

SESION 1

PROYECTO

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

```
• show databases;  
• use classicmodels;
```

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

```
1 • show databases;  
2 • use classicmodels;  
3 • show tables;  
4 • describe employees;  
5 • select lastName from employees;
```

| | lastName |
|---|-----------|
| ▶ | Murphy |
| | Patterson |
| | Firrelli |
| | Patterson |
| | Bondur |
| | Bow |
| | Jennings |
| | Thompson |
| | Firrelli |

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

```
• select lastName, firstName, jobTitle from employees;
```

| | lastName | firstName | jobTitle |
|---|-----------|-----------|----------------------|
| ▶ | Murphy | Diane | President |
| | Patterson | Mary | VP Sales |
| | Firrelli | Jeff | VP Marketing |
| | Patterson | William | Sales Manager (APAC) |
| | Bondur | Gerard | Sale Manager (EMEA) |
| | Bow | Anthony | Sales Manager (NA) |
| | Jennings | Leslie | Sales Rep |
| | Thompson | Leslie | Sales Rep |
| | Firrelli | Julie | Sales Rep |

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

```
7 • select * from employees;
```

| employeeNumber | lastName | firstName | extension | email | officeCode | reportsTo | jobTitle |
|----------------|-----------|-----------|-----------|---------------------------------|------------|-----------|---------------|
| 1002 | Murphy | Diane | x5800 | dmurphy@classicmodelcars.com | 1 | 1002 | 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 |
| 1102 | Bondur | Gerard | x5408 | gbondur@classicmodelcars.com | 4 | 1056 | Sales Manager |
| 1143 | Bow | Anthony | x5428 | abow@classicmodelcars.com | 1 | 1056 | Sales Manager |
| 1165 | Jennings | Leslie | x3291 | ljennings@classicmodelcars.com | 1 | 1143 | Sales Rep |
| 1166 | Thompson | Leslie | x4065 | lthompson@classicmodelcars.com | 1 | 1143 | Sales Rep |

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

```
8 • select lastName, firstName, jobTitle from employees where jobTitle = 'Sales Rep';
```

| lastName | firstName | jobTitle |
|-----------|-----------|-----------|
| Jennings | Leslie | Sales Rep |
| Thompson | Leslie | Sales Rep |
| Firrelli | Julie | Sales Rep |
| Patterson | Steve | Sales Rep |
| Tseng | Foon Yue | Sales Rep |
| Vanauf | George | Sales Rep |
| Bondur | Loui | Sales Rep |
| Hernandez | Gerard | Sales Rep |
| Castillo | Pamela | Sales Rep |

6. 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.

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

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

7. 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.

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

| 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 |
| Thompson | Leslie | Sales Rep | 1 |
| Firrelli | Julie | Sales Rep | 2 |
| Patterson | Steve | Sales Rep | 2 |

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.

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

| lastName | firstName | officeCode |
|-----------|-----------|------------|
| Murphy | Diane | 1 |
| Patterson | Mary | 1 |
| Firrelli | Jeff | 1 |
| Bow | Anthony | 1 |
| Jennings | Leslie | 1 |
| Thompson | Leslie | 1 |
| Firrelli | Julie | 2 |
| Patterson | Steve | 2 |
| Tseng | Foon Yue | 3 |

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

```
11 • select lastName, firstName, jobTitle from employees where not jobTitle = 'Sales Rep';
```

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

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.

```
12 • select lastName, firstName, officeCode from employees where officeCode > 5;
```

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

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

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.

```
12 • select lastName, firstName, officeCode from employees where officeCode <= 4;
```

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

| | lastName | firstName | officeCode |
|---|-----------|-----------|------------|
| ▶ | Murphy | Diane | 1 |
| | Patterson | Mary | 1 |
| | Firrelli | Jeff | 1 |
| | Bondur | Gerard | 4 |
| | Bow | Anthony | 1 |
| | Jennings | Leslie | 1 |
| | Thompson | Leslie | 1 |
| | Firrelli | Julie | 2 |
| | Patterson | Steve | 2 |

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.

```
14 • select customerName, country, state from customers where country = 'USA' and state = 'CA';
```

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

| | customerName | country | state |
|---|------------------------------|---------|-------|
| ▶ | Mini Gifts Distributors Ltd. | USA | CA |
| | Mini Wheels Co. | USA | CA |
| | Technics Stores Inc. | USA | CA |
| | Toys4GrownUps.com | USA | CA |
| | Boards & Toys Co. | USA | CA |
| | Collectable Mini Designs Co. | USA | CA |
| | Corporate Gift Ideas Co. | USA | CA |
| | Men 'R' US Retailers, Ltd. | USA | CA |
| | The Sharp Gifts Warehouse | USA | CA |

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.

```
15 • ame, country, state, creditLimit from customers where country = 'USA' and state = 'CA' and creditLimit > 100000;
```

| 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 |

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

```
16 • select customerName, country from customers where country = 'USA' or country = 'France';
```

| customerName | country |
|------------------------------|---------|
| Atelier graphique | France |
| Signal Gift Stores | USA |
| La Rochelle Gifts | France |
| Mini Gifts Distributors Ltd. | USA |
| Mini Wheels Co. | USA |
| Land of Toys Inc. | USA |
| Saveley & Henriot, Co. | France |
| Musde Machine Inc | USA |
| Diecast Classics Inc. | USA |

15. Dentro de la tabla customers, obtén el nombre, pas 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.

```
17 • ame, country, creditLimit from customers where (country = 'USA' or country = 'France') and creditLimit > 100000;
```

| customerName | country | creditLimit |
|------------------------------|---------|-------------|
| La Rochelle Gifts | France | 118200.00 |
| Mini Gifts Distributors Ltd. | USA | 210500.00 |
| Land of Toys Inc. | USA | 114900.00 |
| Saveley & Henriot, Co. | France | 123900.00 |
| Musde Machine Inc | USA | 138500.00 |
| Diecast Classics Inc. | USA | 100600.00 |
| Collectable Mini Designs Co. | USA | 105000.00 |
| Marta's Replicas Co. | USA | 123700.00 |
| Mini Classics | USA | 102700.00 |

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.

```
19 • select officeCode, city, phone, country from offices where country = 'USA' or country = 'France';
```

| | 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 |

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.

```
19 • select officeCode, city, phone, country from offices where not (country = 'USA' or country = 'France');
```

| | 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 |

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.

```
21 • n, status, shippedDate from orders where orderNumber = 10165 or orderNumber = '10287' or orderNumber = '10310';
```

| | orderNumber | customerNumber | status | shippedDate |
|---|-------------|----------------|---------|-------------|
| ▶ | 10165 | 148 | Shipped | 2003-12-26 |
| | 10287 | 298 | Shipped | 2004-09-01 |
| | 10310 | 259 | Shipped | 2004-10-18 |

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

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

| | contactFirstName | contactLastName |
|---|------------------|-----------------|
| ▶ | Paolo | Accorti |
| | Raanan | Altagar, G M |
| | Mel | Andersen |
| | Carmen | Anton |
| | Rachel | Ashworth |
| | Miguel | Barajas |
| | Violeta | Benitez |
| | Helen | Bennett |
| | Christina | Berglund |

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

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

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

| | contactFirstName | contactLastName |
|---|------------------|-----------------|
| ▶ | Jeff | Young |
| | Julie | Young |
| | Mary | Young |
| | Dorothy | Young |
| | Juri | Yoshida |
| | Brydey | Walker |
| | Wendy | Victorino |
| | Braun | Urs |
| | Jerry | Tseng |

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.

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

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

| | contactFirstName | contactLastName |
|---|------------------|-----------------|
| ▶ | Adrian | Huxley |
| | Akiko | Shimamura |
| | Alejandra | Camino |
| | Alexander | Feuer |
| | Alexander | Semenov |
| | Allen | Nelson |
| | Ann | Brown |
| | Anna | O'Hara |
| | Annette | Roulet |

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).

26 • `select customerNumber, contactFirstName, creditLimit from customers order by creditLimit desc limit 5;`

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

| | 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 | NULL |

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.

```
26 • select customerNumber, contactFirstName, creditLimit from customers order by creditLimit limit 5;
```

27

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

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

SESION 2

PROYECTO

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

```
5 • select employeeNumber, firstName, lastName from employees where firstName like 'A%';
```

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

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

```
6 • select employeeNumber, lastName, firstName from employees where firstName like '%n';
```

| employeeNumber | lastName | firstName |
|----------------|----------|-----------|
| 1702 | Gerard | Martin |
| NULL | NULL | NULL |

3. Dentro de la tabla `employees`, obten 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%';
```

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

4. Dentro de la tabla `employees`, obten 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';
```

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

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

```
select employeeNumber, lastName, firstName from employees where not firstName like 'B%';
```

| employeeNumber | lastName | firstName |
|----------------|-----------|-----------|
| 1002 | Murphy | Diane |
| 1056 | Patterson | Mary |
| 1076 | Firrelli | Jeff |
| 1088 | Patterson | William |
| 1102 | Bondur | Gerard |
| 1143 | Bow | Anthony |
| 1165 | Jennings | Leslie |
| 1166 | Thompson | Leslie |
| 1188 | Firrelli | Julie |

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

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

| productCode | productName |
|-------------|-------------------------------------------|
| S10_2016 | 1996 Moto Guzzi 1100i |
| S24_2000 | 1960 BSA Gold Star DBD34 |
| S24_2011 | 18th century schooner |
| S24_2022 | 1938 Cadillac V-16 Presidential Limousine |
| S700_2047 | HMS Bounty |
| NULL | NULL |

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

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

| orderNumber | Ordenes |
|-------------|---------|
| 10106 | 18 |
| 10159 | 18 |
| 10165 | 18 |
| 10168 | 18 |

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

```
select year(orderDate), count(*) as Total from orders group by year(orderDate);
```

| year(orderDate) | Total |
|-----------------|-------|
| 2003 | 111 |
| 2004 | 151 |
| 2005 | 64 |

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

```
select * from employees where officeCode in(select officeCode from offices where country = 'USA');
```

| 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 |
| 1143 | Bow | Anthony | x5428 | abow@classicmodelcars.com | 1 | 1056 | Sales Manager (NA) |
| 1165 | Jennings | Leslie | x3291 | ljennings@classicmodelcars.com | 1 | 1143 | Sales Rep |
| 1166 | Thompson | Leslie | x4065 | lthompson@classicmodelcars.com | 1 | 1143 | Sales Rep |
| 1188 | Firrelli | Julie | x2173 | jfirrelli@classicmodelcars.com | 2 | 1143 | Sales Rep |
| 1216 | Patterson | Steve | x4334 | spatterson@classicmodelcars.com | 2 | 1143 | Sales Rep |
| 1286 | Tseng | Foon Yue | x2248 | ftseng@classicmodelcars.com | 3 | 1143 | Sales Rep |

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

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

| customerNumber | checkNumber | Maximo |
|----------------|-------------|-----------|
| 141 | JE105477 | 120166.58 |

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

```
select avg(amount) from payments;  
select customerNumber, checkNumber, amount from payments where amount > 32431.645531;
```

| customerNumber | checkNumber | amount |
|----------------|-------------|----------|
| 112 | HQ55022 | 32641.98 |
| 112 | ND748579 | 33347.88 |
| 114 | GG31455 | 45864.03 |
| 114 | MA765515 | 82261.22 |
| 114 | NR27552 | 44894.74 |
| 119 | LN373447 | 47924.19 |
| 119 | NG94694 | 49523.67 |
| 121 | DB889831 | 50218.95 |
| 121 | MA302151 | 34638.14 |

