SESIÓN 01

RETO 1

01. ***Usando la base de datos `tienda`, muestra la descripción de las tablas `articulo`, `puesto` y `venta`.***

```
use tienda;

show databases;

show tables;

describe articulo;
```

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

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	

```
use tienda;

show databases;

show tables;

describe puesto;
```

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

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

```
use tienda;

show databases;

show tables;

describe venta;
```

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

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

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

```
select * from empleado where id_puesto = 4;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap C

id_empleado	id_puesto	nombre	apellido_paterno	apellido_materno	rfc
317	4	Norrie	McGarrie	Hartopp	HDOW177256O16
361	4	Maxy	Udden	Kose	MBZF302490Y06
NULL	NULL	NULL	NULL	NULL	NULL

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

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

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

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

Result 99 | Result 100 | Result 101 | empleado 102 | empleado 103 | empleado 104 | empleado 105 | empl

Output

Action Output

#	Time	Action	Message
135	20:59:52	select * from empleado where id_puesto = 4	2 row(s) returned
136	20:59:52	select * from puesto where salario > 10000	1000 row(s) returned

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

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

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

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

Result 110 | Result 111 | Result 112 | empleado 113 | empleado 114 | empleado 115 | empleado 116 | empleado

Output

Action Output

#	Time	Action	Message
148	21:01:40	select * from puesto where salario > 10000	1000 row(s) returned
149	21:01:40	select * from articulo where precio > 1000 and iva > 100	787 row(s) returned

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

```
27 • select * from venta where id_articulo in (135,963) and id_empleado in (835,369);
```

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

id_venta	id_articulo	id_empleado	clave	fecha
7	963	369	47335-894	2019-06-08 00:00:00
6	135	835	0049-0032	2020-02-03 15:05:27
NULL	NULL	NULL	NULL	NULL

RETO 3

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

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

Result Grid | Filter Rows: | Edit: | Export/Import:

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

PROYECTO

01. **-- Dentro del mismo servidor de bases de datos, conéctate al esquema classicmodels.**
use classicmodels;

02. **-- Dentro de la tabla employees, obtén el apellido de todos los empleados.**
select lastName from employees;

03. **-- Dentro de la tabla employees, obtén el apellido, nombre y puesto de todos los empleados**
select lastName, firstName, jobTitle from employees;

04. **-- Dentro de la tabla employees, obtén todos los datos de cada empleado.**
select * from employees;

05. **-- 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";

06. **-- 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;

07. **-- 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;

08. **-- 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, officeCode from employees where officeCode in (1,2,3);

09. **-- Dentro de la tabla employees, obten el apellido, nombre y puesto de todos los empleados que tengan un puesto distinto a Sales Rep.**
select lastName, firstName, jobTitle from employees where jobTitle != "Sales Rep";

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;

11. **-- Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados cuyo cdigo de oficina sea menor o igual 4.**
select lastName, firstName, officeCode from employees where officeCode <= 4;

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";

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;

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");

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

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");

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 country != "USA" and country != "France";

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

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;

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;

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;

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, customerName, creditLimit from customers order by creditLimit DESC limit 5;

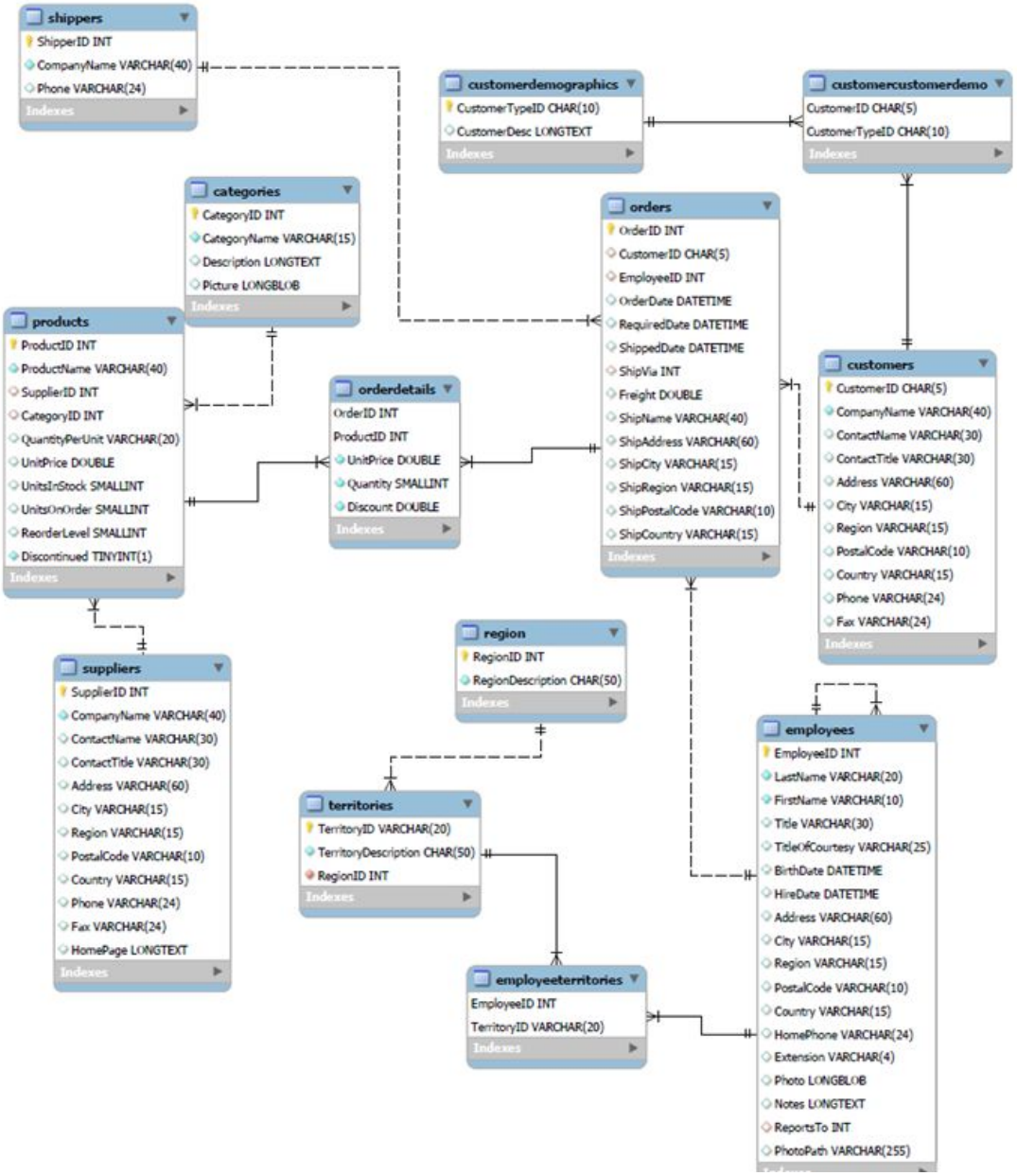
23. **-- Dentro de la tabla customers, obtén el número de cliente, nombre de cliente y el límite de crédito de los cinco clientes con el límite de crédito más bajo.**
select customerNumber, customerName, creditLimit from customers order by creditLimit ASC limit 5;

POSTWORK

01. Definir proyecto del curso.

ANÁLISIS DE VENTAS EMPRESA “ABCD”

02. Diagrama Entidad-Relación



03. Define cinco preguntas que te interesen sobre la base de datos que definiste.

1. ***¿Cuántos empleados tiene la Empresa “ABCD”?***

SELECT count(EmployeeID) FROM employees;

	count(EmployeeID)
▶	9

2. ***¿Qué tipos de productos vende?***

SELECT CategoryName,CategoryID from categories;

	CategoryName	CategoryID
▶	Beverages	1
	Condiments	2
	Confections	3
	Dairy Products	4
	Grains/Cereals	5
	Meat/Poultry	6
	Produce	7
	Seafood	8
*	NULL	NULL

3. ¿Qué puestos tienen los empleados?

SELECT Title FROM employees;

Title
Sales Representative
Vice President, Sales
Sales Representative
Sales Representative
Sales Manager
Sales Representative
Sales Representative
Inside Sales Coordinator
Sales Representative

4. ¿Cuál es el producto más caro a la venta?

SELECT ProductName,UnitPrice FROM products ORDER BY UnitPrice DESC limit 1;

ProductName	UnitPrice
C?te de Blaye	263.5

5. ¿Cuáles son los nombres de los productos de bebidas y cereales?

SELECT ProductName,CategoryID FROM products WHERE CategoryID IN (1,5);

ProductName	CategoryID
Wimmers gute Semmelkn?del	5
Tunnbr?d	5
Steeleye Stout	1
Singaporean Hokkien Fried...	5
Sasquatch Ale	1
Rh?nbr?u Klosterbier	1
Ravioli Angelo	5
Outback Lager	1
Laughing Lumberjack Lager	1
Lakkalik??ri	1
Ipoh Coffee	1
Gustaf's Kn?ckebr?d	5
Guaran? Fant?stica	1
Gnocchi di nonna Alice	5

SESIÓN 02

RETO 1

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

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

