

Taller #10 – Optimización de consultas

Base de Datos I

I. Álgebra relacional y árbol de consulta

1. Presente el código SQL y la sentencia de Algebra Relacional equivalente, para realizar las siguientes consultas, considerando las tablas que se presentan abajo.

- Nombres y apellidos de los estudiantes que tienen resultados del ejercicio A – 1, con la cantidad de puntos que obtuvieron.
- Correos de los estudiantes que tienen resultados para ejercicios del tópico SQL.

ESTUDIANTES			
CODIGO(PK)	NOMBRES	APELLIDOS	CORREO
101	Pedro	Perez	pperez@uaq
102	Monica	Mejia	mmejia@uaq
103	Ramiro	Gonzalez	rgonza@uaq
104	Juan	Martinez	jmartin@uaq

RESULTADOS			
COD_EST	CAT_EJER	NUM_EJER	PUNTOS
101	A	1	10
101	A	2	8
102	A	1	9
102	A	2	9
102	B	1	12

EJERCICIOS			
CATEGORIA(PK)	NUMERO(PK)	TOPICO	MAXPUNTO
A	1	Modelamiento	10
A	2	SQL	10
B	1	SQL	15

Solución:

1.

a):

```
1 SELECT ESTUDIANTES.NOMBRES, ESTUDIANTES.APELLIDOS, RESULTADOS.PUNTOS
2 FROM ESTUDIANTES
3 JOIN RESULTADOS ON ESTUDIANTES.CODIGO = RESULTADOS.COD_EST
4 WHERE RESULTADOS.CAT_EJER = 'A' AND RESULTADOS.NUM_EJER = 1;
```

Álgebra Relacional:

```
1  $\pi$  NOMBRES, APELLIDOS, PUNTOS ( $\sigma$  CAT_EJER='A' AND NUM_EJER=1 (ESTUDIANTES
 $\bowtie$  COD_EST = CODIGO RESULTADOS))
```

b):

```
1 SELECT ESTUDIANTES.CORREO
2 FROM ESTUDIANTES
3 JOIN RESULTADOS ON ESTUDIANTES.CODIGO = RESULTADOS.COD_EST
4 JOIN EJERCICIOS ON RESULTADOS.CAT_EJ = EJERCICIOS.CATEGORIA AND
5 RESULTADOS.NUM_EJER = EJERCICIOS.NUMERO
6 WHERE EJERCICIOS.TOPICO = 'SQL';
```

Álgebra Relacional:

```
1  $\pi$  CORREO ( $\sigma$  TOPICO='SQL' (ESTUDIANTES  $\bowtie$  COD_EST = CODIGO (RESULTADOS  $\bowtie$ 
CAT_EJER = CATEGORIA AND NUM_EJER = NUMERO EJERCICIOS)))
```

2. Considerando la descripción de tablas que se presenta a continuación, describa la consulta (mostrada al final) como expresión de Álgebra Relacional:

AEROPUERTOS (aerold, aeronombre, ciudad)

VUELOS (vuelold, fecha, compañía, **aeroorigen**, **aerodestino**)

RESERVAS (**numtiquete**, nombre, nacionalidad, **numvuelor**, **silla**)

SILLAS (numsilla, **numvuelo**, clase)

Observación: los campos subrayados corresponden a las llaves primarias de cada tabla, los campos en negrilla corresponden a Llaves foráneas.

```
SELECT nombre, silla, vuelold
FROM SILLAS, VUELOS inner join AEROPUERTOS on aeroorigen = aerold, RESERVAS
WHERE nacionalidad = 'Colombia' and compañía = 'Avianca' and ciudad = 'Miami' or ciudad = 'Los Angeles' and numvuelor = numvuelo and silla = numsilla and numvuelo = vuelold
```

Solución:

2.

