



UNIVERSIDAD
SAN SEBASTIAN

Taller 1 de Minería de datos y Big data:
Inteligencia de negocios.

Integrantes:
Sebastián Díaz
Yasmin Villagra

Profesor:
Mauricio Sepulveda

10 de Mayo

Índice

KPI's	3
Cantidad de productos por tienda:	3
Flujo de clientes por mes:	3
Ventas por vendedor mensual:	3
Ventas por tienda anualmente:	3
Modelo lógico multidimensional	4
Modelo físico	5
Implementación de carga inicial	8
Implementación de carga mensual	9
Cuadro de mando	11
Conclusión	13

KPI's

- Cantidad de productos por tienda:

Se buscará medir la cantidad de productos por tienda, lo que permitirá llevar un mejor registro del stock de productos.

- Flujo de clientes por mes:

Se realizará un seguimiento de cantidad de clientes cada mes, lo que permitirá conocer cuales son los meses del año en los que hay más tendencia tanto negativa como positiva.

- Ventas por vendedor mensual:

Se busca conocer la cantidad de ventas por vendedor, lo que permitirá evaluar el desempeño de cada vendedor.

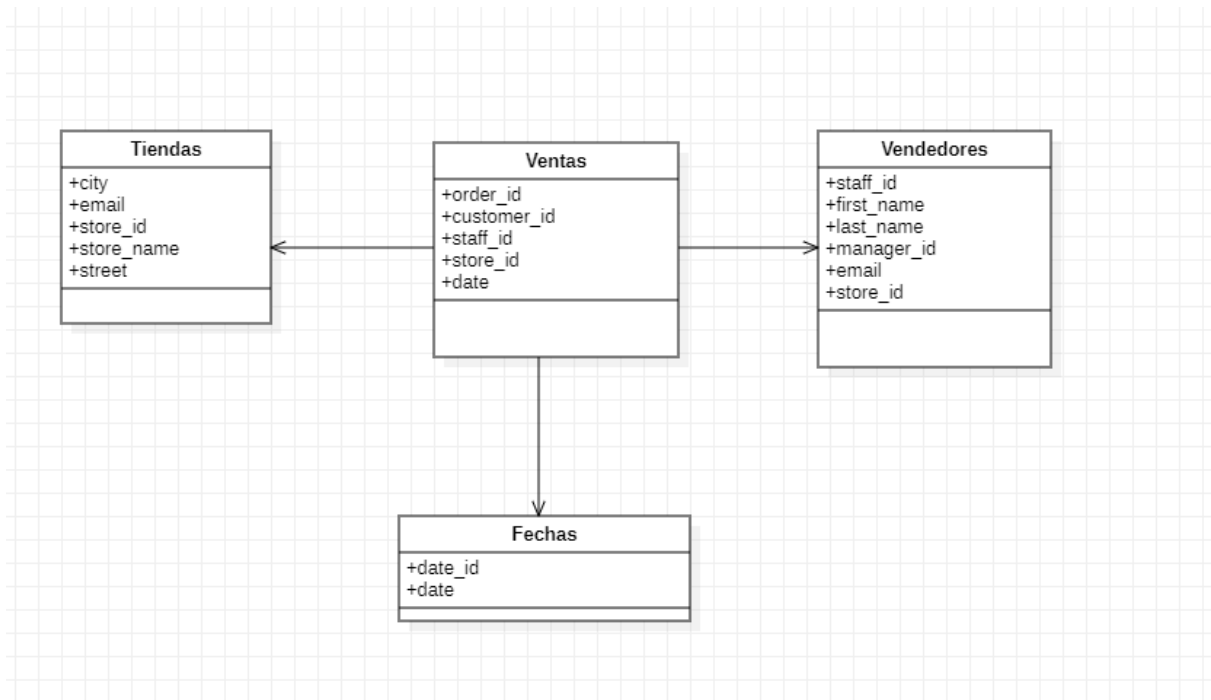
- Ventas por tienda anualmente:

Se revisará el desempeño de cada tienda, lo que permitirá conocer los sectores donde existen más clientes.