```
11 •  select * from articulo where nombre like "%Pasta%";
```

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

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

empleado 17 | empleado 18 | empleado 19 | empleado 20 | articulo 21

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

```
13 •  select * from articulo where nombre like "%Cannelloni%";
```

Result Grid | Filter Rows: | Edit: | Export/Import:

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

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

```
15 •  select * from articulo where nombre like "%-%";
```

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

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 Cktl.	5579.47	1012.33	527
8	Wine - Fontanafredda Barolo	2684.64	327.16	682
9	Lotus Rootlets - Canned	1996.46	324.72	636

empleado 34 | empleado 35 | empleado 36 | empleado 37 | articulo 38 | articulo 39 | arti

RETO 2

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

01. **¿Cuál es el promedio de salario de los puestos?**

```
21 •  select avg(salario) from puesto;
```

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

avg(salario)
19595.051179999973

02. **¿Cuántos artículos incluyen la palabra `Pasta` en su nombre?**

```
23 •  select count(*) from articulo where nombre like "%Pasta%";
```

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

count(*)
17

03. **¿Cuál es el salario mínimo y máximo?**

```
31 •  select min(salario), max(salario) from puesto;
```

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

min(salario)	max(salario)
10013.44	29996.58

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

```
29 •  select sum(salario) from puesto where id_puesto >= 995;
```

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

sum(salario)
98919.69

RETO 3

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

01. ¿Cuántos registros hay por cada uno de los puestos?

```
35 •  select nombre, count(id_puesto) from puesto group by nombre;
```

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

nombre	count(id_puesto)
Analog Circuit Design manager	8
Junior Executive	8
Director of Sales	8
Staff Scientist	9
Desktop Support Technician	5
Budget/Accounting Analyst III	4
Accounting Assistant III	3
Programmer Analyst II	2
Nurse Practicioner	16

Result 413 | Result 414 | Result 415 | Result 416 | Result 417 | Result 418 | Result 419

02. ¿Cuánto dinero se paga en total por puesto?

```
37 •  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 Practicioner	296384.04

Result 456 | Result 457 | Result 458 | Result 459 | Result 460 | Result 461 | Result 462

03. ¿Cuál es el número total de ventas por vendedor?

```
39 •  select id_empleado, count(id_venta) from venta group by id_empleado;
```

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

id_empleado	count(id_venta)
2	2
3	2
4	1
5	1
6	2
12	5
15	4
18	2
22	1

Result 479 | Result 480 | Result 481 | Result 482 | Result 483 | Result 484 | Result 485 | Result 486

04. ¿Cuál es el número total de ventas por artículo?

```
select id_articulo, count(id_venta) from venta group by id_articulo;
```

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

id_articulo	count(id_venta)
2	1
3	1
4	2
8	1
10	1
11	1
12	1
13	2
15	1

Result 503 | Result 504 | Result 505 | Result 506 | Result 507 | Result 508 | Result 509 | Result 510

RETO 4

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

01. ¿Cuál es el nombre de los empleados cuyo sueldo es menor a $20,000?

```
select nombre, apellido_paterno from empleado
where id_puesto in (select id_puesto from puesto where salario < 20000);
```

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
Penny	Dargie
Doy	Pepperrall

Result 1 | Result 2 | Result 3 | Result 4 | Result 5 | Result 6 | empleado 7

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

```
select clave, id_empleado, count(*) total_ventas from venta
group by  clave, id_empleado;
select id_empleado, min(total_ventas), max(total_ventas)
from (select clave, id_empleado, count(*) total_ventas
from venta
group by  clave, id_empleado) as sq group by id_empleado;
-- cuál es el nombre del puesto de cada 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

03. ¿Cuál es el nombre del puesto de cada empleado?

```
select nombre, apellido_paterno, (select nombre from puesto p where p.id_puesto = e.id_puesto)
from empleado as e;
```

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

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

Result 1216 | Result 1217 | Result 1218 | Result 1219 | Result 1220 | Result 1221 | puesto 1222 | empleado 1223 | Result

PROYECTO

01. ***Dentro de la tabla `employees`, obten 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%";
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Ce

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

02. ***Dentro de la tabla `employees`, obten 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";
```

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

employeeNumber	lastName	firstName
NULL	NULL	NULL

03. ***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%";
```

esult Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Conte

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

04. ***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`.***

```
5 •  select employeeNumber, lastName, firstName from employees where firstName like "t_m";
6
```

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

employeeNumber	lastName	firstName
1619	King	Tom
NULL	NULL	NULL

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

```
6 •  select employeeNumber, lastName, firstName from employees where firstName not like "b%";
7
```

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

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
1216	Patterson	Steve
1286	Tseng	Foon Yue

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

```
7 •  select productCode, productName from products where productCode like "%_20%";
8
```

esult Grid | Filter Rows: | Edit: | Export/Import:

productCode	productName
S10_2016	1996 Moto Guzzi 1100i
S18_3320	1917 Maxwell Touring Car
S24_2000	1960 BSA Gold Star DBD34
S24_2011	18th century schooner
S24_2022	1938 Cadillac V-16 Presidential Limousine
S24_3420	1937 Horch 930V Limousine
S24_4620	1961 Chevrolet Impala
S32_2206	1982 Ducati 996 R
S32_3207	1950's Chicago Surface Lines Streetcar
S700_2047	HMS Bounty
NULL	NULL

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

