

Introducción a bases de datos 2020

José Alberto Oscoy

Proyecto DOFactory



Objetivos

01

Crear una base de datos y analizar la estructura de distintas tablas.

02

Realizar agrupamientos en los resultados de una consulta o tabla y escribir consultas que relacionen dos o más tablas.

03

Analizar las agregaciones de agrupamiento y generar vistas que almacenen el resultado de una agregación y consulta.



• • •

¿Cuál es la necesidad o problema a resolver? □

La digitalización de los procesos de administración, incluso antes de la situación actual ya era esencial para competir en el mercado actual donde el acceso a la información de manera ordenada es crucial para la toma de decisiones.



Una organización necesita ordenar los datos de sus compras a proveedores, ordenes a clientes y catalogo de productos ya que al tener toda su información separada y en hojas de calculo, tareas como generación reportes de resultados toman mucho tiempo además de que la información no es 100% confiable al ser un proceso principalmente manual.

• • •

¿Cuál es la necesidad o problema a resolver? □

La digitalización de los procesos de administración, incluso antes de la situación actual ya era esencial para competir en el mercado actual donde el acceso a la información de manera ordenada es crucial para la toma de decisiones.



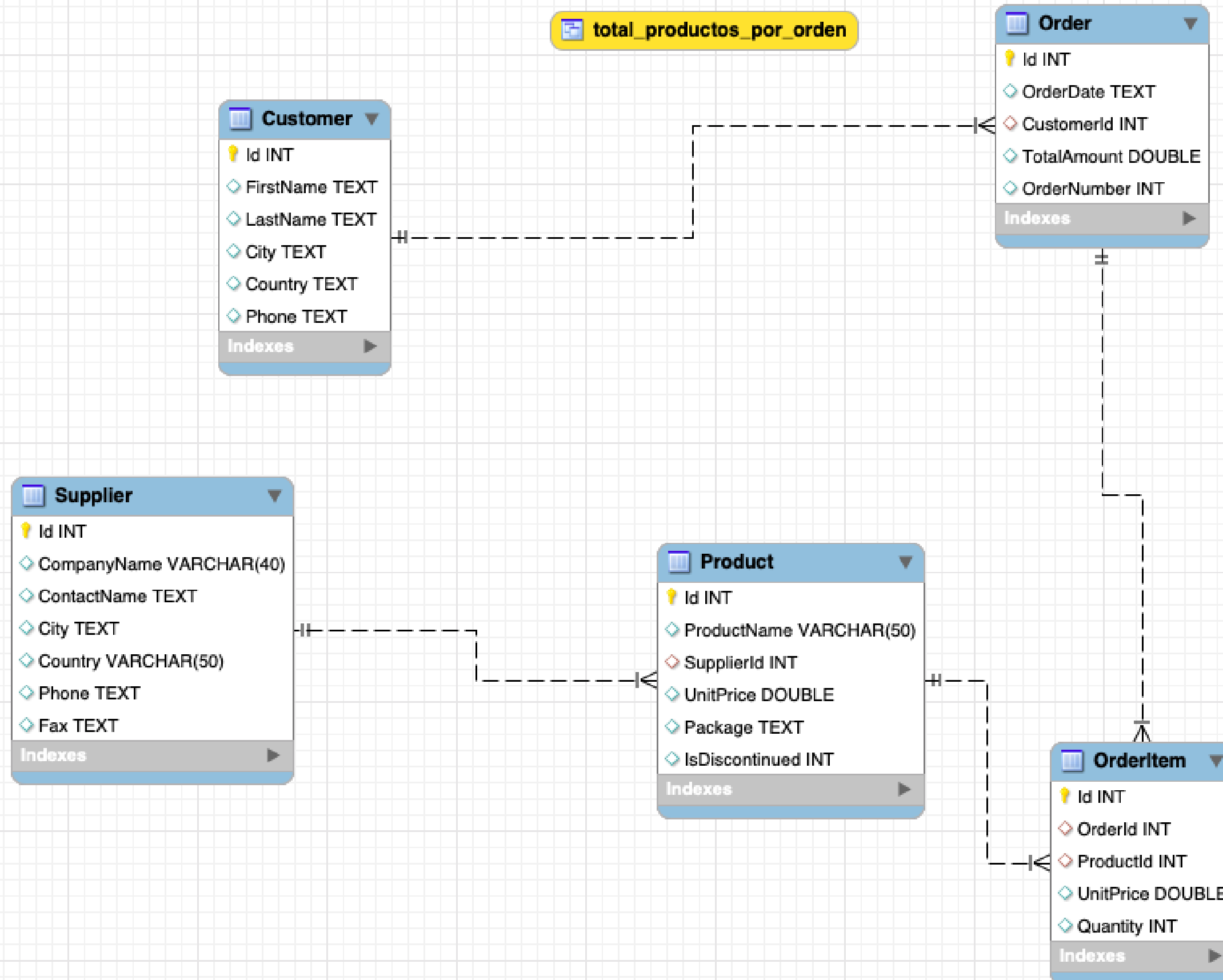
Los socios quieren diseñar una estrategia agresiva para crecer en los mercados mas prósperos en este panorama actual y necesitan la información más exacta y fácil de obtener posible.