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

```
select customerName from customers where customerNumber  
in(select customerNumber from orders where status = 'Cancelled');
```

| customerName |
|-------------------------|
| Scandinavian Gift Ideas |
| Kelly's Gift Shop |
| Land of Toys Inc. |
| UK Collectables, Ltd. |
| GiftsForHim.com |
| Euro+ Shopping Channel |

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

```
select max(quantityOrdered) as maximo, min(quantityOrdered) as minimo, avg(quantityOrdered) as promedio  
from orderdetails;
```

| maximo | minimo | promedio |
|--------|--------|----------|
| 97 | 6 | 35.2190 |

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

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

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

SESION 3

PROYECTO

Para estas consultas usa *RIGHT JOIN*

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

```
select productCode, productName, productDescription from products;
```

| productCode | productName | productDescription |
|-------------|---------------------------------------|------------------------------------------------------|
| S10_1678 | 1969 Harley Davidson Ultimate Chopper | This replica features working kickstand, front su... |
| S10_1949 | 1952 Alpine Renault 1300 | Turnable front wheels; steering function; detail... |
| S10_2016 | 1996 Moto Guzzi 1100i | Official Moto Guzzi logos and insignias, saddle b... |
| S10_4698 | 2003 Harley-Davidson Eagle Drag Bike | Model features, official Harley Davidson logos a... |
| S10_4757 | 1972 Alfa Romeo GTA | Features include: Turnable front wheels; steeri... |
| S10_4962 | 1962 LanciaA Delta 16V | Features include: Turnable front wheels; steeri... |
| S12_1099 | 1968 Ford Mustang | Hood, doors and trunk all open to reveal highly ... |
| S12_1108 | 2001 Ferrari Enzo | Turnable front wheels; steering function; detail... |
| S12_1666 | 1958 Setra Bus | Model features 30 windows, skylights & glare re... |

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

```
select o.orderNumber Orden, o.status Estado, sum(p.amount) Cantidad from orders o right join payments p  
on p.customerNumber = o.customerNumber group by Orden;
```

| Orden | Estado | Cantidad |
|-------|---------|-----------|
| 10123 | Shipped | 22314.36 |
| 10298 | Shipped | 22314.36 |
| 10345 | Shipped | 22314.36 |
| 10124 | Shipped | 80180.98 |
| 10278 | Shipped | 80180.98 |
| 10346 | Shipped | 80180.98 |
| 10120 | Shipped | 180585.07 |
| 10125 | Shipped | 180585.07 |
| 10223 | Shipped | 180585.07 |
| 10342 | Shipped | 180585.07 |

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

```
select o.orderNumber Orden, o.orderDate Fecha, od.orderLineNumber Linea,  
       p.productName Nombre, od.quantityOrdered Cantidad, od.priceEach Precio  
from orders o right join orderdetails od on o.orderNumber = od.orderNumber  
right join products p on od.productCode = p.productCode order by Linea;
```

| Orden | Fecha | Linea | Nombre | Cantidad | Precio |
|-------|------------|-------|---------------------------------------|----------|--------|
| NULL | NULL | NULL | 1985 Toyota Supra | NULL | NULL |
| 10208 | 2004-01-02 | 1 | The USS Constitution Ship | 46 | 63.61 |
| 10206 | 2003-12-05 | 1 | 1982 Camaro Z28 | 33 | 89.01 |
| 10168 | 2003-10-28 | 1 | 1969 Harley Davidson Ultimate Chopper | 36 | 94.74 |
| 10188 | 2003-11-18 | 1 | 1969 Harley Davidson Ultimate Chopper | 48 | 95.70 |
| 10103 | 2003-01-29 | 1 | 1962 Volkswagen Microbus | 36 | 107.34 |
| 10223 | 2004-02-20 | 1 | 1969 Harley Davidson Ultimate Chopper | 37 | 80.39 |
| 10354 | 2004-12-04 | 1 | 1960 BSA Gold Star DBD34 | 28 | 62.46 |
| 10275 | 2004-07-23 | 1 | 1969 Harley Davidson Ultimate Chopper | 45 | 81.35 |
| 10152 | 2003-09-25 | 1 | 1970 Triumph Spitfire | 35 | 117.77 |

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

```
select od.orderNumber Orden, p.productName Producto, p.MSRP MSRP, p.buyPrice
from products p right join orderdetails od on p.productCode = od.productCode order by Orden;
```

| Orden | Producto | MSRP | buyPrice |
|-------|-------------------------------------------|--------|----------|
| 10100 | 1917 Grand Touring Sedan | 170.00 | 86.70 |
| 10100 | 1911 Ford Town Car | 60.54 | 33.30 |
| 10100 | 1932 Alfa Romeo 8C2300 Spider Sport | 92.03 | 43.26 |
| 10100 | 1936 Mercedes Benz 500k Roadster | 41.03 | 21.75 |
| 10101 | 1932 Model A Ford J-Coupe | 127.13 | 58.48 |
| 10101 | 1928 Mercedes-Benz SSK | 168.75 | 72.56 |
| 10101 | 1939 Chevrolet Deluxe Coupe | 33.19 | 22.57 |
| 10101 | 1938 Cadillac V-16 Presidential Limousine | 44.80 | 20.61 |
| 10102 | 1937 Lincoln Berline | 102.74 | 60.62 |
| 10102 | 1936 Mercedes-Benz 500K Special Roadster | 53.91 | 24.26 |

Para estas consultas usa LEFT JOIN

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

```
select c.customerNumber No_Cliente, c.customerName Cliente, o.orderNumber No_Orden, c.state Estado
from customers c left join orders o on c.customerNumber = o.customerNumber;
```

| No_Cliente | Cliente | No_Orden | Estado |
|------------|----------------------------|----------|----------|
| 103 | Atelier graphique | 10123 | NULL |
| 103 | Atelier graphique | 10298 | NULL |
| 103 | Atelier graphique | 10345 | NULL |
| 112 | Signal Gift Stores | 10124 | NV |
| 112 | Signal Gift Stores | 10278 | NV |
| 112 | Signal Gift Stores | 10346 | NV |
| 114 | Australian Collectors, Co. | 10120 | Victoria |
| 114 | Australian Collectors, Co. | 10125 | Victoria |
| 114 | Australian Collectors, Co. | 10223 | Victoria |
| 114 | Australian Collectors, Co. | 10342 | Victoria |

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

```
select c.customerNumber No_Cliente, c.customerName Cliente, o.orderNumber No_Orden, c.state Estado
from customers c left join orders o on c.customerNumber = o.customerNumber
where isnull(o.orderNumber);
```

| No_Cliente | Cliente | No_Orden | Estado |
|------------|----------------------------|----------|--------|
| 125 | Havel & Zbyszek Co | NULL | NULL |
| 168 | American Souvenirs Inc | NULL | CT |
| 169 | Porto Imports Co. | NULL | NULL |
| 206 | Asian Shopping Network, Co | NULL | NULL |
| 223 | Natürlich Autos | NULL | NULL |
| 237 | ANG Resellers | NULL | NULL |
| 247 | Messner Shopping Network | NULL | NULL |
| 273 | Franken Gifts, Co | NULL | NULL |
| 293 | BG&E Collectables | NULL | NULL |
| 303 | Schuvler Imports | NULL | NULL |

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 concat(e.lastName, ' ', e.firstName) Empleado, c.customerName Cliente, p.checkNumber Cheque
from employees e left join customers c on e.reportsTo = c.salesRepEmployeeNumber
left join payments p on c.customerNumber = p.customerNumber order by Cheque desc;
```

| Empleado | Cliente | Cheque |
|--------------|-------------------------|----------|
| Kato Yoshimi | Dragon Souvenirs, Ltd. | ME497970 |
| Kato Yoshimi | Dragon Souvenirs, Ltd. | KM172879 |
| Kato Yoshimi | Tokyo Collectables, Ltd | KB54275 |
| Kato Yoshimi | Tokyo Collectables, Ltd | JPMR4544 |
| Kato Yoshimi | Cruz & Sons Co. | EK785462 |
| Kato Yoshimi | Tokyo Collectables, Ltd | DO787644 |
| Kato Yoshimi | Dragon Souvenirs, Ltd. | DD635282 |
| Kato Yoshimi | Cruz & Sons Co. | CP804873 |
| Kato Yoshimi | Osaka Souvenirs Co. | CI381435 |
| Kato Yoshimi | Cruz & Sons Co. | BN347084 |

Para estas consultas usa *RIGHT JOIN*

8. Repite los ejercicios 5 a 7 usando *RIGHT JOIN*.

```
select c.customerNumber No_Cliente, c.customerName Cliente, o.orderNumber No_Orden, c.state Estado
from customers c right join orders o on c.customerNumber = o.customerNumber;
```

| No_Cliente | Cliente | No_Orden | Estado |
|------------|----------------------------|----------|----------|
| 103 | Atelier graphique | 10123 | NULL |
| 103 | Atelier graphique | 10298 | NULL |
| 103 | Atelier graphique | 10345 | NULL |
| 112 | Signal Gift Stores | 10124 | NV |
| 112 | Signal Gift Stores | 10278 | NV |
| 112 | Signal Gift Stores | 10346 | NV |
| 114 | Australian Collectors, Co. | 10120 | Victoria |
| 114 | Australian Collectors, Co. | 10125 | Victoria |
| 114 | Australian Collectors, Co. | 10223 | Victoria |
| 114 | Australian Collectors, Co. | 10342 | Victoria |

```
select concat(e.lastName, ' ', e.firstName) Empleado, c.customerName Cliente, p.checkNumber Cheque
from employees e right join customers c on e.reportsTo = c.salesRepEmployeeNumber
right join payments p on c.customerNumber = p.customerNumber order by Cheque desc;
```

| Empleado | Cliente | Cheque |
|----------|--------------------------------|----------|
| NULL | Signal Collectibles Ltd. | PT550181 |
| NULL | Down Under Souvenirs, Inc | PQ803830 |
| NULL | Double Decker Gift Stores, Ltd | PO860906 |
| NULL | Online Diecast Creations Co. | PN238558 |
| NULL | Extreme Desk Decorations, Ltd | PJ434867 |
| NULL | Mini Wheels Co. | PI42991 |
| NULL | Gift Depot Inc. | PI15215 |
| NULL | Mini Caravy | PH785937 |
| NULL | Salzburg Collectables | PH29054 |
| NULL | FunGiftIdeas.com | PE176846 |

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

```
1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`%`
4     SQL SECURITY DEFINER
5     VIEW `RCA_orden` AS
6     SELECT
7         `o`.`orderNumber` AS `Orden`,
8         `o`.`status` AS `Estado`,
9         SUM(`p`.`amount`) AS `Cantidad`
10    FROM
11        (`payments` `p`
12     LEFT JOIN `orders` `o` ON ((`p`.`customerNumber` = `o`.`customerNumber`)))
13    GROUP BY `Orden`
```

```
34 • select * from RCA_orden;
```

| Result Grid | | | |
|--------------|-------|---------|-----------|
| Filter Rows: | | | |
| | Orden | Estado | Cantidad |
| ▶ | 10123 | Shipped | 22314.36 |
| | 10298 | Shipped | 22314.36 |
| | 10345 | Shipped | 22314.36 |
| | 10124 | Shipped | 80180.98 |
| | 10278 | Shipped | 80180.98 |
| | 10346 | Shipped | 80180.98 |
| | 10120 | Shipped | 180585.07 |
| | 10125 | Shipped | 180585.07 |
| | 10223 | Shipped | 180585.07 |
| | 10342 | Shipped | 180585.07 |

```
1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`%`
4     SQL SECURITY DEFINER
5     VIEW `RCA_detalle_orden` AS
6     SELECT
7         `o`.`orderNumber` AS `Orden`,
8         `o`.`orderDate` AS `Fecha`,
9         `od`.`orderLineNumber` AS `Linea`,
10        `p`.`productName` AS `Nombre`,
11        `od`.`quantityOrdered` AS `Cantidad`,
12        `od`.`priceEach` AS `Precio`
```

