



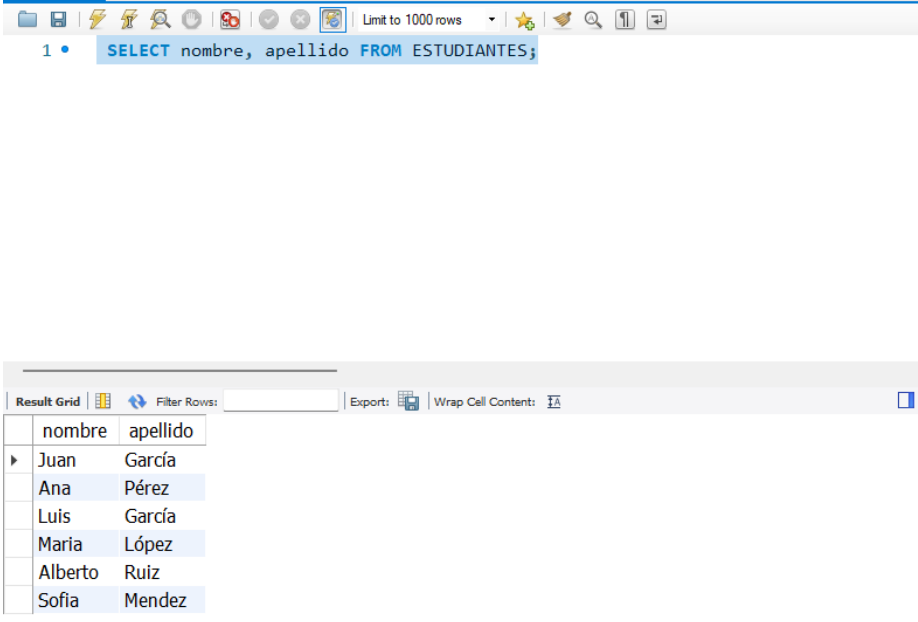
Basado en el documento SQL realizar los siguientes enunciados y pegar el resultado y captura. Subir a GitHub en la carpeta correspondiente

1. Seleccionar todos los estudiantes

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the query: `SELECT * FROM ESTUDIANTES;`. The Results window displays the following data:

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

2. Listar solo los nombres y apellidos



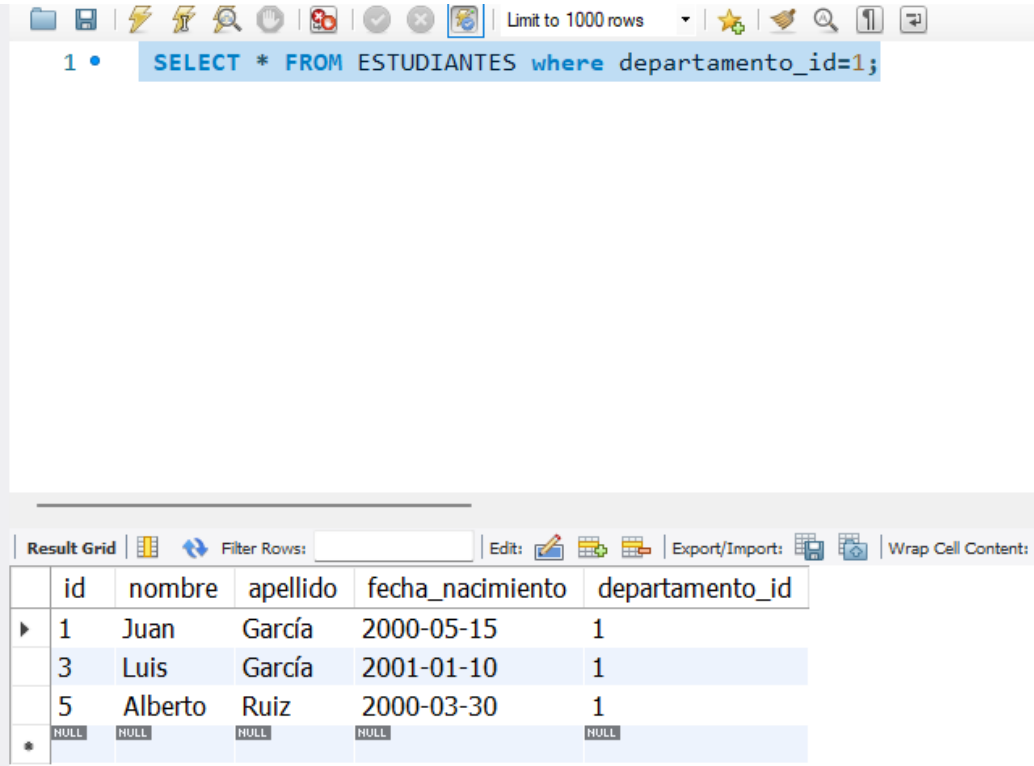
The screenshot shows a database query editor with a toolbar at the top. The SQL query entered is:

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

Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The grid has two columns: 'nombre' and 'apellido'. The results are as follows:

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

3. Filtrar estudiantes de un departamento



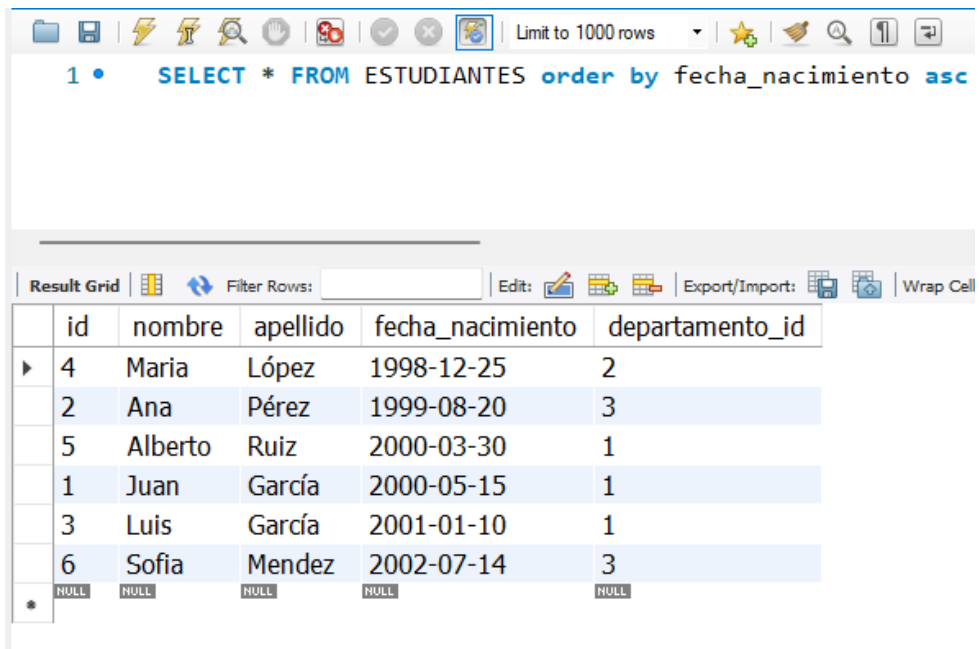
The screenshot shows a database query editor with a toolbar at the top. The SQL query entered is:

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

Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The grid has six columns: 'id', 'nombre', 'apellido', 'fecha_nacimiento', and 'departamento_id'. The results are as follows:

	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
*	NULL	NULL	NULL	NULL	NULL

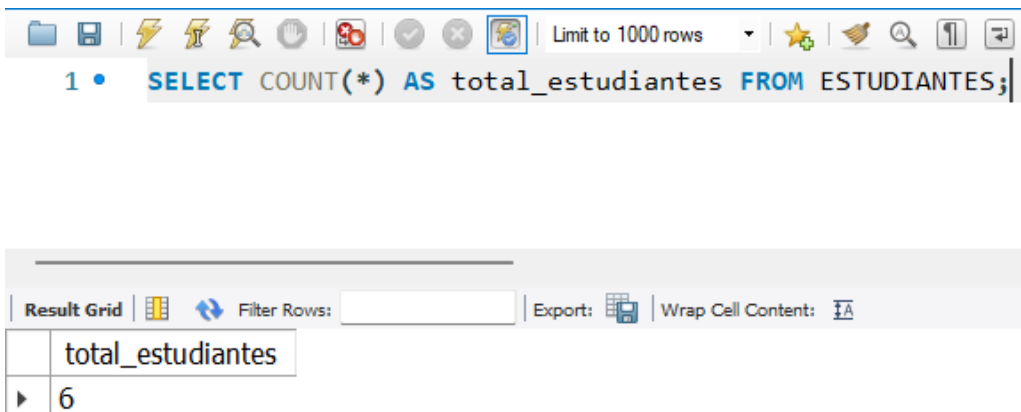
4. Ordenar estudiantes por fecha de nacimiento (más viejos primero)



