WALMART SALES ANALYSIS USING MYSQL TECHNIQUES

SQL Queries And Business Insights

Name : RITU JAWALKAR
DATA SCIENTIST

INTRODUCTION

- Walmart operates across multiple cities and offering a wide range of products.
- The dataset includes transaction data, customer demographic, product lines, sales figure and payment method.
- The project uses MySQL to analyzesales performance, customer behavior, operational efficiencies and trends.

PROBLEM STATEMENT OF WALMARTSALES

- How sales are changing over time (sales trends)
- Different types of customers and their buying habits (customer segmentation)
- Which products make the most profit (product profitability)

DATASET DESCRIPTION

Dataset includes:

- Branch – wise sales data
- Customer purchase behaviour
- Payment methods and product lines
- Sales volume and transaction trends

Source: Walmart Sales Dataset

1

Task 1st : Identifying the Top Branch by Sales Growth Rate

Walmart wants to find out which store has increased its sales the most over time.

We will check the total sales for each store every month and compare their growth rates to see which one is doing the best.

Step 1st Extract Monthly sales Data – First, I calculate total sales for each branch group by month.

INTPUT :

```sql
SELECT
    Branch,
   Date_format(STR_To_Date(Date, '%d-%m-%Y'), '%Y-%m') AS Sales_month,
    round(SUM(Total),2) AS Monthly_Sales
FROM WalmartSales
GROUP BY Branch, Sales_month
Order by Branch, Sales_month;
```

Branch	Sales_month	Monthly_Sales
A	2019-01	38681.19
A	2019-02	29860.2
A	2019-03	37659.18
B	2019-01	37176.16
B	2019-02	34424.39
B	2019-03	34597.45
C	2019-01	40434.76
C	2019-02	32934.99
C	2019-03	37199.11

Step 2nd : Calculate Month-over-Month Growth

INTPUT :

```
SELECT Branch, Sales_Month,
Monthly_Sales, Monthly_Sales - LAG (Monthly_sales)
OVER (Partition BY Branch ORDER BY Sales_Month) AS Sales_Growth
FROM (SELECT Branch, date_format(STR_TO_DATE(Date, '%d-%m-%Y'), '%Y-%m') AS Sales_month,
Round(SUM(Total),2) AS Monthly_sales From Walmartsales group by Branch, Sales_Month) AS Monthly_Data;
```

OUTPUT :

Branch	Sales_Month	Monthly_Sales	Sales_Growth
A	2019-01	38681.19	NULL
A	2019-02	29860.2	-8820.990000000002
A	2019-03	37659.18	7798.98
B	2019-01	37176.16	NULL
B	2019-02	34424.39	-2751.770000000004
B	2019-03	34597.45	173.05999999999767
C	2019-01	40434.76	NULL
C	2019-02	32934.99	-7499.770000000004
C	2019-03	37199.11	4264.120000000003

Step 3rd : Identify the top Branch

INTPUT :

```sql
SELECT Branch, MAX(Sales_Growth) AS MAx_growth
FROM (Select Branch, Sales_Month, Monthly_sales, Monthly_sales - LAG (Monthly_Sales)
Over(Partition by Branch ORDER BY Sales_Month) AS Sales_Growth FROM
(Select Branch, date_format(STR_To_Date(Date, '%d-%m-%Y'), '%Y-%m') AS Sales_Month,
round(Sum(Total),2) AS Monthly_sales From walmartsales Group by Branch, Sales_Month)
AS Monthly_data) AS Growth_Data GROUP BY Branch ORDER BY MAX_Growth DESC;
```

OUTPUT :

Branch	AverageGrowthRate
A	1.66
C	-2.8
B	-3.45

INSIGHTS :

The store with the biggest sales growth over time probably keeps customers coming back, runs good promotions, or sells more during certain seasons.
By looking at monthly growth, Walmart can find ways to improve sales in stores that are not doing as well.

Plan: Use the LAG() function to compare each month's sales with the previous month's sales and spot patterns or changes.

2 Task 2nd :

Finding the Most Profitable Product Line for Each Branch

Walmart wants to find out which product category makes the most profit in each store.

The profit margin will be found by subtracting the cost of the products from the total money earned from sales.

Step 1st: Calculate Total income per category

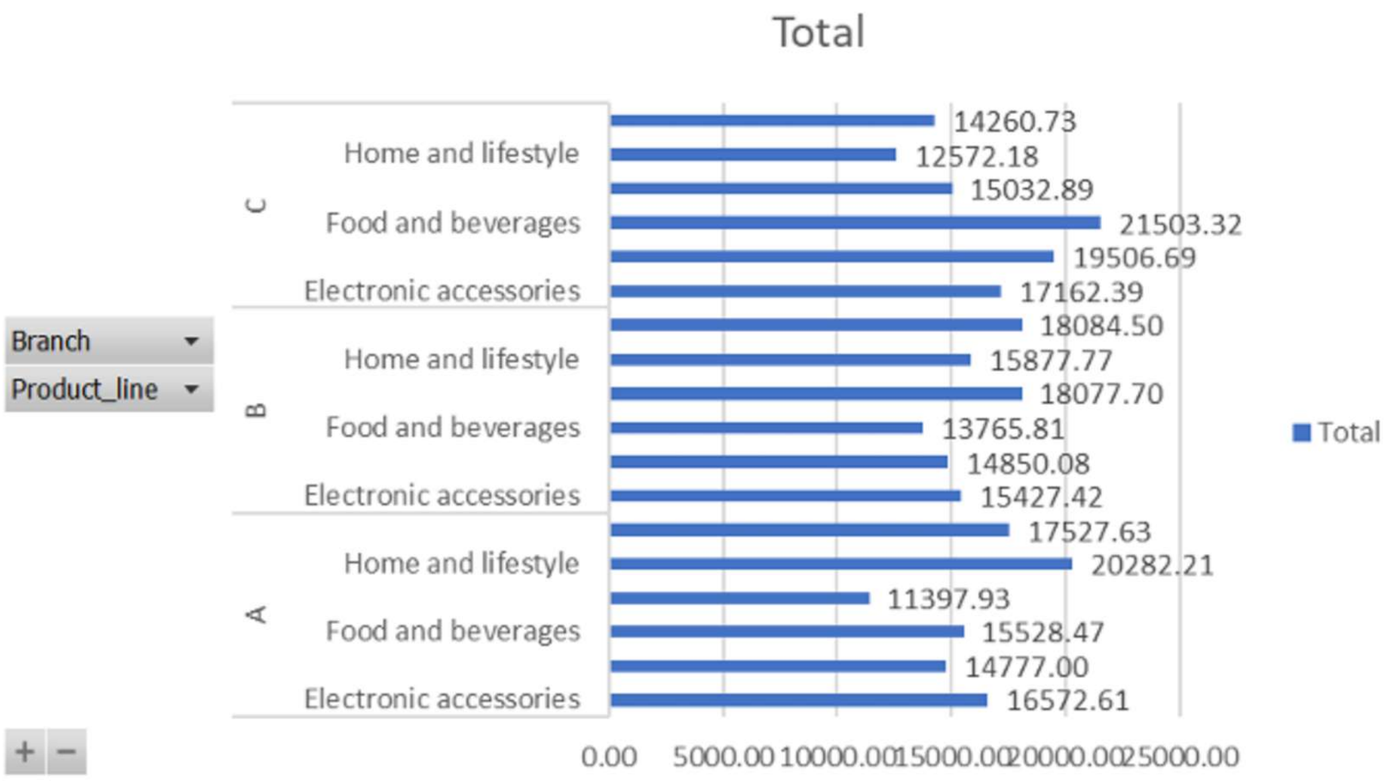
INPUT :

```sql
WITH ProductLineProfit AS (
    -- Calculate the total gross income for each product line per branch, rounded to 2 decimal places
    SELECT
        Branch,
        `Product line` AS ProductLine,
        ROUND(SUM(`gross income`), 2) AS TotalGrossIncome
    FROM walmartsales
    GROUP BY
        Branch,
        `Product line`
),
```

