Andrea Valdez Hernández

Sesión 1 - SQL

Código

#si no sabes que bases de datos contiene el documento
show databases;

#decir cual base de dato vas a utilizar
use tienda;

#ver las tablas
show tables;

#ver la estructura de la tabla
describe empleado;
#int son numeros enteros
#varchar() son textos en el parentesis se pone la longitud llega hasta 250 caracteres
#imágenes es mejor no guardarlos, mejor las ligas

Reto 1

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.**

describe articulo;
describe puesto;
describe venta;

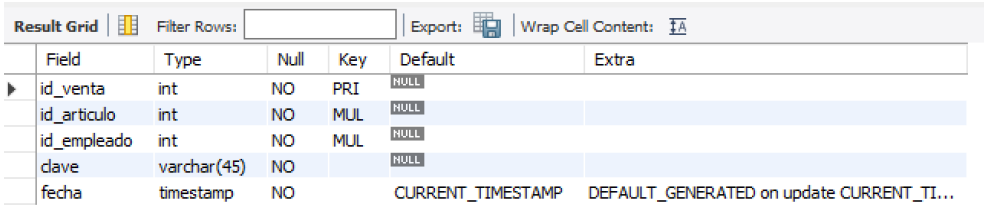
```
13 •  describe articulo;
14 •  describe puesto;
15 •  describe venta;
```

Result Grid | Filter Rows: | Export: | Wr

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	

Result Grid | Filter Rows: | Export: | Wr

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	



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...

```
#seleccionar datos de una tabla en especifico
select nombre from empleado;

#para todos los campos de la tabla
select * from empleado;

#se usa WHERE para filtrar
#para que te de un resultado deseado
select *
from empleado
where apellido_paterno = 'Risom';

#para que sea mayor a algo deseado
select *
from empleado
where id_puesto > 100;

#se usa ORDER BY para ordenar
SELECT *
FROM empleado
WHERE id_puesto >= 100
        AND id_puesto <= 200
ORDER BY id_puesto ASC;

SELECT *
FROM empleado
WHERE id_puesto = 100
        OR id_puesto = 200;
#otra forma de decirlo sería
SELECT *
FROM empleado
WHERE id_puesto IN (100, 200, 300);
```

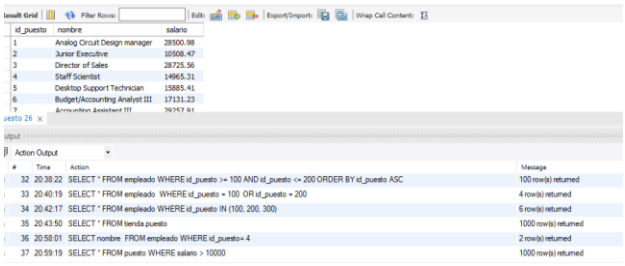
Reto 2

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

```
SELECT nombre
FROM empleado
WHERE id_puesto     = 4;
```

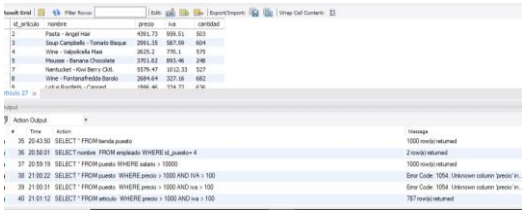
2. **¿Qué puestos tienen un salario mayor a $10,000?**

```
SELECT *
FROM puesto
WHERE salario > 10000;
```



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

```
SELECT *
FROM articulo
WHERE precio > 1000
        AND iva > 100;
```



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

```
SELECT *
FROM venta
WHERE id_articulo IN (135, 963)
        AND id_empleado IN (835, 369);
```


```
#para ordenar descendente
SELECT *
FROM puesto
ORDER BY salario DESC;

#para ordenar ascendente puede llevar o no ASC
SELECT *
FROM puesto
ORDER BY salario;
```

```
SELECT *
FROM puesto
ORDER BY nombre;

#es para limitar el numero de resultados
SELECT *
FROM empleado
LIMIT 5;
```

Reto 3

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

```
SELECT *
FROM puesto
ORDER BY salario DESC
LIMIT 5;
```

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

#proyecto

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

```
USE classicmodels;
```

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

```
SELECT lastName
FROM employees;
```

lastName
Murphy
Patterson
Firrelli
Patterson
Bondur
Bow
Jennings
Thompson
Firrelli
Patterson
Tseng
Vanauf
Bondur
Hernandez

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
Patterson	Steve	Sales Rep
Tseng	Foon Yue	Sales Rep
Vanauf	George	Sales Rep
Bondur	Loui	Sales Rep
Hernandez	Gerard	Sales Rep

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

SELECT *
FROM employees;

employeeNumber	lastName	firstName	extension	email	officeCode	reportsTo	jobTitle
1002	Murphy	Diane	x5800	dmurphy@classicmodelcars.com	1	NULL	President
1056	Patterson	Mary	x4611	mpatterso@classicmodelcars.com	1	1002	VP Sales
1076	Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com	1	1002	VP Marketing
1088	Patterson	William	x4871	wpatterson@classicmodelcars.com	6	1056	Sales Manager (APAC)
1102	Bondur	Gerard	x5408	gbondur@classicmodelcars.com	4	1056	Sale Manager (EMEA)
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
1323	Vanauf	George	x4102	gvanauf@classicmodelcars.com	3	1143	Sales Rep
1337	Bondur	Loui	x6493	lbondur@classicmodelcars.com	4	1102	Sales Rep
1370	Hernandez	Gerard	x2028	ghernande@classicmodelcars.com	4	1102	Sales Rep

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

SELECT lastName, firstName, jobTitle
FROM employees
WHERE jobTitle = 'Sales Rep';

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
Bott	Larry	Sales Rep
Jones	Barry	Sales Rep
Fixter	Andy	Sales Rep
Marsh	Peter	Sales Rep
King	Tom	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`.**

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`.**

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
Tseng	Foon Yue	Sales Rep	3
Vanauf	George	Sales Rep	3
Bondur	Loui	Sales Rep	4
Hernandez	Gerard	Sales Rep	4
Castillo	Pamela	Sales Rep	4
Bott	Larry	Sales Rep	7

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

```
SELECT lastName, firstName, officeCode
FROM employees
WHERE officeCode IN (1, 2, 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
Vanauf	George	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`.**

```
SELECT lastName, firstName, jobTitle
FROM employees
WHERE 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`.**

```
SELECT lastName, firstName, officeCode
FROM employees
WHERE officeCode > 5;
```

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 cdigo de oficina sea menor o igual 4.**

```
SELECT lastName, firstName, officeCode
FROM employees
WHERE officeCode <= 4;
```

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
Tseng	Foon Yue	3
Vanauf	George	3
Bondur	Loui	4
Hernandez	Gerard	4
Castillo	Pamela	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';
```

customerName	country	stat
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
West Coast Collectables Co.	USA	CA
Signal Collectibles Ltd.	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`.**

```
SELECT customerName, 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`.**

```
SELECT customerName, country
FROM customers
WHERE country IN ('USA', '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
Muscle Machine Inc	USA
Diecast Classics Inc.	USA
Technics Stores Inc.	USA
American Souvenirs Inc	USA
Daedalus Designs Imports	France
La Corne D'abondance, Co.	France
Cambridge Collectables Co.	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.**

```
SELECT customerName, country, state, creditLimit
FROM customers
WHERE country IN ('USA', 'France')
        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

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');
```

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`.**

```
SELECT officeCode, city, phone, country
FROM offices
WHERE country NOT IN ('USA', '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
NULL	NULL	NULL	NULL

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);
```

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

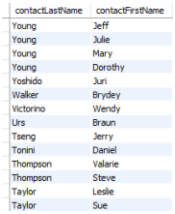
19. **Dentro de la tabla `customers`, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma ascendente.**

```
SELECT contactLastName, contactFirstName
FROM customers
ORDER BY contactLastName;
```

contactLastName	contactFirstName
Accorti	Paolo
Altagar,G M	Raanan
Andersen	Mel
Anton	Carmen
Ashworth	Rachel
Barajas	Miguel
Benitez	Violeta
Bennett	Helen
Berglund	Christina
Bergulfsen	Jonas
Bertrand	Marie
Brown	Julie
Brown	Ann
Brown	William

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

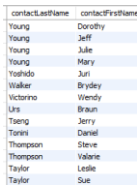
```
SELECT contactLastName, contactFirstName
FROM customers
ORDER BY contactLastName DESC;
```



contactLastName	contactFirstName
Young	Jeff
Young	Julie
Young	Mary
Young	Dorothy
Yoshido	Juri
Walker	Brydey
Victorino	Wendy
Urs	Braun
Tseng	Jerry
Tonini	Daniel
Thompson	Valarie
Thompson	Steve
Taylor	Leslie
Taylor	Sue

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.**

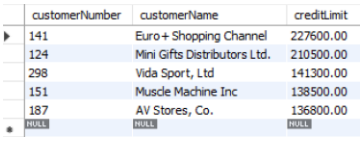
```
SELECT contactLastName, contactFirstName
FROM customers
ORDER BY contactLastName DESC, contactFirstName;
```



contactLastName	contactFirstName
Young	Dorothy
Young	Jeff
Young	Julie
Young	Mary
Yoshido	Juri
Walker	Brydey
Victorino	Wendy
Urs	Braun
Tseng	Jerry
Tonini	Daniel
Thompson	Steve
Thompson	Valarie
Taylor	Leslie
Taylor	Sue

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).**

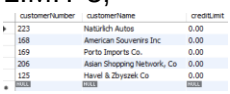
```
SELECT customerNumber, customerName, creditLimit
FROM customers
ORDER BY creditLimit DESC
LIMIT 5;
```



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

```
SELECT customerNumber, customerName, creditLimit
FROM customers
ORDER BY creditLimit
LIMIT 5;
```



customerNumber	customerName	creditLimit
223	Natürlich Autos	0.00
168	American Souvenirs Inc	0.00
169	Porto Imports Co.	0.00
206	Asian Shopping Network, Co	0.00
125	Havel & Zbyszek Co	0.00
NULL	NULL	NULL

Sesión 2

Código

```
#empieza con M
SELECT *
FROM empleado
WHERE nombre LIKE 'M%';

#termina con a
SELECT *
FROM empleado
WHERE nombre LIKE '%A';

#empeiza con m y termina con a
SELECT *
FROM empleado
WHERE nombre LIKE 'M%A';

#_ solo busca una letra
SELECT *
FROM empleado
WHERE nombre LIKE 'M_losa';

SELECT *
FROM empleado
WHERE nombre LIKE 'klara';
```

Reto 1

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

```
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

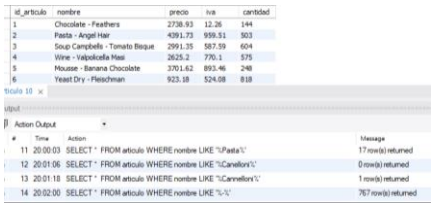
2. **¿Qué artículos incluyen la palabra `Cannelloni` en su nombre?**

```
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

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

