

Andrea Valdez Hernández

Sesión 1 - SQL	2
Código	2
Reto 1	2
Reto 2	3
Reto 3	5
Proyecto	5
Sesión 2	11
Código	11
Reto 1	11
Reto 2	12
Reto 3	13
Reto 4	15
Proyecto	16
Sesión 3	19
Código	19
Reto 1	20
Reto 2	21
Proyecto	22
Sesión 4 - Mongo DB	26
Reto 1	26
Reto 2	27
Proyecto	28
Sesión 5	35
Reto 1	35
Reto 2	37
Reto 3	38
Proyecto	38
Sesión 6	39
Reto 1	39
Reto 2	40
Reto 3	40
Proyecto	41

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;

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	

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;

Result Grid	Filter Rows:	Export:	Wrap Cell Contents
nombre			
Norrie			
Maxy			

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

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

Result Grid	Filter Rows:	Export:	Wrap Cell Contents
id_puesto	nombre	salario	
1	Analyst Circuit Design manager	28500.00	
2	Junior Executive	10506.47	
3	Director of Sales	28725.56	
4	Staff Scientist	14665.31	
5	Desktop Support Technician	15885.41	
6	Budget/Accounting Analyst III	17131.23	
7	Administrative Assistant III	7674.76	

#	Time	Action	Message
32	20:38:22	SELECT * FROM empleado WHERE id_puesto >= 100 AND id_puesto <= 200 ORDER BY id_puesto ASC	100 row(s) returned
33	20:40:19	SELECT * FROM empleado WHERE id_puesto = 100 OR id_puesto = 200	4 row(s) returned
34	20:42:17	SELECT * FROM empleado WHERE id_puesto IN (100, 200, 300)	6 row(s) returned
35	20:43:50	SELECT * FROM tienda puesto	1000 row(s) returned
36	20:58:01	SELECT nombre FROM empleado WHERE id_puesto = 4	2 row(s) returned
37	20:59:19	SELECT * FROM puesto WHERE salario > 10000	1000 row(s) returned

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;
```

id_articulo	nombre	precio	iva	cantidad
1	Pasta - angel hair	4351.72	999.51	503
2	Soup Campbell - Tomato Basil	2991.35	987.59	604
4	Wine - Cabernet Sauvignon	2652.1	776.1	679
5	Wine - Merlot	1791.42	891.46	248
7	Hammerhead - SeaBarr C&B	5175.47	1012.31	527
8	Wine - Pinot Noir	2694.64	227.16	682
9	Wine - Pinot Noir	1998.46	714.71	476

#

Time

Action

Message

35

20:43:50

SELECT * FROM tienda puesto

1000 row(s) returned

36

20:58:01

SELECT nombre FROM empleado WHERE id_puesto = 4

2 row(s) returned

37

20:59:19

SELECT * FROM puesto WHERE salario > 10000

1000 row(s) returned

38

21:00:22

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

Error Code: 1054. Unknown column 'precio' in 'where clause'

39

21:00:23

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

Error Code: 1054. Unknown column 'precio' in 'where clause'

40

21:01:10

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

787 row(s) returned

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);
```

Result Grid

Filter Rows:

Edit:

Export/Import:

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

#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

Proyecto

#proyecto

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

```
USE classicmodels;
```

✓	18	20:13:09	USE classicmodels	0 row(s) affected
---	----	----------	-------------------	-------------------

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	1002	President
	1056	Patterson	Mary	x4611	mpatterson@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	ghernandez@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	Lou	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
Potter	Andy	6
Morsh	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
Tsieng	Foon Yue	3
Vanauf	George	3
Bondur	Louli	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	state
Mini Gifts Distributors Ltd.	USA	CA
Mini Wheels Co.	USA	CA
Technics Stores Inc.	USA	CA
ToysRUs.com	USA	CA
Boards & Toys Co.	USA	CA
Collectable Mini Designs Co.	USA	CA
Corporate Gift Ideas Co.	USA	CA
Men 'N' US Retailers, Ltd.	USA	CA
The Sharp Gifts Warehouse	USA	CA
West Coast Collectables Co.	USA	CA
Signal Collectables 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, país y límite de crédito de todos los clientes cuyo país sea USA o France y cuyo límite de crédito sea mayor a 100000. Para este ejercicio ten cuidado con los paréntesis.


```
SELECT customerName, country, 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
5	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
8	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
Bentz	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
Yoshida	Juri
Walker	Brydley
Victorino	Wendy
Uris	Braun
Tseng	Jerry
Torini	Daniel
Thompson	Valerie
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
Yoshida	Juri
Walker	Brydley
Victorino	Wendy
Uris	Braun
Tseng	Jerry
Torini	Daniel
Thompson	Steve
Thompson	Valerie
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
HULL	HULL	HULL

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	Naturish Autos	0.00
168	American Souvenirs Inc	0.00
169	Porto Imports Co.	0.00
206	Asian Shopping Network, Co	0.00
125	Havel & Shyazee Co	0.00
1038	1038	1038

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

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

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

id_articulo	nombre	precio	iva	cantidad
1	Chocolate - Feathers	2738.93	12.36	144
2	Pasta - Angel Hair	4391.73	999.51	503
3	Soup Campbell - Tomato Bisque	2991.19	937.59	404
4	Wine - Valpolicella Nera	2625.2	770.1	375
5	Mousse - Banana Chocolate	3701.62	893.46	248
6	Yeast Dry - Fleischman	925.39	524.08	818

#	Time	Action	Message
11	20:00:03	SELECT * FROM articulo WHERE nombre LIKE '%Pasta%'	17 row(s) returned
12	20:01:06	SELECT * FROM articulo WHERE nombre LIKE '%Cavallori%'	0 row(s) returned
13	20:01:18	SELECT * FROM articulo WHERE nombre LIKE '%Cavallori%'	1 row(s) returned
14	20:02:00	SELECT * FROM articulo WHERE nombre LIKE '%-%'	767 row(s) returned

#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;
```

nombre	COUNT(*)
Analog Circuit Design manager	3
Junior Executive	8
Director of Sales	8
Staff Scientist	9
Desktop Support Technician	9
Budget/Accounting Analyst III	4
Accounting Assistant III	3

result 29 x

Output

Action Output

#	Time	Action	Message
36	20:56:28	SELECT nombre, COUNT(*) AS cantidad FROM articulo GROUP BY nombre ORDER BY ca...	562 row(s) returned
37	20:56:06	SELECT nombre, MIN(salario) AS menor, MAX(salario) AS maximo FROM articulo GROUP B...	Error Code: 1054. Unknown column 'salario' in 'field list'
38	20:56:24	SELECT nombre, MIN(salario) AS menor, MAX(salario) AS maximo FROM puesto GROUP BY...	181 row(s) returned
39	20:57:28	SELECT nombre, COUNT(*) FROM puesto GROUP BY nombre	181 row(s) returned

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

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

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

result 30 x

Output

Action Output

#	Time	Action	Message
37	20:56:06	SELECT nombre, MIN(salario) AS menor, MAX(salario) AS maximo FROM articulo GROUP B...	Error Code: 1054. Unknown
38	20:56:24	SELECT nombre, MIN(salario) AS menor, MAX(salario) AS maximo FROM puesto GROUP BY...	181 row(s) returned
39	20:57:28	SELECT nombre, COUNT(*) FROM puesto GROUP BY nombre	181 row(s) returned
40	20:58:30	SELECT nombre, SUM(salario) FROM puesto GROUP BY nombre	181 row(s) returned

