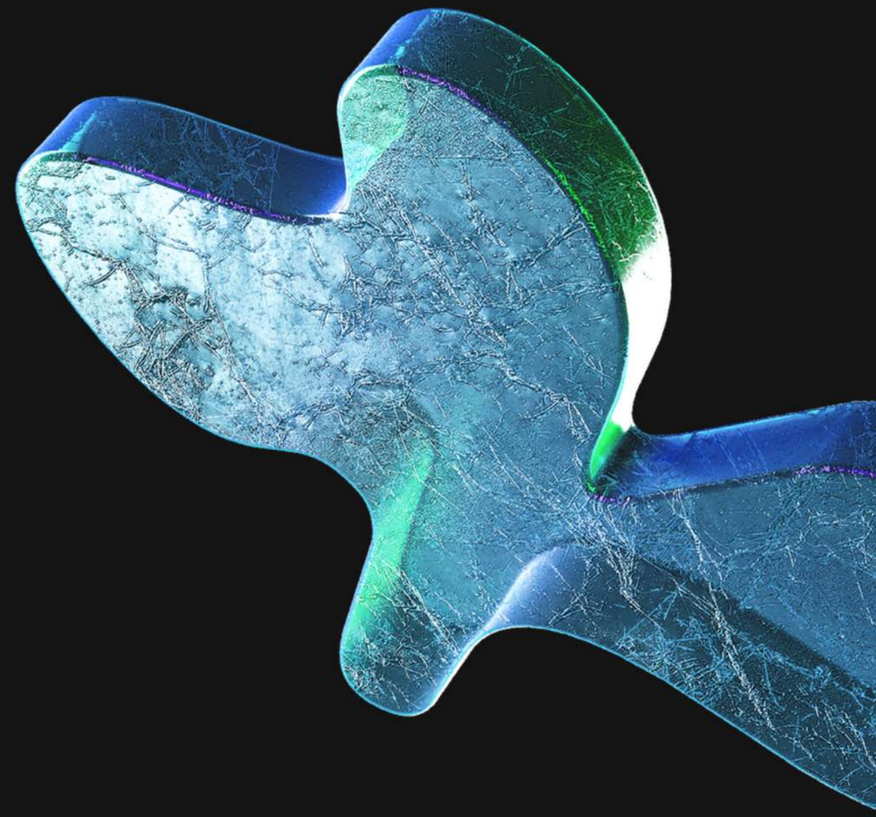
células plenum by Plenumsoft

Plenumsoft ACADEMY

Consultas de SQL

Práctica Laboratorio



Orlando Ezequiel Salazar Cruz

PostgreSQL

células plenum by Plenumsoft

Plenumsoft ACADEMY

Ejercicio UNO

Ejercicio de SELECT que relacione dos tablas por medio de clave primaria y clave foránea (No utilizar JOIN)

CONSULTA

```sql
SELECT a.nombre AS alumno, c.nombre_curso
FROM alumnos a, cursos c
WHERE a.id_curso = c.id_curso;
```

TABLA

	alumno character varying (50)	nombre_curso character varying (50)
1	Ezequiel	Matemáticas
2	María	Historia
3	José	Programación
4	Ana	Matemáticas
5	Luis	Biología
6	Sofía	Historia
7	Pedro	Programación
8	Lucía	Biología

células plenum by Plenumsoft

Plenumsoft ACADEMY

Ejercicio de SELECT con BETWEEN

Lista alumnos cuya edad está entre 20 y 22.
BETWEEN incluye los extremos (20 y 22).

CONSULTA

```
SELECT nombre, edad
FROM alumnos
WHERE edad BETWEEN 20 AND 22;
```

TABLA

	nombre character varying (50)	edad integer
1	Ezequiel	20
2	José	21
3	Ana	22
4	Luis	20
5	Pedro	21

células plenum by Plenumsoft

Plenumsoft ACADEMY

Ejercicio de SELECT con LIKE

Muestra alumnos cuyos nombres empiezan con "A".
LIKE 'A%' significa que debe iniciar con "A".

