



José Armando Alcántara

**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

MySQL Workbench

Local instance MYSQL80

File Edit View Query Database Server Tools Scripting Help

Navegador

SCHEMAS

- biblioteca
- colegio
- sakila
- sys
- universidad2
- Tables
- Views
- Stored Procedures
- Functions
- ventas
- world

practicagabriel\_consultas\*

2096  
2097 -- 1. Seleccionar todos los estudiantes  
2098 • SELECT \* FROM Estudiante;  
2099 -- 2. Listar solo los nombres y apellidos  
2100 • SELECT Nombre, Apellido FROM Estudiante;  
2101 -- 3. Filtrar estudiantes de un departamento  
2102 • SELECT \* FROM Estudiante  
2103 WHERE DepartamentoID = 1;  
2104 -- 4. Ordenar estudiantes por fecha de nacimiento (más viejos primero)

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

EstudianteID	Nombre	Apellido	FechaNacimiento	Email	DepartamentoID
1	Casenro	Manrique	1994-11-02	pferreira@hotmail.com	5
2	Che	Tovar	2001-08-11	ineneperello@outlook.com	4
3	Esfafio	Pes	2002-12-27	sandhyap@duarte.es	3
4	Pepito	Campoy	2004-09-12	castilofrancisco-javier@villar-hervas.es	1
5	Santiago	España	2002-02-03	remar05@fernandez.es	4
6	Encarnacion	Aliona	2009-04-11	apina@yahoo.com	5
7	Nacho	Marilo	2000-03-06	jeaurez@aldana.es	5
8	Lupe	Vilar	2001-11-23	tomasmanzano@yahoo.com	4
9	Iker	Pifol	1995-03-02	ciprianocepida@casas-galvan.es	5
10	Anna	Blanes	1997-09-01	jorgejome@azconia.es	5

No object selected

Output

Action Output

# Time Action

1 18:38:56 SELECT \* FROM Estudiante LIMIT 0, 1000

Message 100 row(s) returned

Duration / Fetch 0.000 sec / 0.000 sec

Object Info Session

## 2. Listar solo los nombres y apellidos

The screenshot shows the MySQL Workbench interface with a query editor titled "practicagrabada\_consultas". The code entered is:

```
2096 -- 1. Seleccionar todos los estudiantes
2098 • SELECT * FROM Estudiante;
2099 -- 2. Listar solo los nombres y apellidos
2100 • SELECT Nombre, Apellido FROM Estudiante;
2101 -- 3. Filtrar estudiantes de un departamento
2102 • SELECT * FROM Estudiante
2103 WHERE DepartamentoID = 1;
2104 -- 4. Ordenar estudiantes por fecha de nacimiento (más viejos primero)
```

The Result Grid displays the following data:

	Nombre	Apellido
1	Cesario	Manrique
2	Chu	Tovar
3	Espinario	Pou
4	Pepeito	Campoy
5	Santiago	España
6	Encarnacion	Alina
7	Nacho	Mariño
8	Lupe	Vilar
9	Iber	Pifol
10	Anna	Blanes

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	18:38:56	SELECT * FROM Estudiante LIMIT 0, 1000	100 row(s) returned	0.000 sec / 0.000 sec
2	18:41:33	SELECT Nombre, Apellido FROM Estudiante LIMIT 0, 1000	100 row(s) returned	0.000 sec / 0.000 sec

## 3. Filtrar estudiantes de un departamento

The screenshot shows the MySQL Workbench interface with a query editor titled "practicagrabada\_consultas". The code entered is:

```
2096 -- 1. Seleccionar todos los estudiantes
2098 • SELECT * FROM Estudiante;
2099 -- 2. Listar solo los nombres y apellidos
2100 • SELECT Nombre, Apellido FROM Estudiante;
2101 -- 3. Filtrar estudiantes de un departamento
2102 • SELECT * FROM Estudiante
2103 WHERE DepartamentoID = 1;
2104 -- 4. Ordenar estudiantes por fecha de nacimiento (más viejos primero)
```

The Result Grid displays the following data:

EstudianteID	Nombre	Apellido	FechaNacimiento	Email	DepartamentoID
4	Pepeito	Campoy	2004-09-12	castillofrancisco-javier@villar-hervas.es	1
11	José	Pérez	1999-10-01	yaza77@abellan-samartin.es	1
15	Espínido	Reguera	2006-01-05	hasenrio@hotmail.com	1
18	Aros	Ceballos	2000-10-15	ponticoyer@gmail.com	1
19	Álbero	Perera	1995-01-06	mmascaro@sierra-domínguez.com	1
30	Eutropio	Mayo	2000-02-04	ycamino@alcarcon.es	1
34	Moisés	Rivas	1999-06-26	hsaera@yahoo.com	1
37	Rico	Cabo	1995-10-12	anabelcarbo@gmail.com	1
43	Eustaquio	Goicoceda	2000-02-06	cruz05@campio.es	1
45	Eugenio	Casals	1997-04-14	garcesmohamed@robles.es	1

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	18:38:56	SELECT * FROM Estudiante LIMIT 0, 1000	100 row(s) returned	0.000 sec / 0.000 sec
2	18:41:33	SELECT Nombre, Apellido FROM Estudiante LIMIT 0, 1000	100 row(s) returned	0.000 sec / 0.000 sec
3	18:42:06	SELECT * FROM Estudiante WHERE DepartamentoID = 1 LIMIT 0, 1000	19 row(s) returned	0.000 sec / 0.000 sec
4	18:42:15	SELECT * FROM Estudiante WHERE DepartamentoID = 1 LIMIT 0, 1000	19 row(s) returned	0.000 sec / 0.000 sec
5	18:42:47	SELECT * FROM Estudiante WHERE DepartamentoID = 1	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server... 0.015 sec	
6	18:42:54	SELECT * FROM Estudiante WHERE DepartamentoID = 1 LIMIT 0, 1000	19 row(s) returned	0.000 sec / 0.000 sec

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

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `universidad2` with its tables, views, stored procedures, functions, and triggers.
- SQL Editor:** Contains the following SQL code:
 

```

2101 -- 3. Filtrar estudiantes de un departamento
2102 • SELECT * FROM Estudiante
2103 WHERE DepartamentoID = 1;
2104 -- 4. Ordenar estudiantes por fecha de nacimiento (más viejos primero)
2105 • SELECT * FROM Estudiante
2106 ORDER BY FechaNacimiento ASC;
2107 -- 5. Contar cuántos estudiantes hay
2108 • SELECT COUNT(*) AS TotalEstudiantes FROM Estudiante;
2109 -- 6. Buscar estudiantes con apellido 'García'
      
```
- Result Grid:** Displays the results of the query, showing 19 rows of student data with columns: EstudianteID, Nombre, Apellido, FechaNacimiento, Email, and DepartamentoID.
- Action Output:** Shows the execution history of the statements, including the time, action, message, and duration.

#### 5. Contar cuántos estudiantes hay

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `universidad2` with its tables, views, stored procedures, functions, and triggers.
- SQL Editor:** Contains the following SQL code:
 

```

2101 -- 3. Filtrar estudiantes de un departamento
2102 • SEL Execute the selected portion of the script or everything, if there is no selection
2103 WHERE DepartamentoID = 1;
2104 -- 4. Ordenar estudiantes por fecha de nacimiento (más viejos primero)
2105 • SELECT * FROM Estudiante
2106 ORDER BY FechaNacimiento ASC;
2107 -- 5. Contar cuántos estudiantes hay
2108 • SELECT COUNT(*) AS TotalEstudiantes FROM Estudiante;
2109 -- 6. Buscar estudiantes con apellido 'García'
      
```
- Result Grid:** Displays the results of the query, showing a single row with the value 100 under the column TotalEstudiantes.
- Action Output:** Shows the execution history of the statements, including the time, action, message, and duration.

## 6. Buscar estudiantes con apellido 'García'

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `universidad2` with its tables (`Estudiante`, `Departamento`, etc.).
- SQL Editor:** Contains the following SQL code:
 

```

2104 -- 4. Ordenar estudiantes por fecha de nacimiento (más viejos primero)
2105 • SELECT * FROM Estudiante
2106 ORDER BY FechaNacimiento ASC;
2107 -- 5. Contar cuántos estudiantes hay
2108 • SELECT COUNT(*) AS TotalEstudiantes FROM Estudiante;
2109 -- 6. Buscar estudiantes con apellido 'García'
2110 • SELECT * FROM Estudiante
2111 WHERE Apellido = 'García';
2112 -- 7. Buscar por patrón (nombres que empiezan con 'A')
      
```
- Result Grid:** Displays the results of the last query, showing columns: EstudianteID, Nombre, Apellido, FechaNacimiento, Email, and DepartamentoID. No rows are present.
- Action Output:** Shows the execution history with the following details:
 