¿Como lo voy a hacer?

Seleccionar y depurar la información.

Definir el tipo de datos y cargar los archivos.

Definir llaves primarias y establecer las relaciones entre tablas para crear llaves foraneas.



¿Como lo voy a hacer?

La carga correcta permite relacionar la información entre tablas.

La relación entre las tablas "Product" y "Supplier" permite saber por proveedor cuantos productos suministra y el detalle de los mismos.

Administration

Schemas

sample-model*

SQL File 4*

SQL File 4*

PostWork*

SCHEMAS

Filter objects

JOG

Postwork

Tables

Customer

Order

OrderItem

Product

Supplier

Views

clientes

clientes_ii

total_productos_por_orden

Stored Procedures

Functions

sys

11

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

12

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

13

-- 7.- ¿Cuales 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

17

group by a.Id

18

order by total desc;

19

-- 8.- ¿el total de ordenes por pais?

20

• select Country, round(count(b.TotalAmount),2) Total from Supplier a

21

left join `Order` b

22

on a.Id=b.CustomerId

23

group by Country

24

order by Total desc;

25

-- 9.- ¿Cual es el pais con más total de ingresos?

26

• select Country, round(sum(b.TotalAmount),2) Total from Supplier a

27

left join `Order` b

100%

20:18

Result Grid

Filter Rows: Search

Export:

	Id	CompanyName	total	
▶	7	Pavlova, Ltd.	5	
	12	Plutzer Lebensmittelgroßmärkte AG	5	
	2	New Orleans Cajun Delights	4	
	8	Specialty Biscuits, Ltd.	4	
	1	Exotic Liquids	3	
	3	Grandma Kelly's Homestead'	3	
	4	Tokyo Traders	3	
	6	Mayumi's'	3	
	11	Heli Süßwaren GmbH & Co. KG	3	
	14	Formaggi Fortini s.r.l.	3	
	15	Norske Meierier	3	
	16	Bigfoot Breweries	3	
	17	Svensk Sjöföda AB	3	
	20	Leka Trading	3	

Result 4

Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 4	17:05:22	select a.Id, CompanyName, count(*) total from Supplier a left join Product b on a.Id=b.SupplierID group by a.Id order by...	29 row(s) returned	0.00059 sec / 0.000...

Query Completed

¿Como lo voy a hacer?

La carga correcta permite relacionar la información entre tablas.

La relación entre las tablas "Supplier" y "Order" permite visualizar las ventas de cada proveedor a detalle.

