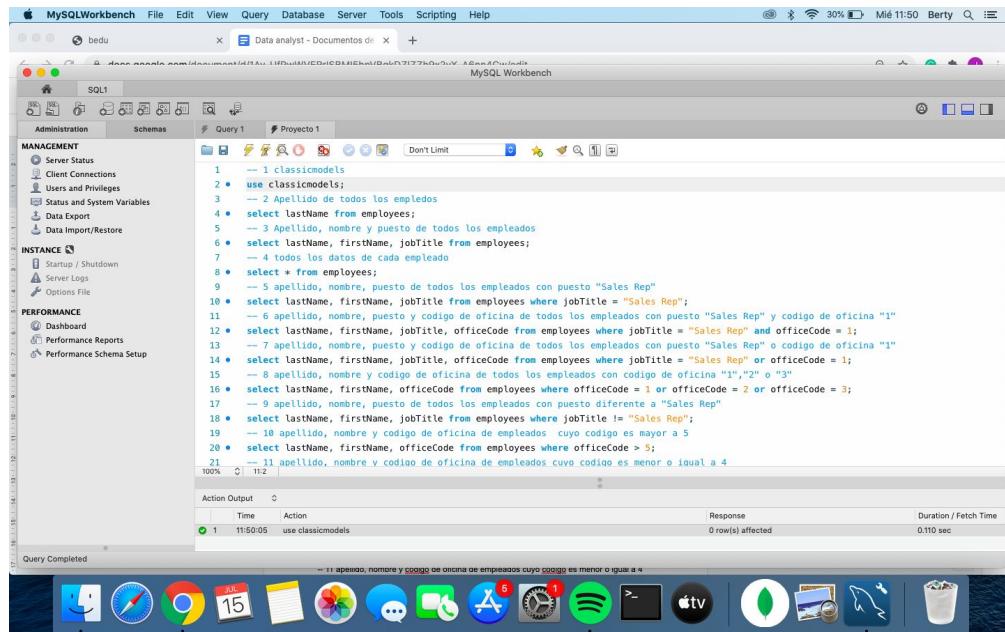


Sesión 01 Fundamentos SQL

Proyecto 1

-- 1 classicmodels
use classicmodels;



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQLWorkbench". The main window has a "SQL" tab selected. On the left is a sidebar with sections like "MANAGEMENT", "INSTANCE", and "PERFORMANCE". The central area contains the following SQL code:

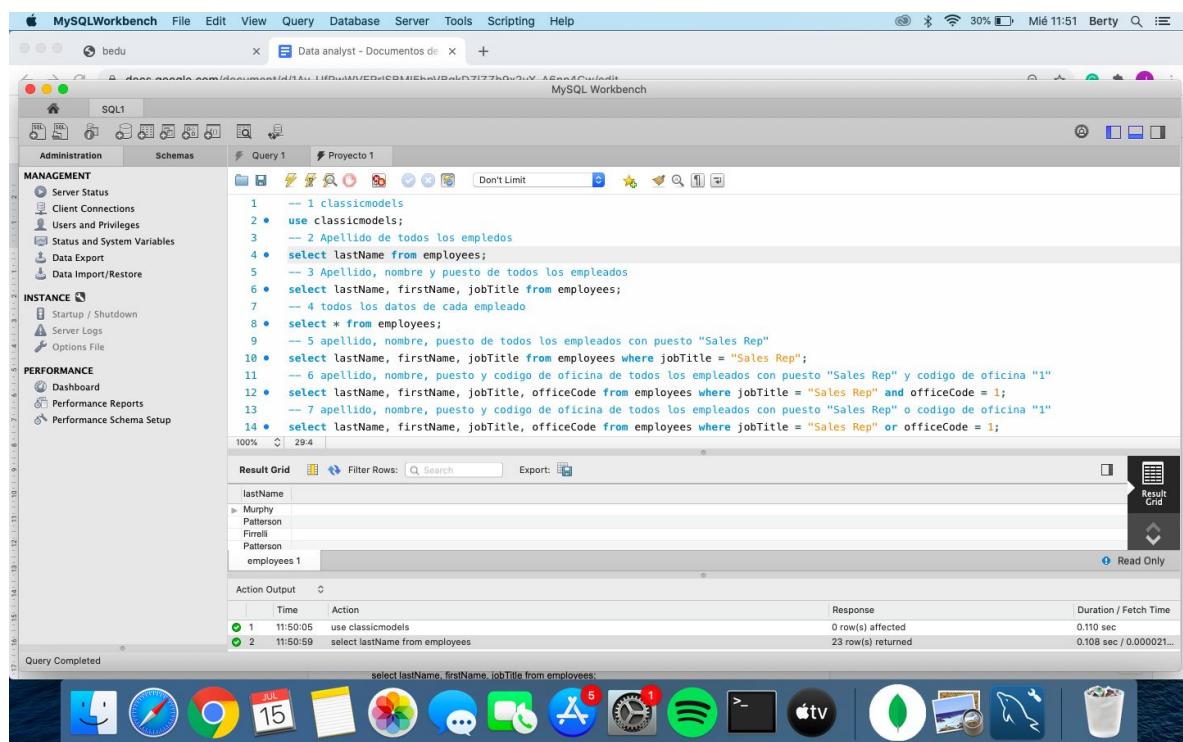
```
1 -- 1 classicmodels
2 • use classicmodels;
3 -- 2 Apellido de todos los empleados
4 • select lastName from employees;
5 -- 3 Apellido, nombre y puesto de todos los empleados
6 • select lastName, firstName, jobTitle from employees;
7 -- 4 todos los datos de cada empleado
8 • select * from employees;
9 -- 5 apellido, nombre, puesto de todos los empleados con puesto "Sales Rep"
10 • select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep";
11 -- 6 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" y codigo de oficina "1"
12 • select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;
13 -- 7 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" o codigo de oficina "1"
14 • select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" or officeCode = 1;
15 -- 8 apellido, nombre y codigo de oficina de todos los empleados con codigo de oficina "1","2" o "3"
16 • select lastName, firstName, officeCode from employees where officeCode = 1 or officeCode = 2 or officeCode = 3;
17 -- 9 apellido, nombre, puesto de todos los empleados con puesto diferente a "Sales Rep"
18 • select lastName, firstName, jobTitle from employees where jobTitle != "Sales Rep";
19 -- 10 apellido, nombre y codigo de oficina de empleados cuyo codigo es mayor a 5
20 • select lastName, firstName, officeCode from employees where officeCode > 5;
21 -- 11 apellido, nombre y codigo de oficina de empleados cuyo codigo es menor o igual a 4
```

Below the code, there is a table titled "Action Output" with two rows:

Action	Time	Action	Response	Duration / Fetch Time
1	11:50:05	use classicmodels	0 row(s) affected	0.110 sec

The status bar at the bottom of the window says "Query Completed".

-- 2 Apellido de todos los empleados
select lastName from employees;



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQLWorkbench". The main window has a "SQL" tab selected. On the left is a sidebar with sections like "MANAGEMENT", "INSTANCE", and "PERFORMANCE". The central area contains the following SQL code:

```
1 -- 1 classicmodels
2 • use classicmodels;
3 -- 2 Apellido de todos los empleados
4 • select lastName from employees;
5 -- 3 Apellido, nombre y puesto de todos los empleados
6 • select lastName, firstName, jobTitle from employees;
7 -- 4 todos los datos de cada empleado
8 • select * from employees;
9 -- 5 apellido, nombre, puesto de todos los empleados con puesto "Sales Rep"
10 • select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep";
11 -- 6 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" y codigo de oficina "1"
12 • select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;
13 -- 7 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" o codigo de oficina "1"
14 • select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" or officeCode = 1;
```

Below the code, there is a table titled "Result Grid" with two rows:

lastName
Murphy
Patterson
Firelli
Patterson

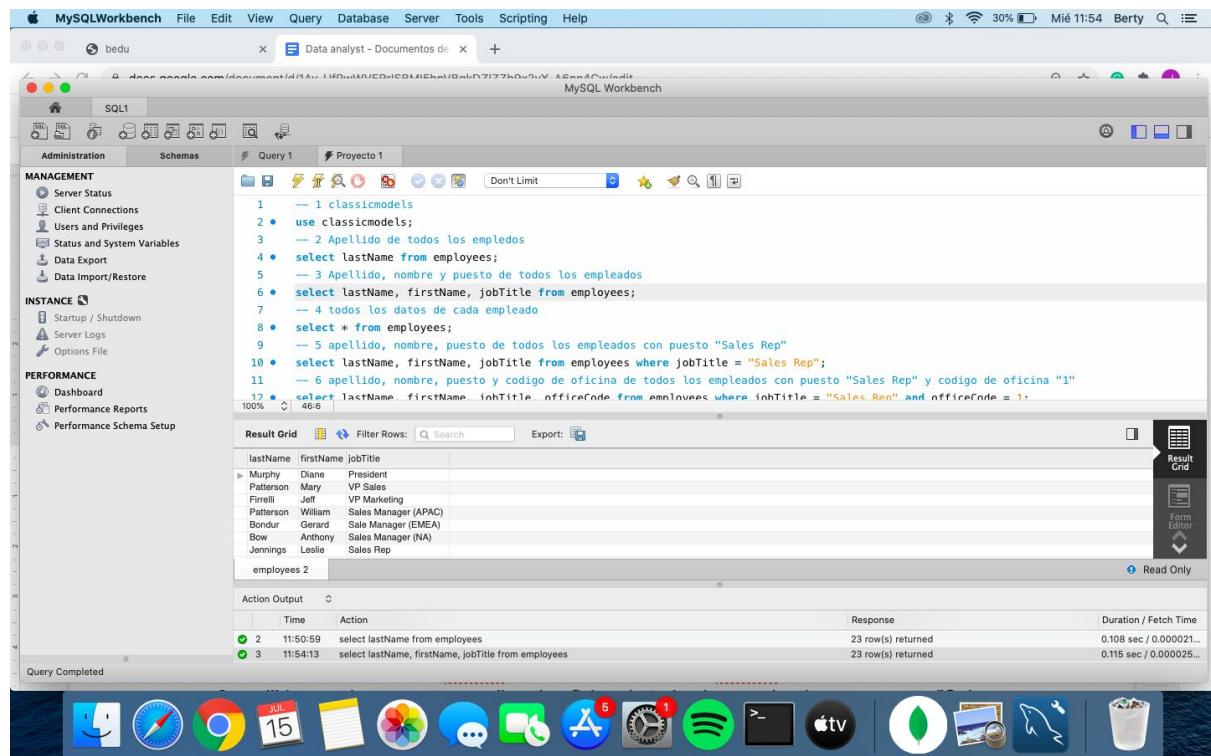
At the bottom of the table, it says "employees 1".

Below the table, there is a table titled "Action Output" with two rows:

Action	Time	Action	Response	Duration / Fetch Time
1	11:50:05	use classicmodels	0 row(s) affected	0.110 sec
2	11:50:59	select lastName from employees;	23 row(s) returned	0.108 sec / 0.000021...

The status bar at the bottom of the window says "Query Completed".

-- 3 Apellido, nombre y puesto de todos los empleados
 select lastName, firstName, jobTitle from employees;



```

MySQLWorkbench  File  Edit  View  Query  Database  Server  Tools  Scripting  Help
bedu          X  Data analyst - Documentos de  +  Mié 11:54  Berty  S

SQL1
Administration  Schemas  Query 1  Proyecto 1

MANAGEMENT
    Server Status
    Client Connections
    Users and Privileges
    Status and System Variables
    Data Export
    Data Import/Restore

INSTANCE
    Startup / Shutdown
    Server Logs
    Options File

PERFORMANCE
    Dashboard
    Performance Reports
    Performance Schema Setup

1 -- 1 classicmodels
2 • use classicmodels;
3 -- 2 Apellido de todos los empleados
4 • select lastName from employees;
5 -- 3 Apellido, nombre y puesto de todos los empleados
6 • select lastName, firstName, jobTitle from employees;
7 -- 4 todos los datos de cada empleado
8 • select * from employees;
9 -- 5 apellido, nombre, puesto de todos los empleados con puesto "Sales Rep"
10 • select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep";
11 -- 6 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" y codigo de oficina "1"
12 • select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;

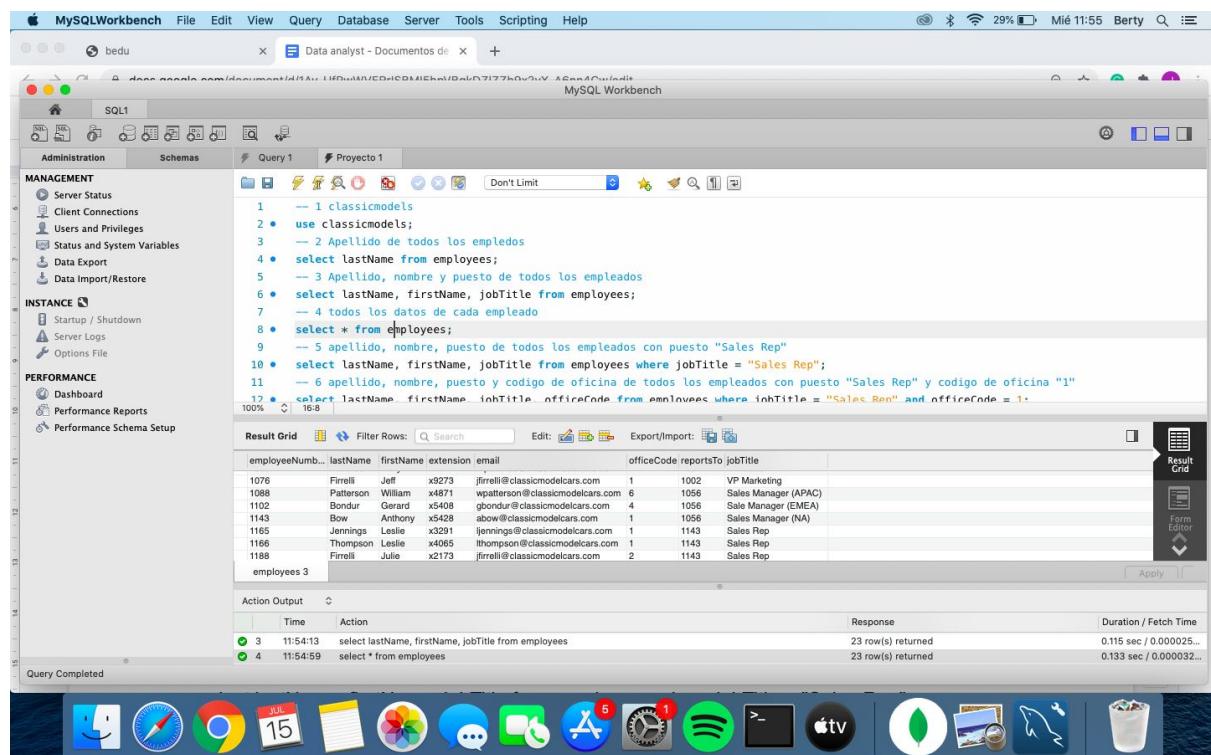
Result Grid  Filter Rows: Q Search  Export:  Result Grid
employees 2  Form Editor  Read Only

Action Output  C
Time Action Response Duration / Fetch Time
2 11:50:59 select lastName from employees 23 row(s) returned 0.108 sec / 0.000021...
3 11:54:13 select lastName, firstName, jobTitle from employees 23 row(s) returned 0.115 sec / 0.000025...

Query Completed

```

-- 4 todos los datos de cada empleado
 select * from employees;



```

MySQLWorkbench  File  Edit  View  Query  Database  Server  Tools  Scripting  Help
bedu          X  Data analyst - Documentos de  +  Mié 11:55  Berty  S

SQL1
Administration  Schemas  Query 1  Proyecto 1

MANAGEMENT
    Server Status
    Client Connections
    Users and Privileges
    Status and System Variables
    Data Export
    Data Import/Restore

INSTANCE
    Startup / Shutdown
    Server Logs
    Options File

PERFORMANCE
    Dashboard
    Performance Reports
    Performance Schema Setup

1 -- 1 classicmodels
2 • use classicmodels;
3 -- 2 Apellido de todos los empleados
4 • select lastName from employees;
5 -- 3 Apellido, nombre y puesto de todos los empleados
6 • select lastName, firstName, jobTitle from employees;
7 -- 4 todos los datos de cada empleado
8 • select * from employees;
9 -- 5 apellido, nombre, puesto de todos los empleados con puesto "Sales Rep"
10 • select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep";
11 -- 6 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" y codigo de oficina "1"
12 • select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;

Result Grid  Filter Rows: Q Search  Edit:  Export/Import:  Result Grid
employees 3  Form Editor  Apply

Action Output  C
Time Action Response Duration / Fetch Time
3 11:54:13 select lastName, firstName, jobTitle from employees 23 row(s) returned 0.115 sec / 0.000025...
4 11:54:59 select * from employees 23 row(s) returned 0.133 sec / 0.000032...

Query Completed

```

```
-- 5 apellido, nombre, puesto de todos los empleados con puesto "Sales Rep"  
select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep";
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQLWorkbench". The left sidebar has sections for MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), and PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup). The main area has tabs for Administration, Schemas, Query 1, and Proyecto 1. The Query 1 tab is active, displaying the following SQL code:

```
1 -- 1 classicmodels
2 • use classicmodels;
3 -- 2 Apellido de todos los empleados
4 • select lastName from employees;
5 -- 3 Apellido, nombre y puesto de todos los empleados
6 • select lastName, firstName, jobTitle from employees;
7 -- 4 todos los datos de cada empleado
8 • select * from employees;
9 -- 5 apellido, nombre, puesto de todos los empleados con puesto "Sales Rep"
10 • select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep";
11 -- 6 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" y codigo de oficina "1"
12 • select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;
100% 19:10
```

The Result Grid shows the following data:

lastName	firstName	jobTitle
Firelli	Julie	Sales Rep
Patterson	Steve	Sales Rep
Tseng	Foon Yue	Sales Rep
Vainau	George	Sales Rep
Bondur	Loui	Sales Rep
Hernandez	Gerard	Sales Rep
Castillo	Pamela	Sales Rep

Below the grid, the Action Output table shows the execution history:

Action	Time	Action	Response	Duration / Fetch Time
select * from employees	4 11:54:59		23 row(s) returned	0.133 sec / 0.000032...
select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep"	5 11:55:50		17 row(s) returned	0.921 sec / 0.000023...

The status bar at the bottom says "Query Completed". The Dock at the bottom shows various application icons.

-- 6 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" y codigo de oficina "1"

```
select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQLWorkbench File Edit View Query Database Server Tools Scripting Help". The main window has tabs for "SQL1" and "Query 1". The left sidebar includes sections for "MANAGEMENT" (Server Status, Client Connections, User and Privileges, Status and System Variables, Data Export, Data Import/Restore), "INSTANCE" (Startup / Shutdown, Server Logs, Options File), and "PERFORMANCE" (Dashboard, Performance Reports, Performance Schema Setup). The central area contains a query editor with the following SQL script:

```
2 • use classicmodels;
3   --- 2 Apellido de todos los empleados
4 • select lastName from employees;
5   --- 3 Apellido, nombre y puesto de todos los empleados
6 • select lastName, firstName, jobTitle from employees;
7   --- 4 todos los datos de cada empleado
8 • select * from employees;
9   --- 5 apellido, nombre, puesto de todos los empleados con puesto "Sales Rep"
10 • select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep";
11   --- 6 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" y codigo de oficina "1"
12 • select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;
13   --- 7 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" y codigo de oficina "1"
```

The "Result Grid" pane shows the results of the last query, which returns 5 rows of employee data:

lastName	firstName	jobTitle	officeCode
Jennings	Leslie	Sales Rep	1
Thompson	Leslie	Sales Rep	1

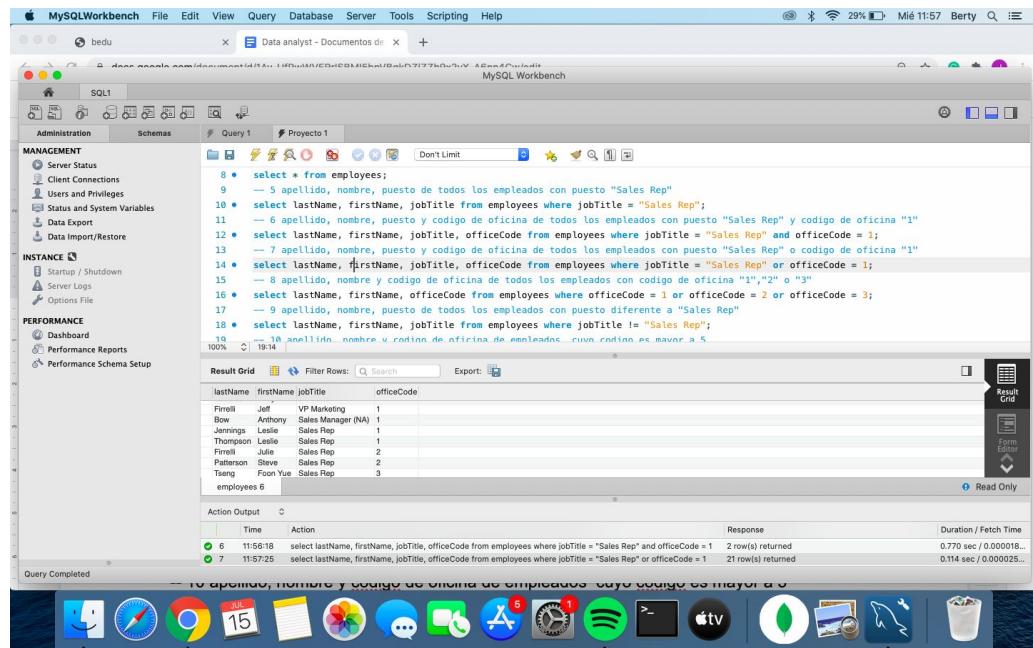
The "Action Output" pane at the bottom shows two recent actions:

Time	Action	Response	Duration / Fetch Time
5 11:55:50	select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep"	17 row(s) returned	0.921 sec / 0.000023...
6 11:56:18	select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1	2 row(s) returned	0.770 sec / 0.000018...

A status bar at the bottom indicates "Query Completed". The system tray shows battery level (29%), signal strength, and the date/time (Mié 11:56).

-- 7 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" o codigo de oficina "1"

```
select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" or officeCode = 1;
```



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The main window displays a SQL query in the Query Grid:

```
8 • select * from employees;
9 -- 5 apellido, nombre, puesto de todos los empleados con puesto "Sales Rep"
10 • select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep";
11 -- 6 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" y codigo de oficina "1"
12 • select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;
13 -- 7 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" o codigo de oficina "1"
14 • select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" or officeCode = 1;
15 -- 8 apellido, nombre y codigo de oficina de todos los empleados con codigo de oficina "1", "2" o "3"
16 • select lastName, firstName, officeCode from employees where officeCode = 1 or officeCode = 2 or officeCode = 3;
17 -- 9 apellido, nombre, puesto de todos los empleados con puesto diferente a "Sales Rep"
18 • select lastName, firstName, jobTitle from employees where jobTitle != "Sales Rep";
19 -- 10 apellido, nombre y codigo de oficina de empleados cuyo codigo es mayor a 3
```

The Result Grid shows the following data:

lastName	firstName	jobTitle	officeCode
Firelli	Jeff	VP Marketing	1
Bow	Anthony	Sales Manager (NA)	1
Jennings	Leslie	Sales Rep	1
Thompson	Leslie	Sales Rep	1
Firelli	Julie	Sales Rep	2
Patterson	Steve	Sales Rep	2
Tseng	Foon Yue	Sales Rep	3

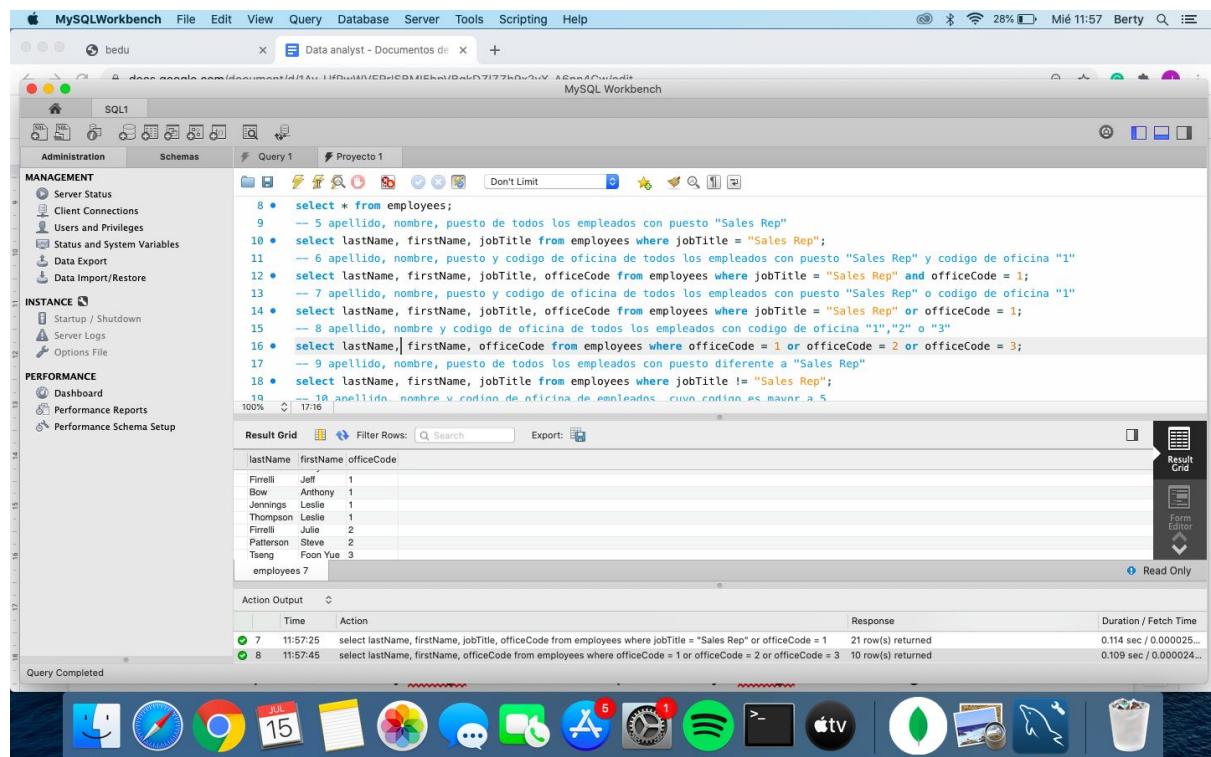
The Action Output pane shows the execution details:

Time	Action	Response	Duration / Fetch Time
6 11:56:18	select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1	2 row(s) returned	0.770 sec / 0.000018...
7 11:57:25	select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" or officeCode = 1	21 row(s) returned	0.114 sec / 0.000025...

-- 8 apellido, nombre y codigo de oficina de todos los empleados con codigo de oficina

"1", "2" o "3"

```
select lastName, firstName, officeCode from employees where officeCode = 1 or officeCode = 2 or officeCode = 3;
```



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The main window displays a SQL query in the Query Grid:

```
8 • select * from employees;
9 -- 5 apellido, nombre, puesto de todos los empleados con puesto "Sales Rep"
10 • select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep";
11 -- 6 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" y codigo de oficina "1"
12 • select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;
13 -- 7 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" o codigo de oficina "1"
14 • select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" or officeCode = 1;
15 -- 8 apellido, nombre y codigo de oficina de todos los empleados con codigo de oficina "1", "2" o "3"
16 • select lastName, firstName, officeCode from employees where officeCode = 1 or officeCode = 2 or officeCode = 3;
17 -- 9 apellido, nombre, puesto de todos los empleados con puesto diferente a "Sales Rep"
18 • select lastName, firstName, jobTitle from employees where jobTitle != "Sales Rep";
19 -- 10 apellido, nombre y codigo de oficina de empleados cuyo codigo es mayor a 3
```

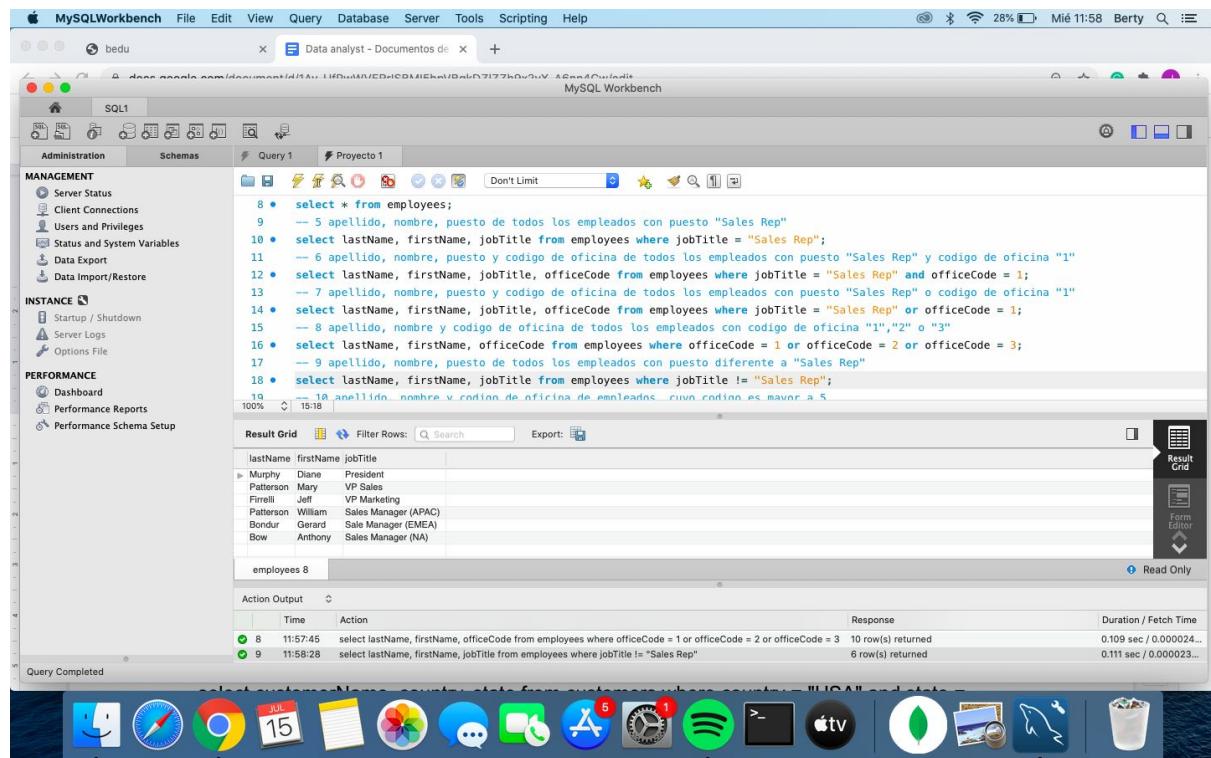
The Result Grid shows the following data:

lastName	firstName	officeCode
Firelli	Jeff	1
Bow	Anthony	1
Jennings	Leslie	1
Thompson	Leslie	1
Firelli	Julie	2
Patterson	Steve	2
Tseng	Foon Yue	3

The Action Output pane shows the execution details:

Time	Action	Response	Duration / Fetch Time
7 11:57:25	select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" or officeCode = 1	21 row(s) returned	0.114 sec / 0.000025...
8 11:57:45	select lastName, firstName, officeCode from employees where officeCode = 1 or officeCode = 2 or officeCode = 3	10 row(s) returned	0.109 sec / 0.000024...

