**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

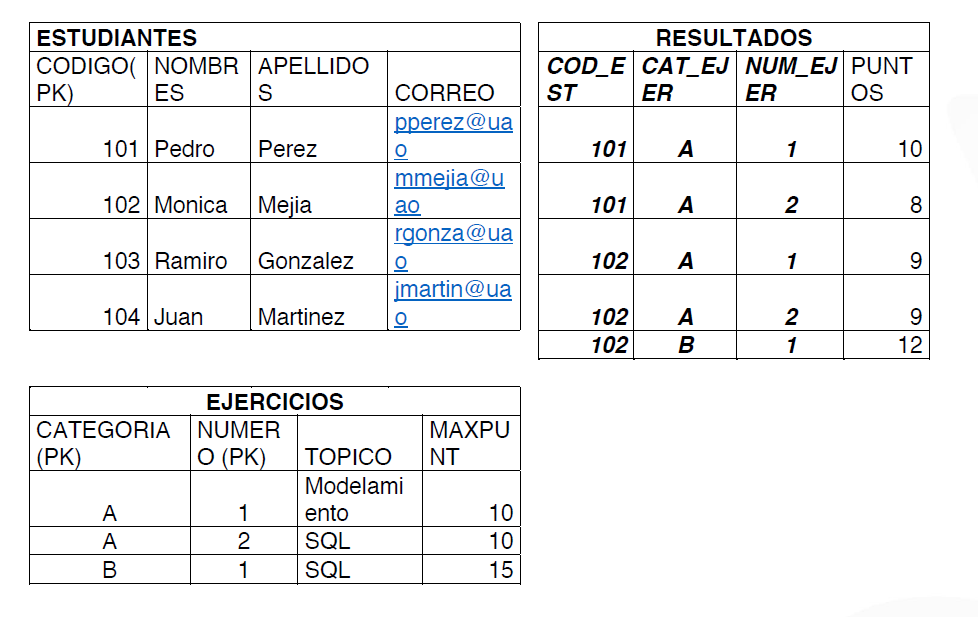
abajo.

1. Nombres y apellidos de los estudiantes que tienen resultados del ejercicio

A – 1, con la cantidad de puntos que obtuvieron.

1. Correos de los estudiantes que tienen resultados para ejercicios del tópico

SQL.



**Solución:**

**1.**

**a):**

|  |  |
| --- | --- |
| 1  2  3  4 | **SELECT** ESTUDIANTES.NOMBRES, ESTUDIANTES.APELLIDOS, RESULTADOS.PUNTOS  **FROM** ESTUDIANTES  **JOIN** RESULTADOS **ON** ESTUDIANTES.CODIGO = RESULTADOS.COD\_EST  **WHERE** RESULTADOS.CAT\_EJER = 'A' **AND** RESULTADOS.NUM\_EJER = 1; |

**Álgebra Relacional:**

|  |  |
| --- | --- |
| 1 | π\_NOMBRES, APELLIDOS, PUNTOS (σ\_CAT\_EJER='A' **AND** NUM\_EJER=1 (ESTUDIANTES ⨝ COD\_EST = CODIGO RESULTADOS)) |

**b):**

|  |  |
| --- | --- |
| 1  2  3  4  5 | **SELECT** ESTUDIANTES.CORREO  **FROM** ESTUDIANTES  **JOIN** RESULTADOS **ON** ESTUDIANTES.CODIGO = RESULTADOS.COD\_EST  **JOIN** EJERCICIOS **ON** RESULTADOS.CAT\_EJ = EJERCICIOS.CATEGORIA **AND** RESULTADOS.NUM\_EJER = EJERCICIOS.NUMERO  **WHERE** EJERCICIOS.TOPICO = 'SQL'; |

**Álgebra Relacional:**

|  |  |
| --- | --- |
| 1 | π\_CORREO (σ\_TOPICO='SQL' (ESTUDIANTES ⨝ COD\_EST = CODIGO (RESULTADOS ⨝ 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 Algebra Relacional:**

AEROPUERTOS (aeroId, aeronombre, ciudad)

VUELOS (vueloId, 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, vueloId

FROM SILLAS, VUELOS inner join AEROPUERTOS on aeroorigen = aeroId, RESERVAS

WHERE nacionalidad = ‘Colombia’ and compañia = ‘Avianca’ and ciduad = ‘Miami’ or ciudad = ‘Los