```

13 FROM
14     (`products` `p`
15     LEFT JOIN (`orderdetails` `od`
16     LEFT JOIN `orders` `o` ON ((`o`.`orderNumber` = `od`.`orderNumber`))) ON ((`od`.`productCode`
17     ORDER BY `od`.`orderLineNumber`

```

35 • `select * from RCA_detalle_orden;`



| Result Grid | | | | | | |
|------------------------------------|-------|------------|-------|---------------------------------------|----------|--------|
| Filter Rows: <input type="text"/> | | | | | | |
| Export: Wrap Cell Content: Fetch | | | | | | |
| | Orden | Fecha | Linea | Nombre | Cantidad | Precio |
| ▶ | NULL | NULL | NULL | 1985 Toyota Supra | NULL | NULL |
| | 10208 | 2004-01-02 | 1 | The USS Constitution Ship | 46 | 63.61 |
| | 10206 | 2003-12-05 | 1 | 1982 Camaro Z28 | 33 | 89.01 |
| | 10168 | 2003-10-28 | 1 | 1969 Harley Davidson Ultimate Chopper | 36 | 94.74 |
| | 10188 | 2003-11-18 | 1 | 1969 Harley Davidson Ultimate Chopper | 48 | 95.70 |
| | 10103 | 2003-01-29 | 1 | 1962 Volkswagen Microbus | 36 | 107.34 |
| | 10223 | 2004-02-20 | 1 | 1969 Harley Davidson Ultimate Chopper | 37 | 80.39 |
| | 10354 | 2004-12-04 | 1 | 1960 BSA Gold Star DBD34 | 28 | 62.46 |
| | 10275 | 2004-07-23 | 1 | 1969 Harley Davidson Ultimate Chopper | 45 | 81.35 |
| | 10152 | 2003-09-25 | 1 | 1970 Triumph Spitfire | 35 | 117.77 |

```

1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`%`
4     SQL SECURITY DEFINER
5     VIEW `RCA_producto` AS
6     SELECT
7         `od`.`orderNumber` AS `Orden`,
8         `p`.`productName` AS `Producto`,
9         `p`.`MSRP` AS `MSRP`,
10        `p`.`buyPrice` AS `buyPrice`
11     FROM
12         (`orderdetails` `od`
13         LEFT JOIN `products` `p` ON ((`p`.`productCode` = `od`.`productCode`)))
14     ORDER BY `od`.`orderNumber`

```

36 • `select * from RCA_producto;`

| Result Grid | | | | |
|-------------|-------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------|-----------------------------------------------------------------------------------------------------|
| | |  | Filter Rows: <input type="text"/> | Export:  Wrap Cell |
| Orden | Producto | MSRP | buyPrice | |
| 10100 | 1917 Grand Touring Sedan | 170.00 | 86.70 | |
| 10100 | 1911 Ford Town Car | 60.54 | 33.30 | |
| 10100 | 1932 Alfa Romeo 8C2300 Spider Sport | 92.03 | 43.26 | |
| 10100 | 1936 Mercedes Benz 500k Roadster | 41.03 | 21.75 | |
| 10101 | 1932 Model A Ford J-Coupe | 127.13 | 58.48 | |
| 10101 | 1928 Mercedes-Benz SSK | 168.75 | 72.56 | |
| 10101 | 1939 Chevrolet Deluxe Coupe | 33.19 | 22.57 | |
| 10101 | 1938 Cadillac V-16 Presidential Limousine | 44.80 | 20.61 | |
| 10102 | 1937 Lincoln Berline | 102.74 | 60.62 | |
| 10102 | 1936 Mercedes-Benz 500K Special Roadster | 53.91 | 24.26 | |

SESION 4

PROYECTO

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

PS4-1

PROJECT

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

FILTER

PROJECT {name:1, email_address:1, phone_number:1}

SORT

COLLATION



VIEW



```
_id: ObjectId("52cdef7c4bab8bd675297d8a")
name: "Wetpaint"
email_address: "info@wetpaint.com"
phone_number: "206.859.6300"
```

```
_id: ObjectId("52cdef7c4bab8bd675297d8e")
name: "Facebook"
email_address: ""
phone_number: ""
```

```
_id: ObjectId("52cdef7c4bab8bd675297d8f")
name: "Omnidrive"
email_address: "info@omnidrive.com"
phone_number: "660-675-5052"
```

2.- Obtén la fuente de cada tweet.

PS4-2

PROJECT

```
{
  source: 1
}
```

PROJECT `{source:1}`




SORT

MAXTIMEMS 5000

COLLATION

SKIP 0

LIMIT 0

VIEW   

Displaying documents 1 - 20 of 24

```

_id: ObjectId("5c8eccb0caa187d17ca623f5")
source: "web"

_id: ObjectId("5c8eccb0caa187d17ca623f7")
source: "<a href='http://www.tweetdeck.com' rel='nofollow'>TweetDeck</a>"

_id: ObjectId("5c8eccb0caa187d17ca623fa")
source: "<a href='http://blackberry.com/twitter' rel='nofollow'>Twitter for Bla..."

_id: ObjectId("5c8eccb0caa187d17ca623fc")
source: "<a href='http://www.echofon.com/' rel='nofollow'>Echofon</a>"

_id: ObjectId("5c8eccb0caa187d17ca623fe")
source: "<a href='http://83degrees.com/to/powertwitter' rel='nofollow'>Power Tw..."

```

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

PS4-3

FILTER

```
{
  founded_month: 10
}
```

PROJECT

```
{
  name: 1,
  founded_month: 1
}
```


`_id: ObjectId("52cdef7c4bab8bd675297da8")`
`name: "OpenX"`
`founded_year: 2008`

`_id: ObjectId("52cdef7c4bab8bd675298218")`
`name: "WonderHowTo"`
`founded_year: 2008`

`_id: ObjectId("52cdef7c4bab8bd675298232")`
`name: "First30Days"`
`founded_year: 2008`

`_id: ObjectId("52cdef7c4bab8bd675298244")`
`name: "Mibura"`
`founded_year: 2008`

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

PS4-5

PROJECT

```
{
  author: 1,
  body: 1
}
```

FILTER

{author: Machine}

PROJECT

{author:1, body:1}

SORT

MAXTIMEMS

5000

COLLATION

SKIP

0

LIMIT

0

VIEW

Displaying documents 1 - 20 of 20

_id: ObjectId("50ab0f8bbcf1bfe2536dc3f9")
body: "Amendment I
<p>Congress shall make no law respecting an establishment ..."
author: "machine"

_id: ObjectId("50ab0f8bbcf1bfe2536dc3fa")
body: "We the People of the United States, in Order to form a more perfect Un..."
author: "machine"

_id: ObjectId("50ab0f8bbcf1bfe2536dc3fb")
body: "Four score and seven years ago our fathers brought forth on this conti..."
author: "machine"

_id: ObjectId("50ab0f8bbcf1bfe2536dc401")
body: "We the People of the United States, in Order to form a more perfect Un..."
author: "machine"

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

PS4-6

FILTER

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

PROJECT

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

| | | |
|------------------|--------------------|---------------------|
| FILTER | {source: "web"} | OPTIONS |
| PROJECT | {text:1, source:1} | |
| SORT | | MAXMEMS 5000 |
| COLLATION | SKIP 0 | LIMIT 0 |

```
_id: ObjectId("5c8eccb0caa187d17ca623f5")
text: "eu preciso de terminar de fazer a minha tabela, está muito foda ***"
source: "web"
```

```
_id: ObjectId("5c8eccb0caa187d17ca623ff")
text: "First week of school is over :P"
source: "web"
```

```
_id: ObjectId("5c8eccb0caa187d17ca62400")
text: "fair today!!!! then jersey shore!!!=D"
source: "web"
```

```
_id: ObjectId("5c8eccb0caa187d17ca62404")
text: "@teetoolegit lmfao!! No BS! hahaha"
source: "web"
```

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

PS4-7

FILTER

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

PROJECT

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

| | | |
|------------------|-------------------------------------------|---------------------|
| FILTER | {founded_year: 2008, founded_month: 10} | OPTIONS |
| PROJECT | {name:1, founded_year:1, founded_month:1} | |
| SORT | | MAXTIME 5000 |
| COLLATION | SKIP 0 | LIMIT 0 |

```
_id: ObjectId("52cdef7c4bab8bd6752985ca")
name: "tunes8ag"
founded_year: 2008
founded_month: 10
```

```
_id: ObjectId("52cdef7d4bab8bd675298d21")
name: "Muecs"
founded_year: 2008
founded_month: 10
```

```
_id: ObjectId("52cdef7d4bab8bd675299a43")
name: "Rush Hour"
founded_year: 2008
founded_month: 10
```

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

PS4-8

FILTER

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

PROJECT

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

FILTER {number_of_employees: {\$gte: 50}}

▼ OPTIONS

PROJECT {name:1, number_of_employees:1}

SORT

MAXTIME 5000

COLLATION

SKIP 0

LIMIT 0

```
_id: ObjectId("52cdef7c4bab8bd675297d8e")
name: "Facebook"
number_of_employees: 5299
```

```
_id: ObjectId("52cdef7c4bab8bd675297d94")
name: "Twitter"
number_of_employees: 1300
```

```
_id: ObjectId("52cdef7c4bab8bd675297d97")
name: "Scribd"
number_of_employees: 50
```

```
_id: ObjectId("52cdef7c4bab8bd675297da1")
name: "Plaxo"
number_of_employees: 50
```

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

PS4-9

FILTER

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

PROJECT

```
{
  href: 1,
  title: 1,
  comments: 1,
  description: 1,
  link: 1
}
```

| | | |
|------------------|-------------------------------------------------------------------------------------|----------------|
| FILTER | <code>{ \$and: [{ comments: { \$gte: 10 } }, { comments: { \$lte: 30 } }] }</code> | OPTIONS |
| PROJECT | <code>{ href: 1, title: 1, comments: 1, description: 1, link: 1 }</code> | |
| SORT | MAXTIME 5000 | |
| COLLATION | SKIP 0 | LIMIT 0 |

```
_id: ObjectId("4ba267dc238d3ba3ca000006")
href: "http://digg.com/travel_places/11_Amazing_Treehouses_from_Around_the_Wo..."
title: "11 Amazing Treehouses from Around the World"
comments: 15
description: "Treehouses bring us closer to nature, and appeal to the kid in all of ..."
link: "http://www.thedailygreen.com/green-homes/latest/treehouse-photos-46031..."
```

```
_id: ObjectId("4ba267dc238d3ba3ca00000b")
href: "http://digg.com/space/NASA_The_Wizard_Nebula"
title: "NASA - The Wizard Nebula "
comments: 14
description: "This image of the open star cluster NGC 7380, also known as the Wizard..."
link: "http://www.nasa.gov/multimedia/imagegallery/image_feature_1615.html"
```

```
_id: ObjectId("4ba267dc238d3ba3ca00000c")
href: "http://digg.com/space/WISE_Captures_a_Cosmic_Rose"
title: "WISE Captures a Cosmic Rose"
comments: 12
description: "A new infrared image from NASA's Wide-field Infrared Survey Explorer, ..."
link: "http://www.physorg.com/news187977528.html"
```

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

PS4-10

FILTER

```
{
  $and: [
    {
      number_of_employees: {
        $ne: null
      }
    },
    {
      number_of_employees: {
        $ne: 0
      }
    }
  ]
}
```

PROJECT

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

LIMIT

```
1
```


12.- Obtén la historia más comentada.

PS4-12

PROJECT

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

SORT

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

LIMIT

```
1
```

| | | | |
|------------------|----------------------|----------------|------|
| FILTER | OPTIONS | | |
| PROJECT | {comments:1,title:1} | | |
| SORT | {comments:-1} | MAXTIME | 5000 |
| COLLATION | SKIP 0 | LIMIT | 1 |

```
_id: ObjectId("4ba27ea0238d3ba3ca002251")
title: "Republican Brown wins Massachusetts Senate seat!"
comments: 1864
```

13.- Obtén la historia menos comentada.

PS4-13

PROJECT

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

SORT

```
{
  comments: 1
}
```

LIMIT

```
1
```

| | | | | |
|-----------|----------------------|---------|-------|---|
| FILTER | | OPTIONS | | |
| PROJECT | {comments:1,title:1} | | | |
| SORT | {comments:1} | MAXTIME | 5000 | |
| COLLATION | SKIP | 0 | LIMIT | 1 |

```
_id: ObjectId("4ba27e1a238d3ba3ca002161")
title: "UA Tech Park chosen for $32 million 'Solar Zone' project"
comments: 0
```


SESION 5

PROYECTO

1. 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.

The screenshot displays two stages of a MongoDB pipeline in the Atlas console. Each stage has a query editor on the left and a preview of the output on the right.

Stage 1:

- Query:**

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   amenities:{$in:['Ethernet/']}
6 }
```
- Output:** A sample document showing fields like `last_scraped`, `calendar_last_scraped`, `first_review`, `last_review`, `accommodates`, `bedrooms`, and `beds`.

