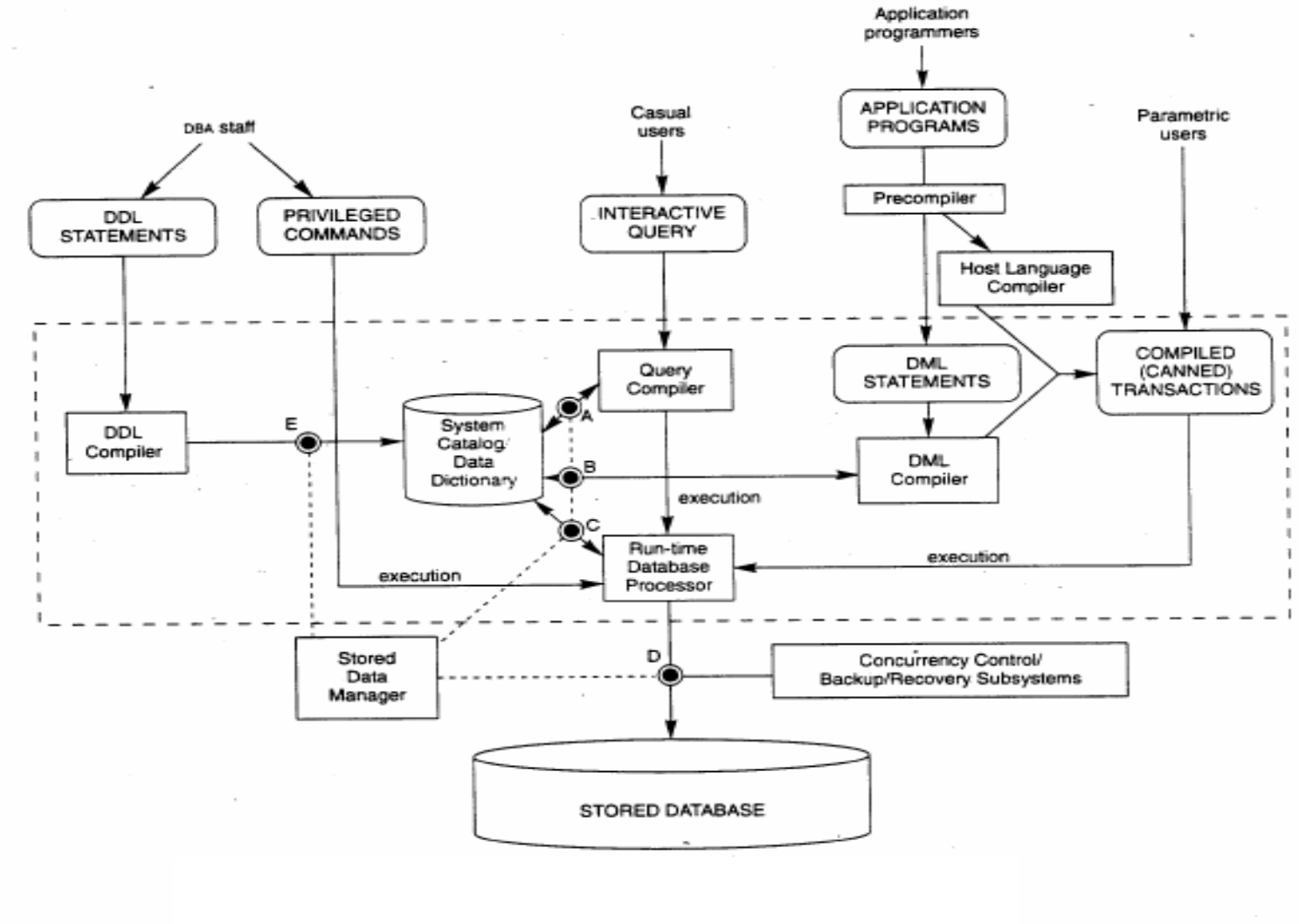
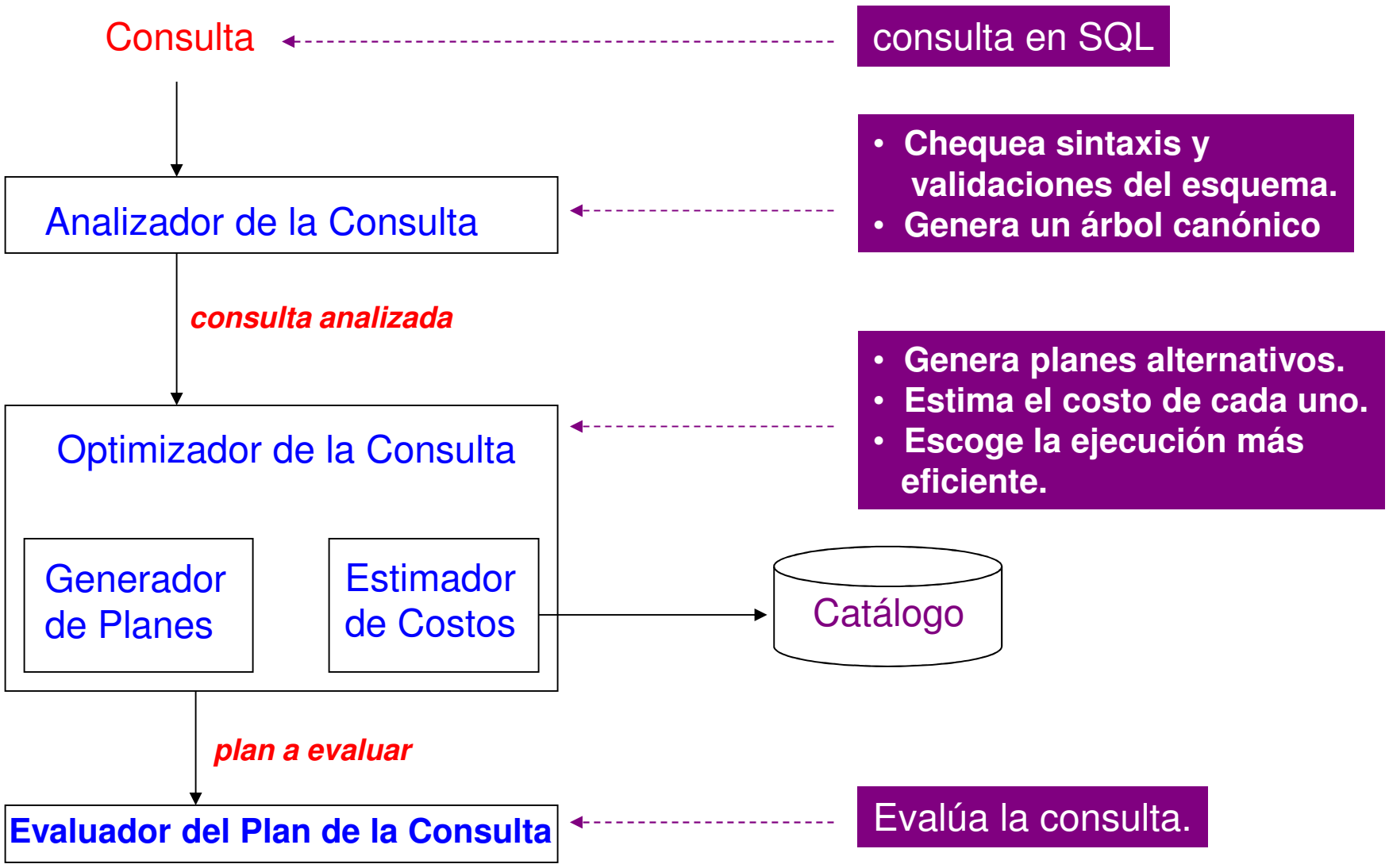
Optimización de Consultas

Optimización



Optimización



Tomado de [2]

Optimizar una expresión del álgebra relacional:

Implica:

1. Enumerar los planes alternativos para la evaluación de la expresión. (Espacio de planes o de soluciones).

2. Estimar el costo de cada plan y escoger el plan con el menor costo.

Para estimar el costo del plan, se debe estimar el costo de cada operador relacional de forma individual, usando información acerca de las propiedades como tamaño de la relación, índices, ordenamiento, etc.

Plan de Evaluación

Consiste de un árbol del álgebra relacional, extendido con anotaciones en cada nodo que indican el método de acceso de cada relación y la implementación de cada operador relacional.

Para la siguiente consulta:

Select E.nombre-e
From Preparador P, Estudiante E
Where P.ci-p = E.ci-e AND E.promedio > 15 AND P.asig=3044

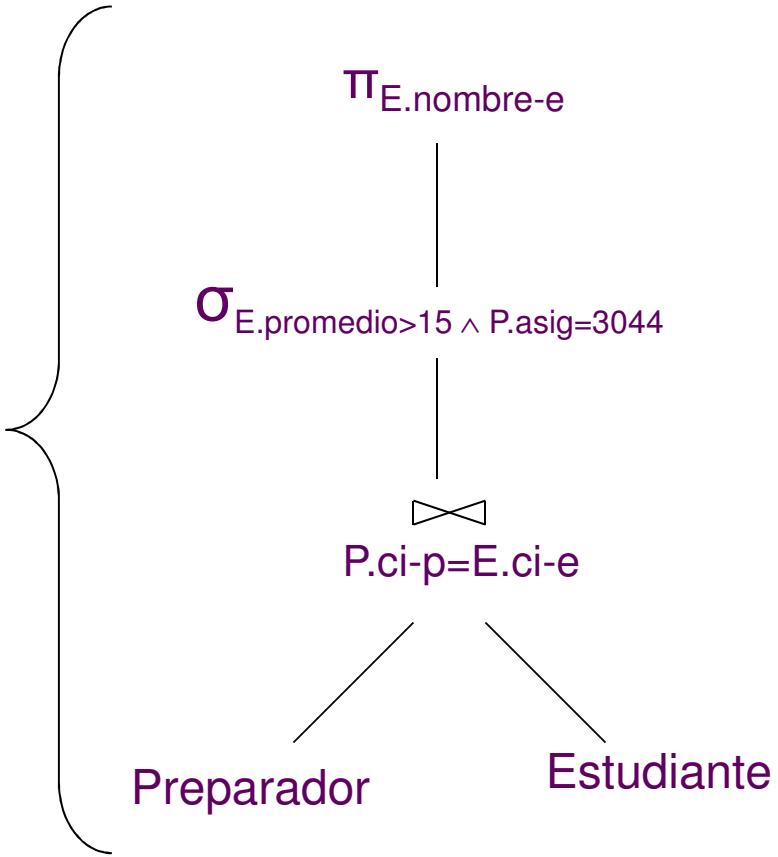
La expresión en álgebra relacional es:

$\pi_{E.nombre\text{-}e}(\sigma_{E.promedio>15 \wedge P.asig=3044}(Preparador \bowtie_{P.ci\text{-}p=E.ci\text{-}e} Estudiante))$

La expresión en álgebra relacional es:

$$\pi_{E.nombre\text{-}e}(\sigma_{E.promedio>15 \wedge P.asig=3044}(Preparador \bowtie_{P.ci\text{-}p=E.ci\text{-}e} Estudiante))$$

