

1 • CREATE DATABASE GestionAcademica;

2 • USE GestionAcademica;

Output

Action Output

#	Time	Action	Message
✓ 1	22:53:43	CREATE DATABASE GestionAcademica	1 row(s) affected
✓ 2	22:53:43	USE GestionAcademica	0 row(s) affected

### Creación de Tablas



```
4 -- Tabla Departamento
5 • CREATE TABLE Departamento (
6     id_departamento INT AUTO_INCREMENT
7     nombre VARCHAR(100)
8 );
```

```
10  -- Tabla Estudiante
11  CREATE TABLE Estudiante (
12      id_estudiante INT AUTO_INCREMENT
13      nombre VARCHAR(100),
14      apellido VARCHAR(100),
15      correo VARCHAR(100),
16      id_departamento INT,
17      FOREIGN KEY (id_departamento)
18  );
19
```

Output

Action Output

#	Time	Action	Message
1	23:09:14	CREATE TABLE Estudiante ( id_estudiante INT AUTO_...	0 row(s) affected

```
19
20  -- Tabla Profesor
21  CREATE TABLE Profesor (
22      id_profesor INT AUTO_INCREMENT
23      nombre VARCHAR(100),
24      apellido VARCHAR(100),
25      correo VARCHAR(100),
26      id_departamento INT,
27      FOREIGN KEY (id_departamento)
28  );
29
```

Output

Action Output

#	Time	Action	Message
1	23:12:48	CREATE TABLE Profesor ( id_profesor INT AUTO_INC...	0 row(s) affected

```
30  -- Tabla Curso
31  CREATE TABLE Curso (
32      id_curso INT AUTO_INCREMENT P
33      nombre VARCHAR(100),
34      creditos INT,
35      id_departamento INT,
36      FOREIGN KEY (id_departamento)
37  );
38
```

Output

Action Output

#	Time	Action	Message
1	23:14:22	CREATE TABLE Curso ( id_curso INT AUTO_INCREM...	0 row(s) affected

```
39  -- Tabla Clase
40  CREATE TABLE Clase (
41      id_clase INT AUTO_INCREMENT P
42      id_curso INT,
43      id_profesor INT,
44      horario VARCHAR(100),
45      aula VARCHAR(50),
46      FOREIGN KEY (id_curso) REFERE
47      FOREIGN KEY (id_profesor) REF
48  );
49
```

Output

Action Output

#	Time	Action	Message
✓ 1	23:15:32	CREATE TABLE Clase ( id_clase INT AUTO_INCREME...	0 row(s) affected

```
61 -- Tabla Calificacion
62 CREATE TABLE Calificacion (
63     id_calificacion INT AUTO_INCR
64     id_inscripcion INT,
65     nota DECIMAL(5,2),
66     FOREIGN KEY (id_inscripcion)
67 );
```

Output

Action Output

#	Time	Action	Message
✓ 1	23:22:04	CREATE TABLE Calificacion ( id_calificacion INT AUT...	0 row(s) affected

## Insertar Registros

```
68
69 INSERT INTO Departamento (nombre)
70
71 INSERT INTO Estudiante (nombre, a
72 VALUES
73 ('Ana', 'Ramirez', 'ana.ramirez@e
```

Context Help Snippets

Output

Action Output

#	Time	Action	Message
✓ 1	23:32:13	INSERT INTO Departamento (nombre) VALUES ('Ciencias'...	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0

## Consulta básicas

### Estudiantes ordenados por apellido

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The query editor contains the following SQL statement:

```
101 • SELECT nombre, apellido
102 FROM Estudiante
103 ORDER BY apellido;
```

Below the query editor, the 'Result Grid' tab is active, displaying a table with two columns: 'nombre' and 'apellido'. The table is currently empty. To the right of the table are buttons for 'Filter Rows', 'Export', and 'Wrap Cell Content'. A 'Result Grid' button is also visible on the right side of the interface.

At the bottom, the 'Output' pane shows the 'Action Output' tab. It contains a table with the following data:

#	Time	Action	Message
1	23:42:07	SELECT nombre, apellido FROM Estudiante ORDER ...	0 row(s) returned

## 2. Cursos con más de 3 créditos

105 • `SELECT *`  
 106 `FROM Curso`  
 107 `WHERE credits > 3;`

Result Grid

	id_curso	nombre	credits	id_departamento
*	NULL	NULL	NULL	NULL

Curso 2 x Apply Context

Output

Action Output

#	Time	Action	Message
✓ 1	23:44:32	SELECT * FROM Curso WHERE credits > 3 LIMIT 0...	0 row(s) returned

3. Mostrar el nombre del estudiante y el nombre del curso en el que está inscrito (INNER JOIN).

110 `FROM Estudiante E`  
 111 `INNER JOIN Inscripcion I ON E.id_estudiante = I.id_estudiante`  
 112 `INNER JOIN Clase CL ON I.id_clase = CL.id_clase`

Result Grid

	estudiante	curso
--	------------	-------

Result 3 x Read Only Context Help

Output

Action Output

#	Time	Action	Message
✓ 1	23:46:13	SELECT E.nombre AS estudiante, C.nombre AS curso...	0 row(s) returned

4. Mostrar todos los estudiantes y, si están inscritos, el curso correspondiente (LEFT JOIN).

The screenshot shows a SQL query editor with the following query:

```
115 • SELECT E.nombre AS estudiante, C.nombre AS curso
116 FROM Estudiante E
117 LEFT JOIN Incripcion I ON E.id_estudiante = I.id_estudiante
```

Below the query editor, there is a toolbar with options like "Limit to 1000 rows", "Filter Rows:", "Export:", and "Wrap Cell Content:". A "Result Grid" button is also visible.