Modelo lógico multidimensional

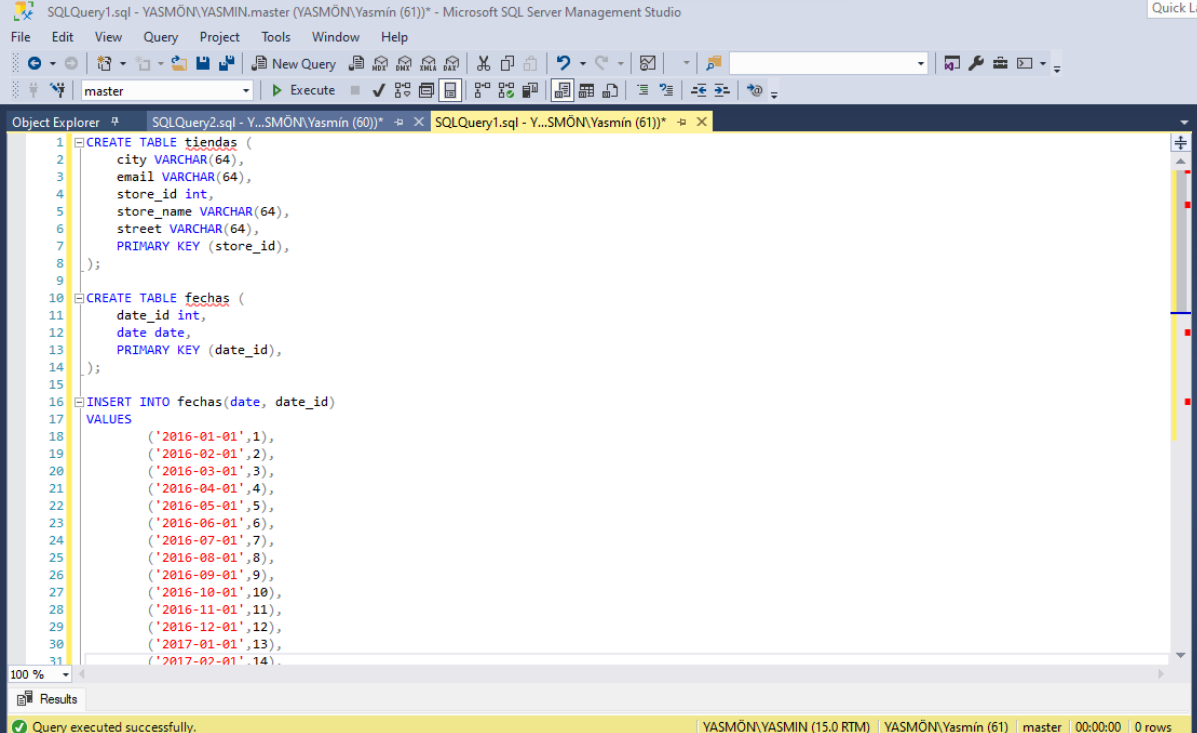
Adaptamos este modelo lógico multidimensional de 4 dimensiones: Tiendas, Vendedores, Ventas y Fechas.

Para poder luego armar nuestro cuadro de mando necesitamos disponer de de campos como por ejemplo, cantidad de pedidos, cantidad de vendedores, id de vendedores, etc.



