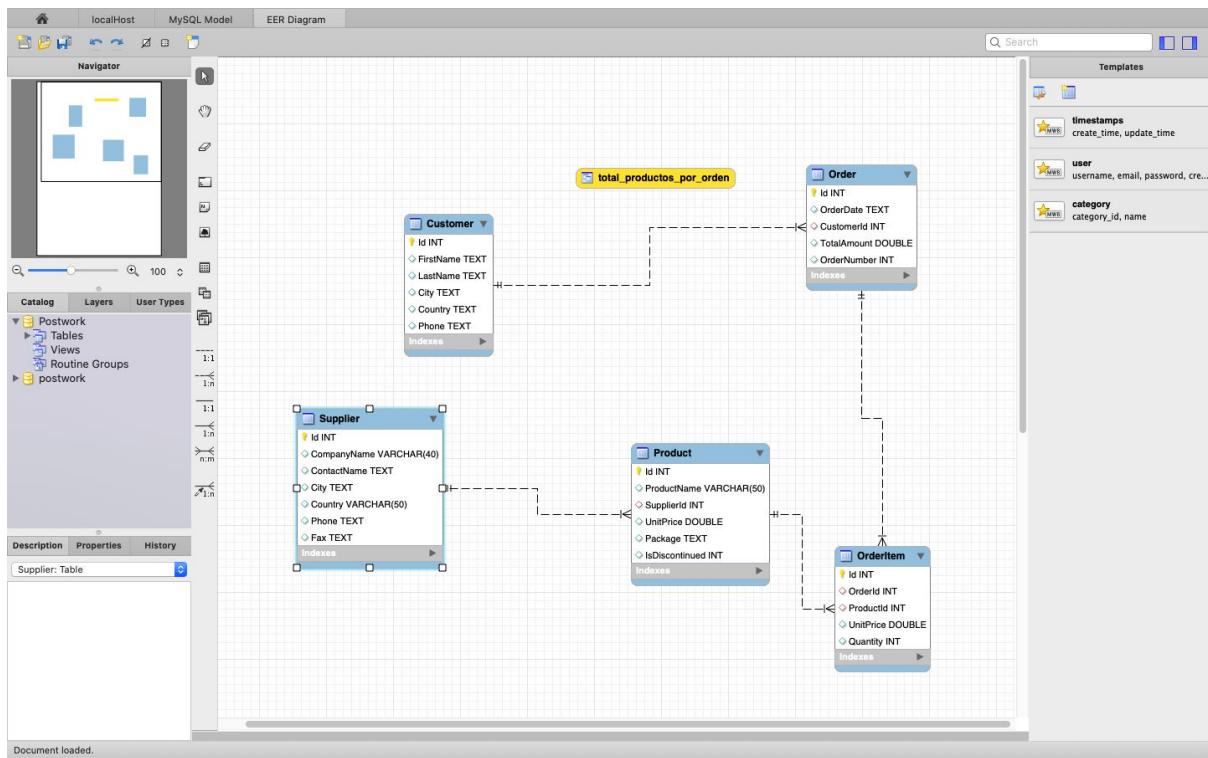


Post Work

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";

```

1. -- 1.- Seleccionar a todos los clientes que radiquen en Mexico
2. select * from Customer where Country = "Mexico";
3. -- 2.- Selecciona a todos los clientes que no radiquen en USA Y Argentina
4. select * from Customer where Country <> "USA" and Country <> "Argentina";
5. -- 3.- Seleccionar a todos los clientes que su apellido empiece con a o L
6. select * from Customer where LastName like 'a%' or LastName like 'l%';
7. -- 4.- ¿Cuál son todos los datos del producto más barato y más caro
8. select * from Product where UnitPrice = (select min(UnitPrice) from Product) or UnitPrice = (select max(UnitPrice) from Product);
9. -- 5.- ¿Cuántos productos surte cada proveedor?
10. select SupplierId, count(*) from Product group by SupplierId;
11. -- 6.- ¿Cuál es el precio promedio de los artículos suministrados por el proveedor?
12. select SupplierId, round(avg(UnitPrice), 2) as Total from Product group by SupplierId order by Total desc;
    
```

ID	First Name	Last Name	City	Country	Phone
2	Ana	Trujillo	México D.F. México	(5) 555-4729	
3	Antonio	Moreno	México D.F. México	(5) 555-3932	
13	Francisco	Chang	México D.F. México	(5) 555-3392	
58	Guillermo	Fernández	México D.F. México	(5) 552-3745	
80	Miguel	Angel Paolino	México D.F. México	(5) 555-2933	
NULL	NULL	NULL	NULL	NULL	NULL

Customer 1

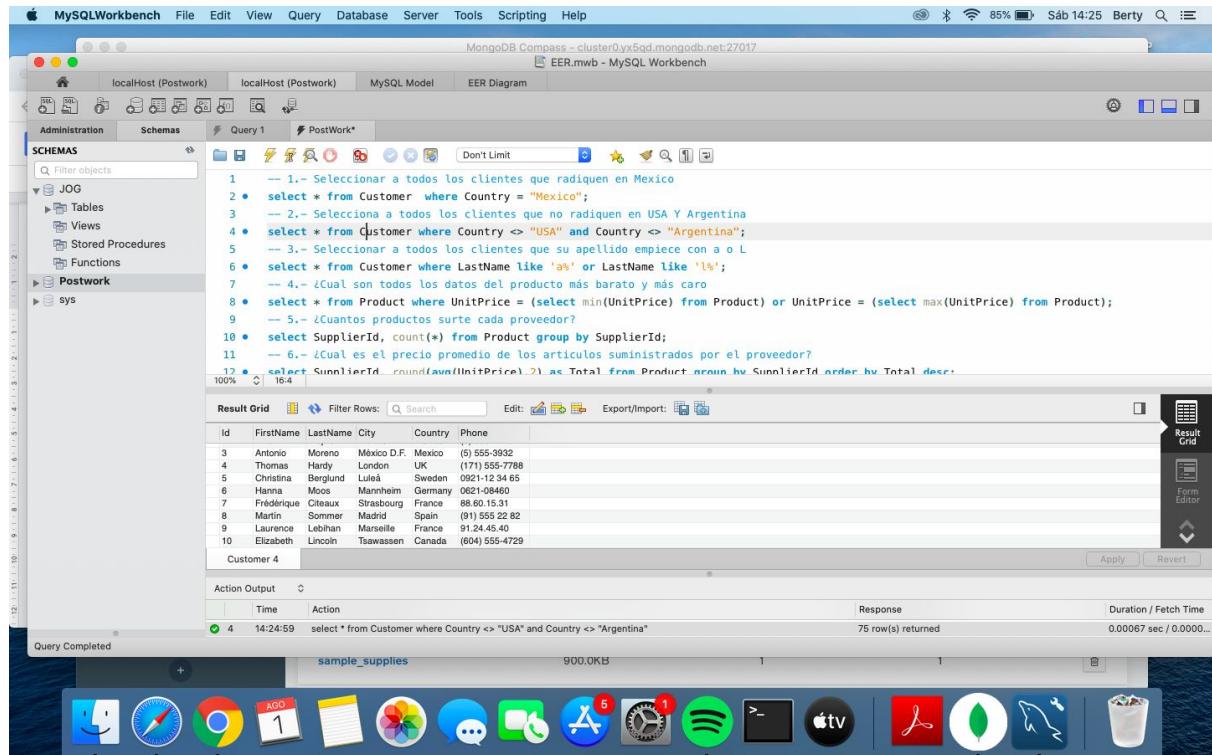
Action Output

Time	Action	Response	Duration / Fetch Time
14:21:20	select * from Customer where Country = "Mexico"	5 row(s) returned	0.018 sec / 0.000015...

