



Proyecto grocery_store (1)

Definición del proyecto

Fuentes BD

Problema a resolver

Definición de la base de datos

Diagrama entidad relación

Glosario

Creación de la base de datos

Fundamentos SQL y MongoDB

1. ¿Cuáles productos se vendieron del 01 enero al 31 de enero del 2018?
2. ¿Cuáles han sido los 5 días con las mayores ventas?
3. ¿Cuáles son los productos que se vendieron de la categoría vegetales?
4. ¿Cuáles productos han vendido más de 10 unidades?
5. ¿Cuál fue el día que más impuestos se cobraron?

Agrupaciones y sub consultas SQL. Consultas MongoDB

1. ¿Cuáles presentaciones se cuenta del producto “Bambino”?
2. ¿Cuántas variedades de producto tiene cada categoría y cuántas unidades promedio tiene cada producto?
3. ¿Cuáles son los productos en presentación de 200ml?
4. ¿Cuáles son los productos hechos de maíz?
5. Presenta una tabla donde se aprecie el nombre del producto, su categoría y las unidades vendidas en total del mismo.

Join SQL. Agregaciones MongoDB

1. ¿Cuál es la venta neta, utilidad neta y utilidad sobre la venta mensual de cada categoría?
2. ¿Cuáles son las 5 categorías de productos que aportaron menor utilidad neta en el año?
3. ¿Cuál es la categoría, la utilidad y rotación de cada producto?
- 4.
- 5.

Vistas

Producto_categoria_unidadesvendidas

Utilidad_rotacion

Definición del proyecto

Fuentes BD

Los datos corresponden a una tienda de conveniencia (grocery_sotre) en Polonia tomados de:

<https://www.kaggle.com/agatii/total-sale-2018-yearly-data-of-grocery-shop>. Se conforma de tres tablas:

1. Day_sell: Información sobre la venta total por día del 2018.
2. SELL_1: Reporte mensual de la venta de productos durante 2018.
3. ROTATION: La rotación de los productos en aparador desde enero a septiembre del 2018.

Problema a resolver

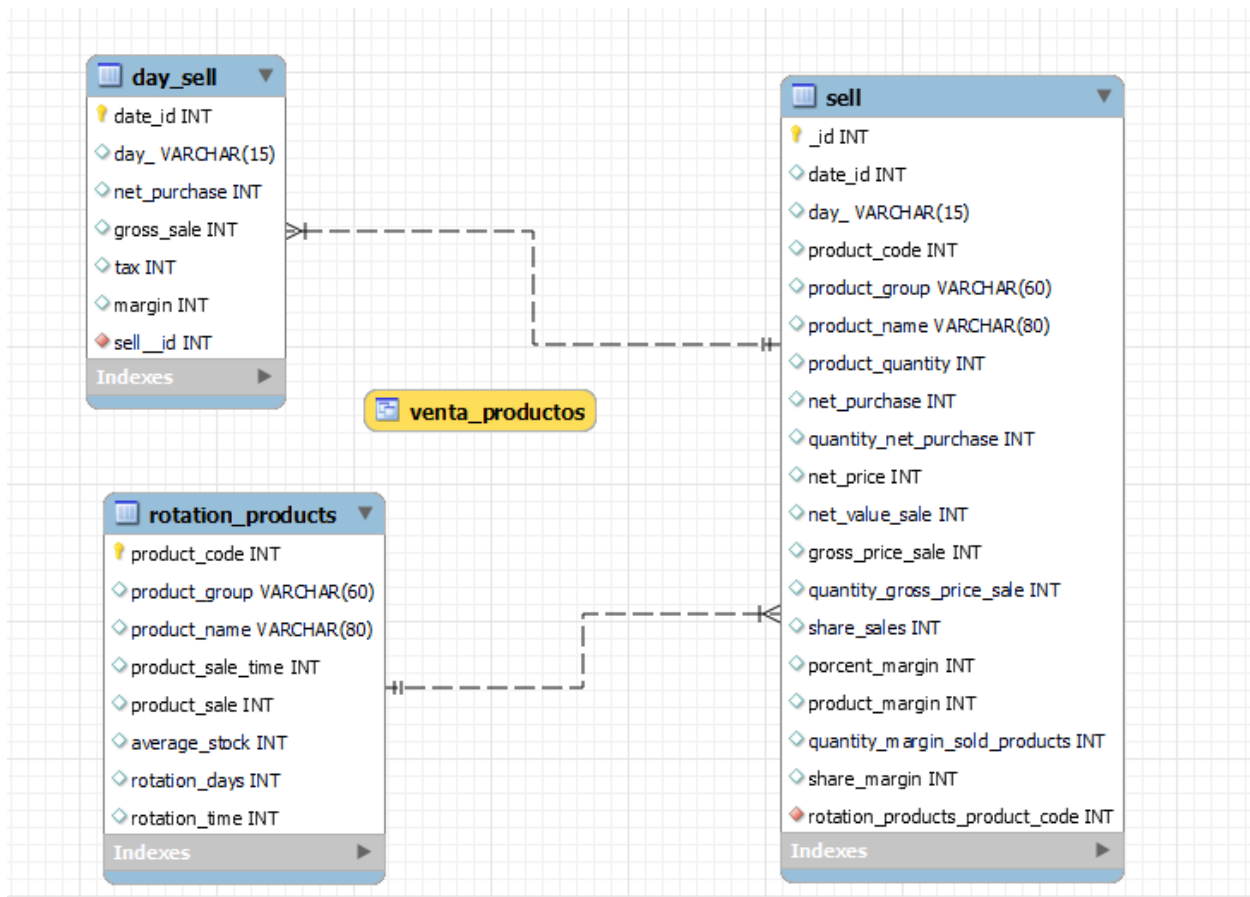
La tienda de conveniencia sufre un decremento en el ingreso del flujo de efectivo. Dicho problema se pretende resolver en el presente proyecto de dos maneras: primero al identificar y eliminar del inventario las categorías de productos que generan merma en la operación y, segundo identificar fechas de baja operación en la que se puedan ejecutar actividades promocionales para aumentar la venta promedio.