Modelo físico

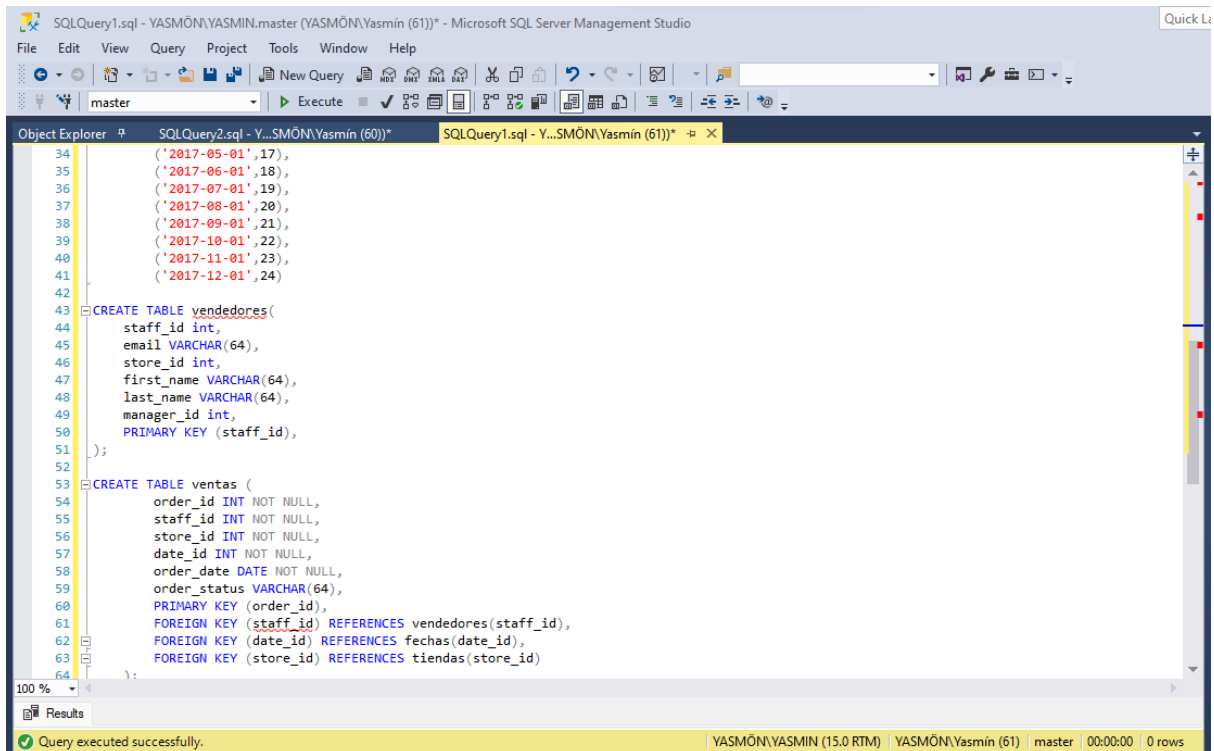
Este código (modelo físico) considera el modelo lógico para la construcción de este y los datos que se extraen para las variables, así tendremos distintas tablas entre los datos y el Data Warehouse.



The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays a SQL query script with the following content:

```
1 CREATE TABLE tiendas (
2     city VARCHAR(64),
3     email VARCHAR(64),
4     store_id int,
5     store_name VARCHAR(64),
6     street VARCHAR(64),
7     PRIMARY KEY (store_id),
8 );
9
10 CREATE TABLE fechas (
11     date_id int,
12     date_date,
13     PRIMARY KEY (date_id),
14 );
15
16 INSERT INTO fechas(date, date_id)
17 VALUES
18     ('2016-01-01',1),
19     ('2016-02-01',2),
20     ('2016-03-01',3),
21     ('2016-04-01',4),
22     ('2016-05-01',5),
23     ('2016-06-01',6),
24     ('2016-07-01',7),
25     ('2016-08-01',8),
26     ('2016-09-01',9),
27     ('2016-10-01',10),
28     ('2016-11-01',11),
29     ('2016-12-01',12),
30     ('2017-01-01',13),
31     ('2017-02-01',14);
```

The status bar at the bottom indicates: "Query executed successfully." followed by "YASMÓN\YASMIN (15.0 RTM) | YASMÓN\Yasmin (61) | master | 00:00:00 | 0 rows".



```

CREATE TABLE tiendas (
    city VARCHAR(64),
    email VARCHAR(64),
    store_id int,
    store_name VARCHAR(64),
    street VARCHAR(64),
    PRIMARY KEY (store_id),
);

```

```

CREATE TABLE fechas (
    date_id int,
    date date,
    PRIMARY KEY (date_id),
);

```

```

INSERT INTO fechas(date, date_id)
VALUES
    ('2016-01-01',1),
    ('2016-02-01',2),
    ('2016-03-01',3),
    ('2016-04-01',4),
    ('2016-05-01',5),
    ('2016-06-01',6),
    ('2016-07-01',7),
    ('2016-08-01',8),
    ('2016-09-01',9),
    ('2016-10-01',10),

```

```
('2016-11-01',11),
('2016-12-01',12),
('2017-01-01',13),
('2017-02-01',14),
('2017-03-01',15),
('2017-04-01',16),
('2017-05-01',17),
('2017-06-01',18),
('2017-07-01',19),
('2017-08-01',20),
('2017-09-01',21),
('2017-10-01',22),
('2017-11-01',23),
('2017-12-01',24)
```

