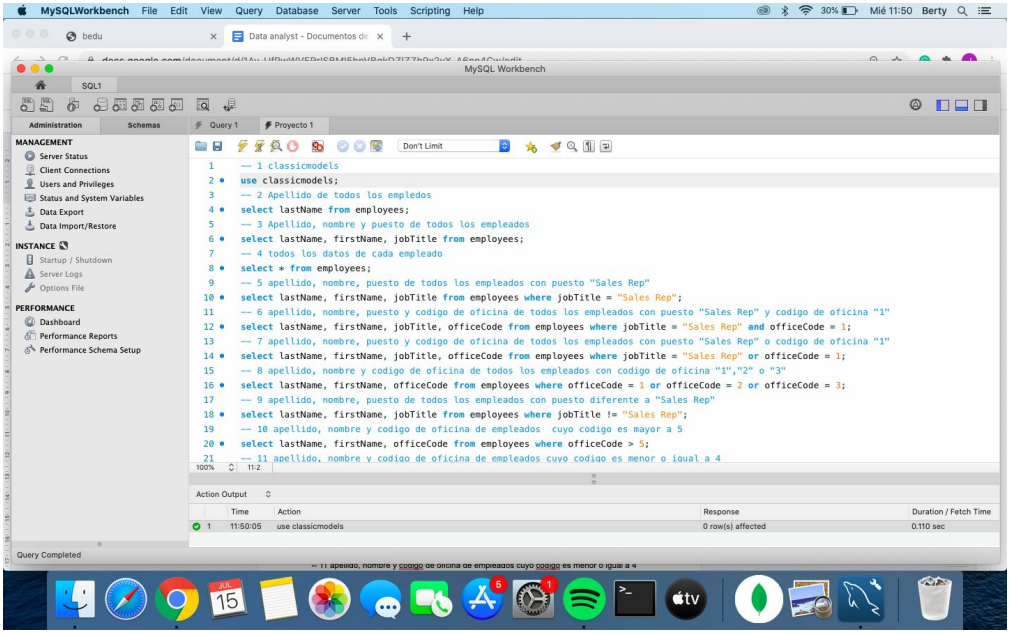
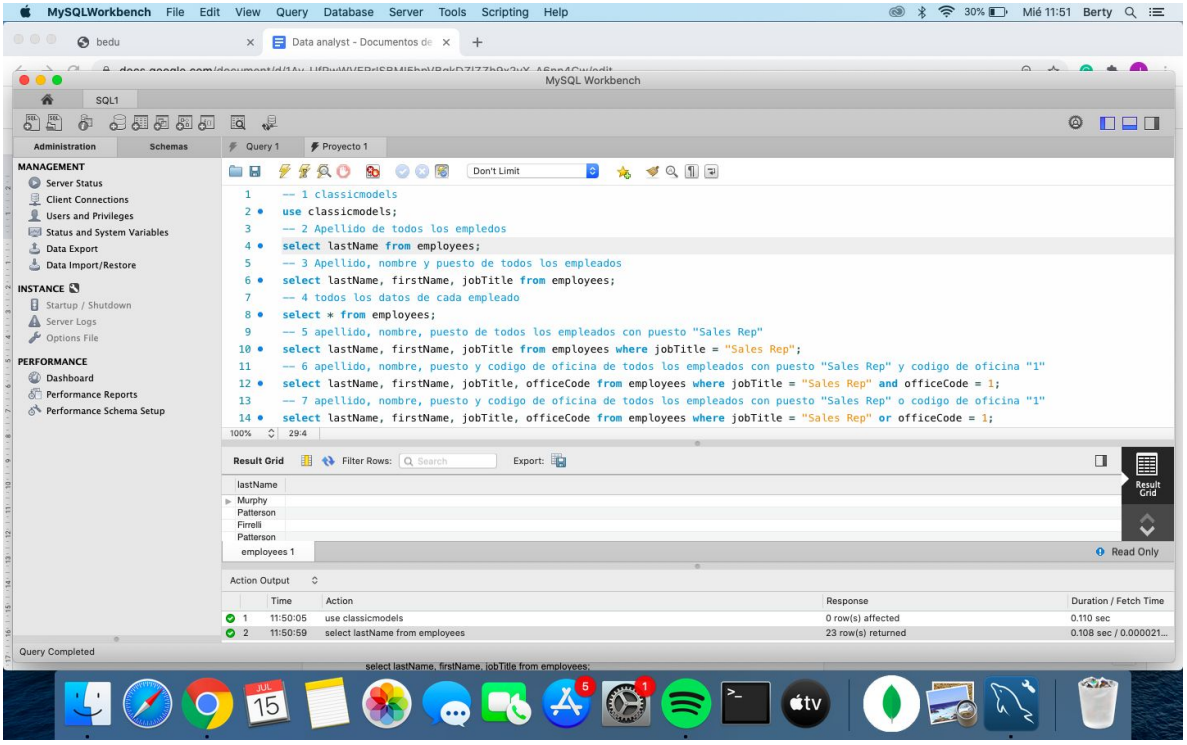
Sesión 01 Fundamentos SQL

Proyecto 1

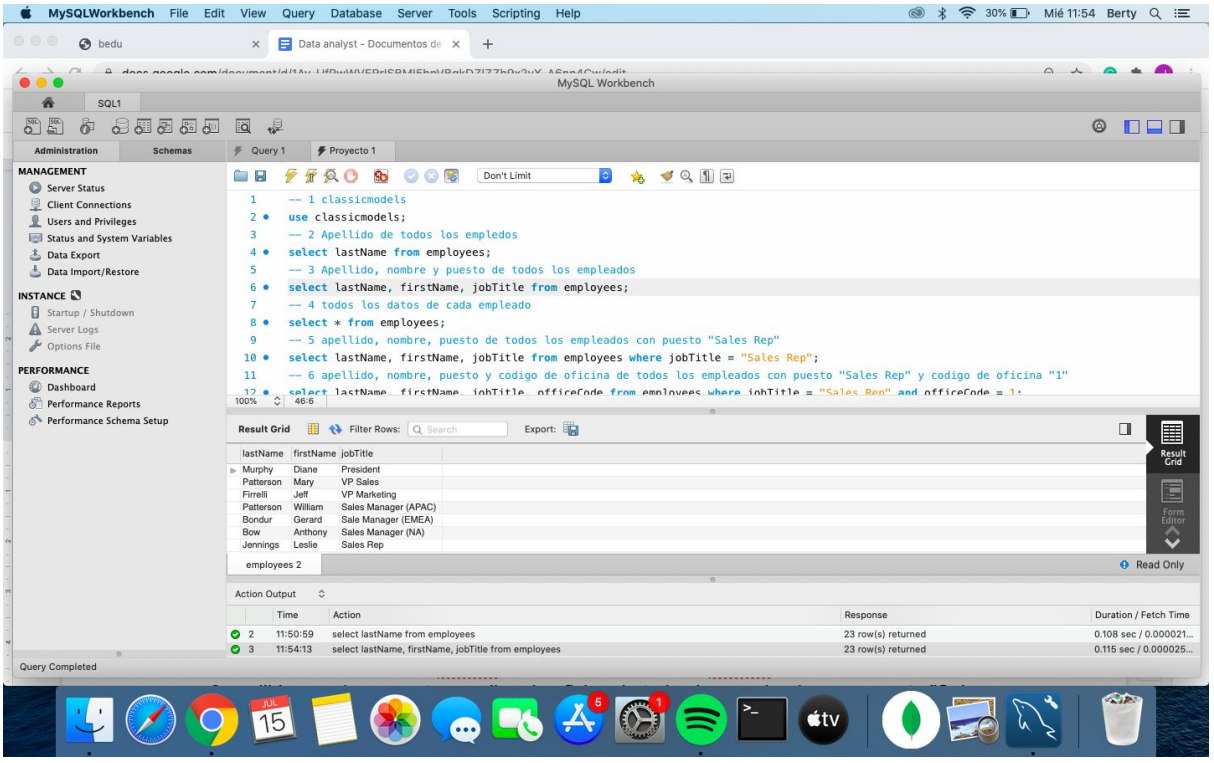
-- 1 classicmodels
use classicmodels;



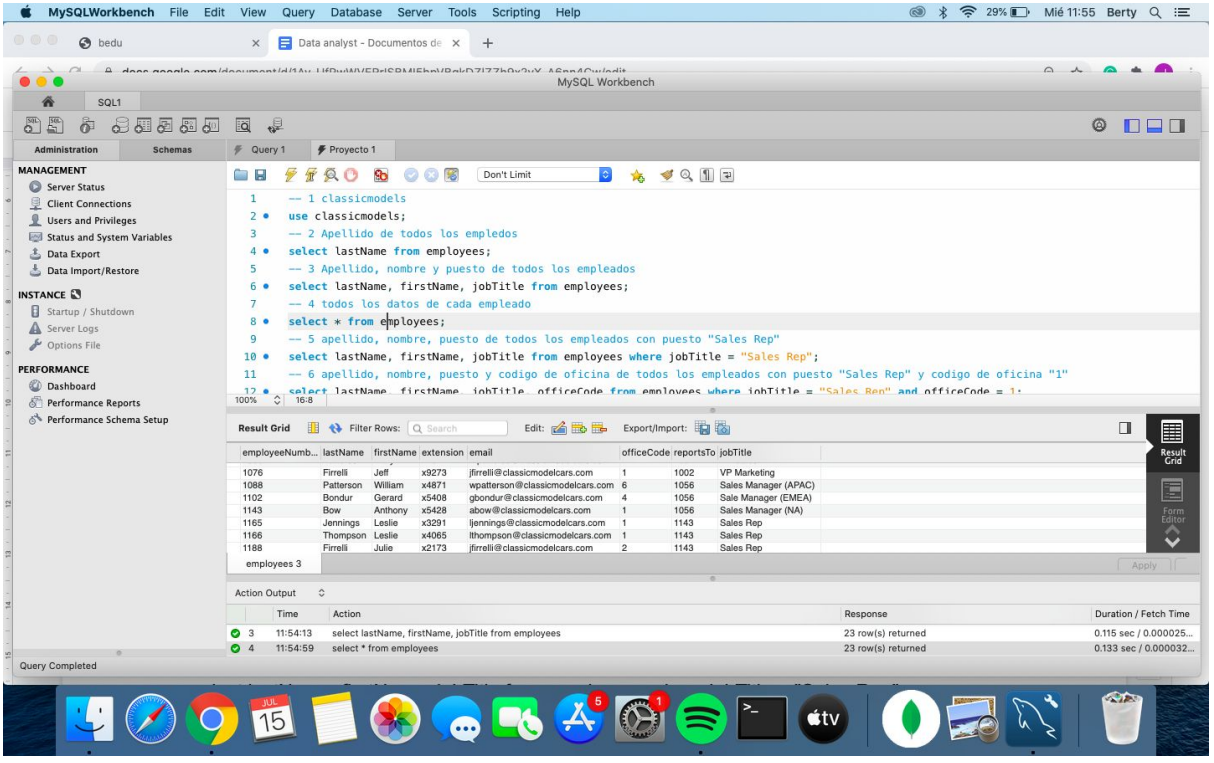
-- 2 Apellido de todos los empleados
select lastName from employees;



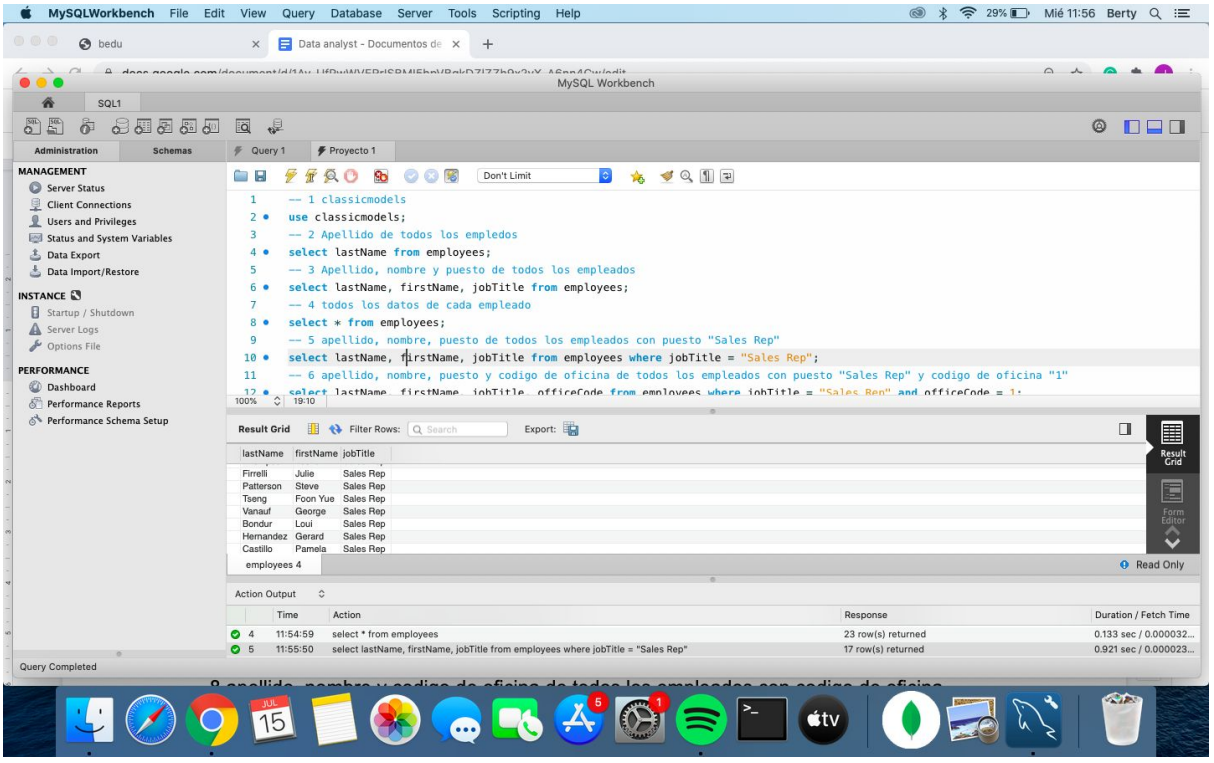
-- 3 Apellido, nombre y puesto de todos los empleados
select lastName, firstName, jobTitle from employees;



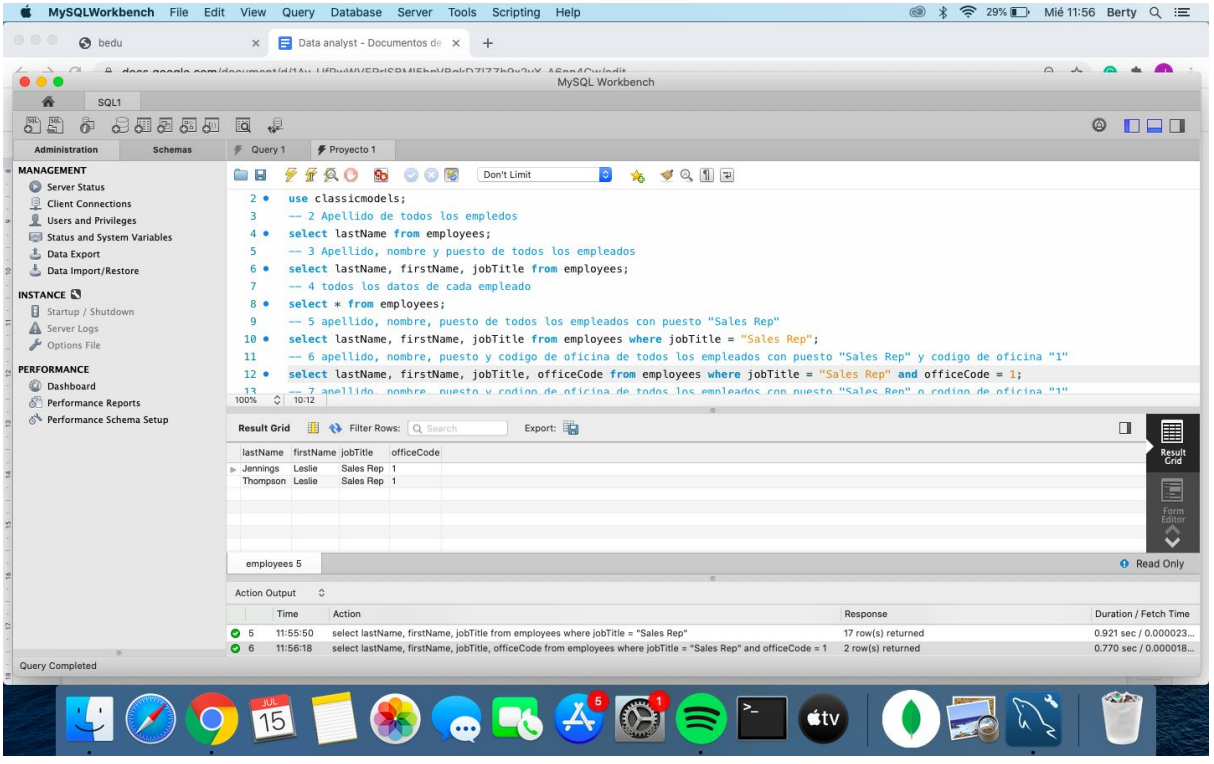
-- 4 todos los datos de cada empleado
select * from employees;



-- 5 apellido, nombre, puesto de todos los empleados con puesto "Sales Rep"
select lastName, firstName, jobTitle from employees where jobTitle = "Sales Rep";

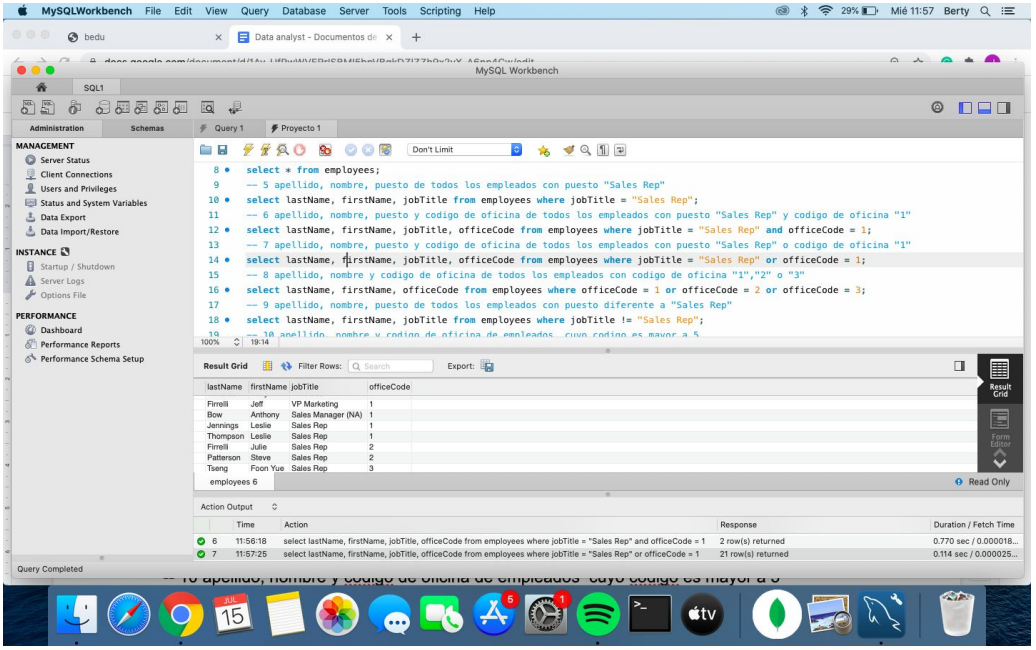


-- 6 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" y codigo de oficina "1"
select lastName, firstName, jobTitle, officeCode from employees where jobTitle = "Sales Rep" and officeCode = 1;



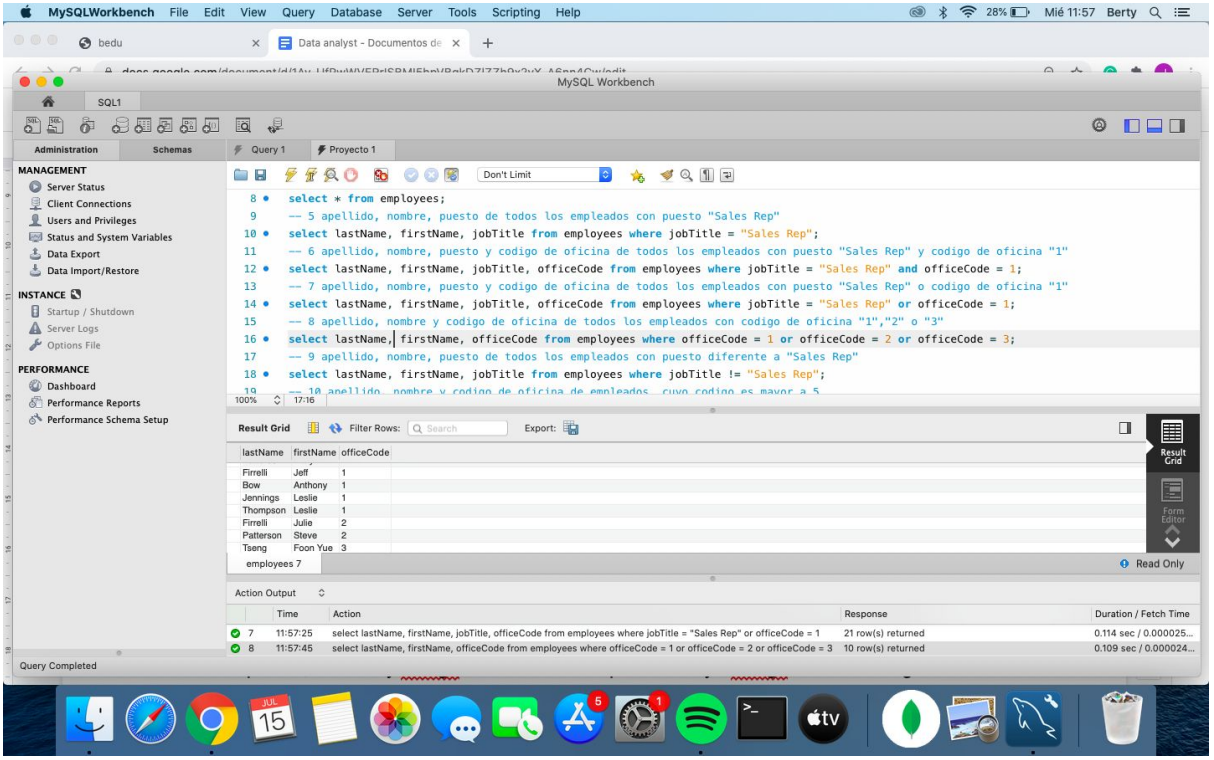
-- 7 apellido, nombre, puesto y codigo de oficina de todos los empleados con puesto "Sales Rep" o codigo de oficina "1"

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

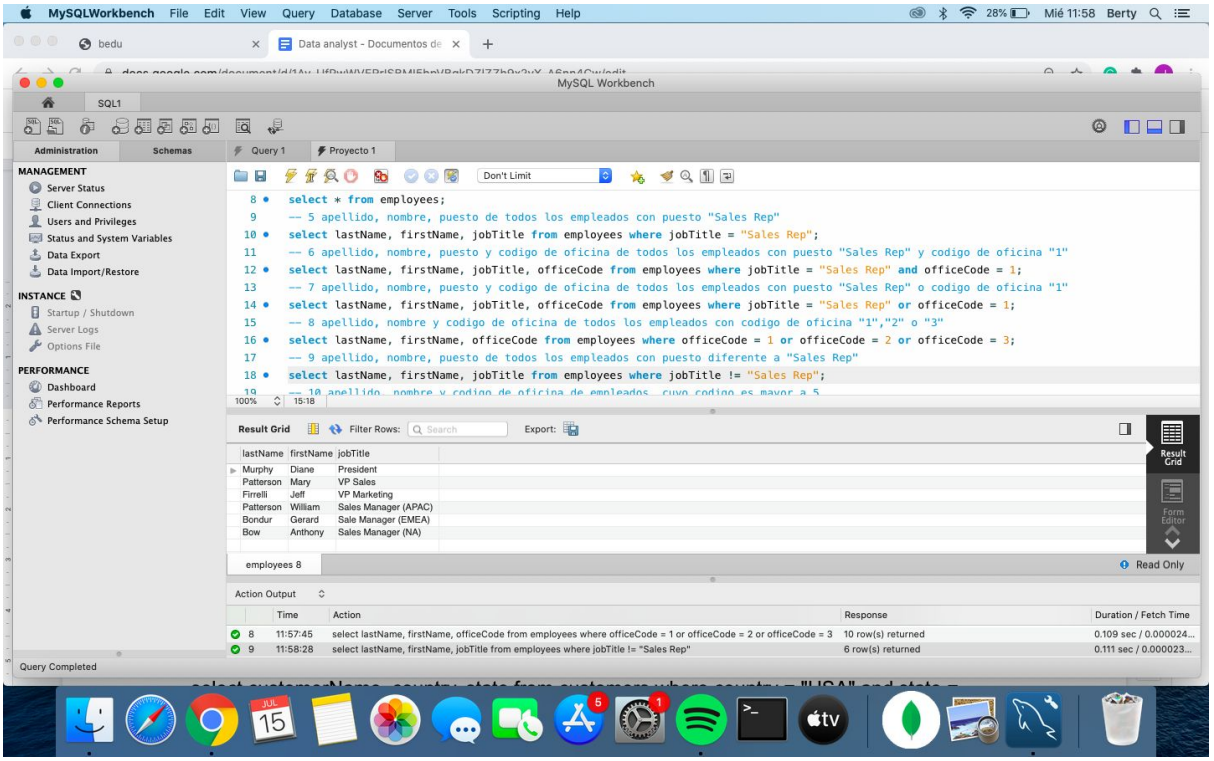


-- 8 apellido, nombre y codigo de oficina de todos los empleados con codigo de oficina "1","2" o "3"

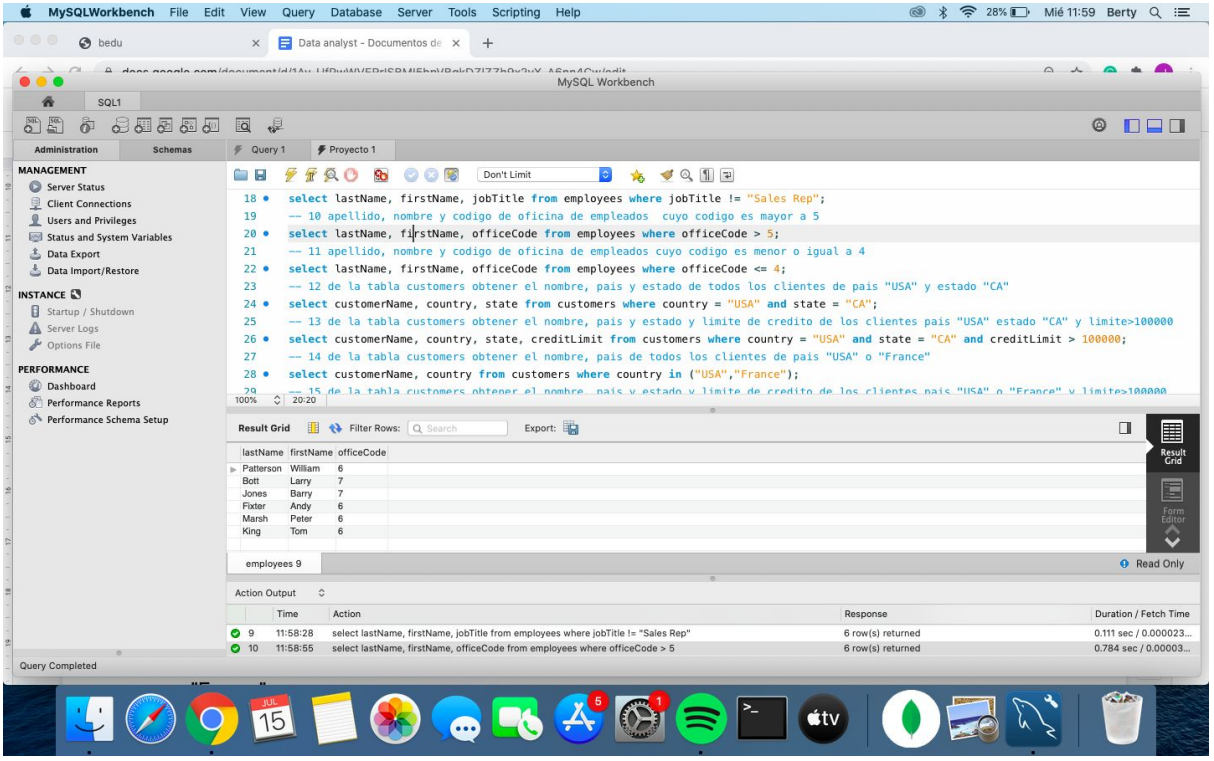
```
select lastName, firstName, officeCode from employees where officeCode = 1 or officeCode = 2 or officeCode = 3;
```



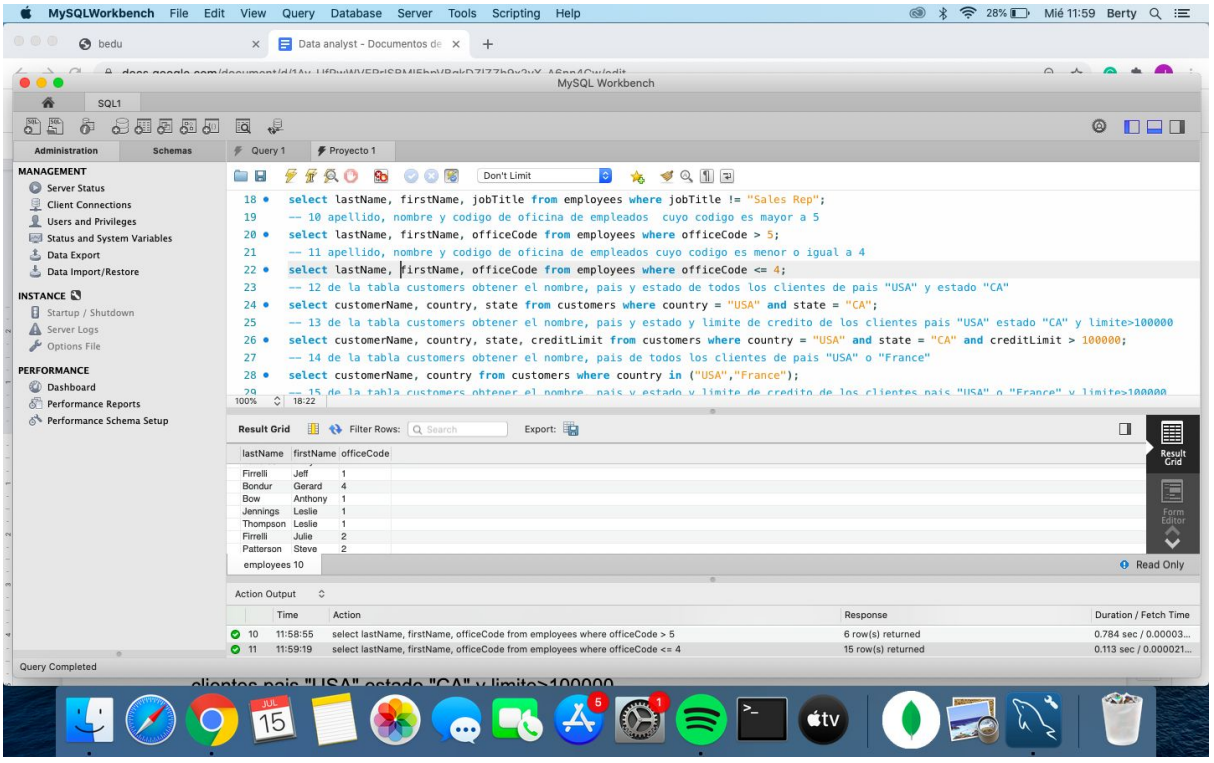
-- 9 apellido, nombre, puesto de todos los empleados con puesto diferente a "Sales Rep"
select lastName, firstName, jobTitle from employees where jobTitle != "Sales Rep";



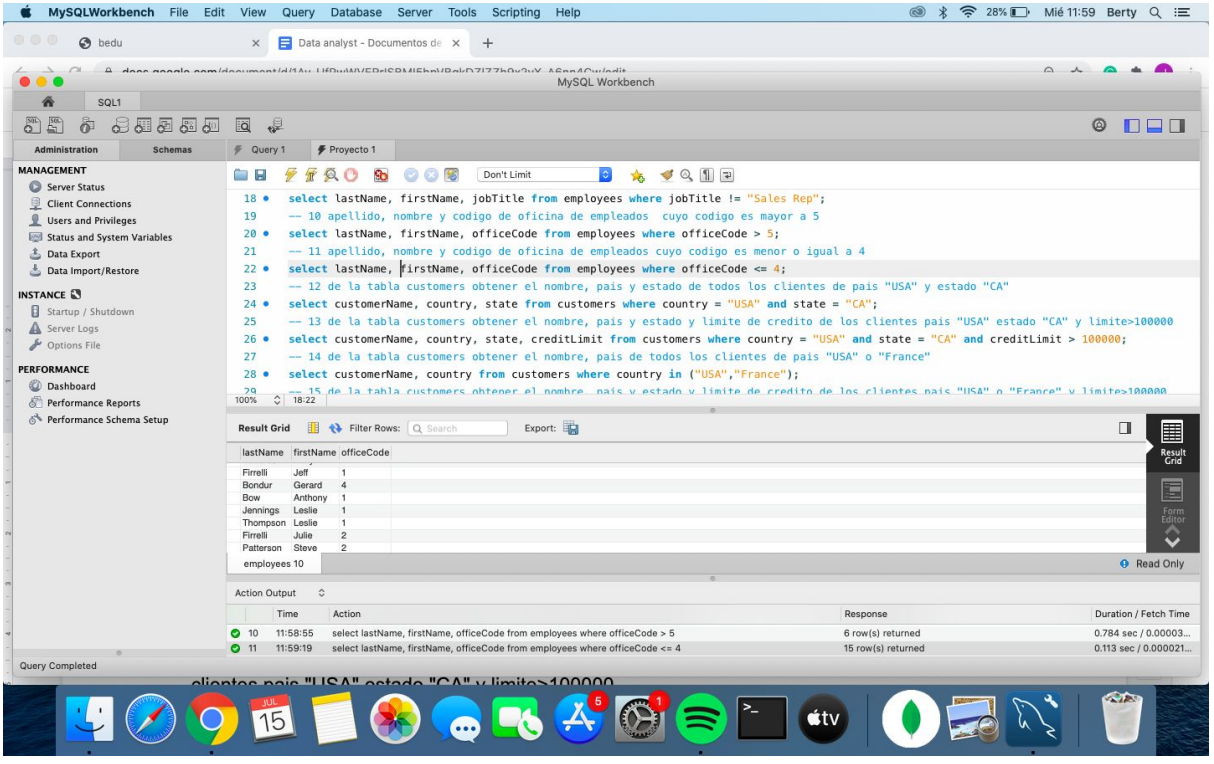
-- 10 apellido, nombre y codigo de oficina de empleados cuyo codigo es mayor a 5
select lastName, firstName, officeCode from employees where officeCode > 5;



-- 11 apellido, nombre y codigo de oficina de empleados cuyo codigo es menor o igual a 4
select lastName, firstName, officeCode from employees where officeCode <= 4;

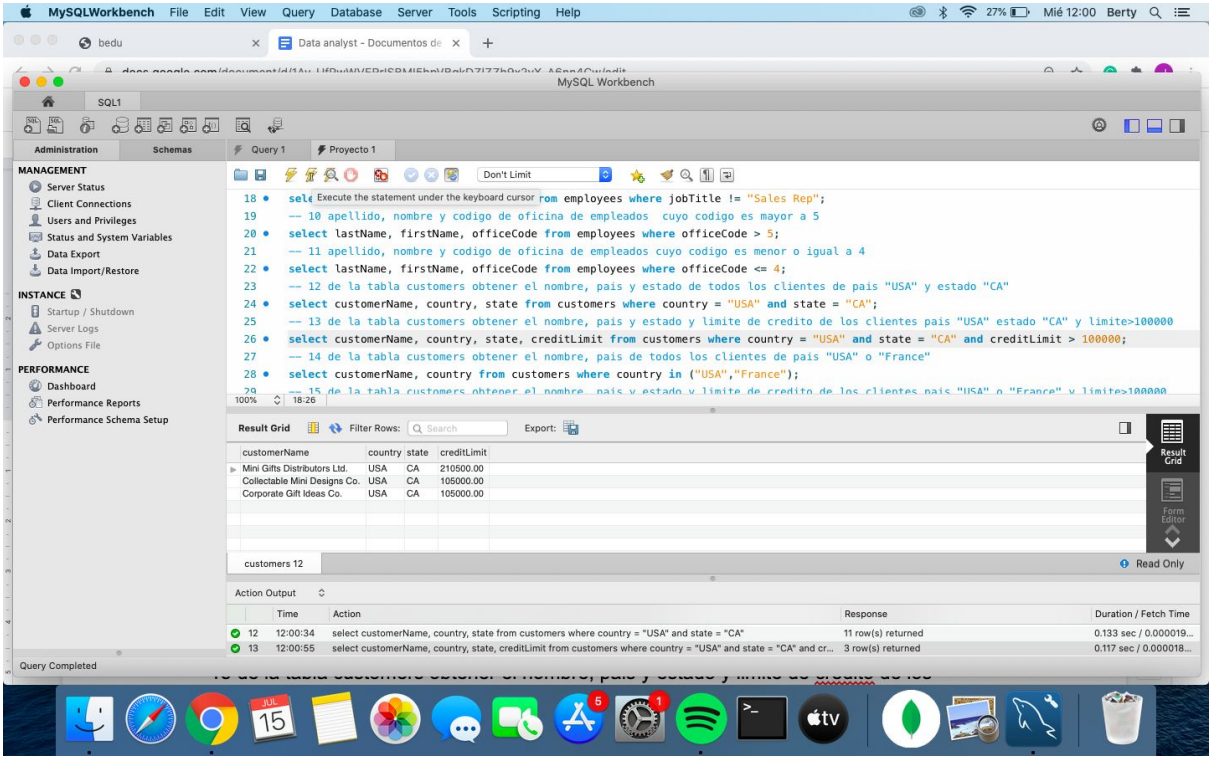


-- 12 de la tabla customers obtener el nombre, pais y estado de todos los clientes de pais "USA" y estado "CA"
select customerName, country, state from customers where country = "USA" and state = "CA";



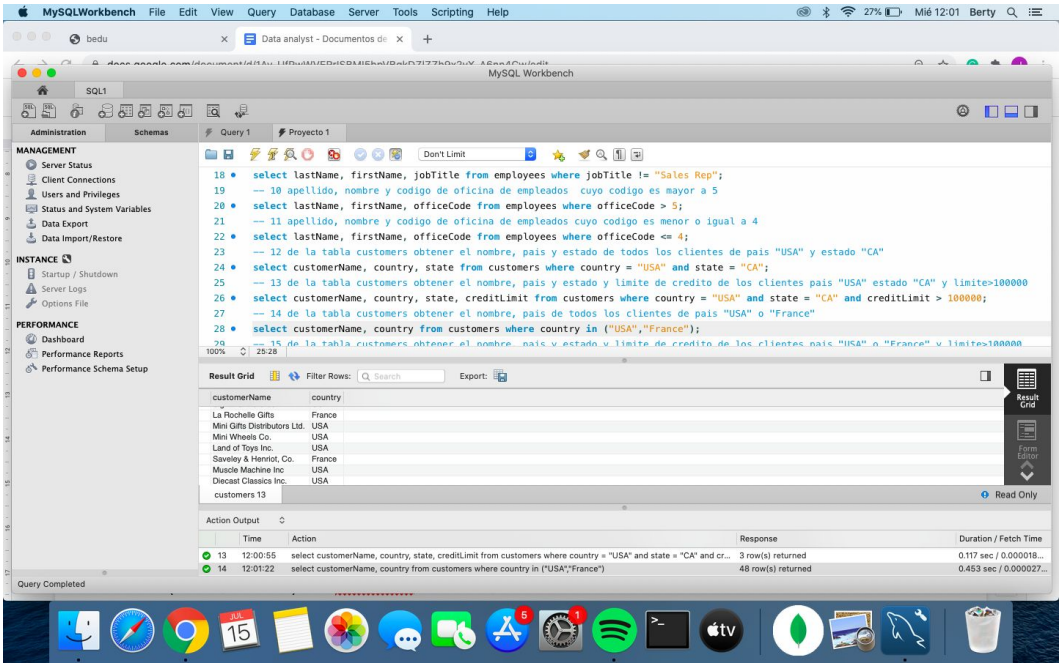
-- 13 de la tabla customers obtener el nombre, pais y estado y limite de credito de los clientes pais "USA" estado "CA" y limite>100000

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



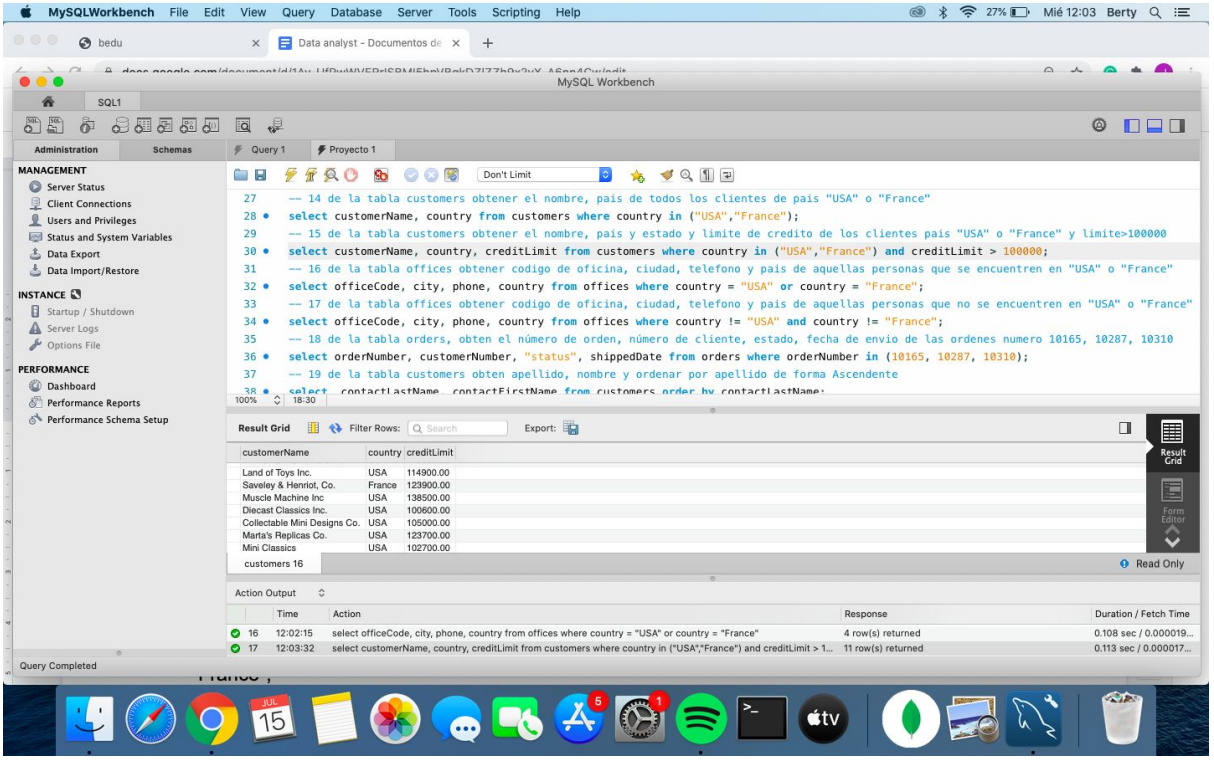
-- 14 de la tabla customers obtener el nombre, pais de todos los clientes de pais "USA" o "France"

```
select customerName, country from customers where country in ("USA","France");
```



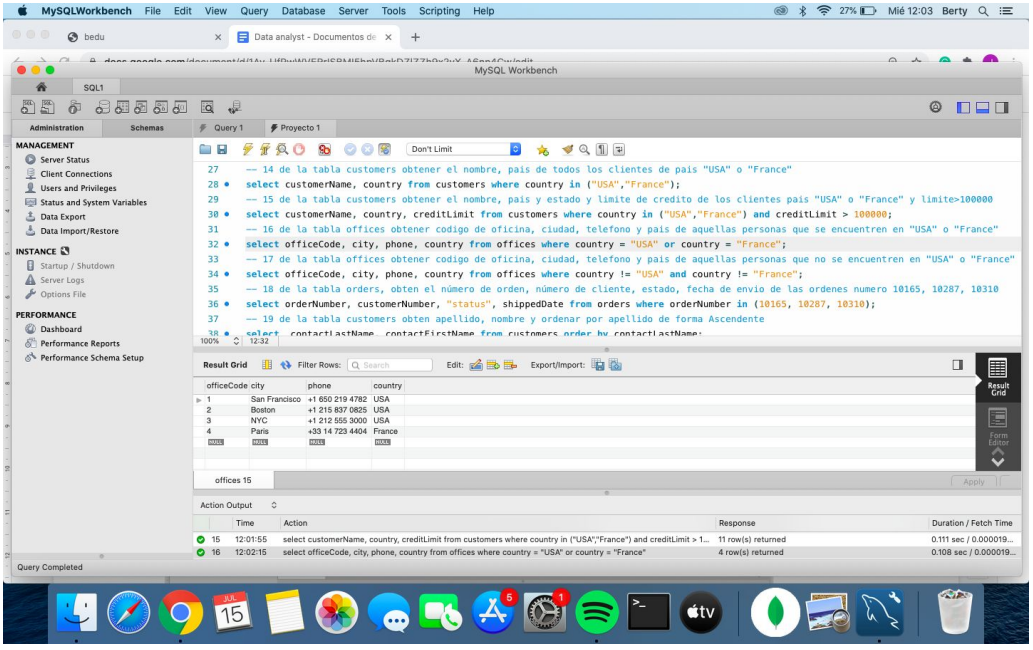
-- 15 de la tabla customers obtener el nombre, pais y estado y limite de credito de los clientes pais "USA" o "France" y limite>100000

