**📘 Project Overview**

**This project focuses on analysing and optimising the logistics and supply chain operations of a retail company using data-driven techniques.**

**The goal is to enhance understanding of delivery efficiency, customer segmentation, and operational performance by integrating Python, SQL, and Tableau.**

**The analysis uncovers key insights around delivery delays, profit-driving categories, and customer behaviour — helping improve supply chain decision-making.**

**Data cleaning and exploratory data analysis with Python**

**🧹 Data Cleaning & Preparation (Python)**

**Performed comprehensive preprocessing on the DataCoSupplyChainDataset.csv:**

**• Removed unnecessary columns with high missing values (Product Description, Order Zipcode)**

**• Standardised column names to snake\_case for consistency**

**• Converted date fields (order\_date, shipping\_date) to datetime format**

**• Engineered key features such as:**

**• delivery\_delay\_days – actual shipping days minus scheduled days**

**• order\_to\_ship\_days – days between order and shipment**

**• processing\_speed – categorised as Same Day, Next Day, Standard, or Slow**

## importing necessary library

"""

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

pd.options.display.max\_columns= None

df= pd.read\_csv('DataCoSupplyChainDataset.csv',encoding='latin1')

df.head()

df.shape

df.info()

"""## check for Null values"""

df.isna().sum()

"""#order zip code and product description has null values hence dropping those columns"""

df.drop(columns=['Product Description','Order Zipcode'],inplace=True)

df.isna().sum()

df.columns= df.columns.str.lower().str.replace(' ','\_')

df.columns

"""## converting the datatype of Date columns"""

cols\_to\_update= ['shipping\_date\_(dateorders)','order\_date\_(dateorders)']

for cols in cols\_to\_update:

df[cols]= pd.to\_datetime(df[cols],errors='coerce')

df.dtypes

## renaming the columns

df.rename(columns={'order\_date\_(dateorders)':'order\_date','shipping\_date\_(dateorders)':'shipping\_date'},inplace=True)

df.head()

df['delivery\_delay\_days'] = df['days\_for\_shipping\_(real)']- df['days\_for\_shipment\_(scheduled)']

df['order\_to\_ship\_days'] = (df['shipping\_date'] - df['order\_date']).dt.days

# Processing speed category

def categorize\_processing(days):

if days == 0:

return 'Same Day'

elif days == 1:

return 'Next Day'

elif days <= 3:

return 'Standard (2-3 days)'

else:

return 'Slow (>3 days)'

df['processing\_speed'] = df['order\_to\_ship\_days'].apply(categorize\_processing)

df[['order\_to\_ship\_days','processing\_speed']].head(5)

df.sample(5)

df.shape

df.order\_state.nunique()

df.order\_region.value\_counts()

df.order\_status.value\_counts()

df.shipping\_mode.value\_counts()

df.customer\_segment.value\_counts()

df.category\_name.value\_counts()

df[['delivery\_status']].head()

pip install psycopg2-binary sqlalchemy

'''from sqlalchemy import create\_engine

import urllib.parse

username = 'postgres'

password = "S@ndy36batra"

host='localhost'

port='5432'

database='customer'

# URL encode the password to handle special characters

encoded\_password = urllib.parse.quote\_plus(password)

# Use the encoded\_password in the connection string, not the original password

engine = create\_engine(f"postgresql+psycopg2://{username}:{encoded\_password}@{host}:{port}/{database}")

# Step 2: Load DataFrame into PostgreSQL

table\_name = "supply\_chain" # choose any table name

df.to\_sql(table\_name, engine, if\_exists="replace", index=False)

print(f"Data successfully loaded into table '{table\_name}' in database '{database}'.")'''

df.to\_excel('supplychain.xlsx')

**Data analysis using SQL queries**

**🧮 SQL Business Analysis**

**Performed key analytical queries to identify actionable business insights:**

**• Top Profitable Categories → Identified top 10 categories by total profit**

**• Regional Delivery Delays → Found regions with highest average delay**

**• Cancellation Rate by Category → Calculated cancellation % per product line**

**• Average Shipping Days by Mode → Compared delivery performance by shipping type**

**• Top Customer Segments → Ranked segments by total sales contribution**

**• Delivery Status Distribution → Measured on-time vs delayed vs canceled orders**

--Which product categories drive most profit?

