**Insights from Walmart Sales Analysis**

**1] Top performing Branch**

→A is the top performing branch with Average Growth Rate of 1.6570.

**2] Most Profitable Customer Type**

→ Member customers tend to generate more profit overall compared to Normal customers. And, they have best product line as observed – ‘Food and Beverages’.

**3] Best Performing Product Line**

→ The product line with the highest total sales is typically Health and Beauty for Branch A, Food and Beverages for Branch B, and Home and Lifestyle for Branch C.

**4] Most Profitable Day of the Week**

→ Saturday, Tuesdays, and Thursdays often see the highest total profits, making them ideal for promotions and marketing.

**5] Top-Performing City / Branch**

→ The branch in Yangon usually contributes the most to overall profitability, followed by Mandalay or Naypyitaw.

**6] Customer Spending Tiers**

→ Most customers fall into the ‘Medium‘ spending category, Low is quite less in count and High is just 1.

**7] Anomalies in sales transactions.**

→We see High anomalies throughout the data, using the z-score method.

**8] Most Popular Payment Method by City.**

→ Most Popular Payment Method for ‘Mandalay’ is observed as Ewallet.

For ‘Naypyitaw’ it is observed to be Cash and for ‘Yangon’ it is Ewallet.

**9] Best Product Line by Customer Type**

→ Members prefer Health and Beauty, while Normal customers often choose Food and Beverages or Electronic Accessories.

**10] Repeat Purchases**

→ A significantly good number of customers make repeat purchases within 30 days, showing potential for loyalty programs.

**11] Top 5 Customers**

→ The top 5 customers contribute a noticeable set of total sales. These are key targets for exclusive rewards.

**-- Walmart Sales Analysis Report**

**-- Author: [Afsha Ajaz Ahmed Khan]**

**-- Date: [05-04-2025]**

**-- Description: This script contains SQL queries for a comprehensive analysis of Walmart sales data.**

**-- Each section addresses a specific business question.**

/\*-------------------------------- Task 1: Top Branch by Sales Growth Rate ------------------------\*/

WITH Sales\_Growth AS

(

SELECT Branch, DATE\_FORMAT(Date, '%m-%Y') AS Month, SUM(Total)

AS Monthly\_Sales,

LAG(SUM(Total)) OVER (PARTITION BY Branch ORDER BY DATE\_FORMAT(Date, '%m-%Y')) AS Prev\_Month\_Sales

FROM walmartsales

GROUP BY Branch, Month

)

SELECT Branch,

AVG( (Monthly\_Sales - Prev\_Month\_Sales) / Prev\_Month\_Sales ) \* 100 AS Avg\_GrowthRate

FROM Sales\_Growth

GROUP BY Branch

ORDER BY Avg\_GrowthRate DESC

LIMIT 1;

/\*------------------------------- Task 2: Most Profitable Product Line by Branch -------------------\*/

WITH Product\_Profit AS

(

SELECT Branch, Product\_Line, SUM(Gross\_Income - cogs) AS Total\_Profit

FROM walmartsales

GROUP BY Branch, Product\_Line

)

SELECT pp.Branch, pp.Product\_Line, pp.Total\_Profit

FROM Product\_Profit pp

JOIN

(

SELECT Branch, MAX(Total\_Profit) AS Max\_Profit

FROM Product\_Profit

GROUP BY Branch

) pm

ON pp.Branch = pm.Branch AND pp.Total\_Profit = pm.Max\_Profit

ORDER BY pp.Branch;

/\*------------------------ Task 3: Customer Segmentation Based on Spending -------------------\*/

WITH Customer\_Spendings AS

(

SELECT `Customer ID` AS CustomerID, SUM(Total) AS Total\_Spent, AVG(Total) AS Avg\_Spent

FROM walmartsales

GROUP BY `Customer ID`

)

SELECT CustomerID, Total\_Spent, Avg\_Spent,

CASE

WHEN Avg\_Spent < 300 THEN 'Low'

WHEN Avg\_Spent BETWEEN 301 AND 350 THEN 'Medium'

ELSE 'High'

END AS Spending\_Tier

FROM Customer\_Spendings

ORDER BY Avg\_Spent DESC;