OUTPUT:

Branch	Product_line	Profit
A	Home and lifestyle	20282.21
A	Sports and travel	17527.63
A	Electronic accessories	16572.61
A	Food and beverages	15528.47
A	Fashion accessories	14777
A	Health and beauty	11397.93
B	Sports and travel	18084.5
B	Health and beauty	18077.7
B	Home and lifestyle	15877.77
B	Electronic accessories	15427.42
B	Fashion accessories	14850.08
B	Food and beverages	13765.81

Step 1st: Calculate Total income per category



Step 2 – Rank product lines within each branch by total gross income

INPUT :

```sql
RankedProfit AS (
    -- Rank product lines by their total gross income within each branch
    SELECT
        Branch,
        ProductLine,
        TotalGrossIncome,
        ROW_NUMBER() OVER (PARTITION BY Branch ORDER BY TotalGrossIncome DESC) AS RankNum
    FROM ProductLineProfit
)
```

OUTPUT:

Branch	ProductLine	TotalGrossIncome
A	Home and lifestyle	1067.49
B	Sports and travel	951.82
C	Food and beverages	1131.75

Step 3 – Select the top product line for each branch

INPUT :

```sql
SELECT
    Branch,
    ProductLine,
    TotalGrossIncome
FROM RankedProfit
WHERE RankNum = 1
ORDER BY
    Branch;
```

OUTPUT:

Branch	ProductLine	TotalGrossIncome
A	Home and lifestyle	1067.49
B	Sports and travel	951.82
C	Food and beverages	1131.75

INSIGHTS :

Product lines with high profits might mean Walmart sells them at higher prices, spends less to make them, or they're in high demand. Walmart can use this to sell more of these items and manage stock better.

Approach: Find profit by subtracting the cost from the money earned, then rank product categories for each store.

3

Task 3rd : Analyzing Customer Segmentation Based on Spending.

Walmart wants to segment customers based on their average spending behavior. Classify customers into three tiers: High, Medium, and Low spenders based on their total purchase amounts.

Step 1 – Calculate total spending per customer

INTPUT :

```sql
WITH CustomerSpending AS (
    -- Calculate the total spending for each customer, rounded to 2 decimal places
    SELECT
        `Customer ID`,
        ROUND(SUM(Total), 2) AS TotalSpent
    FROM walmartsales
    GROUP BY
        `Customer ID`
),
```

OUTPUT:

Customer_Id	Total_Spending
8	26634.38
3	23402.25
2	23392.31
15	22674.51
1	22634.61
12	21720.7
11	21398.88
13	21063.69
14	21049.42
10	20723.99
6	20694.03
7	20628.13
9	19661.65
5	19632.12

Step 2 – Divide customers into 3 spending tiers

INTPUT :

```sql
CustomerTiers AS (
    -- Assign each customer to a spending tier using NTILE(3)
    SELECT
        `Customer ID`,
        TotalSpent,
        NTILE(3) OVER (ORDER BY TotalSpent) AS SpendingTierRank
    FROM CustomerSpending
)
```

Step 3 – Label each tier as Low, Medium, or High spender

INTPUT :

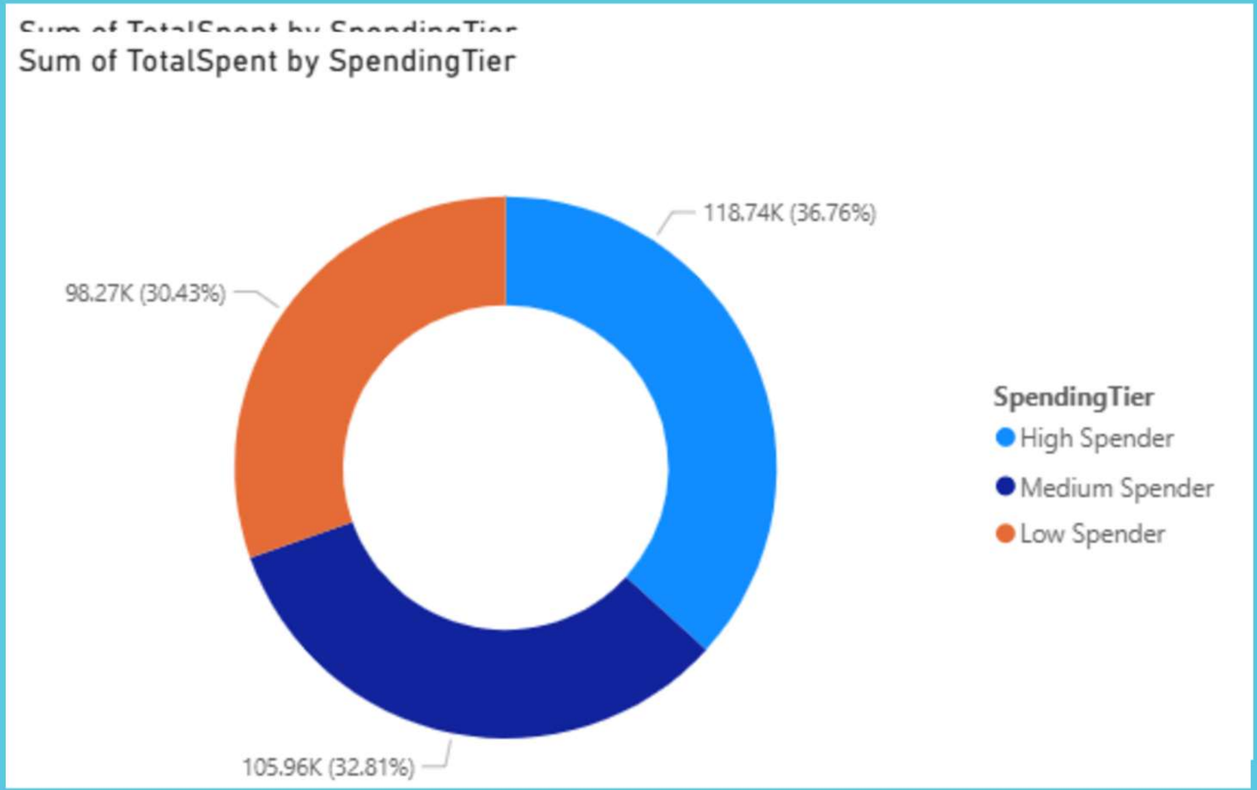
```sql
SELECT
    `Customer ID`,
    TotalSpent,
    CASE
        WHEN SpendingTierRank = 1 THEN 'Low Spender'
        WHEN SpendingTierRank = 2 THEN 'Medium Spender'
        WHEN SpendingTierRank = 3 THEN 'High Spender'
    END AS SpendingTier
FROM CustomerTiers
ORDER BY
    TotalSpent DESC;
```

