

Instituto Tecnológico de Mexicali



Ingeniería Sistemas Computacionales

Fundamento de Base de Datos

Tema:

Tarea 1 Unidad 3 Problema del día 9/04/2025

Alumno:

Delgado Rodriguez Jonhatan Eduardo

Profesor:

José Ramón Bogarin Valenzuela

Mexicali, B.C. a 10 de Marzo de 2025

alter table estudiantes

ADD column telefono ;

ALTER TABLE estudiantes

ALTER COLUMN telefono SET NOT NULL;

```
1  alter table estudiantes
2  ADD column telefono ;
3
4  ALTER TABLE estudiantes
5  ALTER COLUMN telefono SET NOT NULL;
```

Data Output Messages Notifications

	id [PK] integer	nombre character varying (100)	email character varying (100)	fecha_nacimiento date	telefono character varying (10)
1	1	Ana Torres	ana@example.com	1998-03-12	6861111111
2	3	Carla Ruiz	carla@example.com	1995-11-05	6863333333
3	2	Luis Gómez	luigomez@universidad.edu	2000-07-22	6861112222

ALTER TABLE cursos

add constraint nombre_curso

unique(nombre_curso);

```
7  ALTER TABLE cursos
8  add constraint nombre_curso
9  unique(nombre_curso);
10
```

Data Output Messages Notifications

	id [PK] integer	nombre_curso character varying (100)	duracion_meses integer
1	1	Bases de Datos	4
2	2	Programación Web	6

INSERT INTO cursos (nombre_curso, duracion_meses) VALUES

('Bases de Datos', 10);

```
15  INSERT INTO cursos (nombre_curso, duracion_meses) VALUES
16  ('Bases de Datos', 10);
17
```

Data Output Messages Notifications

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

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

--Update de correo LuisGómez--

update estudiantes

set email='luisgomez@universidad.edu'

where(nombre)='Luis Gómez';

```
20 update estudiantes
21 set email ='luisgomez@universidad.edu'
22 where(nombre)='Luis Gómez';
23
```

	id [PK] integer	nombre character varying (100)	email character varying (100)	fecha_nacimiento date	telefono character varying (10)
1	1	Ana Torres	ana@example.com	1998-03-12	6861111111
2	3	Carla Ruiz	carla@example.com	1995-11-05	6863333333
3	2	Luis Gómez	luisgomez@universidad.edu	2000-07-22	6861112222

--insert de una 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, 1, '2025-04-01');

```
24 --insert de una matrícula para "Carla Ruiz" en el curso “Bases de Datos”, con fecha 2025-04-01.--
25 INSERT INTO matriculas (id_estudiante, id_curso, fecha_matricula) VALUES
26 (3, 1, '2025-04-01');
27
```

	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

--Eliminacion de la matrícula de "Ana Torres" del curso “Bases de Datos”--

delete from matriculas

where (id) ='1';

```
28 --Eliminacion de la matrícula de "Ana Torres" del curso “Bases de Datos”--
29 delete from matriculas
30 where (id) ='1';
31
```

	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

--Parte 3--

--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 AS Nombre_Curso,  
m.fecha_matricula AS 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;
```

```
33 --Muestra un listado con el nombre de cada estudiante, el nombre del curso al que está matriculado y la fecha de matrícula.--  
34 SELECT e.nombre AS Nombre_estudiante, c.nombre_curso AS Nombre_Curso, m.fecha_matricula AS Fecha_Matricula FROM estudiantes e  
35 INNER JOIN matriculas m ON e.id = m.id_estudiante  
36 INNER JOIN cursos c ON m.id_curso = c.id;
```

	nombre_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

--Muestra cuántos cursos ha tomado cada estudiante.--

```
SELECT e.nombre AS Nombre_estudiante, COUNT(m.id_curso) AS Cursos_tomados FROM  
estudiantes e  
  
LEFT JOIN Matriculas m ON e.id = m.id_estudiante  
  
GROUP BY e.id, e.nombre  
  
ORDER BY Cursos_tomados DESC;
```

```
38 --Muestra cuántos cursos ha tomado cada estudiante.--  
39 SELECT e.nombre AS Nombre_estudiante, COUNT(m.id_curso) AS Cursos_tomados FROM estudiantes e  
40 LEFT JOIN Matriculas m ON e.id = m.id_estudiante  
41 GROUP BY e.id, e.nombre  
42 ORDER BY Cursos_tomados DESC;
```

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

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

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

