

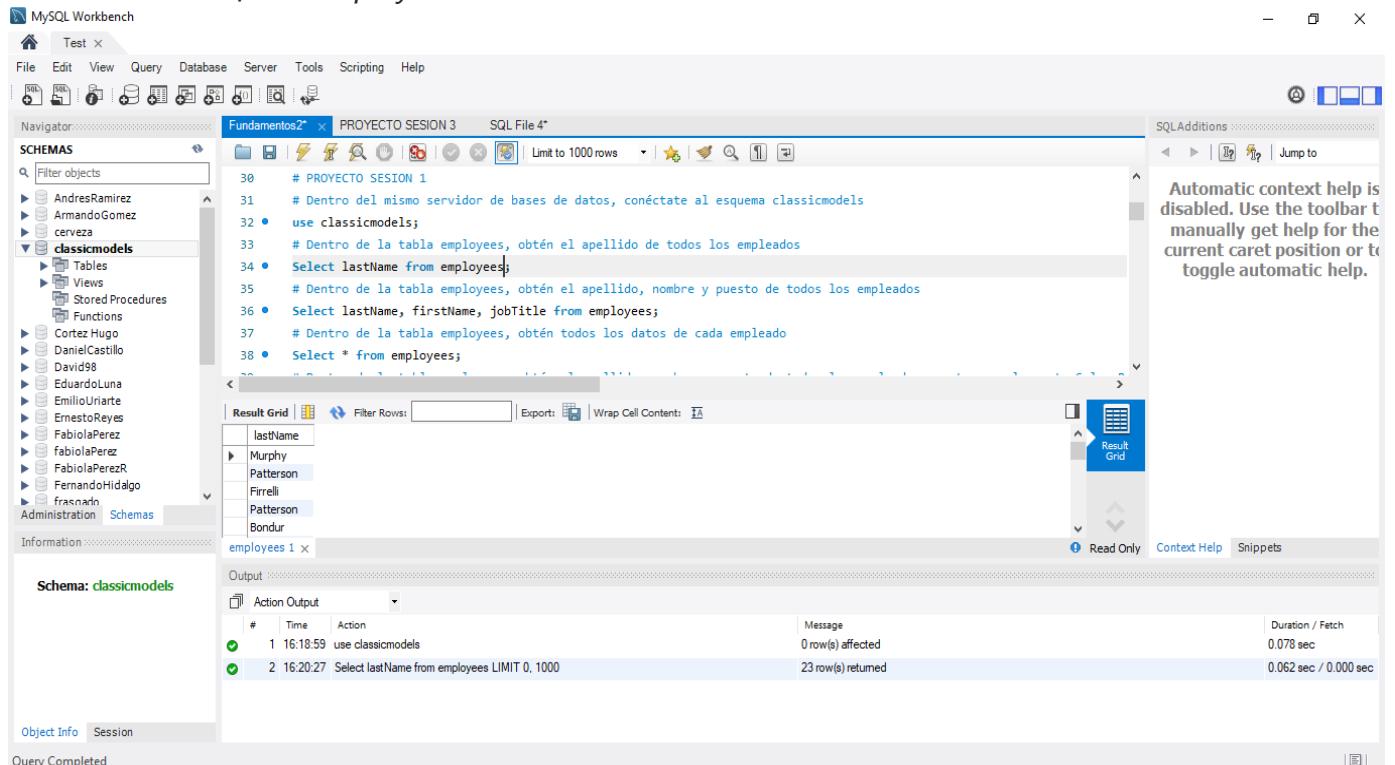
## PROYECTO SESIÓN 1

Todas las consultas que realices deberás mantenerlas dentro del editor de textos de MySQL Workbench. Al finalizar, guarda este archivo, yendo al menú File > Save script. Recuerda que para hacer consultas a una tabla debes conocer primero su estructura.

1. Dentro del mismo servidor de bases de datos, conéctate al esquema `classicmodels`.  
`use classicmodels;`

2. Dentro de la tabla `employees`, obtén el apellido de todos los empleados.

`Select lastName from employees;`



The screenshot shows the MySQL Workbench interface. The SQL Editor tab is active, displaying the following code:

```

30  # PROYECTO SESIÓN 1
31  # Dentro del mismo servidor de bases de datos, conéctate al esquema classicmodels
32  use classicmodels;
33  # Dentro de la tabla employees, obtén el apellido de todos los empleados
34  Select lastName from employees;
35  # Dentro de la tabla employees, obtén el apellido, nombre y puesto de todos los empleados
36  Select lastName, firstName, jobTitle from employees;
37  # Dentro de la tabla employees, obtén todos los datos de cada empleado
38  Select * from employees;
    
```

The Results Grid shows the following data:

| lastName  |
|-----------|
| Murphy    |
| Patterson |
| Firrelli  |
| Patterson |
| Bondur    |

The Output pane shows the following log entries:

| # | Time     | Action                                       | Message            | Duration / Fetch      |
|---|----------|--|--------------------|-----------------------|
| 1 | 16:18:59 | use classicmodels                            | 0 row(s) affected  | 0.078 sec             |
| 2 | 16:20:27 | Select lastName from employees LIMIT 0, 1000 | 23 row(s) returned | 0.062 sec / 0.000 sec |

3. Dentro de la tabla `employees`, obtén el apellido, nombre y puesto de todos los empleados.

`Select lastName, firstName, jobTitle from employees;`

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

classicmodels

employees 2 x

Schema: classicmodels

Action Output

| # | Time     | Action  | Message            | Duration / Fetch      |
|---|----------|---|--------------------|-----------------------|
| 1 | 16:18:59 | use classicmodels   | 0 row(s) affected  | 0.078 sec             |
| 2 | 16:20:27 | Select lastName from employees LIMIT 0, 1000                      | 23 row(s) returned | 0.062 sec / 0.000 sec |
| 3 | 16:21:13 | Select lastName, firstName, jobTitle from employees LIMIT 0, 1000 | 23 row(s) returned | 0.063 sec / 0.000 sec |

Query Completed

Fundamentos2\* PROYECTO SESION 3 SQL File 4\*

```

30 # PROYECTO SESION 1
31 # Dentro del mismo servidor de bases de datos, conéctate al esquema classicmodels
32 • use classicmodels;
33 # Dentro de la tabla employees, obtén el apellido de todos los empleados
34 • Select lastName from employees;
35 # Dentro de la tabla employees, obtén el apellido, nombre y puesto de todos los empleados
36 • Select lastName, firstName, jobTitle from employees;
37 # Dentro de la tabla employees, obtén todos los datos de cada empleado
38 • Select * from employees;

```

Result Grid

| lastName  | firstName | jobTitle             |
|-----------|-----------|----------------------|
| Murphy    | Diane     | President            |
| Patterson | Mary      | VP Sales             |
| Firrelli  | Jeff      | VP Marketing         |
| Patterson | William   | Sales Manager (APAC) |
| Bondur    | Gerard    | Sale Manager (EMEA)  |

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

4. Dentro de la tabla employees, obtén todos los datos de cada empleado.

*Select \* from employees;*

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

classicmodels

employees 3 x

Schema: classicmodels

Action Output

| # | Time     | Action  | Message            | Duration / Fetch      |
|---|----------|---|--------------------|-----------------------|
| 1 | 16:18:59 | use classicmodels   | 0 row(s) affected  | 0.078 sec             |
| 2 | 16:20:27 | Select lastName from employees LIMIT 0, 1000                      | 23 row(s) returned | 0.062 sec / 0.000 sec |
| 3 | 16:21:13 | Select lastName, firstName, jobTitle from employees LIMIT 0, 1000 | 23 row(s) returned | 0.063 sec / 0.000 sec |
| 4 | 16:26:52 | Select * from employees LIMIT 0, 1000                             | 23 row(s) returned | 0.063 sec / 0.000 sec |

Fundamentos2\* PROYECTO SESION 3 SQL File 4\*

```

37 # Dentro de la tabla employees, obtén todos los datos de cada empleado
38 • Select * from employees;
39 # Dentro de la tabla employees, obtén el apellido, nombre y puesto de todos los empleados que tengan el puesto Sales Rep
40 • Select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep";
41 # Dentro de la tabla employees, obtén el apellido, nombre, puesto y código de oficina de todos los empleados que tengan el puesto Sales Rep
42 • Select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;
43 # Dentro de la tabla employees, obtén el apellido, nombre, puesto y código de oficina de todos los empleados que tengan el puesto Sales Rep
44 • Select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" or officeCode = 1;
45 # Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados que tenga código de oficina

```

Result Grid

| employeeNumber | lastName  | firstName | extension | email                           | officeCode | reportsTo | jobTitle             |
|----------------|-----------|-----------|-----------|---------------------------------|------------|-----------|----------------------|
| 1002           | Murphy    | Diane     | x5800     | dmurphy@classicmodelcars.com    | 1          | NULL      | President            |
| 1056           | Patterson | Mary      | x4611     | mpatterso@classicmodelcars.com  | 1          | 1002      | VP Sales             |
| 1076           | Firrelli  | Jeff      | x9273     | jfirrelli@classicmodelcars.com  | 1          | 1002      | VP Marketing         |
| 1088           | Patterson | William   | x4871     | wpatterson@classicmodelcars.com | 6          | 1056      | Sales Manager (APAC) |
| 1102           | Bondur    | Gerard    | x5408     | gbondur@classicmodelcars.com    | 4          | 1056      | Sale Manager (EMEA)  |

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

5. Dentro de la tabla employees, obtén el apellido, nombre y puesto de todos los empleados que tengan el puesto Sales Rep.

*Select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep";*

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Fundamentos2 PROYECTO SESION 3 SQL File 4\*

```

39 # Dentro de la tabla employees, obtén el apellido, nombre y puesto de todos los empleados que tengan el puesto Sales Rep;
40 • Select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep";
41 # Dentro de la tabla employees, obtén el apellido, nombre, puesto y código de oficina de todos los empleados que tengan el puesto Sales Rep;
42 • Select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;
43 # Dentro de la tabla employees, obtén el apellido, nombre, puesto y código de oficina de todos los empleados que tengan el puesto Sales Rep o código de oficina 1;
44 • Select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" or officeCode = 1;
45 # Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados que tenga código de oficina 1;
46 • Select lastName, firstName, jobTitle, officeCode from employees where officeCode in(1,2,3);
47 # Dentro de la tabla employees, obtén el apellido, nombre y puesto de todos los empleados que tengan un puesto distinto de Sales Rep;

```

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

| lastName  | firstName | jobTitle  |
|-----------|-----------|-----------|
| Jennings  | Leslie    | Sales Rep |
| Thompson  | Leslie    | Sales Rep |
| Firrelli  | Julie     | Sales Rep |
| Patterson | Steve     | Sales Rep |
| Tseng     | Foon Yue  | Sales Rep |

employees 4 x

Output:

| # | Time     | Action   | Message            | Duration / Fetch      |
|---|----------|--|--------------------|-----------------------|
| 2 | 16:20:27 | Select lastName from employees LIMIT 0, 1000   | 23 row(s) returned | 0.062 sec / 0.000 sec |
| 3 | 16:21:13 | Select lastName, firstName, jobTitle from employees LIMIT 0, 1000                              | 23 row(s) returned | 0.063 sec / 0.000 sec |
| 4 | 16:26:52 | Select * from employees LIMIT 0, 1000  | 23 row(s) returned | 0.063 sec / 0.000 sec |
| 5 | 16:27:50 | Select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep" LIMIT 0, 1... | 17 row(s) returned | 0.078 sec / 0.000 sec |

Object Info Session

Query Completed

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

- Dentro de la tabla employees, obtén el apellido, nombre, puesto y código de oficina de todos los empleados que tengan el puesto Sales Rep y código de oficina 1.

*Select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;*

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Fundamentos2 PROYECTO SESION 3 SQL File 4\*

```

41 # Dentro de la tabla employees, obtén el apellido, nombre, puesto y código de oficina de todos los empleados que tengan el puesto Sales Rep;
42 • Select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;
43 # Dentro de la tabla employees, obtén el apellido, nombre, puesto y código de oficina de todos los empleados que tengan el puesto Sales Rep o código de oficina 1;
44 • Select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" or officeCode = 1;
45 # Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados que tenga código de oficina 1;
46 • Select lastName, firstName, jobTitle, officeCode from employees where officeCode in(1,2,3);
47 # Dentro de la tabla employees, obtén el apellido, nombre y puesto de todos los empleados que tengan un puesto distinto de Sales Rep;
48 • Select lastName, firstName, jobTitle from employees where not jobTitle = "Sales Rep";
49 # Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados cuyo código de oficina sea 1;

```

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

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

employees 5 x

Output:

| # | Time     | Action  | Message            | Duration / Fetch      |
|---|----------|---|--------------------|-----------------------|
| 3 | 16:21:13 | Select lastName, firstName, jobTitle from employees LIMIT 0, 1000                               | 23 row(s) returned | 0.063 sec / 0.000 sec |
| 4 | 16:26:52 | Select * from employees LIMIT 0, 1000   | 23 row(s) returned | 0.063 sec / 0.000 sec |
| 5 | 16:27:50 | Select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep" LIMIT 0, 1...  | 17 row(s) returned | 0.078 sec / 0.000 sec |
| 6 | 16:28:51 | Select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep..." | 2 row(s) returned  | 0.078 sec / 0.000 sec |

Object Info Session

Query Completed

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

- Dentro de la tabla employees, obtén el apellido, nombre, puesto y código de oficina de todos los empleados que tengan el puesto Sales Rep o código de oficina 1.

*Select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" or officeCode = 1;*

The screenshot shows the MySQL Workbench interface. The main window displays a SQL query in the 'Fundamentos2' tab:

```
43 # Dentro de la tabla employees, obtén el apellido, nombre, puesto y código de oficina de todos los empleados que tengan
44 • Select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" or officeCode = 1;
45 # Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados que tenga código
46 • Select lastName, firstName, jobTitle, officeCode from employees where officeCode in(1,2,3);
47 # Dentro de la tabla employees, obtén el apellido, nombre y puesto de todos los empleados que tengan un puesto distinto
48 • Select lastName, firstName, jobTitle from employees where not jobTitle = "Sales Rep";
49 # Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados cuyo código de ofi
50 • Select lastName, firstName, officeCode from employees where officeCode > 5;
51 # Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados cuyo código de ofi
```

The results grid shows the following data:

| lastName  | firstName | jobTitle           | officeCode |
|-----------|-----------|--------------------|------------|
| Murphy    | Diane     | President          | 1          |
| Patterson | Mary      | VP Sales           | 1          |
| Firrelli  | Jeff      | VP Marketing       | 1          |
| Bow       | Anthony   | Sales Manager (NA) | 1          |
| Jennings  | Leslie    | Sales Rep          | 1          |

The 'Output' pane at the bottom shows the execution log:

| # | Time     | Action  | Message            | Duration / Fetch      |
|---|----------|---|--------------------|-----------------------|
| 4 | 16:26:52 | Select * from employees LIMIT 0, 1000   | 23 row(s) returned | 0.063 sec / 0.000 sec |
| 5 | 16:27:50 | Select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep" LIMIT 0, 1...  | 17 row(s) returned | 0.078 sec / 0.000 sec |
| 6 | 16:28:51 | Select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep..." | 2 row(s) returned  | 0.078 sec / 0.000 sec |
| 7 | 16:29:55 | Select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep..." | 21 row(s) returned | 0.093 sec / 0.000 sec |

A context help message is displayed on the right side of the interface:

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

8. Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados que tenga código de oficina 1, 2 o 3.

Select lastName, firstName, jobTitle, officeCode from employees where officeCode in(1,2,3);

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar displays the Navigator, Schemas (classicmodels), and Information sections. The main area shows a query editor titled "Fundamentos2" under "PROYECTO SESION 3". The query text is in Spanish, listing various SQL statements to retrieve employee and customer data based on specific criteria. Below the query text is a "Result Grid" showing the results of the last query, which retrieves employees with job titles like President, VP Sales, VP Marketing, Sales Manager, and Sales Rep. The results grid has columns for lastName, firstName, jobTitle, and officeCode. At the bottom, there are tabs for Action Output, Read Only, Context Help, and Snippets.

9. Dentro de la tabla employees, obtén el apellido, nombre y puesto de todos los empleados que tengan un puesto distinto a Sales Rep.

Select lastName, firstName, jobTitle from employees where not jobTitle = "Sales Rep";

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Fundamentos2 PROYECTO SESION 3 SQL File 4\*

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

**Result Grid**

| lastName  | firstName | jobTitle             |
|-----------|-----------|----------------------|
| Murphy    | Diane     | President            |
| Patterson | Mary      | VP Sales             |
| Firrelli  | Jeff      | VP Marketing         |
| Patterson | William   | Sales Manager (APAC) |
| Bondur    | Gerard    | Sale Manager (EMEA)  |

employees 8

**Output**

| # | Time     | Action  | Message            | Duration / Fetch      |
|---|----------|---|--------------------|-----------------------|
| 6 | 16:28:51 | Select lastName, firstName, jobTitle, officeCode from employees where job Title = "Sales Rep"   | 2 row(s) returned  | 0.078 sec / 0.000 sec |
| 7 | 16:29:55 | Select lastName, firstName, jobTitle, officeCode from employees where job Title = "Sales Rep"   | 21 row(s) returned | 0.093 sec / 0.000 sec |
| 8 | 16:32:35 | Select lastName, firstName, jobTitle, officeCode from employees where officeCode in(1,2,3) L... | 10 row(s) returned | 0.078 sec / 0.000 sec |
| 9 | 16:34:06 | Select lastName, firstName, jobTitle from employees where not job Title = "Sales Rep" LIMIT ... | 6 row(s) returned  | 0.062 sec / 0.000 sec |

Action Output

Object Info Session

Query Completed

10. Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados cuyo código de oficina sea mayor a 5.

Select lastName, firstName, officeCode from employees where officeCode > 5;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Fundamentos2 PROYECTO SESION 3 SQL File 4\*

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

**Result Grid**

| lastName  | firstName | officeCode |
|-----------|-----------|------------|
| Patterson | William   | 6          |
| Bott      | Larry     | 7          |
| Jones     | Barry     | 7          |
| Fixter    | Andy      | 6          |
| Marsh     | Peter     | 6          |

employees 9

**Output**

| #  | Time     | Action  | Message            | Duration / Fetch      |
|----|----------|---|--------------------|-----------------------|
| 7  | 16:29:55 | Select lastName, firstName, jobTitle, officeCode from employees where job Title = "Sales Rep"   | 21 row(s) returned | 0.093 sec / 0.000 sec |
| 8  | 16:32:35 | Select lastName, firstName, jobTitle, officeCode from employees where officeCode in(1,2,3) L... | 10 row(s) returned | 0.078 sec / 0.000 sec |
| 9  | 16:34:06 | Select lastName, firstName, jobTitle from employees where not job Title = "Sales Rep" LIMIT ... | 6 row(s) returned  | 0.062 sec / 0.000 sec |
| 10 | 16:34:47 | Select lastName, firstName, officeCode from employees where officeCode > 5 LIMIT 0, 1000        | 6 row(s) returned  | 0.062 sec / 0.000 sec |

Action Output

Object Info Session

Query Completed

11. Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados cuyo código de oficina sea menor o igual 4.

Select lastName, firstName, officeCode from employees where officeCode <= 4;

MySQL Workbench

Fundamentos2\* PROYECTO SESION 3 SQL File 4\*

```

51 # Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados cuyo código de oficina sea menor o igual a 4;
52 • Select lastName, firstName, officeCode from employees where officeCode <= 4;
53 # Dentro de la tabla customers, obtén el nombre, país y estado de todos los clientes cuyo país sea USA y cuyo estado sea CA;
54 • select customerName, country, state from customers where country = "USA" and state = "CA";
55 # Dentro de la tabla customers, obtén el nombre, país, estado y límite de crédito de todos los clientes cuyo país sea USA y cuyo estado sea CA y cuyo límite de crédito sea mayor a 100000;
56 • select customerName, country, state, creditLimit from customers where country = "USA" and state = "CA" and creditLimit > 100000;
57 # Dentro de la tabla customers, obtén el nombre y país de todos los clientes cuyo país sea USA o France;
58 • select customerName, country from customers where country in("USA","France");
59 •

```

Result Grid

| lastName  | firstName | officeCode |
|-----------|-----------|------------|
| Murphy    | Diane     | 1          |
| Patterson | Mary      | 1          |
| Firrelli  | Jeff      | 1          |
| Bondur    | Gerard    | 4          |
| Bow       | Anthony   | 1          |

employees 10 x

Action Output

| #  | Time     | Action   | Message            | Duration / Fetch      |
|----|----------|--|--------------------|-----------------------|
| 8  | 16:32:35 | Select lastName, firstName, jobTitle, officeCode from employees where officeCode in(1,2,3) LIMIT 0, 1000 | 10 row(s) returned | 0.078 sec / 0.000 sec |
| 9  | 16:34:06 | Select lastName, firstName, jobTitle from employees where not jobTitle = "Sales Rep" LIMIT 0, 1000       | 6 row(s) returned  | 0.062 sec / 0.000 sec |
| 10 | 16:34:47 | Select lastName, firstName, officeCode from employees where officeCode > 5 LIMIT 0, 1000                 | 6 row(s) returned  | 0.062 sec / 0.000 sec |
| 11 | 16:35:54 | Select lastName, firstName, officeCode from employees where officeCode <= 4 LIMIT 0, 1000                | 15 row(s) returned | 0.063 sec / 0.000 sec |

Output

Object Info Session

Query Completed

12. Dentro de la tabla customers, obtén el nombre, país y estado de todos los clientes cuyo país sea USA y cuyo estado sea CA.

