INSTITUTO TECNOLOGICO DE MEXICALI

Carrera: ING. en Sistemas.

Materia: Fundamento de base de datos.

Alumno: Marin Salazar Juan Sebastian 22490423.

Correo Institucional: a22490423@itmexicali.edu.mx

Profesor: Jose Ramon Bogarin Valenzuela

> Fecha de entrega: 10 de Abril del 2025

```
Querys del profe
```

```
-- Crear tablas
CREATE TABLE estudiantes (
 id SERIAL PRIMARY KEY,
 nombre VARCHAR(100),
 email VARCHAR(100),
 fecha nacimiento DATE
);
CREATE TABLE cursos (
 id SERIAL PRIMARY KEY,
 nombre curso VARCHAR(100),
 duracion meses INT
);
CREATE TABLE matriculas (
 id SERIAL PRIMARY KEY,
 id estudiante INT REFERENCES estudiantes(id),
 id curso INT REFERENCES cursos(id),
 fecha matricula DATE
);
-- Insertar datos en estudiantes
INSERT INTO estudiantes (nombre, email, fecha nacimiento) VALUES
('Ana Torres', 'ana@example.com', '1998-03-12'),
('Luis Gómez', 'luis@example.com', '2000-07-22'),
('Carla Ruiz', 'carla@example.com', '1995-11-05');
-- Insertar datos en cursos
INSERT INTO cursos (nombre curso, duracion meses) VALUES
('Bases de Datos', 4),
('Programación Web', 6);
-- Insertar datos en matriculas
INSERT INTO matriculas (id estudiante, id curso, fecha matricula) VALUES
(1, 1, '2025-01-10'),
(2, 1, '2025-01-12'),
(3, 2, '2025-02-05'),
(1, 2, '2025-02-10');
```

Problema a resolver: "Analítica y Gestión Académica"

Contexto

Una institución educativa quiere aprovechar su sistema de base de datos para obtener información útil sobre sus estudiantes, los cursos ofrecidos y las matrículas realizadas. Como analista de datos, se te solicita realizar una serie de tareas para mejorar la toma de decisiones académicas.

Parte 1: Verificación y Ajustes de Estructura (LDD)

• Verifica si la base de datos contiene una columna para almacenar el número de teléfono de los estudiantes. Si no existe, agrégala a la tabla estudiantes.

R= ALTER TABLE Estudiantes ADD Telefono VARCHAR(14);

• Modifica la tabla cursos para que el nombre del curso no pueda repetirse.

R= ALTER TABLE Cursos ADD UNIQUE(nombre_curso);

Parte 2: Carga y Ajuste de Datos (LMD)

• Actualiza el email de "Luis Gómez" a luisgomez@universidad.edu.

R=

UPDATE Estudiantes SET email = 'luisgomez@universidad.edu' WHERE nombre = 'Luis Gómez';

• Registra una nueva matrícula para "Carla Ruiz" en el curso "Bases de Datos", con fecha 2025-04-01.

R=

INSERT INTO matriculas (id_estudiante, id_curso, fecha_matricula) VALUES (3, 1, '2025-04-01');

• Elimina la matrícula de "Ana Torres" del curso "Bases de Datos".

R= DELETE FROM matriculas M WHERE M.id = 1;

Parte 3: Consultas Avanzadas (CLE)

• Muestra un listado con el nombre de cada estudiante, el nombre del curso al que está matriculado y la fecha de matrícula.

R=

SELECT E.nombre, C.nombre_curso, M.fecha_matricula FROM estudiantes E INNER JOIN matriculas M ON E.id = M.id_estudiante INNER JOIN cursos C ON M.id curso = C.id;

• Muestra cuántos cursos ha tomado cada estudiante.

R=

SELECT E.nombre, COUNT(M.id_estudiante) AS Numero_cursos FROM estudiantes E INNER JOIN matriculas M ON E.id = M.id_estudiante GROUP BY E.nombre;

• Calcula la edad actual de cada estudiante y ordénalos de mayor a menor edad.