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

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

id_empleado	COUNT(*)
2	2
3	2
4	1
5	1
6	2
10	5
15	4

result 31 x

Output

Action Output

#	Time	Action	Message
38	20:56:24	SELECT nombre, MIN(salario) AS menor, MAX(salario) AS maximo FROM puesto GROUP BY...	181 row(s) returned
39	20:57:28	SELECT nombre, COUNT(*) FROM puesto GROUP BY nombre	181 row(s) returned
40	20:58:30	SELECT nombre, SUM(salario) FROM puesto GROUP BY nombre	181 row(s) returned
41	21:00:56	SELECT id_empleado, COUNT(*) FROM venta GROUP BY id_empleado	542 row(s) returned

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

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

id_articulo	COUNT(*)
2	1
3	1
4	2
6	1
10	1
11	1
12	1

result 32 x

Output

Action Output

#	Time	Action	Message
40	20:58:30	SELECT nombre, SUM(salario) FROM puesto GROUP BY nombre	181 row(s) returned
41	21:00:56	SELECT id_empleado, COUNT(*) FROM venta GROUP BY id_empleado	542 row(s) returned
42	21:03:14	SELECT id_venta, COUNT(*) FROM venta GROUP BY id_venta	Error Code: 1055. Sig
43	21:03:36	SELECT id_articulo, COUNT(*) FROM venta GROUP BY id_articulo	633 row(s) returned

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	Rose
Celia	Fulbrook	Harrie
Katya	Baribridge	Fossitt
Robyn	Harcord	Leibold
Hayyim	Verdon	Eastcott
Anaise	Beteriss	Tant

Empleado 39 x

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 x

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
Jeanette	Potes	Heisler	Occupational Therapist
Cassey	Womersley	Chapell	Financial Advisor
Gnni	Rison	Kalnowsky	Physical Therapy Assistant
Lisle	Carlsson	Marquot	Marketing Assistant
Andre	Theurer	Craighill	Tax Accountant

#	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. You
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';
```


[illegible]

customerNumber	checkNumber	amount
141	JE105477	120166.58
NULL	NULL	NULL

customerNumber	checkNumber	amount
213	C343306	43785.34
214	B355020	49460.79
214	L324478	41632.47
224	N742498	43824.36
114	N627552	44894.74
181	OK36719	44460.50
149	B536703	41380.15
266	DC37937	41220.92

payments 15 x

Output

#	Action	Time	Action	Message
156	15 10 16	SELECT A.customerNumber AS pcustomer, B.	1 row(s) returned	
157	15 10 16	SELECT customerNumber, checkNumber,	134 row(s) returned	

	Customer Name
1	Musical Instruments Co.
2	American Securities Inc.
3	Park Imports Co.
4	Asian Shipping Network, Co.
5	Tenishini Autos
6	440-Readers
7	Hessner Shipping Network
8	Kaplan Gifts, Co.

	Action Output
1	Time Action
2	197 21 06 30 SELECT customerNumber FROM payee ... 27 rows returned
3	198 21 06 30 SELECT customerName FROM custom ... 24 rows returned

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

	orderNumber	maximo	minimo	promedio
1	10100	50	22	37.7500
	10101	46	25	35.5000
	10102	41	39	40.0000
	10103	46	32	39.8333
	10104	49	23	34.5769
	10105	50	22	36.3333
	10106	50	26	37.5000
	10107	39	20	28.6250

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
```

Reto 1

ORDER BY clave;

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

ORDER BY clave;

3. ¿Cuál es el total de cada venta?

ORDER BY clave;

JOIN empleado AS e

1. Obtener el puesto de un empleado

CREATE VIEW AVpuesto AS

```
(SELECT p.nombre AS puesto, concat(e.nombre.' 'e.apellido_paterno.'
```

FROM empleado AS e

JOIN puesto AS p

ON e.id_puesto = p.id_puesto):

FROM AVpuerto:

person	employee
Product Engineer	Simcha Goldstein-Libson
Supporting Product ID	Henry Goldstein-Libson
Supporting Product ID	Simcha Goldstein-Libson

```
CREATE VIEW AVarticulosxempleado AS
```

```
(SELECT v.clave, concat(e.nombre,' ', e.apellido_paterno,' ', e.apellido_materno) AS
```

FROM venta AS y

JOIN empleado AS e

ON y id_empleado = e id_empleado

JOIN articulo AS a

ON y id_articulo = a id_articulo):

FROM AVarticulosxempleado:

date	nombre	article
0002-8149	Leslie Jonsson Be-off	Sprouts - Pea
0002-8149	Leslie Conley Puchner	Wine - Sant - Bro 2002, Seco
0002-8149	Ellen Miles Threlkeld	Sauce Tomato Peach
0002-8149	Julia Carter Heavner	Wine - Harlands Rioja Crianza
0002-8149	Sydney Houston Jagel	Tofu/steak
0002-8149	Aquelin Richard Burge	Norwalk - Spencers, Pine, D.o...
0002-8149	Sightl Ted Pearson	Glass - Clear
0002-8149	Rebecca Kuchawich Sells	Beans - Green
0002-8149	Nora O'Suaird Macleale	Walter, Tap
0002-8149	Lucret Lamm Benthley	Butter Sweet
0002-8149	Gustaf Grydenås Gilely	Sour Puss - Tangerine
0002-8149	Pete Stangl Varcoe	Bread - White, Sand
0002-8149	Donna Cohen Kugel	Juice - Orange 1.09

 W:\Hilatic\asmp\leads 10

Output

Action	Time	Action	Message
55	21:12:34	Apply changes to Hilatic\asmp\leads	Changes applied
56	21:12:48	SELECT * FROM Hilatic\asmp\leads	1000 records returned

```
CREATE VIEW AVventas AS
```

(SEI ECT n nombre AS puesto COUNT(y clave) AS ventas

FROM venta AS v

IOIN empleado AS e

ON y 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;
```

productCode	productName	productDescription
S10_1670	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 1100	Official Moto Guzzi logo and insignia, saddle b...
S10_4698	2003 Harley Davidson Eagle Drag Bike	Model features, official Harley Davidson logo a...
S10_4757	1972 Alfa Romeo GTA	Features include: Turnable front wheels, steeri...
S10_4962	1962 Lancia Delta 16V	Features include: Turnable front wheels, steeri...
S12_1099	1968 Ford Mustang	Hood, doors and trunk all open to reveal highl...
S12_1108	2001 Ferrari Enzo	Turnable front wheels, steering function, detail...
S12_1666	1978 Seta Bui	Model features 30 windows, 40 lights & glare re...
S12_2813	2002 Subaru WRX	Official logo and insignia, saddle bags, front en...
S12_2749	1968 Corvair Monza	1:18 scale die-cast about 10" long, doors open...
S12_3380	1968 Dodge Charger	1:12 scale model of a 1968 Dodge Charger. Ho...
S12_3891	1968 Ford F-100	Turnable front wheels, steering function, detail...
S12_3990	1970 Plymouth Hemi Cuda	Very detailed 1970 Plymouth Cuda model in 1:1...

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;
```

orderNumber	status	priceEach
10100	Shipped	136.50
10101	Shipped	51.00
10102	Shipped	75.46
10103	Shipped	56.26
10104	Shipped	108.06
10105	Shipped	107.06
10106	Shipped	52.53
10107	Shipped	45.55
10108	Shipped	95.55
10109	Shipped	45.53
10110	Shipped	214.30
10111	Shipped	110.47
10112	Shipped	121.44
10113	Shipped	94.50

