

Instituto Tecnológico de Mexicali

Fundamentos de bases de datos

LDD, MDD Y CLE

Analitica y gestión academica

Luis Alonso Guevara Quiñonez - 23490377

- Docente -

Mtro. Jose Ramon Bogarin Valenzuela

```
-- 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);
```

```
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
SELECT * FROM
estudiantes:
SELECT * FROM
matriculas;
SELECT * FROM
cursos;
-- 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;
```

-- Insertar datos en matriculas

-- 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; --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. **ALTER TABLE estudiantes** ADD telefono INT; --Modifica la tabla cursos para que el nombre del curso no pueda repetirse. **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. 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. INSERT INTO matriculas (id_estudiante, id_curso, fecha_matricula) VALUES (3, 2, '2025-04-01');--Elimina la matrícula de "Ana Torres" del curso "Bases de Datos". **DELETE FROM matriculas** WHERE id = 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 nombre estudiante, c.nombre curso, m.fecha matricula

FROM matriculas m

INNER JOIN estudiantes e ON m.id estudiante = e.id

INNER JOIN cursos c ON m.id curso = c.id

-- Muestra cuántos cursos ha tomado cada estudiante.

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;

--Calcula la edad actual de cada estudiante y ordénalos de mayor a menor edad.6+

SELECT e.nombre, AGE(e.fecha nacimiento) AS edad

FROM estudiantes e

ORDER BY e.fecha_nacimiento ASC;

-- Muestra qué curso tiene más estudiantes matriculados.

SELECT c.nombre_curso, COUNT(m.id_estudiante) AS total_matriculados

FROM cursos c

JOIN matriculas m ON c.id = m.id_curso

GROUP BY c.nombre_curso

ORDER BY total matriculados DESC

LIMIT 1;

--5. Porcentaje de estudiantes matriculados respecto al total por curso:

SELECT c.nombre_curso,

 $ROUND(COUNT(DISTINCT\ m.id_estudiante), 2) * 100.0 / (SELECT\ COUNT(*)\ FROM\ estudiantes)\ AS\ porcentaje_matriculados$

FROM cursos c

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

GROUP BY c.nombre curso;

Parte 1: Verificacion 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 exisste, agrégara a la tabla estudiantes.

No existe una columna con el atributo teléfono en la tabla estudiantes:

```
1 -- Crear tablas
2 v CREATE TABLE estudiantes (
3    id SERIAL PRIMARY KEY,
4    nombre VARCHAR(100),
5    email VARCHAR(100),
6    fecha_nacimiento DATE
7 );
```

Se altera la tabla y se crea un atributo llamado teléfono:

```
78 \rightarrow ALTER TABLE estudiantes
79 ADD telefono INT;
```

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

Se crea la columna teléfono con valores null debido a que no se han insertado datos.

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

```
82  ALTER TABLE cursos
83  ADD UNIQUE (nombre_curso);
```

Se altera la columna nombre_curso para que su valor sea unico.

Parte 2: Carga y ajuste de datos (LMD)

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

```
87 • UPDATE estudiantes SET email = 'luisgomez@universidad.edu.'
88 WHERE nombre = 'Luis Gómez'
```

Actualizamos la columna email de la tabla "Estudiantes" a "luisgomez@universidad.edu."

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

—Registra una nueva matricula para "Carla Ruiz" en el curso "Base de datos", con fecha 2025-04-01.

```
91 INSERT INTO matriculas (id_estudiante, id_curso, fecha_matricula) VALUES
92 (3, 2, '2025-04-01');
```

-ESTA ERA LA FECHA ANTERIOR AGREGADA EN LA INSERCIÓN DE DATOS-

```
37 (3, 2, '2025-02-05'),
```

Cambiamos fecha con un insert.

-Elimina la matricula de "Ana Torres" del curso "Base de datos"

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

Como se ve ana torres contiene el id "1" por eso se manda a eliminar ese:

	id [PK] integer	id_estudiante integer	id_curso integer	fecha_matricula date
1	2	2	1	2025-01-12
2	3	3	2	2025-02-05
3	4	1	2	2025-02-10
4	5	3	2	2025-04-01

No existe id 1.

Parte 3.

Consultas avanzadas (CLE)

—Muestra un lsitado con el nombre de cada estudiante, el nombre del curso al que esta matriculado la fecha de la matricula.

```
SELECT e.nombre AS nombre_estudiante, c.nombre_curso, m.fecha_matricula

FROM matriculas m

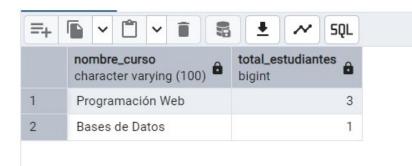
INNER JOIN estudiantes e ON m.id_estudiante = e.id

INNER JOIN cursos c ON m.id_curso = c.id
```

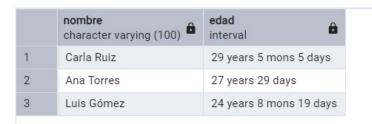
	nombre_estudiante character varying (100)	nombre_curso character varying (100)	fecha_matricula date
1	Ana Torres	Programación Web	2025-02-10
2	Carla Ruiz	Programación Web	2025-04-01
3	Carla Ruiz	Programación Web	2025-02-05
4	Luis Gómez	Bases de Datos	2025-01-12

-Muestra cuantos cursos ha tomado cada estudiante

```
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;
```



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



—Muestra qué curso tiene más estudiantes matriculados.

	nombre_curso character varying (100)	total_matriculados bigint
1	Programación Web	3

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

```
125 v SELECT c.nombre_curso,
126 ROUND(COUNT(DISTINCT m.id_estudiante),2) * 100.0 / (SELECT COUNT(*) FROM estudiantes) AS porcentaje_matriculados
127 FROM cursos c
128 LEFT JOIN matriculas m ON c.id = m.id_curso
129 GROUP BY c.nombre_curso;
```

	nombre_curso character varying (100)	porcentaje_matriculados numeric
1	Bases de Datos	33.3333333333333333
2	Programación Web	66.66666666666667