AdministrationSchemas

SCHEMAS

Filter objects

JOG

Postwork

Tables

Customer

Order

OrderItem

Product

Supplier

Views

clientes

clientes_ii

total_productos_por_orden

Stored Procedures

Functions

sys

sample-model*SQL File 4*SQL File 4*PostWork*

Don't Limit

```
30 group by Country
31 order by Total desc
32 limit 1;
33 -- 10.- ¿Cual son las 3 ciudades con mas ordenes?
34 • select Country, round(sum(b.TotalAmount), 2) Total from Supplier a
35 left join `Order` b
36 on a.Id=b.CustomerId
37 group by Country
38 order by Total desc;
39 -- 11.- Crea una vista que muestra el total de unidades vendidas por producto
40 • create view Total_productos_por_orden as (select a.Id, a.ProductName, sum(b.Quantity) Total from Product a
41 left join OrderItem b
42 on a.Id=b.ProductId
43 group by a.Id
44 order by Total desc);
45 -- 11a.- ¿Cual es el nombre de los 5 productos más vendidos?
46 • select * from Total_productos_por_orden;
```

100%45:34

Result GridFilter Rows:SearchExport:

Country	Total
Singapore	113236.68
Sydney	32555.55
Canada	29559.41
Sweden	27614.16
Spain	26968.15
USA	25671.06
Brazil	22607.70
Australia	19088.00
Japan	17046.30
Italy	16058.46
Finland	11666.90
France	10313.15
UK	9894.00
Germany	8005.50

Result 11Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
13	17:20:26	select Country, round(sum(b.TotalAmount), 2) Total from Supplier a left join `Order` b on a.Id=b.CustomerId group by C...	17 row(s) returned	0.00090 sec / 0.000...

Query Completed

¿Como lo voy a hacer?

Administration

Schemas

SCHEMAS

Filter objects

JOG

Postwork

Tables

Customer

Order

OrderItem

Product

Supplier

Views

clientes

clientes_ii

total_productos_por_orden

Stored Procedures

Functions

sys

sample-model*

SQL File 4*

SQL File 4*

PostWork*

Don't Limit

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 desc;

73

100%

53:72

Result Grid

Filter Rows:

Search

Export:

	Nombre	Total_Quantity	Total_Amount
▶	Jose Pavarotti	481076.03	4958
	Roland Mendel	426348.25	4543
	Horst Kloss	417617.35	3961
	Paula Wilson	179124.60	1383
	Patricia McKenna	156949.88	1684
	Philip Cramer	114561.63	903
	Lúcia Carvalho	106095.65	1031
	Peter Franken	94919.07	1525
	Jean Fresnière	94535.40	966
	Maria Larsson	93623.10	1234
	Karl Jablonski	91666.50	1063
	Pascale Cartrain	91125.40	1072
	Christina Berglund	81376.45	1001
	Laurence Lebihan	70917.35	980
	Clientes 2		

