Got it ✅

Let’s finalize everything for Dashboard 1 (Global Sales Overview).

I’ll break it down in 4 parts for you:

1. Charts & KPIs in this dashboard.

2. SQL queries for Power BI exports (small summary tables).

3. SQL practice queries (basic → advanced) for Viva.

4. DAX measures you can add directly in Power BI.

---

📊 Dashboard 1: Global Sales Overview

✅ KPIs (top row in Power BI)

Total Revenue → SUM of Total\_Amount

Total Orders → COUNT of Transaction\_ID

Unique Customers → DISTINCT COUNT of Customer\_ID

📈 Charts

1. Bar Chart: Top 5 Cities by Sales (City vs Total\_Sales)

2. Line Chart: Monthly Revenue Trend (Year-Month vs Total\_Amount)

3. Donut Chart: Order Status Share (Order\_Status vs Count)

4. Map: Revenue by Country (Country vs Total\_Sales)

5. Table: Country-wise sales summary

🎛 Filters → Year, Month, Country, Segment

---

🗂 SQL Queries for Power BI Exports

We’ll use SQLite3 + Pandas (serverless) to generate summary CSVs.

import sqlite3

import pandas as pd

# Step 1: Connect to SQLite

conn = sqlite3.connect("retail\_analytics.db")

# Step 2: Load dataset

df\_time = pd.read\_csv("df\_time\_cleaned.csv")

df\_time.to\_sql("RetailData", conn, if\_exists="replace", index=False)

# ---------- EXPORT QUERIES ----------

# Export 1: City-level Sales (for Bar Chart + Map)

query\_city\_sales = """

SELECT City, State, Country, SUM(Total\_Amount) AS Total\_Sales

FROM RetailData

GROUP BY City, State, Country

ORDER BY Total\_Sales DESC;

"""

df\_city\_sales = pd.read\_sql\_query(query\_city\_sales, conn)

df\_city\_sales.to\_csv("output\_city\_sales.csv", index=False)

# Export 2: Monthly Revenue Trend (for Line Chart)

query\_monthly\_sales = """

SELECT Year, Month, SUM(Total\_Amount) AS Monthly\_Revenue

FROM RetailData

GROUP BY Year, Month

ORDER BY Year, Month;

"""

df\_monthly\_sales = pd.read\_sql\_query(query\_monthly\_sales, conn)

df\_monthly\_sales.to\_csv("output\_monthly\_sales.csv", index=False)

# Export 3: Order Status Distribution (for Donut Chart)

query\_order\_status = """

SELECT Order\_Status, COUNT(\*) AS Order\_Count

FROM RetailData

GROUP BY Order\_Status;

"""

df\_order\_status = pd.read\_sql\_query(query\_order\_status, conn)

df\_order\_status.to\_csv("output\_order\_status.csv", index=False)

# Export 4: Country-level Sales (for Map + Table)

query\_country\_sales = """

SELECT Country, SUM(Total\_Amount) AS Total\_Sales

FROM RetailData

GROUP BY Country

ORDER BY Total\_Sales DESC;

"""

df\_country\_sales = pd.read\_sql\_query(query\_country\_sales, conn)

df\_country\_sales.to\_csv("output\_country\_sales.csv", index=False)

print("✅ CSVs exported for Dashboard 1")

Generated CSVs to import into Power BI:

output\_city\_sales.csv

output\_monthly\_sales.csv

output\_order\_status.csv

output\_country\_sales.csv

---

🧑‍💻 SQL Practice Queries (for Viva)

Here’s a set of queries you can show (basic → advanced).

🔹 Beginner

-- Count total transactions

SELECT COUNT(\*) FROM RetailData;

-- Find unique customers

SELECT COUNT(DISTINCT Customer\_ID) FROM RetailData;

-- Total revenue

SELECT SUM(Total\_Amount) FROM RetailData;

🔹 Intermediate

-- Top 5 cities by revenue

SELECT City, SUM(Total\_Amount) AS Sales

FROM RetailData

GROUP BY City

ORDER BY Sales DESC

LIMIT 5;

-- Monthly average revenue

SELECT Year, Month, AVG(Total\_Amount) AS Avg\_Sales

FROM RetailData

GROUP BY Year, Month

ORDER BY Year, Month;

🔹 Advanced

-- Revenue per customer (LTV style)

SELECT Customer\_ID, SUM(Total\_Amount) AS Customer\_LTV

FROM RetailData

GROUP BY Customer\_ID

ORDER BY Customer\_LTV DESC

LIMIT 10;

-- Churn proxy: customers with no orders after Dec 2023

SELECT Customer\_ID

FROM RetailData

GROUP BY Customer\_ID

HAVING MAX(Date) < '2023-12-31';

---

🧮 DAX Measures in Power BI

When you load your fact table (df\_time\_cleaned.csv), you can add these DAX formulas:

-- Total Revenue

Total\_Revenue = SUM(RetailData[Total\_Amount])

-- Total Orders

Total\_Orders = COUNT(RetailData[Transaction\_ID])

-- Unique Customers

Unique\_Customers = DISTINCTCOUNT(RetailData[Customer\_ID])

-- Avg Revenue per Order

Avg\_Revenue\_Per\_Order = DIVIDE([Total\_Revenue], [Total\_Orders])

-- Revenue Growth % (Month over Month)

Revenue\_Growth =

VAR CurrentMonth = [Total\_Revenue]

VAR PrevMonth = CALCULATE([Total\_Revenue], DATEADD(RetailData[Date], -1, MONTH))

RETURN DIVIDE(CurrentMonth - PrevMonth, PrevMonth, 0)

---

✅ So for Dashboard 1, you’ll have:

5 visuals + 3 KPI cards

4 SQL export queries for Power BI

6–8 SQL practice queries for Viva

5 useful DAX measures in Power BI

This is more than enough to impress examiners because you are covering Python → SQL → Power BI → DAX in one dashboard itself.