```
8 •  select orderNumber, count(*) from orderdetails group by orderNumber
```

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

orderNumber	count(*)
10100	4
10101	4
10102	2
10103	16
10104	13
10105	15
10106	18
10107	8
10108	16
10109	6
10110	16

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

pendiente

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

```
10 •  select lastName, firstName from employees where officeCode in
11    (select officeCode from offices where country = "USA");
```

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

lastName	firstName
Murphy	Diane
Patterson	Mary
Firrelli	Jeff
Bow	Anthony
Jennings	Leslie
Thompson	Leslie
Firrelli	Julie
Patterson	Steve
Tseng	Foon Yue
Vanauf	George

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

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

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

customerNumber	checkNumber	amount
141	JE105477	120166.58
NULL	NULL	NULL

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

pendiente

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

pendiente

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

```
16 •  select min(quantityOrdered), max(quantityOrdered), avg(quantityOrdered) from orderdetails;
```

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

min(quantityOrdered)	max(quantityOrdered)	avg(quantityOrdered)
6	97	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;
```

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

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

SESIÓN 03

RETO 1

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

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

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

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

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

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

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

esult Grid | Filter Rows: | Export: |

clave	nombre
0002-8149	Sprouts - Pea
0002-8149	Wine - Saint - Bris 2002, Sauv
0002-8149	Sauce Tomato Pouch
0002-8149	Wine - Montecillo Rioja Crianza
0002-8149	Yokaline
0002-8149	Marsala - Sperone, Fine, D.o.c.
0002-8149	Glaze - Clear
0002-8149	Beans - Green

03. ¿Cuál es el total de cada venta?

```
select clave, sum(precio) as total
from venta as v
join articulo as a
on v.id_articulo = a.id_articulo
group by clave
order by clave;
```

Result Grid | Filter Rows: | Export:

clave	total
0002-8149	38642.55000000001
0008-0833	862.6
0009-0347	6879.56
0019-9888	178.57
0023-9177	3891.89
0049-0032	321524.6099999999
0065-0396	5571.99

RETO 2

Usando la base de datos `tienda`, define las siguientes vistas que permitan obtener la siguiente información.

01. Obtener el puesto de un empleado.

```
create view YCMPuesto as
(select concat(e.nombre, " ", e.apellido_paterno) as empleado, p.nombre
from  empleado e
join puesto p
on e.id_puesto = p.id_puesto);
```

02. Saber qué artículos ha vendido cada empleado.

```
create view YCMEmpleado as
(select v.clave, concat(e.nombre, ' ', e.apellido_paterno) as empleado, a.nombre as articulo
from venta v
join empleado e
on v.id_empleado = e.id_empleado
join articulo a
on v.id_articulo = a.id_articulo
order by v.clave);
```

03. Saber qué puesto ha tenido más ventas.

```
create view YCMVenta as
(select p.nombre, count(v.clave) totalventa
from venta v
join empleado e
on v.id_empleado = e.id_empleado
join puesto p
on e.id_puesto = p.id_puesto
group by p.nombre);
```

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;
```

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

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...
S12_2823	2002 Suzuki XREO	Official logos and insignias, saddle bags located ...

products 1

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

```
select o.orderNumber, o.status, sum(od.priceEach) costototal
from orders o
right join orderdetails od
on o.orderNumber = od.orderNumber
group by o.orderNumber;
```

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

orderNumber	status	costototal
10100	Shipped	301.84
10101	Shipped	352.00
10102	Shipped	138.68
10103	Shipped	1520.37
10104	Shipped	1251.89

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, o.orderDate, od.orderLineNumber, p.productName, od.quantityOrdered, od.priceEach
from orders o
right join orderdetails od
on o.orderNumber = od.orderNumber
right join products p
on od.productCode = p.productCode
order by od.orderNumber;
```

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

orderNumber	orderDate	orderLineNumber	productName	quantityOrdered	priceEach
NULL	NULL	NULL	1985 Toyota Supra	NULL	NULL
10100	2003-01-06	3	1917 Grand Touring Sedan	30	136.00
10100	2003-01-06	2	1911 Ford Town Car	50	55.09
10100	2003-01-06	4	1932 Alfa Romeo 8C2300 Spider Sport	22	75.46
10100	2003-01-06	1	1936 Mercedes Benz 500k Roadster	49	35.29
10101	2003-01-09	4	1932 Model A Ford J-Coupe	25	108.06
10101	2003-01-09	1	1928 Mercedes-Benz SSK	26	167.06
10101	2003-01-09	3	1939 Chevrolet Deluxe Coupe	45	32.53
10101	2003-01-09	2	1938 Cadillac V-16 Presidential Limousine	46	44.35
10102	2003-01-10	2	1937 Lincoln Berline	39	95.55
10102	2003-01-10	1	1936 Mercedes-Benz 500K Special Road...	41	43.13
10103	2003-01-29	11	1952 Alpine Renault 1300	26	214.30

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

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

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

orderNumber	productName	MSRP	priceEach
NULL	1985 Toyota Supra	107.57	NULL
10100	1917 Grand Touring Sedan	170.00	136.00
10100	1911 Ford Town Car	60.54	55.09
10100	1932 Alfa Romeo 8C2300 Spider Sport	92.03	75.46
10100	1936 Mercedes Benz 500k Roadster	41.03	35.29
10101	1932 Model A Ford J-Coupe	127.13	108.06
10101	1928 Mercedes-Benz SSK	168.75	167.06
10101	1939 Chevrolet Deluxe Coupe	33.19	32.53
10101	1938 Cadillac V-16 Presidential Limousine	44.80	44.35
10102	1937 Lincoln Berline	102.74	95.55
10102	1936 Mercedes-Benz 500K Special Road...	53.91	43.13
10103	1952 Alpine Renault 1300	214.30	214.30
10103	1962 LanciaA Delta 16V	147.74	119.67

Para estas consultas usa LEFT JOIN

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

```
select c.customerNumber, c.customerName, o.orderNumber, c.state
from customers c
left join orders o
on c.customerNumber = o.customerNumber
order by c.customerNumber;
```

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

customerNumber	customerName	orderNumber	state
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
114	Australian Collectors, Co.	10347	Victoria
119	La Rochelle Gifts	10275	NULL
119	La Rochelle Gifts	10315	NULL

