Contenido

Sesión 1: Fundamentos de SQL

Work 1

Reto 1
Usando la base de datos tienda, muestra la descripción de las tablas articulo, puesto y venta. Por cada tipo de dato que encuentras llena la siguiente tabla, a mano. Usa la Documentación de MySQL como referencia.

int	Datos números enteros
varchar	Cadenas de caracteres
timestamp	Fechas
double	Datos números con decimales

describe articulo;

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

describe puesto;

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

describe venta;

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

Reto 2

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

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

SELECT nombre FROM empleado WHERE id_puesto = 4;

nombre
Norrie
Maxy

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

SELECT * FROM puesto WHERE salario > 10000;

nombre
Norrie
Maxy

- **¿Qué articulos 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
2	Pasta - Angel Hair	4391.73	959.51	503
3	Soup Campbells - Tomato Bisque	2991.35	587.59	604
4	Wine - Valpolicella Masi	2625.2	770.1	575
5	Mousse - Banana Chocolate	3701.62	893.46	248
7	Nantucket - Kiwi Berry Cktl.	5579.47	1012.33	527
8	Wine - Fontanafredda Barolo	2684.64	327.16	682
9	Lotus Rootlets - Canned	1996.46	324.72	636
10	Wine - Vovray Sec Domaine Huet	6066.99	890.47	397
11	Cake - Pancake	5271.11	821.28	64
12	Chocolate Liqueur - Godet White	1616.78	612.63	929
13	Appetizer - Southwestern	1771.99	248.16	589

- **¿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);

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

Reto 3

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

SELECT * FROM tienda 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 Sesión 1

-- 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

-- 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

-- 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)

-- 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

-- 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

/*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

/*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
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

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

select customername, country, creditlimit from customers where country in("USA", "France") and creditlimit > 100000;

customername	country	creditlimit
La Rochelle Gifts	France	118200.00
Mini Gifts Distributors Ltd.	USA	210500.00
Land of Toys Inc.	USA	114900.00
Saveley & Henriot, Co.	France	123900.00
Muscle Machine Inc	USA	138500.00
Diecast Classics Inc.	USA	100600.00
Collectable Mini Designs Co.	USA	105000.00
Marta's Replicas Co.	USA	123700.00
Mini Classics	USA	102700.00
Corporate Gift Ideas Co.	USA	105000.00
Online Diecast Creations Co.	USA	114200.00

/*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

/*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

/*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

/*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

/*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

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

select contactlastname, contactfirstname from customers order by contactlastname desc, contactfirstname asc;

contactlastname	contactfirstname
Young	Dorothy
Young	Jeff
Young	Julie
Young	Mary
Yoshido	Juri
Walker	Brydey
Victorino	Wendy
Urs	Braun
Tseng	Jerry

/*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, contactfirstname, creditlimit from customers order by creditlimit desc limit 5;

customernumber	customername	contactfirstname	creditlimit
141	Euro+ Shopping Channel	Diego	227600.00
124	Mini Gifts Distributors Ltd.	Susan	210500.00
298	Vida Sport, Ltd	Mihael	141300.00
151	Muscle Machine Inc	Jeff	138500.00
187	AV Stores, Co.	Rachel	136800.00

/*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, contactfirstname, creditlimit from customers where creditlimit > 0
order by creditlimit limit 5;
```

customernumber	customername	contactfirstname	creditlimit
219	Boards & Toys Co.	Mary	11000.00
103	Atelier graphique	Carine	21000.00
198	Auto-Moto Classics Inc.	Leslie	23000.00
381	Royale Belge	Pascale	23500.00
473	Frau da Collezione	Franco	34800.00

Sesión 2: Agrupaciones y subconsultas

Work 2

Reto 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
233	Pasta - Cannelloni, Sheets, Fresh	2316.37	605.55	307
327	Pasta - Cappellini, Dry	6994.49	766.18	828
361	Pasta - Penne, Rigate, Dry	2222.62	584.88	276

¿Qué artículos incluyen la palabra Cannelloni en su nombre?

select * from articulo where nombre like '%Cannelloni%';

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

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

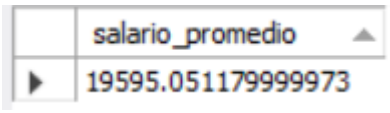
select * from articulo where nombre like '%-%';

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

Reto 2

-- ¿Cuál es el promedio de salario de los puestos?

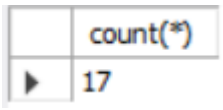
select avg(salario) as salario_promedio from puesto;



	salario_promedio
▶	19595.051179999973

-- ¿Cuántos artículos incluyen la palabra Pasta en su nombre?

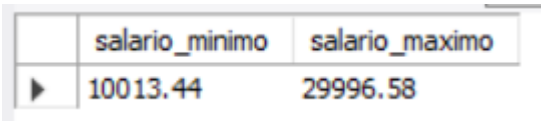
select count(*) from articulo where nombre like 'Pasta%';



	count(*)
▶	17

-- ¿Cuál es el salario mínimo y máximo?

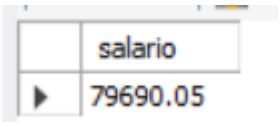
select min(salario) salario_minimo, max(salario) salario_maximo from puesto;



	salario_minimo	salario_maximo
▶	10013.44	29996.58

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

select sum(a.salario) as salario from (select * from puesto order by id_puesto desc limit 5) as a;



	salario
▶	79690.05

Reto 3

-- ¿Cuántos registros hay por cada uno de los puestos?

select nombre, count(*) from puesto group by nombre;

	nombre	count(*)
▶	Nurse Practicioner	16
	Marketing Manager	16
	Project Manager	15
	Research Associate	14
	Editor	14
	Business Systems Developme...	14
	Data Coordiator	13
	Physical Therapy Assistant	13
	Assistant Professor	13

-- ¿Cuánto dinero se paga en total por puesto?

select nombre, sum(salario) as salario from puesto group by nombre;

nombre	salario
Project Manager	319920.94999999995
Marketing Manager	314634.11000000004
Nurse Practicioner	296384.04
Editor	295937.8
Business Systems Developme...	287700.92000000004
Data Coordiator	270557.2
Assistant Professor	259330.11999999997
Sales Representative	258233.40000000002
Physical Therapy Assistant	250663.83000000002

-- Cuál es el número total de ventas por vendedor?

select id_empleado, count(*) total_ventas_vendedor from venta group by id_empleado;

id_empleado	total_ventas_vendedor
510	6
12	5
227	5
544	5
15	4
142	4
216	4
233	4

-- ¿Cuál es el número total de ventas por artículo?

select id_articulo, count(*) total_ventas_articulo from venta group by id_articulo;

id_articulo	total_ventas_articulo
966	5
32	4
43	4
98	4
289	4
313	4
322	4
343	4

Reto 4

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

select nombre, apellido_paterno from empleado where id_puesto in

(select id_puesto from puesto where salario < 20000);

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

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

select id_empleado, min(total_ventas) cantidad_min_ventas, max(total_ventas) cantidad_max_ventas from

(select clave, id_empleado, count(*) as total_ventas from venta group by clave, id_empleado) as a group by id_empleado;

	id_empleado	cantidad_min_ventas	cantidad_max_ventas
▶	263	2	2
	775	2	2
	417	2	2
	108	2	2
	898	2	2
	569	1	1
	413	1	2
	765	1	1

-- ¿Cuál es el nombre del puesto de cada empleado?

select nombre, apellido_paterno, nombre_puesto from (select nombre, apellido_paterno, id_puesto from empleado) a left join (select id_puesto, nombre nombre_puesto from puesto) b

on a.id_puesto = b.id_puesto;

	nombre	apellido_paterno	nombre_puesto
▶	Enrichetta	Bodechon	Product Engineer
	Morey	Bowskill	Budget/Accounting Analyst IV
	Jeannette	Potes	Occupational Therapist
	Cassey	Womersley	Financial Advisor
	Gnni	Risom	Physical Therapy Assistant
	Lisle	Carlsson	Marketing Assistant
	Andre	Theurer	Tax Accountant
	Land	Locksley	Product Engineer

Proyecto Sesión 2

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

select id_empleado, apellido_paterno, nombre from empleado where nombre like "a%";

id_empleado	apellido_paterno	nombre
7	Theurer	Andre
10	Bidewell	Aldridge
23	Cornehl	Alleyn
38	Mea	Allina
41	Ilchenko	Ailbert

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

select id_empleado, apellido_paterno, nombre from empleado where nombre like "%on";

id_empleado	apellido_paterno	nombre
54	Corkish	Dalston
152	Osban	Kevon
189	Selby	Rhiamon
240	Haselgrove	Milton
273	Baldelli	Vinson

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

select id_empleado, apellido_paterno, nombre from empleado where nombre like "%on%";

id_empleado	apellido_paterno	nombre
15	Chaize	Donaugh
54	Corkish	Dalston
55	Yitshak	Conney
128	Prickett	Lonnie
152	Osban	Kevon

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

select id_empleado, apellido_paterno, nombre from empleado where length(nombre) = 3 and nombre like 'T%m';

id_empleado	apellido_paterno	nombre
NULL	NULL	NULL

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

select id_empleado, apellido_paterno, nombre from empleado where nombre not like "b%";

id_empleado	apellido_paterno	nombre
911	Stepto	Aurelia
936	Melhuish	Austina
972	Bourley	Avivah
119	Sambedge	Cad
822	Calow	Camila

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

use classicmodels;

select productcode, productname from products where productcode like "%|_20%" escape '|';

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

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

select orderNumber, sum(quantityordered) total from orderdetails group by orderNumber;

orderNumber	total
10222	717
10106	675
10165	670
10386	650
10168	642

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

select year(orderDate), count(*) numero_ordenes from orders group by year(orderDate);

year(orderDate)	numero_ordenes
2003	111
2004	151
2005	64

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

select lastname, firstname, officecode from employees where officecode in

(select officecode from offices where country = 'USA');

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

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

select customernumber, checknumber, amount pago from payments where amount in
(select max(amount) pago from payments);

customernumber	checknumber	pago
141	JE105477	120166.58

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

select avg(amount) pago_promedio from payments;

select customernumber, checknumber, amount cantidad from payments where amount >
(select avg(amount) pago from payments);

pago_promedio
32431.645531

customernumber	checknumber	cantidad
189	NM916675	32538.74
112	HQ55022	32641.98
379	FR499138	32680.31
311	FA728475	32723.04
458	DD995006	33145.56
112	ND748579	33347.88
172	EH208589	33383.14
202	IQ627690	33594.58

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

select contactfirstname, contactlastname, customernumber from customers where customernumber not in (select customernumber from orders);

contactfirstname	contactlastname	customernumber
Zbyszek	Piestrzeniewicz	125
Keith	Franco	168
Isabel	de Castro	169
Brydey	Walker	206
Horst	Kloss	223
Alejandra	Camino	237
Renate	Messner	247
Peter	Franken	273

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

select max(numero_productos) max_productos, min(numero_productos) min_productos,

avg(numero_productos) promedio_productos from (select productcode, count(*) as numero_productos from orderdetails group by productcode) a;

max_productos	min_productos	promedio_productos
53	24	27.4862

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

select estado, count(*) numero_ordenes from (select customernumber, ordernumber, (select state from customers a where a.customernumber = b.customernumber) as estado from orders b) c group by estado;

estado	numero_ordenes
NULL	180
CA	45
MA	23
NY	18
PA	9
Victoria	8
CT	8
NSW	8
BC	4

Sesión 3: Joins y Vistas

Work 3

Reto 1

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

```
select nombre, apellido_paterno from empleado a
join venta b
on a.id_empleado = b.id_empleado
order by 1;
```

nombre	apellido_paterno
Aaren	Pryce
Aaren	Pryce
Aaron	Klossmann
Aaron	Klossmann
Abagael	Buzzing
Abbie	Tibald
Abigail	Shama
Adan	Berthelmot

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

```
select nombre from articulo a
join venta b
on a.id_articulo = b.id_articulo
order by 1;
```

nombre
Alize Gold Passion
Alize Gold Passion
Alize Red Passion
Alize Sunset
Alize Sunset
Alize Sunset
Allspice - Jamaican
Amaretto

-- ¿Cuál es el total de cada venta?

```
select clave, round(sum(precio), 0) as precio_total from venta a
```

join articulo b

on a.id_articulo = b.id_articulo

group by clave;

clave	precio_total
0228-3661	3714
52125-277	340583
0049-0032	321525
13107-062	249071
47335-894	223650
51655-951	190821
52380-1865	162361
69128-001	174311

Proyecto Sesión 3

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

use classicmodels;

select p.productcode, p.productname, p.productdescription from products p;

productcode	productname	productdescription
S10_1678	1969 Harley Davidson Ultimate Chopper	This replica features working kickstand, front suspension, gear-shi...
S10_1949	1952 Alpine Renault 1300	Turnable front wheels; steering function; detailed interior; detaile...
S10_2016	1996 Moto Guzzi 1100i	Official Moto Guzzi logos and insignias, saddle bags located on side...
S10_4698	2003 Harley-Davidson Eagle Drag Bike	Model features, official Harley Davidson logos and insignias, detac...
S10_4757	1972 Alfa Romeo GTA	Features include: Turnable front wheels; steering function; detaile...
S10_4962	1962 LanciaA Delta 16V	Features include: Turnable front wheels; steering function; detaile...
S12_1099	1968 Ford Mustang	Hood, doors and trunk all open to reveal highly detailed interior fe...
S12_1108	2001 Ferrari Enzo	Turnable front wheels; steering function; detailed interior; detaile...

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

select c.orderNumber, a.state, sum(c.priceEach) as costo_total from customers a

join orders b

on a.customerNumber = b.customerNumber

join orderdetails c

on b.orderNumber = c.orderNumber

group by c.orderNumber, a.state

order by 3 desc;

orderNumber	state	costo_total
10287	NULL	1801.52
10165	NULL	1794.94
10181	NULL	1760.39
10159	CA	1687.00
10310	NULL	1656.26
10126	NULL	1623.71
10204	NY	1619.73
10306	NULL	1612.26
10185	MA	1605.19

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

```
select a.ordernumber, c.orderdate, a.orderLineNumber, b.productName, a.quantityOrdered,
priceEach, b.buyprice

from orderdetails a

join products b

on a.productCode = b.productCode

join orders c

on a.orderNumber = c.orderNumber

order by 1;
```

ordernumber	orderdate	orderLineNumber	productName	quantityOrdered	priceEach	buyprice
10100	2003-01-06	3	1917 Grand Touring Sedan	30	136.00	86.70
10100	2003-01-06	2	1911 Ford Town Car	50	55.09	33.30
10100	2003-01-06	4	1932 Alfa Romeo 8C2300 Spider ...	22	75.46	43.26
10100	2003-01-06	1	1936 Mercedes Benz 500k Roadster	49	35.29	21.75
10101	2003-01-09	4	1932 Model A Ford J-Coupe	25	108.06	58.48
10101	2003-01-09	1	1928 Mercedes-Benz SSK	26	167.06	72.56
10101	2003-01-09	3	1939 Chevrolet Deluxe Coupe	45	32.53	22.57
10101	2003-01-09	2	1938 Cadillac V-16 Presidential Li...	46	44.35	20.61
10102	2003-01-10	2	1937 Lincoln Berline	39	95.55	60.62

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

```
select a.orderNumber, b.productname, b.MSRP, a.priceeach, b.buyprice from orderdetails a

left join products b

on a.productCode = b.productCode
```

order by 1;

orderNumber	productname	MSRP	priceeach	buyprice
10100	1917 Grand Touring Sedan	170.00	136.00	86.70
10100	1911 Ford Town Car	60.54	55.09	33.30
10100	1932 Alfa Romeo 8C2300 Spider Sport	92.03	75.46	43.26
10100	1936 Mercedes Benz 500k Roadster	41.03	35.29	21.75
10101	1932 Model A Ford J-Coupe	127.13	108.06	58.48
10101	1928 Mercedes-Benz SSK	168.75	167.06	72.56
10101	1939 Chevrolet Deluxe Coupe	33.19	32.53	22.57
10101	1938 Cadillac V-16 Presidential Limousine	44.80	44.35	20.61
10102	1937 Lincoln Berline	102.74	95.55	60.62

-- Para estas consultas usa LEFT JOIN

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

```
select c.customernumber, concat(c.contactfirstname,  ' ' ,c.contactlastname) as
nombre_cliente, o.ordernumber, c.state from customers c
left join orders o
on c.customernumber = o.customernumber
order by 1;
```

customernumber	nombre_cliente	ordernumber	state
103	Carine Schmitt	10123	NULL
103	Carine Schmitt	10298	NULL
103	Carine Schmitt	10345	NULL
112	Jean King	10124	NV
112	Jean King	10278	NV
112	Jean King	10346	NV
114	Peter Ferguson	10120	Victoria

sult 4 ×

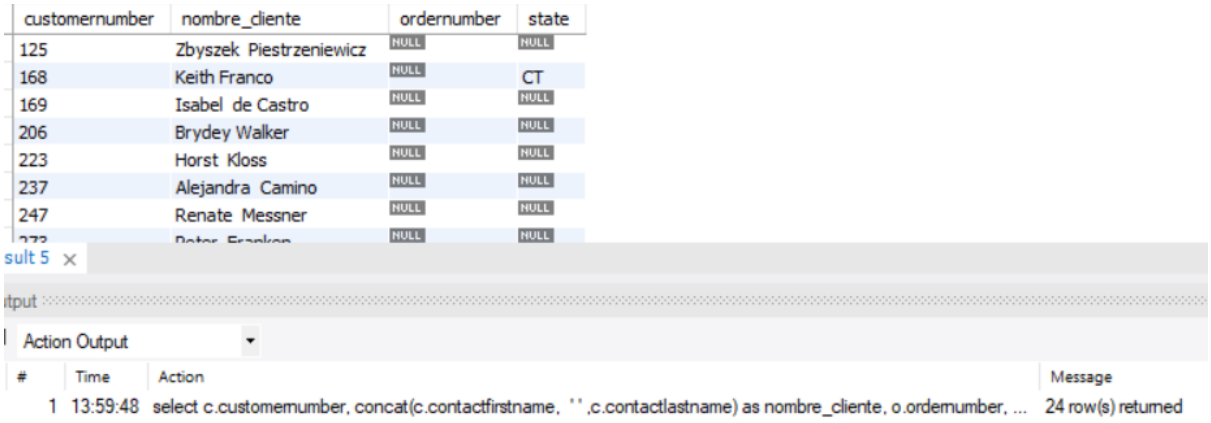
tput

Action Output

#	Time	Action	Message
1	13:59:07	select c.customernumber, concat(c.contactfirstname, ' ' ,c.contactlastname) as nombre_cliente, o.ordernumber, ...	350 row(s) returned

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

