

ABD 21/22 Práctica 2 - Optimización de consultas

Clara María Romero Lara

Tablas

Profesor(DNI, Apellidos, Nombre, Departamento)	Asignatura(ID, DNI, Créditos, Curso)
N(Profesor) = 100	N(Asignatura) = 250
Size(DNI) = 10, char	Size(DNI) = 10, char
Size(Apellidos) = 200, varchar2	
Size (Nombre) = 100, varchar2	
Size (Departamento) = 6, varchar2	
	Size(ID) = 5, varchar2
	Size(Créditos) = 4, int
	Size (Curso) = 4, int
V (Apellidos) = 80	
V (Nombre) = 70	
	V(Créditos) = 2
	V(Curso) = 4

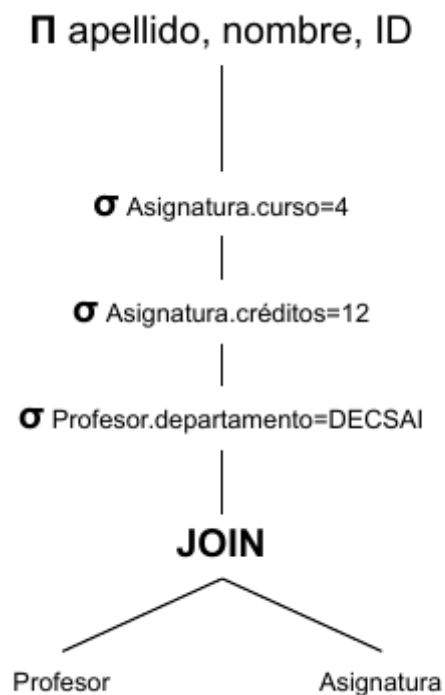
- DNI es **clave primaria** del profesor. Son 10 char (9 cifras y una letra)
- ID es **clave primaria** de asignatura. Son hasta 5 varchar (las siglas de la asignatura)
 - *Asignatura.DNI* (profesor que imparte la asignatura) es clave externa a la relación Profesor.

- Las asignaturas pueden valer 6 o 12 créditos.
- Las asignaturas pueden pertenecer a 1º, 2º, 3º o 4º curso.
- La variabilidad de V(DNI) y V(ID) es igual a N(Profesor) y N(Asignatura) respectivamente, ya que son claves primarias únicas.

Tamaño de bloque 5KB, cabecera 50B. Usamos bloqueo fijo y bloques homogéneos. Selección y proyección respetan el índice.

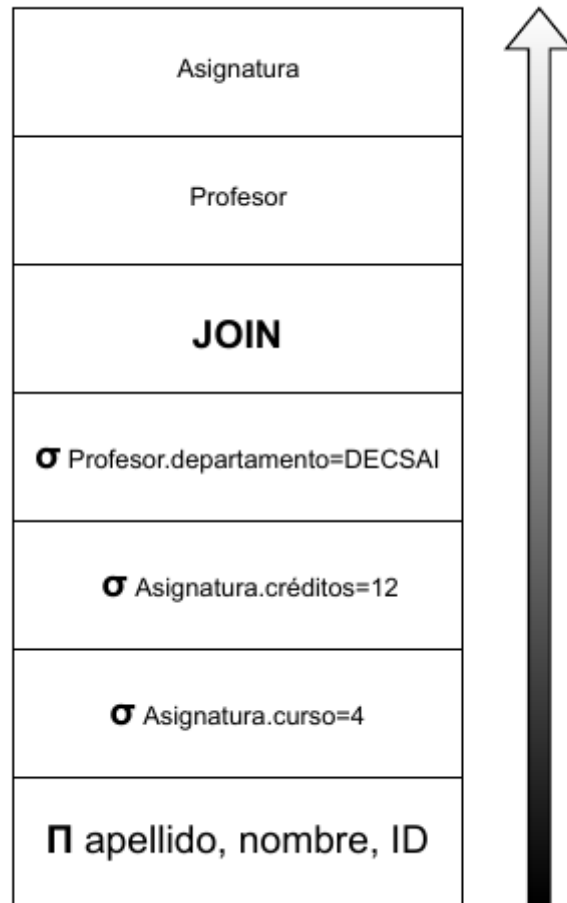
Consulta

```
SELECT Profesor.Apellidos, Profesor.Nombre, Asignatura.ID FROM Profesor,  
Asignatura WHERE (Asignatura.Curso=4 AND Asignatura.Creditos=12) JOIN  
Profesor.Departamento=DECSAI
```



Representación en memoria y plan lógico

La representación en memoria escogida será una pila en la que se introducirán los elementos en el orden en el que aparecen en el árbol: primero la proyección, luego las selecciones, el join y finalmente las tablas.



De esta forma, podemos resolver la consulta planteada empezando desde las tuplas de las tablas, e ir obteniendo con las operaciones de selección y proyección las tuplas que nos interesan.

Plan físico

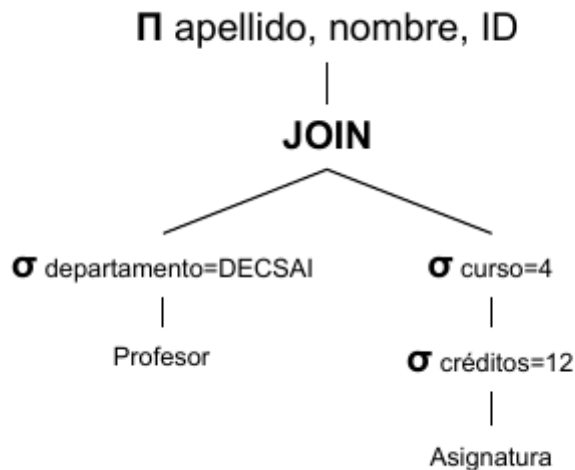
Manejaremos las operaciones del plan físico con una cola: la estructura FIFO nos permite que las operaciones de lectura y escritura se completen en el orden correcto, ya que es necesario que las operaciones de lectura sean previas a la escritura (Orden L>E>L>E...).

Si usamos las operaciones básicas de una cola (`push` para introducir un elemento al final, `pop` para sacar el elemento más antiguo), la secuencia de acontecimientos sería `push(operación 1)` para la lectura de los resultados de la operación, `push(operación 2)` (donde operación 2 usa los resultados de la primera); y finalmente `pop()`, que sacaría los resultados de operación 1 de la cola, pues ya los hemos utilizado.

Optimización de la consulta

Algoritmo 1: selección lo más pronto posible

En esta primera mejora, las operaciones de selección se han adelantado al JOIN. De esta forma juntar las dos tablas será menos costoso, ya que se hará sobre un número menor de tuplas, el resultado de las selecciones.



Algoritmo 2: proyección y selección lo más pronto posible

En esta segunda mejora, se incluye a lo anterior una proyección previa al JOIN con los atributos que nos interesan. Estos son: los atributos que queremos obtener de la consulta; y el atributo DNI, puesto que es el criterio común con el que se van a ordenar las tablas.

De esta forma no solo hemos reducido las tuplas que obtenemos, sino que también su factor de bloqueo será menor.