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

MySQL Workbench

Fundamentos2\* PROYECTO SESION 3 SQL File 4\*

```

53 # Dentro de la tabla customers, obtén el nombre, país y estado de todos los clientes cuyo país sea USA y cuyo estado sea CA;
54 • Select customerName, country, state from customers where country = "USA" and state = "CA";
55 # Dentro de la tabla customers, obtén el nombre, país, estado y límite de crédito de todos los clientes cuyo país sea USA y cuyo estado sea CA y cuyo límite de crédito sea mayor a 100000;
56 • select customerName, country, state, creditLimit from customers where country = "USA" and state = "CA" and creditLimit > 100000;
57 # Dentro de la tabla customers, obtén el nombre y país de todos los clientes cuyo país sea USA o France;
58 • select customerName, country from customers where country in("USA","France");
59 •
60 # Dentro de la tabla customers, obtén el nombre, país y límite de crédito de todos los clientes cuyo país sea USA o France y cuyo límite de crédito sea mayor a 100000;
61 • select customerName, country, creditLimit from customers where country in("USA","France") and creditLimit > 100000;

```

Result Grid

| customerName                 | country | state |
|------------------------------|---------|-------|
| Mini Gifts Distributors Ltd. | USA     | CA    |
| Mini Wheels Co.              | USA     | CA    |
| Technics Stores Inc.         | USA     | CA    |
| Toys-GrownUps.com            | USA     | CA    |
| Boards & Toys Co.            | USA     | CA    |

customers 11 x

Action Output

| #  | Time     | Action   | Message            | Duration / Fetch      |
|----|----------|--|--------------------|-----------------------|
| 9  | 16:34:06 | Select lastName, firstName, jobTitle from employees where not jobTitle = "Sales Rep" LIMIT 0, 1000 | 6 row(s) returned  | 0.062 sec / 0.000 sec |
| 10 | 16:34:47 | Select lastName, firstName, officeCode from employees where officeCode > 5 LIMIT 0, 1000           | 6 row(s) returned  | 0.062 sec / 0.000 sec |
| 11 | 16:35:54 | Select lastName, firstName, officeCode from employees where officeCode <= 4 LIMIT 0, 1000          | 15 row(s) returned | 0.063 sec / 0.000 sec |
| 12 | 16:36:52 | Select customerName, country, state from customers where country = "USA" and state = "CA"          | 11 row(s) returned | 0.078 sec / 0.000 sec |

Output

Object Info Session

Query Completed

13. Dentro de la tabla customers, obtén el nombre, país, estado y límite de crédito de todos los clientes cuyo país sea USA, cuyo estado sea CA y cuyo límite de crédito sea mayor a 100000.

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

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

- Navigator:** Shows the **classicmodels** schema with tables like `customers`, `employees`, `offices`, etc.
- SQL Editor:** Contains the following SQL code:
 

```

55 # Dentro de la tabla customers, obtén el nombre, país, estado y límite de crédito de todos los clientes cuyo país sea, ^ 
56 • select customerName, country, state, creditLimit from customers where country = "USA" and state = "CA" 
57 and creditLimit > 100000; 
58 # Dentro de la tabla customers, obtén el nombre y país de todos los clientes cuyo país sea USA o France. 
59 • select customerName, country from customers where country in("USA","France"); 
60 # Dentro de la tabla customers, obtén el nombre, pas y límite de crédito de todos los clientes cuyo país sea USA o France 
61 • select customerName, country, creditLimit from customers where country in("USA","France") and creditLimit > 100000; 
62 # Dentro de la tabla offices, obtén el código de la oficina, ciudad, teléfono y país de aquellas oficinas que se encuentran en USA o France 
63 • select officeCode, city, phone, country from offices where country in("USA","France");
      
```
- Result Grid:** Displays the results of the first query (customers from USA or France):
 

| customerName                 | country | state | creditlimit |
|------------------------------|---------|-------|-------------|
| Mini Gifts Distributors Ltd. | USA     | CA    | 210500.00   |
| Collectable Mini Designs Co. | USA     | CA    | 105000.00   |
| Corporate Gift Ideas Co.     | USA     | CA    | 105000.00   |
- Output:** Shows the execution log with the following entries:
 

| #  | Time     | Action  | Message            | Duration / Fetch      |
|----|----------|---|--------------------|-----------------------|
| 10 | 16:34:47 | Select lastName, firstName, officeCode from employees where officeCode > 5 LIMIT 0, 1000      | 6 row(s) returned  | 0.062 sec / 0.000 sec |
| 11 | 16:35:54 | Select lastName, firstName, officeCode from employees where officeCode <= 4 LIMIT 0, 1000     | 15 row(s) returned | 0.063 sec / 0.000 sec |
| 12 | 16:36:52 | Select customerName, country, state from customers where country = "USA" and state = "CA"     | 11 row(s) returned | 0.078 sec / 0.000 sec |
| 13 | 16:38:52 | select customerName, country, state, creditLimit from customers where country = "USA" and ... | 3 row(s) returned  | 0.078 sec / 0.000 sec |

14. Dentro de la tabla `customers`, obtén el nombre y país de todos los clientes cuyo país sea USA O France.

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

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

- Navigator:** Shows the **classicmodels** schema with tables like `customers`, `employees`, `offices`, etc.
- SQL Editor:** Contains the following SQL code:
 

```

58 # Dentro de la tabla customers, obtén el nombre y país de todos los clientes cuyo país sea USA o France. 
59 • Select customerName, country from customers where country in("USA","France"); 
60 # Dentro de la tabla customers, obtén el nombre, pas y límite de crédito de todos los clientes cuyo país sea USA o France 
61 • Select customerName, country, creditLimit from customers where country in("USA","France") and creditLimit > 100000; 
62 # Dentro de la tabla offices, obtén el código de la oficina, ciudad, teléfono y país de aquellas oficinas que se encuentran en USA o France 
63 • Select officeCode, city, phone, country from offices where country in("USA","France"); 
64 # Dentro de la tabla offices, obtén el código de la oficina, ciudad, teléfono y país de aquellas oficinas que no se encuentran en USA o France 
65 • Select officeCode, city, phone, country from offices where not country in("USA","France"); 
66 # Dentro de la tabla orders, obtén el número de orden, número de cliente, estado y fecha de envío de todas las órdenes 
67 • Select orderID, status, shipVia, shipDate from orders;
      
```
- Result Grid:** Displays the results of the first query (customers from USA or France):
 

| customerName                 | country |
|------------------------------|---------|
| Atelier graphique            | France  |
| Signal Gift Stores           | USA     |
| La Rochelle Gifts            | France  |
| Mini Gifts Distributors Ltd. | USA     |
| Mini Wheels Co.              | USA     |
- Output:** Shows the execution log with the following entries:
 

| #  | Time     | Action  | Message            | Duration / Fetch      |
|----|----------|---|--------------------|-----------------------|
| 11 | 16:35:54 | Select lastName, firstName, officeCode from employees where officeCode <= 4 LIMIT 0, 1000     | 15 row(s) returned | 0.063 sec / 0.000 sec |
| 12 | 16:36:52 | Select customerName, country, state from customers where country = "USA" and state = "CA"     | 11 row(s) returned | 0.078 sec / 0.000 sec |
| 13 | 16:38:52 | select customerName, country, state, creditLimit from customers where country = "USA" and ... | 3 row(s) returned  | 0.078 sec / 0.000 sec |
| 14 | 16:40:02 | Select customerName, country from customers where country in("USA","France") LIMIT 0, 1...    | 48 row(s) returned | 0.078 sec / 0.000 sec |

15. Dentro de la tabla `customers`, obtén el nombre, país y límite de crédito de todos los clientes cuyo país sea USA O France y cuyo límite de crédito sea mayor a 100000. Para este ejercicio ten cuidado con los paréntesis.

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

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Fundamentos2\* PROYECTO SESION 3 SQL File 4\*

Navigator Schemas

SCHEMAS

- AndresRamirez
- ArmandoGomez
- cerveza
- classicmodels**
  - Tables
  - Views
  - Stored Procedures
  - Functions
- Cortez Hugo
- DanielCastillo
- David98
- EduardoLuna
- EmilioUriarte
- ErnestoReyes
- FabiolaPerez
- FabiolaPerezR
- FernandoHidalgo
- franndo
- Administration
- Schemas

Information

Schema: classicmodels

customers 14

| customerName                 | country | creditLimit |
|------------------------------|---------|-------------|
| La Rochelle Gifts            | France  | 118200.00   |
| Mini Gifts Distributors Ltd. | USA     | 210500.00   |
| Land of Toys Inc.            | USA     | 114900.00   |
| Saviley & Herriot, Co.       | France  | 123900.00   |
| Muscle Machine Inc           | USA     | 138500.00   |

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

Output

Action Output

| #  | Time     | Action  | Message            | Duration / Fetch      |
|----|----------|---|--------------------|-----------------------|
| 12 | 16:36:52 | Select customerName, country, state from customers where country = "USA" and state = "C..."   | 11 row(s) returned | 0.078 sec / 0.000 sec |
| 13 | 16:38:52 | select customerName, country, state, creditLimit from customers where country = "USA" and ... | 3 row(s) returned  | 0.078 sec / 0.000 sec |
| 14 | 16:40:02 | Selected customerName, country from customers where country in("USA","France") LIMIT 0,1...   | 48 row(s) returned | 0.078 sec / 0.000 sec |
| 15 | 16:41:21 | Select customerName, country, creditLimit from customers where country in("USA","France")...  | 11 row(s) returned | 0.078 sec / 0.000 sec |

Object Info Session

Query Completed

16. Dentro de la tabla offices, obtén el código de la oficina, ciudad, teléfono y país de aquellas oficinas que se encuentren en USA O France.

Select officeCode, city, phone, country from offices where country in("USA","France");

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Fundamentos2\* PROYECTO SESION 3 SQL File 4\*

Navigator Schemas

SCHEMAS

- AndresRamirez
- ArmandoGomez
- cerveza
- classicmodels**
  - Tables
  - Views
  - Stored Procedures
  - Functions
- Cortez Hugo
- DanielCastillo
- David98
- EduardoLuna
- EmilioUriarte
- ErnestoReyes
- FabiolaPerez
- FabiolaPerezR
- FernandoHidalgo
- franndo
- Administration
- Schemas

Information

Schema: classicmodels

offices 15

| officeCode | city          | phone           | country |
|------------|---------------|-----------------|---------|
| 1          | San Francisco | +1 650 219 4782 | USA     |
| 2          | Boston        | +1 215 837 0825 | USA     |
| 3          | NYC           | +1 212 555 3000 | USA     |
| 4          | Paris         | +33 14 723 4404 | France  |
| NULL       | NULL          | NULL            | NULL    |

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

Output

Action Output

| #  | Time     | Action  | Message            | Duration / Fetch      |
|----|----------|---|--------------------|-----------------------|
| 13 | 16:38:52 | select customerName, country, state, creditLimit from customers where country = "USA" and ...   | 3 row(s) returned  | 0.078 sec / 0.000 sec |
| 14 | 16:40:02 | Selected customerName, country from customers where country in("USA","France") LIMIT 0,1...     | 48 row(s) returned | 0.078 sec / 0.000 sec |
| 15 | 16:41:21 | Select customerName, country, creditLimit from customers where country in("USA","France")...    | 11 row(s) returned | 0.078 sec / 0.000 sec |
| 16 | 16:43:03 | Select officeCode, city, phone, country from offices where country in("USA","France") LIMIT ... | 4 row(s) returned  | 0.125 sec / 0.000 sec |

Object Info Session

Query Completed

17. Dentro de la tabla offices, obtén el código de la oficina, ciudad, teléfono y país de aquellas oficinas que no se encuentren en USA O France.

Select officeCode, city, phone, country from offices where not country in("USA","France");

MySQL Workbench

Fundamentos2\* PROYECTO SESION 3 SQL File 4\*

```

64 # Dentro de la tabla offices, obtén el código de la oficina, ciudad, teléfono y país de aquellas oficinas que no se en...
65 • Select officeCode, city, phone, country from offices where not country in("USA","France");
66 # Dentro de la tabla orders, obtén el número de orden, número de cliente, estado y fecha de envío de todas las órdenes
67 • select orderNumber, customerNumber, shippedDate from orders where orderNumber in(10165,10287,10310);
68 # Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma...
69 • select contactLastName, contactFirstName from customers order by contactLastName ASC;
70 # Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma...
71 • select contactLastName, contactFirstName from customers order by contactLastName DESC;
72 # Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma...

```

Result Grid

| 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        |
| NULL       | HULL   | NULL             | NULL      |

offices 16 x

Output

Action Output

| #  | Time     | Action   | Message            | Duration / Fetch      |
|----|----------|--|--------------------|-----------------------|
| 14 | 16:40:02 | Select customerName, country from customers where country in("USA","France") LIMIT 0,...       | 48 row(s) returned | 0.078 sec / 0.000 sec |
| 15 | 16:41:21 | Select customerName, country, creditLimit from customers where country in("USA","France")...   | 11 row(s) returned | 0.078 sec / 0.000 sec |
| 16 | 16:43:03 | Select officeCode, city, phone, country from offices where country in("USA","France") LIMIT... | 4 row(s) returned  | 0.125 sec / 0.000 sec |
| 17 | 16:44:02 | Select orderNumber, customerNumber, shippedDate from orders where orderNumber in(10165,...     | 3 row(s) returned  | 0.407 sec / 0.000 sec |

Context Help Snippets

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

18. Dentro de la tabla orders, obtén el número de orden, número de cliente, estado y fecha de envío de todas las órdenes con el número 10165, 10287 o 10310.

*Select orderNumber, customerNumber, shippedDate from orders where orderNumber in(10165,10287,10310);*

MySQL Workbench

Fundamentos2\* PROYECTO SESION 3 SQL File 4\*

```

66 # Dentro de la tabla orders, obtén el número de orden, número de cliente, estado y fecha de envío de todas las órdenes
67 • Select orderNumber, customerNumber, shippedDate from orders where orderNumber in(10165,10287,10310);
68 # Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma...
69 • Select contactLastName, contactFirstName from customers order by contactLastName ASC;
70 # Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma...
71 • Select contactLastName, contactFirstName from customers order by contactLastName DESC;
72 # Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma...
73 • Select contactLastName, contactFirstName from customers order by contactLastName DESC, contactFirstName ASC;
74 # Dentro de la tabla customers, obtén el número de cliente, nombre de cliente y el límite de crédito de los cinco clientes...

```

orders 17 x

Output

Action Output

| orderNumber | customerNumber | shippedDate |
|-------------|----------------|-------------|
| 10165       | 148            | 2003-12-26  |
| 10287       | 298            | 2004-09-01  |
| 10310       | 259            | 2004-10-18  |
| NULL        | HULL           | NULL        |

Context Help Snippets

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

19. Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma ascendente.

*Select contactLastName, contactFirstName from customers order by contactLastName ASC;*

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\* PROYECTO SESION 3 SQL File 4\*

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or to toggle automatic help.

SCHEMAS: classicmodels

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

| contactLastName | contactFirstName |
|-----------------|------------------|
| Accorti         | Paolo            |
| Altaiger, G M   | Raanan           |
| Andersen        | Mel              |
| Anton           | Carmen           |
| Ashworth        | Rachel           |

customers 18 x

Output:

Action Output

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 16 | 16:43:03 | Select officeCode, city, phone, country from offices where country in("USA","France") LIMIT...  | 4 row(s) returned   | 0.125 sec / 0.000 sec |
| 17 | 16:44:02 | Select officeCode, city, phone, country from offices where not country in("USA","France") LI... | 3 row(s) returned   | 0.407 sec / 0.000 sec |
| 18 | 16:45:16 | Select orderNumber, customerNumber, shippedDate from orders where orderNumber in(1016...        | 3 row(s) returned   | 0.078 sec / 0.000 sec |
| 19 | 16:45:58 | Select contactLastName, contactFirstName from customers order by contactLastName ASC ...        | 122 row(s) returned | 0.078 sec / 0.000 sec |

Object Info Session

Query Completed

20. Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma descendente.

Select contactLastName, contactFirstName from customers order by contactLastName DESC;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\* PROYECTO SESION 3 SQL File 4\*

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or to toggle automatic help.

SCHEMAS: classicmodels

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

| contactLastName | contactFirstName |
|-----------------|------------------|
| Young           | Jeff             |
| Young           | Julie            |
| Young           | Mary             |
| Young           | Dorothy          |
| Yoshido         | Juri             |

customers 19 x

Output:

Action Output

| #  | Time     | Action   | Message             | Duration / Fetch      |
|----|----------|--|---------------------|-----------------------|
| 17 | 16:44:02 | Select officeCode, city, phone, country from offices where not country in("USA","France") LI...              | 3 row(s) returned   | 0.407 sec / 0.000 sec |
| 18 | 16:45:16 | Select orderNumber, customerNumber, shippedDate from orders where orderNumber in(1016...                     | 3 row(s) returned   | 0.078 sec / 0.000 sec |
| 19 | 16:45:58 | Select contactLastName, contactFirstName from customers order by contactLastName ASC ...                     | 122 row(s) returned | 0.078 sec / 0.000 sec |
| 20 | 16:47:19 | Select contactLastName, contactFirstName from customers order by contactLastName DESC... 122 row(s) returned |                     | 0.094 sec / 0.000 sec |

Object Info Session

Query Completed

21. Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma descendente y luego por nombre de forma ascendente.

Select contactLastName, contactFirstName from customers order by contactLastName DESC, contactFirstName ASC;

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

- Navigator:** Shows the **classicmodels** schema with tables like `customers`, `employees`, etc.
- SQL Editor:** Contains the following SQL code:
 

```

72 # Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma ascendente
73 • Select contactLastName, contactFirstName from customers order by contactLastName ASC, contactFirstName ASC;
74 # Dentro de la tabla customers, obtén el número de cliente, nombre de cliente y el límite de crédito de los cinco clientes con el límite de crédito más alto (top 5).
75 • select customerNumber, contactFirstName, creditLimit from customers order by creditLimit DESC limit 5;
76 # Dentro de la tabla customers, obtén el número de cliente, nombre de cliente y el límite de crédito de los cinco clientes con el límite de crédito más bajo (bottom 5).
77 • select customerNumber, contactFirstName, creditLimit from customers order by creditLimit ASC limit 5;
78
79
80
      
```
- Result Grid:** Displays the results for the first query:
 

| contactLastName | contactFirstName |
|-----------------|------------------|
| Young           | Dorothy          |
| Young           | Jeff             |
| Young           | Julie            |
| Young           | Mary             |
| Yoshido         | Juri             |
| Walker          | Brydey           |
- Output:** Shows the execution log with the following entries:
 

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 18 | 16:45:16 | Select orderNumber, customerNumber, shippedDate from orders where orderNumber in(1016...  | 3 row(s) returned   | 0.078 sec / 0.000 sec |
| 19 | 16:45:58 | Select contactLastName, contactFirstName from customers order by contactLastName ASC ...  | 122 row(s) returned | 0.078 sec / 0.000 sec |
| 20 | 16:47:19 | Select contactLastName, contactFirstName from customers order by contactLastName DESC ... | 122 row(s) returned | 0.094 sec / 0.000 sec |
| 21 | 16:48:38 | Select contactLastName, contactFirstName from customers order by contactLastName DESC ... | 122 row(s) returned | 0.078 sec / 0.000 sec |

22. Dentro de la tabla `customers`, obtén el número de cliente, nombre de cliente y el límite de crédito de los cinco clientes con el límite de crédito más alto (top 5).

*Select customerNumber, contactFirstName, creditLimit from customers order by creditLimit DESC limit 5;*

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

- Navigator:** Shows the **classicmodels** schema with tables like `customers`, `employees`, etc.
- SQL Editor:** Contains the following SQL code:
 

```

74 # Dentro de la tabla customers, obtén el número de cliente, nombre de cliente y el límite de crédito de los cinco clientes con el límite de crédito más alto (top 5).
75 • Select customerNumber, contactFirstName, creditLimit from customers order by creditLimit DESC limit 5;
76 # Dentro de la tabla customers, obtén el número de cliente, nombre de cliente y el límite de crédito de los cinco clientes con el límite de crédito más bajo (bottom 5).
77 • Select customerNumber, contactFirstName, creditLimit from customers order by creditLimit ASC limit 5;
78
79
80
      
```
- Result Grid:** Displays the results for the second query:
 

| customerNumber | contactFirstName | creditLimit |
|----------------|------------------|-------------|
| 141            | Diego            | 227600.00   |
| 124            | Susan            | 210500.00   |
| 298            | Mihael           | 141300.00   |
| 151            | Jeff             | 138500.00   |
| 187            | Rachel           | 136800.00   |
| *              | NULL             | NULL        |
- Output:** Shows the execution log with the following entries:
 

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 19 | 16:45:58 | Select contactLastName, contactFirstName from customers order by contactLastName ASC ...                        | 122 row(s) returned | 0.078 sec / 0.000 sec |
| 20 | 16:47:19 | Select contactLastName, contactFirstName from customers order by contactLastName DESC ...                       | 122 row(s) returned | 0.094 sec / 0.000 sec |
| 21 | 16:48:38 | Select contactLastName, contactFirstName from customers order by contactLastName DESC ...                       | 122 row(s) returned | 0.078 sec / 0.000 sec |
| 22 | 16:49:48 | Select customerNumber, contactFirstName, creditLimit from customers order by creditLimit D... 5 row(s) returned |                     | 0.079 sec / 0.000 sec |

23. Dentro de la tabla `customers`, obtén el número de cliente, nombre de cliente y el límite de crédito de los cinco clientes con el límite de crédito más bajo.

