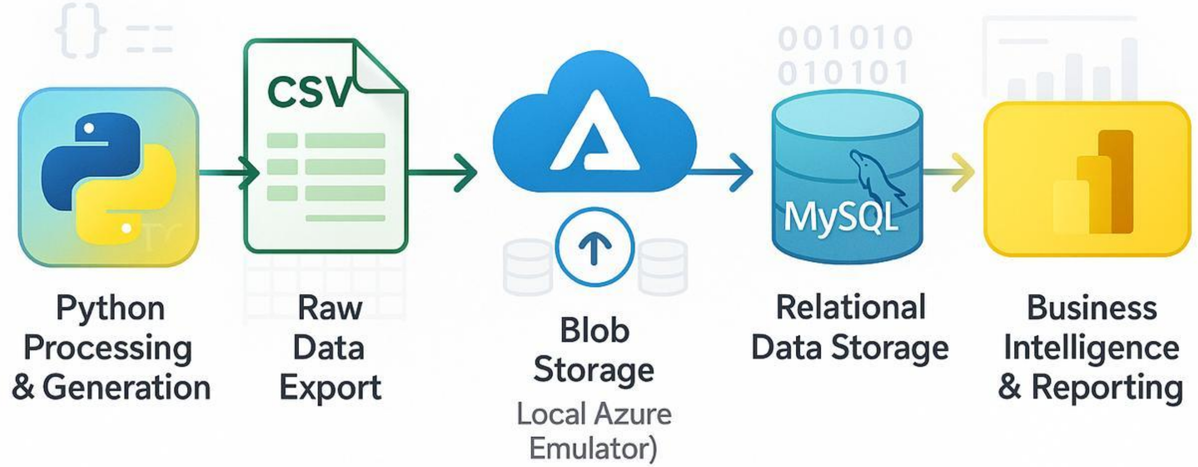
- **Project Name** *Order Management Dashboard*
- Your Name: **Tanmay Sharma**
- Role (Data/Business Analyst)



Problem Statement

- Difficulty in tracking order status, shipment delays, and customer profitability.
- Data was scattered, no central platform to visualize performance.

Project Objectives

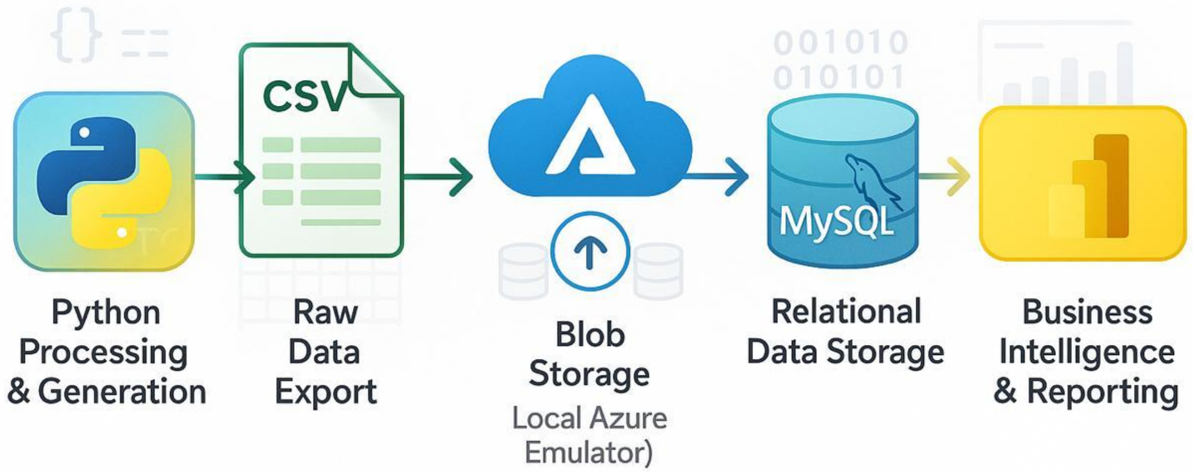
- Automate data generation using Python
- Store raw data in Azurite (Azure Blob Emulator) for staging
- Import data into MySQL for structured storage
- Create interactive dashboards in Power BI
- Enable drill-through for detailed customer analysis

Tools & Technologies

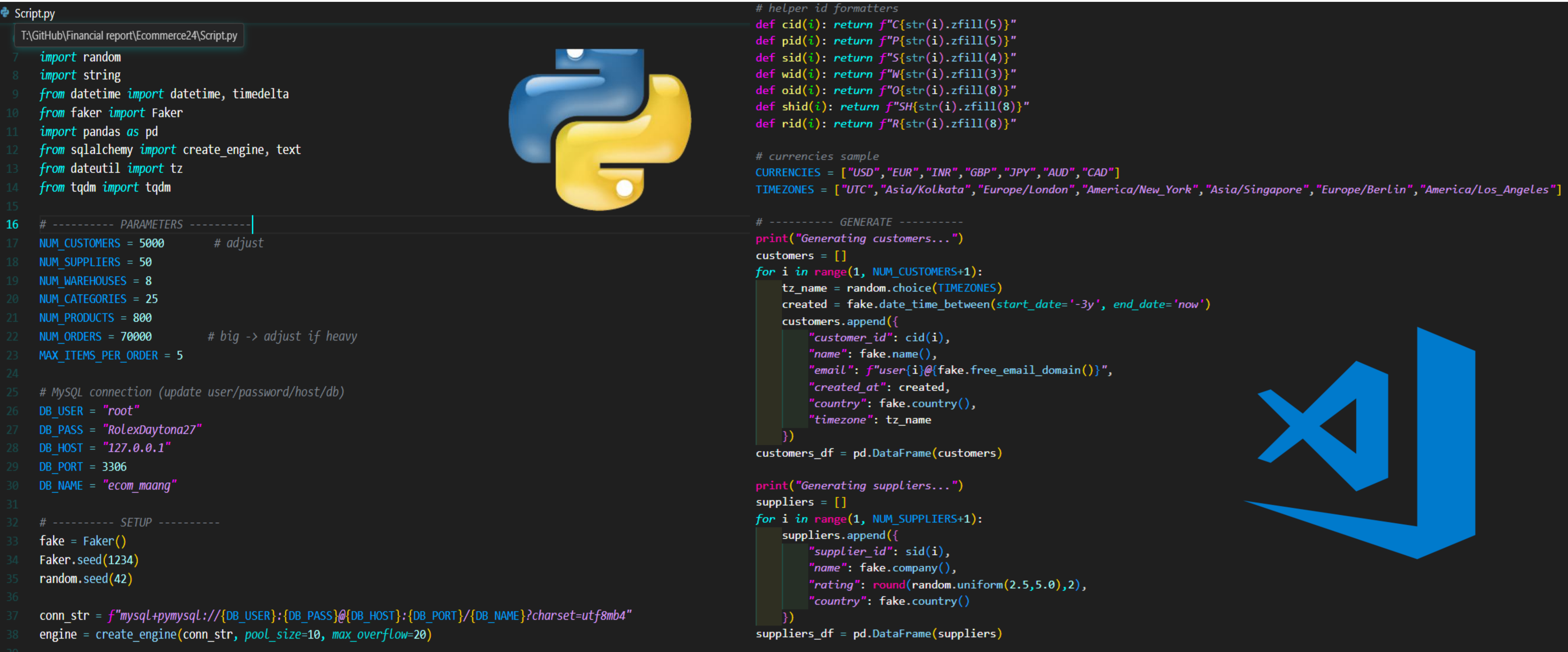
Python – Data generation & preprocessing
Azurite – Local Azure Blob Storage for staging CSV files
MySQL – Relational database for structured storage
Power BI – Data visualization & analysis
DAX – Measures & calculated columns for KPIs

Data Flow Architecture :

- **Python Script** → Generates synthetic sales, order, and customer data
- **Azurite Storage** → Stores CSV files in Azure Blob format locally
- **MySQL Database** → Tables for Orders, Customers, Products, Shipments
- **Power BI** → Connects to MySQL, applies transformations, builds visuals



Python Script to Import CSV Files:



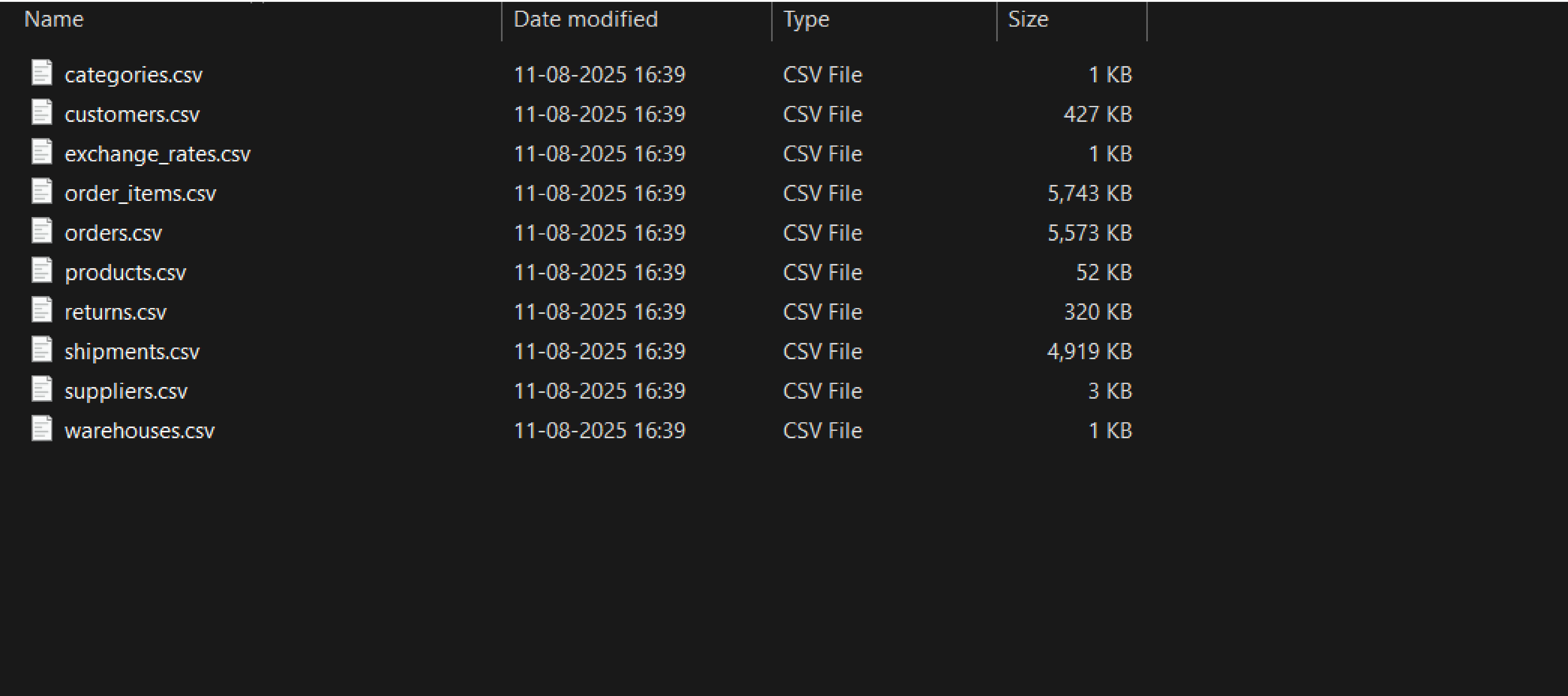
Script.py

T:\GitHub\Financial report\Ecommerce24\Script.py

```python
import random
import string
from datetime import datetime, timedelta
from faker import Faker
import pandas as pd
from sqlalchemy import create_engine, text
from dateutil import tz
from tqdm import tqdm

# ---------- PARAMETERS ----------
NUM_CUSTOMERS = 5000        # adjust
NUM_SUPPLIERS = 50
NUM_WAREHOUSES = 8
NUM_CATEGORIES = 25
NUM_PRODUCTS = 800
NUM_ORDERS = 70000          # big -> adjust if heavy
MAX_ITEMS_PER_ORDER = 5

# MySQL connection (update user/password/host/db)
DB_USER = "root"
DB_PASS = "RolexDaytona27"
DB_HOST = "127.0.0.1"
DB_PORT = 3306
DB_NAME = "ecom_maang"

# ---------- SETUP ----------
fake = Faker()
Faker.seed(1234)
random.seed(42)

conn_str = f"mysql+pymysql://{DB_USER}:{DB_PASS}@{DB_HOST}:{DB_PORT}/{DB_NAME}?charset=utf8mb4"
engine = create_engine(conn_str, pool_size=10, max_overflow=20)
```

```python
# helper id formatters
def cid(i): return f"C{str(i).zfill(5)}"
def pid(i): return f"P{str(i).zfill(5)}"
def sid(i): return f"S{str(i).zfill(4)}"
def wid(i): return f"W{str(i).zfill(3)}"
def oid(i): return f"O{str(i).zfill(8)}"
def shid(i): return f"SH{str(i).zfill(8)}"
def rid(i): return f"R{str(i).zfill(8)}"

# currencies sample
CURRENCIES = ["USD","EUR","INR","GBP","JPY","AUD","CAD"]
TIMEZONES = ["UTC","Asia/Kolkata","Europe/London","America/New_York","Asia/Singapore","Europe/Berlin","America/Los_Angeles"]

# ---------- GENERATE ----------
print("Generating customers...")
customers = []
for i in range(1, NUM_CUSTOMERS+1):
    tz_name = random.choice(TIMEZONES)
    created = fake.date_time_between(start_date='-3y', end_date='now')
    customers.append({
        "customer_id": cid(i),
        "name": fake.name(),
        "email": f"user{i}@{fake.free_email_domain()}",
        "created_at": created,
        "country": fake.country(),
        "timezone": tz_name
    })
customers_df = pd.DataFrame(customers)

print("Generating suppliers...")
suppliers = []
for i in range(1, NUM_SUPPLIERS+1):
    suppliers.append({
        "supplier_id": sid(i),
        "name": fake.company(),
        "rating": round(random.uniform(2.5,5.0),2),
        "country": fake.country()
    })
suppliers_df = pd.DataFrame(suppliers)
```

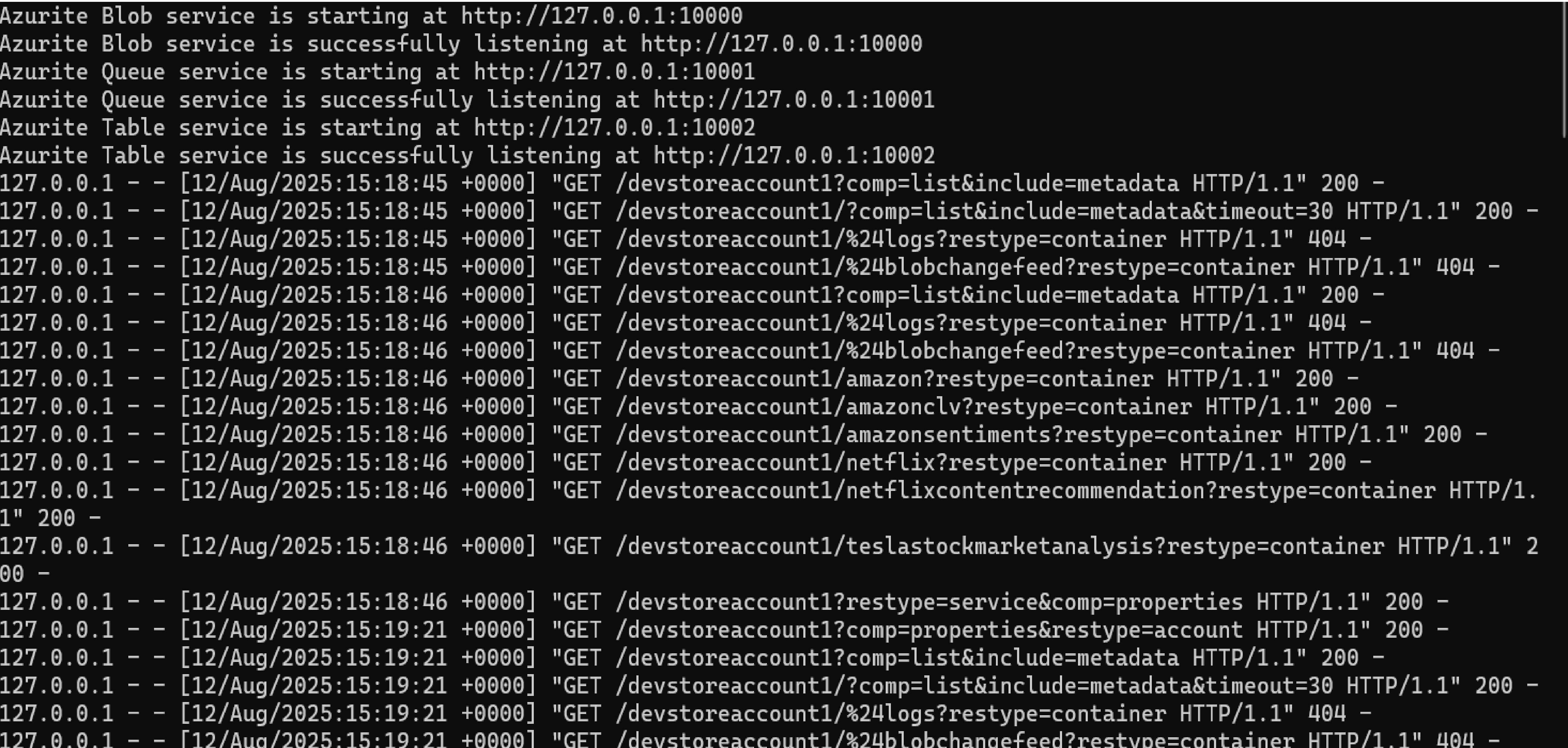
CSV Files after Importing From Python Script:



Name	Date modified	Type	Size
categories.csv	11-08-2025 16:39	CSV File	1 KB
customers.csv	11-08-2025 16:39	CSV File	427 KB
exchange_rates.csv	11-08-2025 16:39	CSV File	1 KB
order_items.csv	11-08-2025 16:39	CSV File	5,743 KB
orders.csv	11-08-2025 16:39	CSV File	5,573 KB
products.csv	11-08-2025 16:39	CSV File	52 KB
returns.csv	11-08-2025 16:39	CSV File	320 KB
shipments.csv	11-08-2025 16:39	CSV File	4,919 KB
suppliers.csv	11-08-2025 16:39	CSV File	3 KB
warehouses.csv	11-08-2025 16:39	CSV File	1 KB

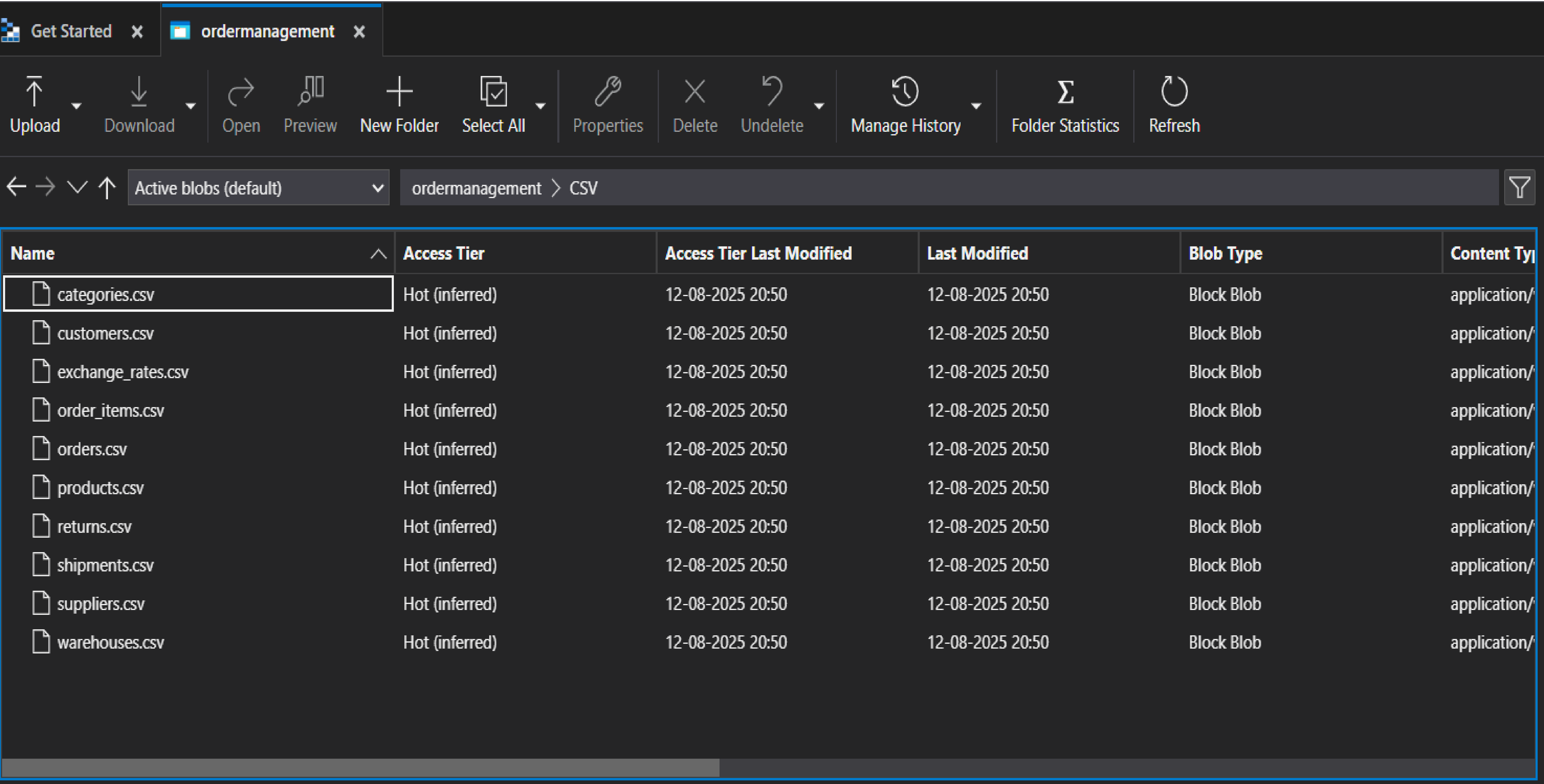
Uploaded CSV File to Azure Blob Storage:

Azurite to establish connection



```
Azurite Blob service is starting at http://127.0.0.1:10000
Azurite Blob service is successfully listening at http://127.0.0.1:10000
Azurite Queue service is starting at http://127.0.0.1:10001
Azurite Queue service is successfully listening at http://127.0.0.1:10001
Azurite Table service is starting at http://127.0.0.1:10002
Azurite Table service is successfully listening at http://127.0.0.1:10002
127.0.0.1 - - [12/Aug/2025:15:18:45 +0000] "GET /devstoreaccount1?comp=list&include=metadata HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2025:15:18:45 +0000] "GET /devstoreaccount1/?comp=list&include=metadata&timeout=30 HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2025:15:18:45 +0000] "GET /devstoreaccount1/%24logs?restype=container HTTP/1.1" 404 -
127.0.0.1 - - [12/Aug/2025:15:18:45 +0000] "GET /devstoreaccount1/%24blobchangefeed?restype=container HTTP/1.1" 404 -
127.0.0.1 - - [12/Aug/2025:15:18:46 +0000] "GET /devstoreaccount1?comp=list&include=metadata HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2025:15:18:46 +0000] "GET /devstoreaccount1/%24logs?restype=container HTTP/1.1" 404 -
127.0.0.1 - - [12/Aug/2025:15:18:46 +0000] "GET /devstoreaccount1/%24blobchangefeed?restype=container HTTP/1.1" 404 -
127.0.0.1 - - [12/Aug/2025:15:18:46 +0000] "GET /devstoreaccount1/amazon?restype=container HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2025:15:18:46 +0000] "GET /devstoreaccount1/amazonclv?restype=container HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2025:15:18:46 +0000] "GET /devstoreaccount1/amazonsentiments?restype=container HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2025:15:18:46 +0000] "GET /devstoreaccount1/netflix?restype=container HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2025:15:18:46 +0000] "GET /devstoreaccount1/netflixcontentrecommendation?restype=container HTTP/1.
1" 200 -
127.0.0.1 - - [12/Aug/2025:15:18:46 +0000] "GET /devstoreaccount1/teslastockmarketanalysis?restype=container HTTP/1.1" 2
00 -
127.0.0.1 - - [12/Aug/2025:15:18:46 +0000] "GET /devstoreaccount1?restype=service&comp=properties HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2025:15:19:21 +0000] "GET /devstoreaccount1?comp=properties&restype=account HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2025:15:19:21 +0000] "GET /devstoreaccount1?comp=list&include=metadata HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2025:15:19:21 +0000] "GET /devstoreaccount1/?comp=list&include=metadata&timeout=30 HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2025:15:19:21 +0000] "GET /devstoreaccount1/%24logs?restype=container HTTP/1.1" 404 -
127.0.0.1 - - [12/Aug/2025:15:19:21 +0000] "GET /devstoreaccount1/%24blobchangefeed?restype=container HTTP/1.1" 404 -
```

Azure Blob Storage(Microsoft)



Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Content Typ
categories.csv	Hot (inferred)	12-08-2025 20:50	12-08-2025 20:50	Block Blob	application/
customers.csv	Hot (inferred)	12-08-2025 20:50	12-08-2025 20:50	Block Blob	application/
exchange_rates.csv	Hot (inferred)	12-08-2025 20:50	12-08-2025 20:50	Block Blob	application/
order_items.csv	Hot (inferred)	12-08-2025 20:50	12-08-2025 20:50	Block Blob	application/
orders.csv	Hot (inferred)	12-08-2025 20:50	12-08-2025 20:50	Block Blob	application/
products.csv	Hot (inferred)	12-08-2025 20:50	12-08-2025 20:50	Block Blob	application/
returns.csv	Hot (inferred)	12-08-2025 20:50	12-08-2025 20:50	Block Blob	application/
shipments.csv	Hot (inferred)	12-08-2025 20:50	12-08-2025 20:50	Block Blob	application/
suppliers.csv	Hot (inferred)	12-08-2025 20:50	12-08-2025 20:50	Block Blob	application/
warehouses.csv	Hot (inferred)	12-08-2025 20:50	12-08-2025 20:50	Block Blob	application/

MySQL Database:

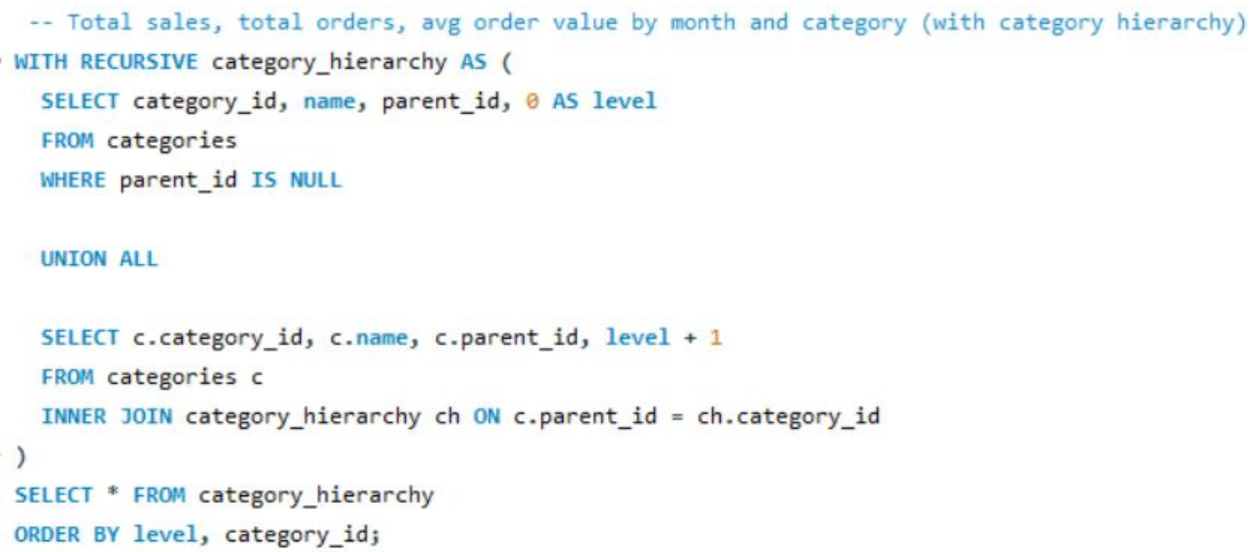
Schema design with **primary & foreign keys**.

- **Normalized structure** (1NF → 3NF).
- Data loaded from CSV into tables.

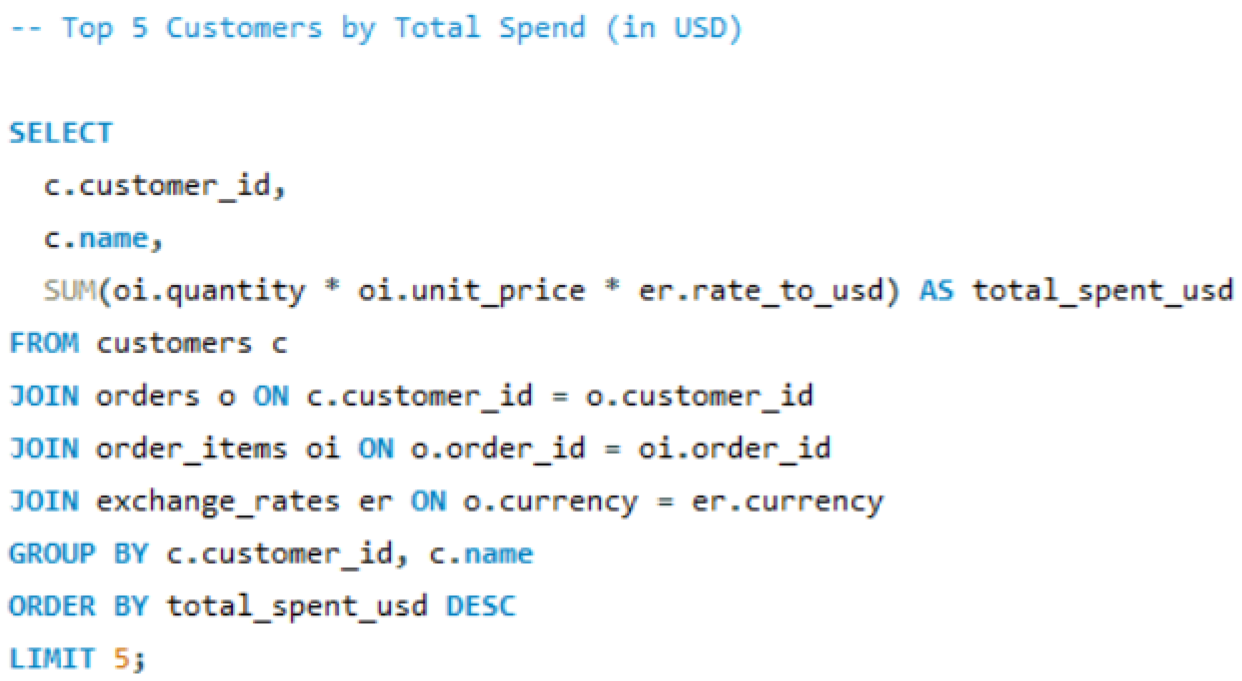
MySQL Role in Detail

- **Database Design**
  - **Tables**: customers, products, orders, order_items, shipments.
  - **Relationships**: Foreign keys ensure data integrity between orders, customers, and products.
- **Data Import Challenges**
  - Fixed **date formats** to match MySQL datetime requirements.
  - Disabled & re-enabled foreign key checks for bulk insert efficiency.
- **SQL Queries for Power BI**
  - Aggregations for **total sales, delayed shipments, top customers**.
  - Joins for combining order, shipment, and customer details.
- **Benefits**
  - Fast retrieval for Power BI.
  - Structured relationships → accurate drill-through in reports.

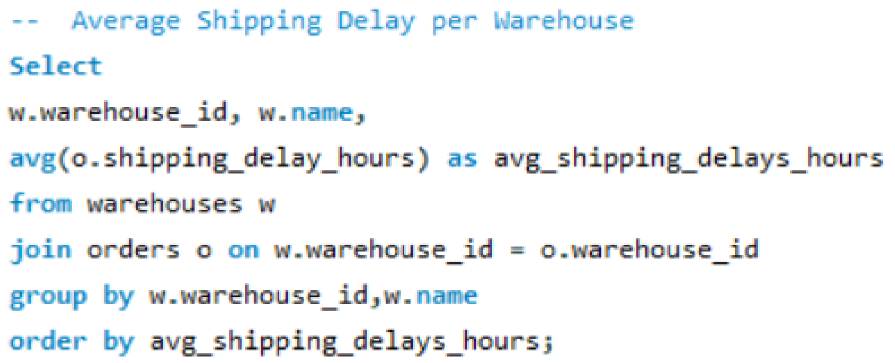
SQL Queries:



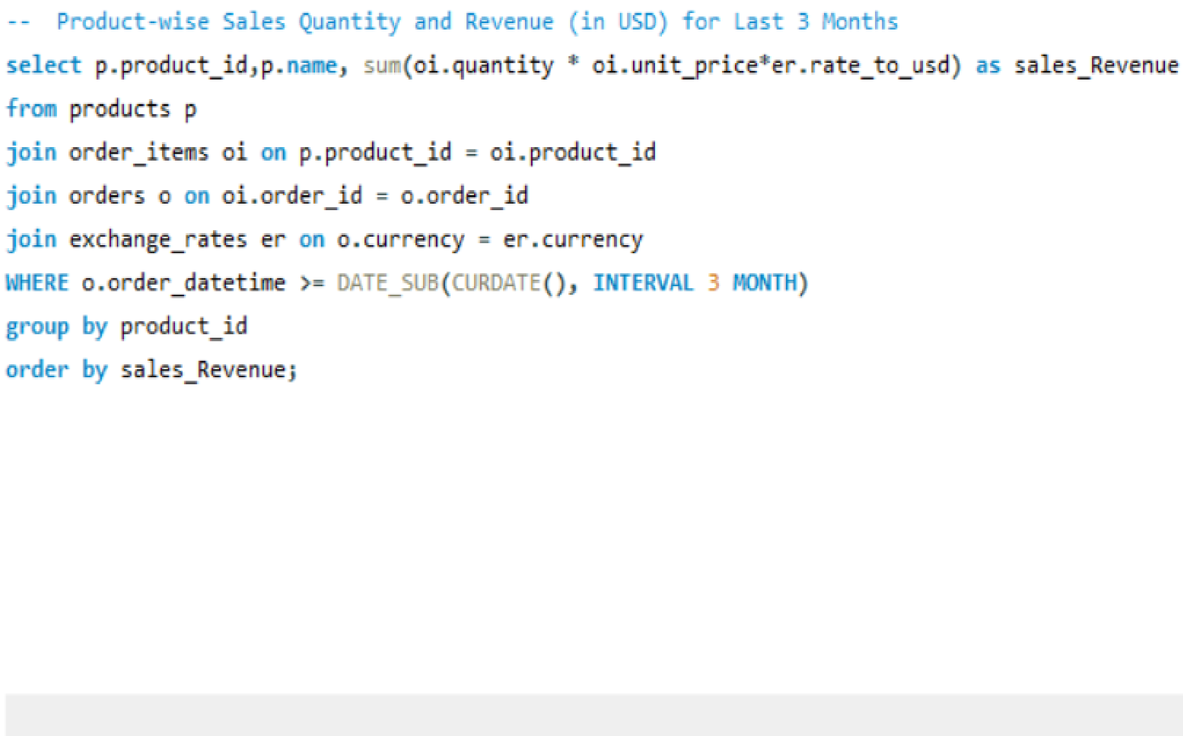
```
-- Total sales, total orders, avg order value by month and category (with category hierarchy)
WITH RECURSIVE category_hierarchy AS (
  SELECT category_id, name, parent_id, 0 AS level
  FROM categories
  WHERE parent_id IS NULL

  UNION ALL

  SELECT c.category_id, c.name, c.parent_id, level + 1
  FROM categories c
  INNER JOIN category_hierarchy ch ON c.parent_id = ch.category_id
)
SELECT * FROM category_hierarchy
ORDER BY level, category_id;
```



```
-- Top 5 Customers by Total Spend (in USD)


SELECT
   c.customer_id,
   c.name,
   SUM(oi.quantity * oi.unit_price * er.rate_to_usd) AS total_spent_usd
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
JOIN order_items oi ON o.order_id = oi.order_id
JOIN exchange_rates er ON o.currency = er.currency
GROUP BY c.customer_id, c.name
ORDER BY total_spent_usd DESC
LIMIT 5;
```