```
select c.customernumber, concat(c.contactfirstname,  ' ' ,c.contactlastname) as
nombre_cliente, o.ordernumber, c.state from customers c
left join orders o
on c.customernumber = o.customernumber
where o.customernumber is null;
```



customernumber	nombre_cliente	ordernumber	state
125	Zbyszek Piestrzeniewicz	NULL	NULL
168	Keith Franco	NULL	CT
169	Isabel de Castro	NULL	NULL
206	Brydey Walker	NULL	NULL
223	Horst Kloss	NULL	NULL
237	Alejandra Camino	NULL	NULL
247	Renate Messner	NULL	NULL

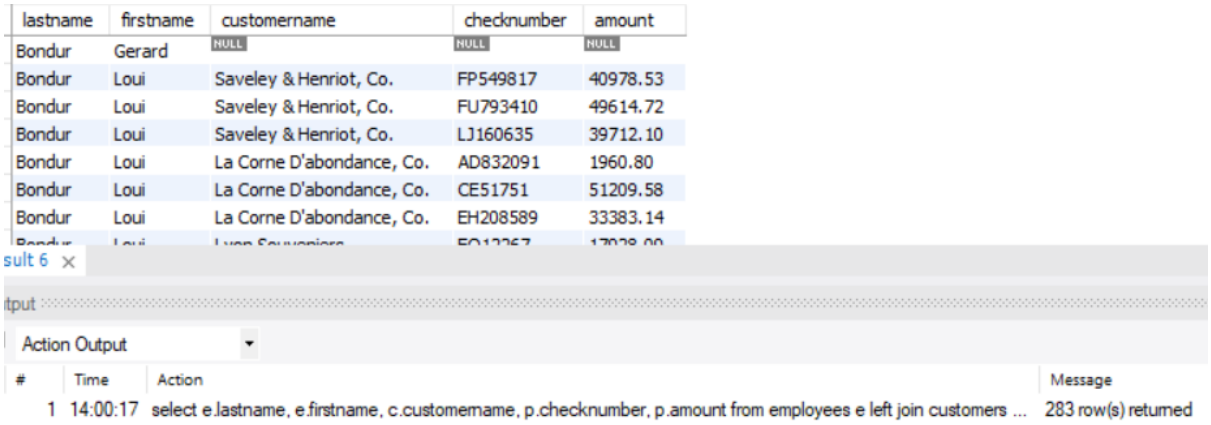
sult 5

Action Output

#	Time	Action	Message
1	13:59:48	select c.customernumber, concat(c.contactfirstname, ' ',c.contactlastname) as nombre_cliente, o.ordernumber, ...	24 row(s) returned

-- 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
order by 1;
```



lastname	firstname	customername	checknumber	amount
Bondur	Gerard	NULL	NULL	NULL
Bondur	Loui	Saveley & Henriot, Co.	FP549817	40978.53
Bondur	Loui	Saveley & Henriot, Co.	FU793410	49614.72
Bondur	Loui	Saveley & Henriot, Co.	LJ160635	39712.10
Bondur	Loui	La Corne D'abondance, Co.	AD832091	1960.80
Bondur	Loui	La Corne D'abondance, Co.	CE51751	51209.58
Bondur	Loui	La Corne D'abondance, Co.	EH208589	33383.14

sult 6

Action Output

#	Time	Action	Message
1	14:00:17	select e.lastname, e.firstname, c.customername, p.checknumber, p.amount from employees e left join customers ...	283 row(s) returned

-- Para estas consultas usa RIGHT JOIN

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

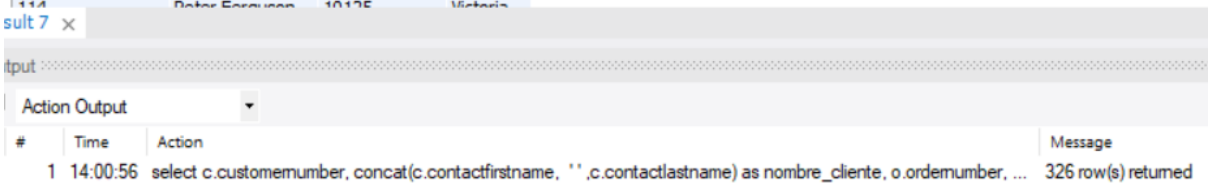
-- 8.5. Obtén el número de cliente, nombre de cliente, número de orden y estado de cada cliente.

```
select c.customernumber, concat(c.contactfirstname, ' ' ,c.contactlastname) as
nombre_cliente, o.ordernumber, c.state from customers c
```

right join orders o

on c.customernumber = o.customernumber;

customernumber	nombre_cliente	ordernumber	state
103	Carine Schmitt	10123	NULL
103	Carine Schmitt	10298	NULL
103	Carine Schmitt	10345	NULL
112	Jean King	10124	NV
112	Jean King	10278	NV
112	Jean King	10346	NV
114	Peter Ferguson	10120	Victoria



-- 8.6. Obtén los clientes que no tienen una orden asociada.

select c.customernumber, concat(c.contactfirstname, ' ' ,c.contactlastname) as nombre_cliente, o.ordernumber, c.state from customers c

right join orders o

on c.customernumber = o.customernumber

where o.customernumber is null;

customernumber	nombre_cliente	ordernumber	state

-- 8.7. 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

right join customers c

on e.employeenumber = c.salesrepemployeenumber

right join payments p

on c.customernumber = p.customernumber

order by 1;

lastname	firstname	customername	checknumber	amount
Bondur	Loui	Saveley & Henriot, Co.	FP549817	40978.53
Bondur	Loui	Saveley & Henriot, Co.	FU793410	49614.72
Bondur	Loui	Saveley & Henriot, Co.	LJ160635	39712.10
Bondur	Loui	La Corne D'abondance, Co.	AD832091	1960.80
Bondur	Loui	La Corne D'abondance, Co.	CE51751	51209.58
Bondur	Loui	La Corne D'abondance, Co.	EH208589	33383.14
Bondur	Loui	Lyon Souveniers	EQ12267	17928.09

sult 9 ×

tput

Action Output

#	Time	Action	Message
1	14:02:20	select e.lastname, e.firstname, c.customername, p.checknumber, p.amount from employees e right join customers...	273 row(s) returned

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

CREATE VIEW WH_vista_p2 AS

(select c.orderNumber, a.state, sum(c.priceEach) as costo_total from customers a

join orders b

on a.customerNumber = b.customerNumber

join orderdetails c

on b.orderNumber = c.orderNumber

group by c.orderNumber, a.state

order by 3 desc

);

select * from WH_vista_p2;

orderNumber	state	costo_total
10287	NULL	1801.52
10165	NULL	1794.94
10181	NULL	1760.39
10159	CA	1687.00
10310	NULL	1656.26
10126	NULL	1623.71
10204	NY	1619.73

```
CREATE VIEW WH_vista_p3 AS
(select a.ordernumber, c.orderdate, a.orderLineNumber, b.productName, a.quantityOrdered,
priceEach, b.buyprice
from orderdetails a
join products b
on a.productCode = b.productCode
join orders c
on a.orderNumber = c.orderNumber
);
select * from WH_vista_p3;
```

ordernumber	orderdate	orderLineNumber	productName	quantityOrdered	priceEach	buyprice
10100	2003-01-06	3	1917 Grand Touring Sedan	30	136.00	86.70
10100	2003-01-06	2	1911 Ford Town Car	50	55.09	33.30
10100	2003-01-06	4	1932 Alfa Romeo 8C2300 Spider Sport	22	75.46	43.26
10100	2003-01-06	1	1936 Mercedes Benz 500k Roadster	49	35.29	21.75
10101	2003-01-09	4	1932 Model A Ford J-Coupe	25	108.06	58.48
10101	2003-01-09	1	1928 Mercedes-Benz SSK	26	167.06	72.56
10101	2003-01-09	3	1939 Chevrolet Deluxe Coupe	45	32.53	22.57

```
CREATE VIEW WH_vista_p4 AS
select a.orderNumber, b.productname, b.MSRP, a.priceeach, b.buyprice from orderdetails a
left join products b
on a.productCode = b.productCode;
select * from WH_vista_p4;
```

orderNumber	productname	MSRP	priceeach	buyprice
10100	1917 Grand Touring Sedan	170.00	136.00	86.70
10100	1911 Ford Town Car	60.54	55.09	33.30
10100	1932 Alfa Romeo 8C2300 Spider Sport	92.03	75.46	43.26
10100	1936 Mercedes Benz 500k Roadster	41.03	35.29	21.75
10101	1932 Model A Ford J-Coupe	127.13	108.06	58.48
10101	1928 Mercedes-Benz SSK	168.75	167.06	72.56
10101	1939 Chevrolet Deluxe Coupe	33.19	32.53	22.57

Sesión 4: Fundamentos de MongoDB

Work 4

Reto 1

Fecha, nombre y texto de cada comentario.

PROJECT {date:1, name:1, text:1, _id:0}

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

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

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

Título, elenco y año de cada película.

PROJECT {title:1, cast:1, year:1, _id:0}

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

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

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


Nombre y contraseña de cada usuario.

PROJECT {name:1, password:1, _id:0}


```
name: "Ned Stark"
password: "$2b$12$UREFwsRUoyF0CRqGNK0LzO0HM/jLhgUCNNIJ9RJAqMUQ74crlJ1Vu"
```

```
name: "Robert Baratheon"
password: "$2b$12$yGqxLG9LZpXA2xVDhuPnSOZd.VURVkz7wgOLY3pnO0s7u2S1ZO32y"
```

```
name: "Jaime Lannister"
password: "$2b$12$6vz7wiwO.EI5Rilvq1zUc./9480gb1uPtXcahDxIadgyC3PS8XCUK"
```


Reto 2

¿Qué comentarios ha hecho Greg Powell?

FILTER {name:"Greg Powell"}

```
_id: ObjectId("5a9427648b0beebeb69579cf")
name: "Greg Powell"
email: "greg_powell@fakegmail.com"
movie_id: ObjectId("573a1390f29313caabcd41b1")
text: "Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Co..."
date: 1987-02-10T00:29:36.000+00:00
```

```
_id: ObjectId("5a9427648b0beebeb6957afe")
name: "Greg Powell"
email: "greg_powell@fakegmail.com"
movie_id: ObjectId("573a1391f29313caabcd754b")
text: "Rem nostrum nobis saepe eaque itaque nemo. Fugit dignissimos nisi sapi..."
date: 2013-03-26T16:20:03.000+00:00
```

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

¿Qué comentarios han hecho Greg Powell o Mercedes Tyler?

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

```
_id: ObjectId("5a9427648b0beebeb69579cf")
name: "Greg Powell"
email: "greg_powell@fakegmail.com"
movie_id: ObjectId("573a1390f29313caabcd41b1")
text: "Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Co..."
date: 1987-02-10T00:29:36.000+00:00
```

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

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

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

PROJECT {num_mflix_comments: 1}

SORT {num_mflix_comments:-1}

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

¿Cuál es título de las cinco películas más comentadas?

PROJECT {title:1, num_mflix_comments: 1, _id:0}

SORT {num_mflix_comments:-1}

LIMIT 5

```
title: "The Mask"
num_mflix_comments: 456
```

```
title: "Dumb & Dumber"
num_mflix_comments: 450
```

```
title: "The Unborn"
num_mflix_comments: 447
```

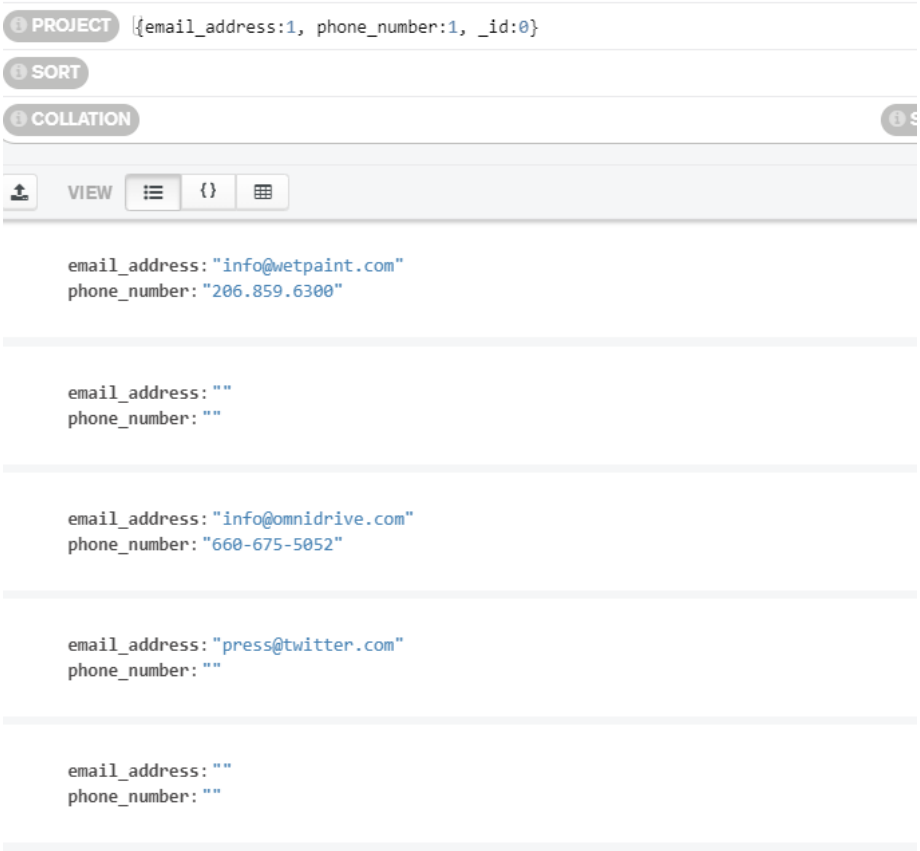
```
title: "About a Boy"
num_mflix_comments: 441
```

```
title: "8 Mile"
num_mflix_comments: 441
```

Proyecto Sesión 4

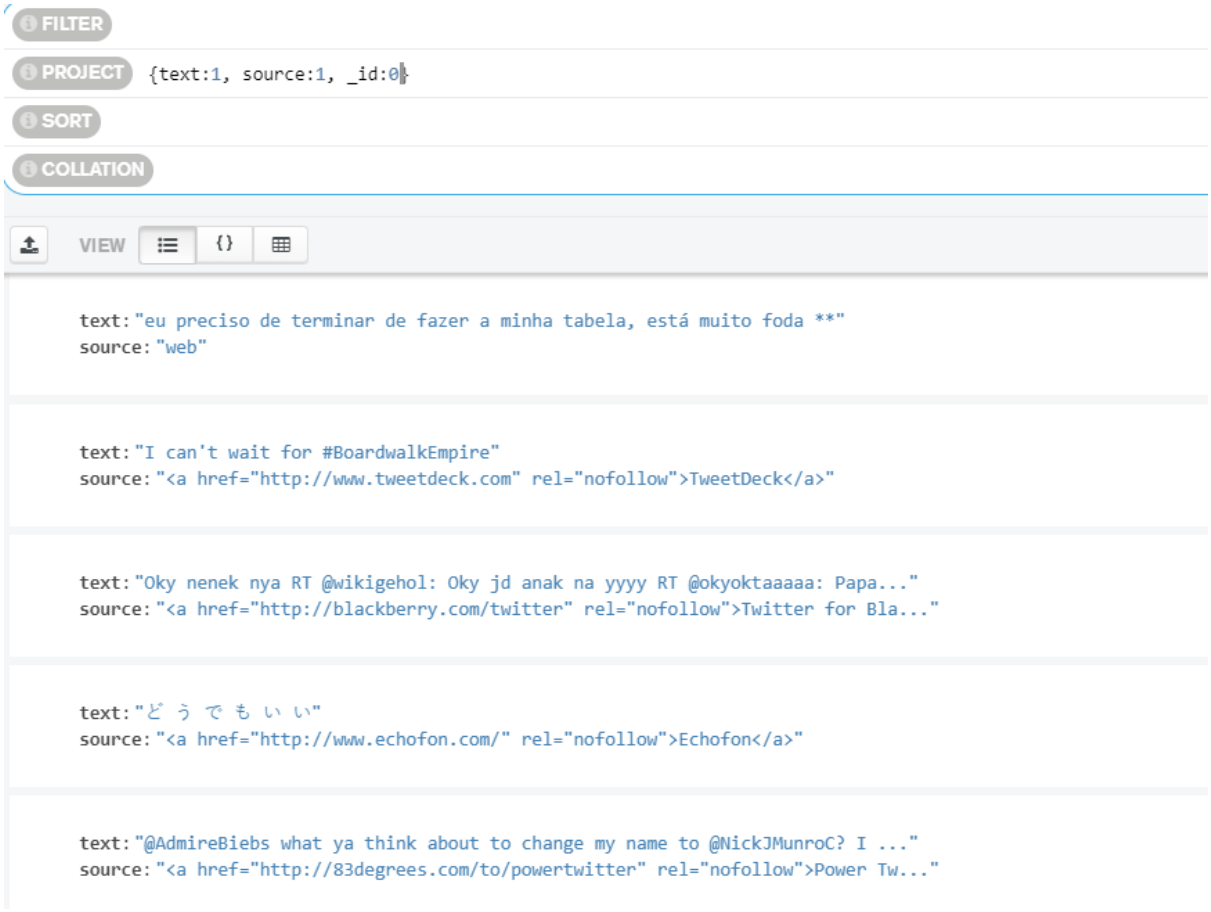
1. Obtén los datos de contacto de cada compañía.

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



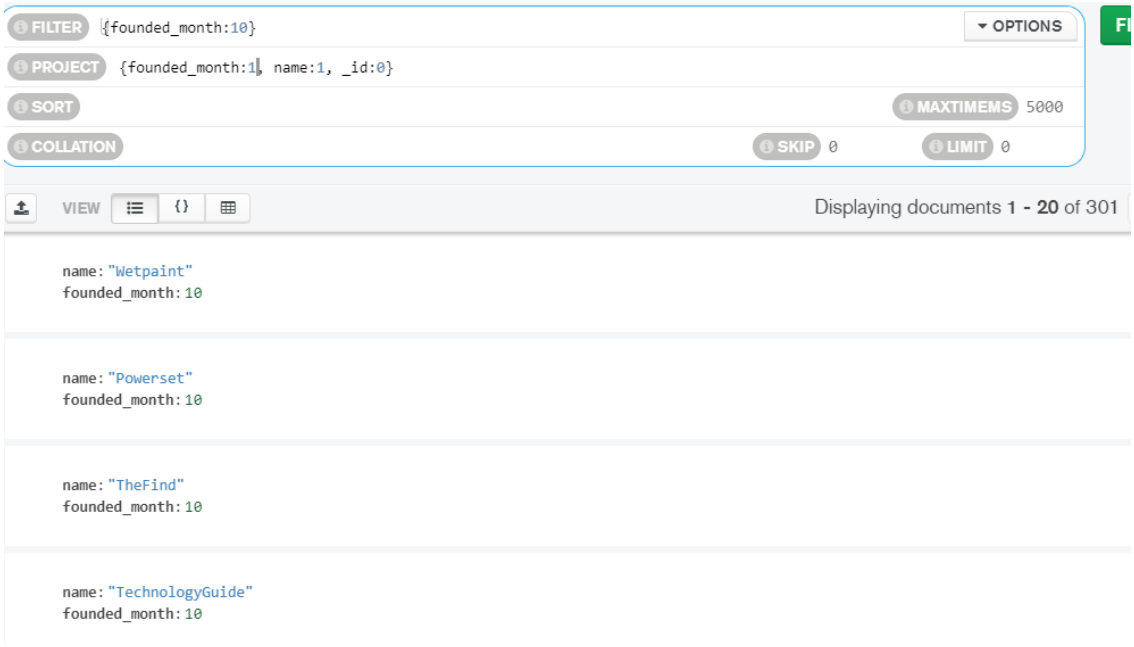
2. Obtén la fuente de cada tweet.