Objetivos:

1. Identificar la categoría de productos y los productos con baja rotación y con menor contribución marginal.
2. Identificar el mes con menos venta así como el comportamiento de venta de las categorías de productos dentro de dicho mes.
3. Detectar si la operación "Trade free Sundays" (3 domingos del mes las grandes cadenas no deben operar con la intención de incentivar el comercio local) ha generado beneficios.

Definición de la base de datos

Diagrama entidad relación



Glosario

Creación de la base de datos

1. SQL

La tabla “day_sell” y “sell” fueron cargadas en su totalidad, sin embargo, en la tabla “rotation_productos” se eliminaron los valores cuya venta era de cero unidades por no ser representativos para el objetivo del proyecto.

```

CREATE DATABASE IF NOT EXISTS grocery_store;
use grocery_store;
create table if not exists day_sell (
    date_id int primary key,
    day_ varchar (15),
    net_purchase int,
    gross_sale int,
    tax int,
    margin int
);

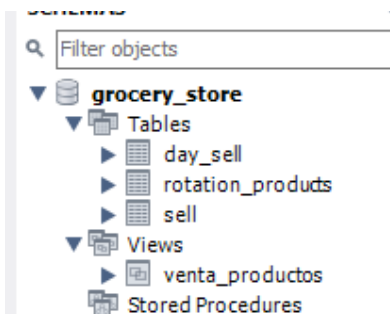
create table if not exists sell (

```

```

_id int primary key,
date_id int,
day_ varchar (15),
product_code int,
product_group varchar (60),
product_name varchar (80),
product_quantity int,
net_purchase int,
quantity_net_purchase int,
net_price int,
net_value_sale int,
gross_price_sale int,
quantity_gross_price_sale int,
share_sales int,
porcent_margin int,
product_margin int,
quantity_margin_sold_products int,
share_margin int
);
create table if not exists rotation_products (
product_code int primary key,
product_group varchar (60),
product_name varchar (80),
product_sale_time int,
product_sale int,
average_stock int,
rotation_days int,
rotation_time int
);

```



1. MongoDB

Los datos fueron cargados de manera completa en MongoDB

Collections

Create collection View [Menu Icon] [Grid Icon]

Sort by Collection Name [Dropdown Arrow]

day_sell

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
32.77 kB	361	102.00 B	1	36.86 kB

product_sell

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
2.48 MB	21 K	325.00 B	1	548.86 kB

producto_categoria_unidades [VIEW] [READ-ONLY]

View on:
product_sell

rotation_products

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
737.28 kB	16 K	193.00 B	1	368.64 kB

Fundamentos SQL y MongoDB

1. ¿Cuáles productos se vendieron del 01 enero al 31 de enero del 2018?

SQL

```
select day_, product_name from sell where date_id >=7 and date_id <=37;
```

Result Grid [Filter Rows:] [Export:] [Wrap]

	day_	product_name
▶	01/01/2018	PATYCZKI BELLA 200
	01/01/2018	ACE 1L
	01/01/2018	ALMETTE CZOSNEK NIEDZWIEDZI 150GR
	01/01/2018	ALMETTE JOGURTOWY 150G HOCHLAND
	01/01/2018	ALMETTE OGOREK/ZIOLA 150G
	01/01/2018	ALMETTE SER POMIDOR 150 G
	01/01/2018	ALMETTE SZCZYPIOREK CEBULA 150G
	01/01/2018	ALMETTE SZPINAK 150 G

sell 4 x

MongoDB

```
[{$addFields: {
  date_: {
    $dateFromString: {
      dateString: '$Date',
      format: '%d/%m/%Y'
    }
  }
}}, {$match: {
  $and: [
    {
      date_: {
        $gte: ISODate('2018-01-01T00:00:00.000Z')
      }
    },
    {
      date_: {
        $lte: ISODate('2018-01-31T00:00:00.000Z')
      }
    }
  ]
}}]
```

grocery_store.product_sell

DOCUMENTS 21.3k STORAGE SIZE 2.5MB AVG. SIZE 325B INDEXES 1 TOTAL SIZE 548.9KB AVG. 548.9

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled - Modified SAVE SAMPLE MODE AUTO PREVIEW

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

```
1 /**
2  * newField: The new field name.
3  * expression: The new field expression.
4  */
5 {
6   date_: {
7     $dateFromString: {
8       dateString: '$Date',
9       format: "%d/%m/%Y"
10    }
11  }
12 }
```

Sample document output:

```
{
  pwa_sn: 2.5,
  pce_sb: 2.78,
  pwa_sb: 2.78,
  pudzsb: 0,
  pmarza: 31.13,
  pmarzajedn: 0.8,
  pkwmarza: 0.8,
  pudzmarza: 0.01,
  date_: 2018-01-01T00:00:00.000+00:00,
  _id: ObjectId("61ee031e8281e78e240f5cdb"),
  id_Date: 7,
  Date: "01/01/2018",
  PKod: 5986,
  Pgroup: "CHEMISTRY",
  Pname: "ACE 1 L",
  Pquantity: 1,
  pce_zn: 3.2
}
```

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

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   $and: [
6     {date_: {$gte: new Date("2018-01-01")}},
7     {date_: {$lte: new Date("2018-01-31")}}
8   ]
9 }
```

Sample document output:

```
{
  _id: ObjectId("61ee031e8281e78e240f5cda"),
  id_Date: 7,
  Date: "01/01/2018",
  PKod: 5307,
  Pgroup: "CHEMISTRY",
  Pname: "PATYCZKI BELLA 200",
  Pquantity: 1,
  pce_zn: 1.77,
  pwa_sn: 5.68,
  pce_sb: 6.99,
  pwa_sb: 6.99,
  pudzsb: 0.01,
  Pname: "ACE 1 L",
  Pquantity: 1,
  pce_zn: 3.2
}
```

2. ¿Cuáles han sido los 5 días con las mayores ventas?

SQL

```
select day_, gross_sale from day_sell  
order by gross_sale desc limit 5;
```

Result Grid			Filter Rows:
	day_	gross_sale	
▶	22/12/2018	6427	
	24/12/2018	5519	
	26/04/2018	5400	
	26/05/2018	5377	
	10/11/2018	5364	

MongoDB

```
[{$sort: {  
  sb: -1  
}}, {$limit: 5}]
```

grocery_store.day_sell

DOCUMENTS 361 STORAGE SIZE 32.8KB AVG. SIZE 102B INDEXES 1 TOTAL SIZE 20.5KB AVG. SIZE 20.5KB

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE SAMPLE MODE AUTO PREVIEW

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

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

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

```
{ "_id": ObjectId("61ee02028281e78e240eb66e"), "ID": 359, "Date": "22/12/2018", "zn": 4416.64, "sb": 6426.81, "tax": 734.23, "marza": 1275.94 }
```

```
{ "_id": ObjectId("61ee02028281e78e240eb670"), "ID": 361, "Date": "24/12/2018", "zn": 3887.9, "sb": 5518.87, "tax": 781.45, "marza": 1009.52 }
```

\$limit Output after \$limit stage (Sample of 5 documents)

```
1 /**
2  * Provide the number of documents to limit.
3  */
4 5
```

Output after \$limit stage (Sample of 5 documents)

```
{ "_id": ObjectId("61ee02028281e78e240eb66e"), "ID": 359, "Date": "22/12/2018", "zn": 4416.64, "sb": 6426.81, "tax": 734.23, "marza": 1275.94 }
```

```
{ "_id": ObjectId("61ee02028281e78e240eb670"), "ID": 361, "Date": "24/12/2018", "zn": 3887.9, "sb": 5518.87, "tax": 781.45, "marza": 1009.52 }
```

3. ¿Cuáles son los productos que se vendieron de la categoría vegetales?

SQL

```
select product_group, product_name from sell where product_group = "vegetables";
```

47 • `select product_group, product_name from sell where product_group = "vegetables";`

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	product_group	product_name
▶	VEGETABLES	AMINO BLYSK ZUREK
	VEGETABLES	DR POLEWA MLECZNA TWARDA
	VEGETABLES	KT ANANAS
	VEGETABLES	KT BANAN
	VEGETABLES	KT BROKUL
	VEGETABLES	KT BURACZEK CZERWONY
	VEGETABLES	KT CEBUL CZERWONNA MODRA
	VEGETABLES	KT CEBUL BIAŁA

sell 7

MongoDB


```
[{$match: {
  Pgroup: 'VEGETABLES'
}}]
```

grocery_store.product_sell

DOCUMENTS 21.3k STORAGE SIZE 2.5MB AVG. SIZE 325B INDEXES 1 TOTAL SIZE 548.9KB AVG. 548.9

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("61ee031e8281e78e240f5cda")
 id_Date: 7
 Date: "01/01/2018"
 PKod: 5307
 Pgroup: "CHEMISTRY"
 Pname: "PATYCZKI BELLA 200"
 Pquantity: 1
 pce_zn: 1.77

_id: ObjectId("61ee031e8281e78e240f5cdb")
 id_Date: 7
 Date: "01/01/2018"
 PKod: 5986
 Pgroup: "CHEMISTRY"
 Pname: "ACE 1 L"
 Pquantity: 1
 pce_zn: 3.2

\$match

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

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   Pgroup: "VEGETABLES"
6 }
```

_id: ObjectId("61ee031e8281e78e240f5ce5")
 id_Date: 7
 Date: "01/01/2018"
 PKod: 8404
 Pgroup: "VEGETABLES"
 Pname: "AMINO BLYSK ZUREK"
 Pquantity: 6
 pce_zn: 0.82

_id: ObjectId("61ee031e8281e78e240f5de1")
 id_Date: 7
 Date: "01/01/2018"
 PKod: 6483
 Pgroup: "VEGETABLES"
 Pname: "DR POLENA MLECZNA TWARDA"
 Pquantity: 1
 pce_zn: 1.94

4. ¿Cuáles productos han vendido más de 10 unidades?

SQL

```
select product_name, product_sale from rotation_products where product_sale >10;
```

```
61 • select product_name, product_sale from rotation_products where product_sale >10;
62
63
```

Result Grid		
Filter Rows:	Export:	Wrap Cell Content:
product_name	product_sale	
BELLA PODPASKI NOVA MAXI 10	18	
BELLA PODPASKI NORMAL MAXI 10	24	
RECZNIK PAPIER SZAST PRAST	11	
NIVEA KREM 50ML	38	
WKŁADKI BELLA PANTY 20SZT	17	
WKŁADKI BELLA PANTY	20	
PLATKI BELLA 80+30%	16	
ODPIAMTACZ VANTSH 100ML PZYN SĄS7FTKA	22	

rotation_products 15 x

MongoDB

```
[{$match: {
  Psale: {
    $gt: 10
  }
}}]
```

grocery_store.rotation_products

DOCUMENTS 16.3k STORAGE SIZE 340.0KB AVG. SIZE 193B INDEXES 1 TOTAL SIZE 172.0KB AVG. S 172.0

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE SAMPLE MODE AUTO PREVIEW

16293 Documents in the Collection

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

Document 1:

```
{
  "_id": "61ee036f8281e78e240fb072",
  "Pgroup": "ART._HYGIENIC",
  "Lp": 1,
  "PKod": 156,
  "Pname": "WATA 100G BELLA",
  "p_sale_in_time": 0.021390374,
  "Psale": 8,
  "Pavarage_stock": 0,
  "Rotation_in_days": 0
}
```

