Proyecto SuperStoreUS

Definición del proyecto

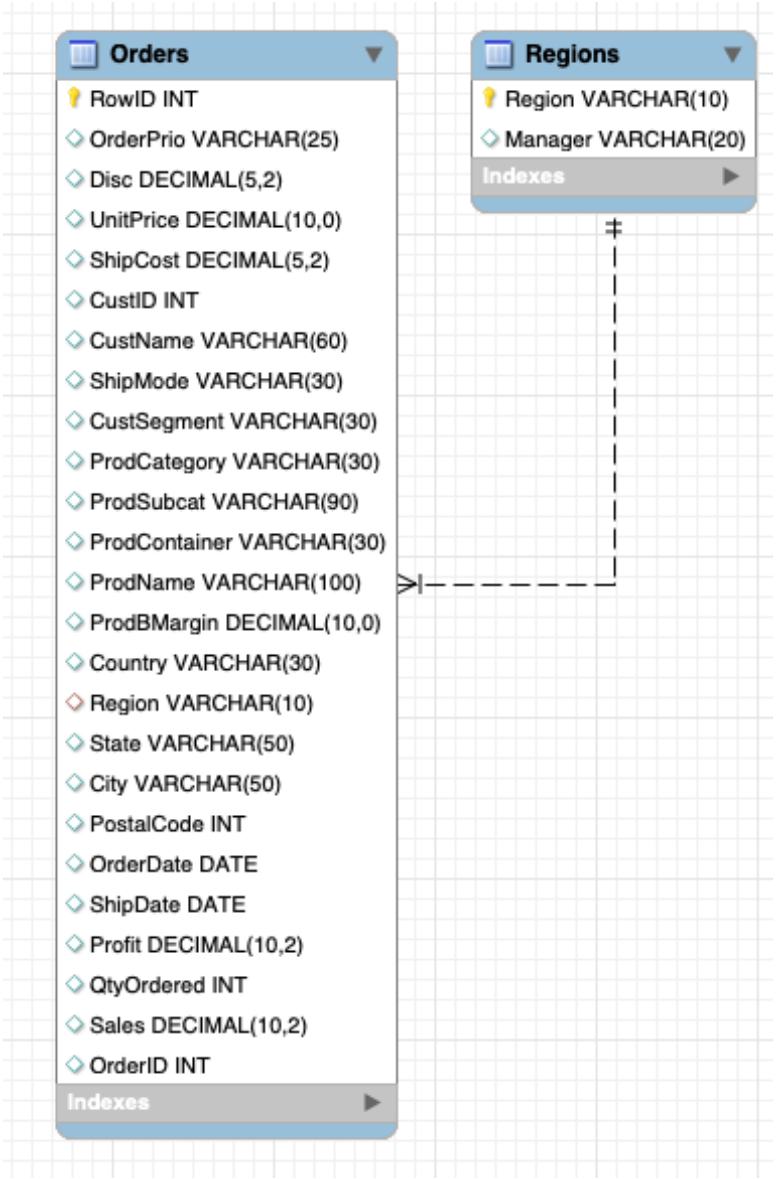
1. **Fuente de BD:** https://frogames.es/tableau-datasets-del-curso/
2. **Problema a resolver**

   Se analizan las ventas de una tienda con sucursales ene 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 envoi 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

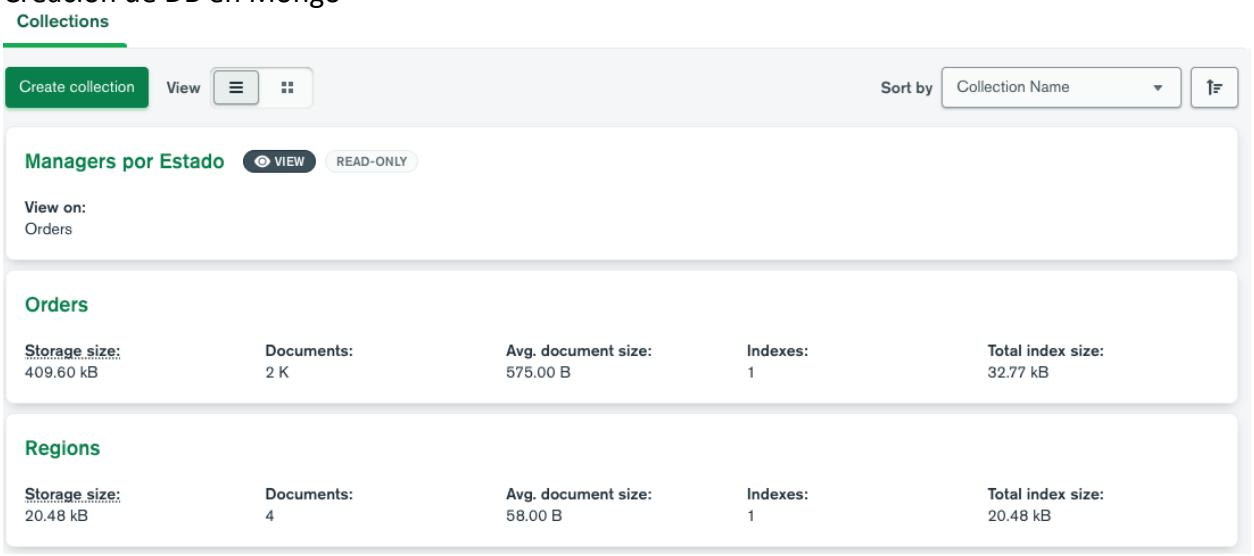
Daniel Eduardo Lobo Carrillo - data-analysis-21-12