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



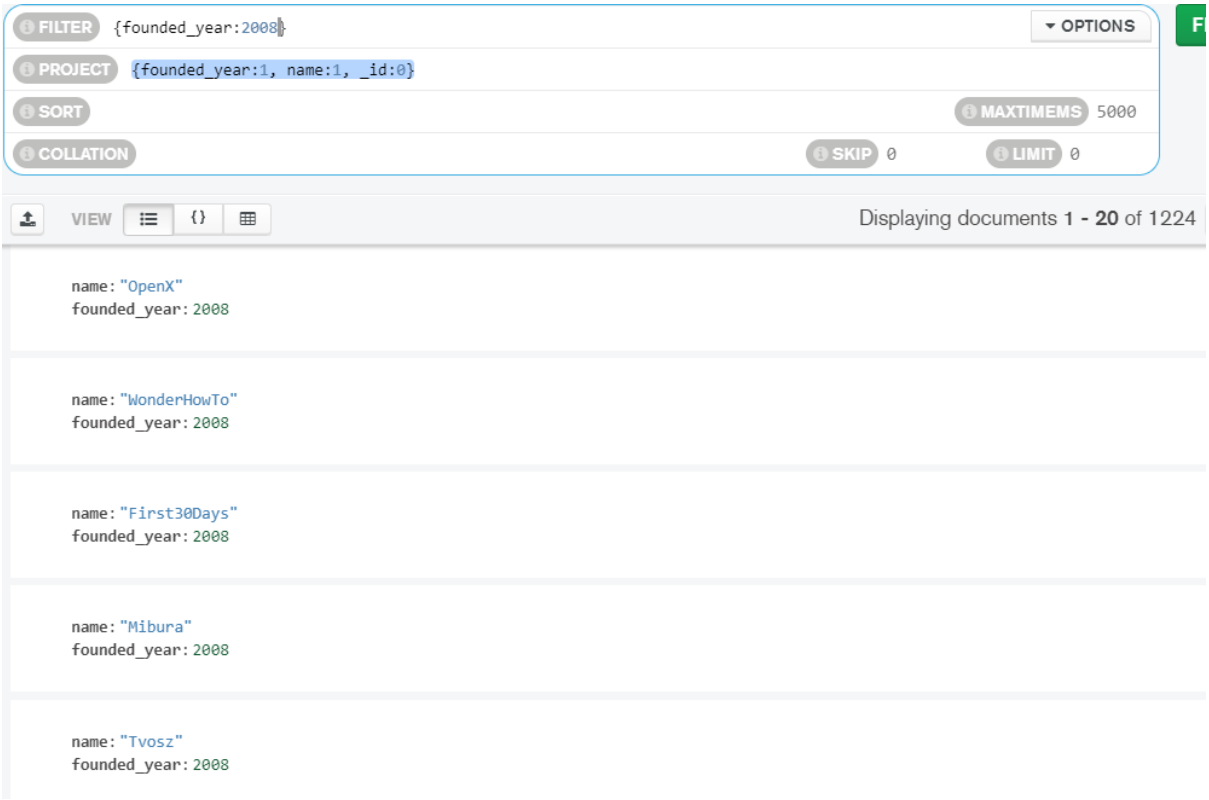
3. Obtén el nombre de todas las compañias fundadas en octubre.

```
{
 filter: {
  founded_month: 10
 },
 project: {
  founded_month: 1,  name: 1,  _id: 0
 }
}
```



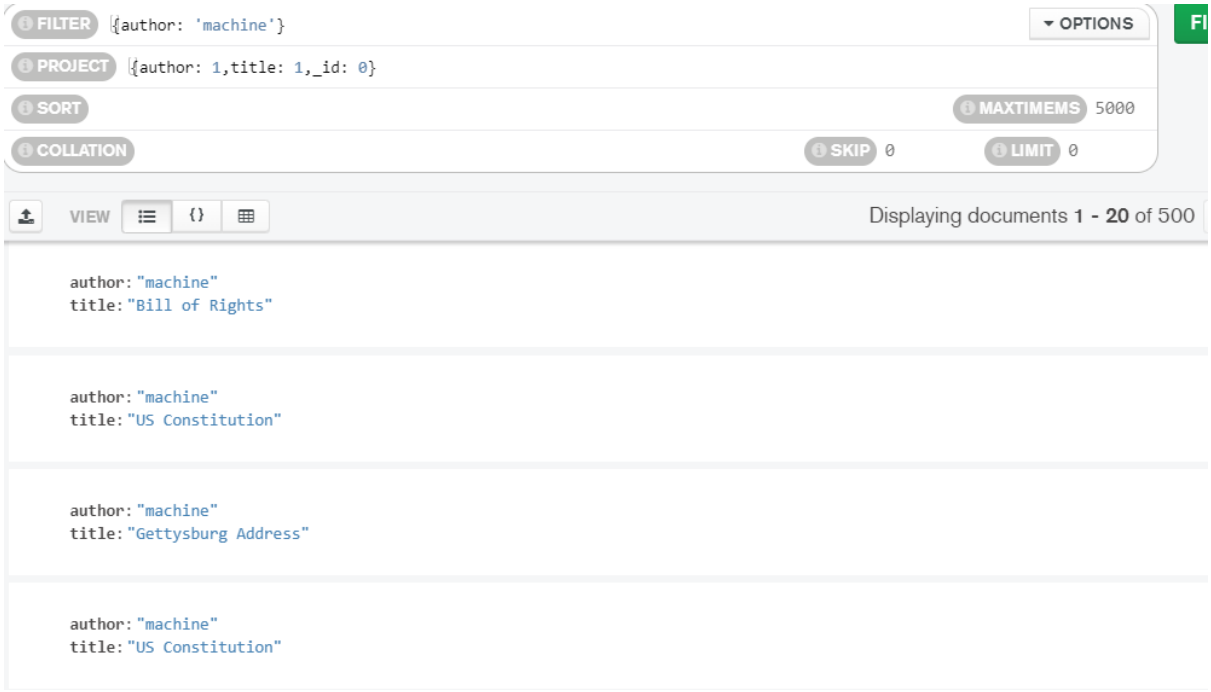
4. Obtén el nombre de todas las compañías fundadas en 2008.

```
{
 filter: {
  founded_year: 2008
 },
 project: {
  founded_year: 1,  name: 1,  _id: 0
 }
}
```



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

```
{
 filter: {
  author: 'machine'
 },
 project: {
  author: 1,  title: 1,  _id: 0
 }
}
```

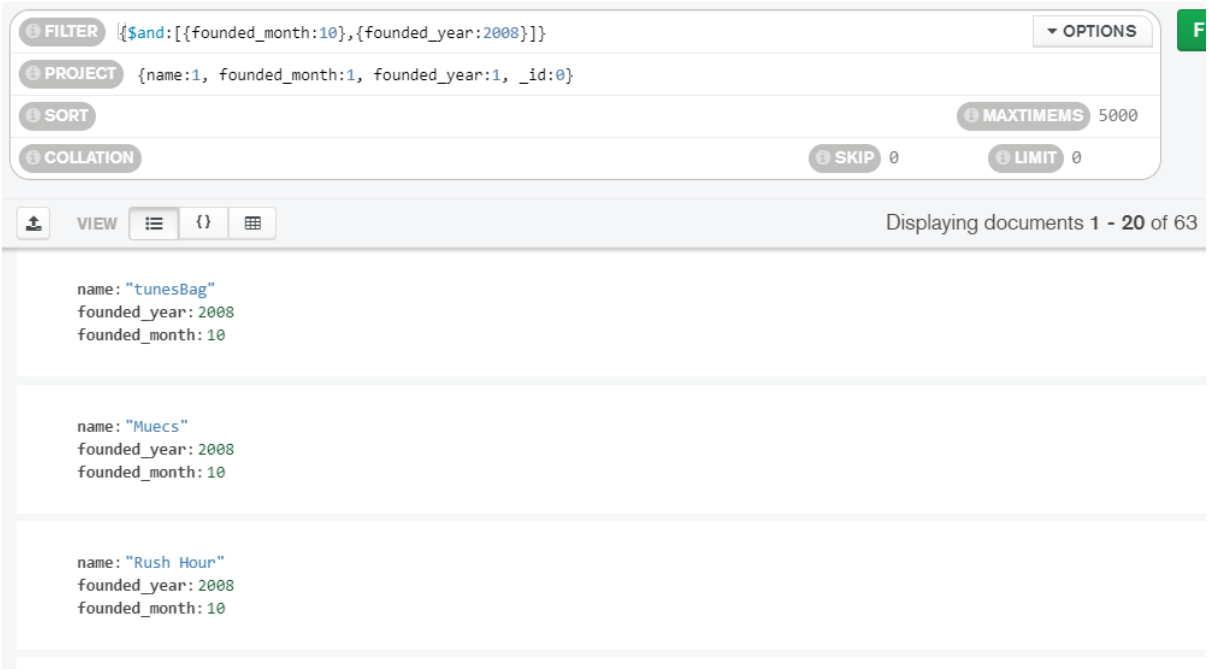


6. Obtén todos los tweets provenientes de la `web`.

```
{
 filter: {
  source: 'web'
 },
 project: {
  source: 1,  text: 1,  _id: 0
 }
}
```

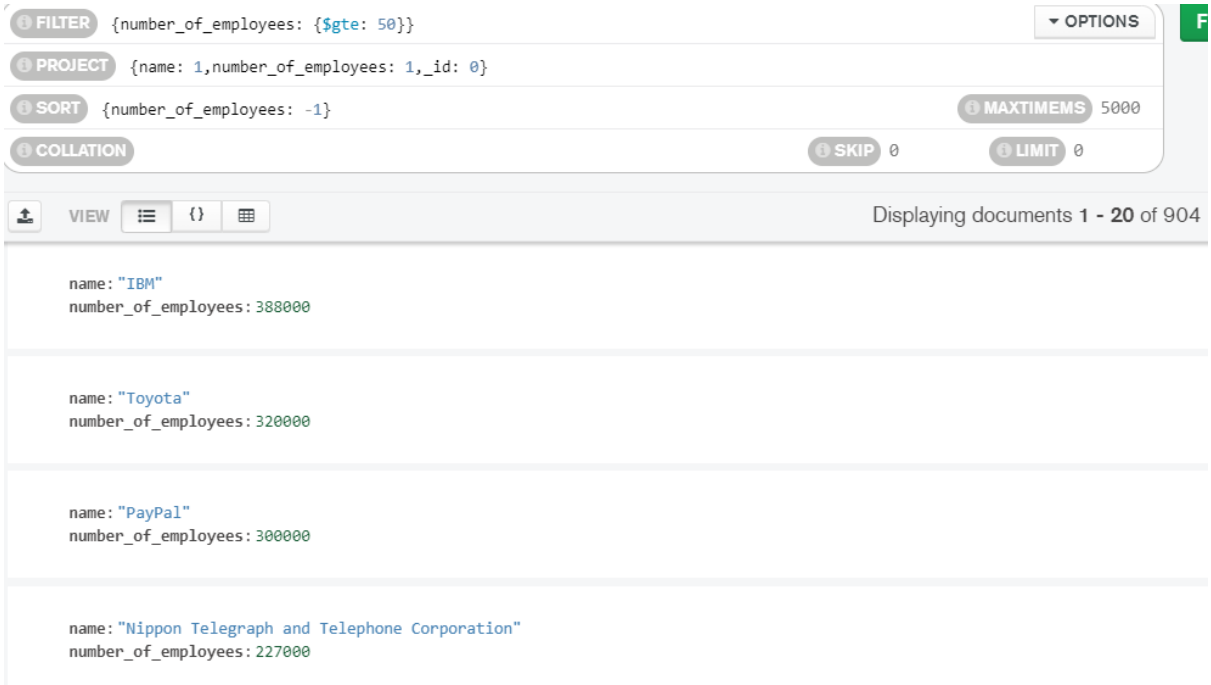
7. Obtén todas las compañías fundadas en octubre del 2008.

```
{
 filter: {
  $and: [
   {
    founded_month: 10
   },
   {
    founded_year: 2008
   }
  ]
 },
 project: {
  name: 1, founded_month: 1, founded_year: 1, _id: 0
 }
}
```



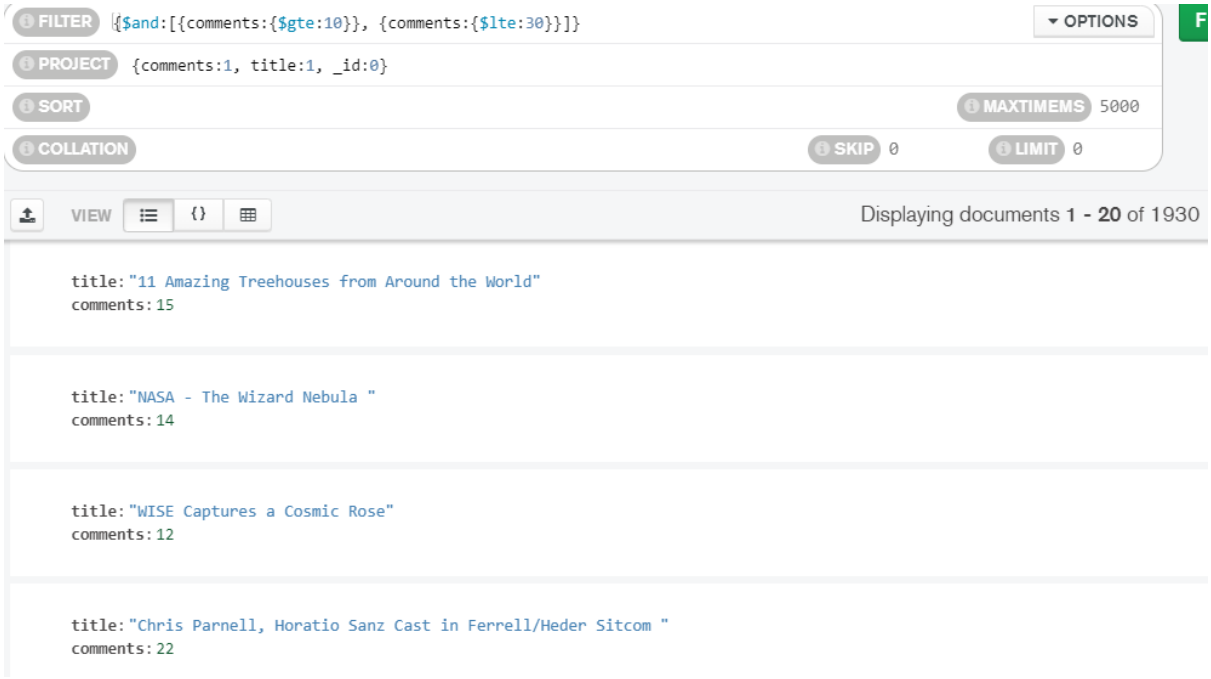
8. Obtén todas las compañias con más de 50 empleados.

```
{
 filter: {
  number_of_employees: {
   $gte: 50
  }
 },
 project: {
  name: 1,  number_of_employees: 1,  _id: 0
 },
 sort: {
  number_of_employees: -1
 }
}
```



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

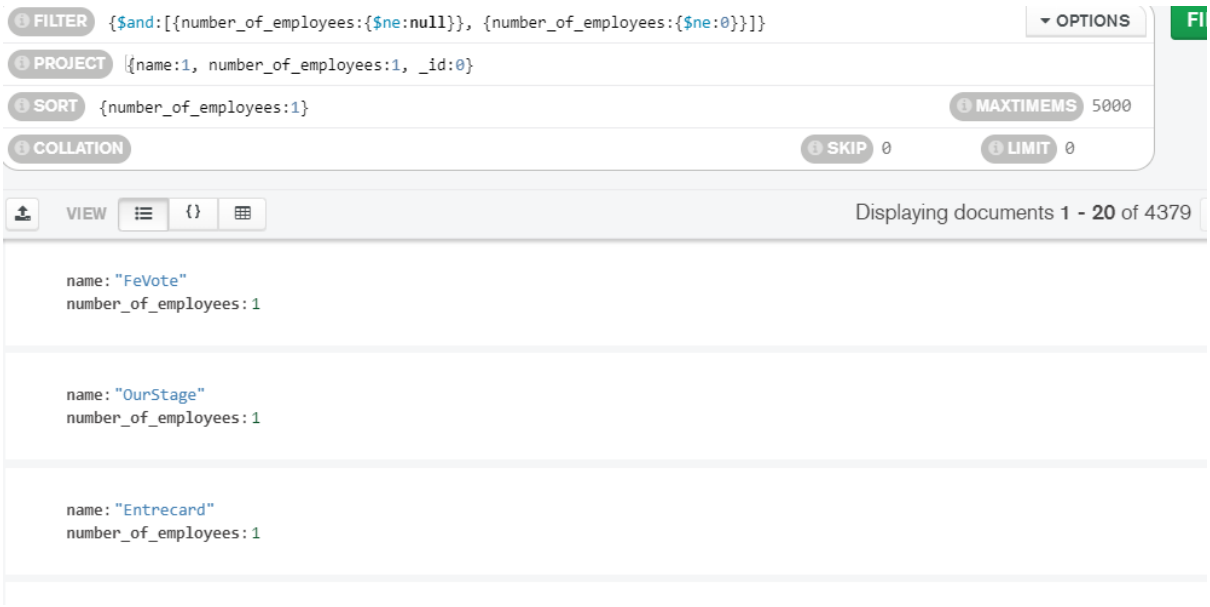
```
{
 filter: {
  $and: [
   {
    comments: {
     $gte: 10
    }
   },
   {
    comments: {
     $lte: 30
    }
   }
  ]
 },
 project: {
  comments: 1,  title: 1,  _id: 0
 }
}
```



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

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

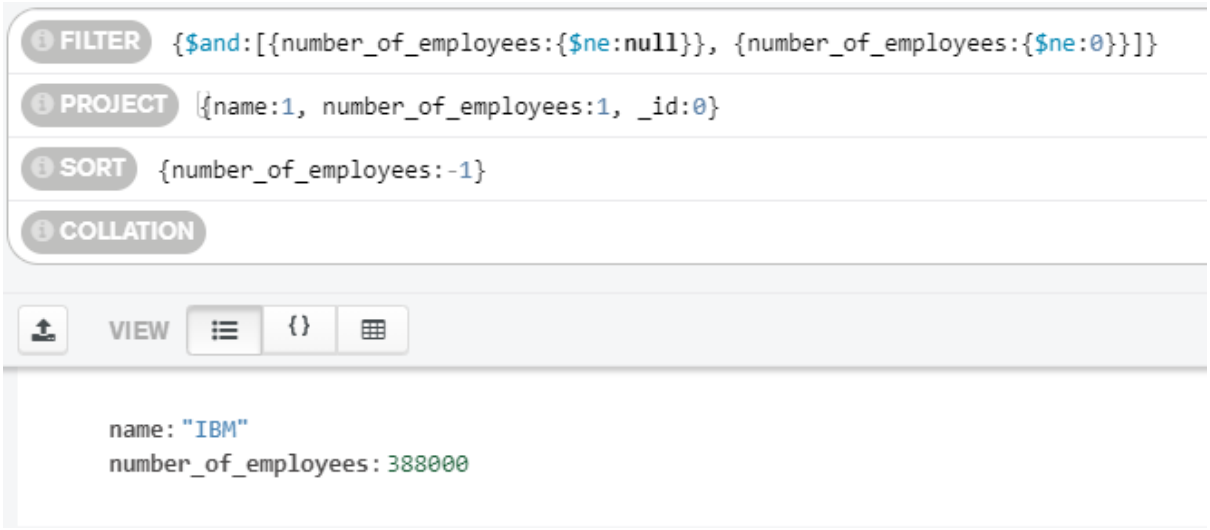
}
}



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

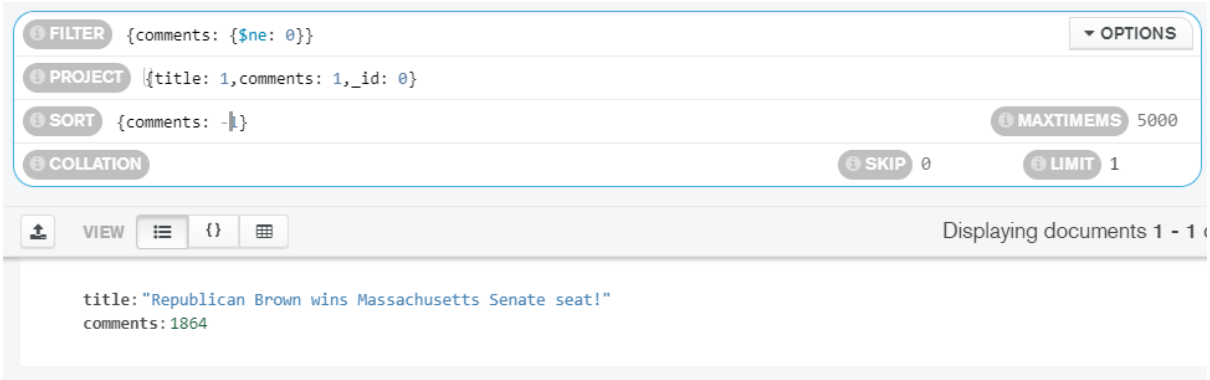
```
{
 filter: {
  $and: [
   {
    number_of_employees: {
     $ne: null
    }
   },
   {
    number_of_employees: {
     $ne: 0
    }
   }
  ]
 },
 project: {
  name: 1,  number_of_employees: 1,  _id: 0
```