CONSULTA

```
SELECT nombre
FROM alumnos
WHERE nombre LIKE 'A%';
```

TABLA

	nombre character varying (50)
1	Ana

células plenum by Plenumsoft

Plenumsoft ACADEMY

Ejercicio de SELECT con IN

Trae alumnos cuya edad sea 19 o 23.
IN actúa como un filtro múltiple.

CONSULTA

```
SELECT nombre, edad
FROM alumnos
WHERE edad IN (19, 23);
```

TABLA

	nombre character varying (50)	edad integer
1	María	19
2	Sofía	23
3	Lucía	19

células plenum by Plenumsoft

Plenumsoft ACADEMY

Ejercicio de SELECT con MAX

Devuelve la edad más alta de todos los alumnos.
Es una función de agregación.

CONSULTA

```
SELECT MAX(edad) AS mayor_edad
FROM alumnos;
```

TABLA

	mayor_edad integer
1	23

células plenum by Plenumsoft

Plenumsoft ACADEMY

Ejercicio de SELECT con HAVING

Calcula el promedio de edad por curso y filtra los que superan 20.
HAVING se usa con funciones agregadas, no WHERE.

CONSULTA

```
SELECT c.nombre_curso, AVG(a.edad) AS promedio
FROM alumnos a
JOIN cursos c ON a.id_curso = c.id_curso
GROUP BY c.nombre_curso
HAVING AVG(a.edad) > 20;
```

TABLA

	nombre_curso character varying (50)	promedio numeric
1	Historia	21.0000000000000000
2	Programación	21.0000000000000000
3	Matemáticas	21.0000000000000000

células plenum by Plenumsoft

Plenumsoft ACADEMY

Ejercicio de INNER JOIN

Muestra alumnos con su curso, solo si hay coincidencia.
Excluye los cursos sin alumnos.

CONSULTA

```
SELECT a.nombre, c.nombre_curso
FROM alumnos a
INNER JOIN cursos c ON a.id_curso = c.id_curso;
```

TABLA

	nombre character varying (50)	nombre_curso character varying (50)
1	Ezequiel	Matemáticas
2	María	Historia
3	José	Programación
4	Ana	Matemáticas
5	Luis	Biología
6	Sofía	Historia
7	Pedro	Programación
8	Lucía	Biología

Ejercicio de LEFT JOIN

La idea será mostrar todos los alumnos aunque no tengan exámenes registrados.

CONSULTA

```sql
SELECT a.id_alumno, a.nombre AS alumno, e.materia, e.calificacion
FROM alumnos a
LEFT JOIN examenes e ON a.id_alumno = e.id_alumno
WHERE e.id_examen IS NULL;
```

TABLA

	id_alumno integer	alumno character varying (50)	materia character varying (50)	calificacion integer
1	8	Lucía	[null]	[null]

células plenum by Plenumsoft

Plenumsoft ACADEMY

Ejercicio de SELECT con MAX

Devuelve la edad más alta de todos los alumnos.
Es una función de agregación.

CONSULTA

```
SELECT MAX(edad) AS mayor_edad
FROM alumnos;
```

TABLA

	mayor_edad integer
1	23

células plenum by Plenumsoft

Plenumsoft ACADEMY

Ejercicio de RIGHT JOIN

Para listar todos los cursos y sus alumnos, mostrando NULL si el curso no tiene alumnos.

CONSULTA

```
SELECT a.nombre, c.nombre_curso
FROM alumnos a
RIGHT JOIN cursos c ON a.id_curso = c.id_curso
```

TABLA

	nombre character varying (50)	nombre_curso character varying (50)
1	Ezequiel	Matemáticas
2	María	Historia
3	José	Programación
4	Ana	Matemáticas
5	Luis	Biología
6	Sofía	Historia
7	Pedro	Programación

Ejercicio de Subconsulta

Muestra los alumnos inscritos en el curso "Programación".
La subconsulta obtiene el id_curso correspondiente.

CONSULTA

```sql
SELECT nombre
FROM alumnos
WHERE id_curso = (
    SELECT id_curso
    FROM cursos
    WHERE nombre_curso = 'Programación'
);
```

TABLA

	nombre character varying (50)
1	José
2	Pedro