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;
```

orderNumber	fecha	orderNumber	productName	quantityOrdered	priceEach
10000	2003-01-06	1	1976 Chevrolet Nova 1000 Roadster	49	75.29
10000	2003-01-06	2	1911 Ford Town Car	30	53.59
10000	2003-01-06	3	1917 Ford Touring Sedan	30	126.39
10000	2003-01-06	4	1912 Alfa Romeo 8C 2300 Spider Sport	22	75.46
10001	2003-01-09	1	1928 Ford Model T 100	30	107.39
10001	2003-01-09	2	1926 Cadillac V-8 Presidential Limousine	46	44.25
10001	2003-01-09	3	1928 Chevrolet Sedan Coupe	45	51.23
10001	2003-01-09	4	1912 Model A Ford 2-Door	25	108.36
10002	2003-01-10	1	1926 Ford Model B 500 Special Roadster	41	41.22
10002	2003-01-10	2	1927 Lincoln Sedan	26	93.55
10003	2003-01-29	1	1962 Volkswagen Beetle	38	107.34
10003	2003-01-29	2	1926 Ford Five-Engine	22	58.24
10003	2003-01-29	3	19815 GM Hummer H1	21	82.46
10003	2003-01-29	4	1962 Lincoln Sedan 800	42	118.87

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;
```

orderNumber	productName	MSRP	priceEach
10000	1917 Ford Touring Sedan	126.39	126.39
10000	1911 Ford Town Car	53.59	53.59
10000	1912 Alfa Romeo 8C 2300 Spider Sport	75.46	75.46
10000	1928 Ford Model T 100	107.39	107.39
10001	1926 Cadillac V-8 Presidential Limousine	44.25	44.25
10001	1928 Chevrolet Sedan Coupe	51.23	51.23
10001	1912 Model A Ford 2-Door	108.36	108.36
10002	1926 Ford Model B 500 Special Roadster	41.22	41.22
10002	1927 Lincoln Sedan	93.55	93.55
10003	1962 Volkswagen Beetle	107.34	107.34
10003	1926 Ford Five-Engine	58.24	58.24
10003	19815 GM Hummer H1	82.46	82.46
10003	1962 Lincoln Sedan 800	118.87	118.87

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;
```

customerNumber	customerName	orderNumber	state
101	Alfred's Pizzeria	10123	MI
102	Alfred's Pizzeria	10234	MI
103	Alfred's Pizzeria	10345	MI
104	Signal-Gift Stores	10124	MI
105	Signal-Gift Stores	10235	MI
106	Signal-Gift Stores	10346	MI
107	Australian Collectors, Co.	10125	Victoria
108	Australian Collectors, Co.	10236	Victoria
109	Australian Collectors, Co.	10347	Victoria
110	Australian Collectors, Co.	10458	Victoria
111	La Rochelle Gifts	10126	MI
112	La Rochelle Gifts	10237	MI
113	La Rochelle Gifts	10348	MI

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;
```

customerName
Patrol & Dispatch Co.
American Souvenirs Inc.
Pinto Imports Co.
Asian Shopping Network, Co.
NetRack Autos
ABC Bookstore
Measner Shopping Network
Franklin Gifts, Co.
RCME Collectables
Schuler Imports
De Hand Imports
Cramer Spezialitäten, Ltd.
Asian Treasures, Inc.
S&B Desk Rulers, Co.

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;
```

name	lastname	customername	checknumber	amount
Murphy	Diane	0208	0208	
Paterson	James	0208	0208	
Perrell	Jeff	0208	0208	
Perrell	William	0208	0208	
Rockley	Gregory	0208	0208	
Row	Anthony	0208	0208	
James	Joe (c/o DeSantis Inc.)	8121443	311244.3	
James	Joe (c/o DeSantis Inc.)	8025566	85457.87	
James	Joe (c/o DeSantis Inc.)	C236787	1104.26	
James	Joe (c/o DeSantis Inc.)	8149468	4123.68	
James	Joe (c/o DeSantis Inc.)	8126616	4712.66	
James	Joe (c/o DeSantis Inc.)	8126616	4712.66	
James	Joe (c/o DeSantis Inc.)	8121376	11554.4	
James	Joe (c/o DeSantis Inc.)	8121376	4206.36	

SQL > *

SQL

Action Output

#	Action	Message
4	SELECT * customername FROM customers WHERE checknumber = 0208	264 records
5	SELECT * FROM customers, checknumber, amount FROM employees	263 records

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;
```

customerNumber	customerName	orderNumber	Status
903	Akster graphique	10173	Nir
903	Akster graphique	10186	Nir
903	Akster graphique	10195	Nir
112	Sigal-GR Stores	10276	Nir
112	Sigal-GR Stores	10278	Nir
112	Sigal-GR Stores	10296	Nir
114	Australian Collections, Co.	10120	Vicente
114	Australian Collections, Co.	10125	Vicente
114	Australian Collections, Co.	10213	Vicente
114	Australian Collections, Co.	10302	Vicente
114	Australian Collections, Co.	10304	Vicente
114	Australian Collections, Co.	10307	Vicente
118	La Rochelle City	10275	Nir
118	La Rochelle City	10313	Nir
118	La Rochelle City	10275	Nir
118	La Rochelle City	10313	Nir

mysql > X

```

mysql> select * from action;
+----+-----+-----+-----+
| #   | Time          | Action                                     | Message                                         |
+----+-----+-----+-----+
| 7   | 10:43:31      | SELECT * FROM sales, allOrders, customerName, firstOrder WHERE ... | 326 rows affected                               |
| 7   | 10:45:12      | SELECT * FROM sales, allOrders, customerName, c_order ... | 326 rows affected                               |

```

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

[illegible]

```
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

Result 26 x

Output

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);
```

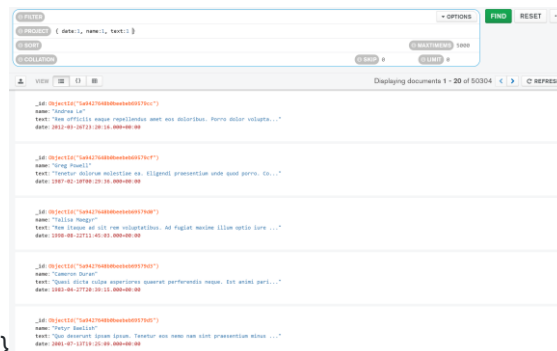
#Menciona las ordenes que se cancelaron y su precio
 SELECT orderNumber, SUM(priceEach * quantityOrdered) AS total, status
 FROM AVO
 GROUP BY orderNumber
 HAVING status LIKE 'Cancelled';

orderNumber	total	status
10167	44167.09	Cancelled
10179	22963.60	Cancelled
10248	41445.21	Cancelled
10253	45445.34	Cancelled
10260	37769.38	Cancelled
10262	47065.36	Cancelled

Sesión 4 - Mongo DB

Reto 1

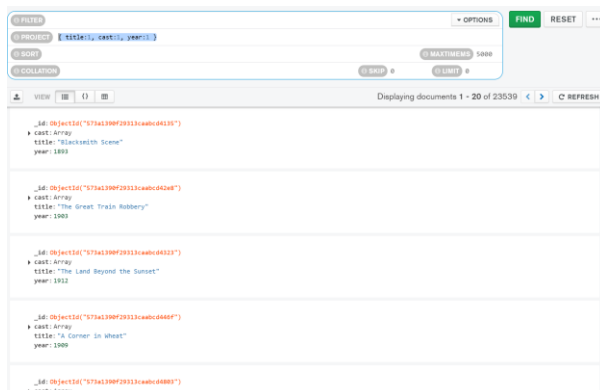
1. Fecha, nombre y texto de cada comentario.