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)
);
```

Daniel Eduardo Lobo Carrillo - data-analysis-21-12

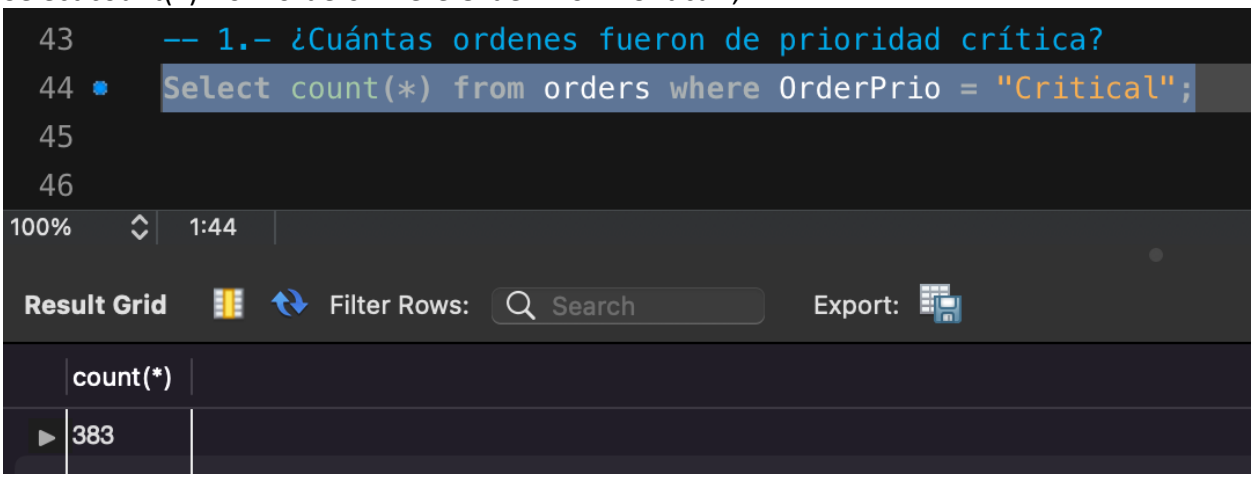
Creación de DB en Mongo



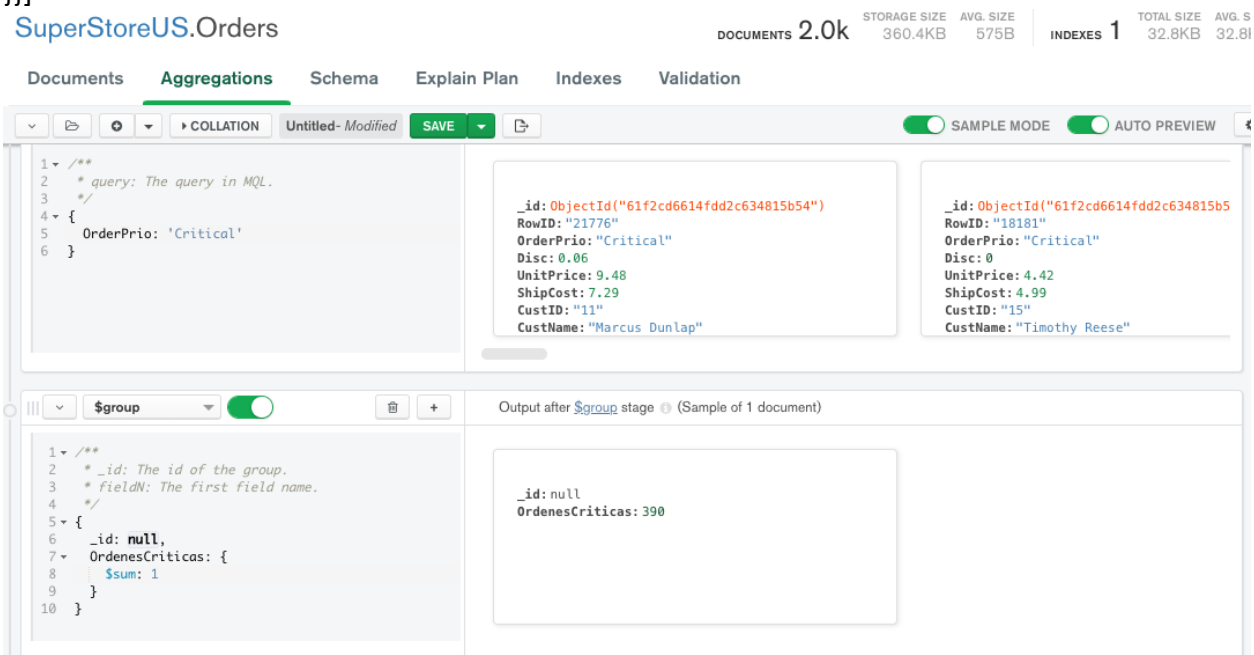
Daniel Eduardo Lobo Carrillo - data-analysis-21-12

CONSULTAS EN MySQL y MongoDB

1.- ¿Cuántas ordenes fueron de prioridad crítica?
Select count(*) from orders where OrderPrio = "Critical";




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



Daniel Eduardo Lobo Carrillo - data-analysis-21-12

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

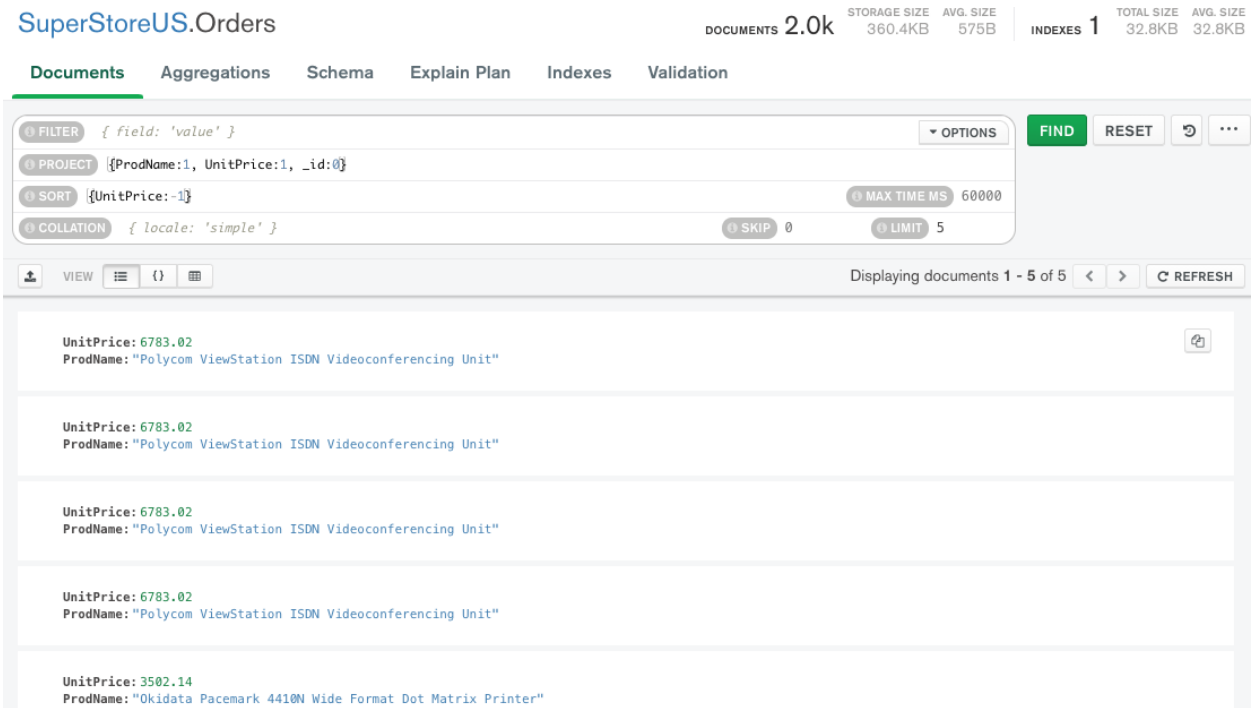
```
45    -- 2.- ¿Cuáles son los 5 productos más caros?