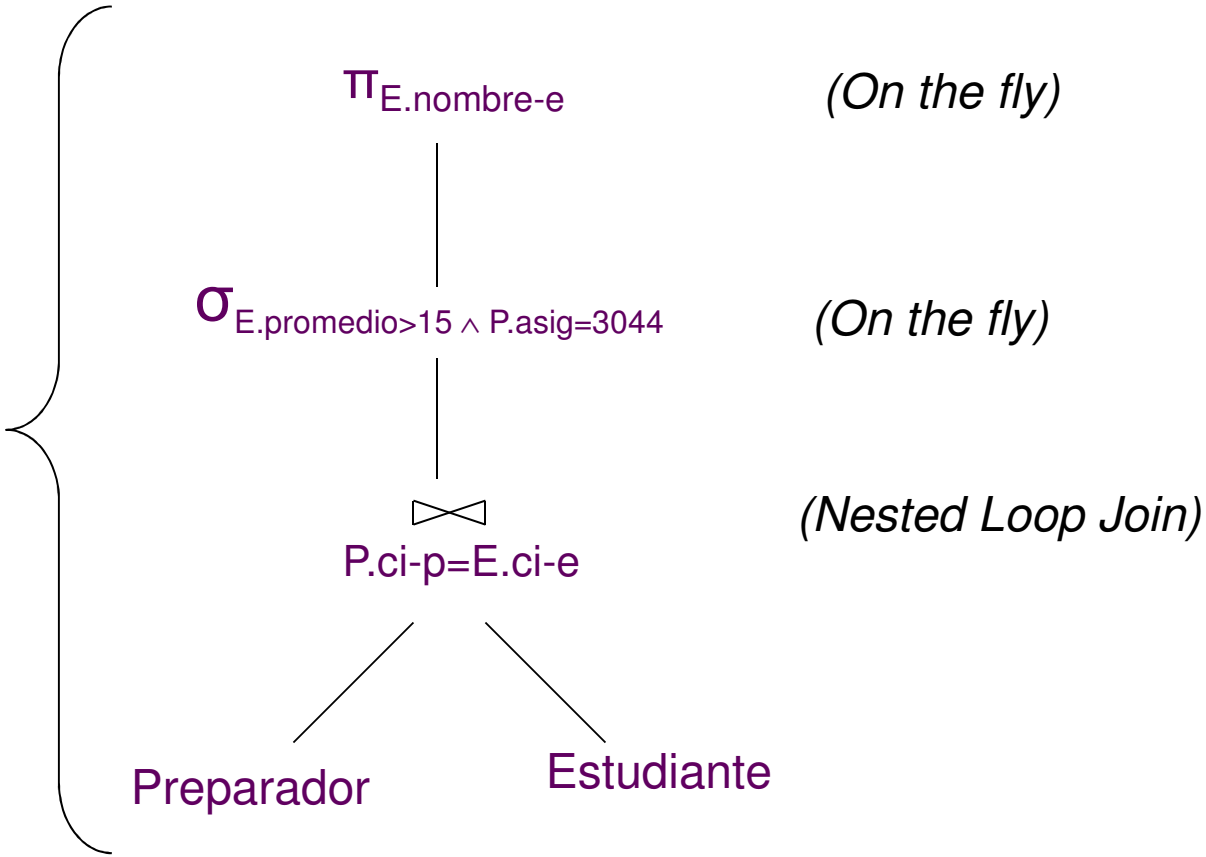
árbol del álgebra relacional



La expresión en álgebra relacional es:

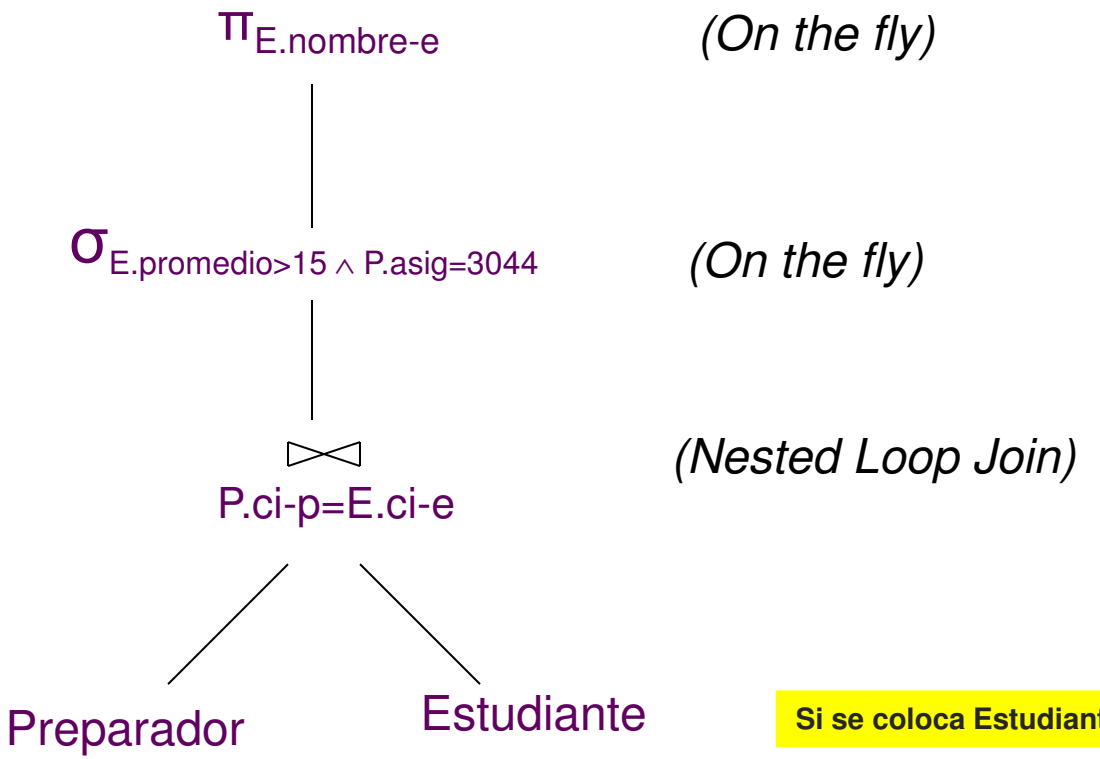
$\pi_{E.nombre\text{-}e}(\sigma_{E.promedio>15 \wedge P.asig=3044}(Preparador \bowtie_{P.ci\text{-}p=E.ci\text{-}e} Estudiante))$

Plan de Evaluación



Catálogo: Preparador tiene 1000 páginas, Estudiante tiene 500 páginas

On the fly: "al vuelo", no usa memoria intermedia.

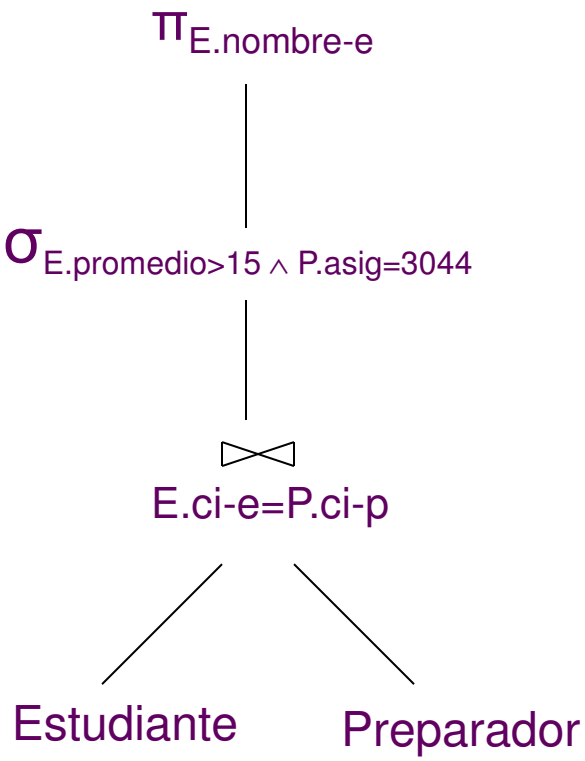


Si se coloca Estudiante como relación externa

Optimización

Catálogo: Preparador tiene 1000 páginas, Estudiante tiene 500 páginas

On the fly: "al vuelo", no usa memoria intermedia.



(On the fly)

(On the fly)

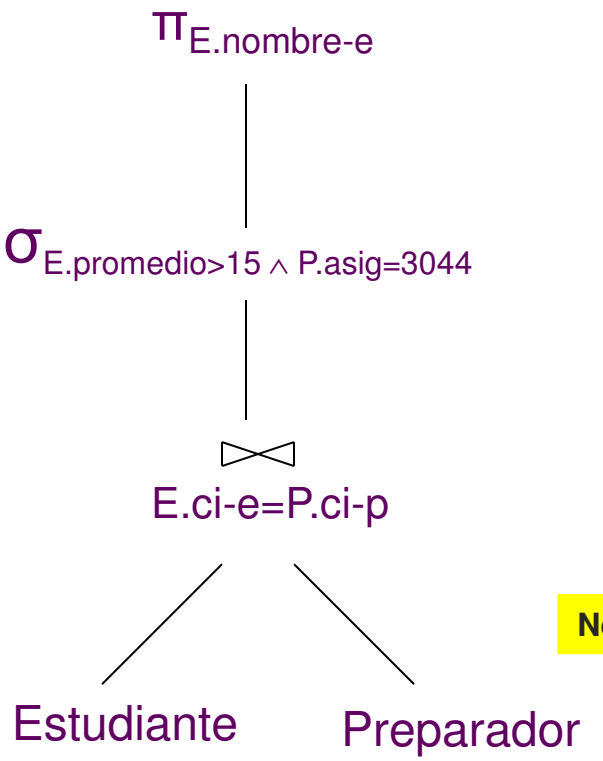
(Nested Loop Join)

500 + 500 * 1000

Catálogo: Preparador tiene 1000 páginas, Estudiante tiene 500 páginas

On the fly: "al vuelo", no usa memoria intermedia.

Total Costo: 500 + 500*1000 = 500.500 I/O's



(On the fly)

(On the fly)

(Nested Loop Join)

500 + 500 * 1000

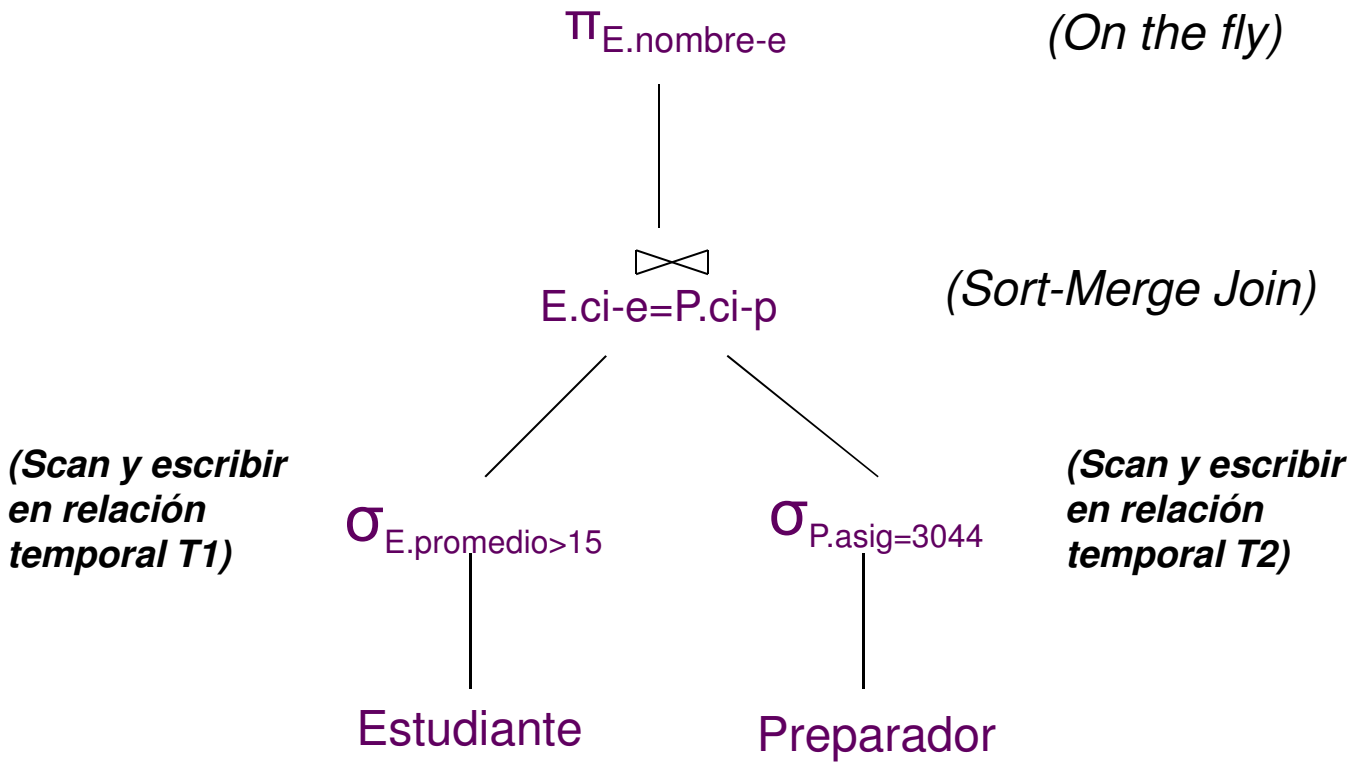
No aprovecha las selecciones para para reducir las relaciones del join