*Select customerNumber, contactFirstName, creditLimit from customers order by creditLimit ASC limit 5;*

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the 'File' menu is open. The 'Navigator' panel on the left shows the 'SCHEMAS' section with 'classicmodels' selected. The main area displays a query editor titled 'PROYECTO SESION 3' containing the following SQL code:

```

76 # Dentro de la tabla customers, obtén el número de cliente, nombre de cliente y el límite de crédito de los cinco clientes con el menor límite de crédito.
77 • Select customerNumber, contactFirstName, creditLimit from customers order by creditLimit ASC limit 5;
78
79
80
81
82

```

The 'Result Grid' shows the following data:

| customerNumber | contactFirstName | creditLimit |
|----------------|------------------|-------------|
| 223            | Horst            | 0.00        |
| 168            | Keith            | 0.00        |
| 169            | Isabel           | 0.00        |
| 206            | Brydey           | 0.00        |
| 125            | Zbyszek          | 0.00        |
| NULL           | NULL             | NULL        |

The 'Output' pane at the bottom shows the execution log:

| #  | Time     | Action   | Message             | Duration / Fetch      |
|----|----------|--|---------------------|-----------------------|
| 20 | 16:47:19 | Select contactLastName, contactFirstName from customers order by contactLastName DESC;         | 122 row(s) returned | 0.094 sec / 0.000 sec |
| 21 | 16:48:38 | Select contactLastName, contactFirstName from customers order by contactLastName DESC;         | 122 row(s) returned | 0.078 sec / 0.000 sec |
| 22 | 16:49:48 | Select customerNumber, contactFirstName, creditLimit from customers order by creditLimit DESC; | 5 row(s) returned   | 0.079 sec / 0.000 sec |
| 23 | 16:50:49 | Select customerNumber, contactFirstName, creditLimit from customers order by creditLimit ASC;  | 5 row(s) returned   | 0.078 sec / 0.000 sec |
| 24 | 17:01:59 | Select employeeNumber, lastName, firstName from employees where firstName like "A%" Li...      | 2 row(s) returned   | 0.063 sec / 0.000 sec |

## PROYECTO SESIÓN 2

Todas las consultas que realices deberás mantenerlas dentro del editor de textos de MySQL Workbench. Al finalizar, guarda este archivo, yendo al menú File > Save script.

- Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyo nombre empiece con a.

*Select employeeNumber, lastName, firstName from employees where firstName like "A%";*

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the 'File' menu is open. The 'Navigator' panel on the left shows the 'SCHEMAS' section with 'classicmodels' selected. The main area displays a query editor titled 'PROYECTO SESION 2' containing the following SQL code:

```

1 # PROYECTO SESIÓN 2
2 # Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyo nombre empiece con A.
3 • Select employeeNumber, lastName, firstName from employees where firstName like "A%";
4 # Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyo nombre termine en on.
5 • Select employeeNumber, lastName, firstName from employees where firstName like "%on";
6 # Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyo nombre incluya on.
7 • Select employeeNumber, lastName, firstName from employees where firstName like "%on%";
8 # Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyos nombres tengan un m.
9 • Select employeeNumber, lastName, firstName from employees where firstName like "%_m";

```

The 'Result Grid' shows the following data:

| employeeNumber | lastName | firstName |
|----------------|----------|-----------|
| 1143           | Bow      | Anthony   |
| 1611           | Fixter   | Andy      |
| NULL           | NULL     | NULL      |

The 'Output' pane at the bottom shows the execution log:

| #  | Time     | Action   | Message             | Duration / Fetch      |
|----|----------|--|---------------------|-----------------------|
| 21 | 16:48:38 | Select contactLastName, contactFirstName from customers order by contactLastName DESC;         | 122 row(s) returned | 0.078 sec / 0.000 sec |
| 22 | 16:49:48 | Select customerNumber, contactFirstName, creditLimit from customers order by creditLimit DESC; | 5 row(s) returned   | 0.079 sec / 0.000 sec |
| 23 | 16:50:49 | Select customerNumber, contactFirstName, creditLimit from customers order by creditLimit ASC;  | 5 row(s) returned   | 0.078 sec / 0.000 sec |
| 24 | 17:01:59 | Select employeeNumber, lastName, firstName from employees where firstName like "A%" Li...      | 2 row(s) returned   | 0.063 sec / 0.000 sec |

- Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyo nombre termina con on.

*Select employeeNumber, lastName, firstName from employees where firstName like "%on";*

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3, SQL File 4\*, employees

**SCHEMAS**

- AndresRamirez
- ArmandoGomez
- cerveza
- classicmodels**
  - Tables
    - customers
    - employees**
      - Columns
      - Indexes
      - Foreign Keys
      - Triggers
  - offices
  - orderdetails
  - orders
  - payments
  - productlines
  - products
  - Views
  - Stored Procedures

Administration Schemas

Information

Table: employees

Columns:

| employeeNumber | int     | PK |
|----------------|---------|----|
| lastName       | varchar |    |
| firstName      | varchar |    |
| extension      | varchar |    |
| email          | varchar |    |
| officeCode     | varchar |    |

Object Info Session

Query Completed

```

4 # Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyo nombre termine en on
5 • Select employeeNumber, lastName, firstName from employees where firstName like "%on%";
6 # Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyo nombre incluya la cadena on
7 • Select employeeNumber, lastName, firstName from employees where firstName like "%on%";
8 # Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyos nombres tengan tres letras e inicien con T y finalizan con m
9 • Select employeeNumber, lastName, firstName from employees where firstName like "T_m";
10 # Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyo nombre no incluya la cadena B
11 • Select employeeNumber, lastName, firstName from employees where firstName not like "%B%";
12 # Dentro de la tabla products, obtén el código de producto y nombre de los productos cuyo código incluye la cadena _20
13 • Select productCode, productName from products where productCode like "%_20%";
14 # Dentro de la tabla orderdetails, obtén el total de cada orden
15 • Select orderNumber, sum(priceEach) from orderdetails group by orderNumber;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Context Help | Snippets

| employeeNumber | lastName | firstName |
|----------------|----------|-----------|
| HULL           | HULL     | HULL      |

Action Output

| #  | Time     | Action   | Message           | Duration / Fetch      |
|----|----------|--|-------------------|-----------------------|
| 31 | 17:09:49 | Select employeeNumber, lastName, firstName from employees where firstName like "%on%" L... | 0 row(s) returned | 0.078 sec / 0.000 sec |
| 32 | 17:11:43 | Select employeeNumber, lastName, firstName from employees where firstName like "%on%" L... | 0 row(s) returned | 0.062 sec / 0.000 sec |

Context Help | Snippets

3. Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyo nombre incluye la cadena on.

Select employeeNumber, lastName, firstName from employees where firstName like "%on%";

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3, SQL File 4\*, employees

**SCHEMAS**

- AndresRamirez
- ArmandoGomez
- cerveza
- classicmodels**
  - Tables
    - customers
    - employees**
      - Columns
      - Indexes
      - Foreign Keys
      - Triggers
  - offices
  - orderdetails
  - orders
  - payments
  - productlines
  - products
  - Views
  - Stored Procedures

Administration Schemas

Information

Table: employees

Columns:

| employeeNumber | int     | PK |
|----------------|---------|----|
| lastName       | varchar |    |
| firstName      | varchar |    |
| extension      | varchar |    |
| email          | varchar |    |
| officeCode     | varchar |    |

Object Info Session

Query Completed

```

6 # Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyo nombre incluya la cadena on
7 • Select employeeNumber, lastName, firstName from employees where firstName like "%on%";
8 # Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyos nombres tengan tres letras e inicien con T y finalizan con m
9 • Select employeeNumber, lastName, firstName from employees where firstName like "T_m";
10 # Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyo nombre no incluya la cadena B
11 • Select employeeNumber, lastName, firstName from employees where firstName not like "%B%";
12 # Dentro de la tabla products, obtén el código de producto y nombre de los productos cuyo código incluye la cadena _20
13 • Select productCode, productName from products where productCode like "%_20%";
14 # Dentro de la tabla orderdetails, obtén el total de cada orden
15 • Select orderNumber, sum(priceEach) from orderdetails group by orderNumber;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Context Help | Snippets

| employeeNumber | lastName | firstName |
|----------------|----------|-----------|
| 1143           | Bow      | Anthony   |
| 1286           | Tseng    | Foon Yue  |
| HULL           | HULL     | HULL      |

Action Output

| #  | Time     | Action   | Message           | Duration / Fetch      |
|----|----------|--|-------------------|-----------------------|
| 32 | 17:11:43 | Select employeeNumber, lastName, firstName from employees where firstName like "%on%" L... | 0 row(s) returned | 0.062 sec / 0.000 sec |
| 33 | 17:12:46 | Select employeeNumber, lastName, firstName from employees where firstName like "%on%" ...  | 2 row(s) returned | 0.062 sec / 0.000 sec |

Context Help | Snippets

4. Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyos nombres tienen tres letras e inician con T y finalizan con m.

Select employeeNumber, lastName, firstName from employees where firstName like "T\_m";

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3, SQL File 4\*, employees

SCHEMAS: Filter objects: AndresRamirez, ArmandoGomez, cerveza, classicmodels, customers, employees, offices, orderdetails, orders, payments, productlines, products, Views, Stored Procedures, Administration, Schemas

Information: Table: employees

Columns:

| employeeNumber | int     |
|----------------|---------|
| lastName       | varchar |
| firstName      | varchar |
| extension      | varchar |
| email          | varchar |
| officeCode     | varchar |

Object Info Session Query Completed

PROYECTO SESION 2\*

```

8 # Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyos nombres tengan una T en la primera letra.
9 • Select employeeNumber, lastName, firstName from employees where firstName like "T_m";
10 # Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyo nombre no inicia con B.
11 • Select employeeNumber, lastName, firstName from employees where firstName not like "B%";
12 # Dentro de la tabla products, obtén el código de producto y nombre de los productos cuyo código incluye la cadena _20.
13 • Select productCode, productName from products where productCode like "%_20%";
14 # Dentro de la tabla orderdetails, obtén el total de cada orden.
15 • Select orderNumber, sum(priceEach) from orderdetails group by orderNumber;
16 # Dentro de la tabla orders obtén el número de órdenes por año.
17 • Select count(orderNumber), year(orderDate) from orders group by year(orderDate);

```

Result Grid: employeeNumber, lastName, firstName

| employeeNumber | lastName | firstName |
|----------------|----------|-----------|
| 1619           | King     | Tom       |
| NULL           | NULL     | NULL      |

Output: Action Output

- # 33 17:12:46 Select employeeNumber, lastName, firstName from employees where firstName like "%on%" ... 2 row(s) returned
- # 34 17:15:14 Select employeeNumber, lastName, firstName from employees where firstName like "T\_m" Li... 1 row(s) returned

Duration / Fetch: 0.062 sec / 0.000 sec 0.078 sec / 0.000 sec

Context Help Snippets

5. Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyo nombre no inicia con B.

Select employeeNumber, lastName, firstName from employees where firstName not like "B%";

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3, SQL File 4\*, employees

SCHEMAS: Filter objects: AndresRamirez, ArmandoGomez, cerveza, classicmodels, customers, employees, offices, orderdetails, orders, payments, productlines, products, Views, Stored Procedures, Administration, Schemas

Information: Table: employees

Columns:

| employeeNumber | int     |
|----------------|---------|
| lastName       | varchar |
| firstName      | varchar |
| extension      | varchar |
| email          | varchar |
| officeCode     | varchar |

Object Info Session Query Completed

PROYECTO SESION 2\*

```

10 # Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyo nombre no inicia con B.
11 • Select employeeNumber, lastName, firstName from employees where firstName not like "B%";
12 # Dentro de la tabla products, obtén el código de producto y nombre de los productos cuyo código incluye la cadena _20.
13 • Select productCode, productName from products where productCode like "%_20%";
14 # Dentro de la tabla orderdetails, obtén el total de cada orden.
15 • Select orderNumber, sum(priceEach) from orderdetails group by orderNumber;
16 # Dentro de la tabla orders obtén el número de órdenes por año.
17 • Select count(orderNumber), year(orderDate) from orders group by year(orderDate);
18 # Obtén el apellido y nombre de los empleados cuya oficina está ubicada en USA.
19 • Select lastName, firstName from employees where officeCode in (select officeCode from offices where country = "USA");

```

Result Grid: employeeNumber, lastName, firstName

| employeeNumber | lastName  | firstName |
|----------------|-----------|-----------|
| 1002           | Murphy    | Diane     |
| 1056           | Patterson | Mary      |
| 1076           | Firelli   | Jeff      |
| 1088           | Patterson | William   |
| 1102           | Bondur    | Gerard    |
| 1143           | Bow       | Anthony   |
| 1165           | Jennings  | Leslie    |

Output: Action Output

- # 34 17:15:14 Select employeeNumber, lastName, firstName from employees where firstName like "T\_m" Li... 1 row(s) returned
- # 35 17:15:42 Select employeeNumber, lastName, firstName from employees where firstName not like "B%"... 22 row(s) returned

Duration / Fetch: 0.078 sec / 0.000 sec 0.078 sec / 0.000 sec

Context Help Snippets

6. Dentro de la tabla products, obtén el código de producto y nombre de los productos cuyo código incluye la cadena \_20.

Select productCode, productName from products where productCode like "%\_20%";

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3, SQL File 4\*, employees

SCHEMAS: Filter objects: AndresRamirez, ArmandoGomez, cerveza, classicmodels

Tables: customers, employees, products, payments, productlines, orders, orderdetails, offices, Views, Stored Procedures, Administration, Schemas

Information: Table: employees

Columns: employeeNumber, lastName, firstName, extension, email, officeCode

Object Info Session

Result Grid: productCode, productName

| productCode | productName                               |
|-------------|---|
| S10_2016    | 1996 Moto Guzzi 1100i                     |
| S18_3320    | 1917 Maxwell Touring Car                  |
| S24_2000    | 1960 BSA Gold Star DB034                  |
| S24_2011    | 18th century schooner                     |
| S24_2022    | 1938 Cadillac V-16 Presidential Limousine |
| S24_3420    | 1937 Horch 930V Limousine                 |
| S24_4620    | 1961 Chevrolet Impala                     |

Output: Action Output

| #  | Time     | Action  | Message            | Duration / Fetch      |
|----|----------|---|--------------------|-----------------------|
| 35 | 17:15:42 | Select employeeNumber, lastName, firstName from employees where firstName not like "B%" ... | 22 row(s) returned | 0.078 sec / 0.000 sec |
| 36 | 17:16:20 | Select productCode, productName from products where productCode like "%_20%" LIMIT 0,...    | 10 row(s) returned | 0.078 sec / 0.000 sec |

Query Completed

7. Dentro de la tabla orderdetails, obtén el total de cada orden.

Select orderNumber, sum(priceEach) from orderdetails group by orderNumber;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3, SQL File 4\*, employees

SCHEMAS: Filter objects: AndresRamirez, ArmandoGomez, cerveza, classicmodels

Tables: customers, employees, products, payments, productlines, orders, orderdetails, offices, Views, Stored Procedures, Administration, Schemas

Information: Schema: classicmodels

Object Info Session

Result Grid: orderNumber, sum(priceEach)

| orderNumber | sum(priceEach) |
|-------------|----------------|
| 10100       | 301.84         |
| 10101       | 352.00         |
| 10102       | 138.68         |
| 10103       | 1520.37        |
| 10104       | 1251.89        |
| 10105       | 1479.71        |
| 10106       | 1427.28        |

Output: Action Output

| #  | Time     | Action   | Message             | Duration / Fetch      |
|----|----------|--|---------------------|-----------------------|
| 47 | 17:26:46 | Select productCode, productName from products where productCode like "%_20%" LIMIT 0,... | 10 row(s) returned  | 0.078 sec / 0.000 sec |
| 48 | 17:27:31 | Select orderNumber, sum(priceEach) from orderdetails group by orderNumber LIMIT 0, 1000  | 326 row(s) returned | 0.078 sec / 0.000 sec |

Query Completed

8. Dentro de la tabla orders obtén el número de órdenes por año.

Select count(orderNumber), year(orderDate) from orders group by year(orderDate);

MySQL Workbench

Test ×

File Edit View Query Database Server Tools Scripting Help

Navigator PROYECTO SESION 1 PROYECTO SESION 2 × PROYECTO SESION 3 SQL File 4\* employees

**SCHEMAS**

Filter objects

AndresRamirez ArmandoGomez cerveza **classicmodels**

Tables

- customers
- employees
- Columns
- Indexes
- Foreign Keys
- Triggers
- offices
- orderdetails
- orders
- payments
- productlines
- products
- Views
- Stored Procedures

Administration Schemas

Information

Schema: classicmodels

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

```

16 # Dentro de la tabla orders obtén el número de órdenes por año.
17 • Select count(orderNumber), year(orderDate) from orders group by year(orderDate);
18 # Obtén el apellido y nombre de los empleados cuya oficina está ubicada en USA.
19 • Select lastName, firstName from employees where officeCode in (select officeCode from offices where country = "USA");
20 # Obtén el número de cliente, número de cheque y cantidad del cliente que ha realizado el pago más alto.
21 • Select customerNumber, checkNumber, amount from payments order by amount DESC limit 1;
22 # Obtén el número de cliente, número de cheque y cantidad de aquellos clientes cuyo pago es más alto que el promedio.
23 • Select customerNumber, checkNumber, amount from payments where ((select avg(amount) from payments) < amount);
24 # Obtén el nombre de aquellos clientes que no han hecho ninguna orden.
25 • Select customerName, customerNumber from customers where customerNumber not in(select customerNumber from orders);

```

Result 25 × Read Only Context Help Snippets

Action Output

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 48 | 17:27:31 | Select orderNumber, sum(priceEach) from orderdetails group by orderNumber LIMIT 0, 1000       | 326 row(s) returned | 0.078 sec / 0.000 sec |
| 49 | 17:28:09 | Select count(orderNumber), year(orderDate) from orders group by year(orderDate) LIMIT 0, 1... | 3 row(s) returned   | 0.062 sec / 0.000 sec |

Object Info Session

Query Completed

9. Obtén el apellido y nombre de los empleados cuya oficina está ubicada en USA.

*Select lastName, firstName from employees where officeCode in (select officeCode from offices where country = "USA");*

MySQL Workbench

Test ×

File Edit View Query Database Server Tools Scripting Help

Navigator PROYECTO SESION 1 PROYECTO SESION 2 × PROYECTO SESION 3 SQL File 4\* employees

**SCHEMAS**

Filter objects

AndresRamirez ArmandoGomez cerveza **classicmodels**

Tables

- customers
- employees
- Columns
- Indexes
- Foreign Keys
- Triggers
- offices
- orderdetails
- orders
- payments
- productlines
- products
- Views
- Stored Procedures

Administration Schemas

Information

Schema: classicmodels

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

```

18 # Obtén el apellido y nombre de los empleados cuya oficina está ubicada en USA.
19 • Select lastName, firstName from employees where officeCode in (select officeCode from offices where country = "USA");
20 # Obtén el número de cliente, número de cheque y cantidad del cliente que ha realizado el pago más alto.
21 • Select customerNumber, checkNumber, amount from payments order by amount DESC limit 1;
22 # Obtén el número de cliente, número de cheque y cantidad de aquellos clientes cuyo pago es más alto que el promedio.
23 • Select customerNumber, checkNumber, amount from payments where ((select avg(amount) from payments) < amount);
24 # Obtén el nombre de aquellos clientes que no han hecho ninguna orden.
25 • Select customerName, customerNumber from customers where customerNumber not in(select customerNumber from orders);
26 # Obtén el máximo, mínimo y promedio del número de productos en las órdenes de venta.
27 • Select orderNumber, max(quantityOrdered), min(quantityOrdered), avg(quantityOrdered) from orderdetails

```

Result 26 × Read Only Context Help Snippets

Action Output

| #  | Time     | Action  | Message            | Duration / Fetch      |
|----|----------|---|--------------------|-----------------------|
| 49 | 17:28:09 | Select count(orderNumber), year(orderDate) from orders group by year(orderDate) LIMIT 0, 1...   | 3 row(s) returned  | 0.062 sec / 0.000 sec |
| 50 | 17:28:49 | Select lastName, firstName from employees where officeCode in (select officeCode from offic...) | 10 row(s) returned | 0.078 sec / 0.000 sec |

Object Info Session

Query Completed

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

*Select customerNumber, checkNumber, amount from payments order by amount DESC limit 1;*

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3, SQL File 4\*, employees

SCHEMAS: AndresRamirez, ArmandoGomez, cerveza, classicomodels (selected), customers, employees, payments, products, offices, orderdetails, orders, Views, Stored Procedures, Administration, Schemas

Information: Schema: classicomodels, Object Info, Session

Result Grid: customerNumber, checkNumber, amount

```

20 # Obten el número de cliente, número de cheque y cantidad del cliente que ha realizado el pago más alto.
21 • Select customerNumber, checkNumber, amount from payments order by amount DESC limit 1;
22 # Obten el número de cliente, número de cheque y cantidad de aquellos clientes cuyo pago es más alto que el promedio.
23 • Select customerNumber, checkNumber, amount from payments where ((select avg(amount) from payments) < amount);
24 # Obten el nombre de aquellos clientes que no han hecho ninguna orden.
25 • Select customerName, customerNumber from customers where customerNumber not in(select customerNumber from orders);
26 # Obten el máximo, mínimo y promedio del número de productos en las órdenes de venta.
27 • Select orderNumber, max(quantityOrdered), min(quantityOrdered), avg(quantityOrdered) from orderdetails
28 group by orderNumber;
29 # Dentro de la tabla orders, obtén el número de órdenes que hay por cada estado.