46 •  Select distinct ProdName, UnitPrice from orders order by UnitPrice desc limit 5;
```

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



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;

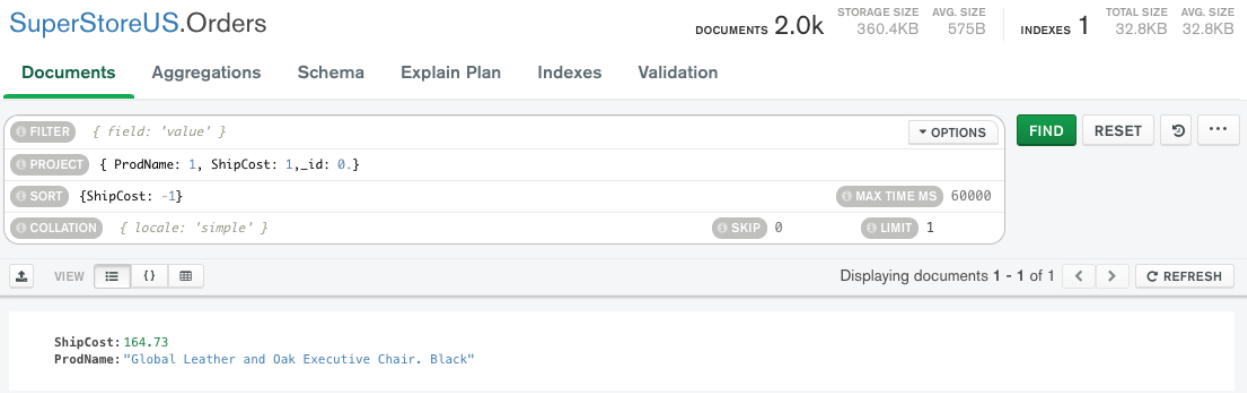
```
56    -- 6.- ¿Cuál es el producto con el costo de envío más caro?
57    Select ProdName, ShipCost from orders
58    order by ShipCost desc limit 1;
59
```

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



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} OPTIONS FIND RESET

PROJECT { CustName: 1,_id: 0.}

SORT {ShipCost: -1} MAX TIME MS 60000

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

VIEW Displaying documents 1 - 6 of 6 REFRESH