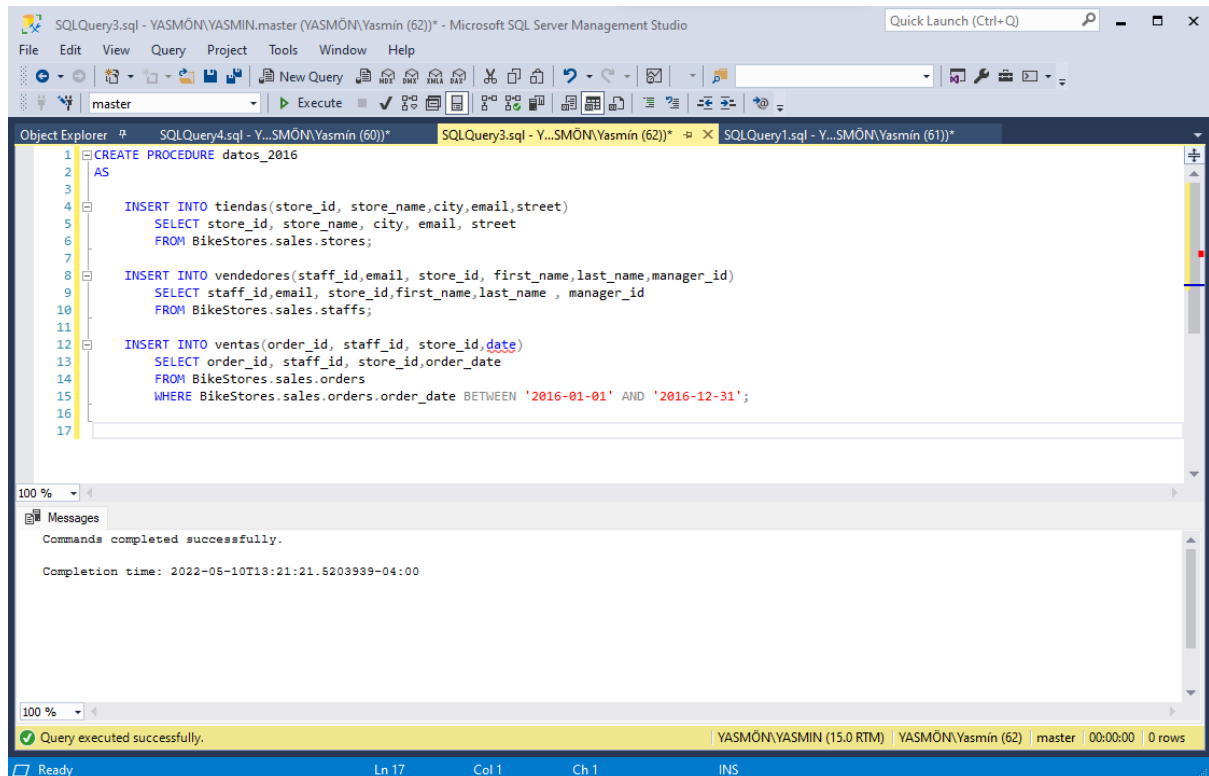
```
CREATE TABLE vendedores(
    staff_id int,
    email VARCHAR(64),
    store_id int,
    first_name VARCHAR(64),
    last_name VARCHAR(64),
    manager_id int,
    PRIMARY KEY (staff_id),
);
```

```
CREATE TABLE ventas (
    order_id INT NOT NULL,
    staff_id INT NOT NULL,
    store_id INT NOT NULL,
    date date NOT NULL,
    PRIMARY KEY (order_id),
    FOREIGN KEY (staff_id) REFERENCES vendedores(staff_id),
    FOREIGN KEY (store_id) REFERENCES tiendas(store_id)
);
```

Implementación de carga inicial

La carga inicial inicia con la información más importante, esta no cambia y si está hecha correctamente marcará las primera ideas para verificar que la base de datos está hecha de forma correcta y es estable.

Aquí consideramos 3 de las 4 dimensiones, ya que la dimensión de “Fecha”, o tabla, la rellenamos anteriormente con datos pedidos en el taller, por esto aqui solo utilizamos los datos de “tiendas”, “vendedores” y “ventas”



```
CREATE PROCEDURE datos_2016
```