Stage 2:

- Query:**

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   "review_scores.review_scores_rating":{$gte:80}
6 }
```
- Output:** A sample document showing fields like `property_type`, `room_type`, `bed_type`, `minimum_nights`, `maximum_nights`, `cancellation_policy`, `last_scraped`, `calendar_last_scraped`, `first_review`, and `last_review`.

▼

\$match

▼

🗑

+

1 ▾ /**

2 * query: The query in MQL.

3 */

4 ▾ {

5 number_of_reviews:{\$gt:50}

6 }

Output after [\\$match](#) stage ⓘ (Sample of 20 documents)

_id: "1001265"

listing_url: "https://www.airbnb.com/rooms/1001265"

name: "Ocean View Waikiki Marina w/prkg"

summary: "A short distance from Honolulu's billion doll mall, and the same dis..."

space: "Great studio located on Ala Moana across the street from Yacht Harbor ..."

description: "A short distance from Honolulu's billion mall, and the same dis..."

▼

\$match

▼

🗑

+

1 ▾ /**

2 * query: The query in MQL.

3 */

4 ▾ {

5 "address.country_code": "BR"

6 }

Output after [\\$match](#) stage ⓘ (Sample of 6 documents)

space: "It is a bedroom and living room, charming, private, finely decorated, ..."

description: "It is a bedroom and living room, charming private, finely decorated, ..."

neighborhood_overview: "There are a lot of bars, restaura supermarkets, bakery, banks... I

notes: "1) Enjoy the apartment 2) Enjoy Rio 3) Come bac soon"

transit: "Buses, taxis, metro and bicycle, that you can rent near to the buildin..."

SESION 6

PROYECTO

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:

| pais | genero | peliculas |
|------|--------|-----------|
| USA | Short | 10 |
| USA | Drama | 20 |
| ... | ... | ... |

\$unwind

1 {

2 path: '\$genres',

3 preserveNullAndEmptyArrays: false

4 }

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

_id: ObjectId("573a1390f29313caabcd4135")

plot: "Three men hammer on an anvil and pass a bottle o

beer around."

genres: "Short"

runtime: 1

▶ cast: Array

num_mflix_comments: 1

title: "Blacksmith Scene"

fullplot: "A stationary camera looks at a large anvil w

\$unwind

1 {

2 path: '\$countries',

3 preserveNullAndEmptyArrays: false

4 }

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

_id: ObjectId("573a1390f29313caabcd4135")

plot: "Three men hammer on an anvil and pass a bottle o

beer around."

genres: "Short"

runtime: 1

▶ cast: Array

num_mflix_comments: 1

title: "Blacksmith Scene"

fullplot: "A stationary camera looks at a large anvil w

```

1 {
2   _id: '$countries',
3   total: {
4     $push: {
5       genero: '$genres',
6       titulo: '$title'
7     }
8   }
9 }

```

```

_id: "Nepal"
total: Array

```

||| ▼ \$unwind 🔴

🗑️ +

Output after [\\$unwind](#) stage ⓘ (Sample of 20 documents)

```

1 {
2   path: '$total',
3   preserveNullAndEmptyArrays: false
4 }

```

```

_id: "Canada"
total: Object

```

||| ▼ \$addFields 🔴

🗑️ +

Output after [\\$addFields](#) stage ⓘ (Sample of 20 documents)

```

1 {
2   genero: '$total.genero'
3 }

```

```

_id: "Greece"
total: Object
genero: "Drama"

```

||| ▼ \$group 🔴

🗑️ +

Output after [\\$group](#) stage ⓘ (Sample of 20 documents)

```

1 {
2   _id: {
3     pais: '$_id',
4     genero: '$genero'
5   },
6   suma: {
7     $sum: 1
8   }
9 }

```

```

_id: Object
suma: 2

```


SESION 7

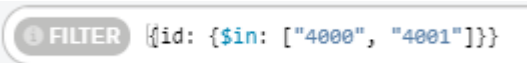
PROYECTO

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

Agregar los siguientes registros en formato CSV a la Colección movies

4000,Avengers: Endgame (2019),Fantasy|Sci-Fi

4001,Glass (2019),Drama|Fantasy

 FILTER `{id: {$in: ["4000", "4001"]}}`

```
_id: ObjectId("5f1f25d8549a9edc90f26dd2")
id: "4000"
titulo: "Avengers: Endgame (2019)"
genres: "Fantasy|Sci-Fi"
```

```
_id: ObjectId("5f1f26b5549a9edc90f26dd3")
id: "4001"
titulo: "Glass (2019)"
genres: "Drama|Fantasy"
```

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

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

```
_id: ObjectId("5f1f26b5549a9edc90f26dd3")
id: "4001"
titulo: "Glass (2019)"
genres: "Drama|Fantasy"
✓ valoraciones: Array
  ✓ 0: Object
    userid: "1563"
    movieid: "4001"
    rating: "4"
  ✓ 1: Object
    userid: "434"
    movieid: "4001"
    rating: "5"
```

RETOS

SESION 1

RETO 1

Usando la base de datos tienda, muestra la descripción de las tablas articulo, puesto y venta. Por cada tipo de dato que encuentras llena la siguiente tabla, a mano. Usa la Documentación de MySQL como referencia.

4 • `describe articulo;`

| Field | Type | Null | Key | Default | Extra |
|-------------|-------------|------|-----|---------|-------|
| id_articulo | int | NO | PRI | NULL | |
| nombre | varchar(45) | NO | | NULL | |
| precio | double | NO | | NULL | |
| iva | double | NO | | NULL | |
| cantidad | int | NO | | 0 | |

4 • `describe puesto;`

| Field | Type | Null | Key | Default |
|-----------|-------------|------|-----|---------|
| id_puesto | int | NO | PRI | NULL |
| nombre | varchar(45) | NO | | NULL |
| salario | double | NO | | NULL |

4 • `describe venta;`

| Field | Type | Null | Key | Default | Extra |
|-------------|-------------|------|-----|-------------------|-------------------------------------------|
| id_venta | int | NO | PRI | NULL | |
| id_articulo | int | NO | MUL | NULL | |
| id_empleado | int | NO | MUL | NULL | |
| clave | varchar(45) | NO | | NULL | |
| fecha | timestamp | NO | | CURRENT_TIMESTAMP | DEFAULT_GENERATED on update CURRENT_TI... |

RETO 2

Usando la base de datos `cursos`, escribe consultas que permitan responder las siguientes preguntas.

¿Cuál es el nombre de los empleados con el puesto 4?

4 • `select * from puesto where id_puesto = 4;`

| id_puesto | nombre | salario |
|-----------|-----------------|----------|
| 4 | Staff Scientist | 14965.31 |
| NULL | NULL | NULL |

¿Qué puestos tienen un salario mayor a \$10,000?


```
5 • select * from puesto where salario > 10000;
```

| Result Grid | | | |
|-----------------------------------|-----------|-------------------------------|----------|
| Filter Rows: <input type="text"/> | | | |
| Edit: | | | |
| | id_puesto | nombre | salario |
| ▶ | 1 | Analog Circuit Design manager | 28500.98 |
| | 2 | Junior Executive | 10508.47 |
| | 3 | Director of Sales | 28725.56 |
| | 4 | Staff Scientist | 14965.31 |
| | 5 | Desktop Support Technician | 15885.41 |
| | 6 | Budget/Accounting Analyst III | 17131.23 |
| | 7 | Accounting Assistant III | 29257.91 |
| | 8 | Programmer Analyst II | 23223.95 |

¿Qué artículos tienen un precio mayor a \$1,000 y un iva mayor a 100?

```
5 • select * from articulo where precio >1000 and iva > 100;
```

| Result Grid | | | | | |
|-----------------------------------|-------------|--------------------------------|---------|---------|----------|
| Filter Rows: <input type="text"/> | | | | | |
| Edit: | | | | | |
| Export/Import: | | | | | |
| | id_articulo | nombre | precio | iva | cantidad |
| | 2 | Pasta - Angel Hair | 4391.73 | 959.51 | 503 |
| | 3 | Soup Campbells - Tomato Bisque | 2991.35 | 587.59 | 604 |
| | 4 | Wine - Valpolicella Masi | 2625.2 | 770.1 | 575 |
| | 5 | Mousse - Banana Chocolate | 3701.62 | 893.46 | 248 |
| | 7 | Nantucket - Kiwi Berry Cktl. | 5579.47 | 1012.33 | 527 |
| | 8 | Wine - Fontanafredda Barolo | 2684.64 | 327.16 | 682 |
| | 9 | Lotus Rootlets - Canned | 1996.46 | 324.72 | 636 |
| | 10 | Wine - Vovray Sec Domaine Huet | 6066.99 | 890.47 | 397 |

¿Qué ventas incluyen los artículo 135 o 963 y fueron hechas por los empleados 835 o 369?




```
5 • select * from venta where id_articulo = 135 or id_articulo =963
6 and id_empleado = 835 and id_empleado = 369;
```

| Result Grid | | | | | |
|-----------------------------------|----------|-------------|-------------|-----------|---------------------|
| Filter Rows: <input type="text"/> | | | | | |
| Edit: | | | | | |
| Export/Import: | | | | | |
| | id_venta | id_articulo | id_empleado | clave | fecha |
| ▶ | 6 | 135 | 835 | 0049-0032 | 2020-02-03 15:05:27 |
| | 614 | 135 | 436 | 52125-277 | 2020-02-03 15:04:57 |
| * | NULL | NULL | NULL | NULL | NULL |

RETO 3

Usando la base de datos `tienda`, escribe una consulta que permita obtener el top 5 de puestos por salarios.

4 • `select * from puesto order by salario desc limit 5;`

| < | | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|-------------------------------|----------|
| Result Grid | | | |
| Filter Rows: <input type="text"/> | | | |
| Edit:    | | | |
| | id_puesto | nombre | salario |
| ▶ | 494 | Sales Representative | 29996.58 |
| | 18 | Speech Pathologist | 29967.17 |
| | 487 | Analog Circuit Design manager | 29923.95 |
| | 79 | Junior Executive | 29916.06 |
| | 893 | Technical Writer | 29912.53 |
| * | NULL | NULL | NULL |

SESION 2

RETO 1

¿Qué artículos incluyen la palabra `Pasta` en su nombre?