```
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%';

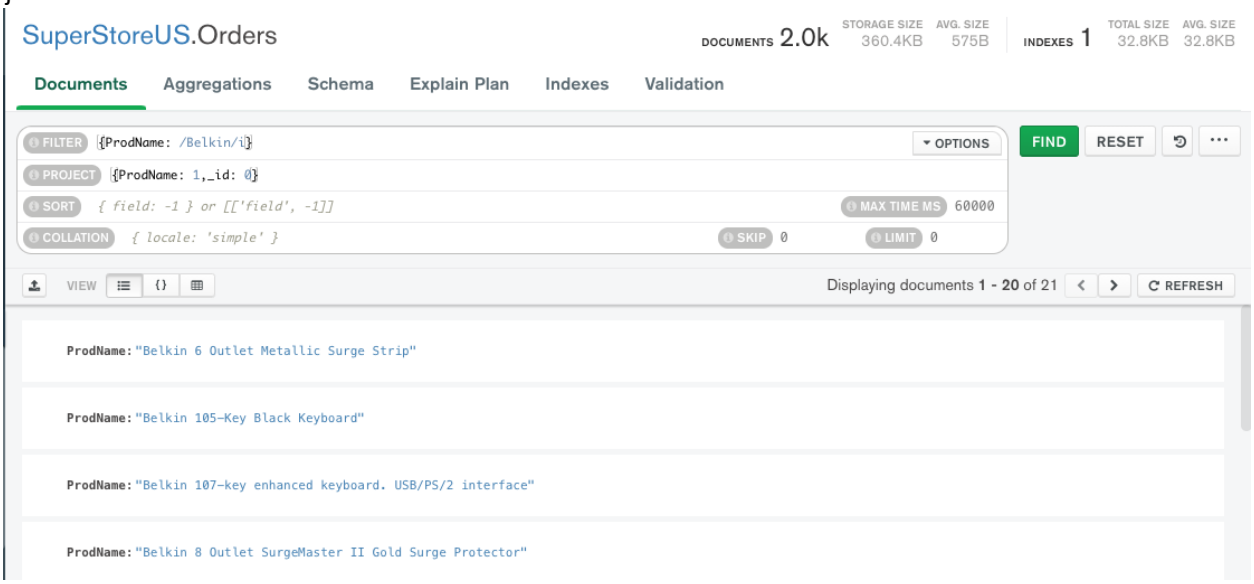
```
62 •  Select distinct ProdName from orders where ProdName like 'Belkin%';
```

100% 46:59

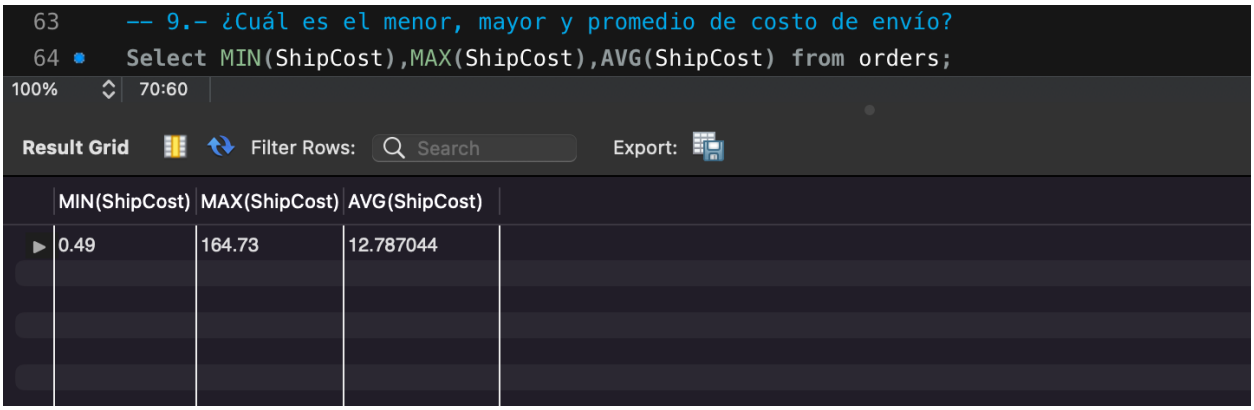
Result Grid | Filter Rows: Search | Export:

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'

