

Proyecto SuperStoreUS

Definición del proyecto

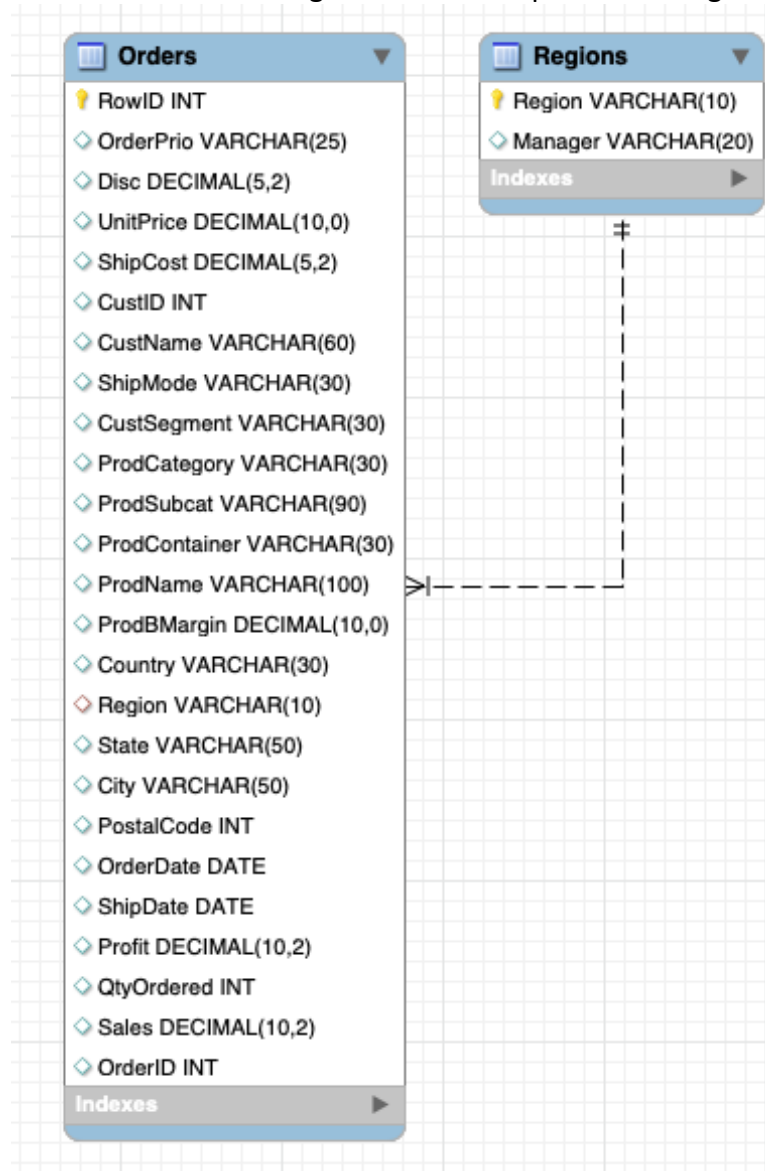
1. **Fuente de BD:** <https://frogames.es/tableau-datasets-del-curso/>
2. **Problema a resolver**

Se analizan las ventas de una tienda con sucursales en todo Estados Unidos en el semestre de Enero a Junio de 2015. Se analizan 4 regiones, entre ellas sus costos, ventas, canales de distribución, entre otras.

Definición de Base de Datos

La base de datos está conformada por dos tablas llamadas:

- Orders: contiene el detalle de las órdenes.
- Regions: contiene el nombre de las regiones con su respectivo manager.