Read Only

Action Output

Time

Action

Response

Duration / Fetch Time

2

16:53:04

select * from Clientes order by Total_Quantity desc

89 row(s) returned

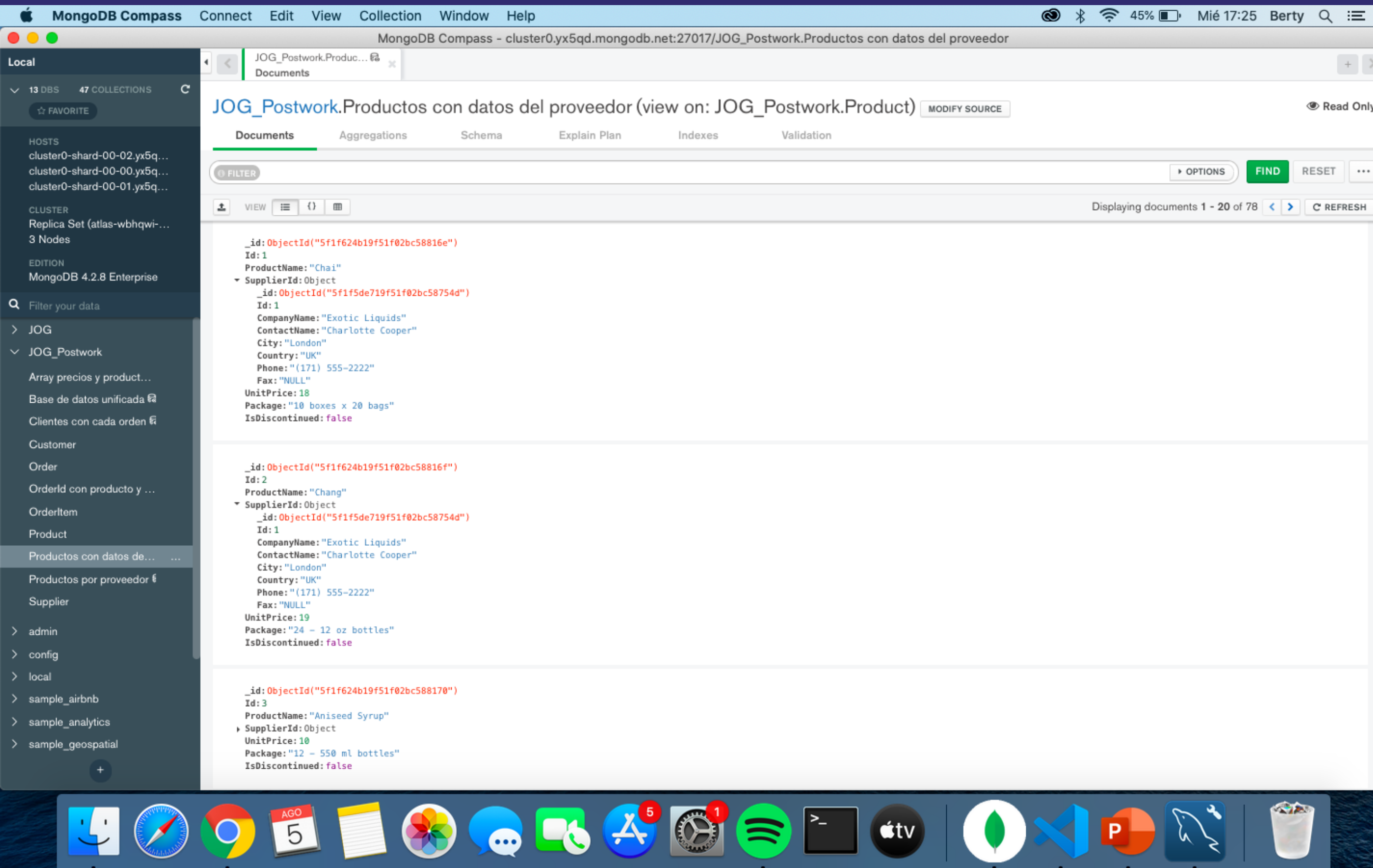
0.013 sec / 0.000025...

Query Completed

La carga correcta permite relacionar la información entre tablas.

La relación entre las tablas "Orders", "OrderItem" y "Customer" permite saber a detalle el estado de la cuenta de cada cliente.

¿Como lo voy a hacer?



Siguiendo esta misma estructura en mongo es posible crear vistas que arrojen información detallada por medio de las llaves de cada tabla.

Es posible generar una vista que arroje por cada producto que proveedor lo suministra.

