**1/Jul/2020 - Sesión 1**

Tablas relacionales tienen llaves únicas y llaves foráneas

**# Seleccionar BD**

USE tienda;

**# Mostrar bases**

SHOW databases;

\*No meter información en las bases de datos internas que se muestran

**# Mostrar tablas dentro de una base**

SHOW tables;

Las llaves primarias normalmente son textos

**# Describir tablas**

DESCRIBE empleado;

**# Mostrar solo la columna de nombre dentro la tabla empleado**

SELECT nombre FROM empleado

**# Mostrar todas las columnas dentro de tabla empleado**

SELECT \* FROM empleado

**# Filtros**

SELECT \* FROM empleado WHERE apellido\_paterno = “Risom”;

SELECT \* FROM empleado WHERE id\_puesto > 100;

**# Filtro con AND**

SELECT \* FROM empleado WHERE id\_puesto >= 100 AND id\_puesto <= 200;

**# Filtro para seleccionar los puestos 100 o 200**

SELECT \* FROM empleado WHERE id\_puesto = 100 OR id\_puesto = 200;

SELECT \* FROM empleado WHERE id\_puesto IN(100,200);

**RETO 2**

**# Nombre de los empleados con el puesto 4**

SELECT \* FROM empleado WHERE id\_puesto = 4;

**# Puestos que tienen un salario mayor a $10,000**

SELECT \* FROM puesto WHERE salario > 10000;

**# Artículos con precio mayor a $1000 y un iva mayor a 100**

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

**# Ventas incluyen artículos 135 o 963 y fueron hechas por empleados 835 o 369**

SELECT \* FROM venta WHERE (id\_articulo = 135 OR id\_articulo = 963) AND (id\_empleado = 835 OR id\_empleado = 369);

SELECT \* FROM venta WHERE id\_articulo IN (135,963) AND id\_empleado IN (835,369);

**# Ordenar datos en orden descendente**

SELECT \* FROM puesto order by salario desc;

**# Ordenar datos en orden ascendente (default es ascendente)**

SELECT \* FROM puesto ORDER BY salario;

SELECT \* FROM puesto ORDER BY salario asc;

**# Ordenar datos por orden alfabético**

SELECT \* FROM puesto ORDER BY nombre asc;

**# Limitar a 100 empleados**

SELECT \* FROM empleado LIMIT 100;

**RETO 3**

**# En base de datos tienda, obtener el top 5 de puestos por salarios**

SELECT \* FROM puesto ORDER BY salario desc LIMIT 5;

**PROYECTO 1**

**# Dentro del mismo servidor de bases de datos, conéctate al esquema classicmodels**

USE classicmodels;

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

SELECT lastName FROM employees;

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

SELECT lastName, firstName, jobTitle FROM employees;

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

SELECT \* FROM employees;

**# 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";

**# 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;

**# 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;

**# 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, jobTitle, officeCode FROM employees WHERE officeCode IN (1,2,3);

**# 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";

**# 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;

**# 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;

**# 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";

**# CHECAR Dentro de la tabla customers, obtén el nombre, país, estado y límite de crédito de todos los clientes cuyo país sea USA y cuyo estado sea CA y cuyo límite de crédito sea mayor a 1000000**

SELECT customerName, country, state, creditLimit FROM customers WHERE country = "USA" AND state = "CA" AND creditLimit > 100000;

**# 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");

**# 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;

SELECT customerName, country, creditLimit FROM customers WHERE (country = "USA" or country = "France") AND (creditLimit > 100000);

**# 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 = "USA" OR country = "France";

SELECT officeCode, city, phone, country FROM offices WHERE country IN ("USA","France");

**# CHECAR Dentro de la tabla offices, obtén el código de la oficina, ciudad, teléfono y país de aquellas oficinas que no se encuentren en USA o France.**

SELECT officeCode, city, phone, country FROM offices WHERE country != "USA" OR country != "France";

**# CHECAR 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 shippedDate IN (10165,10287,10310);

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

SELECT contactLastName, contactFirstName FROM customers ORDER BY contactLastName asc;

**# 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;

**# 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;

**# 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, contactFirstName, creditLimit FROM customers ORDER BY creditLimit DESC LIMIT 5;

**# 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.**

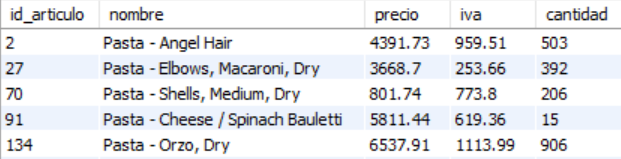
SELECT customerNumber, contactFirstName, creditLimit FROM customers ORDER BY creditLimit ASC LIMIT 5;

**6/Jul/2020 - SESIÓN 2**

***-- RETO 1***

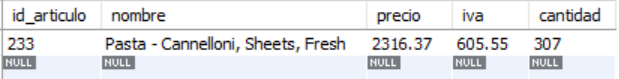
**# Artículos que incluyen la palabra Pasta en su nombre**

SELECT \* FROM articulo WHERE nombre LIKE '%Pasta%';



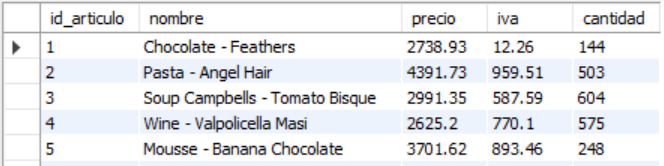
**# Artículos que incluyen la palabra Cannelloni en su nombre**

SELECT \* FROM articulo WHERE nombre LIKE '%Cannelloni%';



**# Nombres que están separados con un (-) por ejemplo Puree - Kiwi**

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



***-- RETO 2***

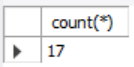
**# Cuál es el promedio de salario de los puestos?**

SELECT AVG(salario) FROM puesto;



**# Cuántos artículos incluyen la palabra Pasta en su nombre?**

SELECT count(\*) FROM articulo WHERE nombre LIKE '%Pasta%';



**# Cuál es el salario mínimo y máximo?**

SELECT max(salario), min(salario) FROM puesto;



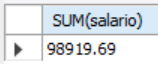
**# Cuál es la suma del salario de los últimos cinco puestos agregados?**

SELECT SUM(salario) FROM puesto ORDER BY id\_puesto DESC LIMIT 5;

#ESTA MAL, NO TOMA EN CUENTA EL LIMIT, SUMA TODOS LOS REGISTROS

SELECT SUM(salario) FROM puesto WHERE id\_puesto >= 995;

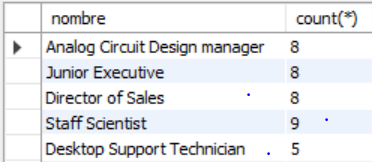
#PRIMERO TENEMOS QUE CONOCER CUÁNTOS PUESTOS HAY PARA PONER LOS 995



***-- RETO 3***

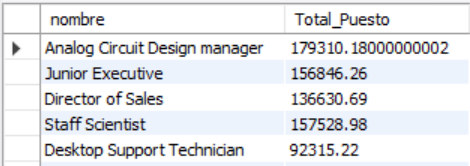
**# Cuántos registros hay por cada uno de los puestos?**

SELECT nombre, count(\*) FROM puesto GROUP BY nombre;



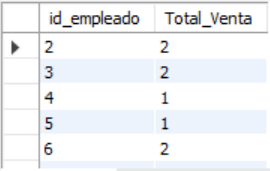
**# Cuánto dinero se paga en total por puesto?**

SELECT nombre, sum(salario) AS Total\_Puesto FROM puesto GROUP BY nombre;



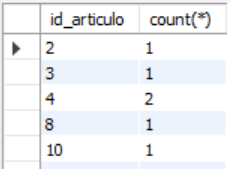
**# ¿Cuál es el número total de ventas por vendedor?**

SELECT id\_empleado, count(clave) AS Total\_Venta FROM venta GROUP BY id\_empleado;



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

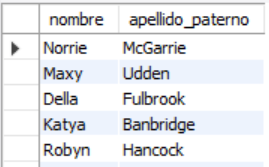
SELECT id\_articulo, count(\*) FROM venta GROUP BY id\_articulo;



***-- RETO 4***

**# Nombre de los empleados cuyo sueldo es menor a 10,000**

SELECT id\_puesto FROM puesto WHERE salario < 20000;



SELECT nombre, apellido\_paterno

FROM empleado

WHERE id\_puesto IN

(SELECT id\_puesto FROM puesto WHERE salario < 20000);

**# Cantidad minima y maxima de ventas de cada empleado**

SELECT clave, id\_empleado, count(\*) total\_ventas

FROM venta

    GROUP BY clave, id\_empleado;

SELECT id\_empleado, min(total\_ventas), max(total\_ventas)

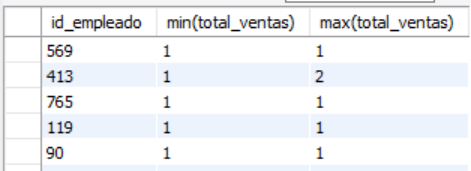
FROM

(SELECT clave, id\_empleado, count(\*) total\_ventas

FROM venta

    GROUP BY clave, id\_empleado) AS sq

GROUP BY id\_empleado;

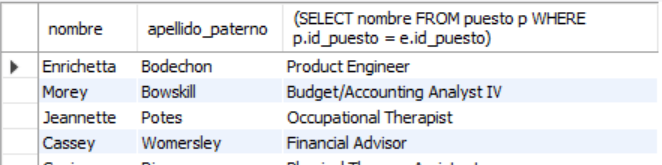


**# Cual es el nombre del puesto de cada empleado?**

SELECT nombre FROM puesto p WHERE p.id\_puesto = 5;

SELECT nombre, apellido\_paterno, (SELECT nombre FROM puesto p WHERE p.id\_puesto = e.id\_puesto)

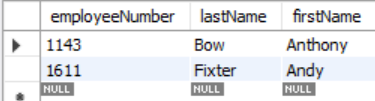
FROM empleado e;



**PROYECTO 2**

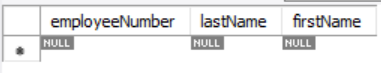
1. Dentro de la tabla employees, obten el número de empleado, apellido y nombre de todos los empleados cuyo nombre empiece con a.

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



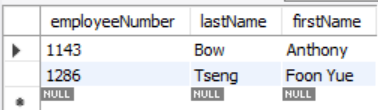
1. Dentro de la tabla employees, obten el número de empleado, apellido y nombre de todos los empleados cuyo nombre termina con on

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



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

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



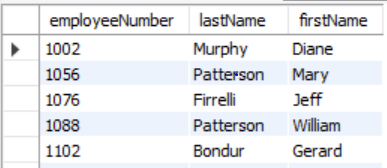
1. Dentro de la tabla employees, obten el número de empleado, apellido y nombre de todos los empleados cuyos nombres tienen tres letras e inician con T y finalizan con m.

SELECT employeeNumber, lastName, firstName FROM employees WHERE firstName LIKE 'T\_m';



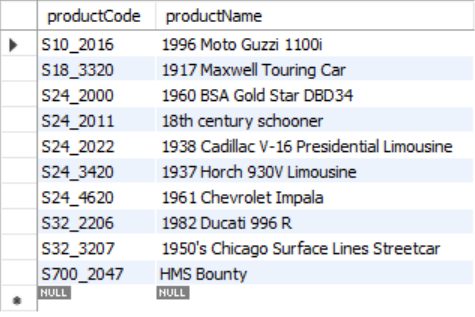
1. Dentro de la tabla employees, obten el número de empleado, apellido y nombre de todos los empleados cuyo nombre *no* inicia con B.

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



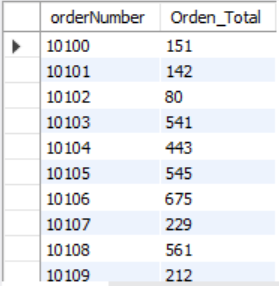
1. Dentro de la tabla products, obten el código de producto y nombre de los productos cuyo código incluye la cadena \_20.

SELECT productCode, productName FROM products WHERE productCode LIKE '%20%';

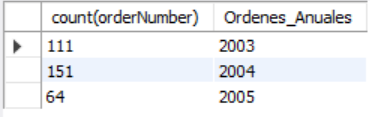


1. Dentro de la tabla orderdetails, obten el total de cada orden.

SELECT orderNumber, sum(quantityOrdered) AS Orden\_Total FROM orderdetails GROUP BY orderNumber;



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



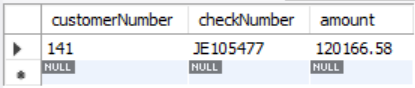
1. Obten el apellido y nombre de los empleados cuya oficina está ubicada en USA.

SELECT firstName, lastName FROM employees WHERE officeCode IN   
(SELECT officeCode FROM offices WHERE country = "USA");



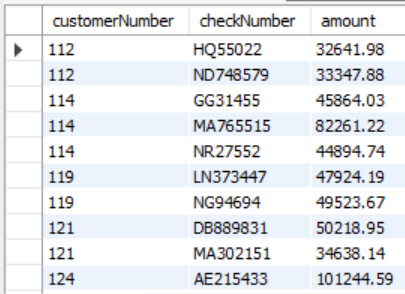
1. Obten el número de cliente, número de cheque y cantidad del cliente que ha realizado el pago más alto.

SELECT customerNumber, checkNumber, amount FROM payments WHERE amount = (SELECT max(amount) FROM payments);



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

SELECT customerNumber, checkNumber, amount FROM payments WHERE amount > (SELECT avg(amount) FROM payments);



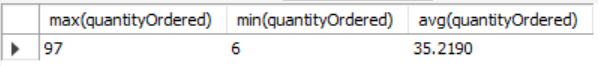
1. Obten el nombre de aquellos clientes que no han hecho ninguna orden.