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



12. Obtén la historia más comentada.

```
{
 filter: {
  comments: {
   $ne: 0
  }
 },
 project: {
  title: 1, comments: 1,  _id: 0
 },
 sort: {
  comments: -1
 },
 limit: 1
}
```



13. Obtén la historia menos comentada.

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

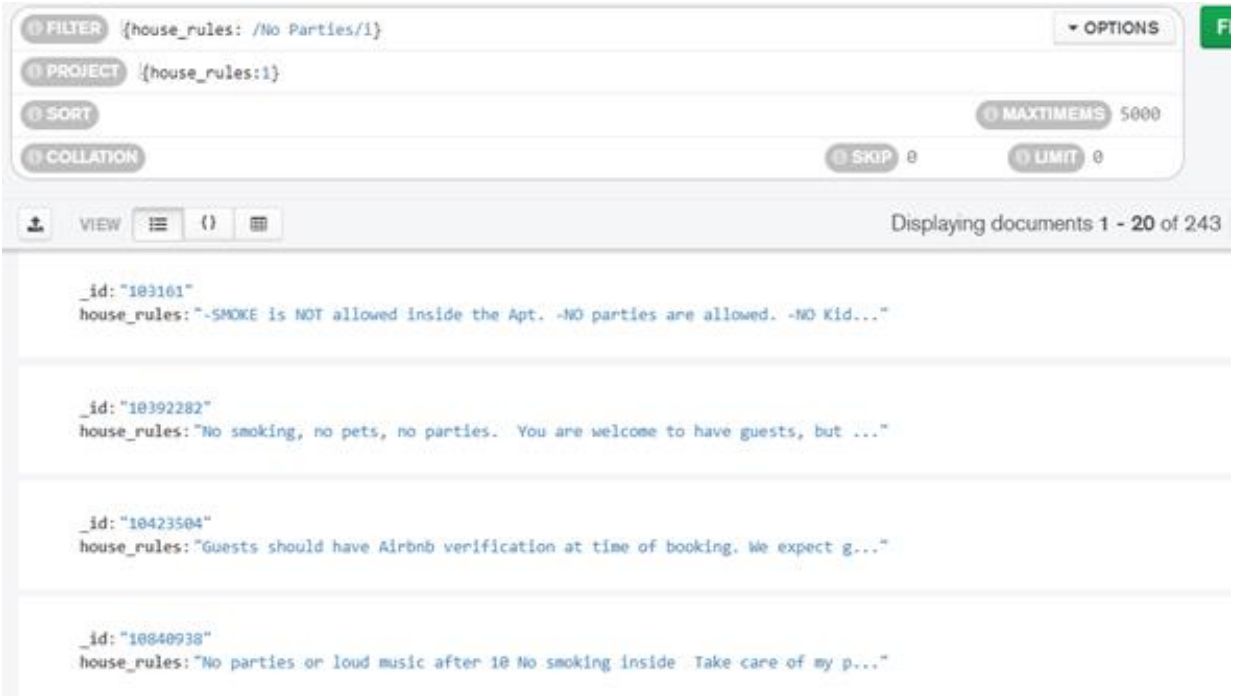
Sesión 5: Consultas en MongoDB

Work 5

Reto 1

Propiedades que no permitan fiestas.

```
{
 filter: {
  house_rules: RegExp('No Parties', i)
 },
 project: {
  house_rules: 1
 }
}
```



Propiedades que admitan mascotas.

```
{
 filter: {
  house_rules: RegExp('Pets Allowed', i)
```

```
 },
 project: {
  house_rules: 1
          }
}
```

Propiedades que no permitan fumadores.

```
{
 filter: {
  house_rules: RegExp('No Smoking', i)
 },
 project: {
  house_rules: 1
 }
}
```

Propiedades que no permitan fiestas ni fumadores.

```
{
 filter: {
  $and: [
   {
    house_rules: RegExp('No smoking', i)
   },
   {
    house_rules: RegExp('no parties', i)
   }
  ]
 },
 project: {
  house_rules: 1
 }
}
```

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'
 },
 project: {
```

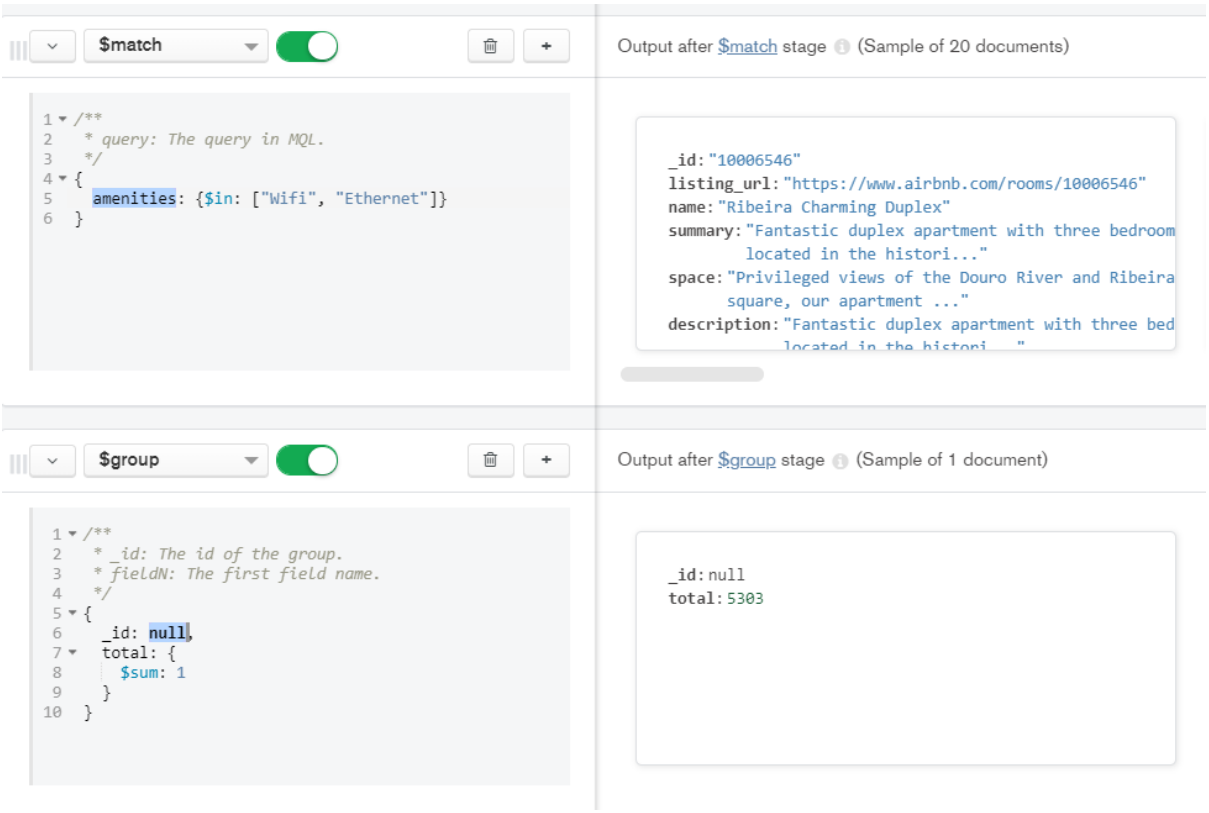
```
  number_of_reviews: 1, 'review_scores.review_scores_rating': 1,
  amenities: 1, 'address.country': 1
 }
}
```

Reto 3

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

```
[{$match: {
  amenities: {$in: ["Wifi", "Ethernet"]}
}}, {$group: {
  _id: null,
  total: {
    $sum: 1
  }
```

}}]


$match
/**
* query: The query in MQL.
*/
{
amenities: {$in: ["Wifi", "Ethernet"]}
}
Output after $match stage (Sample of 20 documents)
_id: "10006546"
listing_url: "https://www.airbnb.com/rooms/10006546"
name: "Ribeira Charming Duplex"
summary: "Fantastic duplex apartment with three bedroom
located in the histori..."
space: "Privileged views of the Douro River and Ribeira
square, our apartment ..."
description: "Fantastic duplex apartment with three bed
$group
/**
* _id: The id of the group.
* fieldN: The first field name.
*/
{
_id: null,
total: {
$sum: 1
}
}
Output after $group stage (Sample of 1 document)
_id: null
total: 5303

Proyecto 5

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},
  "review_scores.review_scores_rating": {$gte: 80},
  amenities: {$in:[/Ethernet/i]},
  "address.country": "Brazil"
}}, {$project: {
  number_of_reviews:1,
  "review_scores.review_scores_rating":1,
  amenities:1,
  "address.country":1
}}, {$group: {
  _id: null,
  Total: {
    $sum: 1
  }
}}]
```

$match

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

Output after $match stage (Sample of 6 documents)

```
_id: "1063491"
listing_url: "https://www.airbnb.com/rooms/1063491"
name: "Charming Apartment,perfect Location"
summary: "Charmoso quarto e sala, privativo, finamente
        decorado, com uma delicio..."
space: "It is a bedroom and living room, charming,
      private, finely decorated, ..."
description: "It is a bedroom and living room, charming
            private, finely decorated, ..."
```

$project

```
/**
 * specifications: The fields to
 *   include or exclude.
 */
{
  number_of_reviews:1,
  "review_scores.review_scores_rating":1,
  amenities:1,
  "address.country":1
}
```

Output after $project stage (Sample of 6 documents)

```
_id: "1063491"
number_of_reviews: 110
amenities: Array
address: Object
review_scores: Object
```

$group

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

Output after $group stage (Sample of 1 document)

```
_id: null
Total: 6
```

Sesión 6: Agregaciones

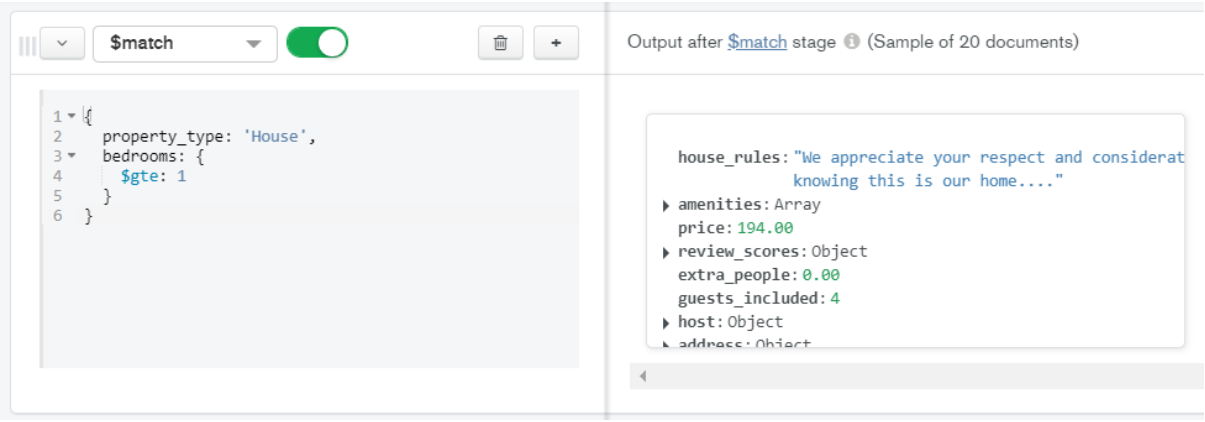
Work 6

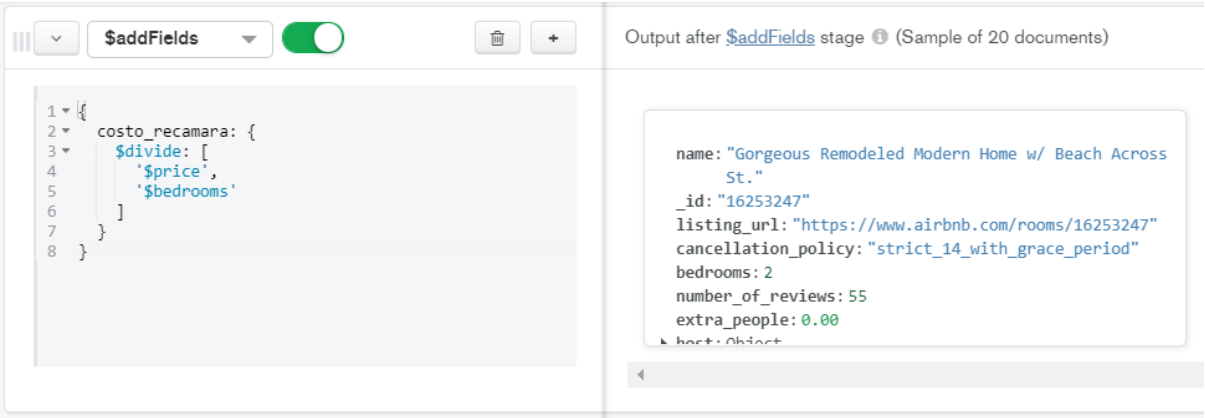
Reto 1

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'
    ]
```

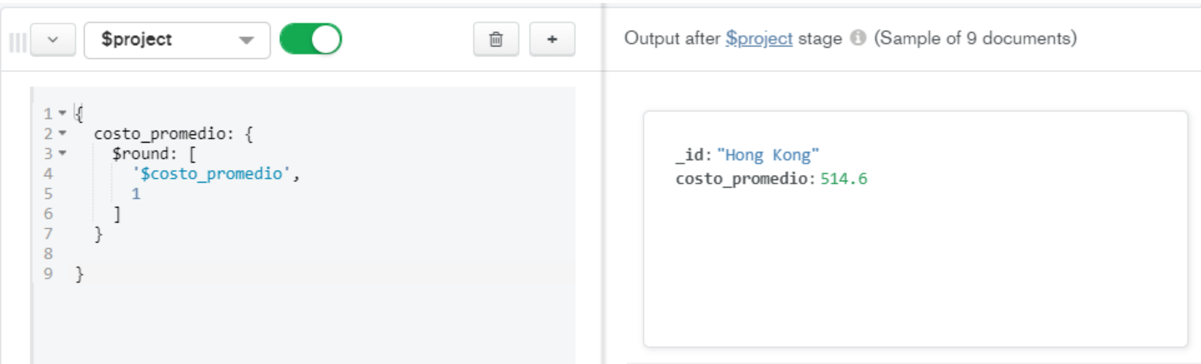
```
  }
}}, {$project: {
  costo_promedio: {
    $round: [
      '$costo_promedio',
      1
    ]
  }
}}, {$sort: {
  _id: 1
}}]
```

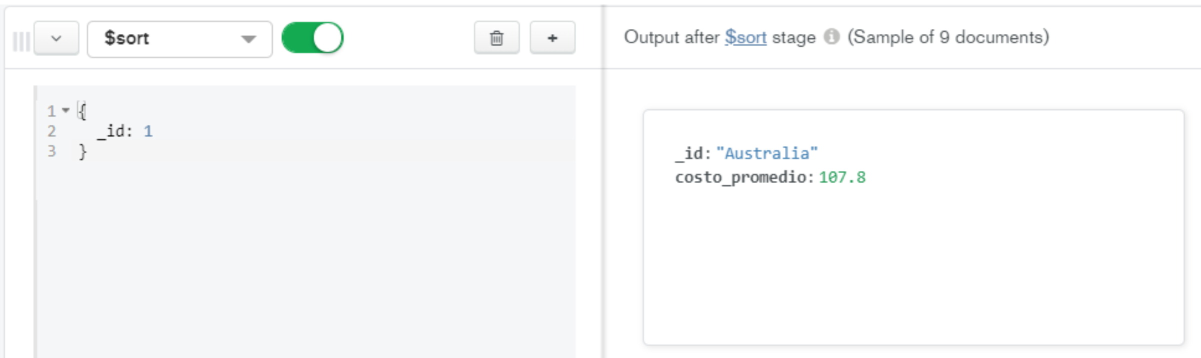





$group
Output after $group stage (Sample of 9 documents)
{
_id: '$address.country',
propiedades: {
$sum: 1
},
total: {
$sum: '$costo_recamara'
}
}
_id: "Australia"
propiedades: 166
total: 17891.32857142857142857142857142857


$addFields
Output after $addFields stage (Sample of 9 documents)
{
costo_promedio: {
$divide: [
'$total',
'$propiedades'
]
}
}
total: 17891.32857142857142857142857142857
costo_promedio: 107.7790877796901893287435456110155
_id: "Australia"
propiedades: 166


$project
Output after $project stage (Sample of 9 documents)
{
costo_promedio: {
$round: [
'$costo_promedio',
1
]
}
}
_id: "Hong Kong"
costo_promedio: 514.6


$sort
Output after $sort stage (Sample of 9 documents)
{
_id: 1
}
_id: "Australia"
costo_promedio: 107.8

Reto 2

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

```
[{$lookup: {
  from: 'users',
  localField: 'email',
  foreignField: 'email',
  as: 'user'
}}, {$addFields: {
  usr_obj: {
    $arrayElemAt: [
      '$user',
      0
    ]
  }
}}, {$match: {
  usr_obj: {
    $exists: true
  }
}}, {$addFields: {
  password: '$usr_obj.password'
}}, {$project: {
  password: 1,
  email: 1,
  name: 1,
  _id: 0
}}, {$group: {
  _id: '$email',
  comments: {
    $sum: 1
  }
}}, {$sort: {
  comments: -1
}}, {$count: '1'}]
```

$lookup

```
{
  from: 'users',
  localField: 'email',
  foreignField: 'email',
  as: 'user'
}
```

Output after $lookup stage (Sample of 20 documents)

```
▸ user: Array
  _id: ObjectId("5a9427648b0beebeb69579cc")
  name: "Andrea Le"
  email: "andrea_le@fakegmail.com"
  movie_id: ObjectId("573a1390f29313caabcd418c")
  text: "Rem officiis eaque repellendus amet eos
       doloribus. Porro dolor volupta..."
  date: 2012-03-26T23:20:16.000+00:00
```

$addFields

```
{
  usr_obj: {
    $arrayElemAt: [
      '$user',
      0
    ]
  }
}
```

Output after $addFields stage (Sample of 20 documents)

```
▸ usr_obj: Object
  _id: ObjectId("5a9427648b0beebeb69579cc")
  name: "Andrea Le"
  email: "andrea_le@fakegmail.com"
  movie_id: ObjectId("573a1390f29313caabcd418c")
  text: "Rem officiis eaque repellendus amet eos
       doloribus. Porro dolor volupta..."
  date: 2012-03-26T23:20:16.000+00:00
▸ user: Array
```

$match

```
{
  usr_obj: {
    $exists: true
  }
}
```

Output after $match stage (Sample of 20 documents)

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

$addFields

```
{
  password: '$usr_obj.password'
}
```

Output after $addFields stage (Sample of 20 documents)

```
  password: "$2b$12$JS87HWuL2y0P1E6kYrcbKOKx22.wsKEdLtS0F
  name: "Andrea Le"
  text: "Rem officiis eaque repellendus amet eos
       doloribus. Porro dolor volupta..."
  date: 2012-03-26T23:20:16.000+00:00
▸ user: Array
▸ usr_obj: Object
  _id: ObjectId("5a9427648b0beebeb69579cc")
```

$project

