1. Create Database

-- =========================================

CREATE DATABASE sales\_analysis;

USE sales\_analysis;

-- =========================================

-- 2. Create Table

-- Keep Transaction\_Date as VARCHAR (raw format: DD-MM-YYYY)

-- Add a real DATE column for analysis

-- =========================================

DROP TABLE IF EXISTS sales\_data;

CREATE TABLE sales\_data (

s\_no INT,

CustomerID INT,

Gender VARCHAR(10),

Location VARCHAR(100),

Tenure\_Months INT,

Transaction\_ID INT,

Transaction\_Date VARCHAR(20), -- raw DD-MM-YYYY format

Transaction\_Date\_dt DATE, -- converted proper DATE

Product\_SKU VARCHAR(50),

Product\_Description TEXT,

Product\_Category VARCHAR(100),

Quantity INT,

Avg\_Price DECIMAL(10,2),

Delivery\_Charges DECIMAL(10,2),

Coupon\_Status VARCHAR(50),

GST DECIMAL(10,2),

Date VARCHAR(20),

Offline\_Spend DECIMAL(10,2),

Online\_Spend DECIMAL(10,2),

Month INT,

Coupon\_Code VARCHAR(50),

Discount\_pct DECIMAL(5,2)

);

-- ✅ NOTE: Insert or load your data here using INSERTs or LOAD DATA INFILE

-- Example:

-- LOAD DATA INFILE '/path/to/file.csv'

-- INTO TABLE sales\_data

-- FIELDS TERMINATED BY ','

-- OPTIONALLY ENCLOSED BY '"'

-- LINES TERMINATED BY '\n'

-- IGNORE 1 ROWS;

-- =========================================

-- 3. Convert Transaction\_Date (DD-MM-YYYY → DATE)

-- Run this after inserting/loading data

-- =========================================

UPDATE sales\_data

SET Transaction\_Date\_dt = STR\_TO\_DATE(Transaction\_Date, '%d-%m-%Y')

WHERE Transaction\_Date IS NOT NULL AND TRIM(Transaction\_Date) <> '';

-- Optional: Index for faster queries

CREATE INDEX idx\_txn\_date\_dt ON sales\_data (Transaction\_Date\_dt);

-- =========================================

-- 4. Monthly Revenue and Order Volume Analysis

-- =========================================

SELECT

DATE\_FORMAT(Transaction\_Date\_dt, '%Y-%m') AS year\_month,

YEAR(Transaction\_Date\_dt) AS year,

MONTH(Transaction\_Date\_dt) AS month\_num,

SUM(Quantity \* Avg\_Price + Delivery\_Charges + GST) AS total\_revenue,

COUNT(DISTINCT Transaction\_ID) AS order\_count

FROM sales\_data

WHERE Transaction\_Date\_dt IS NOT NULL

GROUP BY YEAR(Transaction\_Date\_dt), MONTH(Transaction\_Date\_dt)

ORDER BY YEAR(Transaction\_Date\_dt), MONTH(Transaction\_Date\_dt);

-- =========================================

-- 5. Top Performing Months by Revenue

-- =========================================

SELECT

DATE\_FORMAT(Transaction\_Date\_dt, '%Y-%m') AS year\_month,

SUM(Quantity \* Avg\_Price + Delivery\_Charges + GST) AS total\_revenue,

COUNT(DISTINCT Transaction\_ID) AS order\_count

FROM sales\_data

WHERE Transaction\_Date\_dt IS NOT NULL

GROUP BY YEAR(Transaction\_Date\_dt), MONTH(Transaction\_Date\_dt)

ORDER BY total\_revenue DESC

LIMIT 10;

-- =========================================

-- 6. Customer Buying Patterns

-- =========================================

-- 6a. Average Order Value (AOV) per Month

SELECT

DATE\_FORMAT(Transaction\_Date\_dt, '%Y-%m') AS year\_month,

AVG(order\_revenue) AS avg\_order\_value

FROM (

SELECT

Transaction\_ID,

DATE\_FORMAT(Transaction\_Date\_dt, '%Y-%m') AS ym,

SUM(Quantity \* Avg\_Price + Delivery\_Charges + GST) AS order\_revenue

FROM sales\_data

WHERE Transaction\_Date\_dt IS NOT NULL

GROUP BY Transaction\_ID, DATE\_FORMAT(Transaction\_Date\_dt, '%Y-%m')

) AS sub

GROUP BY ym

ORDER BY ym;

-- 6b. Repeat Purchase Rate (Customers with >1 order in a month)

SELECT

DATE\_FORMAT(Transaction\_Date\_dt, '%Y-%m') AS year\_month,

COUNT(DISTINCT CASE WHEN order\_count > 1 THEN CustomerID END) AS repeat\_customers,

COUNT(DISTINCT CustomerID) AS total\_customers,

ROUND(

COUNT(DISTINCT CASE WHEN order\_count > 1 THEN CustomerID END) / COUNT(DISTINCT CustomerID) \* 100,

2

) AS repeat\_rate\_pct

FROM (

SELECT

CustomerID,

DATE\_FORMAT(Transaction\_Date\_dt, '%Y-%m') AS ym,

COUNT(DISTINCT Transaction\_ID) AS order\_count

FROM sales\_data

WHERE Transaction\_Date\_dt IS NOT NULL

GROUP BY CustomerID, DATE\_FORMAT(Transaction\_Date\_dt, '%Y-%m')

) AS sub

GROUP BY ym

ORDER BY ym;

-- 6c. Monthly Orders per Customer (Engagement Trend)

SELECT

DATE\_FORMAT(Transaction\_Date\_dt, '%Y-%m') AS year\_month,

AVG(order\_count) AS avg\_orders\_per\_customer

FROM (

SELECT

CustomerID,

DATE\_FORMAT(Transaction\_Date\_dt, '%Y-%m') AS ym,

COUNT(DISTINCT Transaction\_ID) AS order\_count

FROM sales\_data

WHERE Transaction\_Date\_dt IS NOT NULL

GROUP BY CustomerID, DATE\_FORMAT(Transaction\_Date\_dt, '%Y-%m')

) AS sub

GROUP BY ym

ORDER BY ym;

-- =========================================

-- 7. Top 5 Most Loyal Customers (Highest Orders)

-- =========================================

SELECT

CustomerID,

COUNT(DISTINCT Transaction\_ID) AS total\_orders,

SUM(Quantity \* Avg\_Price + Delivery\_Charges + GST) AS total\_spent

FROM sales\_data

WHERE Transaction\_Date\_dt IS NOT NULL

GROUP BY CustomerID

ORDER BY total\_orders DESC, total\_spent DESC

LIMIT 5;

-- =========================================

-- 8. Top 5 High-Value Customers (Biggest Spenders)

-- =========================================

SELECT

CustomerID,

SUM(Quantity \* Avg\_Price + Delivery\_Charges + GST) AS total\_spent,

COUNT(DISTINCT Transaction\_ID) AS total\_orders

FROM sales\_data

WHERE Transaction\_Date\_dt IS NOT NULL

GROUP BY CustomerID

ORDER BY total\_spent DESC, total\_orders DESC

LIMIT 5;