OUTPUT :

Customer ID	TotalSpent	SpendingTier
8	26634.34	High Spender
3	23402.26	High Spender
2	23392.28	High Spender
15	22674.46	High Spender
1	22634.55	High Spender
12	21720.65	Medium Spender
11	21398.82	Medium Spender
13	21063.66	Medium Spender
14	21049.4	Medium Spender
10	20723.93	Medium Spender
6	20693.96	Low Spender
7	20628.09	Low Spender
9	19661.6	Low Spender
5	19632.04	Low Spender
4	17656.72	Low Spender

Step 3 – Label each tier as Low, Medium, or High spender

Graph :



INSIGHTS :

Sorting customers into High, Medium, and Low spenders helps Walmart send the right offers to the right people. High spenders are likely regular customers, while low spenders could be encouraged to buy more with discounts.

Approach: Add up each customer’s total spending, split them into three groups using NTILE(3), and label them as High, Medium, or Low spenders for targeted promotions.

4

Task 4th : Detecting Anomalies in Sales Transactions.

Walmart suspects that some transactions have unusually high or low sales compared to the average for the product line. Identify these anomalies.

Step 1 – Calculate average and standard deviation for each product line

INPUT:

```sql
WITH ProductLineStats AS (
    -- Calculate the average and standard deviation of sales for each product line
    SELECT
        `Product line` AS ProductLine,
        AVG(Total) AS AverageSales,
        STDDEV(Total) AS StdDevSales
    FROM walmartsales
    GROUP BY
        `Product line`
)
```

OUTPUT:

Product_line	Avg_sales
Health and beauty	323.644
Electronic accessories	319.633
Home and lifestyle	336.637
Sports and travel	332.066
Food and beverages	322.672
Fashion accessories	305.09

Step 2 – Identify transactions far from the average (outliers)

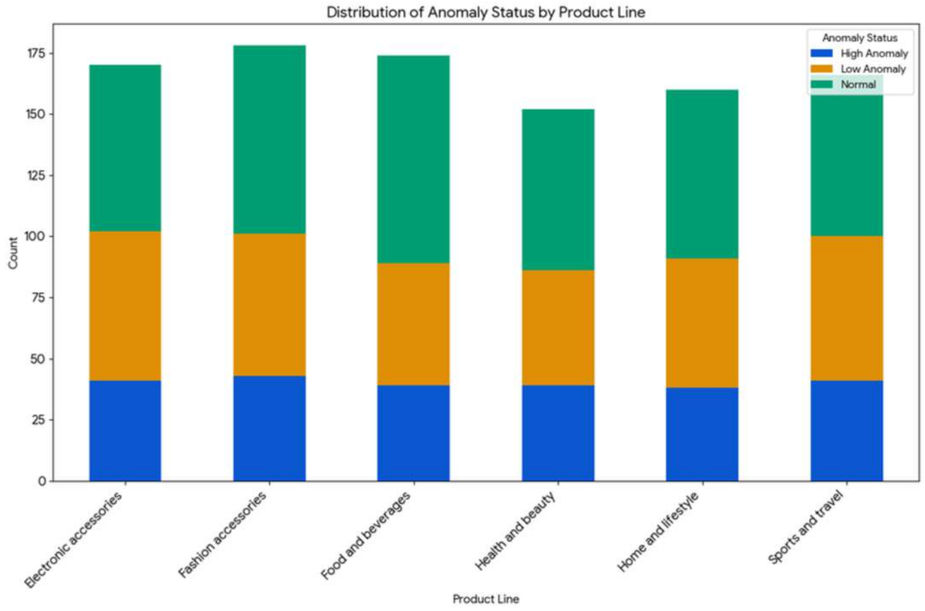
INPUT:

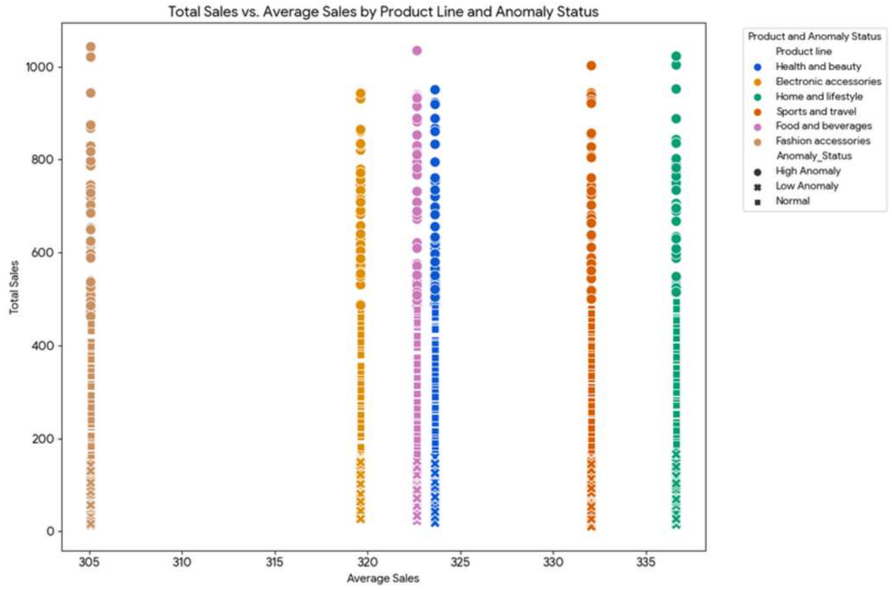
```sql
SELECT Invoice_ID, Product_line, Total,
(SELECT AVG(Total)
FROM walmartsales
AS sub Where Sub.Product_line = main.Product_line) AS Avg_sale,
CASE
    When Total > 1.5 * (Select AVG(Total)
    From walmartsales AS sub
    Where Sub.Product_line = main.Product_line)
THEN 'High Anomaly'
When Total <0.5 * (SELECT avg(Total) From walmartsales AS sub
Where sub.Product_line = main.Product_line)
THEN 'Low Anomaly' ELSE 'Normal'
END
AS Anomaly_Status From walmartsales AS main;
```

OUTPUT:

Invoice_ID	Product_line	Total	Avg_sale	Anomaly_Status
750-67-8 750-67-8428	nd beauty	548.97	323.64368421052643	High Anomaly
226-31-3081	Electronic accessories	80.22	319.6331764705883	Low Anomaly
631-41-3108	Home and lifestyle	340.53	336.63725000000017	Normal
123-19-1176	Health and beauty	489.05	323.64368421052643	High Anomaly
373-73-7910	Sports and travel	634.38	332.066265060241	High Anomaly
699-14-3026	Electronic accessories	627.62	319.6331764705883	High Anomaly
355-53-5943	Electronic accessories	433.69	319.6331764705883	Normal
315-22-5665	Home and lifestyle	772.38	336.63725000000017	High Anomaly
665-32-9167	Health and beauty	76.15	323.64368421052643	Low Anomaly
692-92-5582	Food and beverages	172.75	322.67218390804607	Normal