The screenshot shows a database query editor with a toolbar at the top. The SQL query entered is: `1 • SELECT * FROM ESTUDIANTES order by fecha_nacimiento asc`. Below the query, the 'Result Grid' is displayed, showing a table with 6 rows and 6 columns: `id`, `nombre`, `apellido`, `fecha_nacimiento`, and `departamento_id`. The rows are sorted by `fecha_nacimiento` in ascending order.

	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	NULL

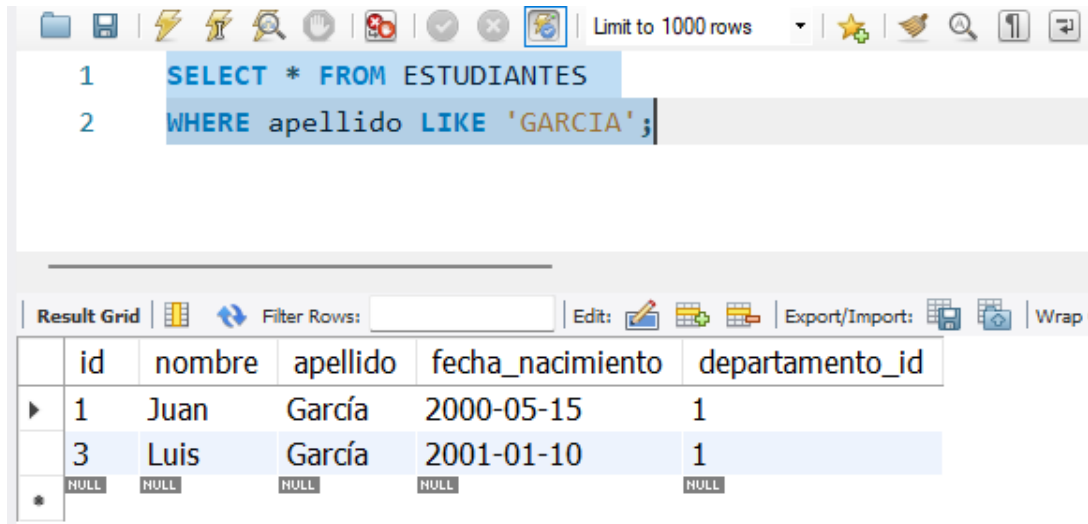
5. Contar cuántos estudiantes hay



The screenshot shows a database query editor with a toolbar at the top. The SQL query entered is: `1 • SELECT COUNT(*) AS total_estudiantes FROM ESTUDIANTES;`. Below the query, the 'Result Grid' is displayed, showing a single row with the column `total_estudiantes` and the value `6`.