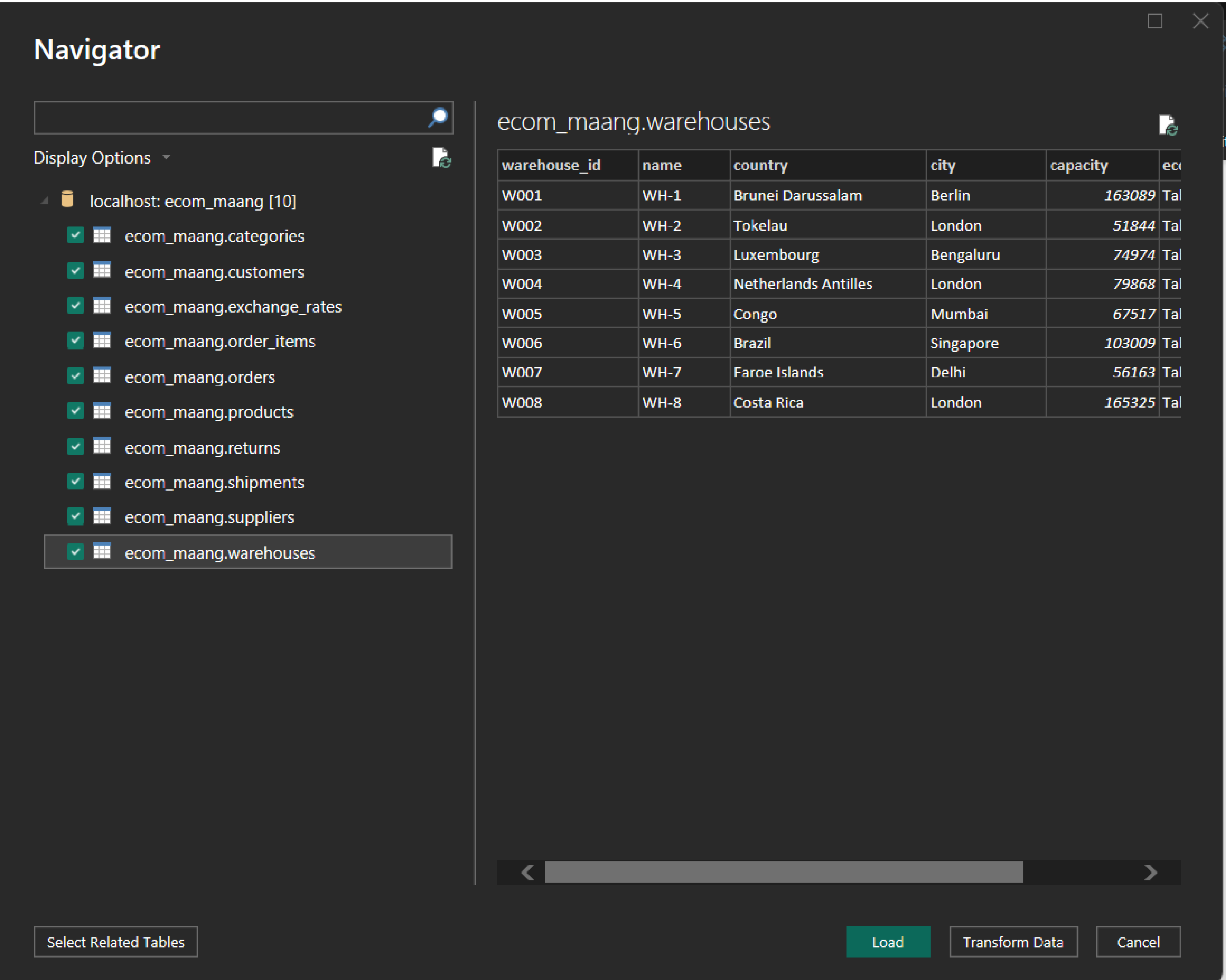


```
--  Average Shipping Delay per Warehouse
Select
w.warehouse_id, w.name,
avg(o.shipping_delay_hours) as avg_shipping_delays_hours
from warehouses w
join orders o on w.warehouse_id = o.warehouse_id
group by w.warehouse_id,w.name
order by avg_shipping_delays_hours;
```



```
--  Product-wise Sales Quantity and Revenue (in USD) for Last 3 Months
select p.product_id,p.name, sum(oi.quantity * oi.unit_price*er.rate_to_usd) as sales_Revenue
from products p
join order_items oi on p.product_id = oi.product_id
join orders o on oi.order_id = o.order_id
join exchange_rates er on o.currency = er.currency
WHERE o.order_datetime >= DATE_SUB(CURDATE(), INTERVAL 3 MONTH)
group by product_id
order by sales_Revenue;
```

Data Import From MySQL to Power BI:



Navigator

Display Options

localhost: ecom_maang [10]

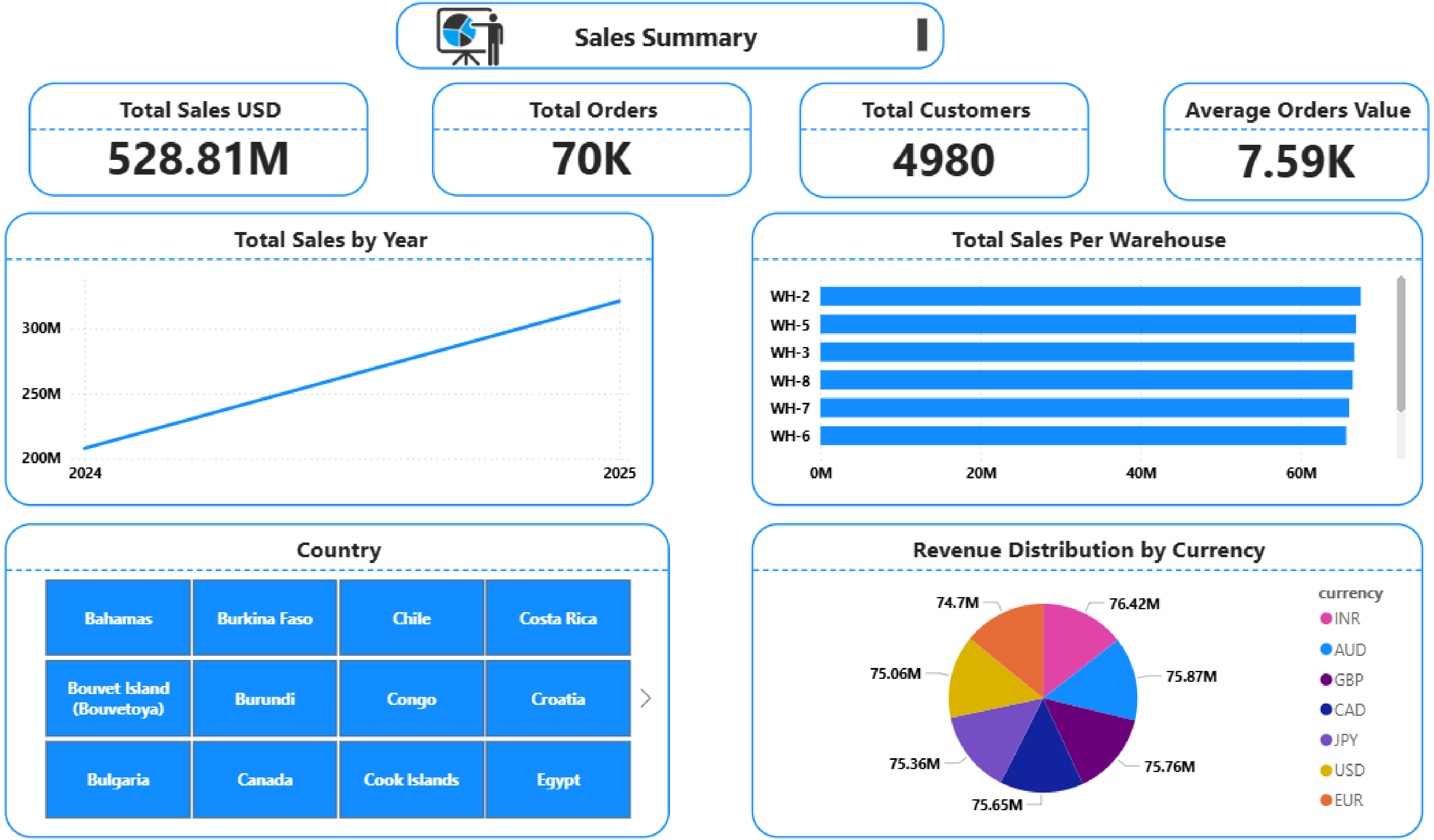
- ☑ ecom_maang.categories
- ☑ ecom_maang.customers
- ☑ ecom_maang.exchange_rates
- ☑ ecom_maang.order_items
- ☑ ecom_maang.orders
- ☑ ecom_maang.products
- ☑ ecom_maang.returns
- ☑ ecom_maang.shipments
- ☑ ecom_maang.suppliers
- ☑ ecom_maang.warehouses