6. ***Obtén los clientes que no tienen una orden asociada.***
   pendiente
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 e.lastName, e.firstName, c.customerName, py.checkNumber, py.amount as total
from employees e
left join customers c
on e.employeeNumber = c.salesRepEmployeeNumber
left join payments py
on c.customerNumber = py.customerNumber
order by e.lastName;
```

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

lastName	firstName	customerName	checkNumber	total
Bondur	Gerard	NULL	NULL	NULL
Bondur	Loui	Saveley & Henriot, Co.	FP549817	40978.53
Bondur	Loui	Saveley & Henriot, Co.	FU793410	49614.72
Bondur	Loui	Saveley & Henriot, Co.	LJ160635	39712.10
Bondur	Loui	La Corne D'abondance, Co.	AD832091	1960.80
Bondur	Loui	La Corne D'abondance, Co.	CE51751	51209.58
Bondur	Loui	La Corne D'abondance, Co.	EH208589	33383.14
Bondur	Loui	Lyon Souveniers	EQ12267	17928.09
Bondur	Loui	Lyon Souveniers	HD284647	26311.63
Bondur	Loui	Lyon Souveniers	HN114306	23419.47
Bondur	Loui	Marseille Mini Autos	BQ602907	18888.31
Bondur	Loui	Marseille Mini Autos	CI471510	50824.66
Bondur	Loui	Marseille Mini Autos	OB648482	1834.56

Result 11

Para estas consultas usa RIGHT JOIN

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

```
select c.customerNumber, c.customerName, o.orderNumber, c.state
from customers c
right join orders o
on c.customerNumber = o.customerNumber
order by c.customerNumber;
```

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

customerNumber	customerName	orderNumber	state
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
114	Australian Collectors, Co.	10347	Victoria
119	La Rochelle Gifts	10275	NULL
119	La Rochelle Gifts	10315	NULL

5.
6. ***pendiente***

```
select e.lastName, e.firstName, c.customerName, py.checkNumber, py.amount as total
from employees e
right join customers c
on e.employeeNumber = c.salesRepEmployeeNumber
right join payments py
on c.customerNumber = py.customerNumber
order by e.lastName;
```

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

lastName	firstName	customerName	checkNumber	total
Bondur	Loui	Saveley & Henriot, Co.	FP549817	40978.53
Bondur	Loui	Saveley & Henriot, Co.	FU793410	49614.72
Bondur	Loui	Saveley & Henriot, Co.	LJ160635	39712.10
Bondur	Loui	La Corne D'abondance, Co.	AD832091	1960.80
Bondur	Loui	La Corne D'abondance, Co.	CE51751	51209.58
Bondur	Loui	La Corne D'abondance, Co.	EH208589	33383.14
Bondur	Loui	Lyon Souveniers	EQ12267	17928.09
Bondur	Loui	Lyon Souveniers	HD284647	26311.63
Bondur	Loui	Lyon Souveniers	HN114306	23419.47
Bondur	Loui	Marseille Mini Autos	BQ602907	18888.31
Bondur	Loui	Marseille Mini Autos	CI471510	50824.66

7.

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

```
create view YCMEstatus as
(select o.orderNumber, o.status, sum(od.priceEach) costototal
from orders o
right join orderdetails od
on o.orderNumber = od.orderNumber
group by o.orderNumber);
```

- YCMEstatus
  - orderNumber
  - status
  - costototal

```
create view YCMPrecios as
(select od.orderNumber, p.productName, p.MSRP, od.priceEach
from orderdetails od
right join products p
on od.productCode = p.productCode
order by od.orderNumber);
```

- YCMPrecios
  - orderNumber
  - productName
  - MSRP
  - priceEach

```
create view YCMSalesRep as
(select e.lastName, e.firstName, c.customerName, py.checkNumber, py.amount as total
from employees e
left join customers c
on e.employeeNumber = c.salesRepEmployeeNumber
left join payments py
on c.customerNumber = py.customerNumber
order by e.lastName);
```

- YCMSalesRep
  - lastName
  - firstName
  - customerName
  - checkNumber
  - total

SESIÓN 04

RETO 1

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

1. ***Fecha, nombre y texto de cada comentario.***
   { date: 1, name: 1, text: 1, _id: 0 }

```
name: "Andrea Le"
text: "Rem officiis eaque repellendus amet eos doloribus. Porro dolor volupta..."
date: 2012-03-26T23:20:16.000+00:00
```

```
name: "Greg Powell"
text: "Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Co..."
date: 1987-02-10T00:29:36.000+00:00
```

```
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
```

```
name: "Cameron Duran"
text: "Quasi dicta culpa asperiores quaerat perferendis neque. Est animi pari..."
date: 1983-04-27T20:39:15.000+00:00
```

2. ***Título, elenco y año de cada película.***
   { title: 1, cast: 1, year: 1, _id: 0}

```
▸ cast: Array
  title: "Blacksmith Scene"
  year: 1893
```

```
▸ cast: Array
  title: "The Great Train Robbery"
  year: 1903
```

```
▸ cast: Array
  title: "The Land Beyond the Sunset"
  year: 1912
```

```
▸ cast: Array
  title: "A Corner in Wheat"
  year: 1909
```

3. ***Nombre y contraseña de cada usuario.***
   {name:1,password:1}

```
_id: ObjectId("59b99db4cfa9a34dcd7885b6")
name: "Ned Stark"
password: "$2b$12$UREFwsRUoyF0CRqGNK0LzO0HM/jLhgUCNNIJ9RJAqMUQ74crlJ1Vu"
```

```
_id: ObjectId("59b99db4cfa9a34dcd7885b7")
name: "Robert Baratheon"
password: "$2b$12$yGqxLG9LZpXA2xVDhuPnSOZd.VURVkz7wgOLY3pnO0s7u2S1ZO32y"
```

```
_id: ObjectId("59b99db5cfa9a34dcd7885b8")
name: "Jaime Lannister"
password: "$2b$12$6vz7wiwO.EI5Rilvq1zUc./9480gb1uPtXcahDxIadgyC3PS8XCUK"
```

```
_id: ObjectId("59b99db5cfa9a34dcd7885b9")
name: "Catelyn Stark"
password: "$2b$12$fiaTH5Sh1zKNFX2i/FTEreWGjxoJxvmV7XL.qlfqCr8CwOxK.mZWS"
```

RETO 2

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

1. ***¿Qué comentarios ha hecho Greg Powell?***
   {name:"Greg Powell"}

```
> _id: ObjectId("5a9427648b0beebeb69579cf")
  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("5a9427648b0beebeb6957afe")
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
```

```
_id: ObjectId("5a9427648b0beebeb6957b56")
name: "Greg Powell"
email: "greg_powell@fakegmail.com"
movie_id: ObjectId("573a1391f29313caabcd7e5d")
text: "Officia atque ullam esse doloribus laborum. Maiores dicta ratione rem ..."
date: 2004-04-08T08:21:05.000+00:00
```

2. ***¿Qué comentarios han hecho Greg Powell o Mercedes Tyler?***
   {$or:[{name:"Greg Powell"},{name:"Mercedes Tyler"}]}

```
_id: ObjectId("5a9427648b0beebeb69579cf")
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("5a9427648b0beebeb69579e7")
name: "Mercedes Tyler"
email: "mercedes_tyler@fakegmail.com"
movie_id: ObjectId("573a1390f29313caabcd4323")
text: "Eius veritatis vero facilis quaerat fuga temporibus. Praesentium exped..."
date: 2002-08-18T04:56:07.000+00:00
```

