



# PROYECTO DMC- SQL FOR BUSINESS INTELLIGENCE: WALMART



Diego Bernales Valdivia

# Descripción de la empresa:



WALMART ES UNA CORPORACIÓN MULTINACIONAL DE TIENDAS DE ORIGEN ESTADOUNIDENSE, QUE OPERA CADENAS DE GRANDES ALMACENES DE DESCUENTO Y CLUBES DE ALMACENES. FUE FUNDADA POR SAM WALTON EL 2 DE JULIO DE 1962, INCORPORADA EL 31 DE OCTUBRE DE 1969, Y HA COTIZADO EN LA BOLSA DE NUEVA YORK DESDE 1972. SU SEDE PRINCIPAL ESTÁ UBICADA EN BENTONVILLE, ARKANSAS.



EN EL PRESENTE PROYECTO SE EXPLORARÁ INFORMACIÓN RESPECTO A LAS VENTAS SEMANALES (INCLUYENDO SEMANAS FESTIVAS) DE 3 TIPOS DE TIENDAS DE WALMART:

- SUPERCENTER
- NEIGHBORHOOD MARKET
- SUPERMERCADO

UBICADAS EN EL NORTE, SUR, ESTE Y OESTE DE ESTADOS UNIDOS, EN EL PERÍODO DE ENERO-2010 A SEPTIEMBRE-2012.



# OBJETIVOS:

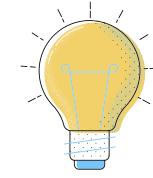
**Objetivo general:** Analizar el comportamiento de las ventas semanales de Walmart en los 3 tipos de tiendas, mediante la implementación de un Data Mart y el uso de herramientas analíticas.

## Objetivos específicos:

- Detectar el comportamiento de ventas en semanas festivas.
- Segmentar los tipos de tiendas de acuerdo al monto de ventas total obtenido en el periodo evaluado. (Caso de Negocio)
- Realizar un análisis exploratorio con la data obtenida en el periodo evaluado.

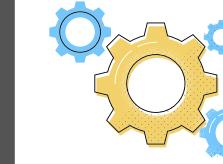


# Metodología del proyecto



## Análisis de requerimiento

DIMENSIONES | MÉTRICAS



## Modelado de datos

DISEÑO



Creación de proceso ingesta  
de data con Integration  
Services - Visual Studio



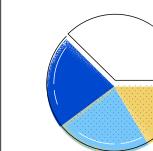
## Aplicación de caso de Segmentación de tiendas