Objetivo:
Buscar planes más eficientes para obtener la misma respuesta

La expresión en álgebra relacional es:

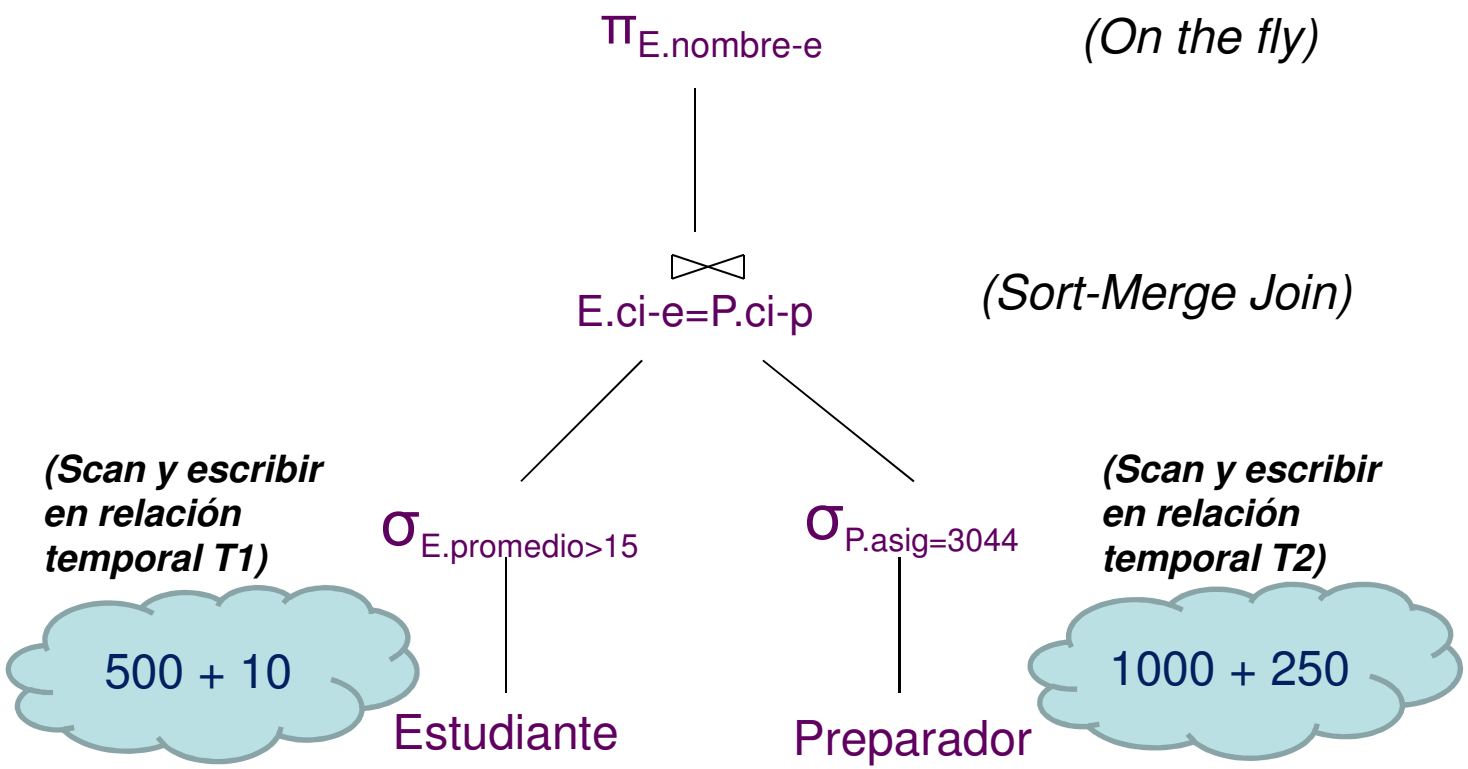
$\pi_{E.nombre\text{-}e}(\sigma_{E.promedio>15 \wedge P.asig=3044}(Preparador \bowtie_{P.ci\text{-}p=E.ci\text{-}e} Estudiante))$

Otro plan



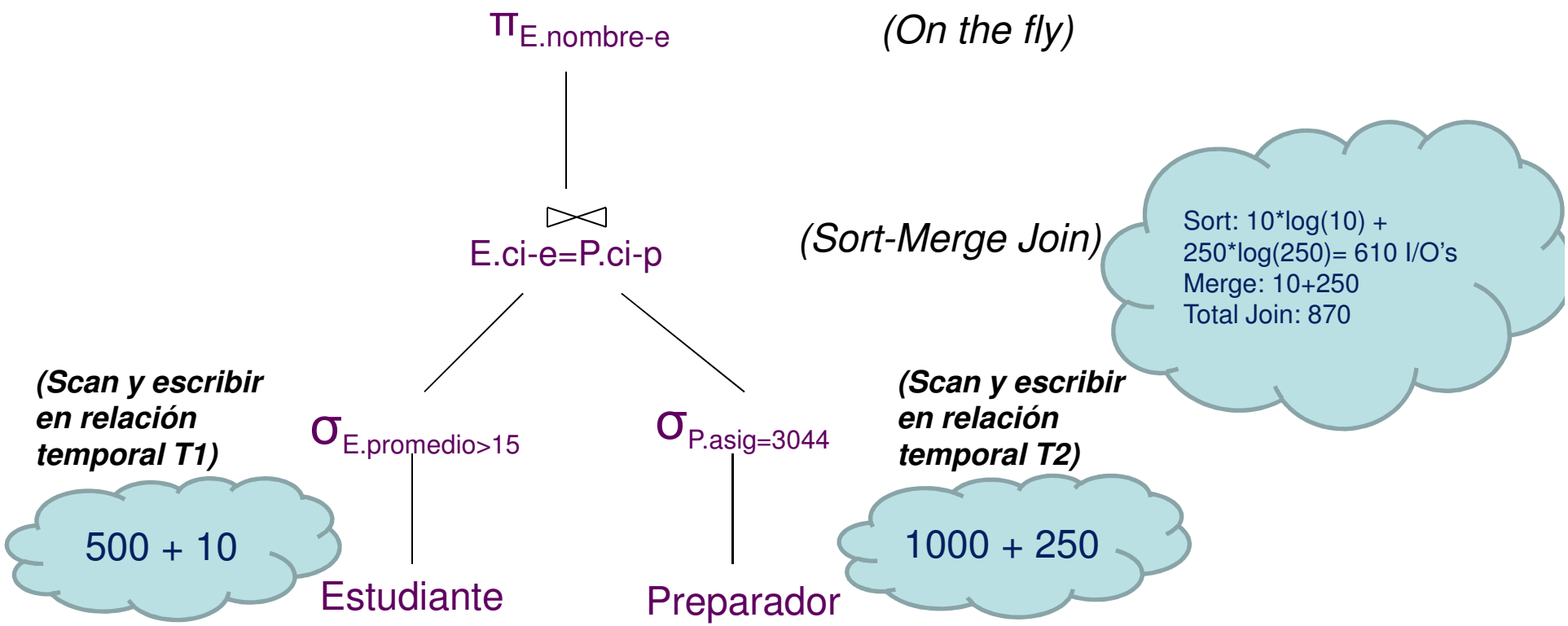
Optimización

Catálogo: Preparador tiene 1000 páginas.
Estudiante tiene 500 páginas.
Suponer que T1 tiene 10 páginas.
Suponer que T2 tiene 250 páginas.



Optimización

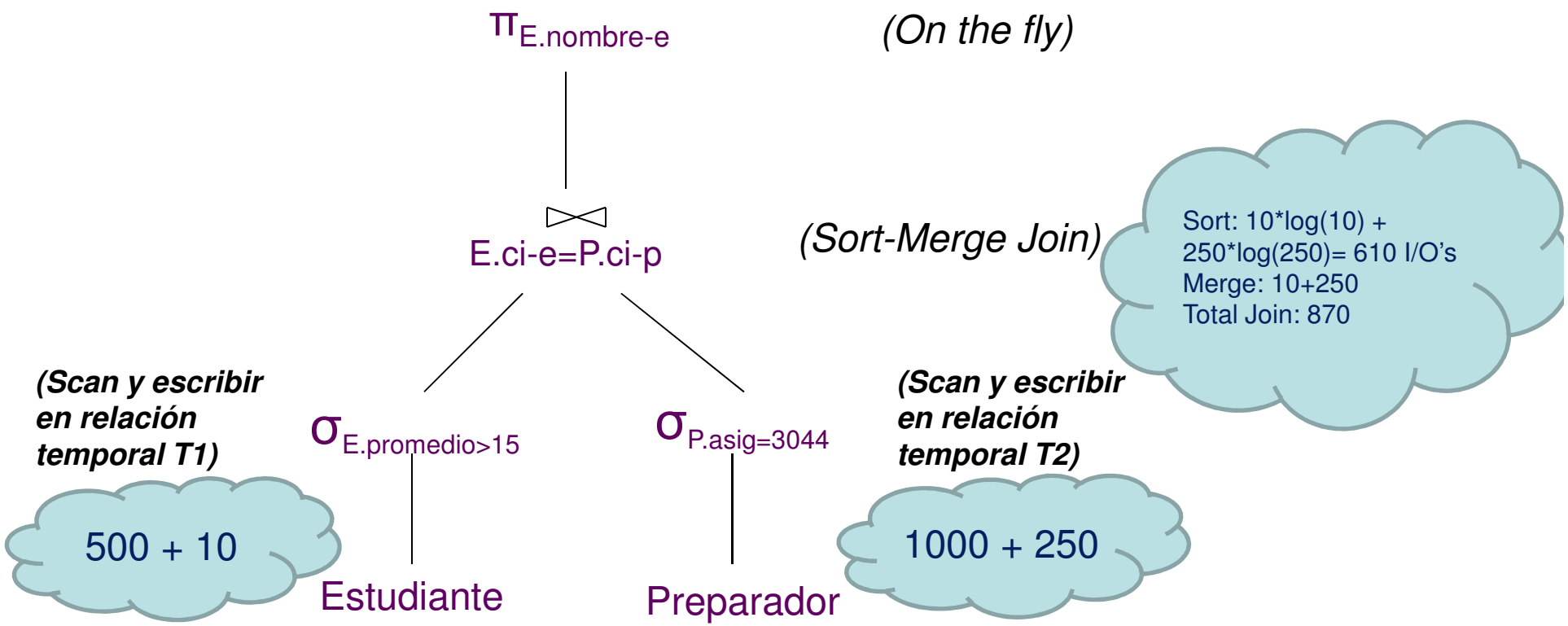
Catálogo: Preparador tiene 1000 páginas.
Estudiante tiene 500 páginas.
Suponer que T1 tiene 10 páginas.
Suponer que T2 tiene 250 páginas.



Optimización

Catálogo: Preparador tiene 1000 páginas.
Estudiante tiene 500 páginas.
Suponer que T1 tiene 10 páginas.
Suponer que T2 tiene 250 páginas.