```
43 --Calcula la edad actual de cada estudiante y ordénalos de mayor a menor edad.--
44 SELECT nombre AS Nombre_estudiante,DATE_PART('year', AGE(fecha_nacimiento)) AS Edad
45 FROM estudiantes
46 ORDER BY Edad DESC;
```

Data Output Messages Notifications

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

--Muestra qué curso tiene más estudiantes matriculados.--

```
SELECT c.nombre_curso AS Nombre_Curso, COUNT(m.id_estudiante) AS
Cantidad_estudiantes FROM cursos c
INNER JOIN matriculas m ON c.id = m.id_curso
GROUP BY c.id, c.nombre_curso
ORDER BY Cantidad_estudiantes DESC
LIMIT 1;
```

```
47 --Muestra qué curso tiene más estudiantes matriculados.--
48 SELECT c.nombre_curso AS Nombre_Curso, COUNT(m.id_estudiante) AS Cantidad_estudiantes FROM cursos c
49 INNER JOIN matriculas m ON c.id = m.id_curso
50 GROUP BY c.id, c.nombre_curso
51 ORDER BY Cantidad_estudiantes DESC
52 LIMIT 2;--puse de limite dos,porque ambos tienen la misma cantidad de alumnos inscritos
```

Data Output Messages Notifications

	nombre_curso character varying (100)	cantidad_estudiantes bigint
1	Bases de Datos	2
2	Programación Web	2

Showing rows: 1 to 2

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

```
SELECT c.nombre_curso AS Curso,ROUND(COUNT(m.id_estudiante) * 100.0 / COUNT(e.id),  
2) AS Porcentaje_matriculados
```

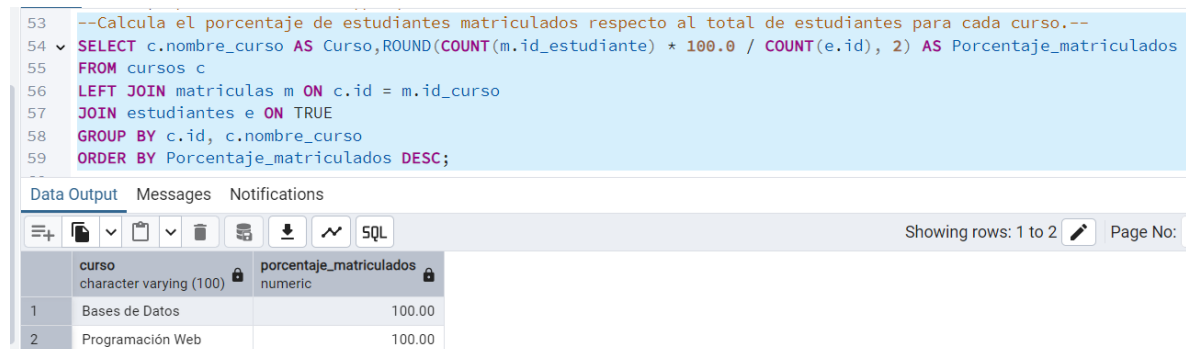
```
FROM cursos c
```

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

```
JOIN estudiantes e ON TRUE
```

```
GROUP BY c.id, c.nombre_curso
```

```
ORDER BY Porcentaje_matriculados DESC;
```



The screenshot shows a SQL IDE interface. The top pane displays a SQL query with line numbers 53 to 59. The query calculates the percentage of enrolled students for each course. The bottom pane shows the 'Data Output' tab with a table of results. The table has two columns: 'curso' (character varying (100)) and 'porcentaje_matriculados' (numeric). There are two rows of data: 'Bases de Datos' with a percentage of 100.00, and 'Programación Web' with a percentage of 100.00. The interface also includes a toolbar with icons for file operations and a status bar indicating 'Showing rows: 1 to 2' and 'Page No:'.

```
53 --Calcula el porcentaje de estudiantes matriculados respecto al total de estudiantes para cada curso.--  
54 SELECT c.nombre_curso AS Curso,ROUND(COUNT(m.id_estudiante) * 100.0 / COUNT(e.id), 2) AS Porcentaje_matriculados  
55 FROM cursos c  
56 LEFT JOIN matriculas m ON c.id = m.id_curso  
57 JOIN estudiantes e ON TRUE  
58 GROUP BY c.id, c.nombre_curso  
59 ORDER BY Porcentaje_matriculados DESC;
```

	curso character varying (100)	porcentaje_matriculados numeric
1	Bases de Datos	100.00
2	Programación Web	100.00