```
SELECT *
FROM articulo
WHERE nombre LIKE '%-%';
```



```
#agrupamiento
SELECT (1 + 7) * (10/ (6 -4));

SELECT AVG(a.precio) AS precio_promedio
FROM articulo a;

SELECT COUNT(*)
FROM empleado
WHERE nombre <> 'Klara';

SELECT MAX(precio)
FROM articulo;

SELECT MIN(precio)
FROM articulo;

SELECT SUM(precio)
FROM articulo;
```

Reto 2

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

```
SELECT AVG(salario)
FROM puesto;
```

AVG(salario)
19595.051179999973

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

```
SELECT COUNT(nombre)
FROM articulo
WHERE nombre LIKE '%pasta%';
```

COUNT(nombre)
17

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

```
SELECT MAX(salario) AS salario_maximo, MIN(salario) AS salario_minimo
FROM puesto;
```

salario_maximo	salario_minimo
29996.58	10013.44

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

```
SELECT MAX(id_puesto) - 5
FROM puesto;

SELECT SUM(salario)
FROM puesto
WHERE id_puesto >= 995;
```

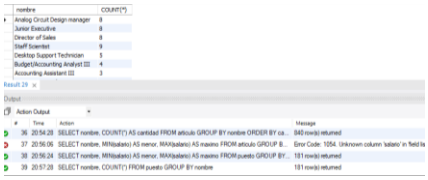
SUM(salario)
98919.69

```
#otra opcion
SELECT SUM(salario)
FROM puesto
WHERE id_puesto >= (SELECT MAX(id_puesto) - 5
FROM puesto);

SELECT nombre, SUM(precio) AS total
FROM articulo
GROUP BY nombre;

SELECT *
FROM articulo;

SELECT nombre, MIN(salario) AS menor, MAX(salario) AS maximo
FROM puesto
GROUP BY nombre;

SELECT nombre, COUNT(*) AS cantidad
FROM articulo
GROUP BY nombre
ORDER BY cantidad DESC;
```

Reto 3

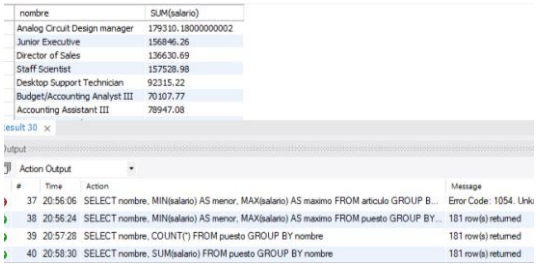
1. **¿Cuántos registros hay por cada uno de los puestos?**

```
SELECT nombre, COUNT(*)
FROM puesto
GROUP BY nombre;
```



2. **¿Cuánto dinero se paga en total por puesto?**

```
SELECT nombre, SUM(salario)
FROM puesto
GROUP BY nombre;
```



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

```
SELECT id_empleado, COUNT(*)
FROM venta
GROUP BY id_empleado;
```

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

```
SELECT id_articulo, COUNT(*)
FROM venta
GROUP BY id_articulo;
```

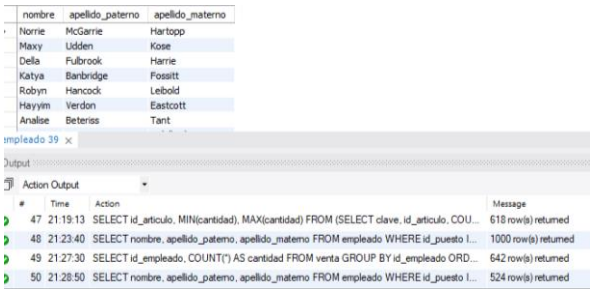
```
SELECT *
FROM empleado where id_puesto IN (SELECT id_puesto
FROM puesto
WHERE nombre = 'Junior Executive');

SELECT clave, id_articulo, COUNT(*) AS cantidad
FROM venta
GROUP BY clave, id_articulo
```

```
ORDER BY clave;

SELECT id_articulo, MIN(cantidad), MAX(cantidad)
FROM (SELECT clave, id_articulo, COUNT(*) AS cantidad
FROM venta
GROUP BY clave, id_articulo
ORDER BY clave) AS subconsulta
GROUP BY id_articulo;
```

Reto 4

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

```
SELECT nombre, apellido_paterno, apellido_materno
FROM empleado
WHERE id_puesto IN (SELECT id_puesto
FROM puesto
WHERE salario < 20000);
```



nombre	apellido_paterno	apellido_materno
Norrie	McGarrie	Hartopp
Maxy	Udden	Kose
Della	Fulbrook	Harrie
Katya	Banbridge	Fossitt
Robyn	Hancock	Leibold
Hayyim	Verdon	Eastcott
Analise	Beteriss	Tant

empleado 39

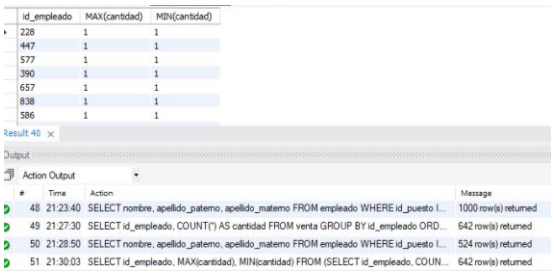
Output

Action Output

#	Time	Action	Message
47	21:19:13	SELECT id_articulo, MIN(cantidad), MAX(cantidad) FROM (SELECT clave, id_articulo, COU...	618 row(s) returned
48	21:23:40	SELECT nombre, apellido_paterno, apellido_materno FROM empleado WHERE id_puesto I...	1000 row(s) returned
49	21:27:30	SELECT id_empleado, COUNT(*) AS cantidad FROM venta GROUP BY id_empleado ORD...	642 row(s) returned
50	21:28:50	SELECT nombre, apellido_paterno, apellido_materno FROM empleado WHERE id_puesto I...	524 row(s) returned

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

```
SELECT id_empleado, MAX(cantidad), MIN(cantidad)
FROM (SELECT id_empleado, COUNT(*) AS cantidad
FROM venta
GROUP BY id_empleado, clave
ORDER BY clave) AS subconsulta
GROUP BY id_empleado;
```



id_empleado	MAX(cantidad)	MIN(cantidad)
228	1	1
447	1	1
577	1	1
390	1	1
657	1	1
838	1	1
586	1	1

Result 40

Output

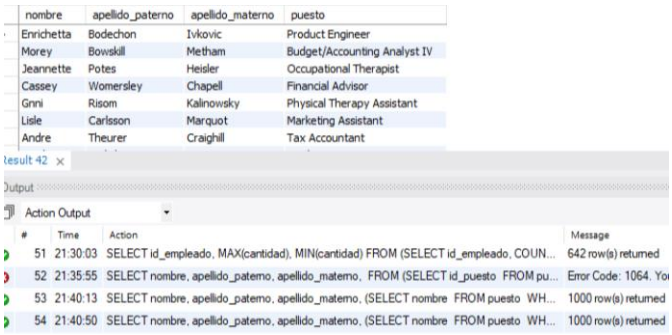
Action Output

#	Time	Action	Message
48	21:23:40	SELECT nombre, apellido_paterno, apellido_materno FROM empleado WHERE id_puesto I...	1000 row(s) returned
49	21:27:30	SELECT id_empleado, COUNT(*) AS cantidad FROM venta GROUP BY id_empleado ORD...	642 row(s) returned
50	21:28:50	SELECT nombre, apellido_paterno, apellido_materno FROM empleado WHERE id_puesto I...	524 row(s) returned
51	21:30:03	SELECT id_empleado, MAX(cantidad), MIN(cantidad) FROM (SELECT id_empleado, COUN...	642 row(s) returned

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

```
SELECT nombre, apellido_paterno, apellido_materno, (SELECT nombre
        FROM puesto
        WHERE id_puesto = e.id_puesto) AS Puesto
```

FROM empleado AS e;



nombre	apellido_paterno	apellido_materno	puesto
Enrichetta	Bodechon	Ivkovic	Product Engineer
Morey	Bowskill	Metham	Budget/Accounting Analyst IV
Jeannette	Potes	Heisler	Occupational Therapist
Cassey	Womersley	Chapell	Financial Advisor
Gnni	Risom	Kalinowsky	Physical Therapy Assistant
Lisle	Carlsson	Marquot	Marketing Assistant
Andre	Theurer	Craighill	Tax Accountant

Result 42

Output

Action Output

#	Time	Action	Message
51	21:30:03	SELECT id_empleado, MAX(cantidad), MIN(cantidad) FROM (SELECT id_empleado, COUN...	642 row(s) returned
52	21:35:55	SELECT nombre, apellido_paterno, apellido_materno, FROM (SELECT id_puesto FROM pu...	Error Code: 1064. Yo
53	21:40:13	SELECT nombre, apellido_paterno, apellido_materno, (SELECT nombre FROM puesto WH...	1000 row(s) returned
54	21:40:50	SELECT nombre, apellido_paterno, apellido_materno, (SELECT nombre FROM puesto WH...	1000 row(s) returned

Proyecto

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

```
SELECT firstName, lastName, employeeNumber
FROM employees
WHERE firstName LIKE 'A%';
```

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

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

```
SELECT firstName, lastName, employeeNumber
FROM employees
WHERE firstName LIKE '%on';
```

firstName	lastName	employeeNumber
NULL	NULL	NULL

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

```
SELECT firstName, lastName, employeeNumber
FROM employees
WHERE firstName LIKE '%on%';
```

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

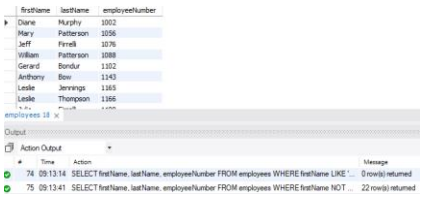
4. **Dentro de la tabla `employees`, obtén el número de empleado, apellido y nombre de todos los empleados cuyos nombres tienen tres letras e inician con `T` y finalizan con `m`.**

```
SELECT firstName, lastName, employeeNumber
FROM employees
WHERE firstName LIKE 'T_M';
```

firstName	lastName	employeeNumber
Tom	King	1619
NULL	NULL	NULL

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

```
SELECT firstName, lastName, employeeNumber
FROM employees
WHERE firstName NOT LIKE 'B%';
```



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

```
SELECT productCode, productName
FROM products
WHERE productCode LIKE '%_20%';
```

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

```
SELECT orderNumber, SUM(quantityOrdered * priceEach) AS total
FROM orderdetails
GROUP BY orderNumber;
```

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

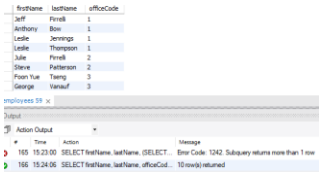
```
SELECT EXTRACT(YEAR FROM orderDate) AS año, COUNT(*) AS ordenes
FROM orders
GROUP BY año;
```

año	ordenes
2003	111
2004	151
2005	64

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

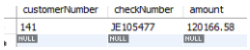
```
SELECT firstName, lastName, officeCode
```