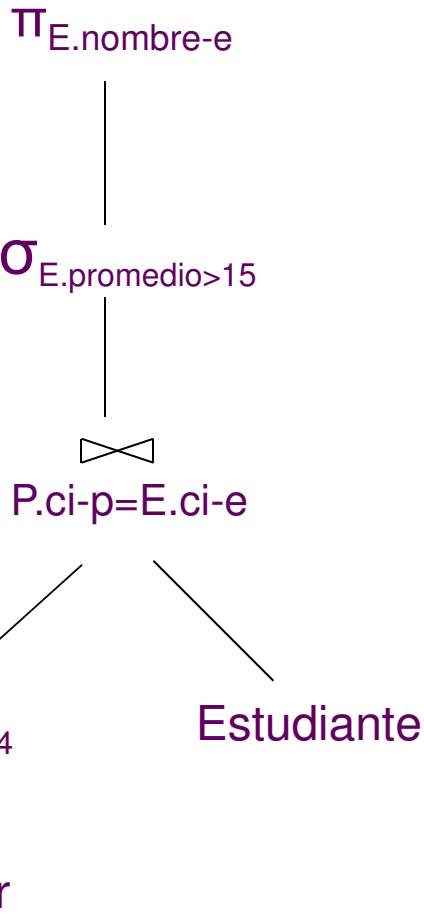
Total Costo: 500 + 10 + 1000 + 250 + 870 = 2.630 I/O's



La expresión en álgebra relacional es:

$\pi_{E.nombre\text{-}e}(\sigma_{E.promedio>15 \wedge P.asig=3044}(Preparador \bowtie_{P.ci\text{-}p=E.ci\text{-}e} Estudiante))$

Otro plan



(On the fly)

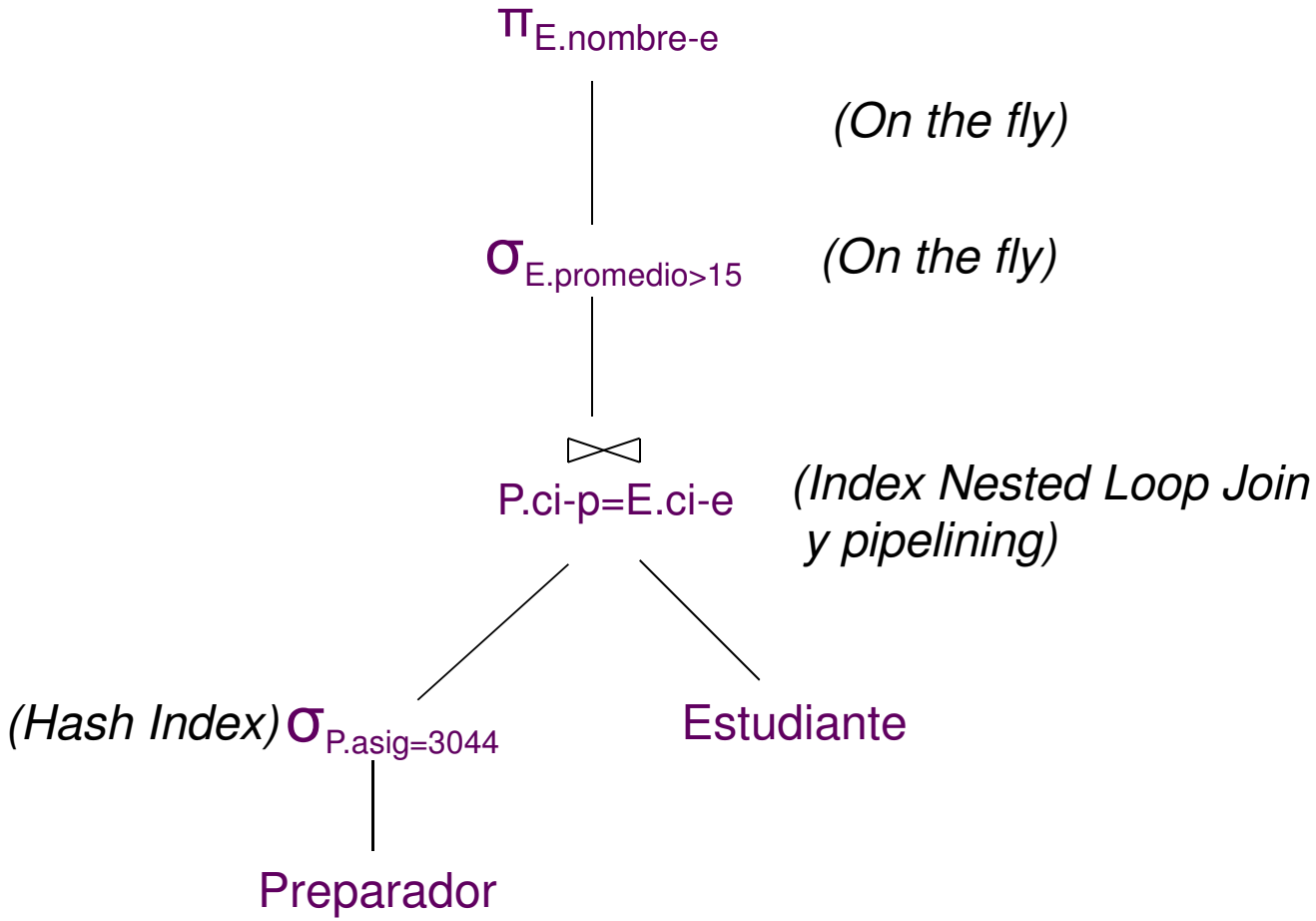
(On the fly)

(Index Nested Loop Join y pipelining)

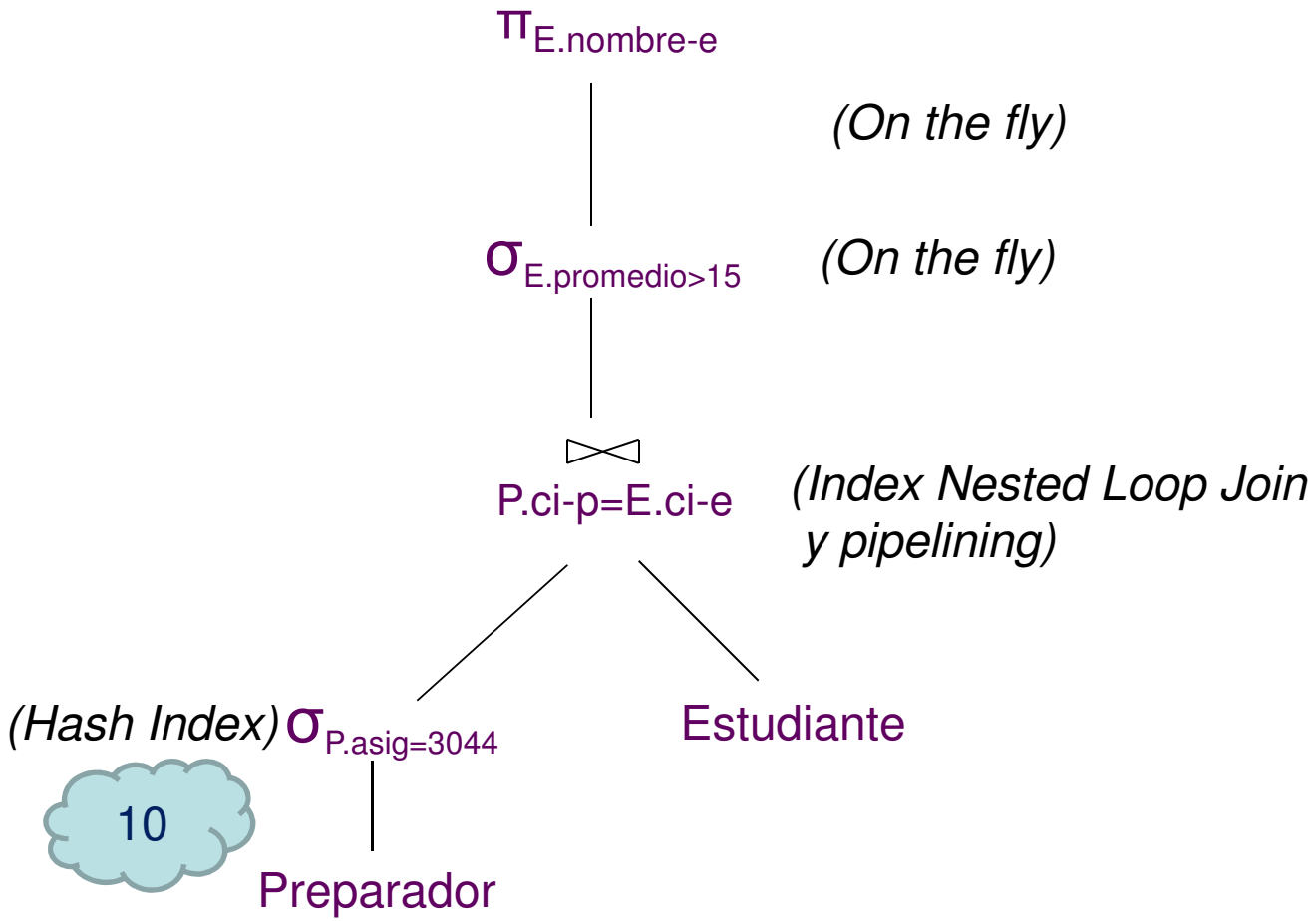
(Hash Index)

Catálogo: Hay un índice clusterizado sobre el atributo **ci-e**.
Hay 1000 tuplas en Preparador con asig=3044

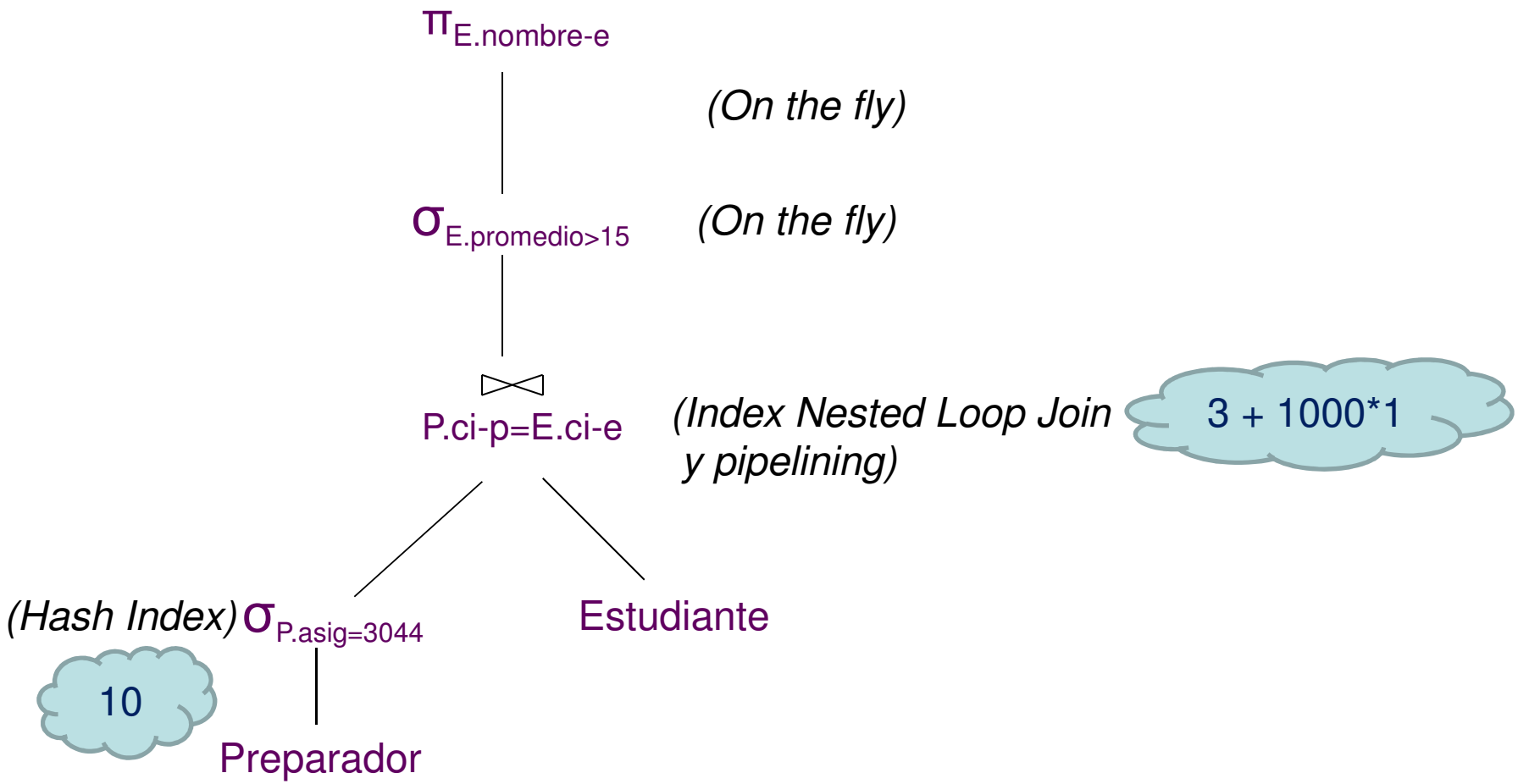
"pipelining" usa memoria intermedia y paralelismo de operadores (no materializa).