```
9 • select * from articulo where nombre like 'Pasta%';
```

| | id_articulo | nombre | precio | iva | cantidad |
|---|-------------|-----------------------------------|---------|---------|----------|
| ▶ | 2 | Pasta - Angel Hair | 4391.73 | 959.51 | 503 |
| | 27 | Pasta - Elbows, Macaroni, Dry | 3668.7 | 253.66 | 392 |
| | 70 | Pasta - Shells, Medium, Dry | 801.74 | 773.8 | 206 |
| | 91 | Pasta - Cheese / Spinach Bauletti | 5811.44 | 619.36 | 15 |
| | 134 | Pasta - Orzo, Dry | 6537.91 | 1113.99 | 906 |
| | 213 | Pasta - Rotini, Colour, Dry | 1830.13 | 373.98 | 309 |
| | 233 | Pasta - Cannelloni, Sheets, Fresh | 2316.37 | 605.55 | 307 |
| | 327 | Pasta - Cappellini, Dry | 6994.49 | 766.18 | 828 |
| | 361 | Pasta - Penne, Rigate, Dry | 2222.62 | 584.88 | 276 |

¿Qué artículos incluyen la palabra `Cannelloni` en su nombre?

```
10 • select * from articulo where nombre like '%Cannelloni%';
```

| | id_articulo | nombre | precio | iva | cantidad |
|--|-------------|-----------------------------------|---------|--------|----------|
| | 233 | Pasta - Cannelloni, Sheets, Fresh | 2316.37 | 605.55 | 307 |
| | NULL | NULL | NULL | NULL | NULL |

¿Qué nombres están separados por un guión (-) por ejemplo `Puree - Kiwi`?

```
11 • select * from articulo where nombre like '%-%';
```

| | id_articulo | nombre | precio | iva | cantidad |
|--|-------------|--------------------------------|---------|---------|----------|
| | 1 | Chocolate - Feathers | 2738.93 | 12.26 | 144 |
| | 2 | Pasta - Angel Hair | 4391.73 | 959.51 | 503 |
| | 3 | Soup Campbells - Tomato Bisque | 2991.35 | 587.59 | 604 |
| | 4 | Wine - Valpolicella Masi | 2625.2 | 770.1 | 575 |
| | 5 | Mousse - Banana Chocolate | 3701.62 | 893.46 | 248 |
| | 6 | Yeast Dry - Fleischman | 923.18 | 524.08 | 818 |
| | 7 | Nantucket - Kiwi Berry Cktd. | 5579.47 | 1012.33 | 527 |
| | 8 | Wine - Fontanafredda Barolo | 2684.64 | 327.16 | 682 |
| | 9 | Lotus Rootlets - Canned | 1996.46 | 324.72 | 636 |

RETO 2

¿Cuál es el promedio de salario de los puestos?

```
22 • select avg(salario) as "promedio" from puesto;
```

| Result Grid | |
|--------------------|--|
| promedio | |
| 19595.051179999973 | |

¿Cuántos artículos incluyen la palabra Pasta en su nombre?

```
23 • select * from articulo where nombre like '%Pasta%';
```

| Result Grid | | | | | |
|-------------|-----------------------------------|---------|--------|----------|--|
| id_articulo | nombre | precio | iva | cantidad | |
| 2 | Pasta - Angel Hair | 4391.73 | 959.51 | 503 | |
| 27 | Pasta - Elbows, Macaroni, Dry | 3668.7 | 253.66 | 392 | |
| 70 | Pasta - Shells, Medium, Dry | 801.74 | 773.8 | 206 | |
| 91 | Pasta - Cheese / Spinach Bauletti | 5811.44 | 619.36 | 15 | |

¿Cuál es el salario mínimo y máximo?

```
24 • select max(salario) as "Maximo" from puesto;
```

| Result Grid | |
|-------------|--|
| Maximo | |
| 29996.58 | |

```
24 • select min(salario) as "Minimo" from puesto;
```

| Result Grid | |
|-------------|--|
| Minimo | |
| 10013.44 | |

¿Cuál es la suma del salario de los últimos cinco puestos agregados?

```
25 • select sum(salario) as "Promedio" from puesto order by salario desc limit 5;
```

| Result Grid | |
|--------------------|--|
| Promedio | |
| 19595051.179999974 | |

RETO 3

¿Cuántos registros hay por cada uno de los puestos?

```
34 • select nombre, count(*) as cantidad from puesto group by nombre;
```

| Result Grid | | Filter Rows: | Export: | Wrap Cell Content: |
|-------------------------------|----------|--------------|---------|--------------------|
| nombre | cantidad | | | |
| Analog Circuit Design manager | 8 | | | |
| Junior Executive | 8 | | | |
| Director of Sales | 8 | | | |
| Staff Scientist | 9 | | | |
| Desktop Support Technician | 5 | | | |
| Budget/Accounting Analyst III | 4 | | | |

¿Cuánto dinero se paga en total por puesto?

```
36 • select nombre, sum(salario) from puesto group by nombre;
```

| Result Grid | | Filter Rows: | Export: | Wrap Cell Content: |
|-------------------------------|--------------------|--------------|---------|--------------------|
| nombre | sum(salario) | | | |
| Analog Circuit Design manager | 179310.18000000002 | | | |
| Junior Executive | 156846.26 | | | |
| Director of Sales | 136630.69 | | | |
| Staff Scientist | 157528.98 | | | |
| Desktop Support Technician | 92315.22 | | | |
| Budget/Accounting Analyst III | 70107.77 | | | |
| Accounting Assistant III | 78947.08 | | | |
| Programmer Analyst II | 35658.78 | | | |
| Nurse Practitioner | 296384.04 | | | |

¿Cuál es el número total de ventas por vendedor?

```
37 • select id_pleado, count(clave) as ventas from venta group by id_pleado;
```

| Result Grid | | Filter Rows: | Export: | Wrap Cell Content: |
|-------------|--------|--------------|---------|--------------------|
| id_pleado | ventas | | | |
| 2 | 2 | | | |
| 3 | 2 | | | |
| 4 | 1 | | | |
| 5 | 1 | | | |

¿Cuál es el número total de ventas por artículo?

```
38 • select id_articulo, count(*) from venta group by id_articulo;
```


| Result Grid | | Filter Rows: | Export: | Wrap Cell Content: |
|-------------|----------|--------------|---------|--------------------|
| id_articulo | count(*) | | | |
| 2 | 1 | | | |
| 3 | 1 | | | |
| 4 | 2 | | | |
| 8 | 1 | | | |


RETO 4

¿Cuál es el nombre de los empleados cuyo sueldo es menor a \$10,000?

```
7 • select nombre, apellido_paterno
8   from empleado
9  where id_puesto IN
10      (select id_puesto from puesto where salario < 20000);
11  -- |
```


Result Grid





Filter Rows:

Export:



Wrap Cell Content:

| nombre | apellido_paterno |
|---------|------------------|
| Norrie | McGarrie |
| Maxy | Udden |
| Della | Fulbrook |
| Katya | Banbridge |
| Robyn | Hancock |
| Hayyim | Verdon |
| Analise | Beteriss |
| Artair | Dearn |

¿Cuál es la cantidad mínima y máxima de ventas de cada empleado?

```
12 • select id_empleado, min(total_ventas), max(total_ventas)
13   from
14   (select clave, id_empleado, count(*) total_ventas
15    from venta
16   group by clave, id_empleado) as sq
17   group by id_empleado;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

| | id_empleado | min(total_ventas) | max(total_ventas) |
|--|-------------|-------------------|-------------------|
| | 569 | 1 | 1 |
| | 413 | 1 | 2 |
| | 765 | 1 | 1 |
| | 119 | 1 | 1 |
| | 90 | 1 | 1 |
| | 835 | 1 | 1 |
| | 369 | 1 | 1 |
| | 555 | 1 | 1 |

¿Cuál es el nombre del puesto de cada empleado?

```
20 • SELECT nombre, apellido_paterno, (SELECT nombre FROM puesto p WHERE p.id_puesto = e.id_puesto)
21 FROM empleado e;
22
```

| Result Grid | | | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows: |
|-------------|------------------|---------------------------------------------------------------|--------------|---------|--------------------|-------------|
| nombre | apellido_paterno | (SELECT nombre FROM puesto p WHERE p.id_puesto = e.id_puesto) | | | | |
| Enrichetta | Bodechon | Product Engineer | | | | |
| Morey | Bowskill | Budget/Accounting Analyst IV | | | | |
| Jeannette | Potes | Occupational Therapist | | | | |
| Cassey | Womersley | Financial Advisor | | | | |
| Gnni | Risom | Physical Therapy Assistant | | | | |
| Lisle | Carlsson | Marketing Assistant | | | | |
| Andre | Theurer | Tax Accountant | | | | |
| Land | Locksley | Product Engineer | | | | |
| Nikki | Fayerbrother | Sales Associate | | | | |

SESION 3

RETO 1

¿Cuál es el nombre de los empleados que realizaron cada venta?

```
4 • select clave, nombre, apellido_paterno from venta as v join empleado as e
5   on v.id_empleado = e.id_empleado order by clave;
```

| Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows: |
|-------------|--------------|------------------|--------------------|-------------|
| clave | nombre | apellido_paterno | | |
| 0002-8149 | Leonidas | Junkinson | | |
| 0002-8149 | Leslie | Cuvley | | |
| 0002-8149 | Edee | Billin | | |
| 0002-8149 | Jillie | Corter | | |
| 0002-8149 | Sydney | Woolway | | |
| 0002-8149 | Aguistin | Richarz | | |
| 0002-8149 | Sigfrid | Teal | | |
| 0002-8149 | Rebecka | Rushworth | | |
| 0002-8149 | Nora | O'Suaird | | |
| 0002-8149 | Luise | Lennard | | |

¿Cuál es el nombre de los artículos que se han vendido?

```
7 • select clave, nombre from venta as v join articulo as a on v.id_articulo = a.id_articulo
8   order by nombre;
```

| Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows: |
|-------------|---------------------|---------|--------------------|-------------|
| clave | nombre | | | |
| 61852-637 | Alize Gold Passion | | | |
| 0049-0032 | Alize Gold Passion | | | |
| 36800-580 | Alize Red Passion | | | |
| 51655-951 | Alize Sunset | | | |
| 47335-894 | Alize Sunset | | | |
| 0049-0032 | Alize Sunset | | | |
| 69128-001 | Allspice - Jamaican | | | |
| 36987-1911 | Amaretto | | | |
| 54868-4536 | Amarula Cream | | | |
| 48951-5065 | Amarula Cream | | | |

¿Cuál es el total de cada venta?

```
10 • select clave, sum(precio) as total from venta as v join articulo as a
11 on v.id_articulo = a.id_articulo group by id_venta;
12
```

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

| clave | total |
|-----------|---------|
| 0228-3661 | 3714.37 |
| 52125-277 | 1157.42 |
| 0049-0032 | 4788.24 |
| 52125-277 | 4786.97 |
| 13107-062 | 5816.84 |
| 0049-0032 | 5066.16 |
| 47335-894 | 2735.46 |
| 52125-277 | 1133.63 |
| 0049-0032 | 2115.44 |
| 13107-062 | 2689.33 |