Document 2:

```
{
  "_id": "61ee036f8281e78e240fb073",
  "Pgroup": "ART._HYGIENIC",
  "Lp": 1,
  "PKod": 155,
  "Pname": "WATA BAWELNIANA BELLA 100G",
  "p_sale_in_time": 0.005347594,
  "Psale": 2,
  "Pavarage_stock": 0,
  "Rotation_in_days": 0
}
```

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

Document 1:

```
{
  "_id": "61ee036f8281e78e240fb072",
  "Pgroup": "ART._HYGIENIC",
  "Lp": 1,
  "PKod": 156,
  "Pname": "WATA 100G BELLA",
  "p_sale_in_time": 0.021390374,
  "Psale": 8,
  "Pavarage_stock": 0,
  "Rotation_in_days": 0
}
```

Document 2:






```
{
  "_id": "61ee036f8281e78e240fb073",
  "Pgroup": "ART._HYGIENIC",
  "Lp": 1,
  "PKod": 155,
  "Pname": "WATA BAWELNIANA BELLA 100G",
  "p_sale_in_time": 0.005347594,
  "Psale": 2,
  "Pavarage_stock": 0,
  "Rotation_in_days": 0
}
```

5. ¿Cuál fue el día que más impuestos se cobraron?

SQL

```
select day_, tax from day_sell order by tax desc limit 1;
```

```
62 • select day_, tax from day_sell order by tax desc limit 1;
```

Result Grid |   Filter Rows: | Export:  | Wrap Cell Content:  | Fetch rows: 

day_	tax
01/09/2018	737

MongoDB

```
[{$sort: {  
  tax: -1  
}}, {$limit: 1}]
```

grocery_store.day_sell DOCUMENTS 361

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE

```

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

```

```

_id: ObjectId("61ee02028281e78e240eb5ff")
ID: 248
Date: "01/09/2018"
zn: 3634.82
sb: 5325.8
tax: 736.87
marza: 954.11

```

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

```

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

```

```

_id: ObjectId("61ee02028281e78e240eb5ff")
ID: 248
Date: "01/09/2018"
zn: 3634.82
sb: 5325.8
tax: 736.87
marza: 954.11

```

Agrupaciones y sub consultas SQL. Consultas MongoDB

1. ¿Cuáles presentaciones se cuenta del producto “Bambino”?

SQL

```
select product_name from sell where product_name like "Bambino%";
```

```
63
64 • select product_name from sell where product_name like "Bambino%";
65
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_name
BAMBINO OLIWKA DLA DZIECI 150G
BAMBINO SZAMPON 400 ML
BAMBINO SZAMPON 400 ML
BAMBINO OLIWKA DLA DZIECI 150G
BAMBINO OLIWKA DLA DZIECI 150G
BAMBINO SZAMPON 400 ML
BAMBINO CHUSTECZKI NASACZ/ 63 SZT
BAMBINO SZAMPON 400 ML

MongoDB

```
{
  filter: {
    pname: RegExp('^BAMBINO')
  }
}
```

groce....rotation_pr... DOCUMENTS 16.3k STORAGE 340.

Documents Aggregations Schema Explain Plan

FILTER {Pname:/^BAMBINO/}

ADD DATA **VIEW** Display

```

_id: ObjectId("61ee036f8281e78e240fb00d")
Pgroup: "ART._HYGIENIC"
Lp: 1
PKod: 8478
Pname: "BAMBINO SZAMPON 400 ML"
p_sale_in_time: 0.013368984
Psale: 5
Pavarage_stock: 0
Rotation_in_days: 0

_id: ObjectId("61ee036f8281e78e240fb024")
Pgroup: "ART._HYGIENIC"
Lp: 1
PKod: 10138
Pname: "BAMBINO CHUSTECZKI NAS?CZ.63"
p_sale_in_time: 0
Psale: 0
Pavarage_stock: 0
Rotation_in_days: NaN
Rotation in times: NaN

```

2. ¿Cuántas variedades de producto tiene cada categoría y cuántas unidades promedio tiene cada producto?

SQL

```

select product_code,
count(product_name) as total_products,
sum(product_quantity) as products_quantity,
(sum(product_quantity) / count(product_name)) as avg_products
from sell group by product_code;

```

```

66 • select product
67 count(product_name) as total_products,
68 sum(product_quantity) as products_quantity,
69 (sum(product_quantity) / count(product_name)) as avg_products
70 from sell group by product_code;

```

result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

product_code	total_products	products_quantity	avg_products
5307	7	16	2.2857
5986	3	4	1.3333
399	6	22	3.6667
2246	7	27	3.8571
2247	9	38	4.2222
1305	6	22	3.6667
1142	7	22	3.1429
17316	8	33	4.1250
3818	4	15	3.7500

result 27 x

Output:

1 Action Output

MongoDB

```

[{$group: {
  _id: '$Pgroup',
  Variedades: {
    $sum: 1
  },
  unidades: {
    $sum: '$Pquantity'
  }
}}, {$addFields: {
  Promedio_unidades: {
    $divide: [
      '$unidades',
      '$Variedades'
    ]
  }
}}]

```

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION

Untitled- Modified

SAVE