```
_id: ObjectId("5a9427648b0beebeb6957a78")
name: "Mercedes Tyler"
email: "mercedes_tyler@fakegmail.com"
movie_id: ObjectId("573a1390f29313caabcd6399")
text: "Voluptate odio minima pariatur recusandae. Architecto illum dicta repu..."
date: 2007-10-17T06:50:56.000+00:00
```

3. ***¿Cuál es el máximo número de comentarios en una película?***
   {num_mflix_comments: 1}
   {num_mflix_comments:-1}
   Limit: 1

```
_id: ObjectId("573a1399f29313caabcee886")
num_mflix_comments: 456
```

4. ***¿Cuál es título de las cinco películas más comentadas?***
   {title: 1}
   {num_mflix_comments: -1}
   Limit:5

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

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

```
_id: ObjectId("573a13bff29313caabd6001f")
title: "The Unborn"
```

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

```
_id: ObjectId("573a13a7f29313caabd1aa55")
title: "8 Mile"
```

PROYECTO

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 `Toogle Query History`. Busca la última consulta y agregala a favoritos presionando el íncono con la estrella ★.

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

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

SORT MAXTIMEMS 5000

COLLATION SKIP 0 LIMIT 0

VIEW { } Displaying documents 1 - 20 of 9500

```
_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"

_id: ObjectId("52cdef7c4bab8bd675297d94")
name: "Twitter"
email_address: "press@twitter.com"
```

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

2. ***Obtén la fuente de cada tweet.***

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..."
```

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

3. ***Obtén el nombre de todas las compañias fundadas en octubre.***

FILTER {founded_month:10} OPTIONS FIND RESET ...
PROJECT {name:1}
SORT MAXTIMEMS 5000
COLLATION SKIP 0 LIMIT 0

VIEW { } Displaying documents 1 - 20 of 301 REFRESH

```
_id: ObjectId("52cdef7c4bab8bd675297d8a")
name: "Wetpaint"

_id: ObjectId("52cdef7c4bab8bd675297da4")
name: "Powerset"

_id: ObjectId("52cdef7c4bab8bd675297dff")
name: "TheFind"

_id: ObjectId("52cdef7c4bab8bd675297e1e")
name: "TechnologyGuide"

_id: ObjectId("52cdef7c4bab8bd675297e22")
name: "Techmeme"
```

```
{
 filter: {
  founded_month: 10
 },
 project: {
  name: 1
 }
}
```

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

FILTER {founded_year:2008} OPTIONS FIND RESET ...
PROJECT {name:1}
SORT MAXTIMEMS 5000
COLLATION SKIP 0 LIMIT 0

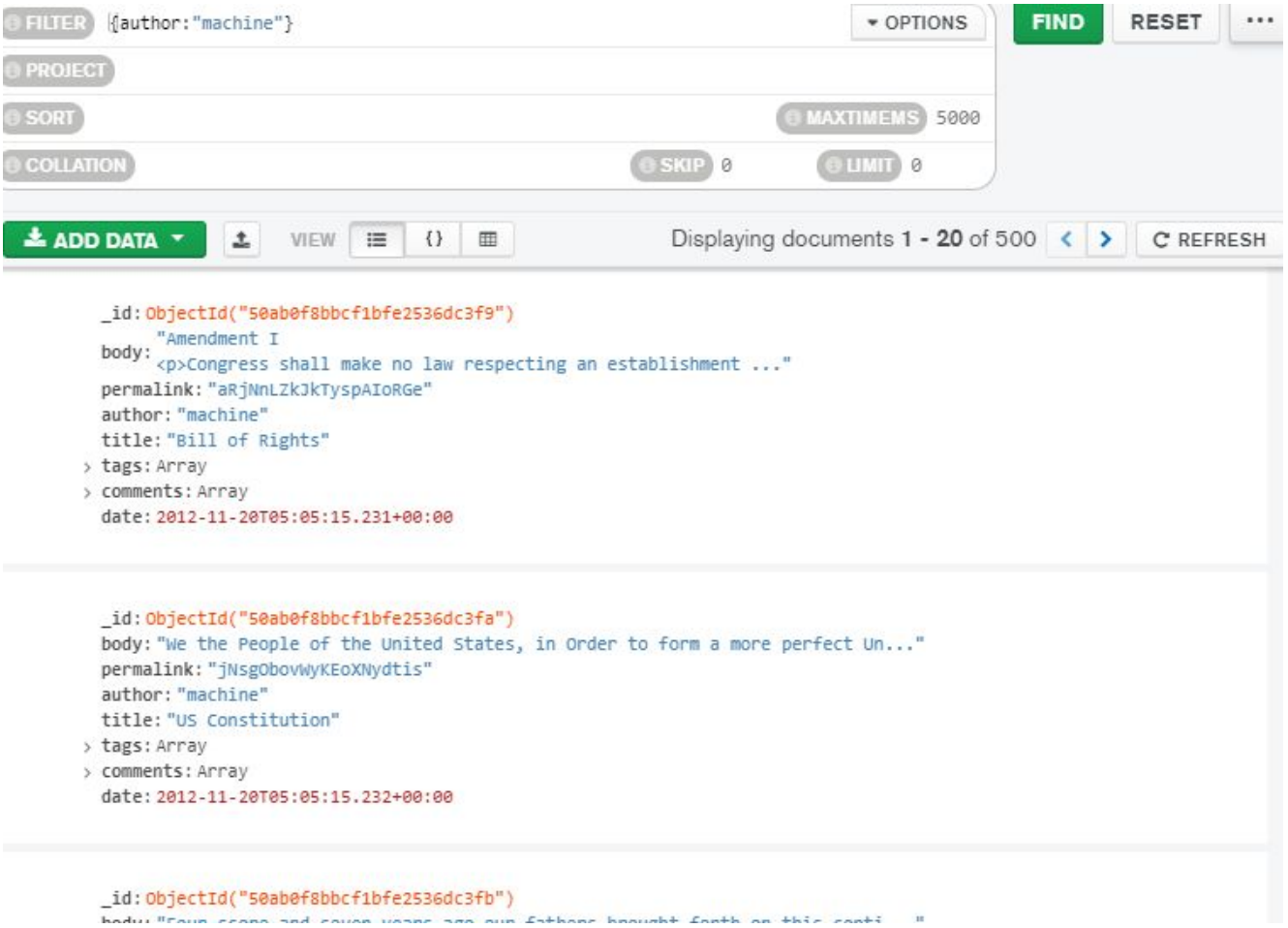
VIEW { } Displaying documents 1 - 20 of 1224 REFRESH

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

_id: ObjectId("52cdef7c4bab8bd675298218")
name: "WonderHowTo"

_id: ObjectId("52cdef7c4bab8bd675298232")
name: "First30Days"

_id: ObjectId("52cdef7c4bab8bd675298244")
name: "Mibura"

_id: ObjectId("52cdef7c4bab8bd675298259")
name: "Tvosz"
```

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

```
 project: {
  name: 1
 }
}
```

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




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

6. ***Obtén todos los tweets provenientes de la `web`.***

FILTER {source:"web"} OPTIONS FIND RESET ...
PROJECT
SORT MAXTIMEMS 5000
COLLATION SKIP 0 LIMIT 0

ADD DATA VIEW Displaying documents 1 - 20 of 11141 REFRESH