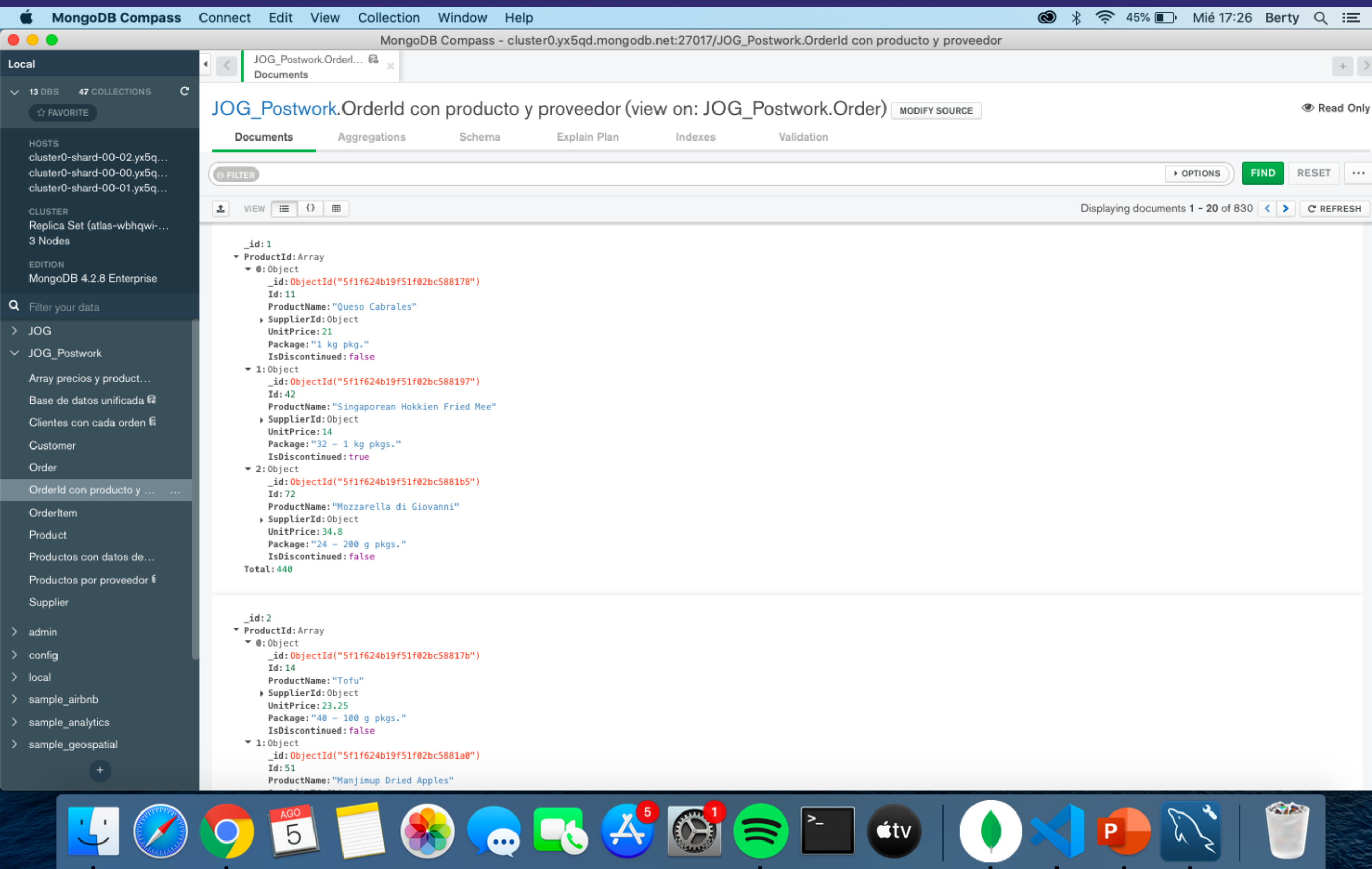
¿Como lo voy a hacer?

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

Siguiendo esta misma estructura en mongo es posible crear vistas que arrojen información detallada por medio de las llaves de cada tabla.

Es posible generar una vista que arroje por cada producto que proveedor lo suministra.

¿Como lo voy a hacer?



Siguiendo esta misma estructura en mongo es posible crear vistas que arrojen información detallada por medio de las llaves de cada tabla.

Esta vista puede agregarse a la tabla orden para indicar que productos fueron vendidos en cada una.