GLOSARIO

Tabla Orders

RowID (INT): identificador de línea
OrderPrio (VARCHAR): prioridad de la orden
Disc (Decimal): descuento del precio
UnitPrice (Decimal): Precio unitario del producto
ShipCost (Decimal): Costo de envío
CustID (INT): identificador del cliente
CustName (VARCHAR): nombre completo del cliente
ShipMode (VARCHAR): método de envío
CustSegment (VARCHAR): segment de negocio del cliente
ProdCategory (VARCHAR): categoría del producto
ProdSubcat (VARCHAR): subcategoría del producto
ProdContainer (VARCHAR): tipo de empaque del producto
ProdName (VARCHAR): nombre del producto
ProdBMargin (DECIMAL): margen base del producto
Country (VARCHAR): país
Region (VARCHAR): nombre de la región
State (VARCHAR): nombre del estado
City (VARCHAR): nombre de la ciudad
PostalCode (INT): código postal
OrderDate (date): fecha en que se creó la orden
ShipDate (date): fecha en que se envi la orden
Profit (Decimal): ganancia
QtyOrdered (INT): cantidad de product ordenada
Sales (Decimal): ingreso por venta
OrderID (INT): identificador de orden)

Tabla Orders

Region (VARCHAR): nombre de la región
Manager (VARCHAR): nombre del manager de la región

Creación de DB en MySQL

-- Crear base de datos

```
CREATE DATABASE IF NOT EXISTS SuperStoreUS;
```

USE SuperStoreUS;

-- Crear tabla Orders

```
CREATE TABLE IF NOT EXISTS Orders (  
    RowID INT PRIMARY KEY,  
    OrderPrio VARCHAR(25),  
    Disc Decimal(5,2),  
    UnitPrice Decimal,  
    ShipCost Decimal(5,2),  
    CustID INT,  
    CustName VARCHAR (60),  
    ShipMode VARCHAR(30),  
    CustSegment VARCHAR(30),  
    ProdCategory VARCHAR(30),  
    ProdSubcat VARCHAR(90),  
    ProdContainer VARCHAR(30),  
    ProdName VARCHAR(100),  
    ProdBMargin DECIMAL,  
    Country VARCHAR(30),  
    Region VARCHAR(10),  
    State VARCHAR(50),  
    City VARCHAR(50),  
    PostalCode INT,  
    OrderDate date,  
    ShipDate date,  
    Profit Decimal(10,2),  
    QtyOrdered INT,  
    Sales Decimal(10,2),  
    OrderID INT,  
    FOREIGN KEY (Region) REFERENCES Regions(Region)  
);
```

-- Crea tabla Regions

```
CREATE TABLE IF NOT EXISTS Regions (  
    Region VARCHAR(10) PRIMARY KEY,  
    Manager VARCHAR(20)  
);
```

Creación de DB en Mongo

Collections

Create collection

View

Sort by

Collection Name

Managers por Estado

VIEW

READ-ONLY

View on:
Orders

Orders

Storage size: 409.60 kB	Documents: 2 K	Avg. document size: 575.00 B	Indexes: 1	Total index size: 32.77 kB
----------------------------	-------------------	---------------------------------	---------------	-------------------------------

Regions

Storage size: 20.48 kB	Documents: 4	Avg. document size: 58.00 B	Indexes: 1	Total index size: 20.48 kB
---------------------------	-----------------	--------------------------------	---------------	-------------------------------

CONSULTAS EN MySQL y MongoDB

1.- ¿Cuántas ordenes fueron de prioridad crítica?

Select count(*) from orders where OrderPrio = "Critical";

The screenshot shows a MySQL query execution interface. The query is: `-- 1.- ¿Cuántas ordenes fueron de prioridad crítica?`
`Select count(*) from orders where OrderPrio = "Critical";`
The interface shows a progress bar at 100% and a duration of 1:44. Below the query, there is a 'Result Grid' section. The grid has one column labeled 'count(*)' and one row with the value '383'.

```
{{ $match: {  
  OrderPrio: 'Critical'  
}}, { $group: {  
  _id: null,  
  OrdenesCriticas: {  
    $sum: 1  
  }  
}}}
```

The screenshot shows the MongoDB Aggregations interface for the 'SuperStoreUS.Orders' collection. The interface includes tabs for Documents, Aggregations (selected), Schema, Explain Plan, Indexes, and Validation. The aggregation pipeline is defined as follows:

```
1- /**  
2-  * query: The query in MQL.  
3-  */  
4- {  
5-   OrderPrio: 'Critical'  
6- }
```

The results of the aggregation are shown in two columns. The first column shows the output of the \$match stage, and the second column shows the output of the \$group stage. The \$group stage output is:

```
_id: null  
OrdenesCriticas: 390
```

2.- ¿Cuáles son los 5 productos más caros?