Catálogo: Hay un índice clusterizado sobre el atributo **ci-e**.
Hay 1000 tuplas en Preparador con asig=3044



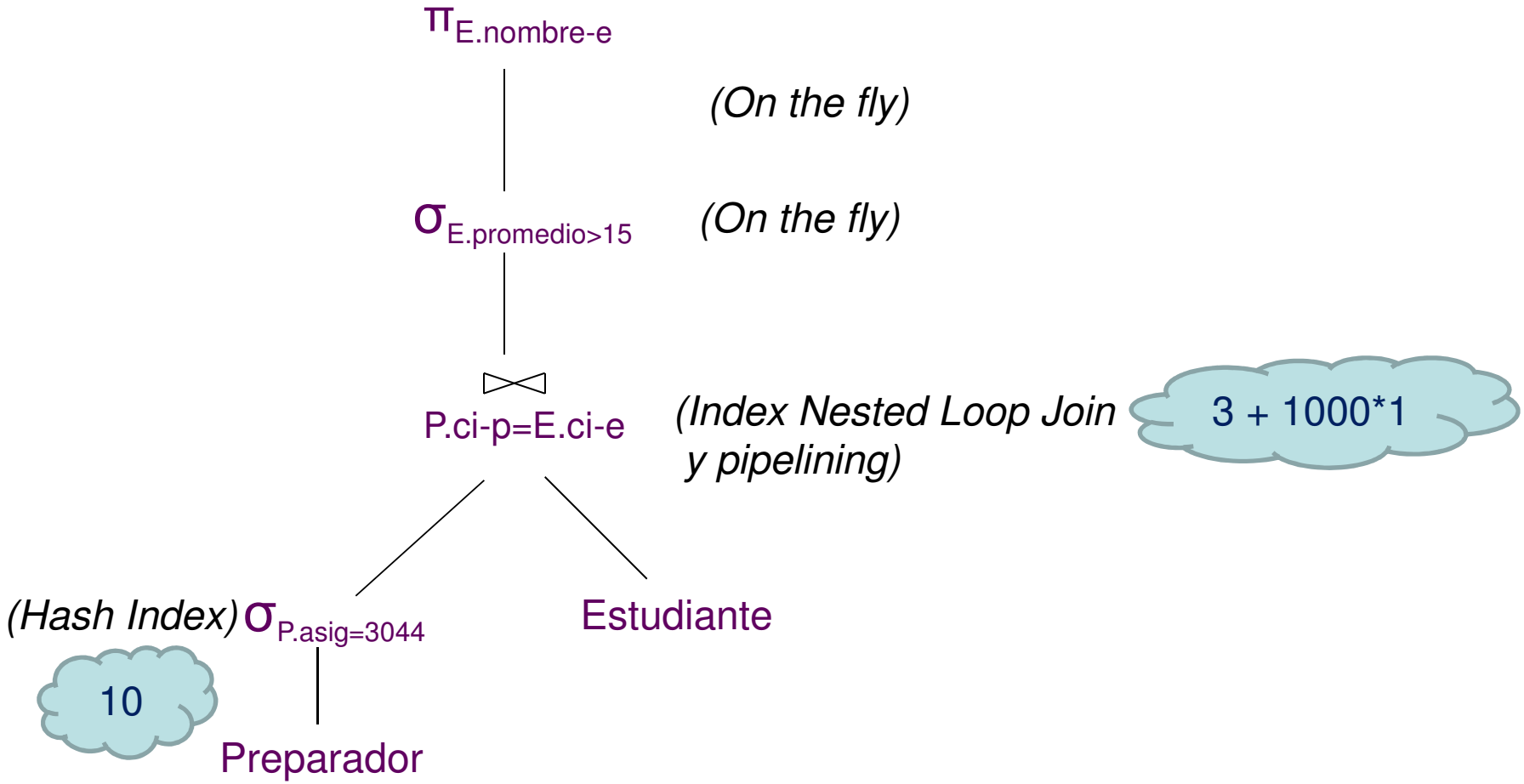
Catálogo: Hay un índice clusterizado sobre el atributo **ci-e**.
Hay 1000 tuplas en Preparador con asig=3044



Optimización

Catálogo: Hay un índice clusterizado sobre el atributo **ci-e**.
Hay 1000 tuplas en Preparador con asig=3044

Total Costo: 10 + 3 + 1000 = 1.013 I/O's



Estrategias de Optimización:

En la optimización de consulta, encontrar la mejor estrategia de optimización, usualmente consume mucho tiempo. Un SGBD debe, sistemáticamente, evaluar las alternativas de ejecución posibles, para una consulta, y escoger la óptima.

Las dos principales estrategias de optimización de consultas son:

Reglas Heurísticas: consiste en ordenar las operaciones en una estrategia de ejecución de consulta. Las reglas, normalmente, reordenan las operaciones en un árbol de consulta o determinan un orden de ejecución de las operaciones especificadas en un grafo de consulta.

Estimación de Costos: se toman diferentes estrategias de ejecución y se selecciona el plan de ejecución de más bajo costo.

Reglas Heurísticas

La principal regla heurística es aplicar operaciones ELEGIR y PROYECTAR antes de aplicar el JOIN u otra operación binaria.

Esto se debe a que el tamaño del archivo resultante de aplicar una operación binaria está en función del tamaño de los archivos de entrada (en algunos casos multiplicativo).

Las operaciones unarias normalmente reducen el tamaño de un archivo, por eso deben aplicar antes que cualquier operación binaria.

Reglas Heurísticas: el reordenamiento de las operaciones en un árbol de consulta es posible debido a las propiedades de los operadores del álgebra relacional:

- Cascada de Selecciones.

  Una condición de selección conjuntiva puede ser dividida en una secuencia de operaciones de conjunción individuales:

  σ c1 AND c2 AND c3 AND...AND cn (R) ≡ σ c1(σ c2(σ c3(...(σ cn(R)....))))

- Conmutatividad de Selecciones.

  σ c1(σ c2(R)) ≡ σ c2 (σ c1 (R))

- Cascada de la proyección.

  En una secuencia de operadores Π, todos excepto el último pueden ser ignorados.

  $$\Pi \text{ lista1 } (\Pi \text{ lista2 } (\ldots \Pi \text{ lista n}(R)\ldots) \equiv \Pi \text{ lista1 } (R)$$

- Conmutatividad de σ con Π : si la condición de selección c sólo involucra los atributos A1,A2,...,An en la lista de proyección, las dos operaciones pueden ser conmutadas.

  $$\Pi \text{ A1,A2,...,An } (\sigma \text{ c } (R)) \equiv \sigma \text{ c } (\Pi \text{ A1,A2,...,An } (R))$$

- Conmutatividad/Asociatividad del Producto Cartesiano/Join.

$$R \times S \equiv S \times R ;$$

$$R \bowtie_c S \equiv S \bowtie_c R$$

$$R \times (S \times P) \equiv (S \times R) \times P ;$$

$$R \bowtie_{c_1} (S \bowtie_{c_2} P) \equiv (P \bowtie_{c2} S) \bowtie_{c_1} R$$

- Conmutatividad de σ con X (o con $\bowtie$):

Si todos los atributos en una condición de selección c son sólo atributos, de una de las relaciones que están siendo reunidas, digamos R, las dos operaciones pueden ser conmutadas.

$$\sigma_c (R \times S) \equiv (\sigma_c (R)) \times S ; \sigma_c (R \bowtie S) \equiv (\sigma_c (R)) \bowtie S$$

- Si la condición de selección c puede ser escrita como (c_1 AND c_2) donde c_1 involucra sólo los atributos de R y c_2 sólo los atributos de S, las operaciones se pueden conmutar:

$$\sigma_c (R \times S) \equiv \sigma_{c_1} (R) \times \sigma_{c_2} (S) ;$$

$$\sigma_c (R \bowtie S) \equiv \sigma_{c_1} (R) \bowtie \sigma_{c_2} (S)$$

Optimización Heurística de Árboles de Consulta

1. Un árbol de consulta es una estructura de árbol que corresponde a una expresión del álgebra relacional, donde las relaciones iniciales son representadas a través de los nodos hojas del árbol y las operaciones son representadas en los nodos internos.

2. Una ejecución de un árbol de consulta consiste en la ejecución de un operación de un nodo interno siempre que sus operadores estén disponibles y luego sustituyendo este nodo por la relación que resulte de ejecutar la operación.

3. La ejecución termina cuando el nodo raíz es ejecutado y produce la relación resultante.

Ejemplo

TRABAJA-EN(c.i., pnum, horas)
EMPLEADO(c.i., enombre,dirección, sexo,dnum, fnac)
DEPTO(dnombre,dnumero,ci-gerente)
PROYECTO(pnombre,pnumero,dnum,plocalización)

Consulta: por cada proyecto localizado en "ccs", indicar el número de proyecto, número de departamento que lo controla e indicar el nombre y sexo del gerente.

TRABAJA-EN(c.i., pnum, horas)
EMPLEADO(c.i., enombre,dirección, sexo,dnum, fnac)
DEPTO(dnombre,dnumero,ci-gerente)
PROYECTO(pnombre,pnumero,dnum,plocalización)

En SQL:

SELECT pnumero,dnumero,enombre,sexo
FROM Proyecto, Depto, Empleado
WHERE dnum=dnumero AND ci-gerente = ci AND plocalización =”ccs”

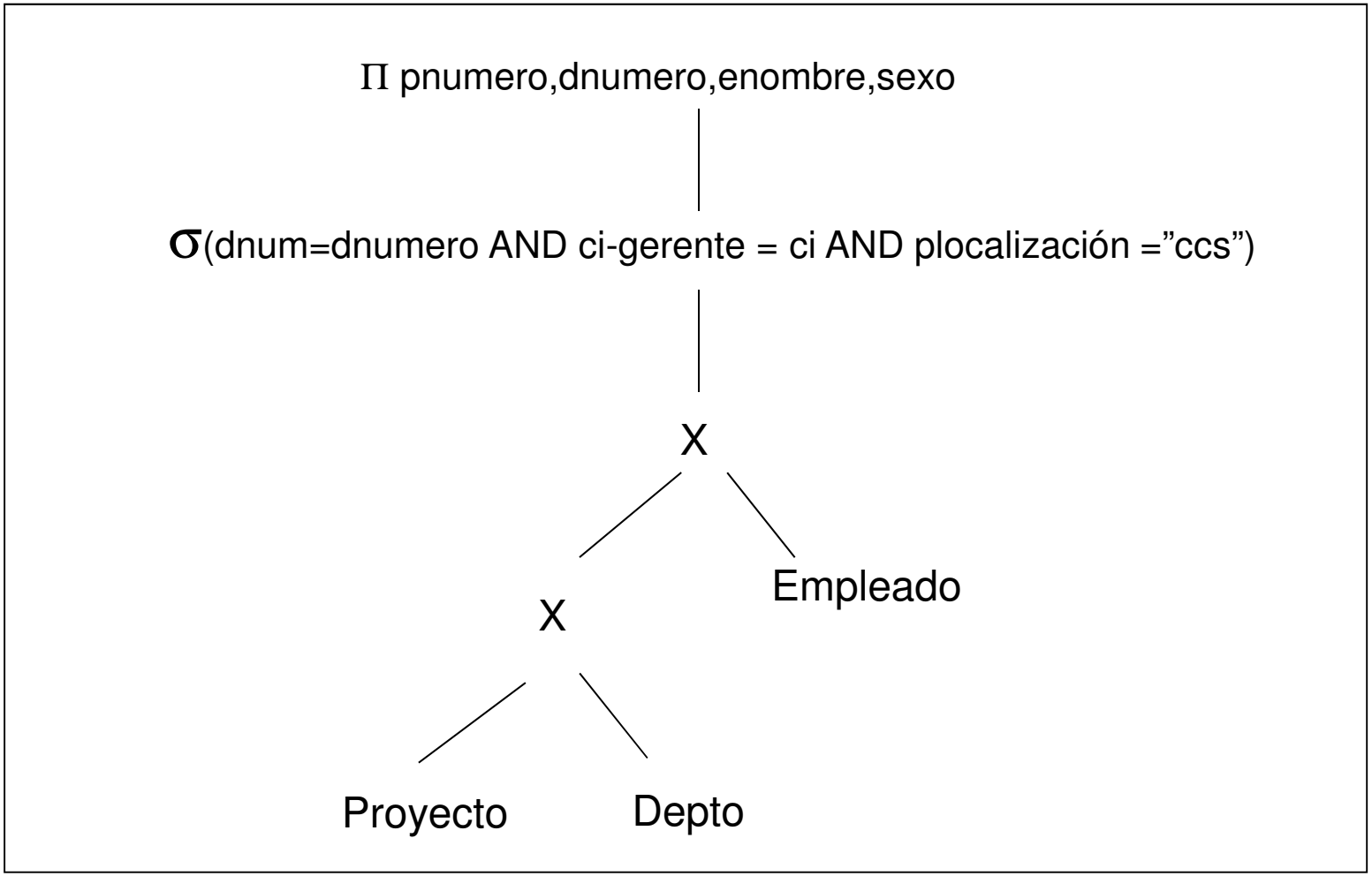
En Algebra Relacional

Πpnumero,dnumero,enombre,sexo

(σ dnum=dnumero $\wedge$ ci-gerente = ci $\wedge$ plocalización=”ccs” (Proyecto X Depto) X Empleado))