```
{
  password: 1,
  email: 1,
  name: 1,
  _id: 0
}
```

Output after $project stage (Sample of 20 documents)

```
name: "Andrea Le"
email: "andrea_le@fakegmail.com"
password: "$2b$12$JS87HWuL2y0P1E6kYrcbKOKx22.wsKEdLtS0F
```

$group

```
{
  _id: '$email',
  comments: {
    $sum: 1
  }
}
```

Output after $group stage (Sample of 20 documents)

```
_id: "liam_cunningham@gameofthron.es"
comments: 280
```

$sort

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

Output after $sort stage (Sample of 20 documents)

```
_id: "roger_ashton-griffiths@gameofthron.es"
comments: 331
```

$count

```
'1'
```

Output after $count stage (Sample of 1 document)

```
1: 184
```

Proyecto 6

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

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

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

$unwind

```
{
  path: "$genres"
}
```

Output after $unwind stage (Sample of 20 documents)

```
imdb: Object
num_mflix_comments: 1
fullplot: "A stationary camera looks at a large anvil w
          blacksmith behind it..."
directors: Array
rated: "UNRATED"
type: "movie"
_id: ObjectId("573a1390f29313caabcd4135")
cast: Array
```

$unwind

```
{
  path: "$countries"
}
```

Output after $unwind stage (Sample of 20 documents)

```
genres: "Short"
year: 1893
tomatoes: Object
cast: Array
num_mflix_comments: 1
directors: Array
plot: "Three men hammer on an anvil and pass a bottle o
      beer around."
title: "Blacksmith Scene"
```

$group

```
{
  _id: {genero:"$genres", pais: "$countries"},
  peliculas:{$sum:1}
}
```

Output after $group stage (Sample of 20 documents)

```
_id: Object
peliculas: 166
```

$addFields

```
{
  pais:"$_id.pais",
  genero:"$_id.genero"
}
```

Output after $addFields stage (Sample of 20 documents)

```
genero: "Family"
_id: Object
peliculas: 14
pais: "Italy"
```


$project
{
pais:1,
genero:1,
peliculas:1,
"_id":0
}
Output after $project stage (Sample of 20 documents)
peliculas: 2
pais: "Hong Kong"
genero: "Western"


$sort
{
pais:1
}
Output after $sort stage (Sample of 20 documents)
pais: "Afghanistan"
genero: "History"
peliculas: 3

Sesión 7: Configuración de Bases de Datos Locales

Work 7

Reto1

Definir los campos y tipos de datos para la tabla `movies` haciendo uso de los archivos `movies.dat` y `README`.

Crear la tabla `movies` (recuerda usar el mismo nombre del archivo sin la extensión para vincular nombres de tablas con archivos).

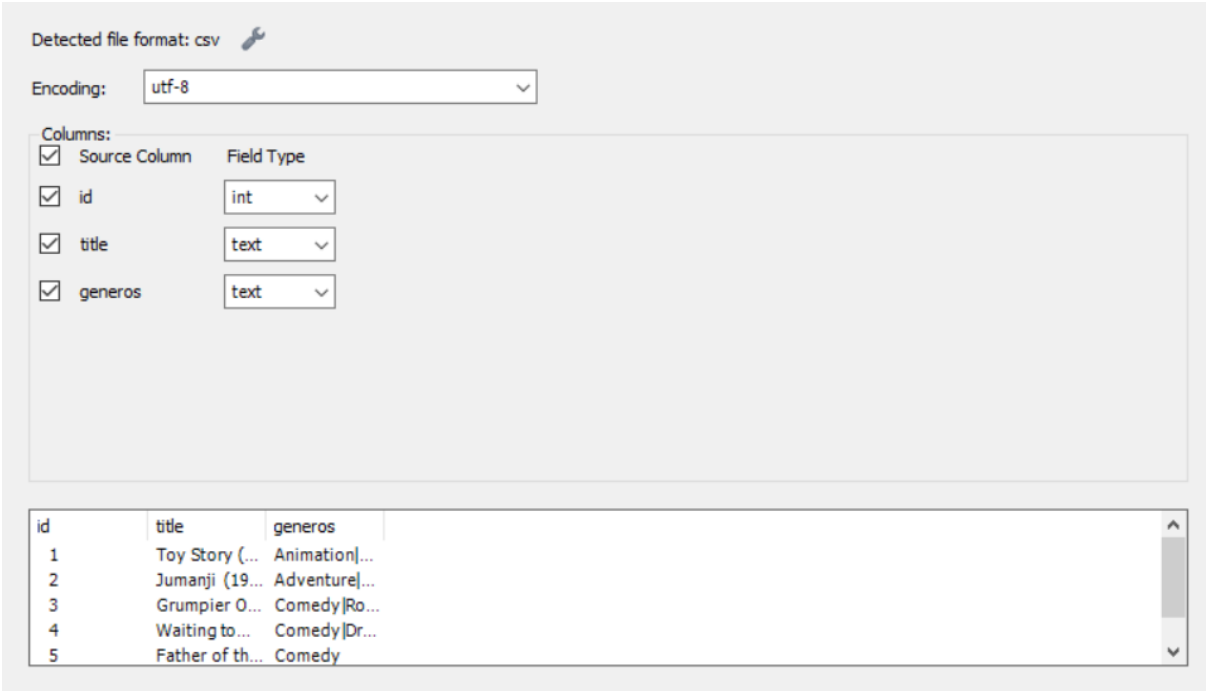
Definir los campos y tipos de datos para la tabla `ratings` haciendo uso de los archivos `ratings.dat` y `README`.

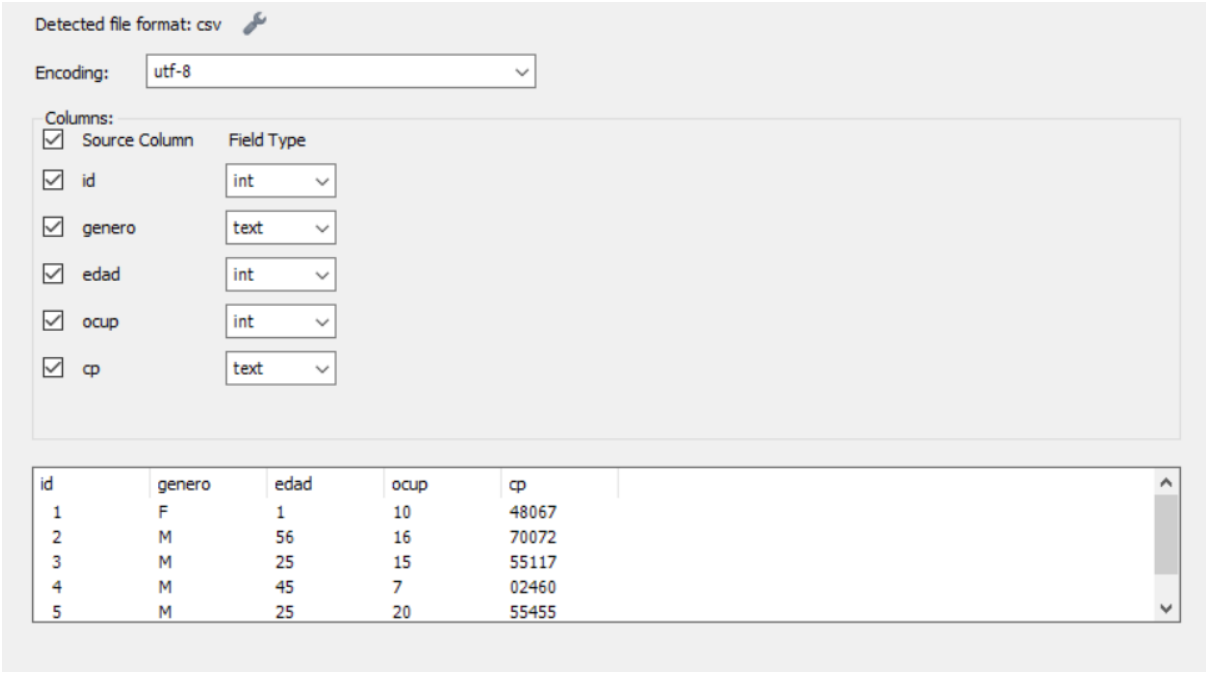
Crear la tabla ratings (recuerda usar el mismo nombre del archivo sin la extensión para vincular nombres de tablas con archivos)

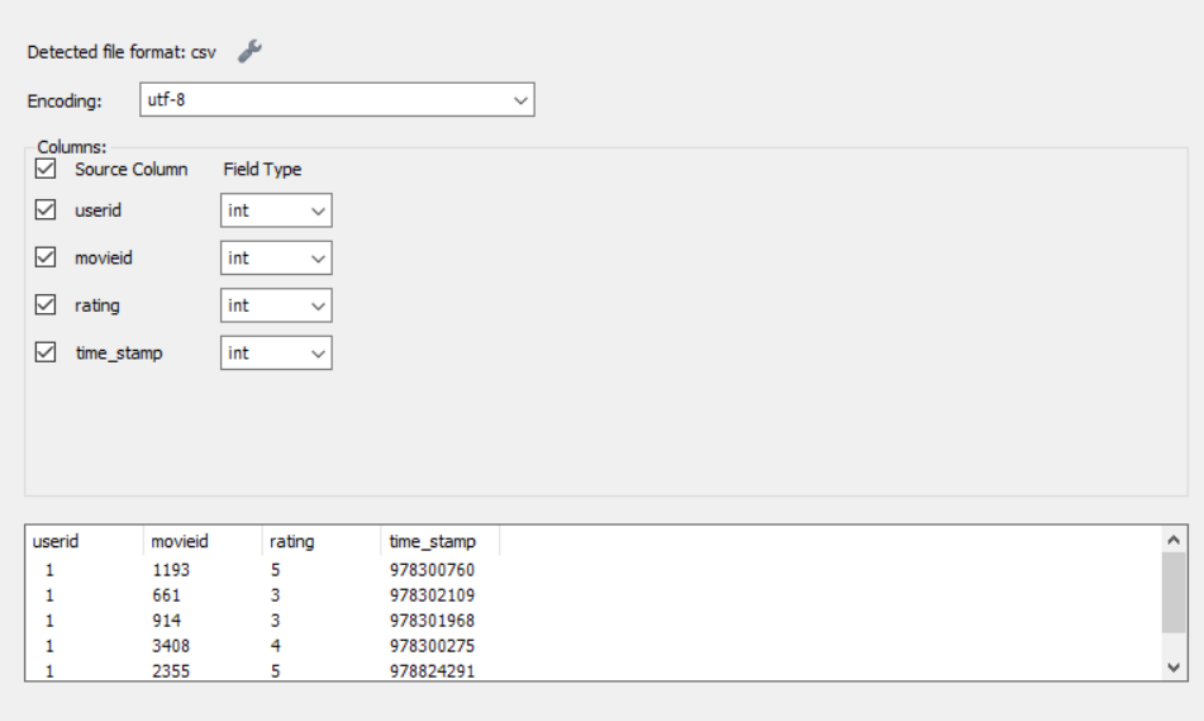
Reto 2

Usando como base el archivo `movies.dat`, limpiarlo e importar los datos en la tabla `movies` creada en el Reto 1.

Usando como base el archivo `ratings.dat`, limpiarlo e importar los datos en la tabla `ratings` creada en el Reto 2.


Detected file format: csv
Encoding:
utf-8
Columns:
Source Column
Field Type
id
int
title
text
generos
text
id
title
generos
1
Toy Story (...
Animation|...
2
Jumanji (19...
Adventure|...
3
Grumpier O...
Comedy|Ro...
4
Waiting to...
Comedy|Dr...
5
Father of th...
Comedy


Detected file format: csv
Encoding:
utf-8
Columns:
Source Column
Field Type
id
int
genero
text
edad
int
ocup
int
cp
text
id
genero
edad
ocup
cp
1
F
1
10
48067
2
M
56
16
70072
3
M
25
15
55117
4
M
45
7
02460
5
M
25
20
55455



Reto 3

Se deberá de crear colecciones e importar los datos de los archivos `movies.csv` y `ratings.csv`

Import To Collection WHR.movies

Select File

C:\Users\IN334839\Desktop\Data Analysis\ml-1m\movies.csv BROWSE

Select Input File Type

JSON | CSV

Options

Select delimiter COMMA

- [x] Ignore empty strings
- [ ] Stop on errors

Specify Fields and Types

	id (Number)	title (String)	generos (String)
1	1	Toy Story (1995)	Animation|Children's|Comedy
2	2	Jumanji (1995)	Adventure|Children's|Fantasy
3	3	Grumpier Old Men (1995)	Comedy|Romance
4	4	Waiting to Exhale (1995)	Comedy|Drama
5	5	Father of the Bride Part II (1995)	Comedy
6	6	Heat (1995)	Action|Crime|Thriller
7	7	Sabrina (1995)	Comedy|Romance
8	8	Tom and Huck (1995)	Adventure|Children's
9	9	Sudden Death (1995)	Action
10	10	GoldenEye (1995)	Action|Adventure|Thriller

Import To Collection WHR.ratings

Select File

C:\Users\IN334839\Desktop\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

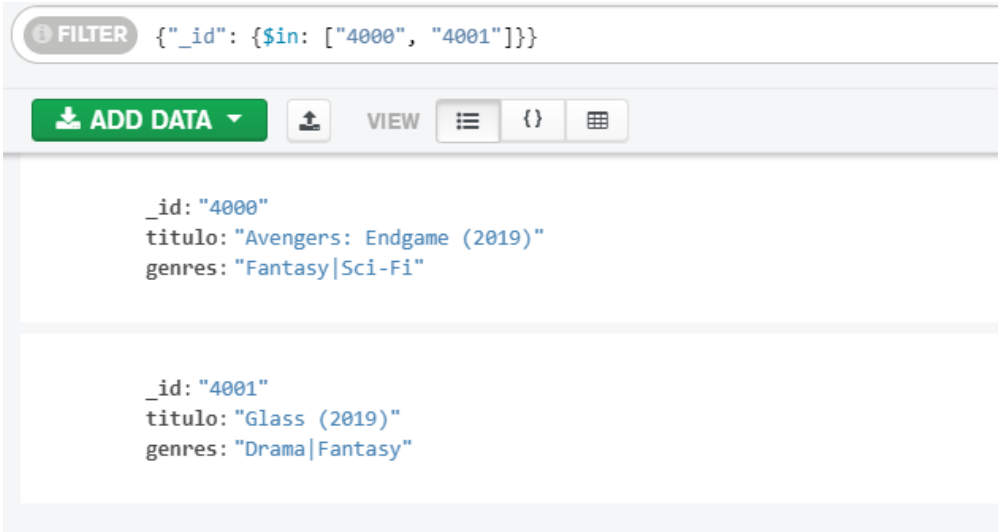
	userid String	movieid String	rating String	time_stamp String
1	1	1193	5	978300760
2	1	661	3	978302109
3	1	914	3	978301968
4	1	3408	4	978300275
5	1	2355	5	978824291
6	1	1197	3	978302268
7	1	1287	5	978302039
8	1	2804	5	978300719
9	1	594	4	978302268
10	1	919	4	978301368

Proyecto 7

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

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


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



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

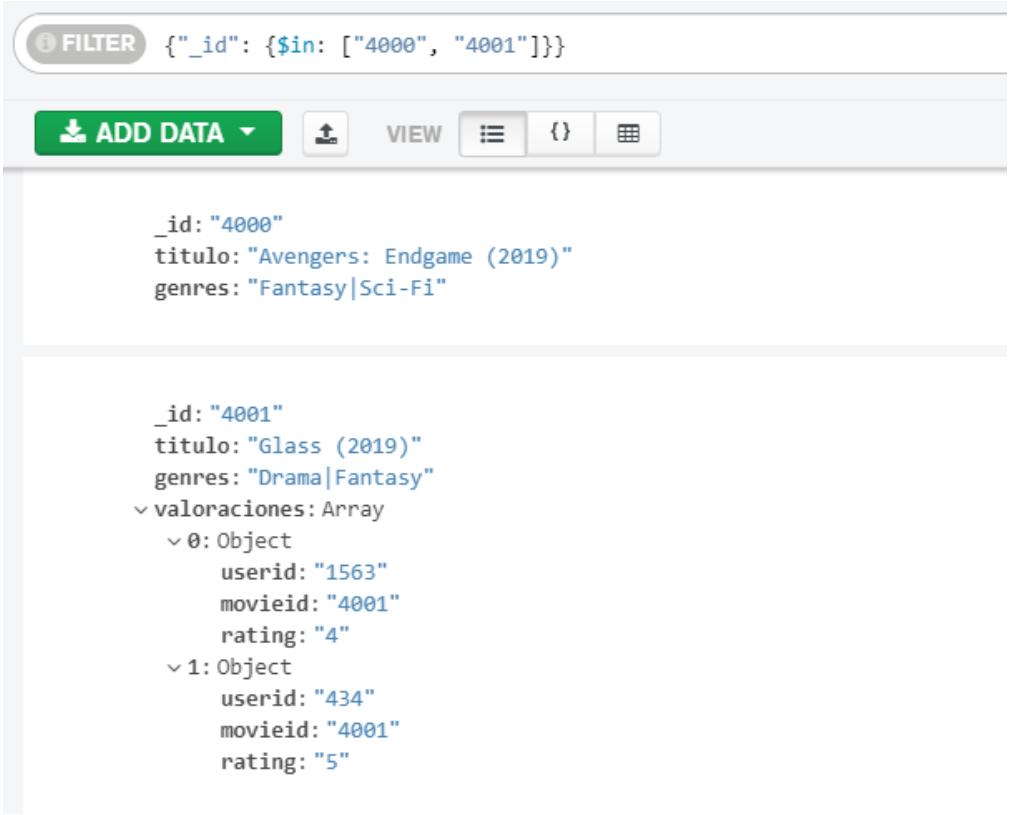
```
{
  id:"4001",
  titulo:"Glass (2019)",
```

```
  genres:"Drama|Fantasy",
  valoraciones: [
    {
      userid: "1563",
      movieid: "4001",
      rating: "4"
    },
    {
      userid: "434",
      movieid: "4001",
      rating: "5"
    }
  ]
}
```



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

```
    }
  ]
}
```


FILTER
{"_id": {$in: ["4000", "4001"]}}
ADD DATA
VIEW
{}
_id: "4000"
titulo: "Avengers: Endgame (2019)"
genres: "Fantasy|Sci-Fi"
_id: "4001"
titulo: "Glass (2019)"
genres: "Drama|Fantasy"
valoraciones: Array
0: Object
userid: "1563"
movieid: "4001"
rating: "4"
1: Object
userid: "434"
movieid: "4001"
rating: "5"

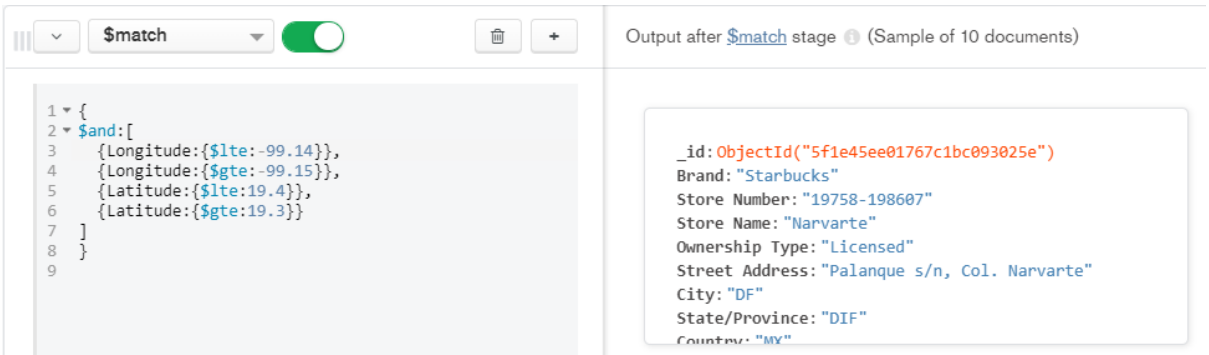
Sesión 8: Query competition

Work 8

Reto 1

- **Descarga la fuente de datos de los locales de Starbucks: `directory.csv`**
- **Analiza los datos, limpia los datos en caso de ser necesario.**
- **Elige MySQL o MongoDB y crea una base de datos para el conjunto de datos del reto.**
- **Carga los datos en la base de datos que elegiste y revisa que éstos se muestren correctamente.**
- **Usando la latitud y longitud de tu posición actual, 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:-99.14}},
  {Longitude:{$gte:-99.15}},
  {Latitude:{$lte:19.4}},
  {Latitude:{$gte:19.3}}
]
}
}, {}]
```