Select distinct ProdName, UnitPrice from orders order by UnitPrice desc limit 5;

```
45 -- 2.- ¿Cuáles son los 5 productos más caros?
46 Select distinct ProdName, UnitPrice from orders order by UnitPrice desc limit 5;
```

100% 81:46

Result Grid Filter Rows: Search Export: Fetch rows:

	ProdName	UnitPrice
▶	Polycom ViewStation ISDN Videoconferencing...	6783
	Okidata Pacemark 4410N Wide Format Dot Mat...	3502
	Canon imageCLASS 2200 Advanced Copier	3500
	Epson DFX-8500 Dot Matrix Printer	2550
	Lexmark 4227 Plus Dot Matrix Printer	2036

PROJECT

```
{
  ProdName: 1,
  UnitPrice: 1,
  _id: 0
}
```

SORT

```
{
  UnitPrice: -1
}
```

LIMIT

5

SuperStoreUS.Orders

DOCUMENTS 2.0k STORAGE SIZE 360.4KB AVG. SIZE 575B INDEXES 1 TOTAL SIZE 32.8KB AVG. SIZE 32.8KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } **OPTIONS** **FIND** **RESET** **...**

PROJECT {ProdName:1, UnitPrice:1, _id:0}

SORT {UnitPrice:-1} **MAX TIME MS** 60000

COLLATION { locale: 'simple' } **SKIP** 0 **LIMIT** 5

VIEW **{}** **...** Displaying documents 1 - 5 of 5 **REFRESH**

UnitPrice: 6783.02 ProdName: "Polycom ViewStation ISDN Videoconferencing Unit"
UnitPrice: 6783.02 ProdName: "Polycom ViewStation ISDN Videoconferencing Unit"
UnitPrice: 6783.02 ProdName: "Polycom ViewStation ISDN Videoconferencing Unit"
UnitPrice: 6783.02 ProdName: "Polycom ViewStation ISDN Videoconferencing Unit"
UnitPrice: 3502.14 ProdName: "Okidata Pacemark 4410N Wide Format Dot Matrix Printer"

3.- ¿Cuál fue la región con menos órdenes?

```
Select Region, count(*) as OrdersRegion from orders
group by Region
order by OrdersRegion limit 1;
```

47 -- 3.- ¿Cuál fue la región con menos órdenes?
48 • Select Region, count(*) as OrdersRegion from orders
49 group by Region order by OrdersRegion limit 1;

100% 47:49

Result Grid Filter Rows: Search Export: Fetch rows:

Region	OrdersRegion
South	435

4.- ¿Quiénes fueron los 3 clientes con más órdenes?

```
Select CustName, count(*) as OrdersCust from orders
group by CustName order by OrdersCust desc limit 3;
```

50 -- 4.- ¿Quiénes fueron los 3 clientes con más órdenes?
51 • Select CustName, count(*) as OrdersCust from orders
52 group by CustName order by OrdersCust desc limit 3;

100% 47:49

Result Grid Filter Rows: Search Export: Fetch rows:

CustName	OrdersCust
Jenny Gold	9
Andrew Gonzalez	8
Louis Parrish	7

5.- Cuál fue el método de envío más utilizado?

```
Select ShipMode, count(*) as OrdersShip from orders
group by ShipMode order by OrdersShip desc limit 1;
```

53 -- 5.- Cuál fue el método de envío más utilizado?
54 • Select ShipMode, count(*) as OrdersShip from orders
55 group by ShipMode order by OrdersShip desc limit 1;
56

00% 1:56

Result Grid Filter Rows: Search Export: Fetch rows:

ShipMode	OrdersShip
Regular Air	1427

6.- ¿Cuál es el producto con el costo de envío más caro?