```
FROM employees
WHERE officeCode IN (SELECT officeCode
FROM offices
WHERE country LIKE 'USA');
```



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

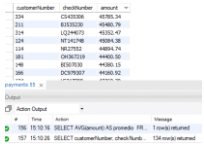
```
SELECT customerNumber, checkNumber, amount
FROM payments
ORDER BY amount DESC
LIMIT 1;
```



customerNumber	checkNumber	amount
141	JE105477	120166.58
NULL	NULL	NULL

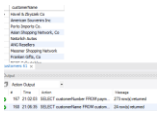
11. **Obtén el número de cliente, número de cheque y cantidad de aquellos clientes cuyo pago es más alto que el promedio.**

```
SELECT customerNumber, checkNumber, amount
FROM payments
WHERE amount > (SELECT AVG(amount) AS promedio
FROM payments);
```



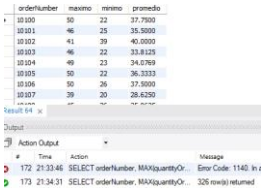
12. **Obtén el nombre de aquellos clientes que no han hecho ninguna orden.**

```
SELECT customerName
FROM customers
WHERE customerNumber NOT IN (SELECT customerNumber
FROM payments);
```



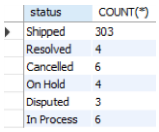
13. **Obtén el máximo, mínimo y promedio del número de productos en las órdenes de venta.**

```
SELECT orderNumber, MAX(quantityOrdered) AS maximo, MIN(quantityOrdered) AS
minimo, AVG(quantityOrdered) AS promedio
FROM orderdetails
GROUP BY orderNumber;
```



14. **Dentro de la tabla `orders`, obtén 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

Sesión 3

Código

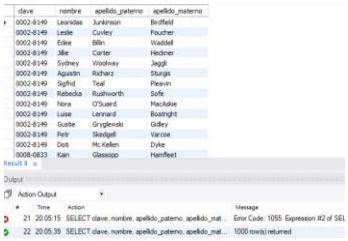
```
USE AVH;

#tambien puede ser AVH.movies y asi no pongo USE AVH;
SELECT *
FROM movies;

USE tienda;

SHOW KEYS
FROM venta;

SELECT *
FROM empleado AS e
JOIN puesto AS p
	ON e.id_puesto = p.id_puesto;

#primer tabla es del FROM es el izquierdo y la segunda tabla es la derecha donde esta el JOIN
SELECT *
FROM puesto AS p
LEFT JOIN empleado AS e
	ON p.id_puesto = e.id_puesto;

SELECT *
FROM empleado AS e
```

```
RIGHT JOIN puesto AS p
        ON e.id_puesto = p.id_puesto;
```

Reto 1

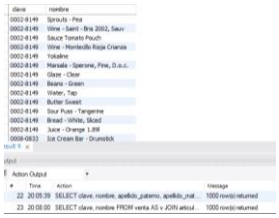
1. **¿Cuál es el nombre de los empleados que realizaron cada venta?**

```
SELECT clave, nombre, apellido_paterno, apellido_materno
FROM venta AS v
JOIN empleado AS e
        ON v.id_empleado = e.id_empleado
ORDER BY clave;
```



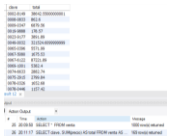
2. **¿Cuál es el nombre de los artículos que se han vendido?**

```
SELECT clave, nombre
FROM venta AS v
JOIN articulo AS a
        ON v.id_articulo = a.id_articulo
ORDER BY clave;
```



3. **¿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;
```



```
CREATE VIEW AVTickets AS
(SELECT v.clave, v.fecha, a.nombre AS producto, a.precio,
concat(e.nombre, ' ', e.apellido_paterno) AS empleado
FROM venta AS v
JOIN empleado AS e
```

```
        ON v.id_empleado = e.id_empleado
JOIN articulo AS a
        ON v.id_articulo = a.id_articulo);
```

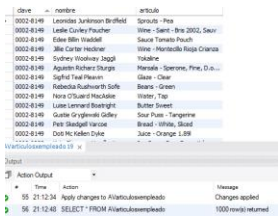
Reto 2

1. **Obtener el puesto de un empleado.**

```
CREATE VIEW AVpuesto AS
(SELECT p.nombre AS puesto, concat(e.nombre,' ',e.apellido_paterno,'
',e.apellido_materno) AS empleado
FROM empleado AS e
JOIN puesto AS p
        ON e.id_puesto = p.id_puesto);

SELECT *
FROM AVpuesto;
```

2. **Saber qué artículos ha vendido cada empleado.**

```
CREATE VIEW AVarticulosxempleado AS
(SELECT v.clave, concat(e.nombre,' ', e.apellido_paterno, ' ', e.apellido_materno) AS
nombre, a.nombre AS articulo
FROM venta AS v
JOIN empleado AS e
        ON v.id_empleado = e.id_empleado
JOIN articulo AS a
        ON v.id_articulo = a.id_articulo);

SELECT *
FROM AVarticulosxempleado;
```



3. **Saber qué puesto ha tenido más ventas.**

```
CREATE VIEW AVventas AS
(SELECT p.nombre AS puesto, COUNT(v.clave) AS ventas
FROM venta AS v
JOIN empleado AS e
        ON v.id_empleado = e.id_empleado
JOIN puesto AS p
```

```
        ON e.id_puesto = p.id_puesto
GROUP BY p.nombre);

SELECT *
FROM AVventas
ORDER BY ventas DESC
LIMIT 1;
```

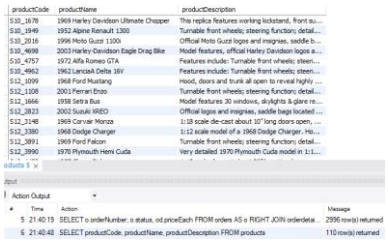
puesto	ventas
Physical Therapy Assistant	23

Proyecto

Para estas consultas usa RIGHT JOIN

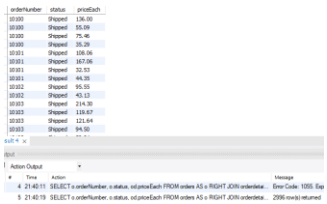
1. **Obtén el código de producto, nombre de producto y descripción de todos los productos.**

```
SELECT productCode, productName, productDescription
FROM products;
```



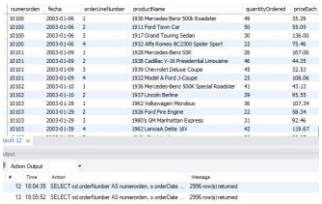
2. **Obtén el número de orden, estado y costo total de cada orden.**

```
SELECT o.orderNumber, o.status, od.priceEach
FROM orders o
RIGHT JOIN orderdetails od
        ON o.orderNumber = od.orderNumber;
```



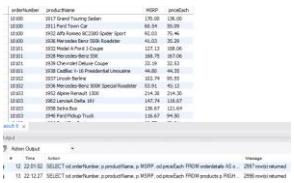
3. **Obtén el número de orden, fecha de orden, línea de orden, nombre del producto, cantidad ordenada y precio de cada pieza que muestre los detalles de cada orden.**

```
SELECT od.orderNumber AS numerorden, o.orderDate AS fecha, od.orderLineNumber,
p.productName, od.quantityOrdered, od.priceEach
FROM orderdetails od
RIGHT JOIN products p
        ON od.productCode = p.productCode
RIGHT JOIN orders o
        ON od.orderNumber = o.orderNumber
ORDER BY numerorden, od.orderLineNumber;
```



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

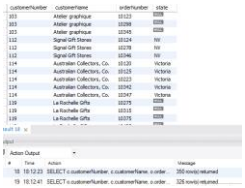
```
SELECT od.orderNumber, p.productName, p.MSRP, od.priceEach
FROM products p
RIGHT JOIN orderdetails od
	ON od.productCode = p.productCode;
```



Para estas consultas usa LEFT JOIN

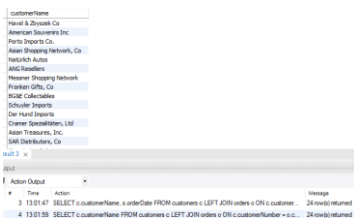
5. **Obtén el número de cliente, nombre de cliente, número de orden y estado de cada cliente.**

```
SELECT o.customerNumber, c.customerName, o.orderNumber, c.state
FROM orders o
LEFT JOIN customers c
	ON o.customerNumber = c.customerNumber;
```



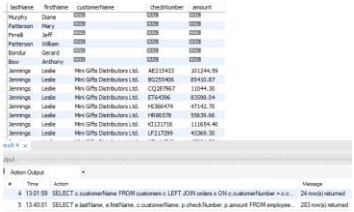
6. **Obtén los clientes que no tienen una orden asociada.**

```
SELECT c.customerName
FROM customers c
LEFT JOIN orders o
	ON c.customerNumber = o.customerNumber
WHERE o.customerNumber IS 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.**

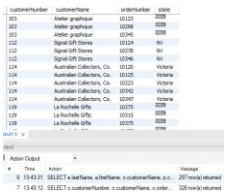
```
SELECT e.lastName, e.firstName, c.customerName, p.checkNumber, p.amount
FROM employees e
LEFT JOIN customers c
        ON e.employeeNumber = c.salesRepEmployeeNumber
LEFT JOIN  payments p
        ON c.customerNumber = p.customerNumber;
```



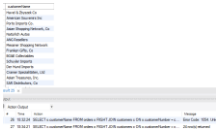
Para estas consultas usa RIGHT JOIN

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

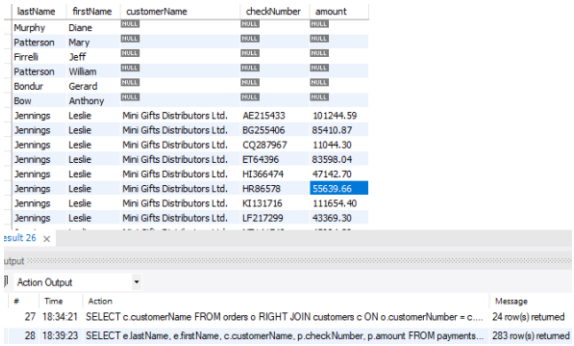
```
SELECT c.customerNumber, c.customerName, o.orderNumber, c.state
FROM customers c
RIGHT JOIN orders o
        ON c.customerNumber = o.customerNumber;
```



```
SELECT c.customerName
FROM orders o
RIGHT JOIN customers c
        ON o.customerNumber = c.customerNumber
WHERE o.customerNumber IS NULL;
```



