físicos de reunión (selectiva y natural).
- **Solución:**
 1. Usar una función de probabilidad llamada **factor de selectividad** para calcular la cantidad de registros del resultado del operador físico de reunión.
 2. A partir de cantidad de registros, calcular la cantidad de bloques del resultado intermedio.
- La cantidad de tuplas del resultado de la reunión viene dada por
$$|r| * |s| * fs(P, r, s)$$

Factor de selectividad para reunión selectiva

- Si r tabla y A atributo de r , entonces $V(A, r)$ número de distintos valores que aparecen en r para A .
- **Regla:** asumiendo uniformidad:
 - $fs(r.A == s.B, r, s) = 1 / \max(V(A, r), V(B, s))$
 - la **uniformidad** significa: para cada valor de atributo A/B en una tabla hay la misma cantidad de tuplas por el atributo B/A que cazan en la otra tabla.

Factor de selectividad para reunión selectiva

- **Justificación de la regla anterior:**

- Para cada valor común de A/B:
 - Hay $|r| / V(A, r)$ tuplas en r y hay $|s| / V(B, s)$ tuplas en s.
- Entonces se generan $|r| / V(A, r) * |s| / V(B, s)$ coincidencias por valor.
- Bajo el supuesto de intersección total entre $V(A, r)$ y $V(B, s)$, hay $\min(V(A, r), V(B, s))$ valores comunes.
- Entonces la cantidad de tuplas de la reunión es:
$$\min(V(A, r), V(B, s)) * |r| / V(A, r) * |s| / V(B, s)$$
- Si $\min(V(A, r), V(B, s)) = V(A, r)$ entonces:
- $V(A, r) * |r| / V(A, r) * |s| / V(B, s) = |r| * |s| / V(B, s) = |r| * |s| / \max(V(A, r), V(B, s))$
- De que la cantidad de tuplas de la reunión es de $|r| * |s| * fs(P, r, s)$ sale entonces que $fs(r.A == s.B, r, s) = 1 / \max(V(A, r), V(B, s))$.

Factor de selectividad para reunión selectiva

- **Observaciones:**

- Si A clave candidata de r: $fs(r.A == s.B, r, s) = 1 / \max(|r|, V(B, s))$
- Si B clave foránea en s referenciando r: $fs(r.A == s.B, r, s) = 1 / |r|$
- Si toda tupla en r produce tuplas en la reunión selectiva:
 - $fs(r.A == s.B, r, s) = 1 / V(B, s)$
- Si se puede asumir independencia:
 - $fs(r.A == s.B \ \&\& \ r.A' = s.B', r, s) = fs(r.A = s.B, r, s) * fs(r.A' = s.B', r, s)$

Cálculo de número de bloques

- **Problema:** ¿Cómo calcular el número de bloques si tengo en el resultado N registros de tamaño R cada uno y B es el tamaño del bloque?
- **Solución:**
$$\text{NumBloques} = \lceil (N \times R) / B \rceil .$$