ecom_maang.warehouses

warehouse_id	name	country	city	capacity	ec
W001	WH-1	Brunei Darussalam	Berlin	163089	Tal
W002	WH-2	Tokelau	London	51844	Tal
W003	WH-3	Luxembourg	Bengaluru	74974	Tal
W004	WH-4	Netherlands Antilles	London	79868	Tal
W005	WH-5	Congo	Mumbai	67517	Tal
W006	WH-6	Brazil	Singapore	103009	Tal
W007	WH-7	Faroe Islands	Delhi	56163	Tal
W008	WH-8	Costa Rica	London	165325	Tal

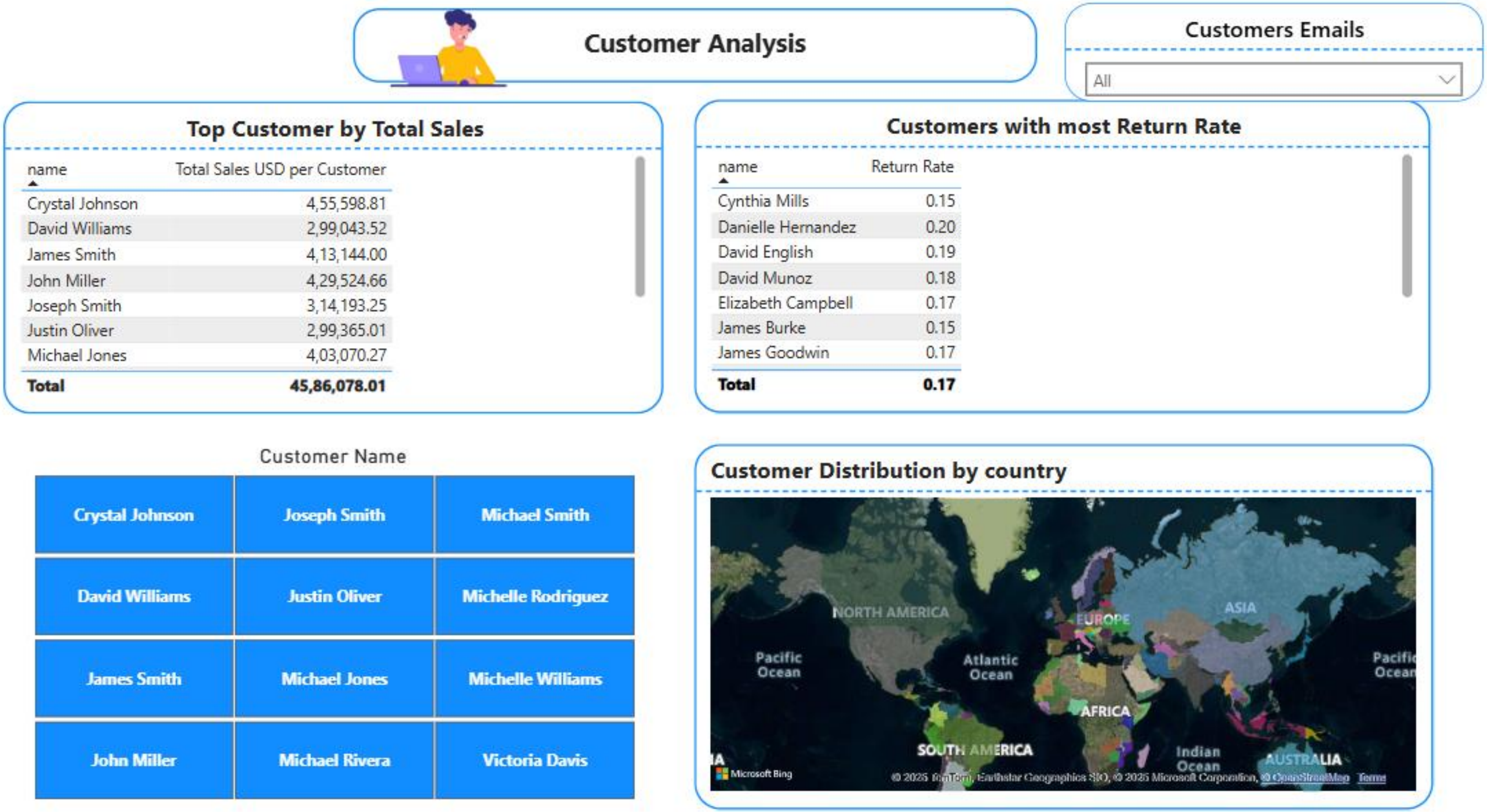
Dashboard Pages Overview

- **Sales Overview** – Revenue trends, category breakdown.
- **Order Status Tracking** – Delivered vs. delayed shipments.
- **Customer Insights** – Top customers, revenue share, drill-through.



Sales Summary

Total Sales USD	Total Orders	Total Customers	Average Orders Value
528.81M	70K	4980	7.59K



Customer Analysis

Customers Emails

All

Top Customer by Total Sales

name	Total Sales USD per Customer
Crystal Johnson	4,55,598.81
David Williams	2,99,043.52
James Smith	4,13,144.00
John Miller	4,29,524.66
Joseph Smith	3,14,193.25
Justin Oliver	2,99,365.01
Michael Jones	4,03,070.27
Total	**45,86,078.01**

Customers with most Return Rate

name	Return Rate
Cynthia Mills	0.15
Danielle Hernandez	0.20
David English	0.19
David Munoz	0.18
Elizabeth Campbell	0.17
James Burke	0.15
James Goodwin	0.17
Total	**0.17**

Customer Name

Crystal Johnson	Joseph Smith	Michael Smith
David Williams	Justin Oliver	Michelle Rodriguez
James Smith	Michael Jones	Michelle Williams
John Miller	Michael Rivera	Victoria Davis

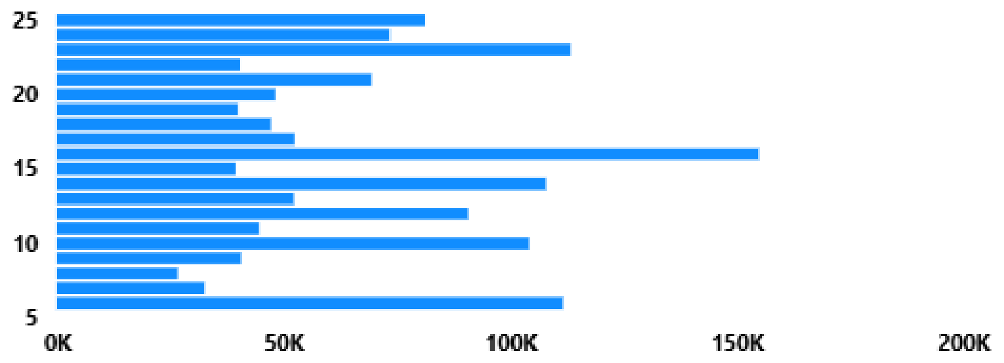
Customer Distribution by country

Product Trend

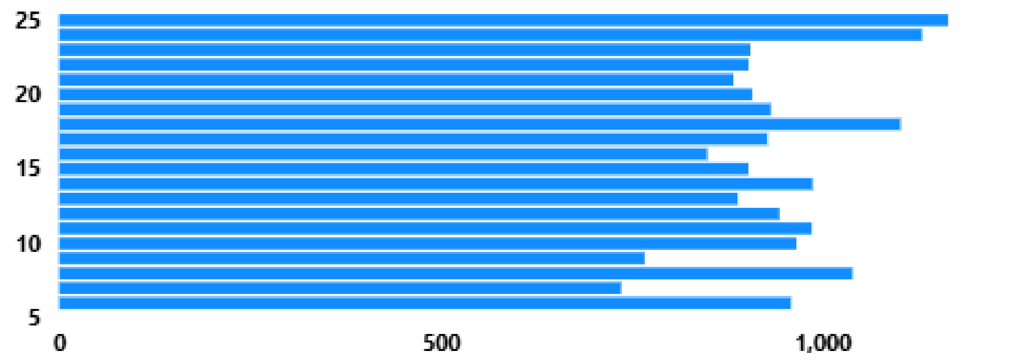
Order Date

04-07-2024 11-08-2025

Product Sales by Category



Avg_price by Category



Price of Products

8.59	22.56	48.29	77.92	86.10	103.34	107.36
14.31	29.66	48.72	78.26	92.34	104.02	113.03
14.64	30.44	54.70	79.74	92.48	104.91	113.86
15.91	37.52	57.04	83.61	94.01	105.66	119.64
18.21	45.36	76.48	84.31	101.05	106.20	121.31