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



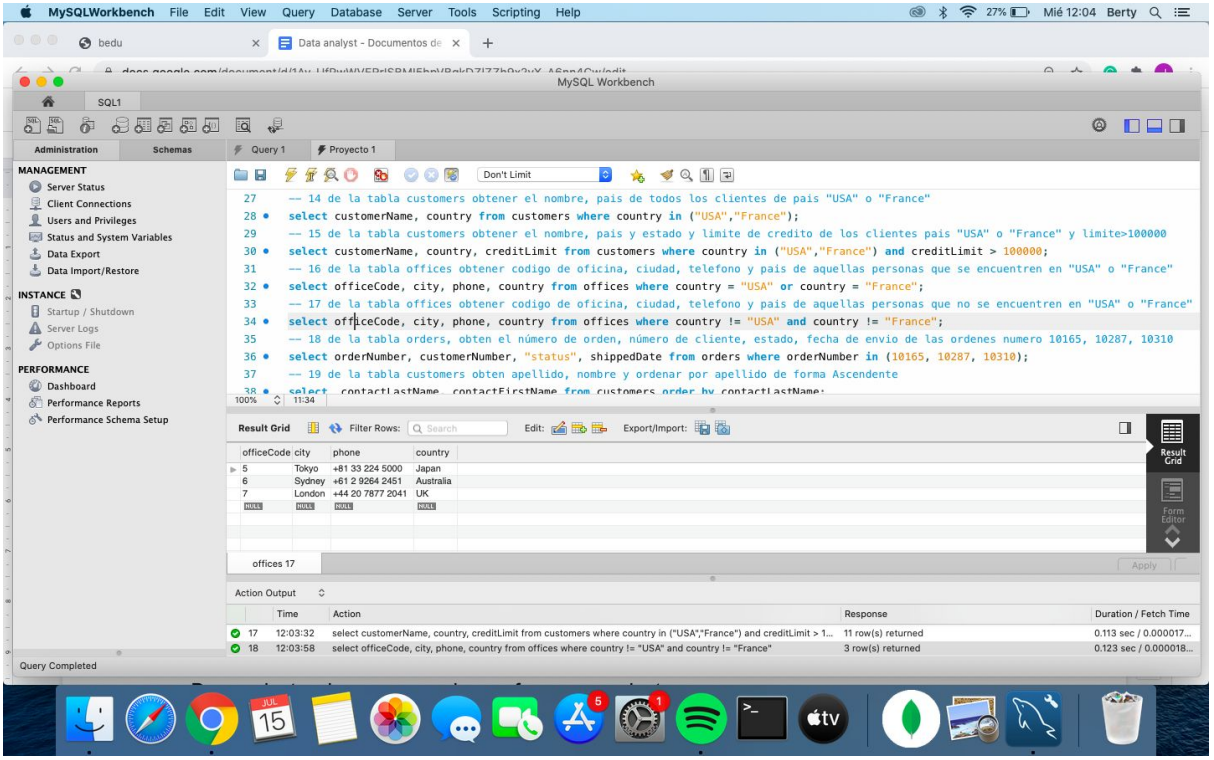
-- 16 de la tabla offices obtener codigo de oficina, ciudad, telefono y pais de aquellas personas que se encuentren en "USA" o "France"

```
select officeCode, city, phone, country from offices where country = "USA" or country = "France";
```



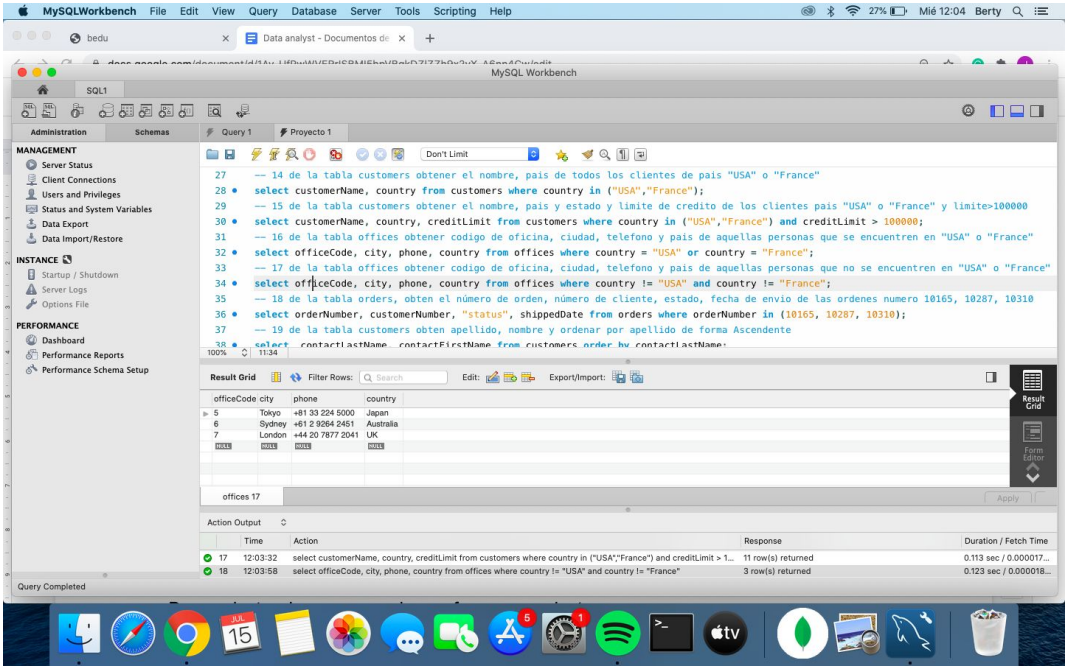
-- 17 de la tabla offices obtener codigo de oficina, ciudad, telefono y pais de aquellas personas que no se encuentren en "USA" o "France"

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



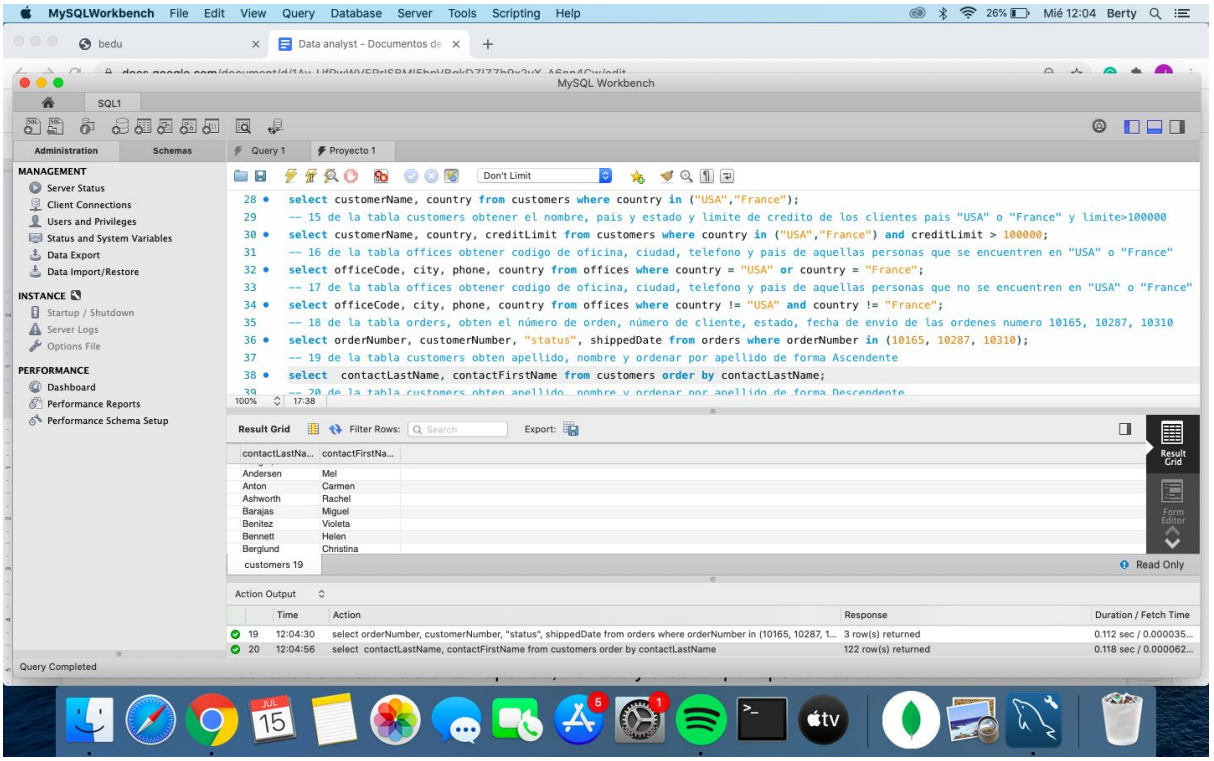
-- 18 de la tabla orders, obten el número de orden, número de cliente, estado, fecha de envio de las ordenes numero 10165, 10287, 10310

```
select orderNumber, customerNumber, "status", shippedDate from orders where orderNumber in (10165, 10287, 10310);
```



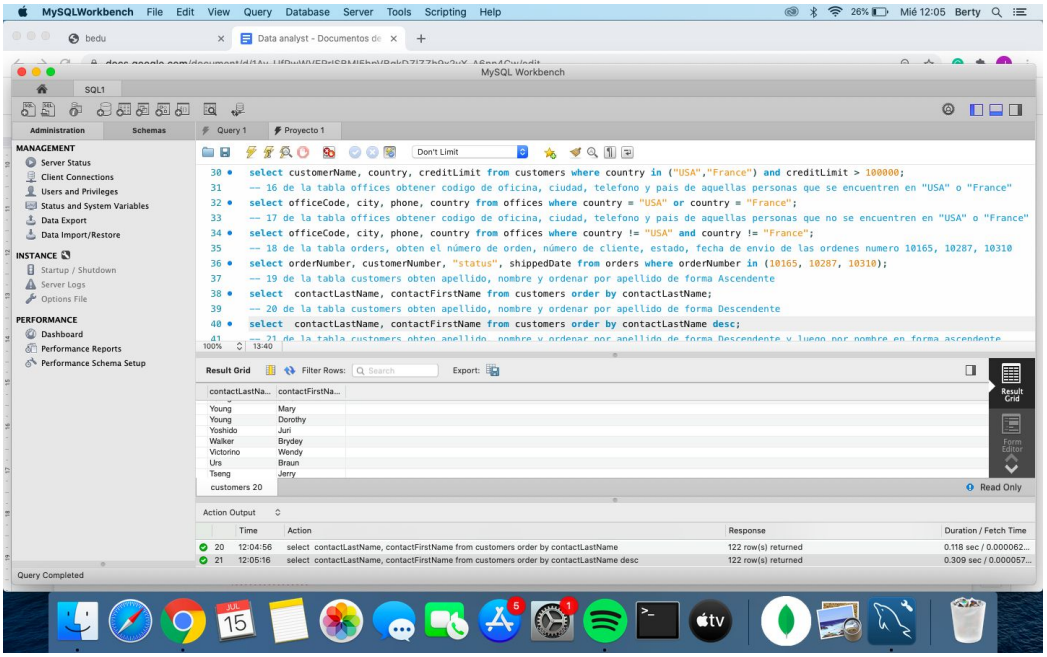
-- 19 de la tabla customers obten apellido, nombre y ordenar por apellido de forma Ascendente

```
select  contactLastName, contactFirstName from customers order by contactLastName;
```



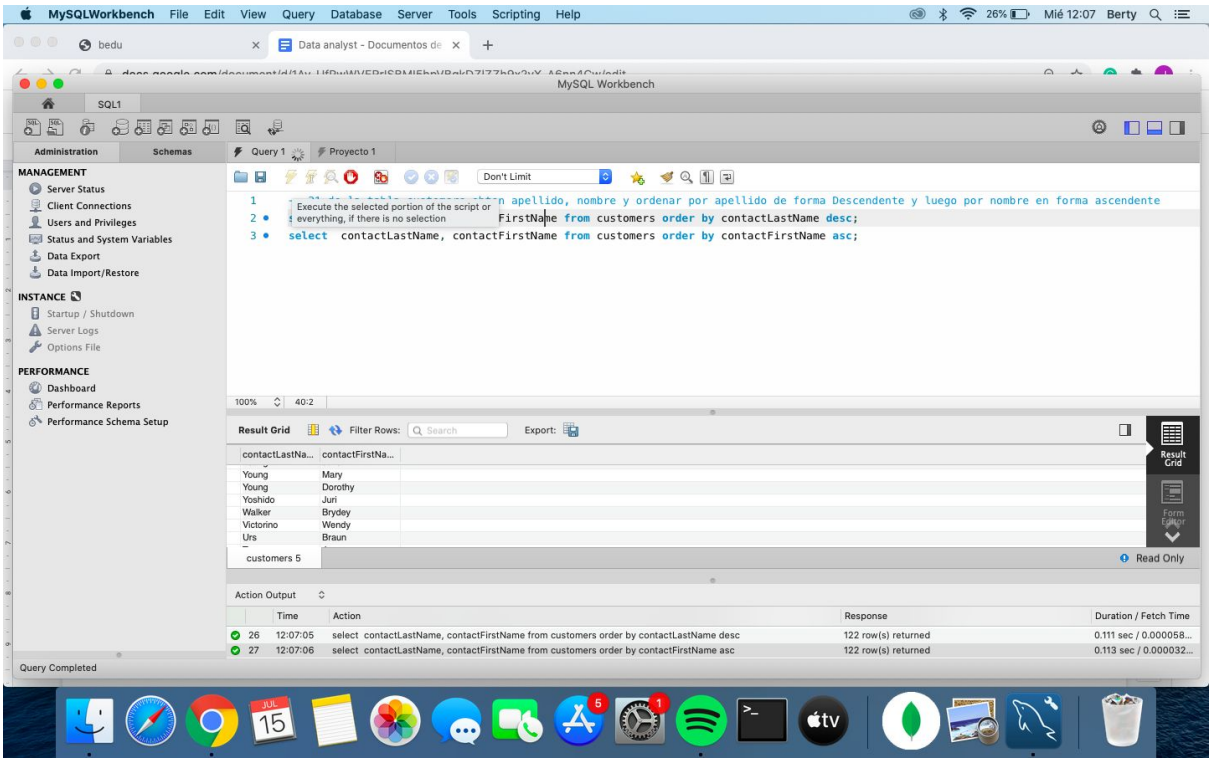
-- 20 de la tabla customers obten apellido, nombre y ordenar por apellido de forma Descendente

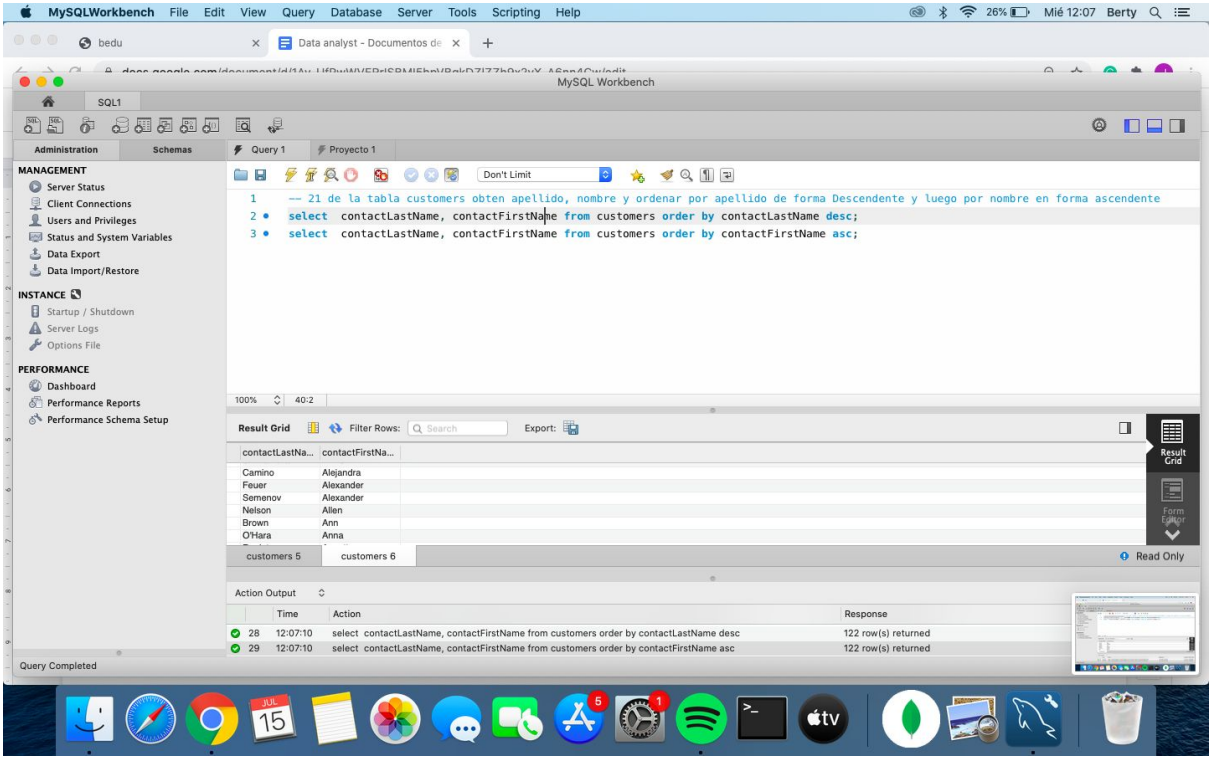
```
select  contactLastName, contactFirstName from customers order by contactLastName
desc;
```



-- 21 de la tabla customers obten apellido, nombre y ordenar por apellido de forma Descendente y luego por nombre en forma ascendente

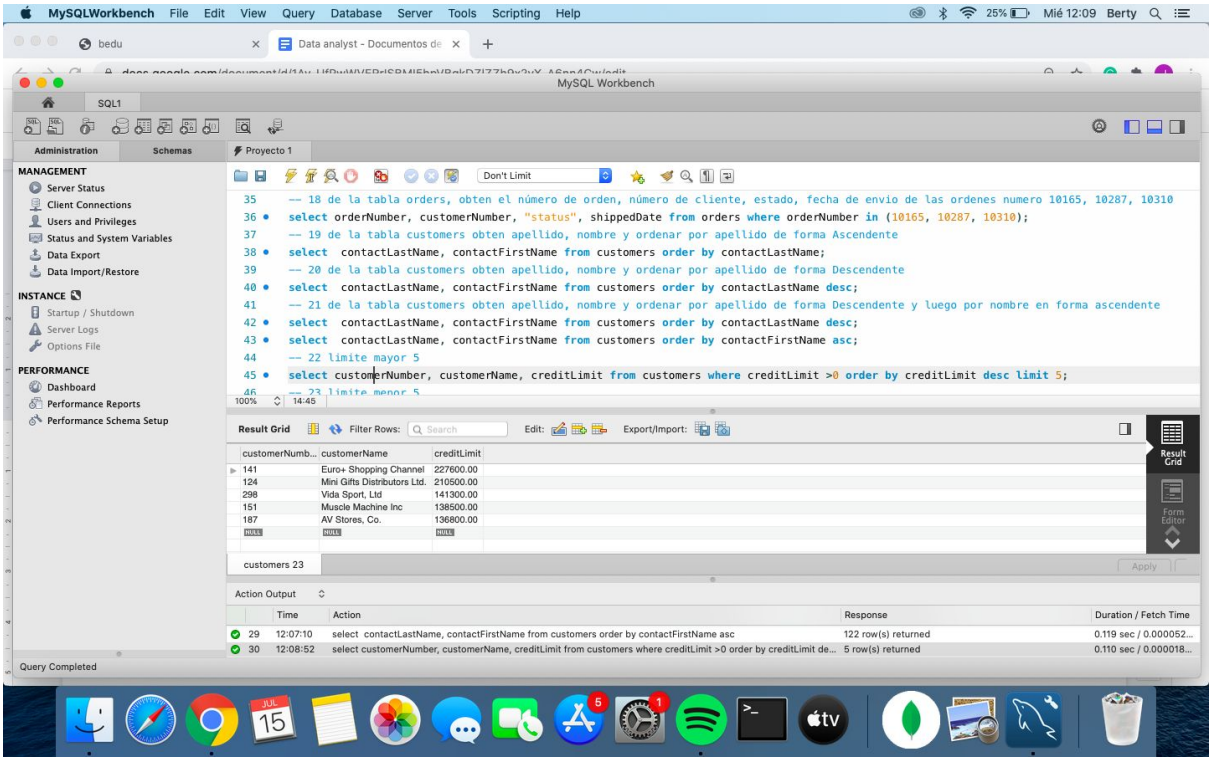
```
select  contactLastName, contactFirstName from customers order by contactLastName desc;
select  contactLastName, contactFirstName from customers order by contactFirstName asc;
```





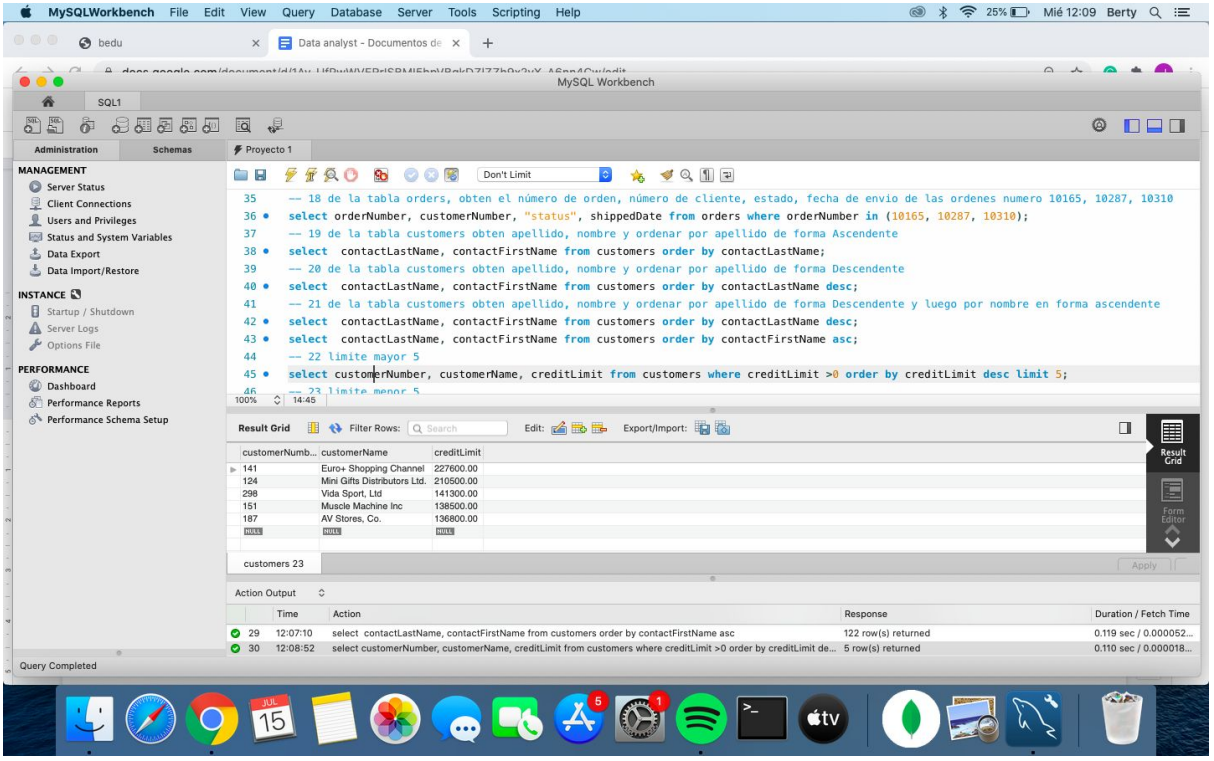
-- 22 limite mayor 5

```
select customerNumber, customerName, creditLimit from customers where creditLimit >0
order by creditLimit desc limit 5;
```



-- 23 limite menor 5

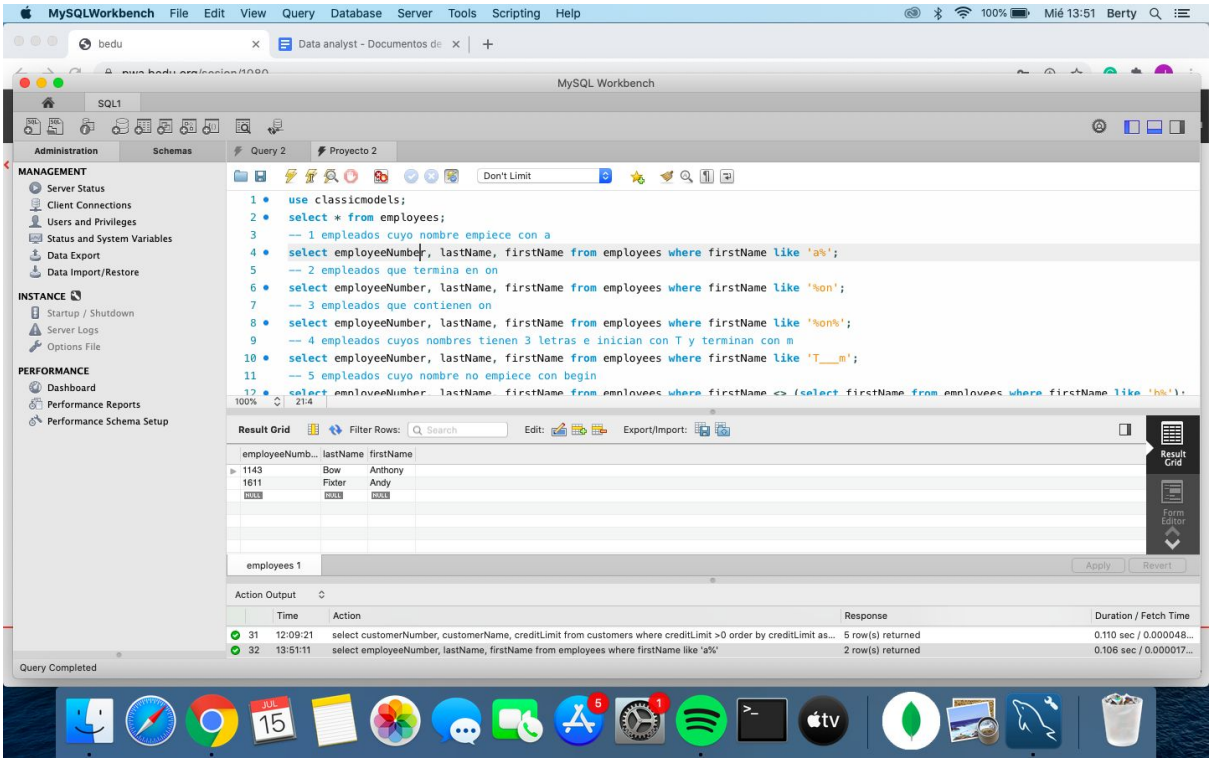
```
select customerNumber, customerName, creditLimit from customers where creditLimit >0
order by creditLimit asc limit 5;
```



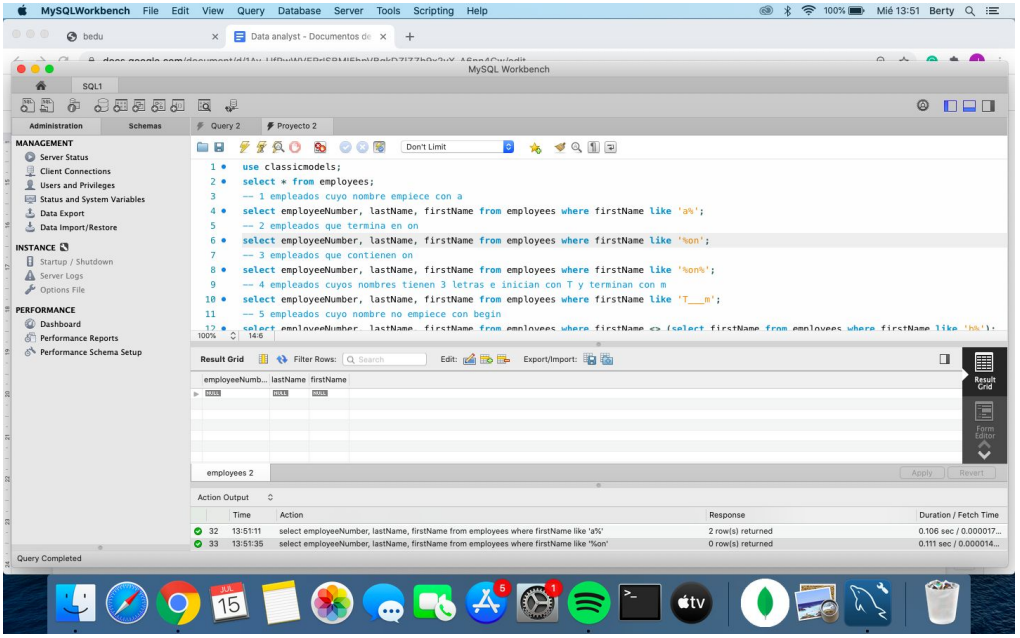
Sesión 02 Agrupaciones y subconsultas

Proyecto 2

-- 1 empleados cuyo nombre empiece con a
select employeeNumber, lastName, firstName from employees where firstName like 'a%';

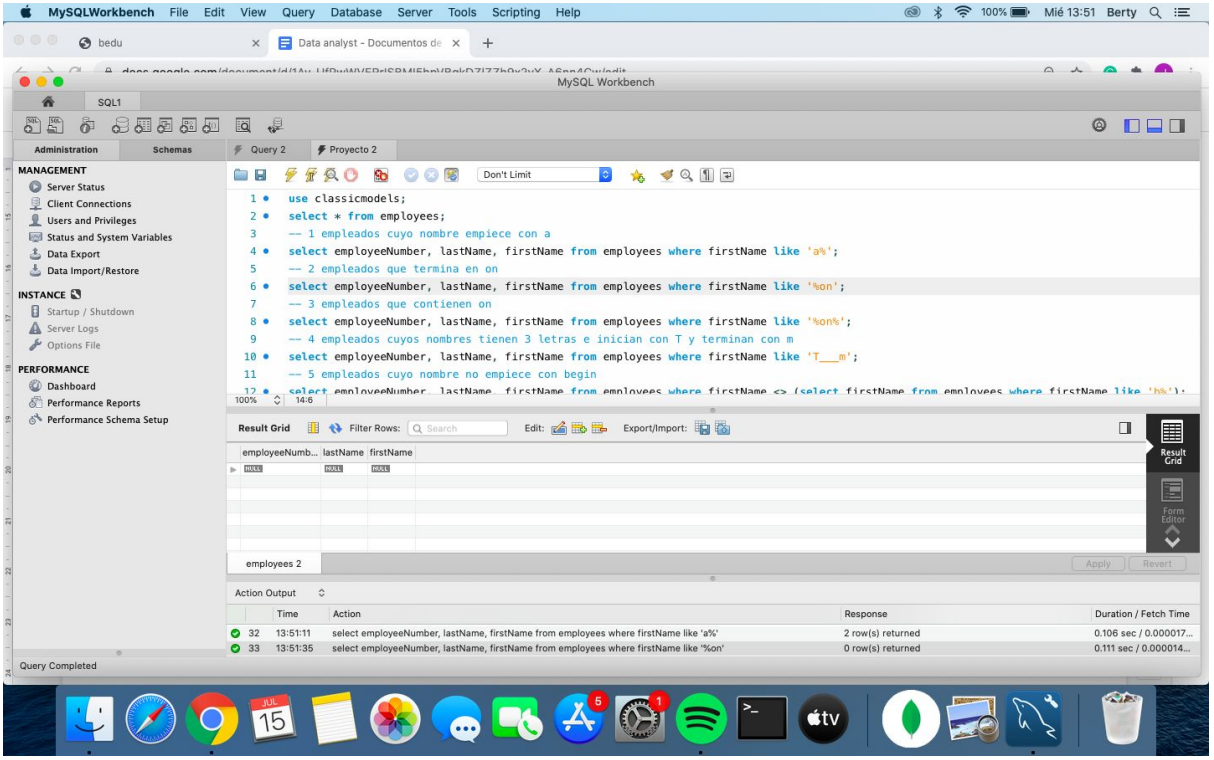


-- 2 empleados que termina en on
select employeeNumber, lastName, firstName from employees where firstName like '%on';



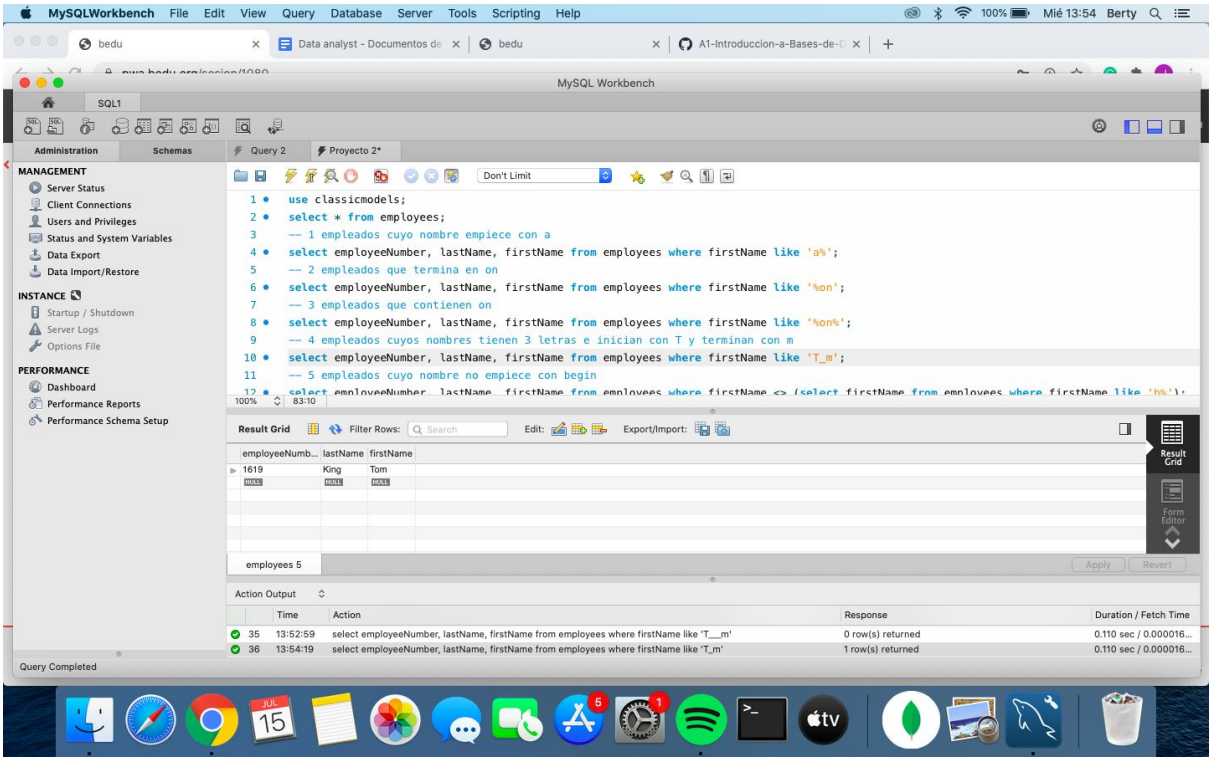
-- 3 empleados que contienen on

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



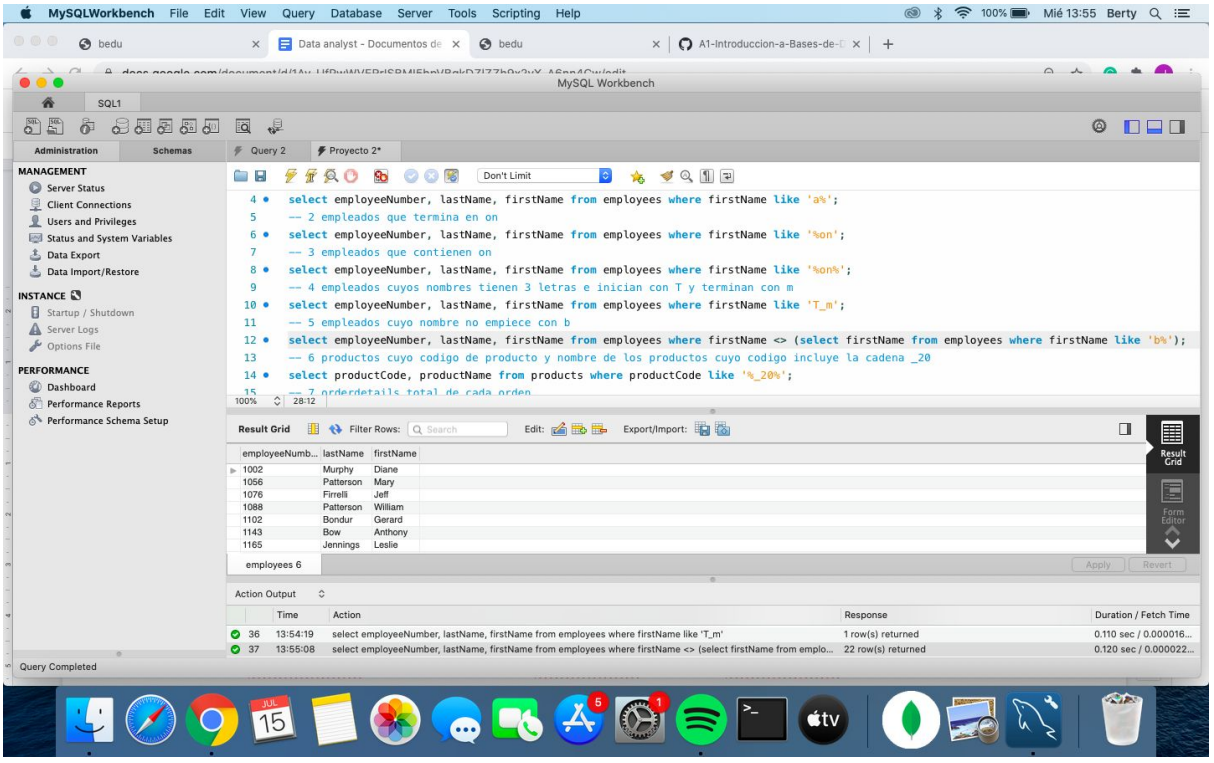
-- 4 empleados cuyos nombres tienen 3 letras e inician con T y terminan con m

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



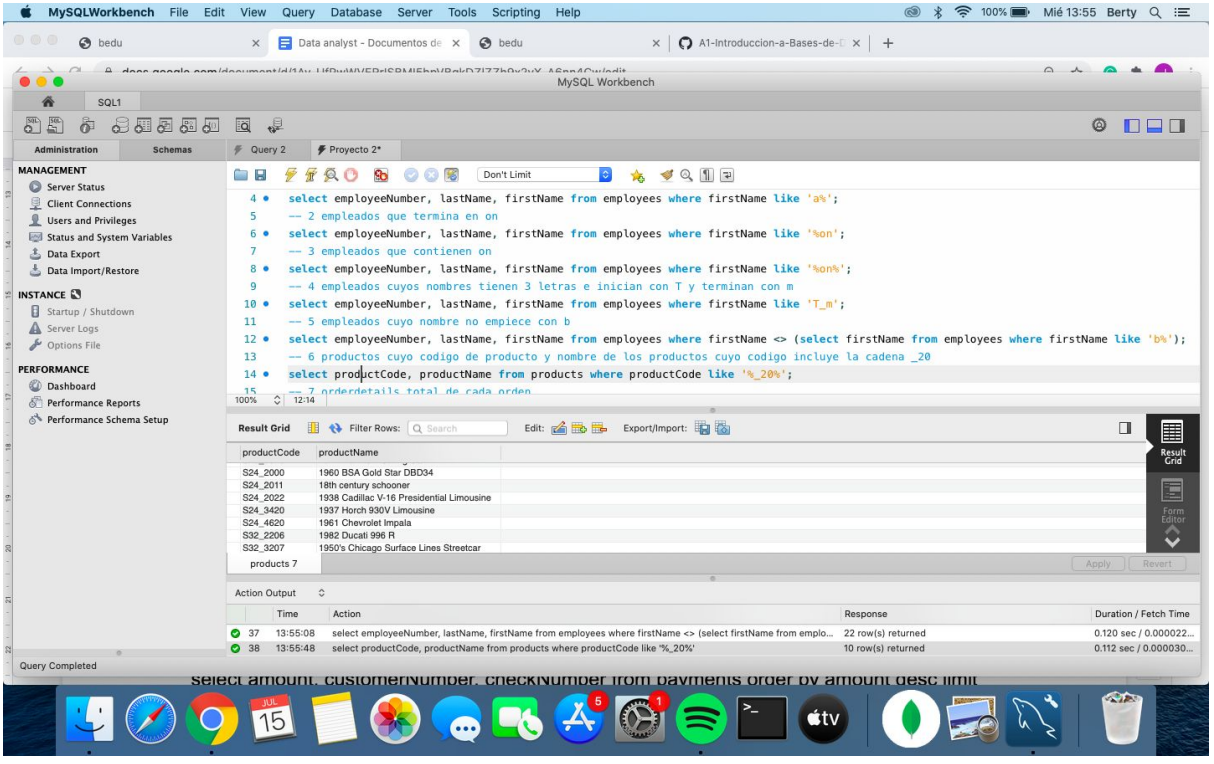
-- 5 empleados cuyo nombre no empiece con b

```
select employeeNumber, lastName, firstName from employees where firstName <> (select firstName from employees where firstName like 'b%');
```



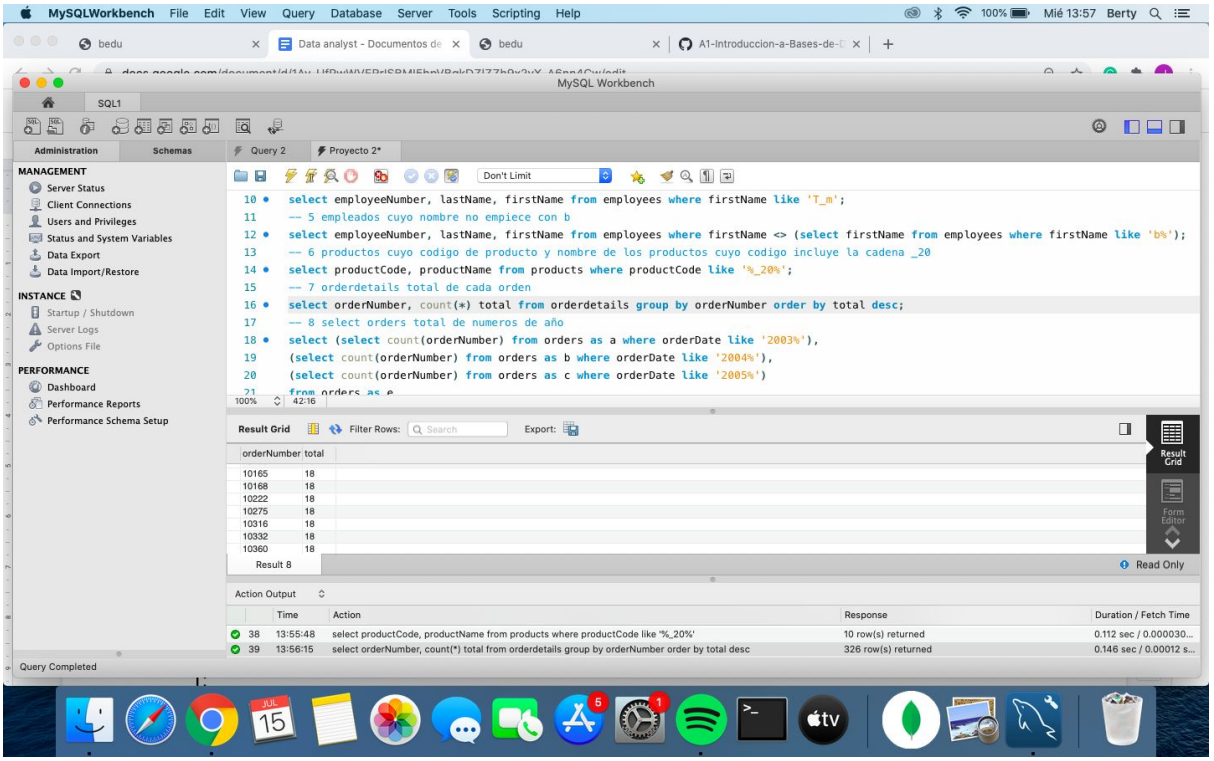
-- 6 productos cuyo codigo de producto y nombre de los productos cuyo codigo incluye la cadena _20

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



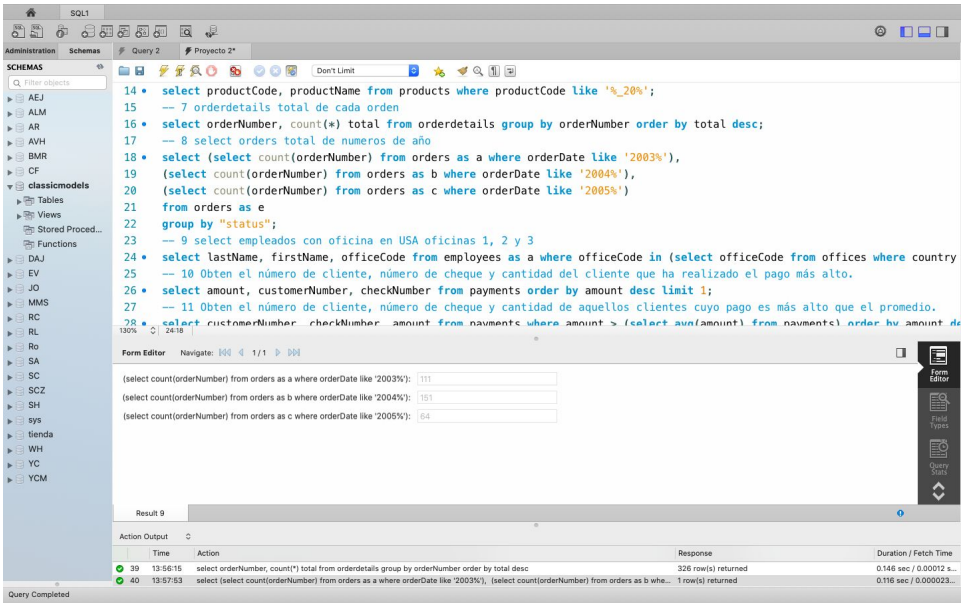
-- 7 orderdetails total de cada orden

```
select orderNumber, count(*) total from orderdetails group by orderNumber order by total desc;
```