Álgebra Relacional:

$\pi_{\text{nombre, silla, vueloId}}$

$(\sigma_{\text{nacionalidad}='Colombia' \wedge \text{compan} \sim \text{ia}='Avianca' \wedge \text{ciudad}='Miami' \vee \text{ciudad}='LosAngeles' \wedge \text{numvuelor}=\text{numvuelo} \wedge \text{silla}=\text{numsilla} \wedge \text{numvuelo}=\text{vueloId}} (SILLAS \bowtie_{\text{numvuelo}=\text{vueloId}} (aeroId/ \text{aeroorigen} (VUELOS) \bowtie_{\text{aeroorigen}=\text{aerold}} AEROPUERTOS) \bowtie_{\text{numvuelor}=\text{numtiquete}} RESERVAS))$

3. Para la siguiente consulta, presente el árbol de consulta equivalente:

PROVEEDOR (nit, nombre, dirección, teléfono, paginaweb)

REPUESTOS (numero, descripción, nomRepuesto, precio)

PROYECTO (código, título, duración, presupuesto)

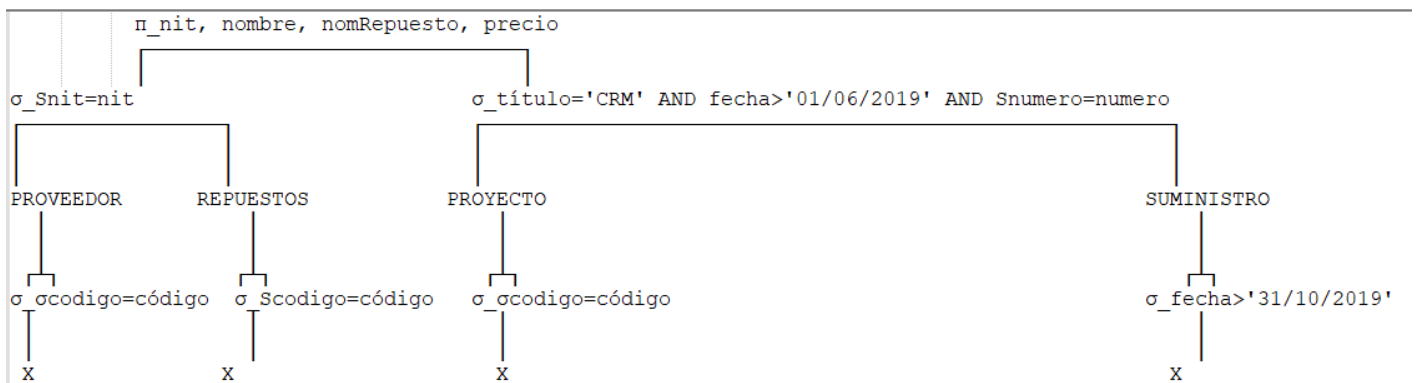
SUMINISTRO (Snit, Snumero, Scódigo, cantidad, fecha)

$\pi_{\text{nit, nombre, nomRepuesto, precio}} (\sigma_{\text{Snit}=\text{nit} \text{ AND } \text{título}='CRM' \text{ AND } \text{fecha} > '01/06/2019' \text{ AND } \text{Snumero}=\text{numero} \text{ AND } \text{precio} > 100.000 \text{ AND } \text{fecha} < '31/10/2019' \text{ AND } \text{Scodigo}=\text{código}} (\text{PROVEEDOR X REPUESTOS X PROYECTO X SUMINISTRO}))$

Solución:

3.

