**SESIÓN 01**

**RETO 1**

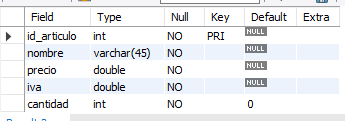
1. ***Usando la base de datos tienda, muestra la descripción de las tablas articulo, puesto y venta.***

use tienda;

show databases;

show tables;

describe articulo;

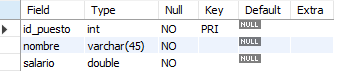


use tienda;

show databases;

show tables;

describe puesto;

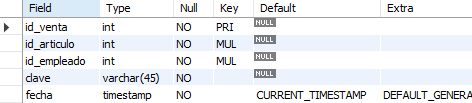


use tienda;

show databases;

show tables;

describe venta;



**RETO 2**

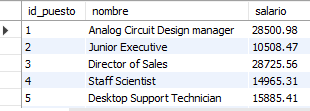
1. ***¿Cuál es el nombre de los empleados con el puesto 4?***

select \* from empleado where id\_puesto =4;



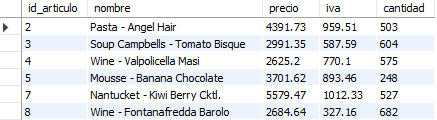
1. ***¿Qué puestos tienen un salario mayor a $10,000?***

select \* from puesto where salario > 1000;



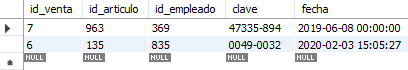
1. ***¿Qué artículos tienen un precio mayor a $1,000 y un iva mayor a 100?***

select \* from articulo where precio > 1000 and iva > 100;



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

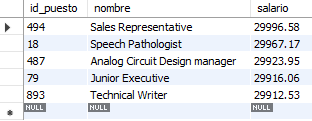
select \* from venta where id\_articulo in (135,963) and id\_empleado in (835,369);

******

**RETO 3**

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

select \* from puesto order by salario desc limit 5;

******

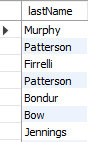
**PROYECTO**

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

use classicmodels;

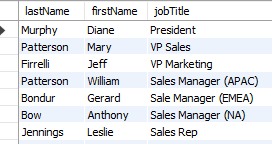
1. ***-- Dentro de la tabla employees, obtén el apellido de todos los empleados.***

select lastName from employees;



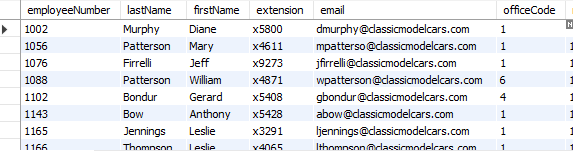
1. ***-- Dentro de la tabla employees, obtén el apellido, nombre y puesto de todos los empleados***

select lastName, firstName, jobTitle from employees;



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

select \* from employees;



1. ***-- 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";



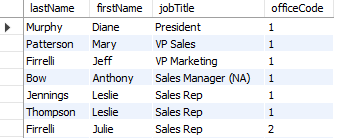
1. ***-- Dentro de la tabla employees, obtén el apellido, nombre, puesto y código de oficina de todos los empleados que tengan el puesto Sales Rep y código de oficina 1.***

select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;

**

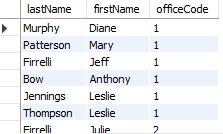
1. ***-- Dentro de la tabla employees, obtén el apellido, nombre, puesto y código de oficina de todos los empleados que tengan el puesto Sales Rep o código de oficina 1.***

select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" or officeCode = 1;

**

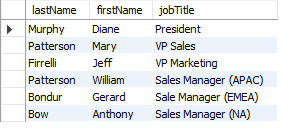
1. ***-- Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados que tenga código de oficina 1, 2 o 3.***

select lastName, firstName, officeCode from employees where officeCode in (1,2,3);



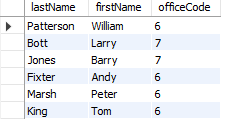
1. ***-- Dentro de la tabla employees, obten el apellido, nombre y puesto de todos los empleados que tengan un puesto distinto a Sales Rep.***

select lastName, firstName, jobTitle from employees where jobTitle != "Sales Rep";

**

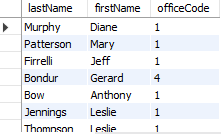
1. ***-- Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados cuyo código de oficina sea mayor a 5.***

select lastName, firstName, officeCode from employees where officeCode > 5;

**

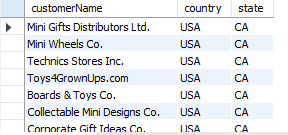
1. ***-- Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados cuyo cdigo de oficina sea menor o igual 4.***

select lastName, firstName, officeCode from employees where officeCode <= 4;



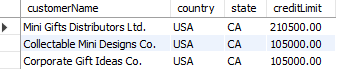
1. ***-- Dentro de la tabla customers, obtén el nombre, país y estado de todos los clientes cuyo país sea USA y cuyo estado sea CA.***

select customerName, country, state from customers where country = "USA" and state = "CA";

**

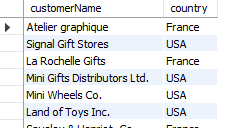
1. ***-- Dentro de la tabla customers, obtén el nombre, país, estado y límite de crédito de todos los clientes cuyo país sea, USA, cuyo estado sea CA y cuyo límite de crédito sea mayor a 100000.***

select customerName, country, state, creditLimit from customers where country = "USA" and state = "CA" and creditLimit > 100000;

**

1. ***-- 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");

**

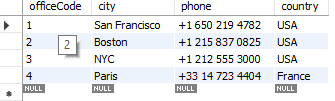
1. ***-- Dentro de la tabla customers, obtén el nombre, pas y límite de crédito de todos los clientes cuyo país sea USA o France y cuyo límite de crédito sea mayor a 100000. Para este ejercicio ten cuidado con los paréntesis.***

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

**

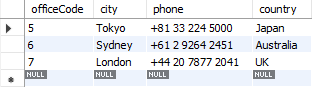
1. ***-- Dentro de la tabla offices, obtén el código de la oficina, ciudad, teléfono y país de aquellas oficinas que se encuentren en USA o France.***

select officeCode, city, phone, country from offices where country in ("USA","France");

**

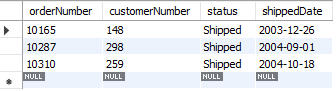
1. ***-- Dentro de la tabla offices, obtén el código de la oficina, ciudad, teléfono y país de aquellas oficinas que no se encuentren en USA o France.***

select officeCode, city, phone, country from offices where country != "USA" and country != "France";

**

1. ***-- 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);

**

1. ***-- 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;

**

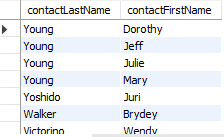
1. ***-- Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma descendente.***

select contactLastName, contactFirstName from customers order by contactLastName DESC;

**

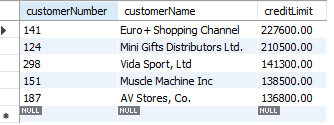
1. ***-- Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma descendente y luego por nombre de forma ascendente.***

select contactLastName, contactFirstName from customers order by contactLastName DESC, contactFirstName ASC;

******

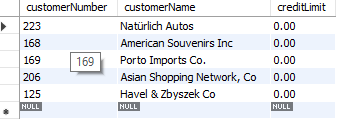
1. ***-- Dentro de la tabla customers, obtén el número de cliente, nombre de cliente y el límite de crédito de los cinco clientes con el límite de crédito más alto (top 5).***

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

******

1. ***-- Dentro de la tabla customers, obtén el número de cliente, nombre de cliente y el límite de crédito de los cinco clientes con el límite de crédito más bajo.***

select customerNumber, customerName, creditLimit from customers order by creditLimit ASC limit 5;



**SESIÓN 02**

**RETO 1**

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

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

select \* from articulo where nombre like "%Pasta%";

******

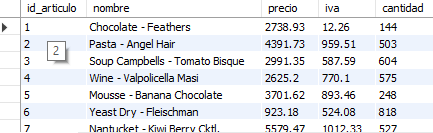
1. ***¿Qué artículos incluyen la palabra Cannelloni en su nombre?***

select \* from articulo where nombre like "%Cannelloni%";

******

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

select \* from articulo where nombre like "%-%";

******

**RETO 2**

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

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

select avg(salario) from puesto;



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

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

******

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

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

******

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

select sum(salario) from puesto where id\_puesto >= 995;

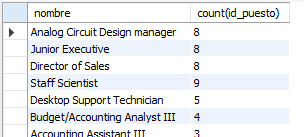


**RETO 3**

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

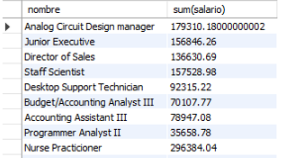
1. ***¿Cuántos registros hay por cada uno de los puestos?***

select nombre, count(id\_puesto) from puesto group by nombre;

******

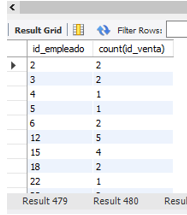
1. ***¿Cuánto dinero se paga en total por puesto?***

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

******

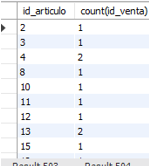
1. ***¿Cuál es el número total de ventas por vendedor?***

SELECT id\_empleado, COUNT(id\_venta) FROM venta GROUP BY id\_empleado;

******

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

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

******

**RETO 4**

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

1. ***¿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");

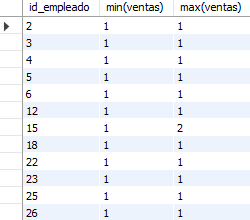


1. ***¿Cuál es la cantidad mínima y máxima de ventas de cada empleado?***

SELECT id\_empleado,min(ventas),max(ventas) FROM

(select clave,id\_empleado,COUNT(\*) AS ventas FROM venta GROUP BY clave,id\_empleado) AS vent

GROUP BY id\_empleado;

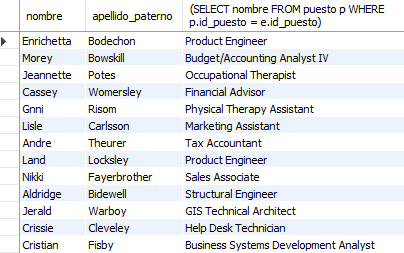
******

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

SELECT nombre,apellido\_paterno,

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

FROM empleado AS e;



**PROYECTO**

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%”;

******

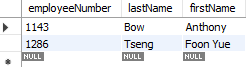
1. ***Dentro de la tabla employees, obten el número de empleado, apellido y nombre de todos los empleados cuyo nombre termina con on.***

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

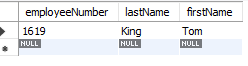
******

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%";

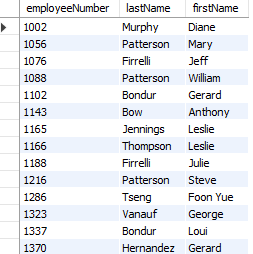
******

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.***
2. SELECT employeeNumber, lastName, firstName FROM employees WHERE firstName LIKE "t\_m";

******

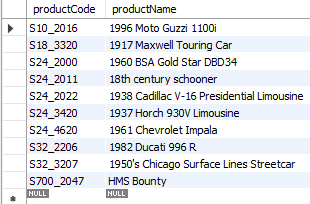
1. ***Dentro de la tabla employees, obten el número de empleado, apellido y nombre de todos los empleados cuyo nombre no inicia con B.***

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

******

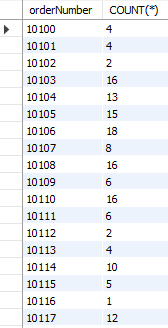
1. ***Dentro de la tabla products, obten el código de producto y nombre de los productos cuyo código incluye la cadena \_20.***

SELECT productCode, productName FROM products WHERE productCode LIKE "%\_20%";

******

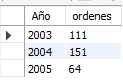
1. ***Dentro de la tabla orderdetails, obten el total de cada orden.***

SELECT orderNumber,COUNT(\*) FROM orderdetails GROUP BY orderNumber;

******

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

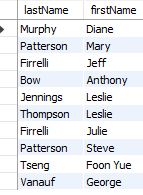
SELECT YEAR(orderDate) AS Año,COUNT(\*) AS ordenes FROM orders GROUP BY year(orderDate);



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

SELECT lastName, firstName 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.***

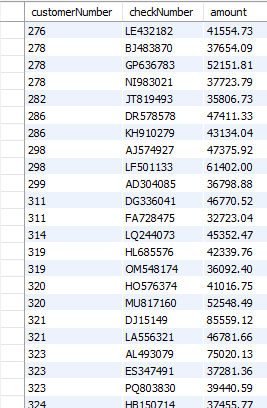
SELECT customerNumber,checkNumber,amount FROM payments ORDER BY amount DESC LIMIT 1;

******

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 customerName FROM customers WHERE customerNumber NOT IN

(SELECT customerNumber FROM orders);



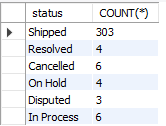
1. ***Obten el máximo, mínimo y promedio del número de productos en las órdenes de venta.***

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;

******

**SESIÓN 03**

**RETO 1**

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

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

SELECT clave,nombre,apellido\_paterno FROM venta AS v

JOIN empleado AS e

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

ORDER BY clave;

******

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

SELECT clave, nombre FROM venta AS v

JOIN articulo AS a

ON v.id\_articulo = a.id\_articulo

ORDER BY clave;

******

1. ***¿Cuál es el total de cada venta?***

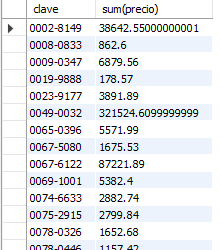
SELECT clave,sum(precio) FROM venta AS v

JOIN articulo AS a

ON v.id\_articulo = a.id\_articulo

GROUP BY clave

ORDER BY clave;



**RETO 2**

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

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

CREATE VIEW ycpuesto AS

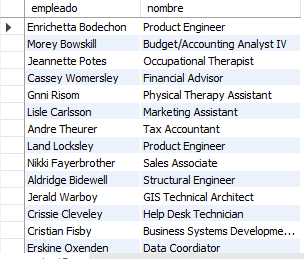
(SELECT CONCAT(e.nombre," ",e.apellido\_paterno) AS empleado,p.nombre

FROM empleado AS e

JOIN puesto AS p

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

SELECT \* FROM ycpuesto;

******

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

CREATE VIEW ycempleado AS

(SELECT CONCAT(e.nombre," ",e.apellido\_paterno) AS empleado,a.nombre

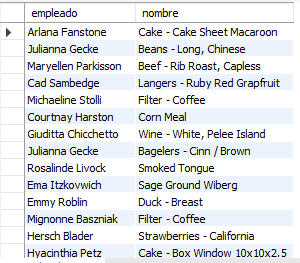
FROM venta AS v

JOIN empleado AS e

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

JOIN articulo AS a

ON v.id\_articulo = a.id\_articulo);

******

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

CREATE VIEW ycventa AS

(SELECT p.nombre,COUNT(v.id\_venta) AS ventas FROM venta AS v

JOIN empleado AS e

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

JOIN puesto AS p

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

GROUP BY p.nombre

ORDER BY ventas DESC

LIMIT 1);

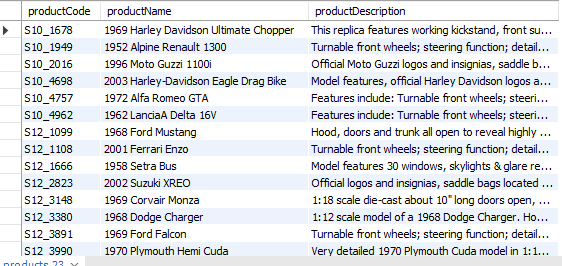
******

**PROYECTO**

***Para estas consultas usa RIGHT JOIN***

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

SELECT productCode,productName,productDescription FROM products;

**

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

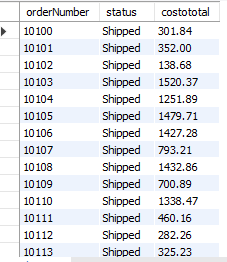
SELECT o.orderNumber,o.status,SUM(od.priceEach) AS costototal

FROM orders AS o

RIGHT JOIN orderdetails AS od

ON o.orderNumber = od.orderNumber

GROUP BY o.orderNumber;



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

SELECT o.orderNumber,o.orderDate,od.orderLineNumber,p.productName,od.quantityOrdered,od.priceEach

FROM orders AS o

RIGHT JOIN orderdetails AS od

ON o.orderNumber = od.orderNumber

RIGHT JOIN products AS p

ON od.productCode = p.productCode

ORDER BY o.orderNumber;



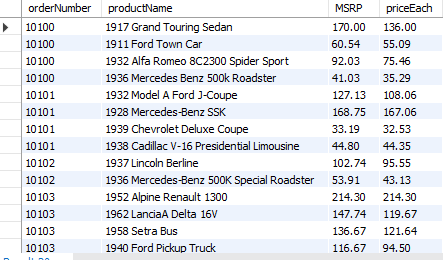
1. ***Obtén el número de orden, nombre del producto, el precio sugerido de fábrica (msrp) y precio de cada pieza.***

SELECT od.orderNumber,p.productName,p.MSRP,od.priceEach FROM orderdetails AS od

JOIN products AS p

ON od.productCode = p.productCode

ORDER BY od.orderNumber;



***Para estas consultas usa LEFT JOIN***

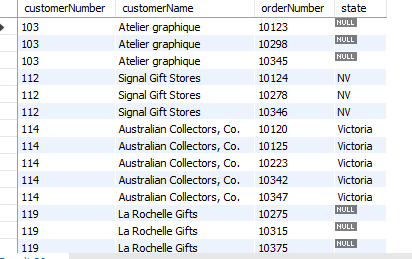
1. ***Obtén el número de cliente, nombre de cliente, número de orden y estado de cada cliente.***

SELECT c.customerNumber,c.customerName,o.orderNumber,c.state FROM customers AS c

LEFT JOIN orders AS o

ON c.customerNumber = o.customerNumber

ORDER BY c.customerNumber;

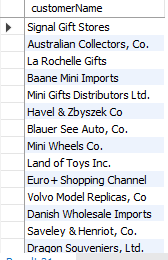
******

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

SELECT c.customerName FROM customers AS c

LEFT JOIN orders AS o

ON c.customerNumber != o.customerNumber;

****

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

SELECT e.lastName,e.firstName,c.customerName,py.checkNumber,py.amount AS total

FROM employees AS e

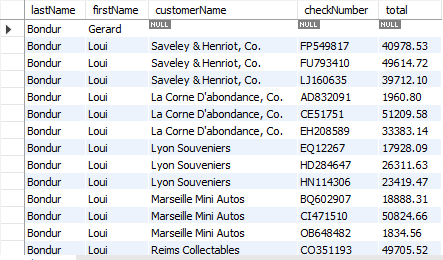
LEFT JOIN customers AS c

ON e.employeeNumber = c.salesRepEmployeeNumber

LEFT JOIN payments AS py

ON c.customerNumber = py.customerNumber

ORDER BY e.lastName;



***Para estas consultas usa RIGHT JOIN***

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

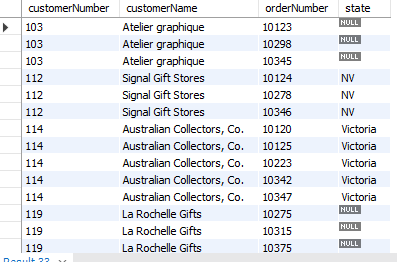
***5.*** SELECT c.customerNumber,c.customerName,o.orderNumber,c.state

FROM customers AS c

RIGHT JOIN orders AS o

ON c.customerNumber = o.customerNumber

ORDER BY c.customerNumber;

******

***6.*** SELECT c.customerName FROM orders AS o

RIGHT JOIN customers AS c

ON o.customerNumber != c.customerName;



***7.*** SELECT e.lastName,e.firstName,c.customerName,py.checkNumber,py.amount AS total

FROM payments AS py

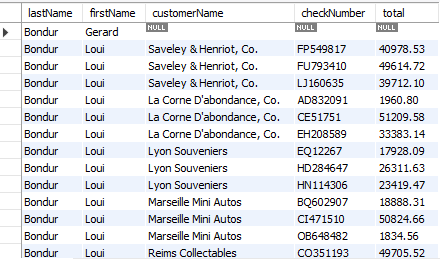
RIGHT JOIN customers AS c

ON py.customerNumber = c.customerNumber

RIGHT JOIN employees AS e

ON c.salesRepEmployeeNumber = e.employeeNumber

ORDER BY e.lastName;



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

CREATE VIEW ycestatus AS

(SELECT o.orderNumber,o.status,SUM(od.priceEach) AS costototal

FROM orders AS o

RIGHT JOIN orderdetails AS od

ON o.orderNumber = od.orderNumber

GROUP BY o.orderNumber);



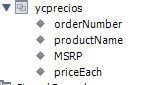
CREATE VIEW ycprecios AS

(SELECT od.orderNumber,p.productName,p.MSRP,od.priceEach FROM orderdetails AS od

JOIN products AS p

ON od.productCode = p.productCode

ORDER BY od.orderNumber);

******

CREATE VIEW ycsalesrep AS

(SELECT e.lastName,e.firstName,c.customerName,py.checkNumber,py.amount AS total

FROM payments AS py

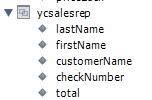
RIGHT JOIN customers AS c

ON py.customerNumber = c.customerNumber

RIGHT JOIN employees AS e

ON c.salesRepEmployeeNumber = e.employeeNumber

ORDER BY e.lastName);



**SESIÓN 04**

**RETO 1**

***Usando la base de datos sample\_mflix, proyecta los datos que se solicitan.***

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

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

****

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

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

****

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

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

******

**RETO 2**

***Usando la base de datos sample\_mflix, agrega proyeccciones, filtros, ordenamientos y límites que permitan contestar las siguientes preguntas.***

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

{name:"Greg Powell"}

******

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

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

******

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

{num\_mflix\_comments: 1}

{num\_mflix\_comments:-1}

Limit: 1

******

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

{title: 1}

{num\_mflix\_comments: -1}

Limit:5

******

**PROYECTO**

***Las consultas se realizarán sobre la base sample\_training.***

***Todas las consultas que realices deberás mantenerlas dentro del MongoDB Compass. Para hacer esto, da clic en el botón con los puntos ··· y en Toogle Query History. Busca la última consulta y agregala a favoritos presionando el íncono con la estrella ⭐.***

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

{

project: {

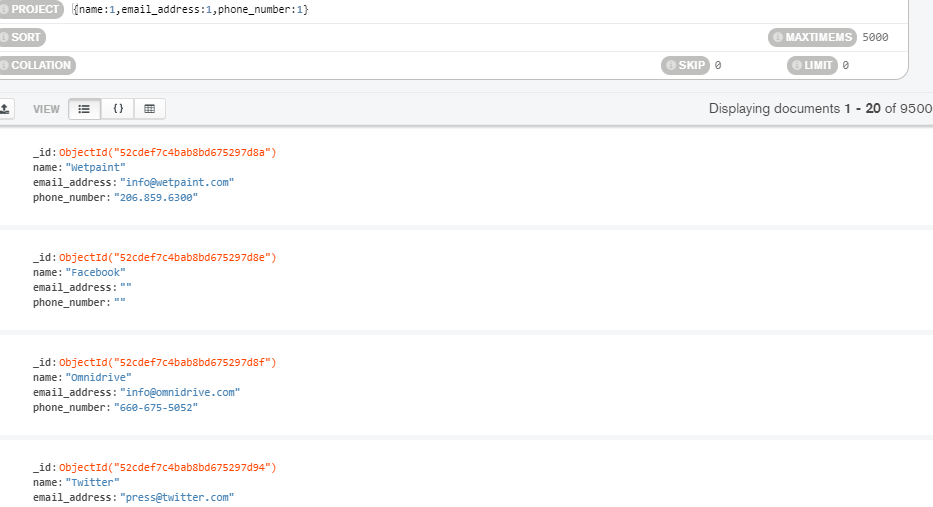
name: 1,

email\_address: 1,

phone\_number: 1

}

}

******

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

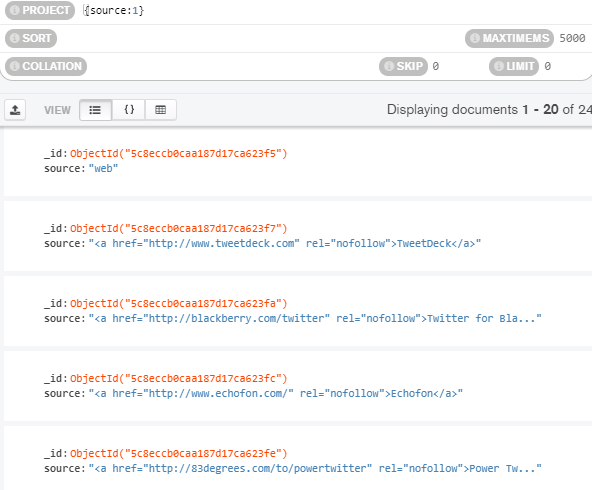
{

project: {

source: 1

}

}

******

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

{

filter: {

founded\_month: 10

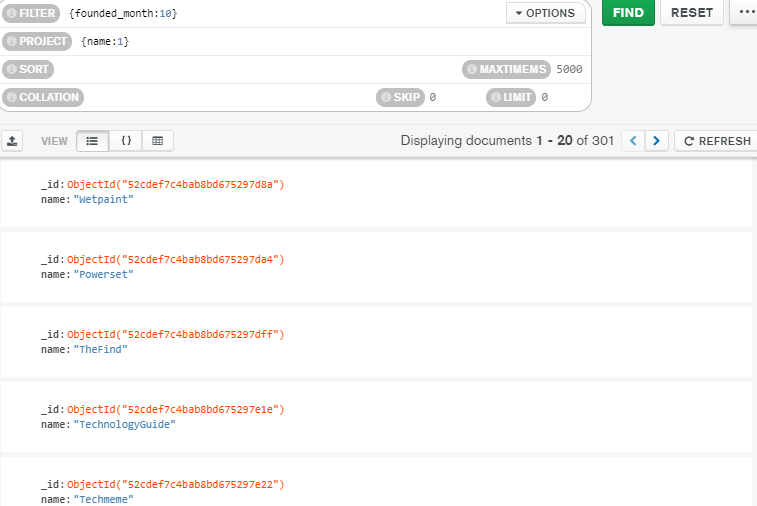
},

project: {

name: 1

}

}

******

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

{

filter: {

founded\_year: 2008

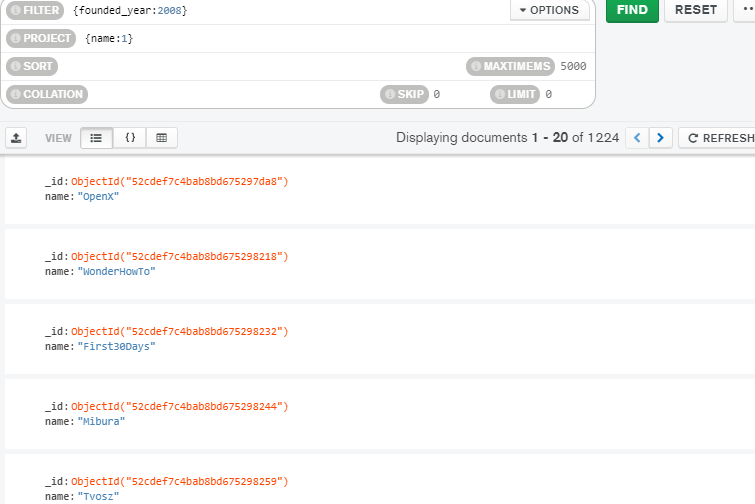
},

project: {

name: 1

}

}

******

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

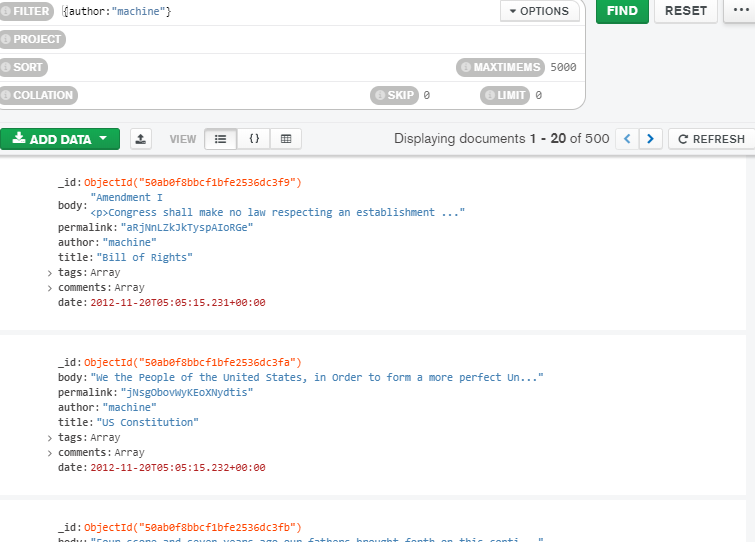
{

filter: {

author: 'machine'

}

}

******

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

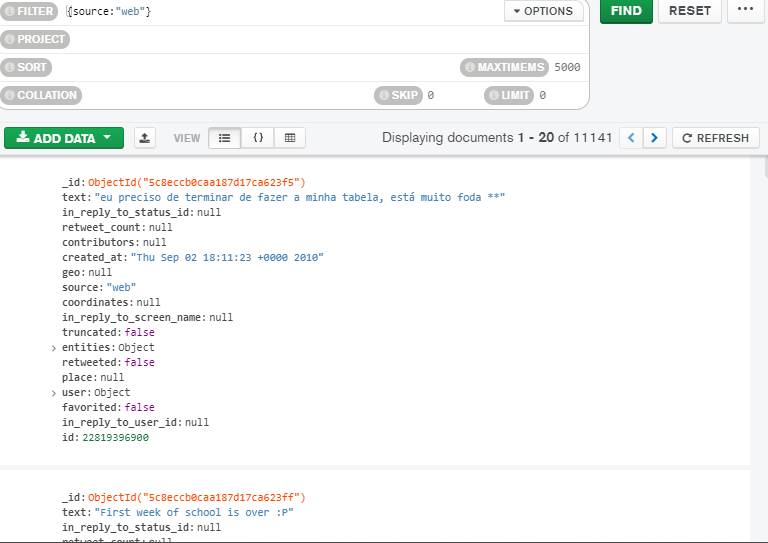
{

filter: {

source: 'web'

}

}

******

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

{

filter: {

$and: [

{

founded\_month: 10

},

{

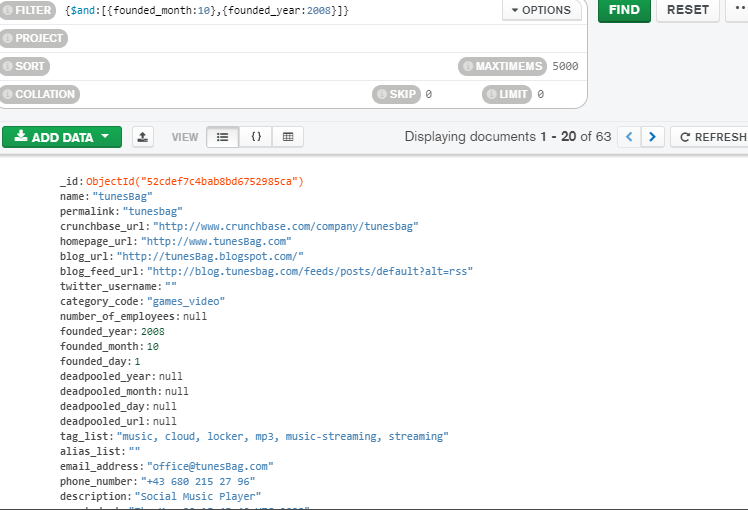
founded\_year: 2008

}

]

}

}

******

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

{

filter: {

number\_of\_employees: {

$gt: 50

}

},

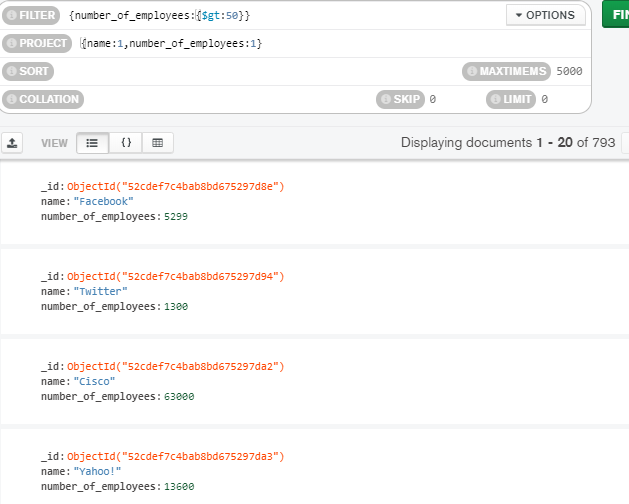
project: {

name: 1,

number\_of\_employees: 1

}

}

******

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

{

filter: {

$and: [

{

comments: {

$gte: 10

}

},

{

comments: {

$lte: 30

}

}

]

},

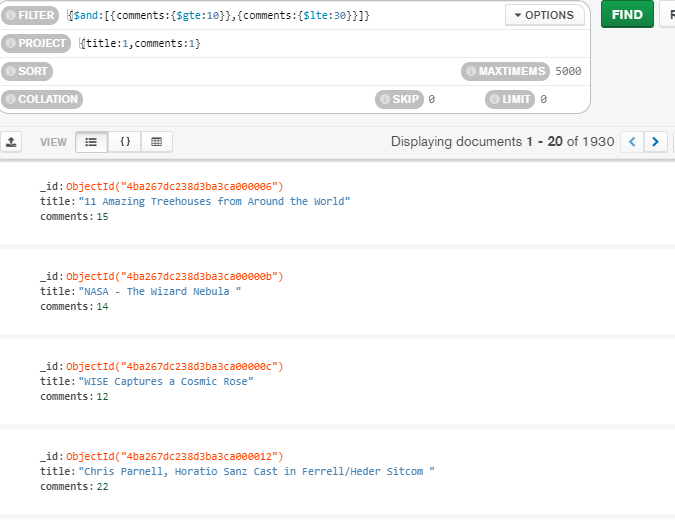
project: {

title: 1,

comments: 1

}

}

******

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

},

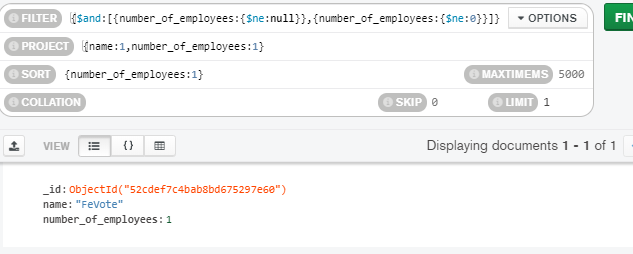
sort: {

number\_of\_employees: 1

},

limit: 1

}

******

1. ***Obtén la empresa con el mayor número de empleados.***

{

project: {

name: 1,

number\_of\_employees: 1

},

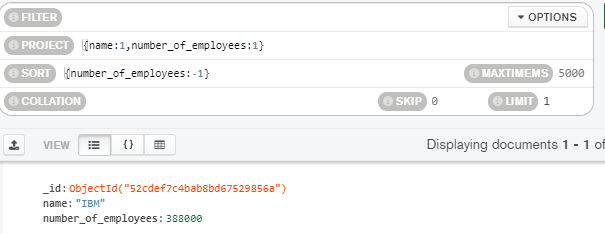
sort: {

number\_of\_employees: -1

},

limit: 1

}

******

1. ***Obtén la historia más comentada.***

{

project: {

title: 1,

comments: 1

},

sort: {

comments: -1

},

limit: 1

}

******

1. ***Obtén la historia menos comentada.***

{

project: {

title: 1,

comments: 1

},

sort: {

comments: 1

},

limit: 1

}

******

**SESIÓN 05**

**RETO 1**

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

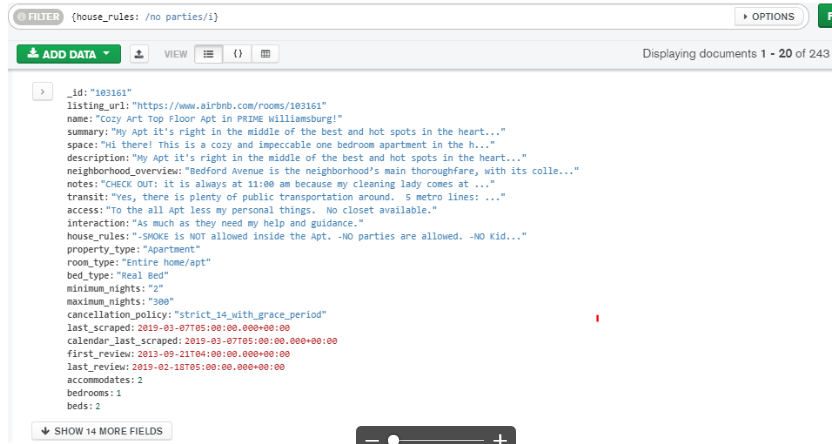
{

filter: {

house\_rules: RegExp('no parties', i)

}

}

******

1. ***Propiedades que admitan mascotas.***

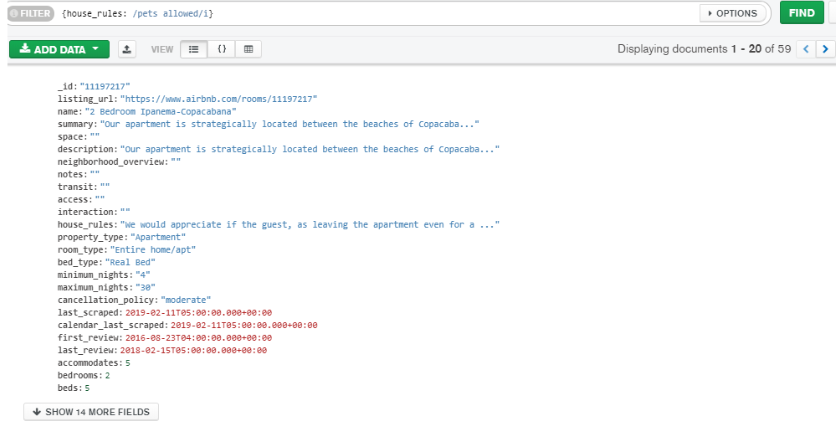
{

filter: {

house\_rules: RegExp('pets allowed', i)

}

}

******

1. ***Propiedades que no permitan fumadores.***

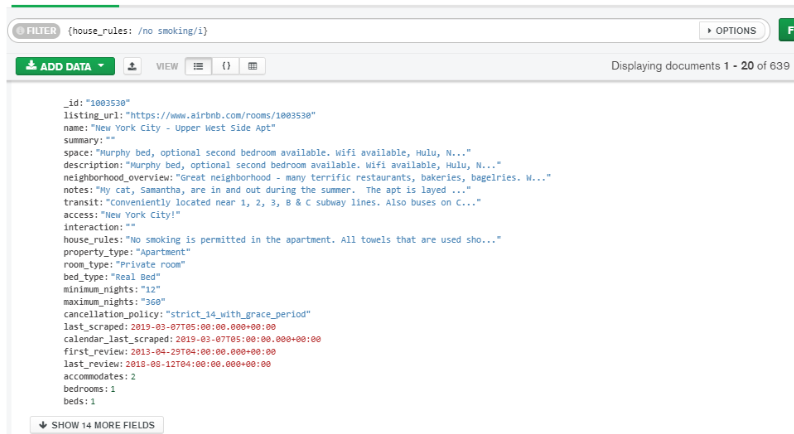
{

filter: {

house\_rules: RegExp('no smoking', i)

}

}

******

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

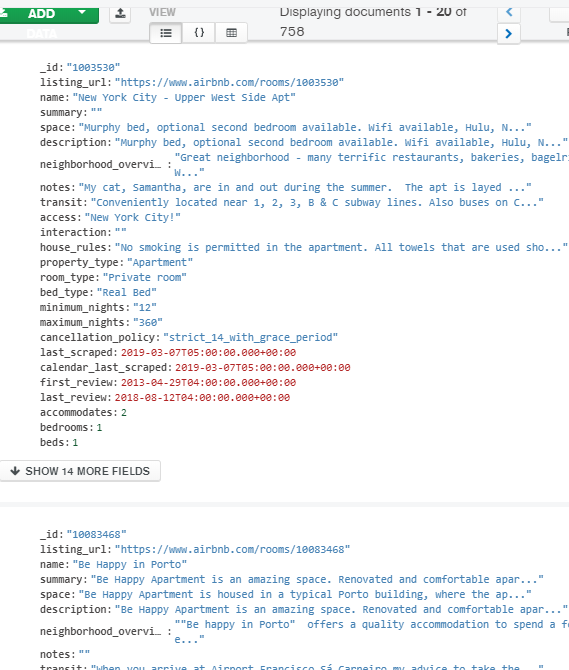
{

filter: {

house\_rules: RegExp('no smoking|no parties', i)

}

}



**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')

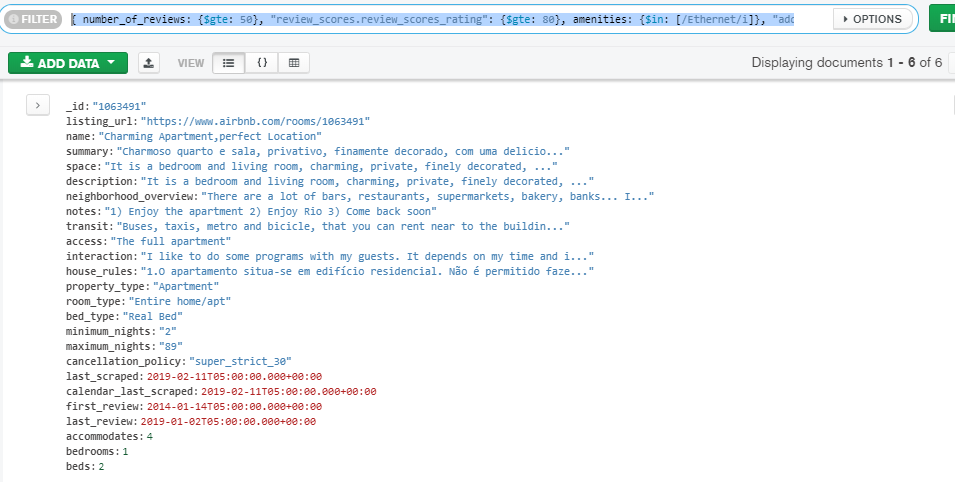
]

},

'address.country\_code': 'BR'

}

}

****

**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"]}

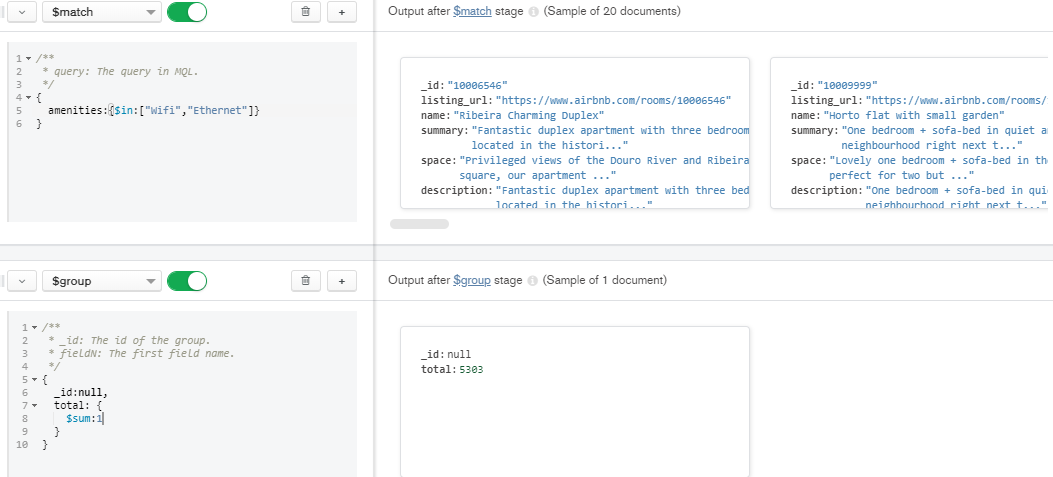
}}, {$group: {

\_id: null,

total:{$sum:1

}

}}]

****

**PROYECTO**

***Para este proyecto deberás practicar en el uso de agregaciones, pues serán usadas durante la siguiente sesión.***

***La base de datos y colección que debes usar es sample\_airbnb.listingsAndReviews.***

***El proyecto consiste en obtener todas las publicaciones que tengan 50 o más comentarios, que la valoración sea mayor o igual a 80, que cuenten con conexión a Internet vía cable y estén ubicadas en Brazil.***

[{$match: {

number\_of\_reviews:{$gte:50}

}}, {$match: {

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

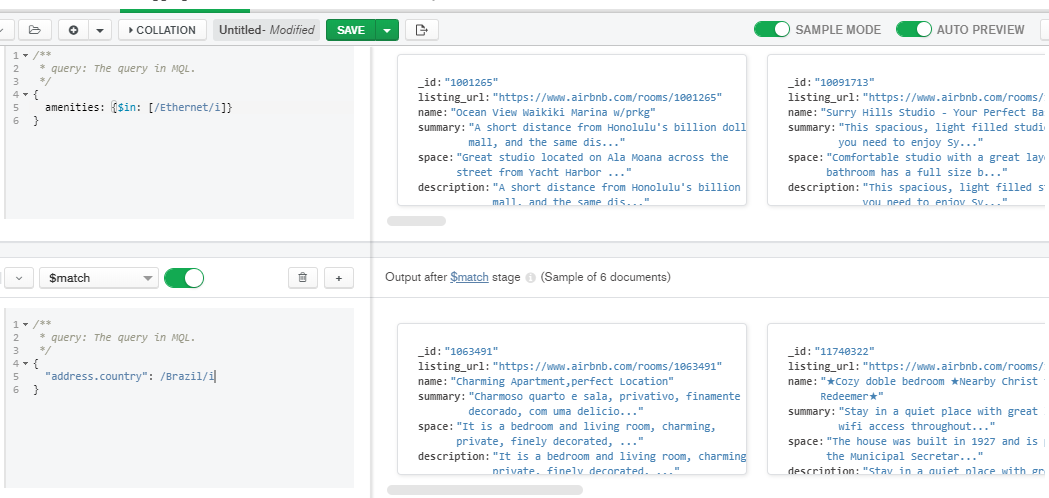
}}, {$match: {

amenities: {$in: [/Ethernet/i]}

}}, {$match: {

"address.country": /Brazil/i

}}]



**SESIÓN 06**

**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'

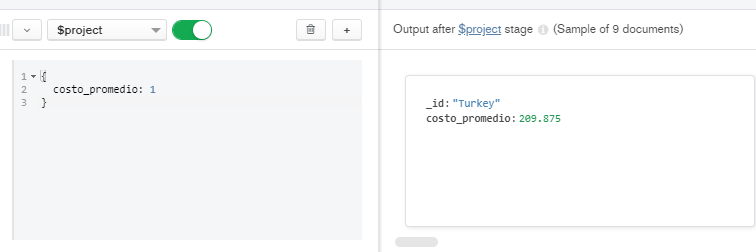
]

}

}}, {$project: {

costo\_promedio: 1

}}]



**RETO 2**

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

***NO CIERES ESTE*** *PIPELINE PUES LO USAREMOS MÁS ADELANTE*

[{$lookup: {

from: 'users',

localField: 'name',

foreignField: 'name',

as: 'usuario'

}}, {$addFields: {

usuario\_objeto:{$arrayElemAt:["$usuario",0]}

}}, {$addFields: {

usuario\_password:"$usuario\_objeto.password"

}}, {$project: {

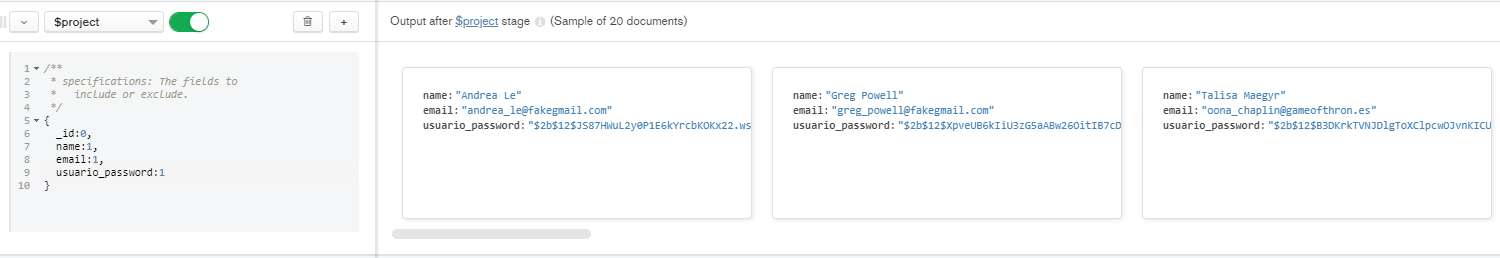
\_id:0,

name:1,

email:1,

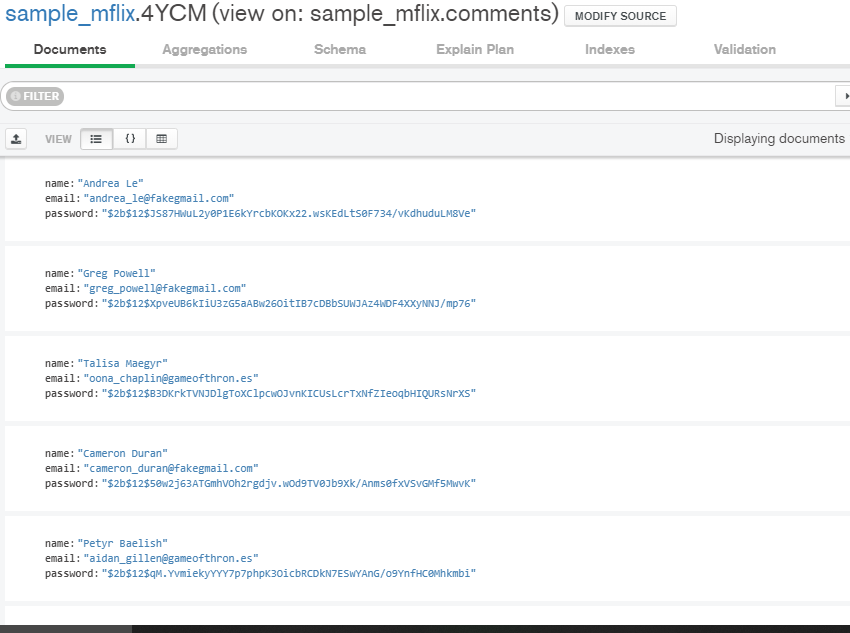
usuario\_password:1

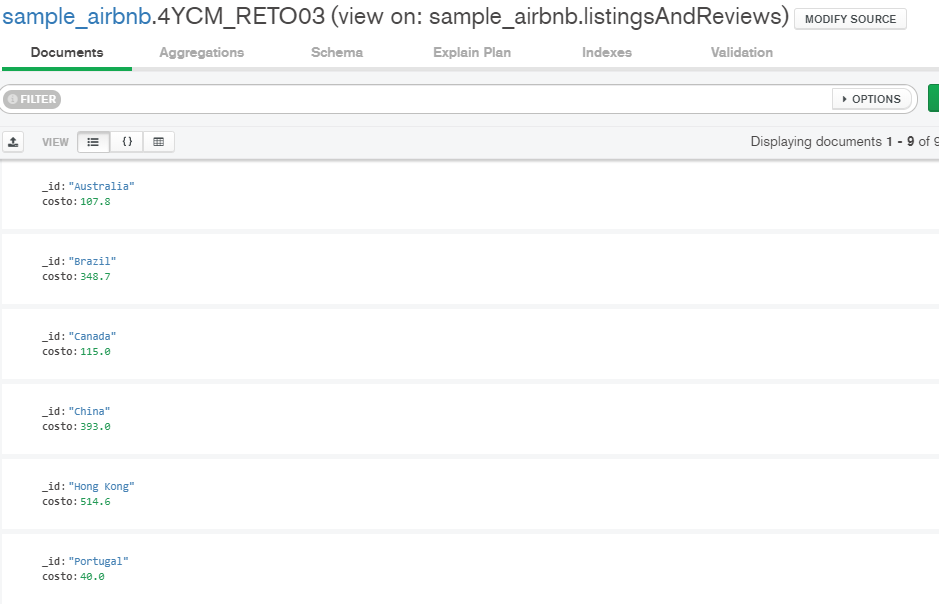
}}]

**

**RETO 3**

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





**PROYECTO**

***Continuaremos trabajando con la base de datos de películas y sus comentarios.***

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

[{$project: {

\_id:0,

genres:1,

countries:1,

}}, {$unwind: {

path: "$genres",

preserveNullAndEmptyArrays:false

}}, {$unwind: {

path: "$countries",

preserveNullAndEmptyArrays:false

}}, {$group: {

\_id: "$countries",

total: {$push:{

genero:"$genres"

}}

}}, {$unwind: {

path: "$total",

preserveNullAndEmptyArrays:false

}}, {$addFields: {

genero:"$total.genero"

}}, {$group: {

\_id:{pais:"$\_id",

genero:"$genero"

},suma:{$sum:1}

}}, {$addFields: {

pais:"$\_id.pais",

genero:"$\_id.genero"

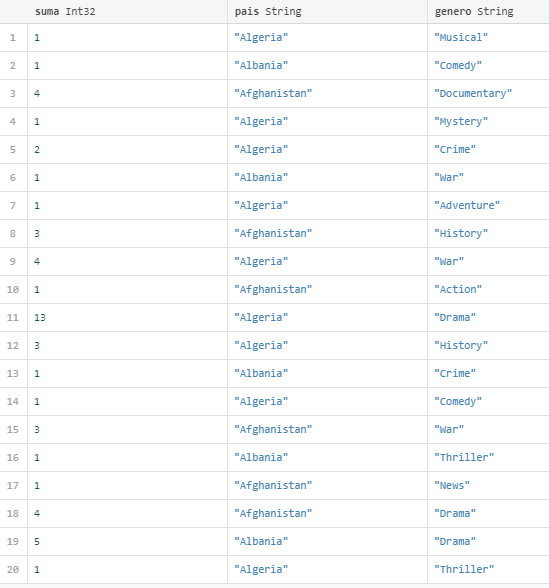
}}, {$project: {

\_id:0

}}, {$sort: {

pais:1

}}]



**SESIÓN 07**

**RETO 1**

1. **Usando como base el archivo movies.dat, limpiarlo e importar los datos en la tabla movies creada en el Reto 1.**
2. **Usando como base el archivo ratings.dat, limpiarlo e importar los datos en la tabla ratings creada en el Reto 2.**

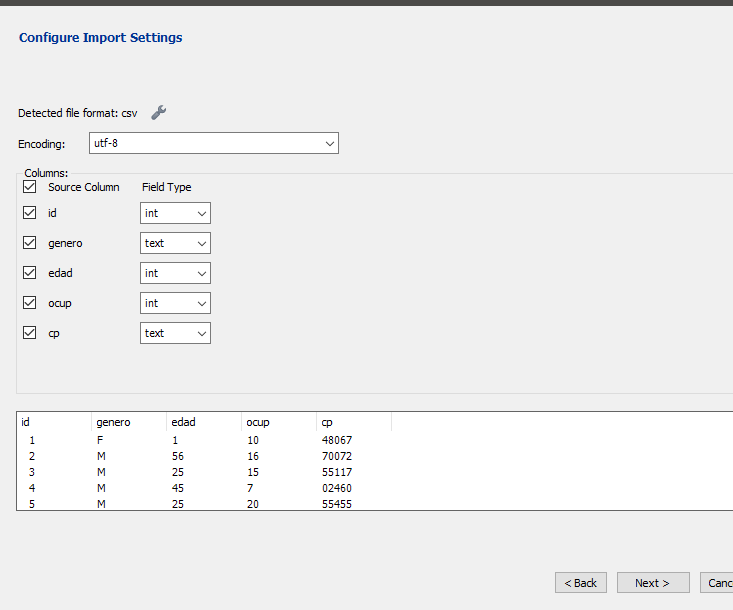
**RETO 2**

1. ***Usando como base el archivo movies.dat, limpiarlo e importar los datos en la tabla movies creada en el Reto 1.***
2. ***Usando como base el archivo ratings.dat, limpiarlo e importar los datos en la tabla ratings creada en el Reto 2.***

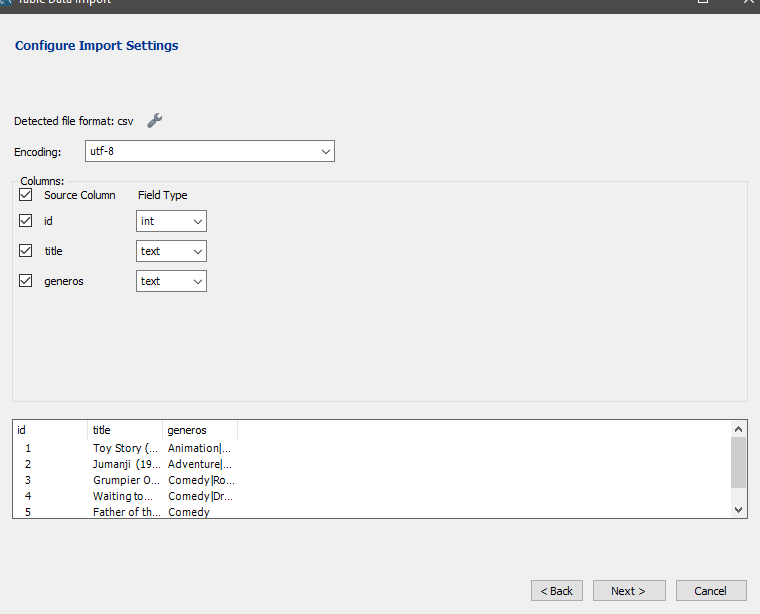
**RETO 3**

1. ***UsandoSe deberá de crear colecciones e importar los datos de los archivos movies.csv y ratings.csv.***

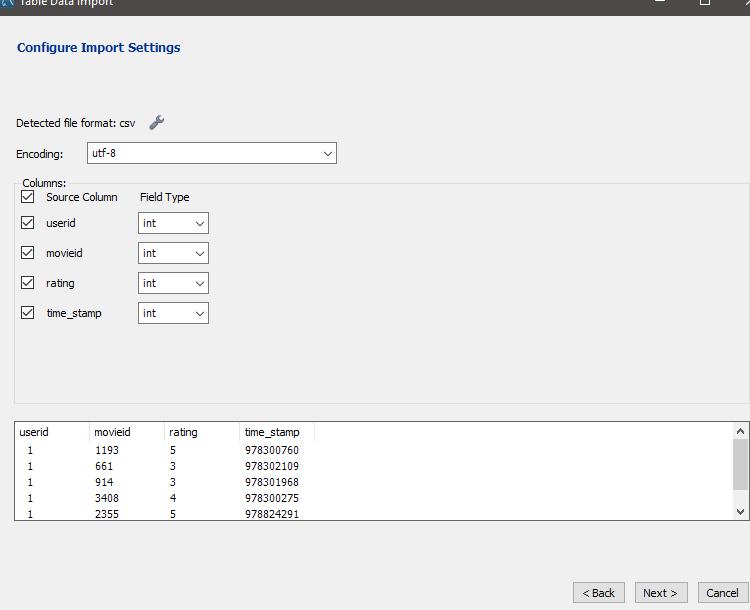
USERS:

******

MOVIES:



RATINGS:



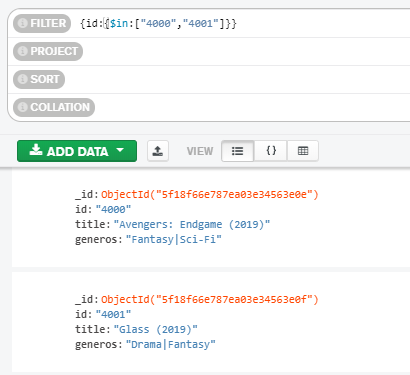
**PROYECTO**

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

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

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

***4001,Glass (2019),Drama|Fantasy***

******

**SESIÓN 08**

**RETO 1**

1. ***Descarga la fuente de datos de los locales de Starbucks:***[***directory.csv***](https://github.com/beduExpert/A1-Introduccion-a-Bases-de-Datos-2020/blob/master/Sesion-08/Reto-01/datos/directory.csv)
2. ***Analiza los datos, limpia los datos en caso de ser necesario.***
3. ***Elige MySQL o MongoDB y crea una base de datos para el conjunto de datos del reto.***
4. ***Carga los datos en la base de datos que elegiste y revisa que éstos se muestren correctamente.***
5. ***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: {

"City":"Mexico"

}}, {$match: {

$and:

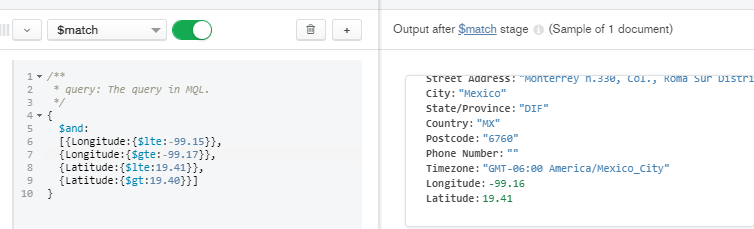
[{Longitude:{$lte:-99.15}},

{Longitude:{$gte:-99.17}},

{Latitude:{$lte:19.41}},

{Latitude:{$gt:19.40}}]

}}]

******

**RETO 2**

1. ***Descarga la fuente de datos sobre la pandemia del 2009 (H1N1):***[***Pandemic (H1N1) 2009.csv***](https://github.com/beduExpert/A1-Introduccion-a-Bases-de-Datos-2020/blob/master/Sesion-08/Reto-02/datos/h1n1.csv)
2. ***Analiza los datos, limpia los datos en caso de ser necesario.***
3. ***Elige MySQL o MongoDB y crea una base de datos para el conjunto de datos del reto.***
4. ***Carga los datos en la base de datos que elegiste y revisa que éstos se muestren correctamente.***
5. ***Responde a las siguientes preguntas usando consultas:***
   1. ***¿Cuál fue el país con mayor número de muertes?***

select Country,Deaths from h1n1 order by Deaths DESC LIMIT 1;

****

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

SELECT Country,Deaths FROM h1n1 ORDER BY Deaths ASC LIMIT 1;



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

SELECT Country,Cases FROM h1n1 ORDER BY Cases DESC LIMIT 1;



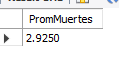
* 1. ***¿Cuál fue el país con el menor número de casos?***

SELECT Country,Cases FROM h1n1 ORDER BY Cases LIMIT 1;



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

SELECT AVG(Deaths) AS PromMuertes FROM h1n1;



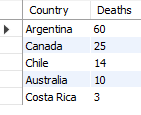
* 1. ***¿Cuál fue el número de casos promedio?***

SELECT AVG(Cases) AS PromMuertes FROM h1n1;



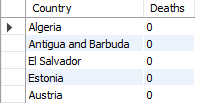
* 1. ***Top 5 de países con más muertes***

select Country,Deaths from h1n1 order by Deaths DESC LIMIT 5;



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

select Country,Deaths from h1n1 order by Deaths LIMIT 5;



**RETO 3**

1. ***Descarga la fuente de datos de los casos sobre la pandemia del COVID-19:***[***2019-nCoV-cases-JHU.csv***](https://github.com/beduExpert/A1-Introduccion-a-Bases-de-Datos-2020/blob/master/Sesion-08/Reto-03/datos/2019-nCoV-cases-JHU.csv)
2. ***Analiza los datos, limpia los datos en caso de ser necesario.***
3. ***Elige MySQL o MongoDB y crea una base de datos para el conjunto de datos del reto.***
4. ***Carga los datos en la base de datos que elegiste y revisa que éstos se muestren correctamente.***
5. ***Responde a las siguientes preguntas usando consultas:***
   1. ***¿Cuál es país con mayor número de casos?***

[{$group: {

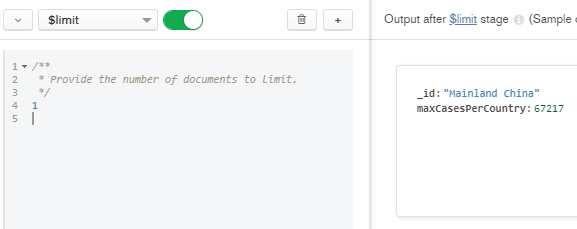
\_id: "$Region",

maxCasesPerCountry:{$max:"$Confirmed"}}}, {$sort: {

maxCasesPerCountry:-1

}}, {$limit: 1

}]

****

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

[{$group: {

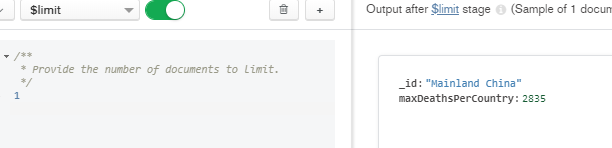
\_id: "$Region",

maxDeathsPerCountry:{$max:"$Deaths"}}}, {$sort: {

maxDeathsPerCountry:-1

}}, {$limit: 1

}]



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

[{$match: {

Lat: { $ne: "" },

Long: { $ne: "" }

}}, {$addFields: {

Lat: { $convert: { input: "$Lat", to: "double"}},

Long: { $convert: { input: "$Long", to: "double"}}

}}, {$group: {

\_id: null, size: {

$sum: 1

},

sumLat: {

$sum: '$Lat'

},

sumLong: {

$sum: '$Long'

},

avgLat: {

$avg: '$Lat'

},

avgLong: {

$avg: '$Long'

}

}}, {$project: { checkAvgLat: {

$divide: [

'$sumLat', '$size'

]

},

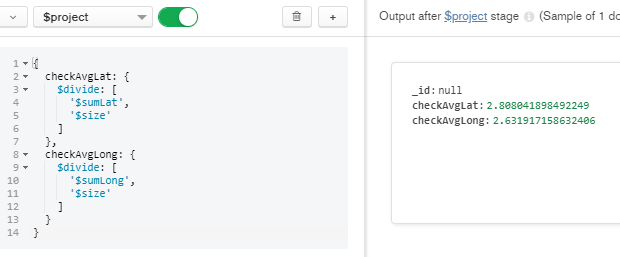
checkAvgLong: {

$divide: [ '$sumLong', '$size'

]

}

}}]



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

[{$match: { Lat: {

$ne: ''

},

Long: {

$ne: ''

}

}}, {$group: {

\_id: "$Region", Lat: {

$max: "$Lat"

},

Long: {

$max: "$Long"

}

}}, {$addFields: { Lat: {

$convert: { input: '$Lat',

to: 'double'

}

},

Long: {

$convert: { input: '$Long', to: 'double'

}

}

}}, {$match: {

$and: [

{

Lat: {

$gte: 21

}

},

{

Lat: {

$lte: 35

}

},

{

Long: {

$gte: 30

}

},

{

Long: {

$lte: 45

}

}

]

}}, {$limit: 5}, {$project: {

\_id: 1

}}]