RETO 2

Obtener el puesto de un empleado.

```
1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`%`
4     SQL SECURITY DEFINER
5     VIEW `vistaRCA` AS
6     SELECT
7         CONCAT(`e`.`nombre`,
8             ' ',
9             `e`.`apellido_paterno`) AS `concat(e.nombre, ' ', e.apellido_paterno)`,
10         `p`.`nombre` AS `nombre`
11     FROM
12         (`empleado` `e`
13     JOIN `puesto` `p` ON ((`e`.`id_puesto` = `p`.`id_puesto`)))
```

```
select * from vistaRCA;
```

| concat(e.nombre, ' ', e.apellido_paterno) | nombre |
|-------------------------------------------|------------------------------|
| Enrichetta Bodechon | Product Engineer |
| Morey Bowskill | Budget/Accounting Analyst IV |
| Jeannette Potes | Occupational Therapist |
| Cassey Womersley | Financial Advisor |
| Gnni Risom | Physical Therapy Assistant |
| Lisle Carlsson | Marketing Assistant |
| Andre Theurer | Tax Accountant |
| Land Locksley | Product Engineer |
| Nikki Fayerbrother | Sales Associate |

Saber qué artículos ha vendido cada empleado.

```
1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`%`
4     SQL SECURITY DEFINER
5     VIEW `RCA_empleado_articulo` AS
6     SELECT
7         CONCAT(`e`.`nombre`,
8             ',
9             `e`.`apellido_paterno`) AS `Nombre`,
10        `a`.`nombre` AS `Articulo`
11     FROM
12         ((`empleado` `e`
13         JOIN `venta` `v` ON ((`e`.`id_empleado` = `v`.`id_empleado`)))
14         JOIN `articulo` `a` ON ((`v`.`id_articulo` = `a`.`id_articulo`)))
15     ORDER BY CONCAT(`e`.`nombre`,
```

```
select * from RCA_empleado_articulo;
```

| Nombre | Articulo |
|-----------------|------------------------------|
| Aaren Pryce | Lemonade - Pineapple Passion |
| Aaren Pryce | Honey - Liquid |
| Aaron Klossmann | Bread - 10 Grain Parisian |
| Aaron Klossmann | Beans - Long, Chinese |
| Abagael Buzing | Food Colouring - Green |
| Abbie Tibald | Chef Hat 25cm |
| Abigail Shama | Tilapia - Fillets |
| Adan Bauckham | Mix Pina Colada |
| Adan Bauckham | Bread - Bistro White |
| Adan Berthelot | Steampan Lid |

Saber qué puesto ha tenido más ventas.

```
1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`%`
4     SQL SECURITY DEFINER
5     VIEW `RCA_puesto_ventas` AS
6     SELECT
7         `p`.`nombre` AS `Puesto`, COUNT(`v`.`clave`) AS `Cantidad`
8     FROM
9         ((`puesto` `p`
10        JOIN `empleado` `e` ON ((`p`.`id_puesto` = `e`.`id_puesto`)))
11        JOIN `venta` `v` ON ((`v`.`id_empleado` = `e`.`id_empleado`)))
12     GROUP BY `Puesto`
13     ORDER BY `Cantidad` DESC
14     LIMIT 1
```

```
select * from RCA_puesto_ventas;
```

| Puesto | Cantidad |
|----------------------------|----------|
| Physical Therapy Assistant | 23 |

SESSION 4

RETO 1

Usando la base de datos `sample_mflix`, proyecta los datos que se solicitan.

Fecha, nombre y texto de cada comentario.

FILTER

PROJECT {date:1, name:1, text:1}

SORT

COLLATION

VIEW

_id:ObjectId("5a9427648b0beeb69579cc")
name:"Andrea Le"
text:"Rem officiis eaque repellendus amet eos doloribus. Porro dolor volupta..."
date:2012-03-26T23:20:16.000+00:00

_id:ObjectId("5a9427648b0beeb69579cf")
name:"Greg Powell"
text:"Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Co..."
date:1987-02-10T00:29:36.000+00:00

_id:ObjectId("5a9427648b0beeb69579d0")
name:"Talisa Maegyr"
text:"Rem itaque ad sit rem voluptatibus. Ad fugiat maxime illum optio iure ..."
date:1998-08-22T11:45:03.000+00:00

Título, elenco y año de cada película.

FILTER

PROJECT

SORT

COLLATION

VIEW

<

Nombre y contraseña de cada usuario.

The screenshot shows a MongoDB query interface with the following settings:

- PROJECT:** `{name:1, password:1 }`
- SORT:** (empty)
- COLLATION:** (empty)
- VIEW:** (List view selected)

The results display three documents:

```
_id: ObjectId("59b99db4cfa9a34dcd7885b6")
name: "Ned Stark"
password: "$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIj9RJAqMUQ74cr1j1Vu"

_id: ObjectId("59b99db4cfa9a34dcd7885b7")
name: "Robert Baratheon"
password: "$2b$12$yGqxLG9LZpXA2xVDhuPnSOzd.VURVxz7wgOLY3pn00s7u2S1Z032y"

_id: ObjectId("59b99db5cfa9a34dcd7885b8")
name: "Jaime Lannister"
password: "$2b$12$6vz7wiw0.EI5Rilvq1zUc./9480gb1uPtXcahDxIadgyC3PS8XCUK"
```

RETO 2

Usando la base de datos `sample_mflix`, agrega proyecciones, filtros, ordenamientos y límites que permitan contestar las siguientes preguntas.

¿Qué comentarios ha hecho Greg Powell?

The screenshot shows a MongoDB query interface with the following settings:

- FILTER:** `{name: "Greg Powell"}`
- PROJECT:** (empty)
- SORT:** (empty)
- COLLATION:** (empty)
- VIEW:** (List view selected)

The results display two documents:

```
_id: ObjectId("5a9427648b0beeb69579cf")
name: "Greg Powell"
email: "greg_powell@fakegmail.com"
movie_id: ObjectId("573a1390f29313caabcd41b1")
text: "Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Co..."
date: 1987-02-10T00:29:36.000+00:00

_id: ObjectId("5a9427648b0beeb6957afe")
name: "Greg Powell"
email: "greg_powell@fakegmail.com"
movie_id: ObjectId("573a1391f29313caabcd754b")
text: "Rem nostrum nobis saepe eaque itaque nemo. Fugit dignissimos nisi sapi..."
date: 2013-03-26T16:20:03.000+00:00
```





¿Qué comentarios han hecho Greg Powell o Mercedes Tyler?

FILTER `{ $or: [{ name: "Greg Powell" }, { name: "Mercedes Tyler" }] }`

PROJECT

SORT

COLLATION

ADD DATA  **VIEW**   

```
_id: ObjectId("5a9427648b0beebeb69579cf")
name: "Greg Powell"
email: "greg_powell@fakegmail.com"
movie_id: ObjectId("573a1390f29313caab41b1")
text: "Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Co..."
date: 1987-02-10T00:29:36.000+00:00
```

```
_id: ObjectId("5a9427648b0beebeb69579e7")
name: "Mercedes Tyler"
email: "mercedes_tyler@fakegmail.com"
movie_id: ObjectId("573a1390f29313caab4323")
text: "Eius veritatis vero facilis quaerat fuga temporibus. Praesentium exped..."
date: 2002-08-18T04:56:07.000+00:00
```

¿Cuál es el máximo número de comentarios en una película?

FILTER **OPTIONS**

PROJECT `{ num_mflix_comments: 1, title: 1 }`

SORT `{ num_mflix_comments: -1 }` **MAXTIME** 5000

COLLATION **SKIP** 0 **LIMIT** 1

VIEW    **Displaying documents 1 - 1**

```
_id: ObjectId("573a1399f29313caabcee886")
title: "The Mask"
num_mflix_comments: 456
```

¿Cuál es título de las cinco películas más comentadas?

PROJECT {num_mflix_comments:1, title:1}

SORT {num_mflix_comments:-1}

MAXTIMEMS 5000

COLLATION

SKIP 0

LIMIT 5

VIEW

Displaying documents 1 - 5

_id: ObjectId("573a1399f29313caabcee886")
title: "The Mask"
num_mflix_comments: 456

_id: ObjectId("573a1399f29313caabcee578")
title: "Dumb & Dumber"
num_mflix_comments: 450

_id: ObjectId("573a13bff29313caabd6001f")
title: "The Unborn"
num_mflix_comments: 447

_id: ObjectId("573a13a5f29313caabd159a9")
title: "About a Boy"
num_mflix_comments: 441

_id: ObjectId("573a13a7f29313caabd1aa55")
title: "8 Mile"
num_mflix_comments: 441

SESION 5

RETO 1

a) Propiedades que no permitan fiestas.

FILTER

{house_rules: /No parties/i}

PROJECT

{name:1, house_rules:1}

SORT

COLLATION

SKIP

VIEW

{}

_id: "103161"

name: "Cozy Art Top Floor Apt in PRIME Williamsburg!"

house_rules: "-SMOKE is NOT allowed inside the Apt. -NO parties are allowed. -NO Kid..."

_id: "10392282"

name: "Banyan Bungalow"

house_rules: "No smoking, no pets, no parties. You are welcome to have guests, but ..."

_id: "10423504"

name: "Bondi Beach Dreaming 3-Bed House"

house_rules: "Guests should have Airbnb verification at time of booking. We expect g..."

_id: "10840938"

name: "Spacious 3bdrm. Artsy and Perfect Location!"

house_rules: "No parties or loud music after 10 No smoking inside Take care of my p..."

b) Propiedades que admitan mascotas.

FILTER

{ \$and: [{ house_rules: /pets allowed/i }, { house_rules: { \$not: /no pets allowed/i } }] }

PROJECT

{name:1, house_rules:1}

SORT

COLLATION

VIEW

{}

_id: "13161668"

name: "Eliseos Luxury Apartment HUTB-001281"

house_rules: "- Only small pets allowed"

_id: "13165887"

name: "Vela Luxury Apartment HUTB-001280"

house_rules: "- Only smalll pets allowed"

_id: "16981802"

name: "Stylish urban loft in Verdun"

house_rules: "- Absolutely no smoking or pets allowed; - No partying;"

_id: "20206764"

name: "Awesome 2-storey home Bronte Beach next to Bondi!"

house_rules: "House Rules please be responsible for all guests; * Pets allowed by pr..."




c) Propiedades que no permitan fumadores.

FILTER {house_rules: /No smoking/i}

PROJECT {{name:1, house_rules:1}}

SORT

COLLATION

VIEW   

```
{
  "_id": "1003530",
  "name": "New York City - Upper West Side Apt",
  "house_rules": "No smoking is permitted in the apartment. All towels that are used sho..."
},
{
  "_id": "10083468",
  "name": "Be Happy in Porto",
  "house_rules": ". No smoking inside the apartment. . Is forbidden receive or lead stra..."
},
{
  "_id": "10084023",
  "name": "City center private room with bed",
  "house_rules": "1. 禁止吸烟, 只限女生入住 (除得到批准) No smoking and only female is allowed 2. 热水是..."
},
{
  "_id": "10091713",
  "name": "Surry Hills Studio - Your Perfect Base in Sydney",
  "house_rules": "No smoking: No smoking any substance, including e-cigarettes. Lost Ke..."
}
```




d) Propiedades que no permitan fiestas ni fumadores.