R=

SELECT E.nombre, E.fecha_nacimiento ,
DATE_PART('year', AGE(CURRENT_DATE, E.fecha_nacimiento)) AS Edad
FROM estudiantes E INNER JOIN matriculas M ON E.id = M.id_estudiante
ORDER BY E.fecha_nacimiento ASC;

• Muestra qué curso tiene más estudiantes matriculados.

R=

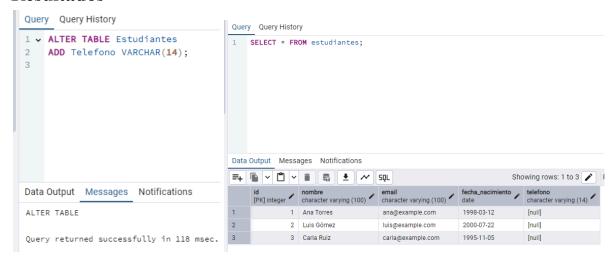
SELECT C.id, C.nombre_curso , COUNT(M.id_estudiante) AS total_estudiantes FROM estudiantes E INNER JOIN matriculas M ON E.id = M.id_estudiante INNER JOIN cursos C ON M.id_curso = C.id GROUP BY C.id, C.nombre_curso ORDER BY total_estudiantes DESC LIMIT 1;

• Calcula el porcentaje de estudiantes matriculados respecto al total de estudiantes para cada curso.

R=

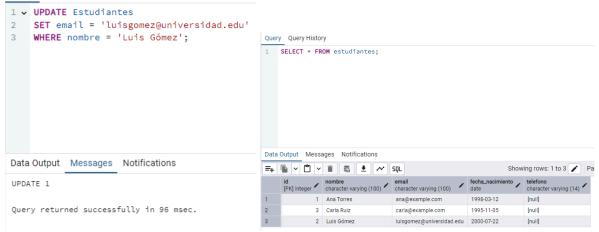
SELECT C.nombre_curso AS curso, COUNT(DISTINCT E.id) AS inscritos, ROUND(COUNT(DISTINCT E.id) * 100.0 / (SELECT COUNT(*) FROM estudiantes), 2) AS porcentaje FROM estudiantes E INNER JOIN matriculas M ON E.id = M.id_estudiante INNER JOIN cursos C ON M.id_curso = C.id GROUP BY C.nombre_curso;