```

payments 27 x

Action Output:

| #  | Time     | Action  | Message | Duration / Fetch      |
|----|----------|---|---------|-----------------------|
| 50 | 17:28:49 | Select lastName, firstName from employees where officeCode in (select officeCode from offic... 10 row(s) returned |         | 0.078 sec / 0.000 sec |
| 51 | 17:29:25 | Select customerNumber, checkNumber, amount from payments order by amount DESC limit 1 1 row(s) returned           |         | 0.078 sec / 0.000 sec |

Query Completed

11. Obtén 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 ((select avg(amount) from payments) < amount);*

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3, SQL File 4\*, employees

SCHEMAS: AndresRamirez, ArmandoGomez, cerveza, classicomodels (selected), customers, employees, payments, products, offices, orderdetails, orders, Views, Stored Procedures, Administration, Schemas

Information: Schema: classicomodels, Object Info, Session

Result Grid: customerNumber, checkNumber, amount

```

22 # Obten el número de cliente, número de cheque y cantidad de aquellos clientes cuyo pago es más alto que el promedio.
23 • Select customerNumber, checkNumber, amount from payments where ((select avg(amount) from payments) < amount);
24 # Obten el nombre de aquellos clientes que no han hecho ninguna orden.
25 • Select customerName, customerNumber from customers where customerNumber not in(select customerNumber from orders);
26 # Obten el máximo, mínimo y promedio del número de productos en las órdenes de venta.
27 • Select orderNumber, max(quantityOrdered), min(quantityOrdered), avg(quantityOrdered) from orderdetails
28 group by orderNumber;
29 # Dentro de la tabla orders, obtén el número de órdenes que hay por cada estado.
30 • Select status, count(orderNumber) from orders group by status;

```

payments 28 x

Action Output:

| #  | Time     | Action   | Message | Duration / Fetch      |
|----|----------|--|---------|-----------------------|
| 51 | 17:29:25 | Select customerNumber, checkNumber, amount from payments order by amount DESC limit 1 1 row(s) returned      |         | 0.078 sec / 0.000 sec |
| 52 | 17:29:59 | Select customerNumber, checkNumber, amount from payments where ((select avg(amount) f... 134 row(s) returned |         | 0.079 sec / 0.000 sec |

Query Completed

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

Select customerName, customerNumber from customers where customerNumber not in(select customerNumber from orders);

```

22 # Obten el número de cliente, número de cheque y cantidad de aquellos clientes cuyo pago es más alto que el promedio.
23 • Select customerNumber, checkNumber, amount from payments where ((select avg(amount) from payments) < amount);
24 # Obten el nombre de aquellos clientes que no han hecho ninguna orden.
25 • Select customerName, customerNumber from customers where customerNumber not in(select customerNumber from orders);
26 # Obten el máximo, mínimo y promedio del número de productos en las órdenes de venta.
27 • Select orderNumber, max(quantityOrdered), min(quantityOrdered), avg(quantityOrdered) from orderdetails
     group by orderNumber;
28 # Dentro de la tabla orders, obtén el número de órdenes que hay por cada estado.
29 • Select status, count(orderNumber) from orders group by status;

```

| customerName               | customerNumber |
|----------------------------|----------------|
| Havel & Bzyszek Co         | 125            |
| American Souvenirs Inc     | 168            |
| Porto Imports Co.          | 169            |
| Asian Shopping Network, Co | 206            |
| Natürlich Autos            | 223            |
| ANG Resellers              | 237            |
| Messner Shopping Network   | 247            |

customers 29 x

Action Output

| #  | Time     | Action   | Message             | Duration / Fetch      |
|----|----------|--|---------------------|-----------------------|
| 52 | 17:29:59 | Select customerNumber, checkNumber, amount from payments where ((select avg(amount) f... | 134 row(s) returned | 0.079 sec / 0.000 sec |
| 53 | 17:30:30 | Select customerName, customerNumber from customers where customerNumber not in(selec...  | 24 row(s) returned  | 0.078 sec / 0.000 sec |

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;

```

22 # Obten el número de cliente, número de cheque y cantidad de aquellos clientes cuyo pago es más alto que el promedio.
23 • Select customerNumber, checkNumber, amount from payments where ((select avg(amount) from payments) < amount);
24 # Obten el nombre de aquellos clientes que no han hecho ninguna orden.
25 • Select customerName, customerNumber from customers where customerNumber not in(select customerNumber from orders);
26 # Obten el máximo, mínimo y promedio del número de productos en las órdenes de venta.
27 • Select orderNumber, max(quantityOrdered), min(quantityOrdered), avg(quantityOrdered) from orderdetails
     group by orderNumber;
28 # Dentro de la tabla orders, obtén el número de órdenes que hay por cada estado.
29 • Select status, count(orderNumber) from orders group by status;

```

| orderNumber | max(quantityOrdered) | min(quantityOrdered) | avg(quantityOrdered) |
|-------------|----------------------|----------------------|----------------------|
| 10100       | 50                   | 22                   | 37.7500              |
| 10101       | 46                   | 25                   | 35.5000              |
| 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              |

Result 30 x

Action Output

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 53 | 17:30:30 | Select customerName, customerNumber from customers where customerNumber not in(selec...       | 24 row(s) returned  | 0.078 sec / 0.000 sec |
| 54 | 17:31:31 | Select orderNumber, max(quantityOrdered), min(quantityOrdered), avg(quantityOrdered) from ... | 326 row(s) returned | 0.078 sec / 0.000 sec |

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

*Select status, count(orderNumber) from orders group by status;*

| status     | count(orderNumber) |
|------------|--------------------|
| Shipped    | 303                |
| Resolved   | 4                  |
| Cancelled  | 6                  |
| On Hold    | 4                  |
| Disputed   | 3                  |
| In Process | 6                  |

## PROYECTO SESIÓN 3

Todas las consultas que realices deberás mantenerlas dentro del editor de textos de MySQL Workbench. Al finalizar, guarda este archivo, llenando al menú File > Save script.

*Para estas consultas usa RIGHT JOIN*

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

*use classicmodels;*

*show keys from products;*

*select productCode, productName, productDescription from products;*

*select productCode, productName, textDescription from products as p right join productlines as l on p.productLine=l.productLine;*

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3\*, SQL File 4\*, employees

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

**Schema: classicmodels**

Tables:

- customers
- employees
- columns
- indexes
- foreign keys
- triggers
- offices
- orderdetails
- orders
- payments
- productlines
- products
- Views
- Stored Procedures

Administration Schemas

Information

Object Info Session

Query Completed

```

1 # PROYECTO SESION 3
2 # Para estas consultas usa RIGHT JOIN
3 # Obten el código de producto, nombre de producto y descripción de todos los productos.
4 • use classicmodels;
5 • show keys from products;
6 • select productCode, productName, productDescription from products;
7 • select productCode, productName, textDescription from products as p right join productlines
8 as l on p.productLine=l.productLine;
9 # Obten el número de orden, estado y costo total de cada orden.

```

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

| Table    | Non_unique | Key_name    | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_col |
|----------|------------|-------------|--------------|-------------|-----------|-------------|----------|--------|------|------------|---------|-----------|
| products | 0          | PRIMARY     | 1            | productCode | A         | 110         | NULL     | NULL   | NULL | BTREE      |         |           |
| products | 1          | productLine | 1            | productLine | A         | 7           | NULL     | NULL   | NULL | BTREE      |         |           |

Result 1 x Read Only Context Help Snippets

Output

Action Output

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 54 | 17:31:31 | Select orderNumber, max(quantityOrdered), min(quantityOrdered), avg(quantityOrdered) from ... | 326 row(s) returned | 0.078 sec / 0.000 sec |
| 55 | 17:31:56 | Select status, count(orderNumber) from orders group by status LIMIT 0, 1000                   | 6 row(s) returned   | 0.078 sec / 0.000 sec |
| 56 | 17:35:19 | use classicmodels   | 0 row(s) affected   | 0.079 sec             |
| 57 | 17:35:23 | show keys from products   | 2 row(s) returned   | 0.078 sec / 0.000 sec |

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3\*, SQL File 4\*, employees

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

**Schema: classicmodels**

Tables:

- customers
- employees
- columns
- indexes
- foreign keys
- triggers
- offices
- orderdetails
- orders
- payments
- productlines
- products
- Views
- Stored Procedures

Administration Schemas

Information

Object Info Session

Query Completed

```

1 # PROYECTO SESION 3
2 # Para estas consultas usa RIGHT JOIN
3 # Obten el código de producto, nombre de producto y descripción de todos los productos.
4 • use classicmodels;
5 • show keys from products;
6 • select productCode, productName, productDescription from products;
7 • select productCode, productName, textDescription from products as p right join productlines
8 as l on p.productLine=l.productLine;
9 # Obten el número de orden, estado y costo total de cada orden.

```

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

| productCode | productName              | textDescription                                    |
|-------------|--------------------------|--|
| S10_1949    | 1952 Alpine Renault 1300 | Attention car enthusiasts: Make your wildest ca... |
| S10_4757    | 1972 Alfa Romeo GTA      | Attention car enthusiasts: Make your wildest ca... |
| S10_4962    | 1962 LanciaA Delta 16V   | Attention car enthusiasts: Make your wildest ca... |
| S12_1099    | 1968 Ford Mustang        | Attention car enthusiasts: Make your wildest ca... |
| S12_1108    | 2001 Ferrari Enzo        | Attention car enthusiasts: Make your wildest ca... |

Result 3 x Read Only Context Help Snippets

Output

Action Output

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 56 | 17:35:19 | use classicmodels   | 0 row(s) affected   | 0.079 sec             |
| 57 | 17:35:23 | show keys from products   | 2 row(s) returned   | 0.078 sec / 0.000 sec |
| 58 | 17:36:07 | select productCode, productName, productDescription from products LIMIT 0, 1000                 | 110 row(s) returned | 0.235 sec / 0.000 sec |
| 59 | 17:36:23 | select productCode, productName, textDescription from products as p right join productlines ... | 110 row(s) returned | 1.297 sec / 0.234 sec |

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

```

select o.orderNumber, status, priceEach, quantityOrdered from orders as o
right join orderdetails as d on o.orderNumber=d.orderNumber;

```

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3\*, SQL File 4\*, employees

SCHEMAS: Filter objects: AndresRamirez, ArmandoGomez, cerveza, classicmodels

Tables: customers, employees, Columns, Indexes, Foreign Keys, offices, orderdetails, orders, payments, productlines, products, Views, Stored Procedures, Administration, Schemas

Information: Schema: classicmodels

Action Output:

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 57 | 17:35:23 | show keys from products   | 2 row(s) returned   | 0.078 sec / 0.000 sec |
| 58 | 17:36:07 | select productCode, productName, productDescription from products LIMIT 0, 1000                                       | 110 row(s) returned | 0.235 sec / 0.000 sec |
| 59 | 17:36:23 | select productCode, productName, textDescription from products as p right join productlines ...                       | 110 row(s) returned | 1.297 sec / 0.234 sec |
| 60 | 17:37:29 | select o.orderNumber, status, priceEach, quantityOrdered from orders as o right join orderdet... 1000 row(s) returned |                     | 0.141 sec / 0.078 sec |

Result Grid: Filter Rows: Export: Wrap Cell Content: Fetch rows: Result Grid

Result 4 x

Output:

Query Completed

```

9 # Obtén el número de orden, estado y costo total de cada orden.
10 • select o.orderNumber, status, priceEach, quantityOrdered from orders as o
11 right join orderdetails as d on o.orderNumber=d.orderNumber;
12 # Obtén el número de orden, fecha de orden, línea de orden, nombre del producto, cantidad ordenada y precio de cada pieza.
13 • select o.orderNumber, orderDate, orderLineNumber, productName, quantityOrdered, priceEach from orders as o
14 right join orderdetails as d on o.orderNumber=d.orderNumber right join products as p on d.productCode=p.productCode;
15 # Obtén el número de orden, nombre del producto, el precio sugerido de fábrica (msrp) y precio de cada pieza.
16 • select o.orderNumber, productName, MSRP, priceEach from orderdetails as o right join products as p
17 on o.productCode=p.productCode;
18
19 # Para estas consultas usa ! FFTZ NOTN

```

3. Obtén 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 o.orderNumber, orderDate, orderLineNumber, productName, quantityOrdered, priceEach  
from orders as o*

*right join orderdetails as d on o.orderNumber=d.orderNumber right join products as p on  
d.productCode=p.productCode;*

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3\*, SQL File 4\*, employees

SCHEMAS: Filter objects: AndresRamirez, ArmandoGomez, cerveza, classicmodels

Tables: customers, employees, Columns, Indexes, Foreign Keys, offices, orderdetails, orders, payments, productlines, products, Views, Stored Procedures, Administration, Schemas

Information: Schema: classicmodels

Action Output:

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 58 | 17:36:07 | select productCode, productName, productDescription from products LIMIT 0, 1000                                       | 110 row(s) returned | 0.235 sec / 0.000 sec |
| 59 | 17:36:23 | select productCode, productName, textDescription from products as p right join productlines ...                       | 110 row(s) returned | 1.297 sec / 0.234 sec |
| 60 | 17:37:29 | select o.orderNumber, status, priceEach, quantityOrdered from orders as o right join orderdet... 1000 row(s) returned |                     | 0.141 sec / 0.078 sec |
| 61 | 17:38:08 | select o.orderNumber, orderDate, orderLineNumber, productName, quantityOrdered, priceEa... 1000 row(s) returned       |                     | 0.156 sec / 0.250 sec |

Result Grid: Filter Rows: Export: Wrap Cell Content: Fetch rows: Result Grid

Result 5 x

Output:

Query Completed

```

12 # Obtén el número de orden, fecha de orden, línea de orden, nombre del producto, cantidad ordenada y precio de cada pieza.
13 • select o.orderNumber, orderDate, orderLineNumber, productName, quantityOrdered, priceEach from orders as o
14 right join orderdetails as d on o.orderNumber=d.orderNumber right join products as p on d.productCode=p.productCode;
15 # Obtén el número de orden, nombre del producto, el precio sugerido de fábrica (msrp) y precio de cada pieza.
16 • select o.orderNumber, productName, MSRP, priceEach from orderdetails as o right join products as p
17 on o.productCode=p.productCode;
18
19 # Para estas consultas usa ! FFTZ NOTN

```

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

```
select o.orderNumber, productName, MSRP, priceEach from orderdetails as o right join products
as p
```

```
on o.productCode=p.productCode;
```

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar displays the Navigator with Schemas (AndresRamirez, ArmandoGomez, cerveza, classicmodels), Tables (customers, employees, columns, indexes, foreign keys, triggers, offices, orderdetails, orders, payments, productlines, products, views), and Administration (Shared Procedures). The central area contains four tabs: PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3 (selected), and SQL File 4\*. The PROYECTO SESION 3 tab contains the following SQL code:

```
15 # Obtén el número de orden, nombre del producto, el precio sugerido de fábrica (msrp) y precio de cada pieza.
16 • select o.orderNumber, productName, MSRP, priceEach from orderdetails as o right join products as p
17 on o.productCode=p.productCode;
18
19 # Para estas consultas usa LEFT JOIN
20 # Obtén el número de cliente, nombre de cliente, número de orden y estado de cada cliente.
21 • select o.customerNumber, customerName, orderNumber, status from customers as c
22 left join orders as o on c.customerNumber=o.customerNumber;
```

The Result Grid shows the following data:

| orderNumber | productName                           | MSRP  | priceEach |
|-------------|---------------------------------------|-------|-----------|
| 10107       | 1969 Harley Davidson Ultimate Chopper | 95.70 | 81.35     |
| 10121       | 1969 Harley Davidson Ultimate Chopper | 95.70 | 86.13     |
| 10134       | 1969 Harley Davidson Ultimate Chopper | 95.70 | 90.92     |
| 10145       | 1969 Harley Davidson Ultimate Chopper | 95.70 | 76.56     |
| 10159       | 1969 Harley Davidson Ultimate Chopper | 95.70 | 81.35     |
| 10168       | 1969 Harley Davidson Ultimate Chopper | 95.70 | 94.74     |
| 10180       | 1969 Harley Davidson Ultimate Chopper | 95.70 | 76.56     |
| 10190       | 1969 Harley Davidson Ultimate Chopper | 95.70 | 81.35     |

The Output section shows the following log entries:

| #  | Time     | Action  | Message               | Duration / Fetch      |
|----|----------|---|-----------------------|-----------------------|
| 59 | 17:36:23 | select productCode, productName, textDescription from products as p right join productlines ...                       | 110 row(s) returned   | 1.297 sec / 0.234 sec |
| 60 | 17:37:29 | select o.orderNumber, status, priceEach, quantityOrdered from orders as o right join orderdet... 1000 row(s) returned | 0.141 sec / 0.078 sec |                       |
| 61 | 17:38:08 | select o.orderNumber, orderDate, orderLineNumber, productName, quantityOrdered, priceEa... 1000 row(s) returned       | 0.156 sec / 0.250 sec |                       |
| 62 | 17:39:10 | select o.orderNumber, productName, MSRP, priceEach from orderdetails as o right join produc... 1000 row(s) returned   | 0.157 sec / 0.203 sec |                       |

Para estas consultas usa LEFT JOIN

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

```
select o.customerNumber, customerName, orderNumber, status from customers as c
left join orders as o on c.customerNumber=o.customerNumber;
```

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3\*, SQL File 4\*, employees

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

**Schema: classicmodels**

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

| customerNumber | customerName               | orderNumber | status  |
|----------------|----------------------------|-------------|---------|
| 103            | Atelier graphique          | 10123       | Shipped |
| 103            | Atelier graphique          | 10298       | Shipped |
| 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 |
| ...            |                            |             |         |

Action Output

| #  | Time     | Action   | Message              | Duration / Fetch      |
|----|----------|--|----------------------|-----------------------|
| 60 | 17:37:29 | select o.orderNumber, status, priceEach, quantityOrdered from orders as o right join orderdet... | 1000 row(s) returned | 0.141 sec / 0.078 sec |
| 61 | 17:38:08 | select o.orderNumber, orderDate, orderLineNumber, productName, quantityOrdered, priceEa...       | 1000 row(s) returned | 0.156 sec / 0.250 sec |
| 62 | 17:39:10 | select o.orderNumber, productName, MSRP, priceEach from orderdetails as o right join produ...    | 1000 row(s) returned | 0.157 sec / 0.203 sec |
| 63 | 17:39:52 | select o.customerNumber, customerName, orderNumber, status from customers as c left join ...     | 350 row(s) returned  | 0.156 sec / 0.000 sec |

Object Info Session

Query Completed

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

*select customerName, contactLastName, status from customers as c left join orders as o*

*on c.customerNumber=o.customerNumber where o.customerNumber is Null;*

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3\*, SQL File 4\*, employees

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

**Schema: classicmodels**

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

| customerName               | contactLastName | status |
|----------------------------|-----------------|--------|
| Havel & Bzyszek Co         | Piestrzewicz    | NULL   |
| American Souvenirs Inc     | Franco          | NULL   |
| Porto Imports Co.          | de Castro       | NULL   |
| Asian Shopping Network, Co | Walker          | NULL   |
| Natürlich Autos            | Kloss           | NULL   |
| ANG Resellers              | Camino          | NULL   |
| Messner Shopping Network   | Messner         | NULL   |
| Ernesto Giese Co.          | Friedrich       | NULL   |

Action Output

| #  | Time     | Action  | Message              | Duration / Fetch      |
|----|----------|---|----------------------|-----------------------|
| 61 | 17:38:08 | select o.orderNumber, orderDate, orderLineNumber, productName, quantityOrdered, priceEa...    | 1000 row(s) returned | 0.156 sec / 0.250 sec |
| 62 | 17:39:10 | select o.orderNumber, productName, MSRP, priceEach from orderdetails as o right join produ... | 1000 row(s) returned | 0.157 sec / 0.203 sec |
| 63 | 17:39:52 | select o.customerNumber, customerName, orderNumber, status from customers as c left join ...  | 350 row(s) returned  | 0.156 sec / 0.000 sec |
| 64 | 17:40:26 | select customerName, contactLastName, status from customers as c left join orders as o on ... | 24 row(s) returned   | 0.063 sec / 0.000 sec |

Object Info Session

Query Completed

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

*select lastName, firstName, customerName, checkNumber, amount from employees as e left join customers as c*

*on e.employeeNumber=c.salesRepEmployeeNumber left join payments as p on c.customerNumber=p.customerNumber;*

The screenshot shows the MySQL Workbench interface. In the center, there is a 'Result Grid' displaying a table with columns: lastName, firstName, customerName, checkNumber, and amount. The data consists of seven rows for the employee 'Jennings' at Mini Gifts Distributors Ltd. The 'Output' pane below shows the execution history of four SQL statements, each with a timestamp, action, message, and duration.

| lastName | firstName | customerName                 | checkNumber | amount    |
|----------|-----------|------------------------------|-------------|-----------|
| Jennings | Leslie    | Mini Gifts Distributors Ltd. | AE215433    | 101244.59 |
| Jennings | Leslie    | Mini Gifts Distributors Ltd. | BG255406    | 85410.87  |
| Jennings | Leslie    | Mini Gifts Distributors Ltd. | CQ287967    | 11044.30  |
| Jennings | Leslie    | Mini Gifts Distributors Ltd. | ET64396     | 83598.04  |
| Jennings | Leslie    | Mini Gifts Distributors Ltd. | HJ366474    | 47142.70  |
| Jennings | Leslie    | Mini Gifts Distributors Ltd. | HR86578     | 55639.66  |
| Jennings | Leslie    | Mini Gifts Distributors Ltd. | KI131716    | 111654.40 |

Output

| #  | Time     | Action  | Message              | Duration / Fetch      |
|----|----------|---|----------------------|-----------------------|
| 62 | 17:39:10 | select o.orderNumber, productName, MSRP, priceEach from orderdetails as o right join produ...               | 1000 row(s) returned | 0.157 sec / 0.203 sec |
| 63 | 17:39:52 | select o.customerNumber, customerName, orderNumber, status from customers as c left join orders as o on ... | 350 row(s) returned  | 0.156 sec / 0.000 sec |
| 64 | 17:40:26 | select customerName, contactLastName, status from customers as c left join orders as o on ...               | 24 row(s) returned   | 0.063 sec / 0.000 sec |
| 65 | 17:40:58 | select lastName, firstName, customerName, checkNumber, amount from employees as e left j...                 | 283 row(s) returned  | 0.141 sec / 0.000 sec |

*Para estas consultas usa RIGHT JOIN*

8. Repite los ejercicios 5 a 7 usando *RIGHT JOIN*.
9. Obtén el número de cliente, nombre de cliente, número de orden y estado de cada cliente.