Query Completed

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

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



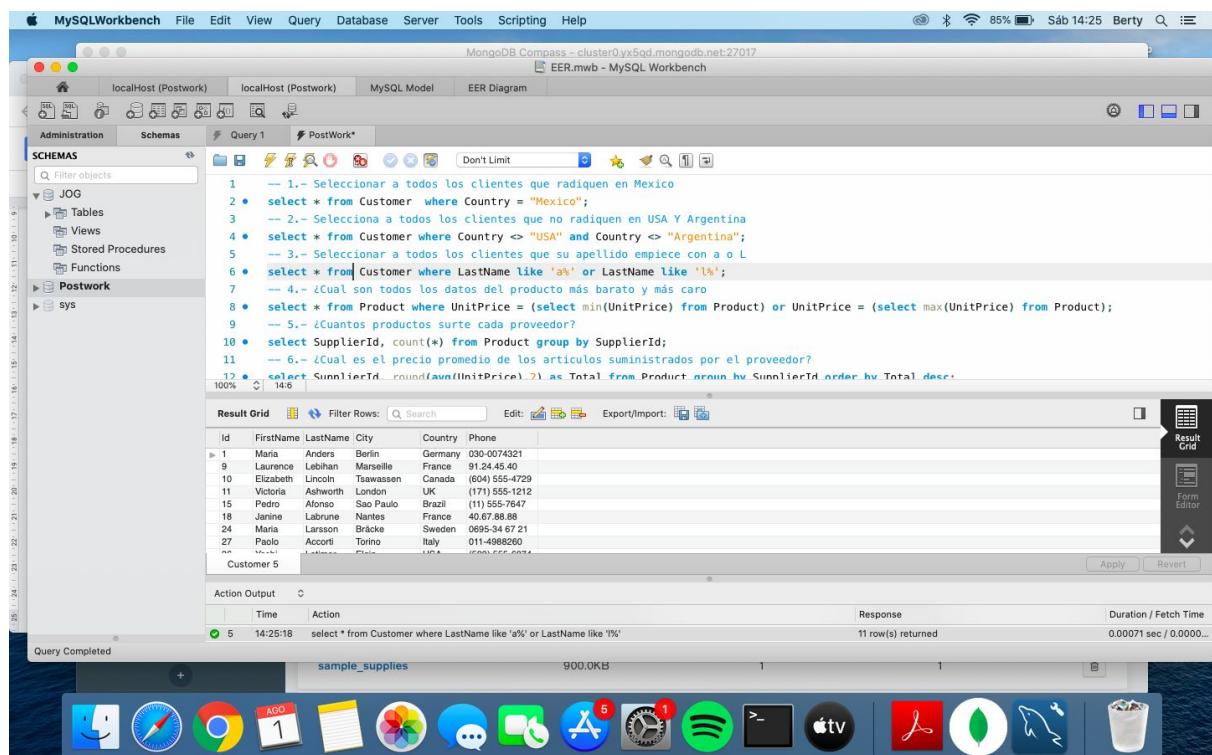
The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** JOG, Postwork (selected), sys
- Query Grid:** Shows the executed query and its results.
- Result Grid:** Displays the data for 75 rows of customers from the Customer table, filtered by Country not being USA or Argentina.
- Action Output:** Shows the execution time and action details.
- Query Completed:** Status message indicating 75 row(s) returned.
- Mac OS X Dock:** Shows various application icons including Finder, Safari, Mail, Calendar, Notes, Photos, iMessage, FaceTime, App Store, Spotify, and others.

ID	FirstName	LastName	City	Country	Phone
3	Antonio	Moreno	México D.F.	Mexico	(5) 555-9872
4	Thomas	Hardy	London	UK	(171) 555-7788
5	Günther	Rundtvedt	Lukas	Norway	(91) 555-8465
6	Hanna	Moe	Mannheim	Germany	0621-09460
7	Frédéric	Citeaux	Strasbourg	France	88.60.15.31
8	Martin	Sommer	Madrid	Spain	(91) 555-2282
9	Laurence	Lebihan	Marseille	France	91.24.45.40
10	Elizabeth	Lincoln	Tsawassen	Canada	(604) 555-4729