```
SELECT e.lastName, e.firstName, c.customerName, p.checkNumber, p.amount
FROM payments p
RIGHT JOIN customers c
        ON p.customerNumber = c.customerNumber
RIGHT JOIN employees e
        ON c.salesRepEmployeeNumber = e.employeeNumber;
```



lastName	firstName	customerName	checkNumber	amount
Murphy	Diane	NULL	NULL	NULL
Patterson	Mary	NULL	NULL	NULL
Firrelli	Jeff	NULL	NULL	NULL
Patterson	William	NULL	NULL	NULL
Bondur	Gerard	NULL	NULL	NULL
Bow	Anthony	NULL	NULL	NULL
Jennings	Leslie	Mini Gifts Distributors Ltd.	AE215433	101244.59
Jennings	Leslie	Mini Gifts Distributors Ltd.	BG255406	85410.87
Jennings	Leslie	Mini Gifts Distributors Ltd.	CQ287967	11044.30
Jennings	Leslie	Mini Gifts Distributors Ltd.	ET64396	83598.04
Jennings	Leslie	Mini Gifts Distributors Ltd.	HI366474	47142.70
Jennings	Leslie	Mini Gifts Distributors Ltd.	HR86578	55639.66
Jennings	Leslie	Mini Gifts Distributors Ltd.	KI131716	111654.40
Jennings	Leslie	Mini Gifts Distributors Ltd.	LF217299	43369.30

esult 26 ×

utput

Action Output

#	Time	Action	Message
27	18:34:21	SELECT c.customerName FROM orders o RIGHT JOIN customers c ON o.customerNumber = c....	24 row(s) returned
28	18:39:23	SELECT e.lastName, e.firstName, c.customerName, p.checkNumber, p.amount FROM payments...	283 row(s) returned

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

```
CREATE VIEW AVco AS
(SELECT c.customerNumber, c.customerName, o.orderNumber, c.state
FROM customers c
RIGHT JOIN orders o
        ON c.customerNumber = o.customerNumber);

#cuantas órdenes han hecho los clientes
SELECT COUNT(*)
FROM AVco;
```

COUNT(*)
326

```
CREATE VIEW AVpce AS
(SELECT e.lastName, e.firstName, c.customerName, p.checkNumber, p.amount
FROM payments p
RIGHT JOIN customers c
        ON p.customerNumber = c.customerNumber
RIGHT JOIN employees e
        ON c.salesRepEmployeeNumber = e.employeeNumber);

#Cuantos empleados no venden
SELECT concat(firstName, ' ', lastName) AS name, SUM(amount)
FROM AVpce
GROUP BY name
HAVING SUM(amount) IS NULL;
```

name	SUM(amount)
Diane Murphy	NULL
Mary Patterson	NULL
Jeff Firrelli	NULL
William Patterson	NULL
Gerard Bondur	NULL
Anthony Bow	NULL
Tom King	NULL
Yoshimi Kato	NULL

```
CREATE VIEW AVo AS
(SELECT o.orderNumber, o.status, od.priceEach, od.quantityOrdered
FROM orders o
RIGHT JOIN orderdetails od
        ON o.orderNumber = od.orderNumber);
```

#Menciona las ordenes que se cancelaron y su precio
SELECT orderNumber, SUM(priceEach * quantityOrdered) AS total, status
FROM AVo
GROUP BY orderNumber
HAVING status LIKE 'Cancelled';

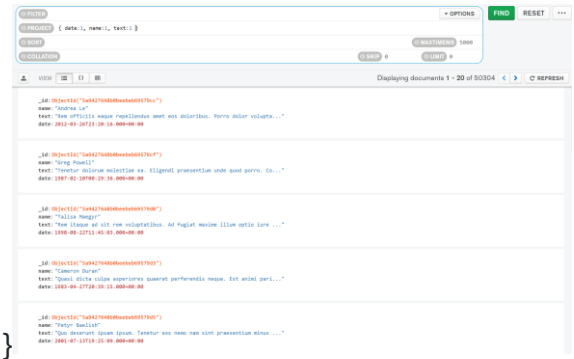
orderNumber	total	status
10167	44167.09	Cancelled
10179	22963.60	Cancelled
10248	41445.21	Cancelled
10253	45443.54	Cancelled
10260	37769.38	Cancelled
10262	47065.36	Cancelled

Sesión 4 - Mongo DB

Reto 1

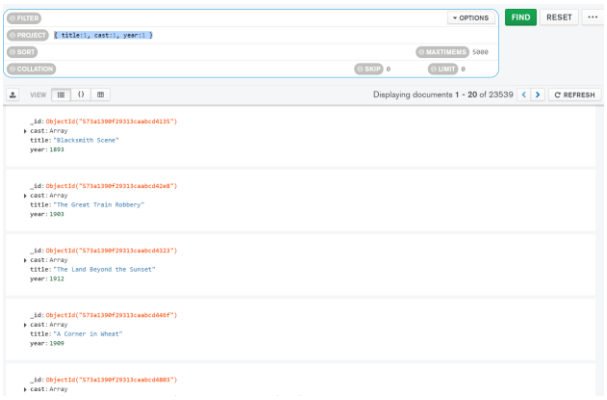
1. **Fecha, nombre y texto de cada comentario.**

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



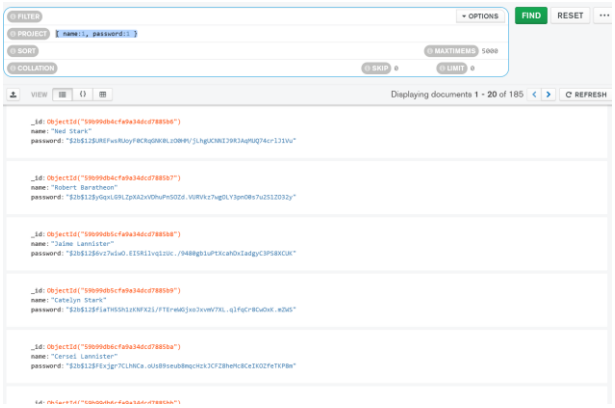
2. **Título, elenco y año de cada película.**

{ title:1, cast:1, year:1 }



3. **Nombre y contraseña de cada usuario.**

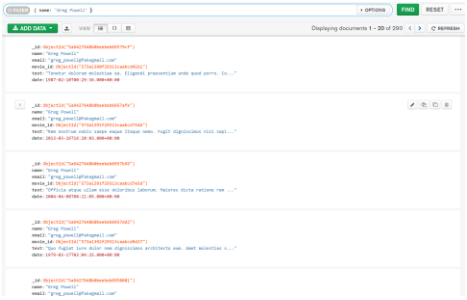
{ name:1, password:1 }



Reto 2

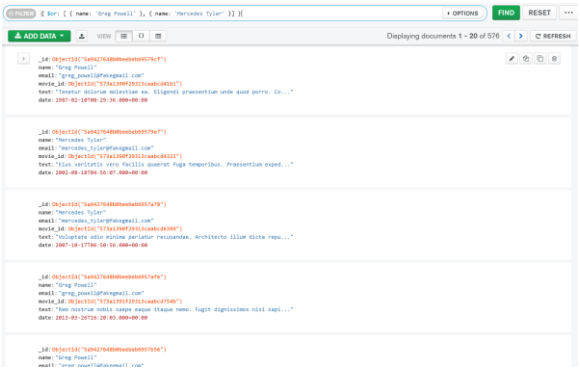
1. **¿Qué comentarios ha hecho Greg Powell?**

Filter: { name: ‘Greg Powell’ }



2. **¿Qué comentarios han hecho Greg Powell o Mercedes Tyler?**

Filter: { $or: [{ name: ‘Greg Powell’ }, { name: ‘Mercedes Tyler’ }] }

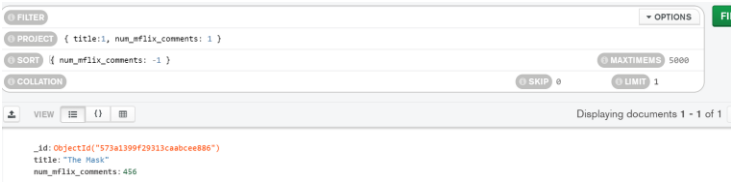


3. **¿Cuál es el máximo número de comentarios en una película?**

Project: { title:1, num_mflix_comments: 1 }

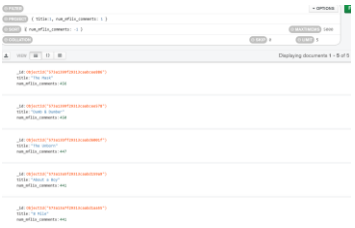
Sort: { num_mflix_comments: -1 }

Limit: 1



4. **¿Cuál es título de las cinco películas más comentadas?**

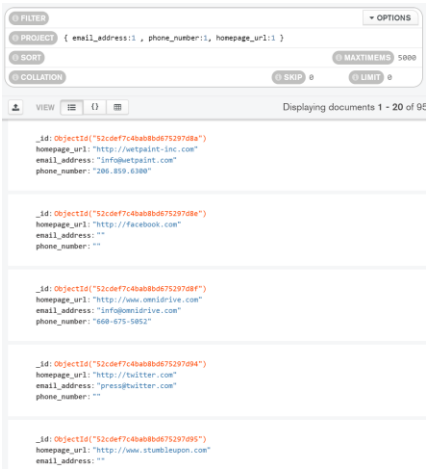
Project: { title:1, num_mflix_comments: 1 }
Sort: { num_mflix_comments: -1 }
Limit: 5



Proyecto

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

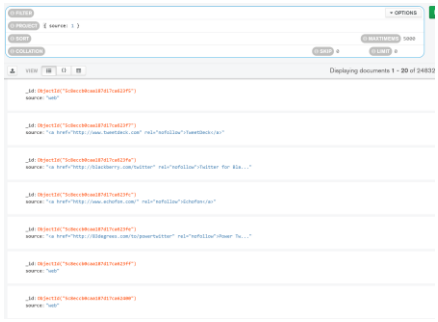
```
{
 project: {
  email_address: 1,
  phone_number: 1,
  homepage_url: 1
 }
}
```



2. **Obtén la fuente de cada tweet.**

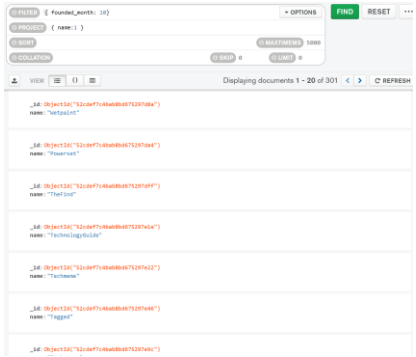
```
{
 project: {
  source: 1
```

}
}



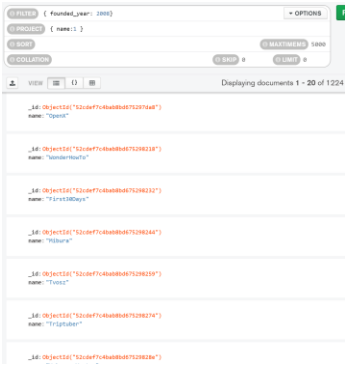
3. **Obtén el nombre de todas las compañías fundadas en octubre.**

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



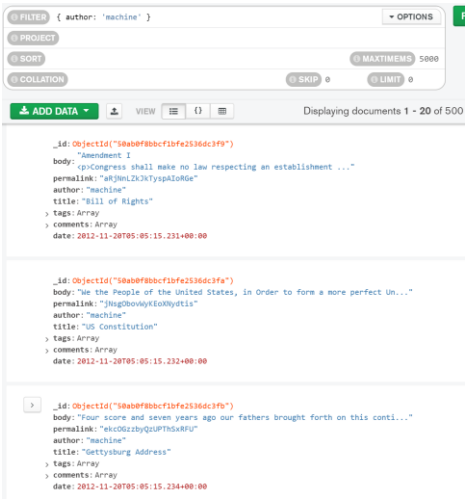
4. **Obtén el nombre de todas las compañías fundadas en 2008**.