The "Result Grid" shows the following columns:

estudiante	curso
------------	-------

Below the result grid, there is a "Result 4" tab with a "Read Only" status. The "Output" section shows the execution results:

#	Time	Action	Message
1	23:47:58	SELECT E.nombre AS estudiante, C.nombre AS curso...	0 row(s) returned

5. Mostrar todos los cursos, incluyendo los que aún no tienen estudiantes inscritos (RIGHT JOIN).

The screenshot shows a database management interface with a SQL query editor and a results pane. The query is as follows:

```
121 • SELECT C.nombre AS curso, E.nombre AS estudiante
122 FROM Curso C
123 RIGHT JOIN Clase CL ON C.id_curso = CL.id_curso
```

Below the query editor, there is a toolbar with options like "Result Grid", "Filter Rows", "Export", and "Wrap Cell Content". A small table header is visible with columns "curso" and "estudiante".

The "Output" pane at the bottom shows the execution results:

#	Time	Action	Message
✓ 1	23:49:06	SELECT C.nombre AS curso, E.nombre AS estudiante...	0 row(s) returned



Contar cuántos estudiantes hay por departamento.

The screenshot shows a database query interface. The SQL query is as follows:

```
127 • SELECT D.nombre AS departamento, COUNT(E.id_estudiante) AS tota
128 FROM Departamento D
129 LEFT JOIN Estudiante E ON D.id_departamento = E.id_departamento
```

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

departamento	total_estudiantes
Ciencias	0
Ingeniería	0

The interface also includes a toolbar with icons for file operations, a 'Limit to 1000 rows' dropdown, and a 'Result Grid' button. The 'Output' pane at the bottom shows the query execution details:

#	Time	Action	Message
1	23:50:33	SELECT D.nombre AS departamento, COUNT(E.id_es...	2 row(s) returned

7. Calcular el promedio de notas por estudiante.

132 • `SELECT E.nombre, AVG(CA.nota) AS promedio`

133 `FROM Estudiante E`

134 `JOIN Inscripcion I ON E.id_estudiante = I.id_estudiante`

Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content: 


nombre	promedio
--------	----------

Result Grid

Result 7 x

Read Only

Output

 Action Output

#	Time	Action	Message
1	23:51:56	SELECT E.nombre, AVG(CA.nota) AS promedio FRO...	0 row(s) returned

### 8. Mostrar la nota máxima y mínima por clase

The screenshot shows a SQL IDE interface. The query editor contains the following SQL code:

```
138 • SELECT CL.id_clase, MAX(CA.nota) AS nota_max, MIN(CA.nota) AS n
139 FROM Clase CL
140 JOIN Inscripcion I ON CL.id_clase = I.id_clase
```

Below the query editor, the 'Result Grid' tab is active, showing a table with the following columns:

id_clase	nota_max	nota_min
----------	----------	----------

The 'Output' pane shows the execution log:

#	Time	Action	Message
1	23:51:56	SELECT E.nombre, AVG(CA.nota) AS promedio FRO...	0 row(s) returned
2	23:53:06	SELECT CL.id_clase, MAX(CA.nota) AS nota_max, MI...	0 row(s) returned

### Mostrar los 5 estudiantes con el mayor promedio de notas

The screenshot shows a SQL IDE interface. The query editor contains the following SQL code:

```
145 • SELECT E.nombre, AVG(CA.nota) AS promedio
146 FROM Estudiante E
147 JOIN Inscripcion I ON E.id_estudiante = I.id_estudiante
```

Below the query editor, the 'Result Grid' tab is active, showing a table with the following columns:

nombre	promedio
--------	----------

The 'Output' pane shows the execution log:

#	Time	Action	Message
1	23:54:36	SELECT E.nombre, AVG(CA.nota) AS promedio FRO...	0 row(s) returned

```
153 • UPDATE Estudiante
154 SET correo = 'nuevo.correo@email.com'
155 WHERE id_estudiante = 1;
```

Context Help Snippets

Output

Action Output

#	Time	Action	Message
✓ 1	23:56:22	UPDATE Estudiante SET correo = 'nuevo.correo@email.c...	0 row(s) affected Rows matched: 0 Changed: 0 Warnings...

## Actualizar y Borrar Datos

```
159
160 -- 10. Cambiar correo de un estudiante
161 • UPDATE Estudiante
162 SET correo = 'nuevo.correo@email.com'
163 WHERE id_estudiante = 1;
```

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	00:00:33	UPDATE Estudiante SET correo = 'nuevo.correo@email....	0 row(s) affected Rows matched: 0 Changed: 0 Warnings...	0.015 sec

```
164
165 -- 11. Eliminar una inscripción
166 • DELETE FROM Inscripcion
167 WHERE id_inscripcion = 1;
```

Output

Action Output

#	Time	Action	Message
✓ 1	00:01:36	DELETE FROM Inscripcion WHERE id_inscripcion = 1	0 row(s) affect