-- 9 apellido, nombre, puesto de todos los empleados con puesto diferente a "Sales Rep"
 select lastName, firstName, jobTitle from employees where jobTitle != "Sales Rep";



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQLWorkbench" and "bedu". The main window has a "SQL1" tab selected. The left sidebar shows "Administration", "Schemas", and "Query 1". The "Query 1" pane contains the following SQL code:

```

8 • select * from employees;
9   -- 5 apellido, nombre, puesto de todos los empleados con puesto "Sales Rep"
10 • select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep";
11   -- 6 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" y codigo de oficina "1"
12 • select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;
13   -- 7 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" o codigo de oficina "1"
14 • select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" or officeCode = 1;
15   -- 8 apellido, nombre y codigo de oficina de todos los empleados con codigo de oficina "1", "2" o "3"
16 • select lastName, firstName, officeCode from employees where officeCode = 1 or officeCode = 2 or officeCode = 3;
17   -- 9 apellido, nombre, puesto de todos los empleados con puesto diferente a "Sales Rep"
18 • select lastName, firstName, jobTitle from employees where jobTitle != "Sales Rep";
19   -- 10 apellido, nombre y codigo de oficina de empleados cuyo codigo es mayor a 5
    
```

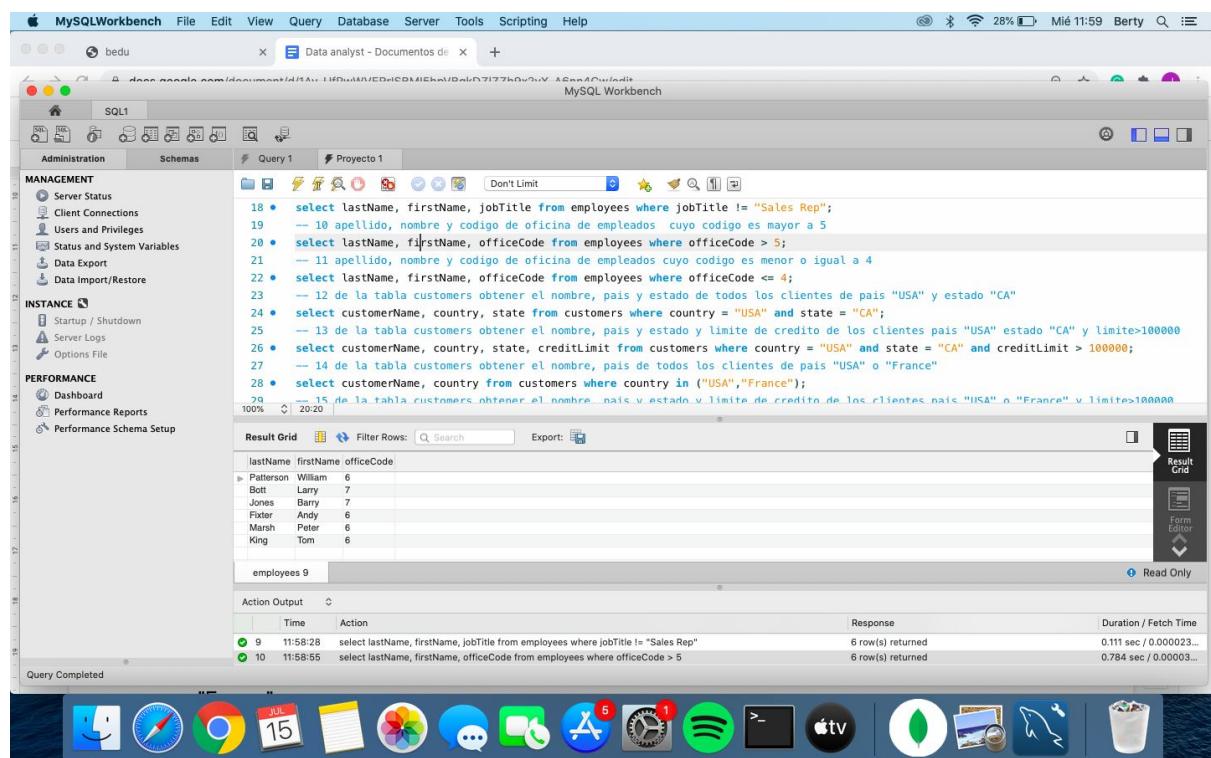
The "Result Grid" pane displays the results of the query:

lastName	firstName	jobTitle
Murphy	Diana	President
Patterson	Mary	VP Sales
Firelli	Jeff	VP Marketing
Patterson	William	Sales Manager (APAC)
Bondur	Gerard	Sale Manager (EMEA)
Bow	Anthony	Sales Manager (NA)

The "Action Output" pane shows the execution details:

Action	Time	Response	Duration / Fetch Time
select lastName, firstName, officeCode from employees where officeCode = 1 or officeCode = 2 or officeCode = 3;	11:57:45	10 row(s) returned	0.109 sec / 0.000024...
select lastName, firstName, jobTitle from employees where jobTitle != "Sales Rep";	11:58:28	6 row(s) returned	0.111 sec / 0.000023...

-- 10 apellido, nombre y codigo de oficina de empleados cuyo codigo es mayor a 5
 select lastName, firstName, officeCode from employees where officeCode > 5;



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQLWorkbench" and "bedu". The main window has a "SQL1" tab selected. The left sidebar shows "Administration", "Schemas", and "Query 1". The "Query 1" pane contains the following SQL code:

```

18 • select lastName, firstName, jobTitle from employees where jobTitle != "Sales Rep";
19   -- 10 apellido, nombre y codigo de oficina de empleados cuyo codigo es mayor a 5
20 • select lastName, firstName, officeCode from employees where officeCode > 5;
21   -- 11 apellido, nombre y codigo de oficina de empleados cuyo codigo es menor o igual a 4
22 • select lastName, firstName, officeCode from employees where officeCode <= 4;
23   -- 12 de la tabla customers obtener el nombre, pais y estado de todos los clientes de pais "USA" y estado "CA"
24 • select customerName, country, state from customers where country = "USA" and state = "CA";
25   -- 13 de la tabla customers obtener el nombre, pais y estado y limite de credito de los clientes pais "USA" estado "CA" y limite>100000
26 • select customerName, country, state, creditLimit from customers where country = "USA" and state = "CA" and creditLimit > 100000;
27   -- 14 de la tabla customers obtener el nombre, pais de todos los clientes de pais "USA" o "France"
28 • select customerName, country from customers where country in ("USA","France");
29   -- 15 de la tabla customers obtener el nombre, pais y estado y limite de credito de los clientes pais "USA" o "France" y limite>100000
    
```

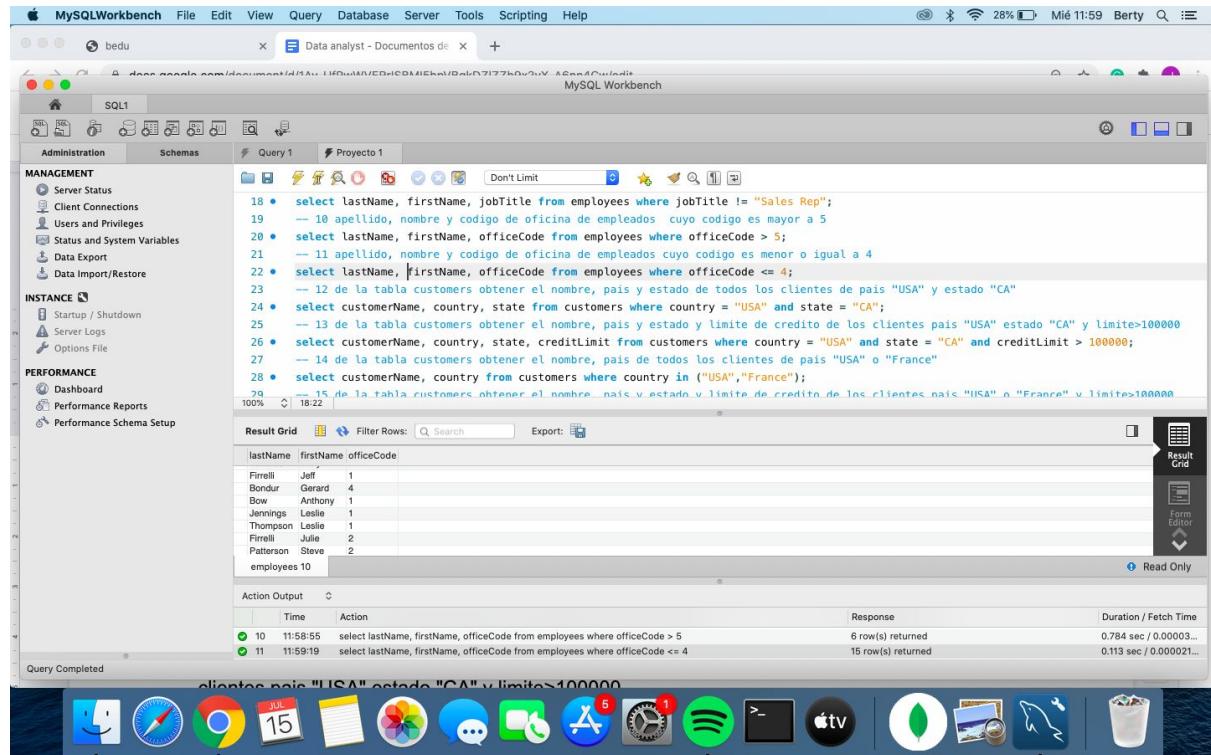
The "Result Grid" pane displays the results of the query:

lastName	firstName	officeCode
Patterson	William	6
Bott	Larry	7
Jones	Barry	7
Fixter	Andy	6
Marsh	Peter	6
King	Tom	6

The "Action Output" pane shows the execution details:

Action	Time	Response	Duration / Fetch Time
select lastName, firstName, jobTitle from employees where jobTitle != "Sales Rep";	11:58:28	6 row(s) returned	0.111 sec / 0.000023...
select lastName, firstName, officeCode from employees where officeCode > 5;	11:58:55	6 row(s) returned	0.784 sec / 0.00003...

-- 11 apellido, nombre y codigo de oficina de empleados cuyo codigo es menor o igual a 4
 select lastName, firstName, officeCode from employees where officeCode <= 4;



```

MySQLWorkbench  File  Edit  View  Query  Database  Server  Tools  Scripting  Help
bedu          Data analyst - Documentos de +  Mié 11:59  Berty  S  MySQL Workbench

SQL1
Administration  Schemas  Query 1  Proyecto 1

MANAGEMENT
  Server Status
  Client Connections
  Users and Privileges
  Status and System Variables
  Data Export
  Data Import/Restore

INSTANCE
  Startup / Shutdown
  Server Logs
  Options File

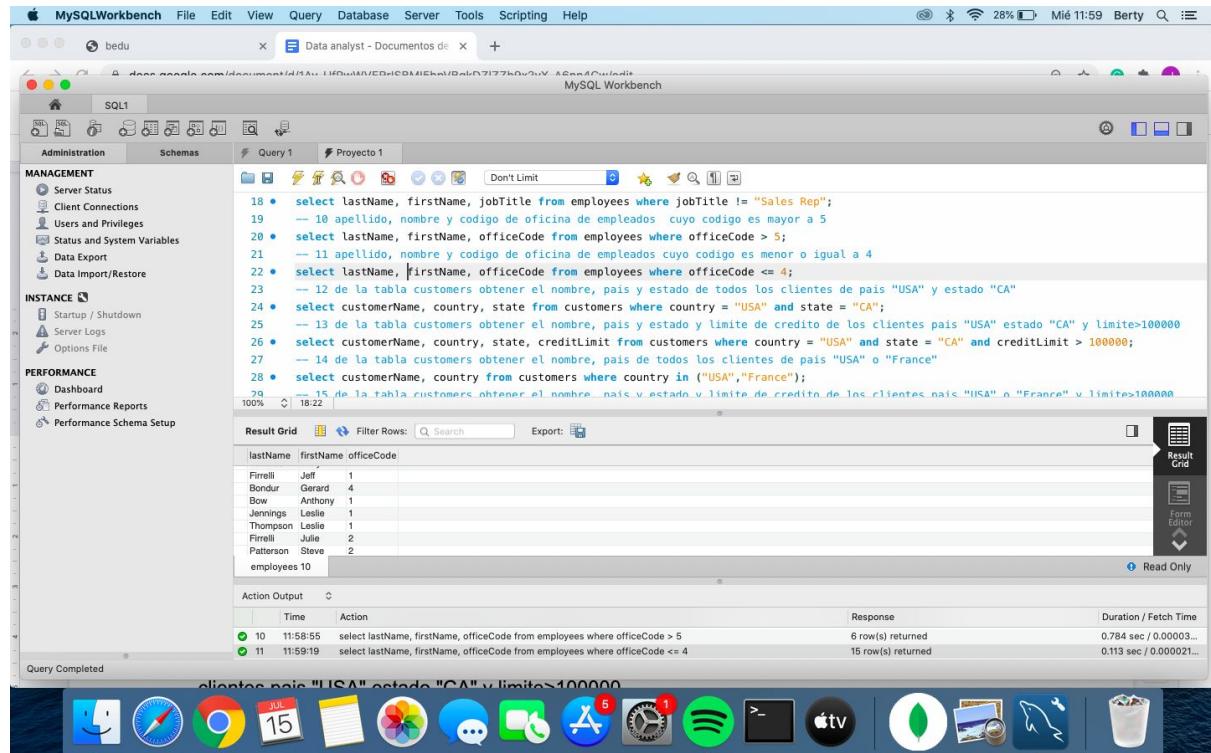
PERFORMANCE
  Dashboard
  Performance Reports
  Performance Schema Setup

Result Grid  Filter Rows: Q Search  Export: 
lastName  firstName  officeCode
Firelli  Jeff  1
Boudur  Gerard  4
Bow  Anthony  1
Jennings  Leslie  1
Thompson  Leslie  1
Firelli  Julie  2
Patterson  Steve  2
employees 10

Action Output  C
Time  Action  Response  Duration / Fetch Time
10  11:58:55  select lastName, firstName, officeCode from employees where officeCode <= 4  15 row(s) returned  0.113 sec / 0.000021...
11  11:59:19  select lastName, firstName, officeCode from employees where officeCode <= 4  6 row(s) returned  0.784 sec / 0.00003...
Query Completed
  
```

-- 12 de la tabla customers obtener el nombre, pais y estado de todos los clientes de pais "USA" y estado "CA"

select customerName, country, state from customers where country = "USA" and state = "CA";



```

MySQLWorkbench  File  Edit  View  Query  Database  Server  Tools  Scripting  Help
bedu          Data analyst - Documentos de +  Mié 11:59  Berty  S  MySQL Workbench

SQL1
Administration  Schemas  Query 1  Proyecto 1

MANAGEMENT
  Server Status
  Client Connections
  Users and Privileges
  Status and System Variables
  Data Export
  Data Import/Restore

INSTANCE
  Startup / Shutdown
  Server Logs
  Options File

PERFORMANCE
  Dashboard
  Performance Reports
  Performance Schema Setup

Result Grid  Filter Rows: Q Search  Export: 
lastName  firstName  officeCode
Firelli  Jeff  1
Boudur  Gerard  4
Bow  Anthony  1
Jennings  Leslie  1
Thompson  Leslie  1
Firelli  Julie  2
Patterson  Steve  2
employees 10

Action Output  C
Time  Action  Response  Duration / Fetch Time
10  11:58:55  select lastName, firstName, officeCode from employees where officeCode <= 4  15 row(s) returned  0.784 sec / 0.00003...
11  11:59:19  select lastName, firstName, officeCode from employees where officeCode <= 4  6 row(s) returned  0.113 sec / 0.000021...
Query Completed
  
```

-- 13 de la tabla customers obtener el nombre, pais y estado y limite de credito de los clientes pais "USA" estado "CA" y limite>100000

select customerName, country, state, creditLimit from customers where country = "USA" and state = "CA" and creditLimit > 100000;

```

18 • select lastName, firstName, jobTitle from employees where jobTitle != "Sales Rep";
19 -- 10 apellido, nombre y codigo de oficina de empleados cuyo codigo es mayor a 5
20 • select lastName, firstName, officeCode from employees where officeCode > 5;
21 -- 11 apellido, nombre y codigo de oficina de empleados cuyo codigo es menor o igual a 4
22 • select lastName, firstName, officeCode from employees where officeCode <= 4;
23 -- 12 de la tabla customers obtener el nombre, pais y estado de todos los clientes de pais "USA" y estado "CA"
24 • select customerName, country, state from customers where country = "USA" and state = "CA";
25 -- 13 de la tabla customers obtener el nombre, pais y estado y limite de credito de los clientes pais "USA" estado "CA" y limite>100000
26 • select customerName, country, state, creditLimit from customers where country = "USA" and state = "CA" and creditLimit > 100000;
27 -- 14 de la tabla customers obtener el nombre, pais de todos los clientes de pais "USA" o "France"
28 • select customerName, country from customers where country in ("USA","France");
29 -- 15 de la tabla customers obtener el nombre, pais y estado y limite de credito de los clientes pais "USA" o "France" v. limite>100000
100% 18:26

```

customerName	country	state	creditLimit
Mini Gifts Distributors Ltd.	USA	CA	210500.00
Collectable Mini Designs Co.	USA	CA	106000.00
Corporate Gift Ideas Co.	USA	CA	105000.00

-- 14 de la tabla customers obtener el nombre, pais de todos los clientes de pais "USA" o "France"

select customerName, country from customers where country in ("USA","France");

```

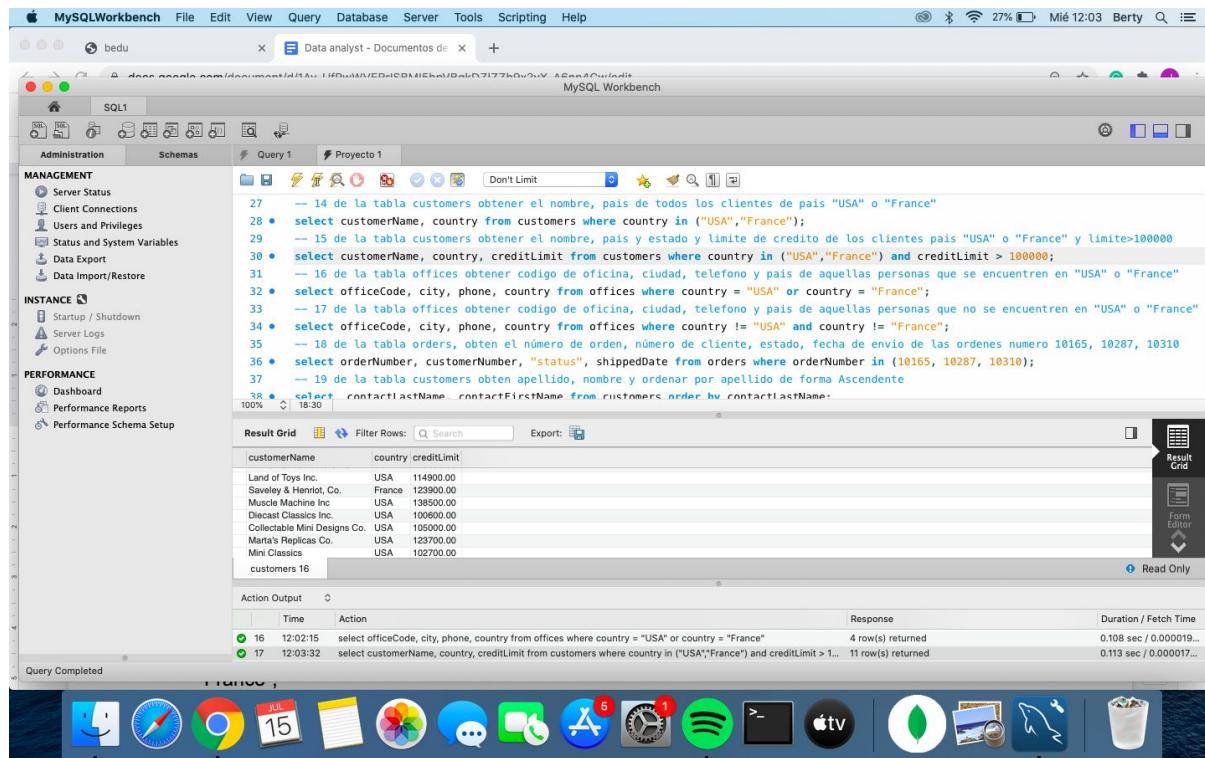
18 • select lastName, firstName, jobTitle from employees where jobTitle != "Sales Rep";
19 -- 10 apellido, nombre y codigo de oficina de empleados cuyo codigo es mayor a 5
20 • select lastName, firstName, officeCode from employees where officeCode > 5;
21 -- 11 apellido, nombre y codigo de oficina de empleados cuyo codigo es menor o igual a 4
22 • select lastName, firstName, officeCode from employees where officeCode <= 4;
23 -- 12 de la tabla customers obtener el nombre, pais y estado de todos los clientes de pais "USA" y estado "CA"
24 • select customerName, country, state from customers where country = "USA" and state = "CA";
25 -- 13 de la tabla customers obtener el nombre, pais y estado y limite de credito de los clientes pais "USA" estado "CA" y limite>100000
26 • select customerName, country, state, creditLimit from customers where country = "USA" and state = "CA" and creditLimit > 100000;
27 -- 14 de la tabla customers obtener el nombre, pais de todos los clientes de pais "USA" o "France"
28 • select customerName, country from customers where country in ("USA","France");
29 -- 15 de la tabla customers obtener el nombre, pais y estado y limite de credito de los clientes pais "USA" o "France" v. limite>100000
100% 28:28

```

customerName	country
La Rochelle Gifts	France
Mini Gifts Distributors Ltd.	USA
Mini World Ltd.	USA
Land of Toys Inc.	USA
Saviley & Henriot, Co.	France
Muscle Machine Inc.	USA
Diecast Classics Inc.	USA

-- 15 de la tabla customers obtener el nombre, pais y estado y limite de credito de los clientes pais "USA" o "France" y limite>100000

select customerName, country, creditLimit from customers where country in ("USA","France") and creditLimit > 100000;



```

MySQLWorkbench  File Edit View Query Database Server Tools Scripting Help
bedu          Data analyst - Documentos de + 
MySQL Workbench

SQL1
Query 1  Proyecto 1

MANAGEMENT
ADMINISTRATION
INSTANCE
PERFORMANCE

Result Grid Filter Rows: Q Search Export: 
customerName country creditLimit
Land of Toys Inc. USA 114900.00
Saviley & Henrot, Co. France 123900.00
Muscle Machine Inc. USA 138500.00
Discast Classics Inc. USA 109000.00
Collectors Mini Designs Co. USA 109000.00
Marta's Replicas Co. USA 123700.00
Min Classics USA 102700.00
customers 16

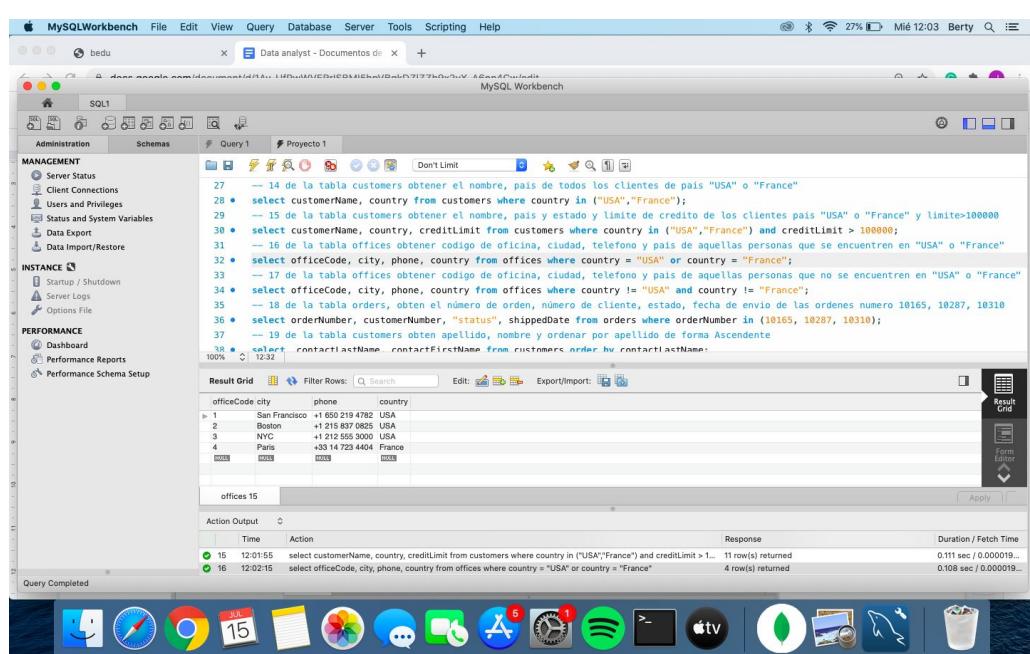
Action Output
Time Action Response Duration / Fetch Time
16 12:02:15 select officeCode, city, phone, country from offices where country = "USA" or country = "France" 4 row(s) returned 0.108 sec / 0.000019...
17 12:03:32 select customerName, country, creditLimit from customers where country in ("USA","France") and creditLimit > 100000 11 row(s) returned 0.113 sec / 0.000017...

Query Completed

```

-- 16 de la tabla offices obtener codigo de oficina, ciudad, telefono y pais de aquellas personas que se encuentren en "USA" o "France"

select officeCode, city, phone, country from offices where country = "USA" or country = "France";



```

MySQLWorkbench  File Edit View Query Database Server Tools Scripting Help
bedu          Data analyst - Documentos de + 
MySQL Workbench

SQL1
Query 1  Proyecto 1

MANAGEMENT
ADMINISTRATION
INSTANCE
PERFORMANCE

Result Grid Filter Rows: Q Search Edit: Export: 
officeCode city phone country
1 San Francisco +1 650 519 1782 USA
2 Seattle +1 206 553 0000 USA
3 NYC +1 212 555 3000 USA
4 Paris +33 14 723 4404 France
offices 15

Action Output
Time Action Response Duration / Fetch Time
15 12:01:55 select customerName, country, creditLimit from customers where country in ("USA","France") and creditLimit > 1... 11 row(s) returned 0.111 sec / 0.000019...
16 12:02:15 select officeCode, city, phone, country from offices where country = "USA" or country = "France" 4 row(s) returned 0.108 sec / 0.000019...

Query Completed

```

-- 17 de la tabla offices obtener codigo de oficina, ciudad, telefono y pais de aquellas personas que no se encuentren en "USA" o "France"

select officeCode, city, phone, country from offices where country != "USA" and country != "France";

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The SQL editor window displays the following query and its execution results:

```
27 -- 14 de la tabla customers obtener el nombre, pais de todos los clientes de pais "USA" o "France"
28 • select customerName, country from customers where country in ("USA","France");
29 -- 15 de la tabla customers obtener el nombre, pais y estado y limite de credito de los clientes pais "USA" o "France" y limite>100000
30 • select customerName, country, creditLimit from customers where country in ("USA","France") and creditLimit > 100000;
31 -- 16 de la tabla offices obtener codigo de oficina, ciudad, telefono y pais de aquellas personas que se encuentren en "USA" o "France"
32 • select officeCode, city, phone, country from offices where country = "USA" or country = "France";
33 -- 17 de la tabla offices obtener codigo de oficina, ciudad, telefono y pais de aquellas personas que no se encuentren en "USA" o "France"
34 • select officeCode, city, phone, country from offices where country != "USA" and country != "France";
35 -- 18 de la tabla orders, obtén el número de orden, número de cliente, estado, fecha de envío de las órdenes numero 10165, 10287, 10310
36 • select orderNumber, customerNumber, "status", shippedDate from orders where orderNumber in (10165, 10287, 10310);
37 -- 19 de la tabla customers obtener apellido, nombre y ordenar por apellido de forma Ascendente
38 • select contactLastName, contactFirstName from customers order by contactLastName;
```

The results grid shows three rows of data from the 'offices' table:

officeCode	city	phone	country
5	Tokyo	+81 33 224 5000	Japan
6	Sydney	+61 2 9264 2451	Australia
7	London	+44 20 7877 2041	UK

The status bar at the bottom indicates "Query Completed".

-- 18 de la tabla orders, obtén el número de orden, número de cliente, estado, fecha de envío de las órdenes numero 10165, 10287, 10310

select orderNumber, customerNumber, "status", shippedDate from orders where orderNumber in (10165, 10287, 10310);

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The SQL editor window displays the following query and its execution results:

```
27 -- 14 de la tabla customers obtener el nombre, pais de todos los clientes de pais "USA" o "France"
28 • select customerName, country from customers where country in ("USA","France");
29 -- 15 de la tabla customers obtener el nombre, pais y estado y limite de credito de los clientes pais "USA" o "France" y limite>100000
30 • select customerName, country, creditLimit from customers where country in ("USA","France") and creditLimit > 100000;
31 -- 16 de la tabla offices obtener codigo de oficina, ciudad, telefono y pais de aquellas personas que se encuentren en "USA" o "France"
32 • select officeCode, city, phone, country from offices where country = "USA" or country = "France";
33 -- 17 de la tabla offices obtener codigo de oficina, ciudad, telefono y pais de aquellas personas que no se encuentren en "USA" o "France"
34 • select officeCode, city, phone, country from offices where country != "USA" and country != "France";
35 -- 18 de la tabla orders, obtén el número de orden, número de cliente, estado, fecha de envío de las órdenes numero 10165, 10287, 10310
36 • select orderNumber, customerNumber, "status", shippedDate from orders where orderNumber in (10165, 10287, 10310);
37 -- 19 de la tabla customers obtener apellido, nombre y ordenar por apellido de forma Ascendente
38 • select contactLastName, contactFirstName from customers order by contactLastName;
```

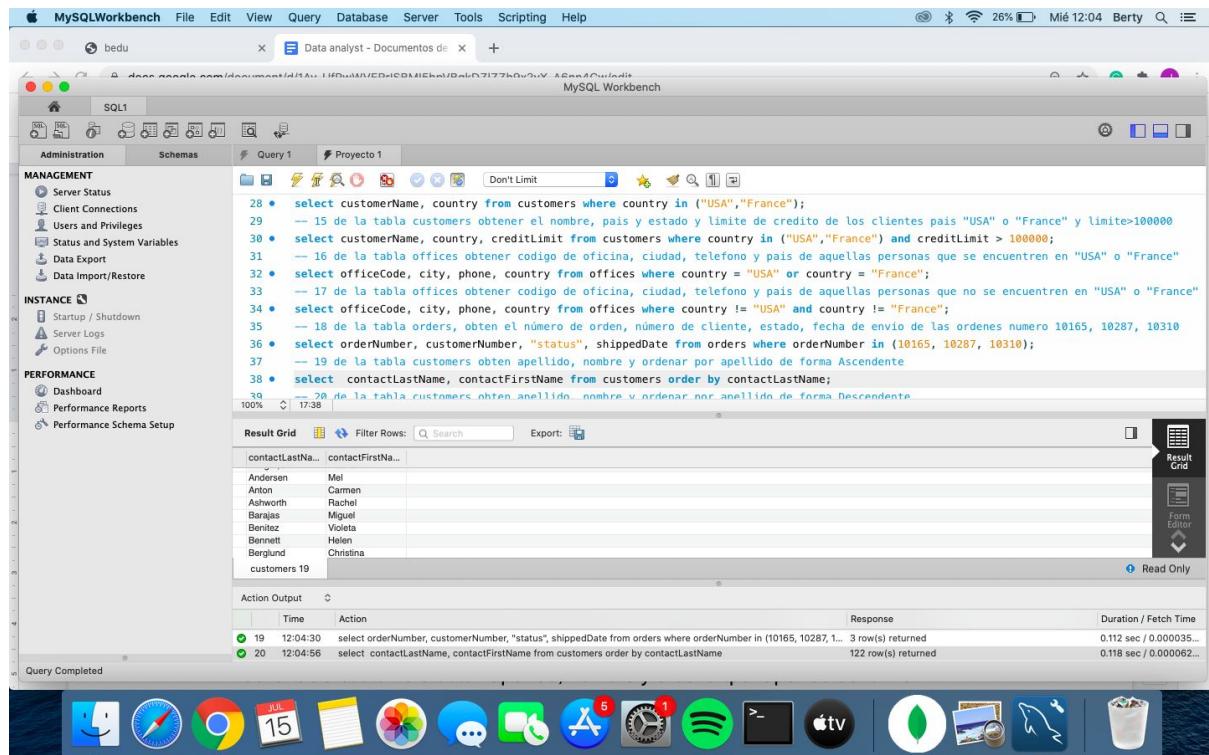
The results grid shows three rows of data from the 'orders' table:

orderNumber	customerNumber	"status"	shippedDate
10165	100017...	In Progress	2003-12-17 00:00:00
10287	100018...	In Progress	2003-12-17 00:00:00
10310	100019...	In Progress	2003-12-17 00:00:00

The status bar at the bottom indicates "Query Completed".