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%';
```



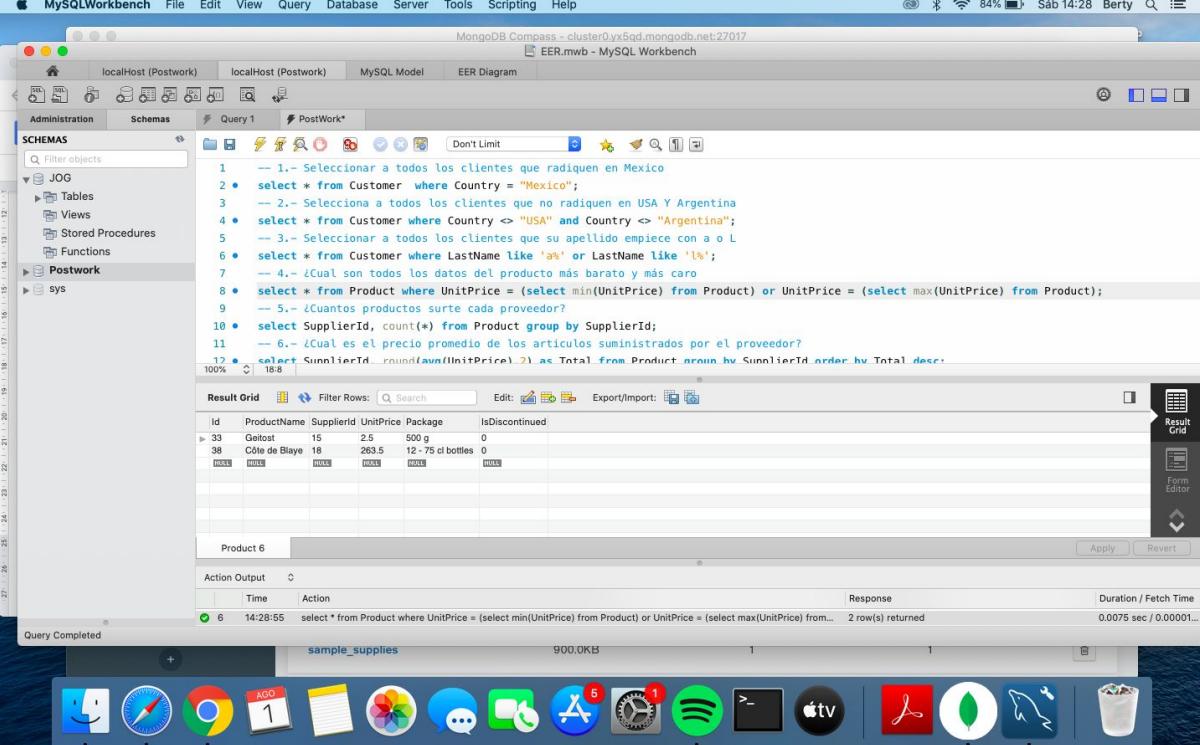
The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** JOG, Postwork (selected), sys
- Query Grid:** Shows the executed query and its results.
- Result Grid:** Displays the data for 11 rows of customers from the Customer table, filtered by LastName starting with 'a' or 'l'.
- Action Output:** Shows the execution time and action details.
- Query Completed:** Status message indicating 11 row(s) returned.
- Mac OS X Dock:** Shows various application icons including Finder, Safari, Mail, Calendar, Notes, Photos, iMessage, FaceTime, App Store, Spotify, and others.

ID	FirstName	LastName	City	Country	Phone
1	Maria	Anaya	Buenos Aires	Argentina	030-0774231
9	Laurence	Lebihan	Marseille	France	91.24.45.40
10	Elizabeth	Lincoln	Tsawassen	Canada	(604) 555-4729
11	Victoria	Ashworth	London	UK	(171) 555-1212
15	Pedro	Afonso	Sao Paulo	Brazil	(11) 555-7647
18	Janine	Larbrun	Nantes	France	40.67.88.88
24	Marie	Larsson	Bräcke	Sweden	0695-34 67 21
27	Paolo	Accorti	Torino	Italy	011-4988260

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);



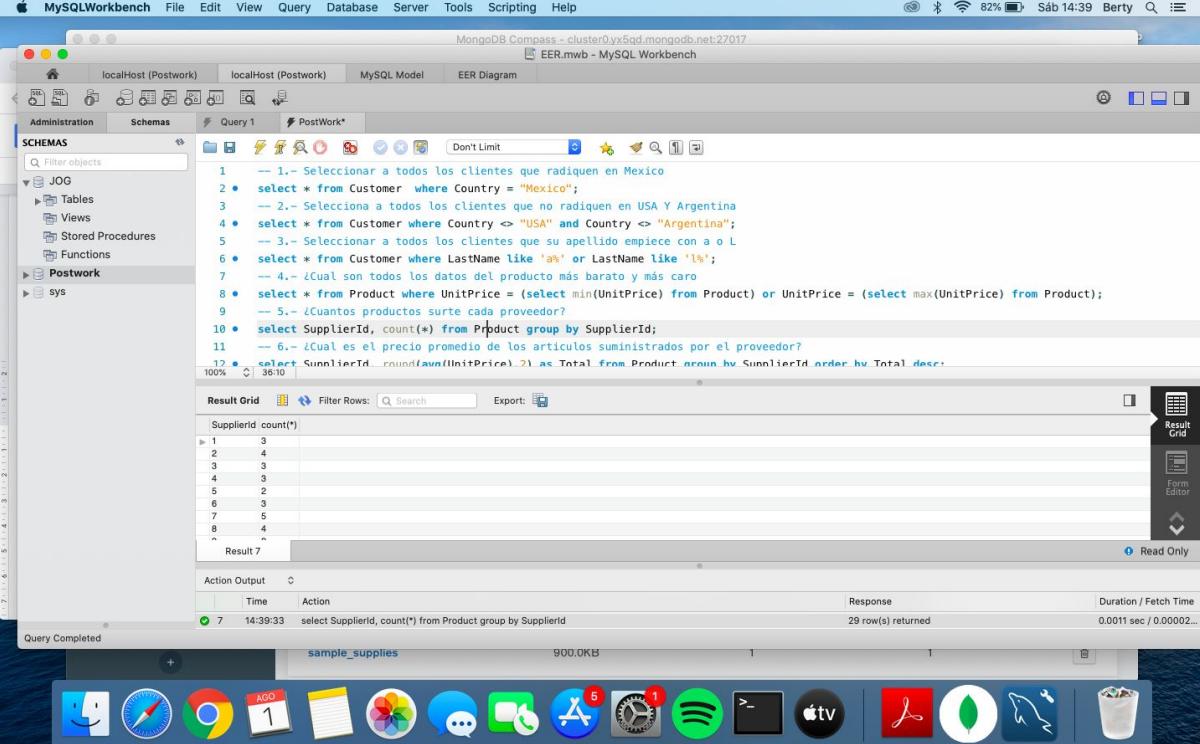
```

MySQLWorkbench  File Edit View Query Database Server Tools Scripting Help
MongoDB Compass - cluster0.yx5qd.mongodb.net:27017
EER.mwb - MySQL Workbench
Administration Schemas
Query 1 PostWork*
1 -- 1.- Seleccionar a todos los clientes que radiquen en Mexico
2 • select * from Customer where Country = "Mexico";
3 -- 2.- Selecciona a todos los clientes que no radiquen en USA Y Argentina
4 • select * from Customer where Country <> "USA" and Country <> "Argentina";
5 -- 3.- Seleccionar a todos los clientes que su apellido empiece con a o L
6 • select * from Customer where LastName like 'a%' or LastName like 'l%';
7 -- 4.- ¿Cuál son todos los datos del producto más barato y más caro
8 • select * from Product where UnitPrice = (select min(UnitPrice) from Product) or UnitPrice = (select max(UnitPrice) from Product);
9 -- 5.- ¿Cuántos productos surte cada proveedor?
10 • select SupplierId, count(*) from Product group by SupplierId;
11 -- 6.- ¿Cuál es el precio promedio de los artículos suministrados por el proveedor?
12 • select SupplierId, round(avg(UnitPrice), 2) as Total from Product group by SupplierId order by Total desc;
100% 18-6
Result Grid Filter Rows: Q Search Edit: Export/Import: Result Grid
Product Grid
Id ProductName SupplierId UnitPrice Package IsDiscontinued
33 Gefilte 15 2.5 500 g 0
38 Côte de Blaye 18 263.5 12 - 75 cl bottles 0
Result 6
Action Output
Time Action Response Duration / Fetch Time
0.0075 sec / 0.00001...
Query Completed
sample_supplies 900.0KB 1 1

```

5.- ¿Cuantos productos surte cada proveedor?

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



```

MySQLWorkbench  File Edit View Query Database Server Tools Scripting Help
MongoDB Compass - cluster0.yx5qd.mongodb.net:27017
EER.mwb - MySQL Workbench
Administration Schemas
Query 1 PostWork*
1 -- 1.- Seleccionar a todos los clientes que radiquen en Mexico
2 • select * from Customer where Country = "Mexico";
3 -- 2.- Selecciona a todos los clientes que no radiquen en USA Y Argentina
4 • select * from Customer where Country <> "USA" and Country <> "Argentina";
5 -- 3.- Seleccionar a todos los clientes que su apellido empiece con a o L
6 • select * from Customer where LastName like 'a%' or LastName like 'l%';
7 -- 4.- ¿Cuál son todos los datos del producto más barato y más caro
8 • select * from Product where UnitPrice = (select min(UnitPrice) from Product) or UnitPrice = (select max(UnitPrice) from Product);
9 -- 5.- ¿Cuántos productos surte cada proveedor?
10 • select SupplierId, count(*) from Product group by SupplierId;
11 -- 6.- ¿Cuál es el precio promedio de los artículos suministrados por el proveedor?
12 • select SupplierId, round(avg(UnitPrice), 2) as Total from Product group by SupplierId order by Total desc;
100% 36-10
Result Grid Filter Rows: Q Search Export: Result Grid
SupplierId count(*)
1 3
2 4
3 3
4 3
5 2
6 3
7 5
8 4
Result 7
Action Output
Time Action Response Duration / Fetch Time
0.0011 sec / 0.00002...
Query Completed
sample_supplies 900.0KB 1 1

```

-- 6.- ¿Cuál 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;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The query editor contains the following SQL code:

```
5 -- 3.- Seleccionar a todos los clientes que su apellido empieza con a o L
6 • select * from Customer where LastName like 'a%' or LastName like 'l%';
7 -- 4.- ¿Cuál son todos los datos del producto más barato y más caro
8 • select * from Product where UnitPrice = (select min(UnitPrice) from Product) or UnitPrice = (select max(UnitPrice) from Product);
9 -- 5.- ¿Cuántos productos surte cada proveedor?
10 • select SupplierId, count(*) from Product group by SupplierId;
11 -- 6.- ¿Cuál es el precio promedio de los artículos suministrados por el proveedor?
12 • select SupplierId, round(avg(UnitPrice),2) as Total from Product group by SupplierId order by Total desc;
13 -- 7.- ¿Cuáles son los 5 nombres de las compañías de los proveedores que más productos surten?
14 • select a.Id, CompanyName, count(*) total from Supplier a
15 left join Product b
16 on a.Id=b.SupplierID
```

The result grid shows the following data:

SupplierId	Total
12	44.68
28	44.50
29	38.90
7	35.57
3	31.67
24	30.93
11	29.71
5	29.50

Action Output shows the query was executed successfully.

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

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

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The query editor contains the same SQL code as the previous screenshot, but the results are different because the order by clause is missing in the original question. The result grid shows the following data:

Id	CompanyName	total
12	Plutzer Lebensmittelgroßmärkte AG	5
7	Pavlova, Ltd.	5
2	New Orleans Cajun Delights	4
8	Specialty Biscuits, Ltd.	4
1	Exotic Liquids	3

Action Output shows the query was executed successfully.

-- 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;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The query editor window displays two queries. The first query is the one above, and the second is:

```
— 9.- ¿Cuál es el país con más total de ingresos?  
27 • select Country, round(sum(b.TotalAmount),2) Total from Supplier a  
28 left join `Order` b  
29 on a.Id=b.CustomerId  
30 group by Country  
31 order by Total desc;
```

The results grid shows the following data:

Country	Total
Singapore	30
Sweden	23
USA	22
Canada	20
Japan	20
Sydney	19
France	18
Spain	18
...	...
Result	13

The status bar at the bottom indicates "Query Completed".

-- 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;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The query editor window displays the same two queries as the previous screenshot. The results grid shows the following data:

Country	Total
Singapore	113236.68

The status bar at the bottom indicates "Query Completed".

-- 10.- ¿Cuáles son las 3 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;
```

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
30 group by Country
31 order by Total desc
32 limit 1;
33 -- 10.- ¿Cuáles son las 3 ciudades con mas órdenes?
34 • select City, count(b.TotalAmount) Total from Supplier a
35 left join `Order` b
36 on a.Id=b.CustomerId
37 group by City
38 order by Total desc
39 limit 3;
40 -- 11.- ¿Cuál es el nombre de los 5 productos más vendidos?
41 • create view Total_productos_por_orden as (select a.Id, a.ProductName, sum(b.Quantity) Total from Product a
```

The result grid shows the top 3 cities with the most orders:

City	Total
Singapore	30
Wendy Mackenzie	19
Oviedo	18

The status bar at the bottom indicates "Query Completed" at 14:50:29.

-- 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);
```

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
37 group by City
38 order by Total desc
39 limit 3;
40 -- 11.- Crea una vista que muestra el total de unidades vendidas por producto
41 • create view Total_productos_por_orden as (select a.Id, a.ProductName, sum(b.Quantity) Total from Product a
42 left join OrderItem b
43 on a.Id=b.ProductId
44 group by a.Id
45 order by Total desc);
46 -- 11a.- ¿Cuál es el nombre de los 5 productos más vendidos?
47 • select * from Total_productos_por_orden limit 5;
48 -- 12.- ¿Cuál es el producto que no se ha vendido?
49 • select * from Total_productos_por_orden where Total is NULL;
50 -- 13.- ¿Cuál fue el total de la venta del producto mostrando el proveedor?
51 • select a.Id, a.ProductName, round(sum(b.UnitPrice),2) Total, c.CompanyName from Product a
52 left join OrderItem b
53 on a.Id=b.ProductId
```

The result grid shows the top 5 products with the highest sales volume:

Id	ProductName	Total
60	Commençant Period	1577
59	Raspberia Courdasmuff	1496
31	Gorgonzola Telino	1387
86	Gnocchi di nonna Alice	1263
16	Pavlova	1158

The status bar at the bottom indicates "Query Completed" at 14:59:27.

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

```
select * from Total_productos_por_orden limit 5;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the schema structure under 'Postwork'. The main area shows a query editor with the following SQL code:

```
37 group by City
38 order by Total desc
39 limit 3;
-- 11.- Crea una vista que muestra el total de unidades vendidas por producto
41 • create view Total_productos_por_orden as (select a.Id, a.ProductName, sum(b.Quantity) Total from Product a
42 left join OrderItem b
43 on a.Id=b.ProductId
44 group by a.Id
45 order by Total desc);
-- 11a.- ¿Cual es el nombre de los 5 productos más vendidos?
47 • select * from Total_productos_por_orden limit 5;
-- 12.- ¿Cual es el producto que no se ha vendido?
49 • select * from Total_productos_por_orden where Total is NULL;
-- 13.- ¿Cuál fue el total de la venta del producto mostrando el proveedor?
51 • select a.Id, a.ProductName, round(sum(b.UnitPrice),2) Total, c.CompanyName from Product a
52 left join OrderItem b
53 on a.Id=b.ProductId
```

The result grid shows the following data:

ID	ProductName	Total
60	Camembert Pierrot	1577
59	Raclette Courteau	1496
31	Gorgonzola Telino	1387
58	Gnocchi di nonna Alice	1263
16	Pavlova	1158

At the bottom, the status bar indicates "Query Completed".

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

```
select * from Total_productos_por_orden where Total is NULL;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the schema structure under 'Postwork'. The main area shows a query editor with the following SQL code:

```
37 group by City
38 order by Total desc
39 limit 3;
-- 11.- Crea una vista que muestra el total de unidades vendidas por producto
41 • create view Total_productos_por_orden as (select a.Id, a.ProductName, sum(b.Quantity) Total from Product a
42 left join OrderItem b
43 on a.Id=b.ProductId
44 group by a.Id
45 order by Total desc);
-- 11a.- ¿Cual es el nombre de los 5 productos más vendidos?
47 • select * from Total_productos_por_orden limit 5;
-- 12.- ¿Cual es el producto que no se ha vendido?
49 • select * from Total_productos_por_orden where Total is NULL;
-- 13.- ¿Cuál fue el total de la venta del producto mostrando el proveedor?
51 • select a.Id, a.ProductName, round(sum(b.UnitPrice),2) Total, c.CompanyName from Product a
52 left join OrderItem b
53 on a.Id=b.ProductId
```

The result grid shows the following data:

ID	ProductName	Total
78	Stroopwafels	NULL

At the bottom, the status bar indicates "Query Completed".

-- 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;
```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** JOG, Postwork, sys.
- Query Editor:** Contains the SQL code for question 13.
- Result Grid:** Displays the results of the query, showing columns: Id, ProductName, Total, CompanyName. One row is highlighted: Manjimup Dried Apples (Id: 51) with a Total of 1971.60.
- Action Output:** Shows the execution time (26 15:04:07) and number of rows returned (78).
- Timeline:** Shows the duration and fetch time of the query (0.0041 sec / 0.00002...).