SELECT customerNumber, customerName FROM customers WHERE customerNumber NOT IN (SELECT customerNumber FROM orders);



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

SELECT max(quantityOrdered), min(quantityOrdered), avg(quantityOrdered) FROM orderdetails;

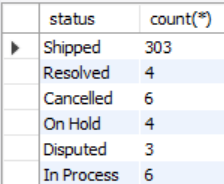


1. Dentro de la tabla orders, obten el número de órdenes que hay por cada estado.

SELECT status, count(\*)

FROM orders

GROUP BY status;

****

**8/Jul/2020 - SESIÓN 3**

**-- RETO 1**

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

select e.nombre, v.clave from empleado e

join venta v

on e.id\_empleado = v.id\_empleado;



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

select a.nombre as articulo

from articulo a

join venta v

on a.id\_articulo = v.id\_articulo

GROUP BY articulo;



# Cuál es el total de cada venta?

select clave, round(sum(precio),2) as total

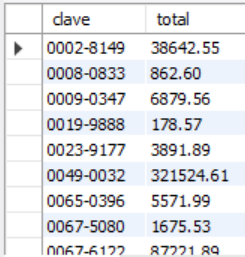
from venta AS v

join articulo AS a

on v.id\_articulo = a.id\_articulo

GROUP BY clave

ORDER BY clave;



**-- RETO 2**

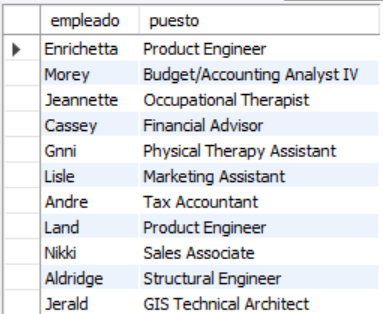
# Obtener puesto de un empleado

create view SHpuesto as

(select e.nombre as empleado , p.nombre as puesto from empleado e

join puesto p

on e.id\_puesto = p.id\_puesto);



# Saber que articulo ha vendido cada empleado

create view SHarticulo as

(select a.nombre as articulo, e.nombre as empleado

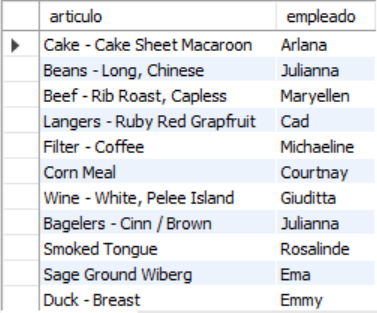
from articulo a

join venta v

on a.id\_articulo = v.id\_articulo

join empleado e

on e.id\_empleado = v.id\_empleado);



# Saber que puesto ha tenido más ventas

SELECT p.nombre, count(v.clave) as total

FROM venta v

JOIN empleado e

ON v.id\_empleado = e.id\_empleado

JOIN puesto p

ON e.id\_puesto = p.id\_puesto

GROUP BY p.nombre

ORDER BY total DESC;



**PROYECTO 3**

*Para estas consultas usa RIGHT JOIN*

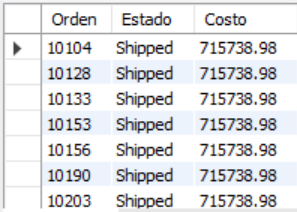
1. Obtén el código de producto, nombre de producto y descripción de todos los productos.

SELECT productCode, productName, productDescription FROM products;



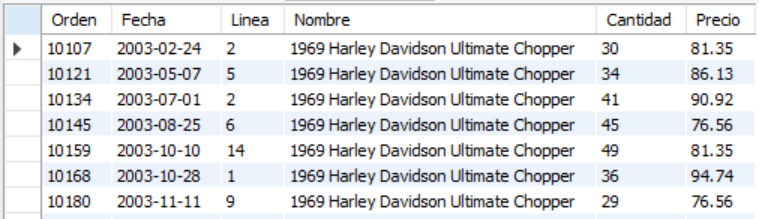
2. Obtén el número de orden, estado y costo total de cada orden.

SELECT o.orderNumber AS Orden, o.status AS Estado, sum(p.amount) as Costo  
FROM orders o  
RIGHT JOIN payments p  
ON p.customerNumber = o.customerNumber  
GROUP BY Orden  
ORDER BY Costo DESC;



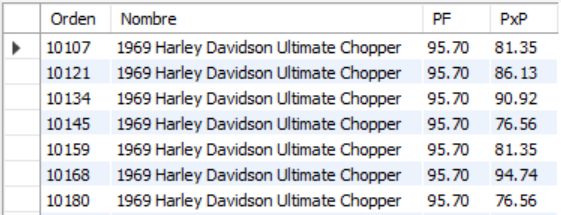
3. Obtén el número de orden, fecha de orden, línea de orden, nombre del   
producto, cantidad ordenada y precio de cada pieza que muestre los detalles de cada orden.

SELECT o.orderNumber AS Orden, o.orderDate AS Fecha, od.orderLineNumber AS Linea, p.productName AS Nombre, od.quantityOrdered AS Cantidad, od.priceEach AS Precio   
FROM orders o  
RIGHT JOIN orderdetails od  
ON o.orderNumber = od.orderNumber  
RIGHT JOIN products p  
ON od.productCode = p.productCode;



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

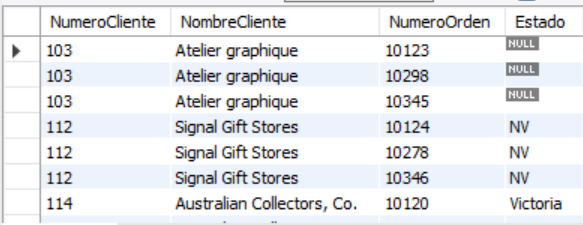
SELECT o.orderNumber AS Orden, p.productName AS Nombre, p.MSRP AS PF, od.priceEach AS PxP  
FROM orders o  
RIGHT JOIN orderdetails od  
ON o.orderNumber = od.orderNumber  
RIGHT JOIN products p  
ON od.productCode = p.productCode;



*Para estas consultas usa LEFT JOIN*

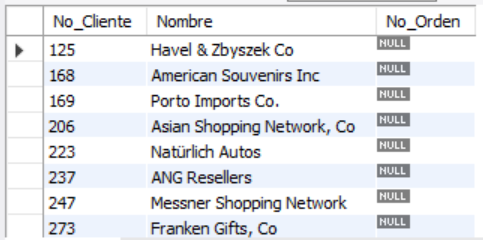
1. Obtén el número de cliente, nombre de cliente, número de orden y estado de cada cliente.

SELECT c.customerNumber AS NumeroCliente, c.customerName AS NombreCliente, o.orderNumber AS NumeroOrden, c.state AS Estado  
FROM customers c  
LEFT JOIN orders o  
ON c.customerNumber = o.customerNumber;



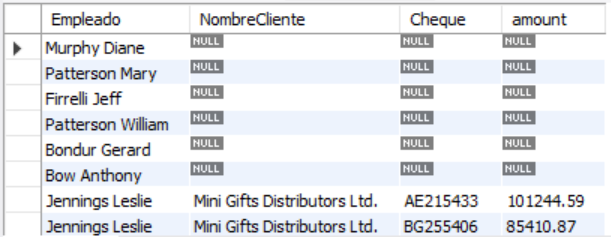
6. Obtén los clientes que no tienen una orden asociada.

SELECT c.customerNumber AS No\_Cliente, c.customerName AS Nombre, o.orderNumber AS No\_Orden  
FROM customers c  
LEFT JOIN orders o  
ON c.customerNumber = o.customerNumber  
WHERE o.customerNumber IS null;



7. Obtén el apellido de empleado, nombre de empleado, nombre de cliente, número de cheque y total, es decir, los clientes asociados a cada empleado.

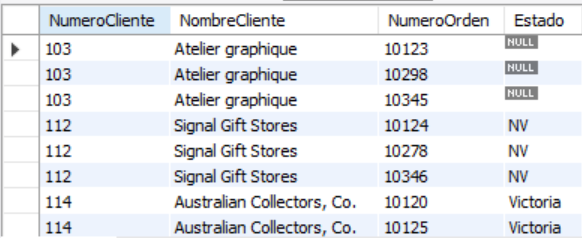
SELECT concat(e.lastName, ' ', e.firstName) AS Empleado, c.customerName AS NombreCliente, p.checkNumber AS Cheque, p.amount  
FROM employees e  
LEFT JOIN customers c  
ON e.employeeNumber = c.salesRepEmployeeNumber  
LEFT JOIN payments p  
ON p.customerNumber = c.customerNumber;



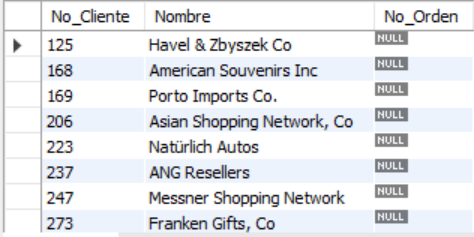
*Para estas consultas usa RIGHT JOIN*

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

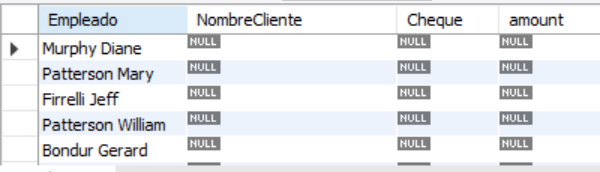
SELECT c.customerNumber AS NumeroCliente, c.customerName AS NombreCliente, o.orderNumber AS NumeroOrden, c.state AS Estado  
FROM customers c  
RIGHT JOIN orders o  
ON c.customerNumber = o.customerNumber;



SELECT c.customerNumber AS No\_Cliente, c.customerName AS Nombre, o.orderNumber AS No\_Orden  
FROM orders o  
RIGHT JOIN customers c  
ON c.customerNumber = o.customerNumber  
WHERE o.customerNumber IS null;



SELECT concat(e.lastName, ' ', e.firstName) AS Empleado, c.customerName AS NombreCliente, p.checkNumber AS Cheque, p.amount  
FROM payments p  
RIGHT JOIN customers c  
ON p.customerNumber = c.customerNumber  
RIGHT JOIN employees e  
ON c.salesRepEmployeeNumber = e.employeeNumber;



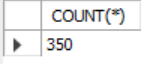
1. Escoge 3 consultas de los ejercicios anteriores, crea una vista y escribe una consulta para cada una.

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

CREATE VIEW SHCo1 AS  
(SELECT c.customerNumber AS NumeroCliente, c.customerName AS NombreCliente, o.orderNumber AS NumeroOrden, c.state AS Estado  
FROM customers c  
LEFT JOIN orders o  
ON c.customerNumber = o.customerNumber);

#Número de órdenes por cliente

SELECT COUNT(\*)  
FROM SHCo1;



CREATE VIEW SHCo2 AS  
(SELECT o.orderNumber AS Orden, p.productName AS Nombre, p.MSRP AS PF, od.priceEach AS PxP  
FROM orders o  
RIGHT JOIN orderdetails od  
ON o.orderNumber = od.orderNumber  
RIGHT JOIN products p  
ON od.productCode = p.productCode);

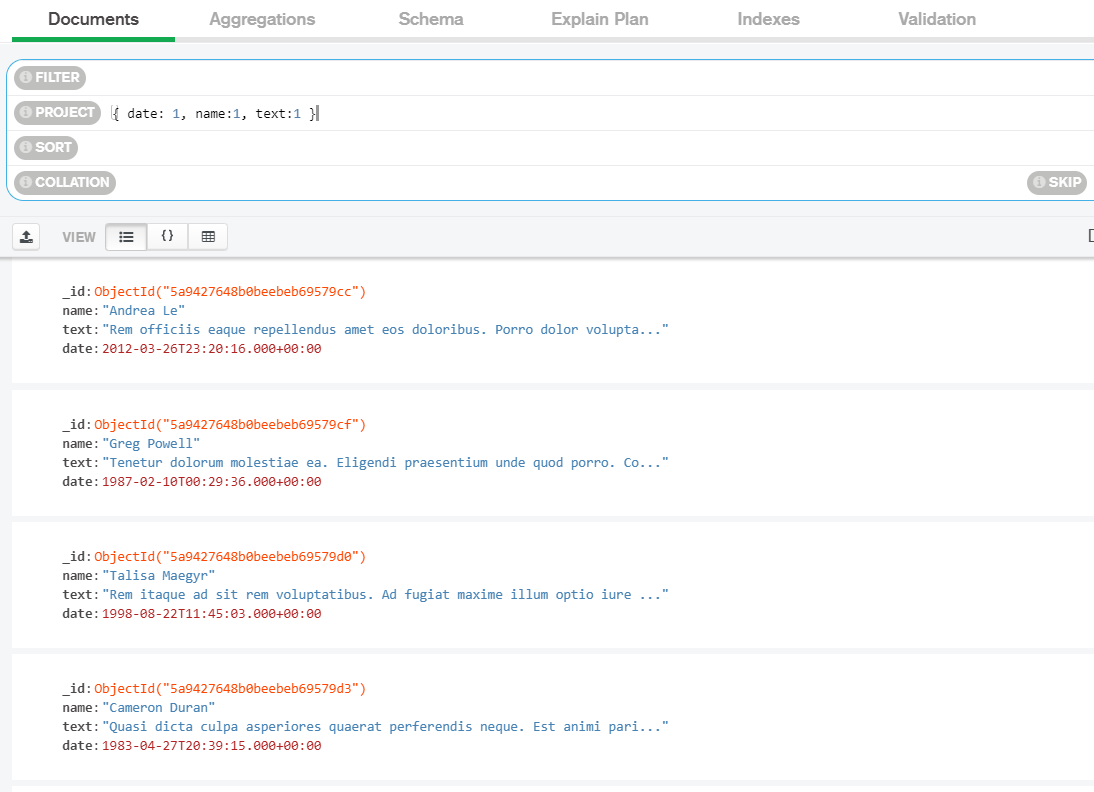
CREATE VIEW SHCo3 AS  
(SELECT o.orderNumber AS Orden, o.orderDate AS Fecha, od.orderLineNumber AS Linea, p.productName AS Nombre, od.quantityOrdered AS Cantidad, od.priceEach AS Precio   
FROM orders o  
RIGHT JOIN orderdetails od  
ON o.orderNumber = od.orderNumber  
RIGHT JOIN products p  
ON od.productCode = p.productCode);