{ 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' }

[illegible]

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

```
Filter: { $or: [ { name: 'Greg Powell' }, { name: 'Mercedes Tyler' } ] }
```

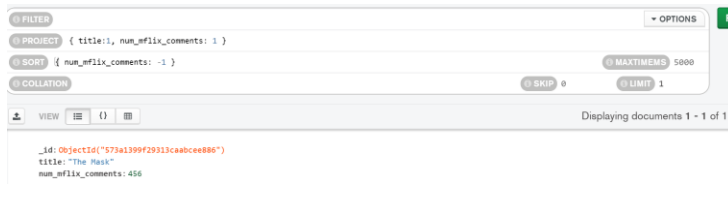
The screenshot shows the AWS IAM console interface. At the top, there's a navigation bar with 'AWS' logo, a search bar, and buttons for 'HOME', 'OPTIONS', 'FIND', and 'RESET'. Below the navigation bar, there's a header 'Adding data...' with a dropdown menu and buttons for 'VIEW', 'EDIT', and 'DELETE'. The main content area displays the details of the 'AmazonECSAccessRole' role. It includes a summary section with the role name, policy name, and a link to the role. Below this, there's a 'Trust relationships' section showing a single trust relationship with the 'AmazonECSAccessRole' role. The role is associated with the 'AmazonECSAccessPolicy' policy. The role is created on 2017-04-18T10:29:38.000-08:00.

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

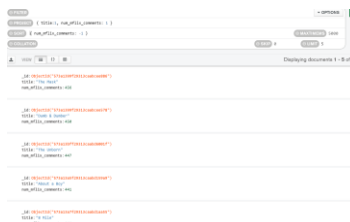


4. ¿Cuál es título de las cinco películas más comentadas?

Project: { title:1, num_mflix_comments: 1 }

Sort: { num_mflix_comments: -1 }

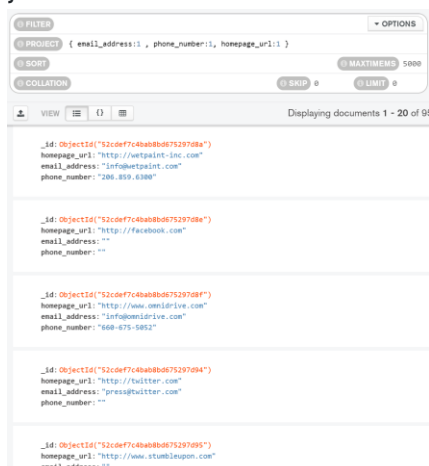
Limit: 5



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
  }
}
```

```
}
```

```
}
```

The screenshot shows the Elasticsearch web interface with a filter applied. The results table displays documents with their source URLs. The first document has a source of "url". The second document has a source of "http://www.banquero.com". The third document has a source of "http://banquero.com". The fourth document has a source of "http://banquero.com". The fifth document has a source of "http://banquero.com". The sixth document has a source of "http://banquero.com". The seventh document has a source of "http://banquero.com". The eighth document has a source of "http://banquero.com". The ninth document has a source of "http://banquero.com". The tenth document has a source of "http://banquero.com".

_id	_source
1	url
2	http://www.banquero.com
3	http://banquero.com
4	http://banquero.com
5	http://banquero.com
6	http://banquero.com
7	http://banquero.com
8	http://banquero.com
9	http://banquero.com
10	http://banquero.com

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

```
{
```

```
  filter: {
```

```
    founded_month: 10
```

```
  },
```

```
  project: {
```

```
    name: 1
```

```
  }
```

```
}
```

The screenshot shows the Elasticsearch web interface with a filter applied. The results table displays documents with their company names. The first document has a name of "Banquero". The second document has a name of "Banquero". The third document has a name of "Banquero". The fourth document has a name of "Banquero". The fifth document has a name of "Banquero". The sixth document has a name of "Banquero". The seventh document has a name of "Banquero". The eighth document has a name of "Banquero". The ninth document has a name of "Banquero". The tenth document has a name of "Banquero".

_id	_source
1	Banquero
2	Banquero
3	Banquero
4	Banquero
5	Banquero
6	Banquero
7	Banquero
8	Banquero
9	Banquero
10	Banquero

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

```
{
```

```
  filter: {
```

```
    founded_year: 2008
```

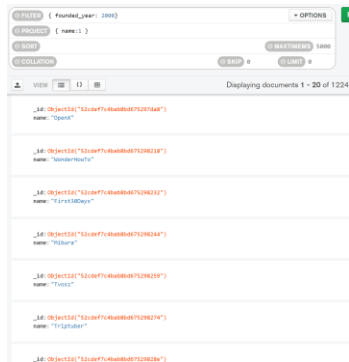
```
  },
```

```
  project: {
```

```
    name: 1
```

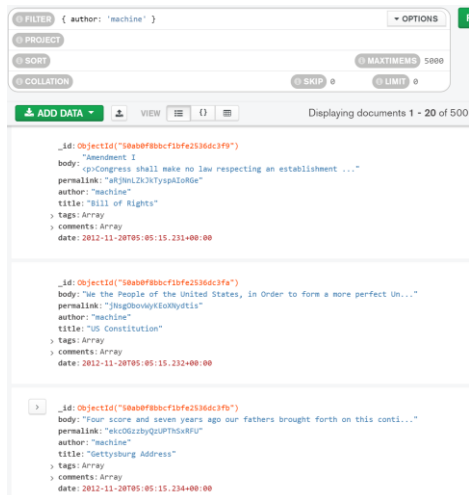
```
  }
```

```
}
```



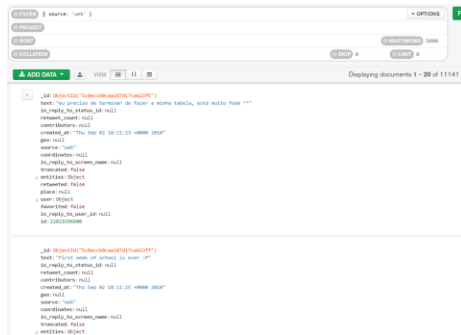
5. Obtén todos los *post* del autor *machine*.

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



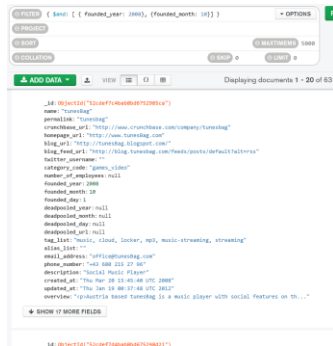
6. Obtén todos los tweets provenientes de la *web*.

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



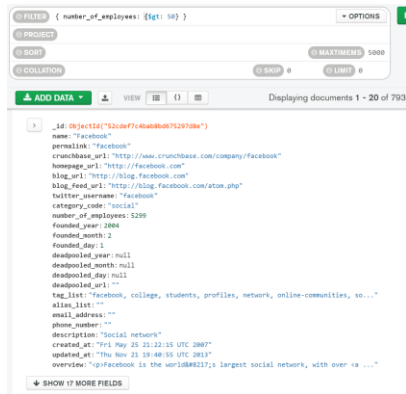
7. Obtén todas las compañías fundadas en octubre del 2008.

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



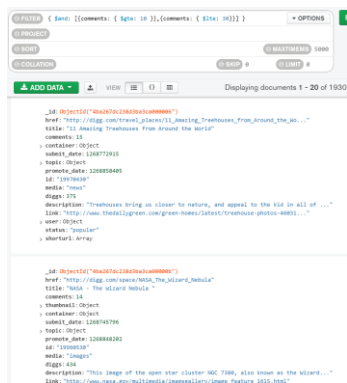
8. Obtén todas las compañías con más de 50 empleados.