UTILIZANDO PYTHON Y SPYDER



## Construcción del Data Mart

FACT TABLE



## Exploración y análisis de datos

POWER BI

# ANÁLISIS DE REQUERIMIENTOS

## DIMENSIONES

LOCALIDAD

TIENDA

TIPO

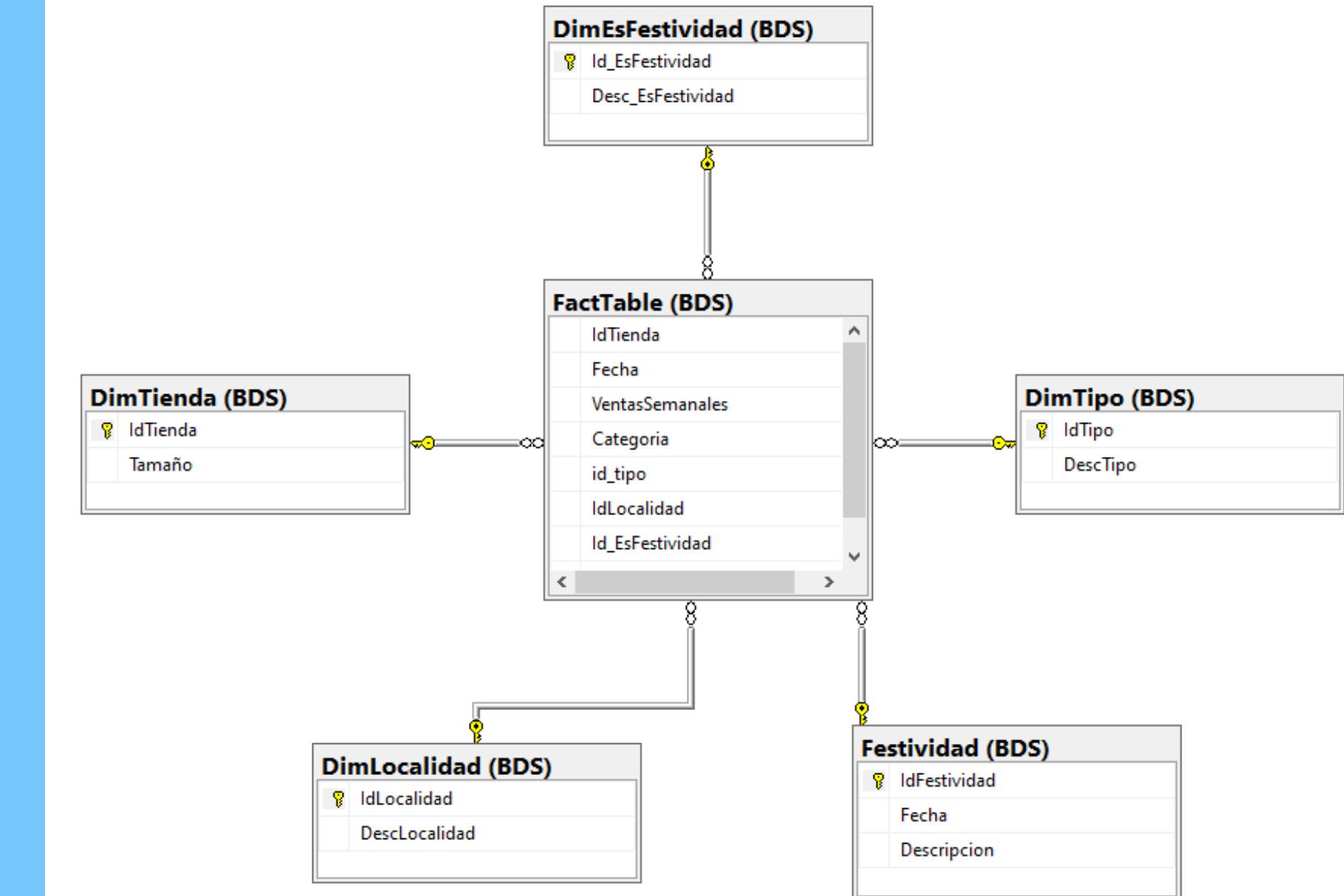
ESFESTIVIDAD

FESTIVIDAD

## MEDIDAS

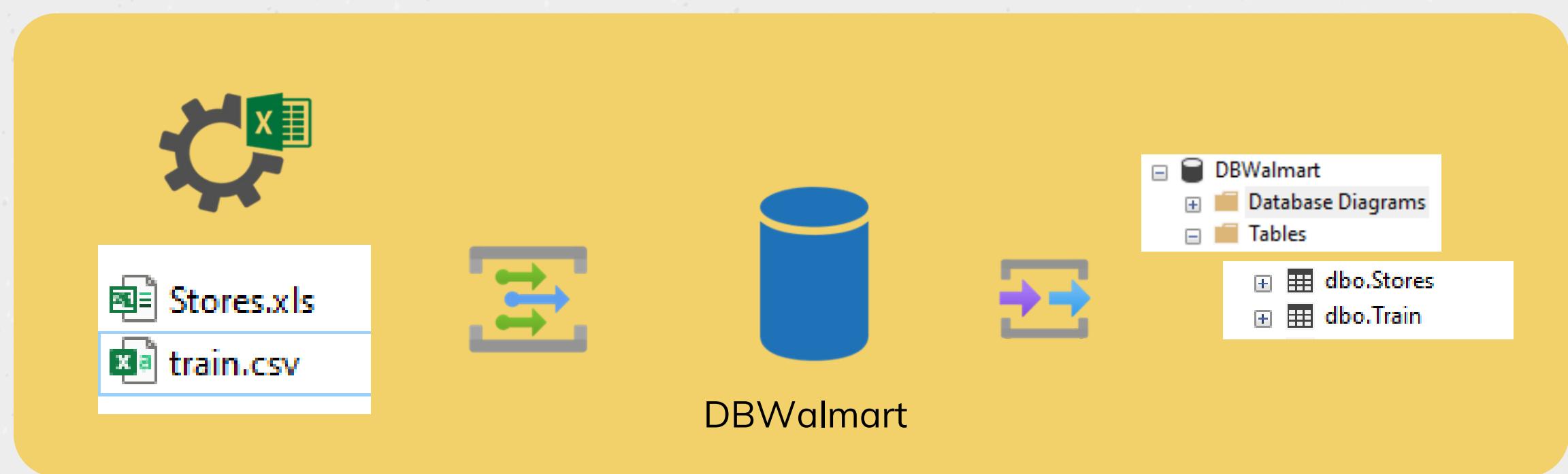
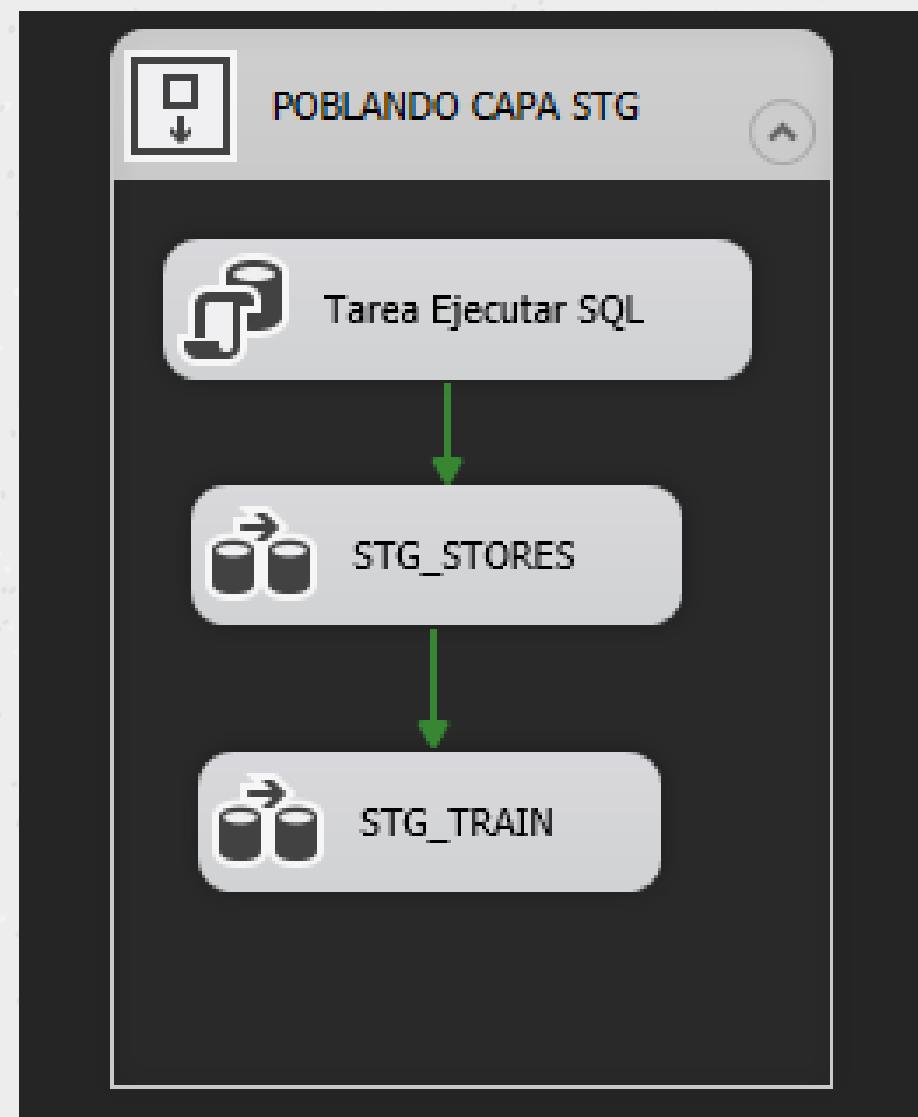
VENTAS  
SEMANALES