GRAPHS :


Distribution of Anomaly Status by Product Line
Anomaly Status
High Anomaly
Low Anomaly
Normal
Count
0
25
50
75
100
125
150
175
Electronic accessories
Fashion accessories
Food and beverages
Health and beauty
Home and lifestyle
Sports and travel
Product Line


Total Sales vs. Average Sales by Product Line and Anomaly Status
Total Sales
0
200
400
600
800
1000
305
310
315
320
325
330
335
Average Sales
Product and Anomaly Status
Product line
Health and beauty
Electronic accessories
Home and lifestyle
Sports and travel
Food and beverages
Fashion accessories
Anomaly_Status
High Anomaly
Low Anomaly
Normal

INSIGHTS :

Finding sales that are much bigger or smaller than usual for a product line helps Walmart spot unusual patterns. These could be big orders, big discounts, mistakes, or fraud, and catching them early helps keep sales accurate.

APPROACH :

- Determine the average sales amount for each product line.
- Evaluate each transaction's total against its respective product line's average.
- Classify it as a **High Anomaly** if it exceeds 1.5 times the average, or a **Low Anomaly** if it falls below 0.5 times the average. Categorize all remaining transactions as **Normal.**

5

Task 5th : Most Popular Payment Method by City

Walmart needs to determine the most popular payment method in each city to tailor marketing strategies.

Step 1 – Count transactions per payment method in each city

INPUT :

```sql
WITH PaymentMethodCounts AS (
    -- Count the number of transactions for each payment method in each city
    SELECT
        City,
        Payment,
        COUNT(`Invoice ID`) AS TransactionCount
    FROM walmartsales
    GROUP BY
        City,
        Payment
),
```

Step 2 – Rank payment methods by transaction count in each city

INPUT :

```sql
RankedPayments AS (
    -- Rank payment methods by transaction count within each city
    SELECT
        City,
        Payment,
        TransactionCount,
        ROW_NUMBER() OVER (PARTITION BY City ORDER BY TransactionCount DESC) AS RankNum
    FROM PaymentMethodCounts
)
```

Step 3 – Select the most popular payment method per city

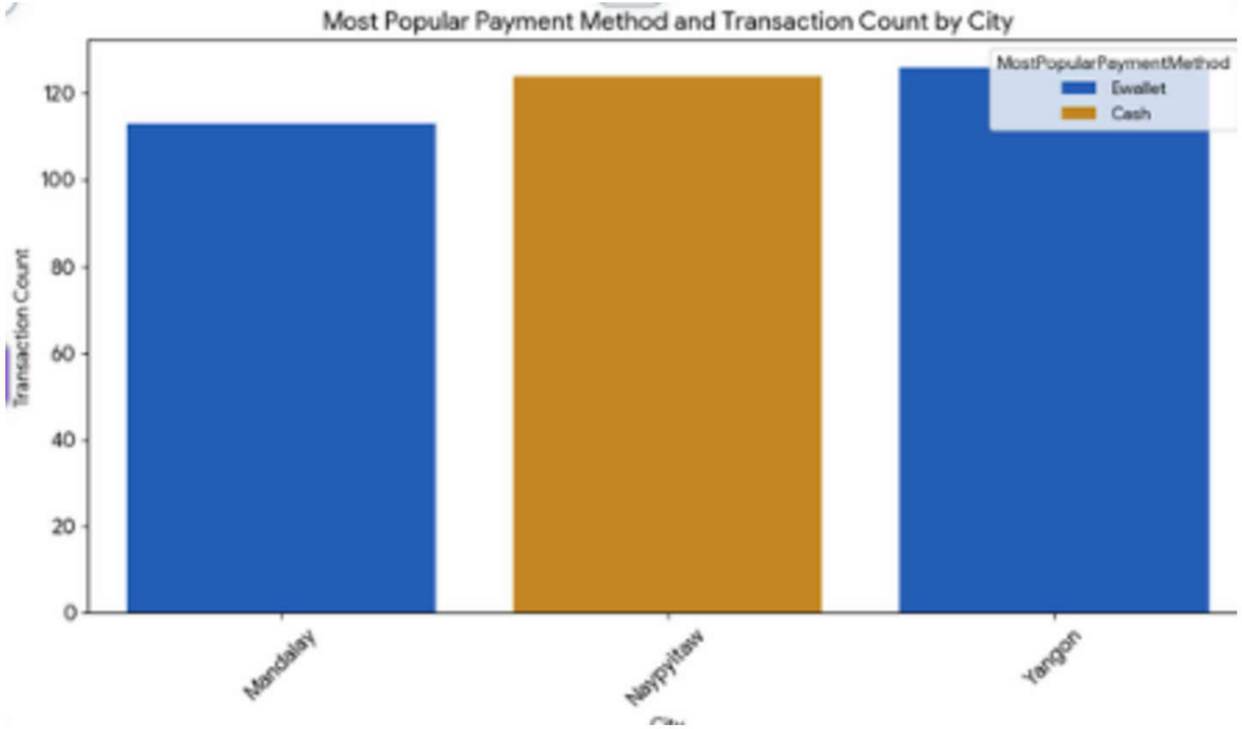
INPUT :

```sql
-- Select the top ranked payment method for
SELECT
    City,
    Payment AS MostPopularPaymentMethod,
    TransactionCount
FROM RankedPayments
WHERE
    RankNum = 1
ORDER BY
    City;
```

OUTPUT :

City	MostPopularPaymentMethod	TransactionCount
Mandalay	Ewallet	113
Naypyitaw	Cash	124
Yangon	Ewallet	126

GRAPHS


Most Popular Payment Method and Transaction Count by City
MostPopularPaymentMethod
Ewallet
Cash
Transaction Count
0
20
40
60
80
100
120
Mandalay
Naypyitaw
Yangon

INSIGHTS :

Knowing which payment method people use most in each city helps Walmart make checkout faster and easier. If a city mostly uses digital payments, Walmart can improve its online or card payment systems there.

Approach : Group the data by city, count how many times each payment method is used, and find the most common one.

6 TASK 6TH :

Monthly Sales Distribution by Gender

Walmart wants to understand the sales distribution between male and female customers on a monthly basis.