-- 19 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Ascendente

```
select contactLastName, contactFirstName from customers order by contactLastName;
```



```
28 • select customerName, country from customers where country in ("USA","France");
29   -- 15 de la tabla customers obtener el nombre, país y estado y límite de crédito de los clientes país "USA" o "France" y límite>100000;
30 • select customerName, country, creditLimit from customers where country in ("USA","France") and creditLimit > 100000;
31   -- 16 de la tabla offices obtener código de oficina, ciudad, teléfono y país de aquellas personas que se encuentren en "USA" o "France"
32 • select officeCode, city, phone, country from offices where country = "USA" or country = "France";
33   -- 17 de la tabla offices obtener código de oficina, ciudad, teléfono y país de aquellas personas que no se encuentren en "USA" o "France"
34 • select officeCode, city, phone, country from offices where country != "USA" and country != "France";
35   -- 18 de la tabla orders, obtén el número de orden, número de cliente, estado, fecha de envío de las órdenes numero 10165, 10287, 10310
36 • select orderNumber, customerNumber, "status", shippedDate from orders where orderNumber in (10165, 10287, 10310);
37   -- 19 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Ascendente
38 • select contactLastName, contactFirstName from customers order by contactLastName;
39   -- 20 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Descendente
40   -- 21 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Descendente v. luces por nombre en forma ascendente
100% 17:38
```

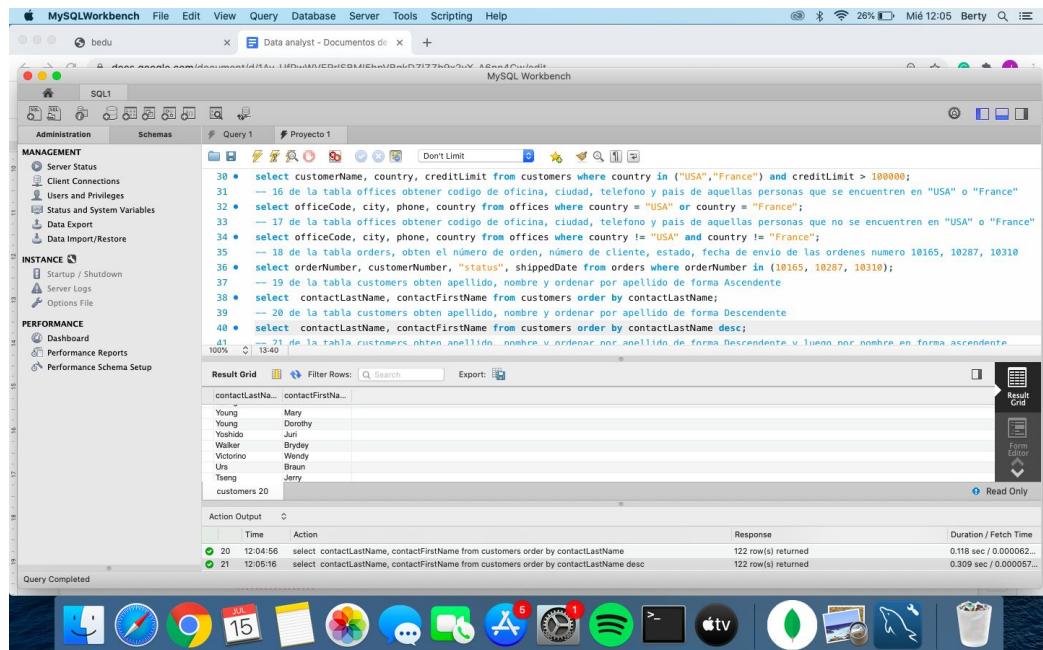
contactLastName	contactFirstName
Andersen	Mel
Anton	Carmen
Ashworth	Rachel
Barajas	Miguel
Bennet	Violeta
Bennett	Helen
Berglund	Christina

Action Output

Time	Action	Response	Duration / Fetch Time
12:04:30	select orderNumber, customerNumber, "status", shippedDate from orders where orderNumber in (10165, 10287, 10310)	3 row(s) returned	0.112 sec / 0.000035...
12:04:56	select contactLastName, contactFirstName from customers order by contactLastName	122 row(s) returned	0.118 sec / 0.000062...

-- 20 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Descendente

```
select contactLastName, contactFirstName from customers order by contactLastName desc;
```



```
30 • select customerName, country, creditLimit from customers where country in ("USA","France") and creditLimit > 100000;
31   -- 16 de la tabla offices obtener código de oficina, ciudad, teléfono y país de aquellas personas que se encuentren en "USA" o "France"
32 • select officeCode, city, phone, country from offices where country = "USA" or country = "France";
33   -- 17 de la tabla offices obtener código de oficina, ciudad, teléfono y país de aquellas personas que no se encuentren en "USA" o "France"
34 • select officeCode, city, phone, country from offices where country != "USA" and country != "France";
35   -- 18 de la tabla orders, obtén el número de orden, número de cliente, estado, fecha de envío de las órdenes numero 10165, 10287, 10310
36 • select orderNumber, customerNumber, "status", shippedDate from orders where orderNumber in (10165, 10287, 10310);
37   -- 19 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Ascendente
38 • select contactLastName, contactFirstName from customers order by contactLastName;
39   -- 20 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Descendente
40 • select contactLastName, contactFirstName from customers order by contactLastName desc;
41   -- 21 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Descendente v. luces por nombre en forma ascendente
100% 13:40
```

contactLastName	contactFirstName
Young	Mary
Young	Dorothy
Yoshida	Juri
Walker	Brydey
Wentzler	Wendy
Uro	Brian
Tseng	Jerry

Action Output

Time	Action	Response	Duration / Fetch Time
12:04:56	select contactLastName, contactFirstName from customers order by contactLastName	122 row(s) returned	0.118 sec / 0.000062...
12:05:16	select contactLastName, contactFirstName from customers order by contactLastName desc	122 row(s) returned	0.309 sec / 0.000057...

-- 21 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Descendente y luego por nombre en forma ascendente

```
select contactLastName, contactFirstName from customers order by contactLastName desc;
select contactLastName, contactFirstName from customers order by contactFirstName asc;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The main window displays three SQL statements:

```

1 -- 21 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Descendente y luego por nombre en forma ascendente
2 • everything, if there is no selection FirstName from customers order by contactLastName desc;
3 • select contactLastName, contactFirstName from customers order by contactFirstName asc;

```

The Result Grid shows the following data:

contactLastNa...	contactFirstNa...
Young	Mary
Young	Dorothy
Yoshido	Juri
Walker	Bryony
Victorino	Wendy
Urs	Braun

The Action Output table shows two recent actions:

Action	Time	Response	Duration / Fetch Time
select contactLastName, contactFirstName from customers order by contactLastName desc	26 12:07:05	122 row(s) returned	0.111 sec / 0.000058...
select contactLastName, contactFirstName from customers order by contactFirstName asc	27 12:07:06	122 row(s) returned	0.113 sec / 0.000032...

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The main window displays three SQL statements:

```

1 -- 21 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Descendente y luego por nombre en forma ascendente
2 • select contactLastName, contactFirstName from customers order by contactLastName desc;
3 • select contactLastName, contactFirstName from customers order by contactFirstName asc;

```

The Result Grid shows the following data:

contactLastNa...	contactFirstNa...
Camino	Alejandra
Fauer	Alexander
Semenov	Alexander
Nelson	Allen
Brown	Ann
O'Hara	Anna

The Action Output table shows two recent actions:

Action	Time	Response	Duration / Fetch Time
select contactLastName, contactFirstName from customers order by contactLastName desc	28 12:07:10	122 row(s) returned	0.111 sec / 0.000058...
select contactLastName, contactFirstName from customers order by contactFirstName asc	29 12:07:10	122 row(s) returned	0.113 sec / 0.000032...

-- 22 limite mayor 5

```
select customerNumber, customerName, creditLimit from customers where creditLimit >0  
order by creditLimit desc limit 5;
```

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

```
35 -- 18 de la tabla orders, obten el número de orden, número de cliente, estado, fecha de envío de las órdenes numero 10165, 10287, 10310;  
36 • select orderNumber, customerNumber, "status", shippedDate from orders where orderNumber in (10165, 10287, 10310);  
37 -- 19 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Ascendente  
38 • select contactLastName, contactFirstName from customers order by contactLastName;  
39 -- 20 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Descendente  
40 • select contactLastName, contactFirstName from customers order by contactLastName desc;  
41 -- 21 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Descendente y luego por nombre en forma ascendente  
42 • select contactLastName, contactFirstName from customers order by contactLastName desc;  
43 • select contactLastName, contactFirstName from customers order by contactFirstName asc;  
44 -- 22 límite mayor 5  
45 • select customerNumber, customerName, creditLimit from customers where creditLimit >0 order by creditLimit desc limit 5;  
46 -- 23 límite menor 5
```

The Result Grid shows the following data:

customerNumber	customerName	creditLimit
141	Euro+ Shopping Channel	227600.00
124	Mini Gifts Distributors Ltd.	210500.00
298	Vida Sport, Ltd	141300.00
151	Muscle Machine Inc	138500.00
187	AV Stores, Co.	136800.00

The Action Output pane shows two queries:

Time	Action	Response	Duration / Fetch Time
29 12:07:10	select contactLastName, contactFirstName from customers order by contactFirstName asc	122 row(s) returned	0.119 sec / 0.000052...
30 12:08:52	select customerNumber, customerName, creditLimit from customers where creditLimit >0 order by creditLimit de...	5 row(s) returned	0.110 sec / 0.000018...

-- 23 límite menor 5

```
select customerNumber, customerName, creditLimit from customers where creditLimit >0  
order by creditLimit asc limit 5;
```

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

```
35 -- 18 de la tabla orders, obten el número de orden, número de cliente, estado, fecha de envío de las órdenes numero 10165, 10287, 10310;  
36 • select orderNumber, customerNumber, "status", shippedDate from orders where orderNumber in (10165, 10287, 10310);  
37 -- 19 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Ascendente  
38 • select contactLastName, contactFirstName from customers order by contactLastName;  
39 -- 20 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Descendente  
40 • select contactLastName, contactFirstName from customers order by contactLastName desc;  
41 -- 21 de la tabla customers obtén apellido, nombre y ordenar por apellido de forma Descendente y luego por nombre en forma ascendente  
42 • select contactLastName, contactFirstName from customers order by contactLastName desc;  
43 • select contactLastName, contactFirstName from customers order by contactFirstName asc;  
44 -- 22 límite mayor 5  
45 • select customerNumber, customerName, creditLimit from customers where creditLimit >0 order by creditLimit desc limit 5;  
46 -- 23 límite menor 5
```

The Result Grid shows the following data:

customerNumber	customerName	creditLimit
141	Euro+ Shopping Channel	227600.00
124	Mini Gifts Distributors Ltd.	210500.00
298	Vida Sport, Ltd	141300.00
151	Muscle Machine Inc	138500.00
187	AV Stores, Co.	136800.00

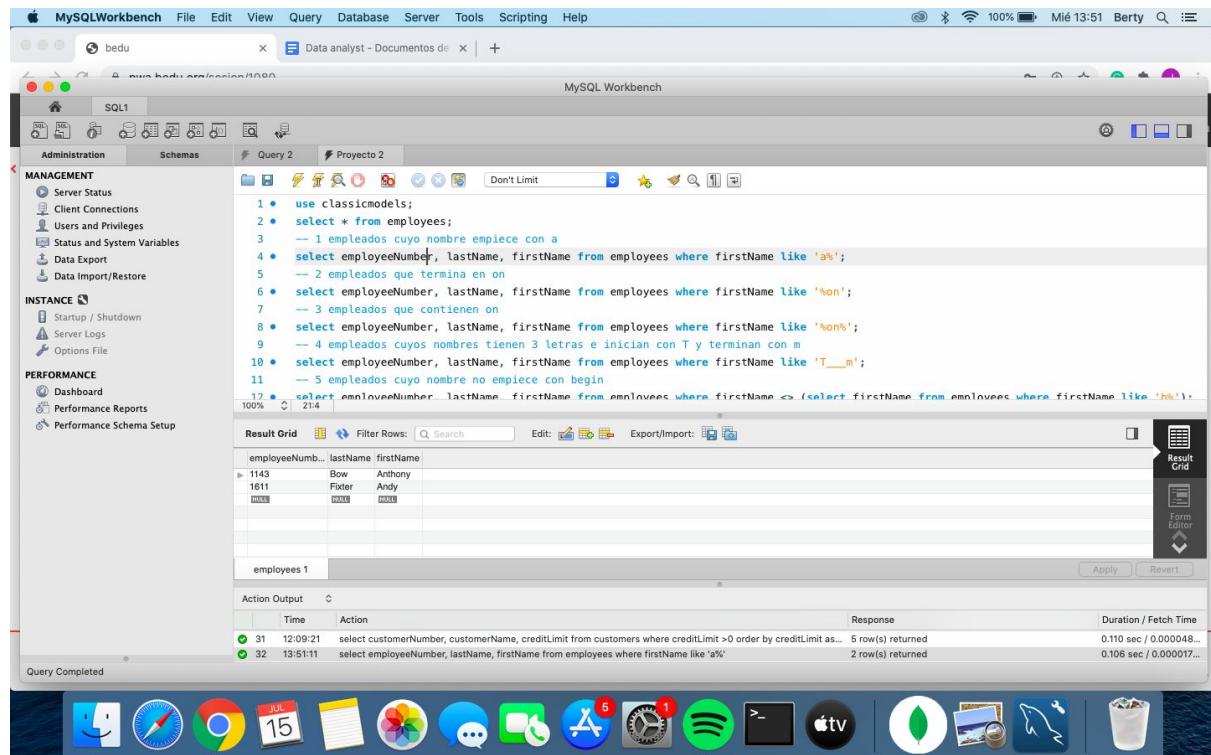
The Action Output pane shows two queries:

Time	Action	Response	Duration / Fetch Time
29 12:07:10	select contactLastName, contactFirstName from customers order by contactFirstName asc	122 row(s) returned	0.119 sec / 0.000052...
30 12:08:52	select customerNumber, customerName, creditLimit from customers where creditLimit >0 order by creditLimit de...	5 row(s) returned	0.110 sec / 0.000018...

Sesión 02 Agrupaciones y subconsultas

Proyecto 2

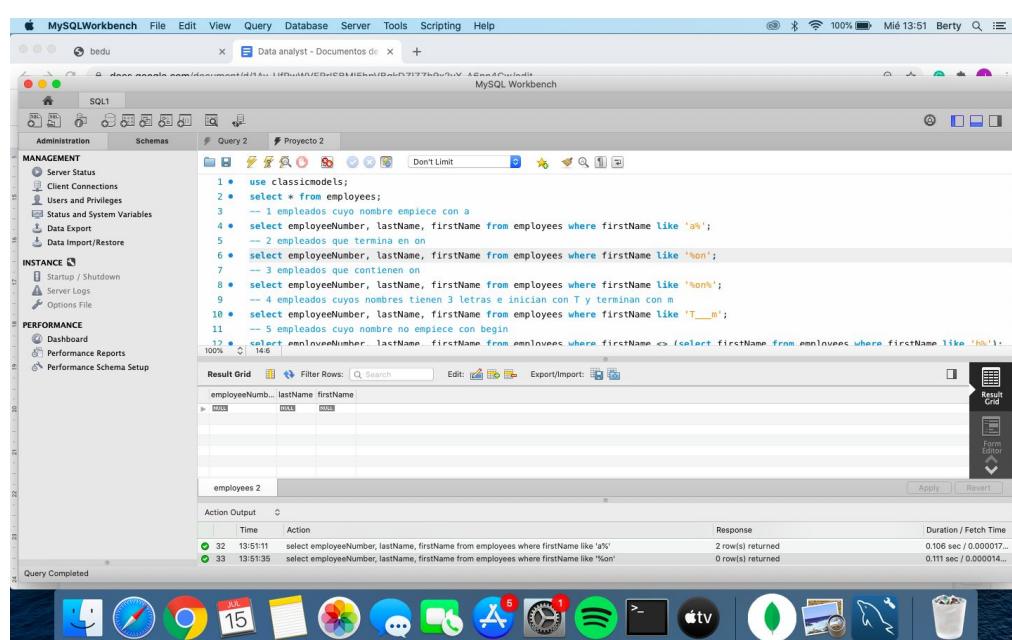
-- 1 empleados cuyo nombre empieza con a
select employeeNumber, lastName, firstName from employees where firstName like 'a%';



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The left sidebar contains navigation links for Management, Instance, and Performance. The main area has tabs for Administration, Schemas, Query 2, and Proyecto 2. The SQL editor tab is active, displaying a numbered list of SQL queries related to employees. The Result Grid tab shows the results of the last query, which retrieves employees whose first name starts with 'a'. The results table has columns: employeeNumber, lastName, and firstName. Two rows are shown: Bow, Anthony and 1611, Fixter, Andy. Below the result grid is an Action Output table showing two recent actions: selecting customer numbers and selecting employees starting with 'a'.

Action	Time	Action	Response	Duration / Fetch Time
31	12:09:21	select customerNumber, customerName, creditLimit from customers where creditLimit > 0 order by creditLimit asc...	5 row(s) returned	0.110 sec / 0.000048...
32	13:51:11	select employeeNumber, lastName, firstName from employees where firstName like 'a%'	2 row(s) returned	0.106 sec / 0.000017...

-- 2 empleados que termina en on
select employeeNumber, lastName, firstName from employees where firstName like '%on';



This screenshot is similar to the previous one, showing the MySQL Workbench interface on a Mac OS X desktop. The left sidebar and tabs are identical. The SQL editor tab shows the same numbered list of queries. The Result Grid tab displays the results of the query for employees ending with 'on'. The results table has columns: employeeNumber, lastName, and firstName. No rows are shown in this specific screenshot, although the query is listed in the history.

Action	Time	Action	Response	Duration / Fetch Time
32	13:51:11	select employeeNumber, lastName, firstName from employees where firstName like 'a%'	2 row(s) returned	0.106 sec / 0.000017...
33	13:51:35	select employeeNumber, lastName, firstName from employees where firstName like '%on'	0 row(s) returned	0.111 sec / 0.000014...

-- 3 empleados que contienen on

select employeeNumber, lastName, firstName from employees where firstName like '%on%';

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The main window displays a SQL editor with the following code:

```
1 • use经典models;
2 • select * from employees;
3 -- 1 empleados cuyo nombre empieza con a
4 • select employeeNumber, lastName, firstName from employees where firstName like 'a%';
5 -- 2 empleados que terminan en on
6 • select employeeNumber, lastName, firstName from employees where firstName like '%on';
7 -- 3 empleados que contienen on
8 • select employeeNumber, lastName, firstName from employees where firstName like '%on%';
9 -- 4 empleados cuyos nombres tienen 3 letras e inician con T y terminan con m
10 • select employeeNumber, lastName, firstName from employees where firstName like 'T__m';
11 -- 5 empleados cuyo nombre no empieza con begin
12 • select employeeNumber, lastName, firstName from employees where firstName > (select firstName from employees where firstName like 'Na%').
```

The Result Grid shows the following data:

employeeNumber	lastName	firstName

The Action Output table shows two rows of activity:

Action	Time	Response	Duration / Fetch Time
32	13:51:11	select employeeNumber, lastName, firstName from employees where firstName like 'a%'	2 row(s) returned 0.106 sec / 0.000017...
33	13:51:35	select employeeNumber, lastName, firstName from employees where firstName like '%on%'	0 row(s) returned 0.111 sec / 0.000014...

-- 4 empleados cuyos nombres tienen 3 letras e inician con T y terminan con m

select employeeNumber, lastName, firstName from employees where firstName like 'T_m';

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The main window displays a SQL editor with the following code:

```
1 • use经典models;
2 • select * from employees;
3 -- 1 empleados cuyo nombre empieza con a
4 • select employeeNumber, lastName, firstName from employees where firstName like 'a%';
5 -- 2 empleados que terminan en on
6 • select employeeNumber, lastName, firstName from employees where firstName like '%on';
7 -- 3 empleados que contienen on
8 • select employeeNumber, lastName, firstName from employees where firstName like '%on%';
9 -- 4 empleados cuyos nombres tienen 3 letras e inician con T y terminan con m
10 • select employeeNumber, lastName, firstName from employees where firstName like 'T__m';
11 -- 5 empleados cuyo nombre no empieza con begin
12 • select employeeNumber, lastName, firstName from employees where firstName > (select firstName from employees where firstName like 'Na%').
```

The Result Grid shows the following data:

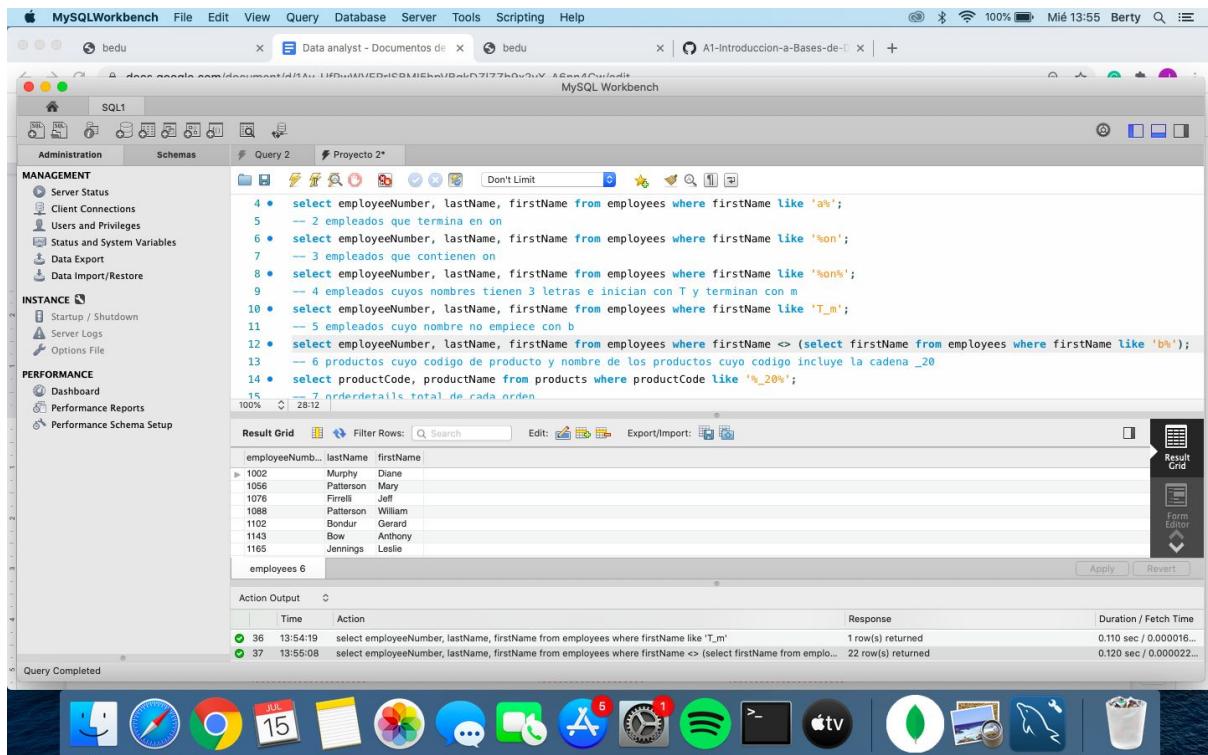
employeeNumber	lastName	firstName
1619	King	Tom

The Action Output table shows two rows of activity:

Action	Time	Response	Duration / Fetch Time
35	13:52:59	select employeeNumber, lastName, firstName from employees where firstName like 'T__m'	0 row(s) returned 0.110 sec / 0.000016...
36	13:54:19	select employeeNumber, lastName, firstName from employees where firstName like 'T_m'	1 row(s) returned 0.110 sec / 0.000016...

-- 5 empleados cuyo nombre no empiece con b

```
select employeeNumber, lastName, firstName from employees where firstName <> (select firstName from employees where firstName like 'b%');
```



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The SQL editor contains the following query:

```
4 • select employeeNumber, lastName, firstName from employees where firstName like 'a%';
5   -- 2 empleados que termina en on
6 • select employeeNumber, lastName, firstName from employees where firstName like '%on';
7   -- 3 empleados que contienen on
8 • select employeeNumber, lastName, firstName from employees where firstName like '%on%';
9   -- 4 empleados cuyos nombres tienen 3 letras e inician con T y terminan con m
10 • select employeeNumber, lastName, firstName from employees where firstName like 'T_m';
11   -- 5 empleados cuyo nombre no empieza con b
12 • select employeeNumber, lastName, firstName from employees where firstName <> (select firstName from employees where firstName like 'b%');
13   -- 6 productos cuyo codigo de producto y nombre de los productos cuyo codigo incluye la cadena _20
14 • select productCode, productName from products where productCode like '%_20%';
15   -- 7 ordenes totales de cada orden
```

The Result Grid shows the following data:

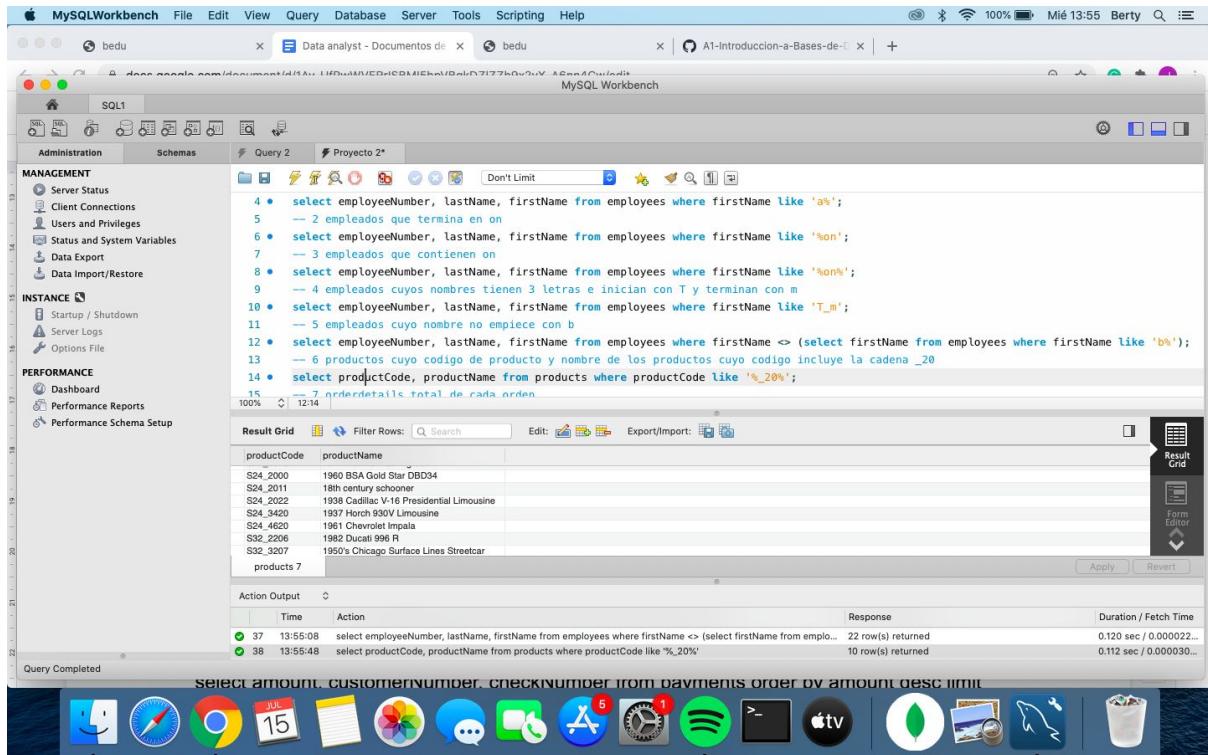
employeeNumber	lastName	firstName
1002	Murphy	Diane
1056	Patterson	Mary
1076	Firelli	Jeff
1089	Patterson	William
1102	Bondur	Gerard
1143	Bow	Anthony
1165	Jennings	Leslie

Action Output shows two log entries:

Time	Action	Response	Duration / Fetch Time
13:54:19	select employeeNumber, lastName, firstName from employees where firstName like 'T_m'	1 row(s) returned	0.110 sec / 0.000016...
13:55:08	select employeeNumber, lastName, firstName from employees where firstName <> (select firstName from employees where firstName like 'b%')	22 row(s) returned	0.120 sec / 0.000022...

-- 6 productos cuyo codigo de producto y nombre de los productos cuyo codigo incluye la cadena _20

```
select productCode, productName from products where productCode like '%_20%';
```



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The SQL editor contains the following query:

```
4 • select employeeNumber, lastName, firstName from employees where firstName like 'a%';
5   -- 2 empleados que termina en on
6 • select employeeNumber, lastName, firstName from employees where firstName like '%on';
7   -- 3 empleados que contienen on
8 • select employeeNumber, lastName, firstName from employees where firstName like '%on%';
9   -- 4 empleados cuyos nombres tienen 3 letras e inician con T y terminan con m
10 • select employeeNumber, lastName, firstName from employees where firstName like 'T_m';
11   -- 5 empleados cuyo nombre no empieza con b
12 • select employeeNumber, lastName, firstName from employees where firstName <> (select firstName from employees where firstName like 'b%');
13   -- 6 productos cuyo codigo de producto y nombre de los productos cuyo codigo incluye la cadena _20
14 • select productCode, productName from products where productCode like '%_20%';
15   -- 7 ordenes totales de cada orden
```

The Result Grid shows the following data:

productCode	productName
S24_2000	1960 BSA Gold Star DBD34
S24_2011	18th century schooner
S24_2022	1938 Cadillac V-16 Presidential Limousine
S24_3420	1937 Horch 930V Limousine
S24_4620	1961 Chevrolet Impala
S32_2206	1982 Ducati 996 R
S32_3207	1950's Chicago Surface Lines Streetcar

Action Output shows two log entries:

Time	Action	Response	Duration / Fetch Time
13:55:08	select employeeNumber, lastName, firstName from employees where firstName <> (select firstName from employees where firstName like 'b%')	22 row(s) returned	0.120 sec / 0.000022...
13:55:48	select productCode, productName from products where productCode like '%_20%'	10 row(s) returned	0.112 sec / 0.000030...

-- 7 orderdetails total de cada orden

```
select orderNumber, count(*) total from orderdetails group by orderNumber order by total desc;
```

The screenshot shows the MySQL Workbench interface. The left sidebar has sections for MANAGEMENT, INSTANCE, and PERFORMANCE. The main area contains a SQL editor with the following code:

```
10 • select employeeNumber, lastName, firstName from employees where firstName like 'T_m';
11   -- 5 empleados cuya nombre no empieza con b
12 • select employeeNumber, lastName, firstName from employees where firstName <> (select firstName from employees where firstName like 'bs');
13   -- 6 productos cuya código de producto y nombre de los productos cuyo código incluye la cadena _20
14 • select productCode, productName from products where productCode like '%_20%';
15   -- 7 orderdetails total de cada orden
16 • select orderNumber, count(*) total from orderdetails group by orderNumber order by total desc;
17   -- 8 select orders total de números de año
18 • select (select count(orderNumber) from orders as a where orderDate like '2003%'),
19   (select count(orderNumber) from orders as b where orderDate like '2004%'),
20   (select count(orderNumber) from orders as c where orderDate like '2005%')
21   from orders ac
100% 42:16
```

The results grid shows the output of the last query:

orderNumber total
10165 18
10168 18
10222 18
10275 18
10316 18
10332 18
10360 18

The status bar at the bottom indicates "Query Completed".

-- 8 select orders total de numeros de año

```
select (select count(orderNumber) from orders as a where orderDate like '2003%'),
```

```
(select count(orderNumber) from orders as b where orderDate like '2004%'),
```

```
(select count(orderNumber) from orders as c where orderDate like '2005%')
```

```
from orders as e
```

```
group by "status";
```

The screenshot shows the MySQL Workbench interface. The left sidebar has sections for SCHEMAS, TABLES, and STORED PROCEDURES. The main area contains a SQL editor with the following code:

```
14 • select productCode, productName from products where productCode like '%_20%';
15   -- 7 orderdetails total de cada orden
16 • select orderNumber, count(*) total from orderdetails group by orderNumber order by total desc;
17   -- 8 select orders total de números de año
18 • select (select count(orderNumber) from orders as a where orderDate like '2003%'),
19   (select count(orderNumber) from orders as b where orderDate like '2004%'),
20   (select count(orderNumber) from orders as c where orderDate like '2005%')
21   from orders as e
22   group by "status";
23   -- 9 select empleados con oficina en USA oficinas 1, 2 y 3
24 • select lastName, firstName, officeCode from employees as a where officeCode in (select officeCode from offices where country
25   -- 10 Obten el número del cliente, número de cheque y cantidad del cliente que ha realizado el pago más alto.
26 • select amount, customerNumber, checkNumber from payments order by amount desc limit 1;
27   -- 11 Obten el número del cliente, número de cheque y cantidad de aquellos clientes cuyo pago es más alto que el promedio.
28 • select customerNumber, checkNumber, amount from payments where amount > (select avg(amount) from payments) order by amount de
130% 24:18
```

The results grid shows the output of the last query:

customerNumber	checkNumber	amount
111	111	111
161	161	161
64	64	64

The status bar at the bottom indicates "Query Completed".

-- 9 select empleados con oficina en USA oficinas 1, 2 y 3

```
select lastName, firstName, officeCode from employees as a where officeCode in (select officeCode from offices where country = "USA");
```

The screenshot shows the SQL Developer interface with the SQL tab selected. The code area contains several numbered comments and queries. Query 41 is highlighted in blue. The result grid shows 10 rows of employee data. The action output table shows two entries for the execution of query 41.

Time	Action	Response	Duration / Fetch Time
40 13:57:53	select (select count(orderNumber) from orders as a where orderDate like '2003%'), (select count(orderNumber) from orders as b where orderDate like '2004%'), (select count(orderNumber) from orders as c where orderDate like '2005%')	1 row(s) returned	0.116 sec / 0.000023...
41 14:00:48	select lastName, firstName, officeCode from employees as a where officeCode in (select officeCode from offices where country = "US...")	10 row(s) returned	0.112 sec / 0.000020...

-- 10 Obten el número de cliente, número de cheque y cantidad del cliente que ha realizado el pago más alto.

```
select amount, customerNumber, checkNumber from payments order by amount desc limit 1;
```

The screenshot shows the SQL Developer interface with the SQL tab selected. The code area contains several numbered comments and queries. Query 42 is highlighted in blue. The result grid shows one row of payment data. The action output table shows two entries for the execution of query 42.

Time	Action	Response	Duration / Fetch Time
41 14:00:48	select lastName, firstName, officeCode from employees as a where officeCode in (select officeCode from offices where country = "US...")	10 row(s) returned	0.112 sec / 0.000020...
42 14:01:10	select amount, customerNumber, checkNumber from payments order by amount desc limit 1	1 row(s) returned	0.118 sec / 0.000018...

-- 11 Obten el número de cliente, número de cheque y cantidad de aquellos clientes cuyo pago es más alto que el promedio.