# MODELO DE DATOS: Multidimensional (Estrella)



# ETL-Capa Staging

Ingesta de data con Integration Services



# Creación de Data Mart



```
-- Crear la base de datos:  
CREATE DATABASE DBWalmart  
USE DBWalmart  
  
select * from Stores  
select * from Train  
  
--CREAR SCHEMA  
CREATE SCHEMA ODS  
  
--CREAR TABLA ODS.STORES  
SELECT  
    CONVERT(INT,Store) IdTienda,  
    CONVERT(CHAR(1),Stores.Type) TipoTienda,  
    CONVERT(VARCHAR(25),dec_type) DescTipo,  
    CONVERT(INT,Size) Tamaño,  
    CONVERT(int,cod_localidad) IdLocalidad,  
    CONVERT(VARCHAR(5),localidad) Localidad INTO ODS.Stores FROM Stores  
  
--CREAR TABLA ODS.TRAIN  
SELECT  
    CONVERT(INT,Store) IdTienda,  
    CONVERT(INT,Dept) Departamento,  
    CONVERT(DATE,Train.Date,103) Fecha,  
    CONVERT(DECIMAL(10,2), Weekly_Sales) VentasSemanales,  
    CONVERT(VARCHAR(5),IsHoliday) EsFestividad INTO ODS.Train FROM Train  
  
--select * from ODS.Train  
  
--OBTENCIÓN DE LAS VENTAS SEMANALES POR TIENDA EN TABLA DETALLE VENTAS  
SELECT IdTienda, Fecha, SUM(VentasSemanales) VentasSemanales, EsFestividad INTO ODS.DetalleVentas  
FROM ODS.Train GROUP BY IdTienda, Fecha, EsFestividad ORDER BY IdTienda
```



# Creación de Data Mart



```
--CREAR SCHEMA BDS
CREATE SCHEMA BDS
--CREAR DIMENSIONES
---DIMENSION TIENDA
SELECT IdTienda , Tamaño  INTO BDS.DimTienda FROM ODS.Stores
SELECT * FROM BDS.DimTienda

ALTER TABLE BDS.DimTienda
ALTER COLUMN IdTienda INT NOT NULL

---DIMENSION ESFESTIVIDAD
CREATE TABLE BDS.DimEsFestividad(
Id_EsFestividad INT IDENTITY(1,1),
Desc_EsFestividad VARCHAR(5)
)

INSERT INTO BDS.DimEsFestividad (Desc_EsFestividad)
SELECT DISTINCT EsFestividad  FROM ODS.DetalleVentas

---DIMENSION TIPO
SELECT DISTINCT TipoTienda IdTipo, DescTipo INTO BDS.DimTipo FROM ODS.Stores
SELECT * FROM BDS.DimTipo

ALTER TABLE BDS.DimTipo
ALTER COLUMN IdTipo char(1) NOT NULL

---DIMENSION LOCALIDAD
SELECT DISTINCT IdLocalidad, Localidad DescLocalidad INTO BDS.DimLocalidad FROM ODS.Stores
--SELECT * FROM BDS.DimLocalidad

ALTER TABLE BDS.DimLocalidad
ALTER COLUMN IdLocalidad INT NOT NULL
```