```
3  * fieldN: The first field name.
4  */
5  {
6    _id: "$Pgroup",
7    Variedades: {
8      $sum: 1
9    },
10   unidades: {
11     $sum: "$Pquantity"
12   }
13 }
```

_id: "DRINK_JUICE"

Variedades: 1856

unidades: 25456

\$addFields

Output after \$addFields stage (Sam

```
1  /**
2   * newField: The new field name.
3   * expression: The new field expression.
4   */
5  {
6    Promedio_unidades: {$divide:[
7      "$unidades", "$Variedades "
8    ]}
9  }
```

_id: "CANNED_PATE"

Variedades: 336

unidades: 2470.723

Promedio_unidades: null

3. ¿Cuáles son los productos en presentación de 200ml?

SQL




```
select product_name from sell where product_name like "%200 ML%";
```



```

71
72 • select product_name from sell where product_name like "%200 ML%";
73

```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	product_name
▶	LODY COFFE LATTE 200 ML
	LODY ICE FRESH 200 ML
	NAP NAPI MULTIWITAMINA 200 ML
	OLEJ LNIANY 200 ML
	LODY COFFE LATTE 200 ML
	NAP NAPI MULTIWITAMINA 200 ML
	NAP TYMB/MULTIW/200 ML
	OLEJ KOKOSOWY 200 ML

MongoDB

```

{
  filter: {
    Pname: RegExp('.*200 ML')
  }
}

```

grocery_store.product_sell DOC

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER {Pname:/. *200 ML/}

ADD DATA **VIEW**

```

_id: ObjectId("61ee031e8281e78e240f5fcd")
id_Date: 7
Date: "01/01/2018"
PKod: 3006
Pgroup: "ICE_CREAMS_FROZEN"
Pname: "LODY COFFE LATTE 200 ML"
Pquantity: 4
pce_zn: 2.31
pwa_zn: 9.24
pce_sn: 3.23
pwa_sn: 12.92
pce_sb: 3.39
pwa_sb: 13.56
pudzs: 0.02
pmarza: 28.48
pmarzajedn: 0.92
pkwmarza: 3.68
pudzmarza: 0.04

```

4. ¿Cuáles son los productos hechos de maíz?

SQL

```
select product_name from rotation_products where product_name like "%CORN%";
```

```

71
72 • select product_name from rotation_products where product_name like "%CORN%";
73

```

Result Grid Filter Rows: | Export: | Wrap Cell Content:

product_name
POP CORN 95 G

MongoDB

```
{
  filter: {
    Pname: RegExp('.*CORN.*')
  }
}
```

The screenshot shows the MongoDB Compass interface. At the top, there are tabs for Documents, Aggregations, Schema, Explain Plan, Indexes, and Validation. Below the tabs, there is a filter bar with a filter query: `{Pname: /.*/CORN.*/}`. To the right of the filter bar are buttons for FILTER, OPTIONS, FIND, RESET, and a refresh icon. Below the filter bar, there is a section for ADD DATA and a VIEW button. The main area displays a document with the following fields: `_id` (ObjectId), `Pgroup` (CHEMISTRY), `Lp` (1), `PKod` (5349), `Pname` (POP CORN 95 G), `p_sale_in_time` (0.254010695), `Psale` (95), `Pavarage_stock` (1), `Rotation_in_days` (3.936842105), and `Rotation_in_tim` (95).

5. Presenta una tabla donde se aprecie el nombre del producto, su categoría y las unidades vendidas en total del mismo.

SQL

```
select product_name, product_group, sum(product_quantity) from sell group by product_name;
```

The screenshot shows a SQL query result in a database client. The query is: `select product_name, product_group, sum(product_quantity) from sell group by product_name;`. The result is displayed in a table with three columns: `product_name`, `product_group`, and `sum(product_quantity)`. The table contains the following data:

product_name	product_group	sum(product_quantity)
PATYCZKI BELLA 200	CHEMISTRY	16
ACE 1 L	CHEMISTRY	4
ALMETTE CZOSNEK NIEDZWIEDZI 150GR	DAIRY_CHESE	22
ALMETTE JOGURTOWY 150G HOCHLAND	DAIRY_CHESE	27
ALMETTE OGOREK/ZIOLA 150G	DAIRY_CHESE	38
ALMETTE SER POMIDOR 150 G	DAIRY_CHESE	22
ALMETTE SZCZYPIOREK CEBULA 150G	DAIRY_CHESE	22

MongoDB