```
_id: ObjectId("5c8eccb0caa187d17ca623f5")
text: "eu preciso de terminar de fazer a minha tabela, está muito foda **"
in_reply_to_status_id: null
retweet_count: null
contributors: null
created_at: "Thu Sep 02 18:11:23 +0000 2010"
geo: null
source: "web"
coordinates: null
in_reply_to_screen_name: null
truncated: false
> entities: Object
retweeted: false
place: null
> user: Object
favorited: false
in_reply_to_user_id: null
id: 22819396900
```

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

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

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

FILTER {$and:[{founded_month:10},{founded_year:2008}]} OPTIONS FIND RESET ...
PROJECT
SORT MAXTIMEMS 5000
COLLATION SKIP 0 LIMIT 0

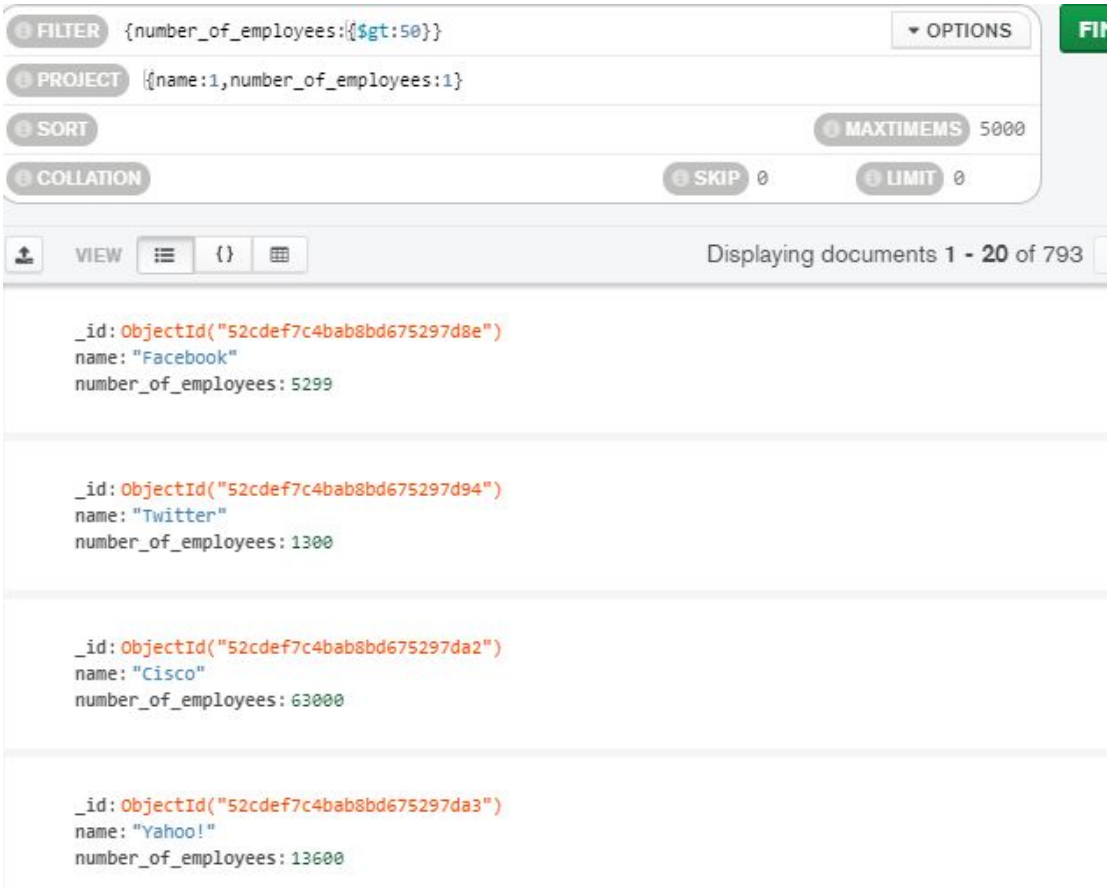
ADD DATA VIEW Displaying documents 1 - 20 of 63 REFRESH

```
_id: ObjectId("52cdef7c4bab8bd6752985ca")
name: "tunesBag"
permalink: "tunesbag"
crunchbase_url: "http://www.crunchbase.com/company/tunesbag"
homepage_url: "http://www.tunesBag.com"
blog_url: "http://tunesBag.blogspot.com/"
blog_feed_url: "http://blog.tunesbag.com/feeds/posts/default?alt=rss"
twitter_username: ""
category_code: "games_video"
number_of_employees: null
founded_year: 2008
founded_month: 10
founded_day: 1
deadpooled_year: null
deadpooled_month: null
deadpooled_day: null
deadpooled_url: null
tag_list: "music, cloud, locker, mp3, music-streaming, streaming"
alias_list: ""
email_address: "office@tunesBag.com"
phone_number: "+43 680 215 27 96"
description: "Social Music Player"
```

```
{
 filter: {
  $and: [
   {
```

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

8. Obtén todas las compañias con más de 50 empleados.




```
{
 filter: {
  number_of_employees: {
   $gt: 50
  }
 },
 project: {
  name: 1,
  number_of_employees: 1
 }
}
```

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

FILTER {$and:[{comments:{$gte:10}},{comments:{$lte:30}}]} OPTIONS FIND

PROJECT {title:1,comments:1}

SORT MAXTIMEMS 5000

COLLATION SKIP 0 LIMIT 0

VIEW {} Displaying documents 1 - 20 of 1930

```
_id: ObjectId("4ba267dc238d3ba3ca000006")
title: "11 Amazing Treehouses from Around the World"
comments: 15
```

```
_id: ObjectId("4ba267dc238d3ba3ca00000b")
title: "NASA - The Wizard Nebula "
comments: 14
```

```
_id: ObjectId("4ba267dc238d3ba3ca00000c")
title: "WISE Captures a Cosmic Rose"
comments: 12
```

```
_id: ObjectId("4ba267dc238d3ba3ca000012")
title: "Chris Parnell, Horatio Sanz Cast in Ferrell/Heder Sitcom "
comments: 22
```

```
{
 filter: {
  $and: [
   {
    comments: {
     $gte: 10
    }
   },
   {
    comments: {
     $lte: 30
    }
   }
  ]
 },
 project: {
  title: 1,
  comments: 1
 }
}
```

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

FILTER {$and:[{number_of_employees:{$ne:null}},{number_of_employees:{$ne:0}}]} OPTIONS FIN

PROJECT {name:1,number_of_employees:1}

SORT {number_of_employees:1} MAXTIMEMS 5000

COLLATION SKIP 0 LIMIT 1

VIEW {} Displaying documents 1 - 1 of 1

```
_id: ObjectId("52cdef7c4bab8bd675297e60")
name: "FeVote"
number_of_employees: 1
```

```
{
```

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

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

```
{
 project: {
  name: 1,
  number_of_employees: 1
 },
 sort: {
  number_of_employees: -1
 },
 limit: 1
}
```

12. Obtén la historia más comentada.