# Creación de Data Mart



```
--DIMENSION FESTIVIDAD CREANDO TABLA FESTIVIDAD
CREATE TABLE BDS.Festividad(
    IdFestividad INT NOT NULL,
    Fecha DATE,
    Descripcion VARCHAR(20) NOT NULL
)
--select * from BDS.Festividad
--drop table Festividad
INSERT INTO BDS.Festividad VALUES (1,CAST('2010-02-12' AS DATE), 'Super Bowl'),
                                    (2,CAST('2011-02-11' AS DATE), 'Super Bowl'),
                                    (3,CAST('2012-02-10' AS DATE), 'Super Bowl'),
                                    (4,CAST('2010-09-10' AS DATE), 'Labor Day'),
                                    (5,CAST('2011-09-09' AS DATE), 'Labor Day'),
                                    (6,CAST('2012-09-07' AS DATE), 'Labor Day'),
                                    (7,CAST('2010-11-26' AS DATE), 'Thaksgiving'),
                                    (8,CAST('2011-11-25' AS DATE), 'Thaksgiving'),
                                    (9,CAST('2012-11-23' AS DATE), 'Thaksgiving'),
                                    (10,CAST('2010-12-31' AS DATE), 'Christmas'),
                                    (11,CAST('2011-12-30' AS DATE), 'Christmas'),
                                    (12,CAST('2012-12-28' AS DATE), 'Christmas'),
                                    (13, NULL, 'No hay festividad')
```

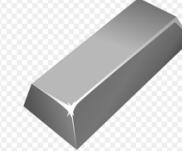


# Creación de Data Mart



## Caso de Negocio:

La Gerencia desea segmentar a las tiendas de cada tipo de acuerdo a las ventas obtenidas en el periodo evaluado. Para esto, decide realizar la segmentación en base a los cuartiles de cada agrupación por Tipo de Tienda y crea las siguientes categorías.

CATEGORÍA	INTERVALOS
DIAMANTE	[Q3,1] 
ORO	[MEDIANA,Q3[ 
PLATA	[Q1,MEDIANA[ 
COBRE	[0,Q1[ 

# Creación de Data Mart

```
-- VENTAS TOTALES, POR TIENDA Y TIPO -----
-- TIENDAS TIPO A --
SELECT A.Store, A.Type , A.dec_type, CONVERT(DECIMAL(10,2), B.Weekly_Sales) Weekly_Sales INTO ODS.VENT_TIE_A
FROM stores A
INNER JOIN train B ON B.Store= A.Store
WHERE A.Type = 'A'
select * from ODS.VENT_TIE_A
--drop table ODS.VENT_TIE_A

select Store, ROUND(SUM(Weekly_Sales),2,2) AS VENTA_TOTAL INTO ODS.VEN_TOT_A
FROM ODS.VENT_TIE_A
GROUP BY Store
ORDER BY Store

--drop table ODS.VEN_TOT_A
--SELECT * FROM ODS.VEN_TOT_A

-- TIENDAS TIPO B --
SELECT A.Store, A.Type , A.dec_type, CONVERT(DECIMAL(10,2), B.Weekly_Sales) Weekly_Sales INTO ODS.VENT_TIE_B
FROM stores A
INNER JOIN train B ON B.Store= A.Store
WHERE A.Type = 'B'
select * from ODS.VENT_TIE_B
--DROP TABLE ODS.VENT_TIE_B

select Store, ROUND(SUM(Weekly_Sales),2,2) AS VENTA_TOTAL INTO ODS.VEN_TOT_B
FROM ODS.VENT_TIE_B
GROUP BY Store
ORDER BY Store

--drop table BDS.VEN_TOT_B
--SELECT * FROM ODS.VEN_TOT_B

-- TIENDAS TIPO C --
SELECT A.Store, A.Type , A.dec_type, CONVERT(DECIMAL(10,2), B.Weekly_Sales) Weekly_Sales
INTO ODS.VENT_TIE_C
FROM stores A
INNER JOIN train B ON B.Store= A.Store
WHERE A.Type = 'C'

select * from ODS.VENT_TIE_C

select Store, ROUND(SUM(Weekly_Sales),2,2) AS VENTA_TOTAL INTO ODS.VEN_TOT_C
FROM ODS.VENT_TIE_C
GROUP BY Store
ORDER BY Store

--drop table ODS.VEN_TOT_C
--SELECT * FROM ODS.VEN_TOT_C
```