Cálculo de estimaciones

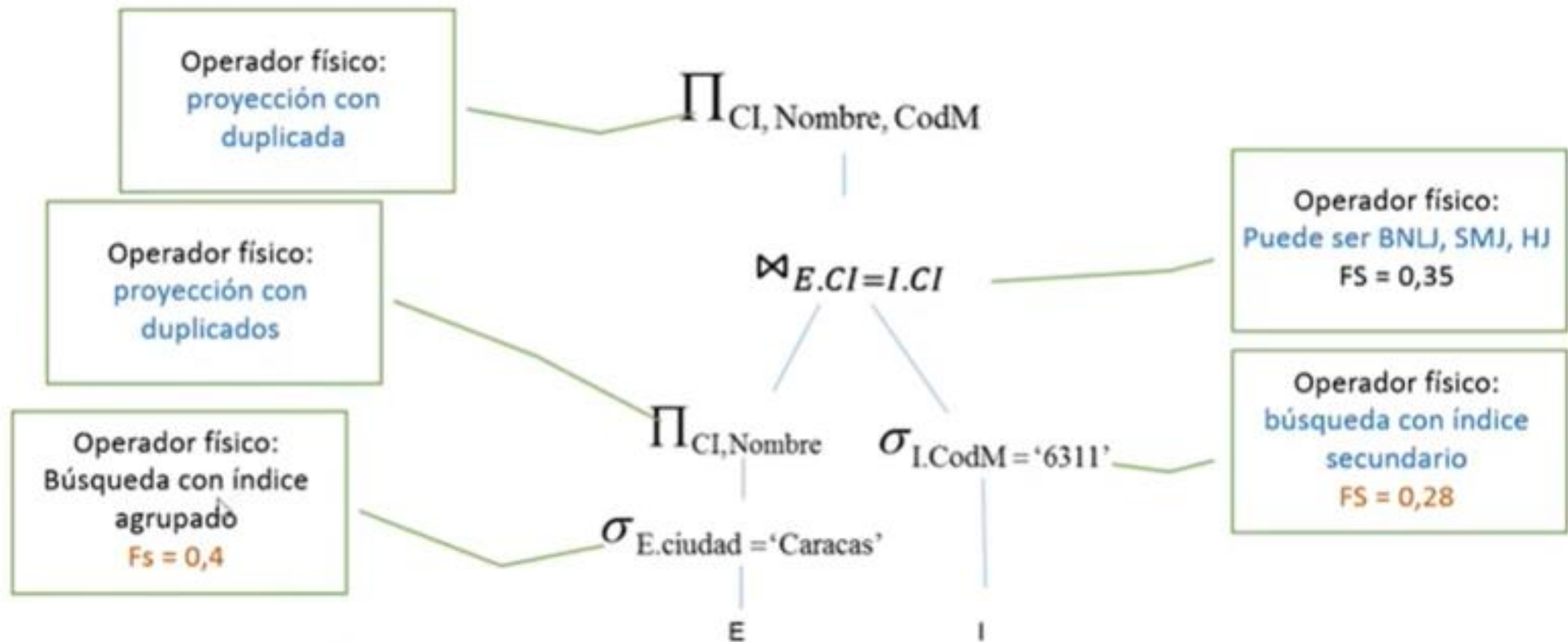
- Estimación de cantidad de claves de búsqueda en un nodo de árbol B+.
- Si en un bloque de C bytes entran n claves de búsqueda, entonces tenemos la siguiente ecuación:
 - $C = n * TCB + (n + 1) * 8$, donde TCB es el tamaño de clave de búsqueda en bytes.
 - Recordar que en un nodo de árbol B+ de n claves de búsqueda hay $n + 1$ punteros de 8 B (para sistemas de 64 bits).
 - Despejando n : $n = (C - 8) / (TCB + 8)$

Materialización

- Ahora vemos cómo procesar y estimar el costo de una consulta con materialización.
- **Fase 1: decidir el plan de ejecución**
 1. Armar el árbol binario de ejecución
 2. Calcular el factor de selectividad para selecciones y reuniones (selectivas y naturales).
 - Explicado en filminas 19 a 21 para selección.
 - Explicado en filmina 23 para reunión natural.
 3. Decidir los operadores físicos más eficientes.
 - Solo se usan índices si la tabla de la BD lo amerita.
 - Ver operadores en archivos de operadores de álgebra de tablas y su implementación.