#	Time	Action	Message	Duration / Fetch
6	18:42:54	SELECT * FROM Estudiante WHERE DepartamentoID = 1 LIMIT 0, 1000	19 row(s) returned	0.000 sec / 0.000 sec
7	18:43:29	SELECT * FROM Estudiante ORDER BY FechaNacimiento ASC LIMIT 0, 1000	100 row(s) returned	0.016 sec / 0.000 sec
8	18:44:01	SELECT COUNT(*) AS TotalEstudiantes FROM Estudiante LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
9	18:44:26	SELECT * FROM Estudiante WHERE Apellido = 'García' LIMIT 0, 1000	0 row(s) returned	0.015 sec / 0.000 sec
10	18:44:55	SELECT * FROM Estudiante WHERE Apellido = 'García' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
11	18:47:29	SELECT * FROM Estudiante WHERE Apellido = 'García' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

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

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `universidad2` with its tables.
- SQL Editor:** Contains the following SQL code:
 

```

2108 • SELECT COUNT(*) AS TotalEstudiantes FROM Estudiante;
2109 - Execute the selected portion of the script or everything, if there is no selection
2110 • SELECT * FROM Estudiante
2111 WHERE Apellido = 'García';
2112 -- 7. Buscar por patrón (nombres que empiezan con 'A')
2113 • SELECT * FROM Estudiante
2114 WHERE Nombre LIKE 'A%';
2115
2116 -- 8. Join: Mostrar nombre del estudiante y el nombre del departamento
      
```
- Result Grid:** Displays the results of the last query, showing columns: EstudianteID, Nombre, Apellido, FechaNacimiento, Email, and DepartamentoID. The results are:
 

EstudianteID	Nombre	Apellido	FechaNacimiento	Email	DepartamentoID
10	Anna	Blanco	1997-09-01	jorge.tome@azcuna.es	5
16	Ariadna	Iriarte	1998-02-20	dionisia.fafonso@lado-moraleda.es	4
17	Asunción	Ulopis	1999-09-29	marupalecios@gmail.com	3
18	Aroa	Caballés	2000-10-15	pioncoquero@gmail.com	1
19	Álvaro	Pereira	1995-01-06	mmascaro@sierra-dominguez.com	1
23	Amaro	Aramburu	1997-08-22	alcantracasmiro@lanos-isern.es	5
27	Ángel	Herranz	2003-06-29	ascensionjume@hotmail.com	5
31	Amando	Lerma	1994-07-13	mauricio.villa@gmail.com	4
36	Addifo	Rebollo	1999-07-20	aristides30@pou.org	4
38	Adelardo	Galindo	1998-10-03	gibertgodofredo@hotmail.com	5
- Action Output:** Shows the execution history with the following details:
 

#	Time	Action	Message	Duration / Fetch
7	18:43:29	SELECT * FROM Estudiante ORDER BY FechaNacimiento ASC LIMIT 0, 1000	100 row(s) returned	0.016 sec / 0.000 sec
8	18:44:01	SELECT COUNT(*) AS TotalEstudiantes FROM Estudiante LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
9	18:44:26	SELECT * FROM Estudiante WHERE Apellido = 'García' LIMIT 0, 1000	0 row(s) returned	0.015 sec / 0.000 sec
10	18:44:55	SELECT * FROM Estudiante WHERE Apellido = 'García' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
11	18:47:29	SELECT * FROM Estudiante WHERE Apellido = 'García' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
12	18:47:53	SELECT * FROM Estudiante WHERE Nombre LIKE 'A%' LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

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

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `universidad2` with tables `estudiante`, `departamento`, and `calificacion`.
- SQL Editor:** Contains the following SQL code:
 

```

2115
2116 -- 8. ⚡ Join: Mostrar nombre del estudiante y el nombre del departamento
2117 • SELECT E.Nombre, E.Apellido, D.Nombre AS Departamento
2118 FROM Estudiante E
2119 JOIN Departamento D ON E.DepartamentoID = D.DepartamentoID
2120
2121 -- 9. ⚡ Promedio de calificaciones por estudiante
2122 • SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio
2123 FROM Calificacion C
      
```
- Result Grid:** Displays the results of the first query, showing columns `Nombre`, `Apellido`, and `Departamento`. The data includes rows for Pepe, Juan, Espinido, Aris, Alvaro, Eutropio, Moisés, Rico, Eustaquio, and Eugenio, all belonging to the `Ingiereria` department.
- Action Output:** Shows the execution log with 13 entries, detailing each query execution and its duration.