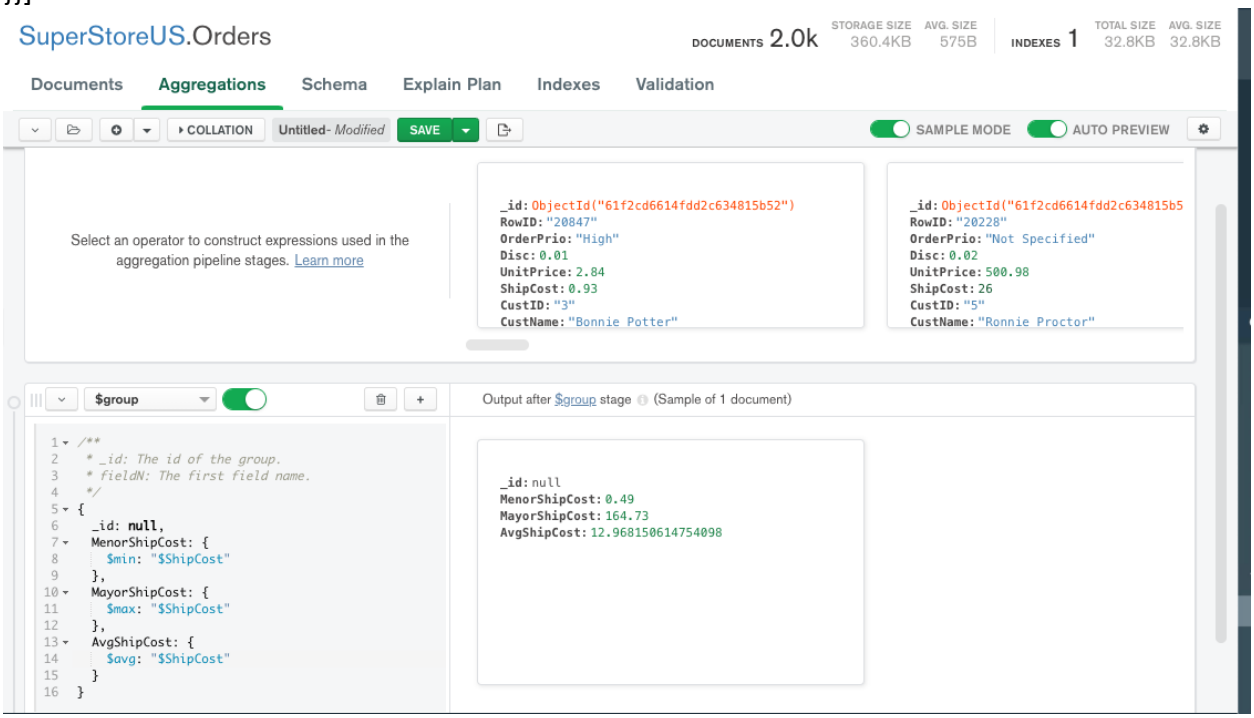
FILTER
{
ProdName: RegExp('Belkin', i)
}
PROJECT
{
ProdName: 1,
_id: 0
}



9.- ¿Cuál es el menor, mayor y promedio de costo de envío?
Select MIN(ShipCost),MAX(ShipCost),AVG(ShipCost) from orders;




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



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

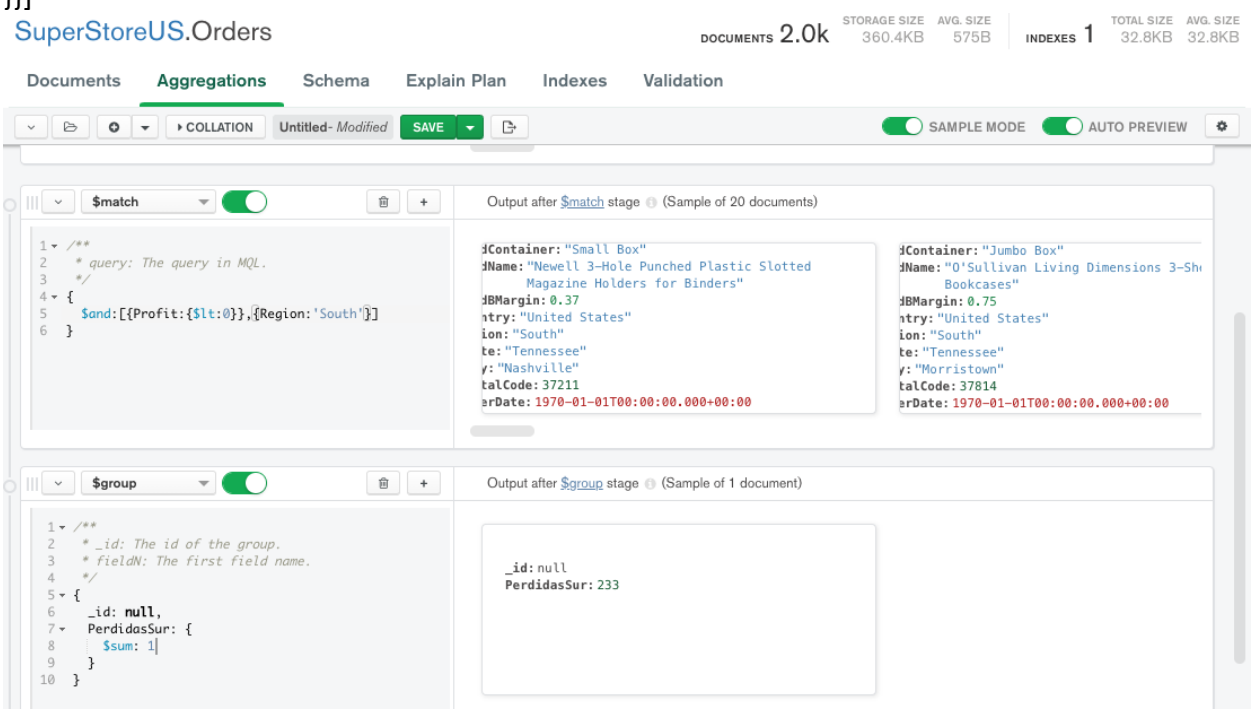
```
65    -- 10.- ¿Cuántas órdenes generaron pérdidas de la región Sur?
66    Select count(*) from orders where Profit < 0 AND Region = 'South';
```