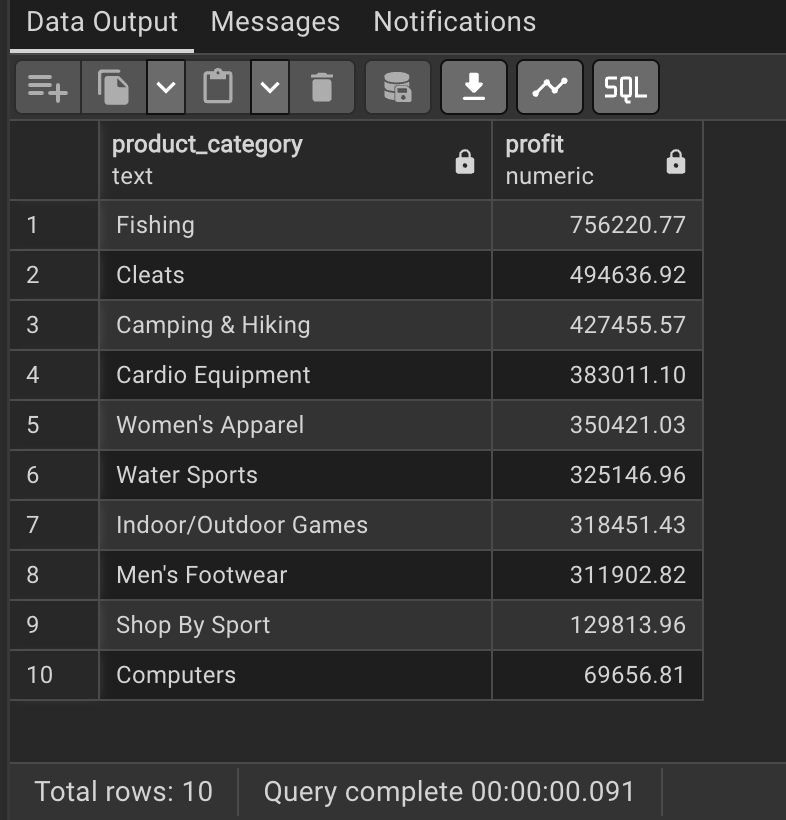
select category\_name as product\_category,round(sum(benefit\_per\_order::Numeric),2) as profit

from supply\_chain

group by category\_name

order by profit desc

limit 10



--Which regions have the longest delivery delays?

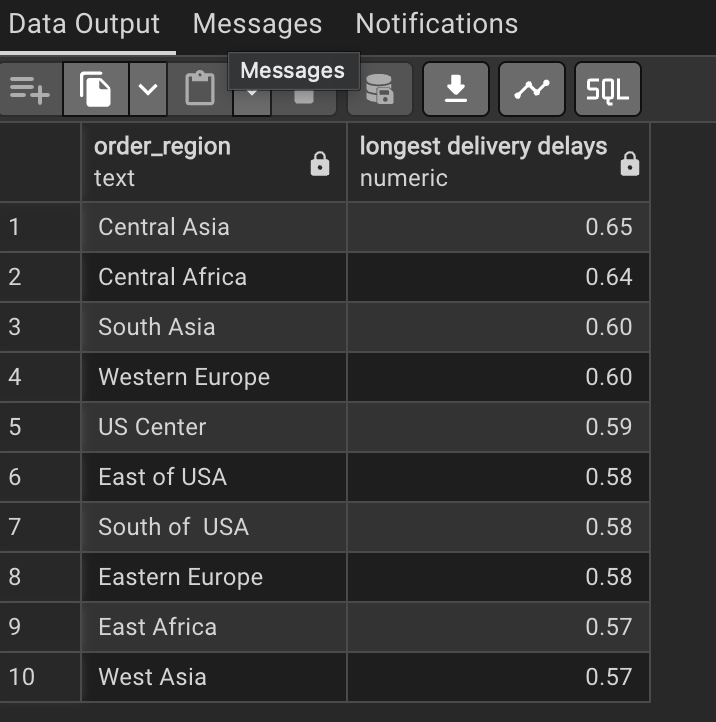
select order\_region,round(avg(delivery\_delay\_days),2) as "longest delivery delays"