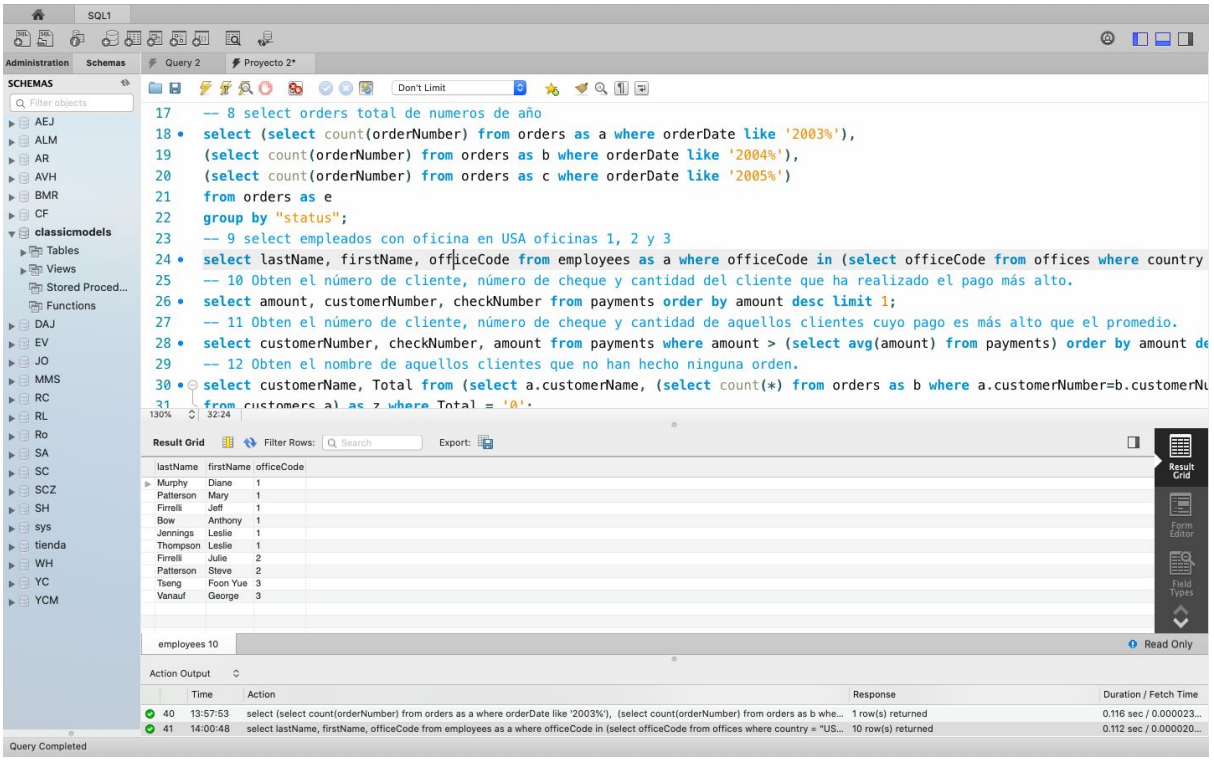
-- 8 select orders total de numeros de año

```
select (select count(orderNumber) from orders as a where orderDate like '2003%'),
(select count(orderNumber) from orders as b where orderDate like '2004%'),
(select count(orderNumber) from orders as c where orderDate like '2005%')
from orders as e
group by "status";
```



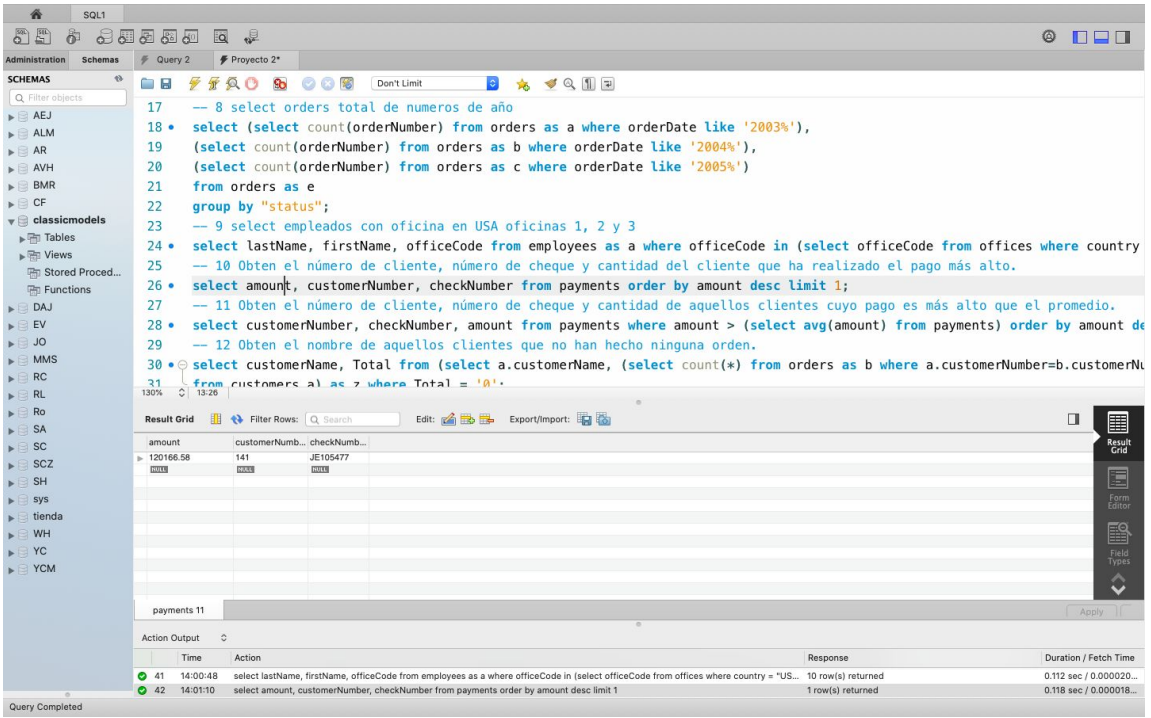
-- 9 select empleados con oficina en USA oficinas 1, 2 y 3

```
select lastName, firstName, officeCode from employees as a where officeCode in (select officeCode from offices where country = "USA");
```



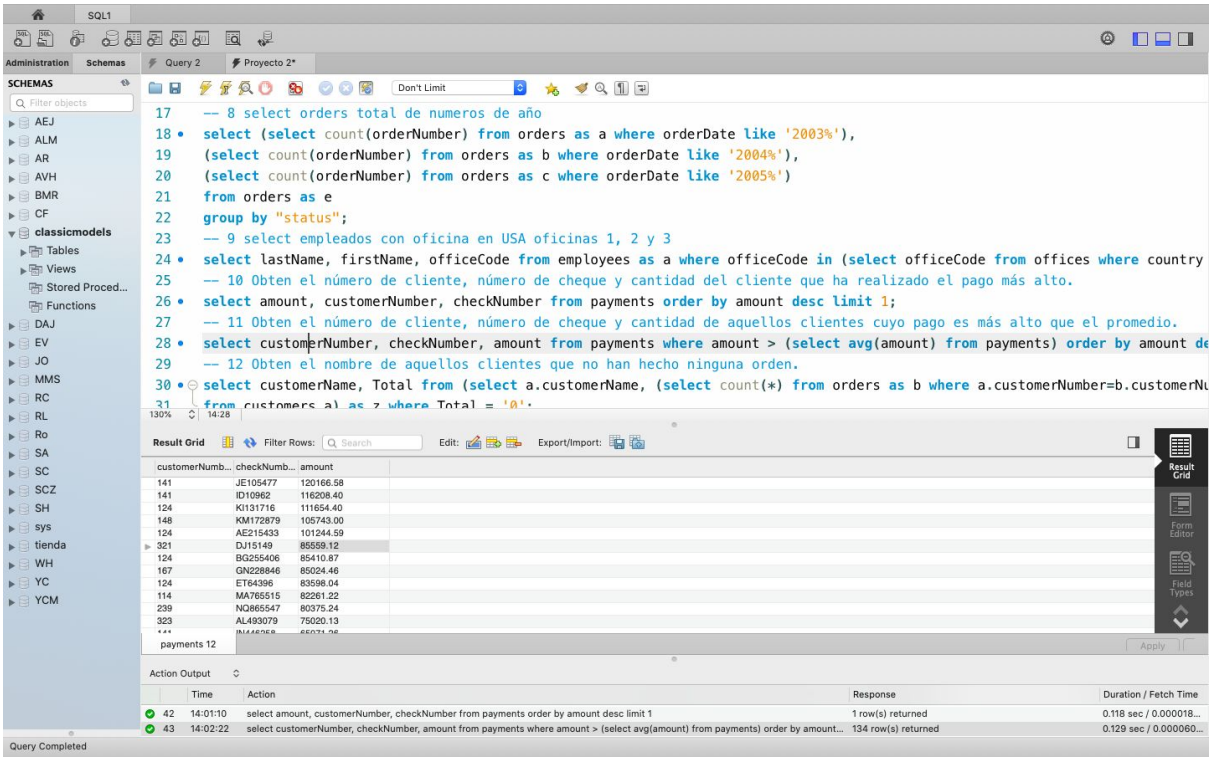
-- 10 Obten el número de cliente, número de cheque y cantidad del cliente que ha realizado el pago más alto.

```
select amount, customerNumber, checkNumber from payments order by amount desc limit 1;
```



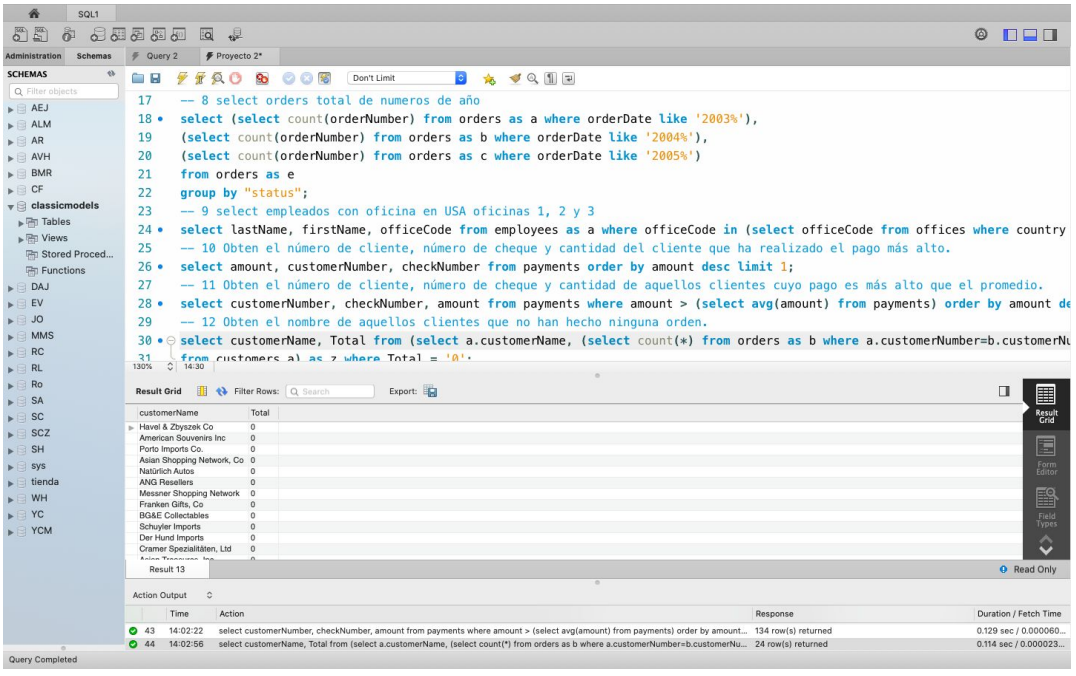
-- 11 Obten el número de cliente, número de cheque y cantidad de aquellos clientes cuyo pago es más alto que el promedio.

```
select customerNumber, checkNumber, amount from payments where amount > (select avg(amount) from payments) order by amount desc;
```



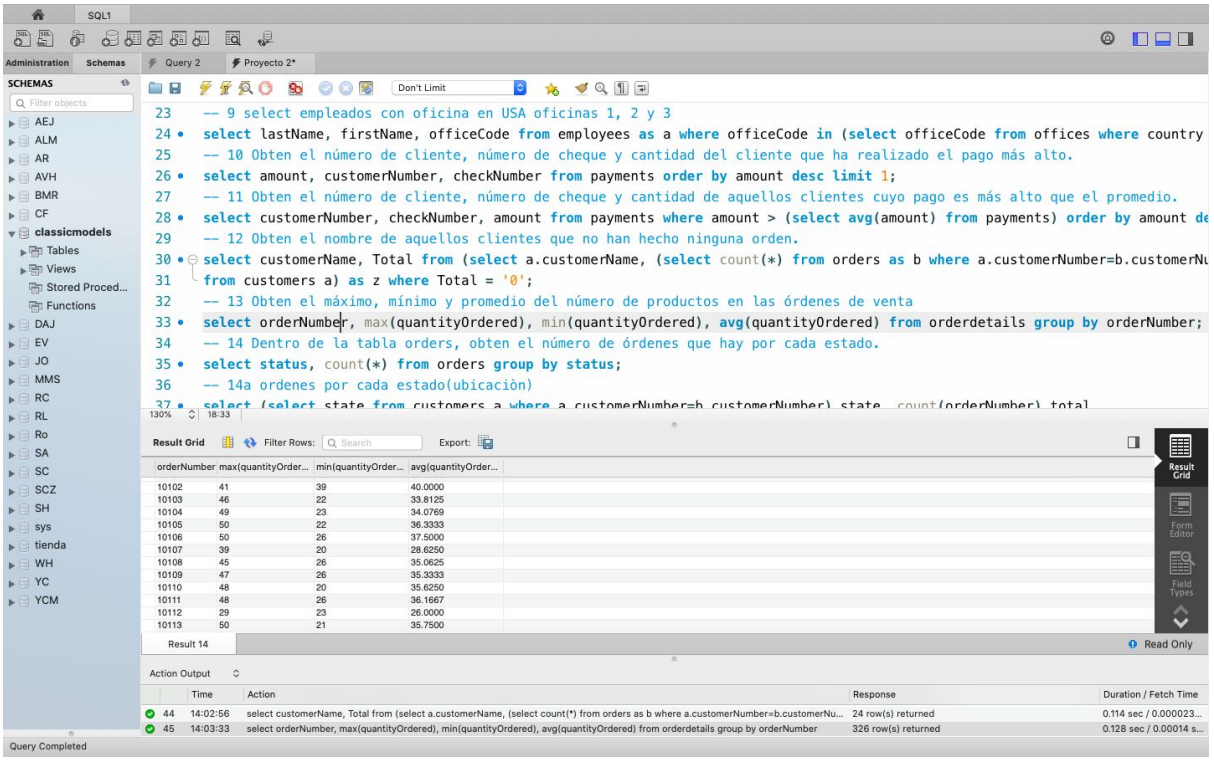
-- 12 Obten el nombre de aquellos clientes que no han hecho ninguna orden.

```
select customerName, Total from (select a.customerName, (select count(*) from orders as b
where a.customerNumber=b.customerNumber) Total
from customers a) as z where Total = '0';
```



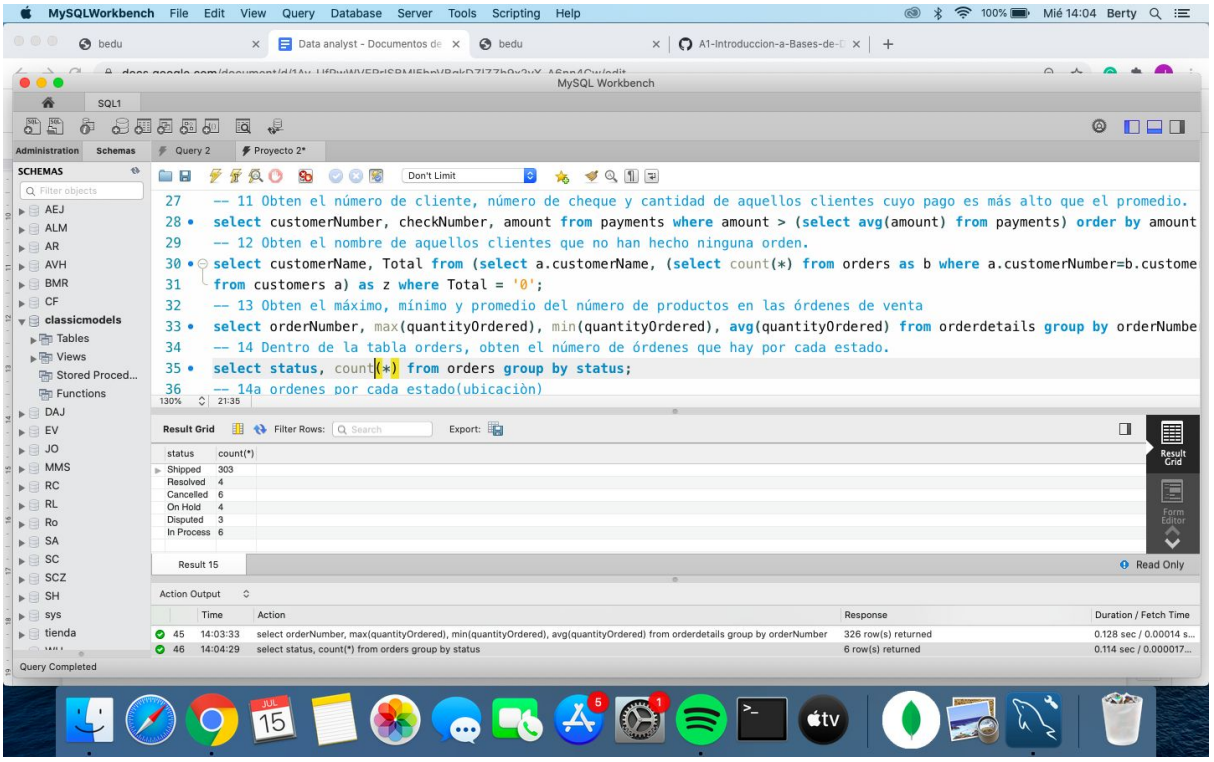
-- 13 Obten el máximo, mínimo y promedio del número de productos en las órdenes de venta

```
select orderNumber, max(quantityOrdered), min(quantityOrdered), avg(quantityOrdered)
from orderdetails group by orderNumber;
```



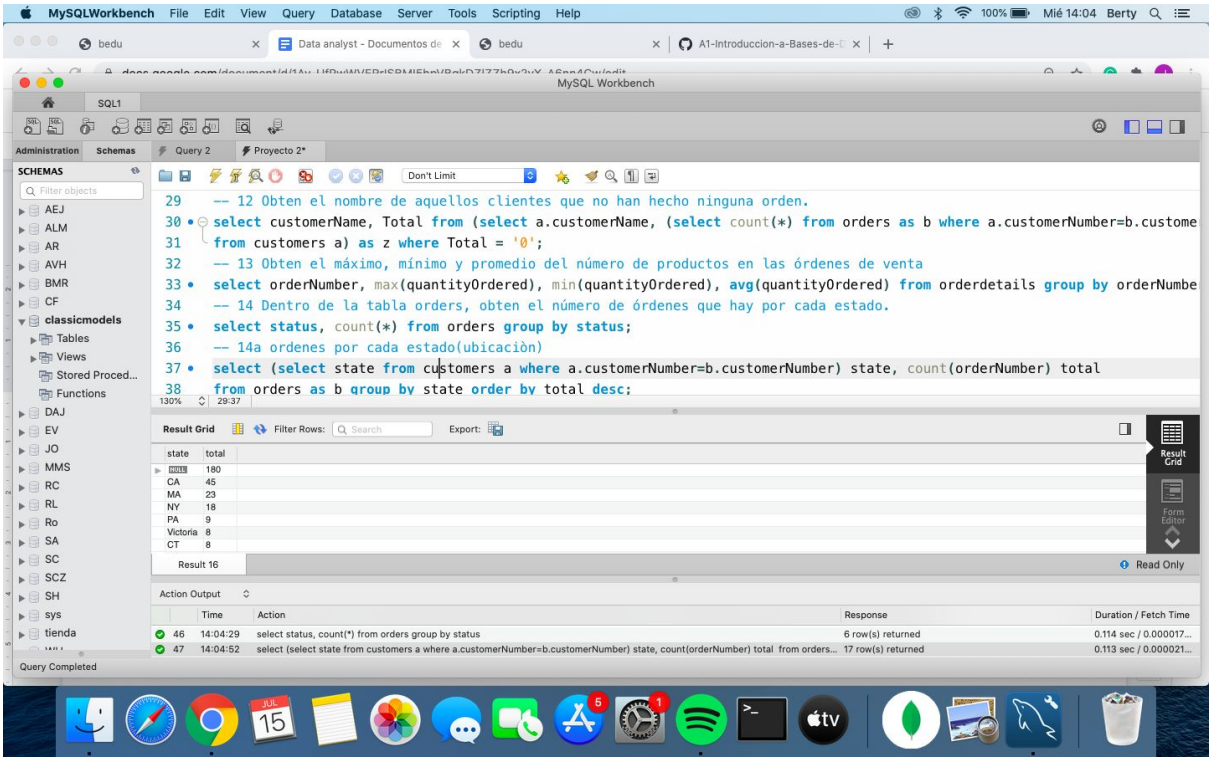
-- 14 Dentro de la tabla orders, obten el número de órdenes que hay por cada estado.

```
select status, count(*) from orders group by status;
```



-- 14a ordenes por cada estado(ubicación)

```
select (select state from customers a where a.customerNumber=b.customerNumber) state,
count(orderNumber) total
from orders as b group by state order by total desc;
```

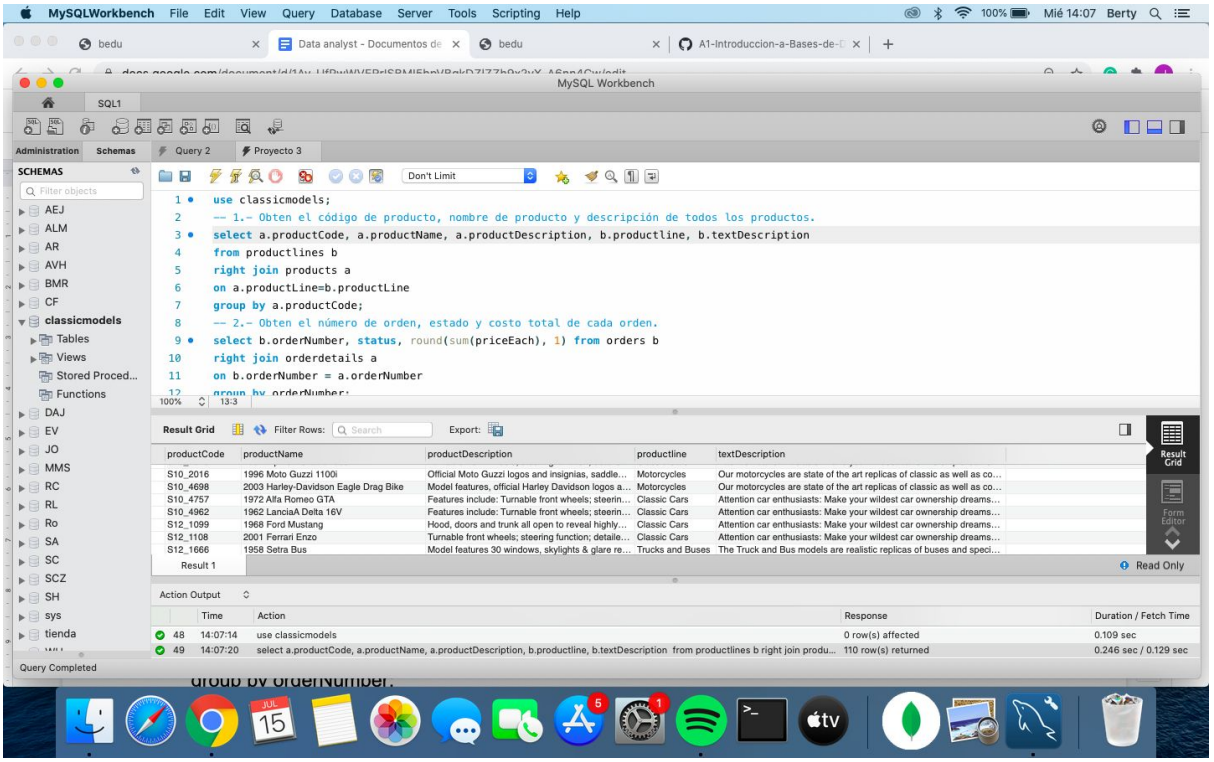


Sesión 03 Agrupaciones y consultas

Proyecto 3

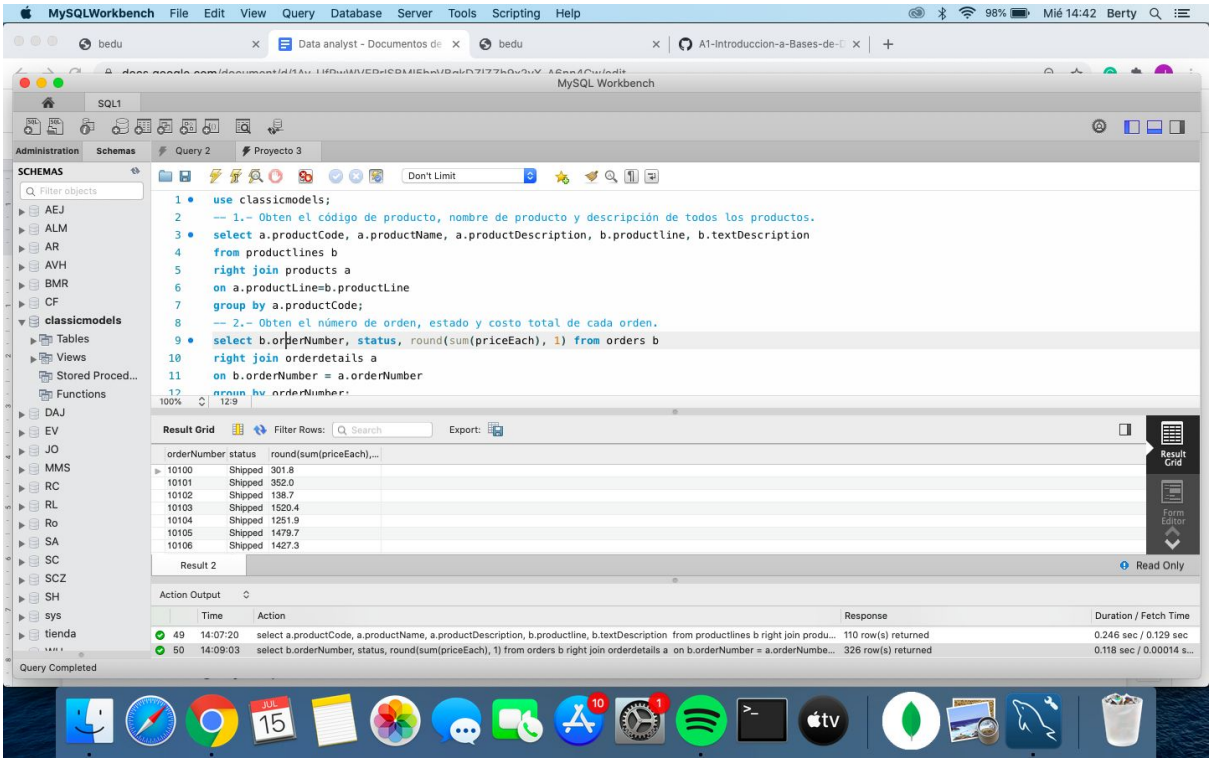
-- 1.- Obten el código de producto, nombre de producto y descripción de todos los productos.

```
select a.productCode, a.productName, a.productDescription, b.productline, b.textDescription
from productlines b
right join products a
on a.productLine=b.productLine
group by a.productCode;
```



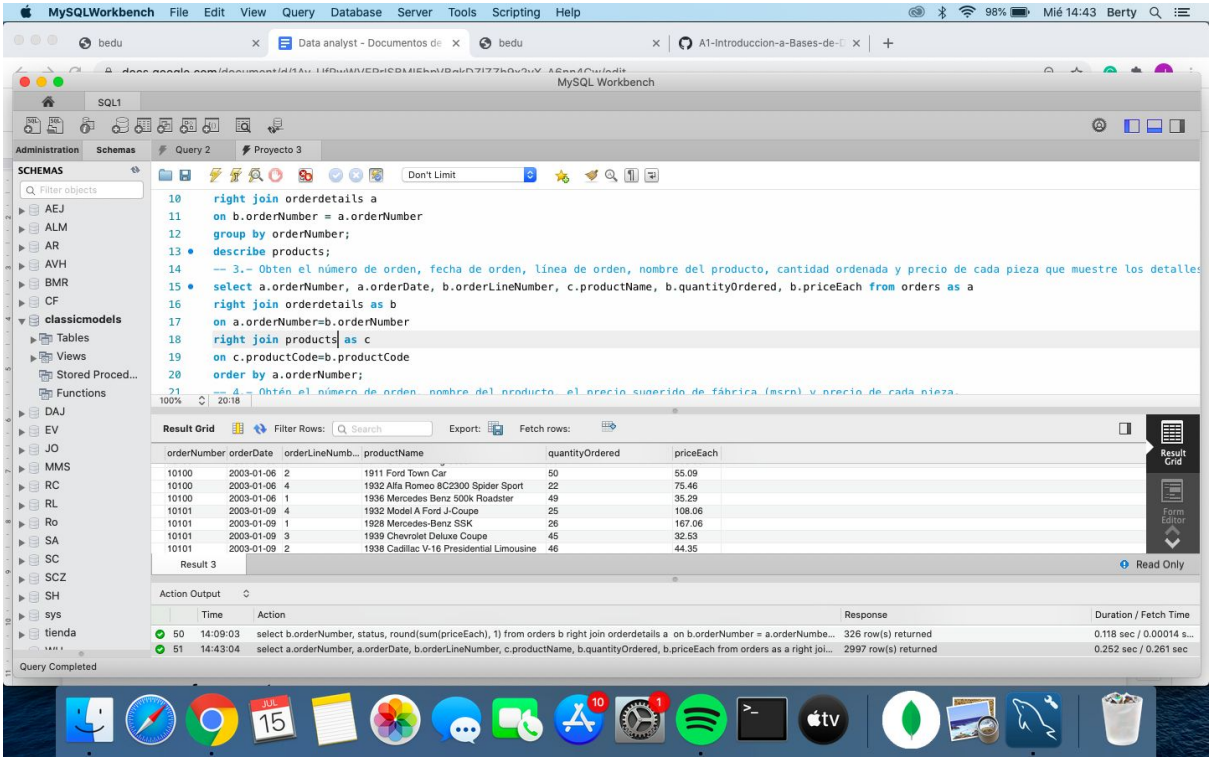
-- 2.- Obten el número de orden, estado y costo total de cada orden.

```
select b.orderNumber, status, round(sum(priceEach), 1) from orders b
right join orderdetails a
on b.orderNumber = a.orderNumber
group by orderNumber;
describe products;
```



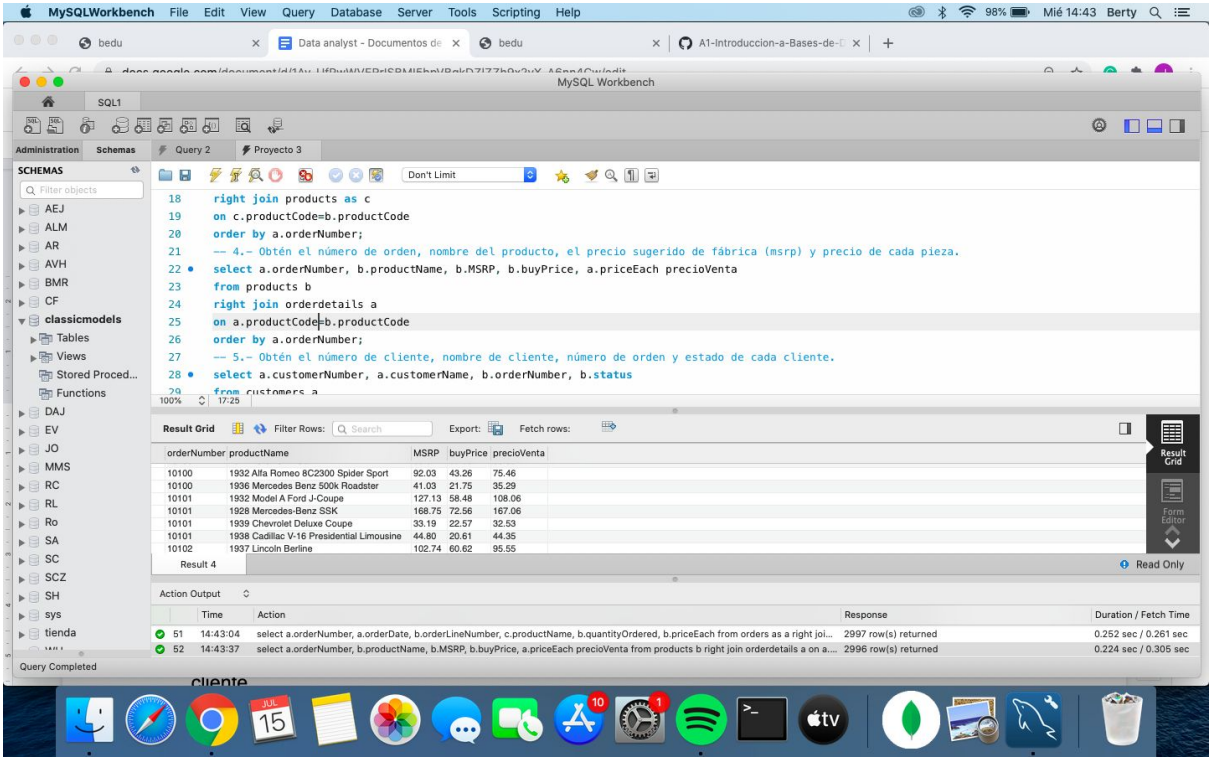
-- 3.- Obten el número de orden, fecha de orden, línea de orden, nombre del producto, cantidad ordenada y precio de cada pieza que muestre los detalles de cada orden.

```
select a.orderNumber, a.orderDate, b.orderLineNumber, c.productName, b.quantityOrdered,
b.priceEach from orders as a
right join orderdetails as b
on a.orderNumber=b.orderNumber
right join products as c
on c.productCode=b.productCode
order by a.orderNumber;
```



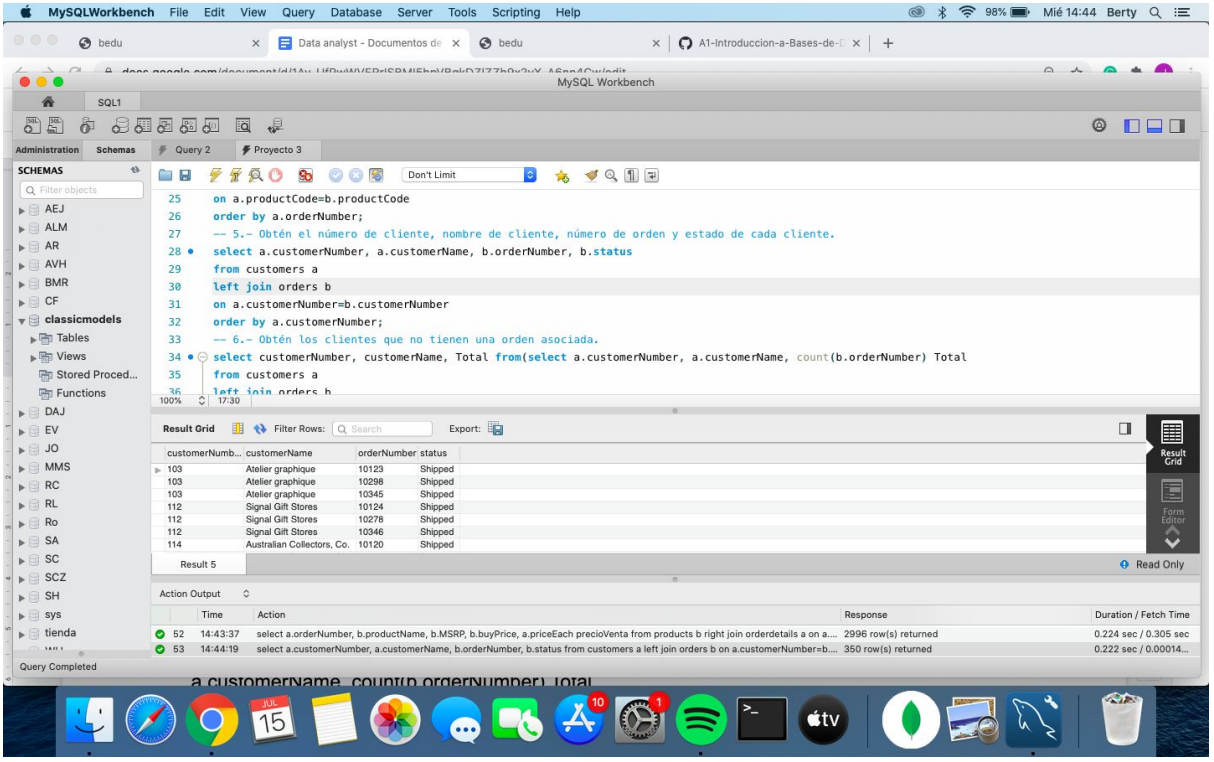
-- 4.- Obtén el número de orden, nombre del producto, el precio sugerido de fábrica (msrp) y precio de cada pieza.

```
select a.orderNumber, b.productName, b.MSRP, b.buyPrice, a.priceEach precioVenta
from products b
right join orderdetails a
on a.productCode=b.productCode
order by a.orderNumber;
```



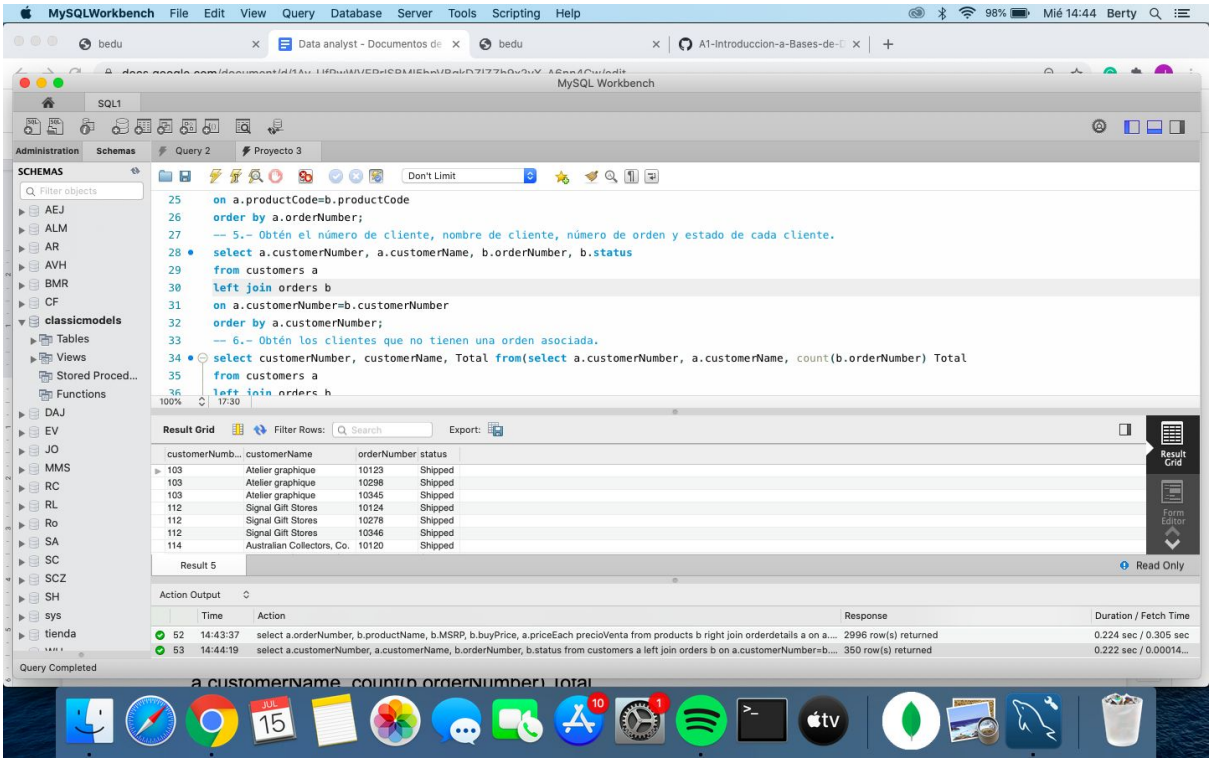
-- 5.- Obtén el número de cliente, nombre de cliente, número de orden y estado de cada cliente.

```
select a.customerNumber, a.customerName, b.orderNumber, b.status
from customers a
left join orders b
on a.customerNumber=b.customerNumber
order by a.customerNumber;
```



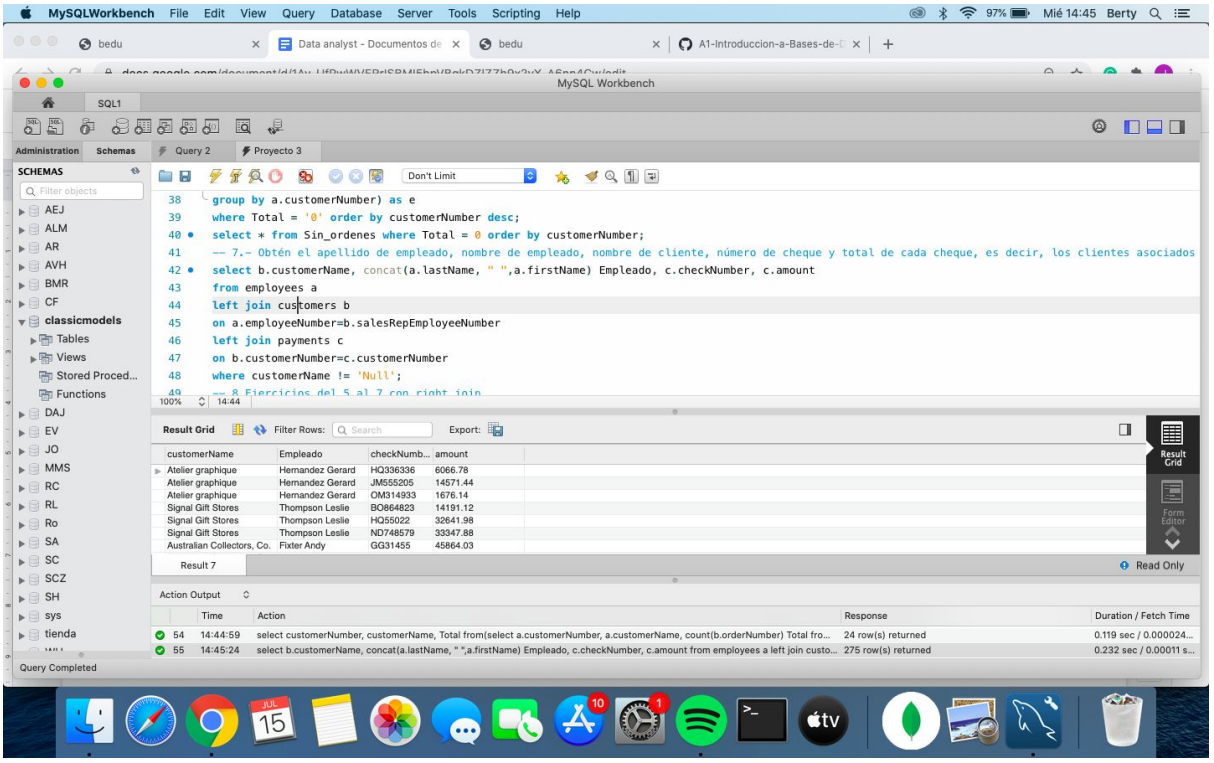
-- 6.- Obtén los clientes que no tienen una orden asociada.

```
select customerNumber, customerName, Total from(select a.customerNumber,
a.customerName, count(b.orderNumber) Total
from customers a
left join orders b
on a.customerNumber=b.customerNumber
group by a.customerNumber) as e
where Total = '0' order by customerNumber desc;
select * from Sin_ordenes where Total = 0 order by customerNumber;
```