100% 16:66

Result Grid | Filter Rows: Search | Export:

count(*)
230

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



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;
```

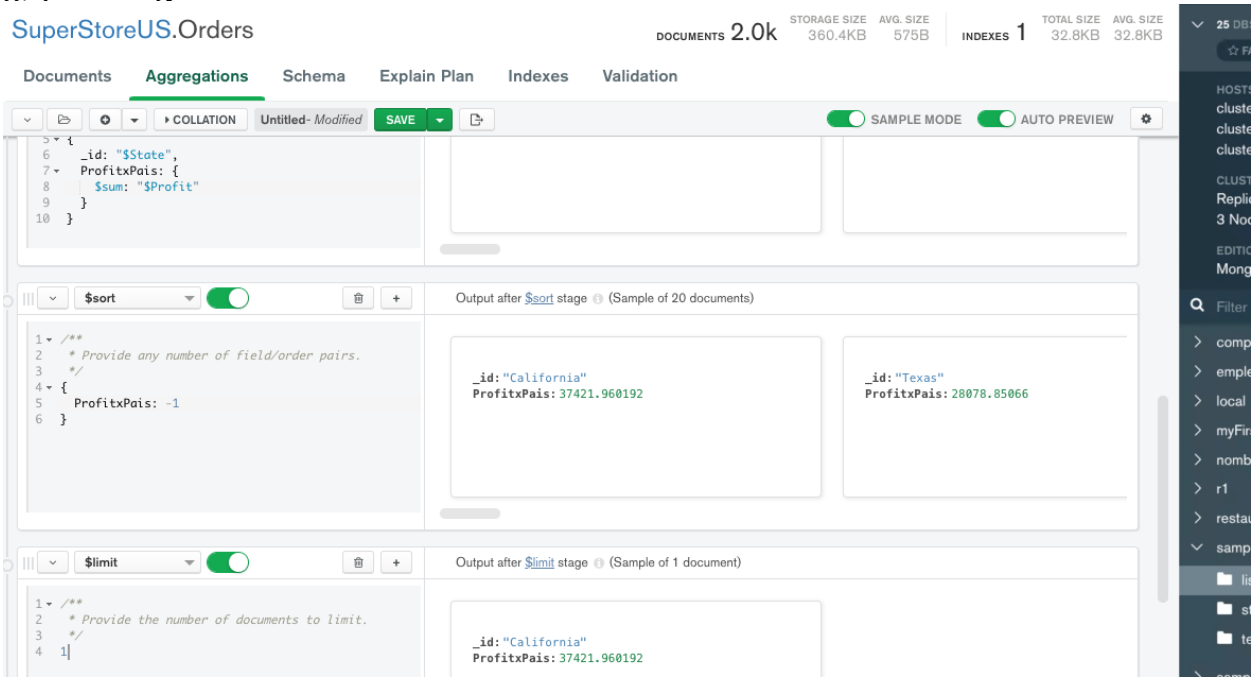
100% 30:67

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}]
```



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;
```

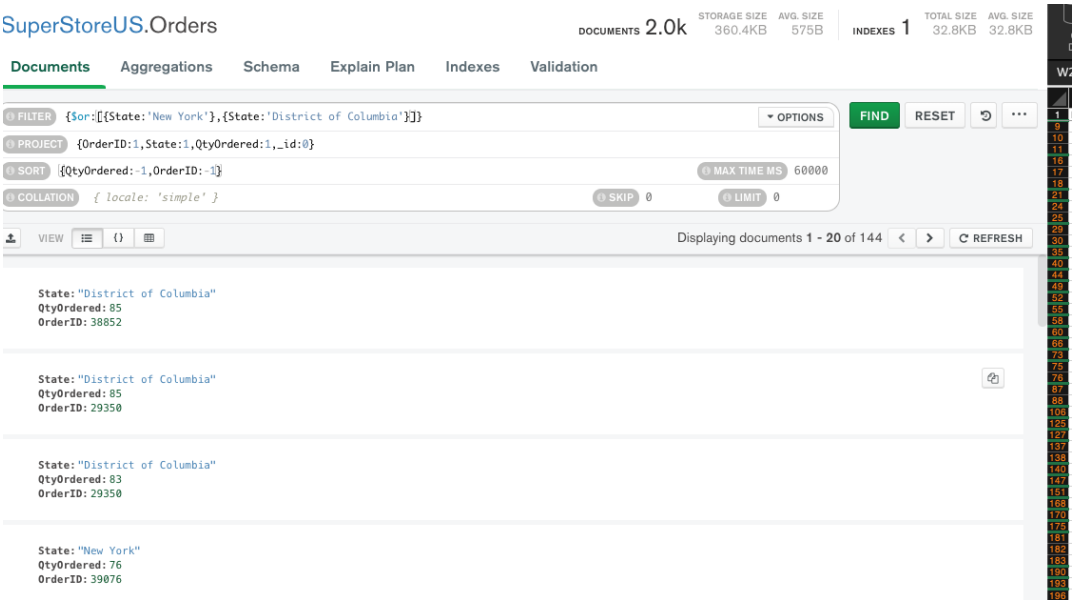
100% 39:67

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
}
```



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;
```

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

100% 22:82 1 error found

Result Grid Filter Rows: Search Export:

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

Result Grid Form Editor Field Types

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

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 Untitled- *Modified* SAVE SAMPLE MODE AUTO PREVIEW

Select an operator to construct expressions used in the aggregation pipeline stages. Learn more

```
RowID: "26267"
OrderPrio: "High"
Disc: 0.04
UnitPrice: 2.98
ShipCost: 1.58
CustID: "16"
CustName: "Sarah Ramsey"
```