```
{
 filter: {
  founded_year: 2008
 },
 project: {
  name: 1
 }
}
```



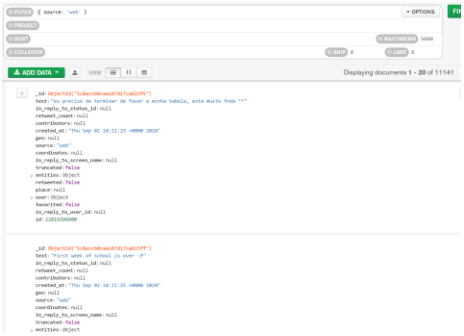
5. **Obtén todos los *post* del autor `machine`.**

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



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

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

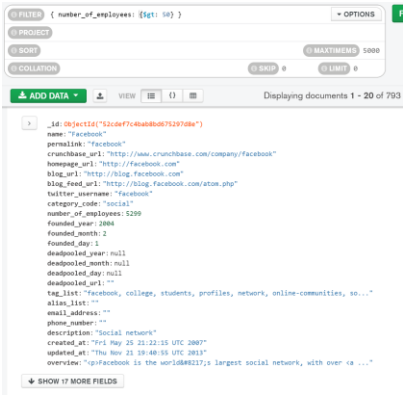


7. **Obtén todas las compañías fundadas en octubre del 2008.**

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

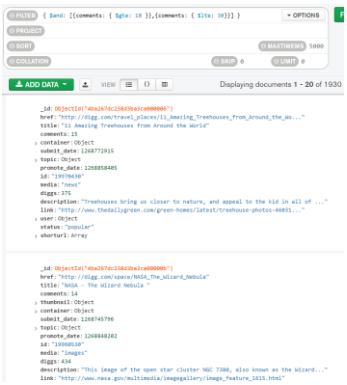
8. **Obtén todas las compañías con más de 50 empleados.**

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



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

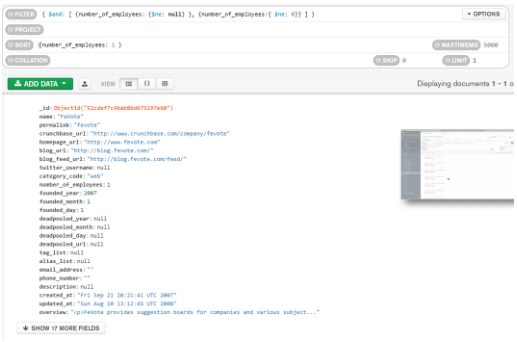
```
{
 filter: {
  $and: [
   {
    comments: {
     $gte: 10
    }
   },
   {
    comments: {
     $lte: 30
    }
   }
  ]
 }
}
```



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

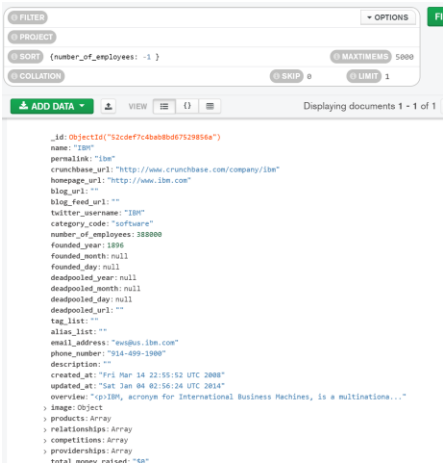
```
{
```

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



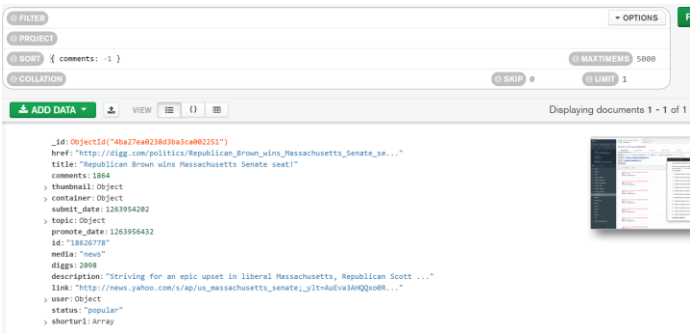
11. Obtén la empresa con el mayor número de empleados.

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



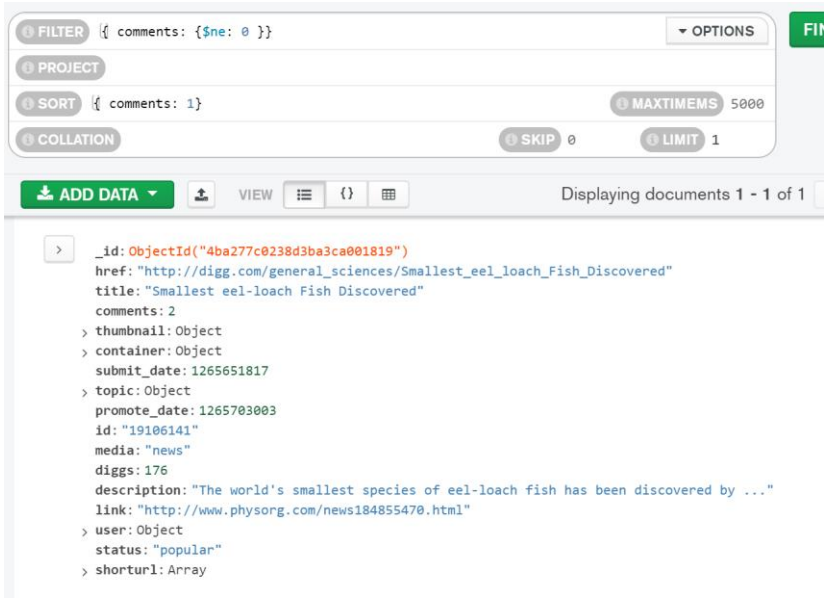
12. Obtén la historia más comentada.

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



13. Obtén la historia menos comentada.

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

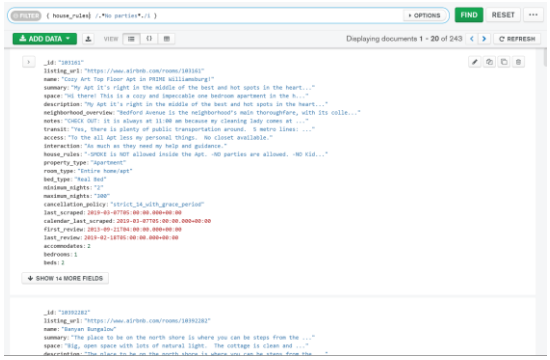


Sesión 5

Reto 1

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

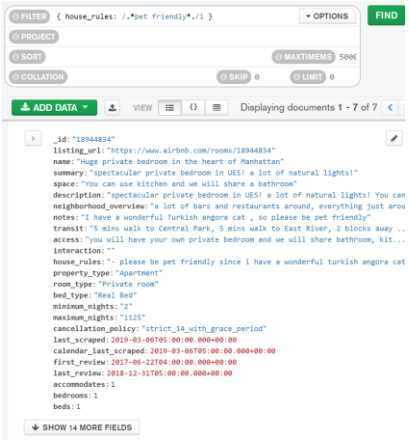
```
{
 filter: {
  house_rules: RegExp('.*No parties*.', i)
 }
}
```



2. **Propiedades que admitan mascotas.**

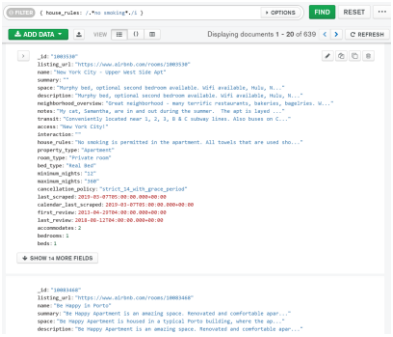
```
{
 filter: {
  house_rules: RegExp('.*pet friendly*.', i)
 }
```

}



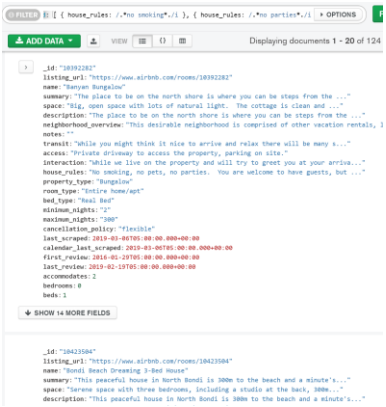
3. Propiedades que no permitan fumadores.