# Creación de Data Mart: Aplicación de caso de negocio

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue May 31 23:50:10 2022
4
5 @author: 51922
6 """
7
8 import pandas as pd
9 import pyodbc as podbc
10 from sqlalchemy import create_engine
11 import urllib
12 from sklearn.preprocessing import MinMaxScaler
13
14 # Declaracion de funciones
15 def conexion_sql_server():
16     connection = podbc.connect(
17         Trusted_Connection='Yes',
18         Driver='{SQL Server Native Client 11.0}',
19         Server='DIEGO-PC',
20         Database='DBWalmart'
21     )
22     return connection
23
24 def select_to_df(sql_query,connection):
25     try:
26         df=pd.read_sql(sql_query, connection)
27     except Exception as e:
28         print(e)
29     #connection.close()
30     return df
31
32 def categoria(x):
33     if (x>=0) and (x<q1):
34         return 'Cobre'
35     elif (x>=q1) and (x<m):
36         return 'Plata'
37     elif (x>=m) and (x<q3):
38         return 'Oro'
39     else:
40         return 'Diamante'
41
42 conexion=conexion_sql_server()
43
44 df1=select_to_df('SELECT * FROM ODS.VEN_TOT_A',conexion)
45 df2=select_to_df('SELECT * FROM ODS.VEN_TOT_B',conexion)
46 df3=select_to_df('SELECT * FROM ODS.VEN_TOT_C',conexion)
47
48 # Transformando columnas
49 df1['IdTienda'] = df1['Store'].astype(int)
50 df2['IdTienda'] = df2['Store'].astype(int)
51 df3['IdTienda'] = df3['Store'].astype(int)
52
53 df1['IdTienda'] = df1['IdTienda'].apply(lambda _: str(_))
54 df2['IdTienda'] = df2['IdTienda'].apply(lambda _: str(_))
55 df3['IdTienda'] = df3['IdTienda'].apply(lambda _: str(_))
56
57 #Para escalar los datos de las ventas y tener valores entre 0 y 1
58 scaler = MinMaxScaler()
59 df1[['Venta_escalada']] = scaler.fit_transform(df1[['VENTA_TOTAL']])
60 df2[['Venta_escalada']] = scaler.fit_transform(df2[['VENTA_TOTAL']])
61 df3[['Venta_escalada']] = scaler.fit_transform(df3[['VENTA_TOTAL']])
62
63 #Categorizando
64 data=[df1,df2,df3]
65
66 for df in data:
67     q1,m,q3=df['Venta_escalada'].quantile([.25, .50, .75])
68     df['Categoria']=df['Venta_escalada'].apply(categoria)
69     df.drop('Store', axis=1, inplace=True)
70     df.drop('Venta_escalada', axis=1, inplace=True)
71     df.drop('VENTA_TOTAL', axis=1, inplace=True)
72
73 #Guardando resultado
74
75 df_final= pd.concat([df1, df2, df3])
76
77 quoted = urllib.parse.quote_plus("DRIVER={SQL Server};SERVER=DIEGO-PC;DATABASE=DBWalmart")
78 engine = create_engine('mssql+pyodbc://?odbc_connect={}'.format(quoted))
79
80 df_final.to_sql('Categoria', schema='ODS', con = engine, chunksize=200, method='multi', index=False, if_exists='replace')
81
82 conexion.close()
```



# Creación de Data Mart



