



Basado en el documento SQL realizar los siguientes enunciados y pegar el resultado y captura. Subir a GitHub en la carpeta correspondiente

1. Seleccionar todos los estudiantes

A screenshot of the MySQL Workbench interface. The left sidebar shows various management and performance metrics. The main area has a query editor window titled "SQL File 3" containing the SQL command "SELECT * FROM ESTUDIANTES;". Below the query is a result grid displaying six rows of student data from the "ESTUDIANTES" table. The columns are labeled id, nombre, apellido, fecha_nacimiento, and departamento_id.

	id	nombre	apellido	fecha_nacimiento	departamento_id
1	1	Juan	García	2000-05-15	1
2	2	Ana	Pérez	1999-08-20	3
3	3	Luis	García	2001-01-10	1
4	4	Maria	López	1998-12-25	2
5	5	Alberto	Ruiz	2000-03-30	1
6	6	Sofia	Mendez	2002-07-14	3

2. Listar solo los nombres y apellidos

The screenshot shows the MySQL Workbench interface. The SQL editor window contains the following query:

```
1 •   SELECT nombre, apellido FROM ESTUDIANTES;
```

The results grid displays the following data:

	nombre	apellido
▶	Juan	García
	Ana	Pérez
	Luis	García
	Maria	López
	Alberto	Ruiz
	Sofia	Mendez

3. Filtrar estudiantes de un departamento

The screenshot shows the MySQL Workbench interface. The SQL editor window contains the following query:

```
1 •   SELECT * FROM ESTUDIANTES where departamento_id=1;
```

The results grid displays the following data:

	id	nombre	apellido	fecha_nacimiento	departamento_id
▶	1	Juan	García	2000-05-15	1
	3	Luis	García	2001-01-10	1
	5	Alberto	Ruiz	2000-03-30	1
*	HULL	HULL	HULL	HULL	HULL

4. Ordenar estudiantes por fecha de nacimiento (más viejos primero)

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```
1 •    SELECT * FROM ESTUDIANTES order by fecha_nacimiento asc
```

The result grid displays the following data:

	id	nombre	apellido	fecha_nacimiento	departamento_id
▶	4	Maria	López	1998-12-25	2
	2	Ana	Pérez	1999-08-20	3
	5	Alberto	Ruiz	2000-03-30	1
	1	Juan	García	2000-05-15	1
	3	Luis	García	2001-01-10	1
	6	Sofia	Mendez	2002-07-14	3
*		NULL	NULL	NULL	NULL

5. Contar cuántos estudiantes hay

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```
1 •    SELECT COUNT(*) AS total_estudiantes FROM ESTUDIANTES;
```

The result grid displays the following data:

	total_estudiantes
▶	6

6. Buscar estudiantes con apellido 'García'

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, the SQL editor contains the following code:

```
1 SELECT * FROM ESTUDIANTES
2 WHERE apellido LIKE 'GARCIA';
```

Below the SQL editor is the Result Grid. It has columns labeled id, nombre, apellido, fecha_nacimiento, and departamento_id. The data shows two rows of results:

	id	nombre	apellido	fecha_nacimiento	departamento_id
▶	1	Juan	García	2000-05-15	1
	3	Luis	García	2001-01-10	1
*	NULL	NULL	NULL	NULL	NULL

7. Buscar por patrón (nombres que empiezan con 'A')

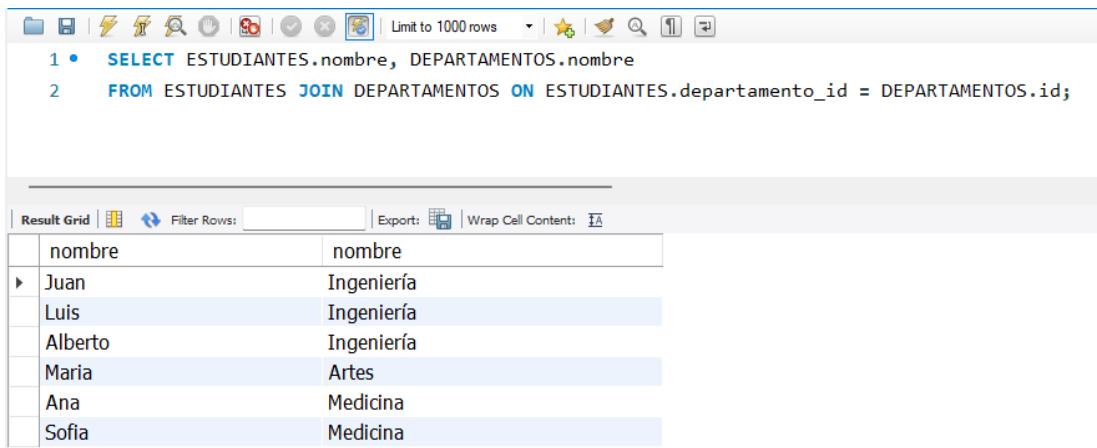
The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, the SQL editor contains the following code:

```
1 SELECT * FROM ESTUDIANTES
2 WHERE nombre LIKE 'A%';
```

Below the SQL editor is the Result Grid. It has columns labeled id, nombre, apellido, fecha_nacimiento, and departamento_id. The data shows two rows of results:

	id	nombre	apellido	fecha_nacimiento	departamento_id
▶	2	Ana	Pérez	1999-08-20	3
	5	Alberto	Ruiz	2000-03-30	1
*	NULL	NULL	NULL	NULL	NULL