from supply\_chain

group by order\_region

order by avg(delivery\_delay\_days) desc

limit 10;



– What is the return rate per category

select

category\_name,

sum(case when order\_status='CANCELED' then 1 else 0 end) as return\_orders,

round((sum(case when order\_status='CANCELED' then 1 else 0 end)\*1.0/

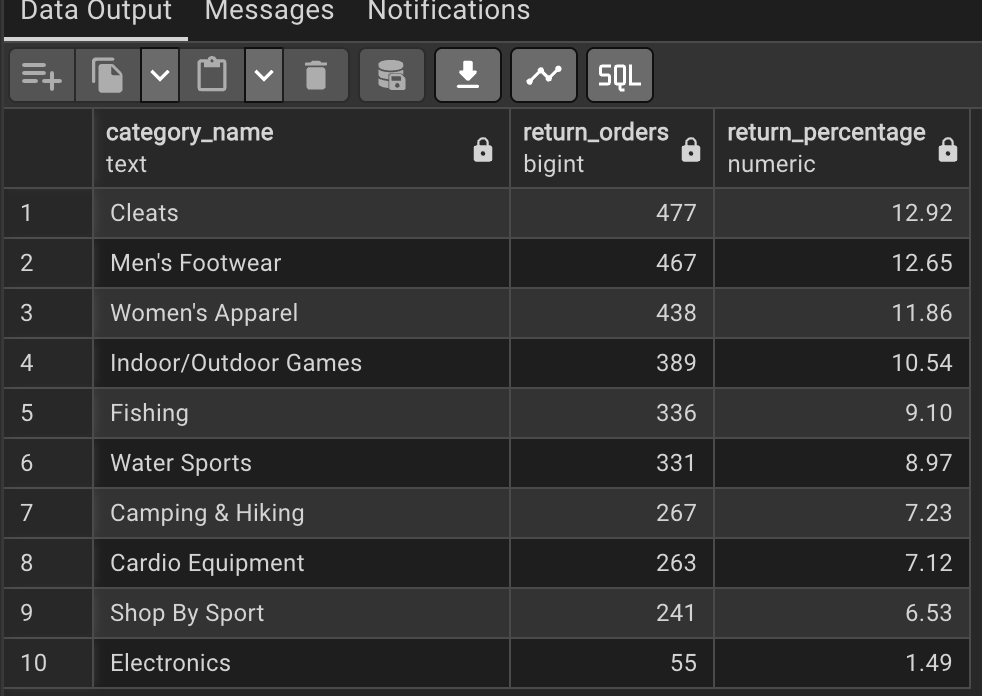
(select count(\*) from supply\_chain where order\_status='CANCELED')\*100),2) as return\_percentage

from supply\_chain

group by category\_name

order by 3 desc

limit 10;