FILTER {\$and:[{house_rules:/no parties/i},{house_rules:{\$not:/no smoking/i}}]}

PROJECT {{name:1, house_rules:1}}

SORT

COLLATION

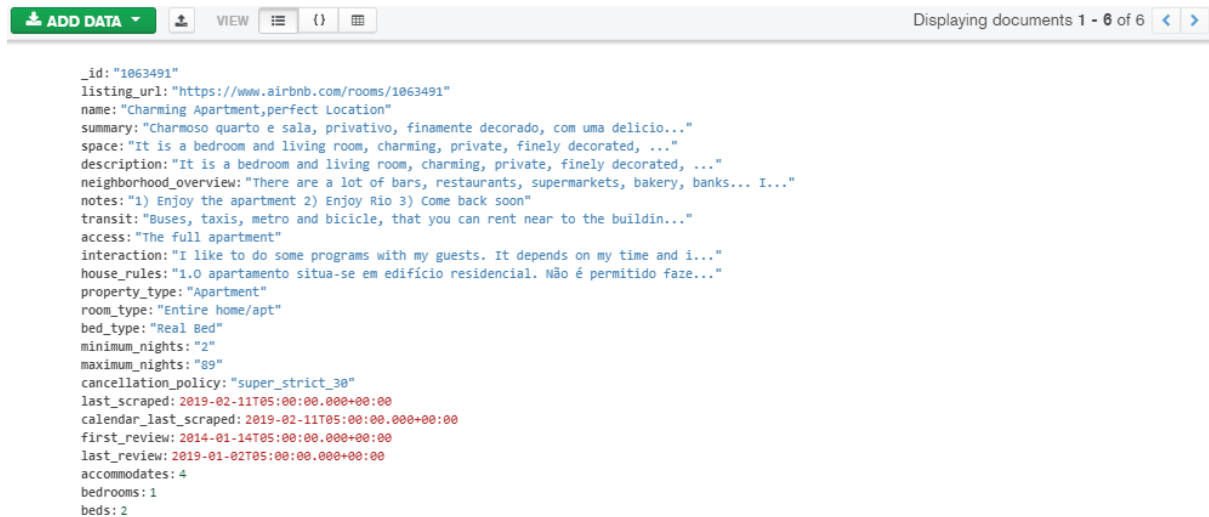
VIEW   

```
{
  "_id": "103161",
  "name": "Cozy Art Top Floor Apt in PRIME Williamsburg!",
  "house_rules": "-SMOKE is NOT allowed inside the Apt. -NO parties are allowed. -NO Kid..."
},
{
  "_id": "1098211",
  "name": "Super location - Walk to the City!",
  "house_rules": "Make yourself at home. Please leave the apartment clean and tidy thoug..."
},
{
  "_id": "1104768",
  "name": "2 Bdrm/2 Bath Family Suite Ocean View",
  "house_rules": "Unit includes all sheets, towels, kitchen equipment and linens. There ..."
},
{
  "_id": "11197217",
  "name": "2 Bedroom Ipanema-Copacabana",
  "house_rules": "We would appreciate if the guest, as leaving the apartment even for a ..."
}
```

RETO 2

- a) Usando la colección `sample_airbnb.listingsAndReviews`, agrega un filtro que permita 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 ubicada en Brazil.

```
{number_of_reviews:{$gt:50},"address.country":  
"Brazil","review_scores.review_scores_rating":{$gte:80},amenities:{$in:[/Ethernet/]}}
```

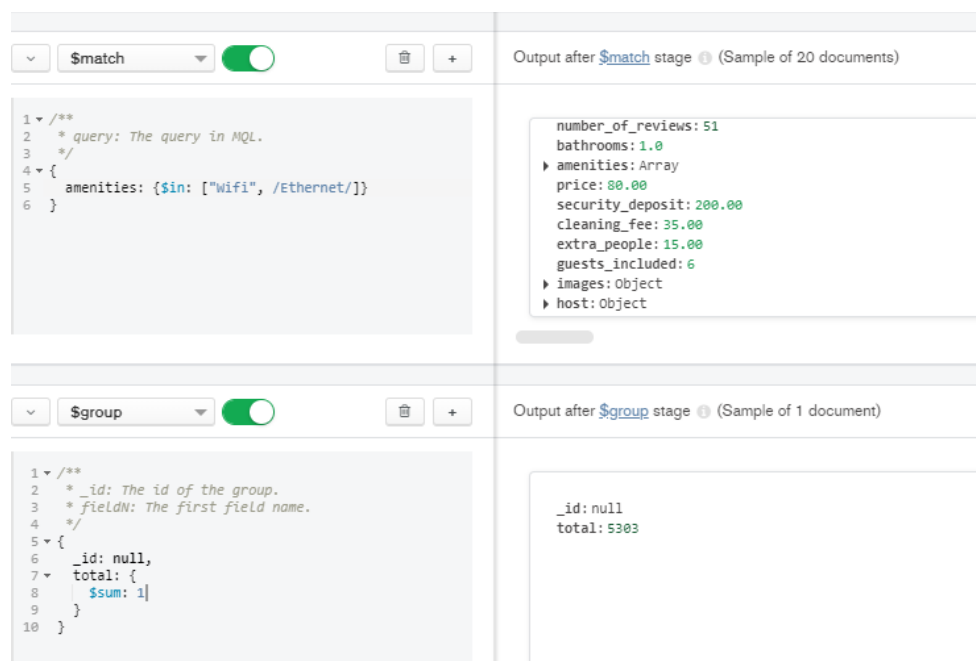


The screenshot shows a MongoDB document with the following fields and values:

- `_id`: "1063491"
- `listing_url`: "https://www.airbnb.com/rooms/1063491"
- `name`: "Charming Apartment,perfect Location"
- `summary`: "Charmoso quarto e sala, privativo, finamente decorado, com uma delicio..."
- `space`: "It is a bedroom and living room, charming, private, finely decorated, ..."
- `description`: "It is a bedroom and living room, charming, private, finely decorated, ..."
- `neighborhood_overview`: "There are a lot of bars, restaurants, supermarkets, bakery, banks... I..."
- `notes`: "1) Enjoy the apartment 2) Enjoy Rio 3) Come back soon"
- `transit`: "Buses, taxis, metro and bicicle, that you can rent near to the buildin..."
- `access`: "The full apartment"
- `interaction`: "I like to do some programs with my guests. It depends on my time and i..."
- `house_rules`: "1.0 apartamento situa-se em edificio residencial. Não é permitido faze..."
- `property_type`: "Apartment"
- `room_type`: "Entire home/apt"
- `bed_type`: "Real Bed"
- `minimum_nights`: "2"
- `maximum_nights`: "89"
- `cancellation_policy`: "super_strict_30"
- `last_scraped`: "2019-02-11T05:00:00.000+00:00"
- `calendar_last_scraped`: "2019-02-11T05:00:00.000+00:00"
- `first_review`: "2014-01-14T05:00:00.000+00:00"
- `last_review`: "2019-01-02T05:00:00.000+00:00"
- `accommodates`: 4
- `bedrooms`: 1
- `beds`: 2

RETO 3

- a) Usando la colección `sample_airbnb.listingsAndReviews`, mediante el uso de agregaciones, encontrar el número de publicaciones que tienen conexión a Internet, sea desde Wifi o desde cable (Ethernet).



The screenshot shows a MongoDB Aggregation Pipeline with two stages:

Stage 1: \$match

```
1 /**  
2  * query: The query in MQL.  
3  */  
4 {  
5   amenities: {$in: ["Wifi", /Ethernet/]}  
6 }
```

Output after `$match` stage (Sample of 20 documents):

```
number_of_reviews: 51  
bathrooms: 1.0  
amenities: Array  
price: 80.00  
security_deposit: 200.00  
cleaning_fee: 35.00  
extra_people: 15.00  
guests_included: 6  
images: Object  
host: Object
```

Stage 2: \$group

```
1 /**  
2  * _id: The id of the group.  
3  * fieldName: The first field name.  
4  */  
5 {  
6   _id: null,  
7   total: {  
8     $sum: 1  
9   }  
10 }
```

Output after `$group` stage (Sample of 1 document):

```
_id: null  
total: 5303
```

SESION 6

RETO 1

Con base en el ejemplo 1, modifica el agrupamiento para que muestre el costo promedio por habitación por país de las propiedades de tipo casa.

```
[{$match: {
  property_type: "House",
  bedrooms:{$gte: 1}
}}, {$addFields: {
  costo_recamara: {$divide: ["$price" , "$bedrooms"]}
}}, {$group: {
  _id: "$address.country",
  recamaras: {
    $sum: 1
  },
  total: {
    $sum: "$costo_recamara"
  }
}}, {$addFields: {
  pais: "$_id",
  costo_promedio: {
    $divide: ["$total", "$recamaras"]
  }
}}, {$project: {
  _id:0,
  pais:1,
  costo_promedio:1
}}]
```

[illegible]



RETO 2

Usando las colecciones `comments` y `users`, se requiere conocer el correo y contraseña de cada persona que realizó un comentario. Construye un pipeline que genere como resultado estos datos.

```

[{$lookup: {
  from: 'users',
  localField: 'email',
  foreignField: 'email',
  as: 'user'
}}, {$addFields: {
  usr_obj :{$arrayElemAt:["$user",0]}
}}, {$match: {
  usr_obj :{$exists:true}
}}, {$addFields: {
  usuario_password: "$usr_obj.password"
}}, {$project: {
  _id:0,
  name:1,
  usuario_password:1,
  email:1
}}]

```

⋮

\$lookup

🔴

🗑️

+

Output after [\\$lookup](#) stage ⓘ (Sample of 20 documents)

```
1 /**
2  * from: The target collection.
3  * localField: The local join field.
4  * foreignField: The target join field.
5  * as: The name for the results.
6  * pipeline: The pipeline to run on the joined colle
7  * let: Optional variables to use in the pipeline fi
8  */
9 {
10   from: 'users',
11   localField: 'email',
12   foreignField: 'email',
13   as: 'user'
14 }
```

```
_id: ObjectId("5a9427648b0beeb69579cc")
name: "Andrea Le"
email: "andrea_le@fakegmail.com"
movie_id: ObjectId("573a1390f29313caabcd418c")
text: "Rem officiis eaque repellendus amet eos
      doloribus. Porro dolor volupta..."
date: 2012-03-26T23:20:16.000+00:00
user: Array
```

⋮

\$addFields

🔴

🗑️

+

Output after [\\$addFields](#) stage ⓘ (Sample of 20 documents)

```
1 /**
2  * newField: The new field name.
3  * expression: The new field expression.
4  */
5 {
6   usr_obj :{$arrayElemAt:["$user",0]}
7 }
```

```
_id: ObjectId("5a9427648b0beeb69579cc")
name: "Andrea Le"
email: "andrea_le@fakegmail.com"
movie_id: ObjectId("573a1390f29313caabcd418c")
text: "Rem officiis eaque repellendus amet eos
      doloribus. Porro dolor volupta..."
date: 2012-03-26T23:20:16.000+00:00
user: Array
usr_obj: Object
```

⋮

\$match

🔴

🗑️

+

Output after [\\$match](#) stage ⓘ (Sample of 20 documents)

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   usr_obj :{$exists:true}
6 }
```

```
_id: ObjectId("5a9427648b0beeb69579cc")
name: "Andrea Le"
email: "andrea_le@fakegmail.com"
movie_id: ObjectId("573a1390f29313caabcd418c")
text: "Rem officiis eaque repellendus amet eos
      doloribus. Porro dolor volupta..."
date: 2012-03-26T23:20:16.000+00:00
user: Array
usr_obj: Object
```

⋮

\$addFields

🔴

🗑️

+

Output after [\\$addFields](#) stage ⓘ (Sample of 20 documents)

```
1 /**
2  * newField: The new field name.
3  * expression: The new field expression.
4  */
5 {
6   usuario_password: "$usr_obj.password"
7 }
```

```
name: "Andrea Le"
email: "andrea_le@fakegmail.com"
movie_id: ObjectId("573a1390f29313caabcd418c")
text: "Rem officiis eaque repellendus amet eos
      doloribus. Porro dolor volupta..."
date: 2012-03-26T23:20:16.000+00:00
user: Array
usr_obj: Object
usuario_password: "$2b$12$JS87HwUL2y0P1E6kYrcbKOKx22.ws"
```

\$project

Output after [\\$project](#) stage ⓘ (Sample of 20 documents)

```

1 ▾ /**
2   * specifications: The fields to
3   *   include or exclude.
4   */
5   ▾ {
6     _id:0,
7     name:1,
8     usuario_password:1,
9     email:1
10  }

```