```
select customerNumber, checkNumber, amount from payments where amount > (select avg(amount) from payments) order by amount desc;
```

Administration Schemas Query 2 Projeto 2*

SCHEMAS

Q Filter objects

AEJ ALM AR AVH BMR CF classicmodels Tables Views Stored Proced... Functions DAJ EV JO MMS RC RL Ro SA SC SCZ SH sys tienda WH YC YCM

SQL1

Don't Limit

17 -- 8 select orders total de numeros de año
18 • select (select count(orderNumber) from orders as a where orderDate like '2003%'),
19 (select count(orderNumber) from orders as b where orderDate like '2004%'),
20 (select count(orderNumber) from orders as c where orderDate like '2005%')
21 from orders as e
22 group by "status";
23 -- 9 select empleados con oficina en USA oficinas 1, 2 y 3
24 • select lastName, firstName, officeCode from employees as a where officeCode in (select officeCode from offices where country
25 -- 10 Obten el número de cliente, número de cheque y cantidad del cliente que ha realizado el pago más alto.
26 • select amount, customerNumber, checkNumber from payments order by amount desc limit 1;
27 -- 11 Obten el número de cliente, número de cheque y cantidad de aquellos clientes cuyo pago es más alto que el promedio.
28 • select customerNumber, checkNumber, amount from payments where amount > (select avg(amount) from payments) order by amount desc
29 -- 12 Obten el nombre de aquellos clientes que no han hecho ninguna orden.
30 • select customerName, Total from (select a.customerName, (select count(*) from orders as b where a.customerNumber=b.customerNumber
31 from customers as a) as Total
130% C 14:28

Result Grid Filter Row Search Edit Export/Import

customerNumber	checkNumber	amount
141	JF105477	120166.58
141	ID10962	116308.40
124	KI131716	111654.40
148	KM172879	105743.00
124	AE15433	101244.59
321	DJ15149	85599.12
124	BG255406	85410.87
167	GN22884	85024.46
124	ET43496	83598.04
114	MAT6555	8221.22
239	NQ865547	80375.24
323	AL493079	75020.13
***	IM44768	45001.96

payments 12

Action Output

Time	Action	Response	Duration / Fetch Time
42 14:01:10	select amount, customerNumber, checkNumber from payments order by amount desc limit 1	1 row(s) returned	0.118 sec / 0.000018...
43 14:02:22	select customerNumber, checkNumber, amount from payments where amount > (select avg(amount) from payments) order by amount...	134 row(s) returned	0.129 sec / 0.000060...

Query Generated

Result Grid Form Editor Field Types

Apply

-- 12 Obtén el nombre de aquellos clientes que no han hecho ninguna orden

```
select customerName, Total from (select a.customerName, (select count(*) from orders as b  
where a.customerNumber=b.customerNumber) Total  
from customers a) as z where Total = '0';
```

The screenshot shows the Oracle SQL Developer interface. The SQL tab contains a query for calculating various statistics from the Northwind database. The Result Grid tab displays the results of the query, showing customer names and their total order counts.

```
17 -- 8 select orders total de numeros de año
18 • select (select count(orderNumber) from orders as a where orderDate like '2003%'),
19 (select count(orderNumber) from orders as b where orderDate like '2004%'),
20 (select count(orderNumber) from orders as c where orderDate like '2005%')
21 from orders as e
22 group by "status";
23 -- 9 select empleados con oficina en USA oficinas 1, 2 y 3
24 • select lastName, firstName, officeCode from employees as a where officeCode in (select officeCode from offices where country
25 -- 10 Obten el número de cliente, número de cheque y cantidad del cliente que ha realizado el pago más alto.
26 • select amount, customerNumber, checkNumber from payments order by amount desc limit 1;
27 -- 11 Obten el número de cliente, número de cheque y cantidad de aquellos clientes cuyo pago es más alto que el promedio.
28 • select customerNumber, checkNumber, amount from payments where amount > (select avg(amount) from payments) order by amount de
29 -- 12 Obten el nombre de aquellos clientes que no han hecho ninguna orden.
30 • select customerName, Total from (select a.customerName, (select count(*) from orders as b where a.customerNumber=b.customerNu
31 from customers a) as ? where Total = '0'.
130% 14:30
```

customerName	Total
Havel & Zyszek Co	0
American Sourcers Inc	0
Portos Unilever Co.	0
Asian Shopping Network, Co	0
Natürlich Autos	0
ANG Resellers	0
Messina Shopping Network	0
Franken Gifts, Co	0
B&G Collectables	0
Schuyler Imports	0
Der Hund Imports	0
Cramer Spezialitäten, Ltd	0

Result 13

Action Output	Time	Action	Response	Duration / Fetch Time
43	14:02:22	select customerNumber, checkNumber, amount from payments where amount > (select avg(amount) from payments) order by amount...	134 row(s) returned	0.129 sec / 0.00006...
44	14:02:56	select customerName, Total from (select a.customerName, (select count(*) from orders as b where a.customerNumber=b.customerNu...	24 row(s) returned	0.114 sec / 0.000023...

-- 13 Obten el máximo, mínimo y promedio del número de productos en las órdenes de venta

```
select orderNumber, max(quantityOrdered), min(quantityOrdered), avg(quantityOrdered)
from orderdetails group by orderNumber;
```

The screenshot shows the MySQL Workbench interface with the SQL tab selected. The query window contains the following code:

```
23 — 9 select empleados con oficina en USA oficinas 1, 2 y 3
24 • select lastName, firstName, officeCode from employees as a where officeCode in (select officeCode from offices where country
25 — 10 Obten el número de cliente, número de cheque y cantidad del cliente que ha realizado el pago más alto.
26 • select amount, customerNumber, checkNumber from payments order by amount desc limit 1;
27 — 11 Obten el número de cliente, número de cheque y cantidad de aquellos clientes cuyo pago es más alto que el promedio.
28 • select customerNumber, checkNumber, amount from payments where amount > (select avg(amount) from payments) order by amount desc
29 — 12 Obten el nombre de aquellos clientes que no han hecho ninguna orden.
30 • select customerName, Total from (select a.customerName, (select count(*) from orders as b where a.customerNumber=b.customerNu
31 from customers a) as z where Total = '0';
32 — 13 Obten el máximo, mínimo y promedio del número de productos en las órdenes de venta
33 • select orderNumber, max(quantityOrdered), min(quantityOrdered), avg(quantityOrdered) from orderdetails group by orderNumber;
34 — 14 Dentro de la tabla orders, obtén el número de órdenes que hay por cada estado.
35 • select status, count(*) from orders group by status;
36 — 14a órdenes por cada estado(ubicación)
37 • select (select state from customers a where a.customerNumber=b.customerNumber).state count(orderNumber).total
```

The result grid displays the following data:

orderNumber	max(quantityOrder...)	min(quantityOrder...)	avg(quantityOrder...)
10102	41	39	40.0000
10103	46	22	33.8125
10104	49	23	34.0769
10105	50	22	36.3333
10106	50	26	37.5000
10107	39	20	28.6250
10108	45	26	35.0625
10109	47	26	35.3333
10110	48	20	35.6250
10111	48	26	36.1667
10112	29	23	26.0000
10113	50	21	35.7500

The action output shows two log entries:

Time	Action	Response	Duration / Fetch Time
14:02:56	select customerName, Total from (select a.customerName, (select count(*) from orders as b where a.customerNumber=b.customerNu...	24 row(s) returned	0.114 sec / 0.000023...
14:03:33	select orderNumber, max(quantityOrdered), min(quantityOrdered), avg(quantityOrdered) from orderdetails group by orderNumber	326 row(s) returned	0.128 sec / 0.00014 s...

-- 14 Dentro de la tabla orders, obtén el número de órdenes que hay por cada estado.

```
select status, count(*) from orders group by status;
```

The screenshot shows the MySQL Workbench interface with the SQL tab selected. The query window contains the following code:

```
27 — 11 Obten el número de cliente, número de cheque y cantidad de aquellos clientes cuyo pago es más alto que el promedio.
28 • select customerNumber, checkNumber, amount from payments where amount > (select avg(amount) from payments) order by amount
29 — 12 Obten el nombre de aquellos clientes que no han hecho ninguna orden.
30 • select customerName, Total from (select a.customerName, (select count(*) from orders as b where a.customerNumber=b.customerNu
31 from customers a) as z where Total = '0';
32 — 13 Obten el máximo, mínimo y promedio del número de productos en las órdenes de venta
33 • select orderNumber, max(quantityOrdered), min(quantityOrdered), avg(quantityOrdered) from orderdetails group by orderNumber;
34 — 14 Dentro de la tabla orders, obtén el número de órdenes que hay por cada estado.
35 • select status, count(*) from orders group by status;
36 — 14a órdenes por cada estado(ubicación)
37 • select (select state from customers a where a.customerNumber=b.customerNumber).state count(orderNumber).total
```

The result grid displays the following data:

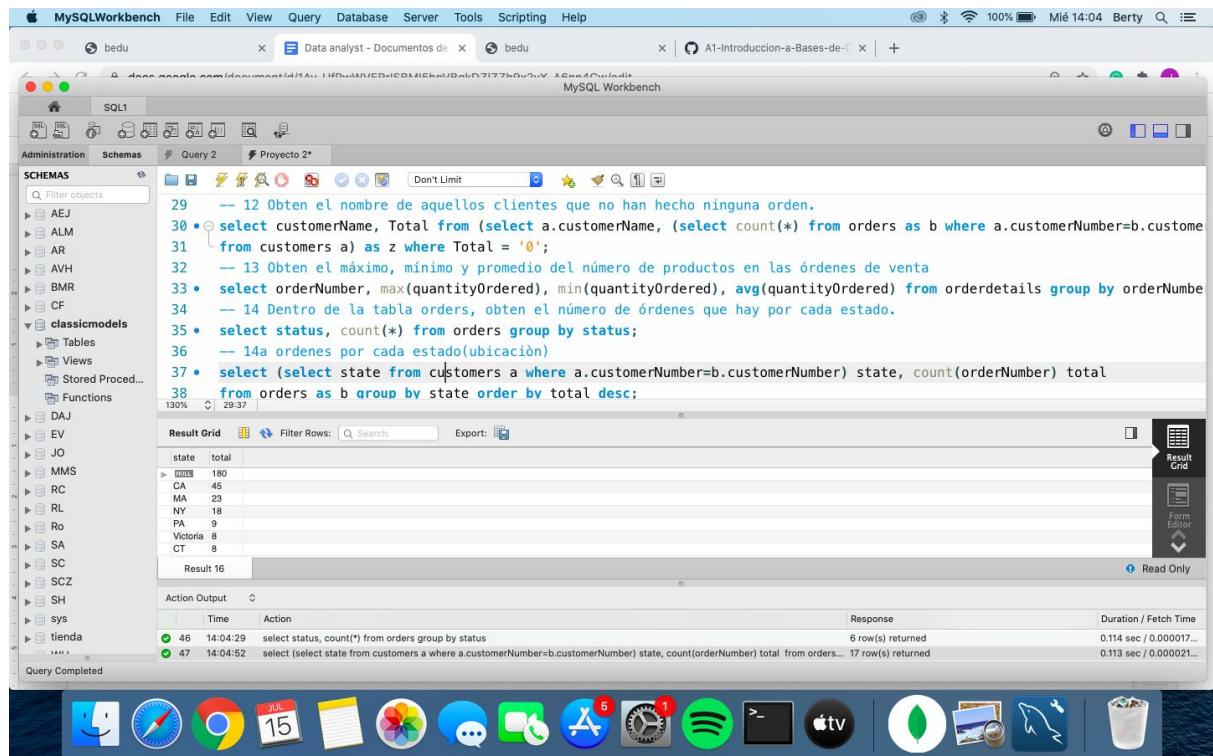
status	count(*)
Shipped	303
Resolved	4
Canceled	6
On Hold	4
Disputed	3
In Process	6

The action output shows two log entries:

Time	Action	Response	Duration / Fetch Time
14:03:33	select orderNumber, max(quantityOrdered), min(quantityOrdered), avg(quantityOrdered) from orderdetails group by orderNumber	326 row(s) returned	0.128 sec / 0.00014 s...
14:04:29	select status, count(*) from orders group by status	6 row(s) returned	0.114 sec / 0.000017...

-- 14a ordenes por cada estado(ubicaciòn)

```
select (select state from customers a where a.customerNumber=b.customerNumber) state,
count(orderNumber) total
from orders as b group by state order by total desc;
```



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

- SQL Editor:** Contains the SQL query provided above.
- Result Grid:** Displays the results of the query, showing the state and total count of orders for each state. The data is as follows:

state	total
HI	180
KS	65
MA	23
NY	18
PA	9
Victoria	8
CT	8

- Action Output:** Shows the history of actions taken, including the execution of the query.
- System Tray:** Shows various application icons at the bottom of the screen.

Sesión 03 Agrupaciones y consultas

Proyecto 3

-- 1.- Obten el código de producto, nombre de producto y descripción de todos los productos.

```
select a.productCode, a.productName, a.productDescription, b.productline, b.textDescription
from productlines b
right join products a
on a.productLine=b.productLine
group by a.productCode;
```

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

```
use classicmodels;
-- 1.- Obten el código de producto, nombre de producto y descripción de todos los productos.
select a.productCode, a.productName, a.productDescription, b.productline, b.textDescription
from productlines b
right join products a
on a.productLine=b.productLine
group by a.productCode;
-- 2.- Obten el número de orden, estado y costo total de cada orden.
select b.orderNumber, status, round(sum(priceEach), 1) from orders b
right join orderdetails a
on b.orderNumber = a.orderNumber
group by b.orderNumber;
```

The results grid displays the following data:

productCode	productName	productDescription	productline	textDescription
S10_2016	1996 Moto Guzzi 1100i	Official Moto Guzzi logos and insignias, saddle...	Motorcycles	Our motorcycles are state of the art replicas of classic as well as co...
S10_4698	2003 Harley-Davidson Eagle Drag Bike	Model features, official Harley Davidson logos a...	Motorcycles	Our motorcycles are state of the art replicas of classic as well as co...
S10_4757	1972 Alfa Romeo GTA	Features include: Turnable front wheels; steerin...	Classic Cars	Attention car enthusiasts: Make your wildest car ownership dreams...
S10_4902	1962 Lancia Delta 16V	Features include: Turnable front wheels; steerin...	Classic Cars	Attention car enthusiasts: Make your wildest car ownership dreams...
S12_1099	1968 Ford Mustang	Hood, doors and trunk all open to reveal highly...	Classic Cars	Attention car enthusiasts: Make your wildest car ownership dreams...
S12_1108	2001 Ferrari Enzo	Turnable front wheels; steering function; details...	Classic Cars	Attention car enthusiasts: Make your wildest car ownership dreams...
S12_1666	1958 Setra Bus	Model features 30 windows, skylights & glare re...	Trucks and Buses	The Truck and Bus models are realistic replicas of buses and spec...

The Action Output section shows two log entries:

Action	Time	Response	Duration / Fetch Time
use classicmodels	48 14:07:14	0 row(s) affected	0.109 sec
select a.productCode, a.productName, a.productDescription, b.productline, b.textDescription from productlines b right join produ...	49 14:07:20	110 row(s) returned	0.246 sec / 0.129 sec

-- 2.- Obten el número de orden, estado y costo total de cada orden.

```
select b.orderNumber, status, round(sum(priceEach), 1) from orders b
right join orderdetails a
on b.orderNumber = a.orderNumber
group by orderNumber;
describe products;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar indicates the application is running at 98% battery life, with the date Mié 14:42 and user Berty. The main window displays two tabs: 'Query 1' and 'Query 2'. The 'Query 1' tab contains the provided SQL code. The 'Query 2' tab shows the results of the second query:

orderNumber	status	round(sum(priceEach), 1)
10100	Shipped	301.8
10101	Shipped	352.0
10102	Shipped	138.7
10103	Shipped	1520.4
10104	Shipped	1251.9
10105	Shipped	1479.7
10106	Shipped	1427.3

Below the results, the 'Action Output' section shows two log entries:

- 49 14:07:20 select a.productCode, a.productName, a.productDescription, b.productline, b.textDescription from productlines b right join produ... 110 row(s) returned 0.246 sec / 0.129 sec
- 50 14:09:03 select b.orderNumber, status, round(sum(priceEach), 1) from orders b right join orderdetails a on b.orderNumber = a.orderNumber... 326 row(s) returned 0.118 sec / 0.00014 s...

The MySQL Workbench interface includes a sidebar for 'SCHEMAS' and a bottom dock with various Mac OS X application icons.

-- 3.- Obten el número de orden, fecha de orden, línea de orden, nombre del producto, cantidad ordenada y precio de cada pieza que muestre los detalles de cada orden.

```
select a.orderNumber, a.orderDate, b.orderLineNumber, c.productName, b.quantityOrdered, b.priceEach from orders as a  
right join orderdetails as b  
on a.orderNumber=b.orderNumber  
right join products as c  
on c.productCode=b.productCode  
order by a.orderNumber;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with several databases listed. The main area contains the SQL editor with the following code:

```
10 right join orderdetails a  
11 on b.orderNumber = a.orderNumber  
12 group by orderNumber;  
13 • describe products;  
14 -- 3.- Obten el número de orden, fecha de orden, línea de orden, nombre del producto, cantidad ordenada y precio de cada pieza que muestre los detalles  
15 • select a.orderNumber, a.orderDate, b.orderLineNumber, c.productName, b.quantityOrdered, b.priceEach from orders as a  
16 right join orderdetails as b  
17 on a.orderNumber=b.orderNumber  
18 right join products as c  
19 on c.productCode=b.productCode  
20 order by a.orderNumber;  
21 -- 4.- Obten el número de orden, nombre del producto, el precio sugerido de fábrica (msrp) y precio de cada pieza.
```

The 'Result Grid' tab is selected, showing the query results:

orderNumber	orderDate	orderLineNumber	productName	quantityOrdered	priceEach
10100	2003-01-06	2	1911 Ford Town Car	50	55.09
10100	2003-01-06	4	1932 Alfa Romeo 8C2300 Spider Sport	22	75.46
10100	2003-01-06	1	1936 Mercedes-Benz 500K Roadster	49	35.29
10101	2003-01-09	4	1932 Model A Ford J-Coupe	25	108.06
10101	2003-01-09	1	1928 Mercedes-Benz SSK	26	167.06
10101	2003-01-09	3	1939 Chevrolet Deluxe Coupe	45	32.53
10101	2003-01-09	2	1938 Cadillac V-16 Presidential Limousine	46	44.35

The 'Action Output' section shows the execution log:

Action	Time	Response	Duration / Fetch Time
select b.orderNumber, status, round(sum(priceEach), 1) from orders b right join orderdetails a on b.orderNumber = a.orderNumber group by orderNumber;	14:09:03	326 row(s) returned	0.118 sec / 0.00014 sec
select a.orderNumber, a.orderDate, b.orderLineNumber, c.productName, b.quantityOrdered, b.priceEach from orders as a right join orderdetails as b on a.orderNumber=b.orderNumber right join products as c on c.productCode=b.productCode order by a.orderNumber;	14:43:04	2997 row(s) returned	0.252 sec / 0.261 sec

-- 4.- Obtén el número de orden, nombre del producto, el precio sugerido de fábrica (msrp) y precio de cada pieza.

```
select a.orderNumber, b.productName, b.MSRP, b.buyPrice, a.priceEach precioVenta  
from products b  
right join orderdetails a  
on a.productCode=b.productCode  
order by a.orderNumber;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar indicates the application is running at 98% battery life, with the date Mié 14:43 and user Berty. The main window has three tabs: SQL1, Query 2, and Projeto 3. The SQL1 tab contains the following SQL code:

```
18 right join products as c  
19 on c.productCode=b.productCode  
20 order by a.orderNumber;  
21 -- 4.- Obtén el número de orden, nombre del producto, el precio sugerido de fábrica (msrp) y precio de cada pieza.  
22 • select a.orderNumber, b.productName, b.MSRP, b.buyPrice, a.priceEach precioVenta  
23 from products b  
24 right join orderdetails a  
25 on a.productCode=b.productCode  
26 order by a.orderNumber;  
27 -- 5.- Obtén el número de cliente, nombre de cliente, número de orden y estado de cada cliente.  
28 • select a.customerNumber, a.customerName, b.orderNumber, b.status  
29 from customers a
```

The code is numbered 18 to 29. The result grid shows the output of the query:

orderNumber	productName	MSRP	buyPrice	precioVenta
10100	1932 Alfa Romeo 8C2800 Spider Sport	92.03	43.26	75.46
10100	1936 Mercedes-Benz 500K Roadster	41.03	21.75	35.29
10101	1932 Model A Ford J-Coupe	127.13	58.48	108.06
10101	1928 Mercedes-Benz SSK	168.75	72.56	167.06
10101	1939 Chevrolet Deluxe Coupe	33.19	22.57	32.53
10101	1938 Cadillac V-16 Presidential Limousine	44.80	20.61	44.35
10102	1937 Lincoln Berline	102.74	60.62	95.55

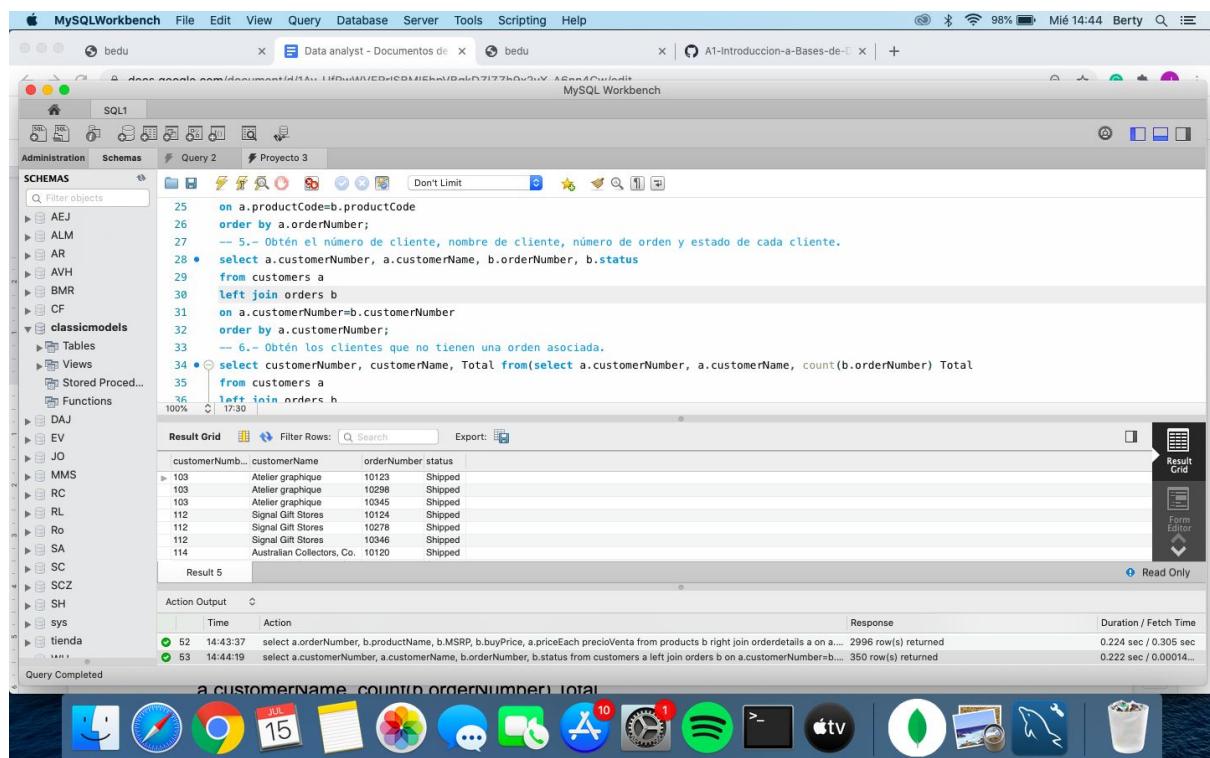
The results are labeled "Result 4". Below the result grid, the "Action Output" section shows two log entries:

Time	Action	Response	Duration / Fetch Time
51 14:43:04	select a.orderNumber, a.orderDate, b.orderLineNumber, c.productName, b.quantityOrdered, b.priceEach from orders as a right join orderdetails b on a.orderNumber = b.orderNumber left join products c on b.productCode = c.productCode	2997 row(s) returned	0.252 sec / 0.261 sec
52 14:43:37	select a.orderNumber, b.productName, b.MSRP, b.buyPrice, a.priceEach precioVenta from products b right join orderdetails a on a.orderNumber = b.orderNumber left join customers c on a.customerNumber = c.customerNumber	2996 row(s) returned	0.224 sec / 0.305 sec

The status bar at the bottom of the screen shows "Cliente".

-- 5.- Obtén el número de cliente, nombre de cliente, número de orden y estado de cada cliente.

```
select a.customerNumber, a.customerName, b.orderNumber, b.status  
from customers a  
left join orders b  
on a.customerNumber=b.customerNumber  
order by a.customerNumber;
```



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

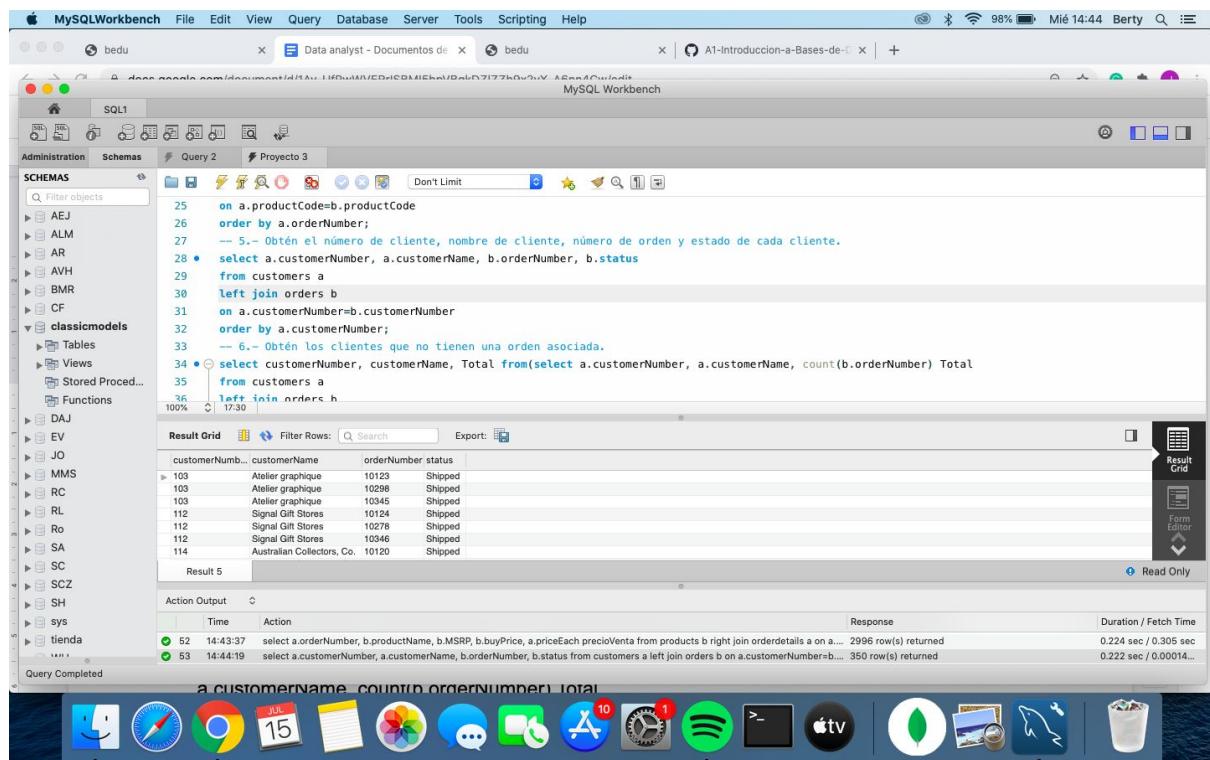
- SQL Editor:** Contains the SQL query provided above.
- Result Grid:** Displays the query results in a tabular format. The columns are customerNumber, customerName, orderNumber, and status. The data includes rows for customers like 'Atelier graphique', 'Signal Gift Stores', and 'Australian Collectors, Co.' with their respective order numbers and statuses.
- Action Output:** Shows the execution log with two entries:

 - Line 52: select a.orderNumber, b.productName, b.MSRP, b.buyPrice, a.priceEach precioVenta from products b right join orderdetails a on a.orderNumber=b.orderNumber left join orders o on a.orderNumber=o.orderNumber 2996 row(s) returned
 - Line 53: select a.customerNumber, a.customerName, b.orderNumber, b.status from customers a left join orders b on a.customerNumber=b.customerNumber 350 row(s) returned

- System Status:** Top right shows battery at 98%, time as Mié 14:44, and user Betty.
- Mac OS Dock:** Bottom shows various Mac OS application icons.

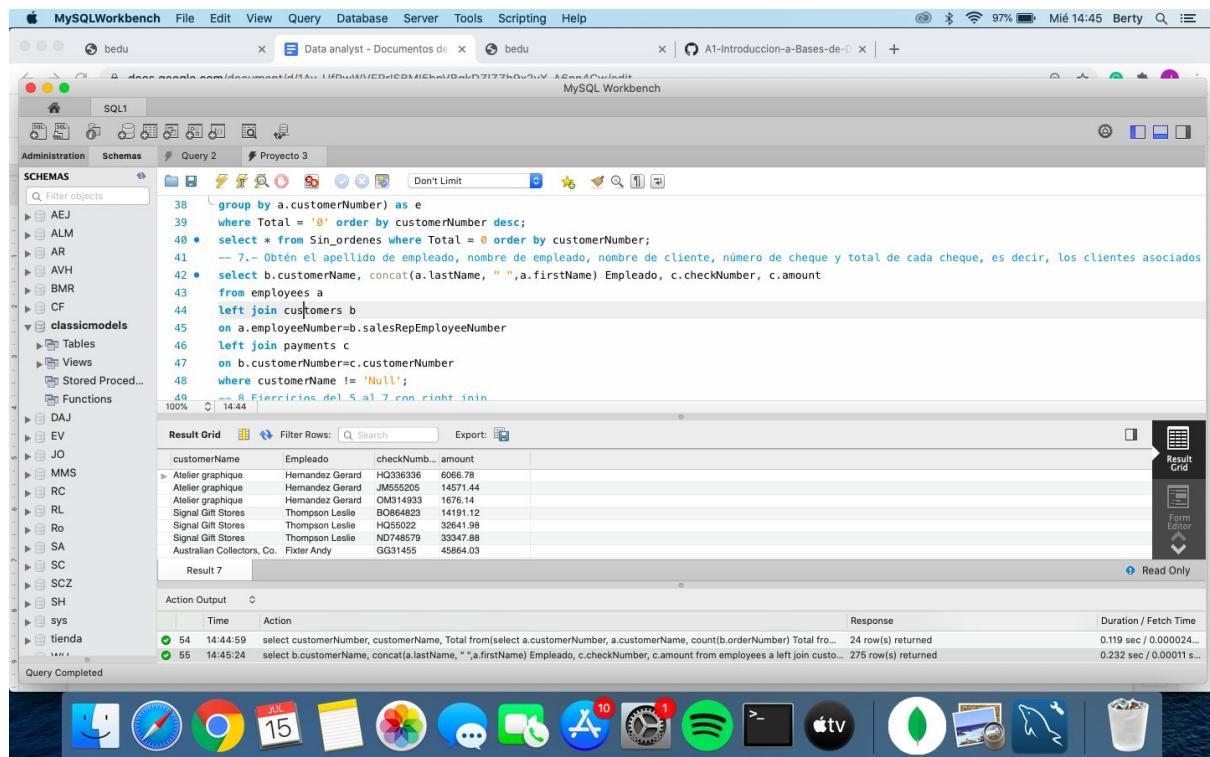
-- 6.- Obtén los clientes que no tienen una orden asociada.

```
select customerNumber, customerName, Total from(select a.customerNumber,
a.customerName, count(b.orderNumber) Total
from customers a
left join orders b
on a.customerNumber=b.customerNumber
group by a.customerNumber) as e
where Total = '0' order by customerNumber desc;
select * from Sin_ordenes where Total = 0 order by customerNumber;
```



-- 7.- Obtén el apellido de empleado, nombre de empleado, nombre de cliente, número de cheque y total de cada cheque, es decir, los clientes asociados a cada empleado.

```
select b.customerName, concat(a.lastName, " ",a.firstName) Empleado, c.checkNumber,
c.amount
from employees a
left join customers b
on a.employeeNumber=b.salesRepEmployeeNumber
left join payments c
on b.customerNumber=c.customerNumber
where customerName != 'Null';
```



The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar indicates it's running on a 'bedu' machine at 14:45. The main window displays a SQL editor with the following query:

```
38   group by a.customerNumber) as e
39   where Total = '0' order by customerNumber desc;
40 •  select * from ordenes where Total = 0 order by customerNumber;
41   -- 7.- Obtén el apellido de empleado, nombre de empleado, nombre de cliente, número de cheque y total de cada cheque, es decir, los clientes asociados
42 •  select b.customerName, concat(a.lastName, " ",a.firstName) Empleado, c.checkNumber, c.amount
43   from employees a
44   left join customers b
45   on a.employeeNumber=b.salesRepEmployeeNumber
46   left join payments c
47   on b.customerNumber=c.customerNumber
48   where customerName != 'Null';
49   -- 8. Fíjate en el 5 al 7 con right join
100% | 14:44
```

The results grid shows the following data:

customerName	Empleado	checkNum...	amount
Atelier graphique	Hernandez Gerard	HQ336336	6066.78
Atelier graphique	Hernandez Gerard	JM55205	14571.44
Atelier graphique	Hernandez Gerard	ND748579	1273.43
Signal Gift Stores	Thompson Leslie	BO86498	14161.12
Signal Gift Stores	Thompson Leslie	HO56022	32641.98
Signal Gift Stores	Thompson Leslie	ND748579	33347.88
Australian Collectors, Co.	Fixter Andy	GG31455	45864.03

The status bar at the bottom shows 'Query Completed'.