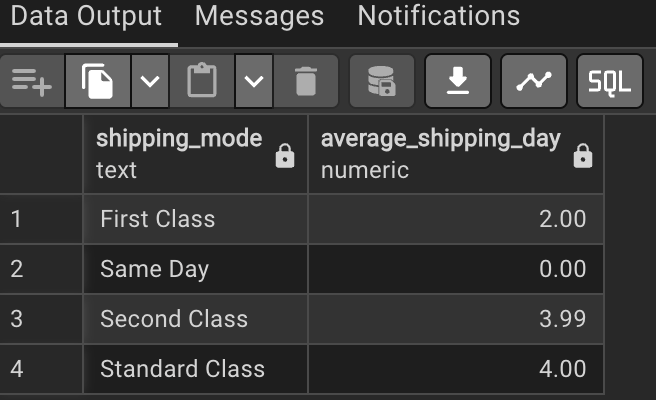
--What is the average shipping day by delivery type?

select

shipping\_mode,round(avg(order\_to\_ship\_days),2) as average\_shipping\_day

from supply\_chain

group by shipping\_mode



-- which customer segment is spending the most

select

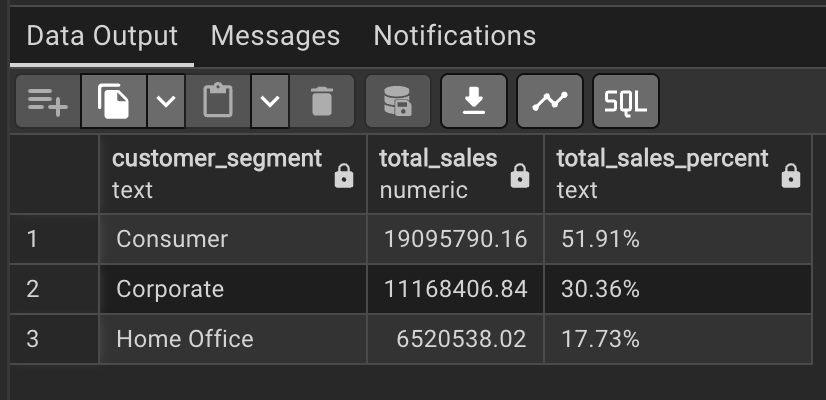
customer\_segment,round(sum(sales::numeric),2) as total\_sales,

concat(round((round(sum(sales::numeric),2)/(select sum(sales::numeric) from supply\_chain)\*100),2),'%') as total\_sales\_percent

from supply\_chain

group by customer\_segment

order by total\_sales desc



--Distribution count of delivery status

select

count(\*) as count\_of\_total\_delivery,

count(case when delivery\_status = 'Advance shipping' then 1 end) as Advance\_shipping,

count(case when delivery\_status = 'Late delivery' then 1 end) as Late\_delivery,

count(case when delivery\_status = 'Shipping canceled' then 1 end )as Shipping\_canceled,

count(case when delivery\_status = 'Shipping on time' then 1 end) as Shipping\_on\_time

from supply\_chain

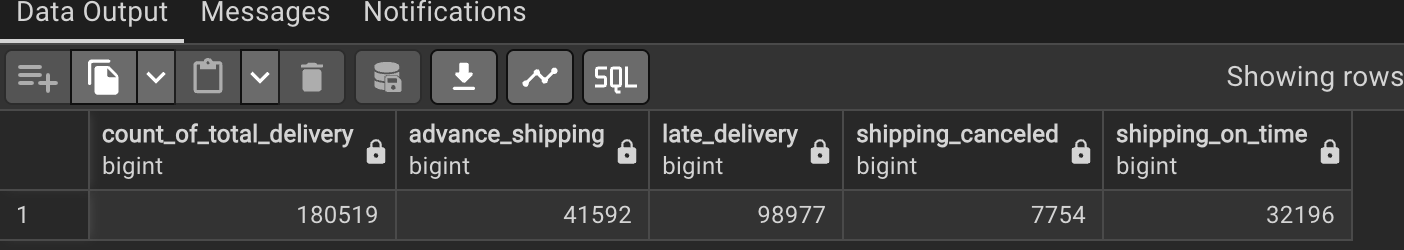


Tableau Dashboard