Materialización

- Ejemplo de fase 1:



Materialización

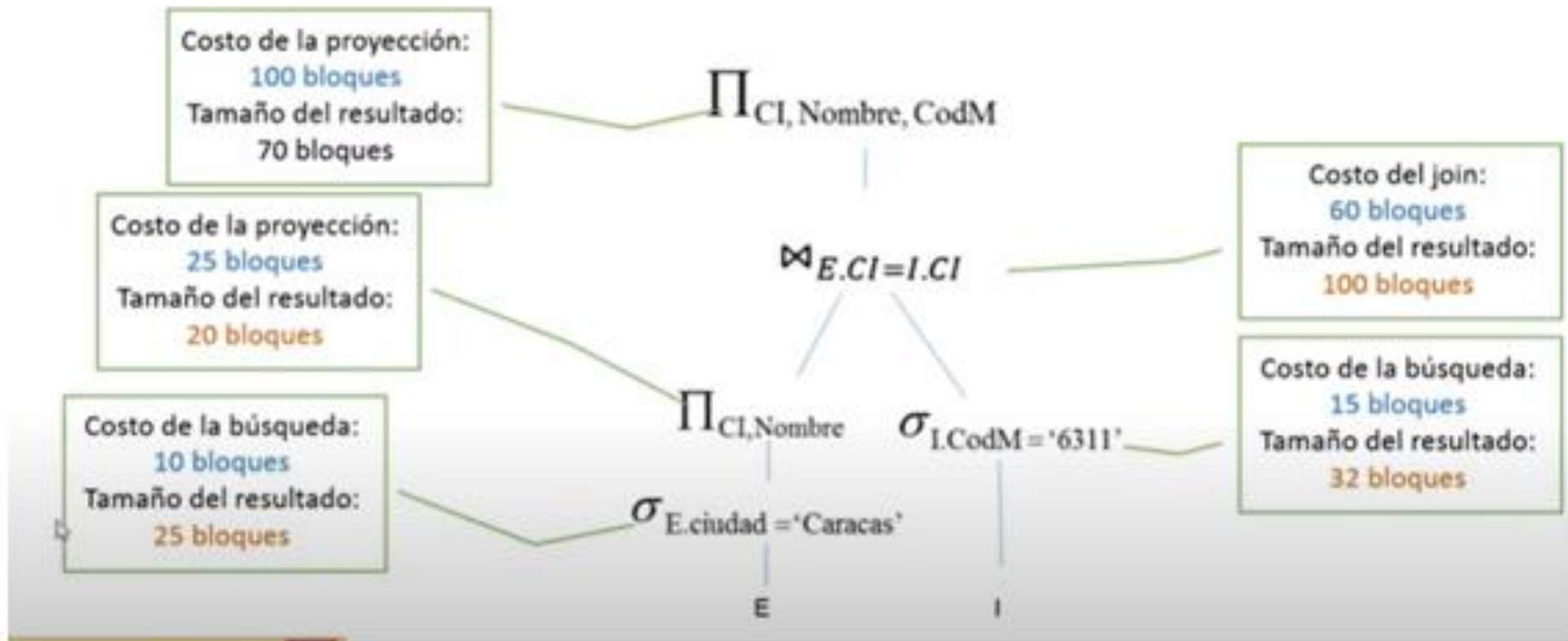
- **Fase 2: estimar el costo de ejecutar el plan de evaluación**

1. Calcular el tamaño en bloques de las tablas de la BD
 - Primero calcular el tamaño en bytes de una tabla (filmina 15), y luego dividir ese tamaño por el tamaño de bloque.
2. Calcular el tamaño de los resultados intermedios primero en cantidad de tuplas y luego en cantidad de bloques:
 - Explicado en filmina 16 para todos los operadores salvo selección y reunión selectiva.
 - Explicado en filminas 18 para selección.
 - Explicado en filminas 22 para reunión selectiva y reunión natural.
3. Calcular el costo de los operadores físicos
 - Explicado en archivos sobre operadores de álgebras de tabla y su implementación.
4. Sumar los costos totales

- Aplicar el método de dos fases.