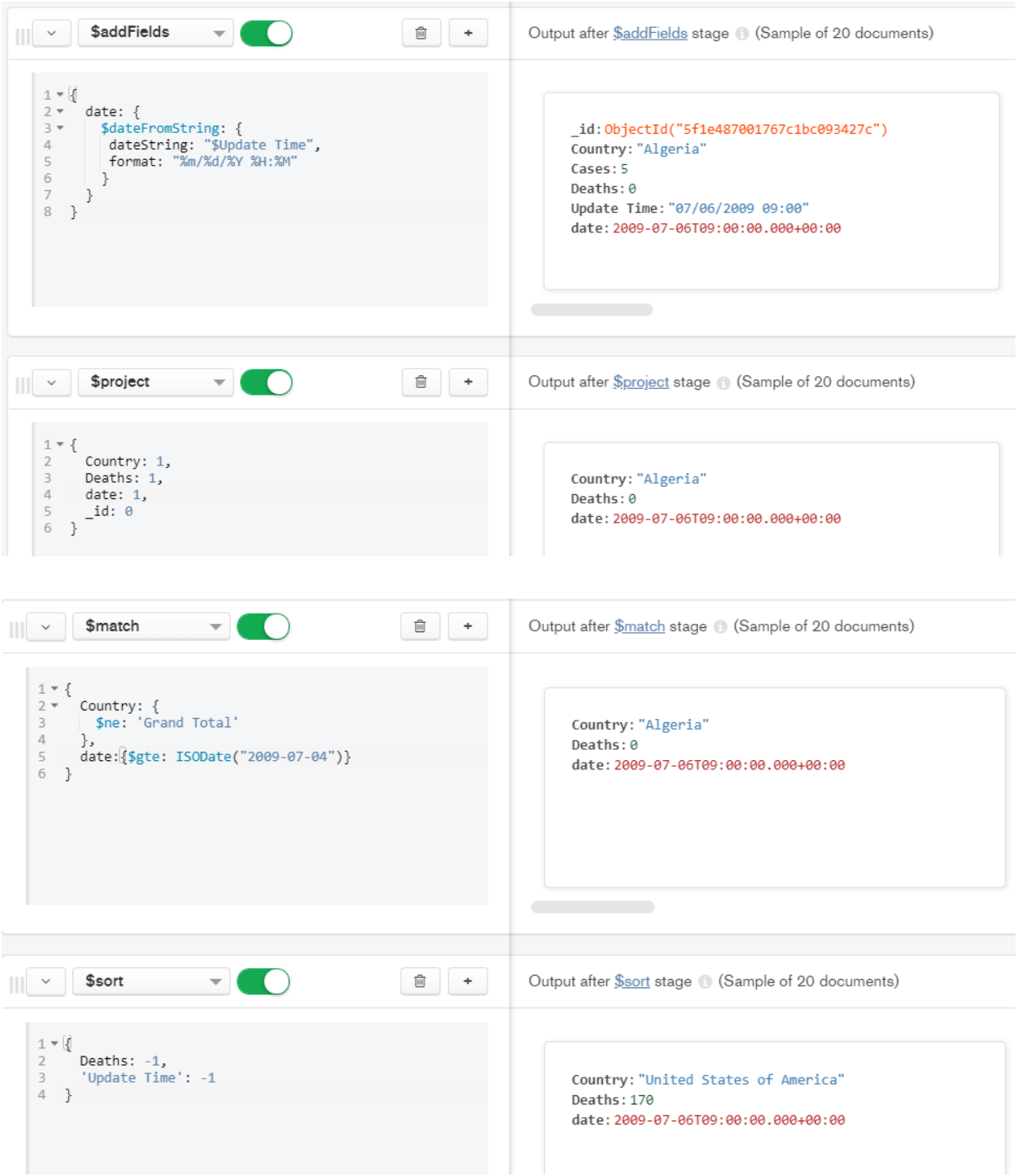


Reto 2

- **Descarga la fuente de datos sobre la pandemia del 2009 (H1N1):** `Pandemic (H1N1) 2009.csv`
- **Analiza los datos, limpia los datos en caso de ser necesario.**
- **Elige MySQL o MongoDB y crea una base de datos para el conjunto de datos del reto.**
- **Carga los datos en la base de datos que elegiste y revisa que éstos se muestren correctamente.**
- **Responde a las siguientes preguntas usando consultas:**

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

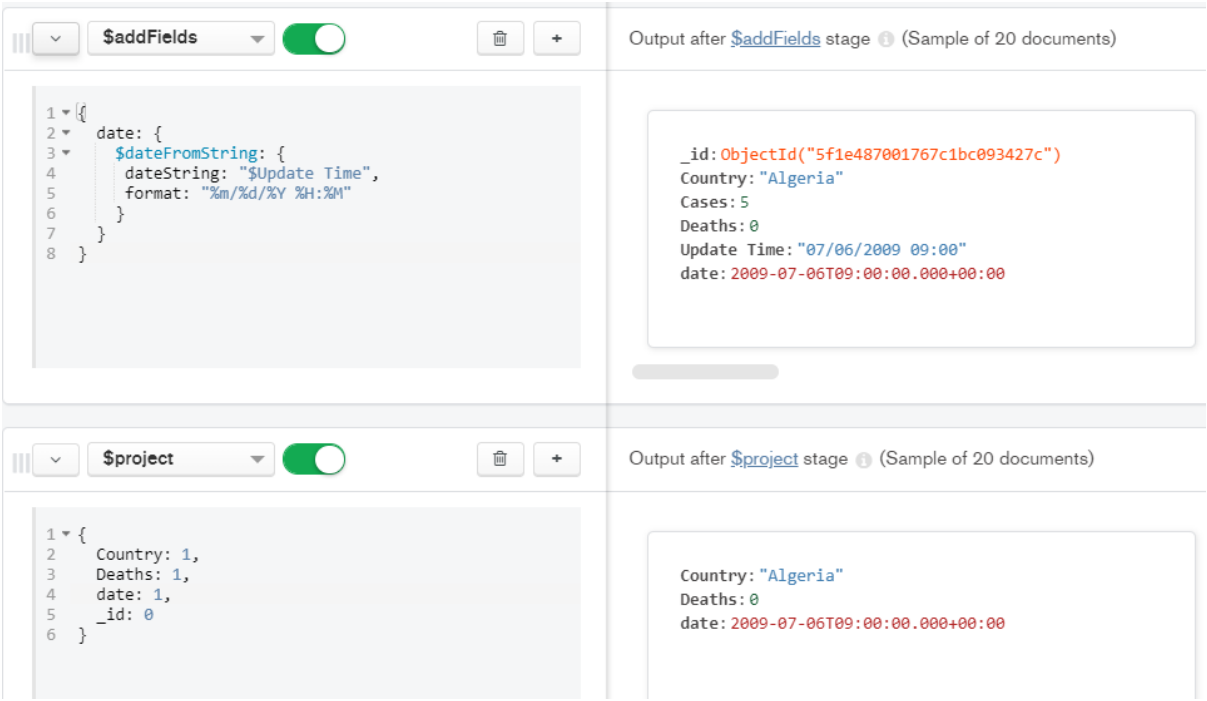
```
[{$addFields: {
  date: {
    $dateFromString: {
     dateString: "$Update Time",
     format: "%m/%d/%Y %H:%M"
    }
  }
}}, {$project: {
  Country: 1,
  Deaths: 1,
  date: 1,
  _id: 0
}}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  date:{$gte: ISODate("2009-07-04")}
}}, {$sort: {
  Deaths: -1,
  'Update Time': -1
}}, {$limit: 1}]
```

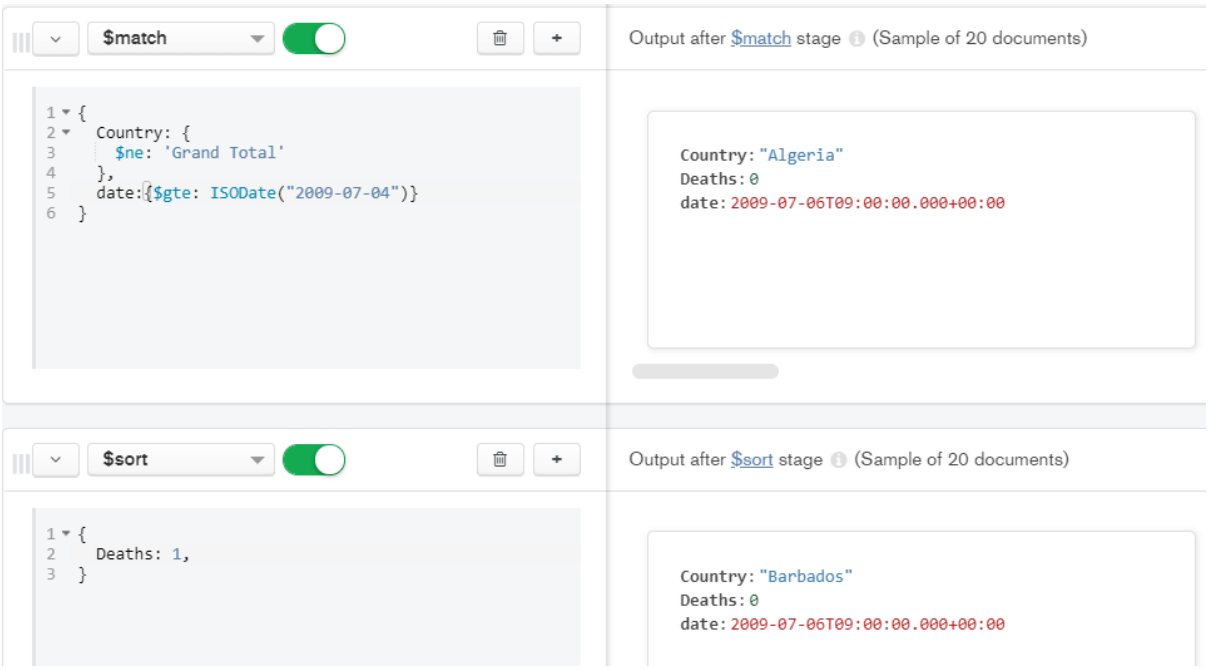


2. ¿Cuál fue el país con menor número de muertes?

```
[{$addFields: {
  date: {
    $dateFromString: {
     dateString: "$Update Time",
     format: "%m/%d/%Y %H:%M"
    }
```

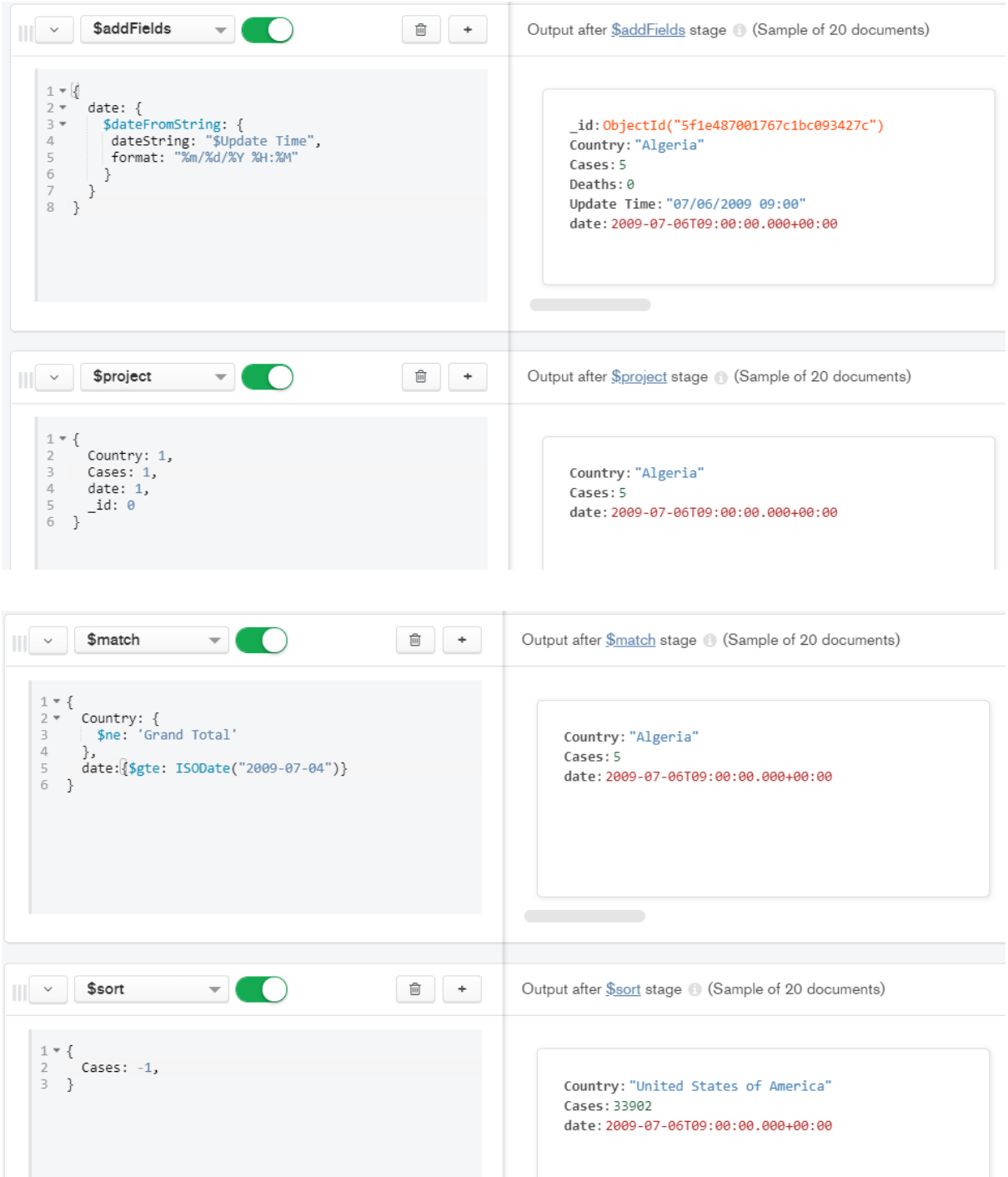
```
  }
}}, {$project: {
  Country: 1,
  Deaths: 1,
  date: 1,
  _id: 0
}}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  date:{$gte: ISODate("2009-07-04")}
}}, {$sort: {
  Deaths: 1,
}}, {$limit: 1}]
```





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

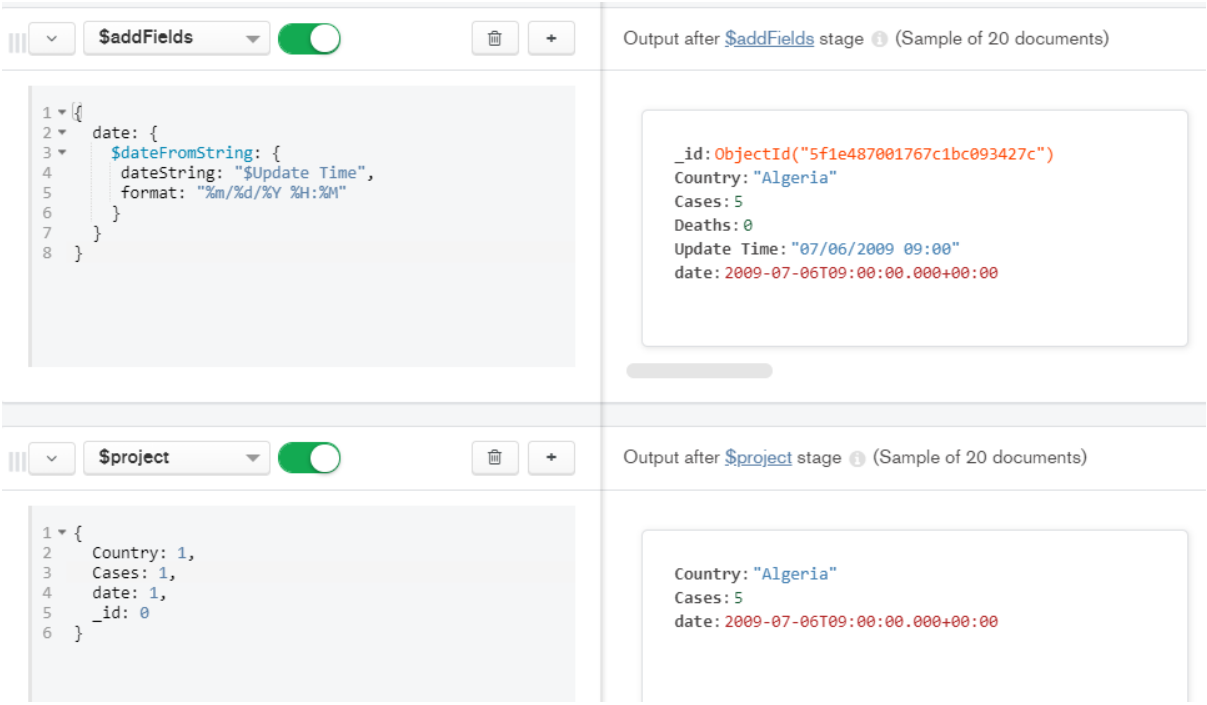
```
[{$addFields: {
  date: {
    $dateFromString: {
     dateString: "$Update Time",
     format: "%m/%d/%Y %H:%M"
    }
  }
}}, {$project: {
  Country: 1,
  Cases: 1,
  date: 1,
  _id: 0
}}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  date:{$gte: ISODate("2009-07-04")}
}}, {$sort: {
  Cases: -1,
}}, {$limit: 1}]
```

