SQL QUERIES

CREATE DATABASE csa

CREATE TABLE customer\_transactions (

InvoiceNo TEXT,

StockCode TEXT,

Description TEXT,

Quantity INT,

InvoiceDate TIMESTAMP,

UnitPrice NUMERIC(10,2),

CustomerID INT,

Country TEXT,

TotalPrice NUMERIC(12,2),

Year INT,

Month INT,

Day INT

);

DROP TABLE customer\_transactions;

CREATE TABLE customer\_transactions (

InvoiceNo TEXT,

StockCode TEXT,

Description TEXT,

Quantity INT,

InvoiceDate TIMESTAMP,

UnitPrice NUMERIC(10,2),

CustomerID INT,

Country TEXT,

TotalPrice NUMERIC(12,2),

Year INT,

Month INT,

Day INT

);

COPY customer\_transactions

FROM 'E:\PROJECTS\Power Bi\Portfolio Project\Customer Segmentation Analysis\customer\_segmentation\_data.csv'

DELIMITER ','

CSV HEADER;

SELECT \* FROM customer\_transactions LIMIT 10;

SELECT Quantity, UnitPrice, (Quantity \* UnitPrice) AS CalculatedTotalPrice, TotalPrice

FROM customer\_transactions

LIMIT 10;

SELECT CustomerID, MAX(InvoiceDate) AS LastPurchaseDate

FROM customer\_transactions

GROUP BY CustomerID;

RFM ANALYSIS

SELECT CustomerID, COUNT(DISTINCT InvoiceNo) AS PurchaseFrequency

FROM customer\_transactions

GROUP BY CustomerID

ORDER BY PurchaseFrequency DESC;

SELECT CustomerID, SUM(TotalPrice) AS TotalSpent

FROM customer\_transactions

GROUP BY CustomerID

ORDER BY TotalSpent DESC;

WITH rfm AS (

-- Calculate Recency, Frequency, and Monetary Value

SELECT

CustomerID,

CURRENT\_DATE - MAX(InvoiceDate) AS Recency, -- Days since last purchase

COUNT(DISTINCT InvoiceNo) AS Frequency, -- Number of purchases

SUM(TotalPrice) AS Monetary -- Total spending

FROM customer\_transactions

WHERE CustomerID IS NOT NULL -- Remove NULL CustomerIDs

GROUP BY CustomerID

)

SELECT \*,

NTILE(4) OVER (ORDER BY Recency DESC) AS R\_Score, -- Higher Recency = lower score

NTILE(4) OVER (ORDER BY Frequency ASC) AS F\_Score, -- Higher Frequency = higher score

NTILE(4) OVER (ORDER BY Monetary ASC) AS M\_Score -- Higher Monetary = higher score

FROM rfm;

WITH rfm AS (

SELECT

CustomerID,

CURRENT\_DATE - MAX(InvoiceDate) AS Recency,

COUNT(DISTINCT InvoiceNo) AS Frequency,

SUM(TotalPrice) AS Monetary

FROM customer\_transactions

WHERE CustomerID IS NOT NULL

GROUP BY CustomerID

),

rfm\_scores AS (

SELECT \*,

NTILE(4) OVER (ORDER BY Recency DESC) AS R\_Score,

NTILE(4) OVER (ORDER BY Frequency ASC) AS F\_Score,

NTILE(4) OVER (ORDER BY Monetary ASC) AS M\_Score

FROM rfm

)

SELECT \*,

CASE

WHEN R\_Score = 4 AND F\_Score = 4 AND M\_Score = 4 THEN 'VIP'

WHEN R\_Score >= 3 AND F\_Score >= 3 THEN 'Loyal Customer'

WHEN R\_Score = 4 THEN 'New Customer'

WHEN F\_Score = 1 AND M\_Score = 1 THEN 'Lost Customer'

ELSE 'Regular Customer'

END AS Customer\_Segment

FROM rfm\_scores;

DROP TABLE IF EXISTS rfm\_scores;

CREATE TABLE rfm\_scores AS

WITH rfm AS (

SELECT

CustomerID,

CURRENT\_DATE - MAX(InvoiceDate) AS Recency, -- Days since last purchase

COUNT(DISTINCT InvoiceNo) AS Frequency, -- Number of purchases

SUM(TotalPrice) AS Monetary -- Total spending

FROM customer\_transactions

WHERE CustomerID IS NOT NULL

GROUP BY CustomerID

),

rfm\_scores\_calc AS (

SELECT \*,

NTILE(4) OVER (ORDER BY Recency DESC) AS R\_Score, -- 4 = least recent, 1 = most recent

NTILE(4) OVER (ORDER BY Frequency ASC) AS F\_Score, -- 4 = most frequent, 1 = least frequent

NTILE(4) OVER (ORDER BY Monetary ASC) AS M\_Score -- 4 = highest spender, 1 = lowest

FROM rfm

)

SELECT \*,

CASE

WHEN R\_Score = 4 AND F\_Score = 4 AND M\_Score = 4 THEN 'VIP'

WHEN R\_Score >= 3 AND F\_Score >= 3 THEN 'Loyal Customer'

WHEN R\_Score = 4 THEN 'New Customer'

WHEN F\_Score = 1 AND M\_Score = 1 THEN 'Lost Customer'

ELSE 'Regular Customer'

END AS Customer\_Segment

FROM rfm\_scores\_calc;

SELECT \* FROM rfm\_scores LIMIT 10;