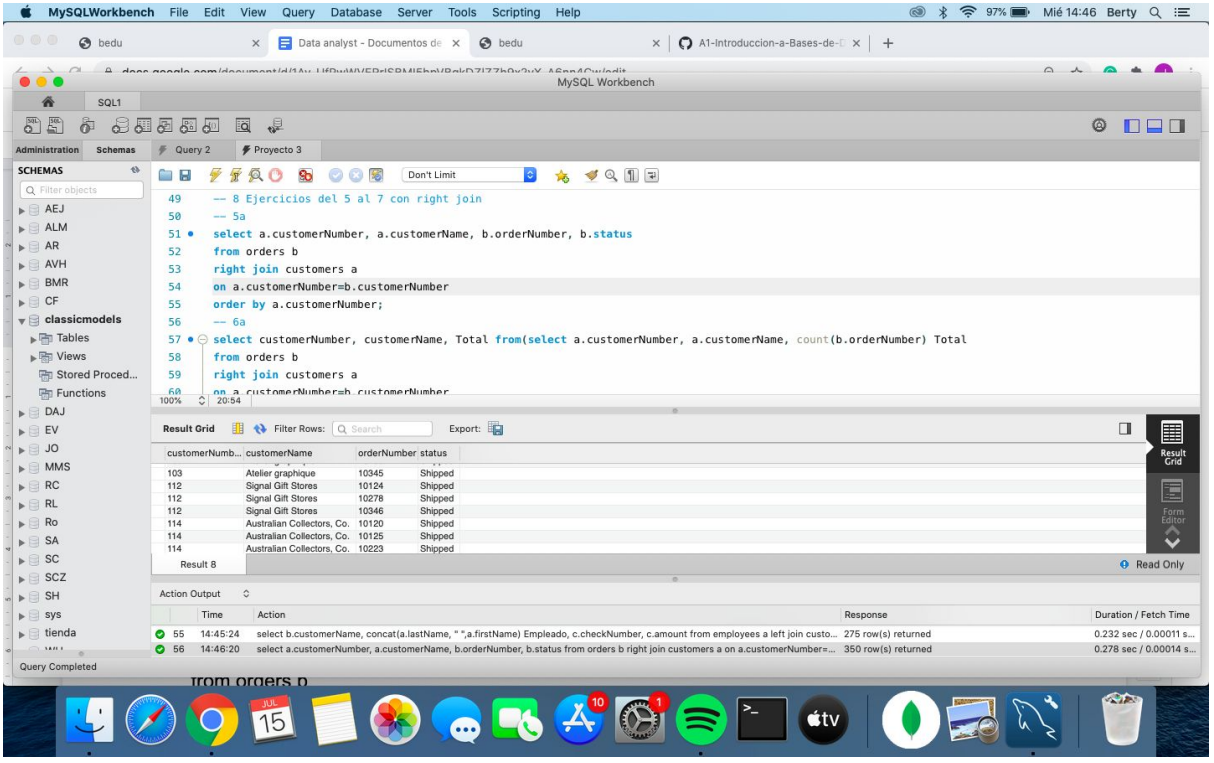
-- 7.- Obtén el apellido de empleado, nombre de empleado, nombre de cliente, número de cheque y total de cada cheque, es decir, los clientes asociados a cada empleado.

```
select b.customerName, concat(a.lastName, " ",a.firstName) Empleado, c.checkNumber,
c.amount
from employees a
left join customers b
on a.employeeNumber=b.salesRepEmployeeNumber
left join payments c
on b.customerNumber=c.customerNumber
where customerName != 'Null';
```



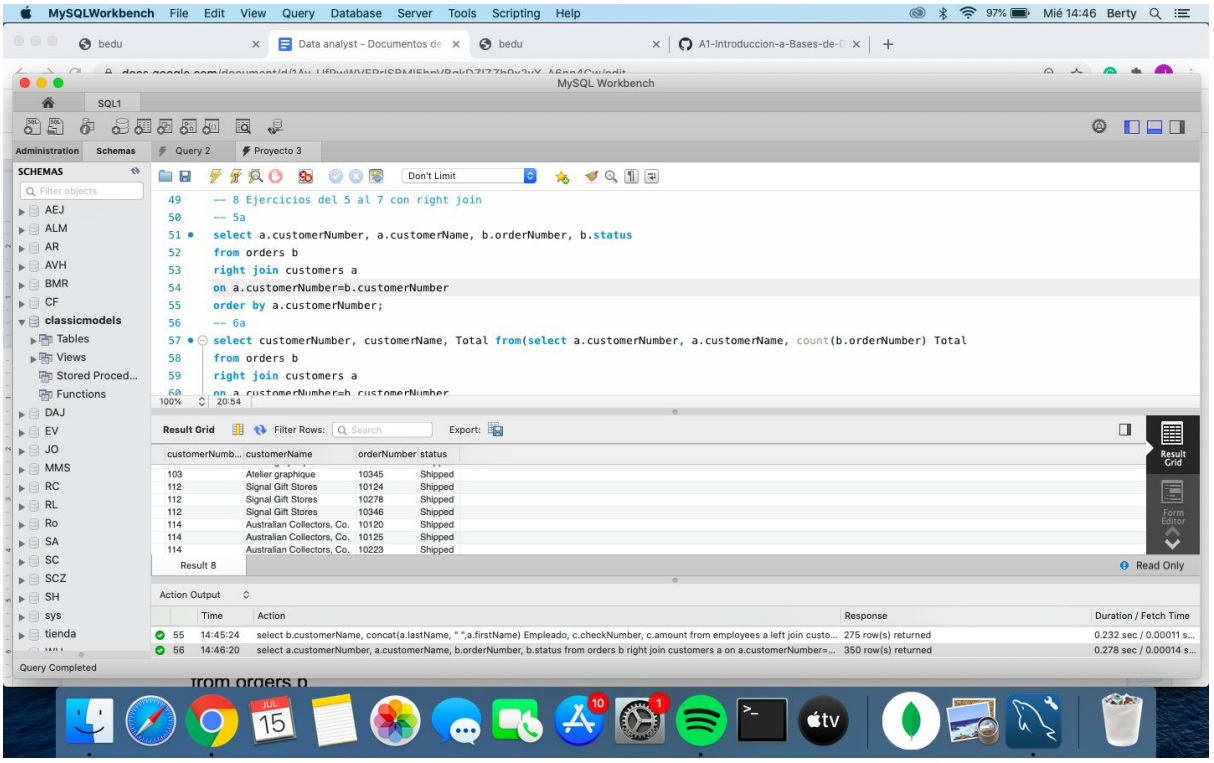
-- 8 Ejercicios del 5 al 7 con right join
-- 5a

```
select a.customerNumber, a.customerName, b.orderNumber, b.status
from orders b
right join customers a
on a.customerNumber=b.customerNumber
order by a.customerNumber;
```

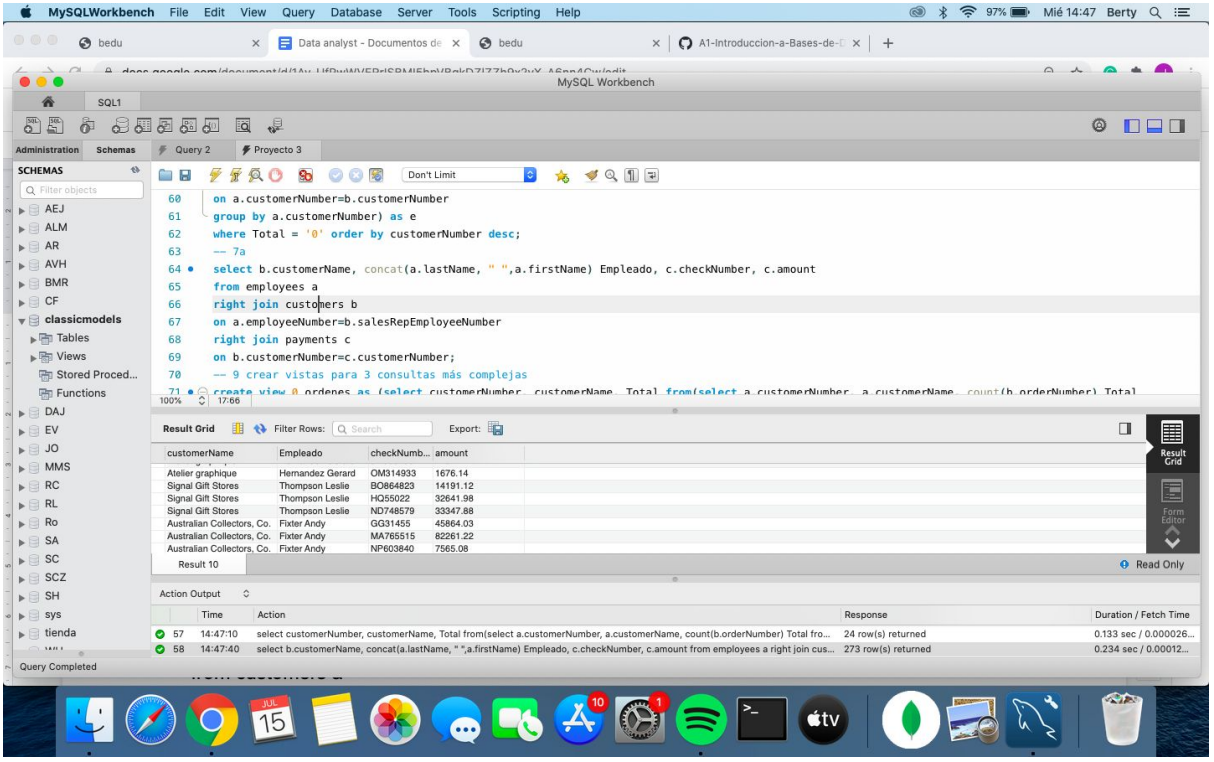


-- 6a

```
select customerNumber, customerName, Total from(select a.customerNumber,
a.customerName, count(b.orderNumber) Total
from orders b
right join customers a
on a.customerNumber=b.customerNumber
group by a.customerNumber) as e
where Total = '0' order by customerNumber desc;
```

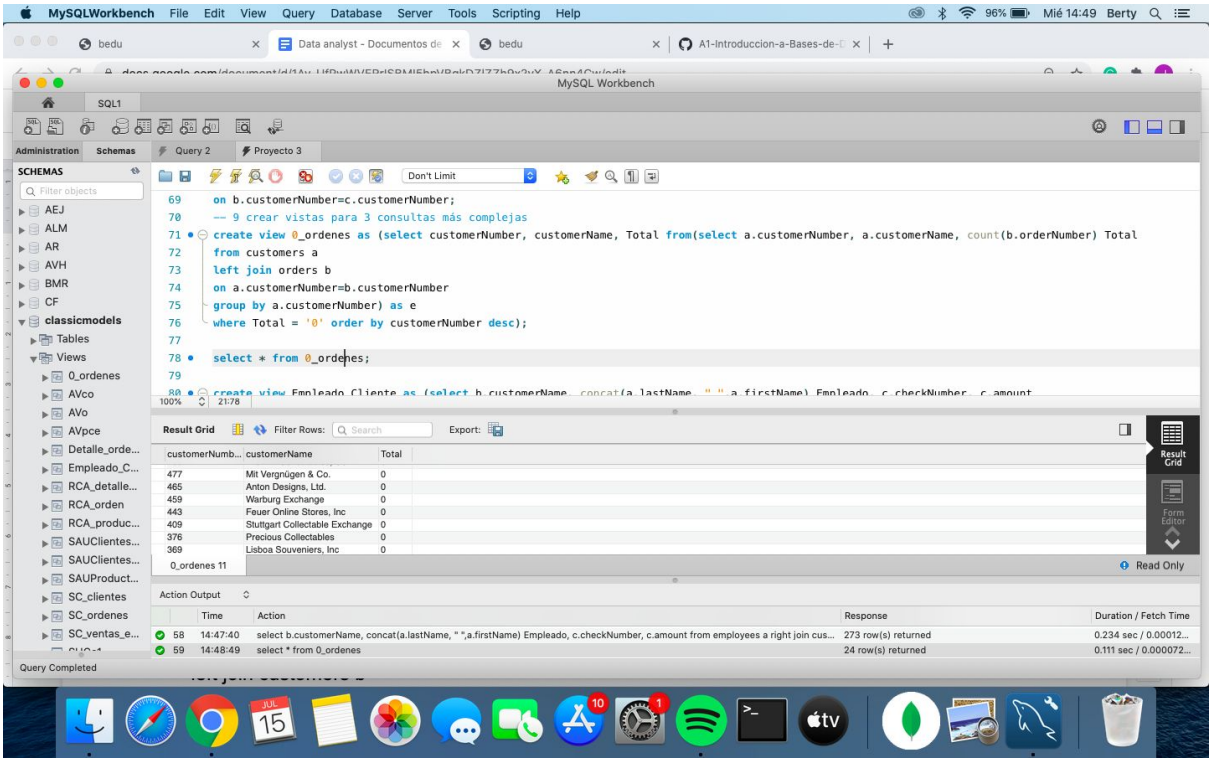


```
-- 7a
select b.customerName, concat(a.lastName, " ",a.firstName) Empleado, c.checkNumber,
c.amount
from employees a
right join customers b
on a.employeeNumber=b.salesRepEmployeeNumber
right join payments c
on b.customerNumber=c.customerNumber;
```



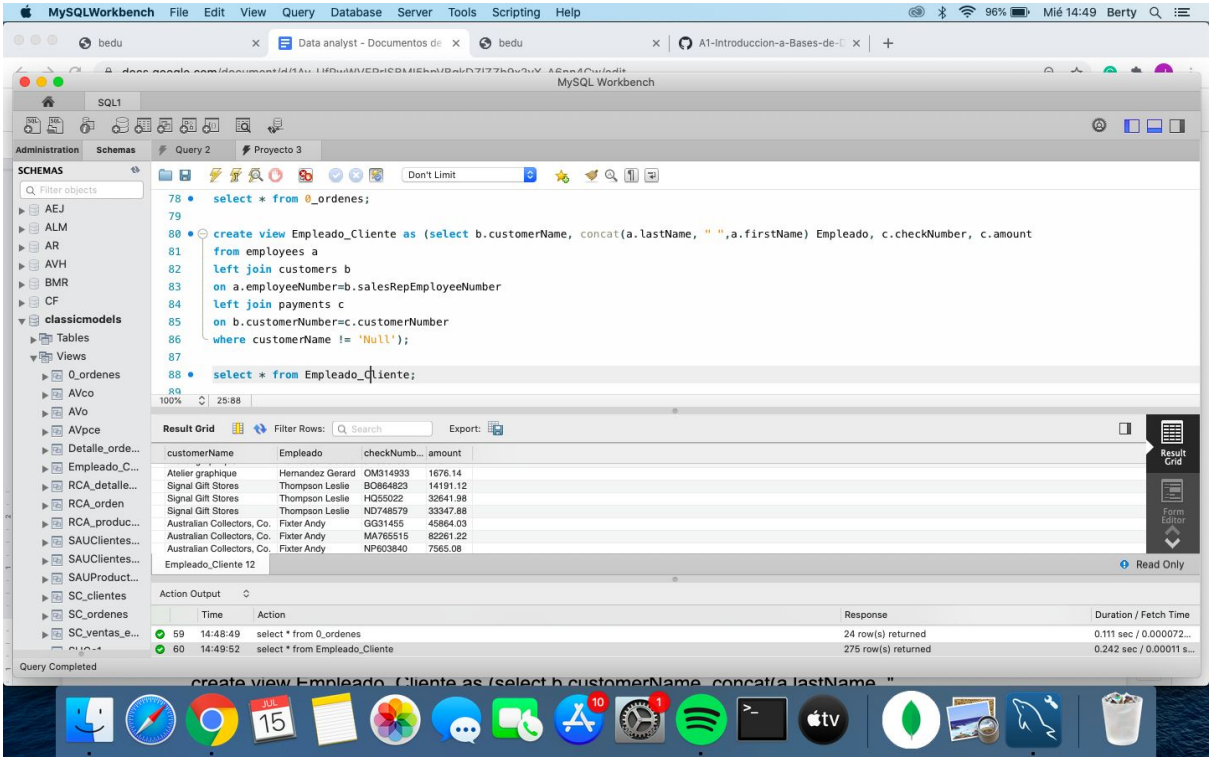
-- 9 crear vistas para 3 consultas más complejas
1:

```
create view 0_ordenes as (select customerNumber, customerName, Total from(select
a.customerNumber, a.customerName, count(b.orderNumber) Total
from customers a
left join orders b
on a.customerNumber=b.customerNumber
group by a.customerNumber) as e
where Total = '0' order by customerNumber desc);

select * from 0_ordenes;
```



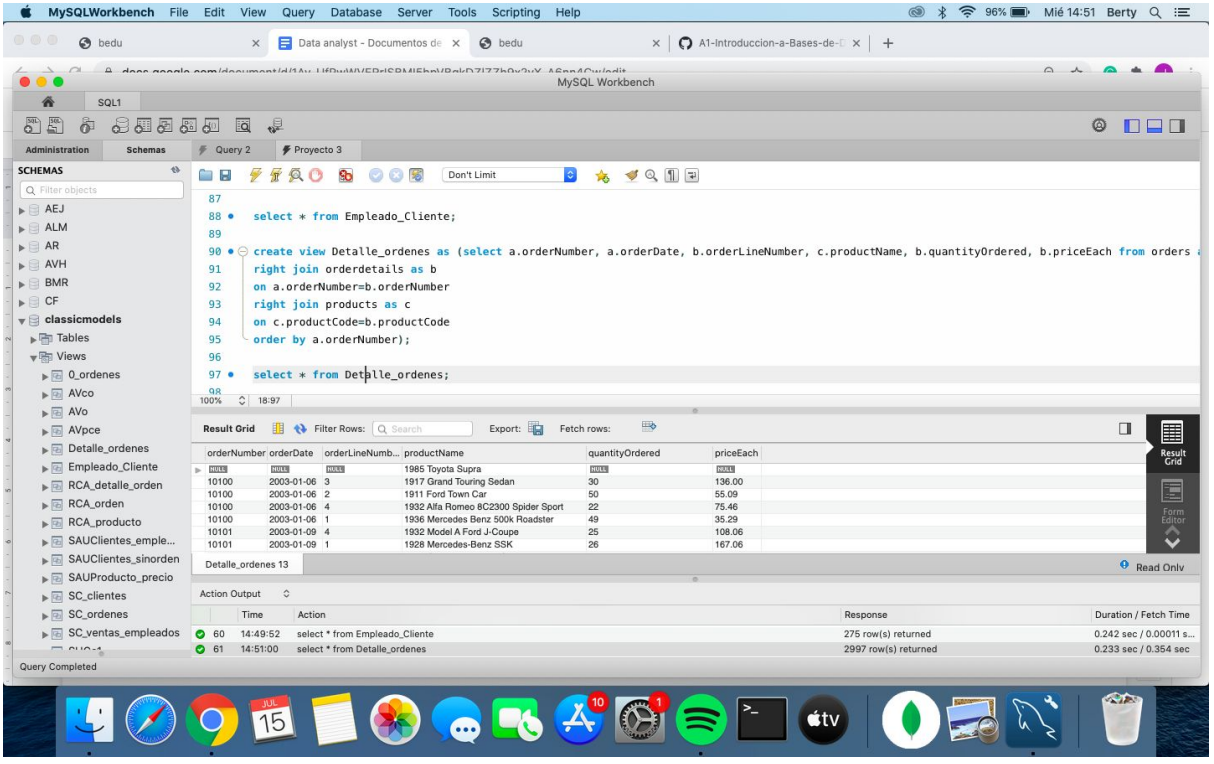
2.-

```
create view Empleado_Cliente as (select b.customerName, concat(a.lastName, "
",a.firstName) Empleado, c.checkNumber, c.amount
from employees a
left join customers b
on a.employeeNumber=b.salesRepEmployeeNumber
left join payments c
on b.customerNumber=c.customerNumber
where customerName != 'Null');

select * from Empleado_Cliente;
```



3.-

```
create view Detalle_ordenes as (select a.orderNumber, a.orderDate, b.orderLineNumber,
c.productName, b.quantityOrdered, b.priceEach from orders as a
right join orderdetails as b
on a.orderNumber=b.orderNumber
right join products as c
on c.productCode=b.productCode
order by a.orderNumber);

select * from Detalle_ordenes;
```

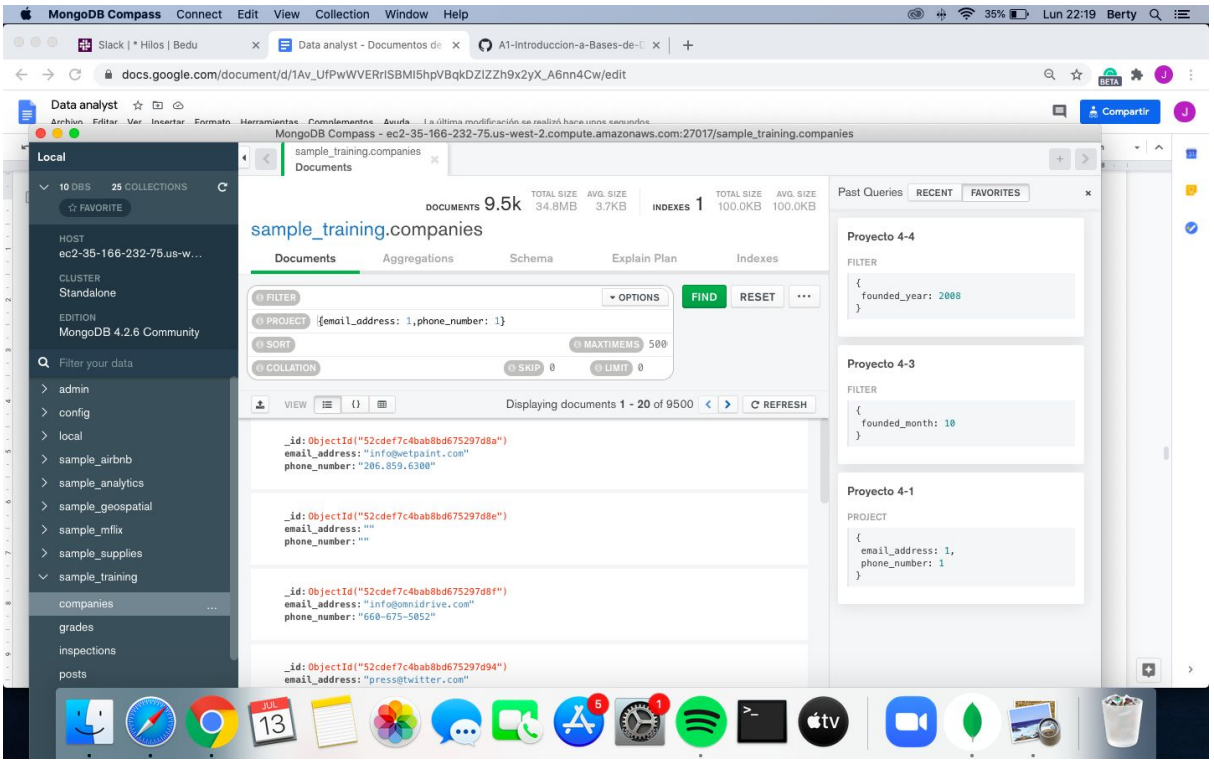


Sesión 04: Fundamentos de MongoDB

Proyecto 4

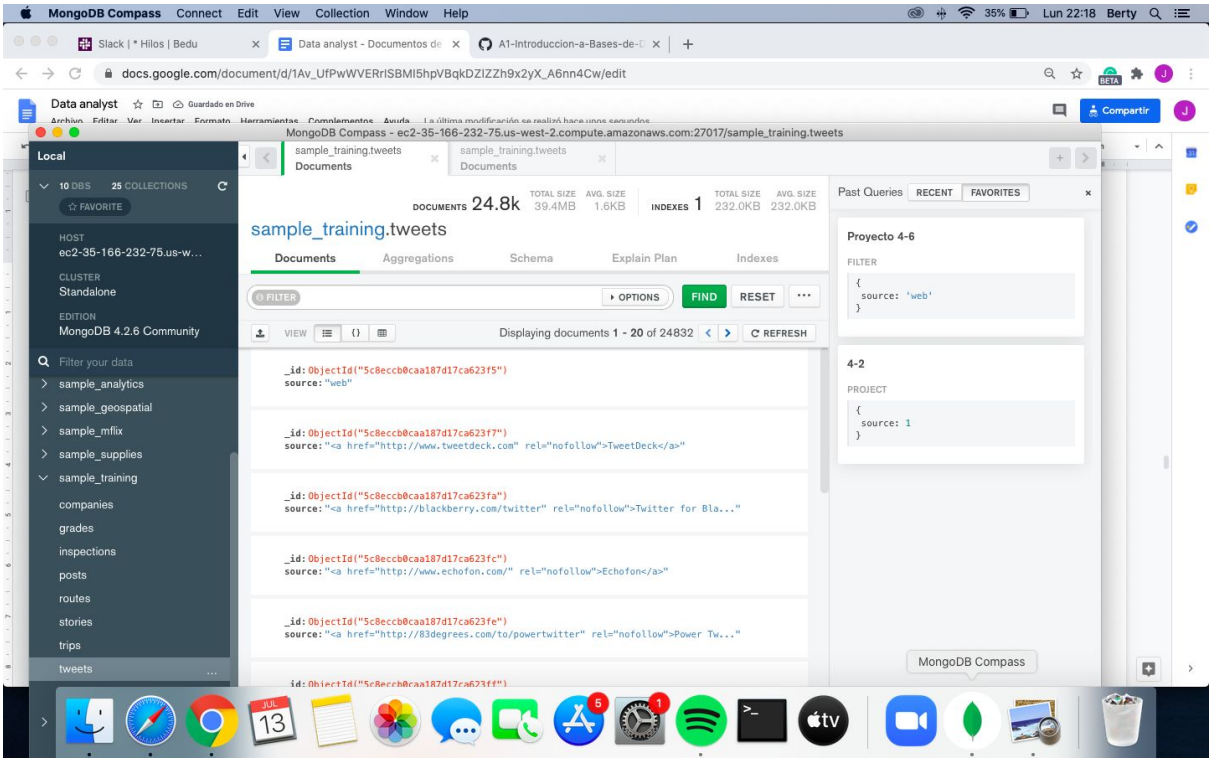
1.- Obtén los datos de contacto de cada compañía.

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



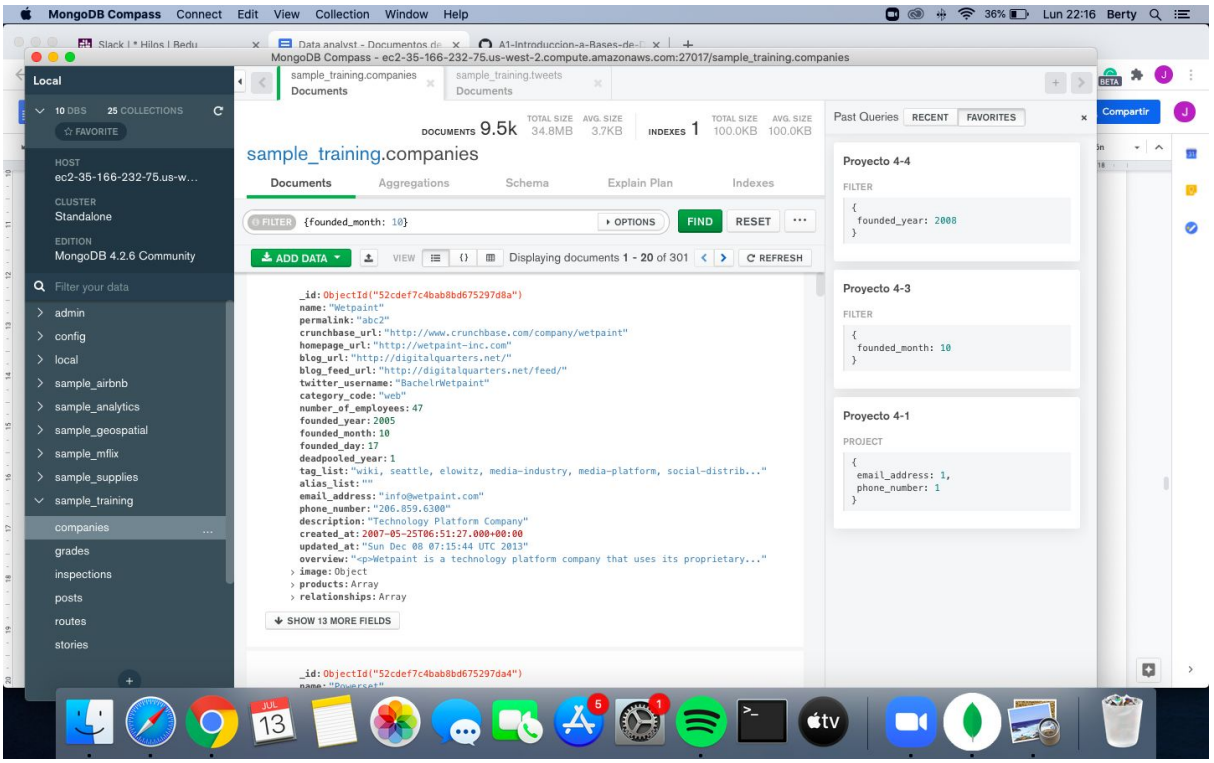
2.- Obtén la fuente de cada tweet.

```
{
 filter: {
 project: {
  source: 1
 }
}
}
```



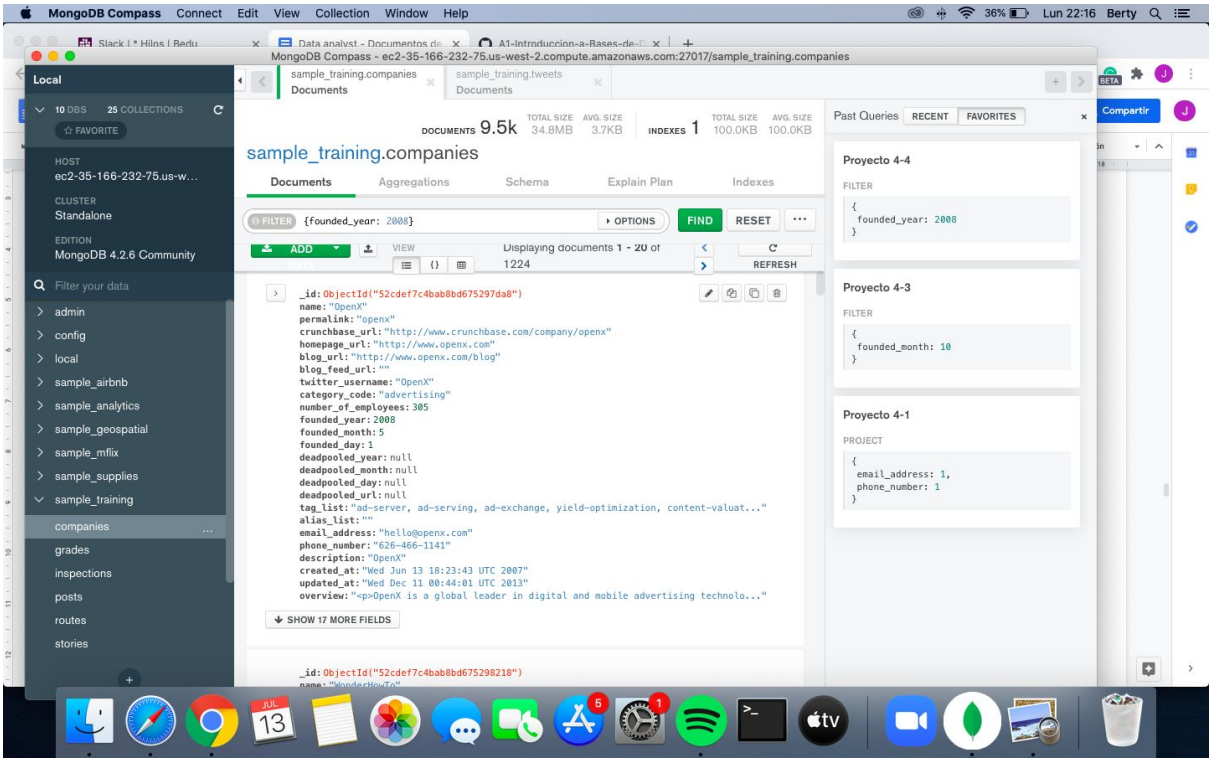
3.- Obtén el nombre de todas las compañias fundadas en octubre.

```
{
 filter: {
  founded_month: 10
 }
}
```



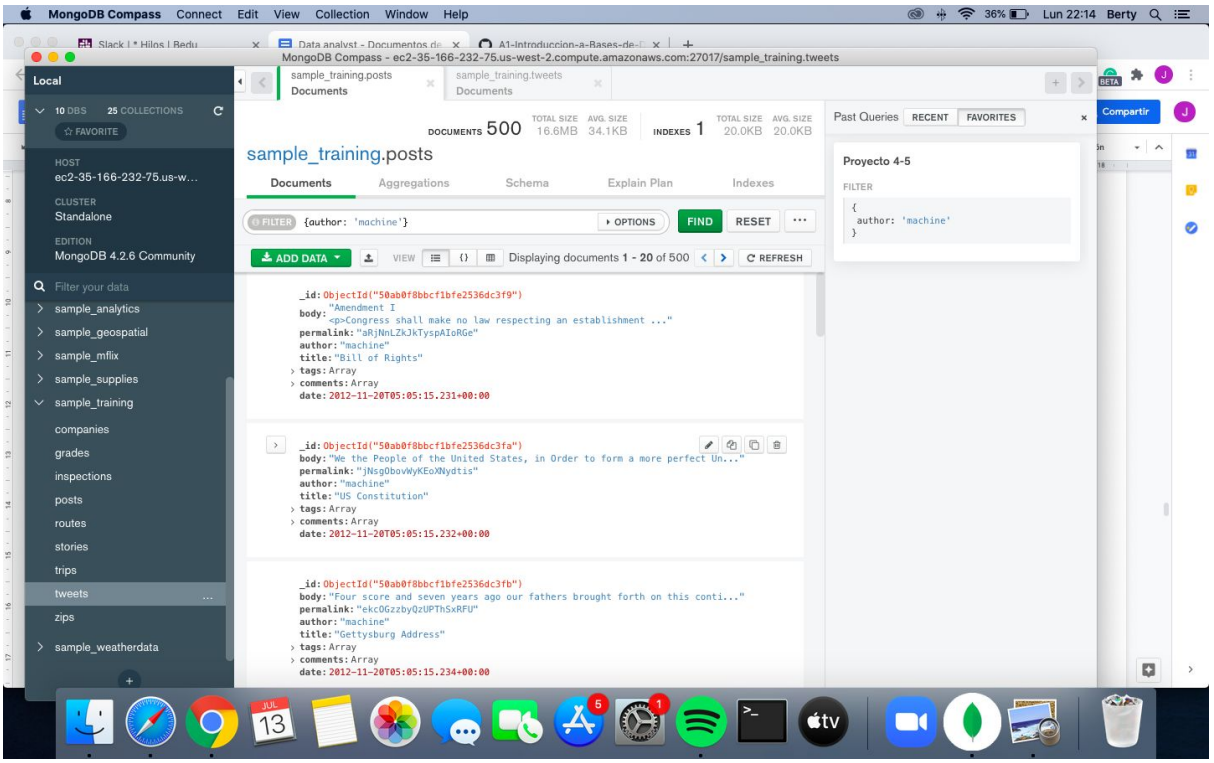
4.- Obtén el nombre de todas las compañías fundadas en 2008.

```
{
 filter: {
  founded_year: 2008
 }
}
```



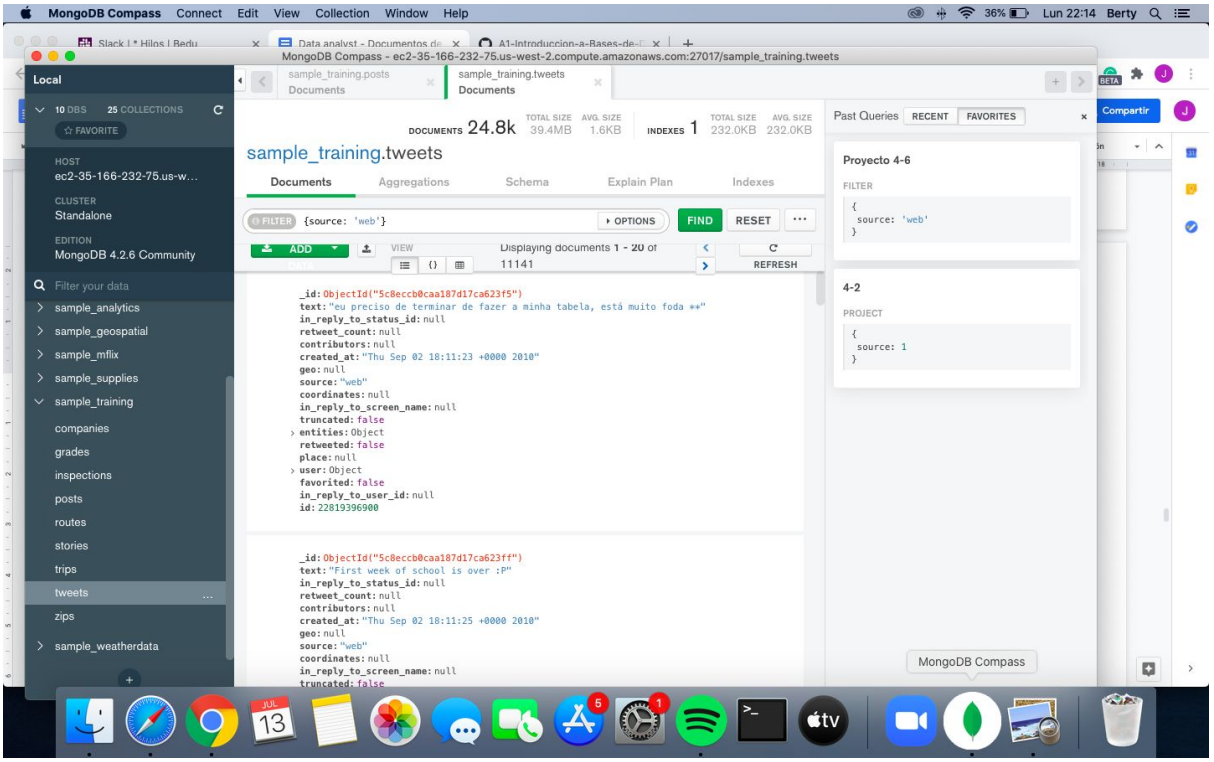
5.- Obtén todos los *post* del autor `machine`.

```
{
 filter: {
  author: 'machine'
 }
}
```



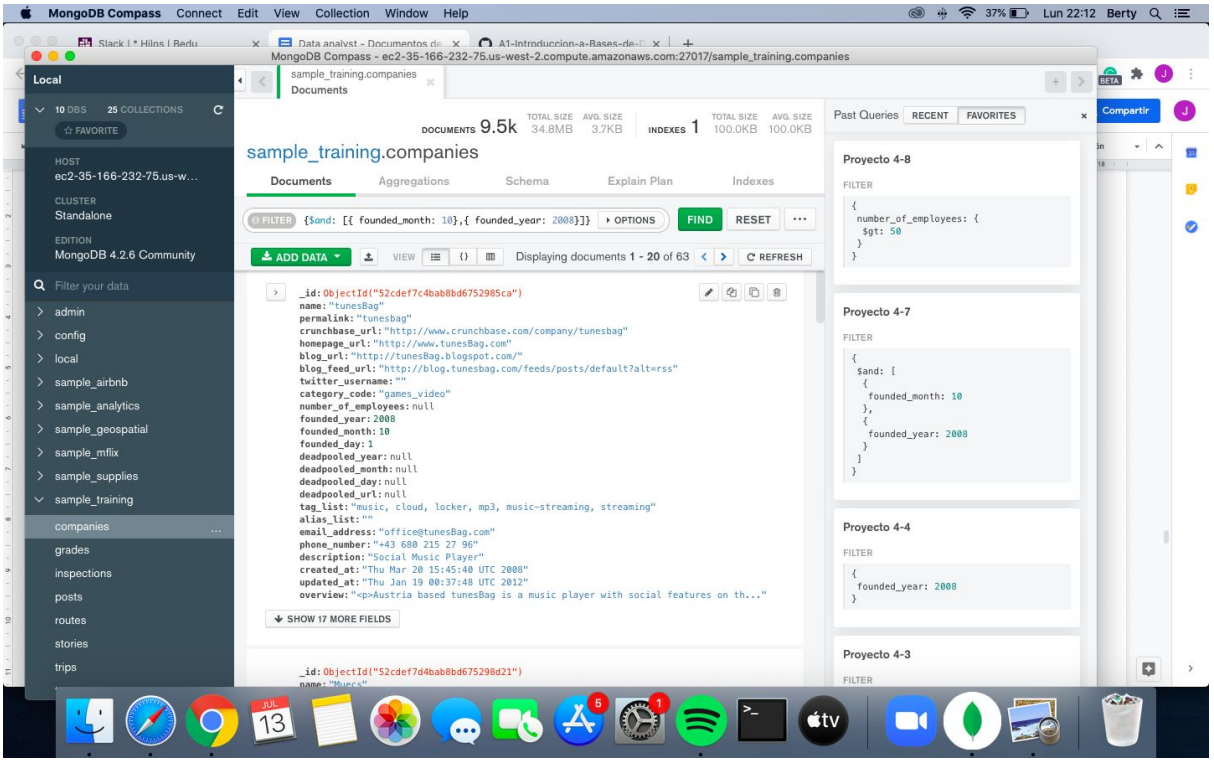
6.- Obtén todos los tweets provenientes de la `web`.

```
{
 filter: {
  source: 'web'
 }
}
```



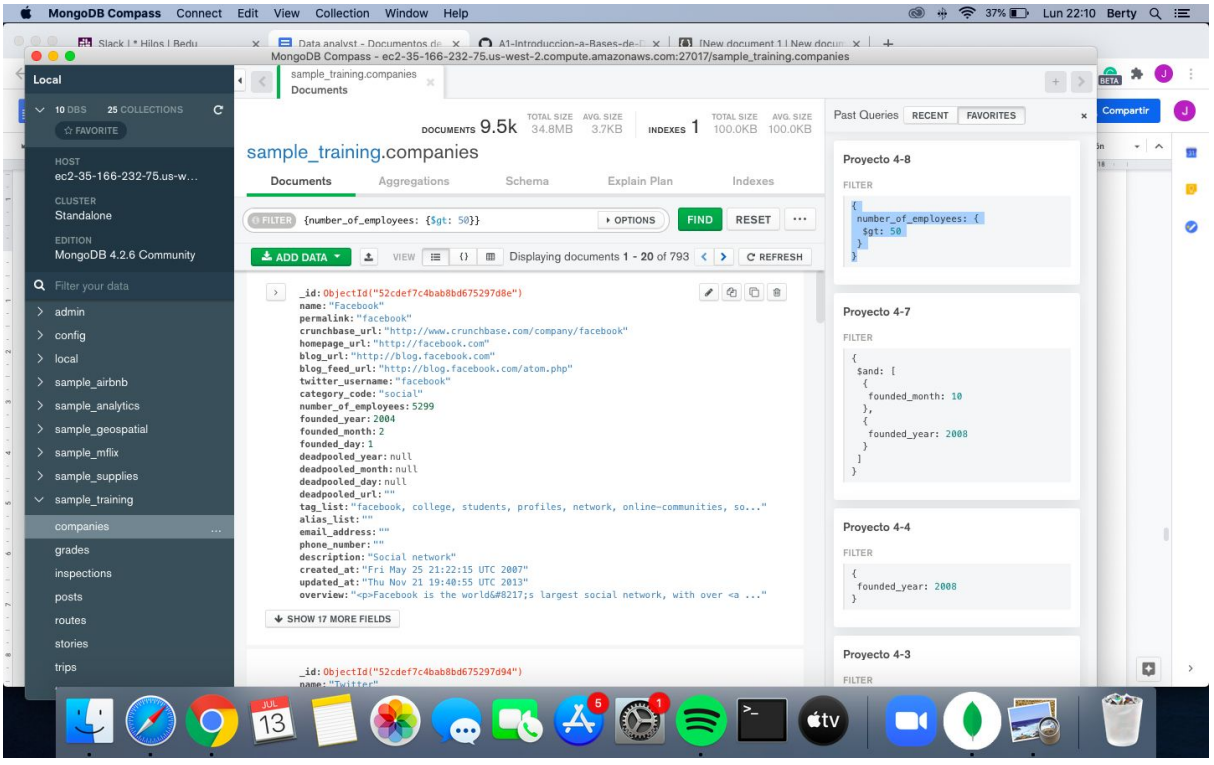
7.- Obtén todas las compañías fundadas en octubre del 2008

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



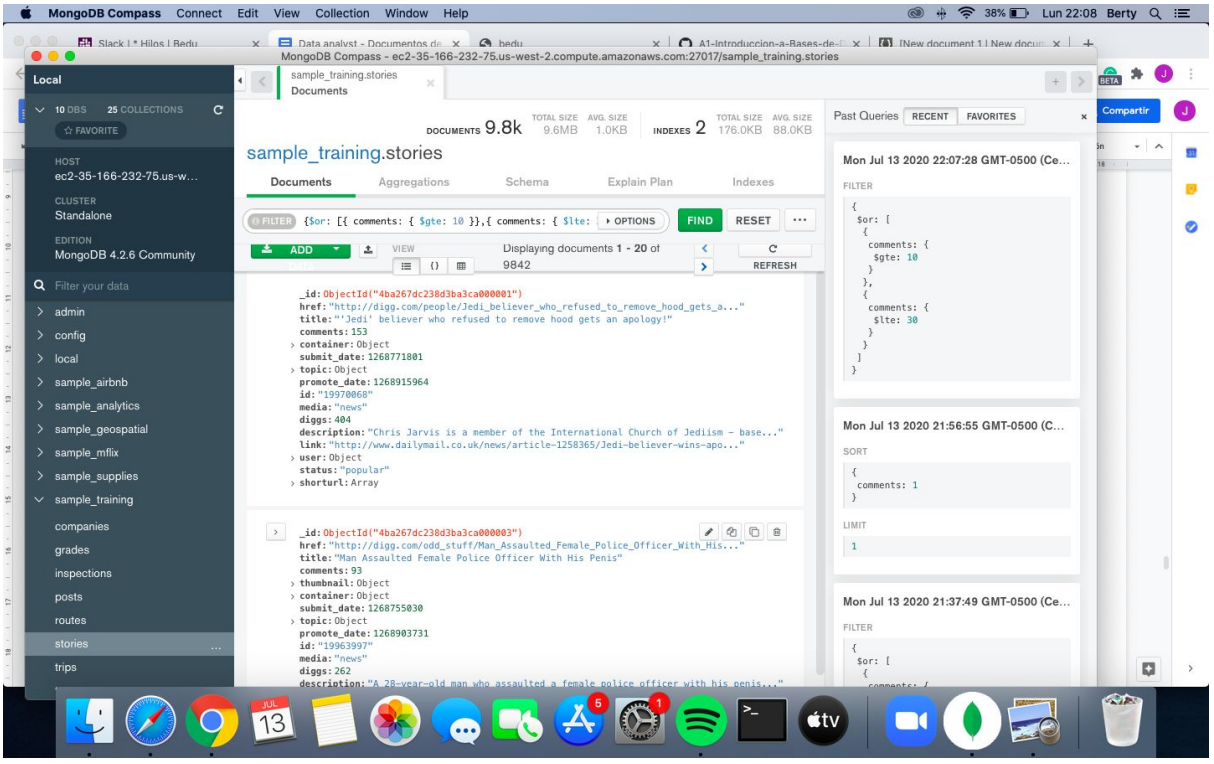
8.- Obtén todas las compañias con más de 50 empleados.