```
{
 project: {
  title: 1,
  comments: 1
 },
 sort: {
  comments: -1
 },
```

limit: 1
}

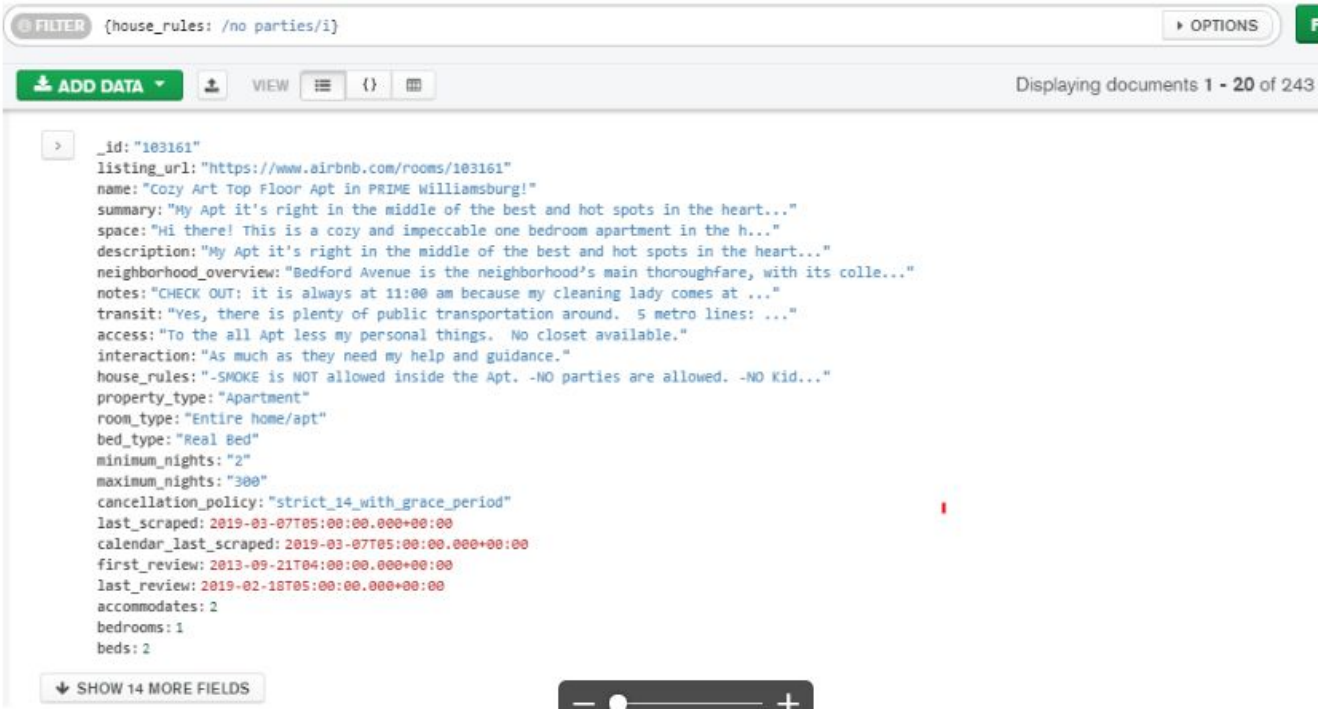
13. Obtén la historia menos comentada.

{
project: {
title: 1,
comments: 1
},
sort: {
comments: 1
},
limit: 1
}

SESIÓN 05

RETO 1

1. **Propiedades que no permitan fiestas.**



2. **Propiedades que admitan mascotas.**

FILTER {house_rules: /pets allowed/i} OPTIONS FIND

ADD DATA VIEW Displaying documents 1 - 20 of 59


```
_id: "11197217"
listing_url: "https://www.airbnb.com/rooms/11197217"
name: "2 Bedroom Ipanema-Copacabana"
summary: "Our apartment is strategically located between the beaches of Copacaba..."
space: ""
description: "Our apartment is strategically located between the beaches of Copacaba..."
neighborhood_overview: ""
notes: ""
transit: ""
access: ""
interaction: ""
house_rules: "We would appreciate if the guest, as leaving the apartment even for a ..."
property_type: "Apartment"
room_type: "Entire home/apt"
bed_type: "Real Bed"
minimum_nights: "4"
maximum_nights: "30"
cancellation_policy: "moderate"
last_scraped: 2019-02-11T05:00:00.000+00:00
calendar_last_scraped: 2019-02-11T05:00:00.000+00:00
first_review: 2016-08-23T04:00:00.000+00:00
last_review: 2018-02-15T05:00:00.000+00:00
accommodates: 5
bedrooms: 2
beds: 5
```


SHOW 14 MORE FIELDS

3. ***Propiedades que no permitan fumadores.***

FILTER {house_rules: /no smoking/i} OPTIONS

ADD DATA VIEW Displaying documents 1 - 20 of 639


```
_id: "1003530"
listing_url: "https://www.airbnb.com/rooms/1003530"
name: "New York City - Upper West Side Apt"
summary: ""
space: "Murphy bed, optional second bedroom available. Wifi available, Hulu, N..."
description: "Murphy bed, optional second bedroom available. Wifi available, Hulu, N..."
neighborhood_overview: "Great neighborhood - many terrific restaurants, bakeries, begelries. W..."
notes: "My cat, Samantha, are in and out during the summer.  The apt is layed ..."
transit: "Conveniently located near 1, 2, 3, B & C subway lines. Also buses on C..."
access: "New York City!"
interaction: ""
house_rules: "No smoking is permitted in the apartment. All towels that are used sho..."
property_type: "Apartment"
room_type: "Private room"
bed_type: "Real Bed"
minimum_nights: "12"
maximum_nights: "360"
cancellation_policy: "strict_14_with_grace_period"
last_scraped: 2019-03-07T05:00:00.000+00:00
calendar_last_scraped: 2019-03-07T05:00:00.000+00:00
first_review: 2013-04-29T04:00:00.000+00:00
last_review: 2018-08-12T04:00:00.000+00:00
accommodates: 2
bedrooms: 1
beds: 1
```


SHOW 14 MORE FIELDS

4. ***Propiedades que no permitan fiestas ni fumadores.***

RETO 2

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

FILTER { number_of_reviews: {$gte: 50}, "review_scores.review_scores_rating": {$gte: 80}, amenities: {$in: [/Ethernet/i]}, "add

OPTIONS

ADD DATA | VIEW | Displaying documents 1 - 6 of 6


```
_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.O apartamento situa-se em edifício 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

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).