```
[{$group: {
  _id: {
    categoria: '$Pgroup',
    producto: '$Pname'
  },
  Unidades_vendidas: {
    $sum: '$Pquantity'
  }
}}, {$addFields: {
  Categoria: '$_id.categoria',
  producto: '$_id.producto'
}}, {$project: {
  _id: 0,
  Unidades_vendidas: 1,
  Categoria: 1,
  producto: 1
}}]
```

grocery_store.producto_categoria_unidades

READ-ONLY

VIEW

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

FILTER { field: 'value' }

VIEW

producto_categoria_unidades

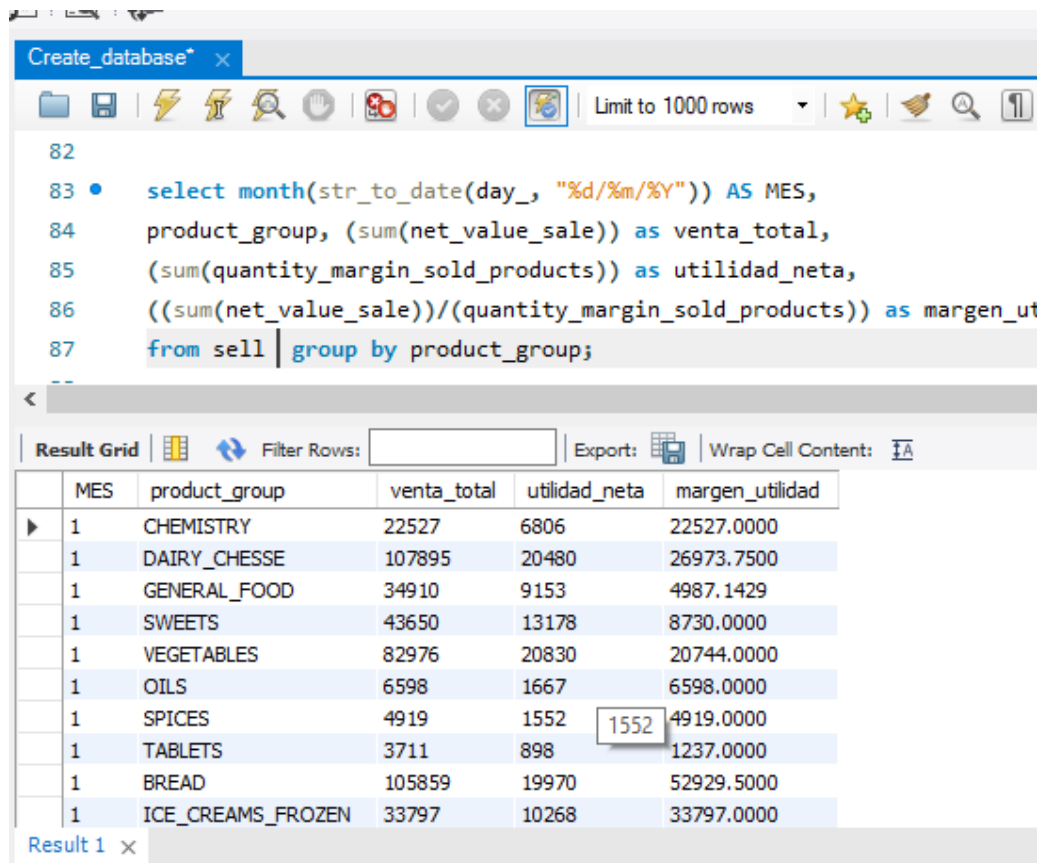
	producto String	Unidades_vendidas Int32	Categoria String
1	"WEDEL CZEKOLADA NADZ TOFFI 100"	5	"SWEETS"
2	"ZNICZ DUZY"	2	"GENERAL"
3	"BREF KULKI WC 50 G"	2	"CHEMISTRY"
4	"MUSZTARDA MIODOWA 270 G"	8	"DRINK_JUICE"
5	"PAP LUCKY STRIKE 22 RED"	69	"CIGARETTES"
6	"SOS BOLONSKI SLOIK DAWT 550G"	6	"GENERAL_FOOD"
7	"ZUREK BUTELKA 0_5 L"	7	"GENERAL"
8	"BELLA CHUSTECZKI ODSWIEZAJACE"	7	"CHEMISTRY"
--	--	--	--

Join SQL. Agregaciones MongoDB

1. ¿Cuál es la venta neta, utilidad neta y utilidad sobre la venta mensual de cada categoría?

SQL

```
select month(str_to_date(day_, "%d/%m/%Y")) AS MES,  
product_group, (sum(net_value_sale)) as venta_total,  
(sum(quantity_margin_sold_products)) as utilidad_neta,  
((sum(net_value_sale))/(quantity_margin_sold_products)) as margen_utilidad  
from sell group by product_group;
```



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
82  
83 • select month(str_to_date(day_, "%d/%m/%Y")) AS MES,  
84 product_group, (sum(net_value_sale)) as venta_total,  
85 (sum(quantity_margin_sold_products)) as utilidad_neta,  
86 ((sum(net_value_sale))/(quantity_margin_sold_products)) as margen_ut  
87 from sell | group by product_group;
```

Below the query, the results are displayed in a table with the following columns: MES, product_group, venta_total, utilidad_neta, and margen_utilidad. The table contains 10 rows of data. A tooltip is visible over the 'utilidad_neta' value of 1552 for the 'SPICES' category.

	MES	product_group	venta_total	utilidad_neta	margen_utilidad
▶	1	CHEMISTRY	22527	6806	22527.0000
	1	DAIRY_CHESSE	107895	20480	26973.7500
	1	GENERAL_FOOD	34910	9153	4987.1429
	1	SWEETS	43650	13178	8730.0000
	1	VEGETABLES	82976	20830	20744.0000
	1	OILS	6598	1667	6598.0000
	1	SPICES	4919	1552	4919.0000
	1	TABLETS	3711	898	1237.0000
	1	BREAD	105859	19970	52929.5000
	1	ICE_CREAMS_FROZEN	33797	10268	33797.0000

MongoDB

```

[{$addFields: {
  date_: {
    $dateFromString: {
      dateString: '$Date',
      format: '%d/%m/%Y'
    }
  }
}}, {$addFields: {
  mes: {
    $month: '$date_'
  }
}}, {$group: {
  _id: {
    categoria: '$Pgroup',
    mes: '$mes'
  },
  venta_neta: {
    $sum: '$pwa_sn'
  },
  utilidad_neta: {
    $sum: '$pkwmarza'
  },
  utilidad_venta: {
    $avg: '$pmarza'
  }
}}, {$addFields: {
  categoria: '$_id.categoria',
  mes: '$_id.mes'
}}, {$project: {
  _id: 0,
  venta_neta: 1,
  utilidad_neta: 1,
  utilidad_venta: 1,
  categoria: 1,
  mes: 1
}}, {$sort: {
  mes: 1
}}]

