

Luis Guillermo Jiménez Bojórquez

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, officeCode
```

```
from employees
```

```
where officeCode in (1, 2, 3);
```

Dentro de la tabla employees, obtén 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 código 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";
```

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;
```

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;
```

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");
```

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");
```

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);
```

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;
```

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;
```

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 limit 5;
```

Proyecto 2

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

```
#COLLECTION
```

```
companies
```

```
#PROJECT
```

```
{name:1, email_address:1, twitter_username:1, phone_number:1,_id:0}
```

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

#COLLECTION

tweets

#PROJECT

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

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

#COLLECTION

companies

#FILTER

{founded_month:{\$eq: 10}}

#PROJECT

{name:1,founded_month:1,_id:0}

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

#COLLECTION

companies

#FILTER

{founded_year: {\$eq:2008}}

#PROJECT

{name:1,founded_year:1,_id:0}

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

#COLLECTION

posts

#FILTER

{author:{\$eq: "machine"}}

#PROJECT

```
{author:1,body:1,_id:0}
```

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

```
#COLLECTION
```

```
tweets
```

```
#FILTER
```

```
{source:{$eq:"web"}}
```

```
#PROJECT
```

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

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

```
#COLLECTION
```

```
companies
```

```
#FILTER
```

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

```
#PROJECT
```

```
{name:1,founded_month:1,founded_year:1,_id:0}
```

8. Obtén el apellido y nombre de los empleados cuya oficina está ubicada en USA.

```
#COLLECTION
```

```
companies
```

```
#FILTER
```

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

```
#PROJECT
```

```
{name:1,number_of_employees:1,_id:0}
```

```
#SORT
```

```
{number_of_employees:1}
```

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

#COLECTION

stories

#FILTER

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

#PROJECT

{title:1, comments:1,_id:0}

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

#COLECTION

companies

#FILTER

{number_of_employees:{\$ne:null}}

#PROJECT

{name:1,number_of_employees:1,_id:0}

#SORT

{number_of_employees:1}

#LIMIT

1

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

#COLECTION

companies

#PROJECT

{name:1,number_of_employees:1,_id:0}

#SORT

{number_of_employees:-1}

#LIMIT

1

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

#COLLECTION

stories

#PROJECT

{title:1,comments:1,_id:0}

#SORT

{comments: -1}

#LIMIT

1

13. Dentro de la tabla orders, obtén el número de órdenes que hay por cada estado.

#COLLECTION

stories

#PROJECT

{title:1,comments:1,_id:0}

#SORT

{comments: 1}

#LIMIT

1

Proyecto 3

#RIGHT JOIN

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

```
SELECT p.productCode AS Product_Code, p.productName AS Product_Name,textDescription AS  
Text_Description
```

```
FROM products p
```



```
RIGHT JOIN productlines l
ON p.productLine = l.productLine
ORDER BY p.productCode;
```

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

```
SELECT o.orderNumber AS Order_Number, o.status AS Estado,SUM ((d.quantityOrdered *
d.priceEach)) AS Costo_Total
FROM orders o
RIGHT JOIN orderdetails d
ON o.orderNumber = d.orderNumber
group by o.orderNumber
```

#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 o.orderNumber AS Order_Number,o.orderDate AS Order_Date,d.orderLineNumber AS
Order_Line_Number,p.productName AS Product_Name,d.quantityOrdered AS
Quantity_Ordered,d.priceEach AS Price_Each
FROM orders o
RIGHT JOIN orderdetails d
ON o.orderNumber = d.orderNumber
RIGHT JOIN products p
ON d.productCode = p.productCode
ORDER BY o.orderNumber, o.orderDate, d.orderLineNumber;
```

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

```
SELECT o.orderNumber AS Order_Number,p.productName AS Product_Name,p.MSRP AS
MSRP,d.priceEach AS Price_Each
FROM orders o
RIGHT JOIN orderdetails d
ON o.orderNumber = d.orderNumber
```

```
RIGHT JOIN products p
ON d.productCode = p.productCode
ORDER BY o.orderNumber, p.productName;
```

#LEFT JOIN

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

```
SELECT c.customerNumber AS Customer_Number,c.customerName AS
Customer_Name,o.orderNumber AS Order_Number,c.state AS State
FROM customers c
LEFT JOIN orders o
ON c.customerNumber = o.customerNumber
ORDER BY c.customerNumber, c.customerName;
```

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