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



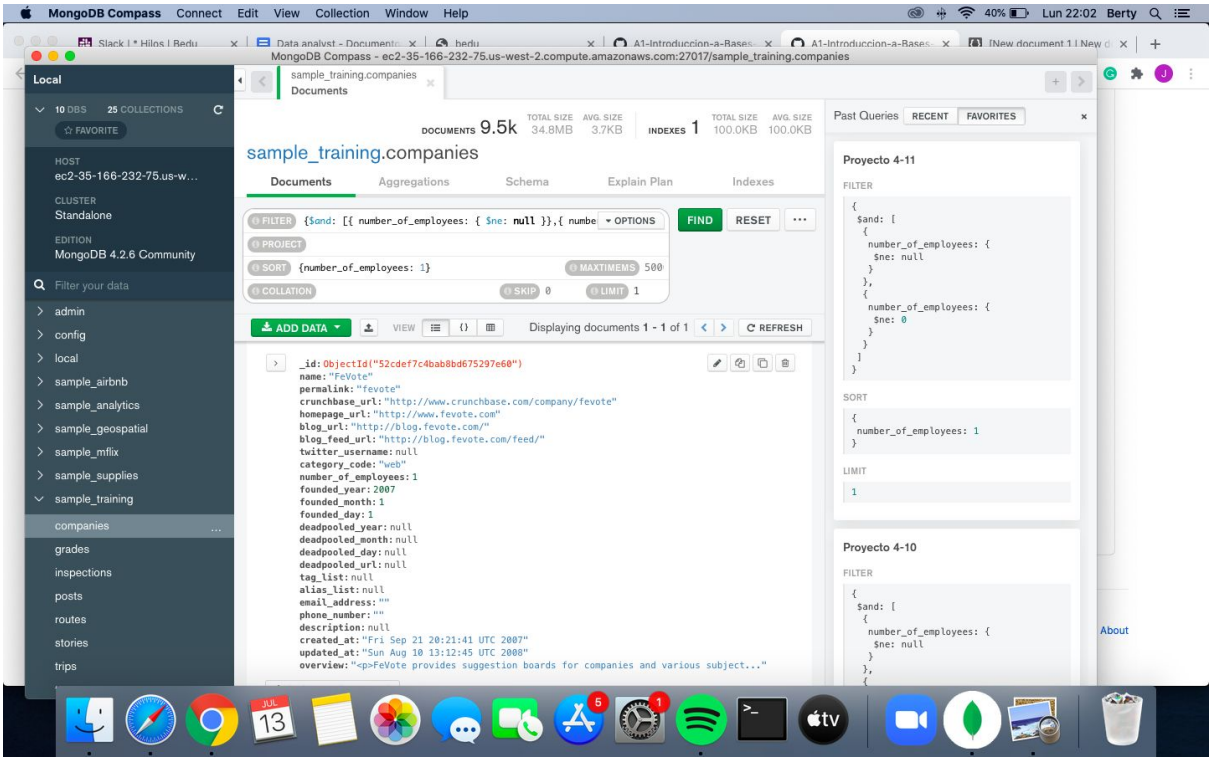
9.- Obtén las historias con número de comentarios entre 10 y 30.

```
{
 filter: {
  $or: [
   {
    comments: {
     $gte: 10
    }
   },
   {
    comments: {
     $lte: 30
    }
   }
  ]
 }
```



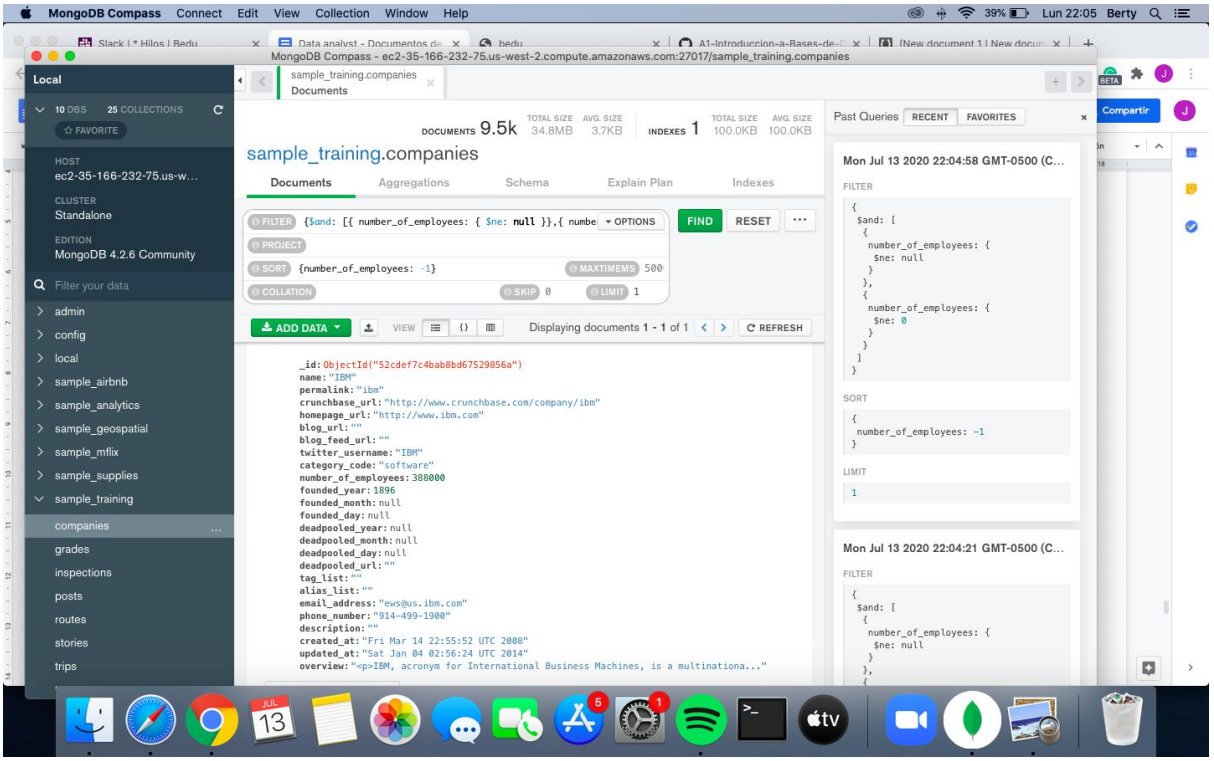
10.- Obtén la empresa con el menor número de empleados.

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



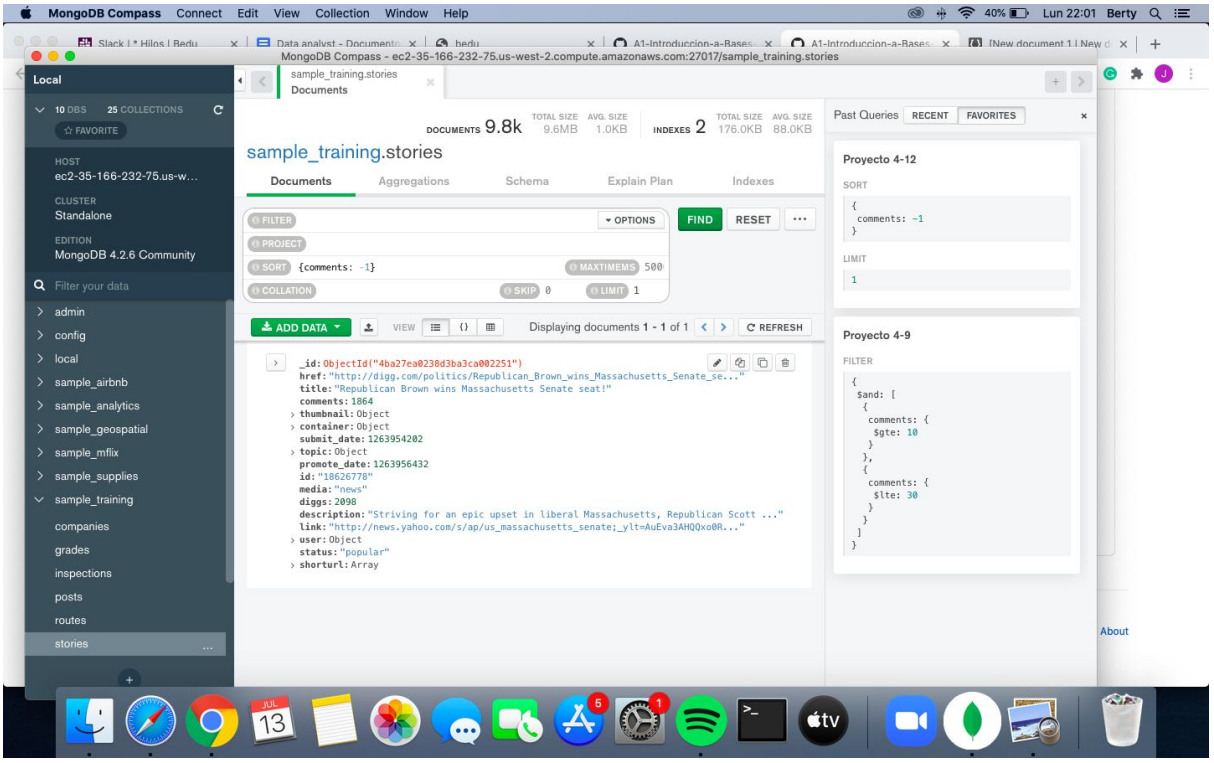
11.- Obtén la empresa con el mayor número de empleados.

```
{
 filter: {
  $and: [
   {
    number_of_employees: {
     $ne: null
    }
   },
   {
    number_of_employees: {
     $ne: 0
    }
   }
  ]
 }
},
 sort: {
  number_of_employees: -1
 },
 limit: 1
}
```



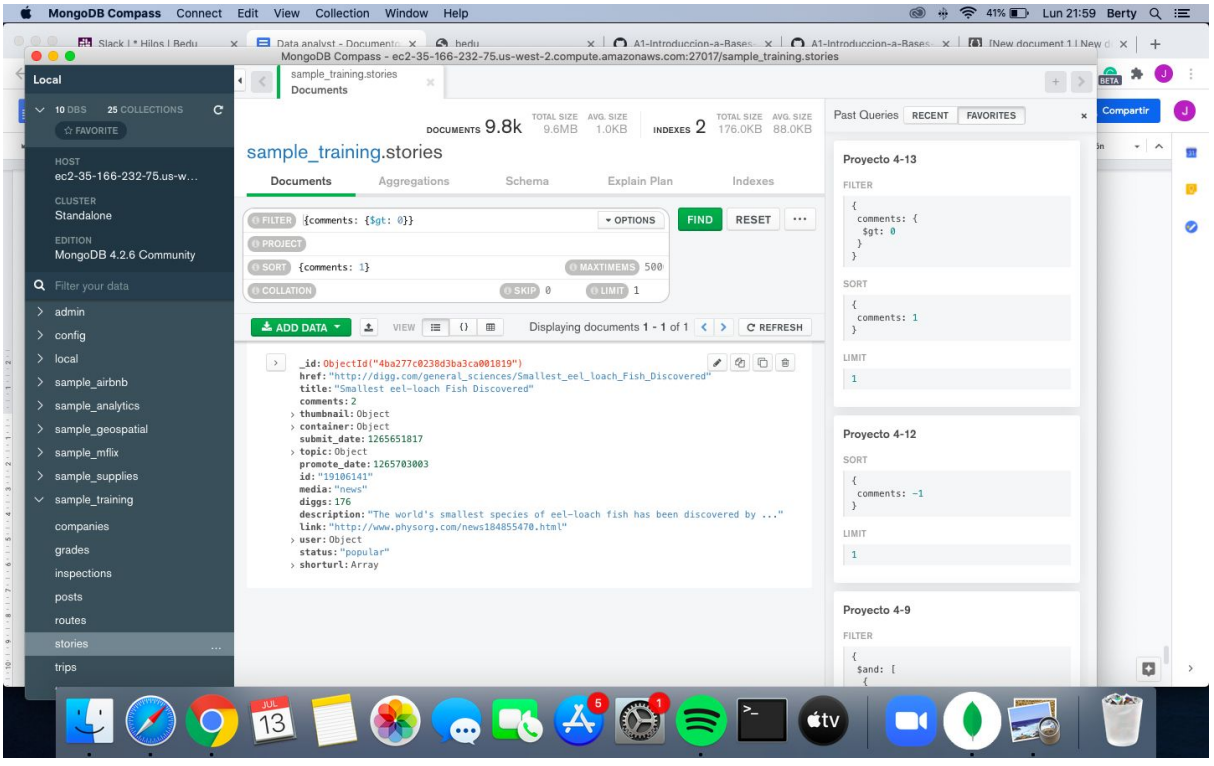
12.- Obtén la historia más comentada.

```
{
 sort: {
  comments: -1
 },
 limit: 1
}
```



13.- Obtén la historia menos comentada.

```
{
 filter: {
  comments: {
   $gt: 0
  }
 },
 sort: {
  comments: 1
 },
 limit: 1
}
```

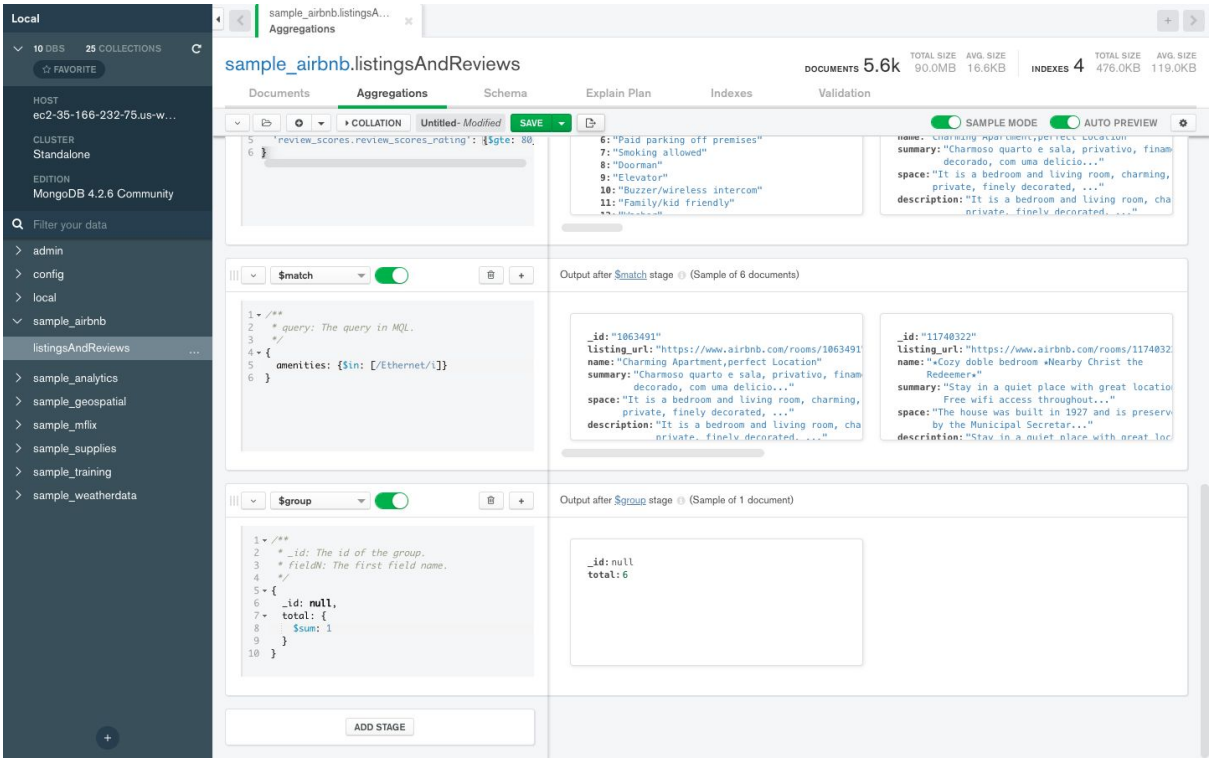


Sesión 05: Consultas en MongoDB

Proyecto 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 ubicadas en Brazil.

```
[{$match: {
  number_of_reviews: {$gt: 50}
}}, {$match: {
  "address.country": "Brazil"
}}, {$match: {
  'review_scores.review_scores_rating': {$gte: 80}
}}, {$match: {
  amenities: {$in: [/Ethernet/i]}
}}, {$group: {
  _id: null,
  total: {
    $sum: 1
  }
}}]
```

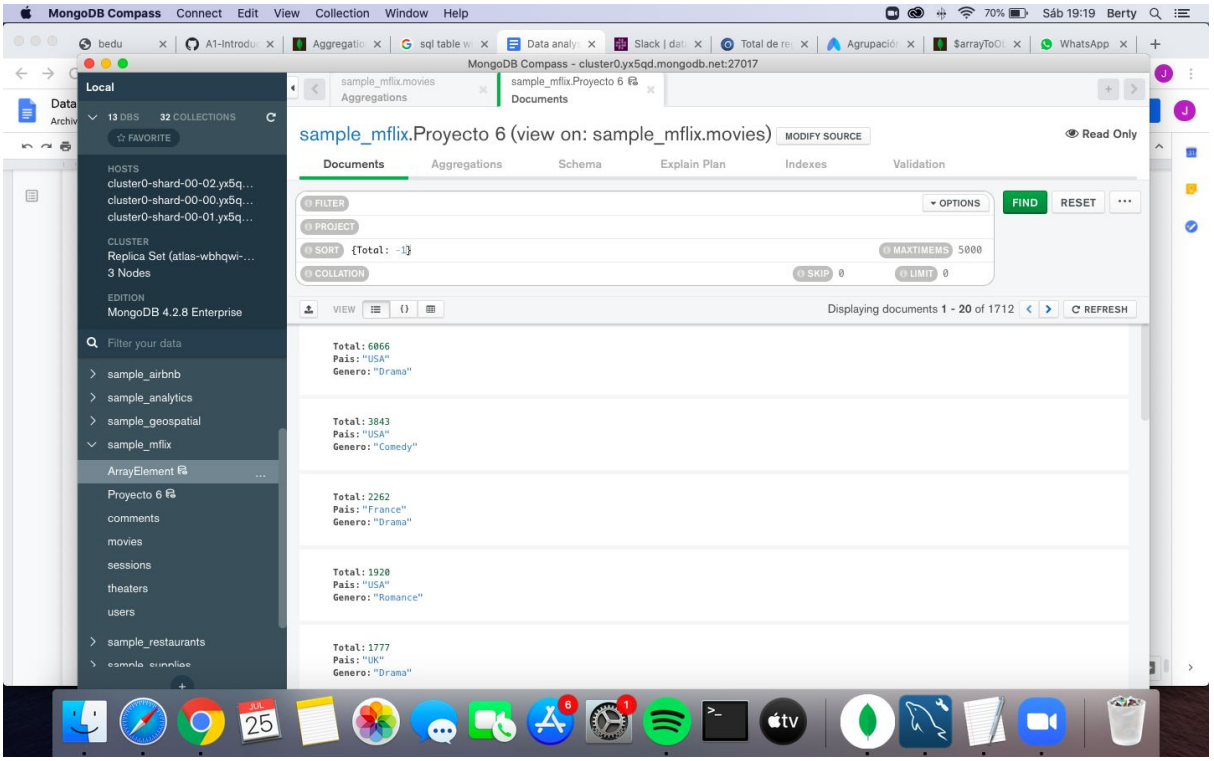


Sesión 06: Agregaciones

Proyecto 6

El proyecto consiste en obtener, por país, el número de películas que hay de cada género:

```
[{$unwind: {
  path: "$genres",
  includeArrayIndex: 'string'
}}, {$unwind: {
  path: "$countries",
  includeArrayIndex: 'string'
}}, {$group: {
  _id:{"genres": "$genres", "countries": "$countries"},
  Total: {$sum: 1}
}}, {$sort: {
  Total: -1
}}, {$addFields: {
  Pais: "$_id.countries",
  Genero: "$_id.genres",
}}, {$project: {
  _id: 0,
  Pais: 1,
  Genero: 1,
  Total: 1
}}]
```



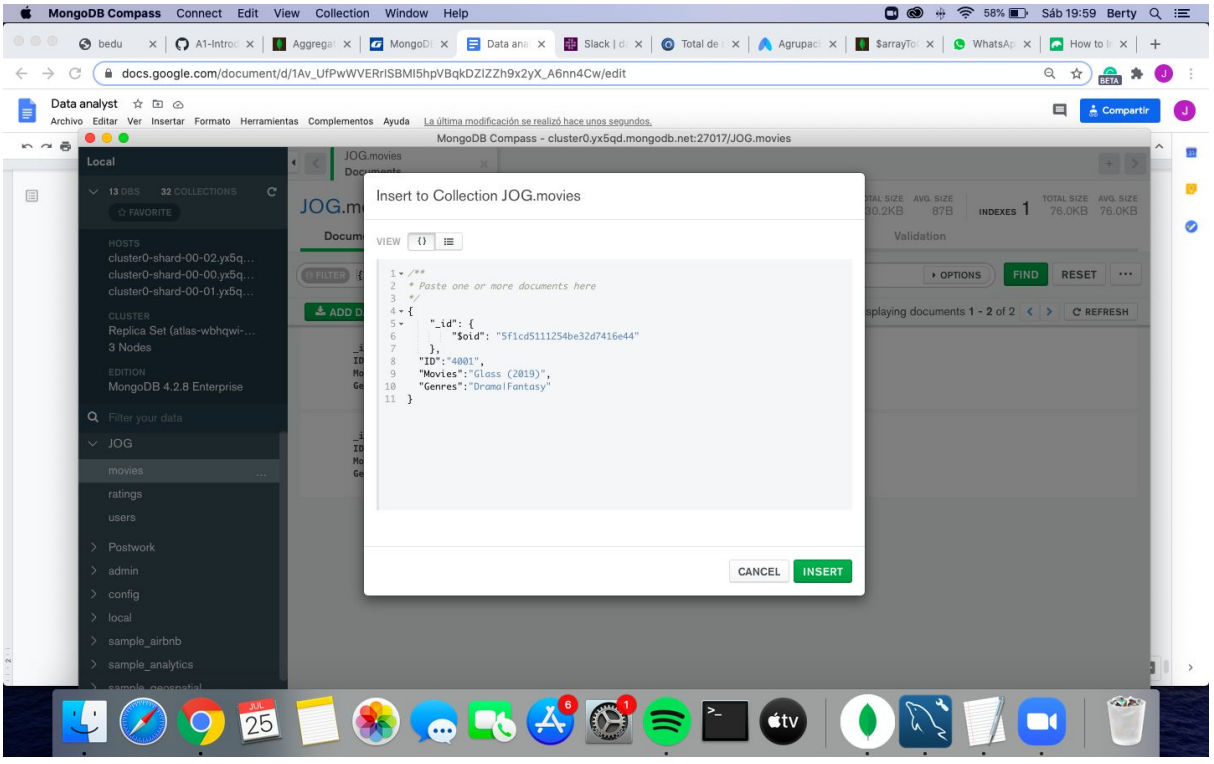
Sesión 07 Configuración de Bases de Datos Locales

Proyecto 7

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

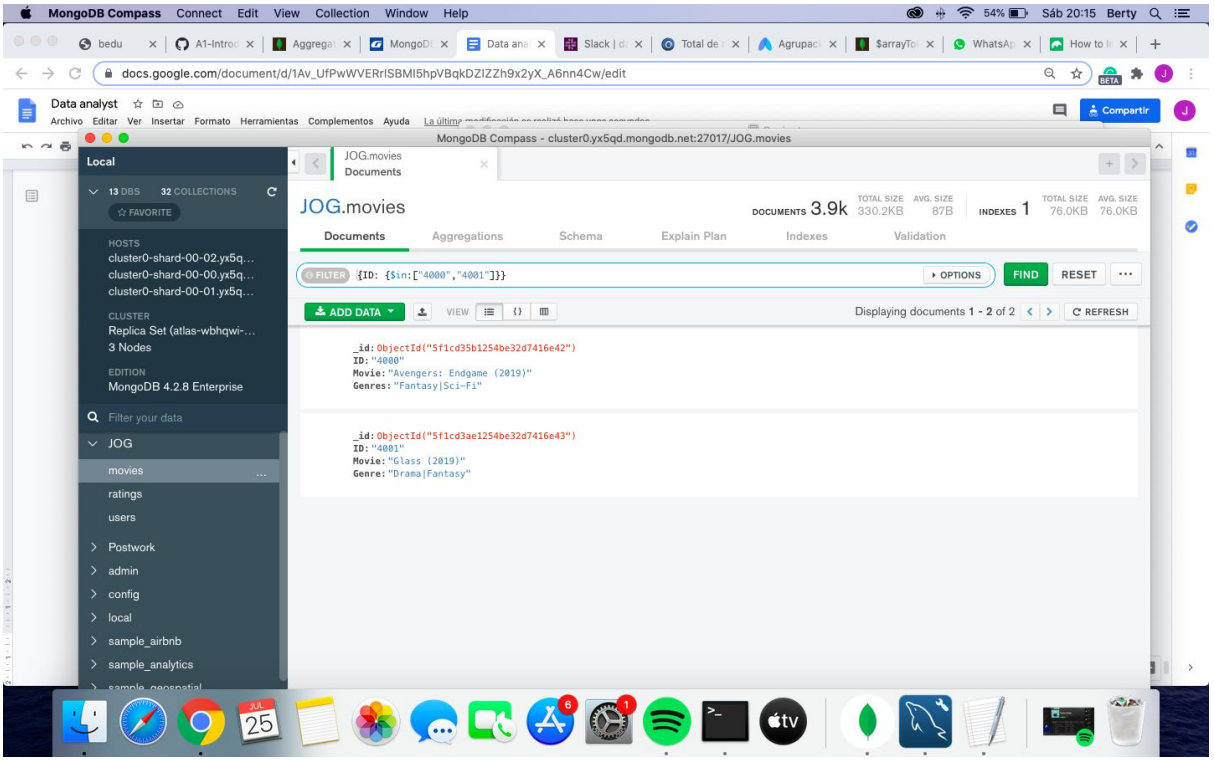
```
4000,Avengers: Endgame (2019),Fantasy|Sci-Fi
4001,Glass (2019),Drama|Fantasy
```


```
{
   "_id": {
      "$oid": "5f1cd63e1254be32d7416e45"
   },
  "ID":"4000",
  "Movie":"Avengers: Endgame (2019)",
  "Genres":"Fantasy|Sci-Fi"
}
{
   "_id": {
      "$oid": "5f1cd63e1254be32d7416e46"
   },
  "ID":"4001",
  "Movie":"Glass (2019)",
  "Genres":"Drama|Fantasy"
}
```

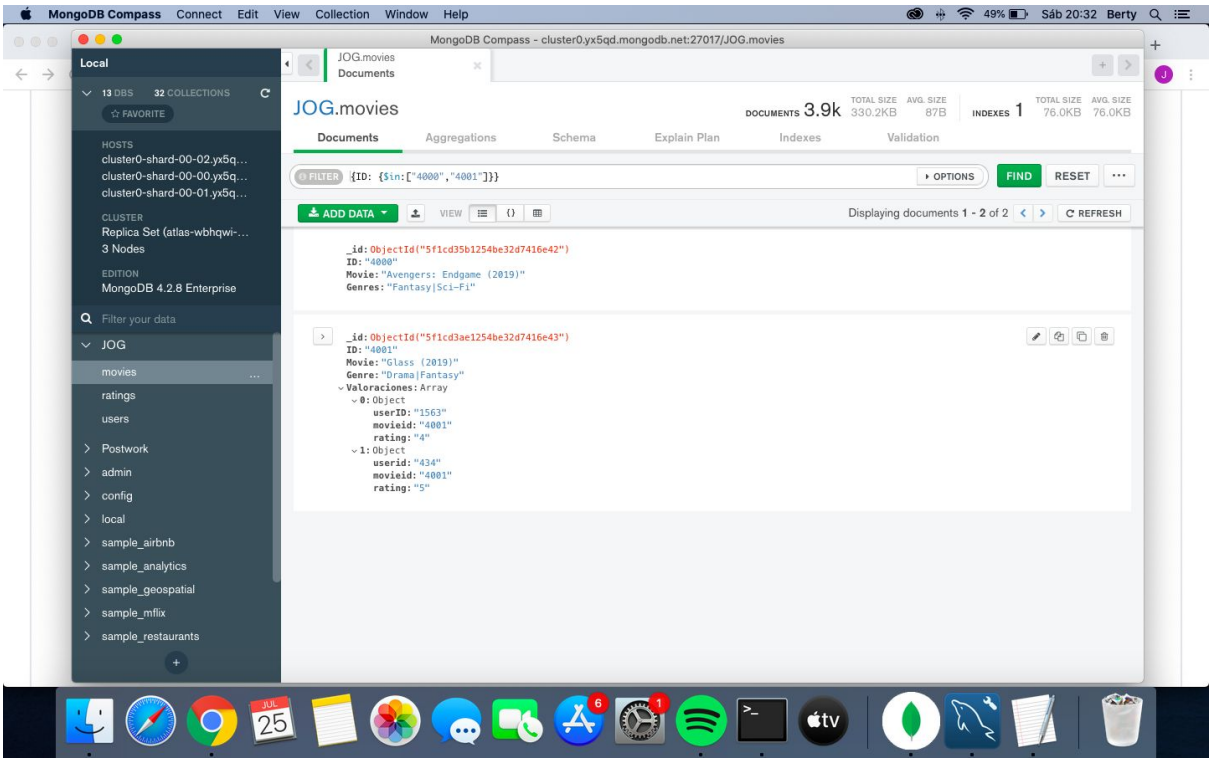
Filtro para ambos documentos:

{ID: {$in:["4000","4001"]}}



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

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


MongoDB Compass Connect Edit View Collection Window Help
Sáb 20:32 Berty
MongoDB Compass - cluster0.yx5qd.mongodb.net:27017/JOG.movies
Local
13 DBS 32 COLLECTIONS
FAVORITE
HOSTS
cluster0-shard-00-02.yx5q...
cluster0-shard-00-00.yx5q...
cluster0-shard-00-01.yx5q...
CLUSTER
Replica Set (atlas-wbhqwi-...
3 Nodes
EDITION
MongoDB 4.2.8 Enterprise
Filter your data
JOG
movies
ratings
users
Postwork
admin
config
local
sample_airbnb
sample_analytics
sample_geospatial
sample_mflix
sample_restaurants
JOG.movies
Documents
JOG.movies
DOCUMENTS 3.9k
TOTAL SIZE 330.2KB
AVG. SIZE 87B
INDEXES 1
TOTAL SIZE 76.0KB
AVG. SIZE 76.0KB
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("5f1cd35b1254be32d7416e42")
ID: "4000"
Movie: "Avengers: Endgame (2019)"
Genres: "Fantasy|Sci-Fi"
_id: ObjectId("5f1cd3ae1254be32d7416e43")
ID: "4001"
Movie: "Glass (2019)"
Genre: "Drama|Fantasy"
Valoraciones: Array
0: Object
userID: "1563"
movieid: "4001"
rating: "4"
1: Object
userid: "434"
movieid: "4001"
rating: "5"

Sesión 08 Query competition

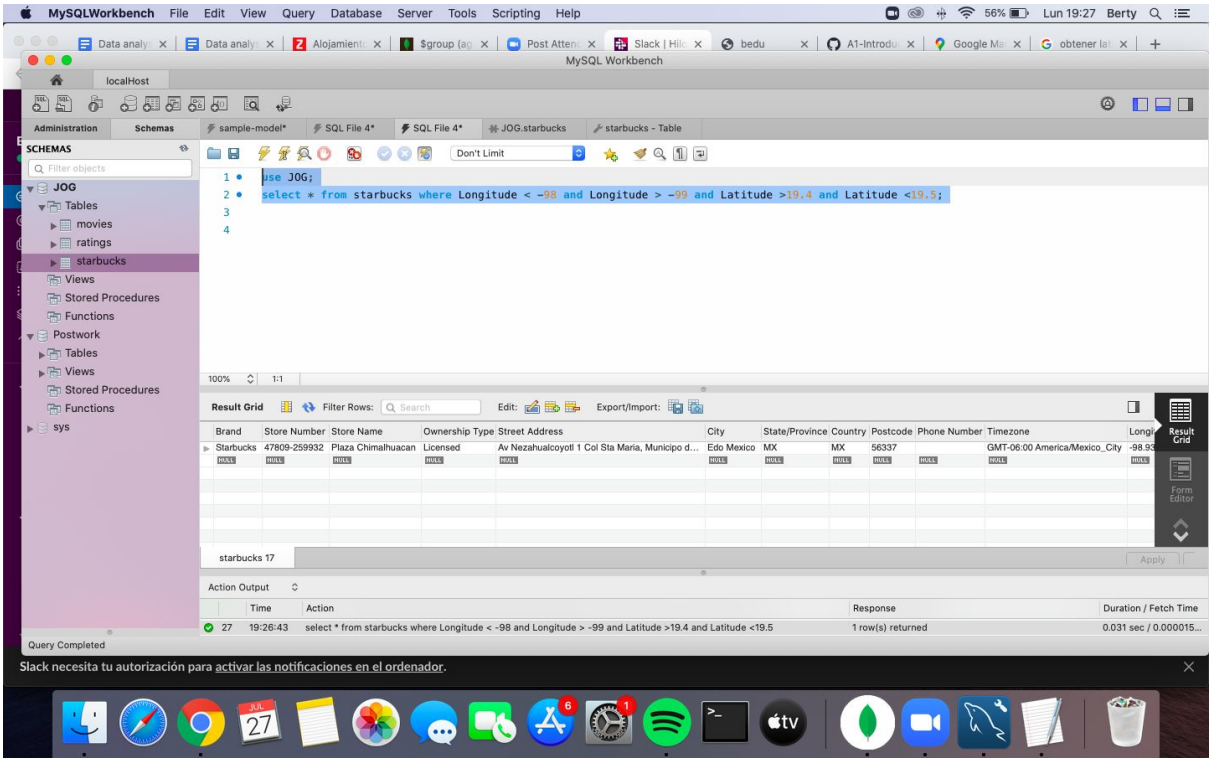
Proyecto 8

Reto 1:

Descarga la fuente de datos de los locales de Starbucks “directory.csv”. Analiza y limpia los datos. Elige MySQL o MongoDB y crea una base de datos para el conjunto de datos del reto. Carga los datos en la base de datos que elegiste y revisa que éstos se muestren correctamente.

Usando la latitud y longitud de tu posición actual, encuenta el Starbucks más cercano a tu posición. Para conocer tu posición actual puedes usar Google Maps para, sólo debes copiar los datos de la URL.

```
use JOG;
select * from starbucks where Longitude < -98 and Longitude > -99 and Latitude >19.4 and Latitude <19.5;
```

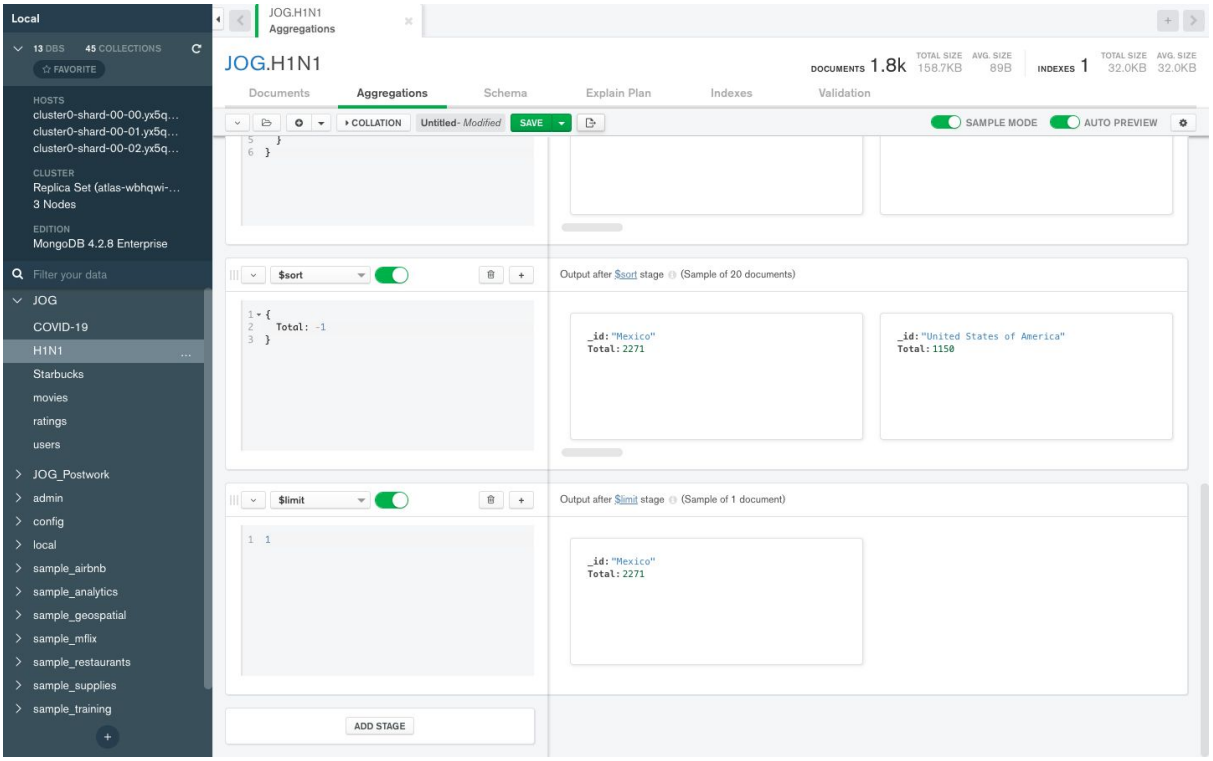


Reto 2

Descarga la fuente de datos de los locales del reto “Pandemic (H1N1) 2009.csv”. Analiza y limpia los datos. Elige MySQL o MongoDB y crea una base de datos para el conjunto de datos del reto. Carga los datos en la base de datos que elegiste y revisa que éstos se muestren correctamente. Responde a las siguientes preguntas usando consultas:

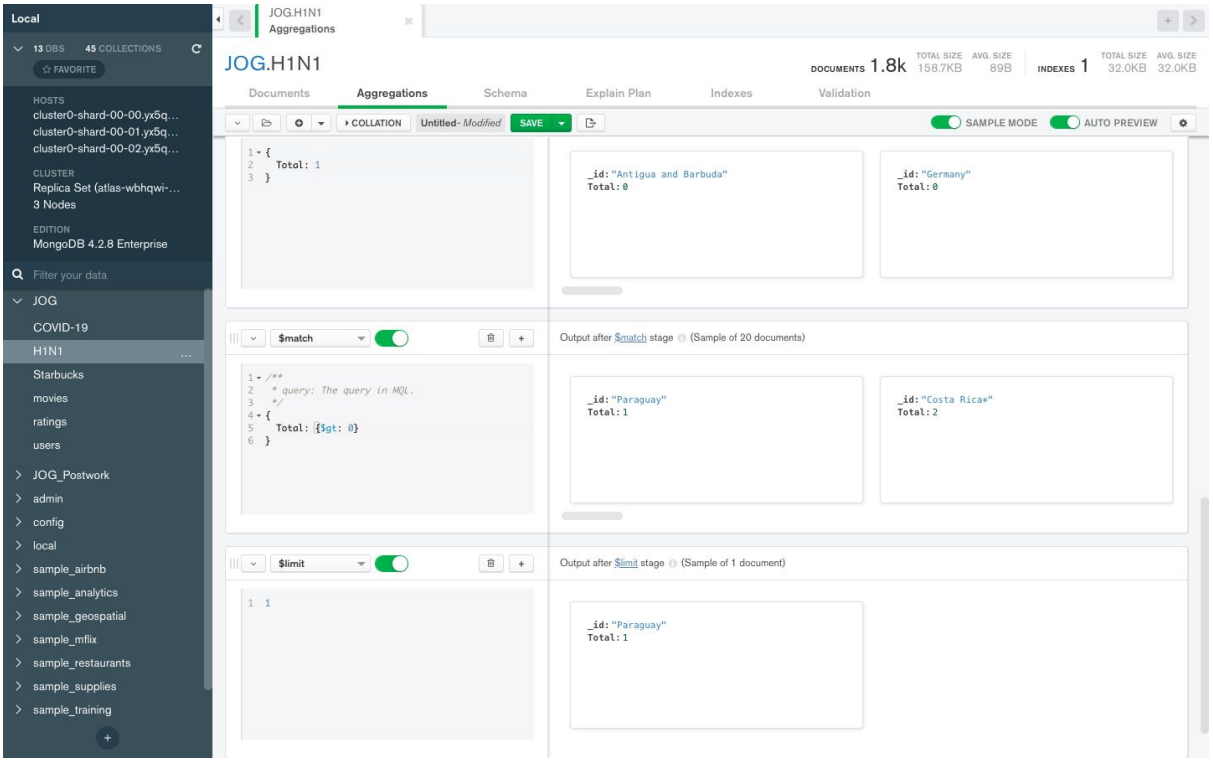
¿Cuál fue el país con mayor número de muertes?

```
[{$sort: {
  Cases: -1
}}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  Deaths: {
    $ne: NaN
  }
}}, {$group: {
  _id: '$Country',
  Total: {
    $sum: '$Deaths'
  }
}}, {$sort: {
  Total: -1
}}, {$limit: 1}]
```



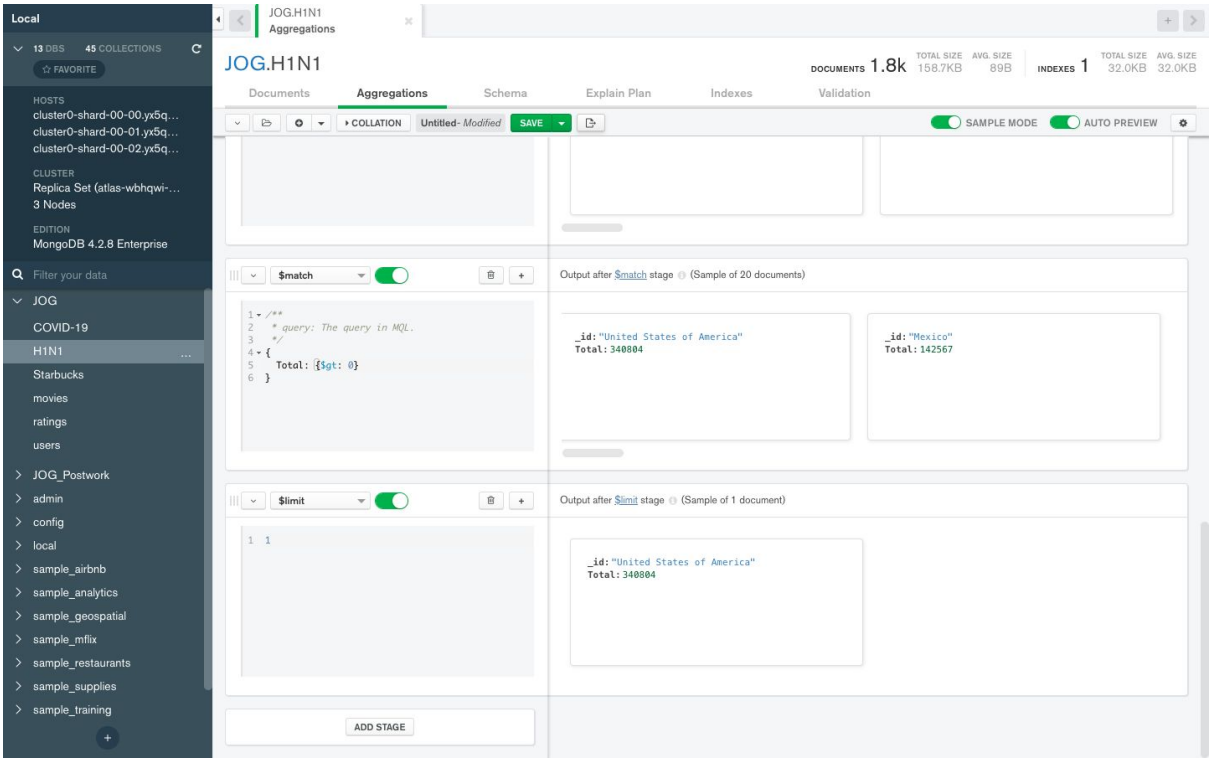
¿Cuál fue el país con menor número de muertes?

```
[{$sort: {
  Cases: -1
}}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  Deaths: {
    $ne: NaN
  }
}}, {$group: {
  _id: '$Country',
  Total: {
    $sum: '$Deaths'
  }
}}, {$sort: {
  Total: 1
}}, {$match: {
  Total: {$gt: 0}
}}, {$limit: 1}]
```