/\*---------------------------- Task 4: Anomaly Detection in Sales -----------------------------\*/

WITH ProductLine\_Stats AS (

SELECT Product\_Line, AVG(Total) AS Avg\_Total, STDDEV(Total) AS StdDev\_Total

FROM walmartsales

GROUP BY Product\_Line

),

Sales\_WithZScore AS

(

SELECT w.InvoiceID, w.Product\_Line, w.Total, p.Avg\_Total, p.StdDev\_Total,

(w.Total - p.Avg\_Total) / p.StdDev\_Total AS ZScore

FROM walmartsales w

JOIN ProductLine\_Stats p ON w.Product\_Line = p.Product\_Line

)

SELECT \*,

CASE

WHEN ZScore > 3 THEN 'Extreme High Anomaly'

WHEN ZScore > 2 THEN 'High Anomaly'

WHEN ZScore < -3 THEN 'Extreme Low Anomaly'

WHEN ZScore < -2 THEN 'Low Anomaly'

ELSE 'Normal'

END AS Anomaly\_Category

FROM Sales\_WithZScore

WHERE ABS(ZScore) > 2

ORDER BY ZScore DESC;

/\*------------------------- Task 5: Most Popular Payment Method by City ------------------------\*/

WITH Payment\_Counts AS

(

SELECT City, Payment, COUNT(\*) AS Payment\_Count

FROM walmartsales

GROUP BY City, Payment

),

Ranked\_Payments AS

(

SELECT \*, RANK() OVER (PARTITION BY City ORDER BY Payment\_Count DESC) AS rk

FROM Payment\_Counts

)

SELECT City, Payment AS Most\_Popular\_Payment\_Method, Payment\_Count

FROM Ranked\_Payments

WHERE rk = 1

ORDER BY City;

/\*--------------------------- Task 6: Monthly Sales Distribution by Gender ------------------------\*/

WITH Monthly\_Sales AS

(

SELECT DATE\_FORMAT(Date, '%m-%Y') AS Month, Gender, SUM(Total) AS Total\_Sales

FROM walmartsales

GROUP BY Month, Gender

)

SELECT \* FROM Monthly\_Sales

ORDER BY Month, Gender;

/\*------------------------------ Task 7: Best Product Line by Customer Type -----------------------\*/

WITH ProductLine\_Sales AS

(

SELECT CustomerType, Product\_Line, ROUND(SUM(Total),4) AS Total\_Sales

FROM walmartsales

GROUP BY CustomerType, Product\_Line

),

Ranked\_ProductLines AS

(

SELECT \*, RANK() OVER (PARTITION BY CustomerType ORDER BY Total\_Sales DESC) AS rk

FROM ProductLine\_Sales

)

SELECT CustomerType, Product\_Line AS Best\_Product\_Line, Total\_Sales

FROM Ranked\_ProductLines

WHERE rk = 1

ORDER BY CustomerType;

/\*----------------- Task 8: Identifying Repeat Customers (Within 30 Days) ----------------\*/

WITH Customer\_Purchases AS

(

SELECT `Customer ID`, Date AS PurchaseDate,

LEAD(Date) OVER (PARTITION BY `Customer ID` ORDER BY Date) AS Next\_PurchaseDate

FROM walmartsales

)

SELECT `Customer ID`, PurchaseDate, Next\_PurchaseDate, DATEDIFF(Next\_PurchaseDate, PurchaseDate) AS Days\_Between\_Purchases

FROM Customer\_Purchases

WHERE DATEDIFF(Next\_PurchaseDate, PurchaseDate) <= 30

ORDER BY `Customer ID`, PurchaseDate;

/\*------------------------- Task 9: Top 5 Customers by Sales Volume ----------------------------\*/

SELECT `Customer ID`, ROUND(SUM(Total), 2) AS Total\_Sales

FROM walmartsales

GROUP BY `Customer ID`

ORDER BY Total\_Sales DESC

LIMIT 5;

/\*-------------------------- Task 10: Sales Trends by Day of the Week ---------------------------\*/

SELECT DAYNAME(Date) AS Day\_of\_Week, ROUND(SUM(Total), 2) AS Total\_Sales

FROM walmartsales

GROUP BY Day\_of\_Week

ORDER BY Total\_Sales DESC;