```
--CREANDO LA FACT TABLE

SELECT* FROM ODS.Stores
SELECT* FROM ODS.DetalleVentas
SELECT* FROM ODS.Categoría -- DEL CASO DE NEGOCIO

SELECT a.IdTienda,a.Fecha,a.VentasSemanales,e.Categoría, b.TipoTienda AS id_tipo , b.IdLocalidad,c.Id_EsFestividad ,
CASE
WHEN d.IdFestividad IS NULL THEN 13
ELSE d.IdFestividad END AS IdFestividad
INTO BDS.FactTable
FROM ods.DetalleVentas a
left join ods.Stores b ON a.IdTienda=b.IdTienda
left join BDS.DimEsFestividad c ON a.EsFestividad=c.Desc_EsFestividad left join BDS.Festividad d ON d.Fecha=a.Fecha
left join ODS.Categoría e on e.IdTienda = a.IdTienda

--SELECT * FROM BDS.FactTable

--CREANDO LOS PRIMARY KEY

ALTER TABLE BDS.DimTienda
ADD CONSTRAINT PK_IdTienda PRIMARY KEY (IdTienda)

ALTER TABLE BDS.DimLocalidad
ADD CONSTRAINT PK_IdLocalidad PRIMARY KEY (IdLocalidad)

ALTER TABLE BDS.DimTipo
ADD CONSTRAINT PK_IdTipo PRIMARY KEY (IdTipo)

ALTER TABLE BDS.DimEsFestividad
ADD CONSTRAINT PK_Id_EsFestividad PRIMARY KEY (Id_EsFestividad)

ALTER TABLE BDS.Festividad
ADD CONSTRAINT PK_IdFestividad PRIMARY KEY (IdFestividad)
```

# Creación de Data Mart



```
--CREANDO LOS FOREIGN KEY

Alter table BDS.FactTable
Add Constraint FK_IdTienda foreign key (IdTienda) references BDS.DimTienda(IdTienda)

Alter table BDS.FactTable
Add Constraint FK_IdLocalidad foreign key (IdLocalidad) references BDS.DimLocalidad(IdLocalidad)

Alter table BDS.FactTable
Add Constraint FK_IdTipo foreign key (Id_Tipo) references BDS.DimTipo(IdTipo)