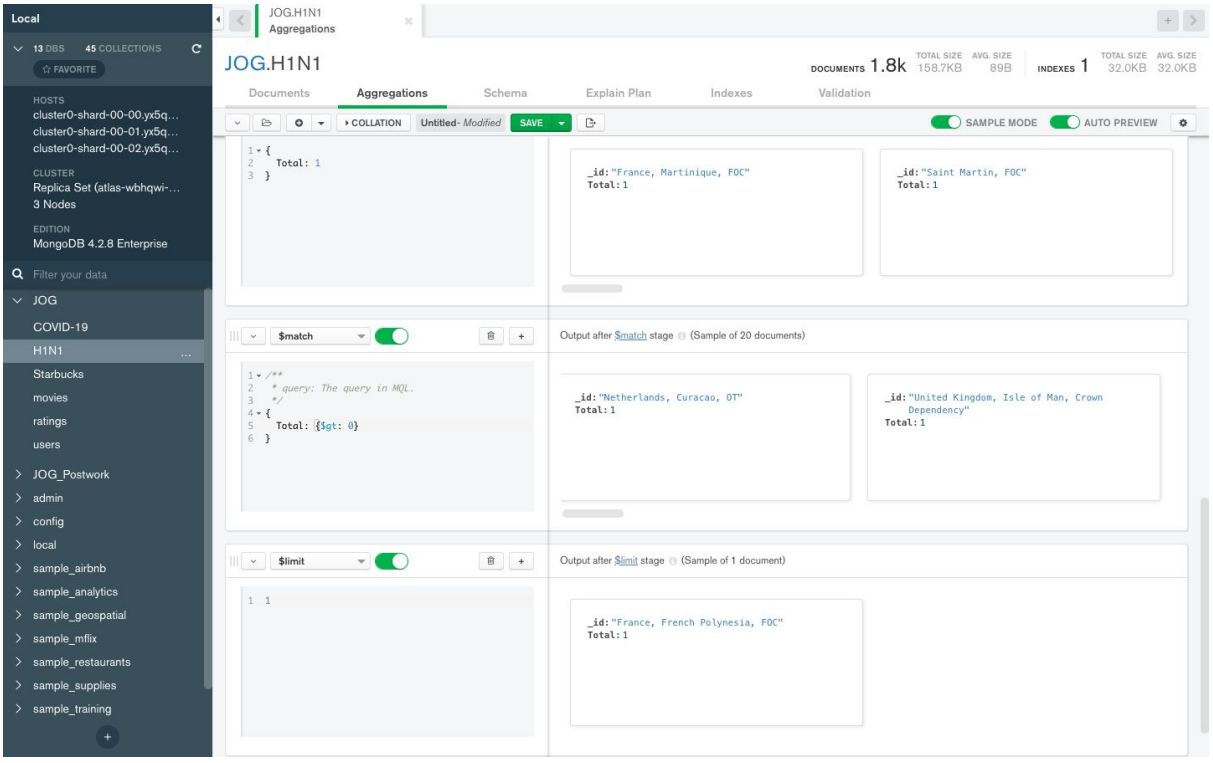
¿Cuál fue el país con el mayor número de casos?

```
[{$sort: {
  Cases: -1
}}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  Deaths: {
    $ne: NaN
  }
}}, {$group: {
  _id: '$Country',
  Total: {
    $sum: '$Cases'
  }
}}, {$sort: {
  Total: -1
}}, {$match: {
  Total: {$gt: 0}
}}, {$limit: 1}]
```



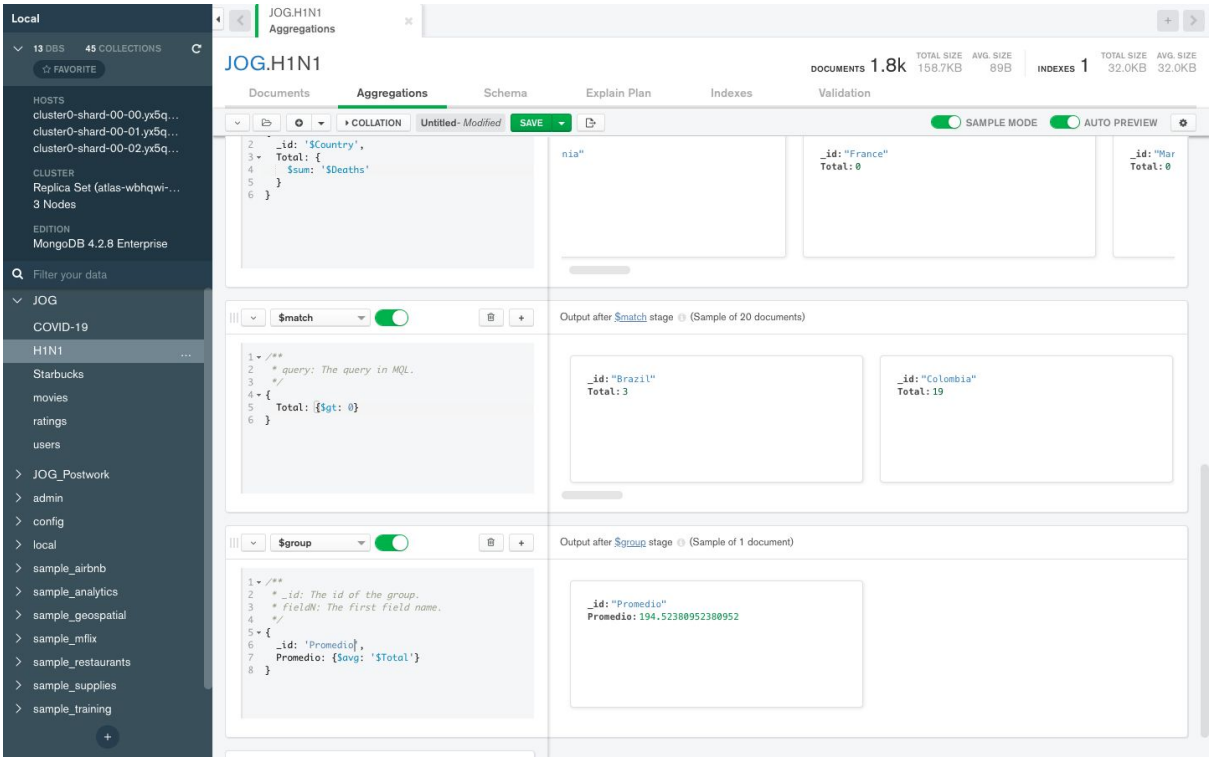
¿Cuál fue el país con el menor número de casos?

```
[{$sort: {
  Cases: -1
}}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  Deaths: {
    $ne: NaN
  }
}}, {$group: {
  _id: '$Country',
  Total: {
    $sum: '$Cases'
  }
}}, {$sort: {
  Total: 1
}}, {$match: {
  Total: {$gt: 0}
}}, {$limit: 1}]
```



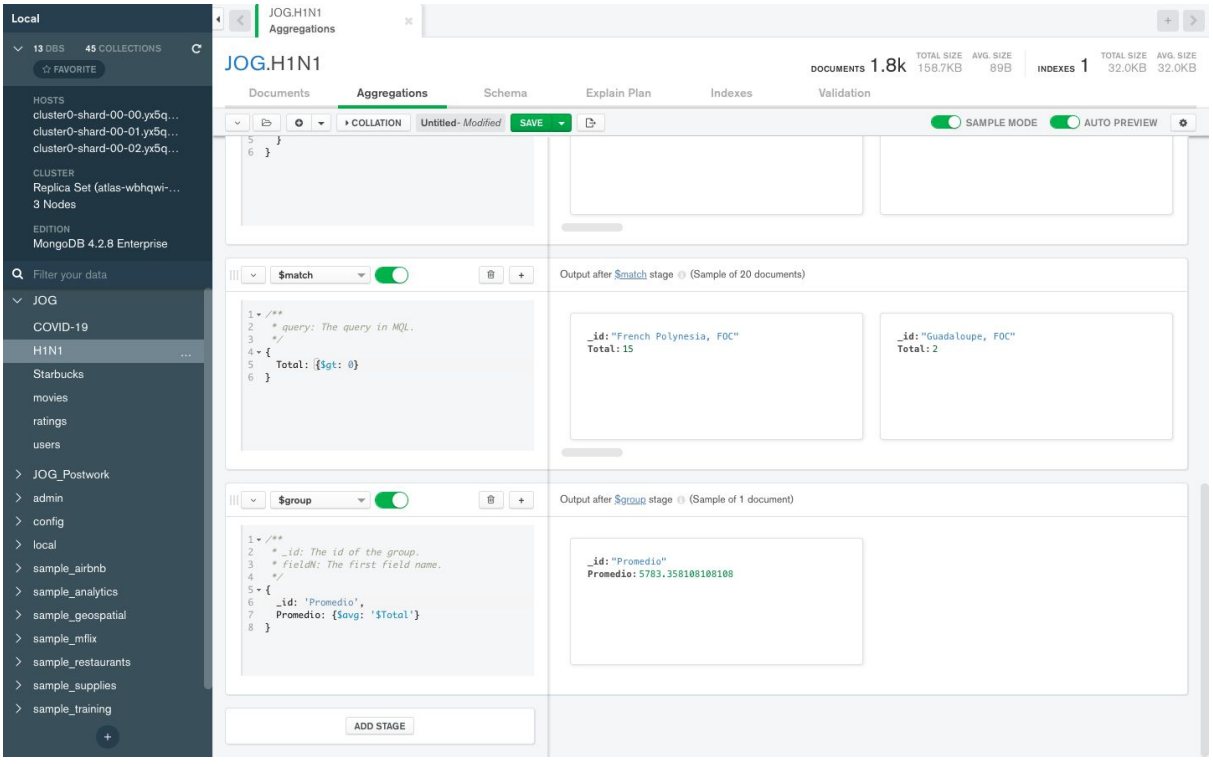
¿Cuál fue el número de muertes promedio?

```
[{$sort: {
  Cases: -1
}}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  Deaths: {
    $ne: NaN
  }
}}, {$group: {
  _id: '$Country',
  Total: {
    $sum: '$Deaths'
  }
}}, {$match: {
  Total: {$gt: 0}
}}, {$group: {
  _id: 'Promedio',
  Promedio: {$avg: '$Total'}
}}]
```



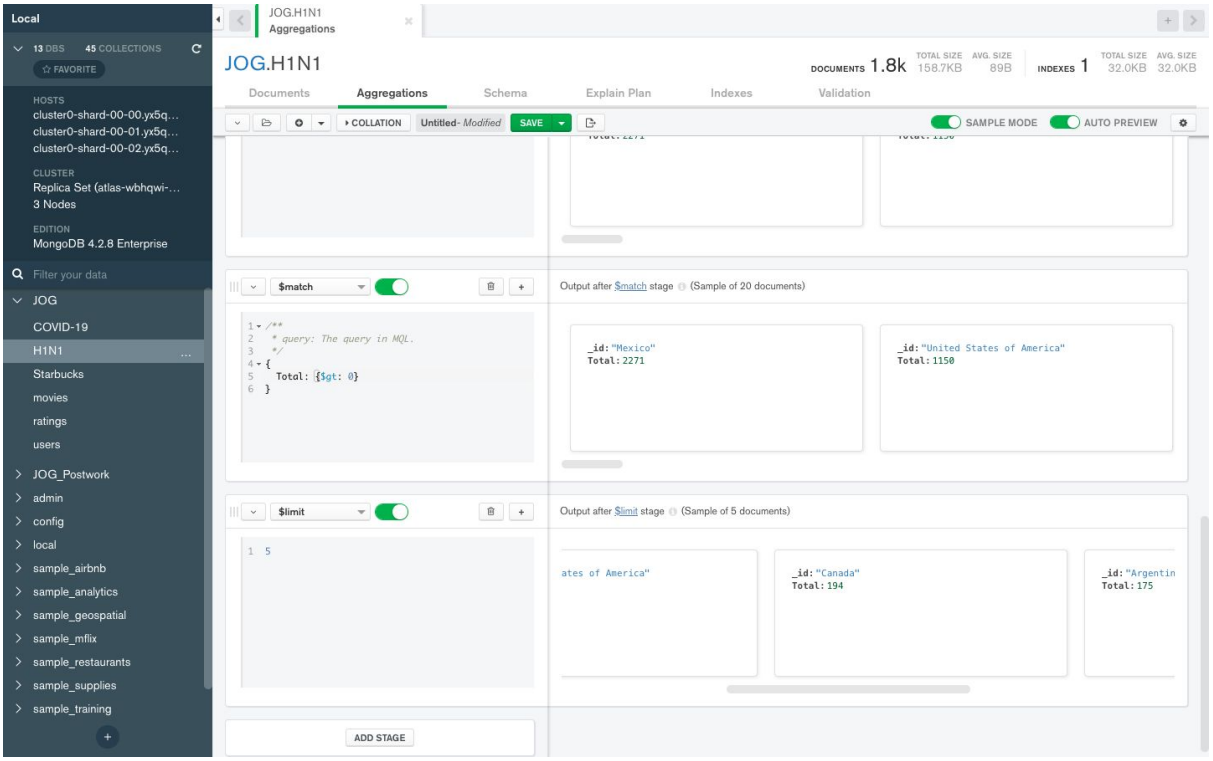
¿Cuál fue el número de casos promedio?

```
[{$sort: {
  Cases: -1
}}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  Deaths: {
    $ne: NaN
  }
}}, {$group: {
  _id: '$Country',
  Total: {
    $sum: '$Cases'
  }
}}, {$match: {
  Total: {$gt: 0}
}}, {$group: {
  _id: 'Promedio',
  Promedio: {$avg: '$Total'}
}}]
```



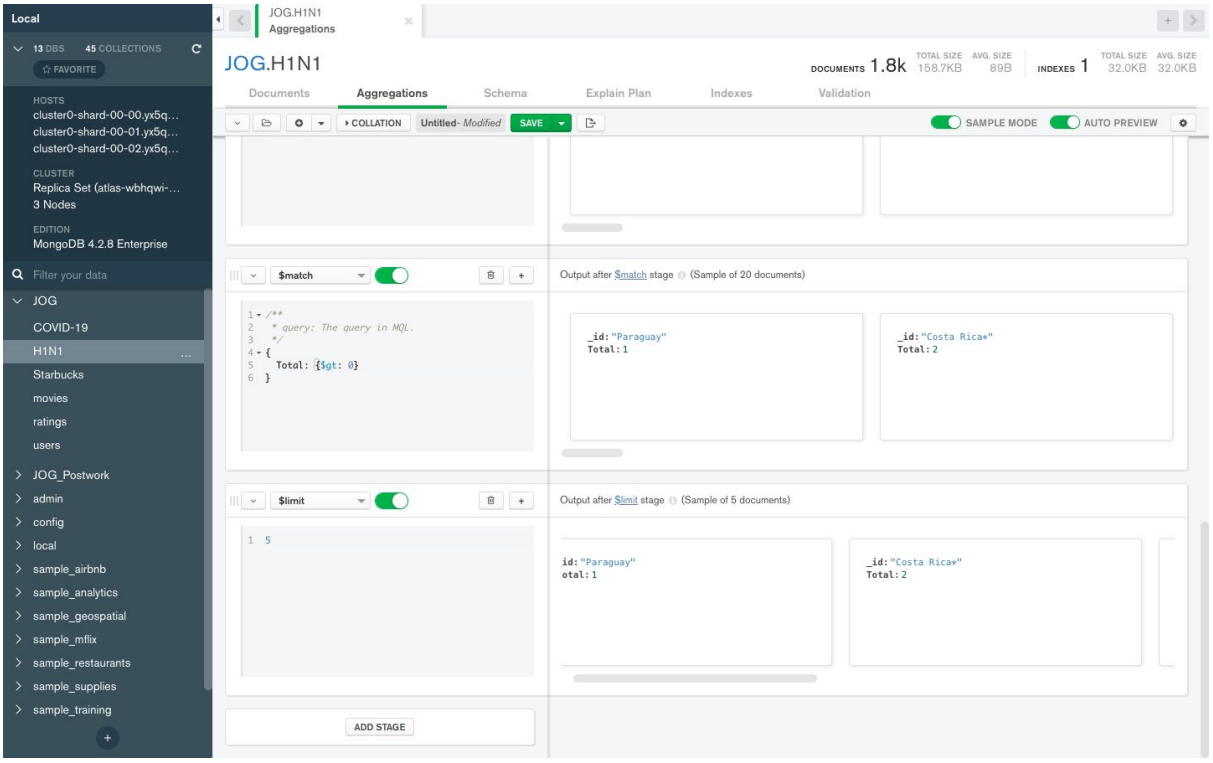
Top 5 de países con más muertes

```
[{$sort: {
  Cases: -1
}}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  Deaths: {
    $ne: NaN
  }
}}, {$group: {
  _id: '$Country',
  Total: {
    $sum: '$Deaths'
  }
}}, {$sort: {
  Total: -1
}}, {$match: {
  Total: {$gt: 0}
}}, {$limit: 5}]
```



Top 5 de países con menos muertes

```
[{$sort: {
  Cases: -1
}}, {$match: {
  Country: {
    $ne: 'Grand Total'
  },
  Deaths: {
    $ne: NaN
  }
}}, {$group: {
  _id: '$Country',
  Total: {
    $sum: '$Deaths'
  }
}}, {$sort: {
  Total: 1
}}, {$match: {
  Total: {$gt: 0}
}}, {$limit: 5}]
```

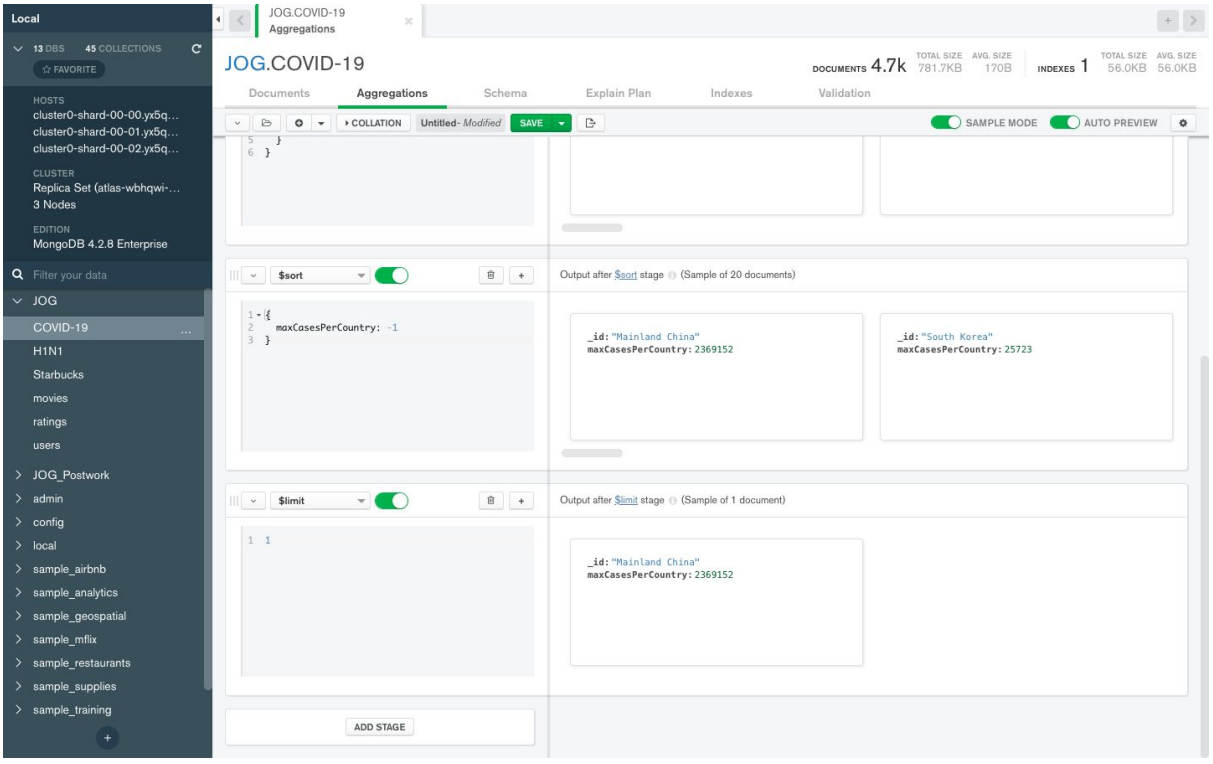


Reto 3

Descarga la fuente de datos de los locales del reto “2019-nCoV-cases-JHU.csv”. Analiza y limpia los datos. Elige MySQL o MongoDB y crea una base de datos para el conjunto de datos del reto. Carga los datos en la base de datos que elegiste y revisa que éstos se muestren correctamente. Responde a las siguientes preguntas usando consultas:

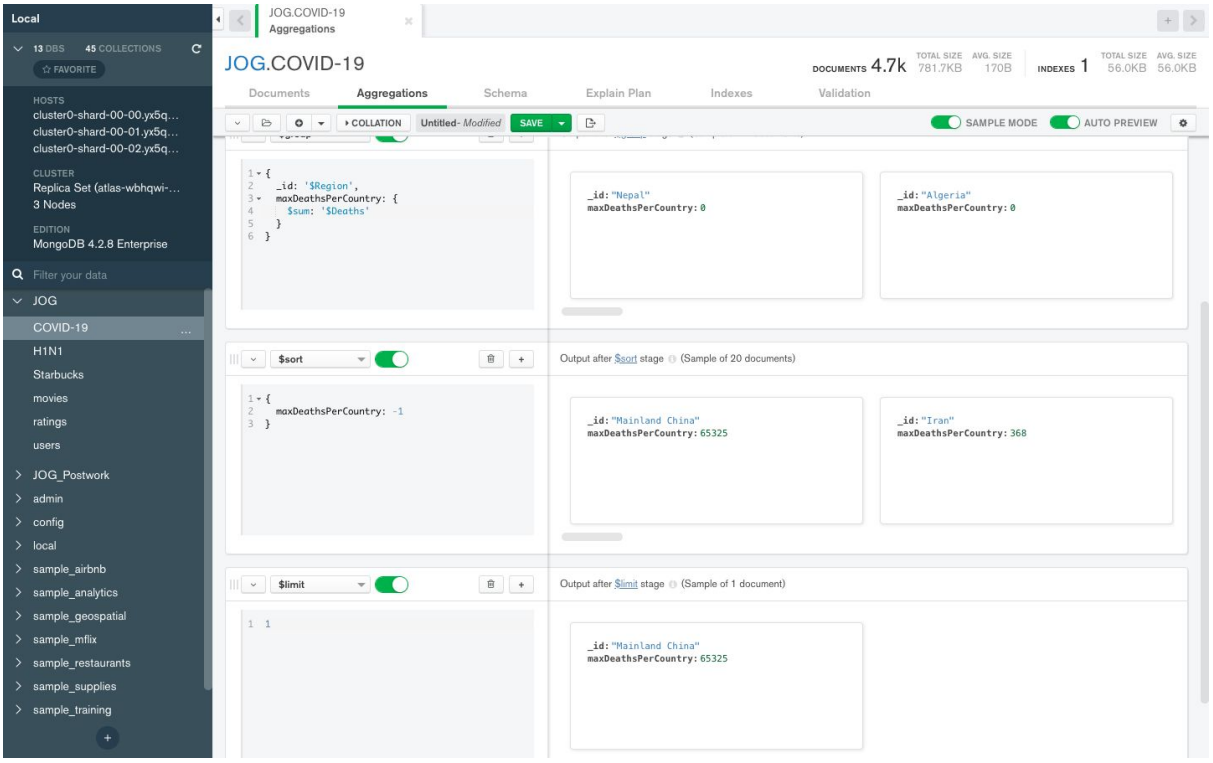
¿Cuál es país con mayor número de casos?

```
[{$group: {
  _id: '$Region',
  maxCasesPerCountry: {
    $sum: '$Confirmed'
  }
}}, {$sort: {
  maxCasesPerCountry: -1
}}, {$limit: 1}]
```



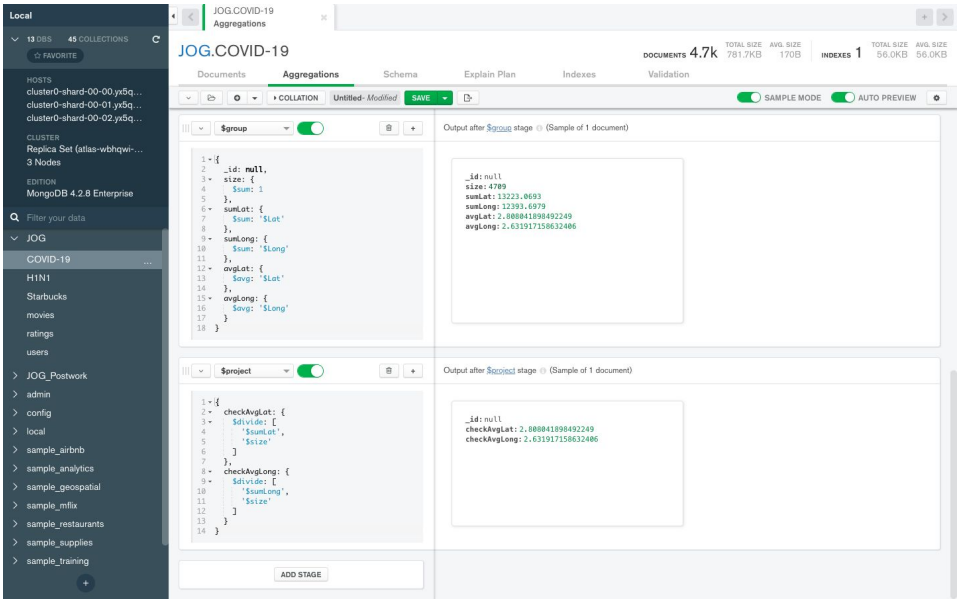
¿Cuál es el país con mayor número de muertes?

```
[{$group: {
  _id: '$Region',
  maxDeathsPerCountry: {
    $sum: '$Deaths'
  }
}}, {$sort: {
  maxDeathsPerCountry: -1
}}, {$limit: 1}]
```



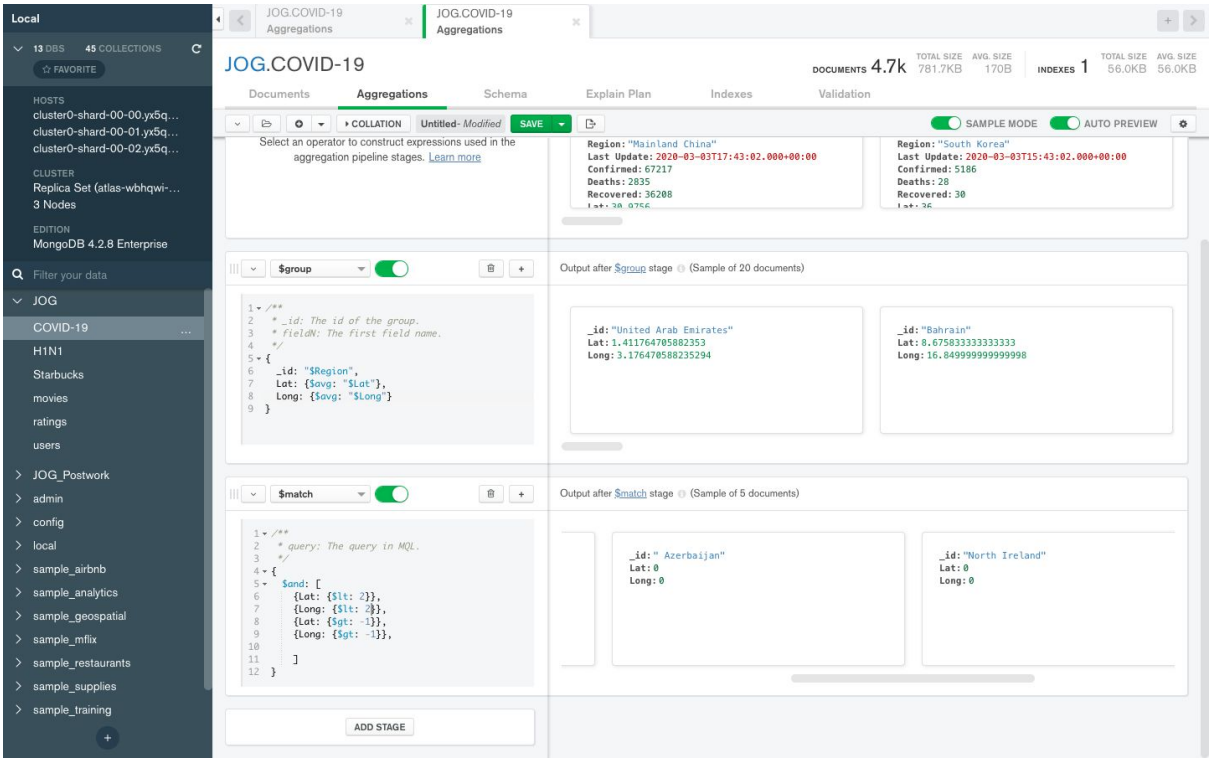
Usando las coordenadas, encuentra el epicentro del virus.

```
[{$match: {
  Lat: {
    $ne: ''
  },
  Long: {
    $ne: ''
  }
}}, {$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'
    ]
  }
}}]
```



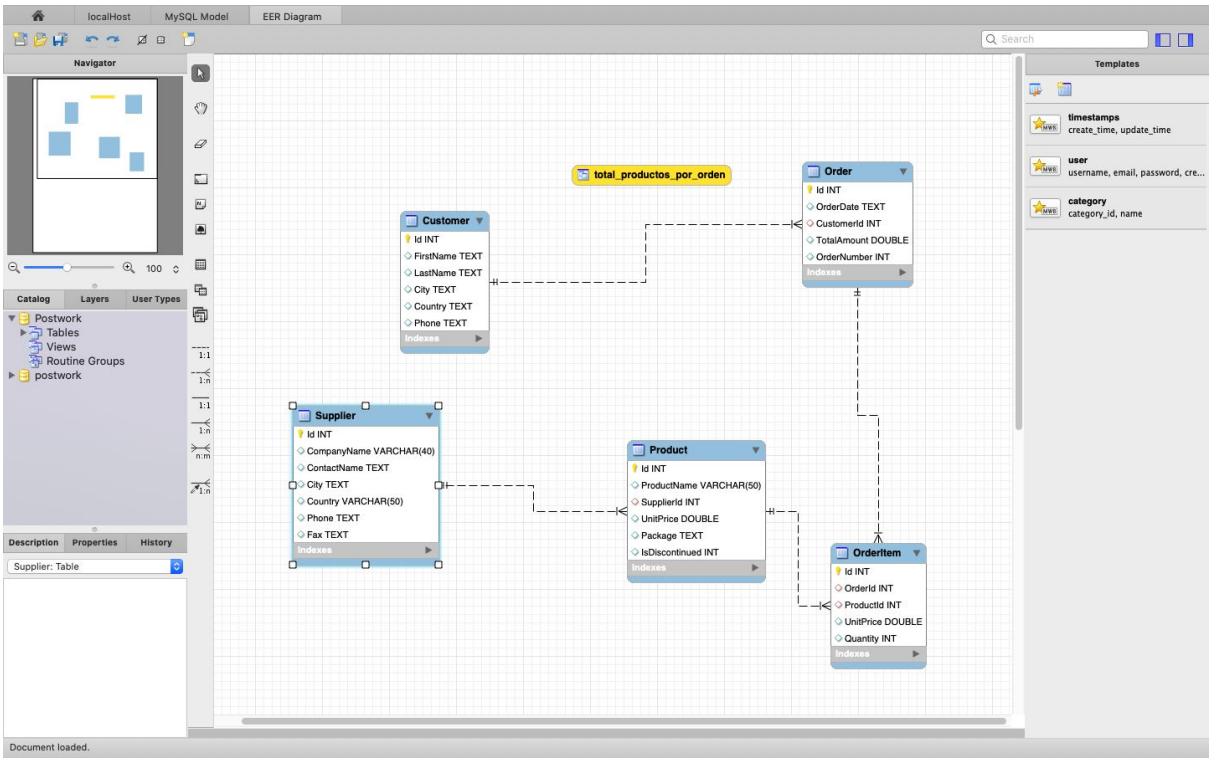
Usando el epicentro, encuentra las 5 regiones más cercanas a dicho epicentro.

```
[{$group: {
  _id: "$Region",
  Lat: {$avg: "$Lat"},
  Long: {$avg: "$Long"}
}}, {$match: {
  $and: [
    {Lat: {$lt: 2}},
    {Long: {$lt: 2}},
    {Lat: {$gt: -1}},
    {Long: {$gt: -1}},

        ]
}}]
```



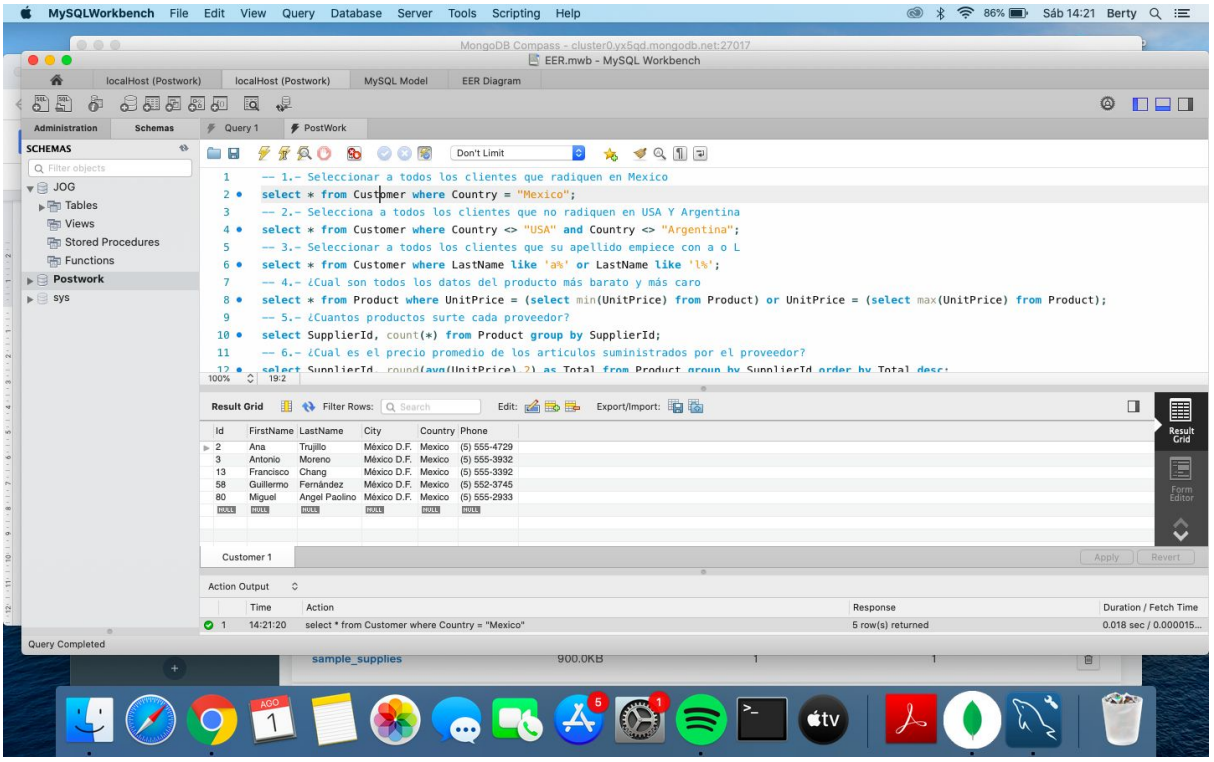
Post Work

0.- Añadir diagrama EER para entender la base de datos cargada:



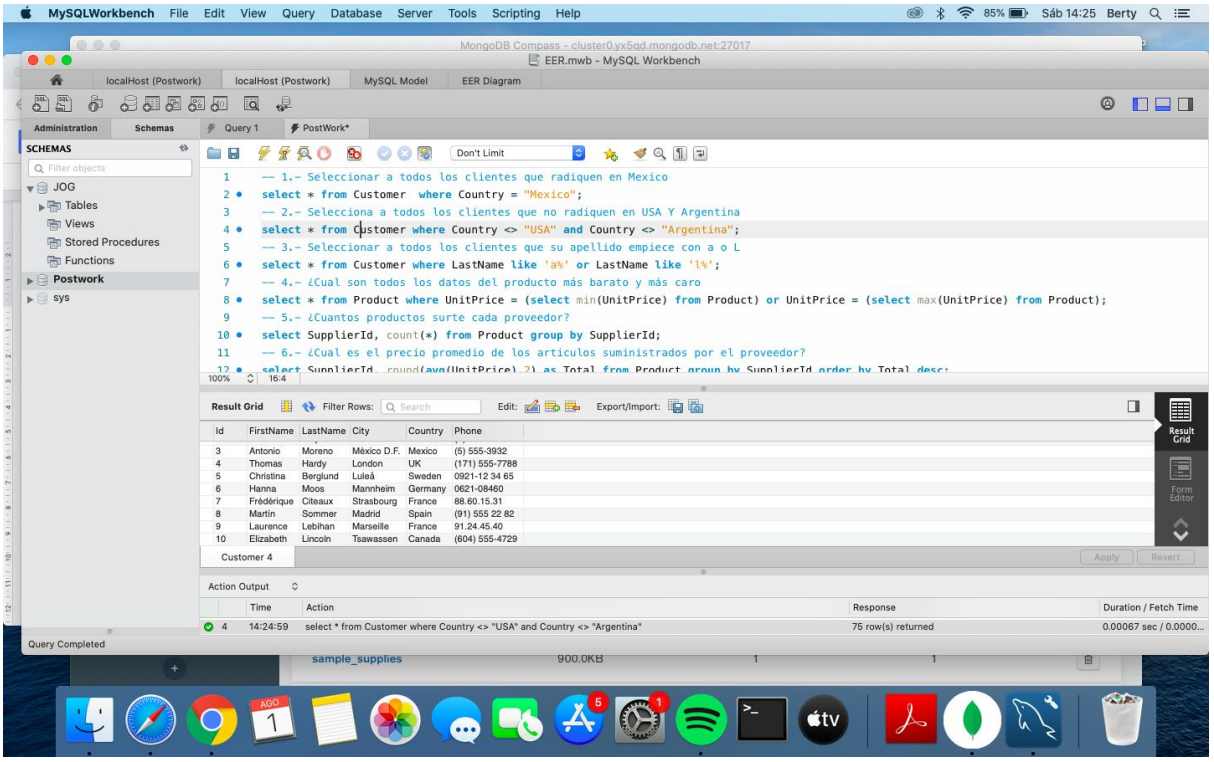
1.- Seleccionar a todos los clientes que radiquen en México.

```
select * from Customer where Country = "Mexico";
```



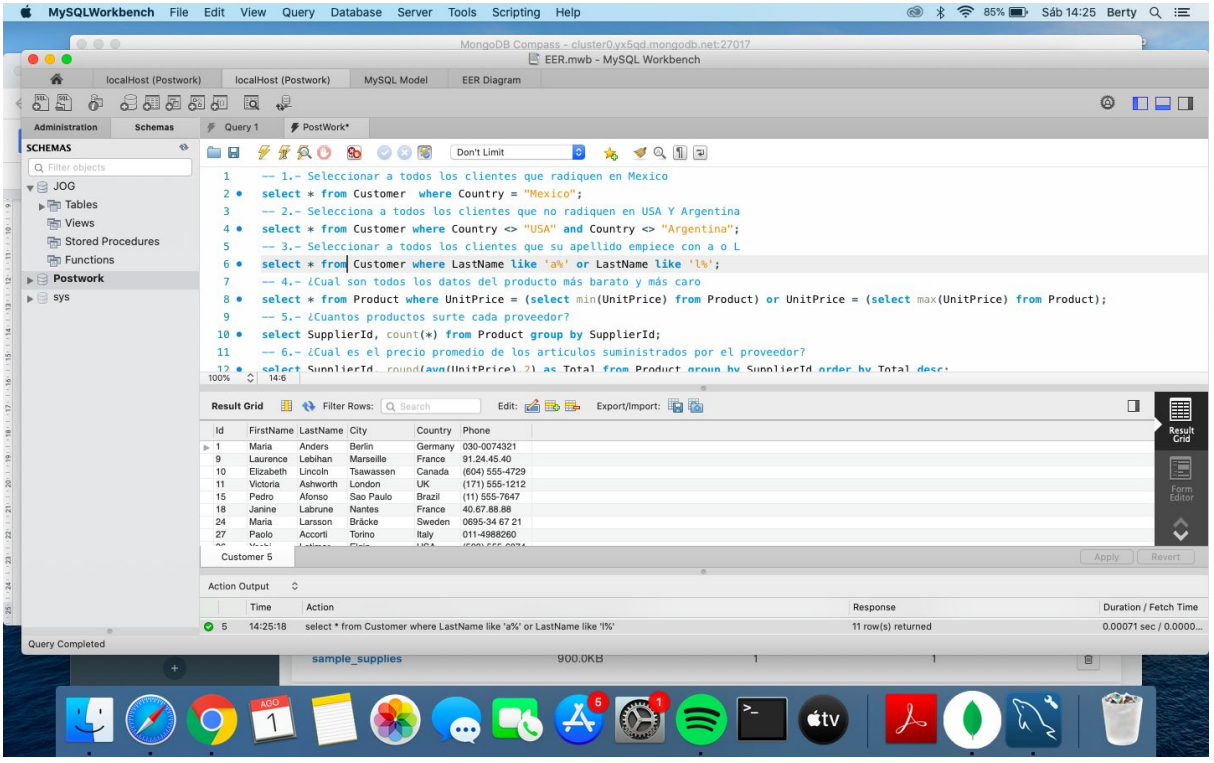
-- 2.- Selecciona a todos los clientes que no radiquen en USA Y Argentina

```
select * from Customer where Country <> "USA" and Country <> "Argentina";
```



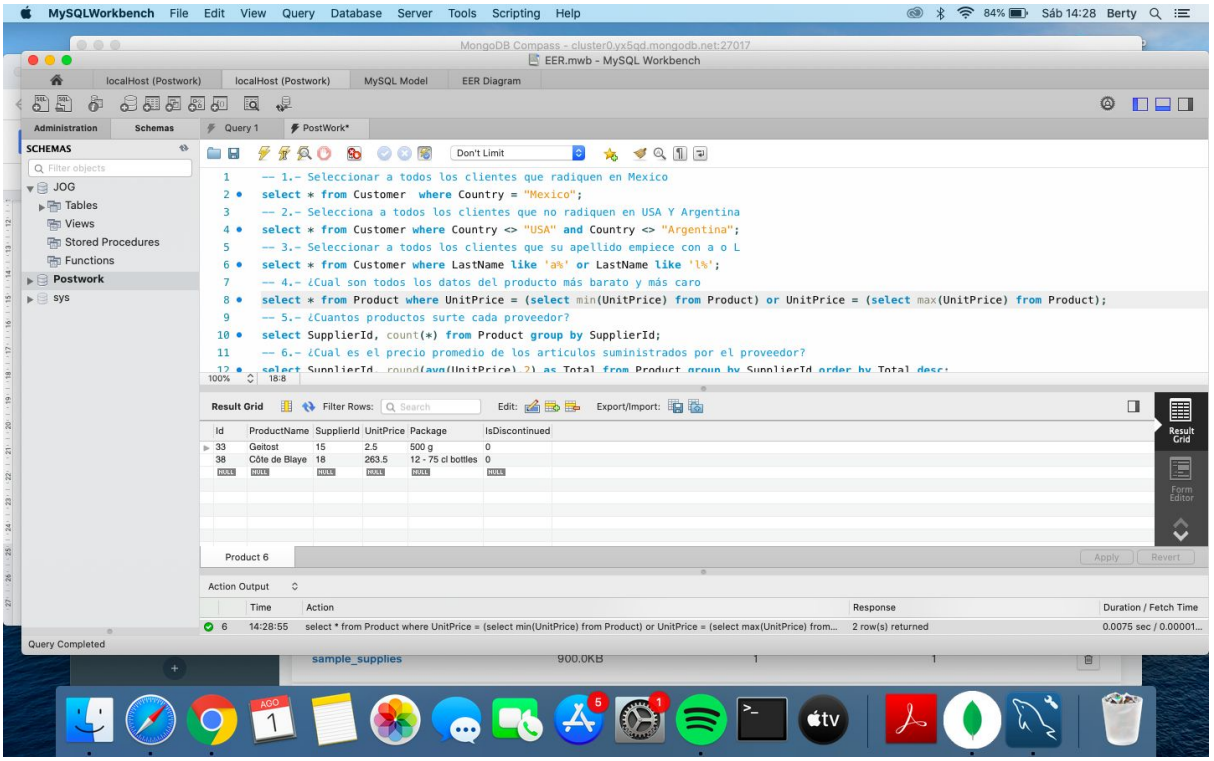
3.- Seleccionar a todos los clientes que su apellido empiece con a o L

```
select * from Customer where LastName like 'a%' or LastName like 'l%';
```



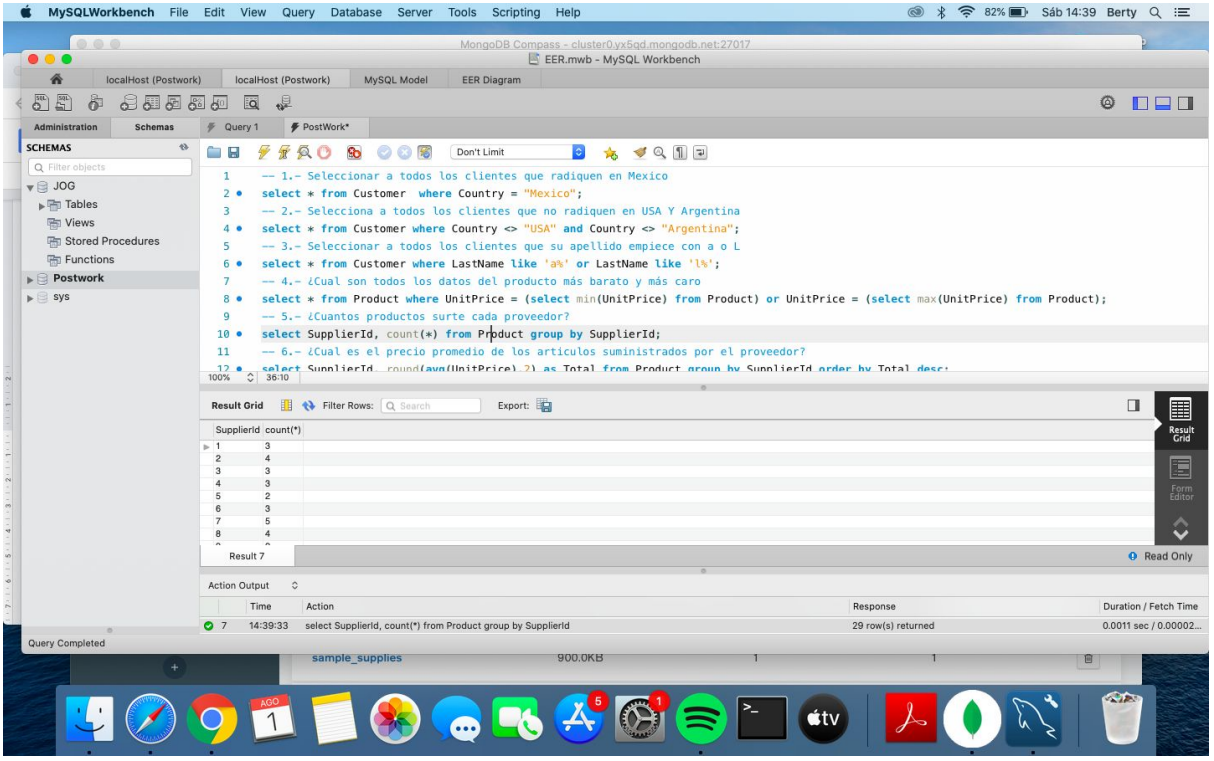
4.- ¿Cual son todos los datos del producto más barato y más caro

```
select * from Product where UnitPrice = (select min(UnitPrice) from Product) or UnitPrice =
(select max(UnitPrice) from Product);
```