Best Selling Products

product_id	Total Quantity Sold	Total Revenue USD
P00090	19	17,739.23
P00094	24	18,558.01
P00121	19	27,446.73
P00123	19	19,853.06
P00140	21	25,010.34
P00187	26	30,174.79
P00188	20	28,202.49
P00326	20	12,270.98
P00358	28	33,735.17
Total	**279**	**3,00,075.81**

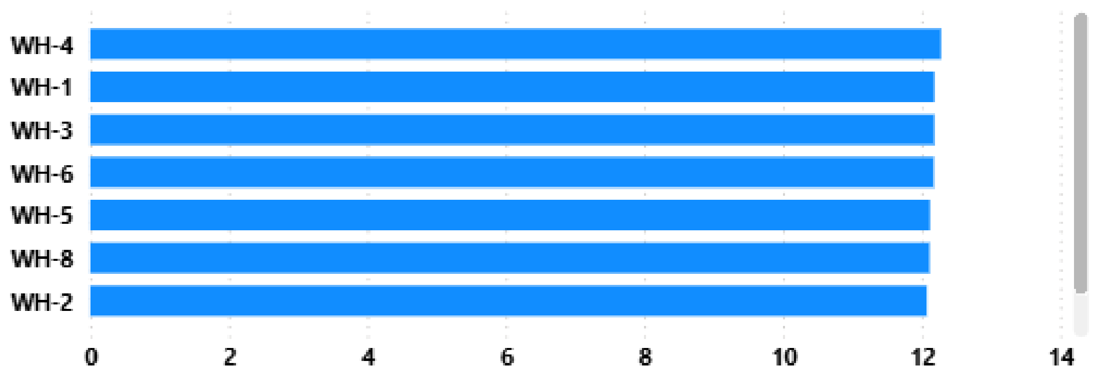
Status

□ delivered

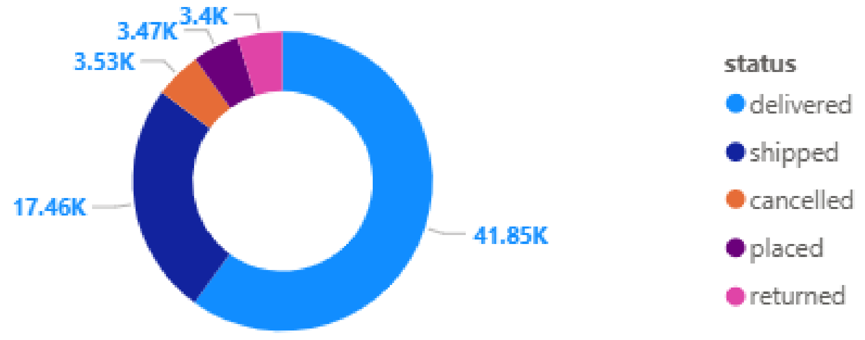
□ returned

Shipping & Orders Performance Page

Average Shipping Delay by Warehouse



Order Status Breakdwon



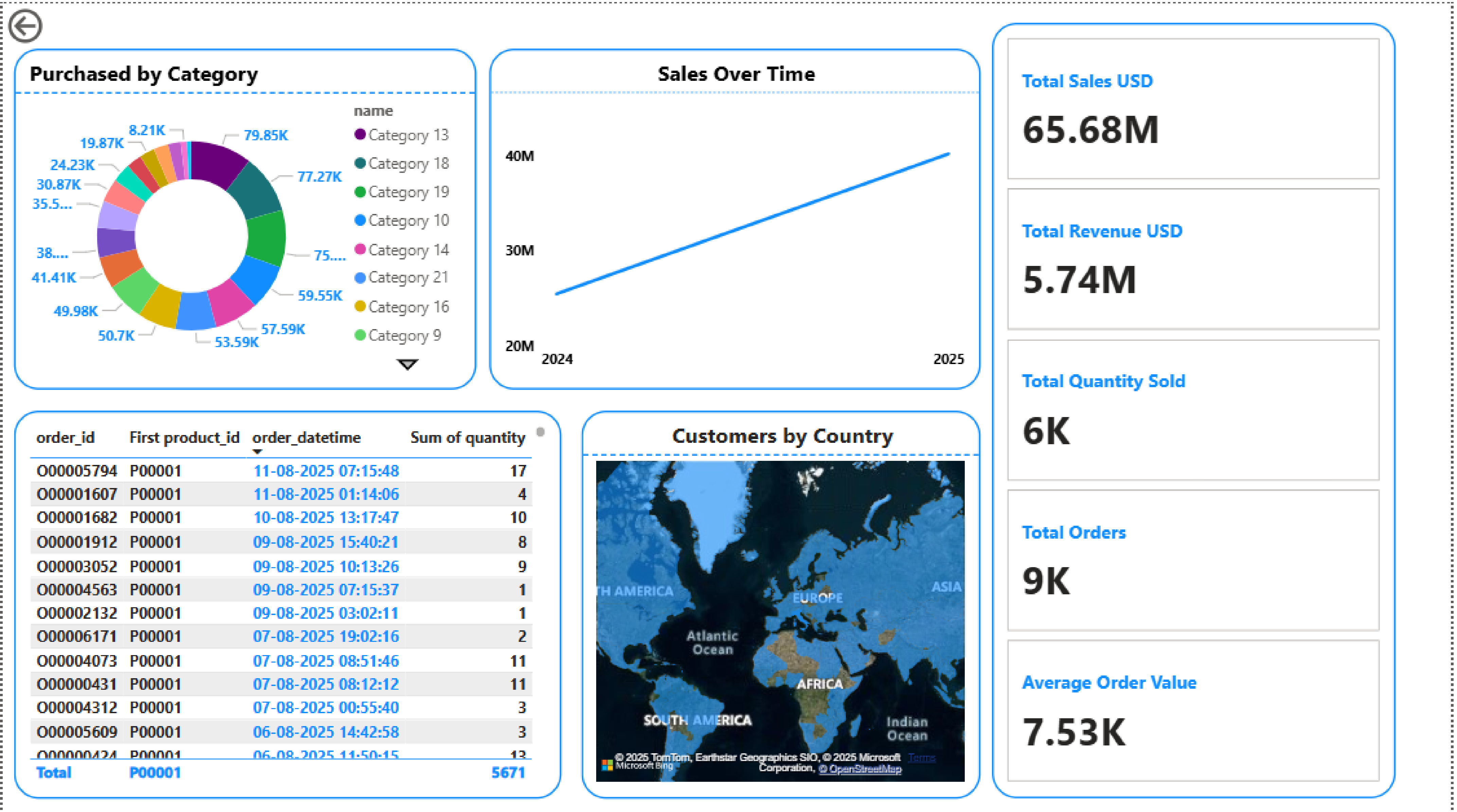
Carrier

BlueDart	IndiaPost
DHL	ShipRocket
FedEx	UPS

return_reason	Sum of refund_amount
Buyer remorse	1,08,17,245.33
Damaged	1,82,90,677.42
Not as described	1,68,68,710.41
Wrong item	1,56,62,709.34
Total	**6,16,39,342.50**

Drill-Through Page – Customer Details

- Order history, shipment timelines, product purchase trends.
- Auto-filtered from main pages for deep analysis.



order_id	First product_id	order_datetime	Sum of quantity
O00005794	P00001	11-08-2025 07:15:48	17
O00001607	P00001	11-08-2025 01:14:06	4
O00001682	P00001	10-08-2025 13:17:47	10
O00001912	P00001	09-08-2025 15:40:21	8
O00003052	P00001	09-08-2025 10:13:26	9
O00004563	P00001	09-08-2025 07:15:37	1
O00002132	P00001	09-08-2025 03:02:11	1
O00006171	P00001	07-08-2025 19:02:16	2
O00004073	P00001	07-08-2025 08:51:46	11
O00000431	P00001	07-08-2025 08:12:12	11
O00004312	P00001	07-08-2025 00:55:40	3
O00005609	P00001	06-08-2025 14:42:58	3
O00000424	P00001	06-08-2025 11:50:15	13
Total	**P00001**		**5671**