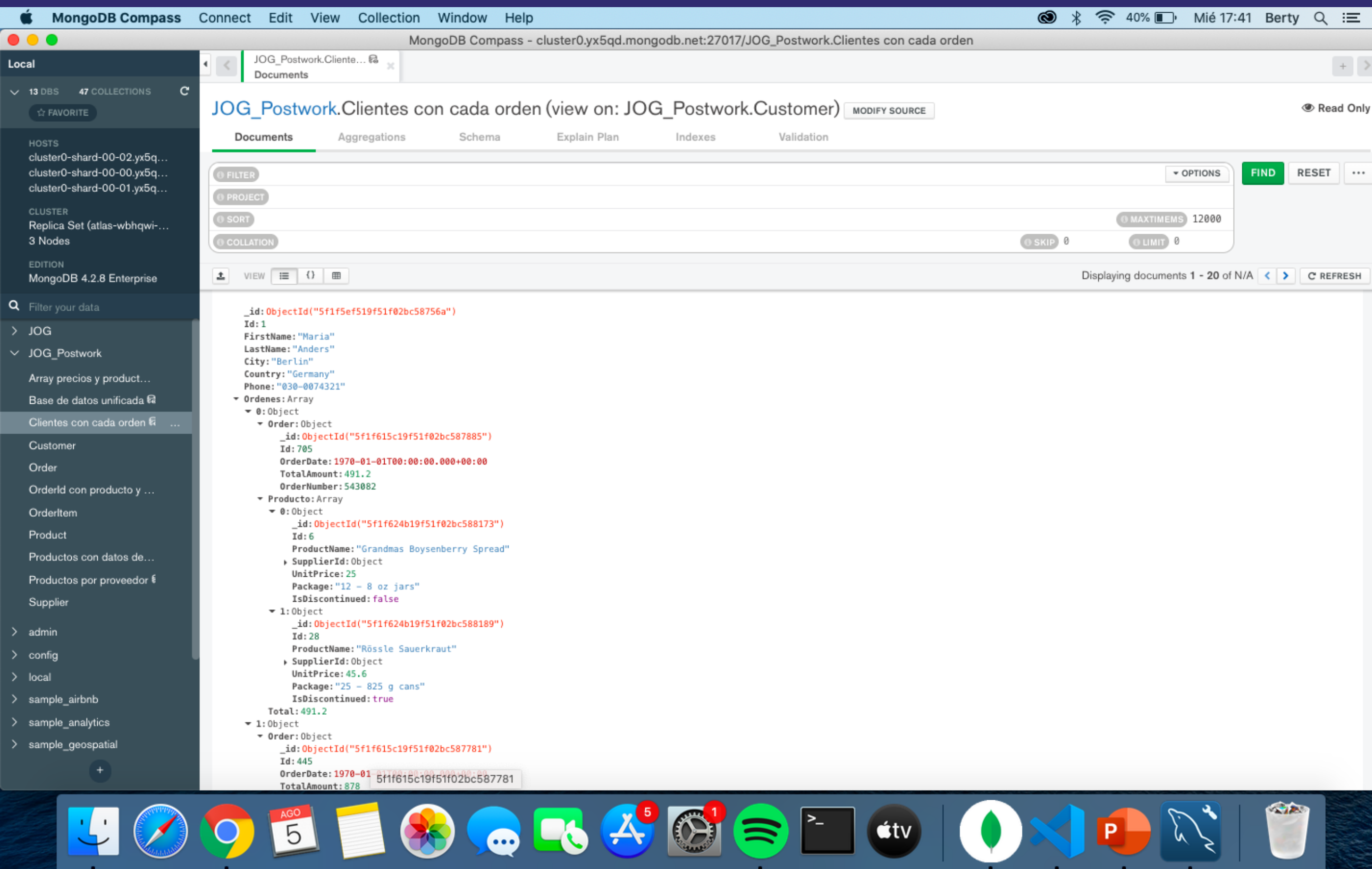
¿Como lo voy a hacer?

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

Siguiendo esta misma estructura en mongo es posible crear vistas que arrojen información detallada por medio de las llaves de cada tabla.

Esta vista puede agregarse a la tabla orden para indicar que productos fueron vendidos en cada una.

¿Como lo voy a hacer?



Siguiendo esta misma estructura en mongo es posible crear vistas que arrojen información detallada por medio de las llaves de cada tabla.

Y finalmente puede añadirse esta información a la tabla "Customer" para indicar todo el detalle de las órdenes compradas por cada cliente.

¿Como lo voy a hacer?

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

Siguiendo esta misma estructura en mongo es posible crear vistas que arrojen información detallada por medio de las llaves de cada tabla.

Y finalmente puede añadirse esta información a la tabla "Customer" para indicar todo el detalle de las órdenes compradas por cada cliente.

Conclusiones



01

Mongo es muy efectivo para mostrar un mayor detalle de cada documento gracias a la propiedad de anidación.

02

MySQL Workbench es muy eficiente para realizar operaciones entre registros y por medio de las vistas acceder a resultados más completos.

03

Seleccionar, depurar y definir el tipo de cada dato es la clave para un sistema útil. Apartir de una buena organización se puede ejecutar cualquier query.