-- 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)
```

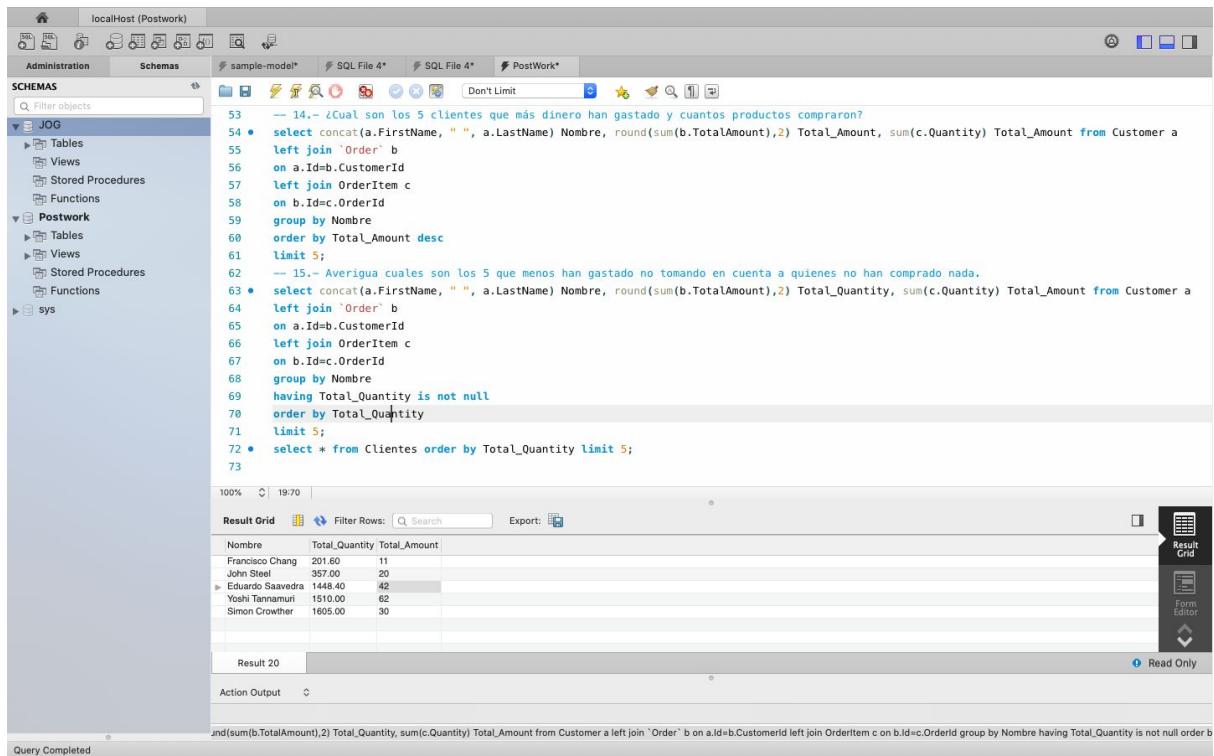
```
Total_Quantity, sum(c.Quantity) Total_Amount from Customer a left join `Order` b on
a.Id=b.CustomerId left join OrderItem c on b.Id=c.OrderId group by Nombre order by
Total_Quantity desc limit 5;
```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** JOG, Postwork, sys.
- Query Editor:** Contains the SQL code for question 14.
- Result Grid:** Displays the results of the query, showing columns: Nombre, Total_Amount, Total_Amount. Five rows are listed, with the first row highlighted: Jose Mendez (Nombre: Jose Mendez) with a Total_Amount of 481075.60.
- Action Output:** Shows the execution time (50 15:30:39) and number of rows returned (5).
- Timeline:** Shows the duration and fetch time of the query (0.013 sec / 0.000017...).

-- 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;
```



The screenshot shows a SQL editor interface with the following details:

- Title Bar:** localhost (Postwork)
- Toolbar:** Includes icons for Home, Database, Schemas, Tables, Views, Procedures, Functions, and a search bar.
- Menu Bar:** Administration, Schemas, sample-model*, SQL File 4*, SQL File 4*, PostWork*.
- Schemas:** JOG (selected), Postwork, sys.
- Code Area:** Contains two SQL queries. The second query is highlighted.

```
53 -- 14.- ¿Cuáles son los 5 clientes que más dinero han gastado y cuantos productos compraron?
54 • select concat(a.FirstName, " ", a.LastName) Nombre, round(sum(b.TotalAmount),2) Total_Amount, sum(c.Quantity) Total_Amount from Customer a
55     left join `Order` b
56     on a.Id=b.CustomerId
57     left join OrderItem c
58     on b.Id=c.OrderId
59     group by Nombre
60     order by Total_Amount desc
61     limit 5;
62 -- 15.- Averigua cuales son los 5 que menos han gastado no tomando en cuenta a quienes no han comprado nada.
63 • select concat(a.FirstName, " ", a.LastName) Nombre, round(sum(b.TotalAmount),2) Total_Quantity, sum(c.Quantity) Total_Amount from Customer a
64     left join `Order` b
65     on a.Id=b.CustomerId
66     left join OrderItem c
67     on b.Id=c.OrderId
68     group by Nombre
69     having Total_Quantity is not null
70     order by Total_Quantity
71     limit 5;
72 • select * from Clientes order by Total_Quantity limit 5;
73
```
- Result Grid:** Shows the results of the second query:

Nombre	Total_Quantity	Total_Amount
Francisco Chang	20.00	11
John Smith	287.00	20
Edwards Saavedra	1448.40	62
Yoshi Tannamuri	1510.00	62
Simon Crowther	1605.00	30
- Status Bar:** Query Completed

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

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

The screenshot shows the MongoDB Compass interface with the following details:

- Projected Fields:** The query includes `CompanyName: 1`, `Phone: 1`, and `Fax: 1`.
- Results:** 29 documents are found, each containing the projected fields.
- Sample Document 1:**

```
_id: ObjectId("5f1f5de719f51f02bc58754d")
CompanyName: "Exotic Liquids"
Phone: "(171) 555-2222"
Fax: "NULL"
```
- Sample Document 2:**

```
_id: ObjectId("5f1f5de719f51f02bc58754e")
CompanyName: "New Orleans Cajun Delights"
Phone: "(180) 555-4822"
Fax: "NULL"
```
- Sample Document 3:**

```
_id: ObjectId("5f1f5de719f51f02bc58754f")
CompanyName: "Grandma Kelly's Homestead"
Phone: "(313) 555-5735"
Fax: "(313) 555-3349"
```
- Sample Document 4:**

```
_id: ObjectId("5f1f5de719f51f02bc587550")
CompanyName: "Tokyo Traders"
Phone: "(03) 3555-5011"
Fax: "NULL."
```

17.- Filtra a los proveedores del reino unido

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

The screenshot shows the MongoDB Compass interface on a Mac OS X desktop. The title bar reads "MongoDB Compass - cluster0.yx5qd.mongodb.net:27017". The left sidebar shows the database structure: "Local" with "13 DBs" and "38 COLLECTIONS", and a "JOG_Postwork" database containing collections like "Supplier", "admin", "config", "local", etc. The main pane displays the "JOG_Postwork.Supplier" collection with 29 documents. A query builder is open with the following filter: {Country: 'UK'}. The results show two documents:

```
_id:ObjectId("5f1f5de719f51f02bc58754d")
Id:1
CompanyName:"Exotic Liquids"
ContactName:"Charlotte Cooper"
City:"London"
Country:"UK"
Phone:"(171) 555-2222"
Fax:"NULL"

_id:ObjectId("5f1f5de719f51f02bc587554")
Id:8
CompanyName:"Specialty Biscuits Ltd."
ContactName:"Peter Wilson"
City:"Manchester"
Country:"UK"
Phone:"(161) 555-4448"
Fax:"NULL"
```