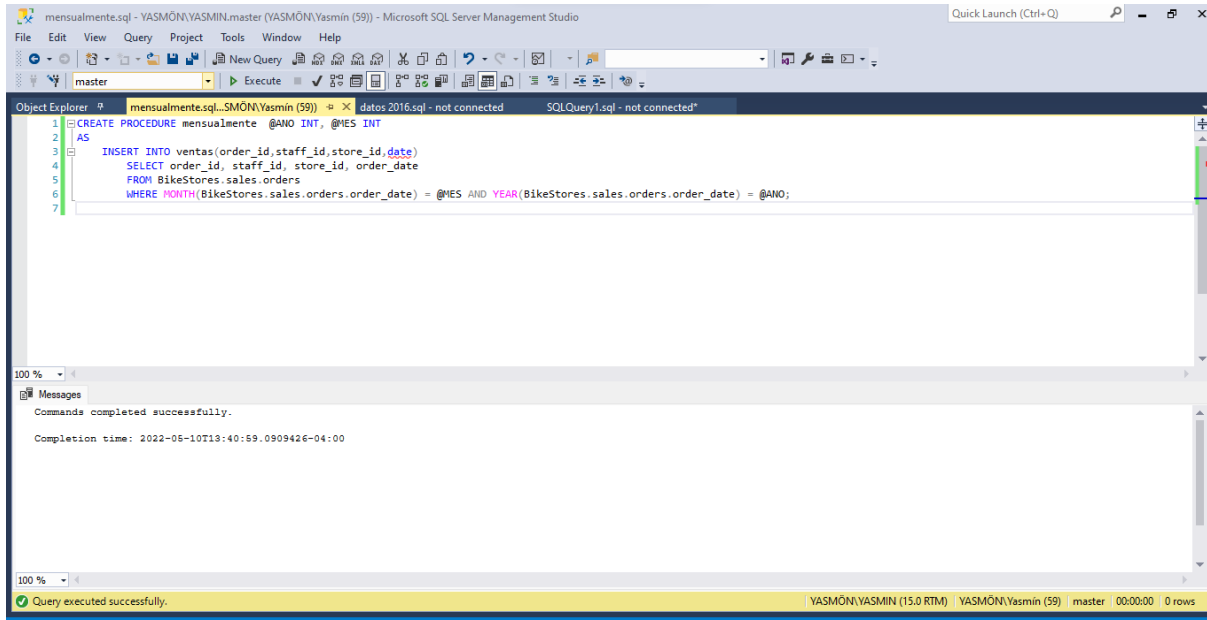
```
AS
```

```
INSERT INTO tiendas(store_id, store_name,city,email,street)
SELECT store_id, store_name, city, email, street
FROM BikeStores.sales.stores;
```

```
INSERT INTO vendedores(staff_id,email, store_id, first_name,last_name,manager_id)
SELECT staff_id,email, store_id,first_name,last_name , manager_id
FROM BikeStores.sales.staffs;
```

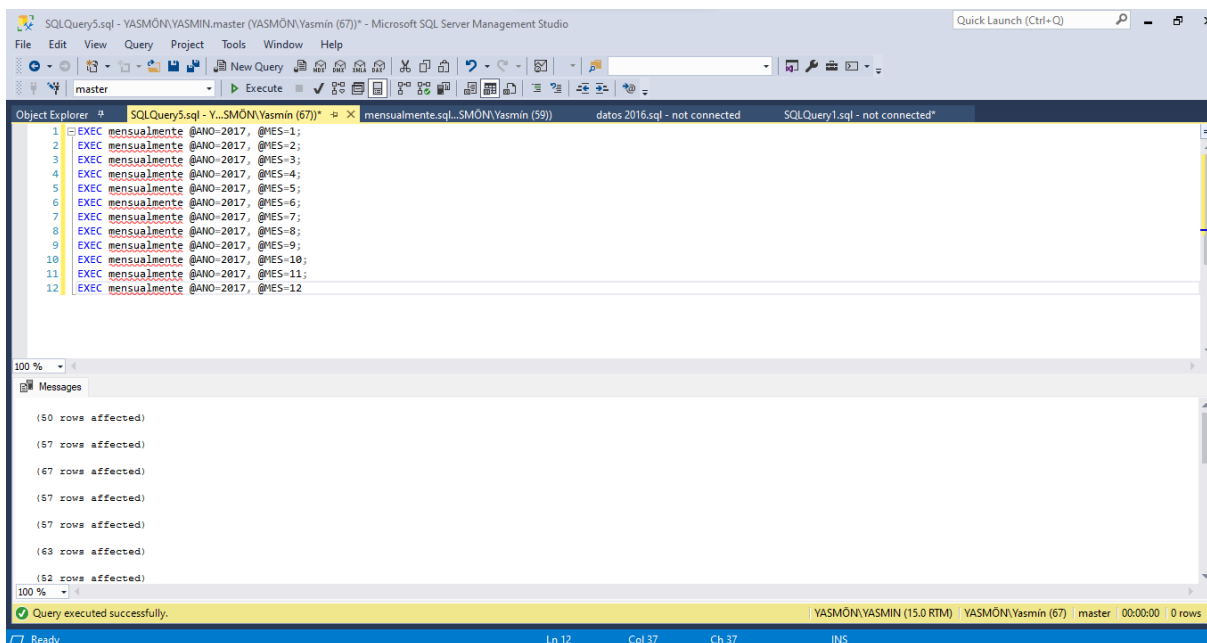
```
INSERT INTO ventas(order_id, staff_id, store_id,date)
SELECT order_id, staff_id, store_id,order_date
FROM BikeStores.sales.orders
WHERE BikeStores.sales.orders.order_date BETWEEN '2016-01-01' AND
'2016-12-31';
```