Step 1 – Prepare monthly sales data by gender

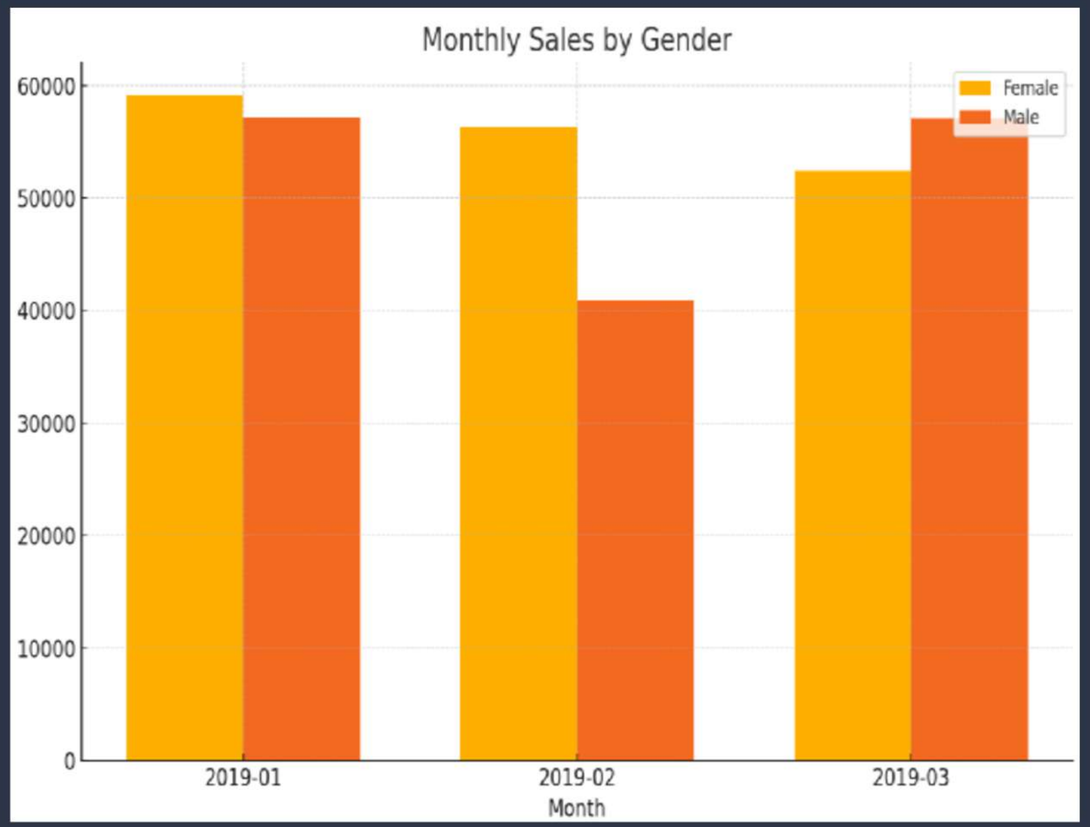
INPUT :

```sql
WITH MonthlySales AS (
    -- Convert the 'Date' string to a DATE type and then format it to get the year and month
    SELECT
        DATE_FORMAT(STR_TO_DATE(Date, '%d-%m-%Y'), '%Y-%m') AS SalesMonth,
        Gender,
        -- Calculate the total sales for each gender in each month, rounded to 2 decimal places
        ROUND(SUM(Total), 2) AS TotalSales
    FROM
        walmartsales
    GROUP BY
        SalesMonth,
        Gender
)
```

OUTPUT :

Sales_Month	Gender	Total_sales
2019-01	Female	59139.15
2019-01	Male	57152.96
2019-02	Female	56335.65
2019-02	Male	40883.93
2019-03	Female	52408.46
2019-03	Male	57047.28

GRAPH


Monthly Sales by Gender
Female
Male
60000
50000
40000
30000
20000
10000
0
2019-01
2019-02
2019-03
Month

Step 2 – Calculate sales percentage for each gender per month

INPUT :

```sql
SELECT
    SalesMonth,
    Gender,
    TotalSales,
    -- Calculate the percentage of sales for each gender within the month
    ROUND(TotalSales * 100.0 / SUM(TotalSales) OVER (PARTITION BY SalesMonth), 2) AS SalesPercentage
FROM
    MonthlySales
ORDER BY
    SalesMonth,
    Gender;
```

OUTPUT :

SalesMonth	Gender	TotalSales	SalesPercentage
2019-01	Female	59138.98	50.85
2019-01	Male	57152.89	49.15
2019-02	Female	56335.56	57.95
2019-02	Male	40883.82	42 57.95
2019-03	Female	52408.39	47.88
2019-03	Male	57047.12	52.12

INSIGHTS :

Knowing sales trends for men and women helps Walmart create promotions for each group and see which products are more popular with them.

Approach : Compare total sales for male and female customers each month.

7

Task 7TH : Best Product line by Customer Type

Walmart wants to know which product lines are preferred by different customer types(Member vs. Normal).

Step 1 – Calculate total sales per product line for each customer type

INPUT :

```sql
WITH ProductLineSales AS (
    -- Calculate the total sales for each product line per customer type
    SELECT
        `Customer type` AS CustomerType,
        `Product line` AS ProductLine,
        ROUND(SUM(Total), 2) AS TotalSales
    FROM walmartsales
    GROUP BY
        `Customer type`,
        `Product line`
),
```

OUTPUT :

Customer_Type	Product_line	Total_Sales
Member	Food and beverages	31357.71
Member	Sports and travel	28234.39
Member	Home and lifestyle	27978.06
Member	Fashion accessories	26324.03
Member	Health and beauty	25831.08
Member	Electronic accessories	24498.54
Normal	Electronic accessories	29839.1
Normal	Fashion accessories	27982
Normal	Sports and travel	26888.61
Normal	Home and lifestyle	25883.9
Normal	Food and beverages	24787.25

Step 2 – Rank product lines within each customer type by total sales

INPUT :

```sql
RankedSales AS (
    -- Rank product lines by their total sales within each customer type
    SELECT
        CustomerType,
        ProductLine,
        TotalSales,
        ROW_NUMBER() OVER (PARTITION BY CustomerType ORDER BY TotalSales DESC) AS RankNum
    FROM ProductLineSales
)
```