Select ProdName, ShipCost from orders

order by ShipCost desc limit 1;

```
56 -- 6.- ¿Cuál es el producto con el costo de envío más caro?
57 * Select ProdName, ShipCost from orders
58   order by ShipCost desc limit 1;
59
```

100% 52:55

Result Grid Filter Rows: Search Export: Fetch rows:

ProdName	ShipCost
Global Leather and Oak Executive Chair. Black	164.73

PROJECT

```
{
  ProdName: 1,
  ShipCost: 1,
  _id: 0
}
```

SORT

```
{
  ShipCost: -1
}
```

LIMIT

1

SuperStoreUS.Orders

DOCUMENTS 2.0k STORAGE SIZE 360.4KB AVG. SIZE 575B INDEXES 1 TOTAL SIZE 32.8KB AVG. SIZE 32.8KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } **OPTIONS** **FIND** **RESET** **...**

PROJECT { ProdName: 1, ShipCost: 1, _id: 0. }

SORT { ShipCost: -1 } **MAX TIME MS** 60000

COLLATION { locale: 'simple' } **SKIP** 0 **LIMIT** 1

VIEW **{} {} {}** Displaying documents 1 - 1 of 1 **REFRESH**

```
ShipCost: 164.73
ProdName: "Global Leather and Oak Executive Chair. Black"
```


7.- ¿Cuáles clientes se apellidan Simpson?

Select distinct CustName from orders where CustName like '%Simpson';

```
59 -- 7.- ¿Cuáles clientes se apellidan Simpson?
60 • Select distinct CustName from orders where CustName like '%Simpson';
```

100% 32:58

Result Grid Filter Rows: Search Export:

	CustName
▶	Ross Simpson
	Eva Simpson
	Helen Simpson

FILTER

```
{
  CustName: RegExp('Simpson', i)
}
```

PROJECT

```
{
  CustName: 1,
  _id: 0
}
```

SORT

```
{
  ShipCost: -1
}
```

SuperStoreUS.Orders

DOCUMENTS 2.0k STORAGE SIZE 360.4KB AVG. SIZE 575B INDEXES 1 TOTAL SIZE 32.8KB AVG. SIZE 32.8KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER `{CustName:/Simpson/i}` **PROJECT** `{ CustName: 1, _id: 0.}` **SORT** `{ShipCost: -1}` **COLLATION** `{ locale: 'simple' }` **MAX TIME MS** 60000 **SKIP** 0 **LIMIT** 0

FIND **RESET** **REFRESH**

VIEW **JSON** **YAML** **SQL** **CSV**

Displaying documents 1 - 6 of 6

CustName: "Helen Simpson"
CustName: "Ross Simpson"
CustName: "Eva Simpson"

8.- ¿Cuáles productos son de la marca Belkin?

Select distinct ProdName from orders where ProdName like 'Belkin%';

62 • Select distinct ProdName from orders where ProdName like 'Belkin%';

100% 46:59

Result Grid Filter Rows: Search Export:

ProdName
Belkin 105-Key Black Keyboard
Belkin 107-key enhanced keyboard. USB/PS/2 i...
Belkin 6 Outlet Metallic Surge Strip
Belkin MediaBoard 104- Keyboard
Belkin 8 Outlet SurgeMaster II Gold Surge Prote...
Belkin F9M820V08 8 Outlet Surge
Belkin ErgoBoard Keyboard
Belkin Premiere Surge Master II 8-outlet surge...
Belkin 325VA UPS Surge Protector. 6'

FILTER

```
{
  ProdName: RegExp('Belkin', i)
}
```

PROJECT

```
{
  ProdName: 1,
  _id: 0
}
```

SuperStoreUS.Orders

DOCUMENTS 2.0k STORAGE SIZE 360.4KB AVG. SIZE 575B INDEXES 1 TOTAL SIZE 32.8KB AVG. SIZE 32.8KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER: {ProdName: /Belkin/i} OPTIONS FIND RESET

PROJECT: {ProdName: 1, _id: 0}

SORT: { field: -1 } or [['field', -1]] MAX TIME MS 60000

COLLATION: { locale: 'simple' } SKIP 0 LIMIT 0

VIEW {}

Displaying documents 1 - 20 of 21 REFRESH

ProdName: "Belkin 6 Outlet Metallic Surge Strip"

ProdName: "Belkin 105-Key Black Keyboard"

ProdName: "Belkin 107-key enhanced keyboard. USB/PS/2 interface"

ProdName: "Belkin 8 Outlet SurgeMaster II Gold Surge Protector"

9.- ¿Cuál es el menor, mayor y promedio de costo de envío?