```
{
  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.

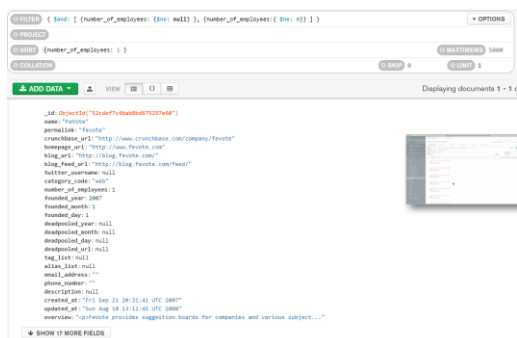
```
{
```



```

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

```

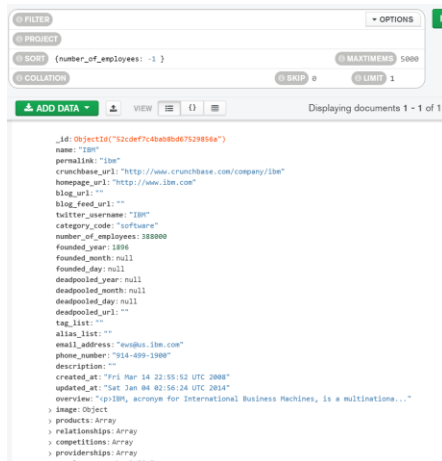


11. Obtén la empresa con el mayor número de empleados.

```

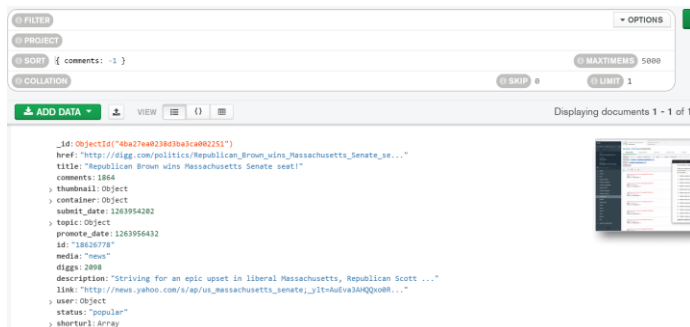
{
  sort: {
    number_of_employees: -1
  },
  limit: 1
}

```



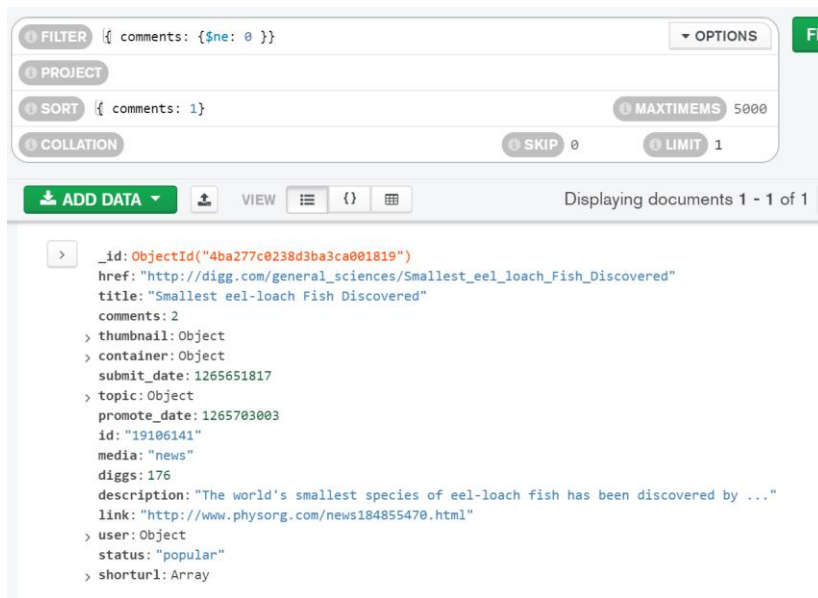
12. Obtén la historia más comentada.

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



13. Obtén la historia menos comentada.

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

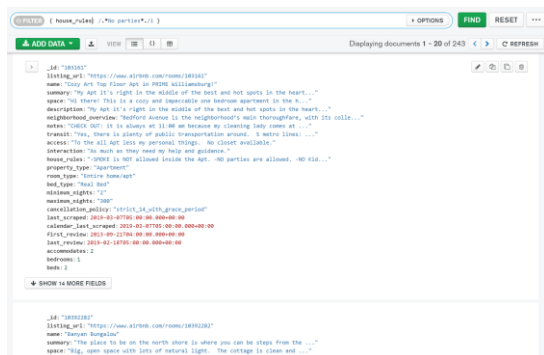


Sesión 5

Reto 1

1. Propiedades que no permitan fiestas.

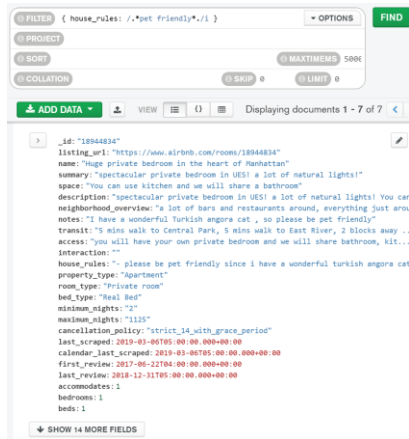
```
{
  filter: {
    house_rules: RegExp('.*No parties*.', i)
  }
}
```



