



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

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

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

The results of the first query are displayed in the Result Grid:

EstudianteID	Nombre	Apellido	FechaNacimiento	Email	DepartamentoID
1	Casemiro	Manrique	1994-11-02	pferreas@hotmail.com	5
2	Che	Tovar	2001-08-11	veneperezo@pont.com	4
3	Epifanio	Pau	2002-12-27	sandhopio@duarte.es	3
4	Pepito	Campoy	2004-09-12	castlofrancisco-javier@villar-hervas.es	1
5	Santiago	España	2002-02-03	romano5@fernandez.es	4
6	Encarnacion	Alcina	2004-04-11	apina@yahoo.com	5
7	hacho	Manfo	2000-03-06	suarez@saldana.es	5
8	Lupe	Villar	2001-11-23	tonasanzano@yahoo.com	4
9	Iter	Pitui	1995-03-02	cprianocepeda@casae-galvan.es	5
10	Anna	Blanes	1997-09-01	jorgetome@azona.es	5

The Output pane shows the execution of the first query:

```
1 18:38:56 SELECT * FROM Estudiante LIMIT 0, 1000
Message: 100 row(s) returned
Duration / Fetch: 0.000 sec / 0.000 sec
```

2. Listar solo los nombres y apellidos

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

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

The Result Grid shows the output of the second query, displaying a list of student names and surnames:

Nombre	Apellido
Casimiro	Manrique
Ché	Tovar
Epifanio	Pou
Pepito	Campoy
Santiago	España
Encarnación	Alaña
Nacho	Manrífo
Lupe	Vilar
Bier	Péfol
Anna	Blanes

The Output pane shows the execution results:

#	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. The SQL editor contains the following queries:

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

The Result Grid shows the output of the third query, displaying a list of students filtered by department (DepartamentoID = 1):

EstudianteID	Nombre	Apellido	FechaNacimiento	Email	DepartamentoID
4	Pepito	Campoy	2004-09-12	castellanojavier@villar-hervas.es	1
11	José	Pérez	1999-10-01	yaiza77@babelan-sanmartin.es	1
15	Espidón	Requera	2006-01-05	hasensio@hotmail.com	1
18	Aroa	Cabañas	2000-10-15	poncioquero@gmail.com	1
19	Albano	Pereira	1995-01-06	mmascaro@sierra-dominguez.com	1
30	Eutropio	Mayo	2000-02-04	ycamino@alarcon.es	1
34	Mosés	Rivas	1999-06-26	hsaenz@yahoo.com	1
37	Rico	Cabo	1995-10-12	anabielcarbo@gmail.com	1
43	Eustaquio	Gacoechea	2000-02-06	crus05@campo.es	1
45	Eugenio	Casale	1997-04-14	garcmohamed@roble.es	1

The Output pane shows the execution results:

#	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 =	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL serv...	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. The SQL editor contains the following queries:

```
-- 3. Filtrar estudiantes de un departamento
SELECT * FROM Estudiante
WHERE DepartamentoID = 1;

-- 4. Ordenar estudiantes por fecha de nacimiento (más viejos primero)
SELECT * FROM Estudiante
ORDER BY FechaNacimiento ASC;

-- 5. Contar cuántos estudiantes hay
SELECT COUNT(*) AS TotalEstudiantes FROM Estudiante;

-- 6. Buscar estudiantes con apellido 'García'
```

The Result Grid shows the output of the first query, displaying student details for DepartamentoID 1:

EstudianteID	Nombre	Apellido	FechaNacimiento	Email	DepartamentoID
31	Amanda	Lerma	1994-07-13	mauriciosevilla@gmail.com	4
1	Casemiro	Manrique	1994-11-02	pferreras@hotmail.com	5
71	Ariel	Olivera	1994-12-22	perlanari@hotmail.com	6
19	Albano	Perera	1995-01-06	mmascaro@sierra-dominguez.com	1
25	Leocadio	Acedo	1995-01-08	mateo16@jimenez.com	2
40	Celestino	Sarmiento	1995-02-04	moreno01@yahoo.com	5
68	Rosalba	Abri	1995-02-19	garridomaria-angeles@gmail.com	3
9	Iker	Pineda	1995-03-02	carriacopepe@casas-galvan.es	5
61	Sara	Redondo	1995-09-24	cabad@terco.com	2
37	Rico	Cabo	1995-10-12	anabekarbo@gmail.com	1

The Output pane shows the execution log for the queries.

5. Contar cuántos estudiantes hay

The screenshot shows the MySQL Workbench interface. The SQL editor contains the same queries as in the previous screenshot. The Result Grid shows the output of the third query, displaying the total number of students:

EstudianteID	Nombre	Apellido	FechaNacimiento	Email	DepartamentoID
TotalEstudiantes					
100					

The Output pane shows the execution log for the queries.