```
SELECT c.customerNumber AS Customer_Number,c.customerName AS Customer_Name,c.state AS
State, o.orderNumber AS Order_Number
FROM customers c
LEFT OUTER JOIN orders o
ON c.customerNumber = o.customerNumber
WHERE o.orderNumber IS NULL
ORDER BY c.customerNumber, c.customerName;
```

#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 AS Last_Name,e.firstName AS First_Name,c.customerName AS
Customer_Name,p.checkNumber AS Check_Number,p.amount AS Total
FROM employees e
LEFT JOIN customers c
ON e.employeeNumber = c.salesRepEmployeeNumber
```

```
LEFT JOIN payments p
ON c.customerNumber = p.customerNumber
ORDER BY e.lastName, e.firstName, c.customerName;
```

#8 (I). Repite los ejercicios 5 a 7 usando RIGHT JOIN.

```
SELECT c.customerNumber AS Customer_Number,c.customerName AS
Customer_Name,o.orderNumber AS Order_Number,c.state AS State
FROM orders o
RIGHT JOIN customers c
ON c.customerNumber = o.customerNumber
ORDER BY c.customerNumber, c.customerName;
```

#8 (II). Repite los ejercicios 5 a 7 usando RIGHT JOIN.

```
SELECT c.customerNumber AS Customer_Number,c.customerName AS Customer_Name,c.state AS
State, o.orderNumber AS Order_Number
FROM orders o
RIGHT OUTER JOIN customers c
ON c.customerNumber = o.customerNumber
WHERE o.orderNumber IS NULL
ORDER BY c.customerNumber, c.customerName;
```

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

```
SELECT e.lastName AS Last_Name,e.firstName AS First_Name,c.customerName AS
Customer_Name,p.checkNumber AS Check_Number,p.amount AS Total
FROM customers c
RIGHT JOIN employees e
ON e.employeeNumber = c.salesRepEmployeeNumber
LEFT JOIN payments p
```

```
ON c.customerNumber = p.customerNumber  
ORDER BY e.lastName, e.firstName, c.customerName;
```

```
CREATE VIEW LuisGuillermoJimenezBojorquez01 AS (SELECT  
c.customerNumber,c.customerName,o.orderNumber,c.state  
FROM orders o  
RIGHT JOIN customers c  
ON c.customerNumber = o.customerNumber  
ORDER BY c.customerNumber, c.customerName);
```

Vista 1

```
SELECT customerNumber as "NumeroDeCliente",customerName as  
"NombreDeCliente",orderNumber as "NumeroDeOrden",state as "EstadoDelCliente"  
FROM LuisGuillermoJimenezBojorquez01;
```

-- No. 09 - 02

-- Vista 2

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

create view LuisGuillermoJimenezBojorquez02 as

```
(  
SELECT  
c.customerNumber,  
c.customerName,  
c.state,  
o.orderNumber  
FROM  
customers c  
LEFT OUTER JOIN orders o  
ON c.customerNumber = o.customerNumber  
WHERE o.orderNumber IS NULL
```

```
ORDER BY c.customerNumber, c.customerName  
);
```

```
-- Consulta Vista 2
```

```
SELECT  
customerNumber as "NumeroDeCliente",  
customerName as "NombreDeCliente",  
state as "EstadoDelCliente",  
orderNumber as "NumeroDeOrden"  
FROM  
LuisGuillermoJimenezBojorquez02 ;
```

```
-- No. 09 - 03
```

```
-- Vista 3
```

```
-- Obtén el apellido de empleado, nombre de empleado, nombre de cliente,  
-- número de cheque y total, es decir, los clientes asociados a cada empleado.  
create view LuisGuillermoJimenezBojorquez03 as
```

```
(  
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 e.lastName, e.firstName, c.customerName
);
```

-- Consulta Vista 3

```
select
lastName as "Apellido del Empleado",
firstName as "Nombre del Empleado",
customerName as "Nombre del Cliente",
checkNumber as "NumeroDeCheque",
amount as "Total"
from
LuisGuillermoJimenezBojorquez03;
```

Proyecto 4

Obtén los datos de contacto de cada compañía.

PROJECT: {email_address: 1, phone_number: 1}

Obtén la fuente de cada tweet.

PROJECT: {source: 1}

Obtén el nombre de todas las compañías fundadas en octubre.

FILTER: {founded_month: 10}

PROJECT: {name: 1}

Obtén el nombre de todas las compañías fundadas en 2008.

FILTER: {founded_year: 2008}

PROJECT: {name: 1}

Obtén todos los posts del autor machine.

FILTER: {author: "machine"}

Obtén todos los tweets provenientes de la web.

FILTER: {source: "web"}

Obtén todas las compañías fundadas en octubre del 2008.

FILTER: {\$and: [{founded_month: 10}, {founded_year: 2008}]}

Obtén todas las compañías con más de 50 empleados.

FILTER: {number_of_employees: {\$gt: 50}}

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

FILTER: {\$and: [{comments: {\$gte: 10}}, {comments: {\$lte: 30}}]}

Obtén la empresa con el menor número de empleados.

SORT: {number_of_employees: 1}

LIMIT: 1

Obtén la empresa con el mayor número de empleados.

SORT: {number_of_employees: -1}

LIMIT: 1

Obtén la historia más comentada.

SORT: {comments: -1}

LIMIT:1

Obtén la historia menos comentada.

SORT: {comments: 1}

LIMIT:1

Proyecto 5

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.

Proyecto Clase 05

[\$match: {

number_of_reviews: {\$gte: 50},

"review_scores.review_scores_rating": {\$gte: 80},

```

"amenities": { $in: [/Ethernet/i] },
"address.country_code" : "BR"
}}, {$sort: {
  number_of_reviews: 1
}}}

```

Proyecto 6

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

```

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

```

Proyecto Sesión 7

A continuación, se realizarán 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

Insert to Collection Eduardo.movies

VIEW

{}

≡

1	_id: ObjectId("5fc7c96330362d905d19a9c5")	ObjectId
2	id : "4000 "	String
+	titulo : "Avengers: Endgame (2019) "	String
4	genres : "Fantasy Sci-Fi "	String

CANCEL

INSERT

Insert to Collection Eduardo.movies

VIEW

{}

≡

1	_id: ObjectId("5fc7ca0130362d905d19a9c6")	ObjectId
2	id : "4001 "	String
3	titulo : "Glass (2019) "	String
4	genres : "Drama Fantasy "	String

```
_id: ObjectId("5fc7c96330362d905d19a9c5")
id: "4000"
titulo: "Avengers: Endgame (2019)"
genres: "Fantasy|Sci-Fi"
```

```
_id: ObjectId("5fc7ca0130362d905d19a9c6")
id: "4001"
titulo: "Glass (2019)"
genres: "Drama|Fantasy"
```

Eduardo.movies

DOCUMENTS 2TOTAL SIZE 185B AVG. SIZE 93BINDEXES 1TOTAL SIZE 20.0KB AVG. SIZE 20.0KB

DocumentsAggregationsSchemaExplain PlanIndexesValidation

FILTER

{id: {\$in: ["4000", "4001"]}}

OPTIONS

FIND

RESET

...

ADD DATA

VIEW

≡

{}

≡

Displaying documents 1 - 2 of 2

REFRESH

```
_id: ObjectId("5fc7c96330362d905d19a9c5")
id: "4000"
titulo: "Avengers: Endgame (2019)"
genres: "Fantasy|Sci-Fi"
```

```
_id: ObjectId("5fc7ca0130362d905d19a9c6")
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

The screenshot shows the MongoDB Compass interface with the 'Documents' tab selected. A filter is applied: `{id: {$in: ["4000", "4001"]}}`. Two documents are displayed:

```
{
  "_id": "5fc7c96320362d905d19a9c5",
  "id": "4000",
  "titulo": "Avengers: Endgame (2019)",
  "genres": "Fantasy|Sci-Fi"
}
```

```
{
  "_id": "5fc7ca0130362d905d19a9c6",
  "id": "4001",
  "titulo": "Glass (2019)",
  "genres": "Drama|Fantasy",
  "valoraciones": [
    {
      "userId": "1563",
      "movieId": "4001",
      "rating": "4"
    },
    {
      "userId": "434",
      "movieId": "4001",
      "rating": "5"
    }
  ]
}
```