---\

👉 Do you want me to do the same full detailing (charts + SQL exports + Viva queries + DAX) for Dashboard 2 (Customer Segmentation) next?

Got it ✅ Thanks for clarifying that you’re running Jupyter Notebook inside VS Code. That means we can use a mix of pandas + sqlite3 for exports and also keep some pure SQL queries for showcasing skills.

Here’s the ready-to-use notebook setup for Dashboard 1 (Global Sales Overview):

---

🔹 Step 1: Setup and Load Dataset

import sqlite3

import pandas as pd

# Load the cleaned dataset

df = pd.read\_csv(r"C:\Users\Mukesh\Desktop\SEM 7 PROJECT PROPOSALS\df\_time\_cleaned.csv")

# Create SQLite connection

conn = sqlite3.connect("retail\_analytics.db")

# Store dataframe into SQL table

df.to\_sql("RetailData", conn, if\_exists="replace", index=False)

print("✅ Data loaded into SQLite successfully!")

---

🔹 Step 2: SQL Queries for Viva (Showcasing Skills)

Beginner Queries

# Count total transactions

pd.read\_sql\_query("SELECT COUNT(\*) AS Total\_Transactions FROM RetailData;", conn)

# Find unique customers

pd.read\_sql\_query("SELECT COUNT(DISTINCT Customer\_ID) AS Unique\_Customers FROM RetailData;", conn)

# Total revenue

pd.read\_sql\_query("SELECT SUM(Total\_Amount) AS Total\_Revenue FROM RetailData;", conn)

Intermediate Queries

# Top 5 cities by revenue

pd.read\_sql\_query("""

SELECT City, SUM(Total\_Amount) AS Sales

FROM RetailData

GROUP BY City

ORDER BY Sales DESC

LIMIT 5;

""", conn)

# Monthly average revenue

pd.read\_sql\_query("""

SELECT Year, Month, AVG(Total\_Amount) AS Avg\_Sales

FROM RetailData

GROUP BY Year, Month

ORDER BY Year, Month;

""", conn)

Advanced Queries

# Revenue per customer (LTV style)

pd.read\_sql\_query("""

SELECT Customer\_ID, SUM(Total\_Amount) AS Customer\_LTV

FROM RetailData

GROUP BY Customer\_ID

ORDER BY Customer\_LTV DESC

LIMIT 10;

""", conn)

# Churn proxy: Customers inactive after Dec 2023

pd.read\_sql\_query("""

SELECT Customer\_ID

FROM RetailData

GROUP BY Customer\_ID

HAVING MAX(Date) < '2023-12-31';

""", conn)

---

🔹 Step 3: Export Queries for Power BI

These queries will create small CSVs (summary tables) that we’ll import into Power BI.

# Export 1: City-level Sales (for Bar Chart + Map)

query\_city\_sales = """

SELECT City, State, Country, SUM(Total\_Amount) AS Total\_Sales

FROM RetailData

GROUP BY City, State, Country

ORDER BY Total\_Sales DESC;

"""

df\_city\_sales = pd.read\_sql\_query(query\_city\_sales, conn)

df\_city\_sales.to\_csv("output\_city\_sales.csv", index=False)

# Export 2: Monthly Revenue Trend (for Line Chart)

query\_monthly\_sales = """

SELECT Year, Month, SUM(Total\_Amount) AS Monthly\_Revenue

FROM RetailData

GROUP BY Year, Month

ORDER BY Year, Month;

"""

df\_monthly\_sales = pd.read\_sql\_query(query\_monthly\_sales, conn)

df\_monthly\_sales.to\_csv("output\_monthly\_sales.csv", index=False)

# Export 3: Order Status Distribution (for Donut Chart)

query\_order\_status = """

SELECT Order\_Status, COUNT(\*) AS Order\_Count

FROM RetailData

GROUP BY Order\_Status;

"""

df\_order\_status = pd.read\_sql\_query(query\_order\_status, conn)

df\_order\_status.to\_csv("output\_order\_status.csv", index=False)

# Export 4: Country-level Sales (for Map + Table)

query\_country\_sales = """

SELECT Country, SUM(Total\_Amount) AS Total\_Sales

FROM RetailData

GROUP BY Country

ORDER BY Total\_Sales DESC;

"""

df\_country\_sales = pd.read\_sql\_query(query\_country\_sales, conn)

df\_country\_sales.to\_csv("output\_country\_sales.csv", index=False)

print("✅ Exported CSVs for Power BI: city, monthly, order status, country")

---

📂 Output files generated in your working directory:

output\_city\_sales.csv

output\_monthly\_sales.csv

output\_order\_status.csv

output\_country\_sales.csv

---

✅ Now you have:

1. SQL queries (basic → advanced) to show in viva.

2. Export queries for Power BI dashboard building.

Would you like me to now also list the exact visuals + slicers you’ll add in Dashboard 1 (with these exported files) so you can directly start building in Power BI?