2. Propiedades que admitan mascotas.

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

```
}
```



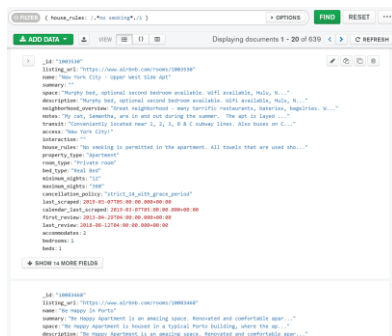
3. Propiedades que no permitan fumadores.

```
{
```

```
  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)
      }
    ]
  }
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

OFFSETS 0 [{ house_rules: /.*no smoking*/i }, { house_rules: /.*no parties*/i }] + OPTIONS	
ADD DATA	VIEW
Displaying documents 1 - 20 of 124	
<pre> { "_id": "10392282", "listing_url": "https://www.airbnb.com/rooms/10392282", "name": "Banyan Bungalow", "summary": "The place to be on the north shore is where you can be steps from the ...", "space": "Big, open space with lots of natural light. The cottage is clean and ...", "description": "The place to be on the north shore is where you can be steps from the ...", "neighborhood_overview": "This desirable neighborhood is comprised of other vacation rentals, i", "notes": "", "transit": "While you might think it nice to arrive and relax there will be many a...", "access": "Private driveway to access the property, parking on site.", "interaction": "While we live on the property and will try to greet you at your arrival...", "house_rules": "No smoking, no pets, no parties. You are welcome to have guests, but ...", "property_type": "Bungalow", "room_type": "Entire home/apt", "bed_type": "Real bed", "minimum_nights": "1", "maximum_nights": "300", "cancellation_policy": "Flexible", "last_scraped": "2019-01-06T05:00:00.000+00:00", "calendar_last_scraped": "2019-01-06T05:00:00.000+00:00", "first_review": "2015-01-15T05:00:00.000+00:00", "last_review": "2019-02-15T05:00:00.000+00:00", "accommodates": 2, "bedrooms": 1, "baths": 1 } </pre>	
SHOW 14 MORE FIELDS	
<pre> { "_id": "10423584", "listing_url": "https://www.airbnb.com/rooms/10423584", "name": "Sunlit Beach Browsing 3-bed House", "summary": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "space": "Spacious space with three bedrooms, including a studio at the back, 300m...", "description": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "neighborhood_overview": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "notes": "", "transit": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "access": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "interaction": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "house_rules": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "property_type": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "room_type": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "bed_type": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "minimum_nights": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "maximum_nights": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "cancellation_policy": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "last_scraped": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "calendar_last_scraped": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "first_review": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "last_review": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "accommodates": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "bedrooms": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "baths": "This peaceful house in North Bondi is 300m to the beach and a minute's..." } </pre>	

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.

```

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

```

OFFSETS 0 [{ review_scores.review_scores_rating': { \$gte: 80 } }, { amenities: [Ethernet] }, { address.country': 'Brazil' }] + OPTIONS	
ADD DATA	VIEW
Displaying documents 1 - 6 of 6	
<pre> { "_id": "10392282", "listing_url": "https://www.airbnb.com/rooms/10392282", "name": "Banyan Bungalow", "summary": "The place to be on the north shore is where you can be steps from the ...", "space": "Big, open space with lots of natural light. The cottage is clean and ...", "description": "The place to be on the north shore is where you can be steps from the ...", "neighborhood_overview": "This desirable neighborhood is comprised of other vacation rentals, i", "notes": "", "transit": "While you might think it nice to arrive and relax there will be many a...", "access": "Private driveway to access the property, parking on site.", "interaction": "While we live on the property and will try to greet you at your arrival...", "house_rules": "No smoking, no pets, no parties. You are welcome to have guests, but ...", "property_type": "Bungalow", "room_type": "Entire home/apt", "bed_type": "Real bed", "minimum_nights": "1", "maximum_nights": "300", "cancellation_policy": "Flexible", "last_scraped": "2019-01-06T05:00:00.000+00:00", "calendar_last_scraped": "2019-01-06T05:00:00.000+00:00", "first_review": "2015-01-15T05:00:00.000+00:00", "last_review": "2019-02-15T05:00:00.000+00:00", "accommodates": 2, "bedrooms": 1, "baths": 1 } </pre>	
SHOW 14 MORE FIELDS	
<pre> { "_id": "10423584", "listing_url": "https://www.airbnb.com/rooms/10423584", "name": "Sunlit Beach Browsing 3-bed House", "summary": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "space": "Spacious space with three bedrooms, including a studio at the back, 300m...", "description": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "neighborhood_overview": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "notes": "", "transit": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "access": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "interaction": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "house_rules": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "property_type": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "room_type": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "bed_type": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "minimum_nights": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "maximum_nights": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "cancellation_policy": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "last_scraped": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "calendar_last_scraped": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "first_review": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "last_review": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "accommodates": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "bedrooms": "This peaceful house in North Bondi is 300m to the beach and a minute's...", "baths": "This peaceful house in North Bondi is 300m to the beach and a minute's..." } </pre>	

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
  }
}}]
```

The screenshot shows the MongoDB Compass interface with two aggregation stages defined and executed.

Stage 1: \$match

- Query:

```
1 = /**
2  * query: The query in MQL.
3  */
4 = {
5   amenities: { $in: [/Ethernet/i, /Wifi/i] }
6 }
```
- Output: A sample of 20 documents. Two documents are visible:
 - `_id: "10009999"`
`listing_url: "https://www.airbnb.com/rooms/10006546"`
`name: "Ribeira Charming Duplex"`
`summary: "Fantastic duplex apartment with three bedrooms located in the historical neighborhood of the Douro River and Ribeira square, our apartment..."`
`space: "Privileged views of the Douro River and Ribeira square, our apartment..."`
`description: "Fantastic duplex apartment with three bedrooms located in the historical neighborhood of the river..."`
`neighborhood_overview: "In the neighborhood of the river several restaurants as..."`
 - `_id: "10009999"`
`listing_url: "https://www.airbnb.com/rooms/10006546"`
`name: "Horto flat with small garden"`
`summary: "One bedroom + sofa-bed in quiet area next to the neighbourhood right next to the Douro River..."`
`space: "Lovely one bedroom + sofa-bed in the perfect for two but..."`
`description: "One bedroom + sofa-bed in quiet area next to the neighbourhood right next to the Douro River..."`

Stage 2: \$group

- Query:

```
1 = /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 = {
6   _id: null,
7   total: {
8     $sum: 1
9   }
10 }
```
- Output: A single document showing the result of the aggregation:

```
_id: null
total: 5308
```

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'}}]
```

The screenshot shows the MongoDB Atlas Query Builder interface. The top stage is '\$match' with a query: `{ "address.country": { "$in": ["Brazil", "I"] } }`. The output shows two sample documents. The bottom stage is '\$count' with a query: `{ "count": { "$sum": 1 } }`. The output shows a single document: `{ "count": 6 }`.

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.

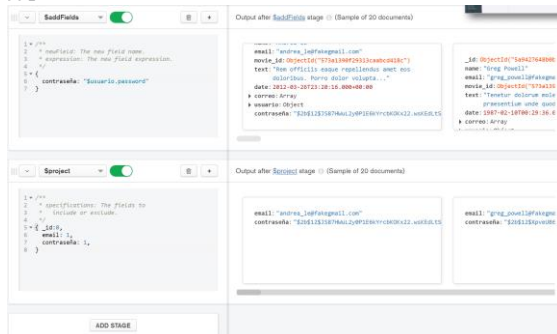
```
{ $match: {
  property_type: "House",
  bedrooms: { $gte: 1 }
}, { $addFields: {
  costo_recamara: { $divide: [ "$price", "$bedrooms" ] }
}, { $group: {
  _id: '$address.country',
  propiedades: {
    $sum: 1
  },
  total: {
    $sum: "$costo_recamara"
  }
}, { $addFields: {
  costo_promedio: {
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

The screenshot shows the MongoDB Atlas Query Builder interface with three stages. The top stage is '\$match' with a query: `{ "property_type": "House", "bedrooms": { "$gte": 1 } }`. The output shows two sample documents. The middle stage is '\$group' with a query: `{ "_id": "$address.country", "propiedades": { "$sum": 1 }, "total": { "$sum": "$costo_recamara" } }`. The output shows two sample documents. The bottom stage is '\$addFields' with a query: `{ "costo_promedio": { "$divide": ["$total", "$propiedades"] } }`. The output shows two sample documents.