```
{
 filter: {
  house_rules: RegExp('.*no smoking*.', i)
 }
}
```



4. Propiedades que no permitan fiestas ni fumadores.

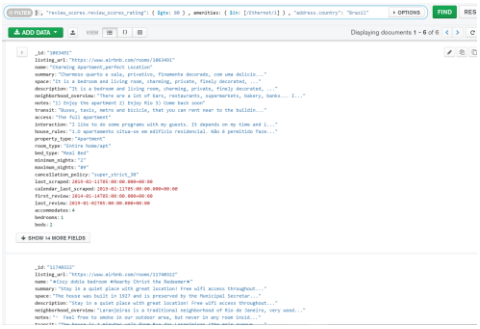
```
{
 filter: {
  $and: [
   {
    house_rules: RegExp('.*no smoking*.', i)
   },
   {
    house_rules: RegExp('.*no parties*.', i)
   }
  ]
 }
}
```



Reto 2

1. **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.**

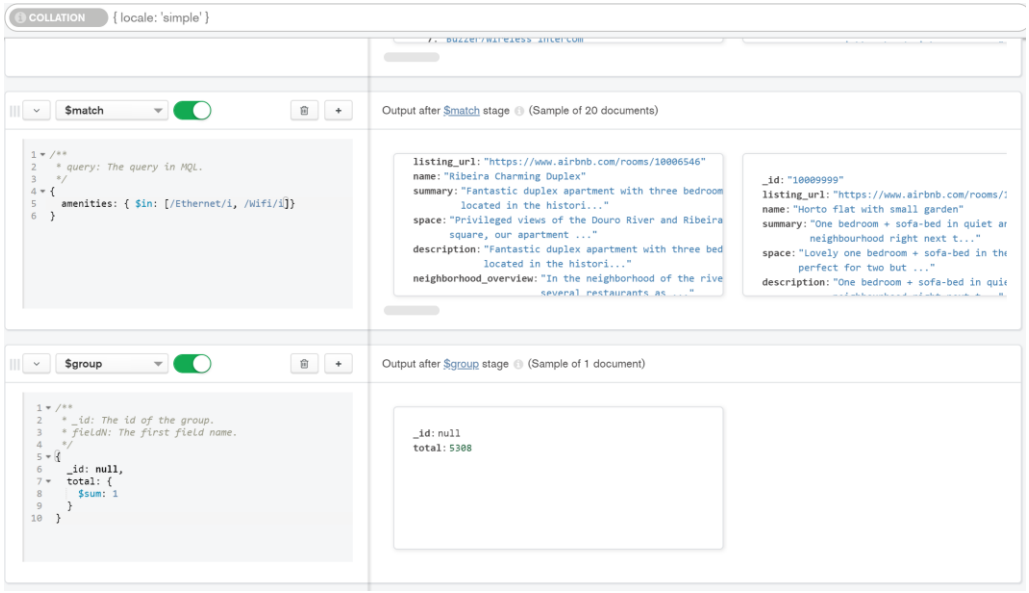
```
{
 filter: {
  number_of_reviews: {
   $gte: 50
  },
  'review_scores.review_scores_rating': {
   $gte: 80
  },
  amenities: {
   $in: [
    RegExp('Ethernet', i)
   ]
  },
  'address.country': 'Brazil'
 }
}
```



Reto 3

1. **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: {
  amenities: { $in: [/Ethernet/i, /Wifi/i]}
}}, {$group: {
  _id: null,
  total: {
    $sum: 1
  }
}}]
```



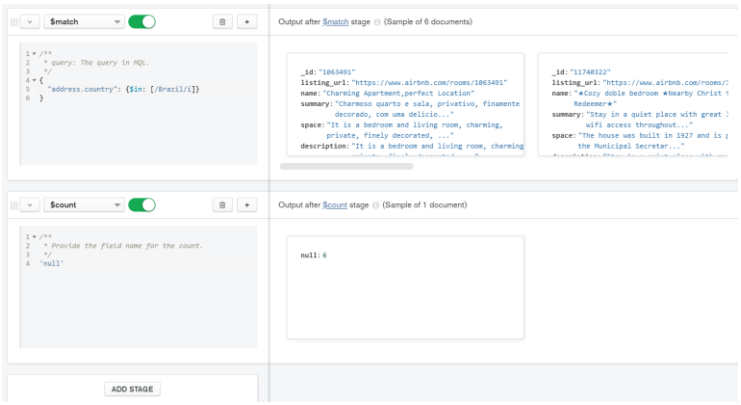
Proyecto

1. **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": {$in: [/Brazil/i]}
```

}}, {$count: 'null'}]

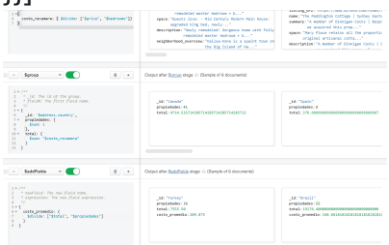


Sesión 6

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.

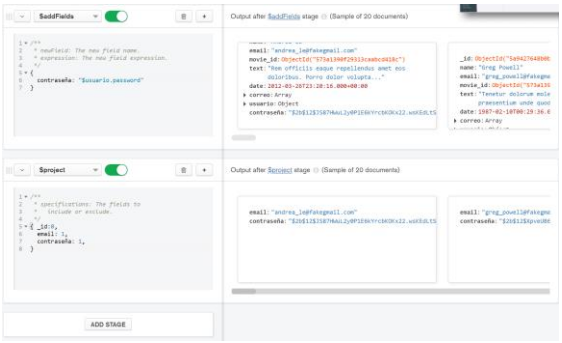
```
[{$match: {
  property_type: "House",
  bedrooms: {$gte: 1 }
}}, {$addFields: {
  costo_recamara: { $divide: ["$price", "$bedrooms"]}
}}, {$group: {
  _id: '$address.country',
  propiedades: {
    $sum: 1
  },
  total: {
    $sum: "$costo_recamara"
  }
}}, {$addFields: {
  costo_promedio: {
    $divide: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



Reto 2

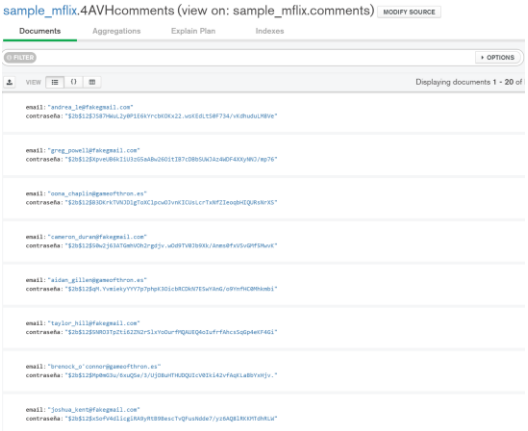
Usando las colecciones `comments` y `users`, se requiere conocer el correo y contraseña de cada persona que realizó un comentario. Construye un pipeline que genere como resultado estos datos.

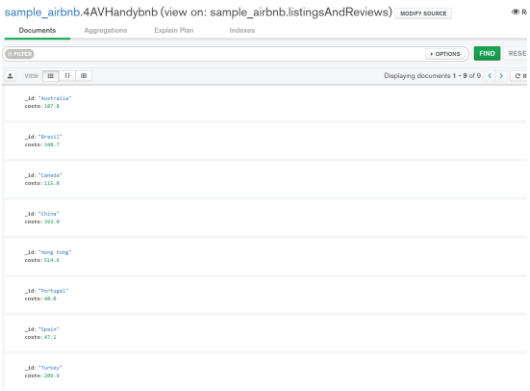
```
[{$lookup: {
  from: 'users',
  localField: 'email',
  foreignField: 'email',
  as: 'correo'
}}, {$addFields: {
  usuario: {$arrayElemAt: ["$correo", 0]}
}}, {$addFields: {
  contraseña: "$usuario.password"
}}, {$project: { _id:0,
  email: 1,
  contraseña: 1,
}}]
```



Reto 3

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

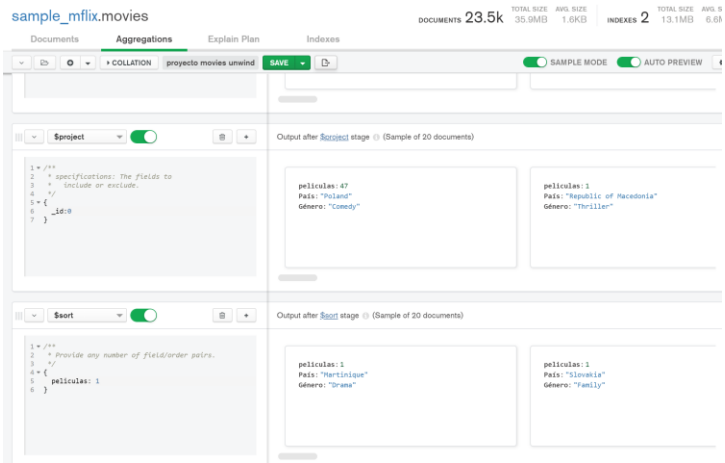




Proyecto

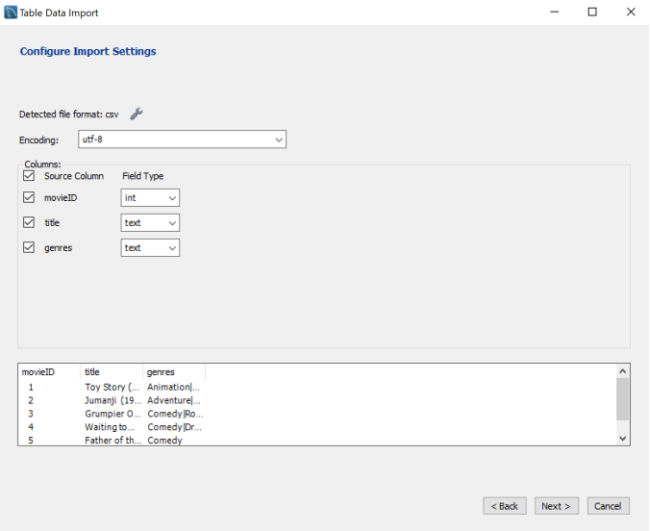
El proyecto consiste en obtener, por país, el número de películas que hay de cada género. Un ejemplo de salida en formato de tabla sería:

```
[{$unwind: {
  path: "$genres"
}}, {$unwind: {
  path: "$countries"
}}, {$group: {
  _id: {genero: "$genres", pais: "$countries"},
  peliculas: {
    $sum : 1
  }
}}, {$addFields: {
  País: "$_id.pais",
  Género: "$_id.genero"
}}, {$project: {
  _id:0
}}, {$sort: {
  peliculas: 1
}}]
```