Árbol de consulta equivalente:



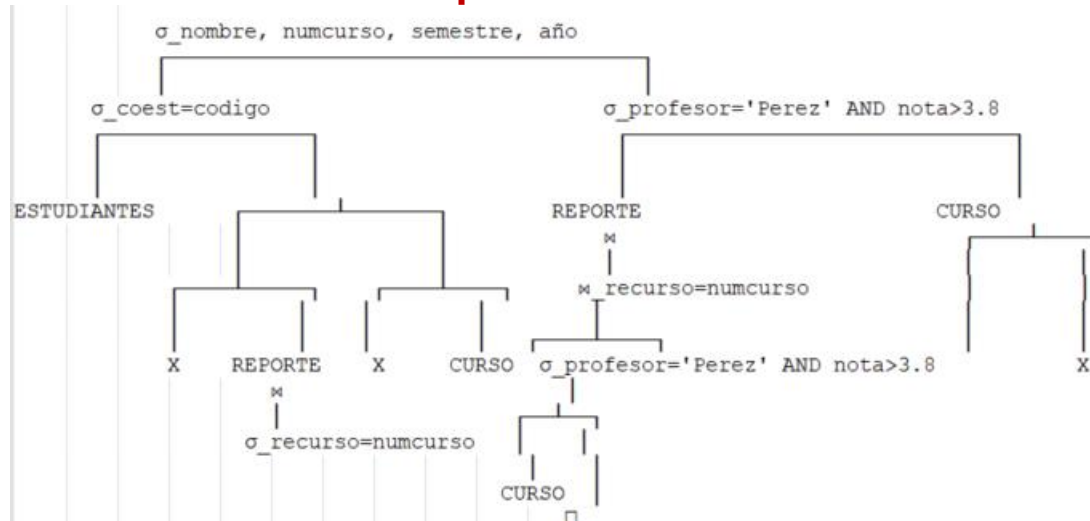
4. Presente el árbol de consulta equivalente a la siguiente expresión de álgebra relacional:

$\sigma_{\text{nombre, numcurso, semestre, año}} (\sigma_{\text{coest=codigo}} (\text{ESTUDIANTES} \times (\text{REPORTE} \bowtie_{\text{recurso=numcurso}} (\sigma_{\text{profesor='Perez' and nota > 3.8}} (\text{CURSO}))))))$

Solución:

4.

Árbol de consulta equivalente:



II. Optimización de consultas

Para las consultas que se plantean a continuación, presente:

- Árbol de consulta inicial
- Justificación de los cambios realizados en el orden de las tablas (hojas del árbol).
- Árbol de consulta optimizado

1. Considerando las siguientes tablas (los campos subrayados corresponden a llaves primarias):

Courses (ID, Name, Room, Time)

Exercises (ID, C_ID, A_ID, Room, Time)

Assistants (ID, Firstname, Lastname)

*En Exercises: C_ID es el ID de courses y A_ID es el ID de Assistants

Consulta:

SELECT C. Name, A. Firstname, A. Lastname, E. Room, E. Time

FROM Courses C, Assistants A, Exercises E

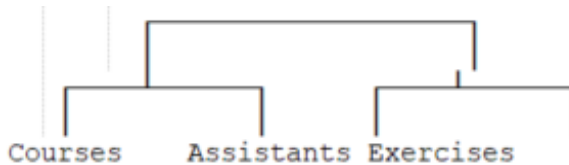
WHERE C.ID = E.C_ID AND A. ID=E.A_ID AND C. Room like '10%' AND E. Room not like 'CAB%'

$\pi_{C. Name, A. Firstname, A. Lastname, E. Room, E. Time} (\sigma_{C.ID = E.C_ID \text{ AND } A. ID=E.A_ID \text{ AND } C.Room \text{ like '10\%' AND } E. Room \text{ not like 'CAB\%'}} ((\text{Courses } C \times \text{Assistants } A) \times \text{Exercises } E))$

Solución II. Optimización de consultas:

1.

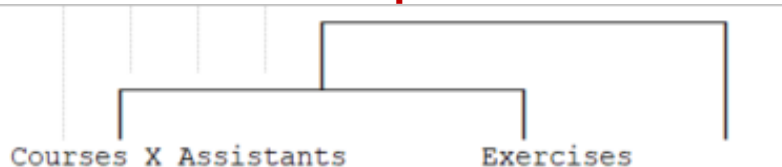
Árbol de consulta inicial:



Justificación del orden de las tablas:

- **Inicio con Courses:** Se inicia con Courses ya que la condición ' $C.ID = E.C_ID$ ' involucra la tabla Courses.
- **Join con Assistants:** Se realiza el join con Assistants utilizando las llaves foráneas que conectan Courses con Assistants ($A.ID = E.A_ID$).
- **Join final con Exercises:** Finalmente, se realiza el join con Exercises utilizando las llaves foráneas que conectan Courses con Exercises ($C.ID = E.C_ID$).