*select o.customerNumber, customerName, orderNumber, status from customers as c*

*right join orders as o on c.customerNumber=o.customerNumber;*

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3\*, SQL File 4\*, employees

SCHEMAS: AndresRamirez, ArmandoGomez, cerveza, classicmodels

Tables: customers, employees, columns, indexes, foreign keys, triggers, offices, orderdetails, orders, payments, productlines, products, views, stored procedures.

Administration: Schemas

Information: Object Info, Session

Schema: classicmodels

Result Grid: customerNumber, customerName, orderNumber, status

| customerNumber | customerName                 | orderNumber | status  |
|----------------|------------------------------|-------------|---------|
| 363            | Online Diecast Creations Co. | 10100       | Shipped |
| 128            | Blauer See Auto, Co.         | 10101       | Shipped |
| 181            | Vitachrome Inc.              | 10102       | Shipped |
| 121            | Baane Mini Imports           | 10103       | Shipped |
| 141            | Euro+ Shopping Channel       | 10104       | Shipped |
| 145            | Danish Wholesale Imports     | 10105       | Shipped |
| 278            | Rovelli Gifts                | 10106       | Shipped |
| ...            | ...                          | ...         | ...     |
| 37             | ...                          | 10107       | Shipped |

Output: Action Output

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 63 | 17:39:52 | select o.customerNumber, customerName, orderNumber, status from customers as c left join ...  | 350 row(s) returned | 0.156 sec / 0.000 sec |
| 64 | 17:40:26 | select customerName, contactLastName, status from customers as c left join orders as o on ... | 24 row(s) returned  | 0.063 sec / 0.000 sec |
| 65 | 17:40:58 | select lastName, firstName, customerName, checkNumber, amount from employees as e left ...    | 283 row(s) returned | 0.141 sec / 0.000 sec |
| 66 | 17:44:03 | select o.customerNumber, customerName, orderNumber, status from customers as c right jo ...   | 326 row(s) returned | 0.141 sec / 0.000 sec |

SQLAdditions: Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

10. Obtén los clientes que no tienen una orden asociada.

```
select customerName, contactLastName, status from customers as c right join orders as o
on c.customerNumber=o.customerNumber where o.customerNumber is = Null;
```

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Fundamentos2\*, PROYECTO SESION 1, PROYECTO SESION 2\*, PROYECTO SESION 3\*, SQL File 4\*, employees, customers, orders

SCHEMAS: AndresRamirez, ArmandoGomez, cerveza, classicmodels

Tables: customers, employees, columns, indexes, foreign keys, triggers, offices, orderdetails, orders, payments, productlines, products, views, stored procedures.

Administration: Schemas

Table: orders

Columns: orderNumber, int Primary Key, orderDate, date, requiredDate, date, shippedDate, date, status, varchar(1), comments, text

Object Info, Session

Result Grid: customerName, contactLastName, status

| customerName | contactLastName | status |
|--------------|-----------------|--------|
|--------------|-----------------|--------|

Output: Action Output

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 73 | 17:50:12 | SELECT * FROM classicmodels.orders LIMIT 0, 1000  | 326 row(s) returned | 0.156 sec / 0.047 sec |
| 74 | 17:50:49 | select customerName, contactLastName, status from customers as c right join orders as o ... | 0 row(s) returned   | 0.063 sec / 0.000 sec |

SQLAdditions: Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

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

```
select lastName, firstName, customerName, checkNumber, amount from employees as e right  
join customers as c
```

```
on e.employeeNumber=c.salesRepEmployeeNumber right join payments as p on  
c.customerNumber=p.customerNumber;
```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the tabs 'PROYECTO SESION 1' and 'PROYECTO SESION 3\*' are selected. The main area displays a query in the SQL pane:

```
38 # Obtén el apellido de empleado, nombre de empleado, nombre de cliente, número de cheque y total, es decir, los clientes que  
39 • select lastName, firstName, customerName, checkNumber, amount from employees as e right join customers as c  
40 on e.employeeNumber=c.salesRepEmployeeNumber right join payments as p on c.customerNumber=p.customerNumber;
```

The results pane below shows a table with columns: lastName, firstName, customerName, checkNumber, and amount. The data includes rows for Hernandez, Gerard, Thompson, Leslie, and Fixter.

| lastName  | firstName | customerName               | checkNumber | amount   |
|-----------|-----------|----------------------------|-------------|----------|
| Hernandez | Gerard    | Atelier graphique          | HQ336336    | 6066.78  |
| Hernandez | Gerard    | Atelier graphique          | JM555205    | 14571.44 |
| Hernandez | Gerard    | Atelier graphique          | OM314933    | 1676.14  |
| Thompson  | Leslie    | Signal Gift Stores         | BO864823    | 14191.12 |
| Thompson  | Leslie    | Signal Gift Stores         | HQ55022     | 32641.98 |
| Thompson  | Leslie    | Signal Gift Stores         | ND748579    | 33347.88 |
| Fixter    | Andy      | Australian Collectors, Co. | GG31455     | 45864.03 |
| Fixter    | Andy      | Australian Collectors, Co. | MA765515    | 82261.22 |
| Fixter    | Andy      | Australian Collectors, Co. | NP603840    | 7565.08  |
| Fixter    | Andy      | Australian Collectors, Co. | NR27552     | 44894.74 |
|           |           | Canadian Catalogue         | PP002204    | 10201.02 |

The bottom pane shows the 'Output' section with two log entries:

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 67 | 17:44:37 | select customerName, contactLastName, status from customers as c right join orders as o on c.customerNumber=o.customerNumber  | 0 row(s) returned   | 0.078 sec / 0.000 sec |
| 68 | 17:46:03 | select lastName, firstName, customerName, checkNumber, amount from employees as e right join customers as c on e.employeeNumber=c.salesRepEmployeeNumber right join payments as p on c.customerNumber=p.customerNumber; | 273 row(s) returned | 0.141 sec / 0.000 sec |

12. Escoge 3 consultas de los ejercicios anteriores, crea una vista y escribe una consulta para cada una.

## PROYECTO SESION 4

Las consultas se realizarán sobre la base sample\_training.

Todas las consultas que realices deberás mantenerlas dentro del MongoDB Compass. Para hacer esto, da clic en el botón con los puntos ... y en Toggle Query History. Busca la última consulta y agrégala a favoritos presionando el ícono con la estrella ⭐.

1. Obtén los datos de contacto de cada compañía.

### PROJECT

```
{  
  name: 1,  
  homepage_url: 1,  
  email_address: 1,  
  phone_number: 1,  
  _id: 0  
}
```

MongoDB Compass - 52.12.20.255:27017/sample\_training.companies

Connect View Collection Help

**Local**

13 DBS 53 COLLECTIONS C

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

sample\_training

- companies
- grades
- inspections
- posts
- routes
- stories
- trips
- tweets
- zips

> sample\_weatherdata

> MongoSH Beta

sample\_training.companies

Documents Aggregations Schema Explain Plan Indexes Validation

DOCUMENTS 9.5k TOTAL SIZE 34.8MB AVG. SIZE 3.7KB INDEXES 1 TOTAL SIZE 100.0KB AVG. SIZE 100.0KB

FILTER PROJECT SORT COLLATION MAXITEMS 5000 SKIP 0 LIMIT 0

VIEW ( )

Displaying documents 1 - 20 of 9500 < > C REFRESH

```

name: "Wetpaint"
homepage_url: "http://wetpaint-inc.com"
email_address: "info@wetpaint.com"
phone_number: "286.859.6300"

name: "Facebook"
homepage_url: "http://facebook.com"
email_address: ""
phone_number: ""

name: "Omnidrive"
homepage_url: "http://www.omnidrive.com"
email_address: "info@omnidrive.com"
phone_number: "660-675-5052"

```

## 2. Obtén la fuente de cada tweet.

### PROJECT

```
{
  source: 1,
  _id: 0
}
```

MongoDB Compass - 52.12.20.255:27017/sample\_training.tweets

Connect View Collection Help

**Local**

13 DBS 53 COLLECTIONS C

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

sample\_training

- companies
- grades
- inspections
- posts
- routes
- stories
- trips
- tweets
- zips

> sample\_weatherdata

> MongoSH Beta

sample\_training.tweets

Documents Aggregations Schema Explain Plan Indexes Validation

DOCUMENTS 24.8k TOTAL SIZE 39.4MB AVG. SIZE 1.6KB INDEXES 1 TOTAL SIZE 232.0KB AVG. SIZE 232.0KB

FILTER PROJECT SORT COLLATION MAXITEMS 5000 SKIP 0 LIMIT 0

VIEW ( )

Displaying documents 1 - 20 of 24832 < > C REFRESH

```

source: "web"

source: "<a href="http://www(tweetdeck.com" rel="nofollow">TweetDeck</a>"

source: "<a href="http://blackberry.com/twitter" rel="nofollow">Twitter for Bla...</a>"

source: "<a href="http://www.echofon.com/" rel="nofollow">Echofon</a>"

source: "<a href="http://83degrees.com/to/powertwitter" rel="nofollow">Power Tw...</a>"

source: "web"

```

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

## FILTER

```
{  
  founded_month: 10  
}  
  
PROJECT  
{  
  name: 1,  
  founded_month: 1,  
  _id: 0  
}
```

MongoDB Compass - 52.12.20.255:27017/sample\_training.companies

Connect View Collection Help

Local

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

sample\_training

- companies
- grades
- inspections
- posts
- routes
- stories
- trips
- tweets
- zips

> sample\_weatherdata

> \_MongoSH Beta

sample\_training.companies

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER {  
 founded\_month: 10  
}  
  
PROJECT {  
 name: 1,  
 founded\_month: 1,  
 \_id: 0  
}  
  
SORT  
COLLATION

MAXITEMS 5000 SKIP 0 LIMIT 0

Displaying documents 1 - 20 of 301

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

## FILTER

```
{  
  founded_year: 2008  
}  
  
PROJECT  
{  
  name: 1,  
  founded_year: 1,  
  _id: 0  
}
```

MongoDB Compass - 52.12.20.255:27017/sample\_training.companies

Connect View Collection Help

**Local**

13 DBS 53 COLLECTIONS C

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

sample\_training

- companies
- grades
- inspections
- posts
- routes
- stories
- trips
- tweets
- zips

> sample\_weatherdata

> MongoSH Beta

sample\_training.companies

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { founded\_year: 2008 }

PROJECT { name: 1, founded\_year: 1, \_id: 0 }

SORT

COLLATION

MAXITEMS 5000 SKIP 0 LIMIT 0

Displaying documents 1 - 20 of 1224 FIND RESET ...

|                      |                    |
|----------------------|--------------------|
| name : "OpenX"       | founded_year: 2008 |
| name : "WonderHowTo" | founded_year: 2008 |
| name : "First30Days" | founded_year: 2008 |
| name : "Mibura"      | founded_year: 2008 |

## 5. Obtén todos los *post* del autor `machine`.

FILTER

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

PROJECT

```
{
  title: 1,
  body: 1,
  author: 1,
  _id: 0
}
```

MongoDB Compass - 52.12.20.255:27017/sample\_training.posts

Connect View Collection Help

Local

13 DBS 53 COLLECTIONS C

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

sample\_training

- companies
- grades
- inspections
- posts**
- routes
- stories
- trips
- tweets
- zips

> sample\_weatherdata

> MongoSH Beta

sample\_training.posts

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER {  
  author: 'machine'  
}  
PROJECT {  
  title: 1,  
  body: 1,  
  author: 1,  
  \_id: 0  
}  
SORT  
COLLATION

OPTIONS FIND RESET ...

DOCUMENTS 500 TOTAL SIZE 16.6MB AVG. SIZE 34.1KB INDEXES 1 TOTAL SIZE 20.0KB AVG. SIZE 20.0KB

MAXITEMS 5000 SKIP 0 LIMIT 0

Displaying documents 1 - 20 of 500 < > C REFRESH

```

body: "Amendment I
      <p>Congress shall make no law respecting an establishment ...
      author: "machine"
      title: "Bill of Rights"
    
```

```

body: "We the People of the United States, in Order to Form a more perfect Un..."
      author: "machine"
      title: "US Constitution"
    
```

```

body: "Four score and seven years ago our fathers brought forth on this conti..."
      author: "machine"
      title: "Gettysburg Address"
    
```

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

FILTER

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

PROJECT

```
{
  source: 1,
  text: 1,
  _id: 0
}
```

MongoDB Compass - 52.12.20.255:27017/sample\_training.tweets

Connect View Collection Help

**Local**

13 DBS 53 COLLECTIONS C

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

sample\_training

- companies
- grades
- inspections
- posts
- routes
- stories
- trips
- tweets**
- ...

> sample\_weatherdata

> MongoSH Beta

sample\_training.tweets

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { source: 'web' }

PROJECT { source: 1, text: 1, \_id: 0 }

SORT

COLLATION

MAXITEMS 5000 SKIP 0 LIMIT 0

Displaying documents 1 - 20 of 11141

text: "eu preciso de terminar de fazer a minha tabela, está muito foda \*\*"  
source: "web"

text: "First week of school is over :P"  
source: "web"

text: "fair today!!!! then jersey shore!!!=D"  
source: "web"

text: "@teetooligit lmfaoo!! No BS!ahaha"  
source: "web"

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

### FILTER

```
{
  founded_month: 10,
  founded_year: 2008
}
```

### PROJECT

```
{
  name: 1,
  founded_month: 1,
  founded_year: 1,
  _id: 0
}
```

MongoDB Compass - 52.12.20.255:27017/sample\_training.companies

Connect View Collection Help

Local

13 DBS 53 COLLECTIONS

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

sample\_training

- companies**
- ...  
grades  
inspections  
posts  
routes  
stories  
trips  
tweets  
zips

> sample\_weatherdata

> MongoSH Beta

sample\_training.companies

Documents Aggregations Schema Explain Plan Indexes Validation

DOCUMENTS 9.5k TOTAL SIZE 34.8MB AVG. SIZE 3.7KB INDEXES 1 TOTAL SIZE 100.0KB AVG. SIZE 100.0KB

FILTER {  
  "Founded\_Month": 10,  
  "Founded\_Year": 2008  
}  
PROJECT {  
  "name": 1,  
  "Founded\_Month": 1,  
  "Founded\_Year": 1,  
  "\_id": 0  
}  
SORT  
COLLATION

MAXITEMS 5000 SKIP 0 LIMIT 0

Displaying documents 1 - 20 of 63

|   | name      | Founded Year | Founded Month |
|---|-----------|--------------|---------------|
| 1 | tunesBag  | 2008         | 10            |
| 2 | Huecs     | 2008         | 10            |
| 3 | Rush Hour | 2008         | 10            |

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

### FILTER

```
{
  "number_of_employees": {
    "$gte": 50
  }
}
```

### PROJECT

```
{
  "name": 1,
  "number_of_employees": 1,
  "_id": 0
}
```

MongoDB Compass - 52.12.20.255:27017/sample\_training.companies

Connect View Collection Help

**Local**

13 DBS 53 COLLECTIONS

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

sample\_training

- companies
- ... (ellipsis)
- grades
- inspections
- posts
- routes
- stories
- trips
- tweets
- zips

> sample\_weatherdata

> MongoSH Beta

sample\_training.companies

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER: { number\_of\_employees: { \$gte: 50 } }

PROJECT: { name: 1, number\_of\_employees: 1, \_id: 0 }

SORT

COLLATION

MAXTIMEMS 5000

SKIP 0

LIMIT 0

Displaying documents 1 - 20 of 904

| name     | number_of_employees |
|----------|---------------------|
| Facebook | 5299                |
| Twitter  | 1300                |
| Scribd   | 50                  |
| Plaxo    | 50                  |

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

### FILTER

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

### PROJECT

```
{
description: 1,
comments: 1,
title: 1,
_id: 0
}
```

### SORT

```
{
comments: 1
}
```

MongoDB Compass - 52.12.20.255:27017/sample\_training.stories

Connect View Collection Help

**Local**

13 DBS 53 COLLECTIONS C

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

sample\_training

- companies
- grades
- inspections
- posts
- routes
- stories
- ... (ellipsis)
- trips
- tweets
- zips

> sample\_weatherdata

> MongoSH Beta

sample\_training.stories

Documents Aggregations Schema Explain Plan Indexes Validation

**FILTER** { \$and: [ { comments: { \$gte: 10 } }, { comments: { \$lt: 30 } } ] }

**PROJECT** { description: 1, comments: 1, title: 1, \_id: 0 }

**SORT** { comments: 1 }

**COLLATION**

OPTIONS MAXTIMEMS 5000 SKIP 0 LIMIT 0

Displaying documents 1 - 20 of 1930

REFRESH

title: "OFIS Unveils Earthquake-Proof Solar Powered Volcano Towers"  
comments: 10  
description: "At a time when earthquakes dominate the headlines, we think it's more ..."

title: "The Financial Checklist Manifesto"  
comments: 10  
description: "A good checklist gets used over and over and is refined to the most im..."

title: "Hyundai Sponsors Oscars , Replaces Bridges With Basinger"  
comments: 10  
description: "You have to hand it to Hyundai: the company is pretty good at making h..."

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

PROJECT

```
{
  name: 1,
  number_of_employees: 1,
  _id: 0
}
```

SORT

```
{
  number_of_employees: 1
}
```

LIMIT

```
1
```

MongoDB Compass - 52.12.20.255:27017/sample\_training.companies

Connect View Collection Help

**Local**

13 DBS 53 COLLECTIONS C

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

sample\_training

- companies
- grades
- inspections
- posts
- routes
- stories
- trips
- tweets
- zips

> sample\_weatherdata

> MongoSH Beta

sample\_training.companies

Documents Aggregations Schema Explain Plan Indexes Validation

DOCUMENTS 9.5k TOTAL SIZE 34.8MB AVG. SIZE 3.7KB INDEXES 1 TOTAL SIZE 100.0KB AVG. SIZE 100.0KB

FILTER

PROJECT

```
{
  name: 1,
  number_of_employees: 1,
  _id: 0
}
```

SORT

```
{
  number_of_employees: 1
}
```

COLLATION

MAXITEMS 5000

Skip 0

LIMIT 1

Displaying documents 1 - 1 of 1 < > C REFRESH

name : "Omnidrive"  
number\_of\_employees: null

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

#### PROJECT

```
{
  name: 1,
  number_of_employees: 1,
  _id: 0
}
```

#### SORT

```
{
  number_of_employees: -1
}
```

#### LIMIT

```
1
```

MongoDB Compass - 52.12.20.255:27017/sample\_training.companies

Local

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

sample\_training

- companies
- grades
- inspections
- posts
- routes
- stories
- trips
- tweets
- zips

> sample\_weatherdata

> MongoSH Beta

sample\_training.companies

Documents Aggregations Schema Explain Plan Indexes Validation

**FILTER**

**PROJECT** {  
  name: 1,  
  number\_of\_employees: 1,  
  \_id: 0  
}

**SORT** {  
  number\_of\_employees: - 1  
}

**COLLATION**

**OPTIONS**

**MAXITEMS** 5000

**Skip** 0    **LIMIT** 1

Displaying documents 1 - 1 of 1

| name | number_of_employees |
|------|---------------------|
| IBM  | 388000              |

## 12. Obtén la historia más comentada.

PROJECT

```
{
  title: 1,
  comments: 1,
  _id: 0
}
```

SORT

```
{
  comments: -1
}
```

LIMIT

```
1
```

MongoDB Compass - 52.12.20.255:27017/sample\_training.stories

Connect View Collection Help

**Local**

13 DBS 53 COLLECTIONS C

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

sample\_training

- companies
- grades
- inspections
- posts
- routes
- stories**
- ...  
trips  
tweets  
zips

> sample\_weatherdata

> MongoSH Beta

sample\_training.stories

Documents Aggregations Schema Explain Plan Indexes Validation

DOCUMENTS 9.8k TOTAL SIZE 9.6MB AVG. SIZE 1.0KB INDEXES 2 TOTAL SIZE 176.0KB AVG. SIZE 88.0KB

FILTER

PROJECT {  
 title: 1,  
 comments: 1,  
 \_id: 0  
}

SORT {  
 comments: -1  
}

COLLATION

MAXITEMS 5000  
SKIP 0  
LIMIT 1

VIEW

Displaying documents 1 - 1 of 1 < > C REFRESH

```
title: "Republican Brown wins Massachusetts Senate seat!"  
comments: 1864
```

### 13. Obtén la historia menos comentada.

PROJECT

```
{
  title: 1,
  comments: 1,
  _id: 0
}
```

SORT

```
{
  comments: 1
}
```

LIMIT

```
1
```

MongoDB Compass - 52.12.20.255:27017/sample\_training.stories

Local

HOST: 52.12.20.255:27017

CLUSTER: Standalone

EDITION: MongoDB 4.2.10 Community

sample\_training

- companies
- grades
- inspections
- posts
- routes
- stories**
- ... (trips, tweets, zips)

> sample\_weatherdata

sample\_training.stories

Documents Aggregations Schema Explain Plan Indexes Validation

DOCUMENTS 9.8k TOTAL SIZE 9.6MB AVG. SIZE 1.0KB INDEXES 2 TOTAL SIZE 176.0KB AVG. SIZE 88.0KB

FILTER PROJECT { title: 1, comments: 1, \_id: 0 }

SORT { comments: 1 } MAXITEMS 5000

COLLATION SKIP 0 LIMIT 1

Displaying documents 1 - 1 of 1

title: "UA Tech Park chosen for \$32 million 'Solar Zone' project"  
comments: 0

## PROYECTO SESIÓN 5

Para este proyecto deberás practicar en el uso de agregaciones, pues serán usadas durante la siguiente sesión.