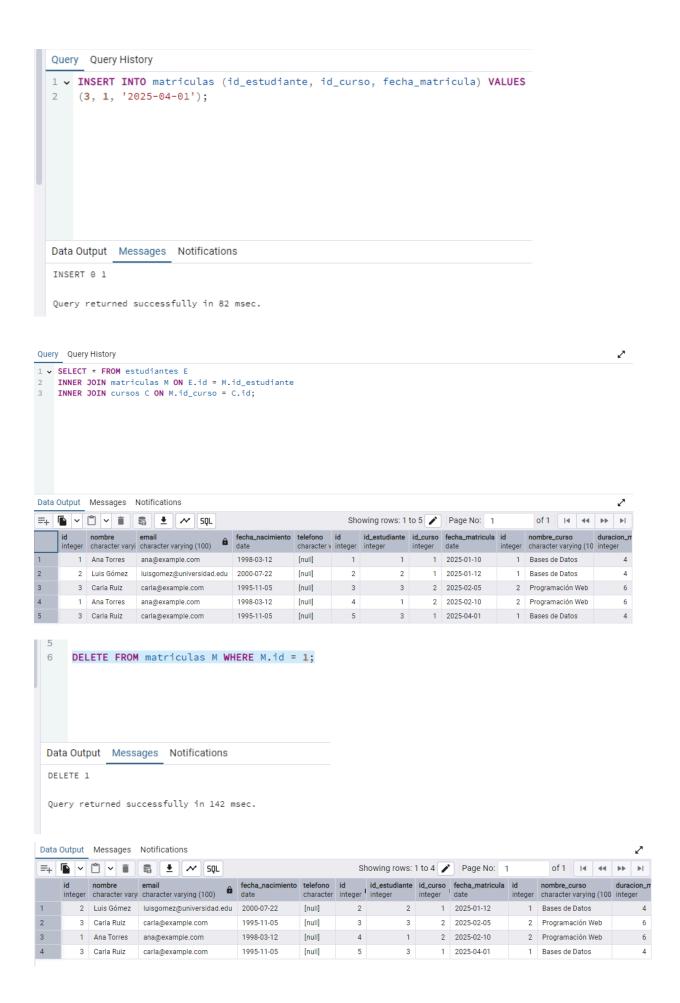
Resultados

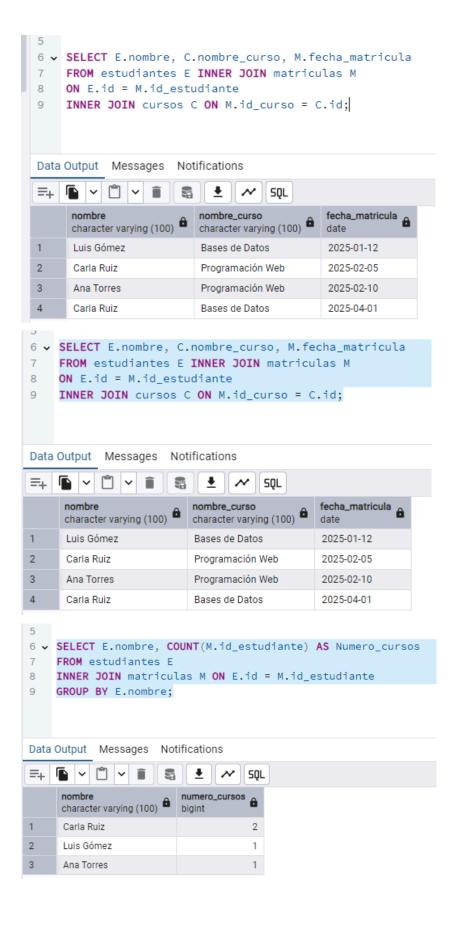












```
6 v SELECT E.nombre, E.fecha_nacimiento ,
     DATE_PART('year', AGE(CURRENT_DATE, E.fecha_nacimiento)) AS Edad
7
8
     FROM estudiantes E
     INNER JOIN matriculas M ON E.id = M.id_estudiante
9
10
     ORDER BY E.fecha_nacimiento ASC;
Data Output Messages Notifications
=+
                                                                           Showing
                        S
                             #
                                       SQL.
                           fecha_nacimiento
                                             edad
      nombre
                                             double precision
     character varying (100)
     Carla Ruiz
                            1995-11-05
                                                         29
2
      Carla Ruiz
                           1995-11-05
                                                         29
3
                           1998-03-12
      Ana Torres
                                                         27
4
     Luis Gómez
                           2000-07-22
                                                         24
13 v SELECT C.id, C.nombre_curso , COUNT(M.id_estudiante) AS total_estudiantes
   FROM estudiantes E
   INNER JOIN matriculas M ON E.id = M.id_estudiante
15
16
   INNER JOIN cursos C ON M.id_curso = C.id
17 GROUP BY C.id, C.nombre_curso
    ORDER BY total_estudiantes DESC
18
19
     LIMIT 1;
Data Output Messages Notifications
                            <u>*</u>
    5QL
                                                                       Showing rows: 1 to 1
                                     total_estudiantes
                 nombre_curso
     [PK] integer
                 character varying (100)
                                     bigint
                  Bases de Datos
                                                   2
29 V SELECT C.nombre_curso AS curso, COUNT(DISTINCT E.id) AS inscritos,
    ROUND(COUNT(DISTINCT E.id) * 100.0 / (SELECT COUNT(*) FROM estudiantes), 2) AS porcentaje
30
31
    FROM estudiantes E
   INNER JOIN matriculas M ON E.id = M.id_estudiante
32
33
   INNER JOIN cursos C ON M.id_curso = C.id
    GROUP BY C.nombre_curso;
34
35
36
37
Data Output Messages Notifications
                                                                  Showing rows: 1 to 2 Page No: 1
=+ | • | • | • | • |
                     8
                          <u>+</u>

✓ SQL

                        inscritos porcentaje
     character varying (100)
1
     Bases de Datos
                               2
                                      66.67
     Programación Web
                               2
```