```

name: "Andrea Le"
email: "andrea_le@fakegmail.com"
usuario_password: "$2b$12$JS87HWUL2y0P1E6kYrcBKOKx22.wS

```

RETO 3

Usando el *pipeline* que generaste en el Reto 2, genera la vista correspondiente.

[sample_mflix.usuario-password](#) (view on: [sample_mflix.comments](#))

Documents

Aggregations

Schema

Explain Plan

Indexes

FILTER

VIEW

{ }

```

name: "Meera Reed"
email: "ellie_kendrick@gameofthron.es"
usuario_password: "$2b$12$SdyXPBMdGScx6DePpPawFeOqcpiwjdHAuTXaPl0mkvWeLZzk6EWti"

```

```

name: "Emily Ellis"
email: "emily_ellis@fakegmail.com"
usuario_password: "$2b$12$UuCb5RqPEgheoLlwOF/Jb.x9gpFVmD30oUwpSRK1jwo8pBUmWT6eG"

```

```

name: "Petyr Baelish"
email: "aidan_gillen@gameofthron.es"
usuario_password: "$2b$12$qM.YvmiekyYY7p7phpK30icbRCDkN7ESwYAnG/o9YnfHC0Mhkmbi"

```

```

name: "Sarah Lewis"
email: "sarah_lewis@fakegmail.com"
usuario_password: "$2b$12$5kCUjCP3lvYSzhouVJTp0eCZ7e7Xke8gDoPPg2Uyz39tNKrp9om1a"

```


SESION 8

RETO 1

1. Descarga la fuente de datos de los locales de Starbucks:
[directory.csv](#)
2. Analiza los datos, limpia los datos en caso de ser necesario.
3. Elige MySQL o MongoDB y crea una base de datos para el conjunto de datos del reto.
4. Carga los datos en la base de datos que elegiste y revisa que éstos se muestren correctamente.
5. 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.

FILTER

`{ $and: [{ Longitude: { $gte: -115.5 } }, { Latitude: { $lte: 33 } }] }`

PROJECT

`{ Brand:1, "Store Name":1, "Street Address":1, City:1, Longitude:1, Latitude:1 }`

SORT

`{ Longitude:1 }`

COLLATION

VIEW

```
_id: ObjectId("5f1f8806f747b32068f0a0fe")
Brand: "Starbucks"
Store Name: "Mexicali Plaza del Valle"
Street Address: "Blvd. Lazaro Cardenas n.1801, Zona Industrial"
City: "Mexicali"
Longitude: -115.46
Latitude: 32.62
```

```
_id: ObjectId("5f1f8806f747b32068f0a101")
Brand: "Starbucks"
Store Name: "Plaza Centenario"
Street Address: "Benito Juarez Boulevard y Lazario, Cardenas sin, Fracc. Residencias"
City: "Mexicali"
Longitude: -115.44
Latitude: 32.63
```

RETO 2

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

```
[$match: {  
  Country: {  
    $ne: 'Grand Total'  
  }  
}], {$sort: {  
  date: -1  
}], {$sort: {  
  Deaths: -1  
}], {$limit: 1}]
```

UI interface for the first query. The left pane shows the query stages: a dropdown menu with '\$limit' selected, a toggle switch that is turned on, and buttons for deleting and adding stages. The right pane shows the output after the '\$limit' stage, displaying a sample of 1 document.

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

```
{  
  "_id": ObjectId("5f1f984ef747b32068f0f007"),  
  "Country": "United States of America",  
  "Cases": 33902,  
  "Deaths": 170,  
  "Update Time": 2009-07-06T16:00:00.000+00:00  
}
```

- ¿Cuál fue el país con menor número de muertes?

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

UI interface for the second query. The left pane shows the query stages: a dropdown menu with '\$limit' selected, a toggle switch that is turned on, and buttons for deleting and adding stages. The right pane shows the output after the '\$limit' stage, displaying a sample of 1 document.

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

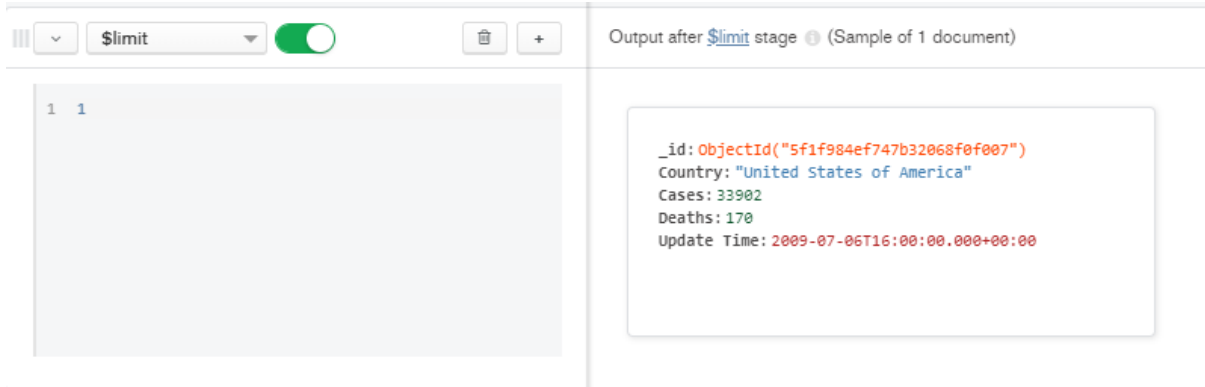
```
{  
  "_id": ObjectId("5f1f984ef747b32068f0ef89"),  
  "Country": "Algeria",  
  "Cases": 5,  
  "Deaths": 0,  
  "Update Time": 2009-07-06T16:00:00.000+00:00  
}
```

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

```

[{$match: {
  Country: {
    $ne: 'Grand Total'
  }
}}, {$sort: {
  date: -1
}}, {$sort: {
  Cases: -1
}}, {$limit: 1}]

```



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

```

_id: ObjectId("5f1f984ef747b32068f0f007")
Country: "United States of America"
Cases: 33902
Deaths: 170
Update Time: 2009-07-06T16:00:00.000+00:00


```

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

```

[{$match: {
  Country: {
    $ne: 'Grand Total'
  }
}}, {$sort: {
  date: -1
}}, {$sort: {
  Cases: 1
}}, {$limit: 1}]

```



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

```

_id: ObjectId("5f1f984ef747b32068f0ef93")
Country: "Bermuda, UKOT"
Cases: 1
Deaths: 0
Update Time: 2009-07-06T16:00:00.000+00:00

```

- ¿Cuál fue el número de muertes promedio?

```

[{$match: {
  Country: {
    $ne: 'Grand Total'
  }
}}, {$group: {
  _id: '$Country',

```

```

    maxDeathsPerCountry: {
      $max: '$Deaths'
    }
  }, {$sort: {
    maxDeathsPerCountry: -1
  }}, {$group: {
    _id: null,
    totalDeaths: {
      $sum: '$maxDeathsPerCountry'
    },
    totalCountries: {
      $sum: 1
    }
  }}, {$addFields: {
    averageDeaths: {
      $divide: [
        '$totalDeaths',
        '$totalCountries'
      ]
    }
  }}, {$project: {
    _id: 0,
    averageDeaths: 1
  }}

```

\$project

Output after [\\$project](#) stage ⓘ (Sample of 1 document)

```

1  {
2    _id: 0,
3    averageDeaths: 1
4  }

```

```

averageDeaths: 3

```

- ¿Cuál fue el número de casos promedio?

```

[{$match: {
  Country: {
    $ne: 'Grand Total'
  }
}}, {$group: {
  _id: '$Country',
  maxDeathsPerCountry: {
    $max: '$Deaths'
  }
}}, {$sort: {
  maxDeathsPerCountry: -1
}}

```

```

}}, {$group: {
  _id: null,
  totalDeaths: {
    $sum: '$maxDeathsPerCountry'
  },
  totalCountries: {
    $sum: 1
  }
}}, {$addFields: {
  averageDeaths: {
    $divide: [
      '$totalDeaths',
      '$totalCountries'
    ]
  }
}}, {$project: {
  _id: 0,
  averageDeaths: 1
}}

```

\$project

Output after [\\$project](#) stage ⓘ (Sample of 1 document)

```
averageDeaths: 3
```

- Top 5 de países con más muertes

```

[{$match: {
  Country: {
    $ne: 'Grand Total'
  }
}}, {$group: {
  _id: '$Country',
  maxDeathsPerCountry: {
    $max: '$Deaths'
  }
}}, {$sort: {
  maxDeathsPerCountry: -1
}}, {$limit: 5}]

```

\$limit

Output after [\\$limit](#) stage (Sample of 5 documents)

15

_id: "United States of America"

maxDeathsPerCountry: 170

_id: "Mexico"

maxDeathsPerCountry: 119

Output after [\\$limit](#) stage (Sample of 5 documents)

_id: "Argentina"

maxDeathsPerCountry: 60

_id: "Canada"

maxDeathsPerCountry: 25

of 5 documents)

_id: "Chile"

maxDeathsPerCountry: 14

- Top 5 de países con menos muertes

```

[{$match: {
  Country: {
    $ne: 'Grand Total'
  }
}}, {$group: {
  _id: '$Country',
  maxDeathsPerCountry: {
    $max: '$Deaths'
  }
}}, {$sort: {
  maxDeathsPerCountry: 1
}}, {$limit: 5}]

```

\$limit

Output after [\\$limit](#) stage (Sample of 5 documents)

15

_id: "Monaco"

maxDeathsPerCountry: 0

_id: "Antigua and Barbuda"

maxDeathsPerCountry: 0

Output after [\\$limit](#) stage ⓘ (Sample of 5 documents)

```
_id: "Croatia"
maxDeathsPerCountry: 0
```

```
_id: "South Africa"
maxDeathsPerCountry: 0
```

5 documents)

```
_id: "Iran"
maxDeathsPerCountry: 0
```

RETO 3

¿Cuál es país con mayor número de casos?

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

||| ☒

Output after [\\$limit](#) stage ⓘ (Sample of 1 document)

```
1 1
```

```
_id: "Mainland China"
maxCasesPerCountry: 67217
```

¿Cuál es el país con mayor número de muertes?

```
[{$group: {
  _id: '$Region',
  maxDeathsPerCountry: {
```

```

    $max: '$Deaths'
  }
}, {$sort: {
  maxDeathsPerCountry: -1
}}, {$limit: 1}]

```

The screenshot shows a MongoDB Atlas query editor interface. At the top, there's a query stage named '\$limit' with a toggle switch turned on. Below the stage name, there are icons for deleting and adding stages. To the right, it says 'Output after \$limit stage (Sample of 1 document)'. The main area is split into two panes. The left pane shows a list of documents with a single document at index 1. The right pane shows the details of that document: { "_id": "Mainland China", "maxDeathsPerCountry": 2835 }.

Usando las coordenadas, encuentra el epicentro del virus

```

[{$match: {
  Lat: {
    $ne: ""
  },
  Long: {
    $ne: ""
  }
}}, {$addFields: {
  Lat: {
    $convert: {
      input: '$Lat',
      to: 'double'
    }
  },
  Long: {
    $convert: {
      input: '$Long',
      to: 'double'
    }
  }
}}, {$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'
      ]
    }
  }}
}
}}

```

\$project

Output after [\\$project](#) stage ⓘ (Sample of 1 document)

```

_id: null
checkAvgLat: 2.808041898492249
checkAvgLong: 2.631917158632406

```

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
}}]
```

\$project

Output after [\\$project](#) stage ⓘ (Sample of 5 documents)

1 {

2 _id: 1

3 }

_id: "Jordan"

_id: "Iraq"

_id: "Israel"

_id: "Saudi Arabia"

_id: "Lebanon"