```

grocery_store.utilidad_... Documents

grocery_store.utilidad_categoria_mes READ-ONLY VIEW view on: grocery_store.product_sell

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET

VIEW { } Displaying documents 1 - 20 of 382

	utilidad_categoria_mes			
	categoria String	mes Int32	venta_neta Double	utilidad_neta Double
1	"ARTICLE_OF_HOUSEHOLD"	1	2.96	0.97
2	"VODKA_ALCOHOL"	1	6753.45	1455.5
3	"ICE_CREAMS_FROZEN"	2	1031.28	274.44
4	"SPICES"	3	495.9	151.54
5	"OILS"	7	735.79	198.05
6	"DRINK_JUICE"	9	6033.12	1946.71
7	"VEGETABLES"	9	7308.4	1243.1299999999999

2. ¿Cuáles son las 5 categorías de productos que aportaron menor utilidad neta en el año?

SQL

```
select month(str_to_date(day_, "%d/%m/%Y")) AS MES,
product_group,
(sum(net_value_sale)) as venta_total,
(sum(IFNULL(quantity_margin_sold_products,0))) as utilidad_neta,
(sum(ifnull(net_value_sale,0)))/(sum(IFNULL(quantity_margin_sold_products,0))) as margen_utilidad
from sell
group by product_group order by utilidad_neta ASC limit 5;
```

Create_database* x

Limit to 1000 rows

```

83 • select month(str_to_date(day_, "%d/%m/%Y")) AS MES,
84     product_group, (sum(net_value_sale)) as venta_total,
85     (sum(IFNULL(quantity_margin_sold_products,0))) as utilidad_neta,
86     (sum(ifnull(net_value_sale,0)))/(sum(IFNULL(quantity_margin_sold_products,0))) as margen_utilidad
87 from sell
88 group by product_group order by utilidad_neta ASC limit 5;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

	MES	product_group	venta_total	utilidad_neta	margen_utilidad
2		MILK	17	4	4.2500
1		ARTICLE_OF_HOUSEHOLD	50	15	3.3333
1		PACKAGES	218	16	13.6250
1		LUNCH DINING DISHES	382	110	3.4727
1		DISHES_FOR_CHILDREN	820	225	3.6444

MongoDB

```

[{$addFields: {
  date_: {
    $dateFromString: {
      dateString: '$Date',
      format: '%d/%m/%Y'
    }
  }
}}, {$addFields: {
  mes: {
    $month: '$date_'
  }
}}, {$group: {
  _id: {
    categoria: '$Pgroup'
  },
  venta_neta: {
    $sum: '$pwa_sn'
  },
  utilidad_neta: {
    $sum: '$pkwmarza'
  },
  utilidad_venta: {
    $avg: '$pmarza'
  }
}}, {$addFields: {
  categoria: '$_id.categoria'
}}, {$project: {
  _id: 0,
  venta_neta: 1,
  utilidad_neta: 1,
  utilidad_venta: 1,
  categoria: 1

```



```
}},{$sort: {
  utilidad_neta: 1
}}, {$limit: 5}]
```

grocery_store.5 productos poca utilidad READ-ONLY VIEW view on: grocery_store.utilidad_catego

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RES

VIEW {}

Displaying documents 1 - 5 of 5

	categoria String	venta_neta Double	utilidad_neta Double	utilidad_venta Double
1	"MILK"	16.5	4.38	26.55
2	"ARTICLE_OF_HOUSEHOLD"	48.67	15.860000000000001	32.96095238095238
3	"PACKAGES"	219.10999999999999	16.85	10.606111111111112
4	"LUNCH DINING DISHES"	381.78999999999996	111.14	29.20151515151515
5	"DISHES_FOR_CHILDREN"	821.47	225.51	26.823529411764707

3. ¿Cuál es la categoría, la utilidad y rotación de cada producto?

SQL

```
select s.product_name, s.product_group, quantity_margin_sold_products, rotation_time
from sell as s
join rotation_products as r
on s.product_code = r.product_code;
```

Create_database* x

Limit to 1000 rows

```

90 • select s.product_name, s.product_group, quantity_margin_sold_products, rotation
91      from sell as s
92      join rotation_products as r
93      on s.product_code = r.product_code;
94
95

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	product_name	product_group	quantity_margin_sold_products	rotation_time
▶	ACE 1 L	CHEMISTRY	2	2
	BELLA HERBS 12 Z WERBENA	ART/_HYGIENIC	1	4
	BELLA PODPASKI NORMAL MAXI 10	ART/_HYGIENIC	4	24
	BELLA PODPASKI NOVA MAXI 10	ART/_HYGIENIC	-1	3
	BREF 4-FORMULA LEMON	CHEMISTRY	13	6
	BREF KULKI BLU AKTIV	CHEMISTRY	1	0
	CHUSTECZKI DO KOLOROW 10 SZT	CHEMISTRY	3	1
	CLIN ZAPAS 500 ML	CHEMISTRY	2	20
	CUKIERKI KROWKA WAWEL/SOLIDARNOSC	COOKIES_BULK	2	1
	CUKIERKI TRUFLE ODRA LUZ	COOKIES_BULK	5	13

MongoDB

```

[{$addFields: {
  date_: {
    $dateFromString: {
      dateString: '$Date',
      format: '%d/%m/%Y'
    }
  }
}}, {$lookup: {
  from: 'rotation_products',
  localField: 'PKod',
  foreignField: 'PKod',
  as: 'rotation_p'
}}, {$addFields: {
  rotation_p: {
    $reduce: {
      input: '$rotation_p',
      initialValue: {
        $arrayElemAt: [
          '$rotation_p',
          0
        ]
      },
      'in': {
        $cond: {