	total_estudiantes
▶	6

6. Buscar estudiantes con apellido 'García'



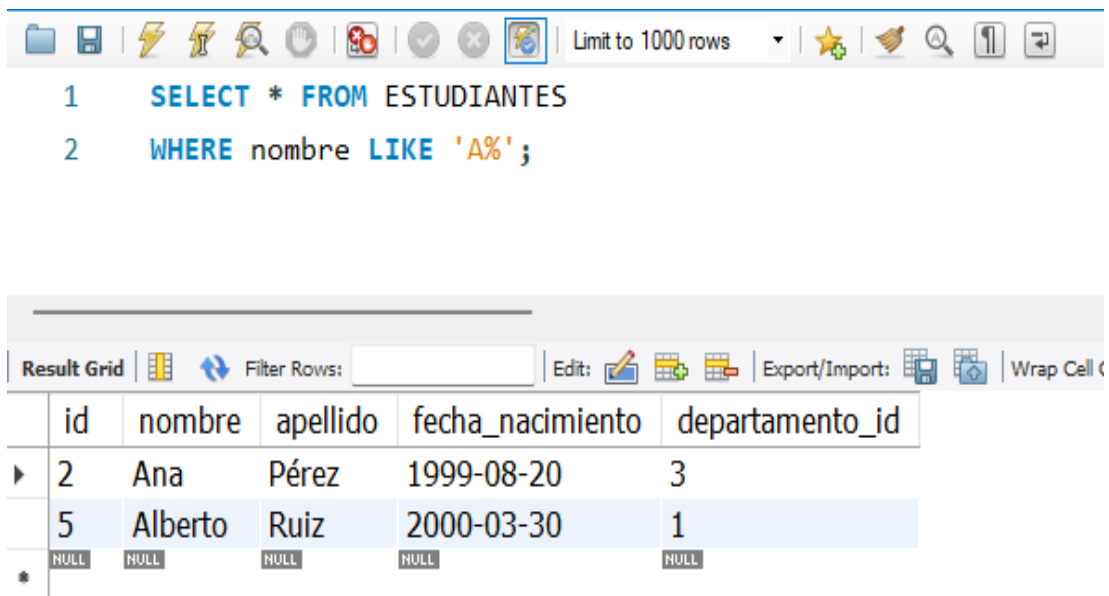
The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 1000 rows". Below the toolbar, a SQL query is entered in a text area:

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

Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Edit:" button, and an "Export/Import:" button. The result 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
*	NULL	NULL	NULL	NULL	NULL

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



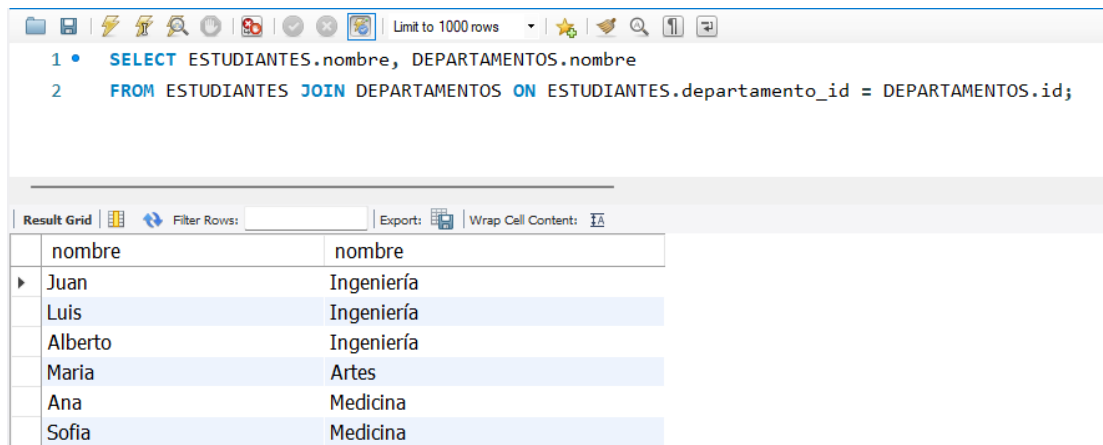
The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 1000 rows". Below the toolbar, a SQL query is entered in a text area:

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

Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Edit:" button, and an "Export/Import:" button. The result grid displays the following data:

	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



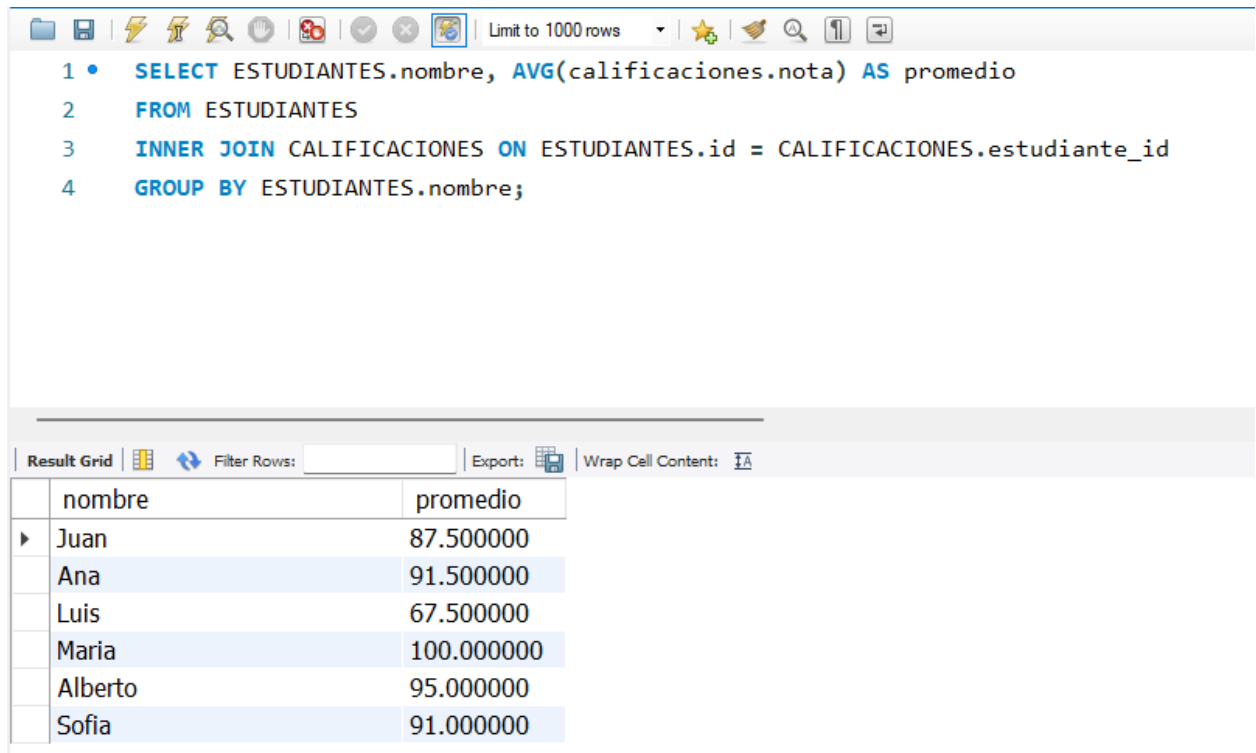
The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

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

Below the query, the 'Result Grid' is displayed with the following data:

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

9. Promedio de calificaciones por estudiante10. Cantidad de estudiantes por departamento



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
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;
```

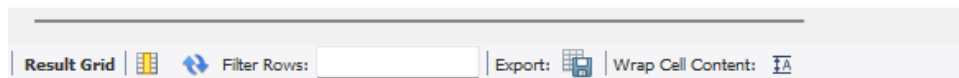
Below the query, the 'Result Grid' is displayed with the following data:

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

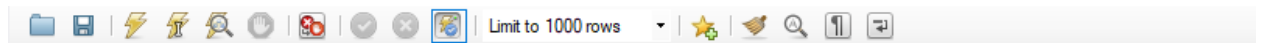


```
1 • SELECT PROFESORES.nombre, CURSOS.nombre
2 FROM PROFESORES
3 INNER JOIN CURSOS ON PROFESORES.id = CURSOS.id
4 ORDER BY PROFESORES.nombre ASC;
```

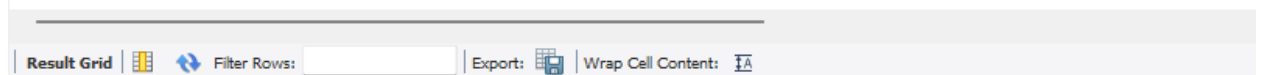


	nombre	nombre
▶	Dr. House	Anatomía
	Ing. Stark	Robótica
	Prof. Picasso	Pintura

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;
```



	nombre	promedio
▶	Ana	91.500000
	Maria	100.000000
	Alberto	95.000000
	Sofia	91.000000

13. Top 5 estudiantes con mejores promedios

Limit to 1000 rows

```
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