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

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

The screenshot shows the MongoDB Compass interface with the following details:

- Left Sidebar:** Shows the database structure with 'Local' selected, 13 DBs, and 38 collections.
- Top Bar:** Title 'JOG_Postwork.Customer Documents', tabs: 'Documents', 'Aggregations', 'Schema', 'Explain Plan', 'Indexes', 'Validation'.
- Search Bar:** Contains the query: '\$or: [{Country: "Germany"}, {Country: "Mexico"}]'.
- Results Table:** Displays 91 documents. The first few results are:
 - Id: 1, FirstName: "Maria", LastName: "Anders", City: "Berlin", Country: "Germany", Phone: "+030-0074321"
 - Id: 2, FirstName: "Ana", LastName: "Trujillo", City: "Mexico D.F.", Country: "Mexico", Phone: "(5) 555-4729"
 - Id: 3, FirstName: "Antonio", LastName: "Moreno", City: "Mexico D.F.", Country: "Mexico", Phone: "(5) 555-3932"
 - Id: 6, FirstName: "Hanna", LastName: "Mann", City: "Mannheim", Country: "Germany", Phone: "+0621-08460"
- Right Panel:** Shows three log entries for filters used in previous queries:
 - Mon Jul 27 2020 18:17:18 GMT-0500 (CET): '\$or: [{Country: 'Germany'}, {Country: 'Mexico'}]' (with JSON code)
 - Mon Jul 27 2020 18:14:29 GMT-0500 (CET): '\$or: [{Country: 'Germany'}, {Country: 'Mexico'}]' (with JSON code)
 - Mon Jul 27 2020 18:14:22 GMT-0500 (CET): '\$and: [{Country: 'Germany'}, {Country: 'Mexico'}]' (with JSON code)
 - Mon Jul 27 2020 18:14:14 GMT-0500 (CET): (empty)

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

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

The screenshot shows the MongoDB Compass interface with the following details:

- Left Sidebar:** Shows the database structure with 13 DBs and 38 collections. The current collection is "JOG_Postwork.Order".
- Top Bar:** Displays the collection name "JOG_Postwork.Order", document count "DOCUMENTS 2.2k", total size "191.5KB", and index count "INDEXES 1".
- Query Filter Panel:** Shows the query: `UnitPrice: {$gt: 30}`. It includes options for FILTER, PROJECT, SORT, and COLLATION. The SORT option has "MAXITEMS 5000" selected.
- Result Preview:** Displays five documents from the results. Each document includes fields: _id, Id, OrderId, ProductId, UnitPrice, and Quantity. The first document's UnitPrice is 34.8, and the last one is 64.8.
- Right Panel:** Shows the query history with the timestamp "Mon Jul 27 2020 18:22:45 GMT-0500 (C...)" and the same query: `UnitPrice: {$gt: 30}`.

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

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

The screenshot shows the MongoDB Compass interface on a Mac OS X desktop. The main window displays the 'JOG_Postwork.Product' collection with 78 documents. A search query is applied, sorting by UnitPrice in descending order (-1) and limiting the result to 1 document. The result is a single document for 'Côte de Blaye' with a UnitPrice of 263.5.

_id	ProductName	SupplierId	UnitPrice	Package	IsDiscontinued
ObjectId("5f1f624b19f51f02bc588193")	Côte de Blaye	18	263.5	12 - 75 cl bottles	false

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

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

The screenshot shows the MongoDB Compass interface with the following details:

- Left Sidebar:** Shows the "Local" database with 13 DBs and 38 collections. The "Product" collection is selected.
- Top Bar:** Shows the title "MongoDB Compass - cluster0.yx5qd.mongodb.net:27017/JOG_Postwork.Product".
- Main Area:** The "Aggregations" tab is active. It displays an aggregation pipeline with one stage: `[$match: { IsDiscontinued: false }]`. The output of this stage is shown in a preview pane, displaying two documents from the "Product" collection.
- Preview Pane:** Shows 78 documents in the collection. Two specific documents are expanded:
 - Document 1: _id: ObjectId("5f1f624b19f51f02bc58816e"), Id: 1, ProductName: "Chai", SupplierId: 1, UnitPrice: 18, Package: "10 boxes x 20 bags", IsDiscontinued: false
 - Document 2: _id: ObjectId("5f1f624b19f51f02bc58816f"), Id: 2, ProductName: "Chang", SupplierId: 1, UnitPrice: 19, Package: "24 - 12 oz bottle", IsDiscontinued: false
- Bottom Bar:** Shows various Mac OS X application icons.

22.- ¿Cuantos productos surte cada proveedor?

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

The screenshot shows the MongoDB Compass interface with the following details:

- Hosts:** cluster0-shard-00-02.y5q... (selected), cluster0-shard-00-00.y5q..., cluster0-shard-00-01.y5q...
- Cluster:** Replica Set (atlas-wbhqw-...)
- Edition:** MongoDB 4.2.8 Enterprise
- Collection:** JOG_Postwork.Product
- Aggregations Tab:** The pipeline consists of two stages:
 - \$match:** Filters documents where IsDiscontinued is false.
 - \$group:** Groups by SupplierId and calculates the total count of products.
- Output after \$match stage:** Shows sample documents for SupplierId 1 and 2.
- Output after \$group stage:** Shows two groups: one for SupplierId 17 with a total of 3, and another for SupplierId 3 with a total of 3.
- Document Count:** 78
- Total Size:** 11.1KB
- Avg. Size:** 145B
- Indexes:** 1
- Total Size:** 20.0KB
- Avg. Size:** 20.0KB

