

Proyectos Sesiones 1 a 7

Proyecto Sesión 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 Sesión 2

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

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

```
select employeeNumber, lastName, firstName from employees where lastName like "%on";
```

Dentro de la tabla employees, obtén 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%" or lastName like "%on%";
```

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

```
select employeeNumber, lastName, firstName from employees where firstName like "T_m";
```

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

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

```
select productCode, productName from products where productCode like "%_20%";
```

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

```
select orderNumber, count(*)
```

```
from orderdetails
```

```
group by orderNumber;
```

Dentro de la tabla orders Obtén el número de órdenes por año.

```
select year(orderDate), count(*)
```

from orders

group by year(orderDate);

Obtén 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");

Obtén 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);

Obtén 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);

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

select customerName

from customers

where customerNumber not in (select customerNumber from orders);

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

select max(cantidad), min(cantidad), avg(cantidad)

from(

select id_articulo, count(*) as cantidad

from venta

group by id_articulo

) as subconsulta;

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

select state, count(cantidad)

from

(select customerNumber, count(*) as cantidad

```
        from orders
    group by customerNumber
) as subconsulta, customers
group by state;
```

Proyecto Sesión 3

- Para estas consultas usa RIGHT JOIN

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

```
select productCode, productName, textDescription
from productlines pl
right join products p
on pl.productLine = p.productLine;
```

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

```
select od.orderNumber, state, sum(priceEach)
from customers c
right join orders o
on c.customerNumber = o.customerNumber
right join orderdetails od
on o.orderNumber = od.orderNumber
group by od.orderNumber;
```

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, orderDate, orderLineNumber, productName, quantityOrdered, buyPrice
from products p
right join orderdetails od
on p.productCode = od.productCode
right join orders o
on od.orderNumber = o.orderNumber;
```

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

```
select orderNumber, od.productCode, MSRP, buyPrice
```

```
from products p
right join orderdetails od
on p.productCode = od.productCode;
```

- **Para estas consultas usa LEFT JOIN**

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

```
select c.customerNumber, customerName, orderNumber, state
from customers c
left join orders o
on c.customerNumber = o.customerNumber;
```

Obtén los clientes que no tienen una orden asociada.

```
select customerName
from customers c
left join orders o
on c.customerNumber = o.customerNumber
where o.customerNumber is null
order by customerName;
```

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 lastName, firstName, customerName, checkNumber, amount
from employees e
left join customers c
on e.employeeNumber = c.salesRepEmployeeNumber
left join payments p
on c.customerNumber = p.customerNumber;
```

- **Para estas consultas usa RIGHT JOIN**

Repite los ejercicios 5 a 7 usando RIGHT JOIN.

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

```
select c.customerNumber, customerName, orderNumber, state
from orders o
```

right join customers c

on o.customerNumber = c.customerNumber;

Obtén los clientes que no tienen una orden asociada.

select customerName

from orders o

right join customers c

on o.customerNumber = c.customerNumber

where o.customerNumber is null

order by customerName;

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 lastName, firstName, customerName, checkNumber, amount

from payments p

right join customers c

on p.customerNumber = c.customerNumber

right join employees e

on c.salesRepEmployeeNumber = e.employeeNumber;

- **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 EduardoLuna31 as

(select c.customerNumber, customerName, orderNumber, state

from orders o

right join customers c

on o.customerNumber = c.customerNumber);

Obtén los clientes que no tienen una orden asociada.

create view EduardoLuna32 as

(select customerName

from orders o

right join customers c

on o.customerNumber = c.customerNumber

where o.customerNumber is null

order by customerName);

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 EduardoLuna33 as

(select lastName, firstName, customerName, checkNumber, amount

from payments p

right join customers c

on p.customerNumber = c.customerNumber

right join employees e

on c.salesRepEmployeeNumber = e.employeeNumber);

Proyecto Sesión 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 Sesión 5

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 ubicada en Brasil.

```
[{$match: {  
  $and:  
  [  
    {number_of_reviews: {$gte: 50}},  
    {"review_scores.review_scores_rating": {$gte: 80}},  
    {amenities: {$in: [/ethernet/i]}},  
    {"address.country": "Brazil"}  
  ]  
}]
```

Proyecto Sesión 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

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

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

Eduardo.movies

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

Documents Aggregations Schema Explain Plan Indexes Validation

Displaying documents 0 - 2 of 2

```
{
  "_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 2 TOTAL SIZE 185B AVG. SIZE 93B INDEXES 1 TOTAL SIZE 20.0KB AVG. SIZE 20.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

Displaying documents 1 - 2 of 2

```
{
  "_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:

Eduardo.movies

DOCUMENTS 2

TOTAL SIZE 185B

AVG. SIZE 93B

INDEXES 1

TOTAL SIZE 20.0KB

AVG. SIZE 20.0KB

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

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"
valoraciones: Array
  0: Object
    userid: "1563"
    movieid: "4001"
    rating: "4"
  1: Object
    userid: "434"
    movieid: "4001"
    rating: "5"
```