Total Sales USD
65.68M

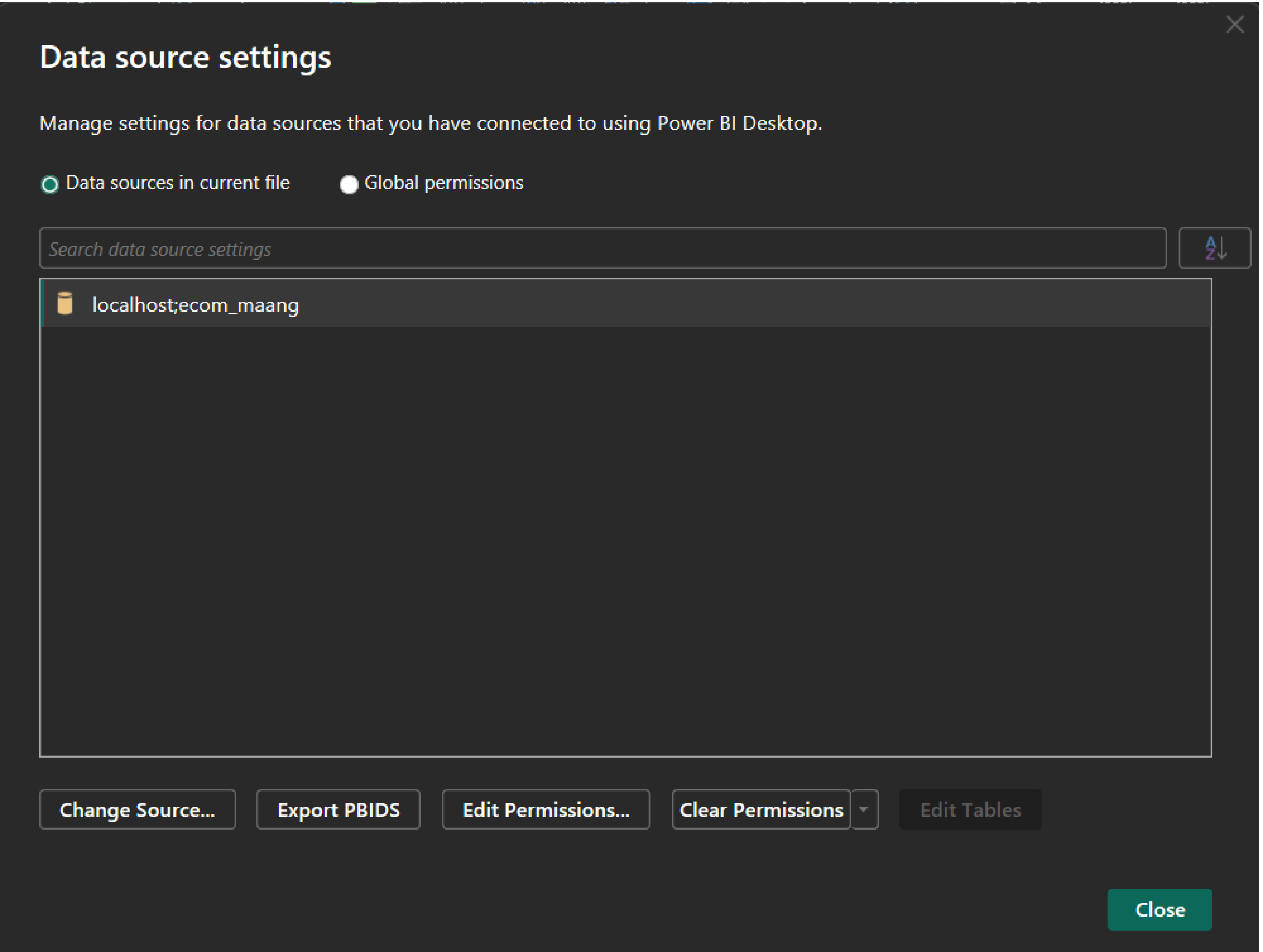
Total Revenue USD
5.74M

Total Quantity Sold
6K

Total Orders
9K

Average Order Value
7.53K

MySQL to Power BI for Dynamic Dashboard:



Data source settings

Manage settings for data sources that you have connected to using Power BI Desktop.

● Data sources in current file ○ Global permissions

localhost;ecom_maang

Key Insights:

- Shipment delays ka regional breakdown
- High-value customers ka top category
- Delivery time ka before-after comparison
- Repeat purchase rate

Data Preparation Challenges

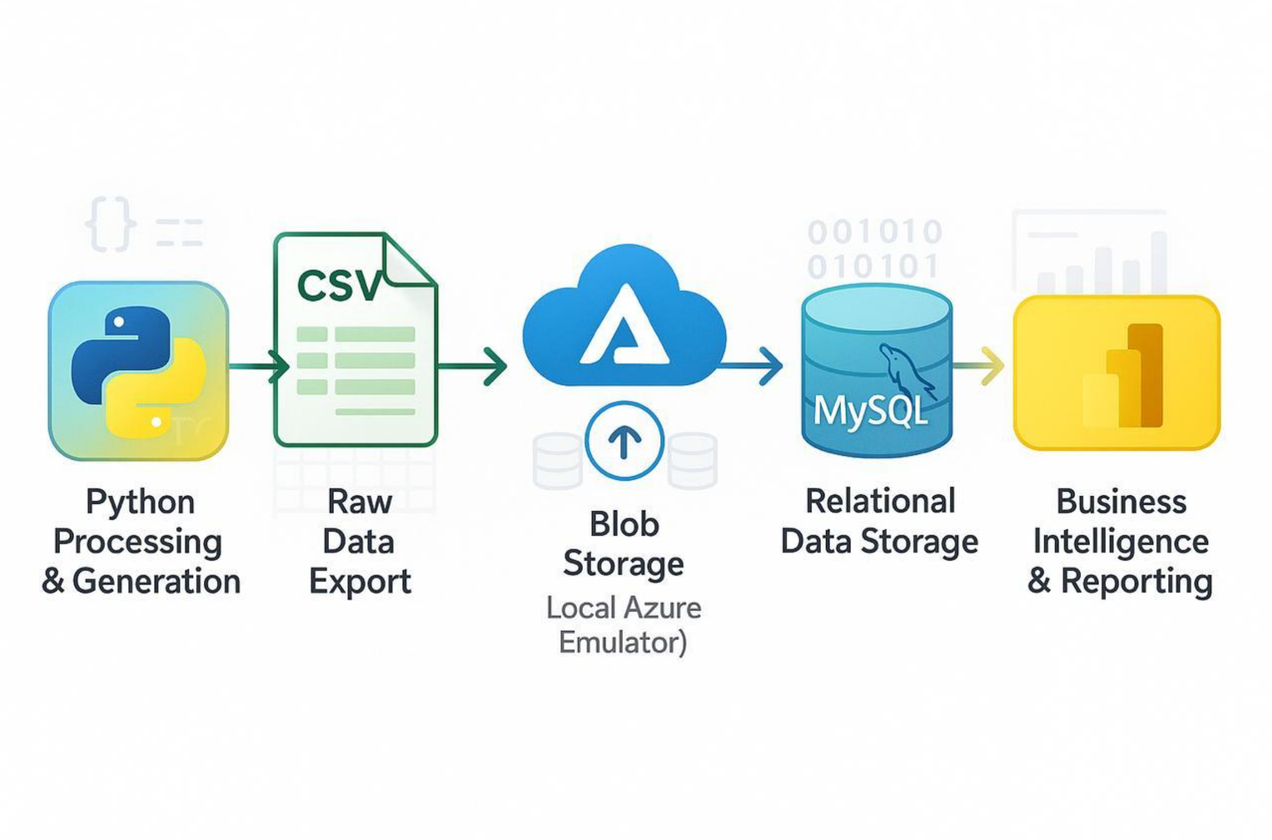
- Cleaning synthetic data before insert.
- Handling **foreign key constraint errors**.
- Large CSV import optimization using LOAD DATA INFILE.
- Maintaining referential integrity after bulk load.

Business Impact

- Faster decisions with single source of truth.
- Improved customer segmentation for marketing.
- Better operational efficiency from delay tracking.

Conclusion & Next Steps

- Move from Azurite to real Azure Blob Storage for cloud scalability.
- Automate ETL pipeline with Python scripts scheduled daily.
- Add predictive models for sales forecasting.



Thank You/Let’s Connect:

LinkedIn: https://www.linkedin.com/in/tanmay-sharma-800599373/

Github: https://github.com/Tanu272004/E2E-Sales-Customer-Analytics-Dashboard-End-to-End-pipeline-with-Python-MySQL-Power-BI-.git