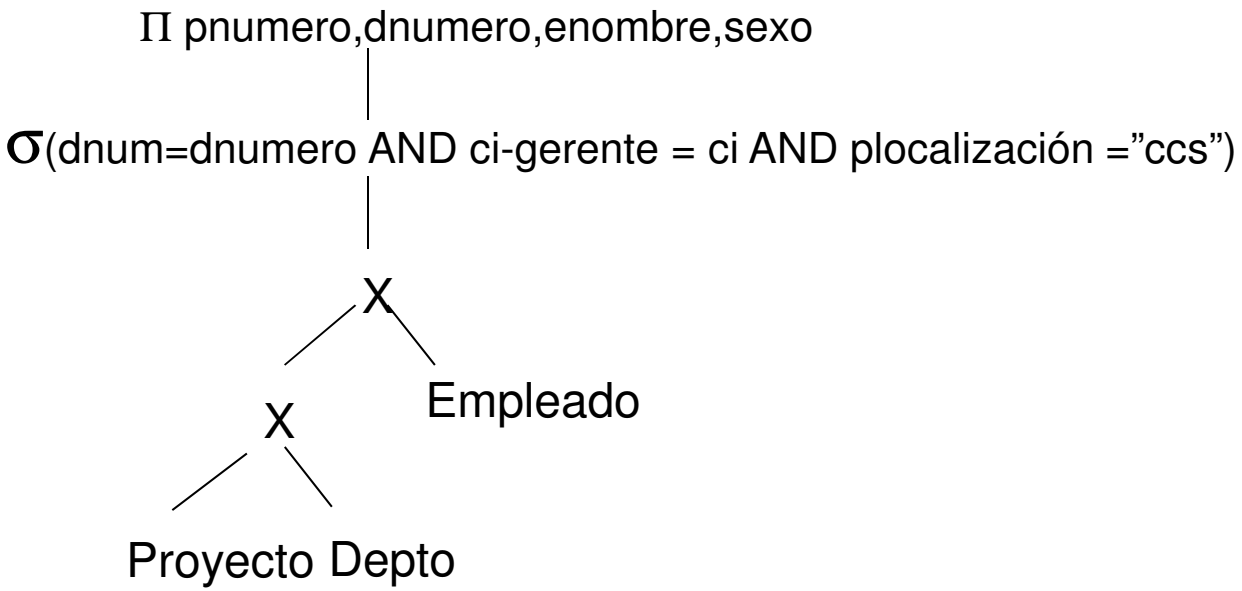
Πpnumero,dnumero,enombre,sexo

(σ dnum=dnumero ∧ ci-gerente = ci ∧ plocalización=”ccs” (Proyecto X Depto) X Empleado))



Arbol Canónico Inicial para la Consulta en SQL

Optimización



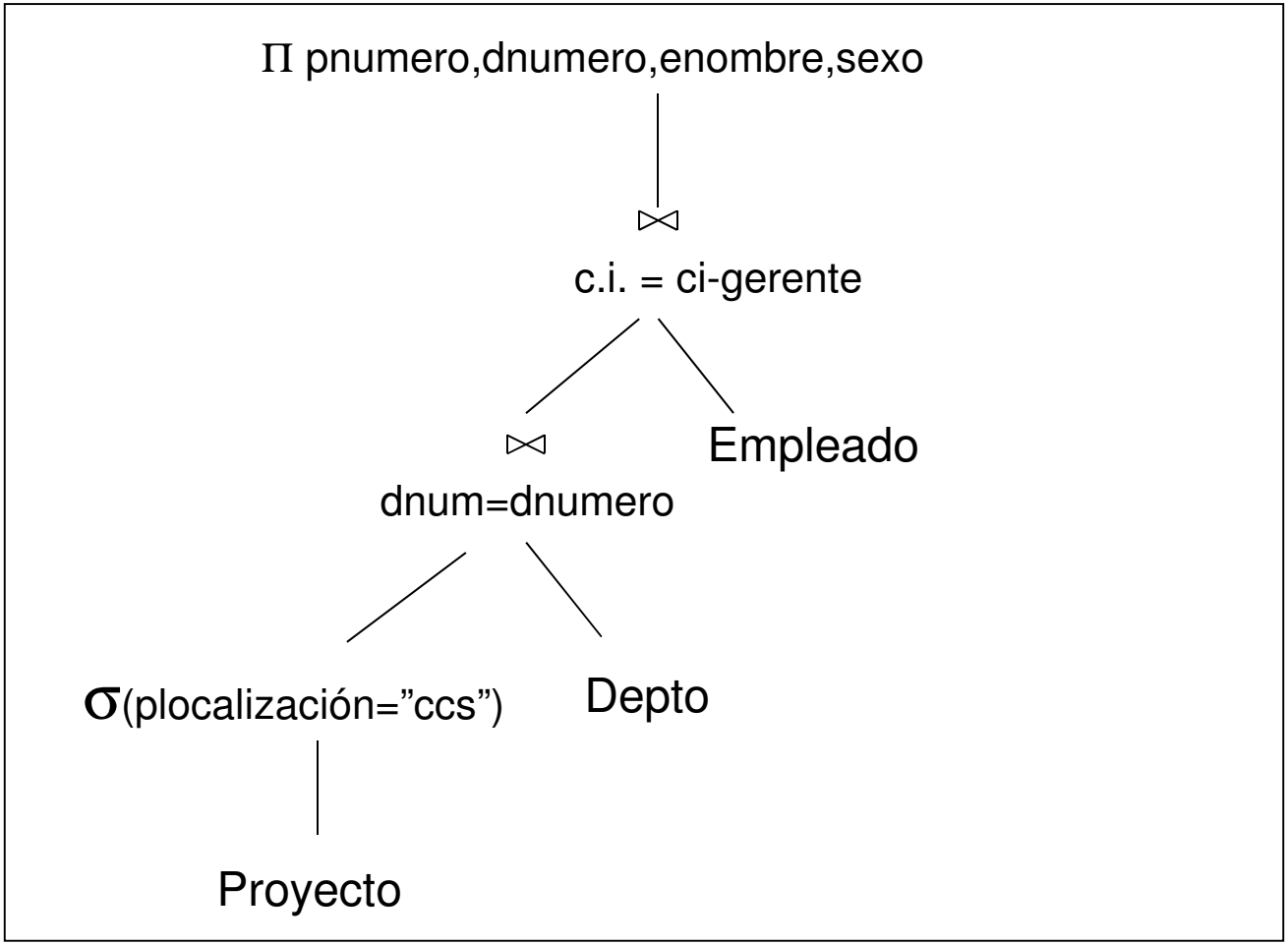
El Analizador de consultas genera una representación del árbol canónico para la consulta en SQL sin hacer optimización alguna. Tal árbol canónico representa una expresión del álgebra relacional que es muy ineficiente, si se ejecuta directamente, debido a los productos cartesianos.

El optimizador de consultas transforma

Árbol inicial de consulta (Canónico) → **Árbol final de consulta**

Πpnumero,dnumero,enombre,sexo

(σ dnum=dnumero ∧ ci-gerente = ci ∧ plocalización=”ccs” (Proyecto X Depto) X Empleado))



Posible Árbol Final

Reglas de Transformación

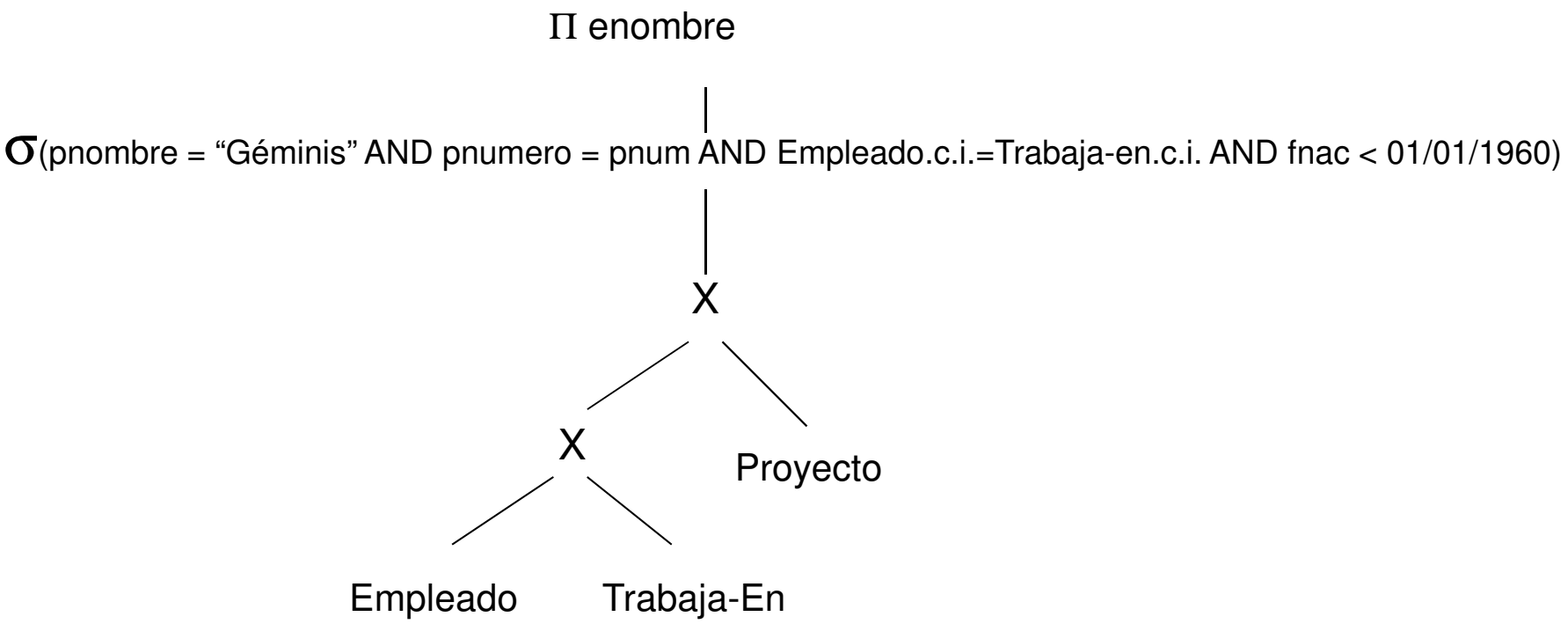
TRABAJA-EN(c.i., pnum, horas)
EMPLEADO(c.i., enombre,dirección, sexo,dnum, fnac)
DEPTO(dnombre,dnumero,ci-gerente)
PROYECTO(pnombre,pnumero,dnum,plocalización)

Consulta: **Listar los *nombres de los empleados* nacidos antes de *1960* que trabajen en un proyecto llamado *Géminis***