Step 3 – Select the top product line for each customer type

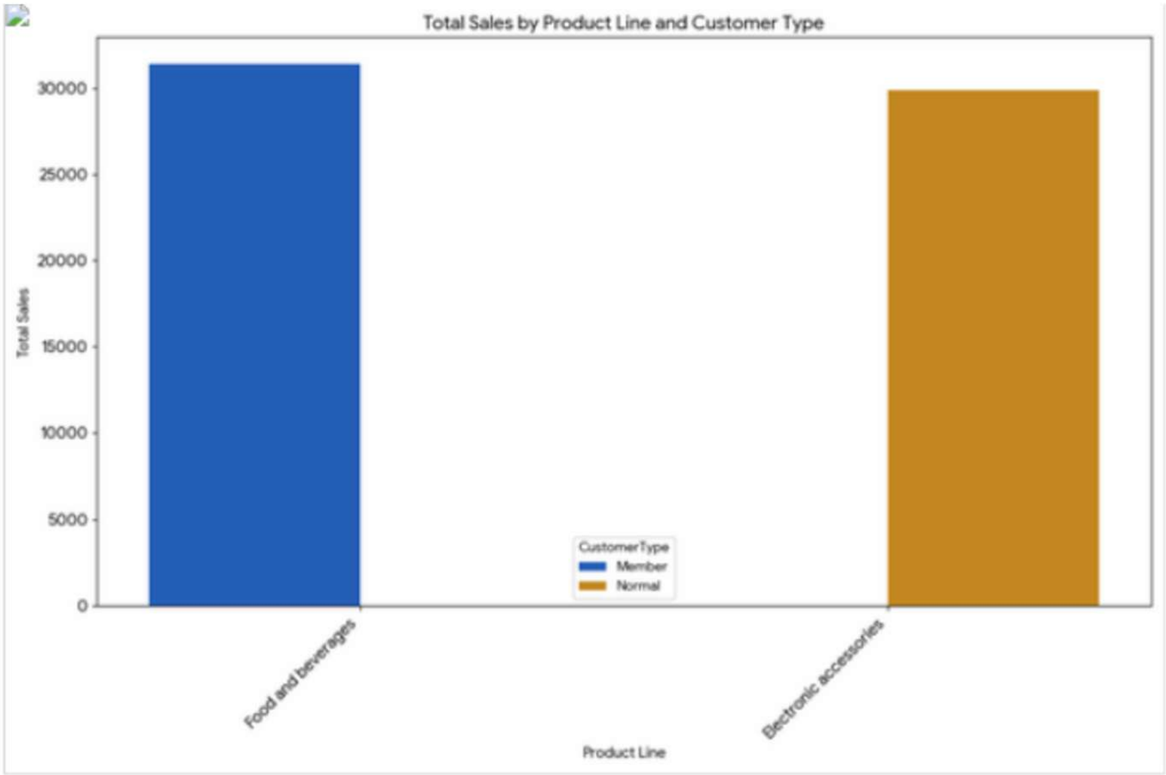
INPUT :

```sql
SELECT
    CustomerType,
    ProductLine,
    TotalSales
FROM RankedSales
WHERE
    RankNum = 1
ORDER BY
    CustomerType;
```

OUTPUT:

	CustomerType	ProductLine	TotalSales
▶	Member	Food and beverages	31357.62
	Normal	Electronic accessories	29839.04

GRAPHS :


Total Sales by Product Line and Customer Type
Total Sales
30000
25000
20000
15000
10000
5000
0
CustomerType
Member
Normal
Food and beverages
Electronic accessories
Product Line

INSIGHTS :

Members and regular customers may like different products. Walmart can use this information to improve loyalty programs and offer products each group prefers.

Approach : List products for each customer type and rank them by total sales to see which ones are most popular.

8

Task 8TH : Identifying Repeat Customers

Walmart needs to identify customers who made repeat purchases within a specific time frame (e.g., within 30 days).

Step 1 – Prepare purchase dates for each customer

INPUT :

```sql
WITH CustomerPurchaseDates AS (
    -- Convert the 'Date' string to a DATE type and order by date for each customer
    SELECT
        `Customer ID`,
        STR_TO_DATE(Date, '%d-%m-%Y') AS PurchaseDate
    FROM walmartsales
    ORDER BY
        `Customer ID`,
        PurchaseDate
),
```

Step 2 – Find the next purchase date for each customer

INPUT :

```sql
NextPurchaseDates AS (
    -- Find the date of the next purchase for each customer
    SELECT
        `Customer ID`,
        PurchaseDate,
        LEAD(PurchaseDate, 1) OVER (PARTITION BY `Customer ID` ORDER BY PurchaseDate) AS NextPurchaseDate
    FROM CustomerPurchaseDates
)
```

Step 3 – Select purchases made within 30 days

INPUT :

```sql
SELECT
    `Customer ID`,
    PurchaseDate,
    NextPurchaseDate,
    DATEDIFF(NextPurchaseDate, PurchaseDate) AS DaysBetweenPurchases
FROM NextPurchaseDates
WHERE
    DATEDIFF(NextPurchaseDate, PurchaseDate) IS NOT NULL
    AND DATEDIFF(NextPurchaseDate, PurchaseDate) <= 30
ORDER BY
    `Customer ID`,
    PurchaseDate;
```

OUTPUT :

Customer_id	Purchase_date	Days_Between
1	2019-01-06	1
1	2019-01-10	4
1	2019-01-10	0
1	2019-01-12	2
1	2019-01-12	0
1	2019-01-13	1
1	2019-01-13	0
1	2019-01-14	1
1	2019-01-14	0
1	2019-01-14	0
1	2019-01-14	0
1	2019-01-15	1
1	2019-01-15	0
1	2019-01-17	2
1	2019-01-17	0
1	2019-01-17	0

STEP 4: FOR UNIQUE IDENTIFICATION

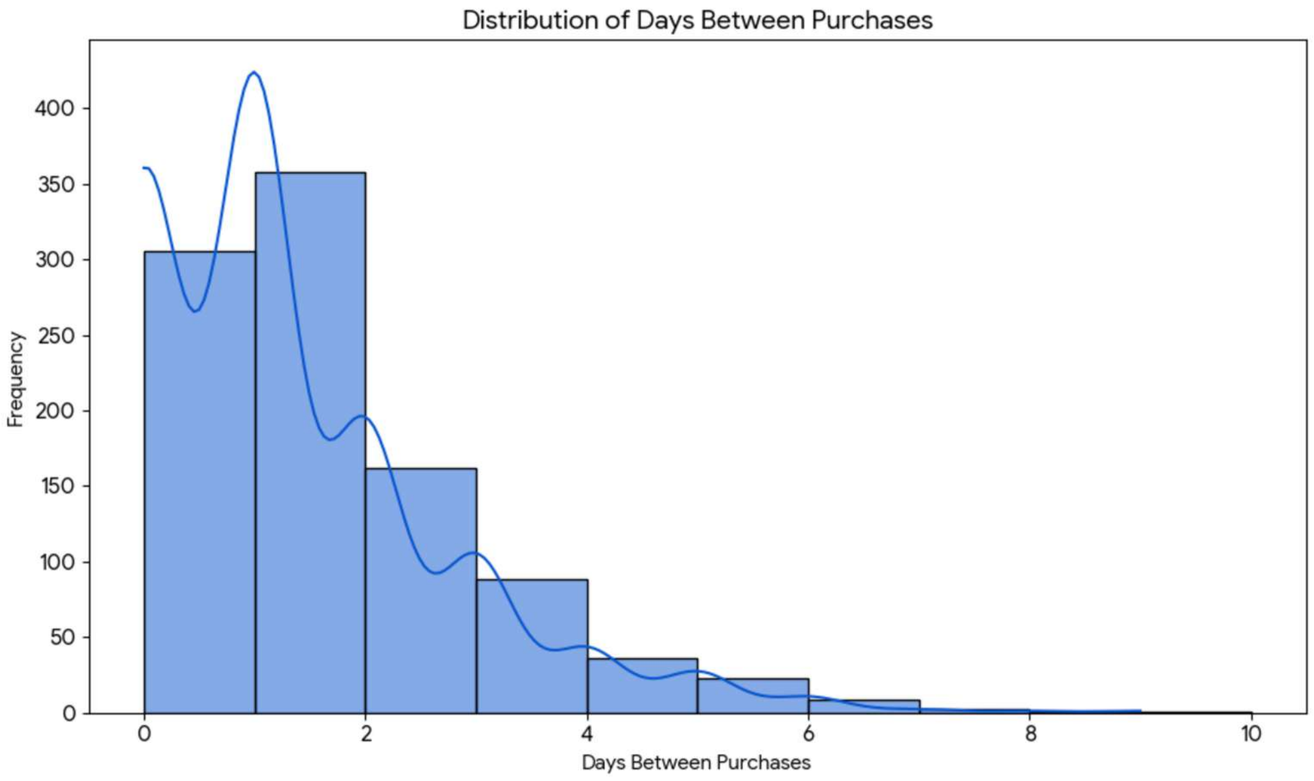
INPUT :

```sql
SELECT DISTINCT
    `Customer ID`
FROM NextPurchaseDates
WHERE
    DATEDIFF(NextPurchaseDate, PurchaseDate) IS NOT NULL
    AND DATEDIFF(NextPurchaseDate, PurchaseDate) <= 30
ORDER BY
    `Customer ID`;
```

