



INSTITUTO TECNOLÓGICO DE MEXICALI

Base De Datos

Tarea BD



ALUMNOS: FRANCISCO RAMOS VAZQUEZ

MAESTRO: JOSE RAMON BOGARIN VALENZUELA

CARRERA: INGENIERIA EN SISTEMAS COMPUTACIONALES

SEMESTRE: 4



Problema a resolver: "Análítica 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 teléfono de los estudiantes. Si no existe, agrégala a la tabla estudiantes.

No tiene la columna Telefono

	id	nombre	email	fecha_nacimiento
1	1	Ana Torres	ana@example.com	1998-03-12
2	2	Luis Gómez	luis@example.com	2000-07-22
3	3	Carla Ruiz	carla@example.com	1995-11-05

Entonces se agrega

```
ALTER TABLE estudiantes
ADD COLUMN telefono VARCHAR(20) UNIQUE ;
```

Y ya queda

	id	nombre	email	fecha_nacimiento	telefono
1	1	Ana Torres	ana@example.com	1998-03-12	<null>
2	2	Luis Gómez	luis@example.com	2000-07-22	<null>
3	3	Carla Ruiz	carla@example.com	1995-11-05	<null>

Solo faltaría rellenar la columna teléfono

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

```
-- Evitar cursos duplicados D|
ALTER TABLE cursos
ADD CONSTRAINT nombre_curso_unico UNIQUE (nombre_curso);
```



Parte 2: Carga y Ajuste de Datos (LMD)

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

Como se puede ver aquí aparece el antiguo correo de Luis

2	2 Luis Gómez	luis@example.com	2000-07-22	<null>
---	--------------	------------------	------------	--------

Ahora aquí esta actualizado

```
-- Actualizar el email de "Luis Gómez"
UPDATE estudiantes
SET email = 'luisgomez@universidad.edu'
WHERE nombre = 'Luis Gómez';
```

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

Aquí esta la matricula y su ID

3	3	3	2 2025-02-05	
---	---	---	--------------	--

```
-- Registrar nueva matrícula de "Carla Ruiz" en "Bases de Datos" (fecha: 2025-04-01)
INSERT INTO matriculas (id_estudiante, id_curso, fecha_matricula)
VALUES (
  (SELECT id_estudiante id FROM estudiantes WHERE nombre = 'Carla Ruiz'),
  (SELECT id_curso id FROM cursos WHERE nombre_curso = 'Bases de Datos'),
  fecha_matricula '2025-04-01'
);
```

Despues

9	9	3	1 2025-04-01	
---	---	---	--------------	--

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

```
-- Eliminar matrícula de "Ana Torres" en "Bases de Datos"
DELETE FROM matriculas
WHERE id_estudiante = (SELECT id FROM estudiantes WHERE nombre = 'Ana Torres')
AND id_curso = (SELECT id FROM cursos WHERE nombre_curso = 'Bases de Datos');
```



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.

```
SELECT e.nombre AS estudiante, c.nombre_curso, m.fecha_matricula
FROM matriculas m
JOIN estudiantes e 1..n<->1: ON m.id_estudiante = e.id
JOIN cursos c 1..n<->1: ON m.id_curso = c.id;
```

	estudiante ▼	nombre_curso ▼	fecha_matricula ▼
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	Luis Gómez	Bases de Datos	2025-01-12
5	Carla Ruiz	Programación Web	2025-02-05
6	Ana Torres	Programación Web	2025-02-10
7	Carla Ruiz	Bases de Datos	2025-04-01

2. Muestra cuántos cursos ha tomado cada estudiante.

```
SELECT e.nombre, COUNT(m.id_curso) AS cantidad_cursos
FROM estudiantes e
LEFT JOIN matriculas m ON e.id = m.id_estudiante
GROUP BY e.nombre;
```

	nombre	cantidad_cursos
1	Ana Torres	2
2	Carla Ruiz	3
3	Luis Gómez	2

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

```
SELECT nombre,
DATE_PART('year', AGE(fecha_nacimiento)) AS edad
FROM estudiantes
ORDER BY edad DESC;
```

	nombre	edad
1	Carla Ruiz	29
2	Ana Torres	27
3	Luis Gómez	24

4. Muestra qué curso tiene más estudiantes matriculados.

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

```
SELECT c.nombre_curso, COUNT(m.id_estudiante) AS total_estudiantes
FROM cursos c
JOIN matriculas m 1<->1..n: ON c.id = m.id_curso
GROUP BY c.nombre_curso
ORDER BY total_estudiantes DESC
LIMIT 1;
```

	nombre_curso	total_estudiantes
1	Programación Web	4

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

```
SELECT
  c.nombre_curso,
  COUNT(DISTINCT m.id_estudiante) AS estudiantes_matriculados,
  ROUND(COUNT(DISTINCT m.id_estudiante) * 100.0 / (SELECT COUNT(*) FROM estudiantes), 2) AS porcentaje_matriculados
FROM cursos c
LEFT JOIN matriculas m 1<->0..n: ON c.id = m.id_curso
GROUP BY c.nombre_curso;
```

	nombre_curso	estudiantes_matriculados	porcentaje_matriculados
1	Bases de Datos	2	66.67
2	Programación Web	2	66.67