Reto 2

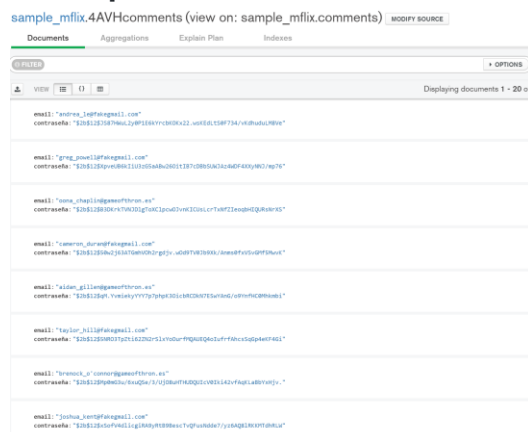
Usando las colecciones `comments` y `users`, se requiere conocer el correo y contraseña de cada persona que realizó un comentario. Construye un pipeline que genere como resultado estos datos.

```
{
  $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.



sample_airbnb.4AVHandybnb (view on: sample_airbnb.listingsAndReviews) userr source

Documents Aggregations Explain Plan Indexes

RESULTS + OPTIONS FIND RESET

view Displaying documents 1 - 9 of 9

1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

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
  }
}
```

sample_mflix.movies

DOCUMENTS 23.5k

TOTAL SIZE 35.9MB

AVG. SIZE 1.6KB

INDEXES 2

TOTAL SIZE 13.1MB

AVG. S 6.6B

Documents

Aggregations

Explain Plan

Indexes

COLLATION project movies unwind

SAVE

SAMPLE MODE

AUTO PREVIEW

\$project

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

```
1 // **
2 * specifications: The fields to
3 * include or exclude.
4 */
5 {
6   _id: 0
7 }
```

peliculas: 47

País: "Poland"

Género: "Comedy"

peliculas: 1

País: "Republic of Macedonia"

Género: "Thriller"

\$sort

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

```
1 // **
2 * Provide any number of field/order pairs.
3 */
4 {
5   peliculas: 1
6 }
```

peliculas: 1

País: "Martinique"

Género: "Drama"

peliculas: 1

País: "Slovakia"

Género: "Family"

Sesión 7

Reto 1

Table Data Import

Configure Import Settings

Detected file format: csv

Encoding: utf-8

Columns:

☒ Source Column

Field Type

☒ movieID

int

☒ title

text

☒ genres

text

movieID	title	genres
1	Toy Story (Animation...
2	Jumanji (19	Adventure...
3	Grumpier O...	ComedyRo...
4	Waiting to...	ComedyDr...
5	Father of th	Comedy

< Back

Next >

Cancel

Reto 2

Reto 3

Import To Collection AVHUsers

Select File

C:\Users\andre\Documents\Data analysis\ml-1m\users.csv

BROWSE

Select Input File Type

JSON

CSV

Options

Select delimiter

COMMA

☒ Ignore empty strings

☐ Stop on errors

cify Fields and Types

userId	gender	age	occupation	zipCode
Number	String	Number	Number	String
1	F	1	10	48067
2	M	56	16	78072
3	M	25	15	55117
4	M	45	7	02460
5	M	25	20	55455
6	F	50	9	55117
7	M	35	1	06810
8	M	25	12	11413
9	M	25	17	61614
10	F	35	1	95370

Import completed

6,040 (100%)

Reto 4

Import To Collection AVH-ratings

Select File

C:\Users\andre\Documents\Data analysis\ml-1m\ratings.csv [BROWSE](#)

Select Input File Type

JSON **CSV**

Options

Select delimiter **COMMA**

☒ Ignore empty strings

☐ Stop on errors

Specify Fields and Types

	<input checked="" type="checkbox"/> userID number	<input checked="" type="checkbox"/> movieID number	<input checked="" type="checkbox"/> rating number	<input checked="" type="checkbox"/> timestamp timestamp
1	1	1193	5	978300760
2	1	661	3	978302109
3	1	914	3	978301968
4	1	3408	4	978300275
5	1	2355	5	978824291
6	1	1197	3	978302268
7	1	1287	5	978302039
8	1	2804	5	978300719
9	1	594	4	978302268
10	1	919	4	978301368

Import completed 5,000 (100%)

Proyecto

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

FILTER { movieID: { \$gte: "4000" } } **OPTIONS** **FIND** **RESET** ...

ADD DATA **VIEW** **LIST** **JSON** **GRID**

Displaying documents 1 - 2 of 2 **REFRESH**

```
{
  "_id": ObjectId("5f18f4e4e62b391a133e7692"),
  "movieID": "4000",
  "title": "Avengers: Endgame (2019)",
  "genres": "Fantasy|Sci-Fi"
}
```

```
{
  "_id": ObjectId("5f18f57fe62b391a133e7693"),
  "movieID": "4001",
  "title": "Glass (2019)",
  "genres": "Drama|Fantasy"
}
```

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

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

The screenshot shows the MongoDB Compass interface. The top bar indicates '25600 Documents in the Collection'. The main area is divided into two panels. The left panel, titled 'Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)', contains a query editor with the following MQL query:

```
1 // **
2 * query: The query in MQL.
3 */
4 {
5   $and: [
6     {Longitude: {$lte: "-103.99"}},
7     {Longitude: {$gte: "-103.40"}},
8     {Latitude: {$lte: "20.90"}},
9     {Latitude: {$gte: "20.70"}}
10  ]
11 }
```

The right panel, titled 'Preview of Documents in the Collection', shows a sample of 5 documents. The first document is:

```
{
  "City": "Andorra la Vella",
  "State/Province": "7",
  "Country": "AD",
  "Postcode": "AD500",
  "Phone Number": "376818720",
  "Timezone": "GMT+1:00 Europe/Andorra",
  "Longitude": "1.53",
  "Latitude": "42.51",
  "_id": "ObjectId(\"5f1f76205a08...")",
  "Brand": "Starbucks",
  "Store Number": "22331-212325",
  "Store Name": "Ajman Drive Th",
  "Ownership Type": "Licensed",
  "Street Address": "1 Street 6",
  "City": "Ajman",
  "State/Province": "AJ"
}
```

The bottom panel, titled 'Output after \$match stage (Sample of 5 documents)', shows the results of the query. The first document is:

```
{
  "_id": "ObjectId(\"5f1f76255a08422dc849cc2e\")",
  "Brand": "Starbucks",
  "Store Number": "30957-101702",
  "Store Name": "Valle Real Guadalajara",
  "Ownership Type": "Licensed",
  "Street Address": "Prolongacion Avenida Santa Margarita, Subancila 3,",
  "City": "Guadalajara",
  "State/Province": "JAL",
  "Brand": "Starbucks",
  "Store Number": "47824-259926",
  "Store Name": "Technology Par",
  "Ownership Type": "Licensed",
  "Street Address": "Carretera a",
  "City": "Guadalajara",
  "State/Province": "JAL"
}
```

Reto 2-H1N1

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

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

```
}
```

The screenshot shows a MongoDB query interface. The filter is set to `{Country: { $ne: "Grand Total" }}`. The sort is set to `{Deaths: -1}`. The result shows a document for the United States of America with 33902 cases and 170 deaths.

```
{
  "_id": ObjectId("5f1f71465a08422dc849a0a1"),
  "Country": "United States of America",
  "Cases": 33902,
  "Deaths": 170,
  "Update Time": "7/6/2009 9:00"
}
```