OUTPUT :

Customer ID
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

GRAPHS:


Distribution of Days Between Purchases
Frequency
400
350
300
250
200
150
100
50
0
0
2
4
6
8
10
Days Between Purchases

INSIGHTS :

This query helps Walmart find customers who return to shop again within 30 days. These quick repeat purchases may mean strong customer loyalty, successful promotions, or high demand for certain products. Tracking this can help Walmart plan retention offers and reward frequent buyers.

APPROACHES :

- Convert purchase dates into a proper date format for each customer.
- Use LEAD() to find the date of their next purchase.
- Calculate the number of days between purchases.
- Keep only customers whose next purchase was within 30 days.

9

Task 9th :

Findings Top 5 Customers by Sales Volume

Walmart wants to reward its top 5 customers who have generated the most sales Revenue.

Finding Top 5 Customers by Sales Volume

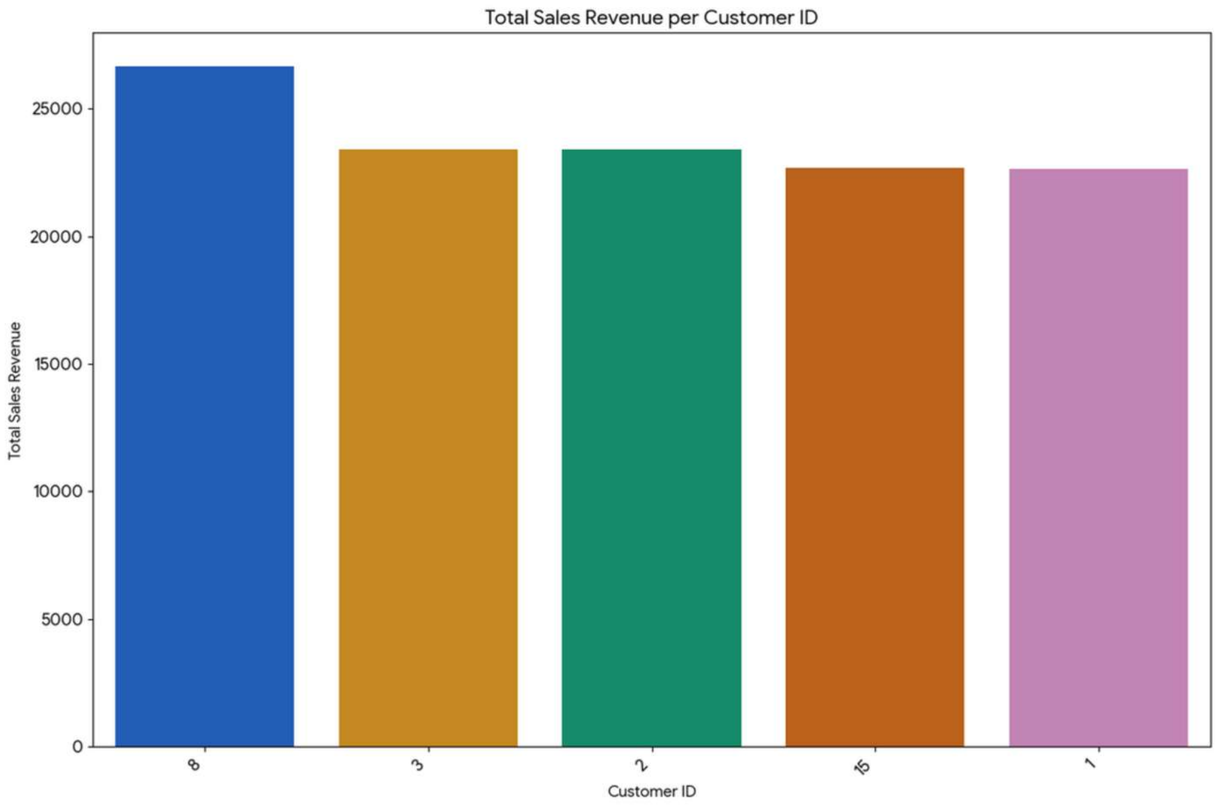
INPUT :

```sql
SELECT
    `Customer ID`,
    -- Calculate the sum of all purchases for each customer, rounded to 2 decimal places
    ROUND(SUM(Total), 2) AS TotalSalesRevenue
FROM
    walmartsales
GROUP BY
    `Customer ID`
ORDER BY
    TotalSalesRevenue DESC
LIMIT 5;
```

OUTPUT :

Customer ID	TotalSalesRevenue
8	26634.34
3	23402.26
2	23392.28
15	22674.46
1	22634.55

GRAPHS :


Total Sales Revenue per Customer ID
Total Sales Revenue
25000
20000
15000
10000
5000
0
8
3
2
15
1
Customer ID

INSIGHTS :

Rewarding top customers with loyalty benefits increases engagement. This insight helps Walmart strengthen customer relationships.

10

Task 10TH : Analyzing Sales Trends by Day of the Week

Walmart wants to analyze the sales patterns to determine which day of the week brings the highest sales.