La base de datos y colección que debes usar es `sample_airbnb.listingsAndReviews`.

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.

```
[{$project: {
    "number_of_reviews": 1,
    "review_scores.review_scores_rating": 1,
    "amenities": 1,
    "address.country": 1,
    "_id: 0
}}, {$match: {
    $and: [
        { number_of_reviews: {$gte: 50} },
        { review_scores.rating: {$gte: 80} },
        { connection: "Cable" }
    ]
}}]
```

```

        { "review_scores.review_scores_rating": {$gte: 80} },
        { amenities: { $in: [ /Ethernet connection/i ] } },
        { "address.country": /Brazil/i }
    ]
})

```

MongoDB Compass - 52.12.20.255:27017/sample\_airbnb.listingsAndReviews

Local

sample\_airbnb.listingsA... Aggregations

sample\_airbnb.listingsAndReviews

DOCUMENTS 5.6k TOTAL SIZE 90.0MB AVG. SIZE 16.6KB INDEXES 4 TOTAL SIZE 476.0KB AVG. SIZE 119.0KB

Aggregations

COLLATION PROYECTO SESION 6.b SAVE SAMPLE MODE AUTO PREVIEW

\$project

Output after \$project stage (Sample of 20 documents)

```

1 {
2   "number_of_reviews": 1,
3   "review_scores.review_scores_rating": 1,
4   "amenities": 1,
5   "address.country": 1,
6   "_id": 0
7 }
8
9

```

number\_of\_reviews: 51  
▶ amenities: Array  
▶ address: Object  
▶ review\_scores: Object

number\_of\_reviews: 0  
▶ amenities: Array  
▶ address: Object  
▶ review\_scores: Object

\$match

Output after \$match stage (Sample of 6 documents)

```

1 {
2   "$and": [
3     { "number_of_reviews": {$gte: 50} },
4     { "review_scores.review_scores_rating": {$gte: 80} },
5     { "amenities": { $in: [ /Ethernet connection/i ] } },
6     { "address.country": /Brazil/i }
7   ]
8 }
9

```

number\_of\_reviews: 110  
▶ amenities: Array  
▶ address: Object  
▶ review\_scores: Object

number\_of\_reviews: 76  
▶ amenities: Array  
▶ address: Object  
▶ review\_scores: Object

## PROYECTO SESION 6

Continuaremos trabajando con la base de datos de películas y sus comentarios.

El proyecto consiste en obtener, por país, el número de películas que hay de cada género. Un ejemplo de salida en formato de tabla sería:

| país | genero | peliculas |
|------|--------|-----------|
|------|--------|-----------|

|     |       |    |
|-----|-------|----|
| USA | Short | 10 |
|-----|-------|----|

|     |       |    |
|-----|-------|----|
| USA | Drama | 20 |
|-----|-------|----|

|     |     |     |
|-----|-----|-----|
| ... | ... | ... |
|-----|-----|-----|

```
[$project:  
{  
  title: 1,  
  genres: 1,  
  countries: 1,  
  _id: 0  
}, {$unwind:  
{  
  path: '$countries'  
}, {$unwind:  
{  
  path: '$genres'  
}, {$group:  
{  
  _id: {  
    pais: '$countries',  
    genero: '$genres'  
  },  
  titulos: {  
    $sum: 1  
  }  
}, {$project:  
{  
  pais: '$_id.pais',
```

```
genero: '$_id.genero',
```

```
titulos: 1,
```

```
_id: 0
```

```
}}, {$sort:
```

```
{
```

```
titulos: -1
```

```
}}]
```

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

- Left Sidebar (Local):** Lists databases (13 DBS) and collections (53 COLLECTIONS). The "movies" collection is selected.
- Top Bar:** Shows the connection information (HOST: 52.12.20.255:27017) and the database (sample\_mflix.movies).
- Aggregations Tab:** Active tab, showing the aggregation pipeline stages.
- Preview of Documents in the Collection:** Shows 23,5k documents with total size of 35.9MB and avg size of 1.6KB. It also shows 2 indexes with total size of 13.1MB and avg size of 6.6MB.
- Aggregation Pipeline Stages:**
  - Stage 1:** \$project (Sample Mode: ON, Auto Preview: ON). The output is a sample of 20 documents, each containing fields: \_id, plot, genres, runtime, cast, num\_mflix\_comments, title, and thumbnail.
  - Stage 2:** \$sort (Sort by titulos: -1).
- Output after \$project Stage:** Shows the sample of 20 documents with the \$project stage applied.

MongoDB Compass - 52.12.20.255:27017/sample\_mflix.movies

Connect View Collection Help

**Local**

13 DBS 53 COLLECTIONS

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

RebecaJuarezEj3

Sesion05ProyectoFabio...

comments

movies

sessions

theaters

users

> sample\_supplies

> sample\_training

> sample\_weatherdata

+ >\_MongoSH Beta

sample\_mflix.movies

Aggregations

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION PROYECTO SESION 6 SAVE SAMPLE MODE AUTO PREVIEW

\$unwind

Output after \$unwind stage (Sample of 20 documents)

```

1
2  {
3     path: '$countries'
4 }

```

```

▶ genres: Array
    title: "Blacksmith Scene"
    countries: "USA"

```

```

▶ genres: Array
    title: "The Great Train Robbery"
    countries: "USA"

```

\$unwind

Output after \$unwind stage (Sample of 20 documents)

```

1
2  {
3     path: '$genres'
4 }

```

```

▶ genres: "Short"
    title: "Blacksmith Scene"
    countries: "USA"

```

```

▶ genres: "Short"
    title: "The Great Train Robbery"
    countries: "USA"

```

MongoDB Compass - 52.12.20.255:27017/sample\_mflix.movies

Connect View Collection Help

**Local**

13 DBS 53 COLLECTIONS

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

RebecaJuarezEj3

Sesion05ProyectoFabio...

comments

movies

sessions

theaters

users

> sample\_supplies

> sample\_training

> sample\_weatherdata

+ >\_MongoSH Beta

sample\_mflix.movies

Aggregations

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION PROYECTO SESION 6 SAVE SAMPLE MODE AUTO PREVIEW

\$group

Output after \$group stage (Sample of 20 documents)

```

1
2  {
3     _id: {
4         pais: '$countries',
5         genero: '$genres'
6     },
7     titulos: {
8         $sum: 1
9     }
10 }

```

```

▶ _id:Object
    titulos: 17

```

```

▶ _id:Object
    titulos: 2

```

\$project

Output after \$project stage (Sample of 20 documents)

```

1
2  [
3     {
4         pais: '_id.pais',
5         genero: '_id.genero',
6         titulos: 1,
7         _id: 0
8     }
9 ]

```

```

titulos: 2
pais: "Hungary"
genero: "Sci-Fi"

```

```

titulos: 2
pais: "Federal Republic of Yugoslavia"
genero: "War"

```

MongoDB Compass - 52.12.20.255:27017/sample\_mflix.movies

Local

HOST: 52.12.20.255:27017

CLUSTER: Standalone

EDITION: MongoDB 4.2.10 Community

Filter your data:

- RebecaJuarezEj3
- Sesion05ProyectoFabio...
- comments
- movies
- sessions
- theaters
- users
- > sample\_supplies
- > sample\_training
- > sample\_weatherdata

sample\_mflix.movies

Aggregations

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION PROYECTO SESION 6 SAVE SAMPLE MODE AUTO PREVIEW

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

ADD STAGE

## PROYECTO SESION 7

A continuación se realizaran algunas operaciones de agregar, modificar y eliminar un documento JSON en una Colección.

1. Agregar los siguientes registros en formato CSV a la Colección movies  
Incluyendo nombres de columnas quedaría:

```

File Edit Selection View Go Run Terminal Help movies.csv - Visual Studio Code
Welcome # SESSION 5 RETO 2 Untitled-1 movies.csv ratings.csv users.dat directory.csv Pandemic (H1N1) 2009.csv 2019-nCoV-cases-JHU.csv ...
C: > Users > rjuarez > Desktop > 2020 > CURSO BASE DE DATOS > 07. Configuracion BD locales > ml-1m > movies.csv
1 MovieID,Title,Genres
2 1,Toy Story (1995),Animation|Children's|Comedy
3 2,Jumanji (1995),Adventure|Children's|Fantasy
4 3,Grumpier Old Men (1995),Comedy|Romance
5 4,Waiting to Exhale (1995),Comedy|Drama
6 5,Father of the Bride Part II (1995),Comedy
7 6,Heat (1995),Action|Crime|Thriller
8 7,Sabrina (1995),Comedy|Romance
9 8,Tom and Huck (1995),Adventure|Children's
10 9,Sudden Death (1995),Action
11 10,GoldenEye (1995),Action|Adventure|Thriller
12 11,American President The (1995),Comedy|Drama|Romance
13 12,Dracula, Dead and Loving It (1995),Comedy|Horror
14 13,Balto (1995),Animation|Children's
15 14,Nixon (1995),Drama
16 15,Cutthroat Island (1995),Action|Adventure|Romance
17 16,Casino (1995),Drama|Thriller
18 17,Sense and Sensibility (1995),Drama|Romance
19 18,Four Rooms (1995),Thriller
20 19,Ace Ventura, When Nature Calls (1995),Comedy
21 20,Money Train (1995),Action
22 21,Get Shorty (1995),Action|Comedy|Drama
23 22,Copycat (1995),Crime|Drama|Thriller
24 23,Assassins (1995),Thriller
25 24,Powder (1995),Drama|Sci-Fi
26 25,Leaving Las Vegas (1995),Drama|Romance
27 26,Othello (1995),Drama
28 27,Now and Then (1995),Drama
29 28,Persuasion (1995),Romance
30 29,City of Lost Children The (1995),Adventure|Sci-Fi
31 30,Shanghai Triad (Yao a yao yao da wo ipo qiao) (1995),Drama
32 31,Dangerous Minds (1995),Drama
33 32,Twelve Monkeys (1995),Drama|Sci-Fi

```

The Marketplace has extensions that can help with '.csv' files

Search Marketplace Don't Show Again for '.csv' files

Ln 1, Col 21 Spaces: 4 UTF-8 LF Plain Text

## QUERY COMPETITION SESION 8

### 1. Starbucks

Descarga la fuente de datos de los locales de Starbucks:

[directory.csv](#)

Analiza los datos, limpia los datos en caso de ser necesario.

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.

```
[$match: {  
$and:  
[ { Latitude: { $gte: 19.44 } },  
{ Longitude: { $gte: -99.2 } },  
{ Latitude: { $lte: 19.44 } },  
{ Longitude: { $lte: -99.2 } }  
]  
}  
}  
}]
```

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

- Left Sidebar:** Shows the local host (52.12.20.255:27017) and a collection named "Starbucks".
- Top Bar:** Shows the connection status and the database name "Starbucks.Location".
- Aggregations Tab:** Selected tab.
- Pipeline Stages:**
  - Stage 1: \$match (shown in the code block above)
  - Stage 2: \$addFields (not shown in the code block)
- Output:** Shows the results of the \$match stage, which includes documents for Starbucks locations in Meritxell, Ajman, and Andorra la Vella.
- Code Block:** Shows the MongoDB query code for the \$match stage.

### 2. Pandemia A (H1N1)

Descarga la fuente de datos sobre la pandemia del 2009 (H1N1):

[Pandemic \(H1N1\) 2009.csv](#)

Analiza los datos, limpia los datos en caso de ser necesario.

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:

- a. ¿Cuál fue el país con mayor número de muertes?

```
[{$match:  
 {  
   Country: {  
     $not: /Grand Total/  
   }  
 }, {$sort:  
 {  
   Deaths: -1  
 }, {$limit:  
 1}]
```

MongoDB Compass - 52.12.20.255:27017/H1N1.Cases

Connect View Collection Help

**Local**

- 13 DBS 53 COLLECTIONS C
- ☆ FAVORITE
- HOST 52.12.20.255:27017
- CLUSTER Standalone
- EDITION MongoDB 4.2.10 Community
- Q Filter your data
- > COVID
- > H1N1
- Cases ...
- > Starbucks
- > admin
- > config
- > local
- > sample\_airbnb
- > sample\_analytics
- > sample\_geospatial
- > sample\_mflix
- > \_MongoSH Beta

**H1N1.Cases**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION: ¿Cuál fue el país con mayor número de muertes? SAVE

DOCUMENTS 1.8k TOTAL SIZE 178.1KB AVG. SIZE 100B INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)

Output after \$match stage (Sample of 20 documents)

```

1
2  [
3   {
4     Country: {
5       $not: /Grand Total/
6     }
7   }
8 ]

```

\_id: ObjectId("5fc2a9998a49fb1de57b313d")  
Country: "Algeria"  
Cases: 5  
Deaths: 0  
Update Time: "7/6/2009 9:00"

\_id: ObjectId("5fc2a9998a49fb1de57b313e")  
Country: "Antigua and Barbuda"  
Cases: 2  
Deaths: 0  
Update Time: "7/6/2009 9:00"

MongoDB Compass - 52.12.20.255:27017/H1N1.Cases

Connect View Collection Help

**Local**

- 13 DBS 53 COLLECTIONS C
- ☆ FAVORITE
- HOST 52.12.20.255:27017
- CLUSTER Standalone
- EDITION MongoDB 4.2.10 Community
- Q Filter your data
- > COVID
- > H1N1
- Cases ...
- > Starbucks
- > admin
- > config
- > local
- > sample\_airbnb
- > sample\_analytics
- > sample\_geospatial
- > sample\_mflix
- > \_MongoSH Beta

**H1N1.Cases**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION: ¿Cuál fue el país con mayor número de muertes? SAVE

DOCUMENTS 1.8k TOTAL SIZE 178.1KB AVG. SIZE 100B INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

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

```

1
2  [
3   {
4     Deaths: -1
5   }
6 ]

```

\_id: ObjectId("5fc2a9998a49fb1de57b31bb")  
Country: "United States of America"  
Cases: 33902  
Deaths: 170  
Update Time: "7/6/2009 9:00"

\_id: ObjectId("5fc2a9998a49fb1de57b323b")  
Country: "United States of America"  
Cases: 33902  
Deaths: 170  
Update Time: "7/3/2009 9:00"

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

```

1
2  1

```

\_id: ObjectId("5fc2a9998a49fb1de57b31bb")  
Country: "United States of America"  
Cases: 33902  
Deaths: 170  
Update Time: "7/6/2009 9:00"

b. ¿Cuál fue el país con menor número de muertes?

[{\$match:

{

Country: {

```
$not: /Grand Total/
```

```
}
```

```
}}, {$sort:
```

```
{
```

```
Deaths: 1
```

```
}}, {$limit:
```

```
1}]
```

MongoDB Compass - 52.12.20.255:27017/H1N1.Cases

Connect View Collection Help

Local

13 DBS 53 COLLECTIONS

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

> COVID

> H1N1

Cases ...

> Starbucks

> admin

> config

> local

> sample\_airbnb

> sample\_analytics

> sample\_geospatial

> sample\_mflix

> \_MongoSH Beta

H1N1.Cases

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION ¿Cuál fue el país con menor número de muertes? SAVE

DOCUMENTS 1.8k TOTAL SIZE 178.1KB AVG. SIZE 100B INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

1822 Documents in the Collection

Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)

Preview of Documents in the Collection

\_id: ObjectId("5fc2a9998a49fb1de57b313d")  
Country: "Algeria"  
Cases: 5  
Deaths: 0  
Update Time: "7/6/2009 9:00"

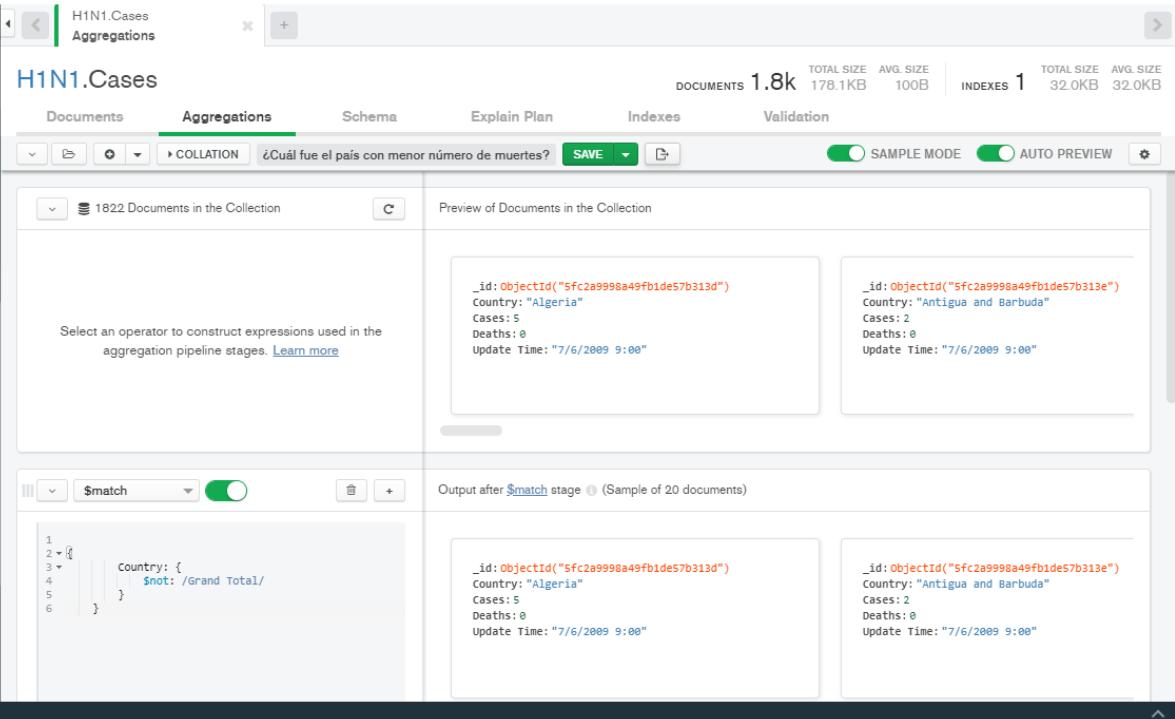
\_id: ObjectId("5fc2a9998a49fb1de57b313e")  
Country: "Antigua and Barbuda"  
Cases: 2  
Deaths: 0  
Update Time: "7/6/2009 9:00"

\$match (Sample of 20 documents)

1  
2  
3  
4     Country: {  
5        \$not: /Grand Total/  
6     }

\_id: ObjectId("5fc2a9998a49fb1de57b313d")  
Country: "Algeria"  
Cases: 5  
Deaths: 0  
Update Time: "7/6/2009 9:00"

\_id: ObjectId("5fc2a9998a49fb1de57b313e")  
Country: "Antigua and Barbuda"  
Cases: 2  
Deaths: 0  
Update Time: "7/6/2009 9:00"



HOST: 52.12.20.255:27017

CLUSTER: Standalone

EDITION: MongoDB 4.2.10 Community

Filter your data:

- > COVID
- > H1N1
  - Cases
  - ...
- > Starbucks
- > admin
- > config
- > local
- > sample\_airbnb
- > sample\_analytics
- > sample\_geospatial
- > sample\_mflix
- > \_MongoSH Beta

H1N1.Cases

Aggregations

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION: &Cuál fue el país con menor número de muertes?