```

```

    'if': {
      $gt: [
        '$$this.Pgroup',
        '$$value.Pgroup'
      ]
    },
    then: '$$this',
    else: '$$value'
  }
}
}
}
}}, {$addFields: {
  rotation_product: '$rotation_p.Rotation_in_days'
}}, {$project: {
  _id: 0,
  Pname: 1,
  Pgroup: 1,
  pmarzajedn: 1,
  rotation_product: 1
}}]

```

grocery_store.product_sell DOCUMENTS 21.3k STORAGE SIZE 2.7MB AVG. 3'

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION rotation_productsell_utilidad SAVE

21

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

```

1 /**
2  * newField: The new field name.
3  * expression: The new field expression.
4  */
5 {
6   rotation_product: "$rotation_p.Rotation_in_days"
7 }

```

```

pwa_sd: 15.96
pudzsb: 0.02
pmarza: 29.47
pmarzajedn: 1.12
pkwmarza: 4.48
pudzmarza: 0.05
date: 2018-01-01T00:00:00.000+00:00
rotation_p: Object
  rotation_product: 0
_id: 0
id_Da:
Date:
PKod:
Pgroup:
Pname:
Pquan:
pce_zi:

```

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

```

1 /**
2  * specifications: The fields to
3  * include or exclude.
4  */
5 {
6   _id: 0,
7   Pname: 1,
8   Pgroup: 1,
9   pmarzajedn: 1,
10  rotation_product: 1

```

```

Pgroup: "DAIRY_CHESSE"
Pname: "ALMETTE CZOSNEK NIEDZWIEDZI 150GR"
pmarzajedn: 1.12
rotation_product: 0
Pgroup:
Pname:
pmarz:
rotat:

```

4.

SQL

MongoDB

5.

SQL

MongoDB

Vistas

Producto_categoria_unidadesvendidas

SQL

Navigator

SCHEMAS

Filter objects

grocery_store

Tables

day_sell

rotation_products

sell

Views

venta_productos

product_name

product_group

sum(product_quantity)

Administration

Schemas

Information

View: venta_productos

Columns:

product_name varchar

product_group varchar

sum(product_quantity) decimal

Create_database

Limit to 1000 rows

```

64 • select product_name from sell where product_name like "Bambino%";
65 • select product_code from sell;
66 • select product_code,
67     count(product_name) as total_products,
68     sum(product_quantity) as products_quantity,
69     (sum(product_quantity) / count(product_name)) as avg_products
70     from sell group by product_code;
71
72 • select product_name from rotation_products where product_name like "%CORN%";
73
74 • select product_name, product_group, sum(product_quantity) from sell group by product_name;
75
76 • create view venta_productos as
77     select product_name, product_group, sum(product_quantity) from sell group by product_name;
78
79

```

MongoDB

grocery_store.producto_categoria_unidades READ-ONLY VIEW

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

VIEW

producto_categoria_unidades

	producto String	Unidades_vendidas Int32	Categoria String
1	"WEDEL CZEKOLADA NADZ TOFFI 100	5	"SWEETS"
2	"ZNICZ DUZY"	2	"GENERAL"
3	"BREF KULKI WC 50 G"	2	"CHEMISTRY"
4	"MUSZTARDA MIODOWA 270 G"	8	"DRINK_JUICE"
5	"PAP LUCKY STRIKE 22 RED"	69	"CIGARETTES"
6	"SOS BOLONSKI SLOIK DAWT 550G"	6	"GENERAL_FOOD"
7	"ZUREK BUTELKA 0_5 L"	7	"GENERAL"
8	"BELLA CHUSTECZKI ODSWIEZAJACE	7	"CHEMISTRY"

Utilidad_rotacion

SQL

The screenshot shows a database management interface with a 'Navigator' pane on the left and a main query editor on the right. The 'Navigator' pane displays a tree view of database objects under the 'SCHEMAS' tab. The 'Information' pane at the bottom left shows the schema for the 'rotation_products' table. The main query editor contains two SQL queries.

Table: rotation_products

Columns:

Column Name	Data Type
product_code	int PK
product_group	varchar(60)
product_name	varchar(80)

```
80
81 • select day_ from day_sell;
82
83 • select month(str_to_date(day_, "%d/%m/%Y")) AS MES,
84 product_group, (sum(net_value_sale)) as venta_total,
85 (sum(IFNULL(quantity_margin_sold_products,0))) as utilidad_neta,
86 (sum(ifnull(net_value_sale,0)))/(sum(IFNULL(quantity_margin_sold_products,0))) as margen_utilidad
87 from sell
88 group by product_group order by utilidad_neta ASC limit 5;
89
90 • create view utilidad_rotacion as
91 select s.product_name, s.product_group, quantity_margin_sold_products, rotation_time
92 from sell as s
93 join rotation_products as r
94 on s.product_code = r.product_code;
95
96
```

MongoDB

Documents

grocery_store.rotacion_utilidad

READ-ONLY

VIEW

view on

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

FILTER { field: 'value' }

VIEW

Displaying docu

Pgroup: "DAIRY_CHESSE"
Pname: "ALMETTE CZOSNEK NIEDZWIEDZI 150GR"
pmarzajedn: 1.12
rotation_product: 0

Pgroup: "GENERAL_FOOD"
Pname: "AMINO BLYSK GULASZOWA 61G"
pmarzajedn: 0.69
rotation_product: 0

Pgroup: "DAIRY_CHESSE"
Pname: "ALMETTE Z CHRZANEM"
pmarzajedn: 1.12
rotation_product: 0

Pgroup: "TABLETS"
Pname: "ASPIRYN ULTRA 2 SZT"
pmarzajedn: 0.85
rotation_product: 53.42857143