

INSTITUTO TECNOLOGICO DE MEXICO  
INGENIERO EN SISTEMAS COMPUTACIONALES

Armas Diaz Erick Hidekio    N.ctrol : 23490411

FUNDAMENTOS DE BASE DE DATOS

CONSULTAS Y SQL

Tarea 1 Unidad 3

Fecha: 10/04/25

-- 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');
```

-- Consultas CLE

-- Estudiantes matriculados en "Bases de Datos"

```
SELECT e.nombre  
FROM estudiantes e  
JOIN matriculas m ON e.id = m.id_estudiante  
JOIN cursos c ON c.id = m.id_curso  
WHERE c.nombre_curso = 'Bases de Datos';
```

-- Cursos con cantidad de estudiantes  
matriculados

```
SELECT c.nombre_curso, COUNT(m.id_estudiante)
AS total_estudiantes

FROM cursos c

LEFT JOIN matriculas m ON c.id = m.id_curso

GROUP BY c.nombre_curso;
```

-- Estudiantes mayores de 25 años

```
SELECT nombre, fecha_nacimiento,
       DATE_PART('year', AGE(fecha_nacimiento)) AS
edad

FROM estudiantes

WHERE DATE_PART('year', AGE(fecha_nacimiento))
> 25;
```

-- Edad promedio de los estudiantes

```
SELECT ROUND(AVG(DATE_PART('year',
AGE(fecha_nacimiento)))) AS edad_promedio

FROM estudiantes;
```

-- Estudiantes ordenados por fecha de nacimiento

SELECT nombre, fecha\_nacimiento

FROM estudiantes

ORDER BY fecha\_nacimiento ASC;

19:02

JOSE RAMON BOGARIN VALENZUELA



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)

1. Verifica si la base de datos contiene una columna para almacenar el número de

2. teléfono de los estudiantes. Si no existe, agrégala a la tabla estudiantes.

	id [PK] integer	nombre character varying (100)	email character varying (100)	fecha_nacimiento date	telefono character varying (50)
1	1	Ana Torres	ana@example.com	1998-03-12	[null]
2	2	Luis Gómez	luisgomez@universidad.edu	2000-07-22	[null]
3	3	Carla Ruiz	carla@example.com	1995-11-05	[null]

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

```
Query  Query History
1  ALTER TABLE cursos
2  ADD CONSTRAINT nombre_curso_unico UNIQUE (nombre_curso);
3
```



## Parte 2: Carga y Ajuste de Datos (LMD)

1. Actualiza el email de "Luis Gómez" a [luisgomez@universidad.edu](mailto:luisgomez@universidad.edu).

```
19  UPDATE estudiantes
20  SET email = 'luisgomez@universidad.edu'
21  WHERE nombre = 'Luis Gómez';
```

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

```
12
13  INSERT INTO matriculas (id_estudiante, id_curso, fecha_matricula)
14  SELECT e.id, c.id, '2025-04-01'
15  FROM estudiantes e
16  JOIN cursos c ON c.nombre_curso = 'Bases de Datos'
17  WHERE e.nombre = 'Carla Ruiz';
```

3	Ana Torres	Programación Web	2025-02-10
4	Carla Ruiz	Bases de Datos	2025-04-01

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

```
5 DELETE FROM matriculas
6 WHERE id_estudiante = (
7     SELECT id FROM estudiantes WHERE nombre = 'Ana Torres'
8 )
9 AND id_curso = (
10     SELECT id FROM cursos WHERE nombre_curso = 'Bases de Datos'
11 );
```



## Parte 3: Consultas Avanzadas (CLE)

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

```
18
19 SELECT e.nombre AS estudiante, c.nombre_curso, m.fecha_matricula
20 FROM matriculas m
21 JOIN estudiantes e ON e.id = m.id_estudiante
22 JOIN cursos c ON c.id = m.id_curso;
23
```

	estudiante character varying (100)	nombre_curso character varying (100)	fecha_matricula date
1	Luis Gómez	Bases de Datos	2025-01-12
2	Carla Ruiz	Programación Web	2025-02-05
3	Ana Torres	Programación Web	2025-02-10
4	Carla Ruiz	Bases de Datos	2025-04-01

### 2. Muestra cuántos cursos ha tomado cada estudiante.



24 ▼ SELECT e.nombre, COUNT(m.id\_curso) AS cursos\_tomados

25 FROM estudiantes e

26 LEFT JOIN matriculas m ON e.id = m.id\_estudiante

27 GROUP BY e.nombre;

28

Data Output

Messages

Notifications

≡

+

📄

▼

📋

▼

🗑️

📤

⬇️

📈

SQL

	nombre character varying (100) 🔒	cursos_tomados bigint 🔒
1	Carla Ruiz	2
2	Luis Gómez	1
3	Ana Torres	1

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

29 SELECT nombre,  
30 DATE\_PART('year', AGE(fecha\_nacimiento)) AS edad  
31 FROM estudiantes  
32 ORDER BY edad ASC;  
33  
34  
35  
36  
37

Data OutputMessagesNotifications

+

▼

▼

⬇

↗

SQL

	nombre character varying (100) 🔒	edad double precision 🔒
1	Luis Gómez	24
2	Ana Torres	27
3	Carla Ruiz	29

4. Muestra qué curso tiene más estudiantes matriculados.

```
34 ✓ SELECT c.nombre_curso, COUNT(m.id_estudiante) AS total_matriculados
35 FROM cursos c
36 JOIN matriculas m ON c.id = m.id_curso
37 GROUP BY c.nombre_curso
38 ORDER BY total_matriculados DESC
39 LIMIT 1;
40
```

Data Output

Messages

Notifications

☰

+

📄

▼

📋

▼

🗑️

📤

⬇️

📈

SQL

	nombre_curso character varying (100) 🔒	total_matriculados bigint 🔒
1	Programación Web	2

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

```
23 SELECT c.nombre_curso,  
24        COUNT(DISTINCT m.id_estudiante) AS estudiantes_matriculados,  
25        ROUND(COUNT(DISTINCT m.id_estudiante) * 100.0 / (SELECT COUNT(*) FROM estudiantes), 2) AS porcentaje_matriculados  
26 FROM cursos c  
27 LEFT JOIN matriculas m ON c.id = m.id_curso  
28 GROUP BY c.nombre_curso;
```

Data Output			Messages	Notifications
	nombre_curso character varying (100)	estudiantes_matriculados bigint	porcentaje_matriculados numeric	
1	Bases de Datos	2	66.67	
2	Programación Web	2	66.67	