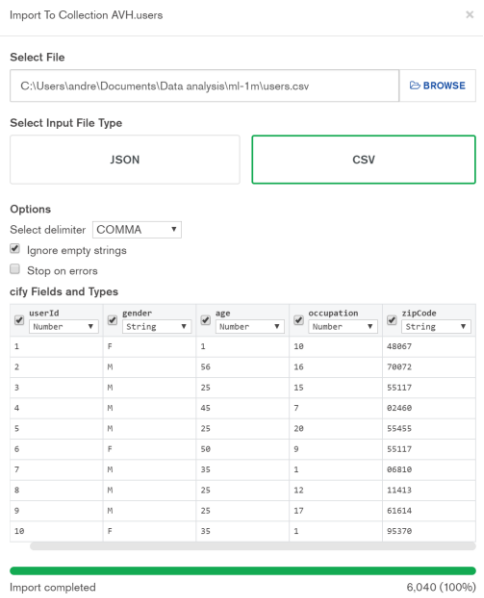
Sesión 7

Reto 1

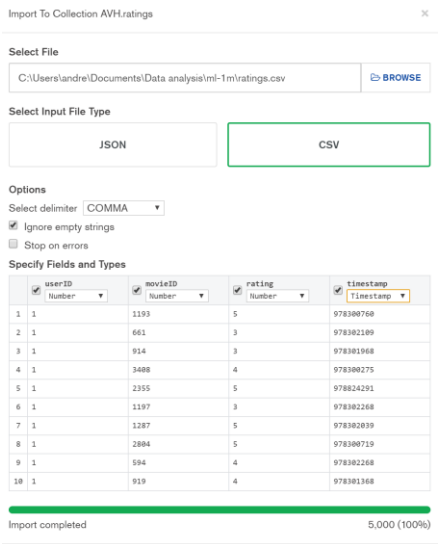


Reto 2

Reto 3

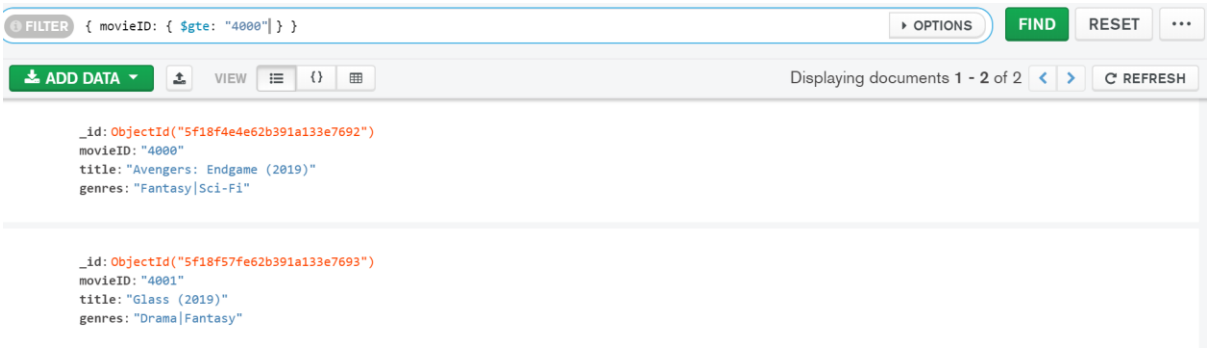


Reto 4



Proyecto

Agregar los siguientes registros en formato CSV a la Colección `movies`



Para recordar

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

Y entonces el correspondiente formato JSON será:

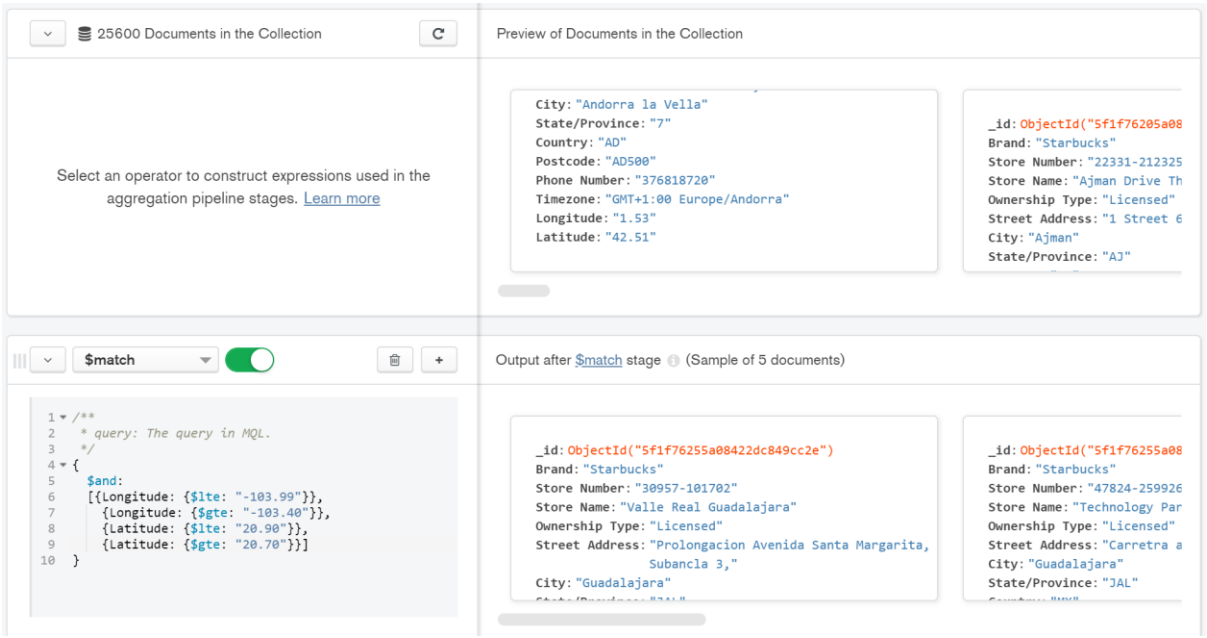
```
{
  id:"4000",
  titulo:"Avengers: Endgame (2019)",
  genres:"Fantasy|Sci-Fi"
}
{
  id:"4001",
  titulo:"Glass (2019)",
  genres:"Drama|Fantasy"
}
```

Sesión 7

Reto 1-Starbucks

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

```
[{$match: {
  $and:
  [{Longitude: {$lte: "-103.99"}},
    {Longitude: {$gte: "-103.40"}},
    {Latitude: {$lte: "20.90"}},
    {Latitude: {$gte: "20.70"}}]
}}]
```

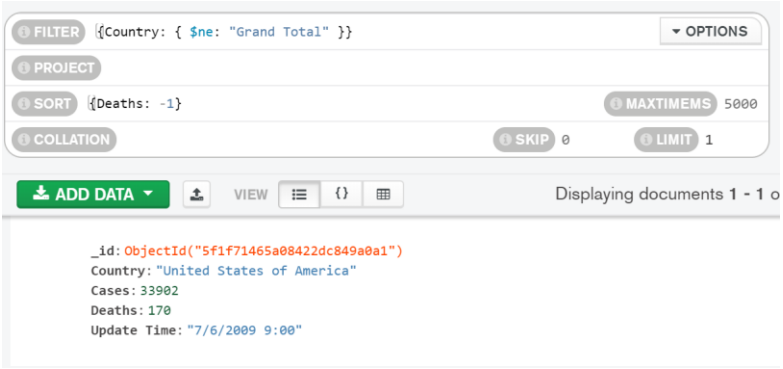


Reto 2-H1N1

1. **¿Cuál fue el país con mayor número de muertes?**

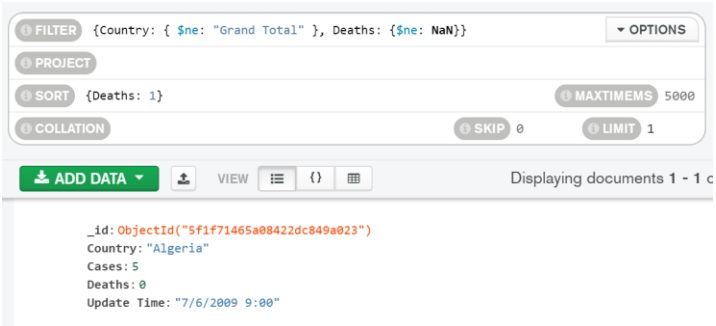
```
{
 filter: {
  Country: {
   $ne: 'Grand Total'
  }
 },
 sort: {
  Deaths: -1
 },
 limit: 1
```

}



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

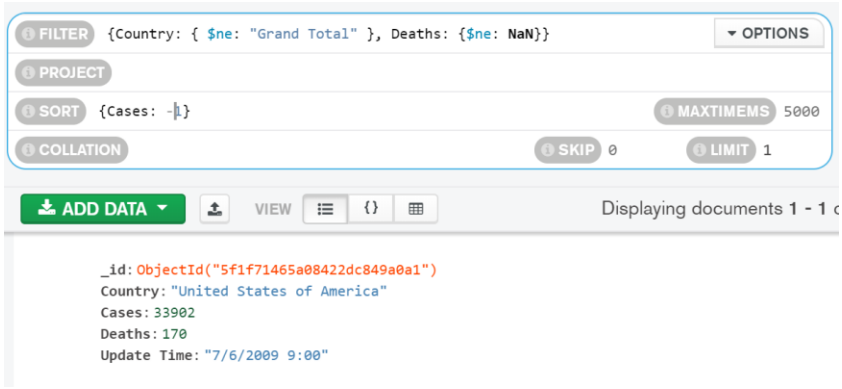
```
{
 filter: {
  Country: {
   $ne: 'Grand Total'
  },
  Deaths: {
   $ne: NaN
  }
 },
 sort: {
  Deaths: 1
 },
 limit: 1
}
```



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

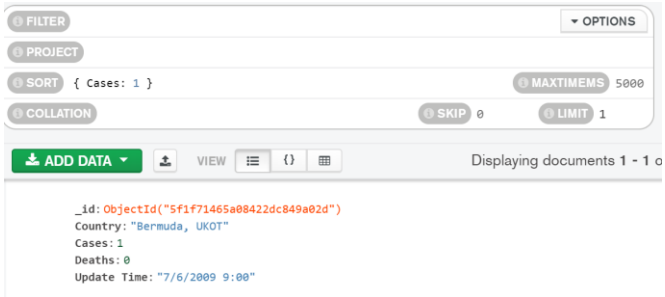
```
{
 filter: {
  Country: {
   $ne: 'Grand Total'
  },
  Deaths: {
```

```
  $ne: NaN
 }
},
sort: {
 Cases: -1
},
limit: 1
}
```



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

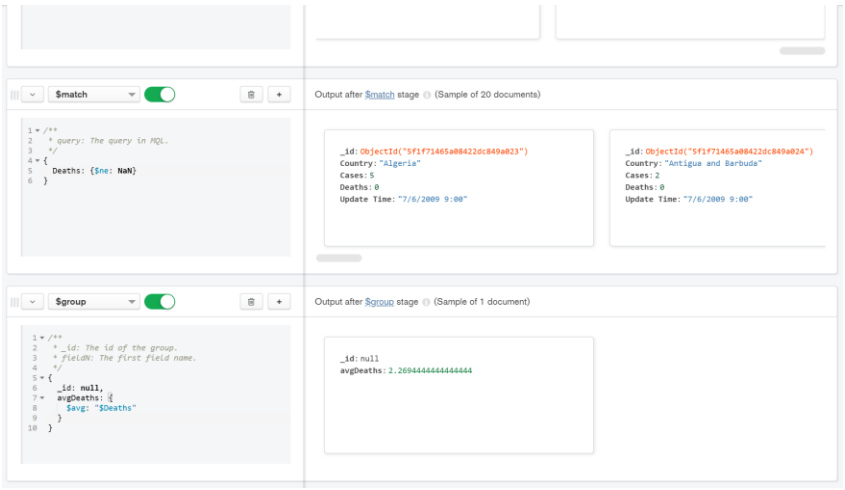
```
{
 sort: {
  Cases: 1
 },
 limit: 1
}
```