-- 8 Ejercicios del 5 al 7 con right join

-- 5a

```
select a.customerNumber, a.customerName, b.orderNumber, b.status  
from orders b
```

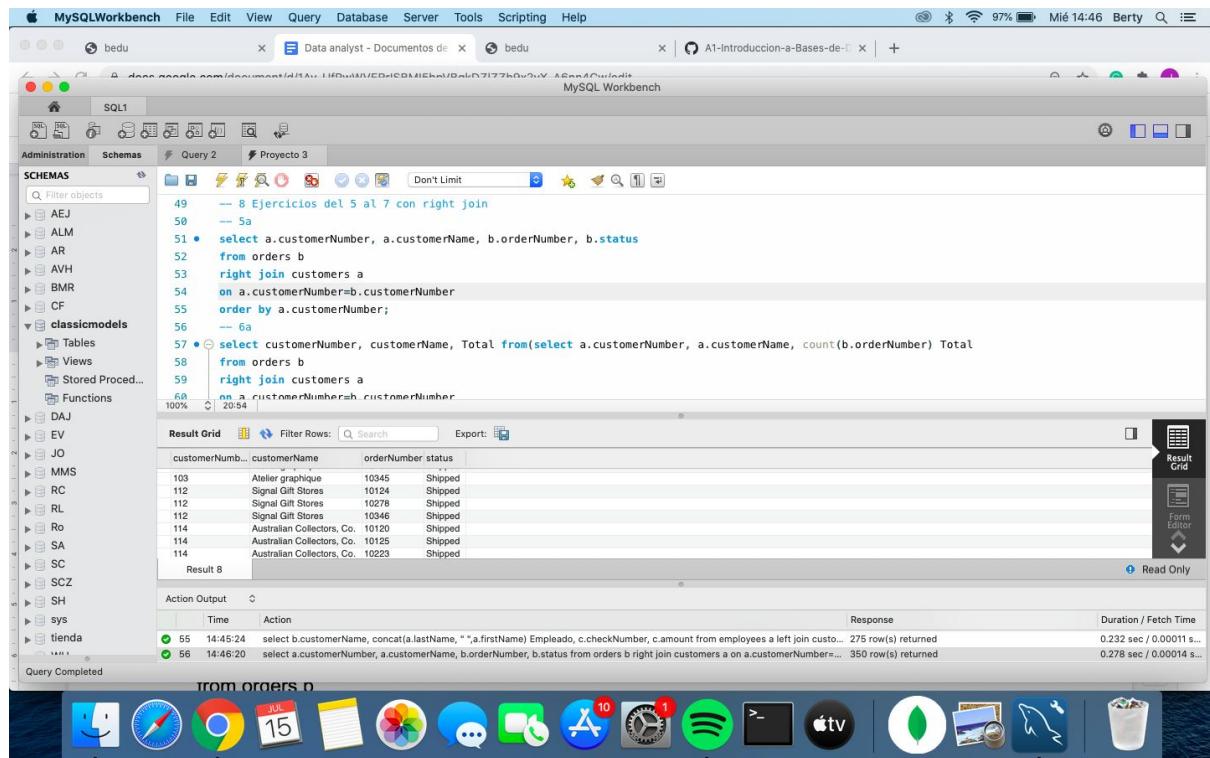
```
right join customers a  
on a.customerNumber=b.customerNumber  
order by a.customerNumber;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar indicates the application is running at 97% battery life, with the date Mié 14:46 and user Betty. The main window has three tabs: SQL1, Query 2, and Projeto 3. The SQL1 tab contains the following code:

```
49 -- 8 Ejercicios del 5 al 7 con right join  
50 -- 5a  
51 • select a.customerNumber, a.customerName, b.orderNumber, b.status  
52 from orders b  
53 right join customers a  
54 on a.customerNumber=b.customerNumber  
55 order by a.customerNumber;  
56 -- 6a  
57 • Ⓜ select customerNumber, customerName, Total from(select a.customerNumber, a.customerName, count(b.orderNumber) Total  
58 from orders b  
59 right join customers a  
60 on a.customerNumber=b.customerNumber  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1098  
1099  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1198  
1199  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1298  
1299  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1398  
1399  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1498  
1499  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1598  
1599  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1698  
1699  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1798  
1799  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877<br
```

-- 6a

```
select customerNumber, customerName, Total from(select a.customerNumber,
a.customerName, count(b.orderNumber) Total
from orders b
right join customers a
on a.customerNumber=b.customerNumber
group by a.customerNumber) as e
where Total = '0' order by customerNumber desc;
```



customerNumber	customerName	orderNumber	status
103	Atelier graphique	10345	Shipped
112	Signal Gift Stores	10124	Shipped
112	Signal Gift Stores	10278	Shipped
112	Signal Gift Stores	10346	Shipped
114	Australian Collectors, Co.	10120	Shipped
114	Australian Collectors, Co.	10125	Shipped
114	Australian Collectors, Co.	10223	Shipped

Result Grid

Action Output

Time Action Response Duration / Fetch Time

55 14:45:24 select b.customerName, concat(a.lastName, " ",a.firstName) Empleado, c.checkNumber, c.amount from employees a left join custo... 275 row(s) returned 0.232 sec / 0.0001 s...

56 14:46:20 select a.customerNumber, a.customerName, b.orderNumber, b.status from orders b right join customers a on a.customerNumber=b.cust... 350 row(s) returned 0.278 sec / 0.0001 s...

-- 7a

```
select b.customerName, concat(a.lastName, " ",a.firstName) Empleado, c.checkNumber,
c.amount
from employees a
right join customers b
on a.employeeNumber=b.salesRepEmployeeNumber
right join payments c
on b.customerNumber=c.customerNumber;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar indicates the application is running at 97% battery life, with the date Mié 14:47. The main window displays a SQL editor with a multi-line query and its execution results.

SQL Editor Content:

```
60   on a.customerNumber=b.customerNumber
61   group by a.customerNumber) as e
62   where Total = 0' order by customerNumber desc;
63   -- 7a
64 •  select b.customerName, concat(a.lastName, " ",a.firstName) Empleado, c.checkNumber,
65   from employees a
66   right join customers b
67   on a.employeeNumber=b.salesRepEmployeeNumber
68   right join payments c
69   on b.customerNumber=c.customerNumber;
70   -- 9 crear vistas para 3 consultas más complejas
71 •  create view ordenes as (select customerNumber, customerName, Total from(select a.customerNumber, a.customerName, count(b.orderNumber) Total fro...
100% 17/66
```

Result Grid:

customerName	Empleado	checkNum... amount
Atelier graphique	Hernandez Gerard	OM314933 1676.14
Signal Gift Stores	Thompson Leslie	BO864823 14191.12
Signal Gift Stores	Thompson Leslie	HO55022 32641.98
Signal Gift Stores	Thompson Leslie	ND748579 33347.88
Australian Collectors, Co.	Fixter Andy	GG31455 45864.03
Australian Collectors, Co.	Fixter Andy	MA765515 82261.22
Australian Collectors, Co.	Fixter Andy	NP603840 7565.08

Action Output:

Time	Action	Response	Duration / Fetch Time
57 14:47:10	select customerNumber, customerName, Total from(select a.customerNumber, a.customerName, count(b.orderNumber) Total fro...	24 row(s) returned	0.133 sec / 0.000026...
58 14:47:40	select b.customerName, concat(a.lastName, " ",a.firstName) Empleado, c.checkNumber, c.amount from employees a right join cus...	273 row(s) returned	0.234 sec / 0.00012...

-- 9 crear vistas para 3 consultas más complejas

1:

```
create view 0_ordenes as (select customerNumber, customerName, Total from(select
a.customerNumber, a.customerName, count(b.orderNumber) Total
from customers a
left join orders b
on a.customerNumber=b.customerNumber
group by a.customerNumber) as e
where Total = '0' order by customerNumber desc);
```

```
select * from 0_ordenes;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar indicates it's running on 'bedu' at 96% battery, with the date Mié 14:49. The main window has tabs for 'SQL1', 'Query 2', and 'Proyecto 3'. The left sidebar shows the 'SCHEMAS' tree with several databases like AEJ, ALM, AR, AVH, BMR, CF, and classicmodels. The 'classicmodels' database is selected, showing tables such as 'Tables', 'Views', '0_ordenes', 'AVco', 'AVo', 'AVpce', 'Detalle_ordenes', 'Empleado_C...', 'RCA_detalle...', 'RCA_orden', 'RCA_produc...', 'SAUClientes...', 'SAUProduct...', 'SC_clientes', 'SC_ordenes', and 'SC_ventas_e...'. The central area contains the SQL editor with the following code:

```
69    on b.customerNumber=c.customerNumber;
70    -- 9 crear vistas para 3 consultas más complejas
71  •  create view 0_ordenes as (select customerNumber, customerName, Total from(select
72      a.customerNumber, a.customerName, count(b.orderNumber) Total
73      from customers a
74      left join orders b
75      on a.customerNumber=b.customerNumber
76      group by a.customerNumber) as e
77      where Total = '0' order by customerNumber desc);
78  •  select * from 0_ordenes;
79
80  •  create view Empleado_Cliente as (select b.customerName, concat(a.lastName, " ", a.firstName) Empleado, c.checkNumber, c.amount
100% 21:78
```

The 'Result Grid' tab is active, displaying the results of the 'select * from 0_ordenes' query. The table has columns 'customerNumber', 'customerName', and 'Total'. The data shows 11 rows where the 'Total' value is 0. The bottom status bar shows 'Query Completed'.

customerNumber	customerName	Total
477	Mt Vergünen & Co.	0
465	Anton Designs, Ltd.	0
459	Wrburg Exchange	0
443	Feuer Online Stores, Inc	0
409	Stuttgart Collectable Exchange	0
376	Precious Collectables	0
369	Uaboa Souveniers, Inc	0
	0_ordenes 11	

Action Output table:

Action	Time	Response	Duration / Fetch Time
select b.customerName, concat(a.lastName, " ", a.firstName) Empleado, c.checkNumber, c.amount from employees a right join cus...	58 14:47:40	273 row(s) returned	0.234 sec / 0.00012...
select * from 0_ordenes	59 14:48:49	24 row(s) returned	0.111 sec / 0.000072...

2.-

```
create view Empleado_Cliente as (select b.customerName, concat(a.lastName, " "
,a.firstName) Empleado, c.checkNumber, c.amount
from employees a
left join customers b
on a.employeeNumber=b.salesRepEmployeeNumber
left join payments c
on b.customerNumber=c.customerNumber
where customerName != 'Null');
```

```
select * from Empleado_Cliente;
```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the title is 'bedu' and the tab is 'Data analyst - Documentos de...'. The bottom status bar shows the date as 'Mié 14:49' and battery level as '96%'. The main window has three tabs: 'SQL1' (selected), 'Query 2', and 'Proyecto 3'. The left sidebar shows the 'Schemas' tree with several databases like AEF, ALM, AR, AVH, BMR, CF, and classicmodels. The 'classicmodels' database is expanded, showing 'Tables' and 'Views'. The 'Views' section contains two entries: '0_ordenes' and 'Empleado_Cliente'. The 'Result Grid' pane displays the results of the 'select * from Empleado_Cliente;' query. The results table has columns: 'customerName', 'Empleado', 'checkNum...', and 'amount'. The data includes rows for Hernandez Gerard, Thompson Leslie, Signal Gift Stores, Australian Collectors, Co., and Fixter Andy. The bottom pane shows the 'Action Output' log with two entries: 'select * from 0_ordenes' at 14:48:49 and 'select * from Empleado_Cliente' at 14:49:52. The status bar at the bottom right indicates 'Read Only'.

customerName	Empleado	checkNum...	amount
Atelier graphique	Hernandez Gerard	OM314933	1676.14
Signal Gift Stores	Thompson Leslie	PD364623	1419.52
Signal Gift Stores	Thompson Leslie	PD364623	320.91-1.58
Signal Gift Stores	Thompson Leslie	ND748579	33347.68
Australian Collectors, Co.	Fixter Andy	GG31455	45861.03
Australian Collectors, Co.	Fixter Andy	MA765515	82261.22
Australian Collectors, Co.	Fixter Andy	NP603840	7565.09

Action Output

Time	Action	Response	Duration / Fetch Time
59 14:48:49	select * from 0_ordenes	24 row(s) returned	0.111 sec / 0.000072...
60 14:49:52	select * from Empleado_Cliente	275 row(s) returned	0.242 sec / 0.0001 s...

3.-

```
create view Detalle_ordenes as (select a.orderNumber, a.orderDate, b.orderLineNumber,
c.productName, b.quantityOrdered, b.priceEach from orders as a
right join orderdetails as b
on a.orderNumber=b.orderNumber
right join products as c
on c.productCode=b.productCode
order by a.orderNumber);
```

```
select * from Detalle_ordenes;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar indicates the application is running at 96% battery life, and the date is Mié 14:51. The main window displays the SQL editor with the following code:

```
88 •    select * from Empleado_Cliente;
89
90 •  create view Detalle_ordenes as (select a.orderNumber, a.orderDate, b.orderLineNumber,
91   right join orderdetails as b
92   on a.orderNumber=b.orderNumber
93   right join products as c
94   on c.productCode=b.productCode
95   order by a.orderNumber);
96
97 •  select * from Detalle_ordenes;
98 OR
100% C | 18:97
```

The SQL editor has two tabs: "Query 1" and "Query 2". The "Query 1" tab contains the code above. The "Result Grid" tab shows the output of the query:

orderNumber	orderDate	orderLineNumber	productName	quantityOrdered	priceEach
10100	2003-01-06	3	1985 Toyota Supra	30	136.00
10100	2003-01-06	2	1917 Grand Touring Sedan	25	35.20
10100	2003-01-06	4	1932 Alfa Romeo 8C2300 Spider Sport	22	75.48
10100	2003-01-06	1	1936 Mercedes-Benz 500k Roadster	49	35.29
10101	2003-01-09	4	1932 Model A Ford J-Coupe	25	108.06
10101	2003-01-09	1	1928 Mercedes-Benz SSK	26	167.06

The "Action Output" section shows the execution history:

Action	Time	Response	Duration / Fetch Time
select * from Empleado_Cliente	60 14:49:52	275 row(s) returned	0.242 sec / 0.00011 s...
select * from Detalle_ordenes	61 14:51:00	2997 row(s) returned	0.233 sec / 0.354 sec

The status bar at the bottom right of the screen shows "Read Only".

Sesión 04: Fundamentos de MongoDB

Proyecto 4

1.- Obtén los datos de contacto de cada compañía.

```
{  
  filter: {  
    project: {  
      email_address: 1,  
      phone_number: 1  
    }  
  }  
}
```

The screenshot shows the MongoDB Compass interface. On the left, the sidebar displays the database structure under 'Local'. The 'sample_training' database is selected, and its 'companies' collection is highlighted. The main pane shows the results of a query on the 'companies' collection. The query is defined in the 'PROJECT' stage of the pipeline:

```
{  
  email_address: 1,  
  phone_number: 1  
}
```

The results grid displays four documents, each containing the '_id' field and either the 'email_address' or 'phone_number' field (both are empty). The interface also shows other stages in the pipeline: 'FILTER' (with a condition for 'founded_year: 2008'), 'SORT' (disabled), and 'COLLATION' (disabled). The 'Aggregations' tab is visible at the top of the main pane.

2.- Obtén la fuente de cada tweet.

```
{  
  filter: {  
    project: {  
      source: 1  
    }  
  }  
}
```

The screenshot shows the MongoDB Compass application interface. On the left, there's a sidebar titled 'Local' showing 10 DBs and 25 collections. The 'sample_training.tweets' collection is selected. In the center, the document viewer displays 24.8k documents with a total size of 39.4MB and an average size of 1.6KB. There are 1 index with a total size of 232.0KB and an average size of 232.0KB. The 'Documents' tab is active, showing the first few documents with their _id and source fields. On the right, there's a 'PROJECT' panel with a query builder containing the following code:

```
Project  
{  
  source: 1  
}
```

3.- Obtén el nombre de todas las compañías fundadas en octubre.

```
{  
  filter: {  
    founded_month: 10  
  }  
}
```

The screenshot shows the MongoDB Compass application running on a Mac OS X desktop. The main window displays the 'sample_training.companies' collection with 9.5k documents. A filter is applied: {founded_month: 10}. The right side of the interface shows three project definitions:

- Proyecto 4-4**: FILTER { founded_year: 2008 }
- Proyecto 4-3**: FILTER { founded_month: 10 }
- Proyecto 4-1**: PROJECT { email_address: 1, phone_number: 1 }

The bottom of the screen shows the Mac OS X dock with various application icons.

4.- Obtén el nombre de todas las compañías fundadas en 2008.

```
{  
  filter: {  
    founded_year: 2008  
  }  
}
```

The screenshot shows the MongoDB Compass interface. In the top navigation bar, the collection 'sample_training.companies' is selected. The main pane displays a list of documents with the following details:

- Total documents: 9.5k
- Total size: 34.8MB
- Avg. size: 3.7KB
- Indexes: 1
- Total size: 100.0KB
- Avg. size: 100.0KB

The 'Documents' tab is active, showing a table with columns: _id, name, created_at, updated_at, and overview. A filter bar at the top of the table area specifies 'founded_year: 2008'. The results table shows two documents, both of which are for 'OpenX'. The first document's details are expanded:

```
_id: ObjectId("52cdef7c4bab8bd675297da8")  
name: "OpenX"  
permalink: "openx"  
crunchbase_url: "http://www.crunchbase.com/company/openx"  
homepage_url: "http://www.openx.com"  
blog_url: "http://www.openx.com/blog"  
blog_feed_url: ""  
twitter_username: "OpenX"  
category_code: "advertising"  
number_of_employees: 305  
founded_year: 2008  
founded_month: 5  
founded_day: 1  
deadpooled_year: null  
deadpooled_month: null  
deadpooled_day: null  
deadpooled_url: null  
tag_list: "ad-server, ad-serving, ad-exchange, yield-optimization, content-valua..."  
alias_list: ""  
email_address: "hello@openx.com"  
phone_number: "626-466-1141"  
description: "OpenX"  
created_at: "Wed Jun 13 18:23:43 UTC 2007"  
updated_at: "Wed Dec 11 08:44:01 UTC 2013"  
overview: "<p>OpenX is a global leader in digital and mobile advertising technolo..."
```

Below the table, there is a 'SHOW 17 MORE FIELDS' button. To the right of the table, there are three project cards labeled 'Proyecto 4-4', 'Proyecto 4-3', and 'Proyecto 4-1', each containing a snippet of code for filtering documents based on their year or month of creation.

5.- Obtén todos los *post* del autor machine.

```
{  
  filter: {  
    author: 'machine'  
  }  
}
```

The screenshot shows the MongoDB Compass interface. In the top navigation bar, the collection 'sample_training.posts' is selected. On the left sidebar, under the 'sample_training' section, the 'posts' item is highlighted. The main pane displays the results of a query with the filter '{author: 'machine'}'. Three documents are listed:

```
_id: ObjectId("50ab0f8bbc1bfe2536dc3f9")  
body: "Amendment I  
<p>Congress shall make no law respecting an establishment ..."  
permalink: "http://tiny.cc/meyarw"  
author: "machine"  
title: "Bill of Rights"  
tags: Array  
comments: Array  
date: 2012-11-20T05:05:15.231+00:00  
  
_id: ObjectId("50ab0f8bbc1bfe2536dc3fa")  
body: "We the People of the United States, in Order to form a more perfect Union..."  
permalink: "http://tiny.cc/meyarw"  
author: "machine"  
title: "US Constitution"  
tags: Array  
comments: Array  
date: 2012-11-20T05:05:15.232+00:00  
  
_id: ObjectId("50ab0f8bbc1bfe2536dc3fb")  
body: "Four score and seven years ago our fathers brought forth on this conti..."  
permalink: "http://tiny.cc/meyarw"  
author: "machine"  
title: "Gettysburg Address"  
tags: Array  
comments: Array  
date: 2012-11-20T05:05:15.234+00:00
```

6.- Obtén todos los tweets provenientes de la web.

```
{  
  filter: {  
    source: 'web'  
  }  
}
```

The screenshot shows the MongoDB Compass interface. On the left, the sidebar lists databases and collections, with 'sample_training.tweets' selected. The main pane displays a list of documents from the 'sample_training.tweets' collection. A search bar at the top contains the query: '{source: "web"}'. The results show two documents, both of which have 'source: "web"'. The bottom right corner of the screen shows the Mac OS X dock with various application icons.

```
_id: ObjectId("5cdec0ca187d17ca623ff")  
text: "eu preciso de terminar de fazer a minha tabela, está muito foda ***"  
in_reply_to_status_id: null  
retweet_count: null  
contributors: null  
created_at: "Thu Sep 02 18:11:23 +0000 2010"  
geo: null  
source: "web"  
coordinates: null  
in_reply_to_screen_name: null  
truncated: false  
> entities: Object  
retweeted: false  
place: null  
> user: Object  
favorited: false  
in_reply_to_user_id: null  
id: 22819396900
```

```
_id: ObjectId("5cdec0ca187d17ca623ff")  
text: "First week of school is over :P"  
in_reply_to_status_id: null  
retweet_count: null  
contributors: null  
created_at: "Thu Sep 02 18:11:25 +0000 2010"  
geo: null  
source: "web"  
coordinates: null  
in_reply_to_screen_name: null  
truncated: false
```

7.- Obtén todas las compañías fundadas en octubre del 2008

```
{  
  filter: {  
    $and: [  
      {  
        founded_month: 10  
      },  
      {  
        founded_year: 2008  
      }  
    ]  
  }  
}
```

The screenshot shows the MongoDB Compass interface with the following details:

- Left Sidebar:** Shows the local host (ec2-35-166-232-75.us-west-2.compute.amazonaws.com) and the sample_training database.
- Central Area:** The "sample_training.companies" collection is selected. It displays 9.5k documents with a total size of 34.8MB and an average size of 3.7KB. There is 1 index with a total size of 100.0KB and an average size of 100.0KB.
- Filter Bar:** A filter is applied: `[$and: [{ founded_month: 10}, { founded_year: 2008}]]`.
- Results:** One document is shown in the list:

```
_id:ObjectId("52cdef7c4bab8bd6752985ca")
name:"tunesBag"
permalink:"tunesbag"
crunchbase_url:"http://www.crunchbase.com/company/tunesbag"
homepage_url:"http://www.tunesBag.com"
blog_url:"http://tunesBag.blogspot.com/"
blog_feed_url:"http://blog.tunesbag.com/feeds/posts/default?alt=rss"
twitter_username:
category:"Social Music Video"
number_of_employees:null
founded_year:2008
founded_month:10
founded_day:1
deadpooled_year:null
deadpooled_month:null
deadpooled_day:null
deadpooled_url:null
tags:"music, cloud, locker, mp3, music-streaming, streaming"
alias_list:
email_address:"office@tunesBag.com"
phone_number:"+43 600 215 27 96"
description:"Social Music Player"
created_at:"Thu Mar 20 15:45:40 UTC 2008"
updated_at:"Thu Jan 19 00:37:48 UTC 2012"
overview:<p><sup></sup>Austria based tunesBag is a music player with social features on th...</p>
```
- Right Side:** Four project cards are visible:
 - Projeto 4-8: Filter for companies with more than 50 employees.
 - Projeto 4-7: Filter for companies founded in October 2008.
 - Projeto 4-4: Filter for companies founded in 2008.
 - Projeto 4-3: Filter for companies founded in 2008.

8.- Obtén todas las compañías con más de 50 empleados.

```
{  
  filter: {  
    number_of_employees: {  
      $gt: 50  
    }  
  }  
}
```

The screenshot shows the MongoDB Compass interface. On the left, the sidebar lists databases and collections, with 'sample_training' selected. The main pane displays the 'sample_training.companies' collection, which contains 9.5k documents. A search bar at the top right shows the query: {number_of_employees: {\$gt: 50}}. Below the search bar, there are tabs for 'Documents', 'Aggregations', 'Schema', 'Explain Plan', and 'Indexes'. The 'Documents' tab is active, showing the results of the query. One document is expanded to show its fields, including _id, name, permalink, created_url, homepage_url, blog_url, blog_feed_url, twitter_username, category_code, number_of_employees, founded_year, founded_month, founded_day, deadpooled_year, deadpooled_month, deadpooled_day, deadpooled_url, tag_list, alias_list, email_address, phone_number, description, created_at, updated_at, and overview. The results table shows 1 document found, totaling 100.0KB. The bottom of the screen shows the macOS dock with various application icons.

9.- Obtén las historias con número de comentarios entre 10 y 30.

```
{  
  filter: {  
    $or: [  
      {  
        comments: {  
          $gte: 10  
        }  
      },  
      {  
        comments: {  
          $lte: 30  
        }  
      }  
    ]  
  }  
}
```

The screenshot shows the MongoDB Compass interface. On the left, the sidebar displays the database structure under 'Local'. The 'sample_training' collection is selected, showing 9842 documents with a total size of 9.8k. The main pane displays two documents from the collection:

```
_id: ObjectId("4ba267dc238d3ba3ca000001")
href: "http://digg.com/people/Jedi_believer_who_refused_to_remove_hood_gets_apology"
title: "Jedi believer who refused to remove hood gets an apology"
comments: 153
container: Object
submit_date: 1268771801
topic: Object
promote_date: 1268915964
id: "19970866"
media: "news"
digg: 1
description: "Chris Jarvis is a member of the International Church of Jediism - base..."
```

```
_id: ObjectId("4ba267dc238d3ba3ca000003")
href: "http://digg.com/odd_stuff/Man_Assaulted_Female_Police_Officer_With_His_Penis"
title: "Man Assaulted Female Police Officer With His Penis"
comments: 93
thumbnail: Object
container: Object
submit_date: 1268755030
topic: Object
promote_date: 1268903731
id: "19963997"
media: "news"
digg: 262
description: "A 28-year-old man who assaulted a female police officer with his penis..."
```

The right side of the interface shows the query builder with the following code:

```
{  
  $or: [  
    {  
      comments: {  
        $gte: 10  
      }  
    },  
    {  
      comments: {  
        $lte: 30  
      }  
    }  
  ]  
}
```

Below the code, there are sections for 'FILTER', 'SORT', and 'LIMIT'.

10.- Obtén la empresa con el menor número de empleados.

```
{  
  filter: {  
    $and: [  
      {  
        number_of_employees: {  
          $ne: null  
        }  
      },  
      {  
        number_of_employees: {  
          $ne: 0  
        }  
      }  
    ]  
  },  
  sort: {  
    number_of_employees: 1  
  },  
  limit: 1  
}
```

The screenshot shows the MongoDB Compass interface with the following details:

- Projecto 4-11:** A query builder window with the following parameters:
 - FILTER:** `{} $and: [{ number_of_employees: { $ne: null } }, { number_of_employees: { $ne: 0 } }]`
 - SORT:** `{ number_of_employees: 1 }`
 - LIMIT:** `1`
- Results:** One document is displayed:

```
_id: ObjectId("52cdef7c4bab8bd675297e68")
name: "FeVote"
permalink: "fevote"
crunchbase_url: "http://www.crunchbase.com/company/fevote"
homepage_url: "http://www.fevote.com"
blog_url: "http://blog.fevote.com/"
blog_feed: "http://blog.fevote.com/feed/"
twitter_username: null
category_code: "web"
number_of_employees: 1
founded_year: 2007
founded_month: 1
founded_day: 1
deadpooled_year: null
deadpooled_month: null
deadpooled_day: null
deadpooled_url: null
tag_list: null
alias_list: null
email_address: ""
phone_number: ""
description: null
created_at: "Fri Sep 21 20:21:41 UTC 2007"
updated_at: "Sun Aug 10 13:12:45 UTC 2008"
overview: "<p>FeVote provides suggestion boards for companies and various subject...</p>"
```
- Projecto 4-10:** A partially visible query builder window with a similar structure to Projecto 4-11.

11.- Obtén la empresa con el mayor número de empleados.

```
{  
  filter: {  
    $and: [  
      {  
        number_of_employees: {  
          $ne: null  
        }  
      },  
      {  
        number_of_employees: {  
          $ne: 0  
        }  
      }  
    ]  
  },  
  sort: {  
    number_of_employees: -1  
  },  
  limit: 1  
}
```

The screenshot shows the MongoDB Compass interface. On the left, the sidebar lists databases and collections, with 'sample_training' selected. The main pane displays the 'companies' collection, showing 9.5k documents. A query builder is open, showing the following aggregation pipeline:

```
{  
  $and: [  
    {  
      number_of_employees: {  
        $ne: null  
      }  
    },  
    {  
      number_of_employees: {  
        $ne: 0  
      }  
    }  
  ],  
  sort: {  
    number_of_employees: -1  
  },  
  limit: 1  
}
```

The results pane shows a single document for IBM:

```
_id: ObjectId("52cdef7c4bab8bd67529856a")  
name: "IBM"  
permalink: "ibm"  
crunchbase_url: "http://www.crunchbase.com/company/ibm"  
homepage_url: "http://www.ibm.com"  
blog_url: ""  
blog_feed_url: ""  
twitter_username: "IBM"  
category_code: "software"  
number_of_employees: 380000  
founded_year: 1898  
founded_month: null  
founded_day: null  
deadpooled_year: null  
deadpooled_month: null  
deadpooled_day: null  
deadpooled_url: ""  
tag_list: ""  
alias_list: ""  
email_address: "ews@us.ibm.com"  
phone_number: "914-499-1900"  
description: ""  
created_at: "Fri Mar 14 22:55:52 UTC 2008"  
updated_at: "Sat Jan 04 02:56:24 UTC 2014"  
overview: "<p>IBM, acronym for International Business Machines, is a multinationa..."
```

12.- Obtén la historia más comentada.

```
{  
  sort: {  
    comments: -1  
  },  
  limit: 1  
}
```

The screenshot shows the MongoDB Compass interface on a Mac OS X desktop. The left sidebar lists databases and collections, with 'sample_training' selected. The main pane displays the 'sample_training.stories' collection, which contains 9.8k documents. A query builder is open, showing a SORT clause: {comments: -1} and a LIMIT clause: 1. To the right, a 'Past Queries' panel shows two previous queries: 'Proyecto 4-12' and 'Proyecto 4-9'. The 'Proyecto 4-12' query is identical to the one in the screenshot. The 'Proyecto 4-9' query uses a '\$and' filter with two conditions: comments: \$gt: 10 and comments: \$lte: 30. Below the queries, a JSON document is shown, representing a single story from the collection. The document includes fields like '_id', 'url', 'title', 'comments', 'submit_date', 'topic', 'promote_date', 'id', 'media', 'digest', 'description', 'link', 'user', and 'status'. The status is listed as 'popular'. The bottom of the screen shows the Mac OS X dock with various application icons.

13.- Obtén la historia menos comentada.

```
{  
  filter: {  
    comments: {  
      $gt: 0  
    }  
  },  
  sort: {  
    comments: 1  
  },  
  limit: 1  
}
```

The screenshot shows the MongoDB Compass interface. On the left, the sidebar lists databases and collections, with 'sample_training' selected. In the main pane, the 'Documents' tab is active, displaying a single document from the 'sample_training.stories' collection. The document details the discovery of the world's smallest eel-loach fish, with its ID, URL, title, and other metadata. Above the document, the query parameters are visible: a filter for 'comments > 0', a sort by 'comments' in ascending order, and a limit of 1. To the right, three previous queries are shown in a sidebar: 'Proyecto 4-13' (filtering by comments > 0), 'Proyecto 4-12' (sorting by comments in descending order), and 'Proyecto 4-9' (filtering by a specific \$and condition). The top status bar shows the date and time as 'Lun 21:59'.