Angeles’ and numvuelor = numvuelo and silla = numsilla and numvuelo = vueloId

**Solución:**

**2.**

**Álgebra Relacional:**

*πnombre*, *silla*, *vueloId* (*σnacionalidad*=′*Colombia*′∧*compan*~*ia*=′*Avianca*′∧*ciudad*=′*Miami*′∨*ciudad*=′*LosAngeles*′∧*numvuelor*=*numvuelo*∧*silla*=*numsilla*∧*numvuelo*=*vueloId*​ (*SILLAS*⋈*numvuelo*=*vueloId*​ (*aeroId*/ *aeroorigen*​ (***VUELOS***)⋈*aeroorigen*=*aeroId*​ ***AEROPUERTOS)*** ⋈*numvuelor*=*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, titulo, duración, presupuesto)

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

πnit, nombre, nomRepuesto, precio (σSnit = nit AND título = ‘CRM’ AND fecha > ‘01/06/2019’ AND Snumero

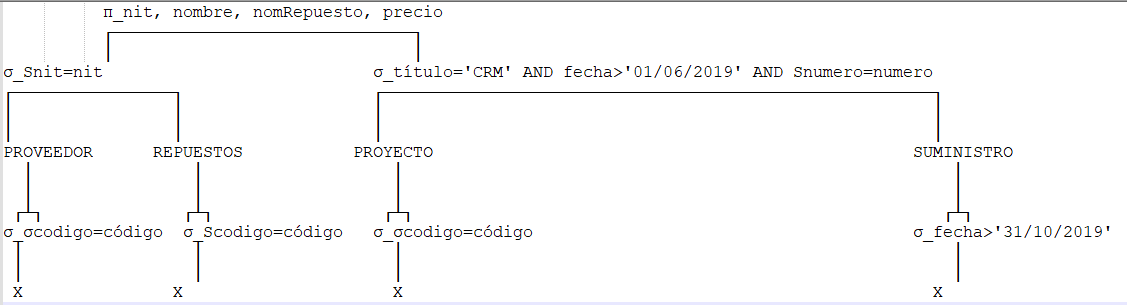
= número AND precio > 100.000 AND fecha < ‘31/10/2019’ AND Scodigo = código (**PROVEEDOR X**

**REPUESTOS X PROYECTO X SUMINISTRO))**

**Solución:**

**3.**

**Árbol de consulta equivalente:**

****

**4. Presente el árbol de consulta equivalente a la siguiente expresión de algebra**

**relacional:**

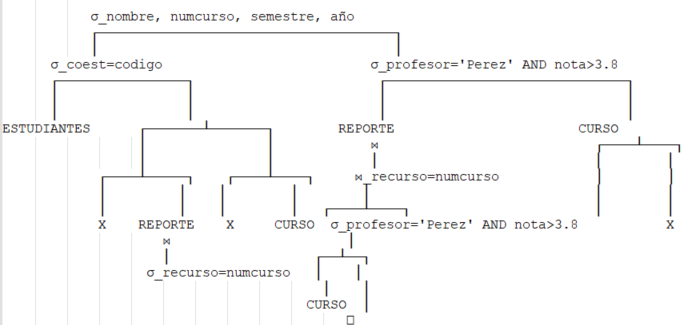
***σ*** nombre, numcurso, semestre, año **(*σ***coest = codigo **(ESTUDIANTES X (REPORTE**

**⋈**recurso=numcurso **(*σ*** profesor=’Perez’ and nota > 3.8 **(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 Excercises: 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%’

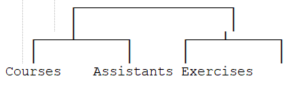
π C. Name, A. Firstname, A. Lastname, E. Room, E. Time (σ C.ID = E.C\_ID AND A. ID=E.A\_ID AND

C.Room like ‘10%’ AND E. Room not like ‘CAB%’ ((Courses C X Assistants A) X Exercises E))

**Solución II. Optimización de consultas:**

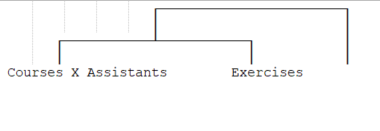
**1.**

**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, srank)

Major (mid, mname, age)

*Consulta:*

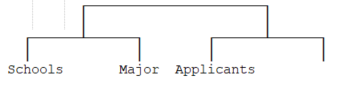
π A.name ( σ A.sid = S.sid AND A.mid=M.mid AND A.city=’Seattle’ AND S.rank

<10 AND mname=’CSE’ ((School x Major) x Applicants))

**Solución II. Optimización de consultas:**

**2.**

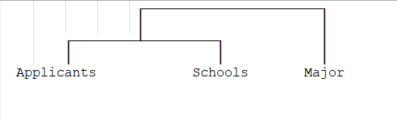
**Árbol de consulta inicial:**

****

**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.