



Instituto Tecnológico de Mexicali

Ingeniería en sistemas computacionales

Bases de Datos

Tarea 1

Maestro: José Ramon Bogarin Valenzuela

Alumno: Génesis Edel Ortiz Pérez

Mexicali, BC a 10 de abril del 2025

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

```
ALTER TABLE estudiantes ADD COLUMN numero_telefono VARCHAR(20);
```

Consulta	Historial de Consultas	Data Output	Mensajes	Notificaciones
1				
2				
3				
4				
5				
6				
7				
8				

id	nombre	email	fecha_nacimiento	numero_telefono
1	Ana Torres	ana@example.com	1998-03-12	[null]
2	Carla Ruiz	carla@example.com	1995-11-05	[null]
3	Luis Gómez	luisgomez@universidad.edu	2000-07-22	[null]

--evitar nombre de cursos duplicados

```
ALTER TABLE cursos ADD CONSTRAINT nombre_curso_unico UNIQUE (nombre_curso);
```

```
INSERT INTO cursos (nombre_curso,duracion_meses)VALUES ('Bases de Datos',5);
```

Consulta	Historial de Consultas	Data Output	Mensajes	Notificaciones
1				
2				
3				
4				
5				
6				
7				

1	--evitar nombre duplicados			
2	ALTER TABLE cursos ADD CONSTRAINT nombre_curso_unico UNIQUE (nombre_curso);			
3	INSERT INTO cursos (nombre_curso,duracion_meses)VALUES ('Bases de Datos',5);			
4				
5				
6				
7				

ERROR: llave duplicada viola restricción de unicidad «nombre_curso_unico»
Ya existe la llave (nombre_curso)=(Bases de Datos).
Estado SQL: 23505
Detalle: Ya existe la llave (nombre_curso)=(Bases de Datos).



Parte 2: Carga y Ajuste de Datos (LMD)

1. Actualiza el email de "Luis Gómez" a `luisgomez@universidad.edu`.
2. Registra una nueva matrícula para "Carla Ruiz" en el curso "Bases de Datos", con fecha `2025-04-01`.
3. Elimina la matrícula de "Ana Torres" del curso "Bases de Datos".

--actualizar el email de luis gomez

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

Consulta	Historial de Consultas	Data Output	Mensajes	Notificaciones
1	--actualizar el email de luis gomez			
2	UPDATE estudiantes			
3	SET email ='luisgomez@universidad.edu'			
4	WHERE nombre LIKE 'Luis Gómez';			
5	select*from estudiantes;			
6				
7				

	id [PK] integer	nombre character varying (100)	email character varying (100)	fecha_nacimiento date	numero_telefono character varying (20)
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]

--registrar una nueva matricula para carla ruiz

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

Consulta	Historial de Consultas	Data Output	Mensajes	Notificaciones
1	--registrar una nueva matricula para carla ruiz			
2	INSERT INTO matriculas (id_estudiante,id_curso,fecha_matricula)			
3	SELECT e.id, c.id, DATE '2025-04-01'			
4	FROM estudiantes e			
5	INNER JOIN cursos c ON c.nombre_curso ='Bases de Datos'			
6	WHERE e.nombre='Carla Ruiz';			
7				
8				
9				
10				

	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	1	2025-04-01
5	6	3	1	2025-04-01

```
--eliminar matricula 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');
```

Consulta	Historial de Consultas	Data Output	Mensajes	Notificaciones
1	--eliminar matricula de ana torres en bases de datos			
2	DELETE FROM matriculas			
3	WHERE id_estudiante=(
4	SELECT id FROM estudiantes WHERE nombre ='Ana Torres'			
5)			
6	AND id_curso=(
7	SELECT id FROM cursos WHERE			
8	nombre_curso='Bases de Datos'			
9);			
10				

	id [PK] integer	id_estudiante integer	id_curso integer	fecha_matricula date
1		2	2	2025-01-12
2		3	3	2025-02-05
3		4	1	2025-02-10
4		5	3	2025-04-01
5		6	3	2025-04-01



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.
2. Muestra cuántos cursos ha tomado cada estudiante.
3. Calcula la edad actual de cada estudiante y ordénalos de mayor a menor edad.
4. Muestra qué curso tiene más estudiantes matriculados.
5. Calcula el porcentaje de estudiantes matriculados respecto al total de estudiantes para cada curso.
- 6.

```
--lista de estudiantes, sus cursos y fecha de matricula
SELECT e.nombre AS estudiante,
c.nombre_curso AS curso,
m.fecha_matricula
FROM estudiantes e
INNER JOIN matriculas m ON e.id=m.id_estudiante
INNER JOIN cursos c ON c.id=m.id_curso;
```

Consulta	Historial de Consultas	Data Output	Mensajes	Notificaciones
1	--lista de estudiantes, sus cursos y fecha de matricula			
2	SELECT e.nombre AS estudiante,			
3	c.nombre_curso AS curso,			
4	m.fecha_matricula			
5	FROM estudiantes e			
6	INNER JOIN matriculas m ON e.id=m.id_estudiante			
7	INNER JOIN cursos c ON c.id=m.id_curso;			
8				
9				
10				

	estudiante character varying (100)	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
5	Carla Ruiz	Bases de Datos	2025-04-01

```
--Numero de cursos tomados por 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;
```

Consulta Historial de Consultas

```
1 --Numero de cursos tomados por cada estudiante
2 SELECT e.nombre, COUNT(m.id_curso)AS
3 cantidad_cursos
4 FROM estudiantes e
5 LEFT JOIN matriculas m ON e.id=m.id_estudiante
6 GROUP BY e.nombre;
```



Data Output Mensajes Notificaciones

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

```
--mostrar que curso tiene mas estudiantes matriculados
SELECT c.nombre_curso,
COUNT (m.id_estudiante)AS total_matriculados
FROM cursos c
INNER JOIN matriculas m ON c.id=m.id_curso
GROUP BY c.nombre_curso
ORDER BY total_matriculados DESC
LIMIT 1;
```

Consulta Historial de Consultas

```
1 --mostrar que curso tiene mas estudiantes matriculados
2 SELECT c.nombre_curso,
3 COUNT (m.id_estudiante)AS total_matriculados
4 FROM cursos c
5 INNER JOIN matriculas m ON c.id=m.id_curso
6 GROUP BY c.nombre_curso
7 ORDER BY total_matriculados DESC
8 LIMIT 1;
```



Data Output Mensajes Notificaciones

	nombre_curso character varying (100)	total_matriculados bigint
1	Bases de Datos	3

```

--Edad actual de cada estudiante ordenados
--de mayor a menor
SELECT nombre,
DATE_PART ('YEAR',
AGE(fecha_nacimiento)) AS edad
FROM estudiantes
ORDER BY edad DESC;

```

Consulta		Historial de Consultas		Data Output		Mensajes	Notificaciones
1	--edad actual de cada estudiante ordenados						
2	--de mayor a menor						
3	SELECT nombre,						
4	DATE_PART ('YEAR',						
5	AGE(fecha_nacimiento)) AS edad						
6	FROM estudiantes						
7	ORDER BY edad DESC;						
8							

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