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
}}]
```

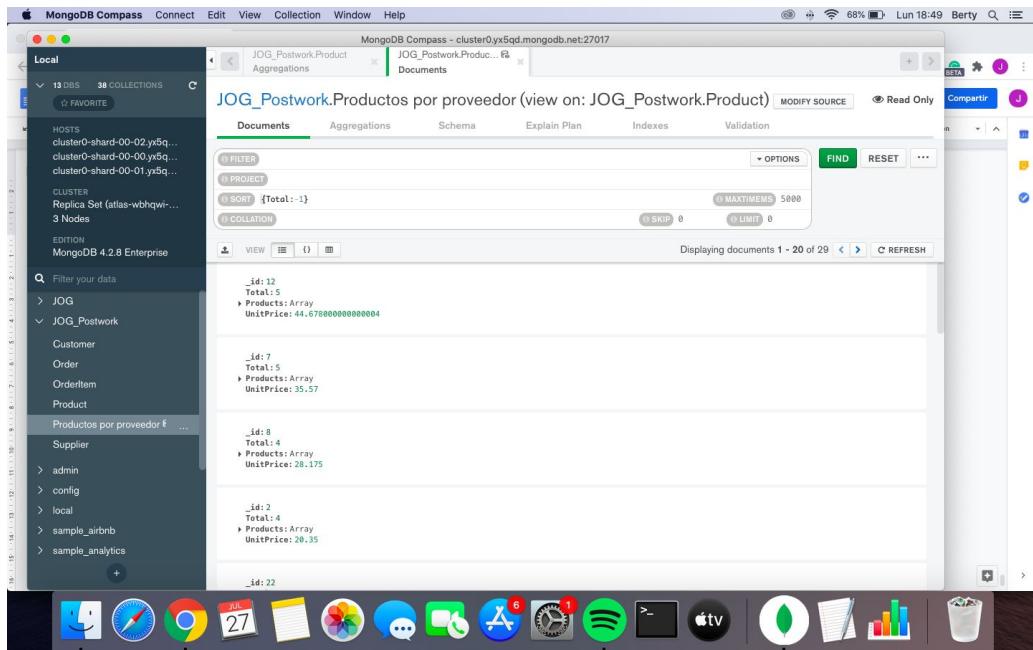
The screenshot shows the MongoDB Compass interface with the following details:

- Hosts:** cluster0-shard-00-02.yx5q... (selected), cluster0-shard-00-00.yx5q..., cluster0-shard-00-01.yx5q...
- Cluster:** Replica Set (atlas-wbhqw-... 3 Nodes)
- Edition:** MongoDB 4.2.8 Enterprise
- Collection:** JOG_Postwork.Product
- Aggregations:** A pipeline is being built with two stages:
 - \$group:** Groups documents by Supplier ID. The output shows two groups: one for Supplier ID 18 (Total: 2) and one for Supplier ID 20 (Total: 2). Each group contains an array of products and a average price.
 - \$sort:** Sorts the groups by Total in descending order. The output shows two sorted groups: one for Supplier ID 7 (Total: 4) and one for Supplier ID 8 (Total: 4).
- Output after \$group stage:**

_id	Total	Productos	Precioneto
18	2	["Product Name 1", "Product Name 2"]	140.75
20	2	["Product Name 1", "Product Name 2"]	32.725
- Output after \$sort stage:**

_id	Total	Productos	Precioneto
7	4	["Product Name 1", "Product Name 2", "Product Name 3", "Product Name 4"]	34.7125
8	4	["Product Name 1", "Product Name 2", "Product Name 3", "Product Name 4"]	28.175

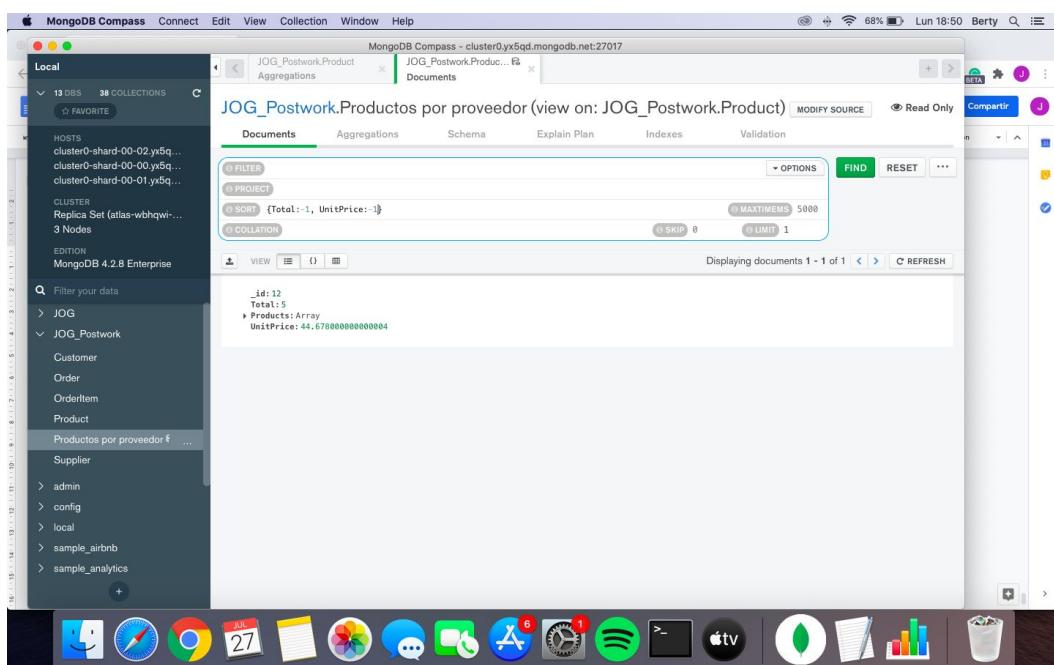
24.- Crea una vista de esta tabla y arroja al proveedor con más productos y el precio promedio más alto



The screenshot shows the MongoDB Compass application running on a Mac OS X desktop. The main window title is "JOG_Postwork.Product" and the sub-title is "Documents". The interface includes a top navigation bar with "MongoDB Compass", "Connect", "Edit", "View", "Collection", "Window", and "Help". Below the navigation is a toolbar with various icons. The left sidebar lists databases and collections, including "Local", "HOSTS", "CLUSTER", and "JOG_Postwork" which contains "Customer", "Order", "OrderItem", and "Product". The "Product" collection is currently selected, showing a document titled "Productos por proveedor". The main content area displays a list of documents with the following fields:

- `_id: 12`
`Total: 5`
`Products: Array`
`UnitPrice: 44.678000000000004`
- `_id: 7`
`Total: 5`
`Products: Array`
`UnitPrice: 35.57`
- `_id: 8`
`Total: 4`
`Products: Array`
`UnitPrice: 28.175`
- `_id: 3`
`Total: 4`
`Products: Array`
`UnitPrice: 20.35`
- `_id: 22`

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



This screenshot shows the same MongoDB Compass interface as the previous one, but with a different aggregation query applied. The "Sort" field in the aggregation pipeline is now set to `{Total:-1, UnitPrice:-1}`. The result shows only one document, which corresponds to the document with `_id: 12` from the previous screenshot.

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'
}]
```

Modifying pipeline backing "JOG_Postwork.Productos con datos del proveedor"

\$lookup

```
1. {  
2.   from: 'Supplier',  
3.   localField: 'SupplierId',  
4.   foreignField: 'Id',  
5.   as: 'SupplierId'  
6. }
```

\$unwind

```
1. {  
2.   path: '$SupplierId'  
3. }
```

_id	ProductName	Supplier
5f1f624b19f51f02bc58816e	Chai	{ "_id": 1, "SupplierId": "5f1f624b19f51f02bc58816e", "UnitPrice": 18, "Package": "10 boxes x 20 bags", "IsDiscontinued": false }
5f1f624b19f51f02bc588170	Aniseed Syrup	{ "_id": 2, "SupplierId": "5f1f624b19f51f02bc588170", "UnitPrice": 19, "Package": "24 - 12 oz bottles", "IsDiscontinued": false }
5f1f624b19f51f02bc588171	Chai Anise Syrup	{ "_id": 3, "SupplierId": "5f1f624b19f51f02bc588171", "UnitPrice": 19, "Package": "550 ml bottles", "IsDiscontinued": false }

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'
  }
}}]
```

The screenshot shows the MongoDB Compass interface with the following details:

- Left Sidebar:** Shows the database structure with "Local" selected, displaying 13 DBs and 45 collections. A "FAVORITE" section lists "JOG_Postwork" and "JOG_Postwork.Order".
- Top Bar:** Shows the connection information: "MongoDB Compass - cluster0.yx5qd.mongodb.net:27017/JOG_Postwork.Order".
- Central Area:**
 - Aggregations Tab:** Selected tab. It displays the aggregation pipeline being modified:


```
1: { $unwind: { path: '$ProductId' } }
2: { $group: {
3:   _id: '$OrderId',
4:   ProductId: { $push: '$ProductId' },
5:   Total: { $sum: '$Total_Amount' }
6: } }
```
 - Output Area:** Shows the results of the aggregation pipeline. Two sample documents are shown:


```
_id: ObjectId("5f1f61a219f51f02bc587903")
Id: 1
OrderId: 1
> ProductId: Object
  UnitPrice: 14
  Quantity: 12
  Total_Amount: 168
```



```
_id: ObjectId("5f1f61a219f51f02bc587903")
Id: 2
OrderId: 1
> ProductId: Object
  UnitPrice: 9.8
  Quantity: 10
  Total_Amount: 98
```
- Bottom:** macOS Dock with various application icons.

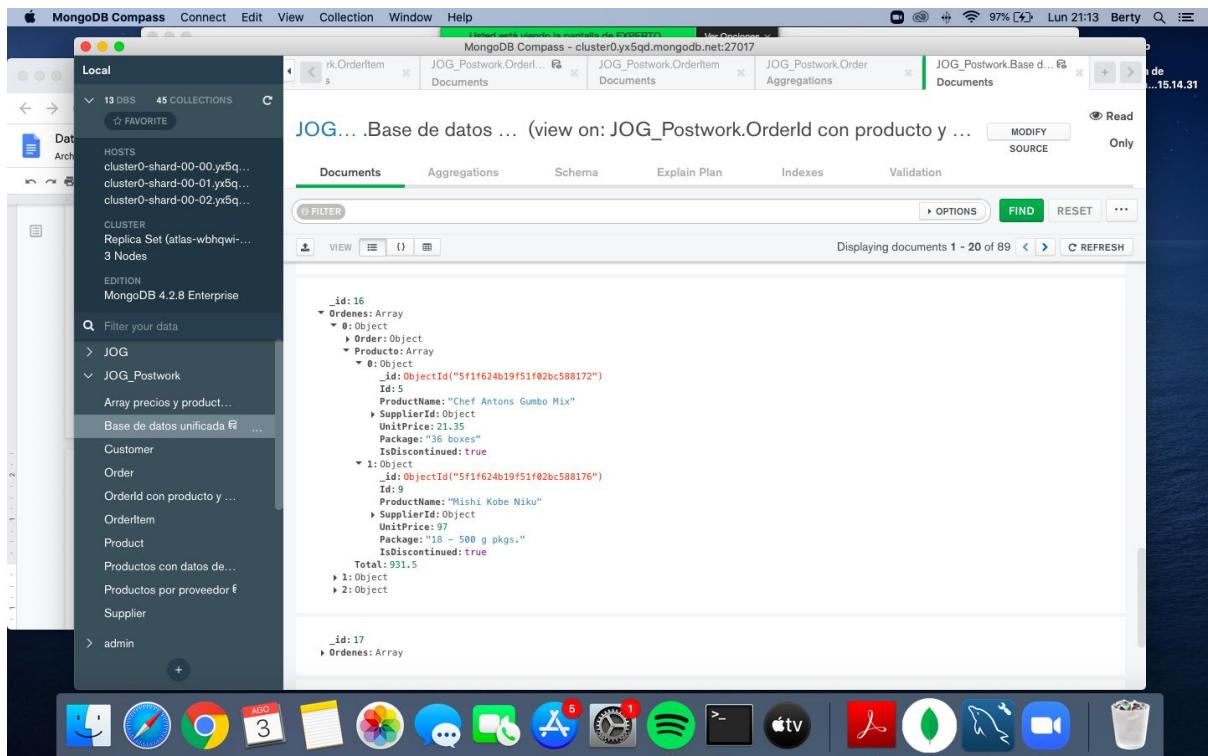
27.- Desde la tabla OrderItem, agrega los documentos de la vista anterior para mostrar por cada cliente las órdenes que ha pedido y qué productos incluye. Cada producto debe tener la información del proveedor.

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

The screenshot shows the MongoDB Compass interface with the following details:

- Aggregation Pipeline:**

```
1. { $group: {
  _id: '$CustomerId',
  Ordenes: {
    $push: {
      Order: '$Id',
      Producto: '$ProductId',
      Total: '$Total'
    }
  }
}}
```
- Results:**
 - Sample document 1: `_id: 210`, `Ordenes: [{ Order: { _id: "5f1f615c19f51f02bc5875d0" }, Producto: { _id: "5f1f615c19f51f02bc5875d0" }, Total: 1584 }]`
 - Sample document 2: `_id: 291`, `Ordenes: [{ Order: { _id: "5f1f615c19f51f02bc5875d0" }, Producto: { _id: "5f1f615c19f51f02bc5875d0" }, Total: 139.8 }]`
 - Sample document 3: `_id: 291`, `Ordenes: [{ Order: { _id: "5f1f615c19f51f02bc5875d0" }, Producto: { _id: "5f1f615c19f51f02bc5875d0" }, Total: 139.8 }]`



28.- Usando la vista anterior, añade a la tabla customer un array con toda la información de cada orden.

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

Local

13 DBS 45 COLLECTIONS

HOSTS
cluster0-shard-00-00.yx5q...
cluster0-shard-00-01.yx5q...
cluster0-shard-00-02.yx5q...

CLUSTER
Replica Set (atlas-wbhqwi...
3 Nodes

EDITION
MongoDB 4.2.8 Enterprise

Filter your data

- > JOG
- > JOG_Postwork
 - Aray precios y product...
 - Base de datos unificada
 - Customer
 - Datos de clientes con o...
 - ...
 - Order
 - OrderId con producto y ...
 - OrderItem
 - Product
 - Productos con datos de...
 - Productos por proveedor
 - Supplier
- > admin
- > config
- > local
- > sample_airbnb
- > sample_analytics

JOG_Postwork.Customer

RETURN TO VIEW DOCUMENTS 91 TOTAL SIZE 12.2KB AVG. SIZE 137B INDEXES 1 TOTAL SIZE 20.0KB AVG. SIZE 20.0KB

Aggregations

Modifying pipeline backing 'JOG_Postwork.Datos de clientes con ordenes'

Output after \$addFields stage (Sample of 20 documents)

```
1 + {
2   Ordenes: '$Lista.Ordenes'
3 }
```

```
_id: ObjectId("5f1f5ef519f51f02bc58756a")
Id: 1
FirstName: "Maria"
LastName: "Anders"
City: "Berglstr."
Country: "Germany"
Phone: "030-0874321"
> Lista: Array
> Ordenes: Object
```

```
_id: ObjectId("5f1f5ef519f51f02bc58756b")
Id: 2
FirstName: "Ana"
LastName: "Trujillo"
City: "Ave. 9123, 12000 D.F."
Country: "Mexico"
Phone: "(5) 555-4729"
> Lista: Array
> Ordenes: Object
```

Output after \$project stage (Sample of 20 documents)

```
1 + {
2   Lista: 0
3 }
```

```
_id: ObjectId("5f1f5ef519f51f02bc58756a")
Id: 1
FirstName: "Maria"
LastName: "Anders"
City: "Berglstr."
Country: "Germany"
Phone: "030-0874321"
> Ordenes: Array
```

```
_id: ObjectId("5f1f5ef519f51f02bc58756b")
Id: 2
FirstName: "Ana"
LastName: "Trujillo"
City: "Ave. 9123, 12000 D.F."
Country: "Mexico"
Phone: "(5) 555-4729"
> Ordenes: Array
```

ADD STAGE