**13/Jul/2020 - SESIÓN 4**

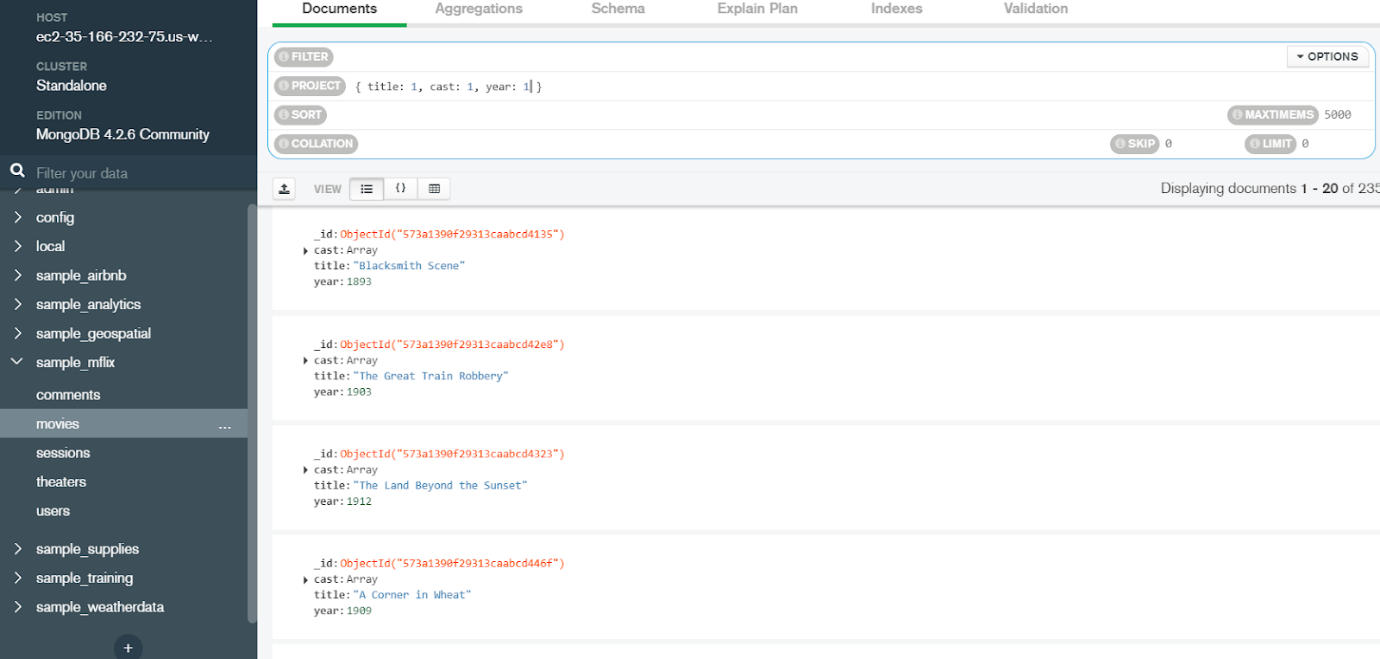
**WORK 4**

**RETO 1**

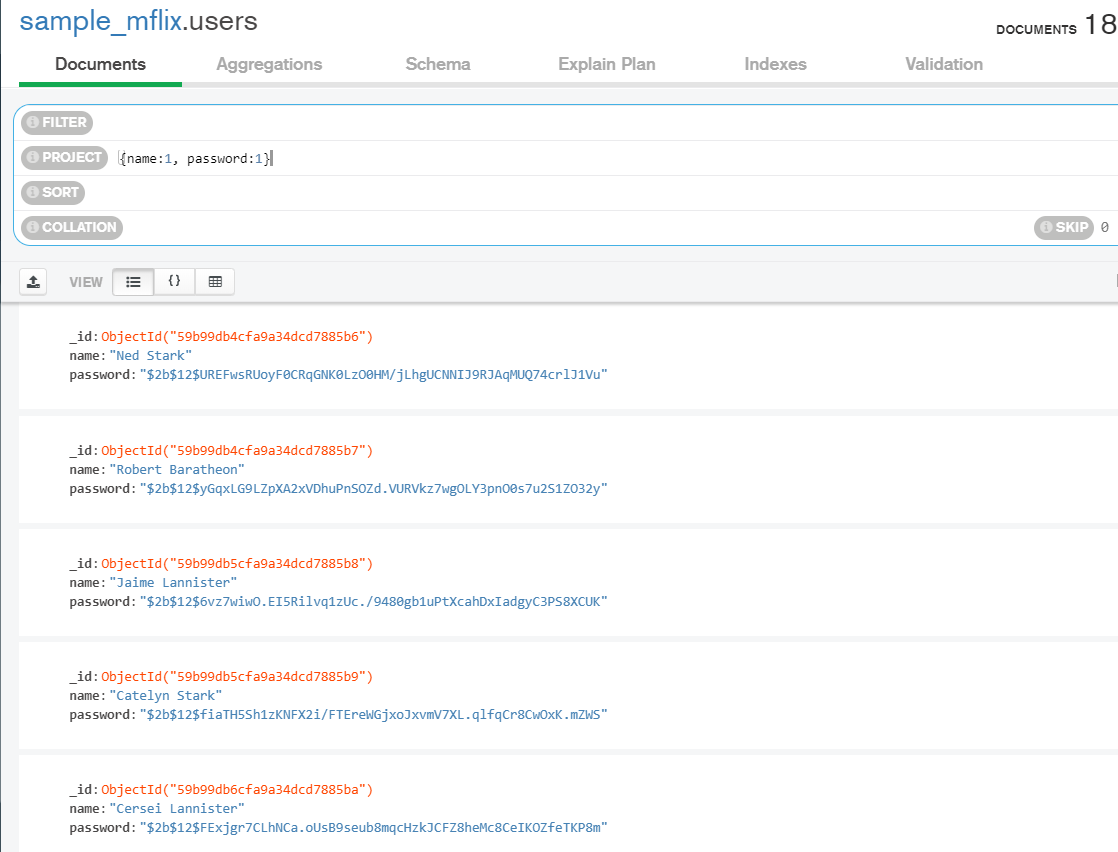
**Fecha, nombre y texto de cada comentario**

****

**Titulo, elenco y año de cada película**

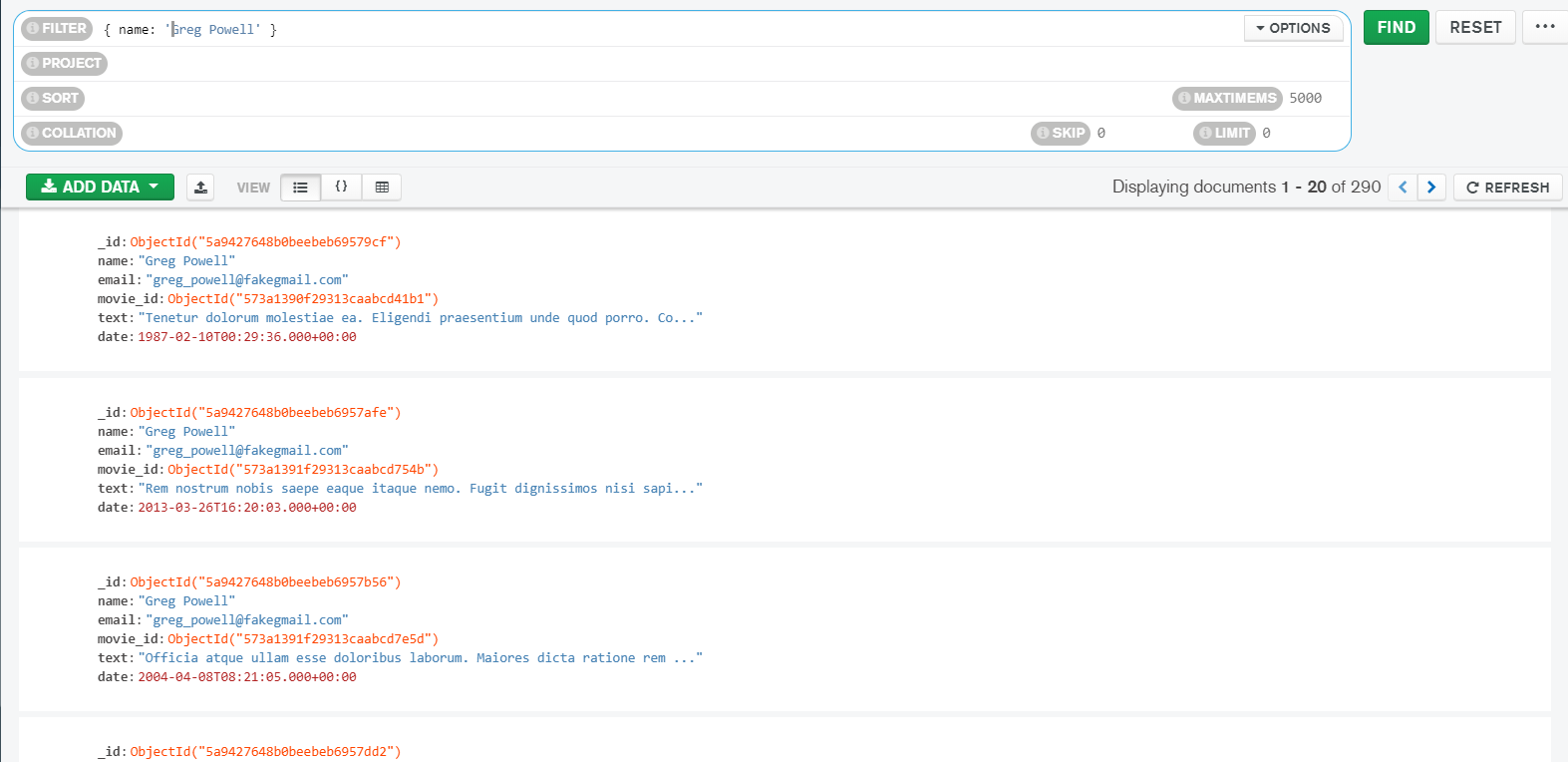
****

**Nombre y contrasena de cada usuario**

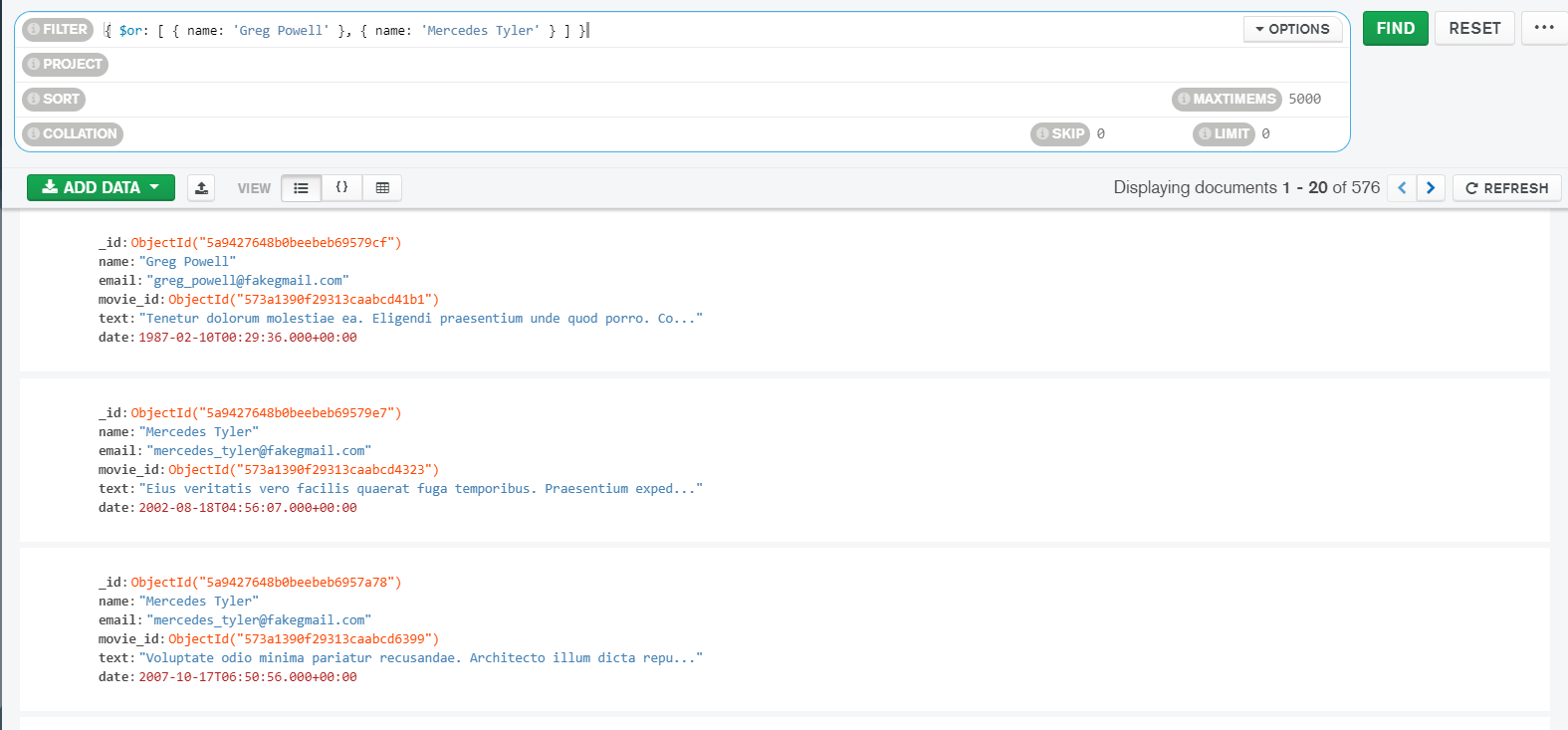
****

**RETO 2**

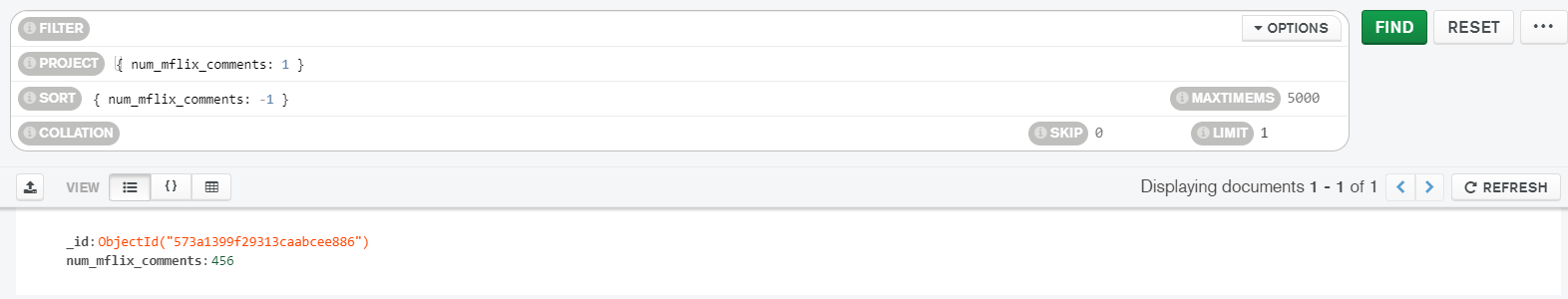
**Comentario hecho por Greg Powell**

****

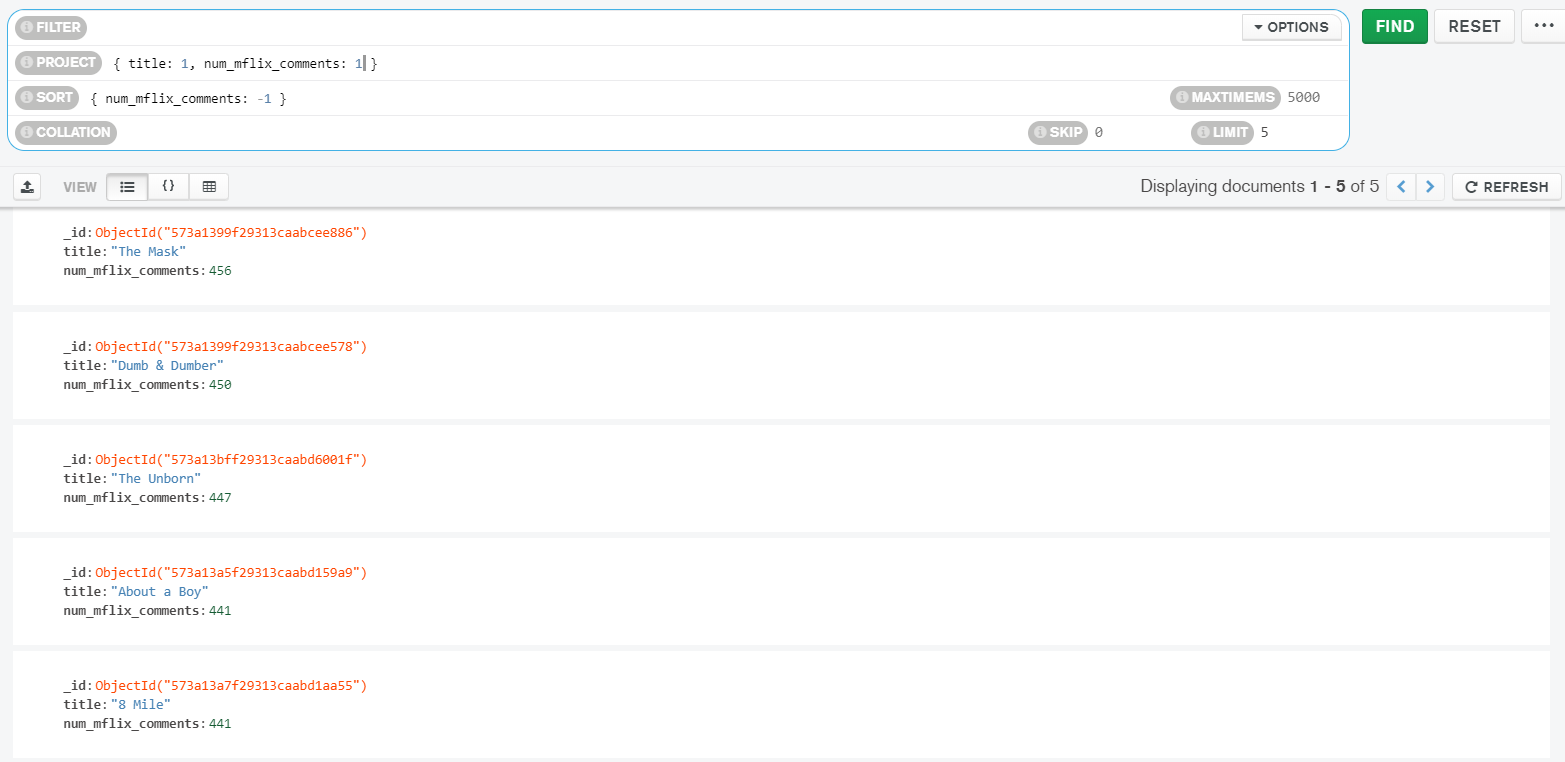
**Comentario hecho por Greg o Mercedes**

****

**Numero maximo de comentarios en una pelicula**

****

**Titulo de las 5 peliculas mas comentadas**

****

**PROYECTO 4**

1. Obtén los datos de contacto de cada compañía.

{

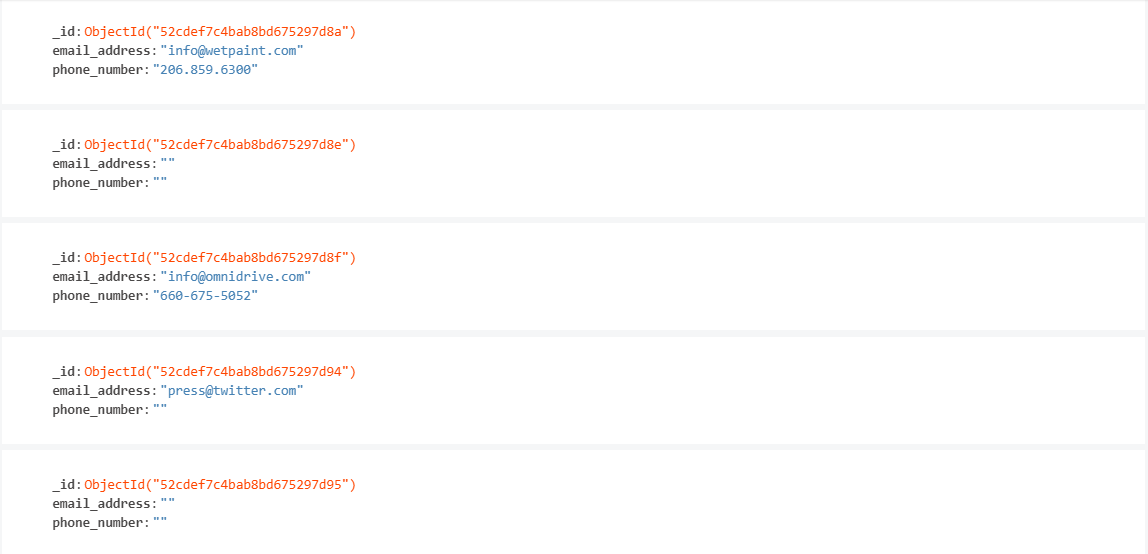
 project: {

  email\_address: 1,

  phone\_number: 1

 }

}



1. Obtén la fuente de cada tweet.

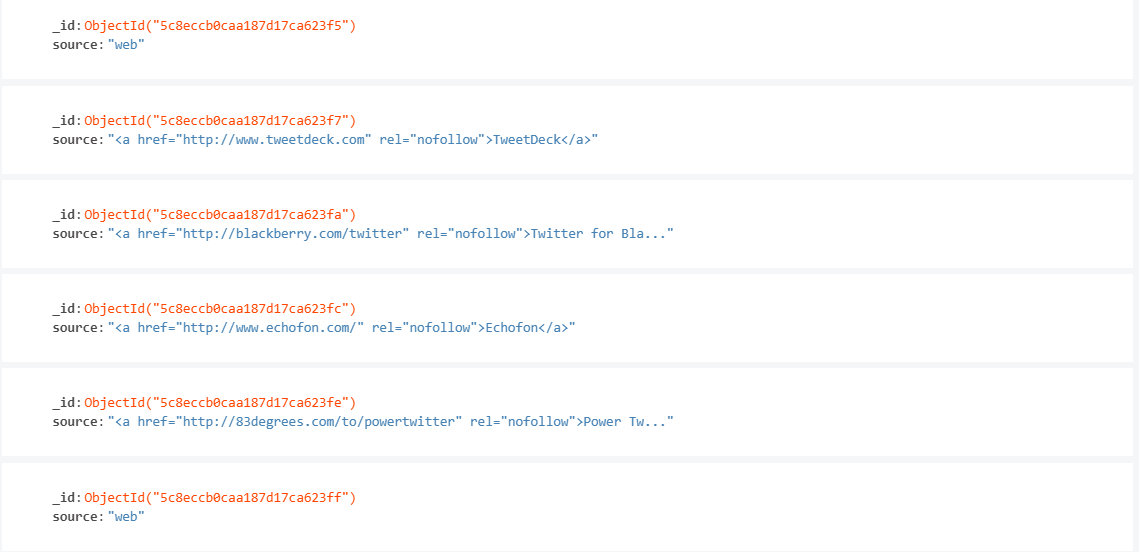
{

 project: {

  source: 1

 }

}



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

{

 filter: {

  founded\_month: 10

 },

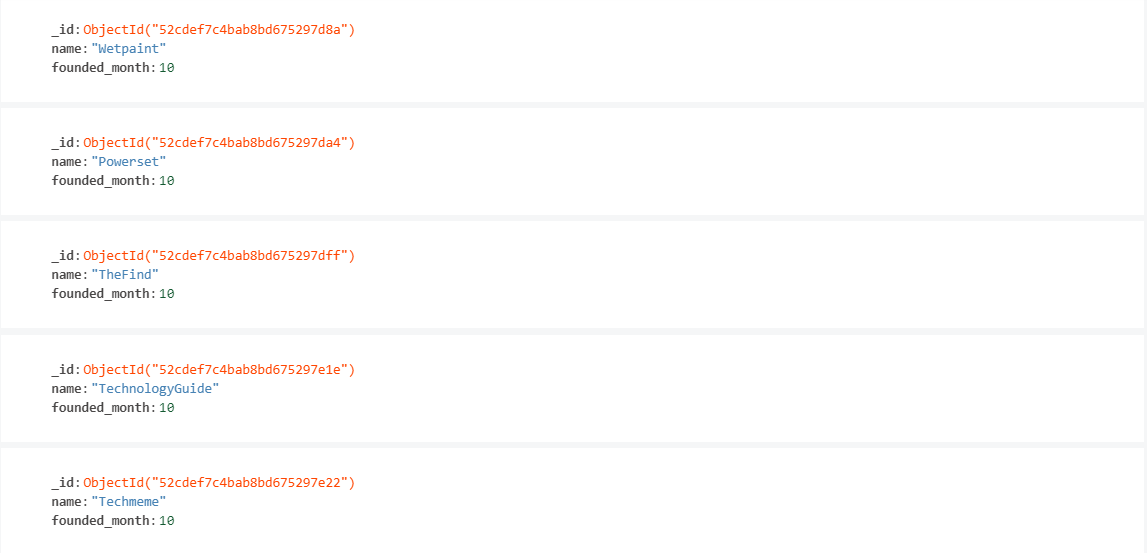
 project: {

  name: 1,

  founded\_month: 1

 }

}



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

{

 filter: {

  founded\_year: 2008

 },

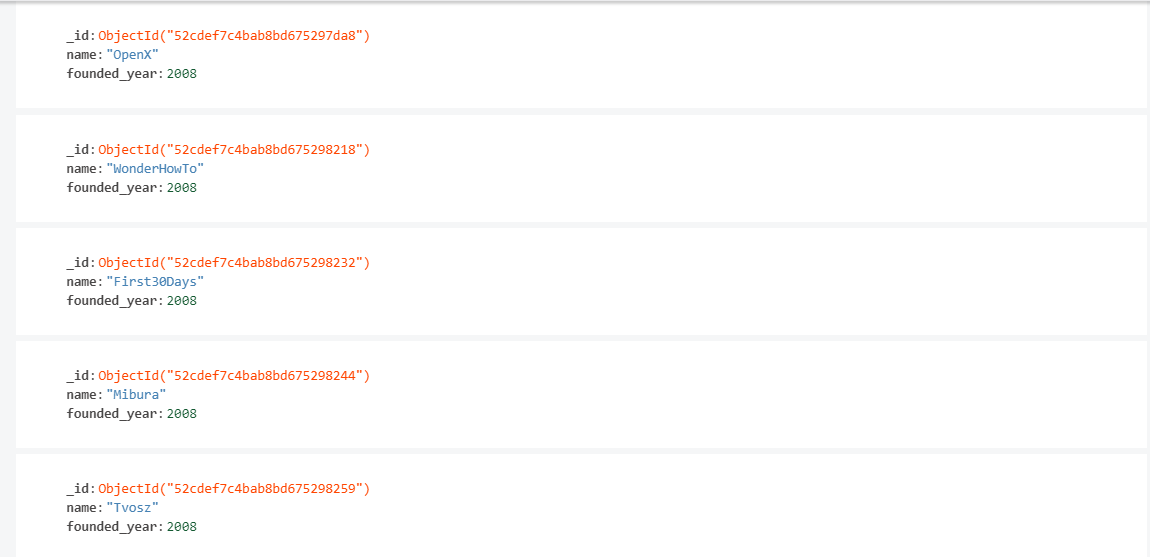
 project: {

  name: 1,

  founded\_year: 1

 }

}



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

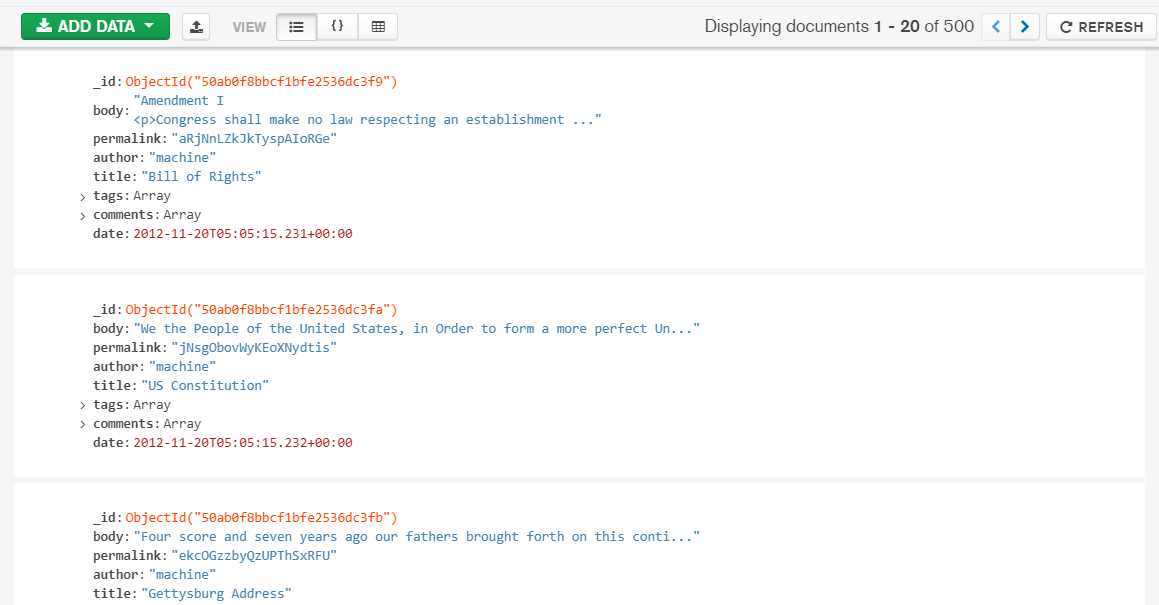
{

 filter: {

  author: 'machine'

 }

}



1. Obtén todos los tweets provenientes de la web.

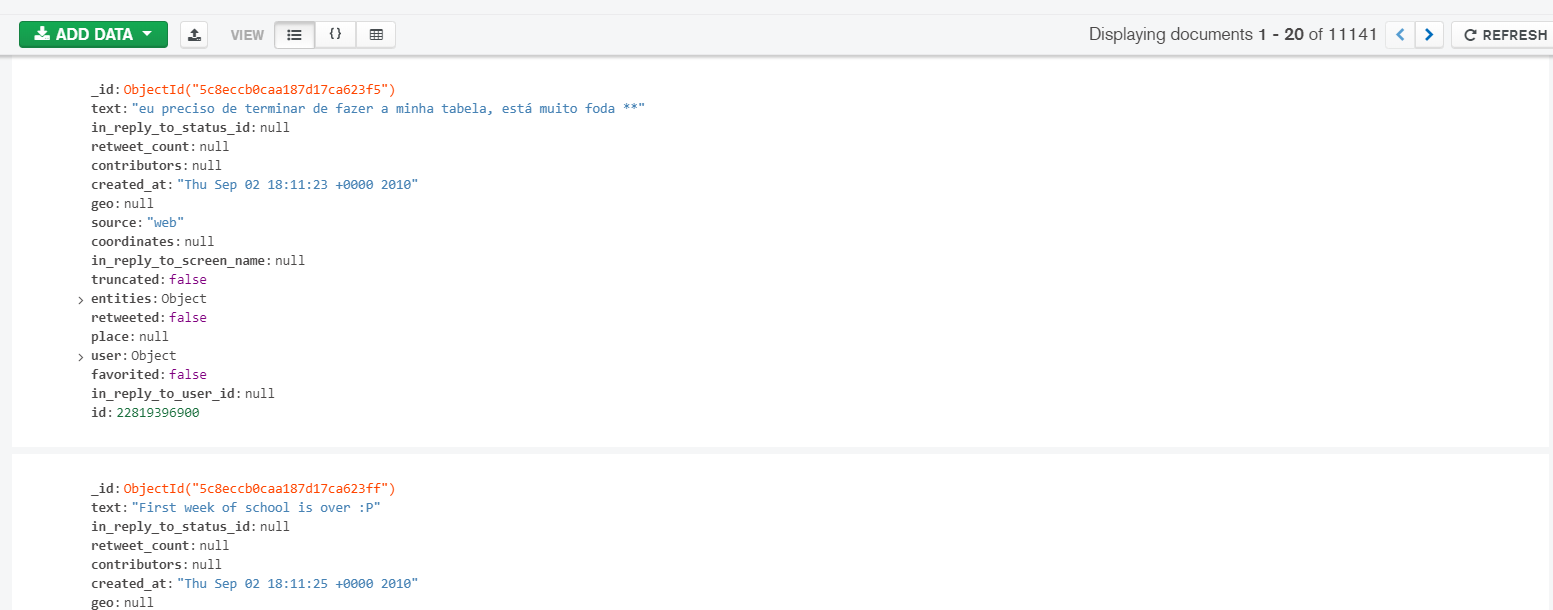
{

 filter: {

  source: 'web'

 }

}



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

{

 filter: {

  $and: [

   {

    founded\_year: 2008

   },

   {

    founded\_month: 10

   }

  ]

 },

 project: {

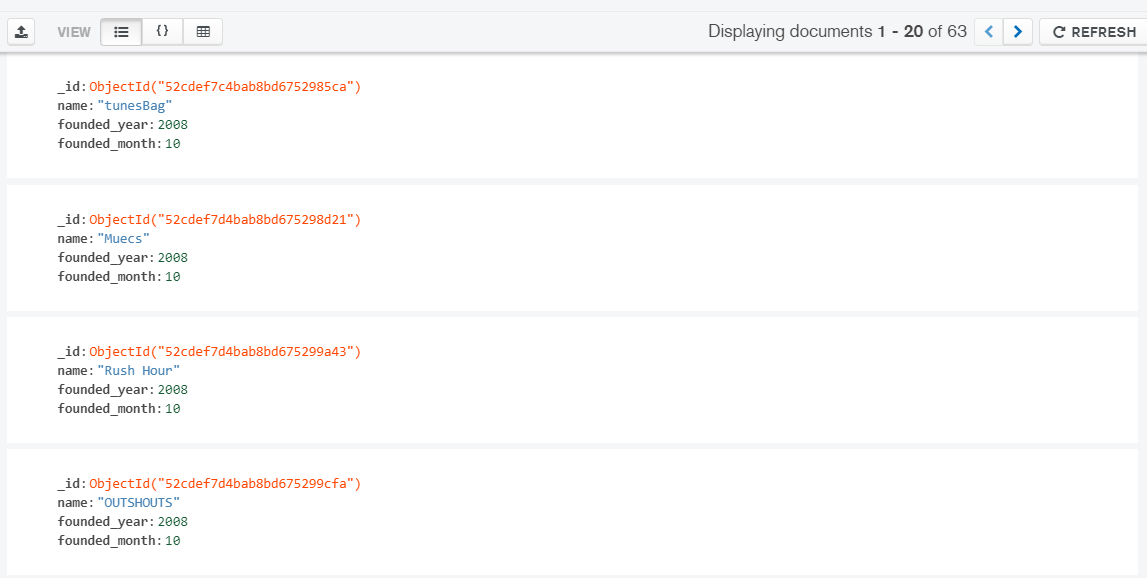
  name: 1,

  founded\_year: 1,

  founded\_month: 1

 }

}



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

{

 filter: {

  number\_of\_employees: {

   $gte: 50

  }

 },

 project: {

  name: 1,

  number\_of\_employees: 1

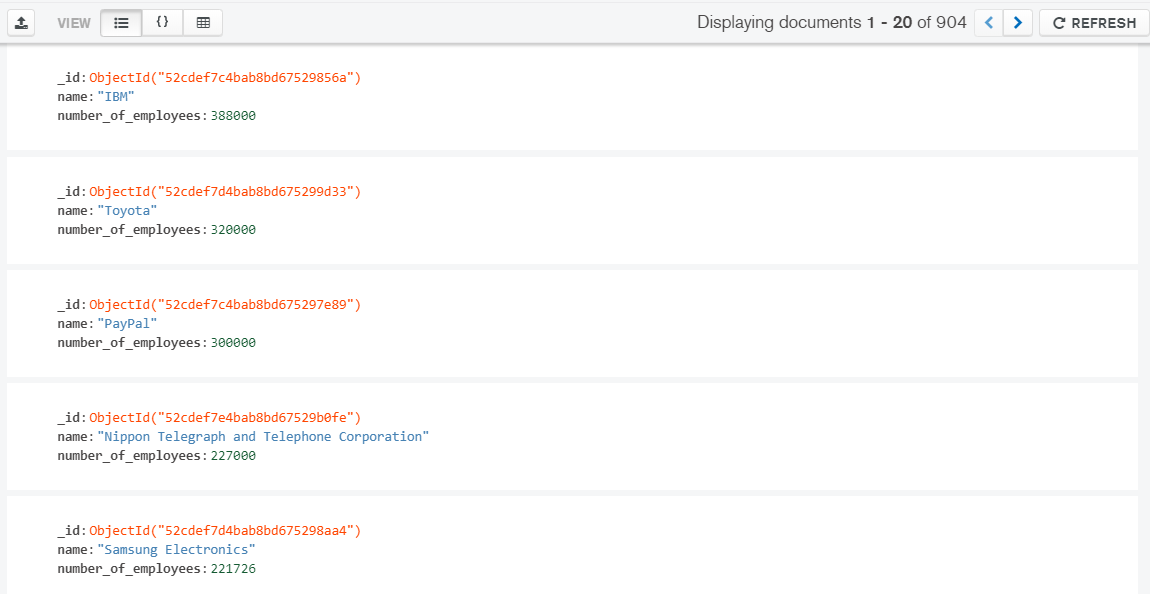
 },

 sort: {

  number\_of\_employees: -1

 }

}



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

{

 filter: {

  $and: [

   {

    comments: {

     $gte: 10

    }

   },

   {

    comments: {

     $lte: 30

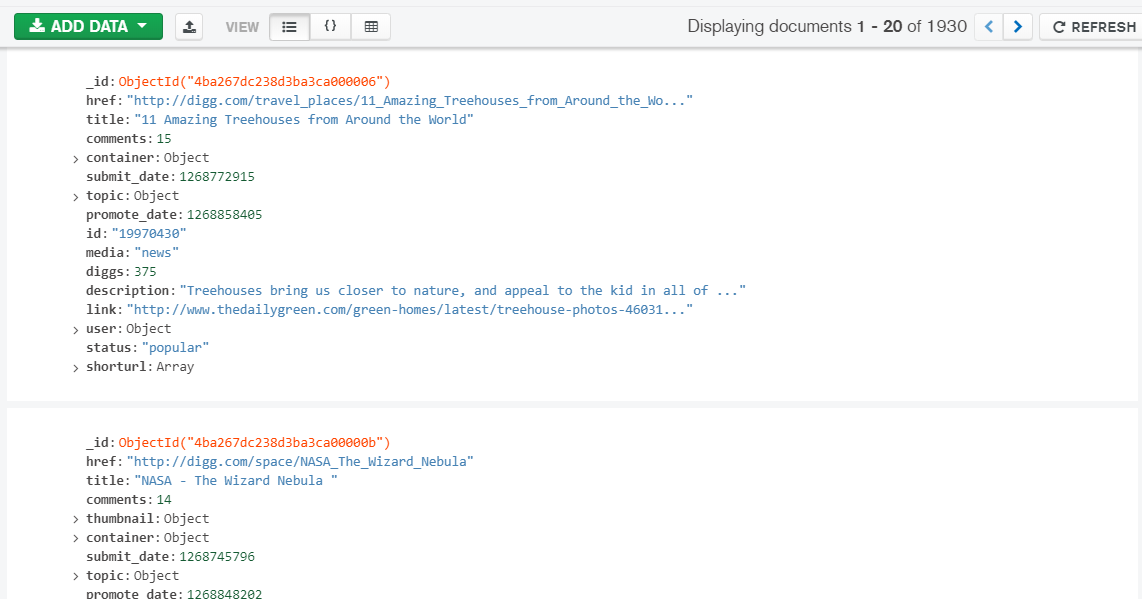
    }

   }

  ]

 }

}



1. 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

 },

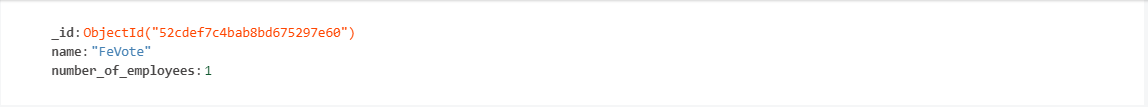
 sort: {

  number\_of\_employees: 1

 },

 limit: 1

}



1. 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

 },

 sort: {

  number\_of\_employees: -1

 },

 limit: 1

}



1. Obtén la historia más comentada.

{

 filter: {

  $and: [

   {

    comments: {

     $ne: null

    }

   },

   {

    comments: {

     $ne: 0

    }

   }

  ]

 },

 project: {

  title: 1,

  comments: 1

 },

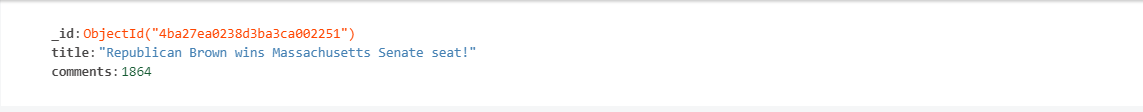
 sort: {

  comments: -1

 },

 limit: 1

}



1. Obtén la historia menos comentada.

{

 filter: {

  $and: [

   {

    comments: {

     $ne: null

    }

   },

   {

    comments: {

     $ne: 0

    }

   }

  ]

 },

 project: {

  title: 1,

  comments: 1

 },

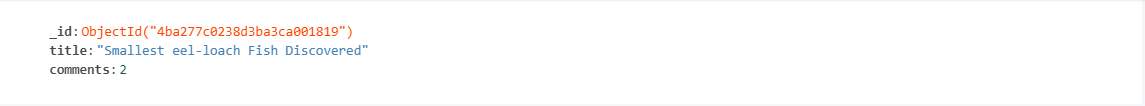
 sort: {

  comments: 1

 },

 limit: 1

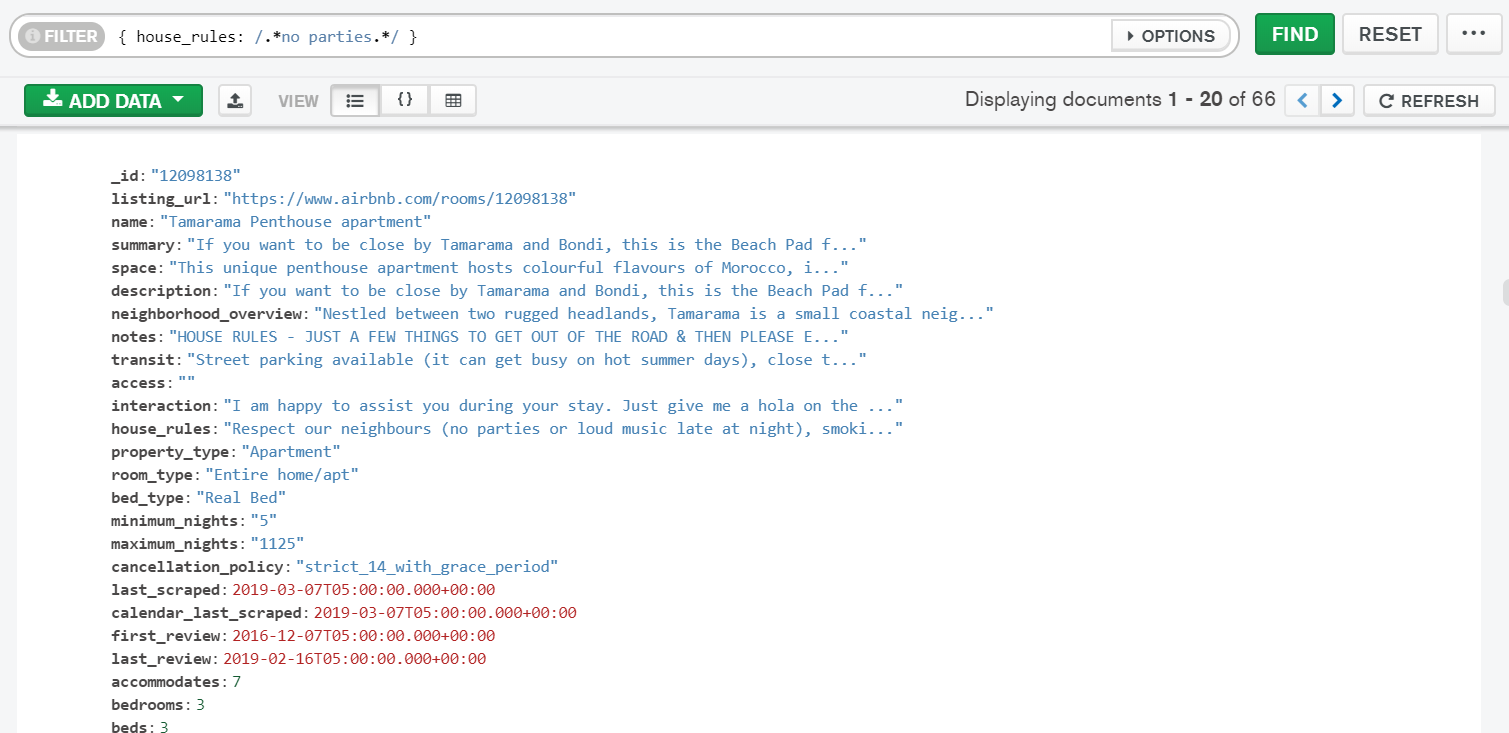
}



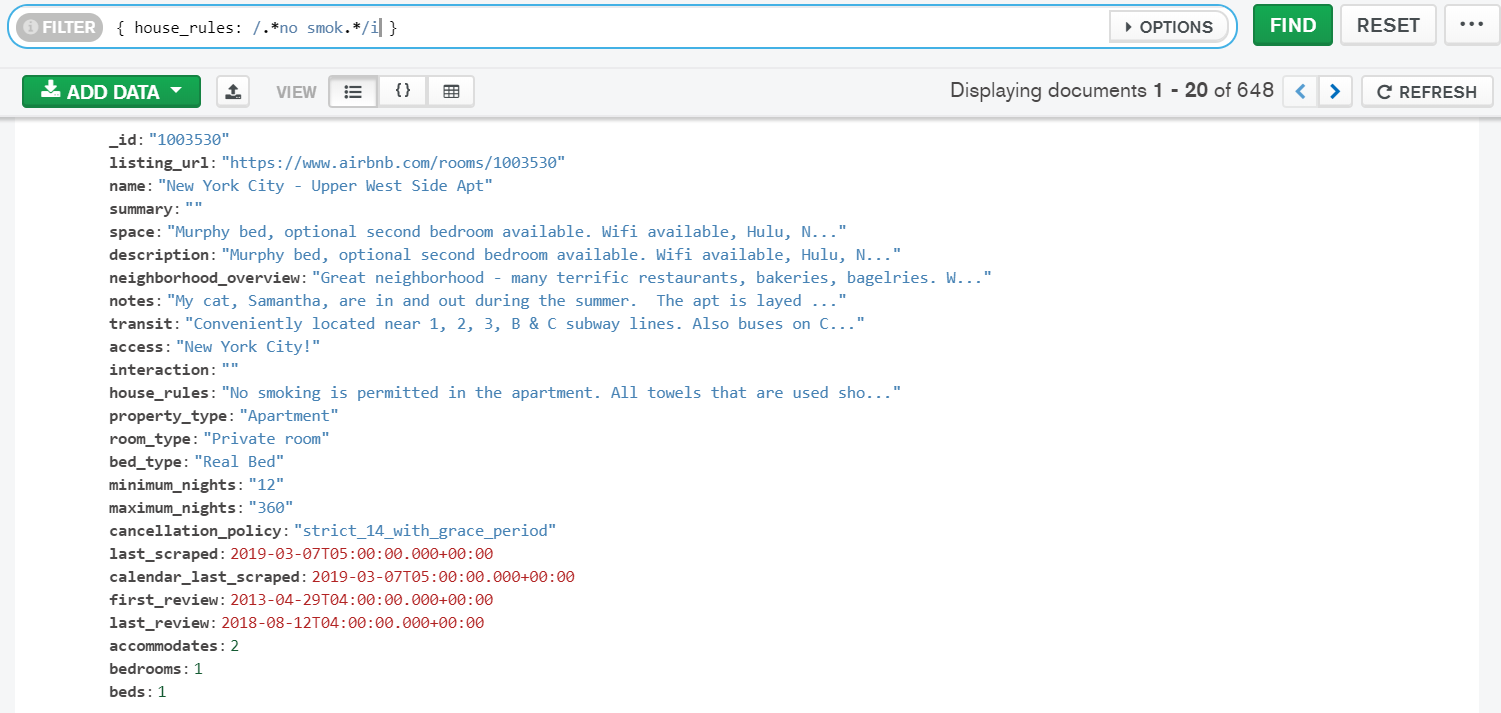
**15/Jul/2020 - Sesión 5**

**RETO 1**

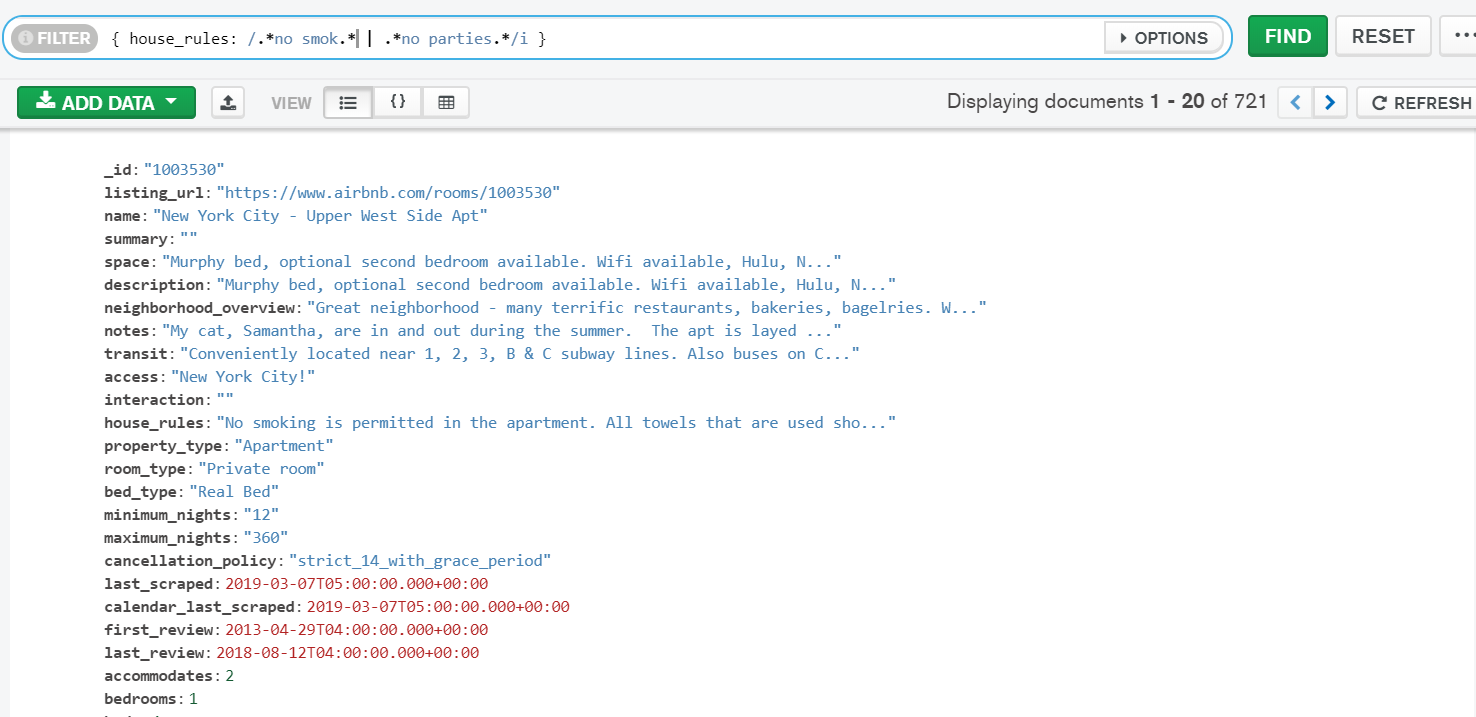
Propiedades que no permitan fiestas



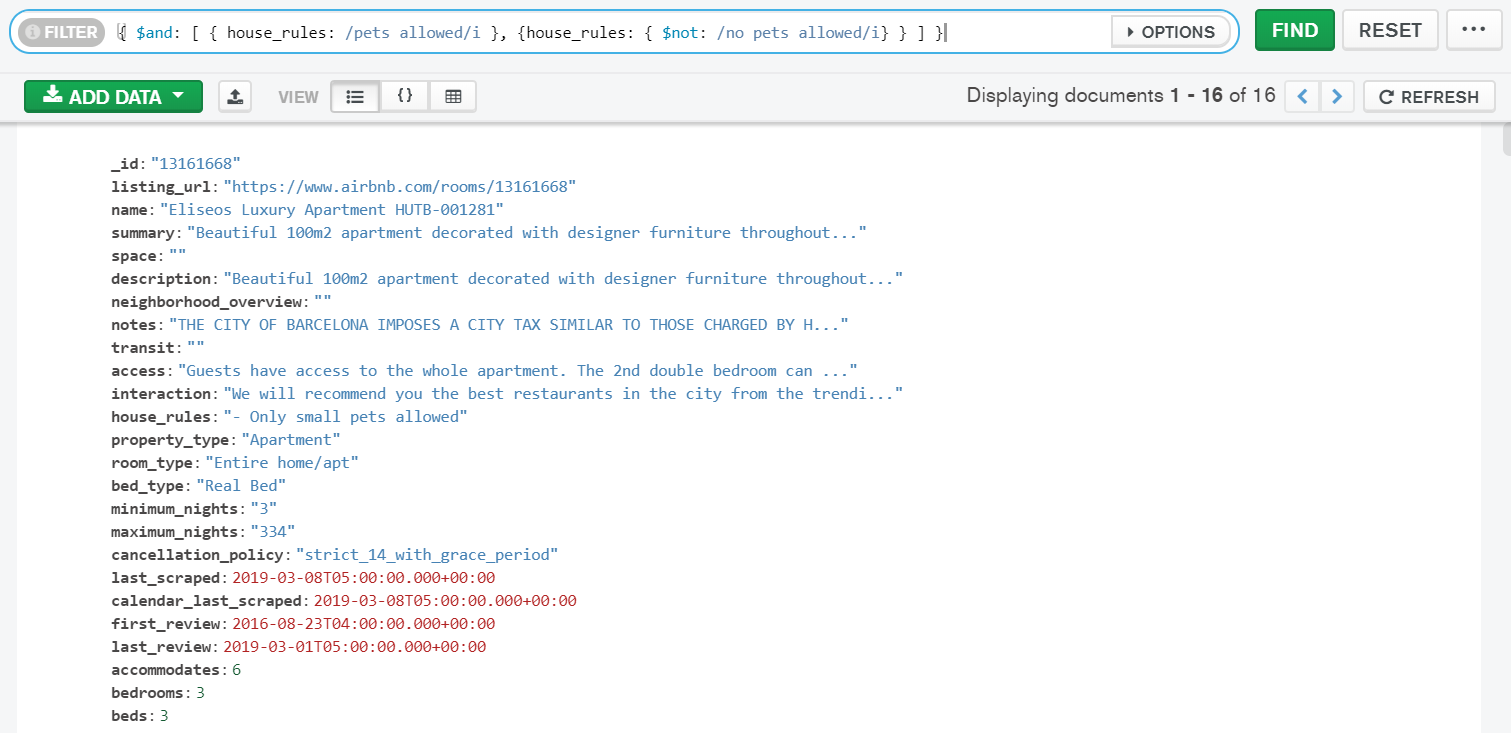
Propiedades que no permitan fumadores



Propiedades que no permitan fiestas ni fumadores.



No permiten mascotas



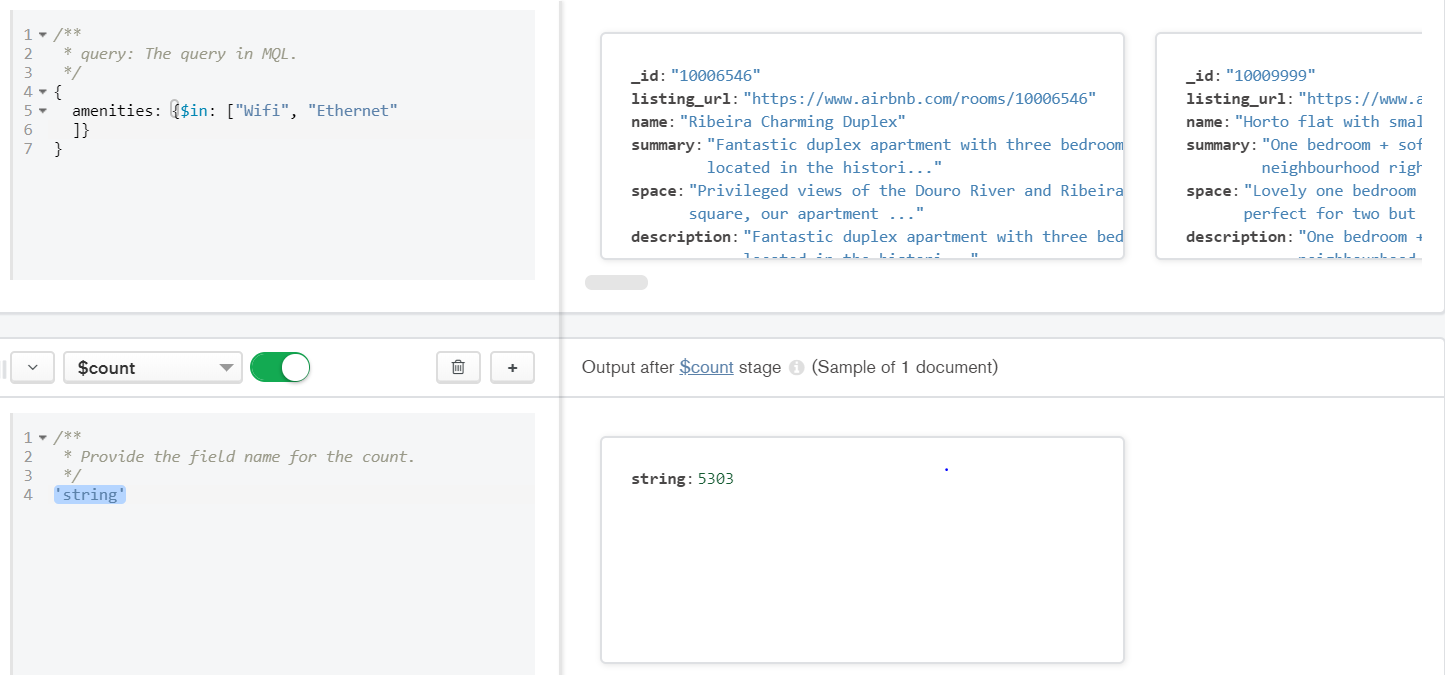
**RETO 2**

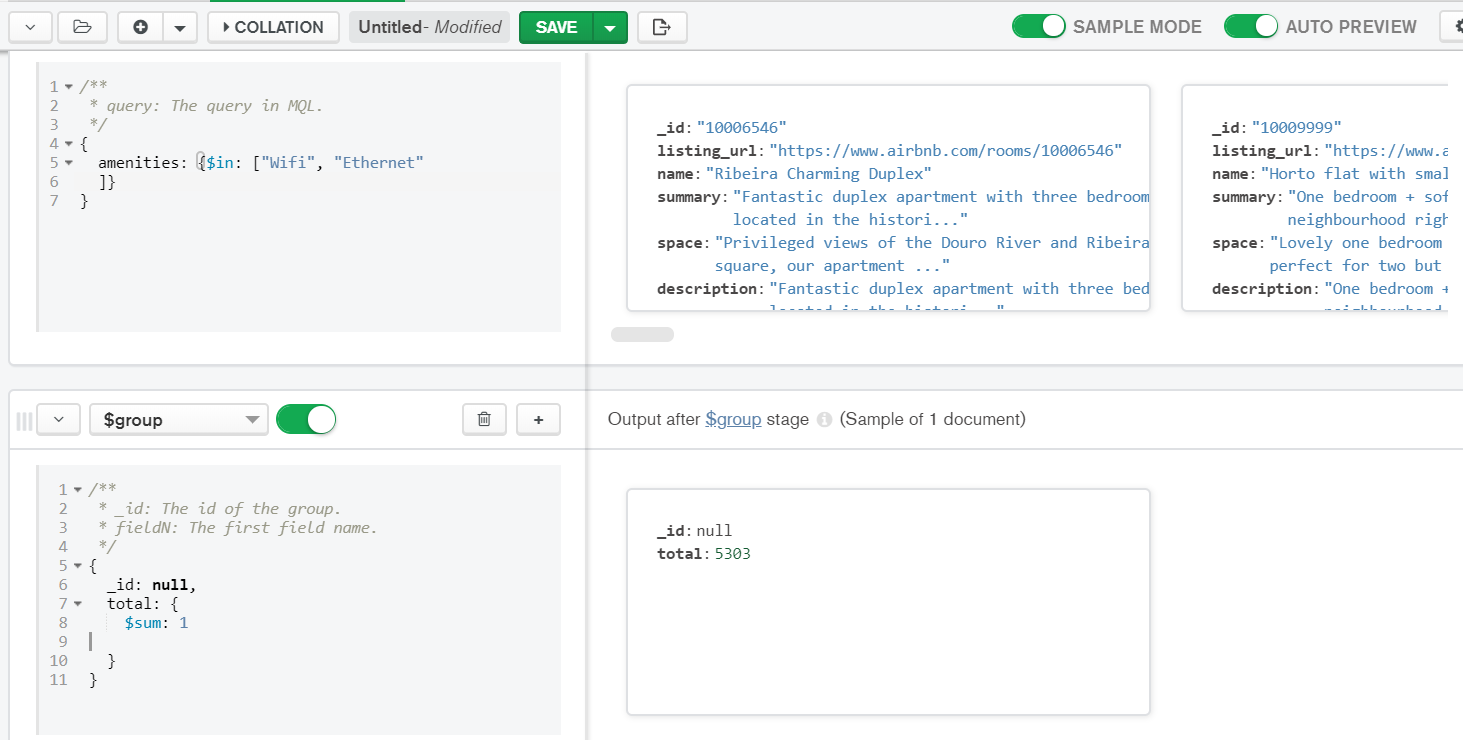
Filtro que permita obtener todas las publicaciones que tengan 50 o mas comentarios, que la valoracion sea mayor o igual a 80, que cuenten con conexion a internet y esten ubicada en Brasil

{ number\_of\_reviews: { $gte: 50 }, "review\_scores.review\_scores\_rating": { $gte: 80 }, amenities: { $in: [ /Ethernet/i ] }, "address.country": "Brazil" }

**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)





**PROYECTO 5**

[{$match: {

  number\_of\_reviews: {$gte: 50}

}}, {$match: {

  "review\_scores.review\_scores\_rating": { $gte: 80 }

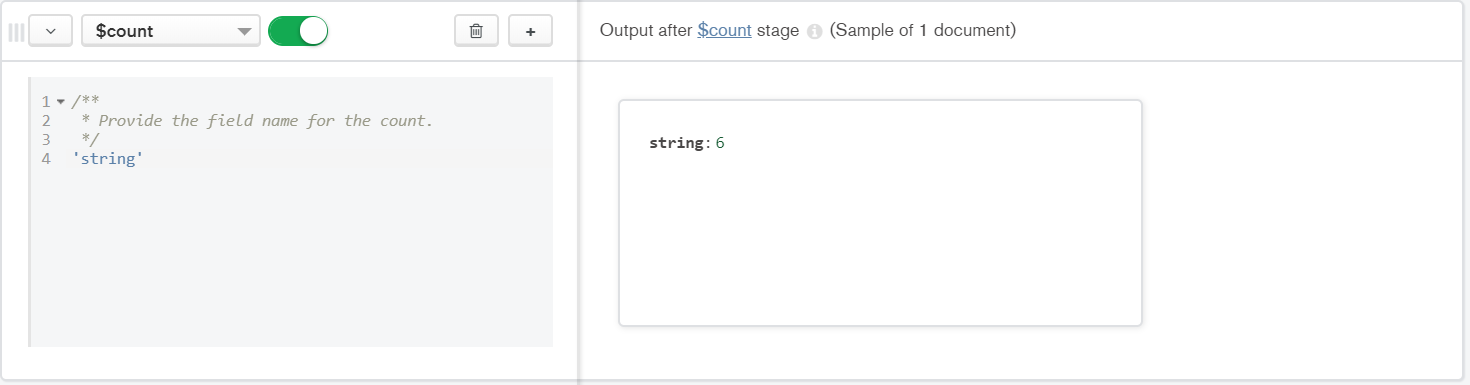
}}, {$match: {

  amenities: { $in: [ /Ethernet/i ] }

}}, {$match: {

  "address.country": "Brazil"

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



**20/Jul/2020 - 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",

  recamaras: {

    $sum: 1

  },

  total: {

    $sum: '$costo\_recamara'

  }

}}, {$addFields: {

  costo\_promedio: {

    $divide: [

      '$total',

      '$recamaras'

    ]

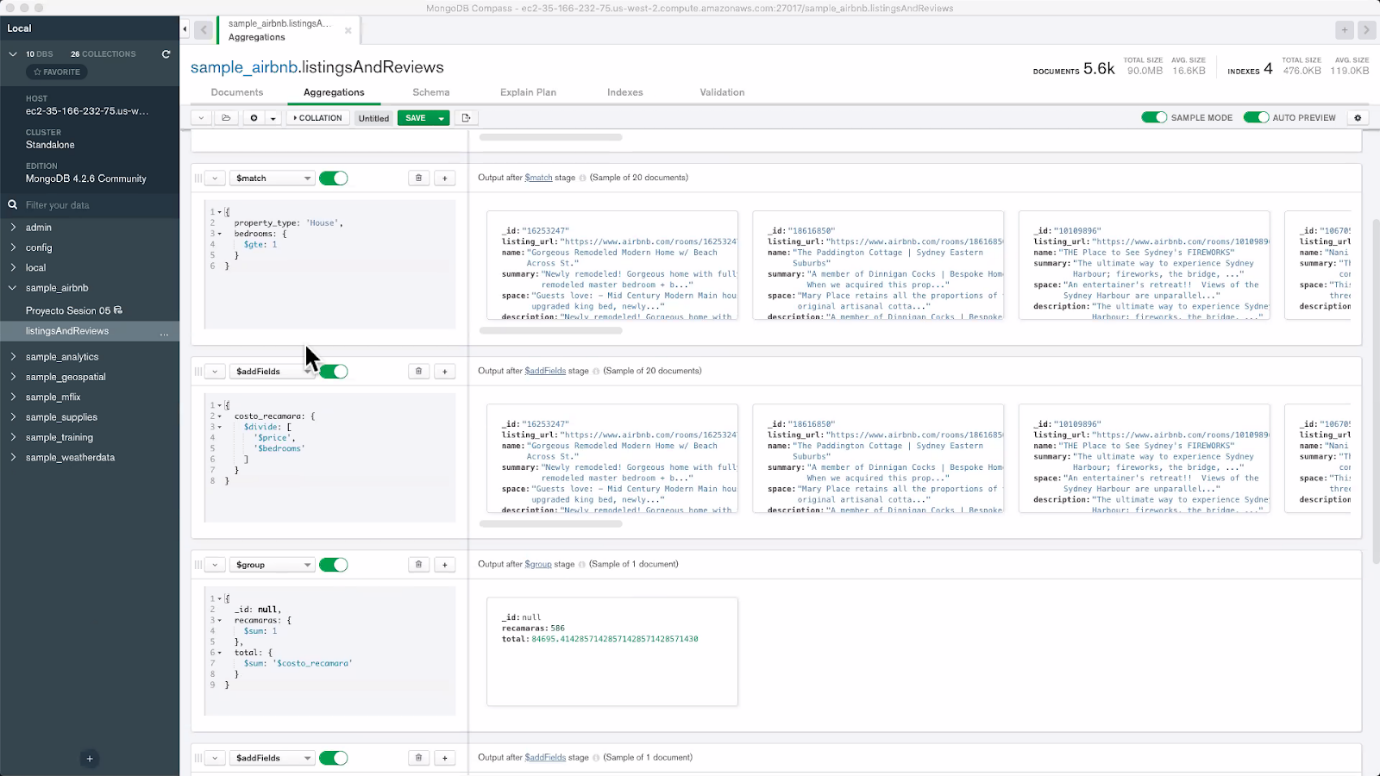
  }

}}, {$project: {

  costo\_promedio: 1,

  \_id: 0

}}]



**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: {

  correo\_obj: {

    $arrayElemAt: [

      '$correo',

      0

    ]

  }

}}, {$match: {

  correo\_obj: {

    $exists: true

  }

}}, {$addFields: {

  correo\_password: '$correo\_obj.password'

}}, {$addFields: {

  correo: '$correo\_obj.email'

}}, {$project: {

  correo\_password: 1,

  correo: 1,

  name: 1,

  \_id: 0

}}]

**RETO 3**

Crear vista

SAVE + Create view

$unwind

path: “$genre”

$unwind

path: “$countries”

**22/Jul/2020 - Sesión 7**

**RETO 1 y 2**

Crear bases de datos en Workbench

**PROYECTO 7**

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

Agregar los siguientes registros en formato CSV a la Colección movies  
4000,Avengers: Endgame (2019),Fantasy|Sci-Fi

4001,Glass (2019),Drama|Fantasy

Incluyendo nombres de columnas quedaría:  
id,titulo,genres

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

1. 4001,Glass (2019),Drama|Fantasy

[

  {

    "id": 4000,

    "Title": "Avengers: Endgame (2019)",

    “Genres”: "Fantasy|Sci-Fi"

  },

 {

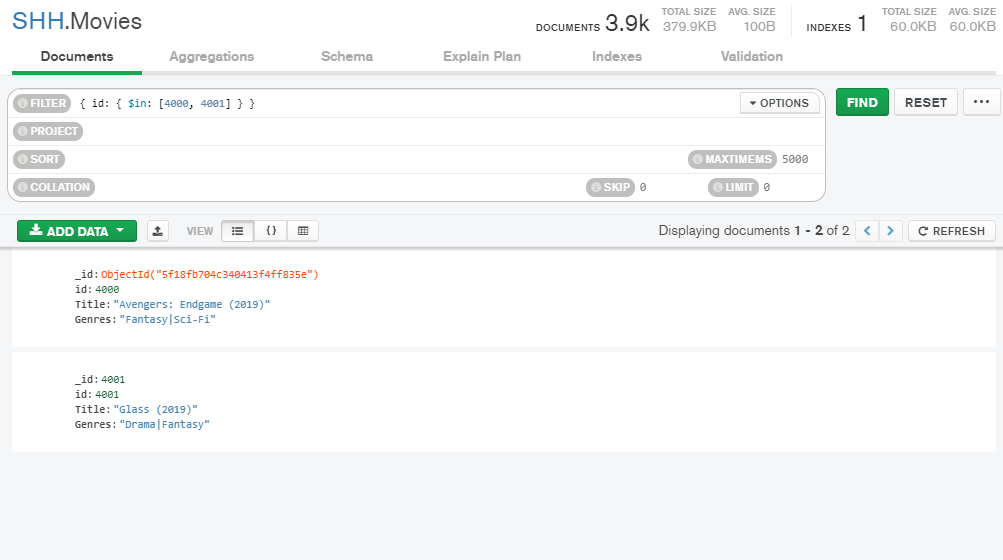
    "id": 4001,

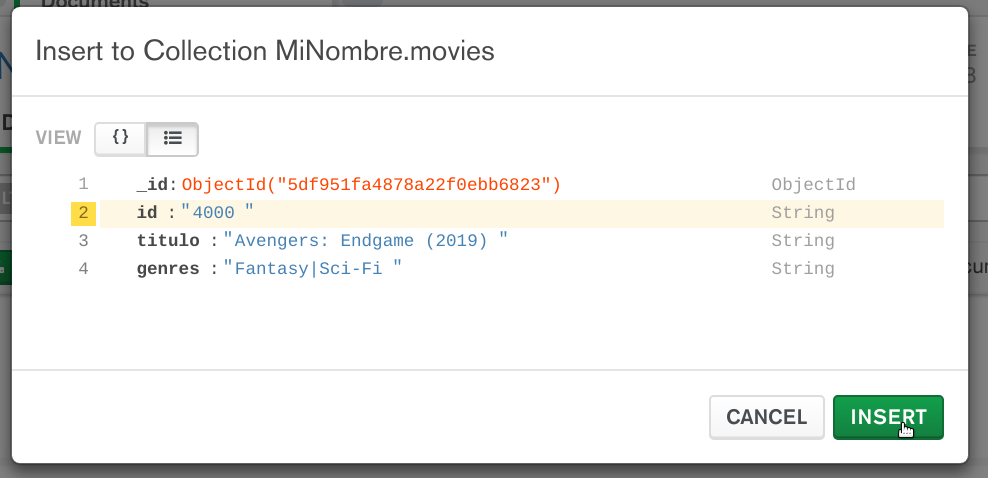
    "Title": "Glass (2019)",

    "Genres": "Drama|Fantasy"

 }

]





**27/Jul/2020 - Sesión 8**

**RETO 1**

**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.**

{

  $and:

  [

    { Latitude: { $gte: 19.41 } },

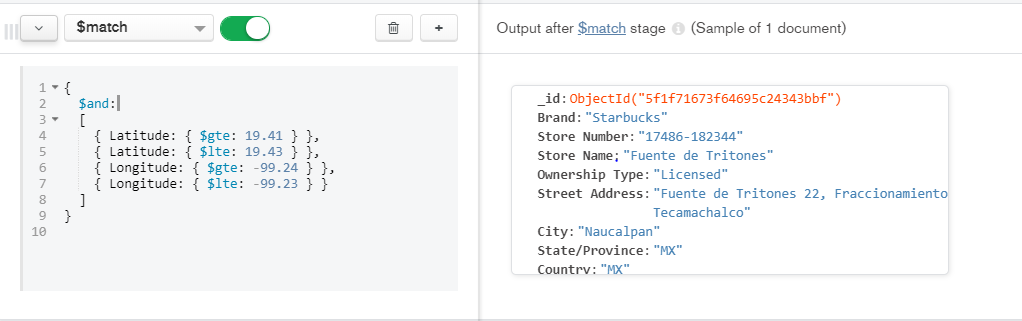
    { Latitude: { $lte: 19.43 } },

    { Longitude: { $gte: -99.24 } },

    { Longitude: { $lte: -99.23 } }

  ]

}

****

**RETO 2**

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

[{$group: {

  \_id: "$Country",

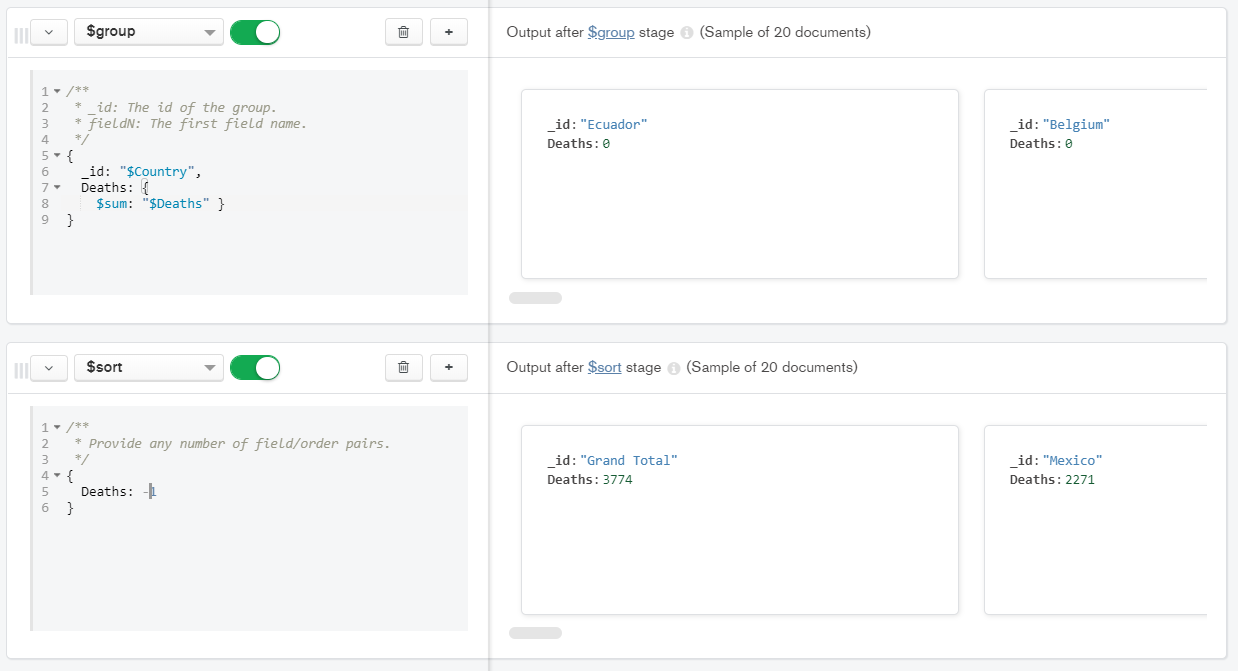
  Deaths: {

    $sum: "$Deaths" }

}}, {$sort: {

  Deaths: -1

}}]

****

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

[{$group: {

  \_id: "$Country",

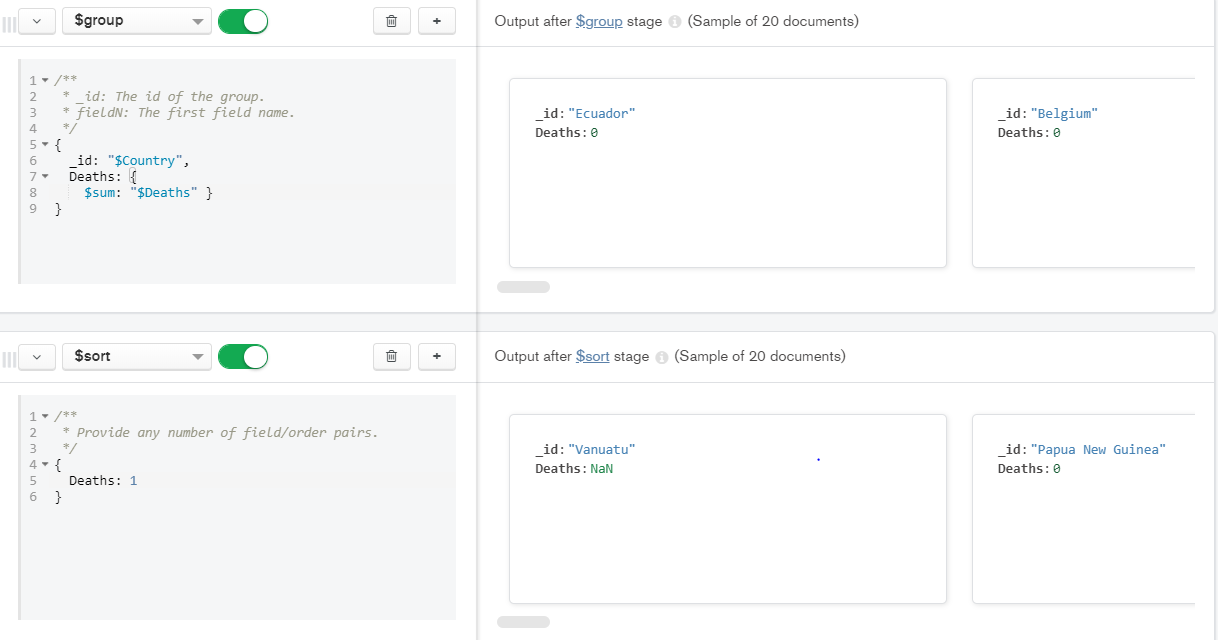
  Deaths: {

    $sum: "$Deaths" }

}}, {$sort: {

  Deaths: 1

}}]

****

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

[{$match: {

  Country: { $ne: "Grand Total" }

}}, {$group: {

  \_id: "$Country",

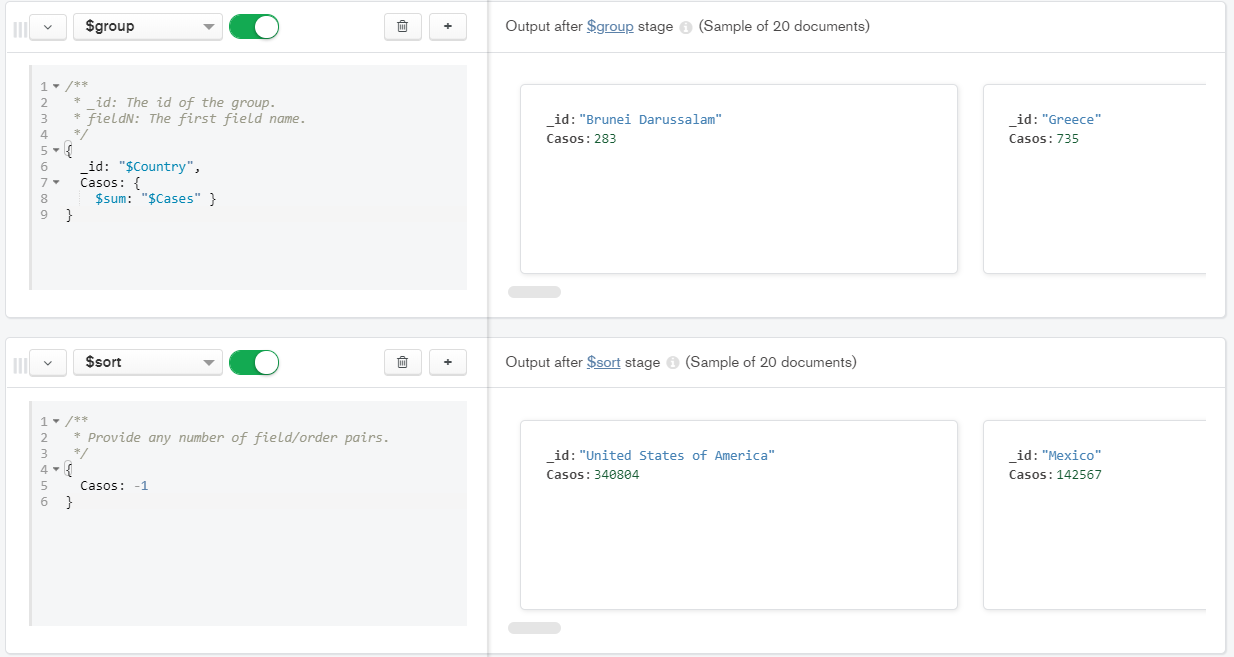
  Casos: {

    $sum: "$Cases" }

}}, {$sort: {

  Casos: -1

}}]



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

[{$match: {

  Country: { $ne: "Grand Total" }

}}, {$group: {

  \_id: "$Country",

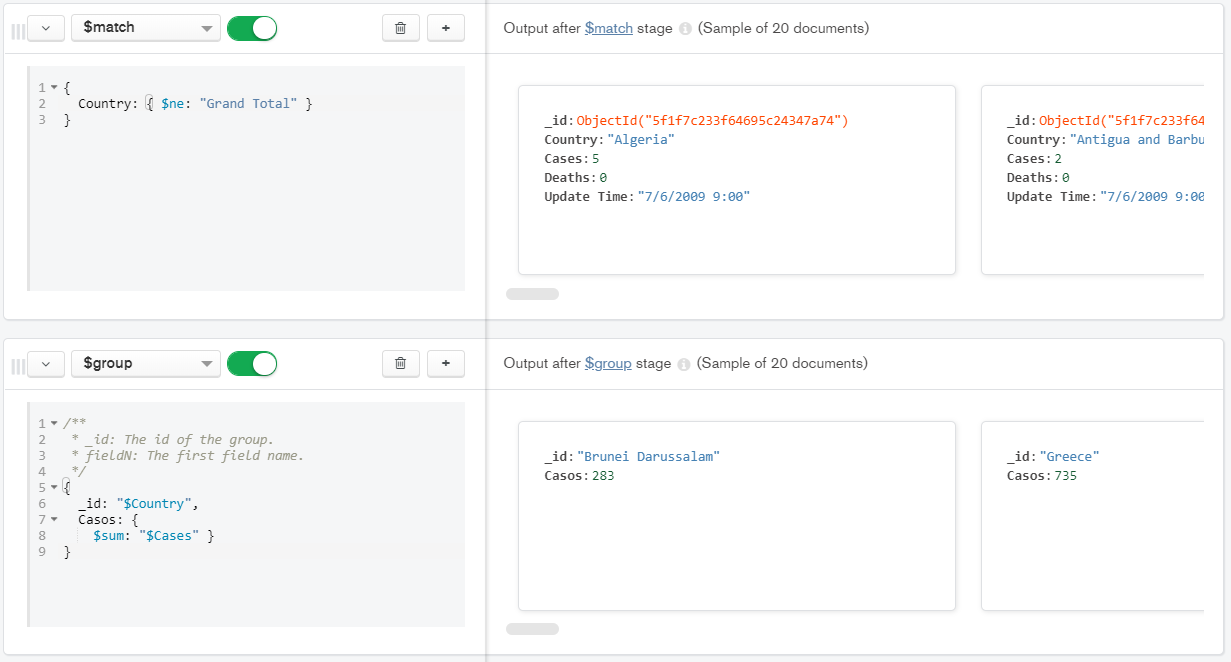
  Casos: {

    $sum: "$Cases" }

}}, {$sort: {

  Casos: 1

}}]

****

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

[{$match: {

  Country: { $ne: "Grand Total" }

}}, {$match: {

  Deaths: {$ne: NaN }

}}, {$group: {

  \_id: null,

  Promedio: {

    $avg: "$Deaths"

  }

}}]



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

[{$match: {

  Country: { $ne: "Grand Total" }

}}, {$match: {

  Cases: {$ne: NaN }

}}, {$group: {

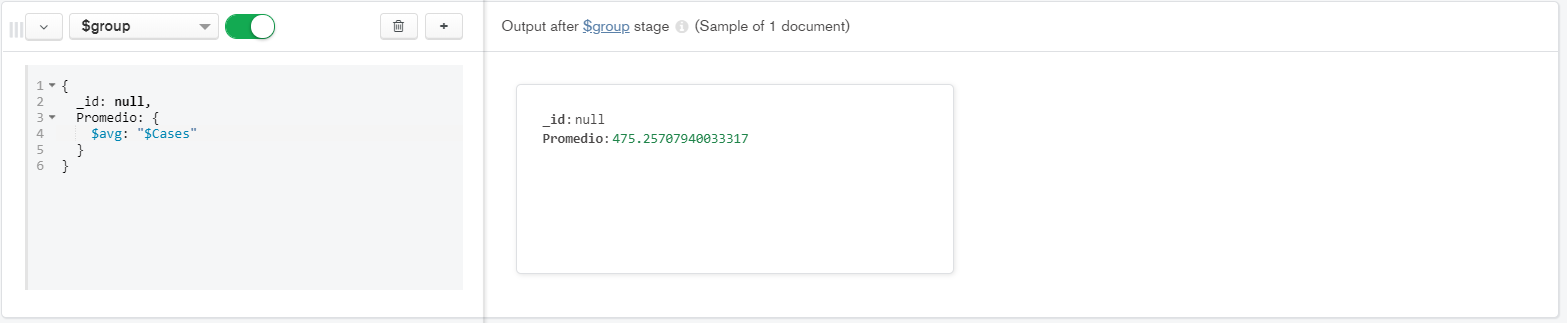
  \_id: null,

  Promedio: {

    $avg: "$Cases"

  }

}}]



* **Top 5 de países con más muertes**

[{$match: {

  Country: { $ne: "Gran Total" }

}}, {$match:

{

  Deaths: { $ne: NaN }

}}, {$group: {

  \_id: "$Country",

  Top5muertes: {

    $sum: "$Deaths"

  }

}}, {$sort: {

  Top5muertes: -1

}}, {$limit: 5}]

**RETO 3**

No lo pude hacer