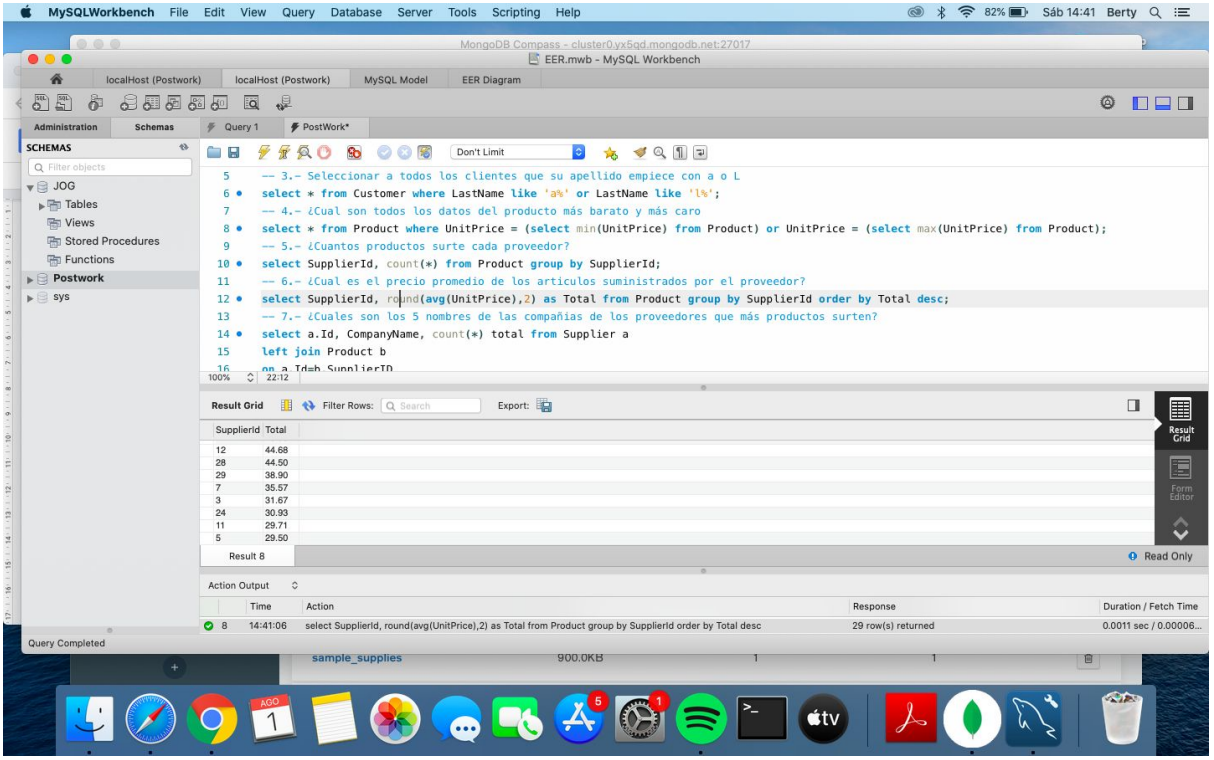
5.- ¿Cuantos productos surte cada proveedor?

```
select SupplierId, count(*) from Product group by SupplierId;
```



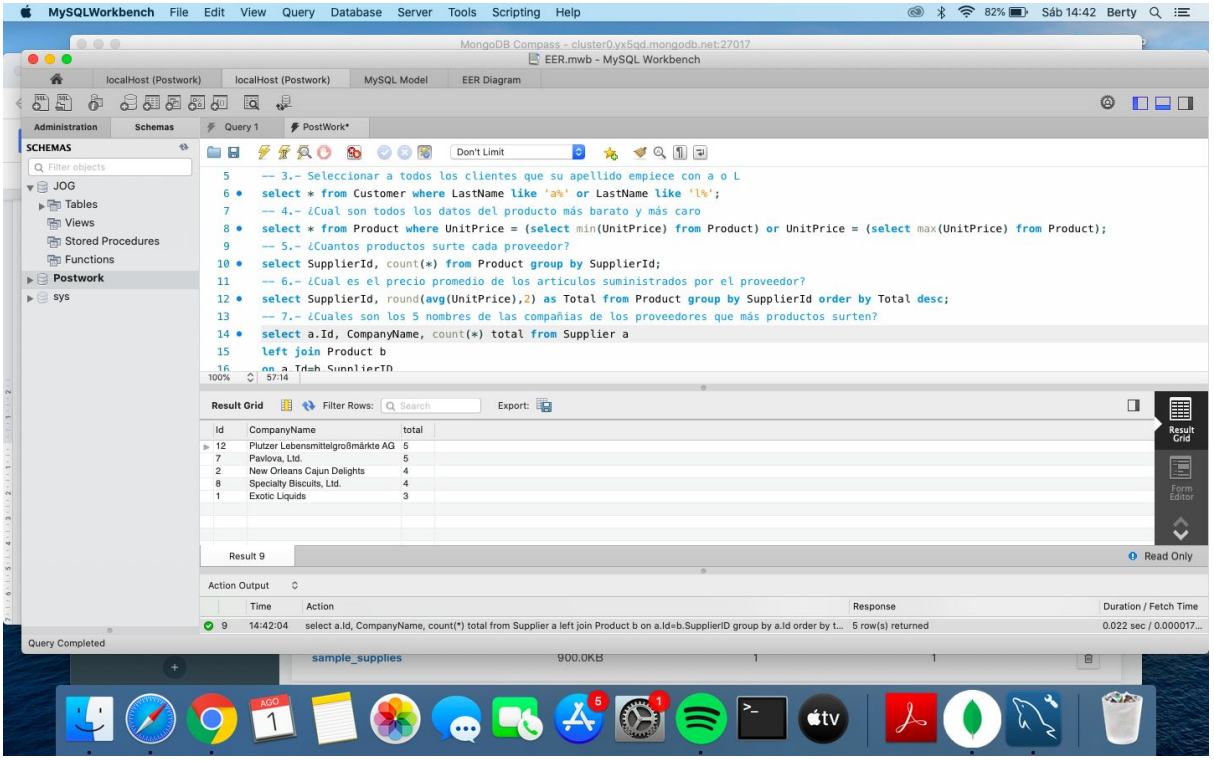
-- 6.- ¿Cual es el precio promedio de los artículos suministrados por el proveedor?

select SupplierId, round(avg(UnitPrice),2) as Total from Product group by SupplierId order by Total desc;



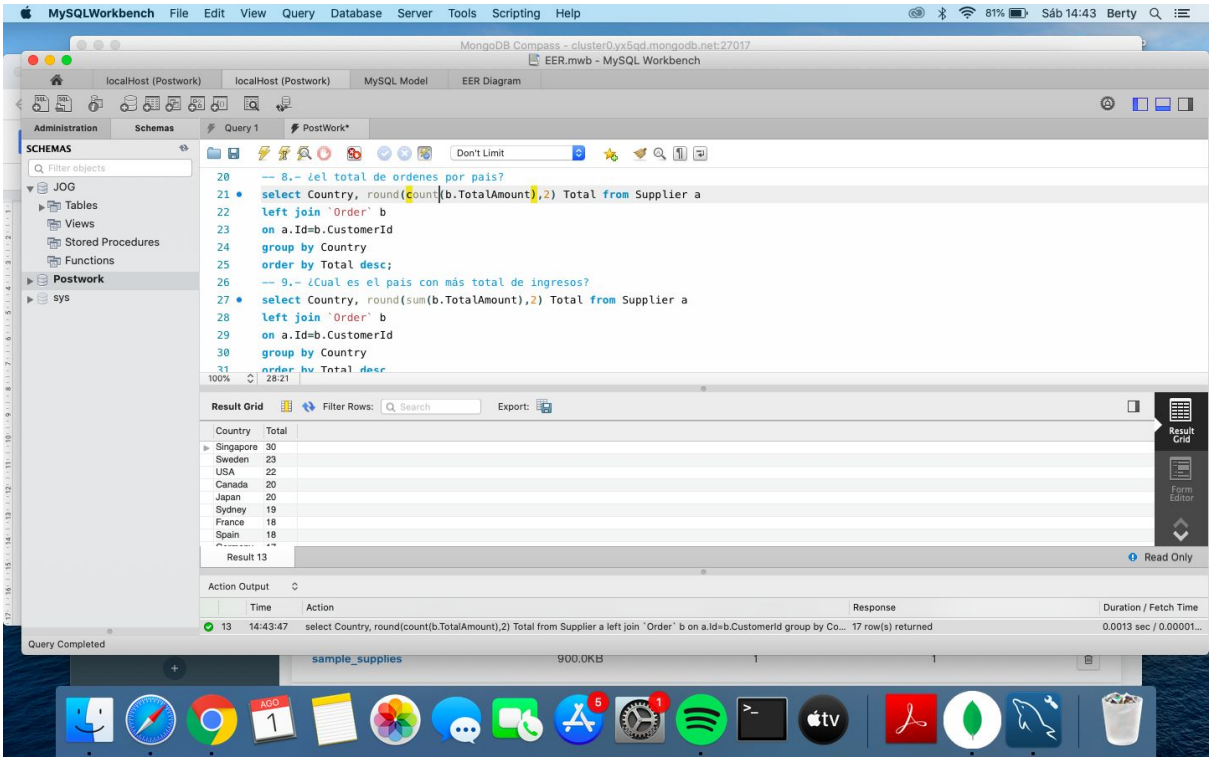
-- 7.- ¿Cuales son los 5 nombres de las compañías de los proveedores que más productos surten?

select a.Id, CompanyName, count(*) total from Supplier a



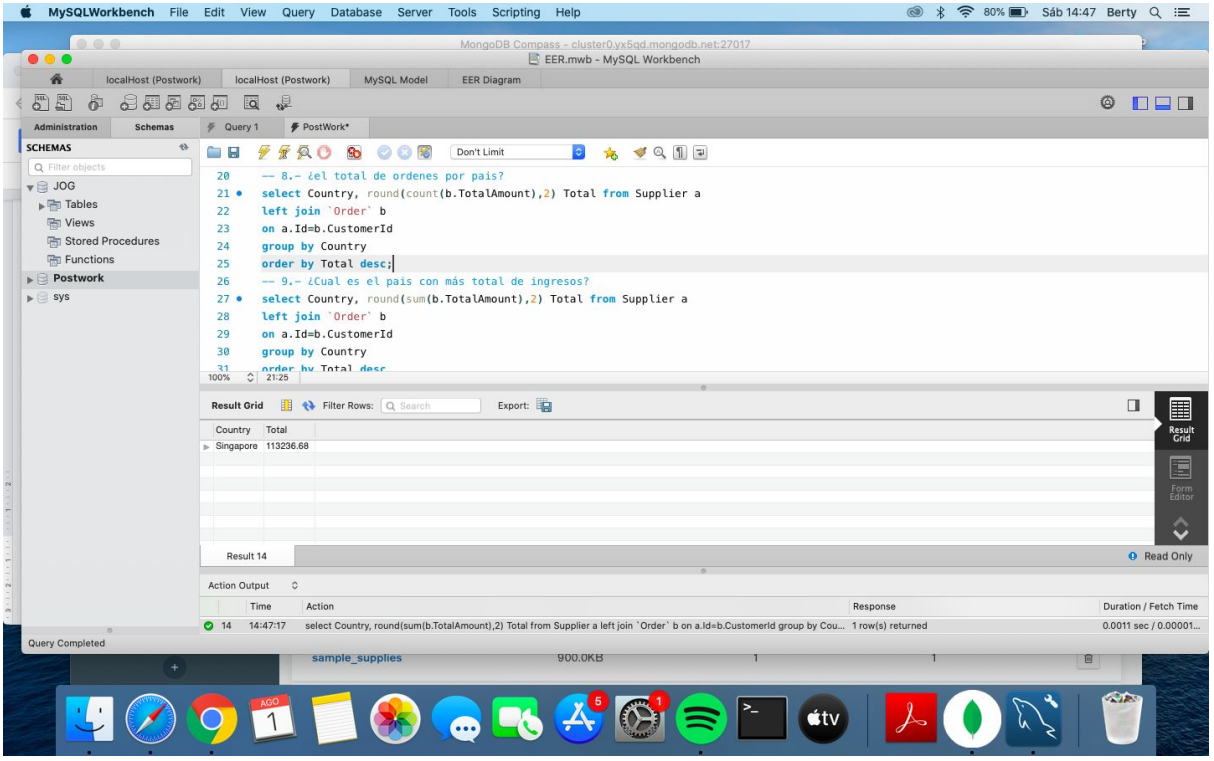
-- 8.- ¿Cual es el total de ordenes por país?

```
select Country, round(count(b.TotalAmount),2) Total from Supplier a
left join `Order` b on a.Id=b.CustomerId group by Country order by Total desc;
```



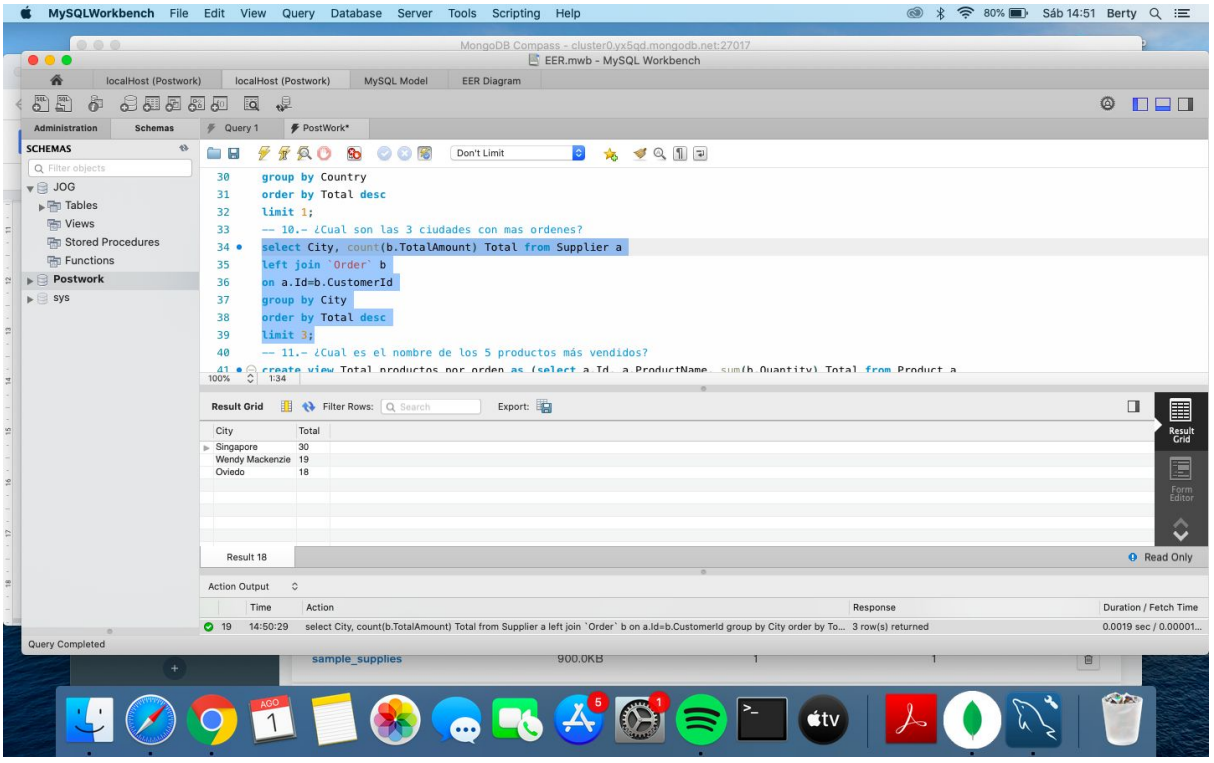
-- 9.- ¿Cuál es el país con más total de ingresos?

```
select Country, round(sum(b.TotalAmount),2) Total from Supplier a
left join `Order` b on a.Id=b.CustomerId group by Country order by Total desc limit 1;
```



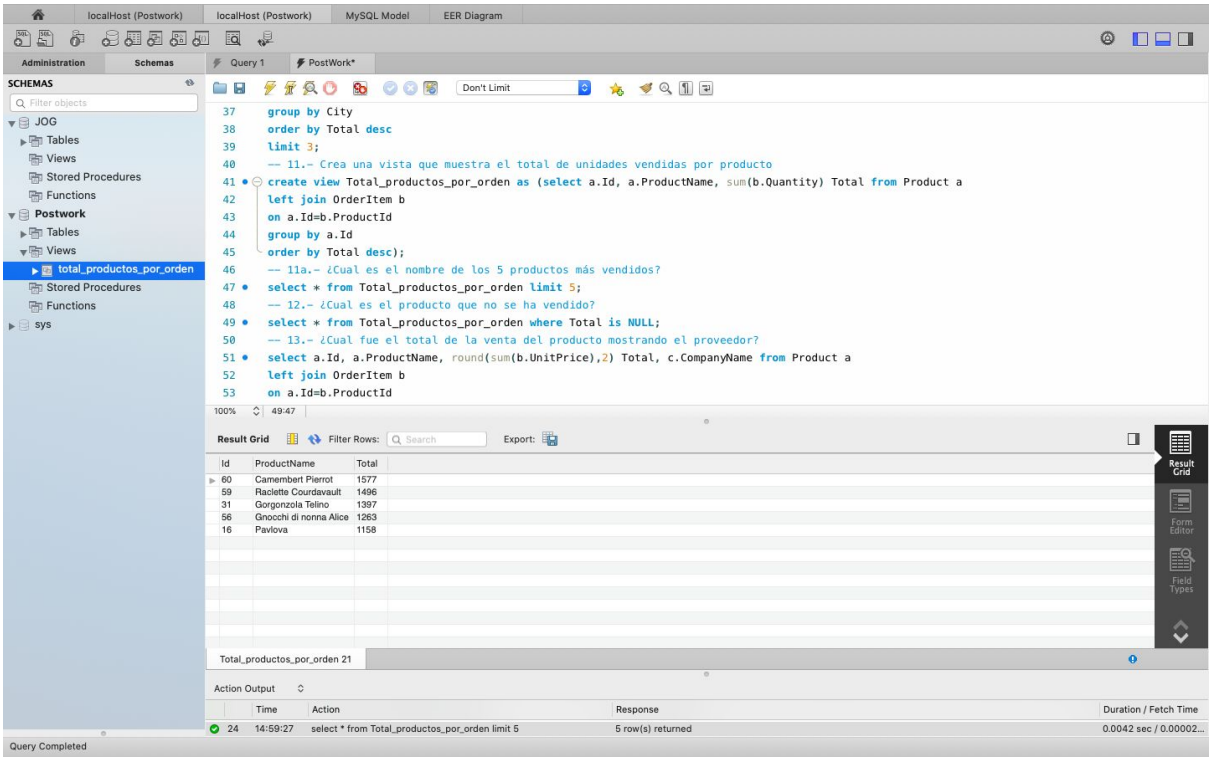
-- 10.- ¿Cual son las 3 de ciudades con más órdenes?

select City, count(b.TotalAmount) Total from Supplier a left join `Order` b on a.Id=b.CustomerId group by City order by Total desc limit 3;



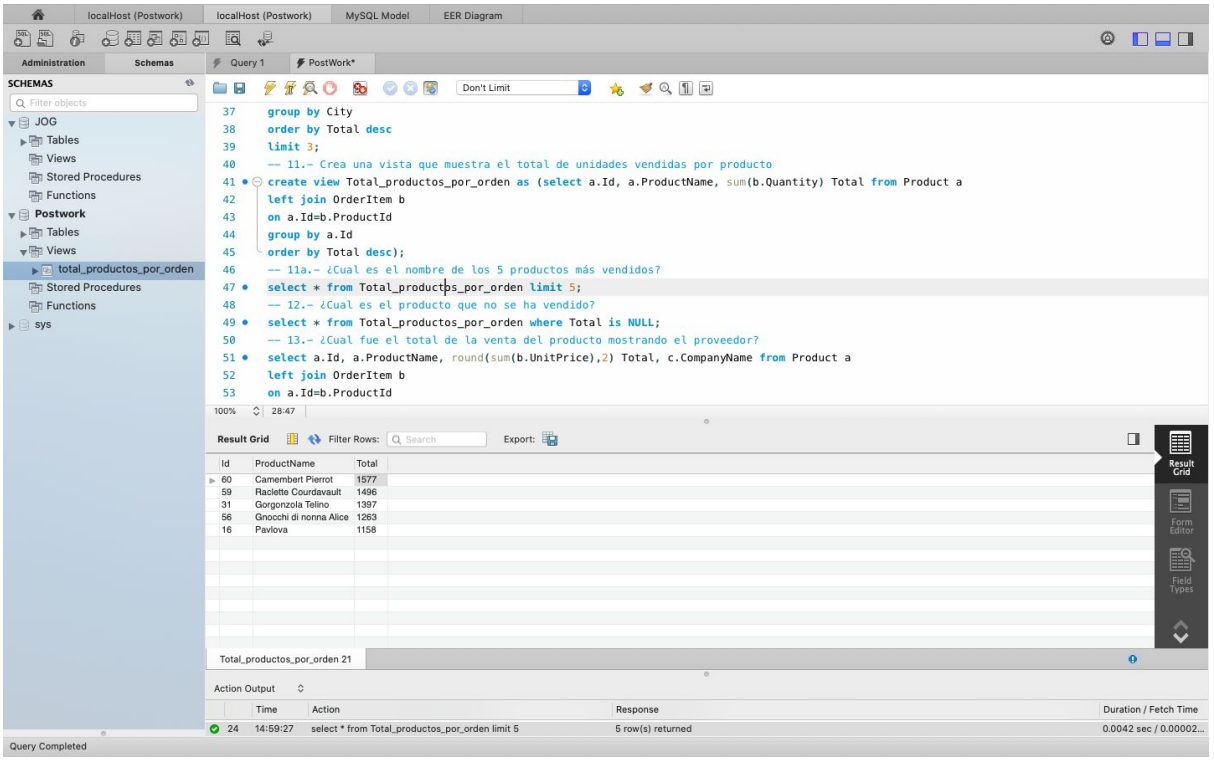
-- 11.- Crea una vista que muestra el total de unidades vendidas por producto

create view Total_productos_por_orden as (select a.Id, a.ProductName, sum(b.Quantity) Total from Product a left join OrderItem b on a.Id=b.ProductId group by a.Id order by Total desc);



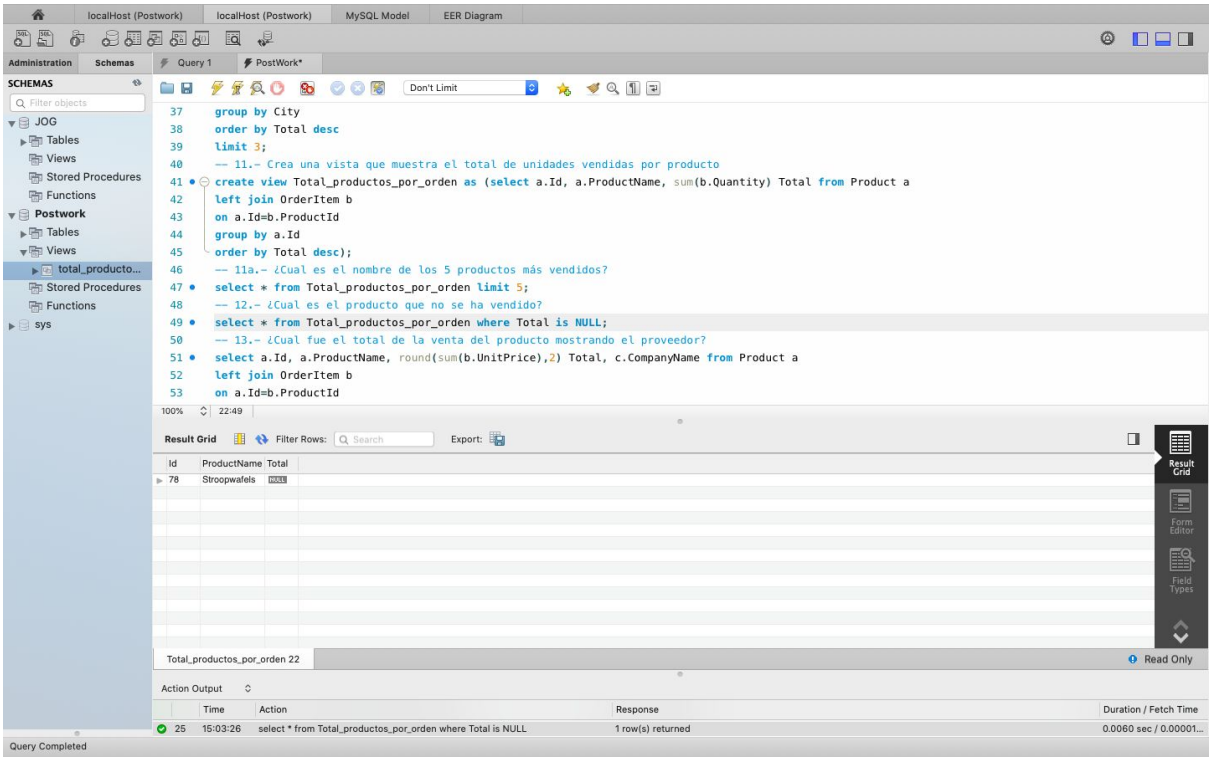
-- 11a.- ¿Cual es el nombre de los 5 productos más vendidos?

select * from Total_productos_por_orden limit 5;



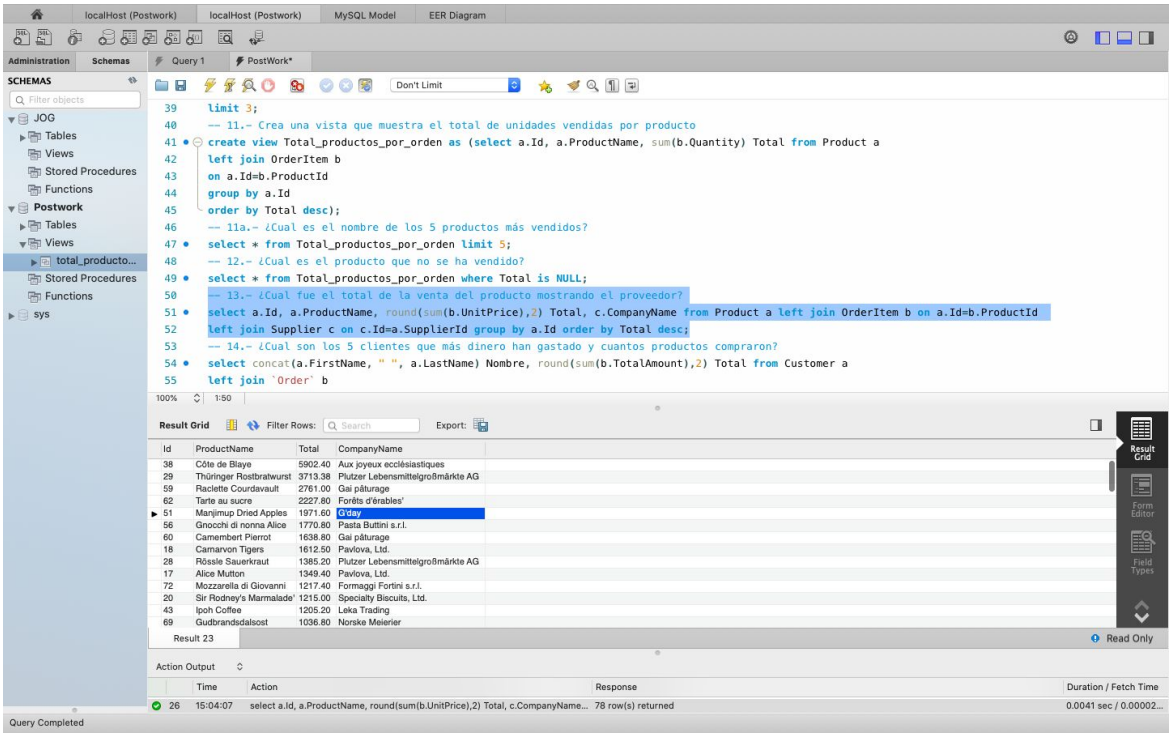
-- 12.- ¿Cual es el producto que no se ha vendido?

select * from Total_productos_por_orden where Total is NULL;



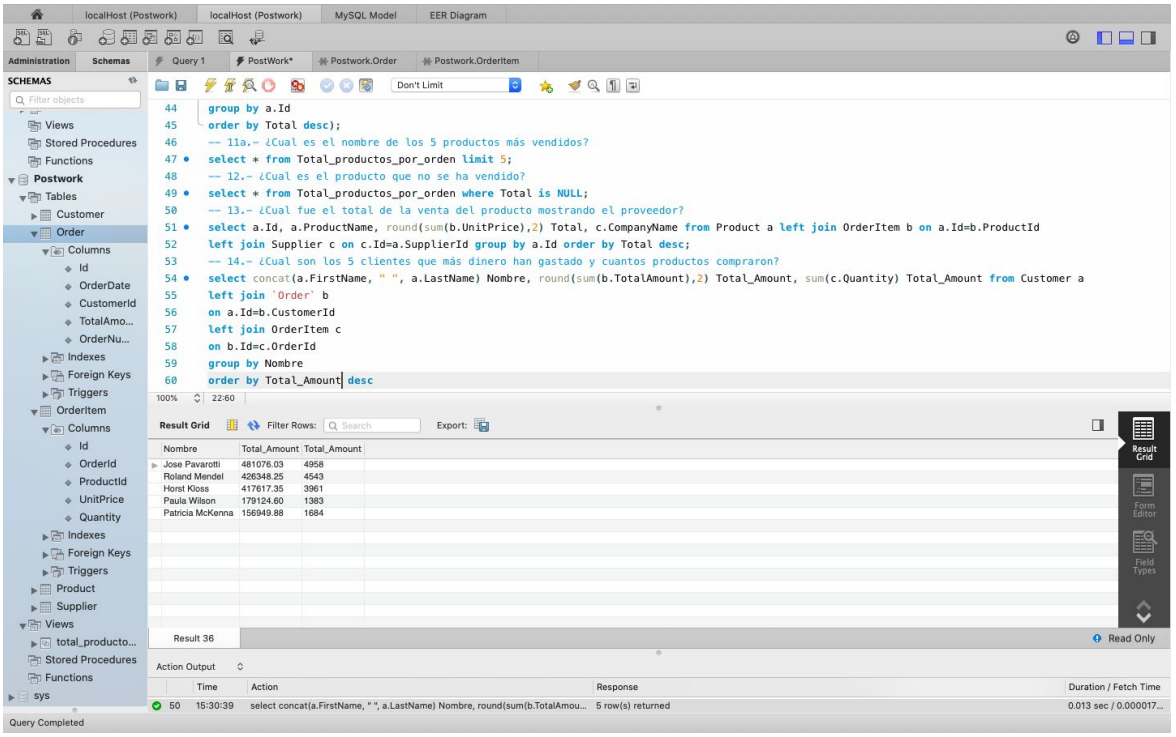
-- 13.- ¿Cual fue el total de la venta del producto mostrando el proveedor?

```
select a.Id, a.ProductName, round(sum(b.UnitPrice),2) Total, c.CompanyName from Product a left join OrderItem b on a.Id=b.ProductId  left join Supplier c on c.Id=a.SupplierId group by a.Id order by Total desc;
```



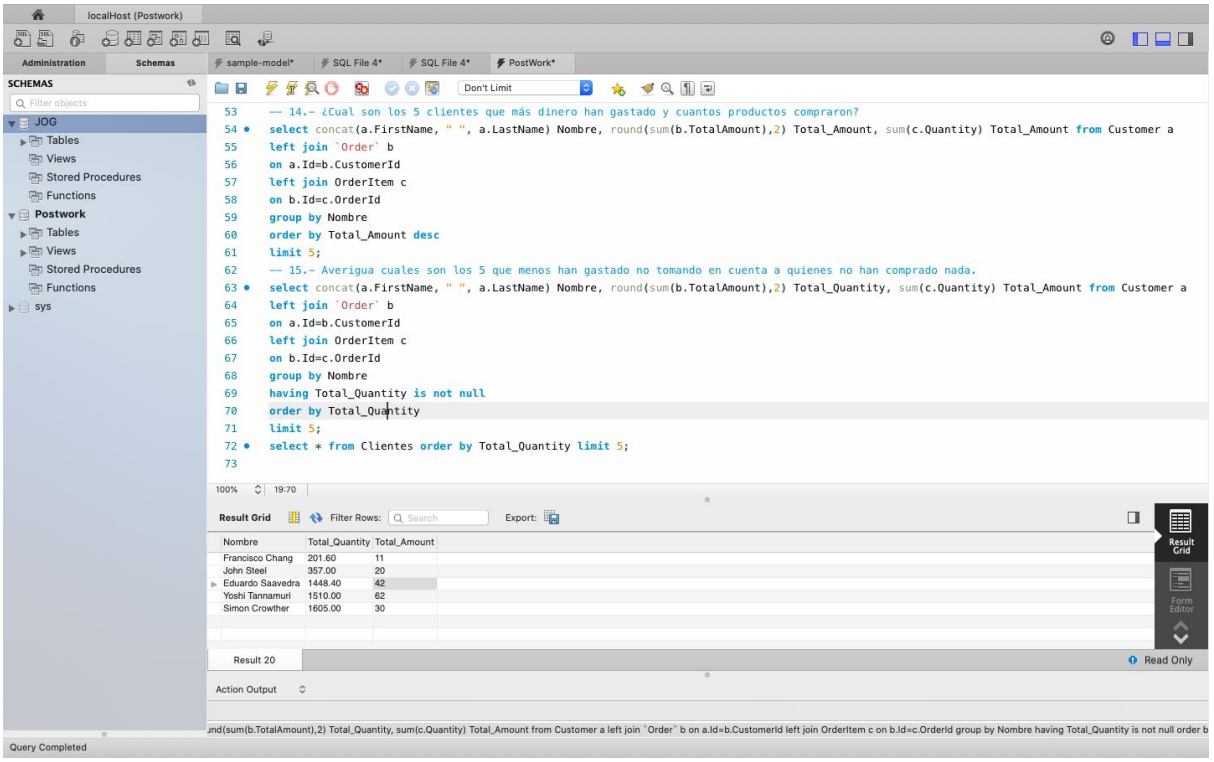
-- 14.- ¿Cual son los 5 clientes que más productos compraron y cuanto dinero han gastado?

```
select concat(a.FirstName, " ", a.LastName) Nombre, round(sum(b.TotalAmount),2) Total_Quantity, sum(c.Quantity) Total_Amount from Customer a left join `Order` b on a.Id=b.CustomerId left join OrderItem c on b.Id=c.OrderId group by Nombre order by Total_Quantity desc limit 5;
```



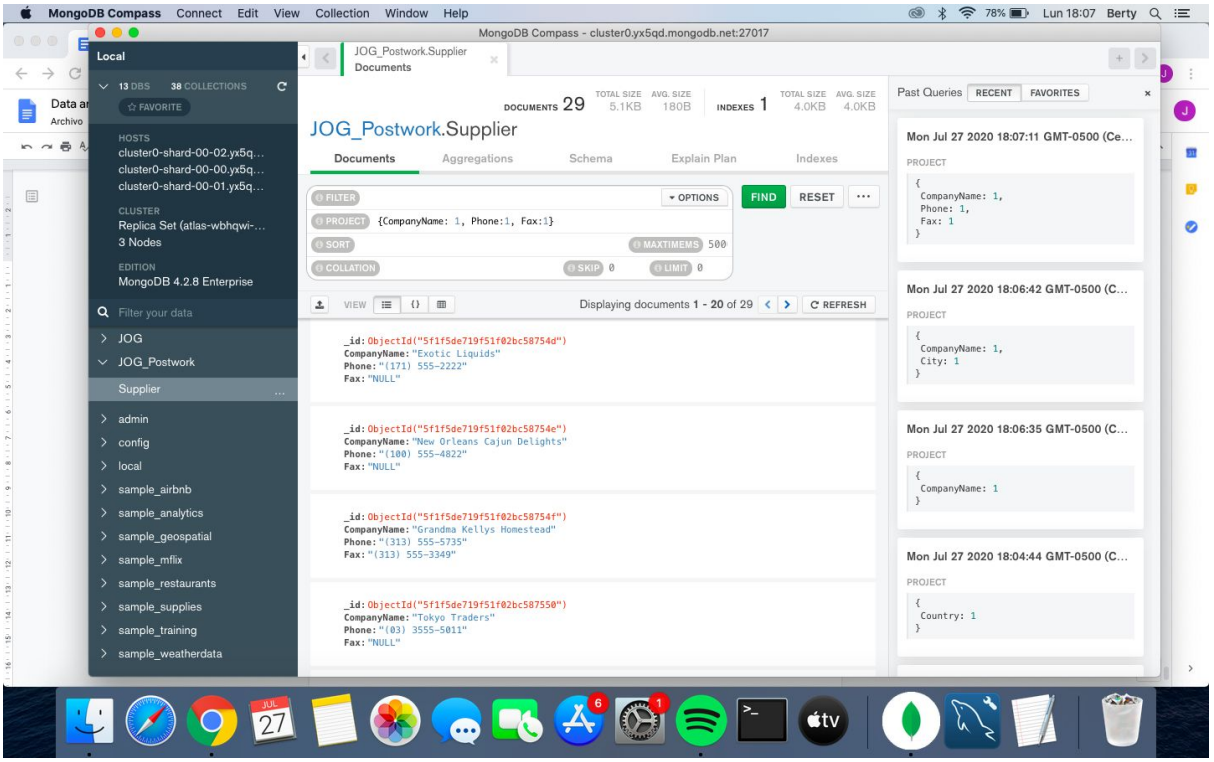
-- 15.- Averigua cuales son los 5 que menos han gastado no tomando en cuenta a quienes no han comprado nada.

```
select concat(a.FirstName, " ", a.LastName) Nombre, round(sum(b.TotalAmount),2) Total_Quantity, sum(c.Quantity) Total_Amount from Customer a left join `Order` b on a.Id=b.CustomerId left join OrderItem c on b.Id=c.OrderId group by Nombre having Total_Quantity is not null order by Total_Quantity limit 5;
```



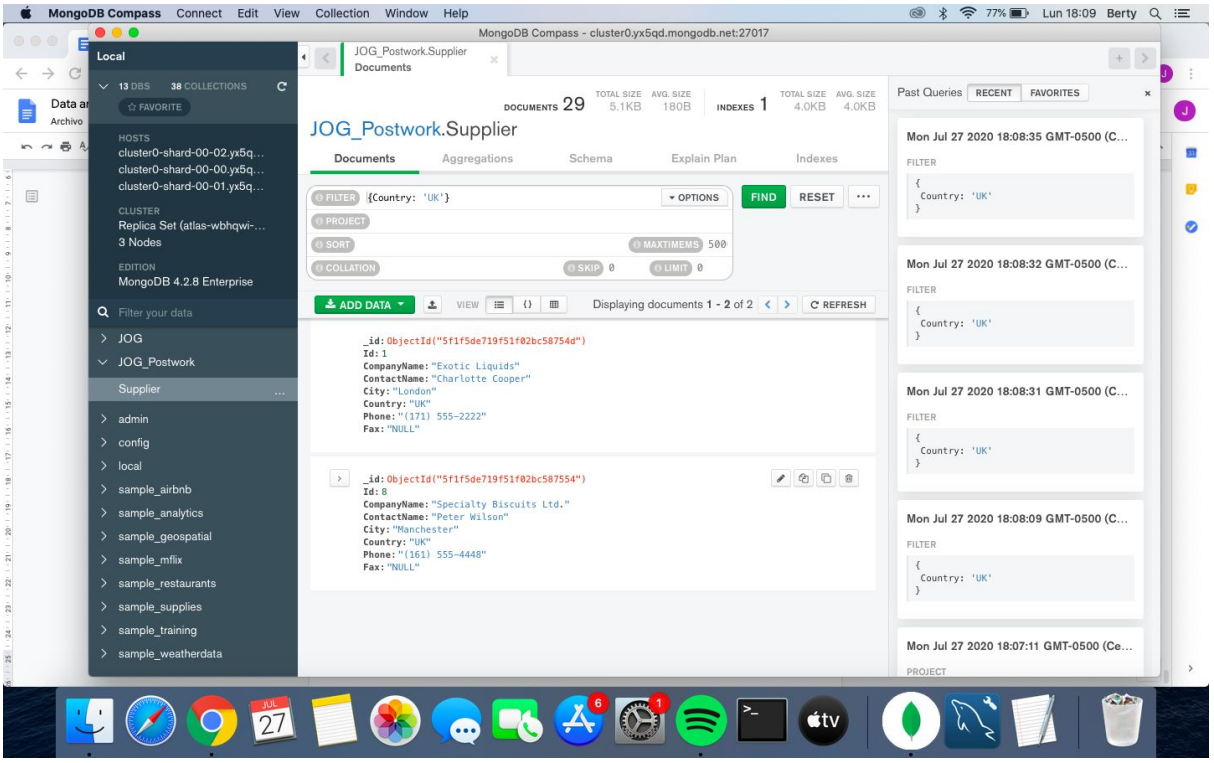
16.- Obtén los datos de contacto de cada compañía de proveedores

```
{
 project: {
  CompanyName: 1,
  Phone: 1,
  Fax: 1
 }
}
```



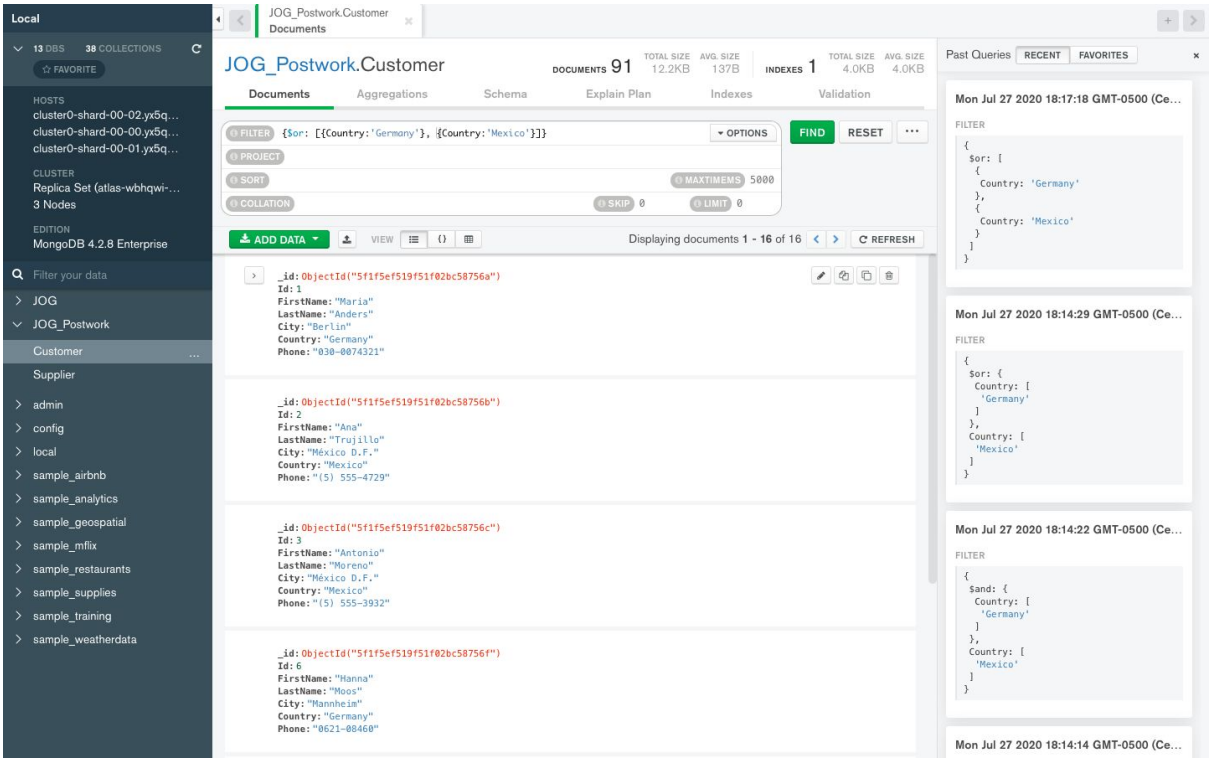
17.- Filtra a los proveedores del reino unido

```
{
 filter: {
  Country: 'UK'
 }
}
```