Select MIN(ShipCost),MAX(ShipCost),AVG(ShipCost) from orders;

63

-- 9.- ¿Cuál es el menor, mayor y promedio de costo de envío?

64

Select MIN(ShipCost),MAX(ShipCost),AVG(ShipCost) from orders;

100%

70:60

Result Grid

Filter Rows:

Search

Export:

MIN(ShipCost)

MAX(ShipCost)

AVG(ShipCost)

0.49

164.73

12.787044

```

[{$group: {
  _id: null,
  MenorShipCost: {
    $min: '$ShipCost'
  },
  MayorShipCost: {
    $max: '$ShipCost'
  },
  AvgShipCost: {
    $avg: '$ShipCost'
  }
}}

```

SuperStoreUS.Orders

DOCUMENTS 2.0k STORAGE SIZE 360.4KB AVG. SIZE 575B INDEXES 1 TOTAL SIZE 32.8KB AVG. SIZE 32.8KB

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE SAMPLE MODE AUTO PREVIEW

Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)

```

_id: ObjectId("61f2cd6614fdd2c634815b52")
RowID: "20847"
OrderPrio: "High"
Disc: 0.01
UnitPrice: 2.84
ShipCost: 0.93
CustID: "3"
CustName: "Bonnie Potter"

```

```

_id: ObjectId("61f2cd6614fdd2c634815b5")
RowID: "20228"
OrderPrio: "Not Specified"
Disc: 0.02
UnitPrice: 500.98
ShipCost: 26
CustID: "5"
CustName: "Ronnie Proctor"

```

Output after \$group stage (Sample of 1 document)

```

1- /**
2-  * _id: The id of the group.
3-  * fieldN: The first field name.
4-  */
5- {
6-   _id: null,
7-   MenorShipCost: {
8-     $min: "$ShipCost"
9-   },
10-   MayorShipCost: {
11-     $max: "$ShipCost"
12-   },
13-   AvgShipCost: {
14-     $avg: "$ShipCost"
15-   }
16- }

```

```

_id: null
MenorShipCost: 0.49
MayorShipCost: 164.73
AvgShipCost: 12.968150614754098

```

10.- ¿Cuántas órdenes generaron pérdidas de la región Sur?

Select count(*) from orders where Profit < 0 AND Region = 'South';

```
65 -- 10.- ¿Cuántas órdenes generaron pérdidas de la región Sur?
66 Select count(*) from orders where Profit < 0 AND Region = 'South';
```

00% 16:66

Result Grid Filter Rows: Search Export:

count(*)
230

```
[{$match: {
  $and: [
    {
      Profit: {
        $lt: 0
      }
    },
    {
      Region: 'South'
    }
  ]
}}, {$group: {
  _id: null,
  PerdidasSur: {
    $sum: 1
  }
}}]
```

SuperStoreUS.Orders

DOCUMENTS 2.0k STORAGE SIZE 360.4KB AVG. SIZE 575B INDEXES 1 TOTAL SIZE 32.8KB AVG. SIZE 32.8KB

Documents Aggregations Schema Explain Plan Indexes Validation

Collation: Untitled- Modified SAVE SAMPLE MODE AUTO PREVIEW

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

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   $and: [{Profit: {$lt: 0}}, {Region: 'South'}]
6 }
```

```
{Container: "Small Box"
  Name: "Newell 3-Hole Punched Plastic Slotted Magazine Holders for Binders"
  Margin: 0.37
  Country: "United States"
  Region: "South"
  State: "Tennessee"
  City: "Nashville"
  PostalCode: 37211
  OrderDate: 1970-01-01T00:00:00.000+00:00}

{Container: "Jumbo Box"
  Name: "O'Sullivan Living Dimensions 3-Shelf Bookcases"
  Margin: 0.75
  Country: "United States"
  Region: "South"
  State: "Tennessee"
  City: "Morristown"
  PostalCode: 37814
  OrderDate: 1970-01-01T00:00:00.000+00:00}
```

\$group Output after \$group stage (Sample of 1 document)