Sesión 05: Consultas en MongoDB

Proyecto 5

El proyecto consiste en obtener todas las publicaciones que tengan 50 o más comentarios, que la valoración sea mayor o igual a 80, que cuenten con conexión a Internet vía cable y estén ubicadas en Brazil.

```
[$match: {  
    number_of_reviews: {$gt: 50}  
}, {$match: {  
    "address.country": "Brazil"  
}, {$match: {  
    'review_scores.review_scores_rating': {$gte: 80}  
}, {$match: {  
    amenities: {$in: [/Ethernet/i]}  
}, {$group: {  
    _id: null,  
    total: {  
        $sum: 1  
    }  
}}]
```

The screenshot shows the MongoDB Aggregations interface with the following details:

- Left Sidebar:** Shows the Local database with 10 DBs and 25 Collections. The `sample_airbnb.listingsAndReviews` collection is selected.
- Top Bar:** Displays the collection name, document count (5.6k), total size (90.0MB), average size (16.6KB), and index count (4).
- Aggregation Pipeline:**
 - Stage 1:** `$match` stage with the query: `'review_scores.review_scores_rating': {$gte: 80}`. It shows a sample of 6 documents with fields like `name`, `summary`, `space`, and `description`.
 - Stage 2:** `$group` stage with the query: `amenities: {$in: [/Ethernet/i]}`. It shows a sample of 1 document with fields `_id`, `listing_url`, `name`, `summary`, `space`, and `description`.
- Output:** The final output shows a single document with `_id: null` and `total: 6`.

Sesión 06: Agregaciones

Proyecto 6

El proyecto consiste en obtener, por país, el número de películas que hay de cada género:

```
[$unwind: {  
    path: "$genres",  
    includeArrayIndex: 'string'  
}, {$unwind: {  
    path: "$countries",  
    includeArrayIndex: 'string'  
}, {$group: {  
    _id: {"genres": "$genres", "countries": "$countries"},  
    Total: {$sum: 1}  
}, {$sort: {  
    Total: -1  
}, {$addFields: {  
    Pais: "$_id.countries",  
    Genero: "$_id.genres",  
}, {$project: {  
    _id: 0,  
    Pais: 1,  
    Genero: 1,  
    Total: 1  
}}]
```

The screenshot shows the MongoDB Compass interface with the following details:

- Project:** sample_mflix.Proyecto 6 (view on: sample_mflix.movies)
- Documents Tab:** Shows the results of the aggregation query.
- Results:**
 - Total: 6066 País: "USA" Genero: "Drama"
 - Total: 3843 País: "USA" Genero: "Comedy"
 - Total: 2262 País: "France" Genero: "Drama"
 - Total: 1920 País: "USA" Genero: "Romance"
 - Total: 1777 País: "UK" Genero: "Drama"
- Toolbar:** Includes buttons for Find, Reset, and Options.
- Bottom Bar:** Shows various Mac OS X application icons.

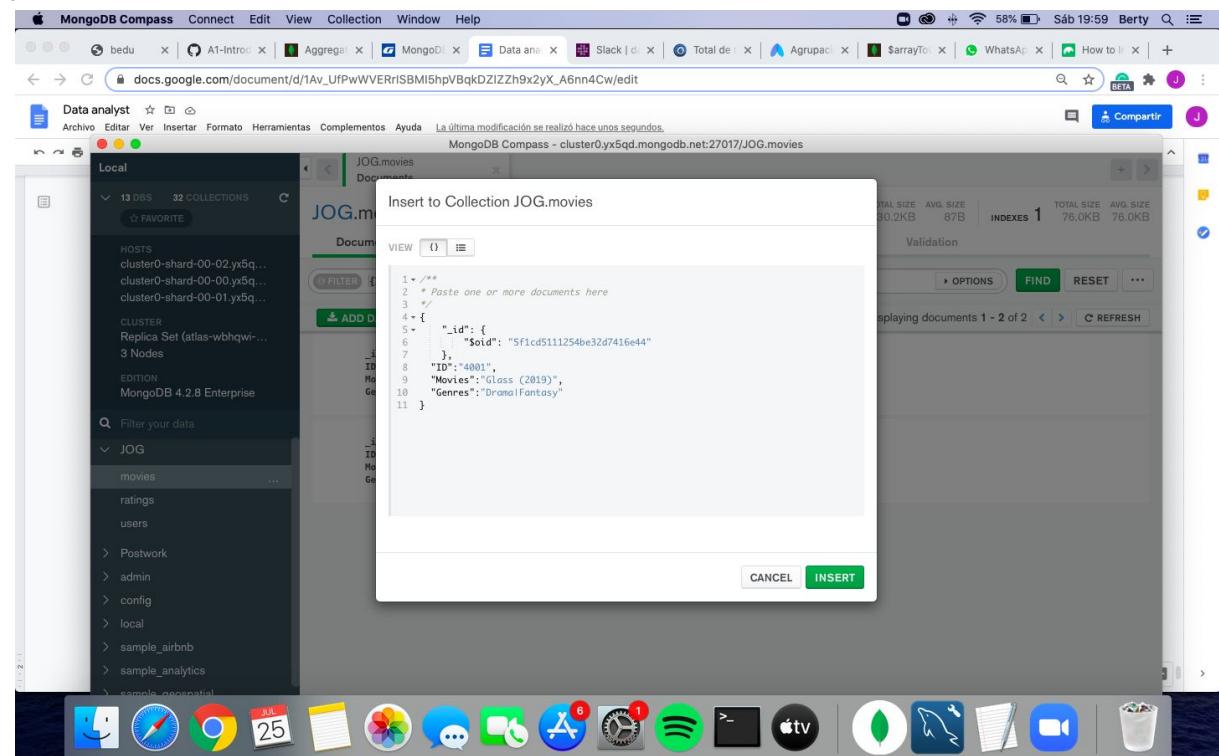
Sesión 07 Configuración de Bases de Datos Locales

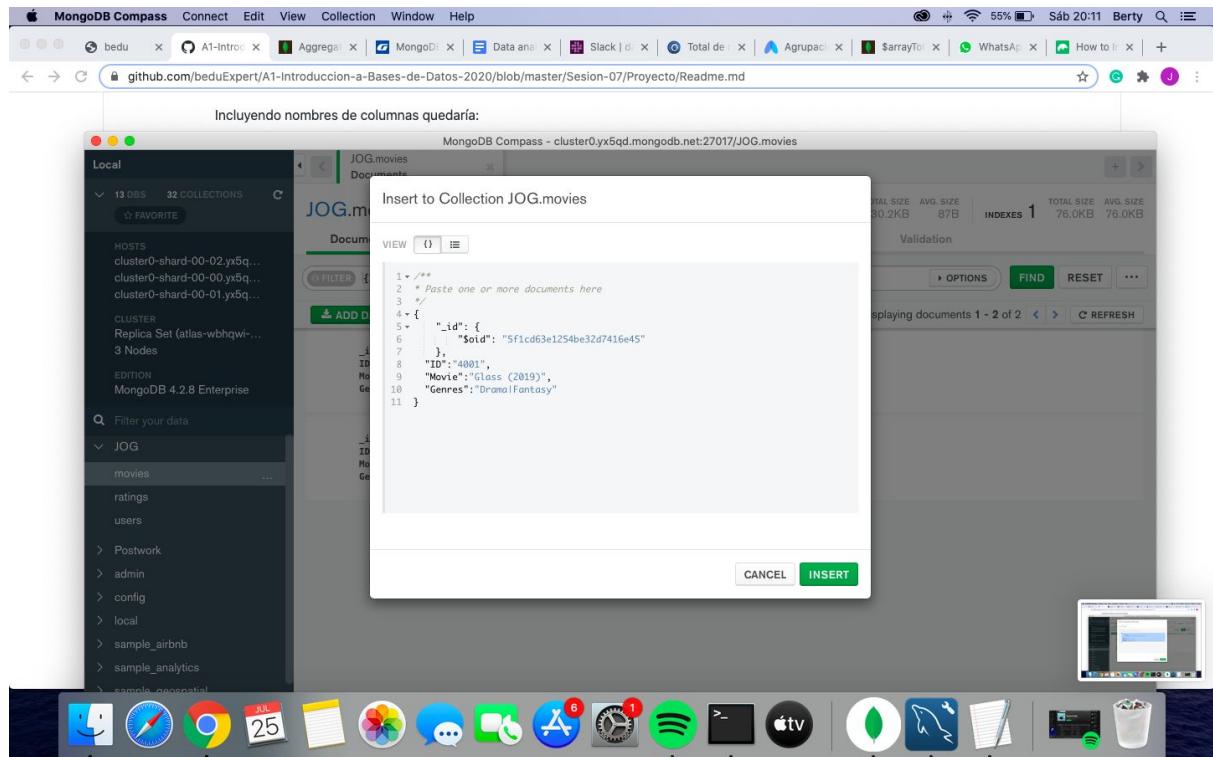
Proyecto 7

1.- Agregar los siguientes registros en formato CSV a la Colección movies

```
4000,Avengers: Endgame (2019),Fantasy|Sci-Fi  
4001,Glass (2019),Drama|Fantasy
```

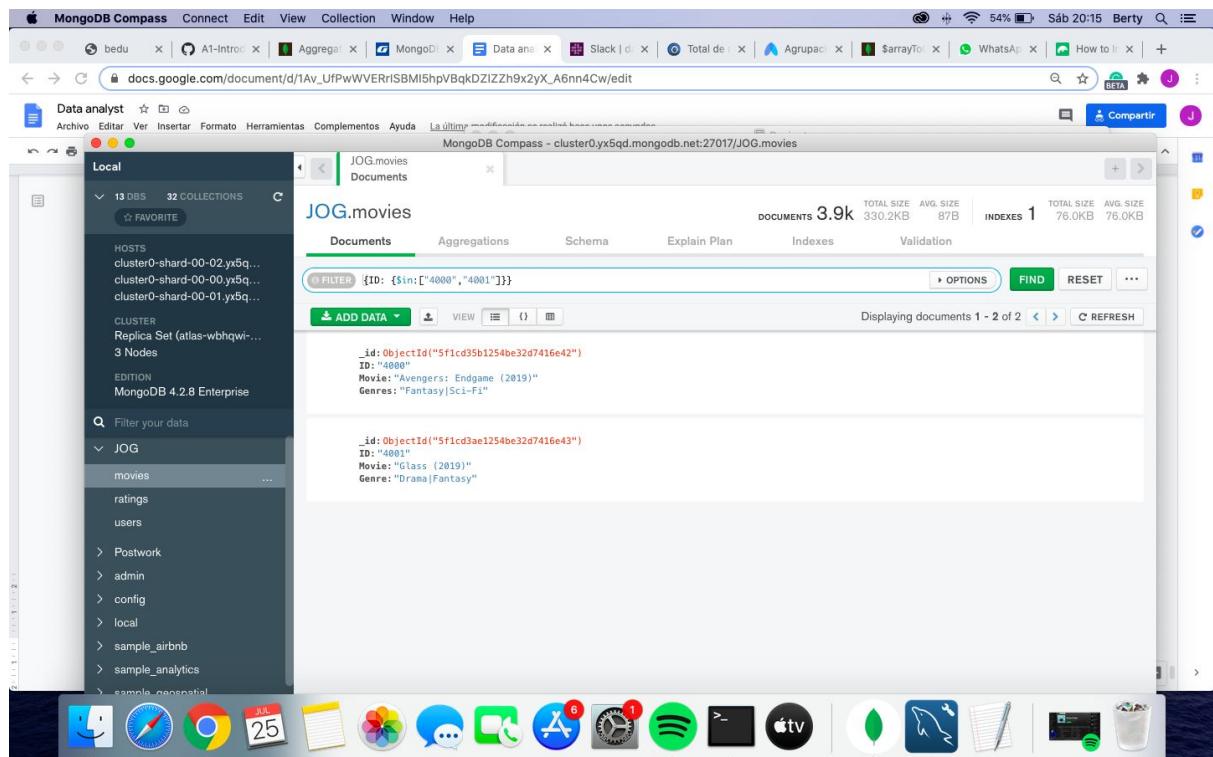
```
{  
    "_id": {  
        "$oid": "5f1cd63e1254be32d7416e45"  
    },  
    "ID":"4000",  
    "Movie":"Avengers: Endgame (2019)",  
    "Genres":"Fantasy|Sci-Fi"  
}  
  
{  
    "_id": {  
        "$oid": "5f1cd63e1254be32d7416e46"  
    },  
    "ID":"4001",  
    "Movie":"Glass (2019)",  
    "Genres":"Drama|Fantasy"  
}
```





Filtro para ambos documentos:

{ID: {\$in:["4000","4001"]}}



2.- Modificar el documento con `id=4001` en la Colección `movies` para que contenga la siguiente información:

```
{  
  id:"4001",  
  titulo:"Glass (2019)",  
  genres:"Drama|Fantasy",  
  valoraciones: [  
    {  
      userid: "1563",  
      movieid: "4001",  
      rating: "4"  
    },  
    {  
      userid: "434",  
      movieid: "4001",  
      rating: "5"  
    }  
  ]  
}
```

The screenshot shows the MongoDB Compass interface on a Mac OS X desktop. The left sidebar lists databases and collections, with 'JOG' selected. The main pane displays the 'movies' collection under the 'JOG.movies' database. A specific document with `_id: "4001"` is selected, showing its detailed structure. The document includes fields for `Movie` ("Glass (2019)") and `Genres` ("Drama|Fantasy"). The `Valoraciones` field is an array containing two objects, each representing a user rating for the movie. The array elements are:

```
_id: ObjectId("5f1cd35b1254be32d7416e42")  
ID: "4001"  
Movie: "Glass (2019)"  
Genres: "Drama|Fantasy"  
  
_id: ObjectId("5f1cd35b1254be32d7416e43")  
ID: "4001"  
Movie: "Glass (2019)"  
Genre: "Drama|Fantasy"  
Valoraciones: Array  
  0: Object  
    userID: "1563"  
    movieid: "4001"  
    rating: "4"  
  1: Object  
    userID: "434"  
    movieid: "4001"  
    rating: "5"
```

Sesión 08 Query competition

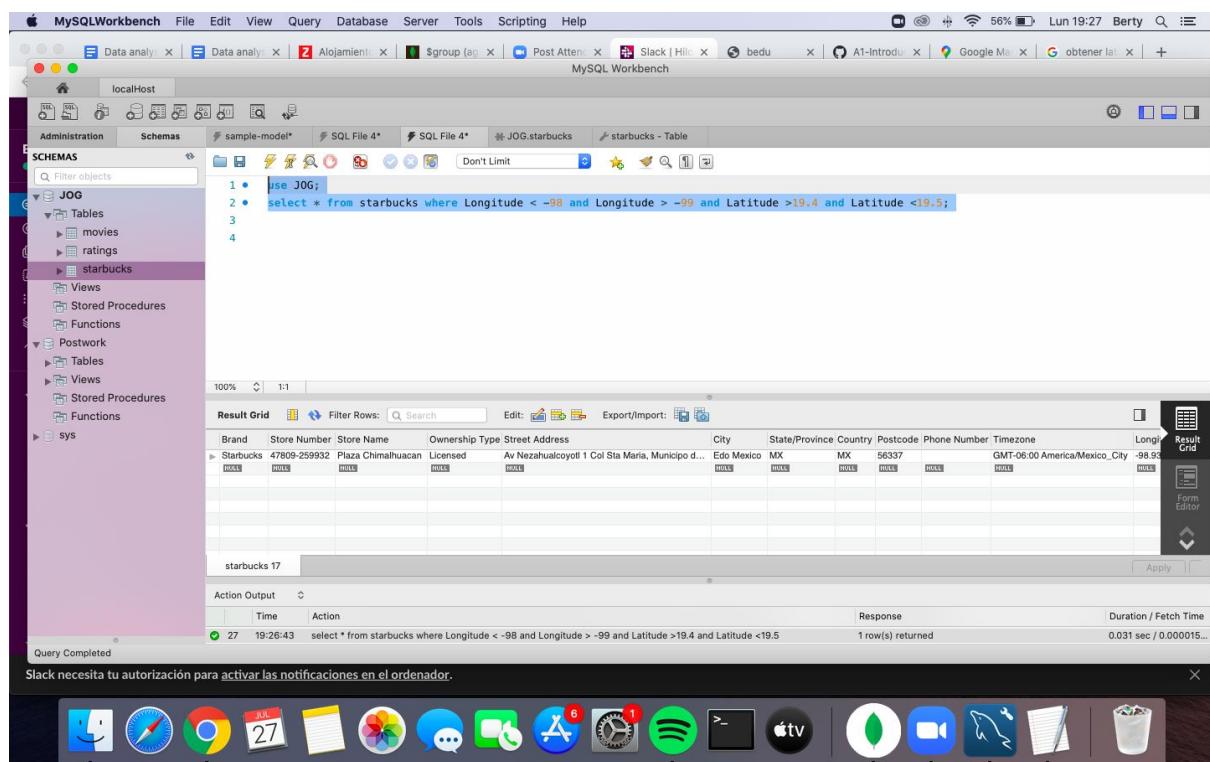
Proyecto 8

Reto 1:

Descarga la fuente de datos de los locales de Starbucks “directory.csv”. Analiza y limpia los datos. Elige MySQL o MongoDB y crea una base de datos para el conjunto de datos del reto. Carga los datos en la base de datos que elegiste y revisa que éstos se muestren correctamente.

Usando la latitud y longitud de tu posición actual, encuentra el Starbucks más cercano a tu posición. Para conocer tu posición actual puedes usar Google Maps para, sólo debes copiar los datos de la URL.

```
use JOG;  
select * from starbucks where Longitude < -98 and Longitude > -99 and Latitude >19.4 and  
Latitude <19.5;
```



The screenshot shows the MySQL Workbench interface. In the top navigation bar, the schema is set to 'JOG'. The SQL editor contains the following query:

```
use JOG;  
select * from starbucks where Longitude < -98 and Longitude > -99 and Latitude >19.4 and  
Latitude <19.5;
```

The results grid displays a single row of data for a Starbucks store in Chimalhuacan, Mexico:

Brand	Store Number	Store Name	Ownership Type	Street Address	City	State/Province	Country	Postcode	Phone Number	Timezone	Longitude	Latitude
Starbucks	47809-259932	Plaza Chimalhuacan	Licensed	Av Nezahualcoyotl 1 Col Sta Maria, Municipio d...	Huixquilucan	Edo Mexico	MX	56337	GMT-06:00	America/Mexico_City	-98.93	19.5

The status bar at the bottom indicates "Query Completed".

Reto 2

Descarga la fuente de datos de los locales del reto “Pandemic (H1N1) 2009.csv”. Analiza y limpia los datos. Elige MySQL o MongoDB y crea una base de datos para el conjunto de datos del reto. Carga los datos en la base de datos que elegiste y revisa que éstos se muestren correctamente. Responde a las siguientes preguntas usando consultas:

¿Cuál fue el país con mayor número de muertes?

```
[$sort: {  
  Cases: -1  
}, {$match: {  
  Country: {  
    $ne: 'Grand Total'  
},  
  Deaths: {  
    $ne: NaN  
}  
}, {$group: {  
  _id: '$Country',  
  Total: {  
    $sum: '$Deaths'  
}  
}, {$sort: {  
  Total: -1  
}, {$limit: 1}]
```

The screenshot shows the MongoDB Aggregations interface. On the left, the 'Local' database is selected, displaying 13 databases and 45 collections. A sidebar on the left lists databases like JOG, COVID-19, and H1N1, with H1N1 currently selected. The main area shows an aggregation pipeline named 'JOG.H1N1'. The pipeline consists of several stages:

- Stage 1: An empty stage.
- Stage 2: A `$sort` stage with the expression `1: { Total: -1 }`.
- Stage 3: An empty stage.
- Stage 4: An empty stage.
- Stage 5: An empty stage.
- Stage 6: An empty stage.

Below the stages, two output samples are shown:

- Output after `$sort` stage (Sample of 20 documents):**

```
_id: "Mexico"  
Total: 2271
```

```
_id: "United States of America"  
Total: 1150
```
- Output after `$limit` stage (Sample of 1 document):**

```
1: 1
```

```
_id: "Mexico"  
Total: 2271
```

At the bottom of the interface, there is a button labeled 'ADD STAGE'.

¿Cuál fue el país con menor número de muertes?

```
[$sort: {  
  Cases: -1  
}, {$match: {  
  Country: {  
    $ne: 'Grand Total'  
},  
  Deaths: {  
    $ne: NaN  
}  
}, {$group: {  
  _id: '$Country',  
  Total: {  
    $sum: '$Deaths'  
}  
}, {$sort: {  
  Total: 1  
}, {$match: {  
  Total: {$gt: 0}  
}, {$limit: 1}]
```

The screenshot shows the MongoDB Aggregations interface with the following details:

- Left Sidebar:** Shows the "Local" database with 13 DBs and 45 collections. The "JOG.H1N1" collection is selected.
- Top Bar:** Title "JOG.H1N1", tabs: "Documents", "Aggregations" (selected), "Schema", "Explain Plan", "Indexes", "Validation". Status: DOCUMENTS 1.8k, TOTAL SIZE 158.7KB, AVG. SIZE 89B; INDEXES 1, TOTAL SIZE 32.0KB, AVG. SIZE 32.0KB.
- Aggregation Pipeline:**
 - Stage 1: `1 {
 2 Total: 1
 3 }`
 - Stage 2: `Output after $match stage (Sample of 20 documents)`
 - `_id: "Antigua and Barbuda"
Total: 0`
 - `_id: "Germany"
Total: 0`
 - Stage 3: `1 {
 2 /* query: The query in MQL.
 3 */
 4 {
 5 Total: {$gt: 0}
 6 }`
 - Stage 4: `Output after $limit stage (Sample of 1 document)`
 - `_id: "Paraguay"
Total: 1`
 - `_id: "Costa Rica"
Total: 2`
 - Stage 5: `1 1`
 - Stage 6: `_id: "Paraguay"
Total: 1`
- Bottom Buttons:** SAMPLE MODE, AUTO PREVIEW.

¿Cuál fue el país con el mayor número de casos?

```
[$sort: {
  Cases: -1
}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  Deaths: {
    $ne: NaN
  }
}}, {$group: {
  _id: '$Country',
  Total: {
    $sum: '$Cases'
  }
}}, {$sort: {
  Total: -1
}}, {$match: {
  Total: {$gt: 0}
}}, {$limit: 1}]
```

The screenshot shows the MongoDB Compass interface with the following details:

- Left Sidebar:** Shows the database structure with "Local" selected, displaying 13 DBs and 45 collections. The "JOG" database is expanded, showing collections like COVID-19, H1N1, Starbucks, movies, ratings, users, and others.
- Top Bar:** The title is "JOG.H1N1 Aggregations". The top right shows statistics: DOCUMENTS 1.8k, TOTAL SIZE 158.7KB, AVG. SIZE 89B; INDEXES 1, TOTAL SIZE 32.0KB, AVG. SIZE 32.0KB.
- Main Area:** The "Aggregations" tab is active. It displays the aggregation pipeline stages:
 - \$match:** Shows a query editor with the following MQL code:

```
1 /**
2  * query: The query in MQL.
3 */
4 {
5   Total: {$gt: 0}
6 }
```
 - Output after \$match stage:** Shows a sample of 20 documents. Two examples are visible:

```
_id:"United States of America"
Total: 340804
```

```
_id:"Mexico"
Total: 142567
```
 - \$limit:** Shows the output after the limit stage, which is a sample of 1 document.

```
_id:"United States of America"
Total: 340804
```

¿Cuál fue el país con el menor número de casos?

```
[$sort: {  
  Cases: -1  
}, {$match: {  
  Country: {  
    $ne: 'Grand Total'  
},  
  Deaths: {  
    $ne: NaN  
}  
}, {$group: {  
  _id: '$Country',  
  Total: {  
    $sum: '$Cases'  
}  
}, {$sort: {  
  Total: 1  
}, {$match: {  
  Total: {$gt: 0}  
}, {$limit: 1}]
```

The screenshot shows the MongoDB Aggregations interface with the following details:

- Left Sidebar:** Shows the database structure with "Local" selected, displaying 13 DBs and 45 collections.
- Top Bar:** Shows the database name "JOG.H1N1" and the stage name "Aggregations".
- Header:** Shows "DOCUMENTS 1.8k" and "INDEXES 1".
- Buttons:** "SAMPLE MODE" and "AUTO PREVIEW".
- Stages:**
 - Stage 1:** An empty stage labeled "1 + {".
 - Stage 2:** A "\$match" stage with the query:

```
1+/**/  
2+  * query: The query in MQL.  
3+ */  
4+ {  
5+   Total: {$gt: 0}  
6+ }
```
 - Stage 3:** An empty stage labeled "1 + {".
 - Stage 4:** A "\$limit" stage with the value "1".
- Output Preview:** Shows the results of each stage:
 - Stage 1:** Total: 1
 - Stage 2:** Two documents:
 - `_id: "France, Martinique, FOC"` Total: 1
 - `_id: "Saint Martin, FOC"` Total: 1
 - Stage 3:** Two documents:
 - `_id: "Netherlands, Curacao, OT"` Total: 1
 - `_id: "United Kingdom, Isle of Man, Crown Dependency"` Total: 1
 - Stage 4:** One document:
 - `_id: "France, French Polynesia, FOC"` Total: 1

¿Cuál fue el número de muertes promedio?

```
[$sort: {
  Cases: -1
}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  Deaths: {
    $ne: NaN
  }
}}, {$group: {
  _id: '$Country',
  Total: {
    $sum: '$Deaths'
  }
}}, {$match: {
  Total: {$gt: 0}
}}, {$group: {
  _id: 'Promedio',
  Promedio: {$avg: '$Total'}
}]]
```

The screenshot shows the MongoDB Aggregations interface with the following details:

- Local** database selected.
- JOG.H1N1** collection selected.
- Aggregations** tab selected.
- Documents** tab selected.
- Aggregation Pipeline:**
 - Stage 1: `$_id: '$Country'`, `Total: { $sum: '$Deaths' }`
 - Stage 2: `$_id: "France"`, `Total: 0`
 - Stage 3: `$_id: "Mar"`, `Total: 0`
- Output after \$match stage:** (Sample of 20 documents)
 - `$_id: "Brazil"`, `Total: 3`
 - `$_id: "Colombia"`, `Total: 19`
- Output after \$group stage:** (Sample of 1 document)
 - `$_id: "Promedio"`,
`Promedio: 194.52380952380952`

¿Cuál fue el número de casos promedio?

```
[$sort: {  
  Cases: -1  
}, {$match: {  
  Country: {  
    $ne: 'Grand Total'  
},  
  Deaths: {  
    $ne: NaN  
}  
}, {$group: {  
  _id: '$Country',  
  Total: {  
    $sum: '$Cases'  
}  
}, {$match: {  
  Total: {$gt: 0}  
}, {$group: {  
  _id: 'Promedio',  
  Promedio: {$avg: '$Total'}  
}}]
```

The screenshot shows the MongoDB Aggregations interface with the following details:

- Left Sidebar:** Shows the Local database with 13 DBs and 45 collections. The JOG.H1N1 collection is selected.
- Top Bar:** JOG.H1N1 Aggregations
- Summary:** DOCUMENTS 1.8k TOTAL SIZE 158.7KB AVG. SIZE 89B INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB
- Stages:**
 - \$match:** query: { Total: { \$gt: 0 } }
 - \$group:** _id: 'Promedio', Promedio: { \$avg: '\$Total' }
- Output after \$match stage:** Sample of 20 documents. Two examples are shown:
 - { _id: "French Polynesia, FOC", Total: 15 }
 - { _id: "Guadalupe, FOC", Total: 2 }
- Output after \$group stage:** Sample of 1 document. One example is shown:
 - { _id: "Promedio", Promedio: 5783.358108108108 }

Top 5 de países con más muertes

```
[{$sort: {
  Cases: -1
}}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  Deaths: {
    $ne: NaN
  }
}}, {$group: {
  _id: '$Country',
  Total: {
    $sum: '$Deaths'
  }
}}, {$sort: {
  Total: -1
}}, {$match: {
  Total: {$gt: 0}
}}, {$limit: 5}]
```

The screenshot shows the MongoDB Aggregations interface with the following details:

- Left Sidebar:** Shows the database structure with "Local" selected, displaying 13 DBs and 45 collections.
- Current Database:** JOG.H1N1
- Aggregation Pipeline:**
 - Stage 1:** \$match (Output after \$match stage)
Query:

```
1 /**
2 * query: The query in MQL.
3 */
4 {
5   Total: {$gt: 0}
6 }
```
 - Stage 2:** \$limit (Output after \$limit stage)
Sample of 5 documents:
 - `_id: "Mexico"` `Total: 2271`
 - `_id: "United States of America"` `Total: 1150`
 - `_id: "Canada"` `Total: 194`
 - `_id: "Argentina"` `Total: 175`
 - `_id: "Brazil"` `Total: 165`
- Top Right Metrics:** DOCUMENTS 1.8k, TOTAL SIZE 156.7KB, AVG. SIZE 89B; INDEXES 1, TOTAL SIZE 32.0KB, AVG. SIZE 32.0KB.
- Bottom Buttons:** ADD STAGE, SAMPLE MODE, AUTO PREVIEW.

Top 5 de países con menos muertes

```
[$sort: {
  Cases: -1
}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  Deaths: {
    $ne: NaN
  }
}}, {$group: {
  _id: '$Country',
  Total: {
    $sum: '$Deaths'
  }
}}, {$sort: {
  Total: 1
}}, {$match: {
  Total: {$gt: 0}
}}, {$limit: 5}]
```

The screenshot shows the MongoDB Compass interface with the following details:

- Left Sidebar (Local):** Shows 13 DBs and 45 collections. The **JOG.H1N1** database is selected.
- Top Bar:** JOG.H1N1 Aggregations. Status: DOCUMENTS 1.8k TOTAL SIZE 168.7KB AVG. SIZE 89B | INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB. Buttons: SAMPLE MODE, AUTO PREVIEW.
- Aggregation Pipeline:**
 - Stage 1:** \$match (Output after \$match stage (Sample of 20 documents))

```
1 /**
2  * query: The query in MQL.
3 */
4 + {
5   Total: {$gt: 0}
6 }
```
 - Stage 2:** \$limit (Output after \$limit stage (Sample of 5 documents))

```
1 5
2
3
4
5
```
- Bottom Buttons:** ADD STAGE, SAVE.

Reto 3

Descarga la fuente de datos de los locales del reto “2019-nCoV-cases-JHU.csv”. Analiza y limpia los datos. Elige MySQL o MongoDB y crea una base de datos para el conjunto de datos del reto. Carga los datos en la base de datos que elegiste y revisa que éstos se muestren correctamente. Responde a las siguientes preguntas usando consultas:

¿Cuál es país con mayor número de casos?

```
[$group: {  
  _id: '$Region',  
  maxCasesPerCountry: {  
    $sum: '$Confirmed'  
  }  
}, {$sort: {  
  maxCasesPerCountry: -1  
}}, {$limit: 1}]
```

The screenshot shows the MongoDB Aggregations interface for the database 'JOG.COVID-19'. The left sidebar lists databases and collections, with 'COVID-19' selected. The main area displays an aggregation pipeline with three stages:

- Stage 1 (\$group):** Groups documents by region and calculates the sum of confirmed cases.
- Stage 2 (\$sort):** Sorts the grouped documents by the maximum cases per country in descending order.
- Stage 3 (\$limit):** Limits the result to the top document (the one with the highest cases).

The output shows the first document from the sorted stage, which is Mainland China with 2369152 cases. The limit stage then outputs this single document.

Document	Output after \$sort stage	Output after \$limit stage
<pre>1: {_id: "Mainland China", maxCasesPerCountry: 2369152}</pre>	<pre>1: {_id: "Mainland China", maxCasesPerCountry: 2369152}</pre>	<pre>1: {_id: "Mainland China", maxCasesPerCountry: 2369152}</pre>

¿Cuál es el país con mayor número de muertes?

```
[$group: {  
  _id: '$Region',  
  maxDeathsPerCountry: {  
    $sum: '$Deaths'  
  }  
}, {$sort: {  
  maxDeathsPerCountry: -1  
}}, {$limit: 1}]
```

The screenshot shows the MongoDB Aggregations interface. On the left, the sidebar lists databases and collections, with 'JOG' and 'COVID-19' selected. The main area displays the aggregation pipeline and its results.