Implementación de carga mensual



```
CREATE PROCEDURE mensualmente @ANO INT, @MES INT
AS
    INSERT INTO ventas(order_id,staff_id,store_id,date)
    SELECT order_id, staff_id, store_id, order_date
    FROM BikeStores.sales.orders
    WHERE MONTH(BikeStores.sales.orders.order_date) = @MES AND
YEAR(BikeStores.sales.orders.order_date) = @ANO;
```

Probando la implementación de carga mensual anteriormente hecha:

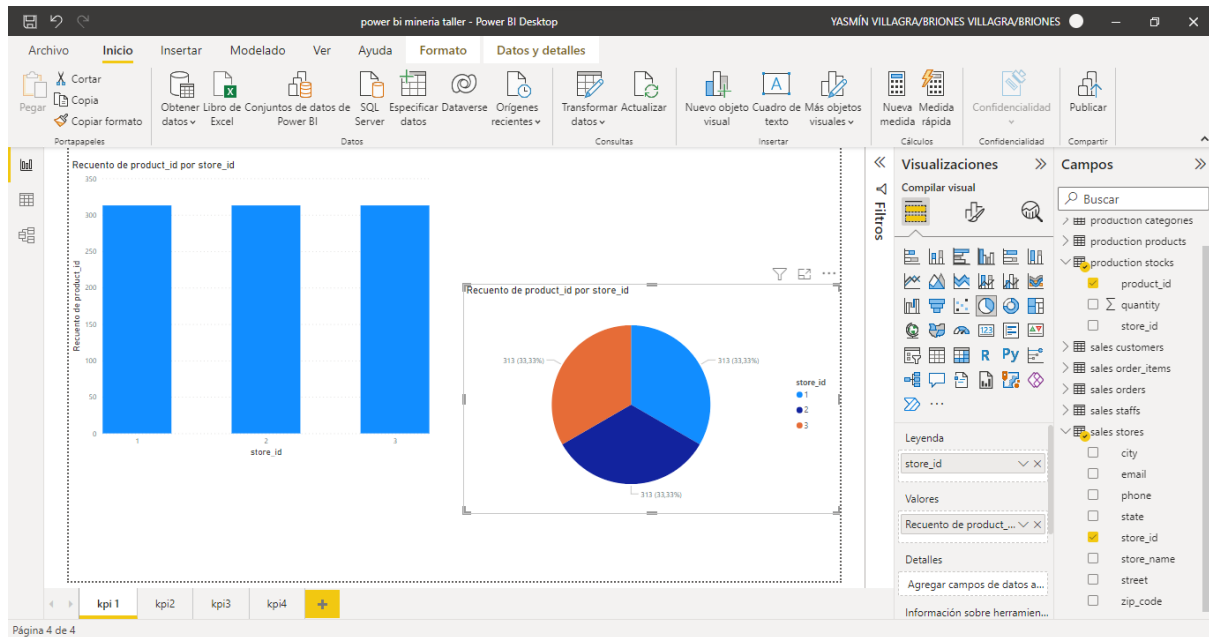


EXEC mensualmente @ANO=2017, @MES=1;
EXEC mensualmente @ANO=2017, @MES=2;
EXEC mensualmente @ANO=2017, @MES=3;
EXEC mensualmente @ANO=2017, @MES=4;
EXEC mensualmente @ANO=2017, @MES=5;
EXEC mensualmente @ANO=2017, @MES=6;
EXEC mensualmente @ANO=2017, @MES=7;
EXEC mensualmente @ANO=2017, @MES=8;
EXEC mensualmente @ANO=2017, @MES=9;
EXEC mensualmente @ANO=2017, @MES=10;
EXEC mensualmente @ANO=2017, @MES=11;
EXEC mensualmente @ANO=2017, @MES=12