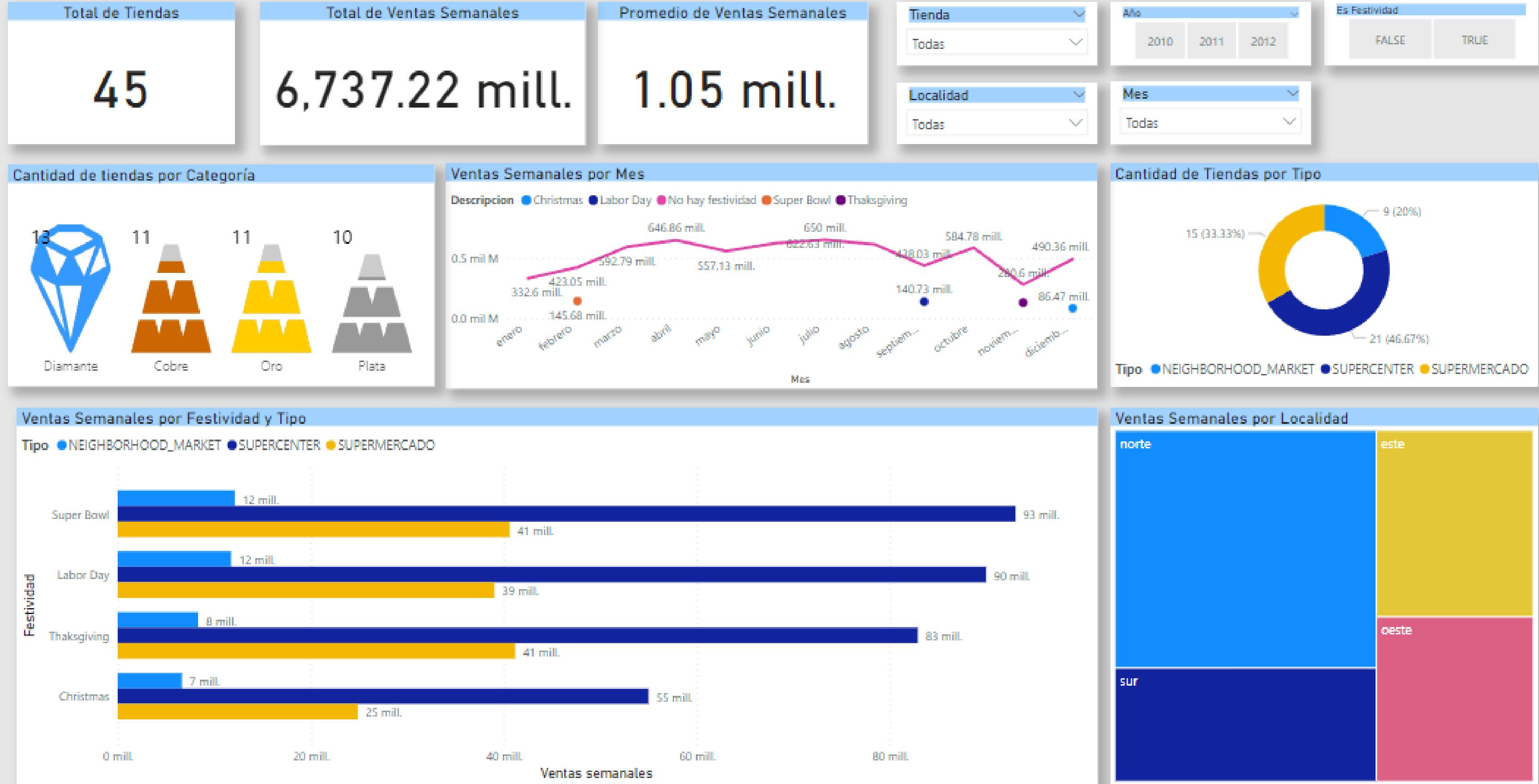
Alter table BDS.FactTable
Add Constraint FK_IdEsFestividad foreign key (Id_EsFestividad) references BDS.DimEsFestividad(Id_EsFestividad)

Alter table BDS.FactTable
Add Constraint FK_IdFestividad foreign key (IdFestividad) references BDS.Festividad(IdFestividad)
```

# Exploración y Análisis de datos

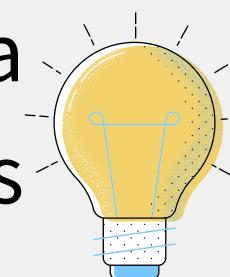


# VENTAS DE WALMART STORE



# Conclusiones:

- El año con mayor promedio de ventas es el 2010 con un monto de 1.06 millones de dólares.
- En el mes de Diciembre se registró la mayor cantidad de ventas en los periodos 2010 y 2011, con un monto de 490.36 millones de dólares.
- La semana festiva con mayores ventas semanales se da en Acción de Gracias, teniendo mayores ventas en las tiendas de tipo Supercenter.
- En base a las ventas semanales del periodo analizado se logró segmentar a las tiendas en 4 categorías: 13 Diamante, 11 cobre, 11 oro y 10 plata.
- Se logra observar una mayor cantidad de ventas en el norte del país a comparación de las otras localidades, provenientes mayormente de las tiendas de Tipo Supercenter.
- La integración de Python y SQL-server es un método bastante factible para la realización de operaciones matemáticas con las tablas y el guardado de resultados en los esquemas adecuados.



**GRACIAS**