SAVE

SAMPLE MODE AUTO PREVIEW

DOCUMENTS 1.8k TOTAL SIZE 178.1KB AVG. SIZE 100B INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

\$sort

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

```

1
2  {
3     |   Deaths: 1
4 }
```

\_id: ObjectId("5fc2a9998a49fb1de57b3408")
Country: "Vanuatu"
Cases: 1
Deaths: NaN
Update Time: "6/24/2009 7:00"

\_id: ObjectId("5fc2a9998a49fb1de57b3150")
Country: "Cap Verde"
Cases: 3
Deaths: 0
Update Time: "7/6/2009 9:00"

\$limit

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

```

1
2  1
```

\_id: ObjectId("5fc2a9998a49fb1de57b3408")
Country: "Vanuatu"
Cases: 1
Deaths: NaN
Update Time: "6/24/2009 7:00"

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

[{\$match:

{

Country: {

\$not: /Grand Total/

}

}}, {\$sort:

{

Cases: - 1

}}, {\$limit:

1}]

MongoDB Compass - 52.12.20.255:27017/H1N1.Cases

Connect View Collection Help

**Local**

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

- > COVID
- > H1N1
  - Cases
  - ...
- > Starbucks
- > admin
- > config
- > local
- > sample\_airbnb
- > sample\_analytics
- > sample\_geospatial
- > sample\_mflix
- > ...

> MongoSH Beta

**H1N1.Cases**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION ¿Cuál fue el país con el mayor número de casos? SAVE SAMPLE MODE AUTO PREVIEW

1822 Documents in the Collection

Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)

Preview of Documents in the Collection

\_id: ObjectId("5fc2a9998a49fb1de57b313d")  
Country: "Algeria"  
Cases: 5  
Deaths: 0  
Update Time: "7/6/2009 9:00"

\_id: ObjectId("5fc2a9998a49fb1de57b313e")  
Country: "Antigua and Barbuda"  
Cases: 2  
Deaths: 0  
Update Time: "7/6/2009 9:00"

\$match (Sample of 20 documents)

```

1
2 {
3   $not: /Grand Total/
4 }
5
6

```

\_id: ObjectId("5fc2a9998a49fb1de57b313d")  
Country: "Algeria"  
Cases: 5  
Deaths: 0  
Update Time: "7/6/2009 9:00"

\_id: ObjectId("5fc2a9998a49fb1de57b313e")  
Country: "Antigua and Barbuda"  
Cases: 2  
Deaths: 0  
Update Time: "7/6/2009 9:00"

MongoDB Compass - 52.12.20.255:27017/H1N1.Cases

Connect View Collection Help

**Local**

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

- > COVID
- > H1N1
  - Cases
  - ...
- > Starbucks
- > admin
- > config
- > local
- > sample\_airbnb
- > sample\_analytics
- > sample\_geospatial
- > sample\_mflix
- > ...

> MongoSH Beta

**H1N1.Cases**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION ¿Cuál fue el país con el mayor número de casos? SAVE SAMPLE MODE AUTO PREVIEW

1.8k DOCUMENTS TOTAL SIZE 178.1KB AVG. SIZE 100B INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

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

\$sort (Sample of 20 documents)

```

1
2 {
3   Cases: -1
4 }

```

\_id: ObjectId("5fc2a9998a49fb1de57b31bb")  
Country: "United States of America"  
Cases: 33902  
Deaths: 170  
Update Time: "7/6/2009 9:00"

\_id: ObjectId("5fc2a9998a49fb1de57b323b")  
Country: "United States of America"  
Cases: 33902  
Deaths: 170  
Update Time: "7/3/2009 9:00"

\$limit (Sample of 1 document)

\$limit (Sample of 1 document)

```

1
2 1

```

\_id: ObjectId("5fc2a9998a49fb1de57b31bb")  
Country: "United States of America"  
Cases: 33902  
Deaths: 170  
Update Time: "7/6/2009 9:00"

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

[{\$match:

{

Country: {

```
$not: /Grand Total/
```

```
}
```

```
}}, {$sort:
```

```
{
```

```
Cases: 1
```

```
}}, {$limit:
```

```
1}]
```

MongoDB Compass - 52.12.20.255:27017/H1N1.Cases

Connect View Collection Help

Local

13 DBS 53 COLLECTIONS

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

> COVID

> H1N1

Cases ...

> Starbucks

> admin

> config

> local

> sample\_airbnb

> sample\_analytics

> sample\_geospatial

> sample\_mflix

> \_MongoSH Beta

H1N1.Cases

Documents Aggregations Schema Explain Plan Indexes Validation

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

DOCUMENTS 1.8k TOTAL SIZE 178.1KB AVG. SIZE 100B INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

Preview of Documents in the Collection

Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)

\_id: ObjectId("5fc2a9998a49fb1de57b313d")  
Country: "Algeria"  
Cases: 5  
Deaths: 0  
Update Time: "7/6/2009 9:00"

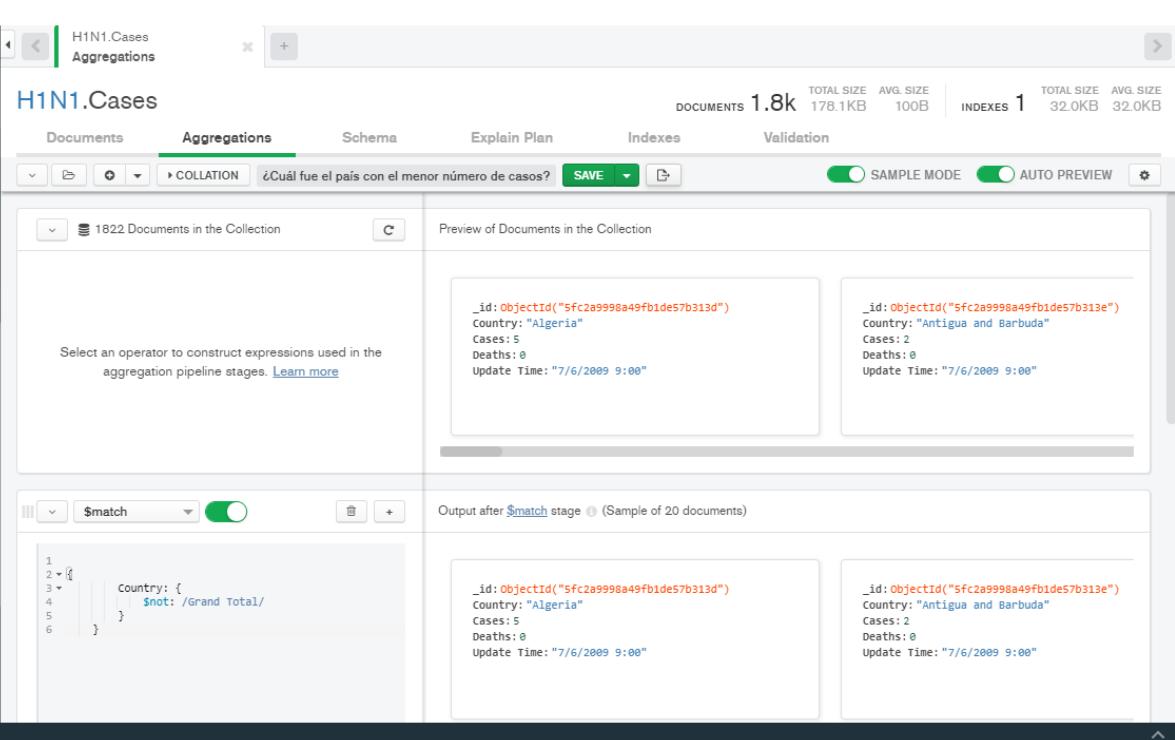
\_id: ObjectId("5fc2a9998a49fb1de57b313e")  
Country: "Antigua and Barbuda"  
Cases: 2  
Deaths: 0  
Update Time: "7/6/2009 9:00"

\$match (Sample of 20 documents)

```
1
2 {
3   $not: /Grand Total/
4   }
5 }
```

\_id: ObjectId("5fc2a9998a49fb1de57b313d")  
Country: "Algeria"  
Cases: 5  
Deaths: 0  
Update Time: "7/6/2009 9:00"

\_id: ObjectId("5fc2a9998a49fb1de57b313e")  
Country: "Antigua and Barbuda"  
Cases: 2  
Deaths: 0  
Update Time: "7/6/2009 9:00"



HOST  
52.12.20.255:27017

CLUSTER  
Standalone

EDITION  
MongoDB 4.2.10 Community

Filter your data

Covid

H1N1

Cases

Starbucks

admin

config

local

sample\_airbnb

sample\_analytics

sample\_geospatial

sample\_mflix

\_MongoSH Beta

H1N1.Cases

Documents Aggregations Schema Explain Plan Indexes Validation

SAVE SAMPLE MODE AUTO PREVIEW

DOCUMENTS 1.8k TOTAL SIZE 178.1KB AVG. SIZE 100B INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

```

1
2 { Cases: 1
3
4 }

```

`_id: ObjectId("5fc2a9998a49fb1de57b3183")  
Country: "Libya"  
Cases: 1  
Deaths: 0  
Update Time: "7/6/2009 9:00"`

`_id: ObjectId("5fc2a9998a49fb1de57b3181")  
Country: "Latvia"  
Cases: 1  
Deaths: 0  
Update Time: "7/6/2009 9:00"`

`_id: ObjectId("5fc2a9998a49fb1de57b3147")  
Country: "Bermuda, UKOT"  
Cases: 1  
Deaths: 0  
Update Time: "7/6/2009 9:00"`

\$limit (Sample of 1 document)

e. ¿Cuál fue el número de muertes promedio?

[{\$match:

{

Country: { \$not: /Grand Total/},

Deaths: { \$ne: NaN}

}}, {\$group:

{

\_id : 1,

avg: { \$avg: "\$Deaths"}

}}, {\$project:

{

promedioMuertes: { \$round : ["\$avg", 0] }

} ]

MongoDB Compass - 52.12.20.255:27017/H1N1.Cases

Connect View Collection Help

**Local**

- 13 DBS 53 COLLECTIONS C
- ☆ FAVORITE
- HOST 52.12.20.255:27017
- CLUSTER Standalone
- EDITION MongoDB 4.2.10 Community
- Filter your data
- > COVID
- > H1N1
  - Cases ...
  - Starbucks
  - admin
  - config
  - local
  - sample\_airbnb
  - sample\_analytics
  - sample\_geospatial
  - sample\_mflix
- > \_MongoSH Beta

**H1N1.Cases**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION ¿Cuál fue el número de muertes promedio? SAVE

DOCUMENTS 1.8k TOTAL SIZE 178.1KB AVG. SIZE 100B INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

Preview of Documents in the Collection

Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)

\_id: ObjectId("5fc2a9998a49fb1de57b313d")  
Country: "Algeria"  
Cases: 5  
Deaths: 0  
Update Time: "7/6/2009 9:00"

\_id: ObjectId("5fc2a9998a49fb1de57b313e")  
Country: "Antigua and Barbuda"  
Cases: 2  
Deaths: 0  
Update Time: "7/6/2009 9:00"

\$match Output after \$match stage (Sample of 20 documents)

```

1
2 <[{
3   Country: { $not: /Grand Total/},
4   Deaths: { $ne: NaN}
5 }

```

\_id: ObjectId("5fc2a9998a49fb1de57b313d")  
Country: "Algeria"  
Cases: 5  
Deaths: 0  
Update Time: "7/6/2009 9:00"

\_id: ObjectId("5fc2a9998a49fb1de57b313e")  
Country: "Antigua and Barbuda"  
Cases: 2  
Deaths: 0  
Update Time: "7/6/2009 9:00"

MongoDB Compass - 52.12.20.255:27017/H1N1.Cases

Connect View Collection Help

**Local**

- 13 DBS 53 COLLECTIONS C
- ☆ FAVORITE
- HOST 52.12.20.255:27017
- CLUSTER Standalone
- EDITION MongoDB 4.2.10 Community
- Filter your data
- > COVID
- > H1N1
  - Cases ...
  - Starbucks
  - admin
  - config
  - local
  - sample\_airbnb
  - sample\_analytics
  - sample\_geospatial
  - sample\_mflix
- > \_MongoSH Beta

**H1N1.Cases**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION ¿Cuál fue el número de muertes promedio? SAVE

DOCUMENTS 1.8k TOTAL SIZE 178.1KB AVG. SIZE 100B INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

\$project Output after \$project stage (Sample of 1 document)

```

1
2 <[{
3   _id : 1,
4   avg: { $avg: "$Deaths" }
5 }

```

\_id: 1  
avg: 2.2694444444444444

```

1
2 <[{
3   promedioMuertes: { $round: ["$avg", 0] }
4 }

```

\_id: 1  
promedioMuertes: 2

f. ¿Cuál fue el número de casos promedio?

[{\$match:

{

Country: { \$not: /Grand Total/},

```

Cases: { $ne: NaN}

}}, {$group:

{

_id : 1,

avg: { $avg: "$Cases" }

}}, {$project:

{

promedioCasos: { $round : ["$avg", 0] }

}}]

```

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

- Database:** H1N1.Cases
- Collection:** H1N1.Cases
- Aggregation Pipeline:**
  - Step 1: `Documents` (Preview: 1822 Documents in the Collection)
  - Step 2: `$match` (Output after `$match` stage: Sample of 20 documents)
    - `_id: ObjectId("5fc2a9998a49fb1de57b313d")`
    - `Country: "Algeria"`
    - `Cases: 5`
    - `Deaths: 0`
    - `Update Time: "7/6/2009 9:00"`
  - Step 3: `Group` (Output after `$group` stage: Sample of 20 documents)
    - `_id: ObjectId("5fc2a9998a49fb1de57b313d")`
    - `Country: "Algeria"`
    - `Cases: 5`
    - `Deaths: 0`
    - `Update Time: "7/6/2009 9:00"`
- Sample Mode:** Enabled
- Auto Preview:** Enabled

MongoDB Compass - 52.12.20.255:27017/H1N1.Cases

Local

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

- > COVID
- > H1N1
  - Cases
  - ...
- > Starbucks
- > admin
- > config
- > local
- > sample\_airbnb
- > sample\_analytics
- > sample\_geospatial
- > sample\_mflix
- > ...
- > MongoSH Beta

H1N1.Cases

Aggregations

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION ¿Cuál fue el número de casos promedio? SAVE SAMPLE MODE AUTO PREVIEW

Output after \$group stage (Sample of 1 document)

```

1
2 > {
3   _id : 1,
4   avg: { $avg: "$Cases" }
5 }

```

\_id: 1  
avg: 475.25707940033317

Output after \$project stage (Sample of 1 document)

```

1
2 > {
3   promedioCasos: { $round : ["$avg", 0] }
4 }

```

\_id: 1  
promedioCasos: 475

g. Top 5 de países con más muertes

```

[{$match:
{
  Country: { $not: /Grand Total/}

}}, {$group:
{
  _id: "$Country",
  sumaDeaths: {

    $sum: "$Deaths"

  }
}}, {$sort:
{
  sumaDeaths: -1

}}, {$limit:

```

5}]

MongoDB Compass - 52.12.20.255:27017/H1N1.Cases

Connect View Collection Help

Local

13 DBS 53 COLLECTIONS

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

> COVID

< H1N1

Cases ...

> Starbucks

> admin

> config

> local

> sample\_airbnb

> sample\_analytics

> sample\_geospatial

> sample\_mflix

> \_MongoSH Beta

H1N1.Cases

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Top 5 de países con más muertes

DOCUMENTS 1.8k TOTAL SIZE 178.1KB AVG. SIZE 100B INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

1822 Documents in the Collection

Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)

Preview of Documents in the Collection

`_id: ObjectId("5fc2a9998a49fb1de57b313d")  
Country: "Algeria"  
Cases: 5  
Deaths: 0  
Update Time: "7/6/2009 9:00"`

`_id: ObjectId("5fc2a9998a49fb1de57b313e")  
Country: "Antigua and Barbuda"  
Cases: 2  
Deaths: 0  
Update Time: "7/6/2009 9:00"`

\$match

Output after \$match stage (Sample of 20 documents)

`1  
2 > {  
3 country: { $not: /Grand Total/}  
4 }`

`_id: ObjectId("5fc2a9998a49fb1de57b313d")  
Country: "Algeria"  
Cases: 5  
Deaths: 0  
Update Time: "7/6/2009 9:00"`

`_id: ObjectId("5fc2a9998a49fb1de57b313e")  
Country: "Antigua and Barbuda"  
Cases: 2  
Deaths: 0  
Update Time: "7/6/2009 9:00"`

MongoDB Compass - 52.12.20.255:27017/H1N1.Cases

Connect View Collection Help

Local

13 DBS 53 COLLECTIONS

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

> COVID

< H1N1

Cases ...

> Starbucks

> admin

> config

> local

> sample\_airbnb

> sample\_analytics

> sample\_geospatial

> sample\_mflix

> \_MongoSH Beta

H1N1.Cases

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Top 5 de países con más muertes

DOCUMENTS 1.8k TOTAL SIZE 178.1KB AVG. SIZE 100B INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

\$group

Output after \$group stage (Sample of 20 documents)

`1  
2 > {  
3 _id: "$country",  
4 sumaDeaths: {  
5 $sum: "$Deaths"  
6 }  
7 }`

`_id: "Slovenia"  
sumaDeaths: 0`

`_id: "Suriname"  
sumaDeaths: 0`

\$sort

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

`1  
2 > [{  
3 sumaDeaths: -1  
4 }]`

`_id: "Mexico"  
sumaDeaths: 2271`

`_id: "United States of America"  
sumaDeaths: 1150`

h. Top 5 de países con menos muertes

[{\$match:

{

```
Country: { $not: /Grand Total/},
```

```
Deaths: { $ne: NaN}
```

```
}}, {$group:
```

```
{
```

```
_id: "$Country",
```

```
sumaDeaths: {
```

```
$sum: "$Deaths"
```

```
}
```

```
}}, {$sort:
```

```
{
```

```
sumaDeaths: 1
```

```
}}, {$limit:
```

```
5}]
```

MongoDB Compass - 52.12.20.255:27017/H1N1.Cases

Connect View Collection Help

Local

13 DBs 53 COLLECTIONS

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

> COVID

< H1N1

Cases ...

> Starbucks

> admin

> config

> local

> sample\_airbnb

> sample\_analytics

> sample\_geospatial

> sample\_mflix

> ...