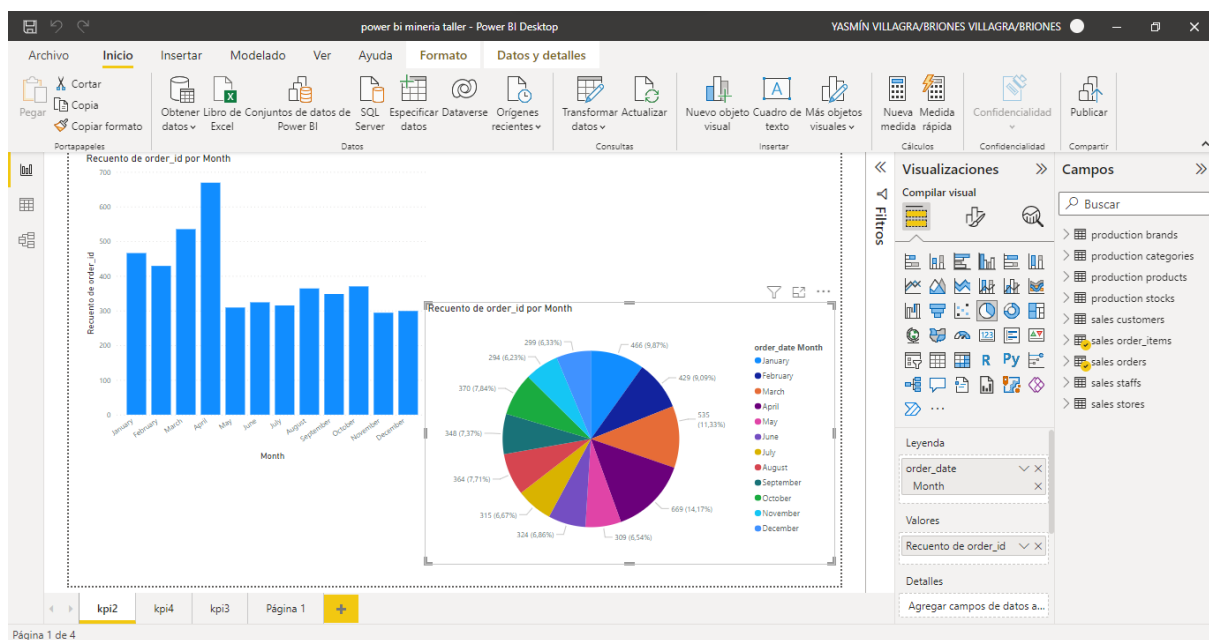
Cuadro de mando

Las siguientes imágenes muestran el cuadro de mando, la visualización de los KPI's definidos anteriormente:

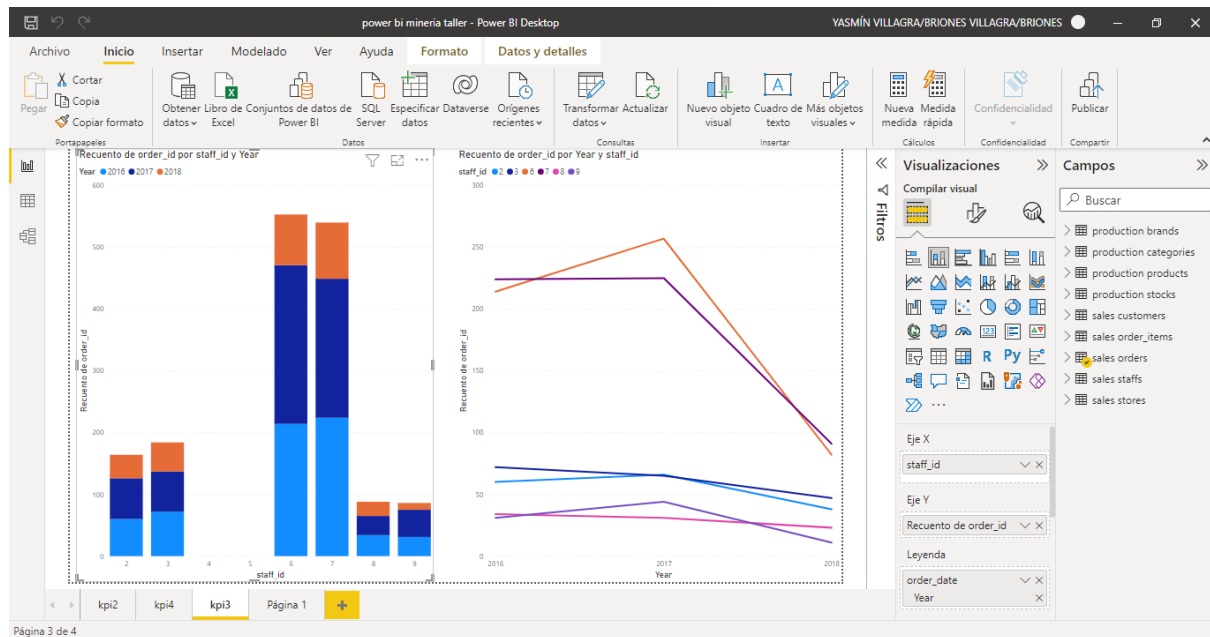
KPI 1



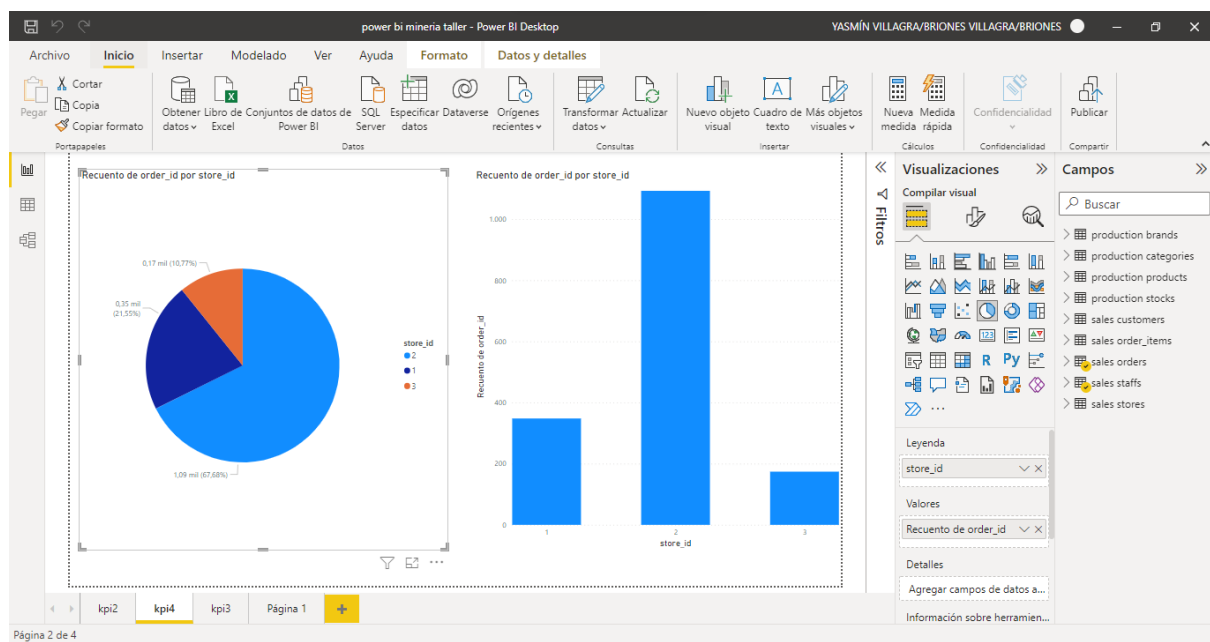
KPI 2



KPI 3



KPI 4



Link para acceder a los cuadros de mando en power BI:

https://app.powerbi.com/links/Ah_poBArMT?ctid=7a599002-001c-432c-846e-1ddca9f6b299&pbi_source=linkShare

Conclusión

La realización de este taller nos ha permitido adentrarnos más en los KPI's, modelos lógicos y físicos, Data Warehouse y cuadros de mando, y entender más acerca de su utilidad y importancia, ya que la realización de estos nos permitió tener un mejor orden en los datos y una visualización de estos que se vuelve mucho más comprensible a la hora de observarlos, lo que nos ayudará a la hora de la toma de decisiones.