```
RowID: "24848"
OrderPrio: "Medium"
Disc: 0.05
UnitPrice: 3.58
ShipCost: 1.63
CustID: "14"
CustName: "Gwendolyn F Tyson"
```

$lookup

Output after $lookup stage (Sample of 20 documents)

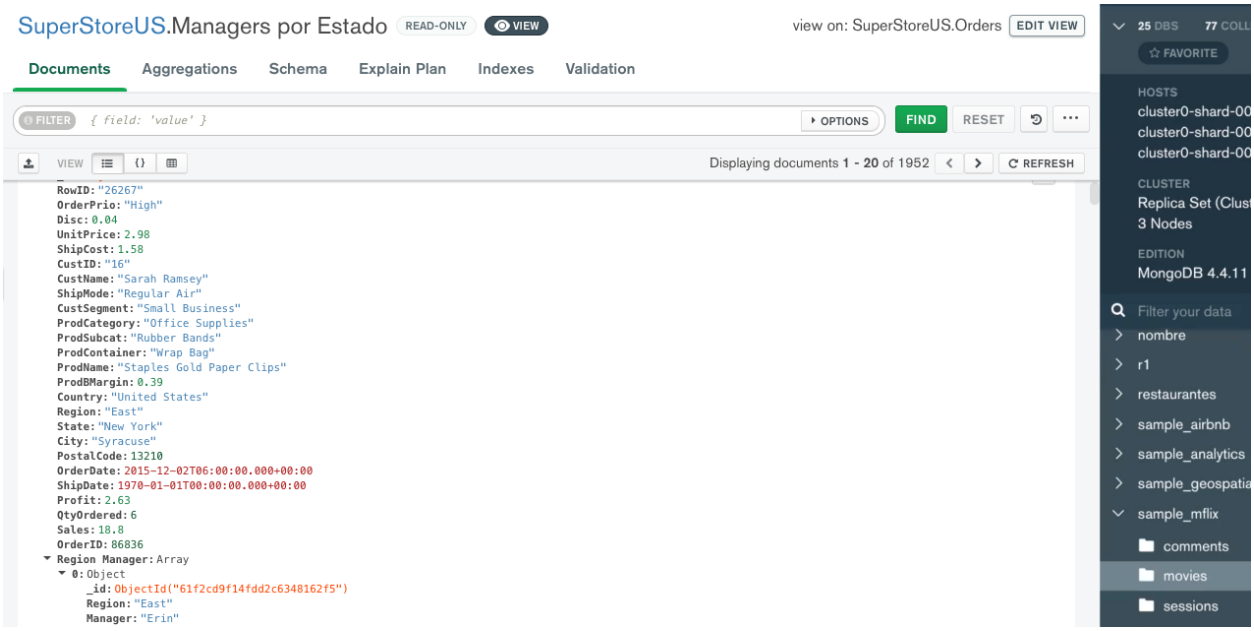
```
/**
 * from: The target collection.
 * localField: The local join field.
 * foreignField: The target join field.
 * as: The name for the results.
 * pipeline: The pipeline to run on the joined coll
 * let: Optional variables to use in the pipeline
 */
{
  from: 'Regions',
  localField: 'Region',
  foreignField: 'Region',
  as: 'ManagersRegion'
}
```