Materialización

- Ejemplo de Fase 2:



Materialización

- Juntando todo:

- $Costo\ total = \sum costo(operaciones) + \sum costo(materialización)$

- Costo total = $(10+20+15+60+100) + (25+20+32+100)$

- Costo total = 382 (accesos a disco)

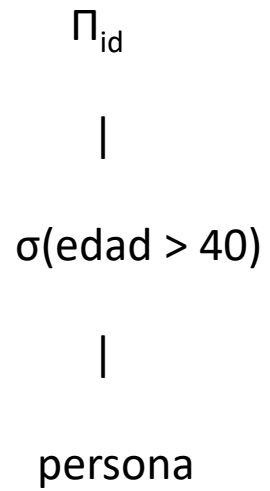
- Esto se multiplica por la velocidad de transferencia y se tiene el tiempo

- A continuación, vamos a hacer un ejemplo detallado de la aplicación de materialización.

Caso de estudio

- Sea la consulta: $\Pi_{id} (\sigma_{edad > 40} (persona))$, donde:
 - Persona ocupa 20 bloques de 10 registros cada uno.
 - Las personas tienen edades de 1 a 100
 - Asumir que entran 50 id por bloque.
 - Entren 20 edades por nodo en el índice primario en edad.

Árbol binario de ejecución



Factor de selectividad

- Usamos la propiedad:
- $fs(A \geq c, r) = (\max(A, r) - c) / (\max(A, r) - \min(A, r))$
- Así obtenemos:

$$fs(\text{edad} > 40, \text{persona})$$

$$= \max(\text{edad}, \text{persona}) - 41 / (\max(\text{edad}, \text{persona}) - \min(\text{edad}, \text{persona}))$$

$$= (100 - 41) / (100 - 1) = 59/99 = 0,595$$

Operadores físicos

- **Proyección:**

- Requiere recorrer todos los registros y realizar una proyección en cada uno.
- Se recorren todos los bloques de la tabla.
- Estimación de costo = b_r transferencias de bloques + 1 acceso a bloque
 - b_r denota el número de bloques conteniendo registros de la tabla r

- **Selección:**

- La tabla está ordenada en A .
- para $\sigma_{A \geq v}(r)$ usar el índice para encontrar el primer registro $\geq v$ y escanar la tabla secuencialmente desde allí.
 - Costo: $h_i + b$ transferencias de bloques, h_i accesos de bloques.
 - b el número de bloques conteniendo registros con $A \geq v$.

Tamaño en bloques de las tablas

- Según el enunciado *persona* tiene 20 bloques.
- Cada bloque de *persona* tiene 10 registros.

Tamaño de los resultados intermedios

- $r = \sigma_{\text{edad} > 40}(\text{persona})$
- $|r| = |\text{persona}| * fs(\text{edad} > 40, \text{persona}) = 200 * 0,595 = 119$
- $B_r = 119 / 10 = 12$ bloques

Costo de los operadores físicos

- **Proyección:** costo es cantidad de transferencia de bloques de resultado intermedio: 12 transferencias de bloques.
- **Selección:** costo $h_i + b$ transferencias de bloques.
 - b es el número de bloques contiendo registros con edad > 40 . Vimos que $b = 12$.
 - h_i altura del árbol B+
 - Tenemos 100 edades y asumimos que entran 20 entradas por nodo del árbol B+. Entonces:
 - $h_i = \lceil \log_{\lceil n/2 \rceil}(K) \rceil = \lceil \log_{\lceil 20/2 \rceil}(100) \rceil = \lceil \log_{10}(100) \rceil = 2$
 - Costo = $12 + 2 = 14$ transferencias de bloques.

Sumar los costos totales

- Costo total = costo de operaciones + costo de materialización
- $= (12 + 14) + 12 = 24 + 14 = 38$ transferencias de bloque.

Tamaño del resultado final

- La proyección se aplica a 119 registros
- La proyección me da 119 id
- Se asumió que entran 50 id por bloque
- Bloques de resultado final = $\text{Techo}(119/50) = 3$