5. **¿Cuál fue el número de muertes promedio?**

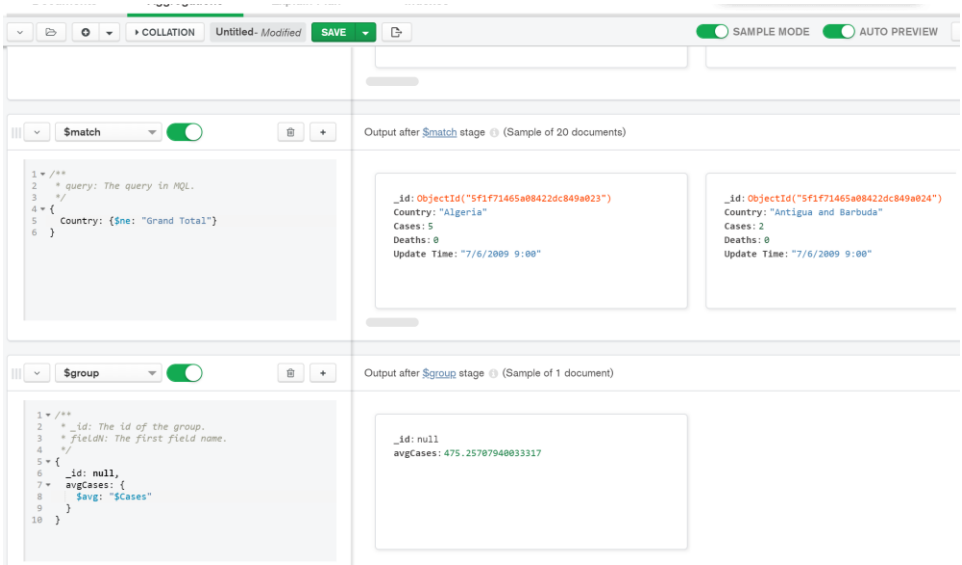
```
[{$match: {
  Country: {$ne: "Grand Total"}
}}, {$match: {
  Deaths: {$ne: NaN}
}}, {$group: {
  _id: null,
  avgDeaths: {
```

```
    $avg: "$Deaths"
  }
}}]
```



6. **¿Cuál fue el número de casos promedio?**

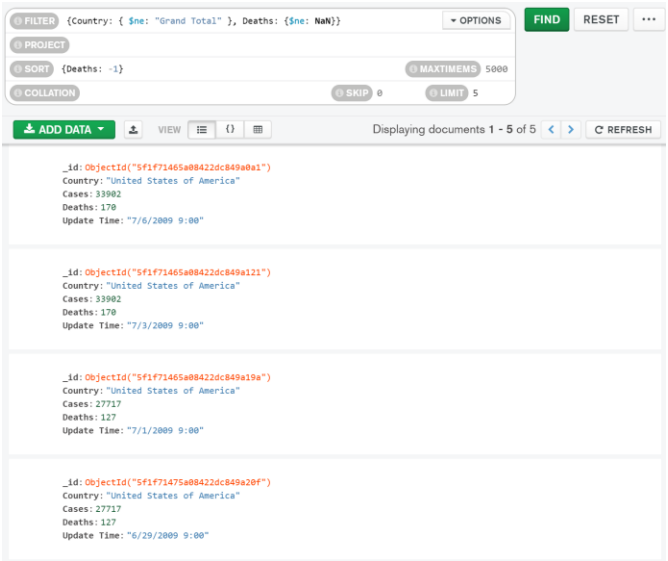
```
[{$match: {
  Country: {$ne: "Grand Total"}
}}, {$group: {
  _id: null,
  avgCases: {
    $avg: "$Cases"
  }
}}]
```



7. **Top 5 de países con más muertes**

```
{
 filter: {
  Country: {
```

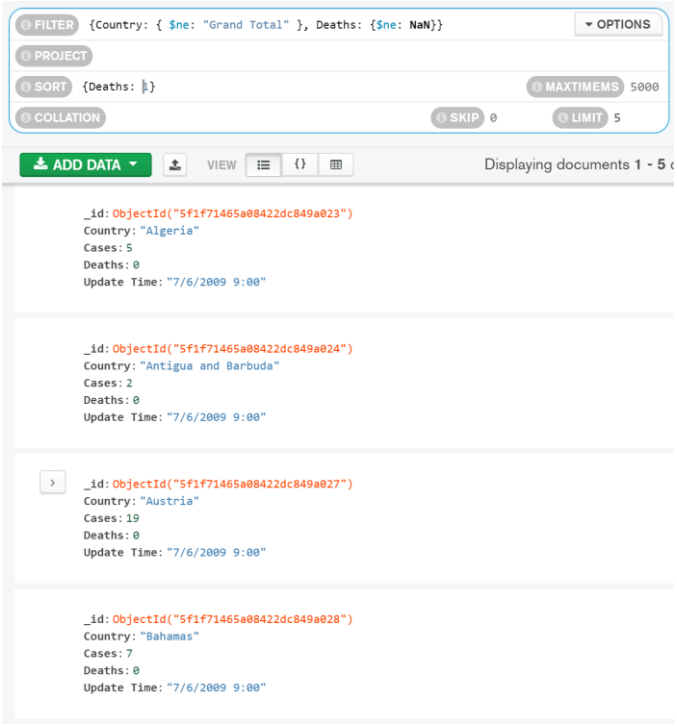
```
  $ne: 'Grand Total'
 },
 Deaths: {
  $ne: NaN
 }
},
sort: {
 Deaths: -1
},
limit: 5
}
```



8. **Top 5 de países con menos muertes**

```
{
 filter: {
  Country: {
   $ne: 'Grand Total'
  },
  Deaths: {
   $ne: NaN
  }
 },
 sort: {
  Deaths: 1
 },
 limit: 5
```

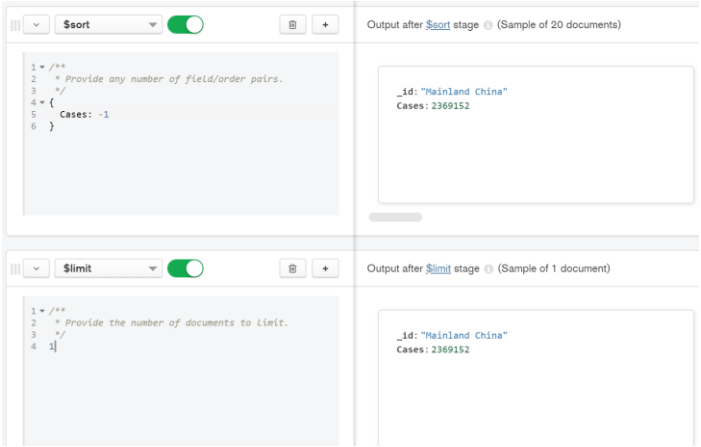
}



Reto 3-Covid

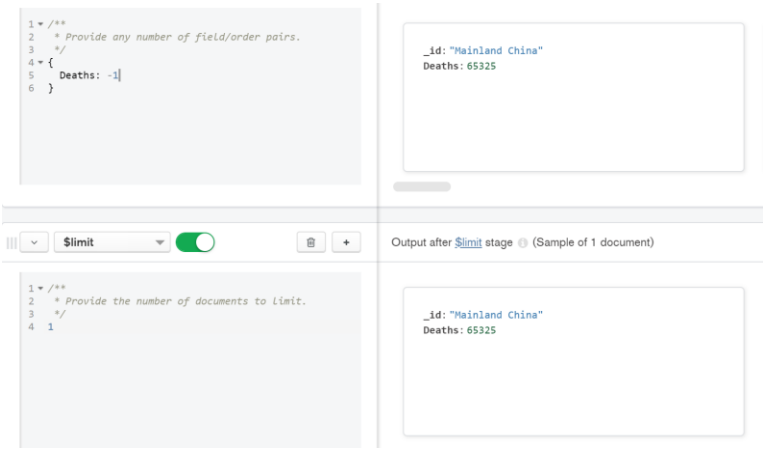
1. **¿Cuál es país con mayor número de casos?**

```
[{$addFields: {
  casos: {$convert: {input: "$Confirmed", to: 16}}

  }}, {$group: {
  _id: "$Region",
  Cases: {
    $sum: "$casos"
  }
}}, {$sort: {
  Cases: -1
}}, {$limit: 1}]
```



2. **¿Cuál es el país con mayor número de muertes?**

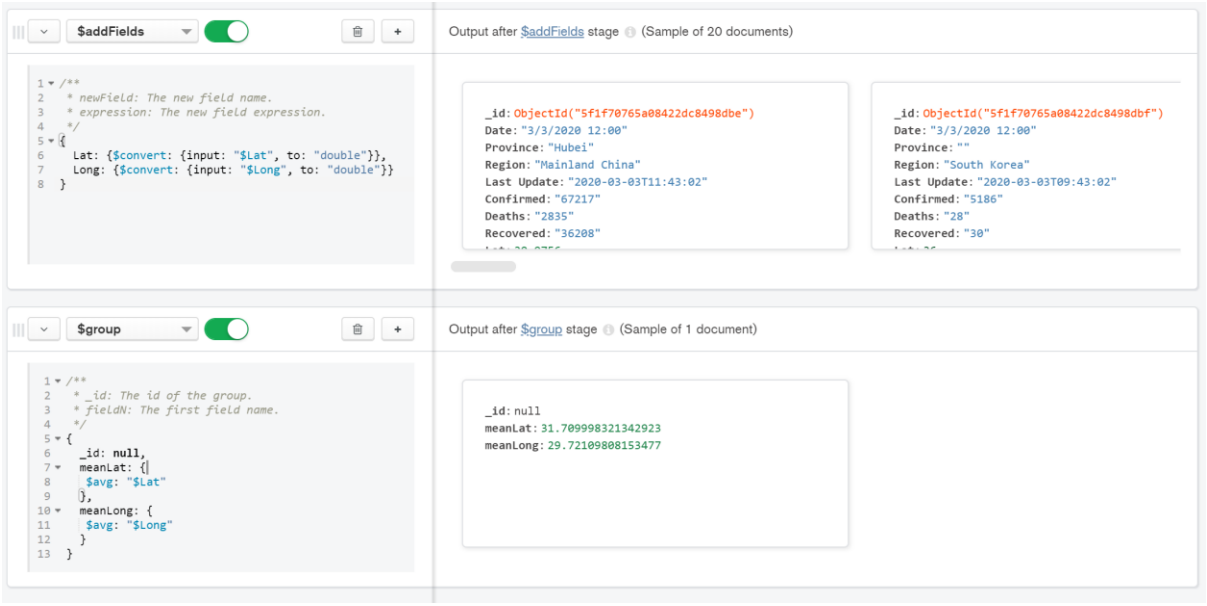
```
[{$addFields: {
  muertes: {$convert: {input: "$Deaths", to: 16}}

  }}, {$group: {
  _id: "$Region",
  Deaths: {
    $sum: "$muertes"
  }
}}, {$sort: {
  Deaths: -1
}}, {$limit: 1}]
```



3. **Usando las coordenadas, encuentra el epicentro del virus.**

```
[{$match: {
  $and:
  [{Lat: {$ne: ""}},
  {Long: {$ne: ""}}]
}}, {$addFields: {
  Lat: {$convert: {input: "$Lat", to: "double"}},
  Long: {$convert: {input: "$Long", to: "double"}}
}}, {$group: {
  _id: null,
  meanLat: {
   $avg: "$Lat"
  },
  meanLong: {
   $avg: "$Long"
  }
```

}}]



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