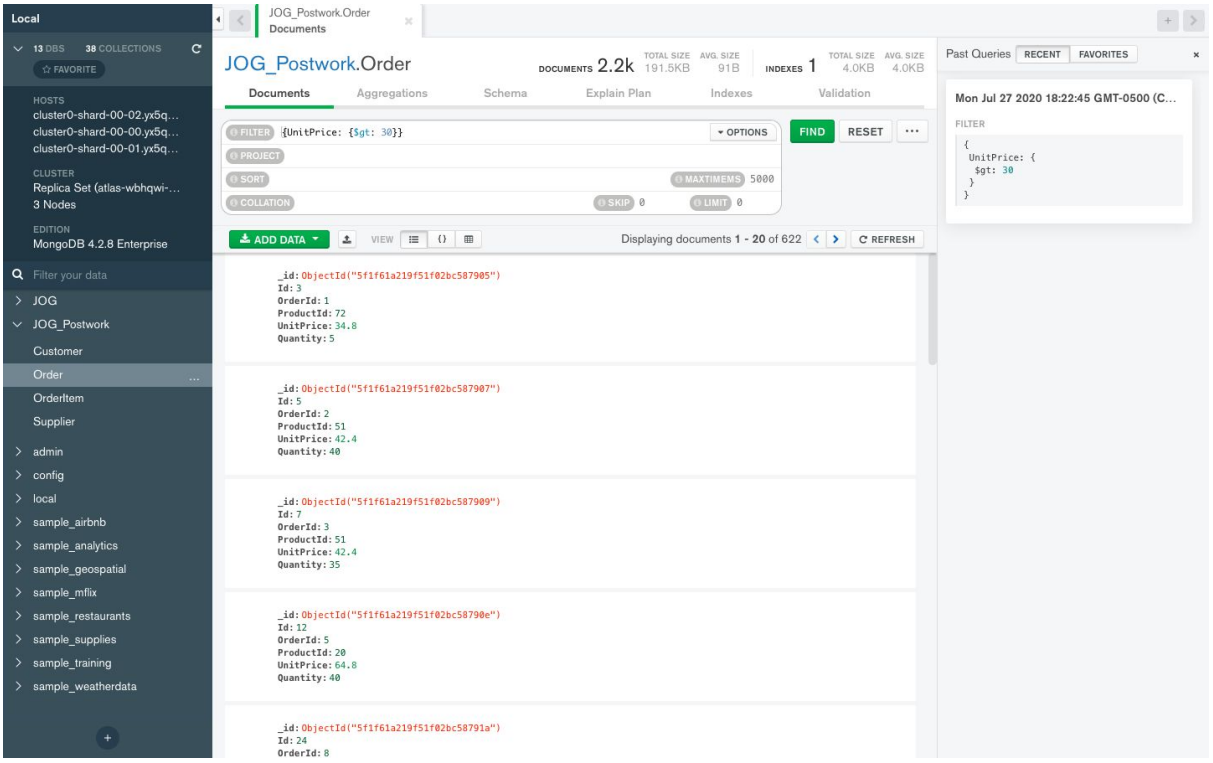
18.- Muestra a los clientes solo de México o Alemania

```
{
 filter: {
  $or: [
   {
    Country: 'Germany'
   },
   {
    Country: 'Mexico'
   }
  ]
 }
}
```



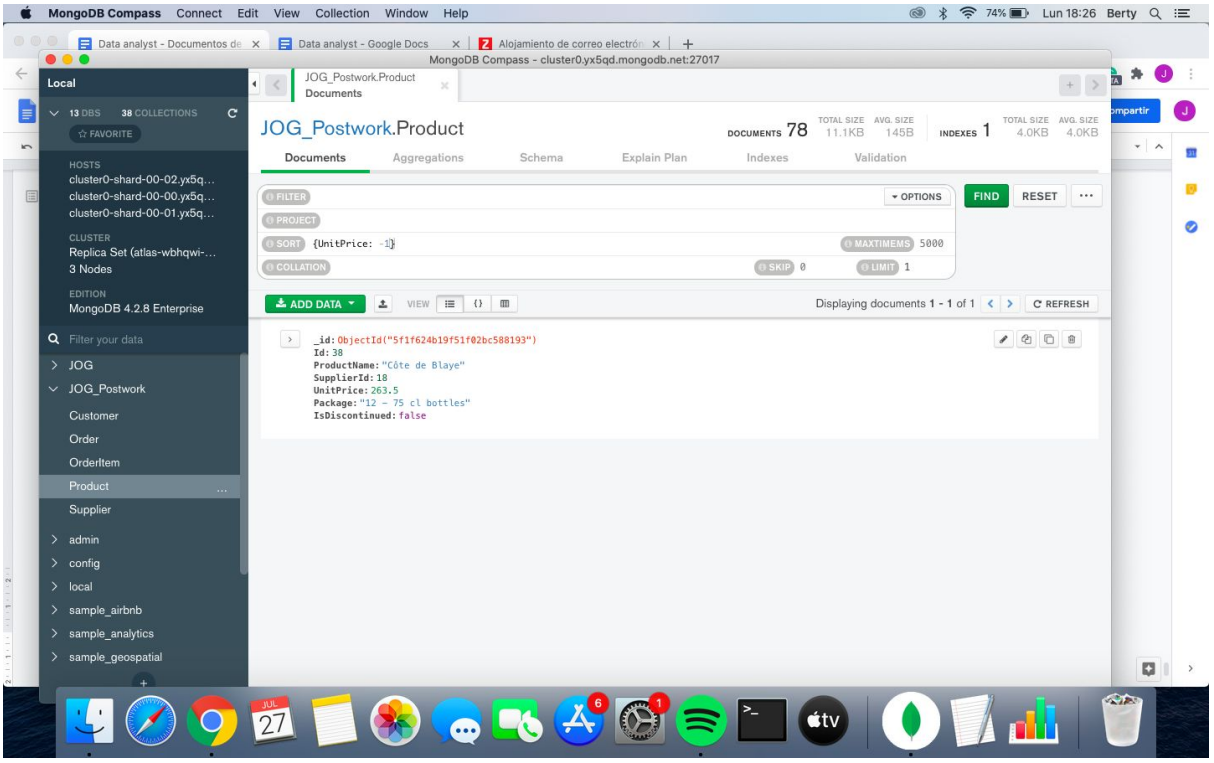
19.- Ubicar todas las órdenes con precio de unidad mayor a 30

```
{
 filter: {
  UnitPrice: {
   $gt: 30
  }
 }
}
```



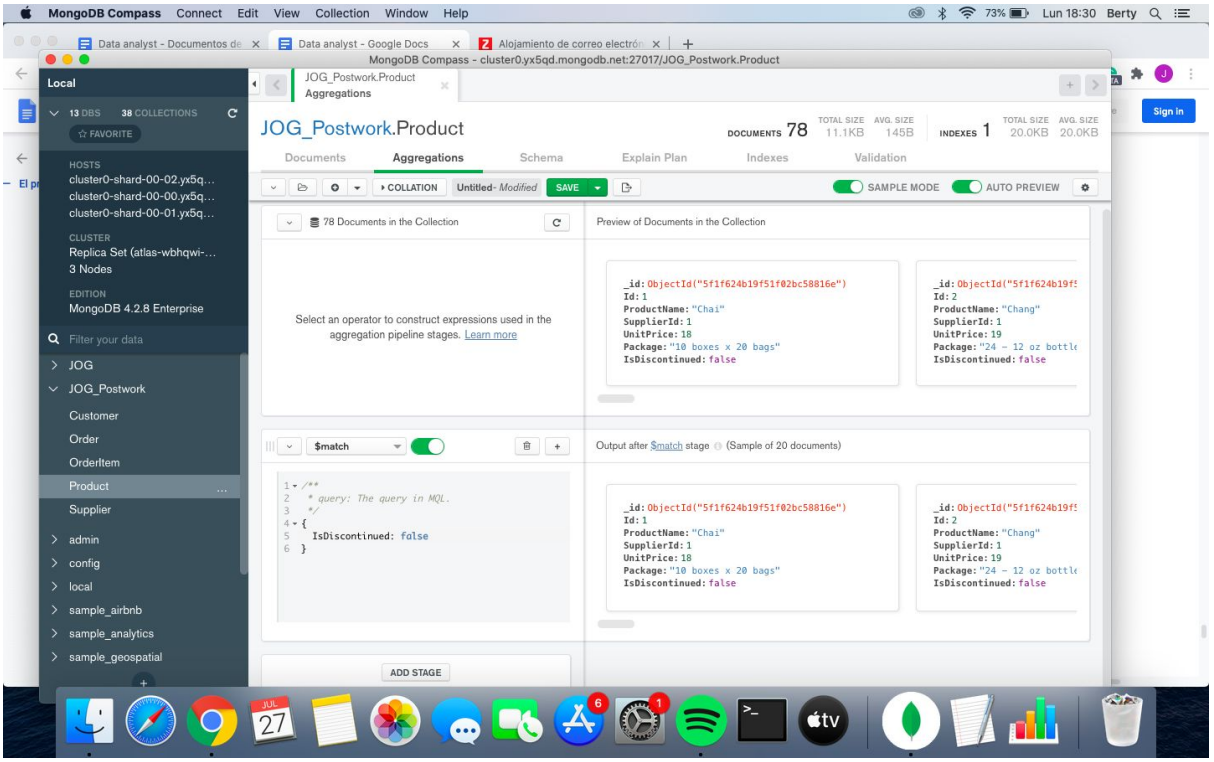
20.- ¿Cual es el producto más caro de la lista?

```
{
 sort: {
  UnitPrice: -1
 },
 limit: 1
}
```



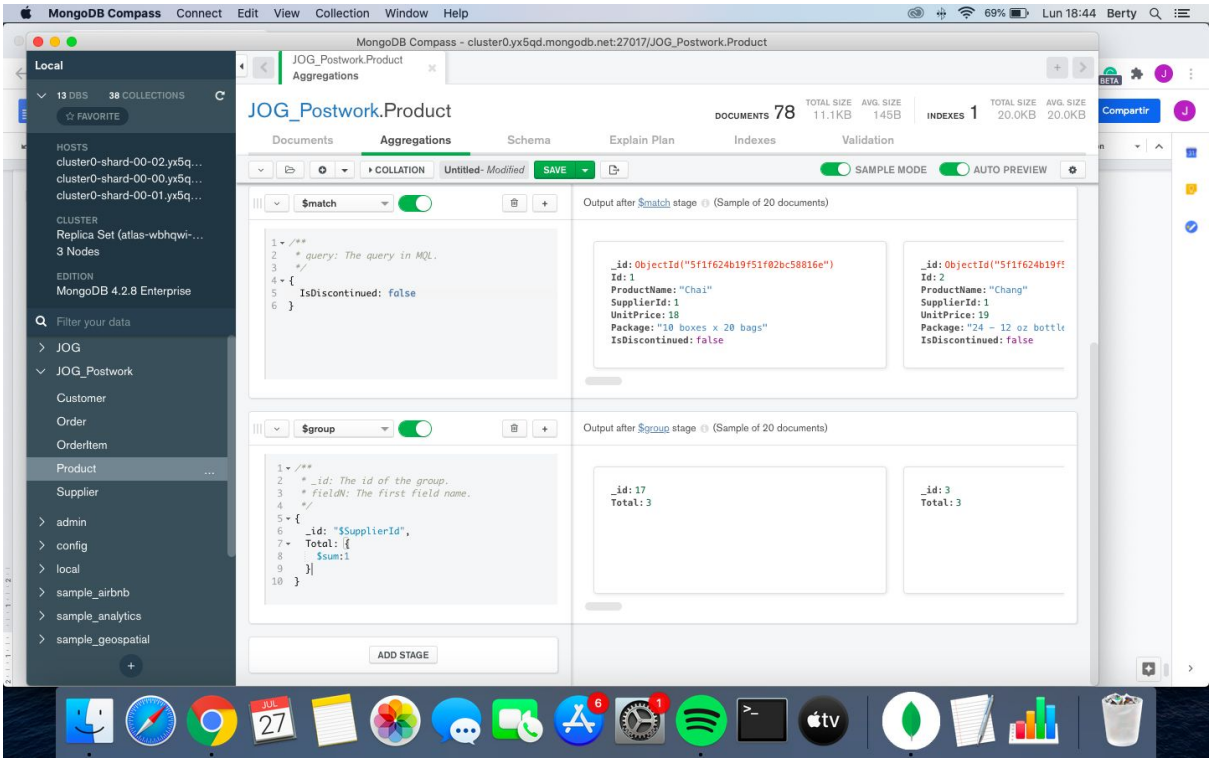
21.- ¿Muestra por agregación todos los productos que no estan descontinuados?

```
[{$match: {
  IsDiscontinued: false
}}]
```



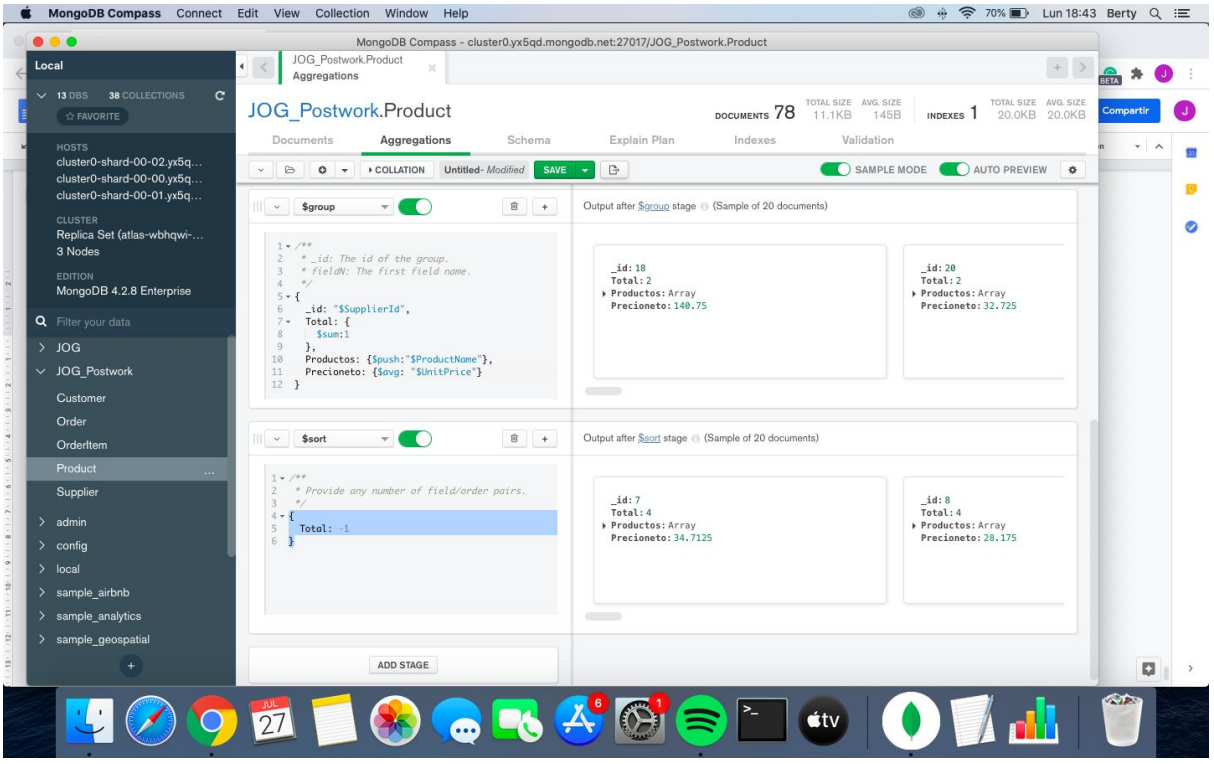
22.- ¿Cuantos productos surte cada proveedor?

```
[{$match: {
  IsDiscontinued: false
}}, {$group: {
  _id: "$SupplierId",
  Total: {
    $sum:1
  }
}}]
```

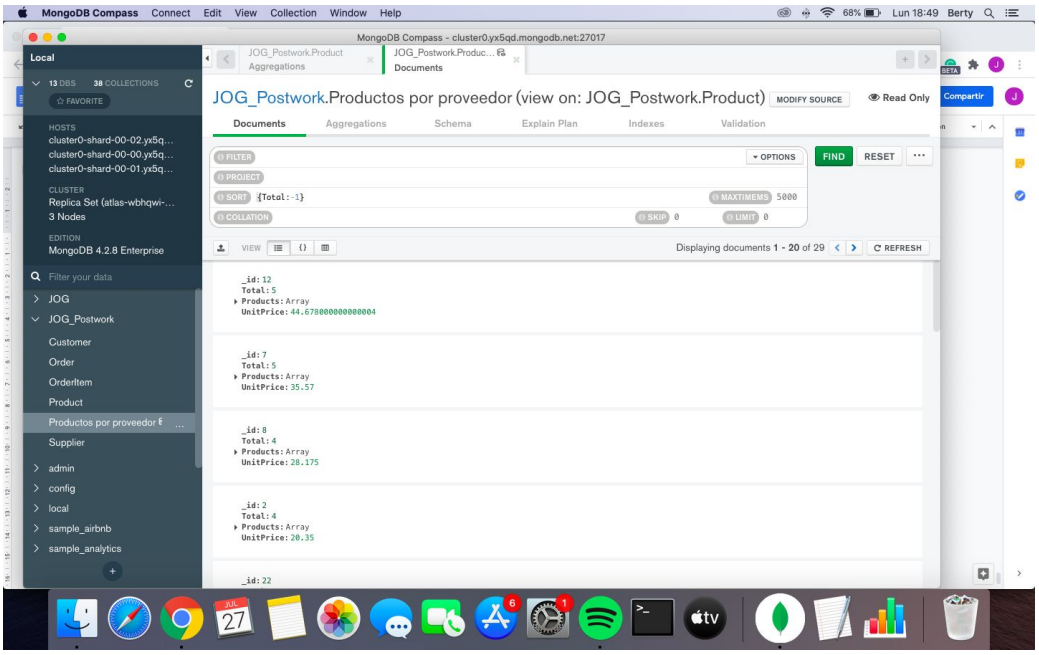


23.- Agrega como un array todos los productos suministrados no descontinuados por cada proveedor y el promedio de los precios de los productos de mayor a menor.

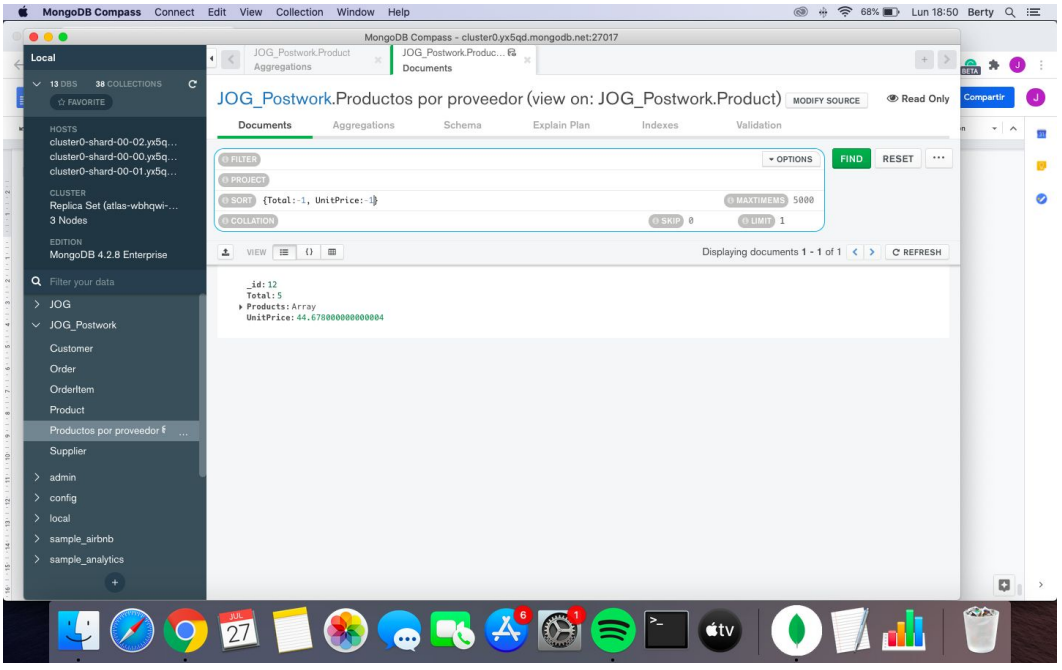
```
[{$match: {
  IsDiscontinued: false
}}, {$group: {
  _id: "$SupplierId",
  Total: {
    $sum:1
  },
  Productos: {$push:"$ProductName"},
  Precioneto: {$avg: "$UnitPrice"}
}}, {$sort: {
  Total: -1
}}]
```



24.- Crea una vista de esta tabla y arroja al proveedor con más productos y el precio promedio más alto

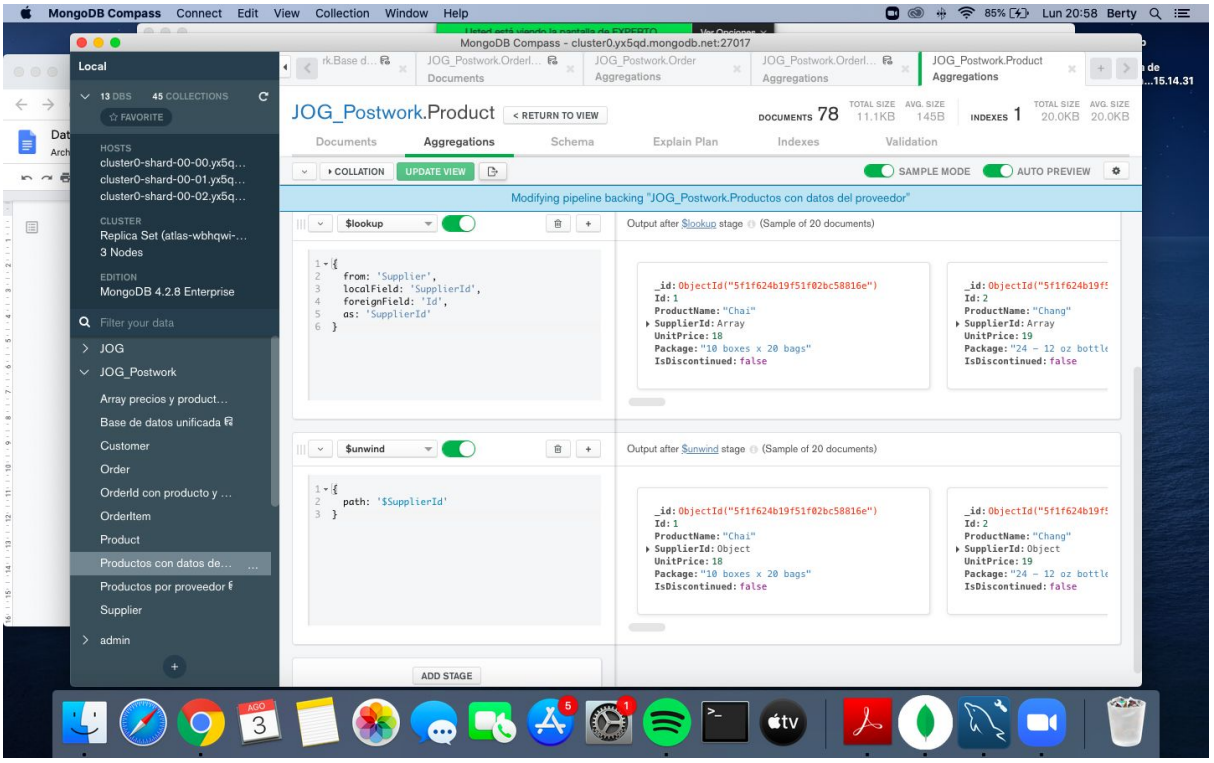


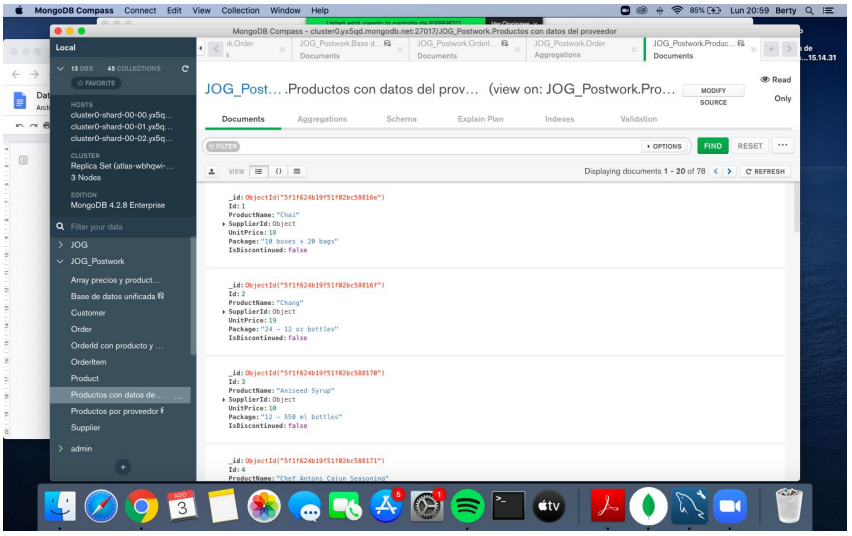

```
{
 sort: {
  Total: 1,
  UnitPrice: -1
 },
 limit: 1
}
```



25.- De la vista generada crea un documento por cada producto existente que incluya los datos de cada proveedor por producto. Usa esta vista para crear una vista

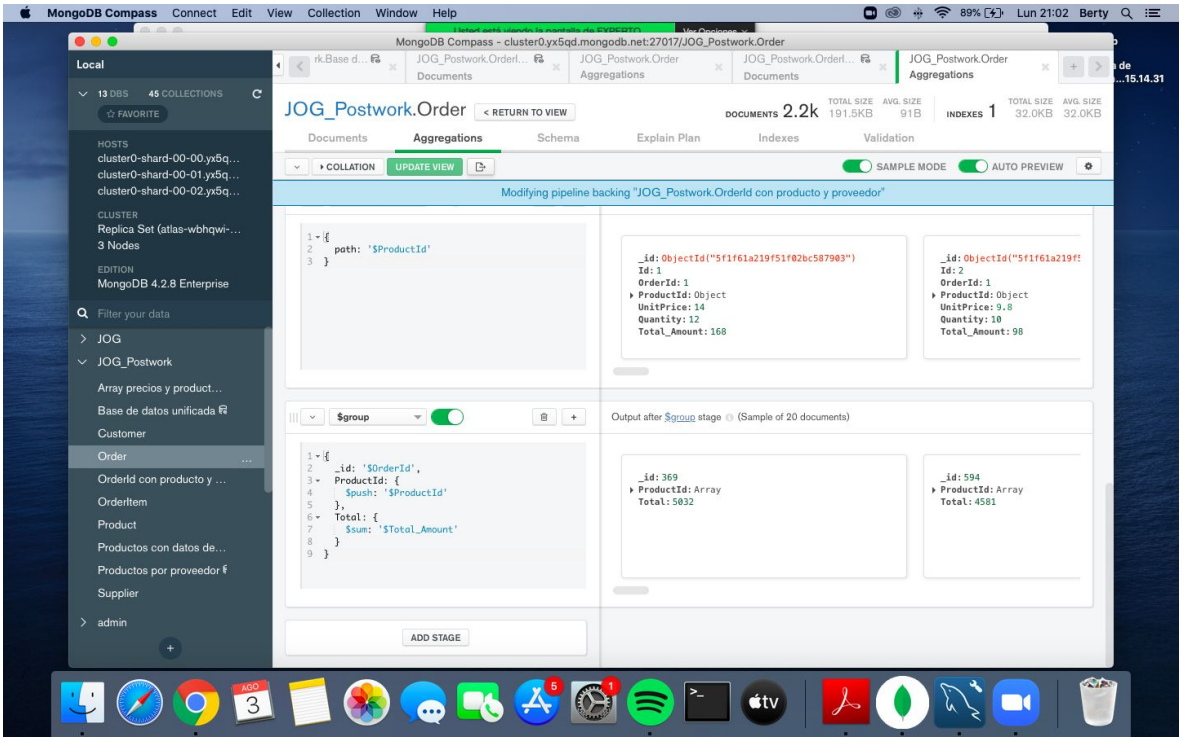
```
[{$lookup: {
  from: 'Supplier',
  localField: 'SupplierId',
  foreignField: 'Id',
  as: 'SupplierId'
}}, {$unwind: {
  path: '$SupplierId'
}}]
```





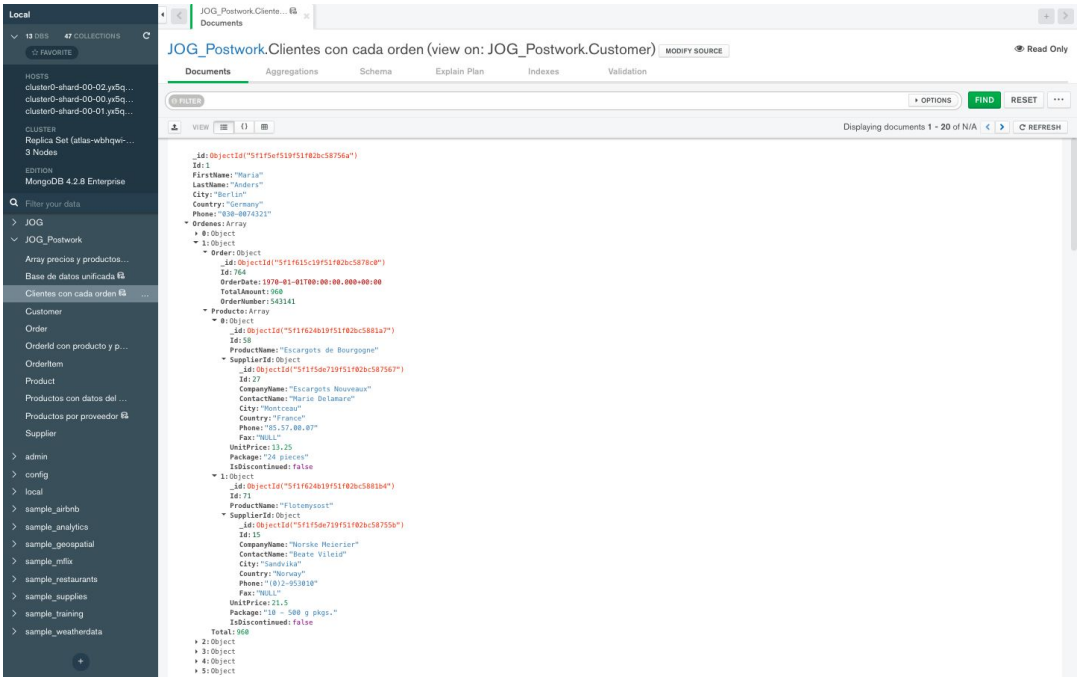
26.- Dentro de la tabla “Order” usa la vista creada para mostrar qué productos componen cada orden. Crea una vista llamada “OrderId con producto y proveedor”.

```
[{$lookup: {
  from: 'Productos con datos del proveedor',
  localField: 'ProductId',
  foreignField: 'Id',
  as: 'ProductId'
}}, {$addFields: {
  Total_Amount: {
    $multiply: [
      '$UnitPrice',
      '$Quantity'
    ]
  }
}}, {$unwind: {
  path: '$ProductId'
}}, {$group: {
  _id: '$OrderId',
  ProductId: {
    $push: '$ProductId'
  },
  Total: {
    $sum: '$Total_Amount'
  }
}}]
```



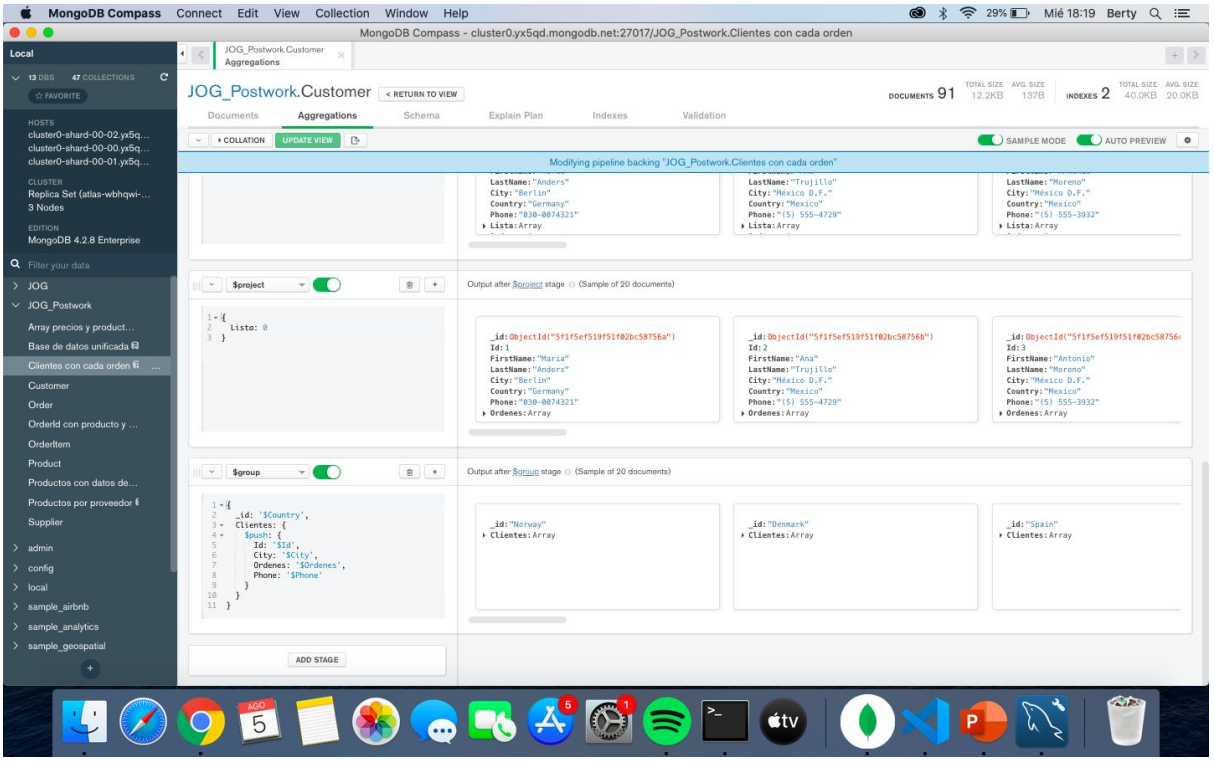
27.- Desde la tabla OrderItem, agrega los documentos de la vista anterior para mostrar por cada cliente las ordenes que ha pedido y que productos incluye. Cada producto debe tener la informaciòn del proveedor.

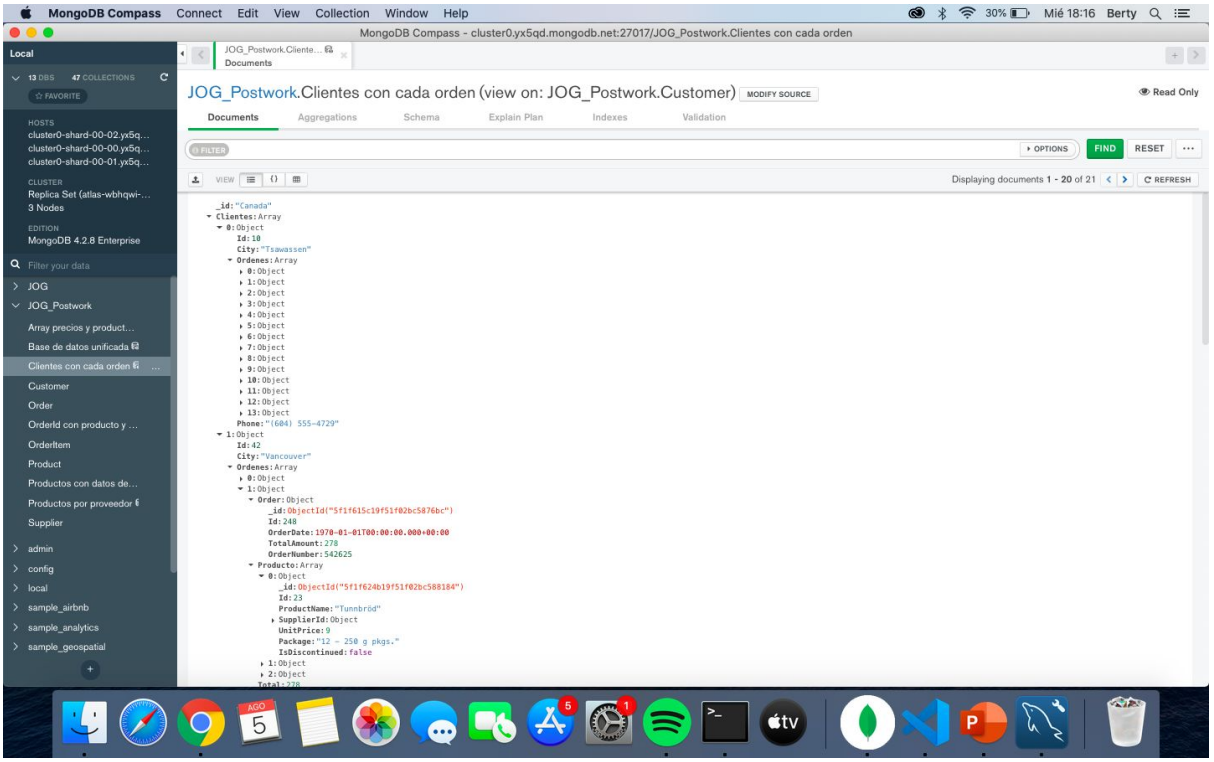
```
[{$lookup: {
  from: 'OrderItem',
  localField: '_id',
  foreignField: 'Id',
  as: 'Id'
}}, {$unwind: {
  path: '$Id'
}}, {$addFields: {
  CustomerId: '$Id.CustomerId'
}}, {$project: {
  'Id.CustomerId': 0
}}, {$group: {
  _id: '$CustomerId',
  Ordenes: {
    $push: {
      Order: '$Id',
      Producto: '$ProductId',
      Total: '$Total'
    }
  }
}}]
```



28.- Usa la agregación anterior para mostrar por ciudad el detalle de cada cliente:

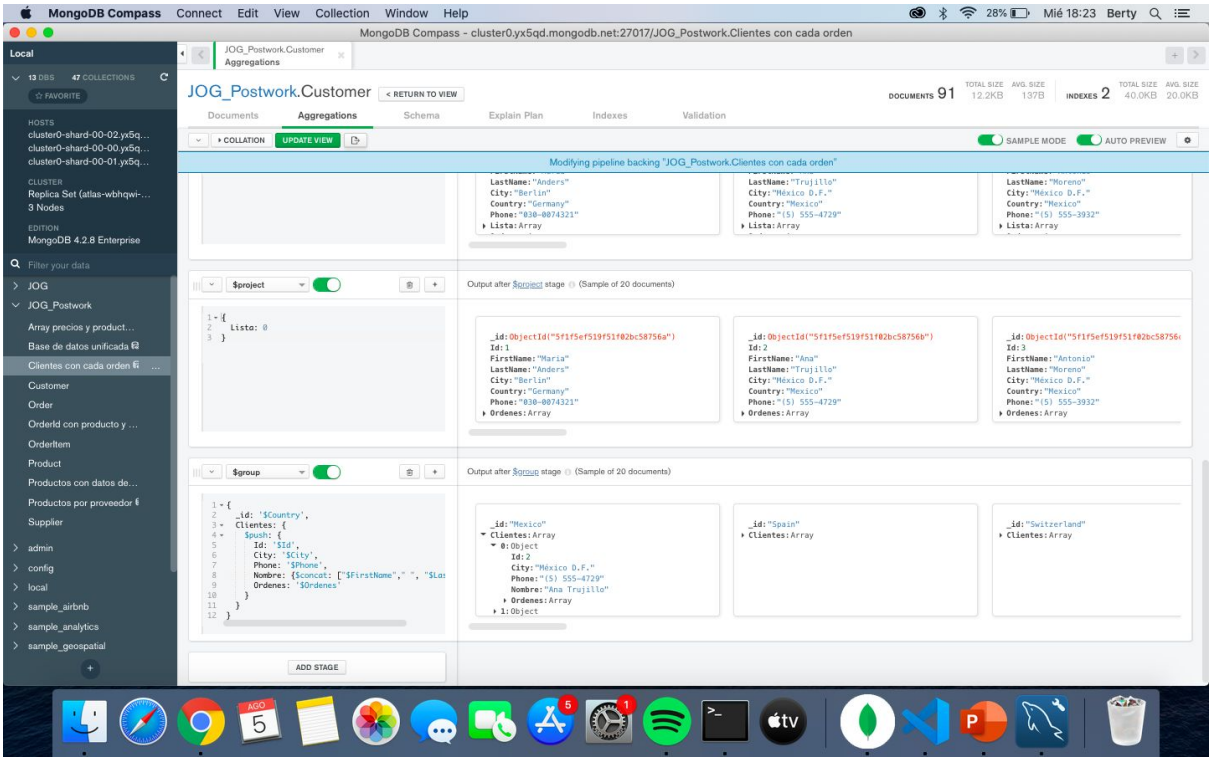
```
[{$lookup: {
  from: 'Base de datos unificada',
  localField: 'Id',
  foreignField: '_id',
  as: 'Lista'
}}, {$addFields: {
  Ordenes: {
    $arrayElemAt: [
      '$Lista',
      0
    ]
  }
}}, {$addFields: {
  Ordenes: '$Ordenes.Ordenes'
}}, {$project: {
  Lista: 0
}}, {$group: {
  _id: '$Country',
  Clientes: {
    $push: {
      Id: '$Id',
      City: '$City',
      Ordenes: '$Ordenes',
      Phone: '$Phone'
    }
  }
}}]
```

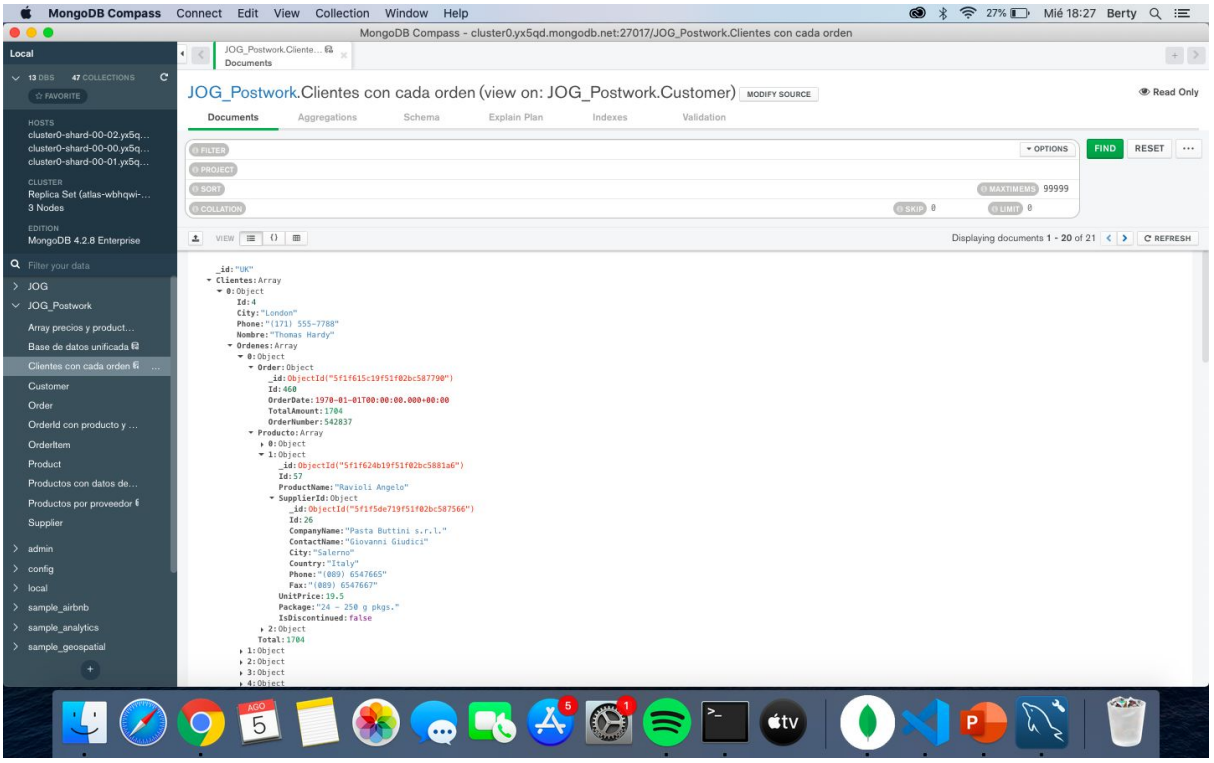

MongoDB Compass Connect Edit View Collection Window Help
MongoDB Compass - cluster0.yx5qd.mongodb.net:27017/JOG_Postwork.Clientes con cada orden
JOG_Postwork.Customer
Aggregations
< RETURN TO VIEW
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
COLLATION
UPDATE VIEW
SAMPLE MODE
AUTO PREVIEW
Modifying pipeline backing "JOG_Postwork.Clientes con cada orden"
$project
Output after $project stage (Sample of 20 documents)
$group
Output after $group stage (Sample of 20 documents)
ADD STAGE


MongoDB Compass Connect Edit View Collection Window Help
MongoDB Compass - cluster0.yx5qd.mongodb.net:27017/JOG_Postwork.Clientes con cada orden
JOG_Postwork.Clientes con cada orden (view on: JOG_Postwork.Customer)
MODIFY SOURCE
Read Only
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
FILTER
OPTIONS
FIND
RESET
Displaying documents 1 - 20 of 21
REFRESH

29.- Añade a esta misma tabla desde el codigo de agregación el nombre concatenado del cliente.

```
[{$lookup: {
  from: 'Base de datos unificada',
  localField: 'Id',
  foreignField: '_id',
  as: 'Lista'
}}, {$addFields: {
  Ordenes: {
    $arrayElemAt: [
      '$Lista',
      0
    ]
  }
}}, {$addFields: {
  Ordenes: '$Ordenes.Ordenes'
}}, {$project: {
  Lista: 0
}}, {$group: {
  _id: '$Country',
  Clientes: {
    $push: {
      Id: '$Id',
      City: '$City',
      Phone: '$Phone',
      Nombre: {$concat: ["$FirstName"," ", "$LastName"]},
      Ordenes: '$Ordenes'
    }
  }
}}]
```


MongoDB Compass Connect Edit View Collection Window Help
MongoDB Compass - cluster0.yx5qd.mongodb.net:27017/JOG_Postwork.Clientes con cada orden
JOG_Postwork.Customer
RETURN TO VIEW
Documents Aggregations Schema Explain Plan Indexes Validation
UPDATE VIEW
SAMPLE MODE
AUTO PREVIEW
Modifying pipeline backing "JOG_Postwork.Clientes con cada orden"
$project
$group
ADD STAGE


MongoDB Compass Connect Edit View Collection Window Help
MongoDB Compass - cluster0.yx5qd.mongodb.net:27017/JOG_Postwork.Clientes con cada orden
JOG_Postwork.Clientes con cada orden (view on: JOG_Postwork.Customer)
MODIFY SOURCE
Read Only
Documents Aggregations Schema Explain Plan Indexes Validation
FIND
RESET

30.- De la agregación actual, indica el número de clientes por país.

```
{
    $project: {
        _id: 1,
        Clientes: {
            $size: '$Clientes'
        }
    }
}
```