Aggregation Pipeline Stages:

```
1. {  
  _id: '$Region',  
  maxDeathsPerCountry: {  
    $sum: '$Deaths'  
  }  
}  
2. {  
  maxDeathsPerCountry: -1  
}  
3. 1
```

Output after \$sort stage (Sample of 20 documents):

Document
<code>_id: "Nepal" maxDeathsPerCountry: 0</code>
<code>_id: "Algeria" maxDeathsPerCountry: 0</code>
<code>_id: "Mainland China" maxDeathsPerCountry: 65325</code>
<code>_id: "Iran" maxDeathsPerCountry: 368</code>

Output after \$limit stage (Sample of 1 document):

Document
<code>_id: "Mainland China" maxDeathsPerCountry: 65325</code>

Summary:

DOCUMENTS	4.7k
TOTAL SIZE	781.7KB
AVG. SIZE	170B
INDEXES	1
TOTAL SIZE	56.0KB
AVG. SIZE	56.0KB

Usando las coordenadas, encuentra el epicentro del virus.

```
[{$match: {  
    Lat: {  
        $ne: "  
    },  
    Long: {  
        $ne: "  
    }  
}}, {$group: {  
    _id: null,  
    size: {  
        $sum: 1  
    },  
    sumLat: {  
        $sum: '$Lat'  
    },  
    sumLong: {  
        $sum: '$Long'  
    },  
    avgLat: {  
        $avg: '$Lat'  
    },  
    avgLong: {  
        $avg: '$Long'  
    }  
}}, {$project: {  
    checkAvgLat: {  
        $divide: [  
            '$sumLat',  
            '$size'  
        ]  
    },  
    checkAvgLong: {  
        $divide: [  
            '$sumLong',  
            '$size'  
        ]  
    }  
}}]
```

The screenshot shows the MongoDB Aggregations interface with a pipeline defined in the 'JOG.COVID-19' collection. The pipeline consists of three stages: a \$group stage, a \$project stage, and a final output stage.

\$group Stage: This stage groups documents by a null key. It calculates the total size (3), the sum of longitudes (sumLong), and the sum of latitudes (sumLat). The output document is:

```
_id: null  
size: 3  
sumLat: 132931.9493  
sumLong: -43931.8979  
avgLat: 2.48804198492249  
avgLong: -2.631917158632406
```

\$project Stage: This stage projects the calculated values. It includes the size, sumLat, sumLong, and the calculated average values (checkAvgLat and checkAvgLong). The output document is:

```
_id: null  
size: 3  
checkAvgLat: 2.488041984932249  
checkAvgLong: -2.631917158632406
```

Usando el epicentro, encuentra las 5 regiones más cercanas a dicho epicentro.

```
[{$group: {  
    _id: "$Region",  
    Lat: {$avg: "$Lat"},  
    Long: {$avg: "$Long"}  
}}, {$match: {  
    $and: [  
        {Lat: {$lt: 2}},  
        {Long: {$lt: 2}},  
        {Lat: {$gt: -1}},  
        {Long: {$gt: -1}},  
    ]  
}}]
```

The screenshot shows the MongoDB Aggregations interface for the 'JOG.COVID-19' database. The left sidebar lists databases and collections, with 'COVID-19' selected. The main area displays an aggregation pipeline with two stages: '\$group' and '\$match'. The '\$group' stage groups documents by region and calculates average latitude and longitude. The '\$match' stage filters documents where the region's latitude is less than 2 and longitude is less than 2, and also includes regions with latitudes greater than -1 and longitudes greater than -1. The output after the '\$group' stage shows five sample documents, each representing a region with its ID, name, last update, confirmed cases, deaths, and recovered cases. The output after the '\$match' stage shows five sample documents, each representing a region that meets the specified spatial criteria.

```
1. /**
2.  * _id: The id of the group.
3.  * fieldN: The first field name.
4. */
5. {
6.     _id: "$Region",
7.     Lat: {$avg: "$Lat"},  
    Long: {$avg: "$Long"}  
}
```

```
Region: "Mainland China"  
Last Update: 2020-03-03T17:43:02.000+00:00  
Confirmed: 67217  
Deaths: 2835  
Recovered: 36208  
l => 3k 075c
```

```
Region: "South Korea"  
Last Update: 2020-03-03T15:43:02.000+00:00  
Confirmed: 5186  
Deaths: 28  
Recovered: 30  
l => 3k 075c
```

```
_id: "United Arab Emirates"  
Lat: 1.41164705882353  
Long: 3.176470588235294
```

```
_id: "Bahrain"  
Lat: 8.67583333333333  
Long: 16.849999999999998
```

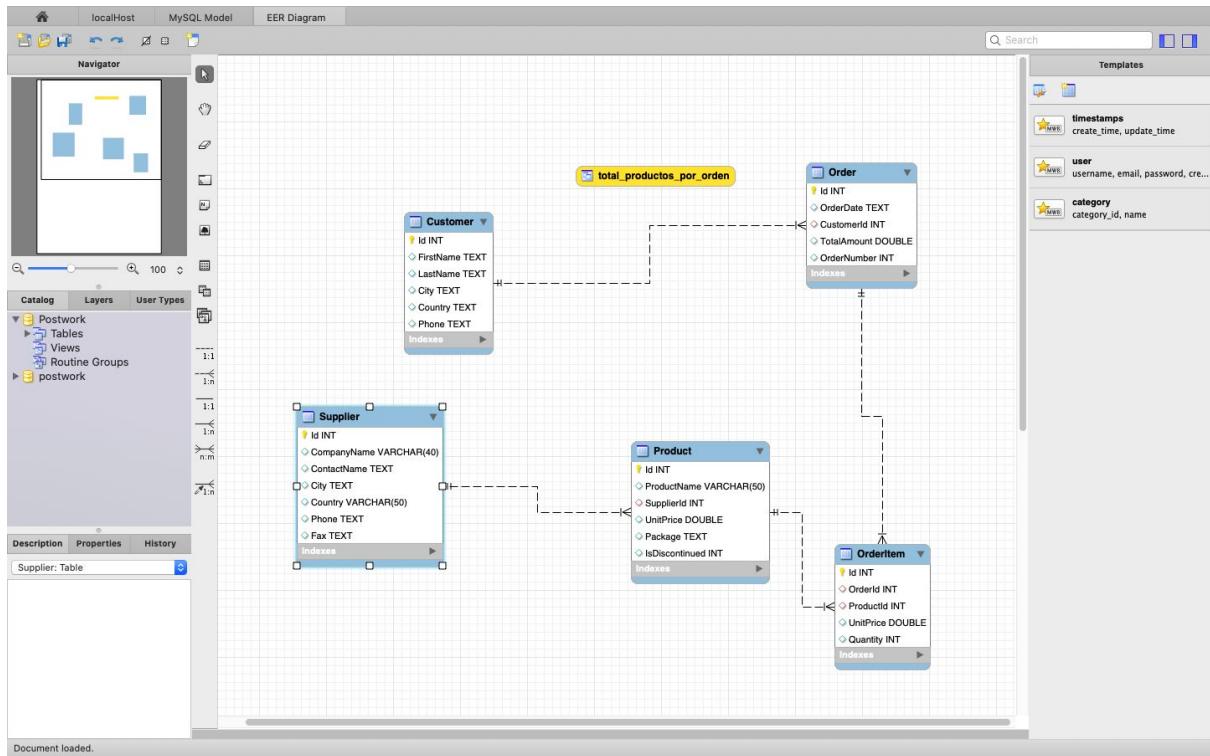
```
1. /**
2.  * query: The query in MQL.
3. */
4. {
5.     $and: [
6.         {Lat: {$lt: 2}},
7.         {Long: {$lt: 2}},
8.         {Lat: {$gt: -1}},
9.         {Long: {$gt: -1}},
10.    ]
11. }
12. }
```

```
_id: "Azerbaijan"  
Lat: 0  
Long: 0
```

```
_id: "North Ireland"  
Lat: 0  
Long: 0
```

Post Work

0.- Añadir diagrama EER para entender la base de datos cargada:



1.- Seleccionar a todos los clientes que radiquen en México.

select * from Customer where Country = "Mexico";

```

1. -- 1.- Seleccionar a todos los clientes que radiquen en Mexico
2. select * from Customer where Country = "Mexico";
3. -- 2.- Selecciona a todos los clientes que no radiquen en USA Y Argentina
4. select * from Customer where Country <> "USA" and Country <> "Argentina";
5. -- 3.- Seleccionar a todos los clientes que su apellido empiece con a o L
6. select * from Customer where LastName like 'a%' or LastName like 'l%';
7. -- 4.- ¿Cuál son todos los datos del producto más barato y más caro
8. select * from Product where UnitPrice = (select min(UnitPrice) from Product) or UnitPrice = (select max(UnitPrice) from Product);
9. -- 5.- ¿Cuántos productos surte cada proveedor?
10. select SupplierId, count(*) from Product group by SupplierId;
11. -- 6.- ¿Cuál es el precio promedio de los artículos suministrados por el proveedor?
12. select SupplierId, round(avg(UnitPrice), 2) as Total from Product group by SupplierId order by Total desc;
  
```

ID	First Name	Last Name	City	Country	Phone
2	Ana	Trujillo	México D.F. México	(5) 555-4729	
3	Antonio	Moreno	México D.F. México	(5) 555-3932	
13	Francisco	Chang	México D.F. México	(5) 555-3392	
58	Guillermo	Fernández	México D.F. México	(5) 552-3745	
80	Miguel	Angel Paolino	México D.F. México	(5) 555-2933	
NULL	NULL	NULL	NULL	NULL	NULL

Customer 1

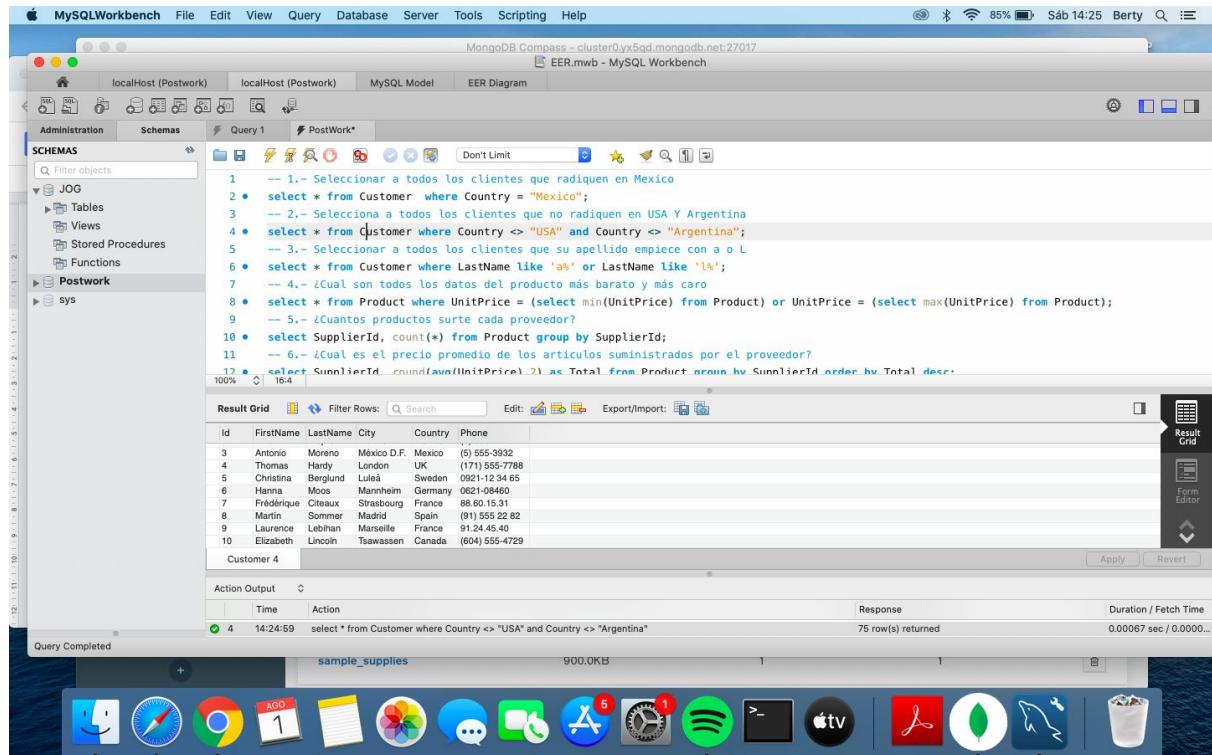
Action Output

Time	Action	Response	Duration / Fetch Time
14:21:20	select * from Customer where Country = "Mexico"	5 row(s) returned	0.018 sec / 0.000015...

Query Completed

-- 2.- Selecciona a todos los clientes que no radiquen en USA Y Argentina

```
select * from Customer where Country <> "USA" and Country <> "Argentina";
```

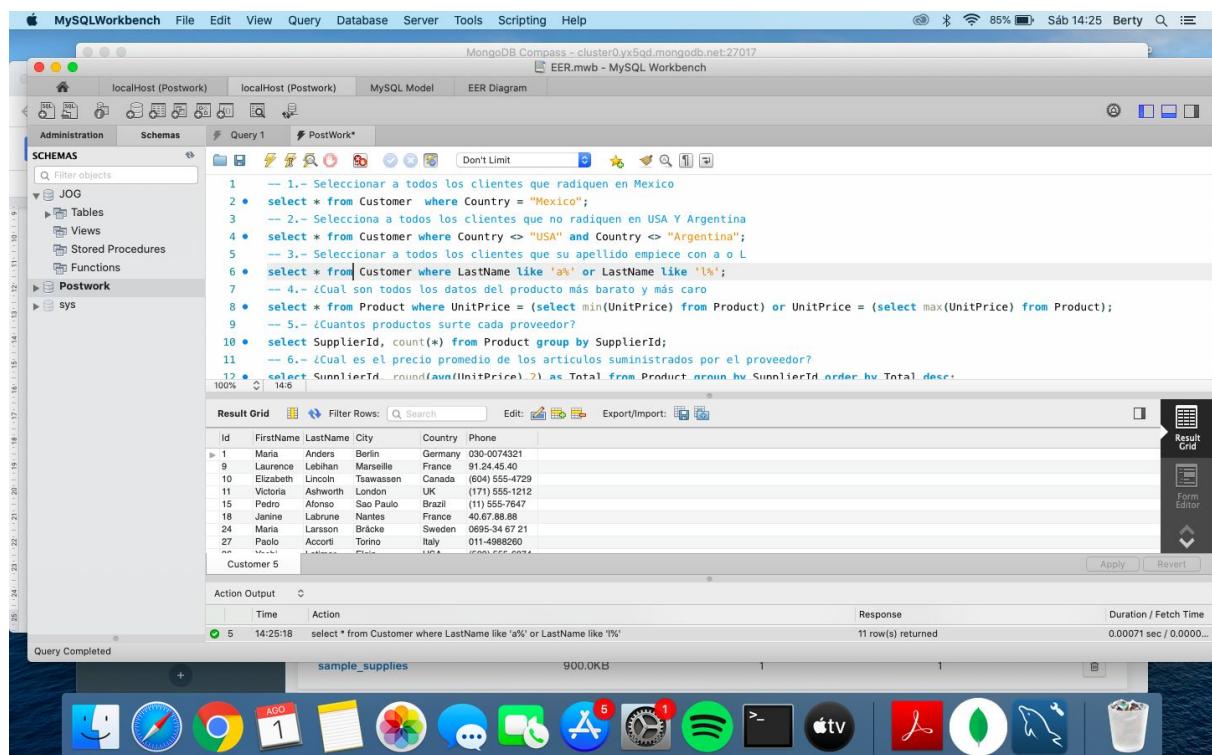


The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** JOG, Postwork (selected), sys.
- Query Editor:** Contains the SQL query: `select * from Customer where Country <> "USA" and Country <> "Argentina";`
- Result Grid:** Displays the results of the query, showing 75 rows of customer information. The columns are Id, FirstName, LastName, City, and Country. The data includes entries like Antonio Moreno from Mexico, Thomas Hardy from UK, and others from various countries like Sweden, Germany, France, Spain, and Canada.
- Action Output:** Shows the execution log: "4 14:24:59 select * from Customer where Country <> "USA" and Country <> "Argentina"" and "75 row(s) returned".
- System Tray:** Shows the date (Sáb 14:25), battery level (85%), and other system icons.

3.- Seleccionar a todos los clientes que su apellido empiece con a o L

```
select * from Customer where LastName like 'a%' or LastName like 'l%';
```

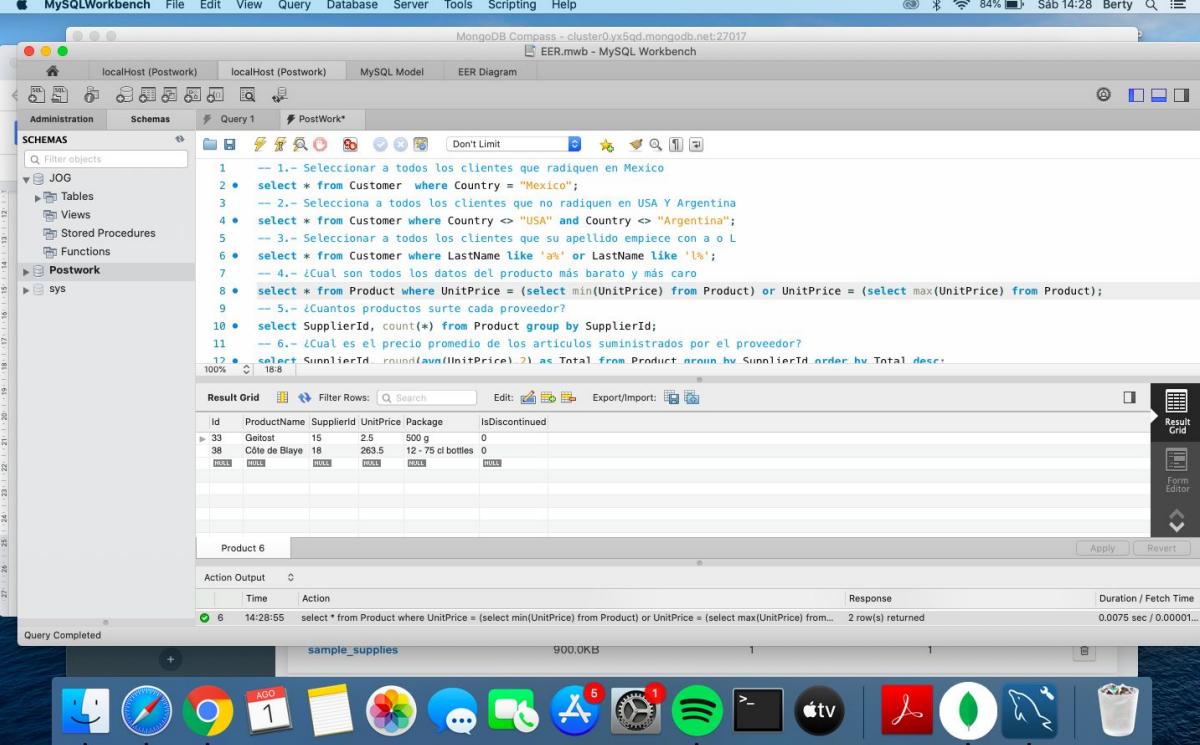


The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** JOG, Postwork (selected), sys.
- Query Editor:** Contains the SQL query: `select * from Customer where LastName like 'a%' or LastName like 'l%'`
- Result Grid:** Displays the results of the query, showing 11 rows of customer information. The columns are Id, FirstName, LastName, City, and Country. The data includes entries like Maria Andress from Berlin, Elizabeth Lincoln from Tsawassen, and others from various countries like France, Brazil, and Italy.
- Action Output:** Shows the execution log: "5 14:25:18 select * from Customer where LastName like 'a%' or LastName like 'l%' " and "11 row(s) returned".
- System Tray:** Shows the date (Sáb 14:25), battery level (85%), and other system icons.

4.- ¿Cual son todos los datos del producto más barato y más caro

select * from Product where UnitPrice = (select min(UnitPrice) from Product) or UnitPrice = (select max(UnitPrice) from Product);



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

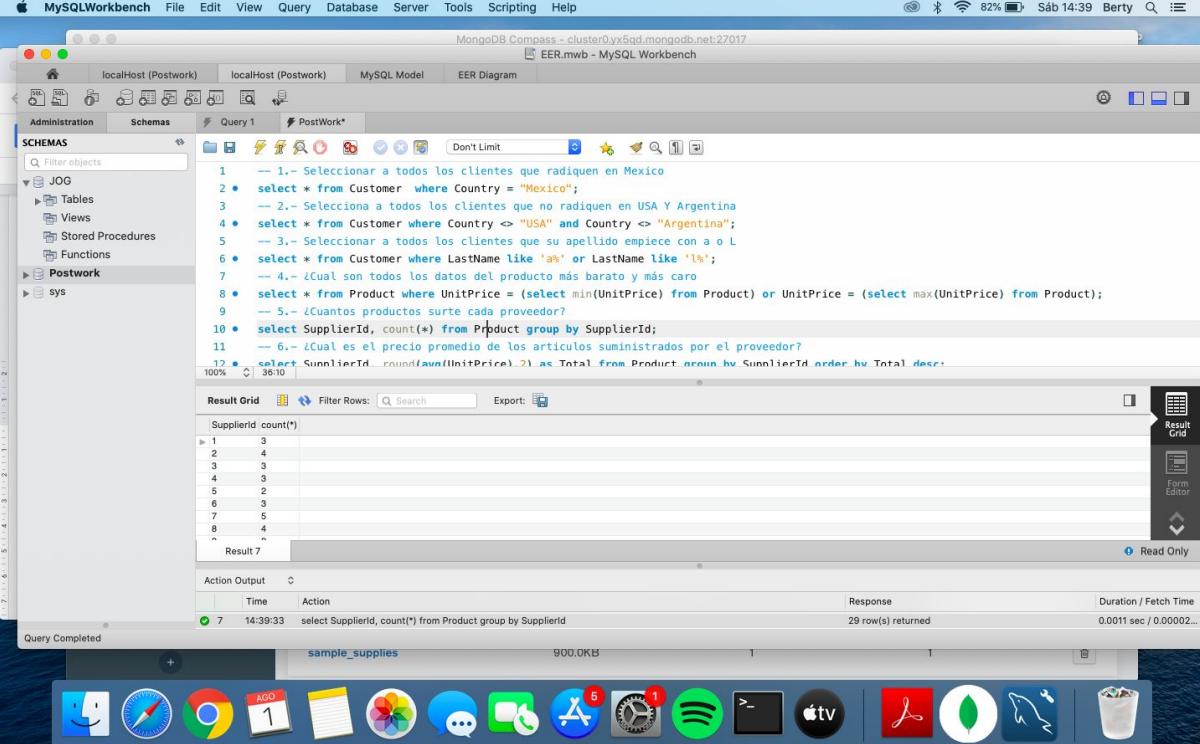
- Query Editor:** Displays the SQL query:

```
1 -- 1.- Seleccionar a todos los clientes que radiquen en Mexico
2 • select * from Customer where Country = "Mexico";
3 -- 2.- Selecciona a todos los clientes que no radiquen en USA Y Argentina
4 • select * from Customer where Country <> "USA" and Country <> "Argentina";
5 -- 3.- Seleccionar a todos los clientes que su apellido empiece con a o L
6 • select * from Customer where LastName like 'a%' or LastName like 'l%';
7 -- 4.- ¿Cuál son todos los datos del producto más barato y más caro
8 • select * from Product where UnitPrice = (select min(UnitPrice) from Product) or UnitPrice = (select max(UnitPrice) from Product);
9 -- 5.- ¿Cuantos productos surte cada proveedor?
10 • select SupplierId, count(*) from Product group by SupplierId;
11 -- 6.- ¿Cuál es el precio promedio de los artículos suministrados por el proveedor?
12 • select SupplierId, round(avg(UnitPrice), 2) as Total from Product group by SupplierId order by Total desc;
13 -- 7.- ¿Cuál es la cantidad de artículos que surten cada proveedor?
```
- Result Grid:** Shows the results for the most expensive and cheapest products.

Id	ProductName	SupplierId	UnitPrice	Package	IsDiscontinued
33	Gelatost	15	2.5	500 g	0
38	Côte de Blaye	18	263.5	12 - 75 cl bottles	0
- Action Output:** Shows the execution details of the query.
- System Tray:** Shows various application icons including Finder, Safari, Mail, Calendar, Notes, Reminders, Stocks, Wallet, App Store, iTunes Store, iBooks, iPhoto, iMovie, GarageBand, Spotify, and others.

5.- ¿Cuantos productos surte cada proveedor?

select SupplierId, count(*) from Product group by SupplierId;



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

- Query Editor:** Displays the SQL query:

```
1 -- 1.- Seleccionar a todos los clientes que radiquen en Mexico
2 • select * from Customer where Country = "Mexico";
3 -- 2.- Selecciona a todos los clientes que no radiquen en USA Y Argentina
4 • select * from Customer where Country <> "USA" and Country <> "Argentina";
5 -- 3.- Seleccionar a todos los clientes que su apellido empiece con a o L
6 • select * from Customer where LastName like 'a%' or LastName like 'l%';
7 -- 4.- ¿Cuál son todos los datos del producto más barato y más caro
8 • select * from Product where UnitPrice = (select min(UnitPrice) from Product) or UnitPrice = (select max(UnitPrice) from Product);
9 -- 5.- ¿Cuantos productos surte cada proveedor?
10 • select SupplierId, count(*) from Product group by SupplierId;
11 -- 6.- ¿Cuál es el precio promedio de los artículos suministrados por el proveedor?
12 • select SupplierId, round(avg(UnitPrice), 2) as Total from Product group by SupplierId order by Total desc;
13 -- 7.- ¿Cuál es la cantidad de artículos que surten cada proveedor?
```
- Result Grid:** Shows the results for the number of products supplied by each supplier.

SupplierId	count(*)
1	3
2	4
3	3
4	3
5	2
6	3
7	5
8	4
- Action Output:** Shows the execution details of the query.
- System Tray:** Shows various application icons including Finder, Safari, Mail, Calendar, Notes, Reminders, Stocks, Wallet, App Store, iTunes Store, iBooks, iPhoto, iMovie, GarageBand, Spotify, and others.

-- 6.- ¿Cuál es el precio promedio de los artículos suministrados por el proveedor?

```
select SupplierId, round(avg(UnitPrice),2) as Total from Product group by SupplierId order by Total desc;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The query editor contains the following SQL code:

```
5 -- 3.- Seleccionar a todos los clientes que su apellido empieza con a o L
6 • select * from Customer where LastName like 'a%' or LastName like 'l%';
7 -- 4.- ¿Cuál son todos los datos del producto más barato y más caro
8 • select * from Product where UnitPrice = (select min(UnitPrice) from Product) or UnitPrice = (select max(UnitPrice) from Product);
9 -- 5.- ¿Cuántos productos surte cada proveedor?
10 • select SupplierId, count(*) from Product group by SupplierId;
11 -- 6.- ¿Cuál es el precio promedio de los artículos suministrados por el proveedor?
12 • select SupplierId, round(avg(UnitPrice),2) as Total from Product group by SupplierId order by Total desc;
13 -- 7.- ¿Cuáles son los 5 nombres de las compañías de los proveedores que más productos surten?
14 • select a.Id, CompanyName, count(*) total from Supplier a
15 left join Product b
16 on a.Id=b.SupplierID
```

The result grid shows the following data:

SupplierId	Total
12	44.68
28	44.50
29	38.90
7	35.57
3	31.67
24	30.93
11	29.71
5	29.50

The status bar at the bottom indicates "Query Completed".

-- 7.- ¿Cuáles son los 5 nombres de las compañías de los proveedores que más productos surten?

```
select a.Id, CompanyName, count(*) total from Supplier a
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The query editor contains the same SQL code as the previous screenshot:

```
5 -- 3.- Seleccionar a todos los clientes que su apellido empieza con a o L
6 • select * from Customer where LastName like 'a%' or LastName like 'l%';
7 -- 4.- ¿Cuál son todos los datos del producto más barato y más caro
8 • select * from Product where UnitPrice = (select min(UnitPrice) from Product) or UnitPrice = (select max(UnitPrice) from Product);
9 -- 5.- ¿Cuántos productos surte cada proveedor?
10 • select SupplierId, count(*) from Product group by SupplierId;
11 -- 6.- ¿Cuál es el precio promedio de los artículos suministrados por el proveedor?
12 • select SupplierId, round(avg(UnitPrice),2) as Total from Product group by SupplierId order by Total desc;
13 -- 7.- ¿Cuáles son los 5 nombres de las compañías de los proveedores que más productos surten?
14 • select a.Id, CompanyName, count(*) total from Supplier a
15 left join Product b
16 on a.Id=b.SupplierID
```

The result grid shows the following data:

Id	CompanyName	total
12	Plutzer Lebensmittelgroßmärkte AG	5
7	Pavlova, Ltd.	5
2	New Orleans Cajun Delights	4
8	Specialty Biscuits, Ltd.	4
1	Exotic Liquids	3

The status bar at the bottom indicates "Query Completed".

-- 8.- ¿Cual es el total de ordenes por país?

```
select Country, round(count(b.TotalAmount),2) Total from Supplier a  
left join `Order` b on a.Id=b.CustomerId group by Country order by Total desc;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The query editor window displays two queries. The first query is the one above, and the second is:

```
— 9.- ¿Cuál es el país con más total de ingresos?  
27 • select Country, round(sum(b.TotalAmount),2) Total from Supplier a  
28 left join `Order` b  
29 on a.Id=b.CustomerId  
30 group by Country  
31 order by Total desc;
```

The results grid shows the following data:

Country	Total
Singapore	30
Sweden	23
USA	22
Canada	20
Japan	20
Sydney	19
France	18
Spain	18
...	...
Result	13

The status bar at the bottom indicates "Query Completed".

-- 9.- ¿Cuál es el país con más total de ingresos?

```
select Country, round(sum(b.TotalAmount),2) Total from Supplier a  
left join `Order` b on a.Id=b.CustomerId group by Country order by Total desc limit 1;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The query editor window displays two queries. The first query is the one above, and the second is:

```
— 9.- ¿Cuál es el país con más total de ingresos?  
27 • select Country, round(sum(b.TotalAmount),2) Total from Supplier a  
28 left join `Order` b  
29 on a.Id=b.CustomerId  
30 group by Country  
31 order by Total desc;
```

The results grid shows the following data:

Country	Total
Singapore	113236.68

The status bar at the bottom indicates "Query Completed".

-- 10.- ¿Cuáles son las 3 ciudades con más órdenes?

```
select City, count(b.TotalAmount) Total from Supplier a left join `Order` b on a.Id=b.CustomerId group by City order by Total desc limit 3;
```

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

```
30 group by Country
31 order by Total desc
32 limit 1;
33 -- 10.- ¿Cuáles son las 3 ciudades con mas órdenes?
34 • select City, count(b.TotalAmount) Total from Supplier a
35 left join `Order` b
36 on a.Id=b.CustomerId
37 group by City
38 order by Total desc
39 limit 3;
40 -- 11.- ¿Cuál es el nombre de los 5 productos más vendidos?
41 • create view Total_productos_por_orden as (select a.Id, a.ProductName, sum(b.Quantity) Total from Product a
```

The result grid shows the top 3 cities with the most orders:

City	Total
Singapore	30
Wendy Mackenzie	19
Oviedo	18

The status bar at the bottom indicates "Query Completed" at 14:50:29.

-- 11.- Crea una vista que muestra el total de unidades vendidas por producto

```
create view Total_productos_por_orden as (select a.Id, a.ProductName, sum(b.Quantity)
Total from Product a left join OrderItem b on a.Id=b.ProductId group by a.Id order by Total
desc);
```

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

```
37 group by City
38 order by Total desc
39 limit 3;
40 -- 11.- Crea una vista que muestra el total de unidades vendidas por producto
41 • create view Total_productos_por_orden as (select a.Id, a.ProductName, sum(b.Quantity) Total from Product a
42 left join OrderItem b
43 on a.Id=b.ProductId
44 group by a.Id
45 order by Total desc);
46 -- 11a.- ¿Cuál es el nombre de los 5 productos más vendidos?
47 • select * from Total_productos_por_orden limit 5;
48 -- 12.- ¿Cuál es el producto que no se ha vendido?
49 • select * from Total_productos_por_orden where Total is NULL;
50 -- 13.- ¿Cuál fue el total de la venta del producto mostrando el proveedor?
51 • select a.Id, a.ProductName, round(sum(b.UnitPrice),2) Total, c.CompanyName from Product a
52 left join OrderItem b
53 on a.Id=b.ProductId
```

The result grid shows the top 5 products with the highest sales volume:

Id	ProductName	Total
60	Commençant Period	1577
59	Raspberia Courdasmuff	1496
31	Gorgonzola Telino	1387
86	Gnocchi di nonna Alice	1263
16	Pavlova	1158

The status bar at the bottom indicates "Query Completed" at 14:59:27.

-- 11a.- ¿Cual es el nombre de los 5 productos más vendidos?

```
select * from Total_productos_por_orden limit 5;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the schema structure under 'Postwork'. The main area shows a query editor with the following SQL code:

```
37 group by City
38 order by Total desc
39 limit 3;
-- 11.- Crea una vista que muestra el total de unidades vendidas por producto
41 • create view Total_productos_por_orden as (select a.Id, a.ProductName, sum(b.Quantity) Total from Product a
42 left join OrderItem b
43 on a.Id=b.ProductId
44 group by a.Id
45 order by Total desc);
-- 11a.- ¿Cual es el nombre de los 5 productos más vendidos?
47 • select * from Total_productos_por_orden limit 5;
-- 12.- ¿Cual es el producto que no se ha vendido?
49 • select * from Total_productos_por_orden where Total is NULL;
-- 13.- ¿Cuál fue el total de la venta del producto mostrando el proveedor?
51 • select a.Id, a.ProductName, round(sum(b.UnitPrice),2) Total, c.CompanyName from Product a
52 left join OrderItem b
53 on a.Id=b.ProductId
```

The result grid shows the following data:

ID	ProductName	Total
60	Camembert Pierrot	1577
59	Raclette Courteau	1496
31	Gorgonzola Telino	1387
58	Gnocchi di nonna Alice	1263
16	Pavlova	1158

At the bottom, the status bar indicates "Query Completed".

-- 12.- ¿Cual es el producto que no se ha vendido?