## 9. Promedio de calificaciones por estudiante

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `universidad2` with tables `estudiante`, `departamento`, `calificacion`, `inscripcion`, and `profesor`.
- SQL Editor:** Contains the following SQL code:
 

```

2120
2121 -- 9. ⚡ Promedio de calificaciones por estudiante
2122 • SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio
2123 FROM Calificacion C
2124 JOIN Inscripcion I ON C.InscripcionID = I.InscripcionID
2125 JOIN Estudiante E ON I.EstudanteID = E.EstudanteID
2126 GROUP BY E.EstudanteID
2127
2128 -- 10. ⚡ Cantidad de estudiantes por departamentos
      
```
- Result Grid:** Displays the results of the second query, showing columns `Nombre`, `Apellido`, and `Promedio`. The data includes rows for Eugenio, Teo, Mar, Valero, Ch, Lorena, Ascension, Trinidad, and Pacifica, with their respective average grades.
- Action Output:** Shows the execution log with 1 entry, indicating 66 rows returned.

## 10. Cantidad de estudiantes por departamento

The screenshot shows the MySQL Workbench interface with a query editor window titled "practicagrabada\_consultas". The query is:

```
2125 JOIN Estudiante E ON I.EstudienteID = E.EstudienteID
2126 GROUP BY E.EstudienteID;
2127
2128 -- 10. ④ Cantidad de estudiantes por departamento
2129 • SELECT D.Nombre AS Departamento, COUNT(E.EstudienteID) AS TotalEstudiantes
2130 FROM Estudiante E
2131 JOIN Departamento D ON E.DepartamentoID = D.DepartamentoID
2132 GROUP BY D.Nombre;
2133
2134 -- 11. ④ Cursos impartidos por cada profesor
```

The result grid shows the following data:

Departamento	TotalEstudiantes
Ingeniería	19
Matemáticas	18
Historia	13
Biológica	14
Filosofía	21
Economía	15

The output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
12	18:47:53	SELECT * FROM Estudiante WHERE Nombre LIKE 'A%' LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
13	18:48:20	SELECT E.Nombre, E.Apellido, D.Nombre AS Departamento FROM Estudiante E JOIN Departamento D ON E.DepartamentoID = D.DepartamentoID	100 row(s) returned	0.000 sec / 0.000 sec
14	18:48:53	SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio FROM Calificacion C JOIN Inscripcion I ON C.InscripcionID = I.InscripcionID	0 row(s) returned	0.000 sec / 0.000 sec
15	18:50:11	SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio FROM Calificacion C JOIN Inscripcion I ON C.InscripcionID = I.InscripcionID	0 row(s) returned	0.016 sec / 0.000 sec
16	18:50:21	SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio FROM Calificacion C JOIN Inscripcion I ON C.InscripcionID = I.InscripcionID	0 row(s) returned	0.000 sec / 0.000 sec
17	18:50:29	SELECT D.Nombre AS Departamento, COUNT(E.EstudienteID) AS TotalEstudiantes FROM Estudiante E JOIN Departamento D ON E.DepartamentoID = D.DepartamentoID	6 row(s) returned	0.000 sec / 0.000 sec

## 11. Cursos impartidos por cada profesor

The screenshot shows the MySQL Workbench interface with a query editor window titled "practicagrabada\_consultas". The query is:

```
2132 GROUP BY D.Nombre;
2133
2134 -- 11. ④ Cursos impartidos por cada profesor
2135 • SELECT P.Nombre, P.Apellido, COUNT(DISTINCT C.ClaseID) AS TotalClases
2136 FROM Profesor P
2137 JOIN Clase C ON P.ProfesorID = C.ProfesorID
2138 GROUP BY P.ProfesorID;
2139
2140 -- 12. ④ Estudiantes con promedio mayor a 90
```

The result grid shows the following data:

Nombre	Apellido	TotalClases
Domingo	Curtio	1
Víctor	Soriano	2
Albina	Múgica	1
Pánfilo	Brito	1
Donato	Tello	2
Chelo	Haro	1
Leonardo	Villalba	1
Elsa	Cerro	1
Cleto	Aller	1
Chelo	Prado	2

The output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
13	18:48:20	SELECT E.Nombre, E.Apellido, D.Nombre AS Departamento FROM Estudiante E JOIN Departamento D ON E.DepartamentoID = D.DepartamentoID	100 row(s) returned	0.000 sec / 0.000 sec
14	18:48:53	SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio FROM Calificacion C JOIN Inscripcion I ON C.InscripcionID = I.InscripcionID	0 row(s) returned	0.000 sec / 0.000 sec
15	18:50:11	SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio FROM Calificacion C JOIN Inscripcion I ON C.InscripcionID = I.InscripcionID	0 row(s) returned	0.016 sec / 0.000 sec
16	18:50:21	SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio FROM Calificacion C JOIN Inscripcion I ON C.InscripcionID = I.InscripcionID	0 row(s) returned	0.000 sec / 0.000 sec
17	18:50:29	SELECT D.Nombre AS Departamento, COUNT(E.EstudienteID) AS TotalEstudiantes FROM Estudiante E JOIN Departamento D ON E.DepartamentoID = D.DepartamentoID	6 row(s) returned	0.000 sec / 0.000 sec
18	18:51:12	SELECT P.Nombre, P.Apellido, COUNT(DISTINCT C.ClaseID) AS TotalClases FROM Profesor P JOIN Clase C ON P.ProfesorID = C.ProfesorID	67 row(s) returned	0.015 sec / 0.000 sec

## 12. Estudiantes con promedio mayor a 90

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (biblioteca, colegio, sakila, sys, universidad2), Tables (clase, curso, departamento, estudiante, inscripcion, profesor), Views, Stored Procedures, Functions, ventas, world.
- SQL Editor:** A query window titled "practicagabada\_consultas" containing the following SQL code:

```
2139
2140   * Execute the selected portion of the script or everything, if there is no selection
2141 •   SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio
2142   FROM Estudiante E
2143   JOIN Inscripcion I ON E.EstudianteID = I.EstudianteID
2144   JOIN Calificacion C ON I.InscripcionID = C.InscripcionID
2145   GROUP BY E.EstudianteID
2146   HAVING AVG(C.Nota) > 90;
```
- Result Grid:** Shows the results of the query:

Nombre	Apellido	Promedio
Violeta	Arranz	93.690000
Ascension	Lopez	94.895000
Sosima	Luna	99.550000
Celestino	Sarmiento	93.270000
Debora	Higueras	92.850000
Rosario	Ulabres	92.480000
Agapito	Balle	92.150000
Trinidad	Castell	93.085000
Carmelita	Sebastian	91.390000
Dolores	Gallego	92.923333
- Output:** Action Output pane showing the execution log:

#	Time	A	Message
1	19:13:57	S.	66 row(s) returned
2	19:14:34	S.	12 row(s) returned

## 13. Top 5 estudiantes con mejores promedios

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (biblioteca, colegio, sakila, sys, universidad2), Tables (clase, curso, departamento, estudiante, inscripcion, profesor), Views, Stored Procedures, Functions, ventas, world.
- SQL Editor:** A query window titled "practicagabada\_consultas" containing the following SQL code:

```
2149
2150
2151 •   SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio
2152   FROM Estudiante E
2153   JOIN Inscripcion I ON E.EstudianteID = I.EstudianteID
2154   JOIN Calificacion C ON I.InscripcionID = C.InscripcionID
2155   GROUP BY E.EstudianteID
2156   ORDER BY Promedio DESC
2157   LIMIT 5;
```
- Result Grid:** Shows the results of the query:

Nombre	Apellido	Promedio
Maria Teresa	Amaya	99.880000
Sosima	Luna	99.550000
Raquel	Carboneil	96.350000
Ascension	Lopez	94.895000
Violeta	Arranz	93.690000
- Output:** Action Output pane showing the execution log:

#	Time	A	Message
1	19:13:57	S.	66 row(s) returned
2	19:14:34	S.	12 row(s) returned
3	19:15:05	S.	5 row(s) returned