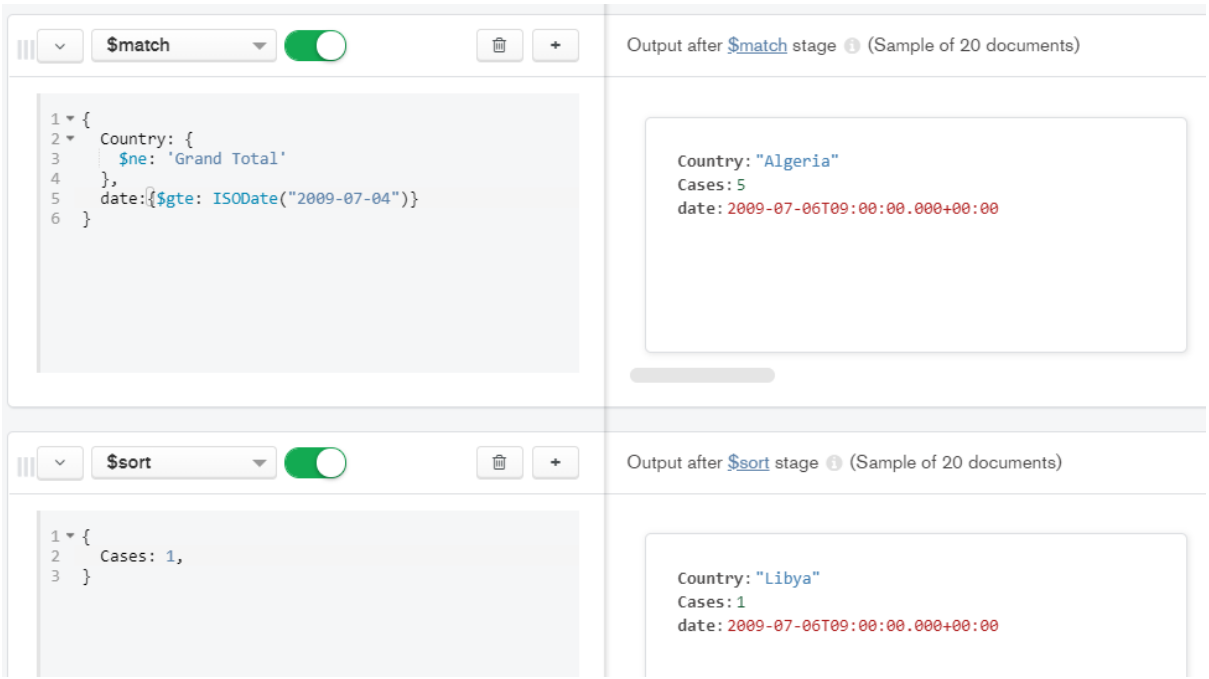


4. **¿Cuál fue el país con el menor número de casos?**

```
[{$addFields: {
  date: {
    $dateFromString: {
     dateString: "$Update Time",
     format: "%m/%d/%Y %H:%M"
    }
```

```
  }
}}, {$project: {
  Country: 1,
  Cases: 1,
  date: 1,
  _id: 0
}}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  date:{$gte: ISODate("2009-07-04")}
}}, {$sort: {
  Cases: 1,
}}, {$limit: 1}]
```

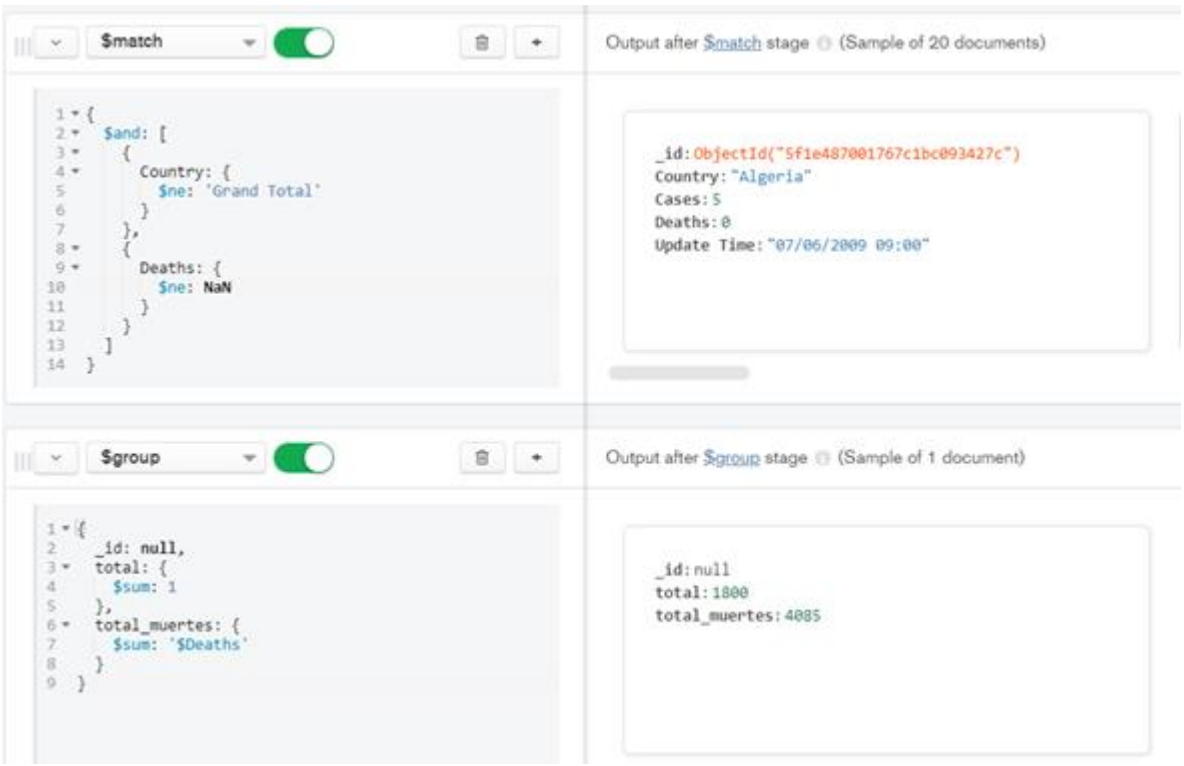


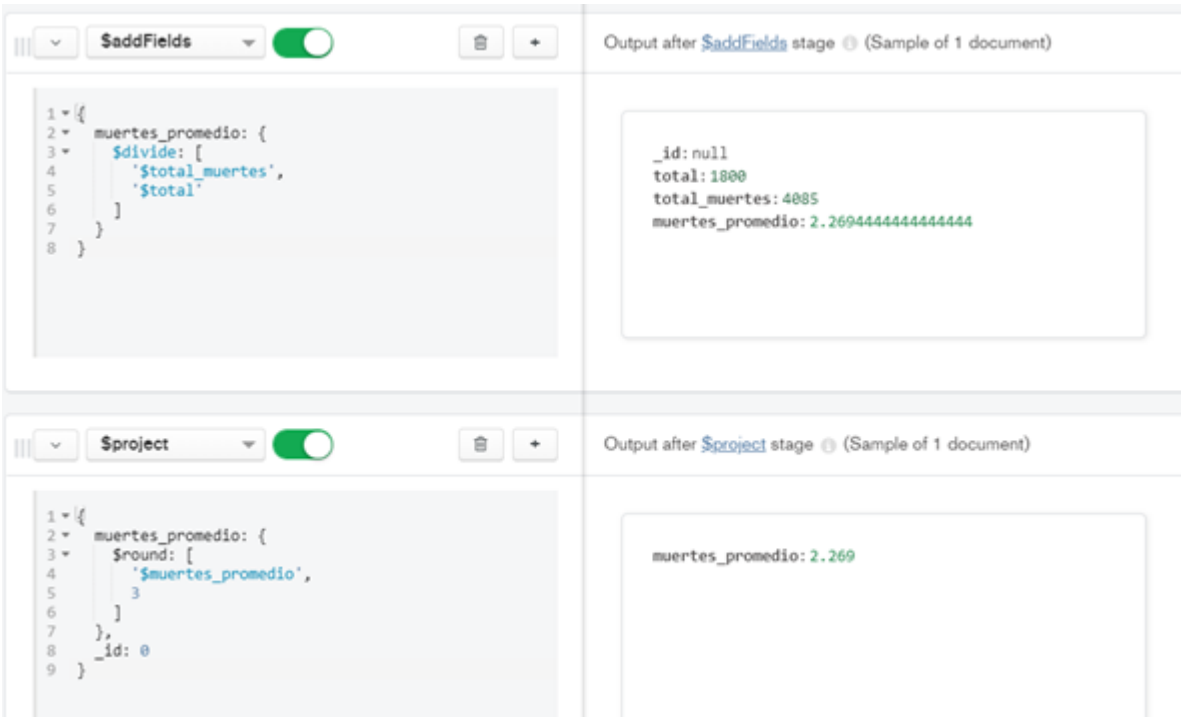


5. **¿Cuál fue el número de muertes promedio?**

```
[{$match: {
  $and: [
    {
      Country: {
        $ne: 'Grand Total'
      }
    },
    {
      Deaths: {
        $ne: NaN
      }
    }
  ]
}}, {$group: {
  _id: null,
  total: {
    $sum: 1
  },
  total_muertes: {
    $sum: '$Deaths'
```

```
  }
}}, {$addFields: {
  muertes_promedio: {
    $divide: [
      '$total_muertes',
      '$total'
    ]
  }
}}, {$project: {
  muertes_promedio: {
    $round: [
      '$muertes_promedio',
      3
    ]
  },
  _id: 0
}}]
```

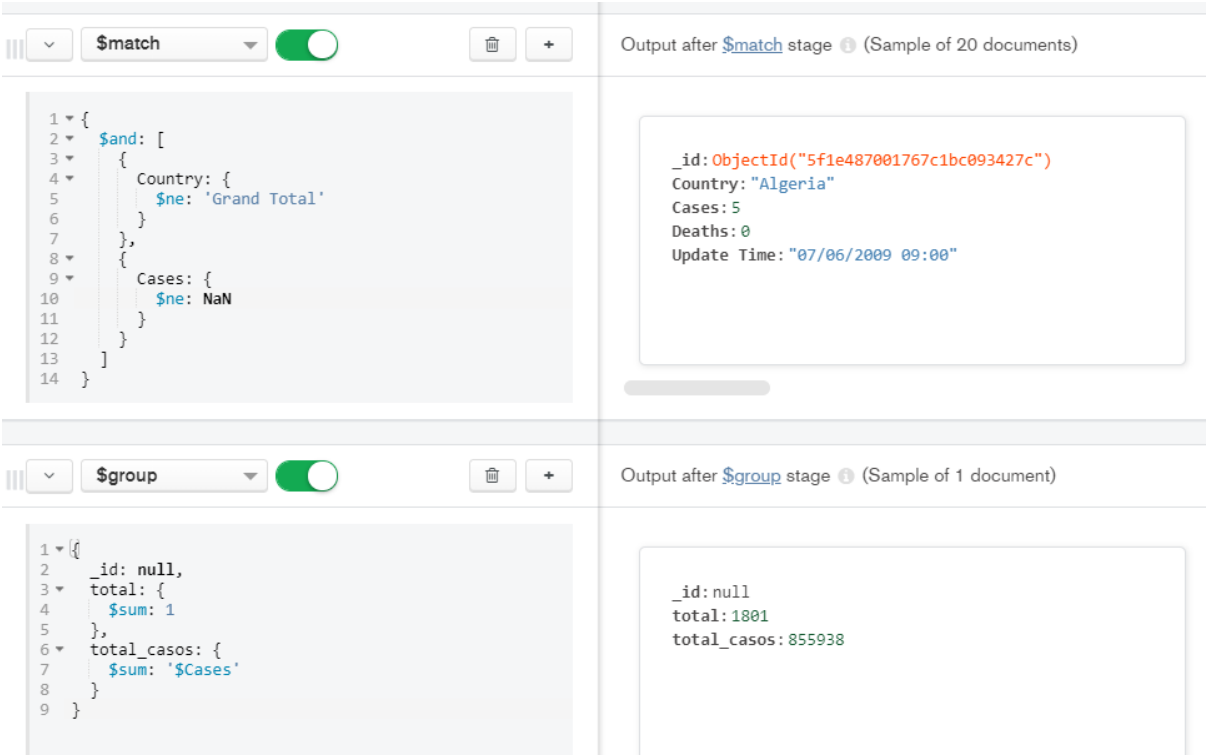


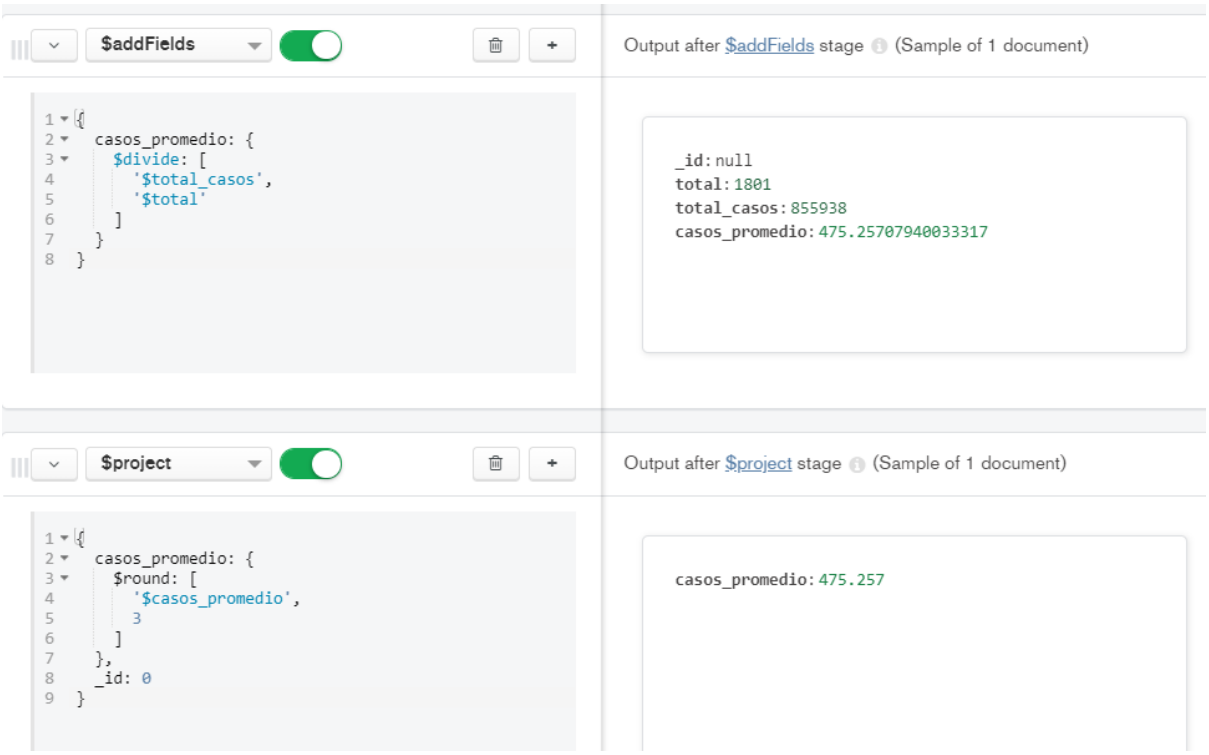


6. **¿Cuál fue el número de casos promedio?**

```
[{$match: {
  $and: [
    {
      Country: {
        $ne: 'Grand Total'
      }
    },
    {
      Cases: {
        $ne: NaN
      }
    }
  ]
}}, {$group: {
  _id: null,
  total: {
    $sum: 1
  },
  total_casos: {
```

```
      $sum: '$Cases'
    }
  }}, {$addFields: {
    casos_promedio: {
      $divide: [
        '$total_casos',
        '$total'
      ]
    }
  }}, {$project: {
    casos_promedio: {
      $round: [
        '$casos_promedio',
        3
      ]
    },
    _id: 0
  }}]
```

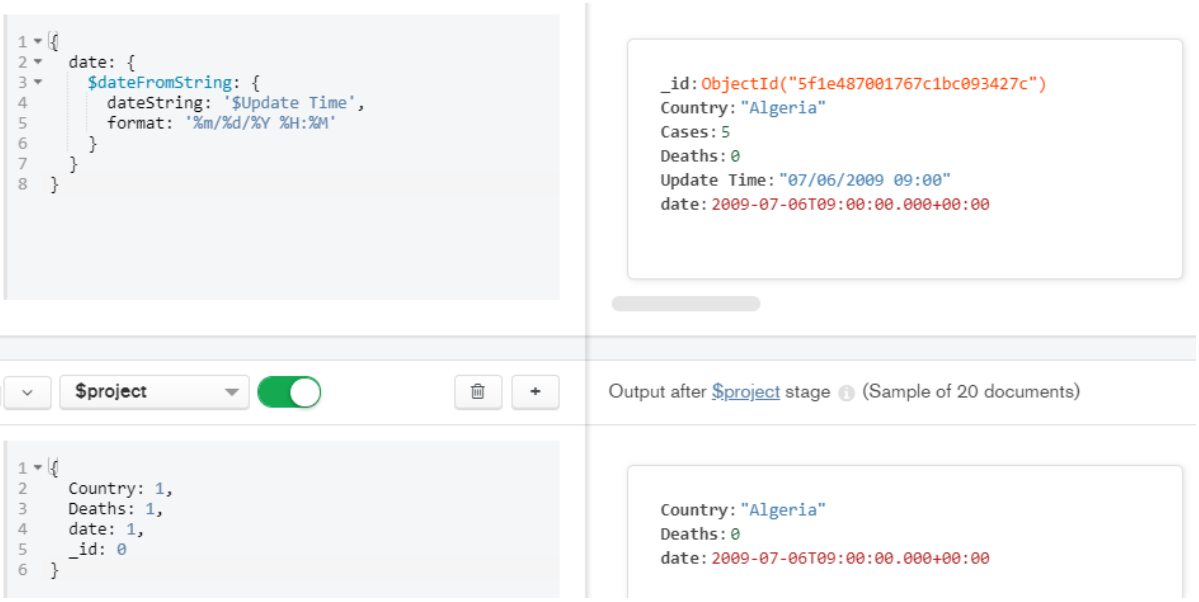


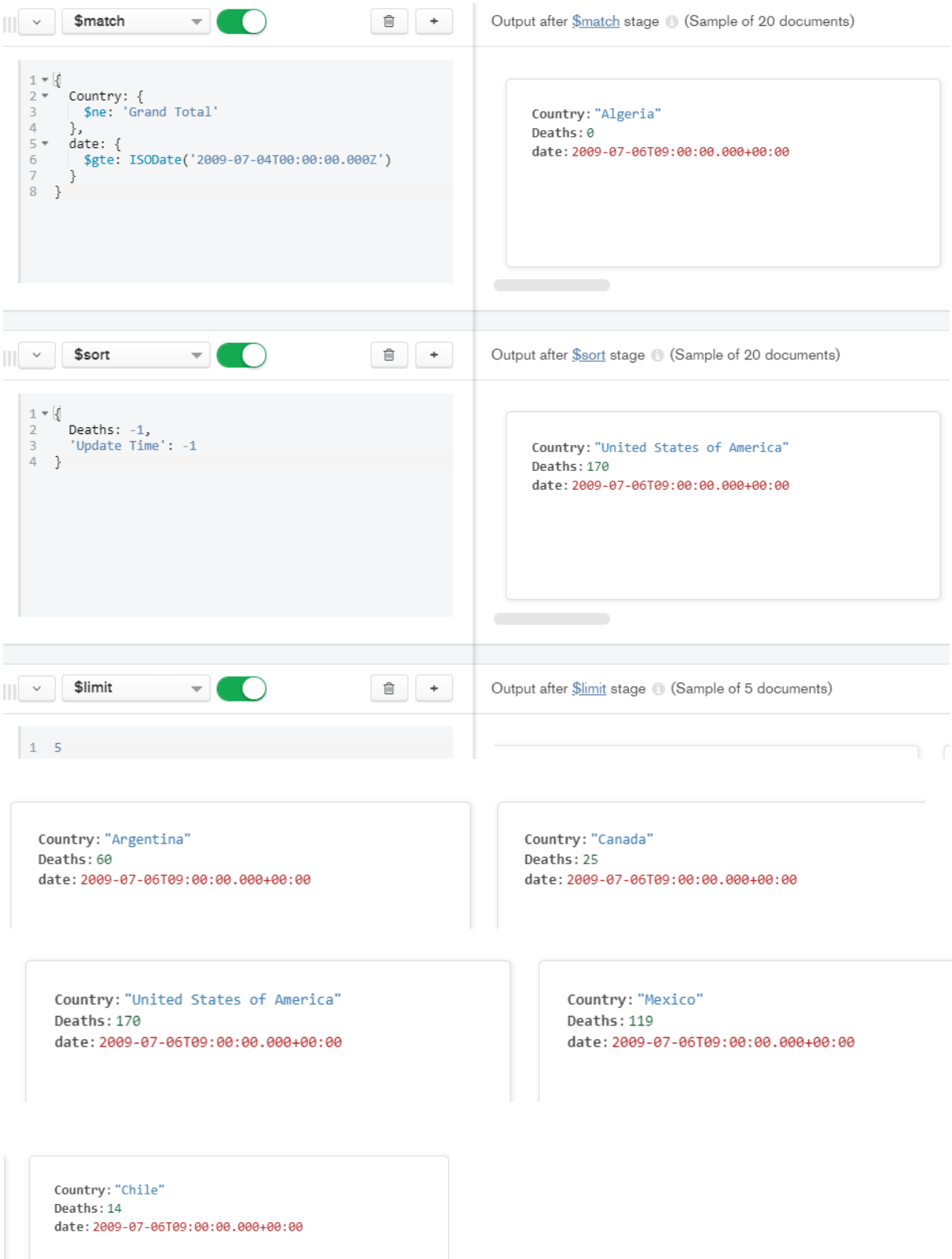


7. **Top 5 de países con más muertes**

```
[{$addFields: {
  date: {
    $dateFromString: {
      dateString: '$Update Time',
      format: '%m/%d/%Y %H:%M'
    }
  }
}}, {$project: {
  Country: 1,
  Deaths: 1,
  date: 1,
  _id: 0
}}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  date: {
    $gte: ISODate('2009-07-04T00:00:00.000Z')
```

```
  }
}}, {$sort: {
  Deaths: -1,
  'Update Time': -1
}}, {$limit: 5}]
```