```
ShipDate: 1970-01-01T00:00:00.000+00:00
Profit: 2.63
QtyOrdered: 6
Sales: 18.8
OrderID: 86836
ManagersRegion: Array
  0: Object
    _id: ObjectId("61f2cd9f14fdd2c6348162f5")
    Region: "East"
    Manager: "Erin"
HIDE 2 FIELDS
```

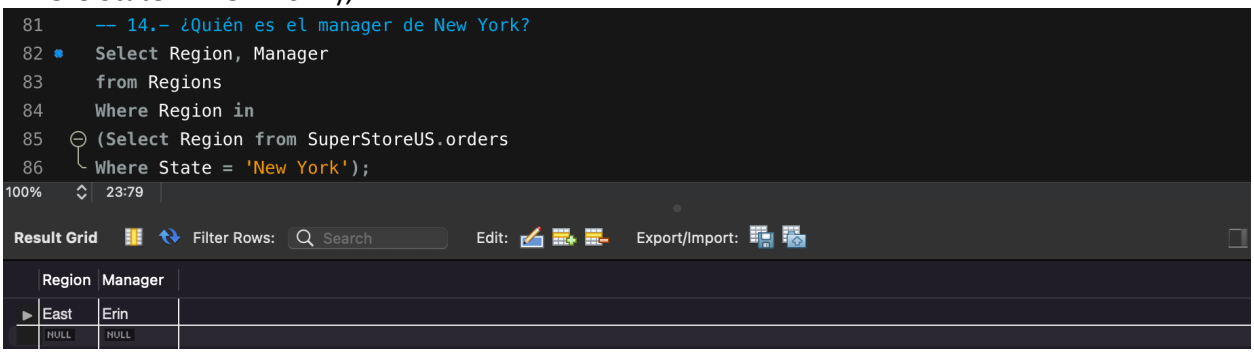
```
ShipDate: 1970-01-01T00:00:00.000+00:0
Profit: -0.71
QtyOrdered: 4
Sales: 14.26
OrderID: 86838
ManagersRegion: Array
  0: Object
    _id: ObjectId("61f2cd9f14fdd2c634
    Region: "Central"
    Manager: "Chris"
HIDE 2 FIELDS
```

Daniel Eduardo Lobo Carrillo - data-analysis-21-12

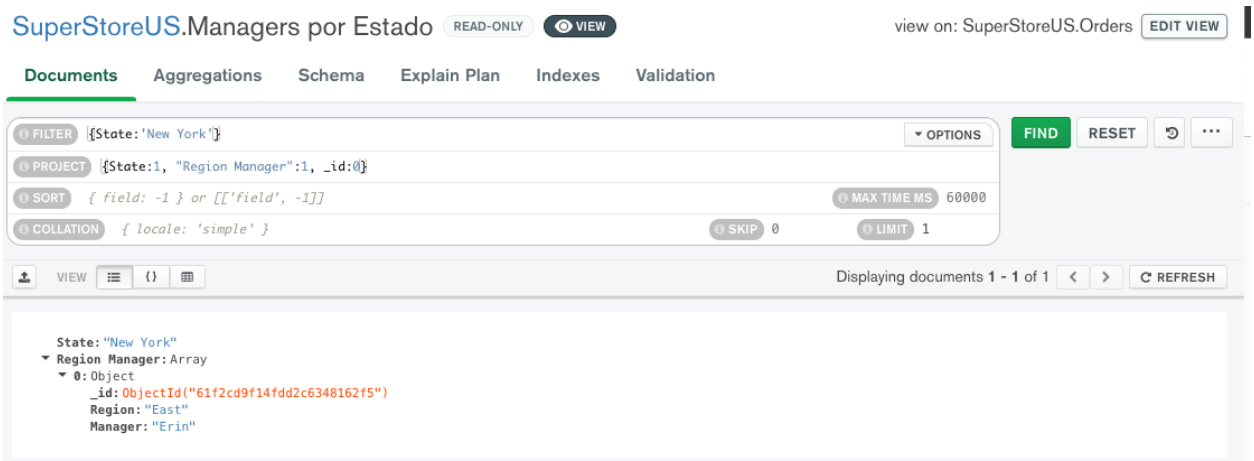
Vista:



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');



FILTER
{
State: 'New York'
}
PROJECT
{
State: 1,
'Region Manager': 1,
_id: 0
}
LIMIT
1



15.- ¿Cuánta ganancia generaron los productos Fellowes?

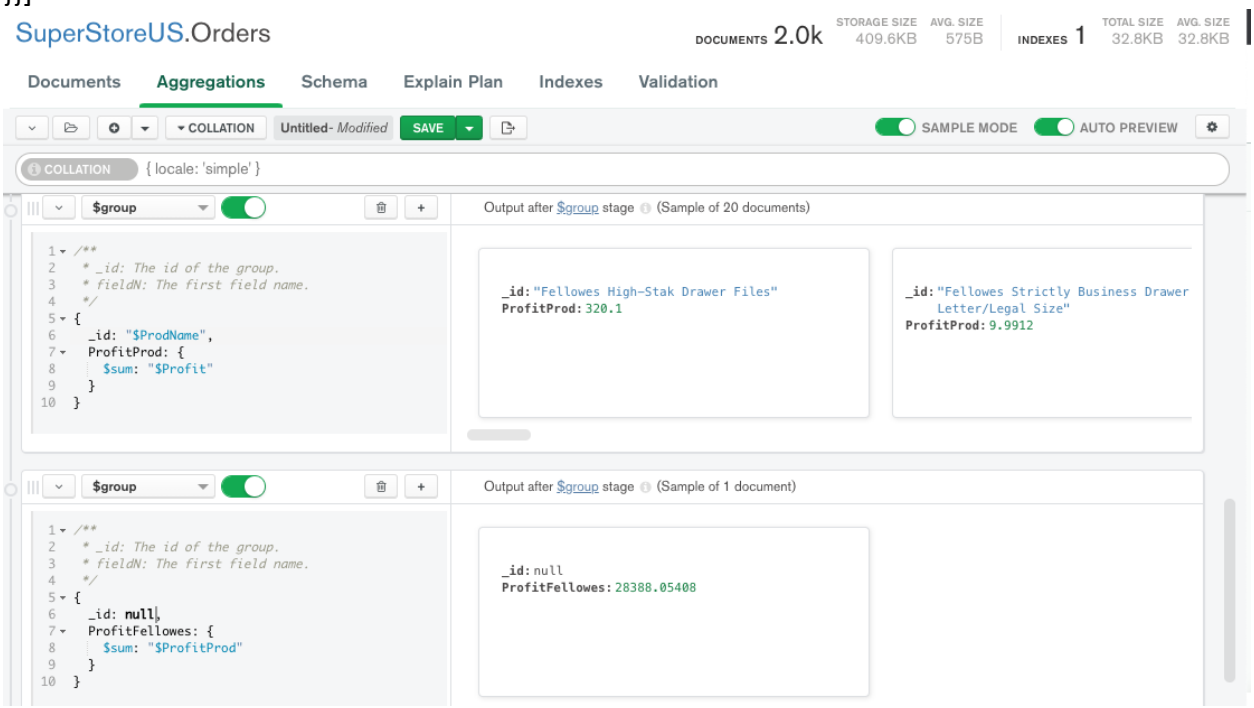
```
89    -- 15.- ¿Cuánta ganancia generaron los productos Fellowes?
90    Select Sum(Profit) as ProfitFellowes from orders
91    where ProdName like 'Fellowes%';
```

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'
 }
}}]
```



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.