Árbol de consulta optimizado:



Justificación de la optimización:

- **Reordenamiento de las tablas:** En la versión optimizada, se reordenaron las tablas para reflejar el orden lógico de las condiciones en la cláusula WHERE. Se inicia con Courses ya que es la tabla principal y luego se unen Assistants y Exercises según las condiciones de igualdad.
- **Eliminación de redundancias:** Se eliminaron redundancias innecesarias en el árbol para simplificar la estructura y mejorar la legibilidad.
- **Uso de notación sigma (σ) y pi (π):** La notación sigma (σ) y pi (π) se utiliza para indicar las operaciones de selección y proyección respectivamente, proporcionando una representación más clara de las operaciones realizadas en la consulta.

2. Considerando las siguientes tablas (los campos subrayados corresponden a llaves primarias):

Applicants (id, mid, name, city, sid)
 Schools (sid, sname, srnk)
 Major (mid, mname, age)

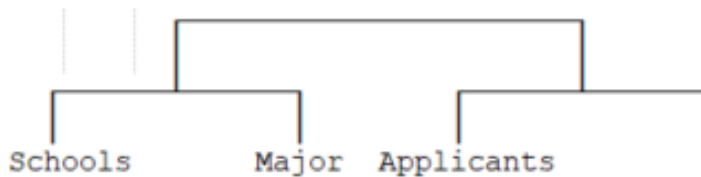
Consulta:

Π A.name (σ A.sid = S.sid AND A.mid=M.mid AND A.city='Seattle' AND S.rank <10 AND mname='CSE' ((School x Major) x Applicants))

Solución II. Optimización de consultas:

2.

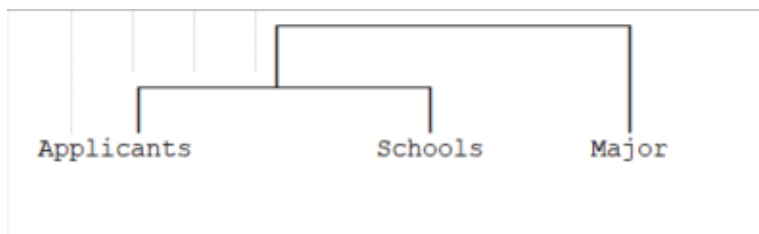
Árbol de consulta inicial:



Justificación del orden de las tablas:

- **Inicio con Schools:** Se inicia con Schools ya que la condición '**A.sid = S.sid**' involucra la tabla Schools.
- **Join con Major:** Se realiza el join con Major utilizando las llaves foráneas que conectan Schools con Major ('**A.mid=M.mid**').
- **Join final con Applicants:** Finalmente, se realiza el join con Applicants utilizando las llaves foráneas que conectan Major con Applicants ('**A.sid = S.sid**').

Árbol de consulta optimizado:



Justificación de la optimización:

- **Reordenamiento de las tablas:** En la versión optimizada, se reordenaron las tablas para reflejar el orden lógico de las condiciones en la cláusula WHERE. Se inicia con Applicants, ya que es la tabla principal, y luego se unen Schools y Major según las condiciones de igualdad.
- **Eliminación de redundancias:** Se eliminaron redundancias innecesarias en el árbol para simplificar la estructura y mejorar la legibilidad.
- **Uso de notación sigma (σ) y pi (π):** La notación sigma (σ) y pi (π) se utilizan para indicar las operaciones de selección y proyección respectivamente, proporcionando una representación más clara de las operaciones realizadas en la consulta SQL.