```
1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   _id: null,
7   PerdidasSur: {
8     $sum: 1
9   }
10 }
```

```
{_id: null
  PerdidasSur: 233}
```

11.- ¿Cuál fue el estado con más ganancias?

Select State, sum(Profit) as StateProfit from orders
group by State order by StateProfit desc limit 1;

```
67 -- 11.- ¿Cuál fue el estado con más ganancias?
68 • Select State, sum(Profit) as StateProfit from orders
69 group by State order by StateProfit desc limit 1;
```

100% 30:67

Result Grid Filter Rows: Search Export: Fetch rows:

State	StateProfit
California	36187.57

***ProfitxPais debía ser ProfitxEstado**

```
{{ $project: {
  State: 1,
  Profit: 1,
  _id: 0
}}, { $group: {
  _id: '$State',
  ProfitxPais: {
    $sum: '$Profit'
  }
}}, { $sort: {
  ProfitxPais: -1
}}, { $limit: 1 }}
```

SuperStoreUS.Orders

DOCUMENTS 2.0k STORAGE SIZE 360.4KB AVG. SIZE 575B INDEXES 1 TOTAL SIZE 32.8KB AVG. SIZE 32.8KB

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE SAMPLE MODE AUTO PREVIEW

```
5 {
6   _id: "$State",
7   ProfitxPais: {
8     $sum: "$Profit"
9   }
10 }
```

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

```
1 /**
2  * Provide any number of field/order pairs.
3  */
4 {
5   ProfitxPais: -1
6 }
```

Output after \$limit stage (Sample of 1 document)

```
1 /**
2  * Provide the number of documents to limit.
3  */
4 1
```

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

_id: "California" ProfitxPais: 37421.960192	_id: "Texas" ProfitxPais: 28078.85066
--	--

Output after \$limit stage (Sample of 1 document)

_id: "California" ProfitxPais: 37421.960192
--

12.- ¿Cuales son los 5 pedidos más grandes entre DC y New York?

Select OrderID, QtyOrdered from orders

where State IN ('New York','District of Columbia')

order by QtyOrdered desc

limit 5;

```
70 -- 12.- ¿Cuales son los 5 pedidos más grandes entre DC y New York?
71 • Select OrderID, QtyOrdered from orders
72   where State IN ('New York','District of Columbia')
73   order by QtyOrdered desc
74   limit 5;
```

00% 39:67

Result Grid Filter Rows: Search Export: Fetch rows:

	OrderID	QtyOrder...
▶	38852	85
	29350	85
	29350	83
	34435	76
	24455	76

FILTER
{
 \$or: [
 {
 State: 'New York'
 },
 {
 State: 'District of Columbia'
 }
]
}

PROJECT
{
 OrderID: 1,
 State: 1,
 QtyOrdered: 1,
 _id: 0
}
SORT
{
 QtyOrdered: -1,
 OrderID: -1
}

SuperStoreUS.Orders

DOCUMENTS 2.0k STORAGE SIZE 360.4KB AVG. SIZE 575B INDEXES 1 TOTAL SIZE 32.8KB AVG. SIZE 32.8KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER `{ $or: [{ State: 'New York' }, { State: 'District of Columbia' }] }` **OPTIONS** **FIND** **RESET** **↺** **⋮**

PROJECT `{ OrderID: 1, State: 1, QtyOrdered: 1, _id: 0 }`

SORT `{ QtyOrdered: -1, OrderID: -1 }` **MAX TIME MS** 60000

COLLATION `{ locale: 'simple' }` **SKIP** 0 **LIMIT** 0

VIEW **{}** **☰**

Displaying documents 1 - 20 of 144 **↶** **↷** **↻** **REFRESH**

State: "District of Columbia"
QtyOrdered: 85
OrderID: 38852

State: "District of Columbia"
QtyOrdered: 85
OrderID: 29350

State: "District of Columbia"
QtyOrdered: 83
OrderID: 29350

State: "New York"
QtyOrdered: 76
OrderID: 39076

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

13.- ¿Cuál es el nombre del mánager de cada estado? Almacenar como vista

Select distinct State, Manager, o.Region

from Orders o

left join Regions r

on o.Region = r.Region

Order by State asc;

The screenshot shows a SQL IDE with a query editor and a results grid. The query is as follows:

```

75 -- 13.- ¿Cuál es el nombre del mánager de cada estado? Almacenar como vista
76 • Create View ManagersState AS
77   Select distinct State, Manager, o.Region
78   from Orders o
79   left join Regions r
80   on o.Region = r.Region
81   Order by State asc;
82 • Select * from ManagersState;
  
```

The results grid shows the following data:

State	Manager	Region
Arkansas	Sam	South
California	William	West
Colorado	William	West
Connecticut	Erin	East
Delaware	Erin	East
District of...	Erin	East
Florida	Sam	South
Georgia	Sam	South
Idaho	William	West
Illinois	Chris	Central
Indiana	Chris	Central

```

[{$lookup: {
  from: 'Regions',
  localField: 'Region',
  foreignField: 'Region',
  as: 'ManagersRegion'
}}]
  
```

The screenshot shows the MongoDB Compass interface for the 'SuperStoreUS.Orders' collection. The 'Aggregations' tab is active, and a \$lookup stage is configured. The pipeline is as follows:

```

1 // **
2 * from: The target collection.
3 * localField: The local join field.
4 * foreignField: The target join field.
5 * as: The name for the results.
6 * pipeline: The pipeline to run on the joined collection.
7 * let: Optional variables to use in the pipeline.
8 **
9 {
10   $lookup: {
11     from: 'Regions',
12     localField: 'Region',
13     foreignField: 'Region',
14     as: 'ManagersRegion'
15   }
16 }
  
```

The output after the \$lookup stage is shown, displaying a sample of 20 documents. The output includes the original order data and the joined 'ManagersRegion' array.

Document 1	Document 2
<pre> RowID: "26267" OrderPrio: "High" Disc: 0.04 UnitPrice: 2.98 ShipCost: 1.58 CustID: "16" CustName: "Sarah Ramsey" </pre>	<pre> RowID: "24848" OrderPrio: "Medium" Disc: 0.05 UnitPrice: 3.58 ShipCost: 1.63 CustID: "14" CustName: "Gwendolyn F. Tyson" </pre>

Vista:

The screenshot displays the MongoDB Atlas web interface for the 'SuperStoreUS.Orders' collection. The top navigation bar includes tabs for 'Documents', 'Aggregations', 'Schema', 'Explain Plan', 'Indexes', and 'Validation'. The 'Documents' tab is active, showing a list of documents. The first document is expanded, revealing its JSON structure. The document contains fields for 'RowID', 'OrderPrio', 'Disc', 'UnitPrice', 'ShipCost', 'CustID', 'CustName', 'ShipMode', 'CustSegment', 'ProdCategory', 'ProdSubcat', 'ProdContainer', 'ProdName', 'ProdBMargin', 'Country', 'Region', 'State', 'City', 'PostalCode', 'OrderDate', 'ShipDate', 'Profit', 'QtyOrdered', 'Sales', 'OrderID', and 'Region Manager'. The 'Region Manager' field is an array containing an object with 'id' and 'name' properties. The right sidebar shows the database structure, including the 'SuperStoreUS.Orders' collection and a search bar.

SuperStoreUS.Managers por Estado READ-ONLY VIEW

view on: SuperStoreUS.Orders EDIT VIEW

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET REFRESH

VIEW {}

Displaying documents 1 - 20 of 1952