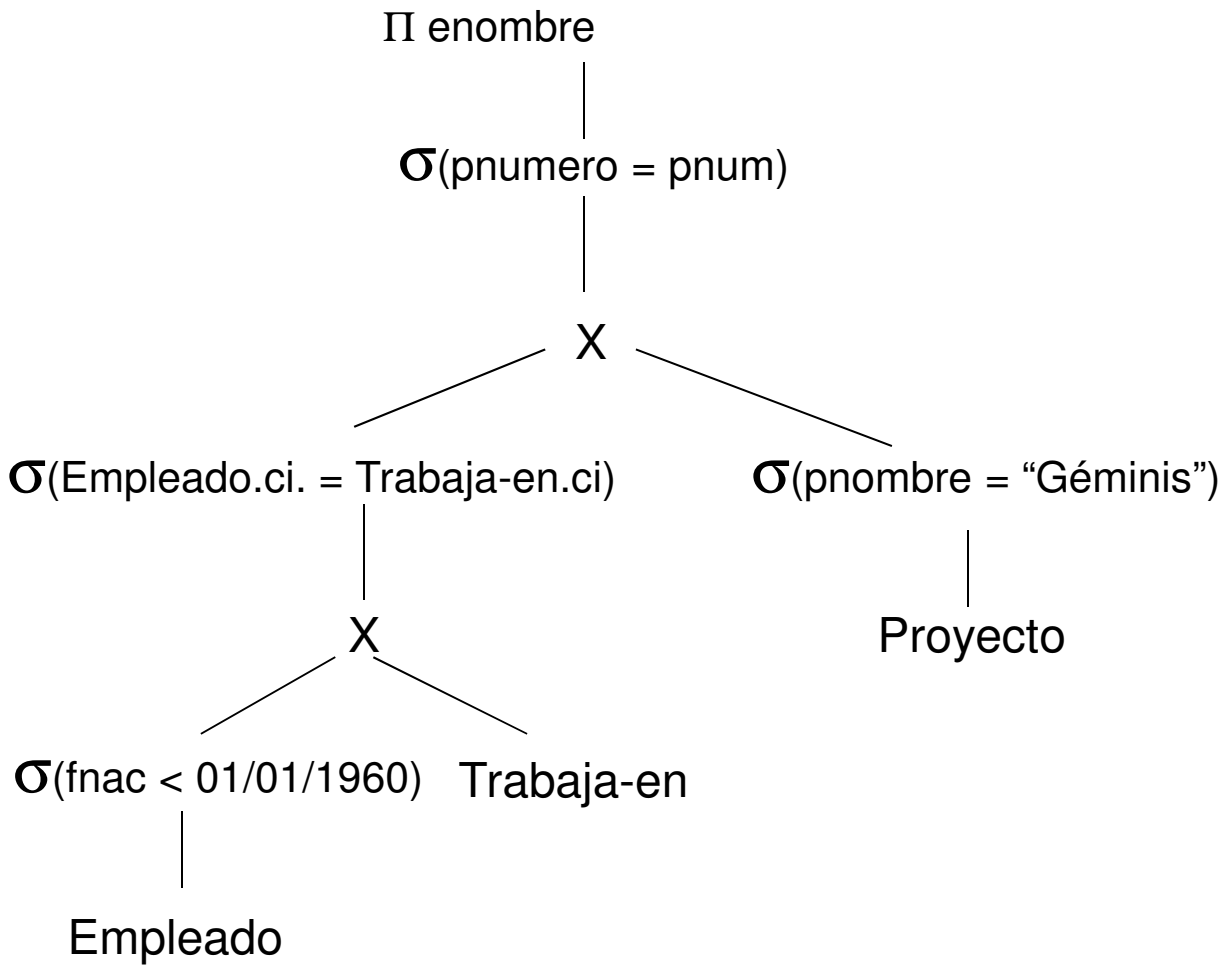
```
SELECT enombre
FROM     Empleado, Trabaja-en, Proyecto
WHERE pnombre = “Géminis” AND
            pnumero = pnum AND
            Empleado.c.i.=Trabaja-en.c.i. AND
            fnac < 01/01/1960
```

Reglas de Transformación

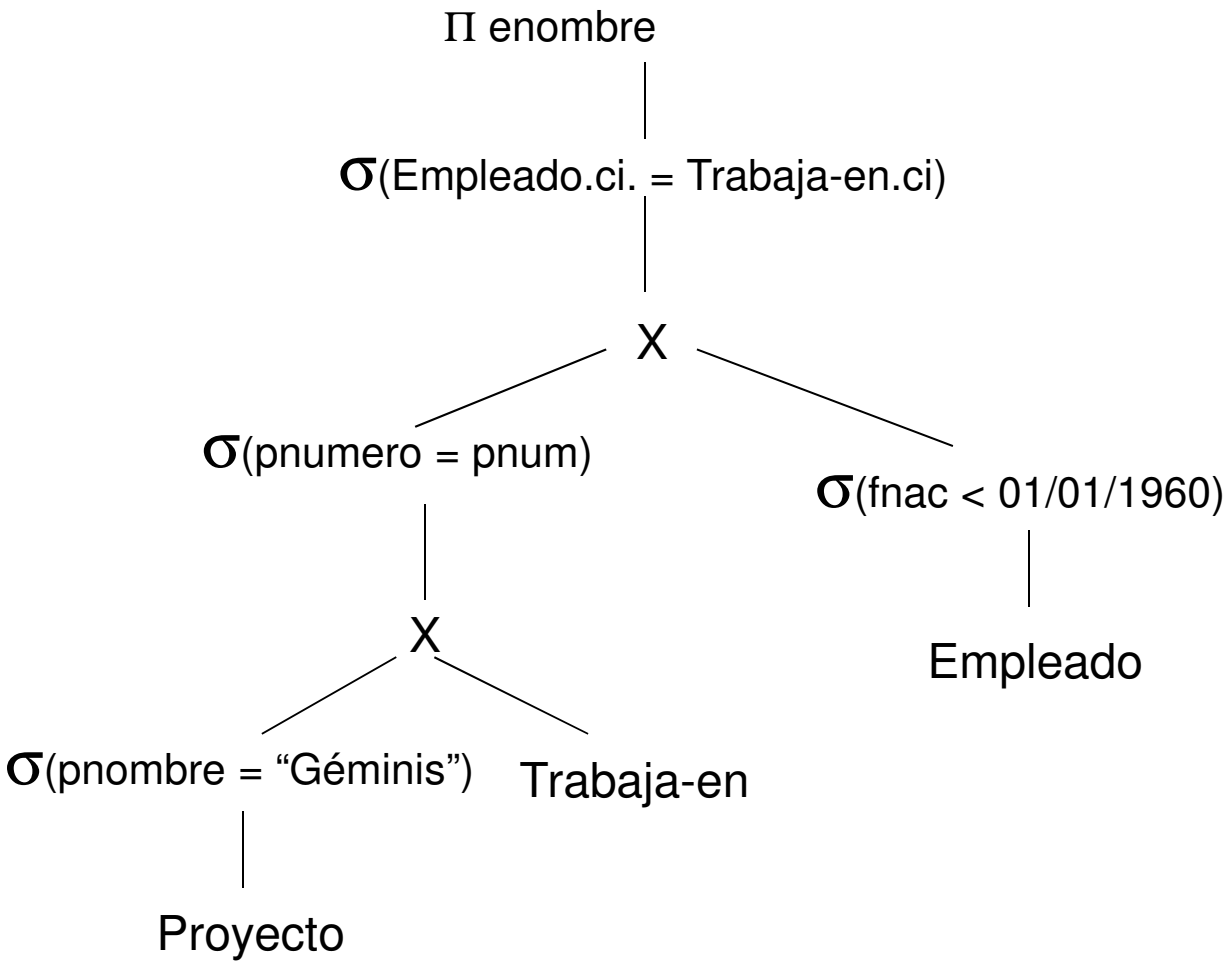
A. Se construye el árbol canónico



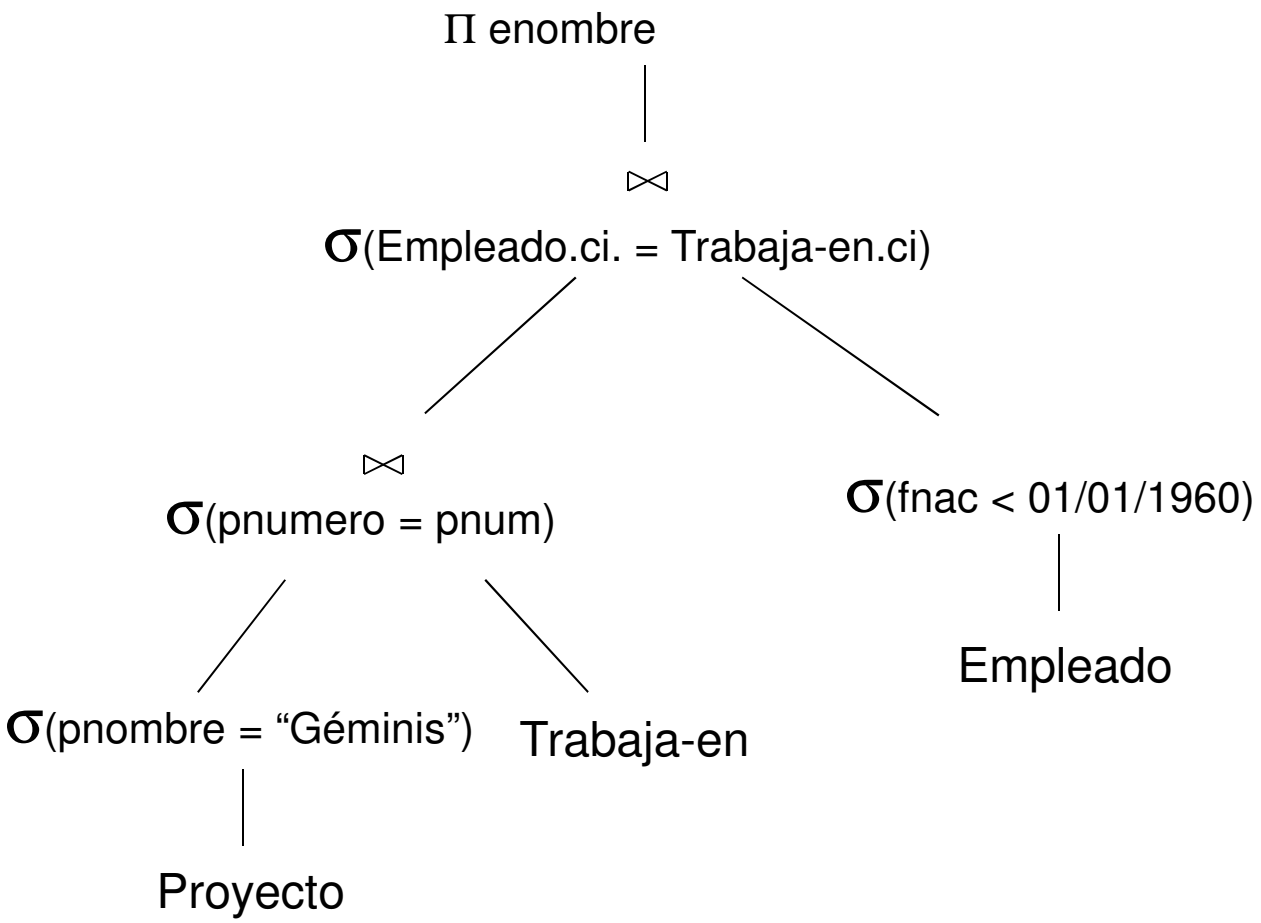
B. Se mueve el operador selección hacia abajo en el árbol.