8. Top 5 de países con menos muertes

[{$addFields: {

```
  date: {
    $dateFromString: {
      dateString: '$Update Time',
      format: '%m/%d/%Y %H:%M'
    }
  }
}}, {$project: {
  Country: 1,
  Deaths: 1,
  date: 1,
  _id: 0
}}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  date: {
    $gte: ISODate('2009-07-04T00:00:00.000Z')
  }
}}, {$sort: {
  Deaths: 1,
}}, {$limit: 5}, {}]
```

$addFields

```
{
  date: {
    $dateFromString: {
      dateString: '$Update Time',
      format: '%m/%d/%Y %H:%M'
    }
  }
}
```

Output after $addFields stage (Sample of 20 documents)

```
_id: ObjectId("5f1e487001767c1bc093427c")
Country: "Algeria"
Cases: 5
Deaths: 0
Update Time: "07/06/2009 09:00"
date: 2009-07-06T09:00:00.000+00:00
```

$project

```
{
  Country: 1,
  Deaths: 1,
  date: 1,
  _id: 0
}
```

Output after $project stage (Sample of 20 documents)

```
Country: "Algeria"
Deaths: 0
date: 2009-07-06T09:00:00.000+00:00
```

$match

```
{
  Country: {
    $ne: 'Grand Total'
  },
  date: {
    $gte: ISODate('2009-07-04T00:00:00.000Z')
  }
}
```

Output after $match stage (Sample of 20 documents)

```
Country: "Algeria"
Deaths: 0
date: 2009-07-06T09:00:00.000+00:00
```

$sort

```
{
  Deaths: 1,
}
```

Output after $sort stage (Sample of 20 documents)

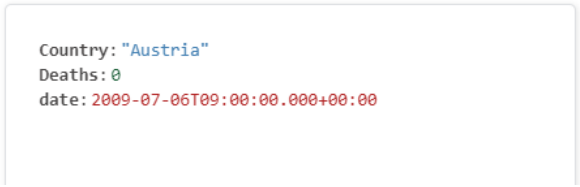
```
Country: "Barbados"
Deaths: 0
date: 2009-07-06T09:00:00.000+00:00
```

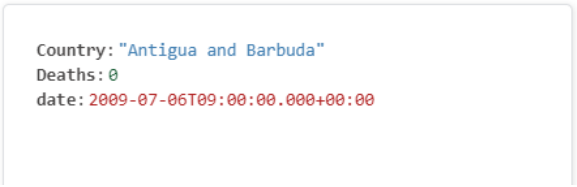
$limit

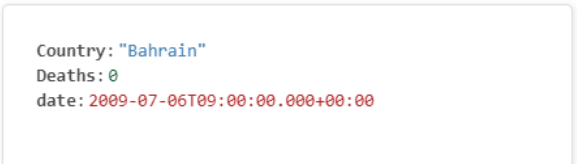
```
5
```

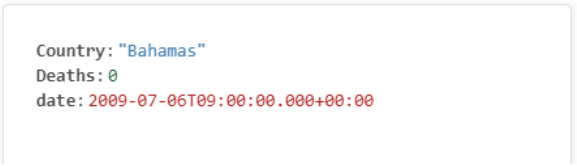
Output after $limit stage (Sample of 5 documents)

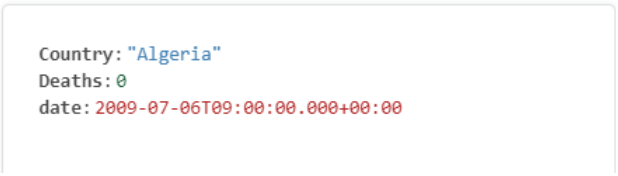
```
Country: "Austria"
Deaths: 0
date: 2009-07-06T09:00:00.000+00:00
```









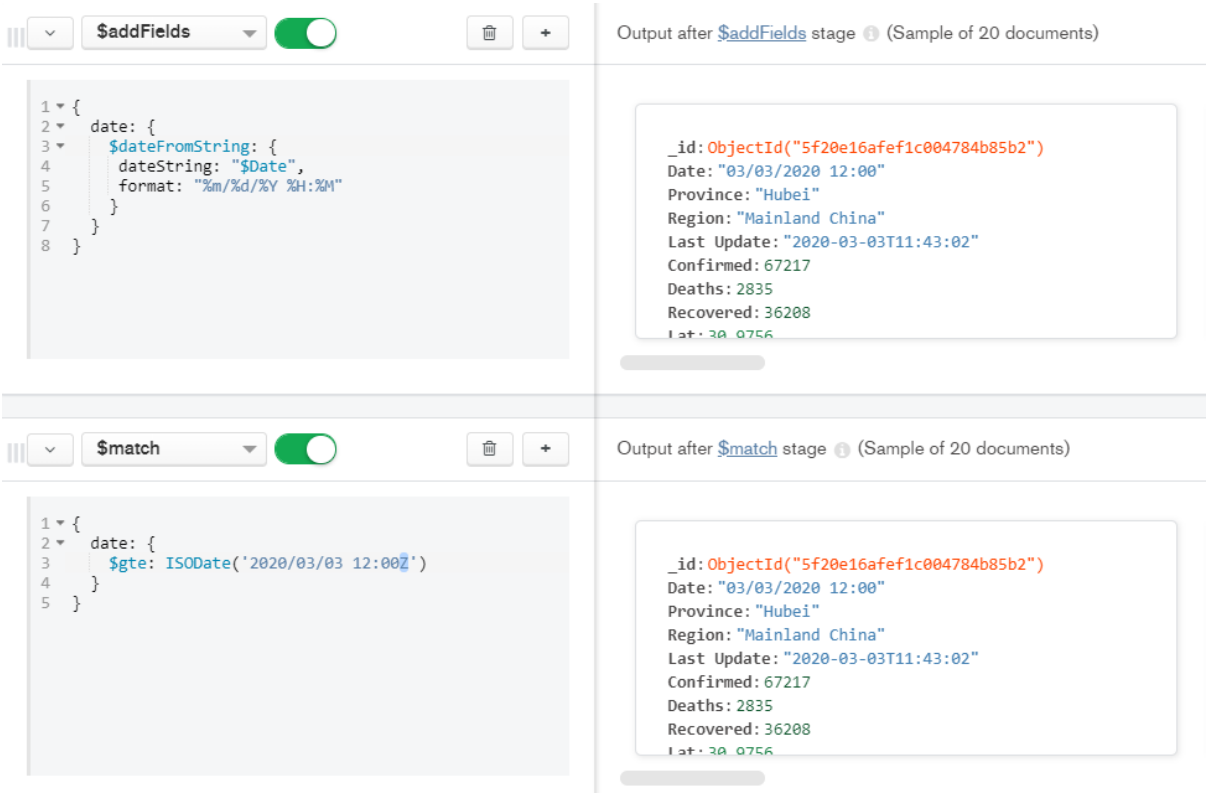


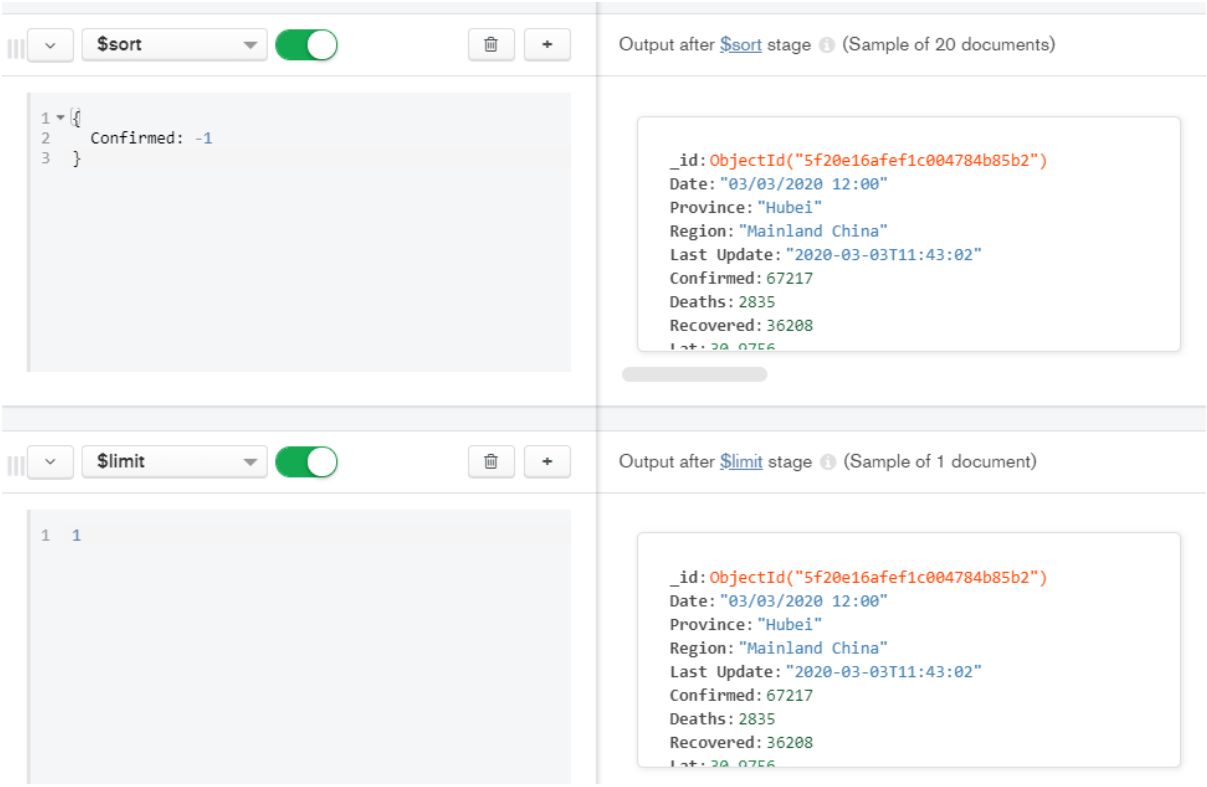
- **Descarga la fuente de datos de los casos sobre la pandemia del COVID-19:** `2019-nCoV-cases-JHU.csv`
- **Analiza los datos, limpia los datos en caso de ser necesario.**
- **Elige MySQL o MongoDB y crea una base de datos para el conjunto de datos del reto.**
- **Carga los datos en la base de datos que elegiste y revisa que éstos se muestren correctamente.**
- **Responde a las siguientes preguntas usando consultas:**

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

```
[{$addFields: {
  date: {
    $dateFromString: {
     dateString: "$Date",
     format: "%m/%d/%Y %H:%M"
    }
  }
}}, {$match: {
  date: {
```

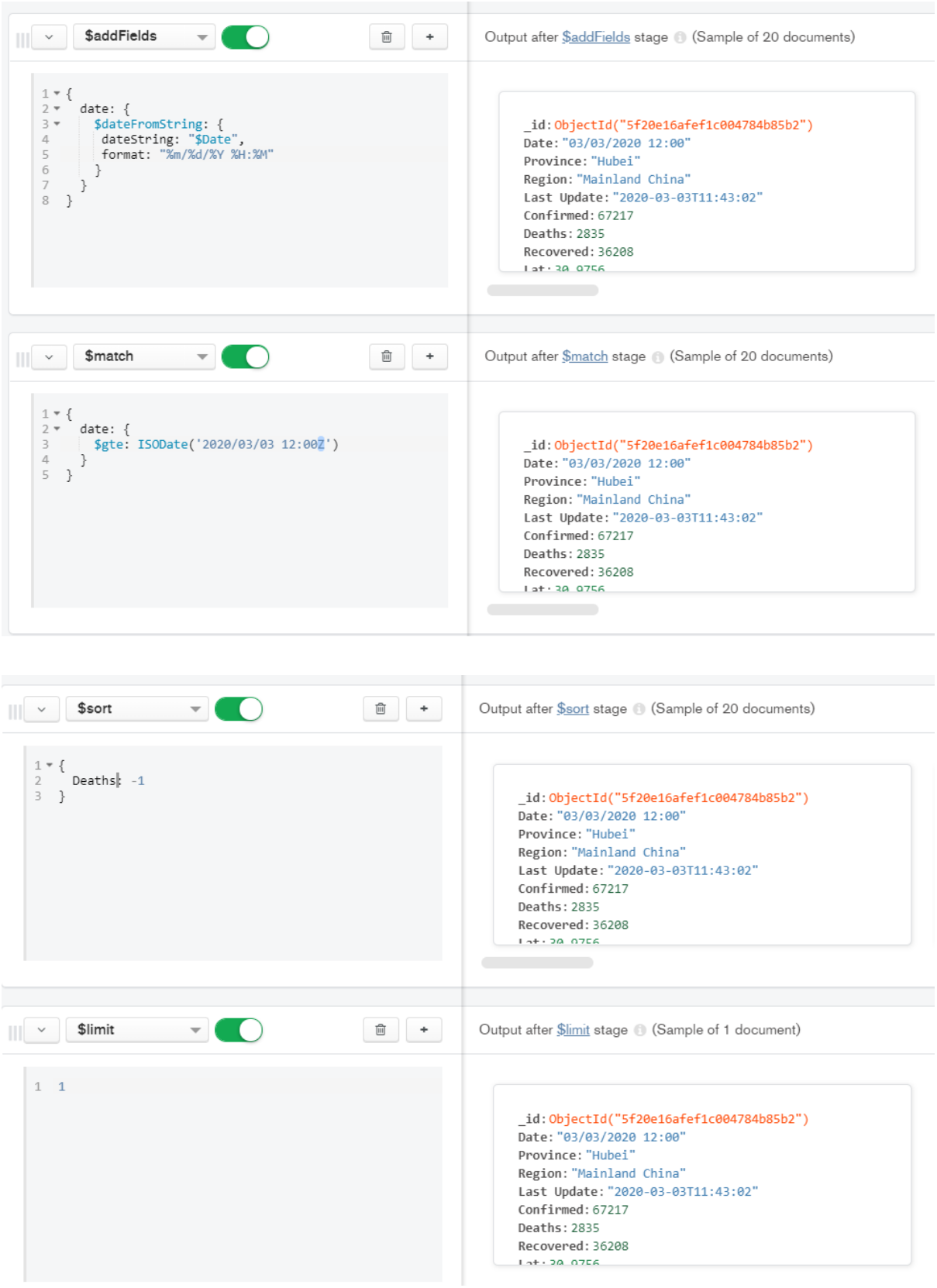
```
    $gte: ISODate('2020/03/03 12:00Z')
   }
}}, {$sort: {
   Confirmed: -1
}}, {$limit: 1}]
```





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

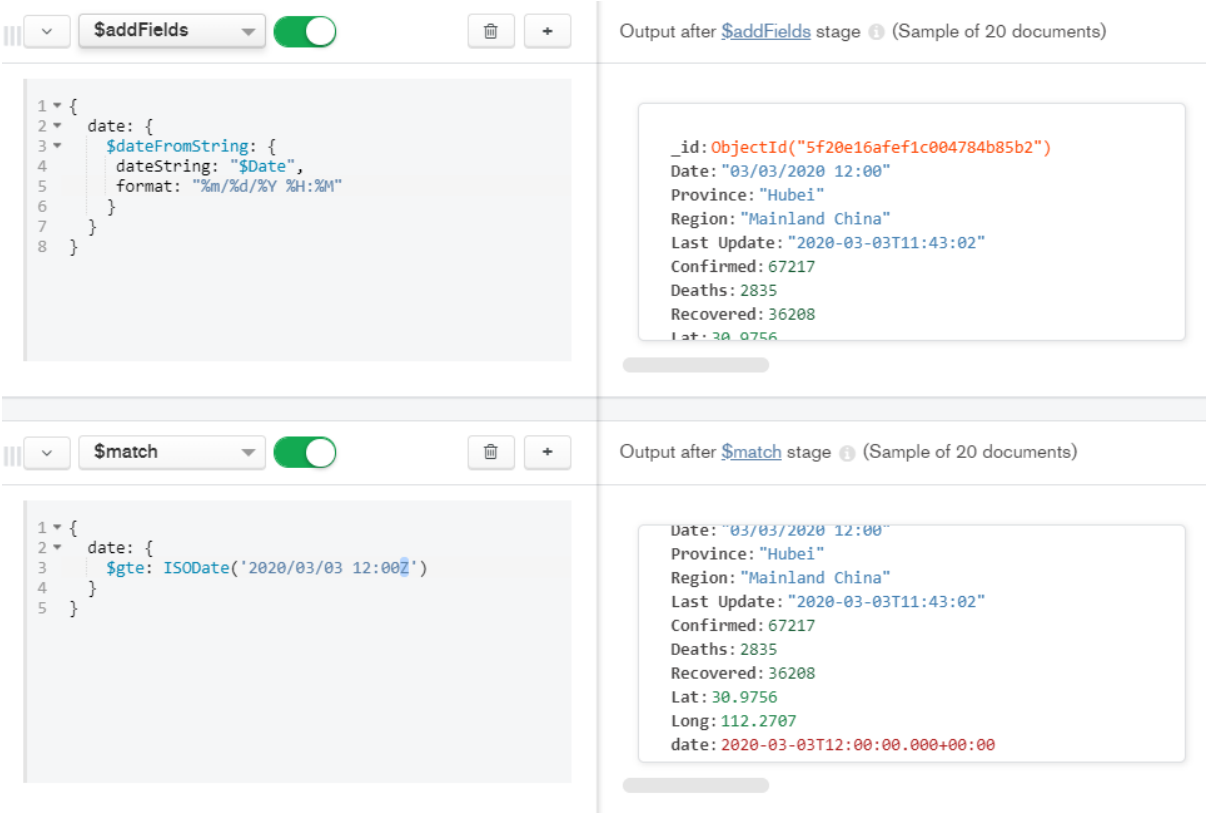
```
[{$addFields: {
  date: {
    $dateFromString: {
     dateString: "$Date",
     format: "%m/%d/%Y %H:%M"
    }
  }
}}, {$match: {
  date: {
    $gte: ISODate('2020/03/03 12:00Z')
  }
}}, {$sort: {
  Deaths: -1
}}, {$limit: 1}]
```



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

[{$addFields: {

```
    date: {
      $dateFromString: {
       dateString: "$Date",
       format: "%m/%d/%Y %H:%M"
      }
    }
  }}, {$match: {
    date: {
      $gte: ISODate('2020/03/03 12:00Z')
    }
  }}, {$group: {
    _id: null,
    Lat_promedio:{$avg:"$Lat"},
    Long_promedio:{$avg:"$Long"}
  }}]
```




$group
{
_id: null,
Lat_promedio:{$avg:"$Lat"},
Long_promedio:{$avg:"$Long"}
}
Output after $group stage (Sample of 1 document)
_id: null
Lat_promedio: 31.236601324503315
Long_promedio: 23.574501986754967