```

RowID: "26267"
OrderPrio: "High"
Disc: 0.04
UnitPrice: 2.98
ShipCost: 1.58
CustID: "16"
CustName: "Sarah Ramsey"
ShipMode: "Regular Air"
CustSegment: "Small Business"
ProdCategory: "Office Supplies"
ProdSubcat: "Rubber Bands"
ProdContainer: "Wrap Bag"
ProdName: "Staples Gold Paper Clips"
ProdBMargin: 0.39
Country: "United States"
Region: "East"
State: "New York"
City: "Syracuse"
PostalCode: 13210
OrderDate: 2015-12-02T06:00:00.000+00:00
ShipDate: 1970-01-01T00:00:00.000+00:00
Profit: 2.63
QtyOrdered: 6
Sales: 18.8
OrderID: 86836
Region Manager: Array
  0: Object
    _id: ObjectId("61f2cd9f14dd2c6348162f5")
    Region: "East"
    Manager: "Erin"
  
```

25 DBS 77 COLL

FAVORITE

HOSTS

- cluster0-shard-00
- cluster0-shard-00
- cluster0-shard-00

CLUSTER

Replica Set (Clus

3 Nodes

EDITION

MongoDB 4.4.11

Filter your data

- nombre
- r1
- restaurants
- sample_airbnb
- sample_analytics
- sample_geospatial
- sample_mflix
- comments
- movies
- sessions

14.- ¿Quién es el manager de New York?

Select Region, Manager

from Regions

Where Region in

(Select Region from SuperStoreUS.orders

Where State = 'New York');

```
81 -- 14.- ¿Quién es el manager de New York?
82 • Select Region, Manager
83   from Regions
84   Where Region in
85   (Select Region from SuperStoreUS.orders
86   Where State = 'New York');
```

100% 23:79

Result Grid Filter Rows: Search Edit: Export/Import:

Region	Manager
East	Erin
HULL	HULL

FILTER

```
{
  State: 'New York'
}
```

PROJECT

```
{
  State: 1,
  'Region Manager': 1,
  _id: 0
}
```

LIMIT

1

SuperStoreUS.Managers por Estado READ-ONLY VIEW view on: SuperStoreUS.Orders EDIT VIEW

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER {State:'New York'} OPTIONS FIND RESET ...

PROJECT {State:1, "Region Manager":1, _id:0}

SORT { field: -1 } or [['field', -1]] MAX TIME MS 60000

COLLATION { locale: 'simple' } SKIP 0 LIMIT 1

VIEW {}

Displaying documents 1 - 1 of 1 REFRESH

```
State: "New York"
Region Manager: Array
  0: Object
    _id: ObjectId("61f2cd9f14fdd2c6348162f5")
    Region: "East"
    Manager: "Erin"
```

15.- ¿Cuánta ganancia generaron los productos Fellowes?

```
89 -- 15.- ¿Cuánta ganancia generaron los productos Fellowes?
90 • Select Sum(Profit) as ProfitFellowes from orders
91 where ProdName like 'Fellowes%';
```

100% 41:87

Result Grid Filter Rows: Search Export:

ProfitFellowes
28388.06

```
{{ $match: {
  ProdName: RegExp('Fellowes', i)
}}, { $project: {
  ProdName: 1,
  Profit: 1,
  _id: 0
}}, { $group: {
  _id: '$ProdName',
  ProfitProd: {
    $sum: '$Profit'
  }
}}, { $group: {
  _id: null,
  ProfitFellowes: {
    $sum: '$ProfitProd'
  }
}}
}}
```

SuperStoreUS.Orders

DOCUMENTS 2.0k STORAGE SIZE 409.6KB AVG. SIZE 575B INDEXES 1 TOTAL SIZE 32.8KB AVG. SIZE 32.8KB

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION { locale: 'simple' }

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

```
1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   _id: "$ProdName",
7   ProfitProd: {
8     $sum: "$Profit"
9   }
10 }
```

Output after \$group stage (Sample of 1 document)

```
1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   _id: null,
7   ProfitFellowes: {
8     $sum: "$ProfitProd"
9   }
10 }
```

Observaciones

- La base datos se pudo haber desmenuzado más para formar más tablas.
- Las consultas más sencillas en SQL se me hicieron complicadas en MongoDB y viceversa.
- Hay detalles de importación en algunos registros de SQL, en MongoDB es más fácil importar.
- Las consultas tanto en MySQL como Mongo facilitan un análisis más visual y estratégico de conjuntos enormes de datos que difícilmente podrían ser analizados utilizando hojas de cálculo como Excel.