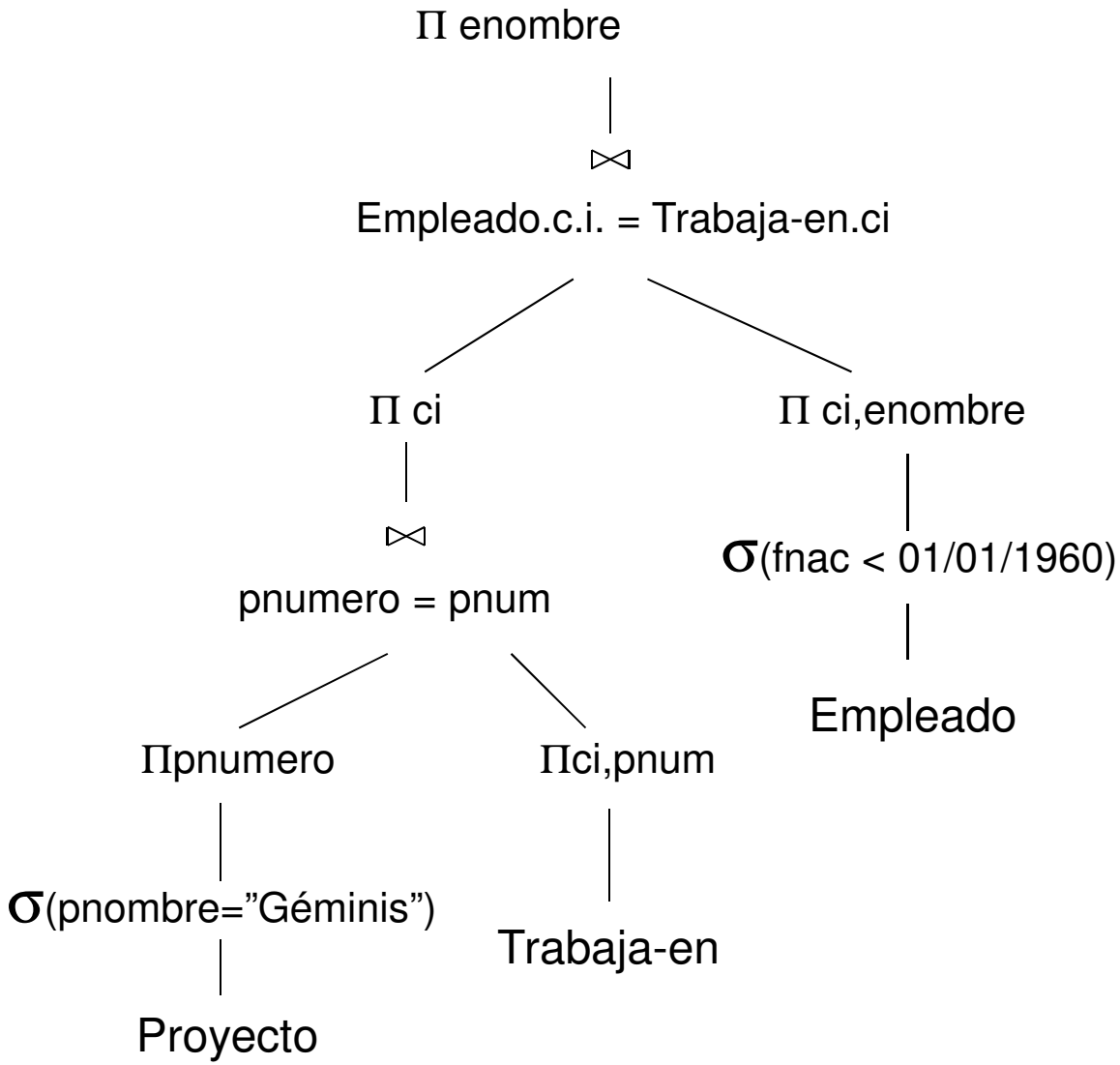
C. Se hacen más selectivas las operaciones en las hojas.



D. Se sustituyen los productos cartesianos por el JOIN.

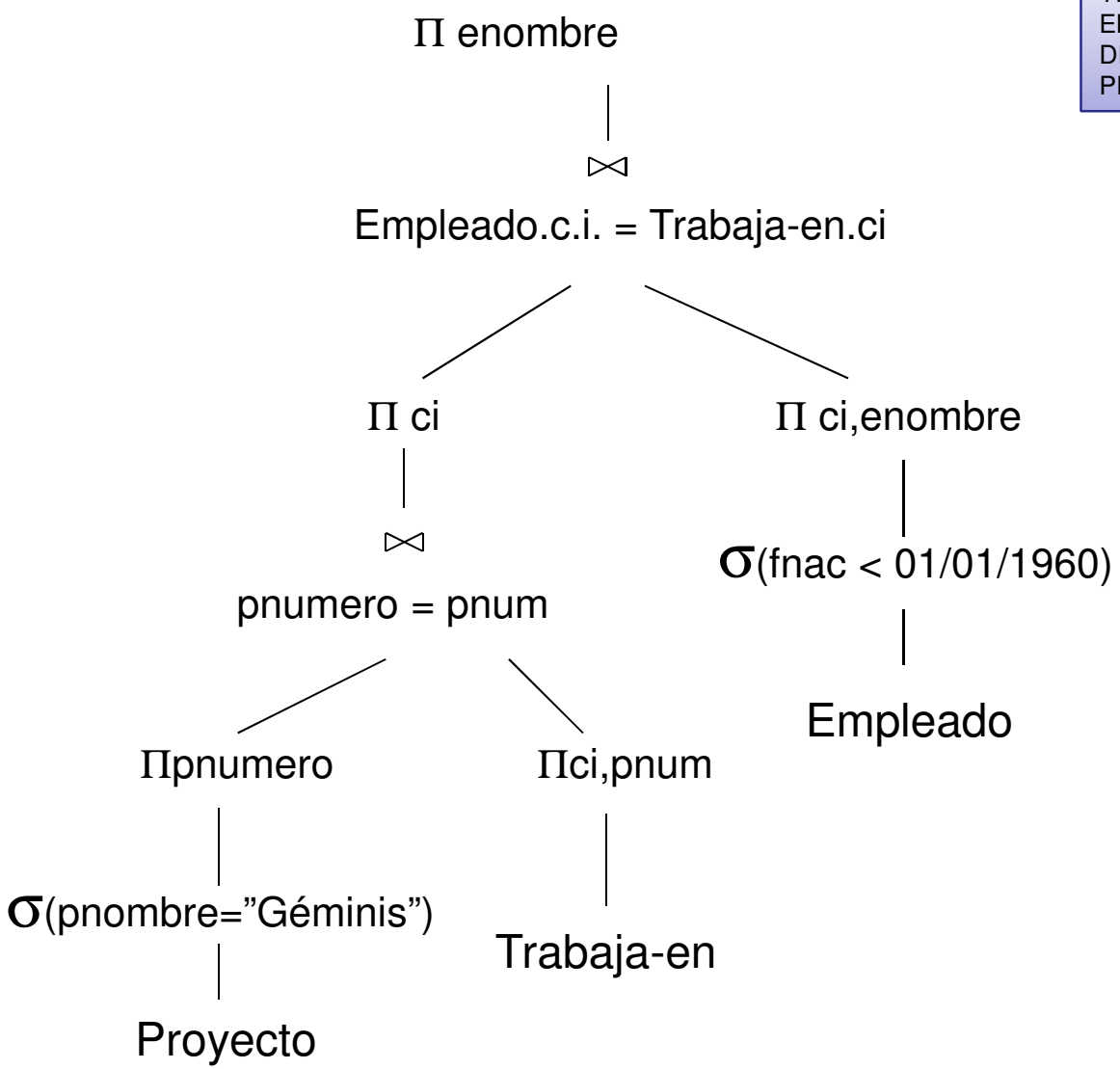


E. Se mueve las PROYECCIONES hacia las hojas para filtrar los atributos no necesarios.



Calculemos el costo de evaluar los operadores con diferentes algoritmos
Según la información del catalogo.

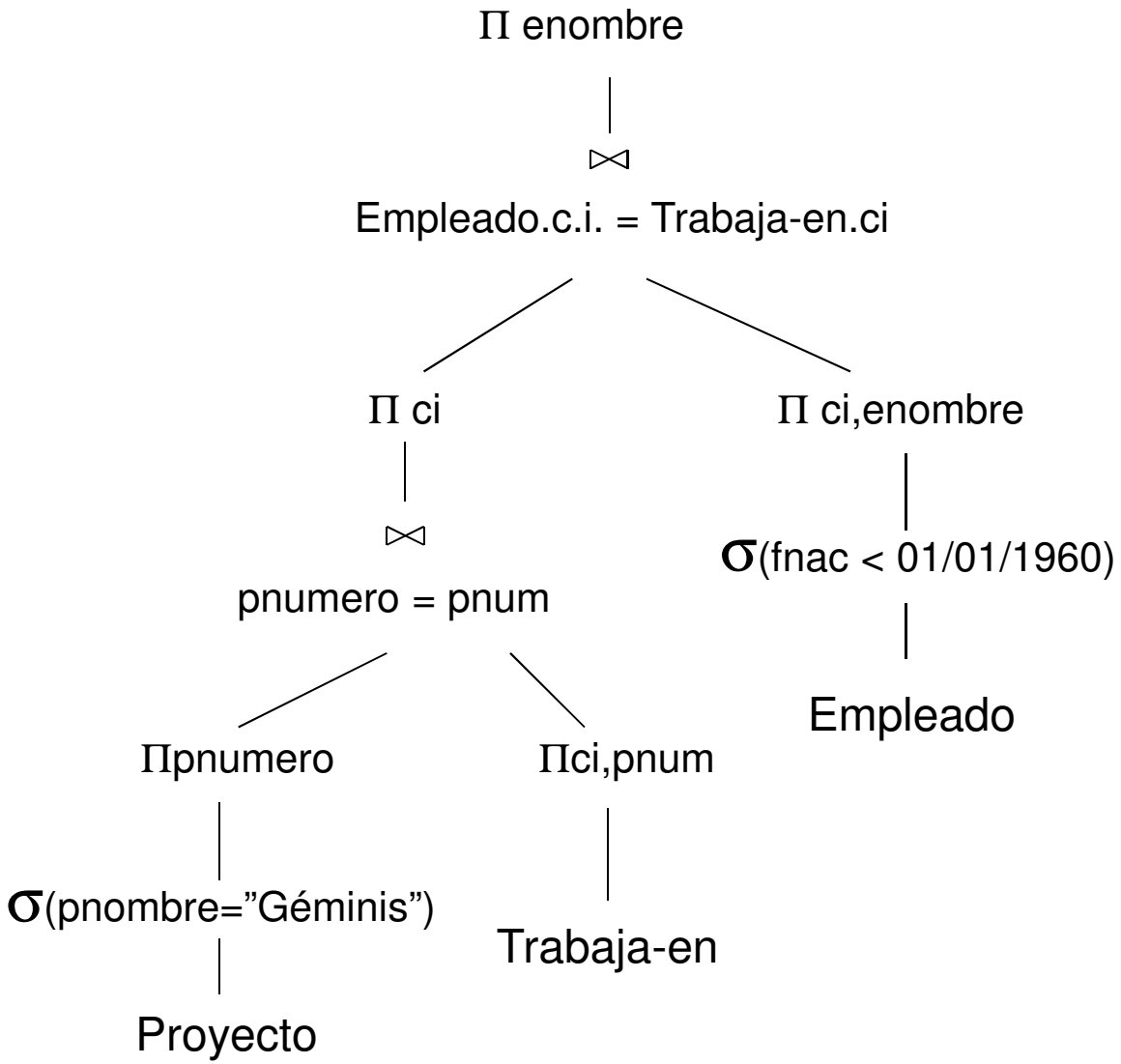
TRABAJA-EN(c.i., pnum, horas)
EMPLEADO(c.i., enombre,dirección, sexo,dnum, fnac)
DEPTO(dnombre,dnumero,ci-gerente)
PROYECTO(pnombre,pnumero,dnum,plocalización)



Optimización

Catálogo: Proyecto tiene 400 páginas, 100 tuplas por página.
Empleado tiene 1000 páginas, 50 páginas por página.
Trabaja-en tiene 2000 páginas, 100 tuplas por página.
200 empleados nacieron antes de 1960.

Probar NLJ, HJ.



Fuentes consultadas:

[1] Prof. Elsa Liliana Tovar.
Notas de clase compiladas.

[2] Ramakrishnan Raghu. ,
“Database Management Systems".

[3] Navathe, Elsamri,
“Fundamentals of DataBase System”.