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

```
{
```

```
  filter: {
```

```
    Country: {
```

```
      $ne: 'Grand Total'
```

```
    },
```

```
    Deaths: {
```

```
      $ne: NaN
```

```
    }  
  },
```

```
  sort: {
```

```
    Deaths: 1
```

```
  },
```

```
  limit: 1
```

```
}
```

The screenshot shows a MongoDB query interface. The filter is set to `{Country: { $ne: "Grand Total" }, Deaths: { $ne: NaN }}`. The sort is set to `{Deaths: 1}`. The result shows a document for Algeria with 5 cases and 0 deaths.

```
{
  "_id": ObjectId("5f1f71465a08422dc849a023"),
  "Country": "Algeria",
  "Cases": 5,
  "Deaths": 0,
  "Update Time": "7/6/2009 9:00"
}
```

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

```
{
```

```
  filter: {
```

```
    Country: {
```

```
      $ne: 'Grand Total'
```

```
    },
```

```
    Deaths: {
```

```

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

```

The screenshot shows a MongoDB query interface. The query is: `{Country: { $ne: "Grand Total" }, Deaths: {$ne: NaN}}`. The results are sorted by `Cases: -1`. The first result is for the United States of America, with 33902 cases and 170 deaths. The update time is 7/6/2009 9:00.

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

```

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

```

The screenshot shows a MongoDB query interface. The query is: `{Cases: 1}`. The results are sorted by `Cases: 1`. The first result is for Bermuda, UKOT, with 1 case and 0 deaths. The update time is 7/6/2009 9:00.

5. ¿Cuál fue el número de muertes promedio?

```

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

```

```

    $avg: "$Deaths"
  }
}}

```

The screenshot shows the MongoDB Compass interface. The top section displays the output of the `$match` stage, showing two documents: one for Algeria with 5 cases and 0 deaths, and one for Antigua and Barbuda with 2 cases and 0 deaths. The bottom section displays the output of the `$group` stage, showing a single document with `avgDeaths: 2.2694444444444444`.

6. ¿Cuál fue el número de casos promedio?

```

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

```

The screenshot shows the MongoDB Compass interface. The top section displays the output of the `$match` stage, showing two documents: one for Algeria with 5 cases and 0 deaths, and one for Antigua and Barbuda with 2 cases and 0 deaths. The bottom section displays the output of the `$group` stage, showing a single document with `avgCases: 475.25707940033317`.

7. Top 5 de países con más muertes

```

{
  filter: {
    Country: {

```

```

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

```

FILTER {Country: { \$ne: "Grand Total" }, Deaths: {\$ne: NaN}} OPTIONS FIND RESET ...
PROJECT
SORT {Deaths: -1} MAXTIMES 5000
COLLATION SKIP 0 LIMIT 5
ADD DATA VIEW {} Refresh Displaying documents 1 - 5 of 5
<pre> _id: ObjectId("5f1f71465a08422dc849a0a1") Country: "United States of America" Cases: 33902 Deaths: 170 Update Time: "7/6/2009 9:00" </pre>
<pre> _id: ObjectId("5f1f71465a08422dc849a121") Country: "United States of America" Cases: 33902 Deaths: 170 Update Time: "7/3/2009 9:00" </pre>
<pre> _id: ObjectId("5f1f71465a08422dc849a19a") Country: "United States of America" Cases: 27717 Deaths: 127 Update Time: "7/1/2009 9:00" </pre>
<pre> _id: ObjectId("5f1f71475a08422dc849a20f") Country: "United States of America" Cases: 27717 Deaths: 127 Update Time: "6/29/2009 9:00" </pre>

8. Top 5 de países con menos muertes

```

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

```


}

FILTER {Country: { \$ne: "Grand Total" }, Deaths: { \$ne: NaN }}

PROJECT

SORT {Deaths: 1}

MAXTIMES 5000

SKIP 0

LIMIT 5

ADD DATA **VIEW** **JSON** **SQL** **TABLE**

Displaying documents 1 - 5 of 5

```
{
  "_id": "ObjectId('5f1f71465a08422dc849a023')",
  "Country": "Algeria",
  "Cases": 5,
  "Deaths": 0,
  "Update Time": "7/6/2009 9:00"
}
```

```
{
  "_id": "ObjectId('5f1f71465a08422dc849a024')",
  "Country": "Antigua and Barbuda",
  "Cases": 2,
  "Deaths": 0,
  "Update Time": "7/6/2009 9:00"
}
```

```
{
  "_id": "ObjectId('5f1f71465a08422dc849a027')",
  "Country": "Austria",
  "Cases": 19,
  "Deaths": 0,
  "Update Time": "7/6/2009 9:00"
}
```

```
{
  "_id": "ObjectId('5f1f71465a08422dc849a028')",
  "Country": "Bahamas",
  "Cases": 7,
  "Deaths": 0,
  "Update Time": "7/6/2009 9:00"
}
```

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 }]
```

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

```
1 // **
2 // Provide any number of field/order pairs.
3 //
4 {
5   Cases: -1
6 }
```

```
{
  "_id": "Mainland China",
  "Cases": 2369152
}
```

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

```
1 // **
2 // Provide the number of documents to limit.
3 //
4 1]
```

```
{
  "_id": "Mainland China",
  "Cases": 2369152
}
```

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}]
```

The screenshot shows a MongoDB Atlas query editor interface. On the left, there's a code editor with a query. On the right, there's a preview of the output. Below the code editor, there's a control bar with a 'Limit' dropdown set to 1, a toggle switch, and a button to view the output after the limit stage. The output preview shows a single document with the following fields:

```
{
  "_id": "Mainland China",
  "Deaths": 65325
}
```

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"
  }
}]
```

}}

The screenshot displays a data processing workflow with two stages. The first stage, **\$addFields**, is active and shows a sample of 20 documents. The second stage, **\$group**, is also active and shows a sample of 1 document.

Stage 1: \$addFields

Code editor content:

```
1 /**
2  * newField: The new field name.
3  * expression: The new field expression.
4  */
5 {
6   Lat: {$convert: {input: "$Lat", to: "double"}},
7   Long: {$convert: {input: "$Long", to: "double"}}
8 }
```

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

Document 1:

```
{
  "_id": ObjectId("5f1f70765a08422dc8498dbf"),
  "Date": "3/3/2020 12:00",
  "Province": "Hubei",
  "Region": "Mainland China",
  "Last Update": "2020-03-03T11:43:02",
  "Confirmed": "67217",
  "Deaths": "2835",
  "Recovered": "36208"
}
```

Document 2:

```
{
  "_id": ObjectId("5f1f70765a08422dc8498dbf"),
  "Date": "3/3/2020 12:00",
  "Province": "",
  "Region": "South Korea",
  "Last Update": "2020-03-03T09:43:02",
  "Confirmed": "5186",
  "Deaths": "28",
  "Recovered": "30"
}
```

Stage 2: \$group

Code editor content:

```
1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   _id: null,
7   meanLat: {
8     $avg: "$Lat"
9   },
10  meanLong: {
11    $avg: "$Long"
12  }
13 }
```

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

```
{
  "_id": null,
  "meanLat": 31.709998321342923,
  "meanLong": 29.72109808153477
}
```

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