H1N1.Cases

Aggregations

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Top 5 de países con menos muertes

DOCUMENTS 1.8k TOTAL SIZE 178.1KB AVG. SIZE 100B INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

SAMPLE MODE AUTO PREVIEW

1822 Documents in the Collection

Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)

Preview of Documents in the Collection

\_id: ObjectId("5fc2a9998a49fb1de57b313d")  
Country: "Algeria"  
Cases: 5  
Deaths: 0  
Update Time: "7/6/2009 9:00"

\_id: ObjectId("5fc2a9998a49fb1de57b313e")  
Country: "Antigua and Barbuda"  
Cases: 2  
Deaths: 0  
Update Time: "7/6/2009 9:00"

\$match

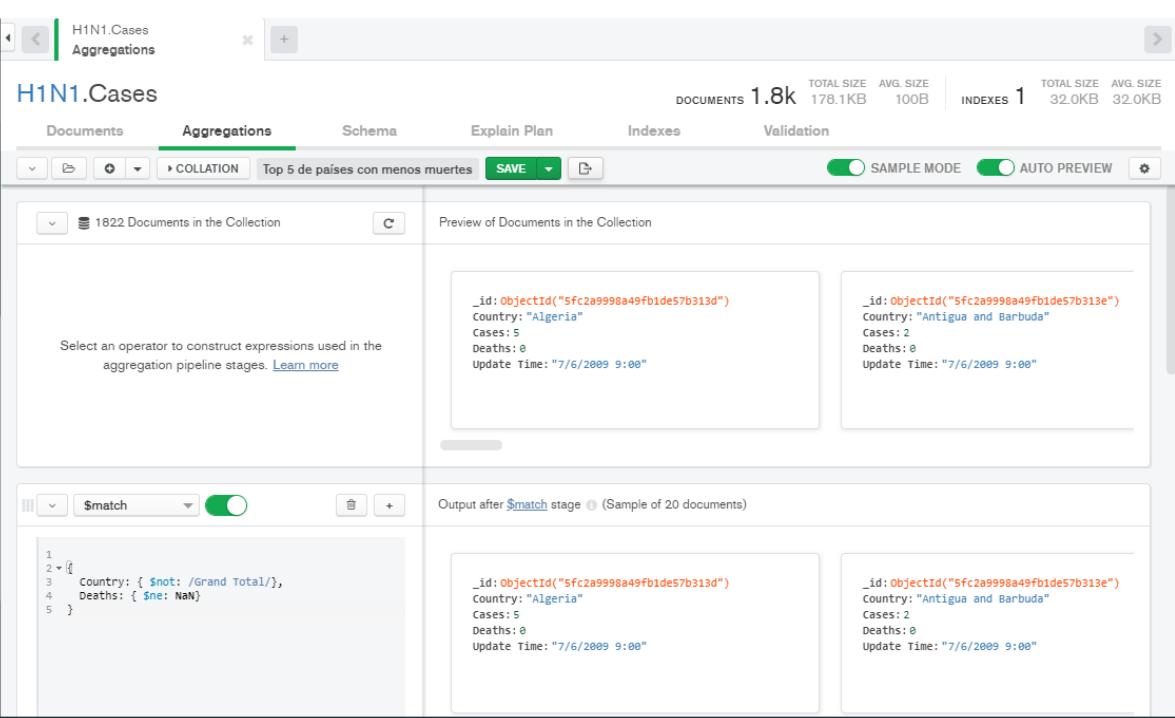
Output after \$match stage (Sample of 20 documents)

1  
2 > [ {  
3 Country: { \$not: /Grand Total/},  
4 Deaths: { \$ne: NaN}  
5 } ]

\_id: ObjectId("5fc2a9998a49fb1de57b313d")  
Country: "Algeria"  
Cases: 5  
Deaths: 0  
Update Time: "7/6/2009 9:00"

\_id: ObjectId("5fc2a9998a49fb1de57b313e")  
Country: "Antigua and Barbuda"  
Cases: 2  
Deaths: 0  
Update Time: "7/6/2009 9:00"

> MongoSH Beta



MongoDB Compass - 52.12.20.255:27017/H1N1.Cases

Connect View Collection Help

**Local**

- 13 DBS 53 COLLECTIONS C
- ☆ FAVORITE
- HOST 52.12.20.255:27017
- CLUSTER Standalone
- EDITION MongoDB 4.2.10 Community
- Filter your data
- > COVID
- > H1N1
- Cases ...
- > Starbucks
- > admin
- > config
- > local
- > sample\_airbnb
- > sample\_analytics
- > sample\_geospatial
- > sample\_mflix
- > \_MongoSH Beta

**H1N1.Cases**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Top 5 de países con menos muertes SAVE

Output after \$group stage (Sample of 20 documents)

```

1
2 < [
3   {
4     "_id": "$Country",
5     "sumaDeaths": {
6       "$sum": "$Deaths"
7     }
8   }
9 ]

```

\_id: "Honduras" sumaDeaths: 5

\_id: "Mauritius" sumaDeaths: 0

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

```

1
2 < [
3   {
4     "sumaDeaths": 1
5   }
6 ]

```

\_id: "Venezuela" sumaDeaths: 0

\_id: "El Salvador" sumaDeaths: 0

MongoDB Compass - 52.12.20.255:27017/H1N1.Cases

Connect View Collection Help

**Local**

- 13 DBS 53 COLLECTIONS C
- ☆ FAVORITE
- HOST 52.12.20.255:27017
- CLUSTER Standalone
- EDITION MongoDB 4.2.10 Community
- Filter your data
- > COVID
- > H1N1
- Cases ...
- > Starbucks
- > admin
- > config
- > local
- > sample\_airbnb
- > sample\_analytics
- > sample\_geospatial
- > sample\_mflix
- > \_MongoSH Beta

**H1N1.Cases**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Top 5 de países con menos muertes SAVE

Output after \$group stage (Sample of 20 documents)

```

1
2 < [
3   {
4     "_id": "$Country",
5     "sumaDeaths": {
6       "$sum": "$Deaths"
7     }
8   }
9 ]

```

\_id: "Venezuela" sumaDeaths: 0

\_id: "El Salvador" sumaDeaths: 0

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

```

1
2 < [
3   {
4     "sumaDeaths": 1
5   }
6 ]

```

\_id: "Suriname" sumaDeaths: 0

\_id: "Cambodia" sumaDeaths: 0

### 3. Pandemia A (H1N1)

Descarga la fuente de datos de los casos sobre la pandemia del COVID-19:

[2019-nCoV-cases-JHU.csv](#)

Analiza los datos, limpia los datos en caso de ser necesario.

Eige 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?

```
[$match:
```

```
{
```

```
Region: { $not: /Null/ }
```

```
}}, {$sort:
```

```
{
```

```
Confirmed: -1
```

```
}}, {$limit:
```

```
1}]
```

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

- Left Sidebar:** Shows "Local" selected, 13 DBs, and 53 collections. A "COVID" section is expanded, showing "Cases" and "Cases2".
- Top Bar:** Shows "COVID.Cases" and "Aggregations".
- Main Area:** The "COVID.Cases" collection is selected. The "Aggregations" tab is active.
- Query Editor:** The pipeline is defined as follows:
  - \$match: { Region: { \$not: /Null/ } }
  - \$sort: { Confirmed: -1 }
- Output:** Two sample documents are shown after each stage.
  - After \$match stage:**

```
_id: ObjectId("5fc2b6948a49fb1de57b385b")
Date: "3/3/2020 12:00"
Province: "Hubei"
Region: "Mainland China"
Last Update: "2020-03-03T11:43:02"
Confirmed: 67217
Deaths: 2835
Recovered: 36208
Lat: 30.9756
```
  - After \$sort stage:**

```
_id: ObjectId("5fc2b6948a49fb1de57b385b")
Date: "3/3/2020 12:00"
Province: ""
Region: "South Korea"
Last Update: "2020-03-03T09:43:02"
Confirmed: 5186
Deaths: 28
Recovered: 30
Lat: 36
```

MongoDB Compass - 52.12.20.255:27017/COVID.Cases

Connect View Collection Help

Local

13 DBS 53 COLLECTIONS

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

COVID

- Cases
- Cases2
- H1N1
- Starbucks
- admin
- config
- local
- sample\_airbnb
- sample\_analytics
- sample\_geospatial
- \_MongoSH Beta

COVID.Cases

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION ¿Cuál es país con mayor número de casos? SAVE

\$sort

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

```

1
2  {
3     Confirmed: -1
4 }
```

\_id: ObjectId("5fc2b6948a49fb1de57b385b")
Date: "3/3/2020 12:00"
Province: "Hubei"
Region: "Mainland China"
Last Update: "2020-03-03T11:43:02"
Confirmed: 67217
Deaths: 2835
Recovered: 36208
Lat: 30.9756

\_id: ObjectId("5fc2b6948a49fb1de57b38f2")
Date: "3/2/2020 12:00"
Province: "Hubei"
Region: "Mainland China"
Last Update: "2020-03-02T15:03:23"
Confirmed: 67103
Deaths: 2803
Recovered: 33934
Lat: 30.9756

\$limit

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

```

1
2  1
```

\_id: ObjectId("5fc2b6948a49fb1de57b385b")
Date: "3/3/2020 12:00"
Province: "Hubei"
Region: "Mainland China"
Last Update: "2020-03-03T11:43:02"
Confirmed: 67217
Deaths: 2835
Recovered: 36208
Lat: 30.9756

- ¿Cuál es el país con mayor número de muertes?

[\$match:

{

Region: { \$not: /Null/ }

}}, {\$sort:

{

Deaths: -1

}}, {\$limit:

1}]

MongoDB Compass - 52.12.20.255:27017/COVID.Cases

Connect View Collection Help

Local

13 DBS 53 COLLECTIONS C

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

COVID

- Cases ...
- Cases2
- H1N1
- Starbucks
- admin
- config
- local
- sample\_airbnb
- sample\_analytics
- sample\_geospatial
- ... \_MongoSH Beta

COVID.Cases

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION ¿Cuál es país con mayor número de casos? SAVE

4709 Documents in the Collection

Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)

Preview of Documents in the Collection

`_id: ObjectId("5fc2b6948a49fb1de57b385b")  
Date: "3/3/2020 12:00"  
Province: "Hubei"  
Region: "Mainland China"  
Last Update: "2020-03-03T11:43:02"  
Confirmed: 67217  
Deaths: 2835  
Recovered: 36208  
Lat: 30.9756`

`_id: ObjectId("5fc2b6948a49fb1de57b385c")  
Date: "3/3/2020 12:00"  
Province: ""  
Region: "South Korea"  
Last Update: "2020-03-03T09:43:02"  
Confirmed: 5186  
Deaths: 28  
Recovered: 30  
Lat: 36`

\$match Output after \$match stage (Sample of 20 documents)

`1  
2 {  
3 Region: { $not: /Null/ }  
4 }`

`_id: ObjectId("5fc2b6948a49fb1de57b385b")  
Date: "3/3/2020 12:00"  
Province: "Hubei"  
Region: "Mainland China"  
Last Update: "2020-03-03T11:43:02"  
Confirmed: 67217  
Deaths: 2835  
Recovered: 36208  
Lat: 30.9756`

`_id: ObjectId("5fc2b6948a49fb1de57b385c")  
Date: "3/3/2020 12:00"  
Province: ""  
Region: "South Korea"  
Last Update: "2020-03-03T09:43:02"  
Confirmed: 5186  
Deaths: 28  
Recovered: 30  
Lat: 36`

MongoDB Compass - 52.12.20.255:27017/COVID.Cases

Connect View Collection Help

Local

13 DBS 53 COLLECTIONS C

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

COVID

- Cases ...
- Cases2
- H1N1
- Starbucks
- admin
- config
- local
- sample\_airbnb
- sample\_analytics
- sample\_geospatial
- ... \_MongoSH Beta

COVID.Cases

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION ¿Cuál es país con mayor número de casos? SAVE

4709 Documents in the Collection

Output after \$match stage (Sample of 20 documents)

`1  
2 {  
3 Deaths: -1  
4 }`

`_id: ObjectId("5fc2b6948a49fb1de57b385b")  
Date: "3/3/2020 12:00"  
Province: "Hubei"  
Region: "Mainland China"  
Last Update: "2020-03-03T11:43:02"  
Confirmed: 67217  
Deaths: 2835  
Recovered: 36208  
Lat: 30.9756`

`_id: ObjectId("5fc2b6948a49fb1de57b38f2")  
Date: "3/2/2020 12:00"  
Province: "Hubei"  
Region: "Mainland China"  
Last Update: "2020-03-02T15:03:23"  
Confirmed: 67103  
Deaths: 2803  
Recovered: 33934  
Lat: 30.9756`

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

`1  
2 1`

`_id: ObjectId("5fc2b6948a49fb1de57b385b")  
Date: "3/3/2020 12:00"  
Province: "Hubei"  
Region: "Mainland China"  
Last Update: "2020-03-03T11:43:02"  
Confirmed: 67217  
Deaths: 2835  
Recovered: 36208  
Lat: 30.9756`

- Usando las coordenadas, encuentra el epicentro del virus.

[{\$match:

{ Lat: { \$ne: "" },

Long: { \$ne: "" } }, {\$group:

```

{ _id: "$Region",
  Lat: { $max: "$Lat" },
  Long: { $max: "$Long" } }, {$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 Compass interface with the following details:

- Left Sidebar:** Shows the "Local" database with 13 DBs and 55 collections. The "COVID" collection is selected.
- Top Bar:** Shows the connection information "52.12.20.255:27017/COVID.Cases2".
- Central Area:**
  - Collection:** COVID.Cases2
  - Aggregations Tab:** Active tab.
  - Pipeline:**

```

1 { $match: { $ne: "" },
  2 { Lat: { $ne: "" },
  3 Long: { $ne: "" } }
}
        
```
  - Output:** Shows a sample of 20 documents from the \$match stage. One document is highlighted:
 

```

_id: ObjectId("5fc68ae12a0c87507996da40")
Date: "3/3/2020 12:00"
Province: "Hubei"
Region: "Mainland China"
Last Update: "2020-03-03T11:43:02"
Confirmed: "67217"
Deaths: "2835"
Recovered: "36288"
Lat: 30.9756
          
```
  - Summary:** DOCUMENTS 4.7k TOTAL SIZE 927.8KB AVG. SIZE 202B INDEXES 1 TOTAL SIZE 68.0KB AVG. SIZE 68.0KB

MongoDB Compass - 52.12.20.255:27017/COVID.Cases2

Connect View Collection Help

**Local**

13 DBS 55 COLLECTIONS C

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

COVID

- Cases
- Cases2
- ...
- Cases3
- H1N1
- Starbucks
- admin
- config
- local
- sample\_airbnb
- sample\_analytics
- 

> \_MongoSH Beta

**COVID.Cases2**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Usando las coordenadas, encuentra el epicentro del virus SAVE

\$group Output after \$group stage (Sample of 20 documents)

```

1
2 { "_id": "$Region",
3   Lat: { $max: "$Lat" },
4   Long: { $max: "$Long" } }

```

\_id: "Switzerland"  
Lat: 46.8182  
Long: 8.2275

\_id: "Croatia"  
Lat: 45.1667  
Long: 15.5

\$group Output after \$group stage (Sample of 1 document)

```

1
2 { "_id": null, size: { $sum: 1 },
3   sumLat: { $sum: "$Lat" },
4   sumLong: { $sum: "$Long" },
5   avgLat: { $avg: "$Lat" },
6   avgLong: { $avg: "$Long" }
7 }
8

```

\_id: null  
size: 84  
sumLat: 2611.9801  
sumLong: 3331.455  
avgLat: 31.095001190476193  
avgLong: 39.660178571428574

MongoDB Compass - 52.12.20.255:27017/COVID.Cases2

Connect View Collection Help

**Local**

13 DBS 55 COLLECTIONS C

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

COVID

- Cases
- Cases2
- ...
- Cases3
- H1N1
- Starbucks
- admin
- config
- local
- sample\_airbnb
- sample\_analytics
- 

> \_MongoSH Beta

**COVID.Cases2**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Usando las coordenadas, encuentra el epicentro del virus SAVE

\$group Output after \$group stage (Sample of 1 document)

```

1
2 { "_id": null,
3   sumLat: { $sum: "$Lat" },
4   sumLong: { $sum: "$Long" },
5   avgLat: { $avg: "$Lat" },
6   avgLong: { $avg: "$Long" }
7 }
8

```

\_id: null  
size: 84  
sumLat: 2611.9801  
sumLong: 3331.455  
avgLat: 31.095001190476193  
avgLong: 39.660178571428574

\$project Output after \$project stage (Sample of 1 document)

```

1
2 { $checkAvgLat: { $divide: [ "$sumLat", "$size" ] },
3   $checkAvgLong: { $divide: [ "$sumLong", "$size" ] } }

```

\_id: null  
checkAvgLat: 31.095001190476193  
checkAvgLong: 39.660178571428574

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

[{\$match:

{ Lat: { \$ne: "" },

Long: { \$ne: "" } }

```

}, {$group:

{ _id: "$Region",

Lat: { $max: "$Lat" },

Long: { $max: "$Long" } }}, {$addFields:

{ Lat: { $convert:

{ input: "$Lat", to: "double" } },

Long: { $convert:

{ input: "$Long", to: "double" } } }}, {$match:

{ $and: [ { Lat: { $gte: 21 } },

{ Lat: { $lte: 35 } },

{ Long: { $gte: 30 } },

{ Long: { $lte: 45 } } ] }}, {$limit:

5}, {$project:

{ _id: 1 }}]

```

MongoDB Compass - 52.12.20.255:27017/COVID.Cases2

Connect View Collection Help

Local

13 DBS 55 COLLECTIONS

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

COVID

Cases Cases2 ... Cases3

H1N1 Starbucks admin config local sample\_airbnb sample\_analytics

COVID.Cases2

Aggregations

4709 Documents in the Collection

Using the epicenter, find the 5 closest regions to this epicenter

DOCUMENTS 4.7k TOTAL SIZE 927.8KB AVG. SIZE 202B INDEXES 1 TOTAL SIZE 68.0KB AVG. SIZE 68.0KB

Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)

\$\_id: ObjectId("5fc68ae12a0c87507996da40")  
Date: "3/3/2020 12:00"  
Province: "Hubei"  
Region: "Mainland China"  
Last Update: "2020-03-03T11:43:02"  
Confirmed: "67217"  
Deaths: "2835"  
Recovered: "36288"

\$\_id: ObjectId("5fc68ae12a0c87507996da41")  
Date: "3/3/2020 12:00"  
Province: ""  
Region: "South Korea"  
Last Update: "2020-03-03T09:43:02"  
Confirmed: "5186"  
Deaths: "28"  
Recovered: "30"

\$\_id: ObjectId("5fc68ae12a0c87507996da40")  
Date: "3/3/2020 12:00"  
Province: "Hubei"  
Region: "Mainland China"  
Last Update: "2020-03-03T11:43:02"  
Confirmed: "67217"  
Deaths: "2835"  
Recovered: "36288"  
LAT: 30.4756

\$\_id: ObjectId("5fc68ae12a0c87507996da41")  
Date: "3/3/2020 12:00"  
Province: ""  
Region: "South Korea"  
Last Update: "2020-03-03T09:43:02"  
Confirmed: "5186"  
Deaths: "28"  
Recovered: "30"  
LAT: 35

\$match

Output after \$match stage (Sample of 20 documents)

\_MongoSH Beta

MongoDB Compass - 52.12.20.255:27017/COVID.Cases2

Connect View Collection Help

**Local**

- 13 DBS 55 COLLECTIONS C
- ☆ FAVORITE
- HOST 52.12.20.255:27017
- CLUSTER Standalone
- EDITION MongoDB 4.2.10 Community
- Filter your data
- COVID
- Cases
- Cases2 ...
- Cases3
- H1N1
- Starbucks
- admin
- config
- local
- sample\_airbnb
- sample\_analytics
- \_MongoSH Beta

**COVID.Cases2**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Usando el epicentro, encuentra las 5 regiones más cercanas a dicho epicentro SAVE SAMPLE MODE AUTO PREVIEW

**\$group** Output after \$group stage (Sample of 20 documents)

```

1
2  {
3     _id: "$Region",
4     Lat: { $max: "$Lat" },
5     Long: { $max: "$Long" }

```

**\$addFields** Output after \$addFields stage (Sample of 20 documents)

```

1
2  {
3     Lat: { $convert:
4         input: "$Lat",
5         to: "double"
6     },
7     Long: { $convert:
8         input: "$Long",
9         to: "double"
10    }

```

**Output**

|   |   |
|---|---|
| <code>_id: "Mainland China"</code><br>Lat: 47.862<br>Long: 127.7615 | <code>_id: "Vietnam"</code><br>Lat: 16<br>Long: 108         |
| <code>_id: "Netherlands"</code><br>Lat: 52.1326<br>Long: 5.2913     | <code>_id: "Algeria"</code><br>Lat: 28.0339<br>Long: 1.6596 |

MongoDB Compass - 52.12.20.255:27017/COVID.Cases2

Connect View Collection Help

**Local**

- 13 DBS 55 COLLECTIONS C
- ☆ FAVORITE
- HOST 52.12.20.255:27017
- CLUSTER Standalone
- EDITION MongoDB 4.2.10 Community
- Filter your data
- COVID
- Cases
- Cases2 ...
- Cases3
- H1N1
- Starbucks
- admin
- config
- local
- sample\_airbnb
- sample\_analytics
- \_MongoSH Beta

**COVID.Cases2**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Usando el epicentro, encuentra las 5 regiones más cercanas a dicho epicentro SAVE SAMPLE MODE AUTO PREVIEW

**\$match** Output after \$match stage (Sample of 6 documents)

```

1
2  {
3     $and: [
4         { Lat: { $gte: 21 } },
5         { Lat: { $lte: 35 } },
6         { Long: { $gte: 30 } },
7         { Long: { $lte: 45 } }
8     ]

```

**\$limit** Output after \$limit stage (Sample of 5 documents)

```

1
2  5

```

**Output**

|  |   |
|--|---|
| <code>_id: "Lebanon"</code><br>Lat: 33.8547<br>Long: 35.8623 | <code>_id: "Saudi Arabia"</code><br>Lat: 24<br>Long: 45 |
| <code>_id: "Lebanon"</code><br>Lat: 33.8547<br>Long: 35.8623 | <code>_id: "Saudi Arabia"</code><br>Lat: 24<br>Long: 45 |

MongoDB Compass - 52.12.20.255:27017/COVID.Cases2

Connect View Collection Help

Local

13 DBS 55 COLLECTIONS C

HOST 52.12.20.255:27017

CLUSTER Standalone

EDITION MongoDB 4.2.10 Community

Filter your data

COVID

Cases

Cases2 ...

Cases3

H1N1

Starbucks

admin

config

local

sample\_airbnb

sample\_analytics

MongoSH Beta

COVID.Cases2

Aggregations

Documents Aggregations Schema Explain Plan Indexes Validation

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

DOCUMENTS 4.7K TOTAL SIZE 927.8KB AVG. SIZE 202B INDEXES 1 TOTAL SIZE 68.0KB AVG. SIZE 68.0KB

Lat: 33.8547 Long: 35.8623

Lat: 24 Long: 45

\$project (Sample of 5 documents)

1  
2 { \_id: 1 }

\_id: "saudi Arabia"

\_id: "Lebanon"

ADD STAGE

The screenshot shows the MongoDB Compass interface with the 'COVID.Cases2' database selected. The left sidebar lists databases and collections, with 'COVID' and 'Cases2' expanded. The main area is titled 'COVID.Cases2' and has tabs for 'Documents', 'Aggregations' (which is selected), 'Schema', 'Explain Plan', 'Indexes', and 'Validation'. An aggregation pipeline is being built, starting with a '\$project' stage. The pipeline code is shown as: 1 and 2 { \_id: 1 }. Below the code, the output is displayed as a sample of 5 documents, showing two results: '\_id: "saudi Arabia"' and '\_id: "Lebanon"'. A button labeled 'ADD STAGE' is visible at the bottom of the pipeline editor. The top right corner shows document statistics: 4.7K DOCUMENTS, 927.8KB TOTAL SIZE, 202B AVG. SIZE, 1 INDEXES, 68.0KB TOTAL SIZE, and 68.0KB AVG. SIZE.

**Nota:** Ten en cuenta que puede haber países o regiones repetidas.