6. Buscar estudiantes con apellido 'García'

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

```
-- 4. Ordenar estudiantes por fecha de nacimiento (más viejos primero)
SELECT * FROM Estudiante
ORDER BY FechaNacimiento ASC;

-- 5. Contar cuántos estudiantes hay
SELECT COUNT(*) AS TotalEstudiantes FROM Estudiante;

-- 6. Buscar estudiantes con apellido 'García'
SELECT * FROM Estudiante
WHERE Apellido = 'García';

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

The Results grid is empty. The Output pane shows the execution log:

#	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. The SQL editor contains the following queries:

```
-- 6. Buscar estudiantes con apellido 'García'
SELECT * FROM Estudiante
WHERE Apellido = 'García';

-- 7. Buscar por patrón (nombres que empiezan con 'A')
SELECT * FROM Estudiante
WHERE Nombre LIKE 'A%';

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

The Results grid shows the following data:

EstudianteID	Nombre	Apellido	FechaNacimiento	Email	DepartamentoID
10	Anna	Blanes	1997-09-01	jorgetome@azona.es	5
16	Ariadna	Irarte	1998-02-20	dionisavalforogledo-moraleda.es	4
17	Ascensión	Llopa	1999-09-29	manupalacios@gmail.com	3
18	Aroa	Cabeñas	2000-10-15	poncoquero@gmail.com	1
19	Albano	Perera	1995-01-06	mascaro@sierra-dominguez.com	1
23	Amaro	Aramburu	1997-08-22	alcantaracasemiro@lanos-iersm.es	5
27	Ángel	Herranz	2003-06-29	ascensionjaume@hotmail.com	5
31	Amando	Lerma	1994-07-13	mauricososvilla@gmail.com	4
36	Adolfo	Raballo	1999-07-20	aristides30@pou.org	4
38	Adelardo	Galdino	1998-10-03	gbertrando@redes@hotmail.com	5

The Output pane shows the execution log:

#	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 MySQL Workbench with a query window titled 'practicagrados_consultas'. The query is a JOIN between 'Estudiante' and 'Departamento' tables. The results grid shows 10 rows of student data with their respective departments.

```
-- 8. Join: Mostrar nombre del estudiante y el nombre del departamento
2116 SELECT E.Nombre, E.Apellido, D.Nombre AS Departamento
2117 FROM Estudiante E
2118 JOIN Departamento D ON E.DepartamentoID = D.DepartamentoID;
```

Nombre	Apellido	Departamento
Pepito	Campoy	Ingeniería
José	Pérez	Ingeniería
Espidón	Riquera	Ingeniería
Aroa	Cabañas	Ingeniería
Albano	Pereira	Ingeniería
Eutropio	Mayo	Ingeniería
Moisés	Rivas	Ingeniería
Rico	Cabo	Ingeniería
Eustaquio	García	Ingeniería
Eugenio	Castro	Ingeniería

The output window shows the execution of the query, indicating that 10 rows were returned.

9. Promedio de calificaciones por estudiante

The screenshot shows MySQL Workbench with a query window titled 'practicagrados_consultas'. The query calculates the average grade for each student. The results grid shows 10 rows of student data with their average grades.

```
-- 9. Promedio de calificaciones por estudiante
2121 SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio
2122 FROM Calificacion C
2123 JOIN Estudiante E ON C.EstudianteID = E.EstudianteID
2124 GROUP BY E.EstudianteID;
```

Nombre	Apellido	Promedio
Eugenio	Casals	64.200000
Teo	Bayón	73.150000
Mar	Almazán	68.385000
Valero	Gaya	78.370000
Eutropio	Mayo	84.906667
Che	Tovar	71.710000
Lorena	Torrealla	68.250000
Ascensión	Llopi	94.895000
Trinidad	Castell	93.085000
Pacifica	Felipe	70.605000

The output window shows the execution of the query, indicating that 10 rows were returned.

10. Cantidad de estudiantes por departamento

The screenshot shows the MySQL Workbench interface with a query window titled 'practicagrabada_consultas'. The query is as follows:

```
2125 JOIN Estudiante E ON I.EstudianteID = E.EstudianteID
2126 GROUP BY E.EstudianteID;
2127
-- 10. Cantidad de estudiantes por departamento
2129 SELECT D.Nombre AS Departamento, COUNT(E.EstudianteID) AS TotalEstudiantes
2130 FROM Estudiante E
2131 JOIN Departamento D ON E.DepartamentoID = D.DepartamentoID
2132 GROUP BY D.Nombre;
```

The result grid shows the following data:

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

The output window shows the execution log with the following messages:

- 12 18:47:53 SELECT * FROM Estudiante WHERE Nombre LIKE 'A%' LIMIT 0, 1000 16 row(s) returned
- 13 18:48:20 SELECT E.Nombre, E.Apellido, D.Nombre AS Departamento FROM Estudiante E JOIN Departamento D ON E... 100 row(s) returned
- 14 18:48:53 SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio FROM Calificacion C JOIN Inscripcion I ON C.Inscrip... 0 row(s) returned
- 15 18:50:11 SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio FROM Calificacion C JOIN Inscripcion I ON C.Inscrip... 0 row(s) returned
- 16 18:50:21 SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio FROM Calificacion C JOIN Inscripcion I ON C.Inscrip... 0 row(s) returned
- 17 18:50:29 SELECT D.Nombre AS Departamento, COUNT(E.EstudianteID) AS TotalEstudiantes FROM Estudiante E JOI... 6 row(s) returned

11. Cursos impartidos por cada profesor

The screenshot shows the MySQL Workbench interface with a query window titled 'practicagrabada_consultas'. The query is as follows:

```
2132 GROUP BY D.Nombre;
2133 -- 11. Cursos impartidos por cada profesor
2134
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;
```

The result grid shows the following data:

Nombre	Apellido	TotalClases
Domingo	Cueto	1
Víctor	Soriano	2
Albina	Múgica	1
Pánfilo	Elorza	1
Donato	Tello	2
Chelo	Haro	1
Leonardo	Vilalba	1
Elsa	Cerro	1
Cleto	Alar	1
Chelo	Prado	2

The output window shows the execution log with the following messages:

- 13 18:48:20 SELECT E.Nombre, E.Apellido, D.Nombre AS Departamento FROM Estudiante E JOIN Departamento D ON E... 100 row(s) returned
- 14 18:48:53 SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio FROM Calificacion C JOIN Inscripcion I ON C.Inscrip... 0 row(s) returned
- 15 18:50:11 SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio FROM Calificacion C JOIN Inscripcion I ON C.Inscrip... 0 row(s) returned
- 16 18:50:21 SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio FROM Calificacion C JOIN Inscripcion I ON C.Inscrip... 0 row(s) returned
- 17 18:50:29 SELECT D.Nombre AS Departamento, COUNT(E.EstudianteID) AS TotalEstudiantes FROM Estudiante E JOI... 6 row(s) returned
- 18 18:51:12 SELECT P.Nombre, P.Apellido, COUNT(DISTINCT C.ClaseID) AS TotalClases FROM Profesor P JOIN Clase C... 67 row(s) returned

12. Estudiantes con promedio mayor a 90

The screenshot shows the MySQL Workbench interface with a query window titled "practicagrados_consultas". The query is as follows:

```
SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio
FROM Estudiante E
JOIN Inscripcion I ON E.EstudianteID = I.EstudianteID
JOIN Calificacion C ON I.InscripcionID = C.InscripcionID
GROUP BY E.EstudianteID
HAVING AVG(C.Nota) > 90;
```

The result grid displays the following data:

Nombre	Apellido	Promedio
Violeta	Arranz	93.690000
Ascensón	Llopi	94.895000
Sosimo	Luna	99.550000
Celestino	Sarmiento	93.270000
Debora	Higuera	92.850000
Rosario	Liabres	92.480000
Agapito	Bartle	92.150000
Trinidad	Castell	93.085000
Carmelita	Sebastián	91.390000
Dolores	Gallego	92.923333

The output window shows the execution results:

#	Time	A	Message	Duration / Fetch
1	19:13:57	S	66 row(s) returned	0.000 sec / 0.000 sec
2	19:14:34	S	12 row(s) returned	0.000 sec / 0.000 sec

13. Top 5 estudiantes con mejores promedios

The screenshot shows the MySQL Workbench interface with a query window titled "practicagrados_consultas". The query is as follows:

```
SELECT E.Nombre, E.Apellido, AVG(C.Nota) AS Promedio
FROM Estudiante E
JOIN Inscripcion I ON E.EstudianteID = I.EstudianteID
JOIN Calificacion C ON I.InscripcionID = C.InscripcionID
GROUP BY E.EstudianteID
ORDER BY Promedio DESC
LIMIT 5;
```

The result grid displays the following data:

Nombre	Apellido	Promedio
Maria Teresa	Amaya	99.880000
Sosimo	Luna	99.550000
Raquel	Carbonell	96.350000
Ascensón	Llopi	94.895000
Violeta	Arranz	93.690000

The output window shows the execution results:

#	Time	A	Message	Duration / Fetch
1	19:13:57	S	66 row(s) returned	0.000 sec / 0.000 sec
2	19:14:34	S	12 row(s) returned	0.000 sec / 0.000 sec
3	19:15:05	S	5 row(s) returned	0.000 sec / 0.000 sec