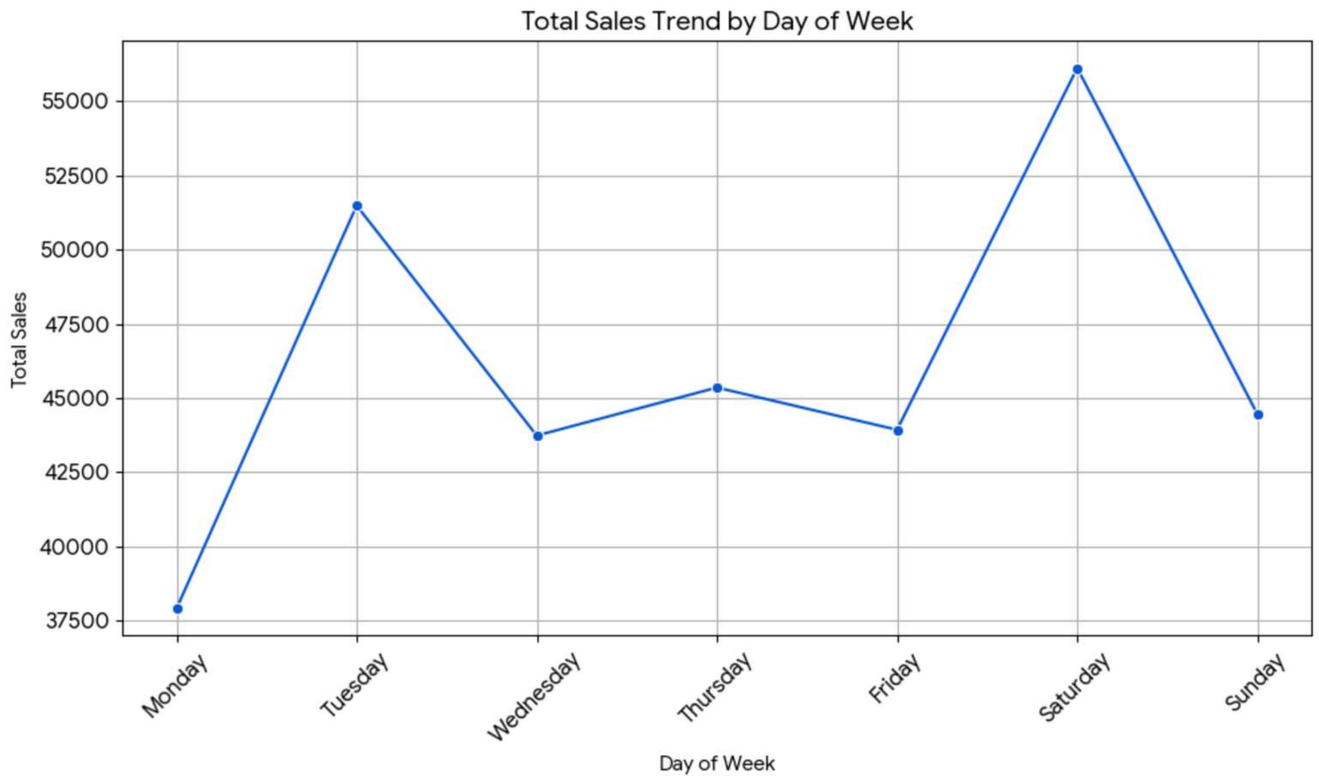
Step 1st : Extract the day of the week

INPUT :

```
Select DAYNAME(STR_To_Date(Date, '%d-%m-%Y')) AS Day_Of_Week,
Round(SUM(Total),2) AS Total_sales
From Walmartsales
Group By Day_Of_Week
Order By Total_sales DESC;
```

OUTPUT :

	DayOfWeek	TotalSales
▶	Saturday	56120.81
	Tuesday	51482.25
	Thursday	45349.25
	Sunday	44457.89
	Friday	43926.34
	Wednesday	43731.14
	Monday	37899.08


Total Sales Trend by Day of Week
Total Sales
55000
52500
50000
47500
45000
42500
40000
37500
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Day of Week

GRAPH

Step 2nd Identify the day with the Highest Sales

INPUT :

```sql
SELECT
    DAYNAME(STR_TO_DATE(Date, '%d-%m-%Y')) AS Day_Of_Week,
    ROUND(SUM(Total), 2) AS Total_Sales
FROM
    walmartsales
GROUP BY
    Day_Of_Week
ORDER BY
    Total_Sales DESC
LIMIT 1;
```

OUTPUT :

	Day_Of_Week	Total_Sales
▶	Saturday	56120.81

Step 3rd Additional Insights

INPUT :

```sql
With Weekly_Sales AS (
 Select Round(SUM(Total),3) AS Total_Weekly_sales
 From walmartsales)
 SELECT DAYNAME(STR_To_Date(Date, '%d-%m-%Y')) AS Day_Of_week,
 round(SUM(Total),3) AS Total_sales,
round((Sum(Total) / (Select total_Weekly_Sales From Weekly_sales)) * 100, 2) AS
Percentage_weekly_Sales
From Walmartsales
Group by Day_Of_Week
Order By total_Sales DESC;
```

OUTPUT :

Day_Of_Week	Total_Sales	Percentage_Weekly_Sales
Saturday	56120.809	17.38
Tuesday	51482.246	15.94
Thursday	45349.248	14.04
Sunday	44457.892	13.77
Friday	43926.341	13.6
Wednesday	43731.135	13.54
Monday	37899.078	11.73

INSIGHTS :

Finding which day has the highest sales helps decide when to keep more staff, run offers, and stock more products.

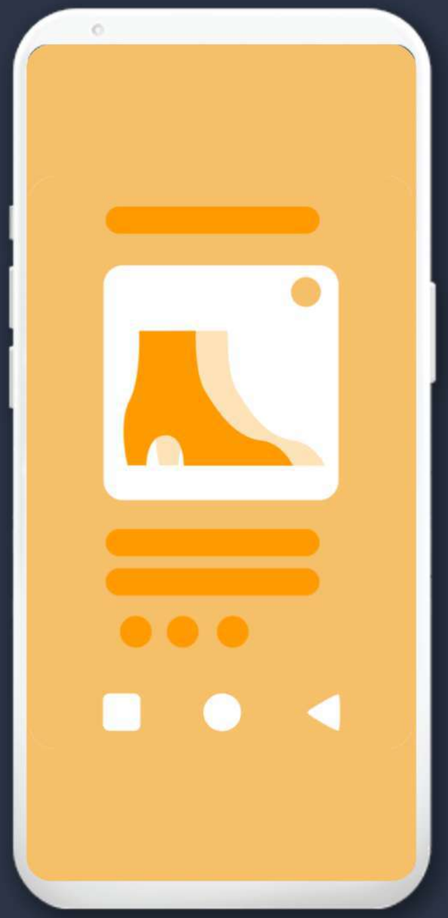
Approach: Group sales by day of the week and check how much each day earns.

KEY INSIGHTS

- Branch Performance: Branch A's sales are growing fast, and Branch C makes the most profit.
- Top Products: Each branch's best-selling product is different, so promotions should match local demand.
- Customer Types: Big spenders and repeat buyers bring in most sales great for loyalty programs.
- Payments: People in Yangon and Mandalay mostly use e-wallets, but in Naypyitaw they prefer cash.
- Gender & Time Trends: Women spent more in the first few months; weekends, especially Saturdays, see the most sales.
- Sales Issues: Some products show unusual sales patterns checking regularly can stop losses.
- Days: Saturday sells the most, then Tuesday good to plan staff and marketing.

CONCLUSION

Walmart's sales data shows that Branch C earns the most profit, while Branch A's sales are growing fast. Different locations have different best-selling products, so each branch needs its own stock and marketing plan. Loyal and high-spending customers are key for growth, and regional differences in payment methods and shopping trends mean strategies should be customized. Fixing unusual sales issues, managing inventory better, and using these insights can boost both profit and customer satisfaction.



VIDEO EXPLAINATION

task 1

CONTINUVATION IS THERE IN THE FOLDER

https://drive.google.com/drive/folders/1iY50vM2YOCbiHcVc05giTSspvU8MfP
usp=drive_link

Thank you!