```
select * from Total_productos_por_orden where Total is NULL;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the schema structure under 'Postwork'. The main area shows a query editor with the following SQL code:

```
37 group by City
38 order by Total desc
39 limit 3;
-- 11.- Crea una vista que muestra el total de unidades vendidas por producto
41 • create view Total_productos_por_orden as (select a.Id, a.ProductName, sum(b.Quantity) Total from Product a
42 left join OrderItem b
43 on a.Id=b.ProductId
44 group by a.Id
45 order by Total desc);
-- 11a.- ¿Cual es el nombre de los 5 productos más vendidos?
47 • select * from Total_productos_por_orden limit 5;
-- 12.- ¿Cual es el producto que no se ha vendido?
49 • select * from Total_productos_por_orden where Total is NULL;
-- 13.- ¿Cuál fue el total de la venta del producto mostrando el proveedor?
51 • select a.Id, a.ProductName, round(sum(b.UnitPrice),2) Total, c.CompanyName from Product a
52 left join OrderItem b
53 on a.Id=b.ProductId
```

The result grid shows the following data:

ID	ProductName	Total
78	Stroopwafels	NULL

At the bottom, the status bar indicates "Query Completed".

-- 13.- ¿Cual fue el total de la venta del producto mostrando el proveedor?

```
select a.Id, a.ProductName, round(sum(b.UnitPrice),2) Total, c.CompanyName from Product
a left join OrderItem b on a.Id=b.ProductId left join Supplier c on c.Id=a.SupplierId group by
a.Id order by Total desc;
```

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

- Schemas:** JOG, Postwork, sys.
- Query Editor:** Contains the SQL code for question 13.
- Result Grid:** Displays the results of the query, showing columns: Id, ProductName, Total, CompanyName. One row is highlighted: Manjimup Dried Apples (Id: 51) with a Total of 1971.60.
- Action Output:** Shows the execution time (26 15:04:07) and number of rows returned (78).

Id	ProductName	Total	CompanyName
36	Côte de Blaye	5902.40	Aux joyeux ecclésiastiques
29	Thüringer Rostbratwurst	3713.38	Plutzer Lebensmittelgroßmärkte AG
59	Raclette Courdavault	2761.00	Gai pâturage
62	Tarte au sucre	2227.80	Foires d'épaves'
51	Manjimup Dried Apples	1971.60	Gray
56	Gnocchi di nonna Alice	1770.00	Pasta Buttini s.r.l.
60	Camembert Pierrot	1638.80	Gai pâturage
19	Le Carré Bleu	1638.00	Le Carré Bleu
28	Rössle Sauerkraut	1385.20	Plutzer Lebensmittelgroßmärkte AG
17	Alice Mutton	1349.40	Pavlova, Ltd.
72	Mozzarella di Giovanni	1217.40	Formaggi Fortini s.r.l.
20	Sir Rodney's Marmalade®	1215.00	Specialty Biscuits, Ltd.
43	Ipoh Coffee	1205.20	Leka Trading
69	Gubbrandsdalost	1036.80	Norske Meierier

-- 14.- ¿Cual son los 5 clientes que más productos compraron y cuanto dinero han gastado?

```
select concat(a.FirstName, " ", a.LastName) Nombre, round(sum(b.TotalAmount),2)
```

```
Total_Quantity, sum(c.Quantity) Total_Amount from Customer a left join `Order` b on
a.Id=b.CustomerId left join OrderItem c on b.Id=c.OrderId group by Nombre order by
Total_Quantity desc limit 5;
```

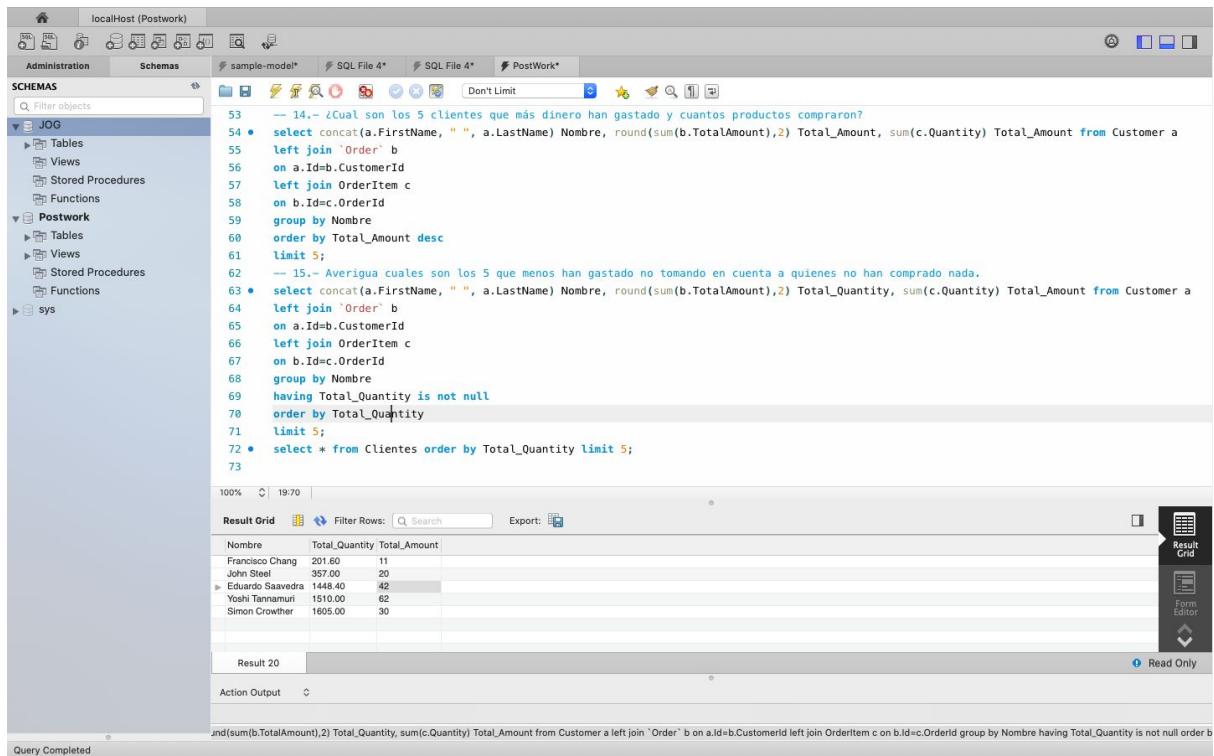
The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** JOG, Postwork, sys.
- Query Editor:** Contains the SQL code for question 14.
- Result Grid:** Displays the results of the query, showing columns: Nombre, Total_Amount, Total_Quantity. Five rows are listed.
- Action Output:** Shows the execution time (50 15:30:39) and number of rows returned (5).

Nombre	Total_Amount	Total_Quantity
Jose Manzanares	48107.60	4958
Howard Mendel	426349.25	3456
Honf Kloss	417617.35	3961
Paula Wilson	179134.60	1383
Patricia McKenna	156949.88	1684

-- 15.- Averigua cuales son los 5 que menos han gastado no tomando en cuenta a quienes no han comprado nada.

```
select concat(a.FirstName, " ", a.LastName) Nombre, round(sum(b.TotalAmount),2)
Total_Quantity, sum(c.Quantity) Total_Amount from Customer a left join `Order` b on
a.Id=b.CustomerId left join OrderItem c on b.Id=c.OrderId group by Nombre having
Total_Quantity is not null order by Total_Quantity limit 5;
```



The screenshot shows a SQL editor interface with the following details:

- Title Bar:** localHost (Postwork)
- Toolbar:** Includes icons for Home, Open, Save, Print, Copy, Paste, Find, Replace, Undo, Redo, and others.
- Menu Bar:** Administration, Schemas, sample-model*, SQL File 4*, SQL File 4*, PostWork*
- Schemas:** JOG (selected), Administration, Postwork, sys
- Query Editor:** Contains two SQL queries. The second query is highlighted.

```
53 -- 14.- ¿Cuáles son los 5 clientes que más dinero han gastado y cuantos productos compraron?
54 • select concat(a.FirstName, " ", a.LastName) Nombre, round(sum(b.TotalAmount),2) Total_Amount, sum(c.Quantity) Total_Amount from Customer a
55   left join `Order` b
56   on a.Id=b.CustomerId
57   left join OrderItem c
58   on b.Id=c.OrderId
59   group by Nombre
60   order by Total_Amount desc
61   limit 5;
62 -- 15.- Averigua cuales son los 5 que menos han gastado no tomando en cuenta a quienes no han comprado nada.
63 • select concat(a.FirstName, " ", a.LastName) Nombre, round(sum(b.TotalAmount),2) Total_Quantity, sum(c.Quantity) Total_Amount from Customer a
64   left join `Order` b
65   on a.Id=b.CustomerId
66   left join OrderItem c
67   on b.Id=c.OrderId
68   group by Nombre
69   having Total_Quantity is not null
70   order by Total_Quantity
71   limit 5;
72 • select * from Clientes order by Total_Quantity limit 5;
73
```
- Result Grid:** Shows the results of the second query:

Nombre	Total_Quantity	Total_Amount
Francisco Chang	20.00	11
John Smith	287.00	20
Edwards Saavedra	1448.40	62
Yoshi Tannamuri	1510.00	62
Simon Crowther	1605.00	30
- Status Bar:** Query Completed

16.- Obtén los datos de contacto de cada compañía de proveedores

```
{  
  project: {  
    CompanyName: 1,  
    Phone: 1,  
    Fax: 1  
  }  
}
```

The screenshot shows the MongoDB Compass interface with the following details:

- Project:** JOG_Postwork.Supplier
- Documents:** 29
- Indexes:** 1
- Filter:** {CompanyName: 1, Phone: 1, Fax: 1}
- Sort:** (None)
- Collation:** (None)
- Options:** MAXITEMS: 500, SKIP: 0, LIMIT: 0
- Results:** Displays 29 documents. The first few are:

 - `_id: ObjectId("5f1f5de719f51f02bc58754d")
CompanyName: "Exotic Liquids"
Phone: "(171) 555-2222"
Fax: "NULL"`
 - `_id: ObjectId("5f1f5de719f51f02bc58754e")
CompanyName: "New Orleans Cajun Delights"
Phone: "(180) 555-4822"
Fax: "NULL"`
 - `_id: ObjectId("5f1f5de719f51f02bc58754f")
CompanyName: "Grandma Kelly's Homestead"
Phone: "(313) 555-5735"
Fax: "(313) 555-3349"`
 - `_id: ObjectId("5f1f5de719f51f02bc587550")
CompanyName: "Tokyo Traders"
Phone: "(03) 3555-5011"
Fax: "NULL"`

17.- Filtra a los proveedores del reino unido

```
{  
  filter: {  
    Country: 'UK'  
  }  
}
```

The screenshot shows the MongoDB Compass interface on a Mac OS X desktop. The title bar reads "MongoDB Compass - cluster0.yx5qd.mongodb.net:27017". The left sidebar shows the database structure: "Local" with "13 DBs" and "38 COLLECTIONS", and a "JOG_Postwork" database containing collections like "Supplier", "admin", "config", "local", etc. The main pane displays the "JOG_Postwork.Supplier" collection with 29 documents. A query builder is open with the following filter: {Country: 'UK'}. The results show two documents:

```
_id:ObjectId("5f1f5de719f51f02bc58754d")
Id:1
CompanyName:"Exotic Liquids"
ContactName:"Charlotte Cooper"
City:"London"
Country:"UK"
Phone:"(171) 555-2222"
Fax:"NULL"

_id:ObjectId("5f1f5de719f51f02bc587554")
Id:8
CompanyName:"Specialty Biscuits Ltd."
ContactName:"Peter Wilson"
City:"Manchester"
Country:"UK"
Phone:"(161) 555-4448"
Fax:"NULL"
```

18.- Muestra a los clientes solo de México o Alemania

```
{
  filter: {
    $or: [
      {
        Country: 'Germany'
      },
      {
        Country: 'Mexico'
      }
    ]
  }
}
```

The screenshot shows the MongoDB Compass interface with the following details:

- Left Sidebar:** Shows the database structure with 'Local' selected, 13 DBs, and 38 collections.
- Top Bar:** Title 'JOG_Postwork.Customer Documents', tabs: 'Documents', 'Aggregations', 'Schema', 'Explain Plan', 'Indexes', 'Validation'.
- Search Bar:** Contains the query: '\$or: [{Country: "Germany"}, {Country: "Mexico"}]'.
- Results Table:** Displays 91 documents. The first few results are:
 - Document 1:** _id: ObjectId("5f1f5ef519f51f02bc58756a"), Id: 1, FirstName: "Maria", LastName: "Anders", City: "Berlin", Country: "Germany", Phone: "+030-0074321"
 - Document 2:** _id: ObjectId("5f1f5ef519f51f02bc58756b"), Id: 2, FirstName: "Ana", LastName: "Trujillo", City: "Mexico D.F.", Country: "Mexico", Phone: "(5) 555-4729"
 - Document 3:** _id: ObjectId("5f1f5ef519f51f02bc58756c"), Id: 3, FirstName: "Antonio", LastName: "Moreno", City: "Mexico D.F.", Country: "Mexico", Phone: "(5) 555-3932"
 - Document 4:** _id: ObjectId("5f1f5ef519f51f02bc58756f"), Id: 6, FirstName: "Hanna", LastName: "Mann", City: "Mannheim", Country: "Germany", Phone: "+0621-08460"
- Right Panel:** Shows three past queries with their respective dates and filters:
 - Mon Jul 27 2020 18:17:18 GMT-0500 (Ce...: '\$or: [{Country: 'Germany'}, {Country: 'Mexico'}]')
 - Mon Jul 27 2020 18:14:29 GMT-0500 (Ce...: '\$or: [{Country: 'Germany'}, {Country: 'Mexico'}]')
 - Mon Jul 27 2020 18:14:22 GMT-0500 (Ce...: '\$and: [{Country: 'Germany'}, {Country: 'Mexico'}]')
 - Mon Jul 27 2020 18:14:14 GMT-0500 (Ce...

19.- Ubicar todas las órdenes con precio de unidad mayor a 30

```
{  
  filter: {  
    UnitPrice: {  
      $gt: 30  
    }  
  }  
}
```

The screenshot shows the MongoDB Compass interface with the following details:

- Left Sidebar:** Shows the database structure with 13 DBs and 38 collections. The current collection is **JOG_Postwork.Order**.
- Top Bar:** Displays the collection name, document count (2.2k), total size (191.5KB), and index count (1).
- Query Filter Panel:** Contains the query: `{UnitPrice: {$gt: 30}}`. It includes options for **FILTER**, **PROJECT**, **SORT**, and **COLLATION**. The **MAXITEMS** field is set to 5000.
- Result Preview:** Shows five documents from the results. Each document includes fields: _id, Id, OrderId, ProductId, UnitPrice, and Quantity. The first document has UnitPrice: 34.8 and the second has UnitPrice: 42.4.
- Right Panel:** Shows the query history with the timestamp **Mon Jul 27 2020 18:22:45 GMT-0500 (C...** and the same query: `{UnitPrice: {$gt: 30}}`.

20.- ¿Cuál es el producto más caro de la lista?

```
{  
  sort: {  
    UnitPrice: -1  
  },  
  limit: 1  
}
```

The screenshot shows the MongoDB Compass interface on a Mac OS X desktop. The main window displays the 'JOG_Postwork.Product' collection with 78 documents. A search query is applied, sorting by UnitPrice in descending order (-1) and limiting the result to 1 document. The result is a single document for 'Côte de Blaye' with a UnitPrice of 263.5.

_id	ProductName	SupplierId	UnitPrice	Package	IsDiscontinued
ObjectId("5f1f624b19f51f02bc588193")	Côte de Blaye	18	263.5	12 - 75 cl bottles	false

21.- ¿Muestra por agregación todos los productos que no estan descontinuados?

```
[{$match: {  
    IsDiscontinued: false  
}}]
```

The screenshot shows the MongoDB Compass interface. On the left, the sidebar lists databases and collections, with 'JOG_Postwork' selected. In the main area, the 'Aggregations' tab is active, showing a pipeline stage: '\$match'. The pipeline stage contains the query:`1 <-- /*
2 * query: The query in MQL.
3 */
4 + {
5 IsDiscontinued: false
6 }`

Below the pipeline, two document samples are shown:

Document 1	Document 2
<pre>_id: ObjectId("5f1f624b19f51f02bc58816e") Id: 1 ProductName: "Chai" SupplierId: 1 UnitPrice: 18 Package: "10 boxes x 20 bags" IsDiscontinued: false</pre>	<pre>_id: ObjectId("5f1f624b19f51f02bc58816e") Id: 2 ProductName: "Chang" SupplierId: 1 UnitPrice: 19 Package: "24 - 12 oz bottle" IsDiscontinued: false</pre>

22.- ¿Cuantos productos surte cada proveedor?

```
[$match: {  
    IsDiscontinued: false  
}, {$group: {  
    _id: "$SupplierId",  
    Total: {  
        $sum:1  
    }  
}}]
```

The screenshot shows the MongoDB Compass interface with the following details:

- Hosts:** cluster0-shard-00-02.y5q... (selected), cluster0-shard-00-00.y5q..., cluster0-shard-00-01.y5q...
- Cluster:** Replica Set (atlas-wbhqw-...)
- Edition:** MongoDB 4.2.8 Enterprise
- Collection:** JOG_Postwork.Product
- Aggregations Tab:** The pipeline consists of two stages:
 - \$match:** Filters documents where IsDiscontinued is false.
 - \$group:** Groups by SupplierId and calculates the total count of products.
- Output after \$match stage:** Shows sample documents for SupplierId 1 and 2.
- Output after \$group stage:** Shows two groups: one for SupplierId 17 with a total of 3, and another for SupplierId 3 with a total of 3.
- Document Count:** 78
- Total Size:** 11.1KB
- Avg. Size:** 145B
- Indexes:** 1
- Total Size:** 20.0KB
- Avg. Size:** 20.0KB

23.- Agrega como un array todos los productos suministrados no descontinuados por cada proveedor y el promedio de los precios de los productos de mayor a menor.

```
[{$match: {  
    IsDiscontinued: false  
}}, {$group: {  
    _id: "$SupplierId",  
    Total: {  
        $sum:1  
    },  
    Productos: {$push:"$ProductName"},  
    Precioneto: {$avg: "$UnitPrice"}  
}}, {$sort: {  
    Total: -1  
}}]
```

The screenshot shows the MongoDB Compass interface with the 'JOG_Postwork.Product' collection selected. The 'Aggregations' tab is active, displaying the following pipeline:

```
1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4 */
5 {
6   "_id": "$SupplierId",
7   "Total": {
8     "$sum:1
9   },
10  "Productos: {$push:"$ProductName"},  
11  "Precioneto: {$avg: "$UnitPrice"}  
12 }
```

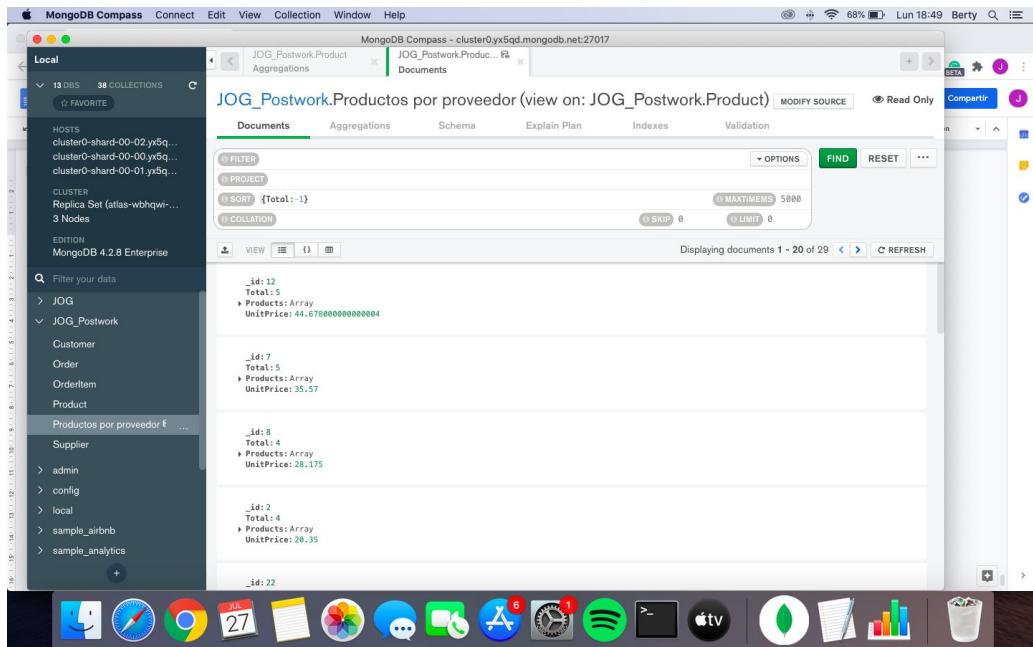
The output after the \$group stage shows two documents:

- Document 1: _id: 18, Total: 2, Productos: Array, Precioneto: 140.75
- Document 2: _id: 20, Total: 2, Productos: Array, Precioneto: 32.725

The output after the \$sort stage shows two documents:

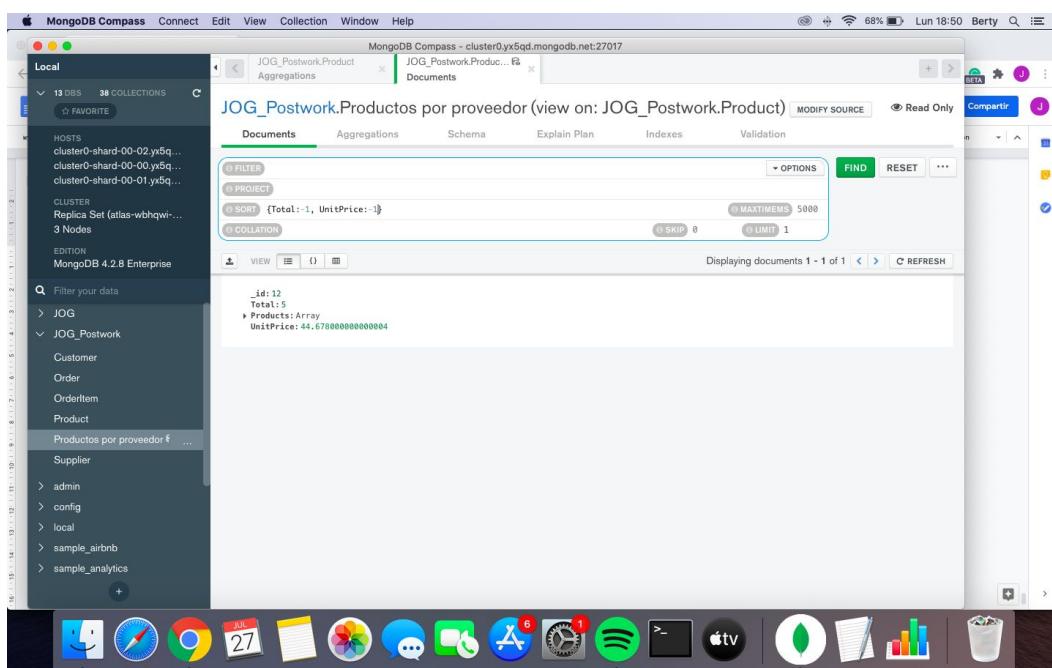
- Document 1: _id: 7, Total: 4, Productos: Array, Precioneto: 34.7125
- Document 2: _id: 8, Total: 4, Productos: Array, Precioneto: 28.175

24.- Crea una vista de esta tabla y arroja al proveedor con más productos y el precio promedio más alto



The screenshot shows the MongoDB Compass application running on a Mac OS X desktop. The main window title is "JOG_Postwork.Product". The left sidebar lists databases "Local" and "JOG_Postwork" containing collections like "Customer", "Order", "OrderItem", "Product", and "Productos por proveedor". The right panel displays a table titled "Productos por proveedor (view on: JOG_Postwork.Product)". It has tabs for "Documents", "Aggregations", "Schema", "Explain Plan", "Indexes", and "Validation". Under "Documents", there are five entries, each showing an _id, Total, and Products array. The first entry has _id: 12, Total: 5, and Products: [{UnitPrice: 44.678}, ...]. The second entry has _id: 7, Total: 5, and Products: [{UnitPrice: 35.5}, ...]. The third entry has _id: 8, Total: 4, and Products: [{UnitPrice: 28.175}, ...]. The fourth entry has _id: 3, Total: 4, and Products: [{UnitPrice: 20.35}, ...]. The fifth entry has _id: 22, Total: 1, and Products: [{UnitPrice: 44.678}, ...]. Below the table are buttons for "FIND", "RESET", and "...". The status bar at the bottom shows the date and time as "Lun 18:49" and the battery level as "68%".

```
{
  sort: {
    Total: 1,
    UnitPrice: -1
  },
  limit: 1
}
```



This screenshot shows the same MongoDB Compass interface after applying the aggregation query. The "Sort" field in the query builder is now set to {Total:-1, UnitPrice:-1}. The result table shows a single document with _id: 12, Total: 5, and Products: [{UnitPrice: 44.678}, ...]. The status bar at the bottom shows the date and time as "Lun 18:50" and the battery level as "68%".

25.- De la vista generada crea un documento por cada producto existente que incluya los datos de cada proveedor por producto. Usa esta vista para crear una vista

```
[$lookup: {
  from: 'Supplier',
  localField: 'SupplierId',
  foreignField: 'Id',
  as: 'SupplierId'
}], {$unwind: {
  path: '$SupplierId'
}]
```

Modifying pipeline backing "JOG_Postwork.Productos con datos del proveedor"

\$lookup

```
1. {  
2.   from: 'Supplier',  
3.   localField: 'SupplierId',  
4.   foreignField: 'Id',  
5.   as: 'SupplierId'  
6. }
```

\$unwind

```
1. {  
2.   path: '$SupplierId'  
3. }
```

_id	ProductName	Supplier
5f1f624b19f51f02bc58816e	Chai	{ "Id": 1, "ProductName": "Chai", "SupplierId": { "Id": 2, "ProductName": "Chang" }, "UnitPrice": 18, "Package": "10 boxes x 20 bags", "IsDiscontinued": false }
5f1f624b19f51f02bc588178	Aniseed Syrup	{ "Id": 2, "ProductName": "Aniseed Syrup", "SupplierId": { "Id": 3, "ProductName": "MandM", "SupplierId": { "Id": 4, "ProductName": "Chang" // This is a self-referencing loop } }, "UnitPrice": 19, "Package": "24 - 12 oz bottles", "IsDiscontinued": false }
5f1f624b19f51f02bc588171	Aniseed Syrup	{ "Id": 3, "ProductName": "Aniseed Syrup", "SupplierId": { "Id": 4, "ProductName": "Chang" // This is a self-referencing loop }, "UnitPrice": 19, "Package": "24 - 12 oz bottles", "IsDiscontinued": false }

26.- Dentro de la tabla “Order” usa la vista creada para mostrar qué productos componen cada orden. Crea una vista llamada “OrderId con producto y proveedor”.

```
[{$lookup: {
  from: 'Productos con datos del proveedor',
  localField: 'ProductId',
  foreignField: 'Id',
  as: 'ProductId'
}}, {$addFields: {
  Total_Amount: {
    $multiply: [
      '$UnitPrice',
      '$Quantity'
    ]
  }
}}, {$unwind: {
  path: '$ProductId'
}}, {$group: {
  _id: '$OrderId',
  ProductId: {
    $push: '$ProductId'
  },
  Total: {
    $sum: '$Total_Amount'
  }
}}]
```

The screenshot shows the MongoDB Compass interface with the following details:

- Left Sidebar:** Shows the database structure with "Local" selected, displaying 13 DBs and 45 collections. A "FAVORITE" section lists "JOG_Postwork" and "JOG_Postwork.Order".
- Top Bar:** Shows the connection information: "MongoDB Compass - cluster0.yx5qd.mongodb.net:27017/JOG_Postwork.Order".
- Central Area:**
 - Aggregations Tab:** Selected tab. It displays the aggregation pipeline being modified:


```
1: { $unwind: { path: '$ProductId' } }
2: { $group: {
3:   _id: '$OrderId',
4:   ProductId: { $push: '$ProductId' },
5:   Total: { $sum: '$Total_Amount' }
6: } }
```
 - Output Area:** Shows the results of the aggregation pipeline. Two sample documents are shown:


```
_id: ObjectId("5f1f61a219f51f02bc587903")
Id: 1
OrderId: 1
> ProductId: Object
  UnitPrice: 14
  Quantity: 12
  Total_Amount: 168
```



```
_id: ObjectId("5f1f61a219f51f02bc587903")
Id: 2
OrderId: 1
> ProductId: Object
  UnitPrice: 9.8
  Quantity: 10
  Total_Amount: 98
```
- Bottom:** macOS Dock with various application icons.

27.- Desde la tabla OrderItem, agrega los documentos de la vista anterior para mostrar por cada cliente las ordenes que ha pedido y que productos incluye. Cada producto debe tener la información del proveedor.

```
[$lookup: {
  from: 'OrderItem',
  localField: '_id',
  foreignField: 'Id',
  as: 'Id'
}], {$unwind: {
  path: '$Id'
}}, {$addFields: {
  CustomerId: '$Id.CustomerId'
}}, {$project: {
  'Id.CustomerId': 0
}}, {$group: {
  _id: '$CustomerId',
  Ordenes: {
    $push: {
      Order: '$Id',
      Producto: '$ProductId',
      Total: '$Total'
    }
  }
}}
}}
```

The screenshot shows the MongoDB Compass interface with the following details:

- Aggregation Pipeline:**

```
1. { $group: {
  _id: '$CustomerId',
  Ordenes: {
    $push: {
      Order: '$Id',
      Producto: '$ProductId',
      Total: '$Total'
    }
  }
}}
```
- Results:**
 - Sample document 1: `_id: 210`, `Ordenes: [{ Order: { _id: "5f1f615c19f51f02bc5875d0" }, Producto: { _id: "5f1f615c19f51f02bc5875d0" }, Total: 1584 }]`
 - Sample document 2: `_id: 291`, `Ordenes: [{ Order: { _id: "5f1f615c19f51f02bc5875d0" }, Producto: { _id: "5f1f615c19f51f02bc5875d0" }, Total: 139.8 }]`
 - Sample document 3: `_id: 291`, `Ordenes: [{ Order: { _id: "5f1f615c19f51f02bc5875d0" }, Producto: { _id: "5f1f615c19f51f02bc5875d0" }, Total: 139.8 }]`

MongoDB Compass Connect Edit View Collection Window Help

Última edición: 15/10/2023 21:13 Lun 21:13 Berty

HOSTS cluster0-shard-00-00.yx5q... cluster0-shard-00-01.yx5q... cluster0-shard-00-02.yx5q...

CLUSTER Replica Set (atlas-wbhqw1... 3 Nodes

EDITION MongoDB 4.2.8 Enterprise

Q Filter your data

> JOG

> JOG_Postwork

Array precios y product...

Base de datos unificada ...

Customer

Order

OrderId con producto y ...

OrderItem

Product

Productos con datos de...

Productos por proveedor €

Supplier

> admin

13 DBs 45 COLLECTIONS

HOSTS cluster0-shard-00-00.yx5q... cluster0-shard-00-01.yx5q... cluster0-shard-00-02.yx5q...

CLUSTER Replica Set (atlas-wbhqw1... 3 Nodes

EDITION MongoDB 4.2.8 Enterprise

Q Filter your data

> JOG

> JOG_Postwork

Array precios y product...

Base de datos unificada ...

Customer

Order

OrderId con producto y ...

OrderItem

Product

Productos con datos de...

Productos por proveedor €

Supplier

> admin

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER

VIEW 0

Displaying documents 1 - 20 of 89

MODIFY SOURCE

Read Only

FIND RESET

... 15.14.31

`_id: 16
 Ordenes: Array
 0: Object
 Order: Object
 Product: Array
 0: Object
 _id: ObjectId("5f1f624b19f51f02bc588172")
 Id: 5
 ProductName: "Chef Antons Gumbo Mix"
 SupplierId: Object
 UnitPrice: 21.35
 Package: "36 boxes"
 IsDiscontinued: true
 1: Object
 _id: ObjectId("5f1f624b19f51f02bc588176")
 Id: 6
 ProductName: "Michi Kobe Niku"
 SupplierId: Object
 UnitPrice: 97
 Package: "18 - 500 g pkgs."
 IsDiscontinued: true
 Total: 931.5
 1: Object
 2: Object`

`_id: 17
 Ordenes: Array`