8. Join: Mostrar nombre del estudiante y el nombre del departamento



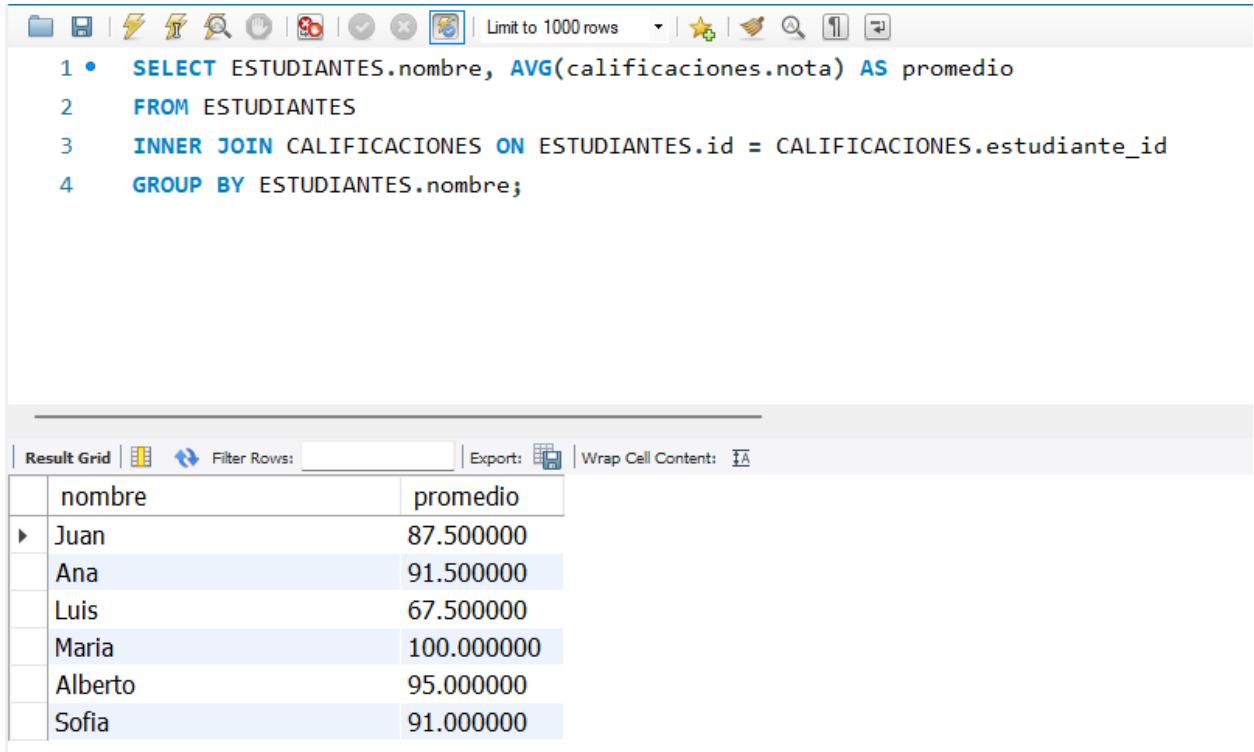
The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query is:

```
1 • SELECT ESTUDIANTES.nombre, DEPARTAMENTOS.nombre
2   FROM ESTUDIANTES JOIN DEPARTAMENTOS ON ESTUDIANTES.departamento_id = DEPARTAMENTOS.id;
```

The results grid displays the following data:

	nombre	nombre
▶	Juan	Ingeniería
	Luis	Ingeniería
	Alberto	Ingeniería
	Maria	Artes
	Ana	Medicina
	Sofia	Medicina

9. Promedio de calificaciones por estudiante
10. Cantidad de estudiantes por departamento



The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query consists of two parts:

```
1 • SELECT ESTUDIANTES.nombre, AVG(calificaciones.nota) AS promedio
2   FROM ESTUDIANTES
3   INNER JOIN CALIFICACIONES ON ESTUDIANTES.id = CALIFICACIONES.estudiante_id
4   GROUP BY ESTUDIANTES.nombre;
```



```
1 • SELECT COUNT(*) AS cantidad
2   FROM ESTUDIANTES
3   WHERE departamento_id = 1;
```

The results grid displays the average grades for each student:

	nombre	promedio
▶	Juan	87.500000
	Ana	91.500000
	Luis	67.500000
	Maria	100.000000
	Alberto	95.000000
	Sofia	91.000000

10. Cursos impartidos por cada profesor

```
1 •  SELECT PROFESORES.nombre, CURSOS.nombre
2   FROM PROFESORES
3   INNER JOIN CURSOS ON PROFESORES.id = CURSOS.id
4   ORDER BY PROFESORES.nombre ASC;
```

Result Grid		
	nombre	nombre
▶	Dr. House	Anatomía
	Ing. Stark	Robótica
	Prof. Picasso	Pintura

11. Estudiantes con promedio mayor a 90

```
1 •  SELECT ESTUDIANTES.nombre, AVG(calificaciones.nota) AS promedio
2   FROM ESTUDIANTES
3   INNER JOIN CALIFICACIONES ON ESTUDIANTES.id = Calificaciones.estudiante_id
4   GROUP BY ESTUDIANTES.nombre
5   having promedio > 90;
```

Result Grid		
	nombre	promedio
▶	Ana	91.500000
	Maria	100.000000
	Alberto	95.000000
	Sofia	91.000000

13. Top 5 estudiantes con mejores promedios

```
1 •  SELECT ESTUDIANTES.nombre, AVG(calificaciones.nota) AS promedio
2   FROM ESTUDIANTES
3   INNER JOIN CALIFICACIONES ON ESTUDIANTES.id = Calificaciones.estudiante_id
4   GROUP BY ESTUDIANTES.nombre order by promedio DESC
5   LIMIT 5;
6
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	nombre	promedio			
▶	Maria	100.000000			
	Alberto	95.000000			
	Ana	91.500000			
	Sofia	91.000000			
	Juan	87.500000			