$match

```
/**
 * query: The query in MQL.
 */
{
  amenities:{$in:["Wifi","Ethernet"]}
}
```

Output after $match stage (Sample of 20 documents)


```
_id: "10006546"
listing_url: "https://www.airbnb.com/rooms/10006546"
name: "Ribeira Charming Duplex"
summary: "Fantastic duplex apartment with three bedroom
located in the histori..."
space: "Privileged views of the Douro River and Ribeira
square, our apartment ..."
description: "Fantastic duplex apartment with three bed
located in the histori..."
```

```
_id: "10009999"
listing_url: "https://www.airbnb.com/rooms/
name: "Horto flat with small garden"
summary: "One bedroom + sofa-bed in quiet a
neighbourhood right next t..."
space: "Lovely one bedroom + sofa-bed in th
perfect for two but ..."
description: "One bedroom + sofa-bed in qui
neighbourhood right next t..."
```


$group

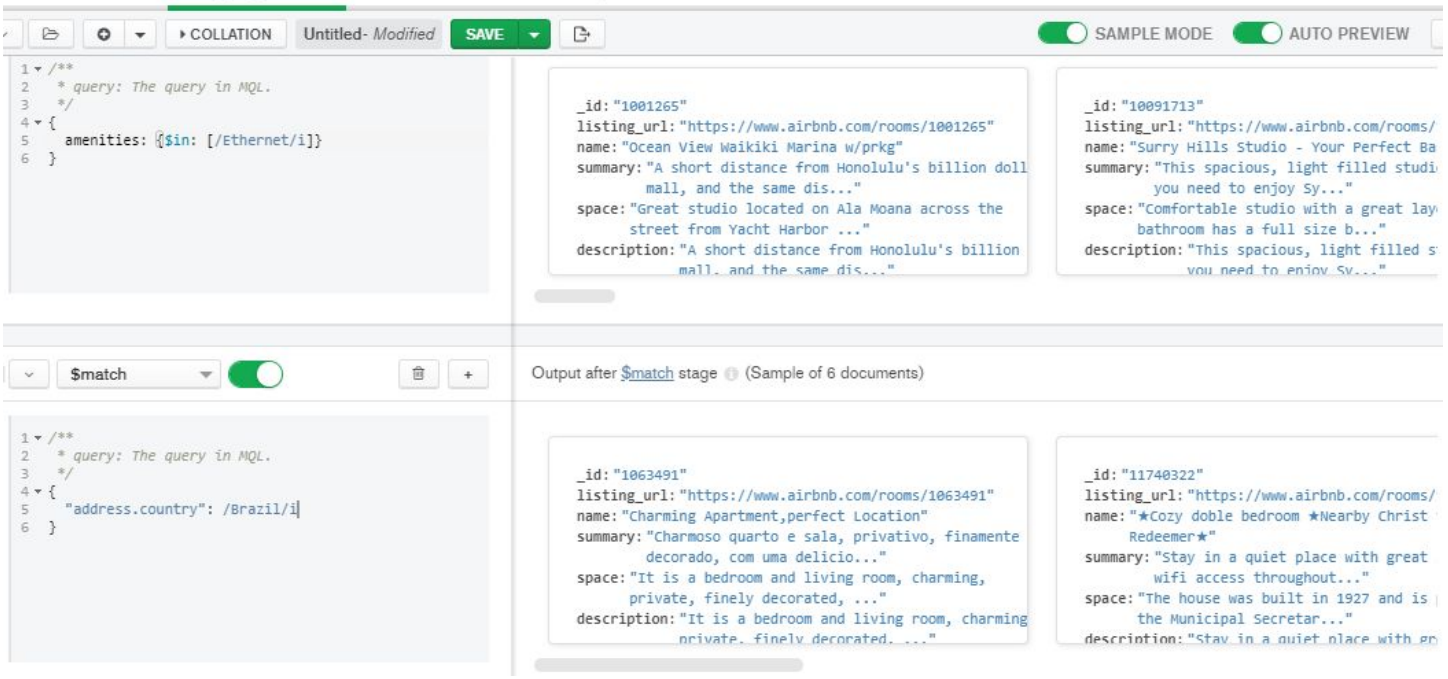
```
/**
 * _id: The id of the group.
 * fieldN: The first field name.
 */
{
  _id:null,
  total: {
    $sum:1
  }
}
```

Output after $group stage (Sample of 1 document)


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


PROYECTO

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.




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

SESIÓN 06

EJEMPLO1

```
[{$match: {
  property_type:"House",
  bedrooms:{$gte:1}
}}, {$addFields: {
  costo_recamara: {$divide: ["$price", "$bedrooms"]}
}}, {$group: {
  _id: null,
  recamaras: {
    $sum:1
  },
  total:{$sum:"$costo_recamara"
  }
}}]
```

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",
  propiedades: {
    $sum: 1
  },
  total: {
```

```
    $sum: '$costo_recamara'
  }
}}, {$addFields: {
  costo_promedio: {
    $divide: [
      '$total',
      '$propiedades'
    ]
  }
}}, {$project: {
  costo_promedio: 1
}}, {$sort: {
  _id: 1
}}]
```

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.

NO CIERES ESTE *PIPELINE PUES LO USAREMOS MÁS ADELANTE*

```
[{$lookup: {
  from: 'users',
  localField: 'email',
  foreignField: 'email',
  as: 'users'
}}, {$addFields: {
  users_objeto:{$arrayElemAt:["$users",0]}
}}, {$addFields: {
  password: "$users_objeto.password"
}}, {$project: {
  password:1,
  email:1,
  name:1,
  _id:0
}}, {$count: '1'}]
```

RETO 3

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

sample_mflix.4YCM (view on: sample_mflix.comments) MODIFY SOURCE

Documents | Aggregations | Schema | Explain Plan | Indexes | Validation

FILTER

VIEW { } Displaying documents

```
name: "Andrea Le"
email: "andrea_le@fakegmail.com"
password: "$2b$12$JS87HWuL2y0P1E6kYrcbKOKx22.wsKEdLtS0F734/vKdhuduLM8Ve"

name: "Greg Powell"
email: "greg_powell@fakegmail.com"
password: "$2b$12$XpveUB6kIiU3zG5aABw26OitIB7cDBbSUWJAz4WDF4XXyNNJ/mp76"

name: "Talisa Maegyr"
email: "oona_chaplin@gameofthron.es"
password: "$2b$12$B3DKrkTVNJDlgToXClpcwOJvnKICUsLcrTxNfZIeoqbHIQURsNrXS"

name: "Cameron Duran"
email: "cameron_duran@fakegmail.com"
password: "$2b$12$50w2j63ATGmhVOh2rgdjv.wOd9TV0Jb9Xk/Anms0fxVSvGMf5MwvK"

name: "Petyr Baelish"
email: "aidan_gillen@gameofthron.es"
password: "$2b$12$qM.YvmiekyYYY7p7phpK3OicbRCDkN7ESwYAnG/o9YnfHC0Mhkmbi"
```

sample_airbnb.4YCM_RETO03 (view on: sample_airbnb.listingsAndReviews) MODIFY SOURCE

Documents | Aggregations | Schema | Explain Plan | Indexes | Validation

FILTER OPTIONS

VIEW { } Displaying documents 1 - 9 of 9

```
_id: "Australia"
costo: 107.8

_id: "Brazil"
costo: 348.7

_id: "Canada"
costo: 115.0

_id: "China"
costo: 393.0

_id: "Hong Kong"
costo: 514.6

_id: "Portugal"
costo: 40.0
```

PROYECTO

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: