

Sales Performance Analysis of Walmart Stores Using Advanced MySQL Techniques

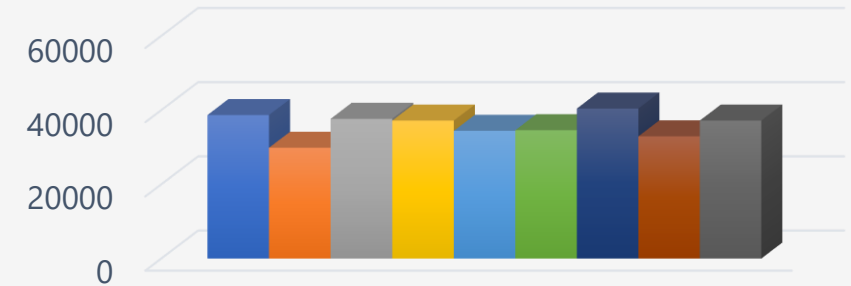


Task 1: Identifying the Top Branch by Sales Growth Rate

Walmart wants to identify which branch has exhibited the highest sales growth over time. Analyze the total sales for each branch and compare the growth rate across months to find the top performer.

Step 1: Aggregate Monthly Sales by Branch

```
1 SELECT
2     Branch,
3     DATE_FORMAT(Date, '%Y-%m') AS Month_Year,
4     SUM(Total) AS Monthly_Sales
5 FROM
6     walmartsales
7 GROUP BY
8     Branch, Month_Year
9 ORDER BY
10    Branch, Month_Year;
```



Monthly_Sales

■ A 2019-01 ■ A 2019-02 ■ A 2019-03
■ B 2019-01 ■ B 2019-02 ■ B 2019-03
■ C 2019-01 ■ C 2019-02 ■ C 2019-03

Result Grid			
Filter Rows:			
	Branch	Month_Year	Monthly_Sales
▶	A	2019-01	38681.1285
	A	2019-02	29860.120500000005
	A	2019-03	37659.121500000001
	B	2019-01	37176.058500000014
	B	2019-02	34424.270999999999
	B	2019-03	34597.3425
	C	2019-01	40434.680999999999
	C	2019-02	32934.982500000006
	C	2019-03	37199.043

Task 1: Identifying the Top Branch by Sales Growth Rate

Walmart wants to identify which branch has exhibited the highest sales growth over time. Analyze the total sales for each branch and compare the growth rate across months to find the top performer.

Step 2: Calculate Monthly Sales Growth Rate.

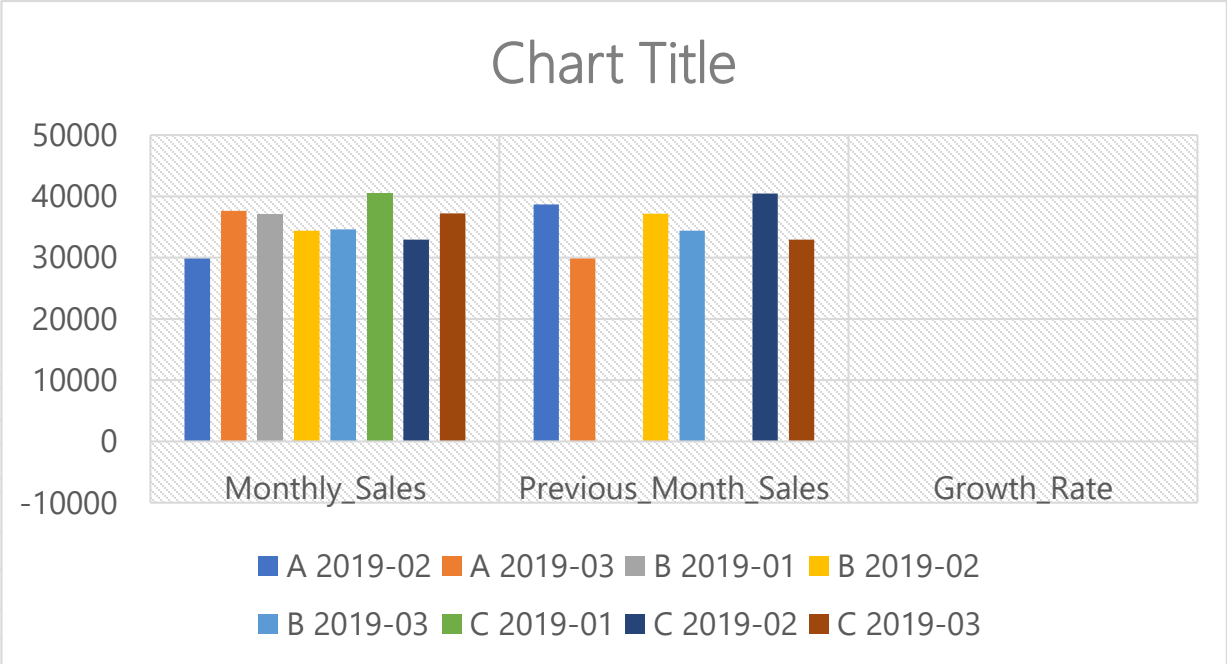
```
1 WITH Monthly_Sales AS (  
2     SELECT  
3         Branch,  
4         DATE_FORMAT(Date, '%Y-%m') AS Month_Year,  
5         SUM(Total) AS Monthly_Sales  
6     FROM  
7         walmartsales  
8     GROUP BY  
9         Branch, Month_Year  
10 )  
11 SELECT  
12     Branch,  
13     Month_Year,  
14     Monthly_Sales,  
15     LAG(Monthly_Sales) OVER (PARTITION BY Branch ORDER BY Month_Year) AS Previous_Month_Sales,  
16     (Monthly_Sales - LAG(Monthly_Sales) OVER (PARTITION BY Branch ORDER BY Month_Year)) /  
17     LAG(Monthly_Sales) OVER (PARTITION BY Branch ORDER BY Month_Year) * 100 AS Growth_Rate  
18 FROM  
19     Monthly_Sales;
```

Task 1: Identifying the Top Branch by Sales Growth Rate

Walmart wants to identify which branch has exhibited the highest sales growth over time. Analyze the total sales for each branch and compare the growth rate across months to find the top performer.

Step 2: Calculate Monthly Sales Growth Rate.

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Branch	Month_Year	Monthly_Sales	Previous_Month_Sales	Growth_Rate
A	2019-01	38681.1285	NULL	NULL
A	2019-02	29860.120500000005	38681.1285	-22.804422575210015
A	2019-03	37659.121500000001	29860.120500000005	26.118451196471238
B	2019-01	37176.0585000000014	NULL	NULL
B	2019-02	34424.270999999999	37176.0585000000014	-7.402042096528386
B	2019-03	34597.3425	34424.270999999999	0.50276010202222
C	2019-01	40434.680999999999	NULL	NULL
C	2019-02	32934.9825000000006	40434.680999999999	-18.547688060158027
C	2019-03	37199.043	32934.9825000000006	12.94690379750465



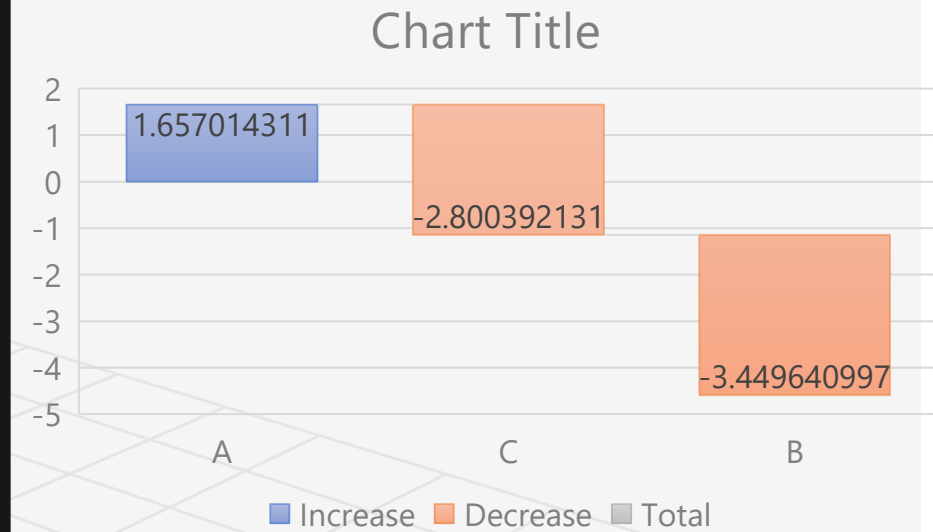
Task 1: Identifying the Top Branch by Sales Growth Rate

Walmart wants to identify which branch has exhibited the highest sales growth over time. Analyze the total sales for each branch and compare the growth rate across months to find the top performer.

Step 3: Identify the Top Branch by Growth Rate.

```
1 WITH Monthly_Sales AS (  
2     SELECT  
3         Branch,  
4         DATE_FORMAT(Date, '%Y-%m') AS Month_Year,  
5         SUM(Total) AS Monthly_Sales  
6     FROM  
7         walmartsales  
8     GROUP BY  
9         Branch, Month_Year  
10 ),  
11 -- SELECT * FROM Monthly_Sales;  
12 Growth_Rates AS (  
13     SELECT  
14         Branch,  
15         Month_Year,  
16         (Monthly_Sales - LAG(Monthly_Sales) OVER (PARTITION BY Branch ORDER BY Month_Year)) /  
17         LAG(Monthly_Sales) OVER (PARTITION BY Branch ORDER BY Month_Year) * 100 AS Growth_Rate  
18     FROM  
19         Monthly_Sales  
20 )  
21 SELECT  
22     Branch,  
23     AVG(Growth_Rate) AS Avg_Growth_Rate  
24 FROM  
25     Growth_Rates  
26 GROUP BY  
27     Branch  
28 ORDER BY  
29     Avg_Growth_Rate DESC;
```

Result Grid		Filter Rows:
	Branch	Avg_Growth_Rate
▶	A	1.6570143106306112
	C	-2.8003921313266886
	B	-3.449640997253083



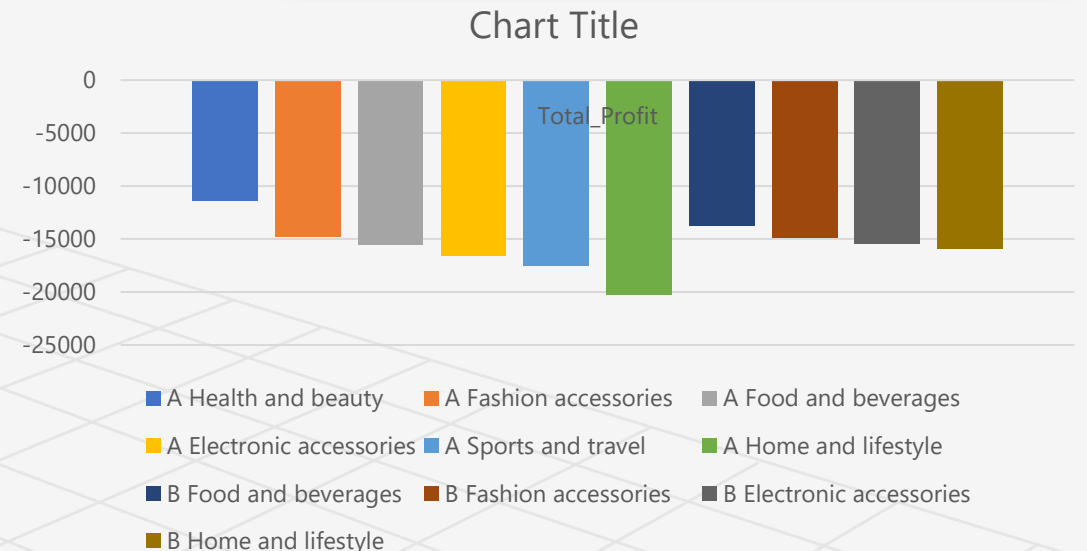
Task 2: Finding the Most Profitable Product Line for Each Branch

Walmart needs to determine which product line contributes the highest profit to each branch. The profit margin should be calculated based on the difference between the gross income and cost of goods sold.

Step 1: Calculate Profit for Each Product Line.

```
1 SELECT
2     Branch,
3     `Product line`,
4     SUM(`gross income` - cogs) AS Total_Profit
5 FROM
6     walmartsales
7 GROUP BY
8     Branch, `Product line`
9 ORDER BY
10    Branch, Total_Profit DESC;
```

	Branch	Product line	Total_Profit
▶	A	Health and beauty	-11397.967000000002
	A	Fashion accessories	-14777.031500000003
	A	Food and beverages	-15528.519500000002
	A	Electronic accessories	-16572.6265
	A	Sports and travel	-17527.680500000006
	A	Home and lifestyle	-20282.2245
	B	Food and beverages	-13765.851499999995
	B	Fashion accessories	-14850.143500000002
	B	Electronic accessories	-15427.4965
	B	Home and lifestyle	-15877.815499999995



Task 2: Finding the Most Profitable Product Line for Each Branch

Walmart needs to determine which product line contributes the highest profit to each branch. The profit margin should be calculated based on the difference between the gross income and cost of goods sold.

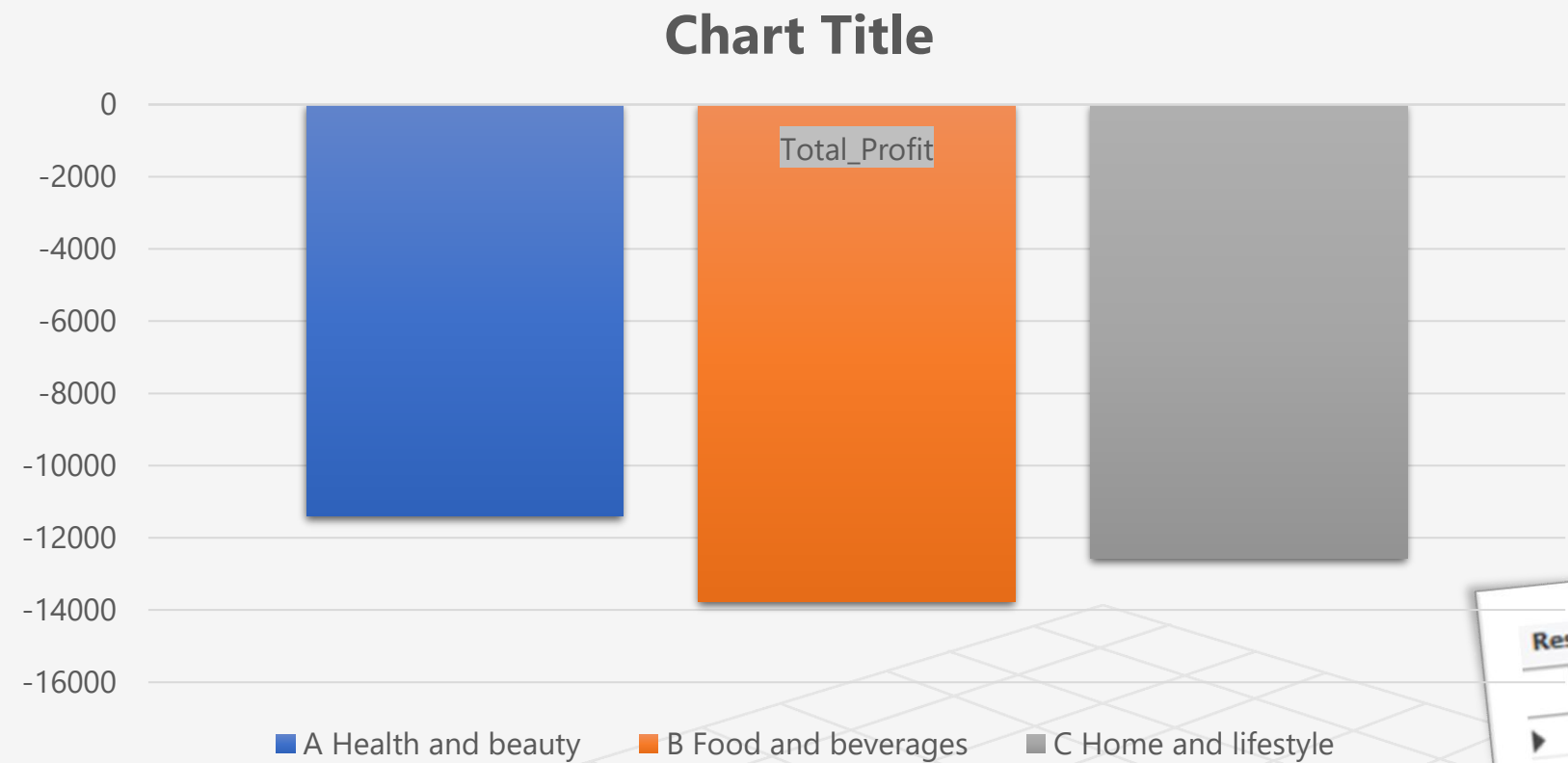
Step 2: Identify the Most Profitable Product Line for Each Branch.

```
1 WITH Profit_Calculation AS (  
2     SELECT  
3         Branch,  
4         `Product line`,  
5         SUM(`gross income` - cogs) AS Total_Profit,  
6         RANK() OVER (PARTITION BY Branch ORDER BY SUM(`gross income` - cogs) DESC) AS Profit_Rank  
7     FROM  
8         walmartsales  
9     GROUP BY  
10        Branch, `Product line`  
11 )  
12 SELECT  
13     Branch,  
14     `Product line`,  
15     Total_Profit  
16 FROM  
17     Profit_Calculation  
18 WHERE  
19     Profit_Rank = 1;
```

Task 2: Finding the Most Profitable Product Line for Each Branch

Walmart needs to determine which product line contributes the highest profit to each branch. The profit margin should be calculated based on the difference between the gross income and cost of goods sold.

Step 2: Identify the Most Profitable Product Line for Each Branch.



Result Grid		Filter Rows:	Export:
Branch	Product line	Total_Profit	
A	Health and beauty	-11397.967000000002	
B	Food and beverages	-13765.851499999995	
C	Home and lifestyle	-12572.167000000001	

Task 3: 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 for Each Customer.

```
1 WITH Customer_Spending AS (  
2     SELECT  
3         `Customer ID`,  
4         SUM(Total) AS Total_Spending  
5     FROM  
6         walmartsales  
7     GROUP BY  
8         `Customer ID`  
9 ),  
10 Spending_Tiers AS (  
11     SELECT  
12         `Customer ID`,  
13         Total_Spending,  
14         NTILE(5) OVER (ORDER BY Total_Spending) AS Tier  
15     FROM  
16         Customer_Spending  
17 )  
18 SELECT  
19     `Customer ID`,  
20     Total_Spending,  
21     CASE  
22         WHEN Tier = 5 THEN 'High Spender'  
23         WHEN Tier IN (3, 4) THEN 'Medium Spender'  
24         WHEN Tier IN (1, 2) THEN 'Low Spender'  
25     END AS Spending_Category  
26 FROM  
27     Spending_Tiers  
28 ORDER BY  
29     Total_Spending DESC;
```

Result Grid				Filter Rows:	Export:
	Customer ID	Total_Spending	Spending_Category		
▶	8	26634.341999999997	High Spender		
	3	23402.263499999997	High Spender		
	2	23392.277999999995	High Spender		
	15	22674.455999999999	Medium Spender		
	1	22634.545499999999	Medium Spender		
	12	21720.646500000003	Medium Spender		
	11	21398.8215	Medium Spender		
	13	21063.6615	Medium Spender		
	14	21049.402500000004	Medium Spender		
	10	20723.934	Low Spender		
	6	20693.9565	Low Spender		
	7	20628.089999999997	Low Spender		
	9	19661.596500000003	Low Spender		
	5	19632.039	Low Spender		
	4	17656.715999999997	Low Spender		

Task 3: 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 for Each Customer.



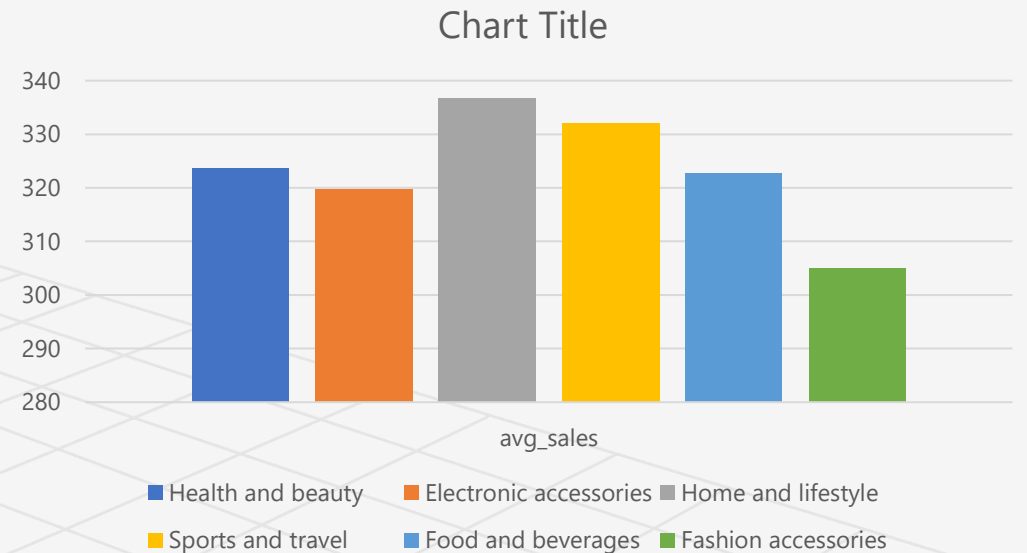
Task 4: Detecting Anomalies in Sales Transactions.

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 Sales per Product Line.

```
1 SELECT
2     `Product line`,
3     AVG(`Total`) AS avg_sales
4 FROM
5     walmartsales
6 GROUP BY
7     `Product line`;
```

Result Grid		Filter Rows:
Product line	avg_sales	
Health and beauty	323.64301973684223	
Electronic accessories	319.63253823529413	
Home and lifestyle	336.63695625	
Sports and travel	332.06521987951805	
Food and beverages	322.6715172413793	
Fashion accessories	305.089297752809	



Task 4: 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 2: Identify Anomalies.

```
1 WITH AvgSales AS (  
2     SELECT  
3         `Product line`,  
4         AVG(`Total`) AS avg_sales  
5     FROM  
6         walmartsales  
7     GROUP BY  
8         `Product line`  
9 )  
10 SELECT  
11     w.*,  
12     a.avg_sales,  
13     CASE  
14         WHEN w.`Total` > a.avg_sales * 1.5 THEN 'High Anomaly'  
15         WHEN w.`Total` < a.avg_sales * 0.5 THEN 'Low Anomaly'  
16         ELSE 'Normal'  
17     END AS anomaly_status  
18 FROM  
19     walmartsales w  
20 JOIN  
21     AvgSales a ON w.`Product line` = a.`Product line`;  
22
```

Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Time	Payment	cogs	gross margin percentage
233-67-5758	C	Naypyitaw	Normal	Male	Health and beauty	40.35	1	2.0175	42.3675	13:46:00	Ewallet	40.35	4.761904762
430-53-4718	B	Mandalay	Member	Male	Health and beauty	75.37	8	30.148	633.108	15:46:00	Credit card	602.96	4.761904762
552-44-5977	B	Mandalay	Member	Male	Health and beauty	62	8	24.8	520.8	19:08:00	Credit card	496	4.761904762
764-44-8999	B	Mandalay	Normal	Female	Health and beauty	14.76	2	1.476	30.996	14:42:00	Ewallet	29.52	4.761904762
148-41-7930	C	Naypyitaw	Normal	Male	Health and beauty	99.96	7	34.986	734.706	10:33:00	Cash	699.72	4.761904762
809-46-1866	A	Yangon	Normal	Female	Health and beauty	58.15	4	11.63	244.23	17:44:00	Cash	232.6	4.761904762
605-03-2706	A	Yangon	Normal	Male	Health and beauty	15.8	3	2.37	49.77	18:02:00	Cash	47.4	4.761904762
503-21-4385	B	Mandalay	Member	Male	Health and beauty	39.91	3	5.9865	125.7165	12:40:00	Ewallet	119.73	4.761904762
333-23-2632	A	Yangon	Member	Male	Health and beauty	10.08	7	3.528	74.088	20:14:00	Cash	70.56	4.761904762
449-16-6770	A	Yangon	Normal	Female	Health and beauty	50.79	5	12.6975	266.6475	14:53:00	Credit card	253.95	4.761904762
131-70-8179	A	Yangon	Member	Male	Health and beauty	92.09	3	13.8135	290.0835	16:27:00	Cash	276.27	4.761904762
433-08-7822	C	Naypyitaw	Normal	Female	Health and beauty	64.08	7	27.6115	579.8415	19:48:00	Ewallet	552.23	4.761904762
458-10-8612	C	Naypyitaw	Normal	Male	Health and beauty	39.62	9	17.829	374.409	12:27:00	Ewallet	448.56	4.761904762
397-25-8725	A	Yangon	Member	Female	Health and beauty	82.88	5	20.72	435.12	14:08:00	Credit card	414.4	4.761904762
851-98-3555	B	Mandalay	Normal	Female	Health and beauty	84.61	10	42.305	888.405	18:58:00	Credit card	846.1	4.761904762
450-42-3339	C	Naypyitaw	Normal	Male	Health and beauty	58.32	2	5.832	122.472	12:42:00	Ewallet	116.64	4.761904762
545-07-8534	C	Naypyitaw	Normal	Female	Health and beauty	27.73	5	6.9325	145.5825	20:21:00	Credit card	138.65	4.761904762
595-94-9924	A	Yangon	Member	Female	Health and beauty	85.6	7	29.96	629.16	13:50:00	Cash	599.2	4.761904762
725-54-0677	C	Naypyitaw	Member	Male	Health and beauty	99.16	8	39.664	832.944	17:47:00	Credit card	793.28	4.761904762
699-88-1972	B	Mandalay	Normal	Male	Health and beauty	83.66	5	20.915	439.215	10:26:00	Cash	418.3	4.761904762
389-70-2397	C	Naypyitaw	Normal	Female	Health and beauty								

Task 4: 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 3: Filter and Display Anomalies.

```

1 WITH AvgSales AS (
2     SELECT
3         `Product line`,
4         AVG(`Total`) AS avg_sales
5     FROM
6         walmartsales
7     GROUP BY
8         `Product line`
9 ),
10 Anomalies AS (
11     SELECT
12         w.*,
13         a.avg_sales,
14         CASE
15             WHEN w.`Total` > a.avg_sales * 1.5 THEN 'High
16 Anomaly'
17             WHEN w.`Total` < a.avg_sales * 0.5 THEN 'Low Anomaly'
18             ELSE 'Normal'
19         END AS anomaly_status
20     FROM
21         walmartsales w
22     JOIN
23         AvgSales a ON w.`Product line` = a.`Product line`
24 )
25 SELECT
26     *
27 FROM
28     Anomalies
29 WHERE
30     anomaly_status IN ('High Anomaly', 'Low Anomaly');

```

assess type	filter notes	exports	wrap cell contents															
Customer type	Gender	Product line	Unit price	Quantity	Tax %	Total	Time	Payment	cogs	gross margin percentage	gross income	Rating	Customer ID	Date	avg_sales		anomaly_status	
pyritaw	Normal	Male	Health and beauty	40.35	1	2.0175	42.3675	13:46:00	Ewallet	40.35	4.761904762	2.0175	6.2	6	2019-01-29	323.64301973684223	Low Anomaly	
ndalay	Member	Male	Health and beauty	75.37	8	30.148	633.108	15:46:00	Credit card	602.96	4.761904762	30.148	8.4	7	2019-01-28	323.64301973684223	High Anomaly	
ndalay	Member	Male	Health and beauty	62	8	24.8	520.8	19:08:00	Credit card	496	4.761904762	24.8	6.2	11	2019-01-03	323.64301973684223	High Anomaly	
ndalay	Normal	Female	Health and beauty	14.76	2	1.476	30.996	14:42:00	Ewallet	29.52	4.761904762	1.476	4.3	2	2019-02-18	323.64301973684223	Low Anomaly	
pyritaw	Normal	Male	Health and beauty	99.96	7	34.986	734.706	10:33:00	Cash	699.72	4.761904762	34.986	6.1	2	2019-01-23	323.64301973684223	High Anomaly	
igon	Normal	Female	Health and beauty	15.8	3	2.37	49.77	18:02:00	Cash	47.4	4.761904762	2.37	9.5	7	2019-03-25	323.64301973684223	Low Anomaly	
ndalay	Member	Male	Health and beauty	39.91	3	5.9865	125.7165	12:40:00	Ewallet	119.73	4.761904762	5.9865	9.3	5	2019-02-21	323.64301973684223	Low Anomaly	
igon	Member	Male	Health and beauty	10.08	7	3.528	74.088	20:14:00	Cash	70.56	4.761904762	3.528	4.2	8	2019-03-28	323.64301973684223	Low Anomaly	
pyritaw	Normal	Female	Health and beauty	78.89	7	27.6115	579.8415	19:48:00	Ewallet	552.23	4.761904762	27.6115	7.5	2	2019-01-05	323.64301973684223	High Anomaly	
pyritaw	Normal	Male	Health and beauty	84.61	10	42.305	888.405	18:58:00	Credit card	846.1	4.761904762	42.305	8.8	12	2019-02-09	323.64301973684223	High Anomaly	
pyritaw	Normal	Female	Health and beauty	58.32	2	5.832	122.472	12:42:00	Ewallet	116.64	4.761904762	5.832	6	10	2019-02-14	323.64301973684223	Low Anomaly	
pyritaw	Normal	Female	Health and beauty	27.73	5	6.9325	145.5825	20:21:00	Credit card	138.65	4.761904762	6.9325	4.2	3	2019-03-26	323.64301973684223	Low Anomaly	
igon	Member	Male	Health and beauty	85.6	7	29.96	629.16	13:50:00	Cash	599.2	4.761904762	29.96	5.3	9	2019-03-02	323.64301973684223	High Anomaly	
pyritaw	Member	Male	Health and beauty	99.16	8	39.664	832.944	17:47:00	Credit card	793.28	4.761904762	39.664	4.2	12	2019-01-28	323.64301973684223	High Anomaly	
ndalay	Normal	Female	Health and beauty	62.82	2	6.282	131.922	12:36:00	Ewallet	125.64	4.761904762	6.282	4.9	10	2019-01-17	323.64301973684223	Low Anomaly	
pyritaw	Member	Male	Health and beauty	33.81	3	5.0715	106.5015	15:11:00	Ewallet	479.75	4.761904762	5.0715	7.3	3	2019-01-26	323.64301973684223	Low Anomaly	
pyritaw	Member	Male	Health and beauty	95.95	5	23.9875	503.7375	14:21:00	Ewallet	479.75	4.761904762	23.9875	8.8	14	2019-01-23	323.64301973684223	High Anomaly	
igon	Member	Female	Health and beauty	91.3	1	4.565	95.865	14:42:00	Ewallet	91.3	4.761904762	4.565	9.2	5	2019-02-14	323.64301973684223	Low Anomaly	
ndalay	Member	Female	Health and beauty	72.11	9	32.4495	681.4395	13:53:00	Credit card	648.99	4.761904762	32.4495	7.7	14	2019-01-28	323.64301973684223	High Anomaly	

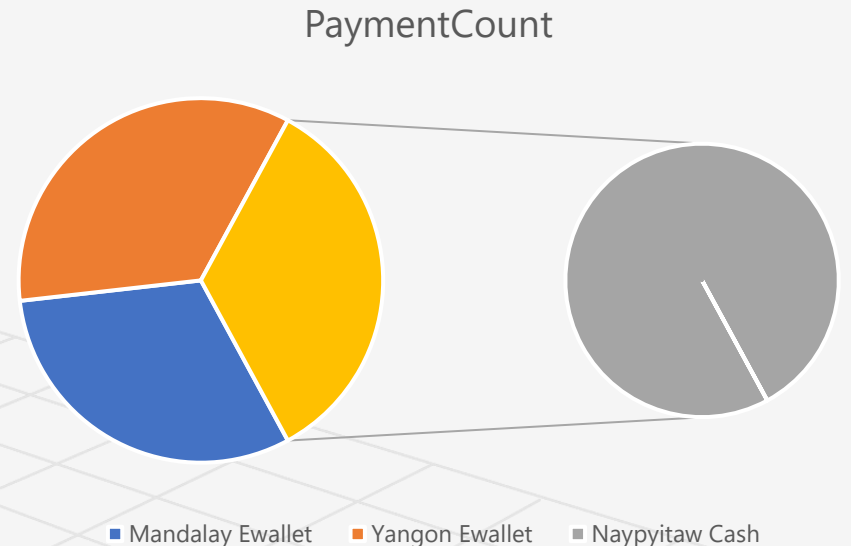
Task 5: Most Popular Payment Method by City.

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

Step 1: Most Popular Payment Method by City.

```
1 WITH PaymentCounts AS (  
2     SELECT  
3         City,  
4         Payment,  
5         COUNT(*) AS PaymentCount,  
6         ROW_NUMBER() OVER (PARTITION BY City ORDER BY COUNT(*) DESC) AS PaymentRank  
7     FROM  
8         walmartsales  
9     GROUP BY  
10        City, Payment  
11 )  
12 SELECT  
13     City,  
14     Payment,  
15     PaymentCount  
16 FROM  
17     PaymentCounts  
18 WHERE  
19     PaymentRank = 1;
```

Result Grid				Filter Rows:
	City	Payment	PaymentCount	
▶	Mandalay	Ewallet	113	
	Naypyitaw	Cash	124	
	Yangon	Ewallet	126	



Task 6: Monthly Sales Distribution by Gender.

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

Step 1: Calculate total sales for each gender on a monthly basis.

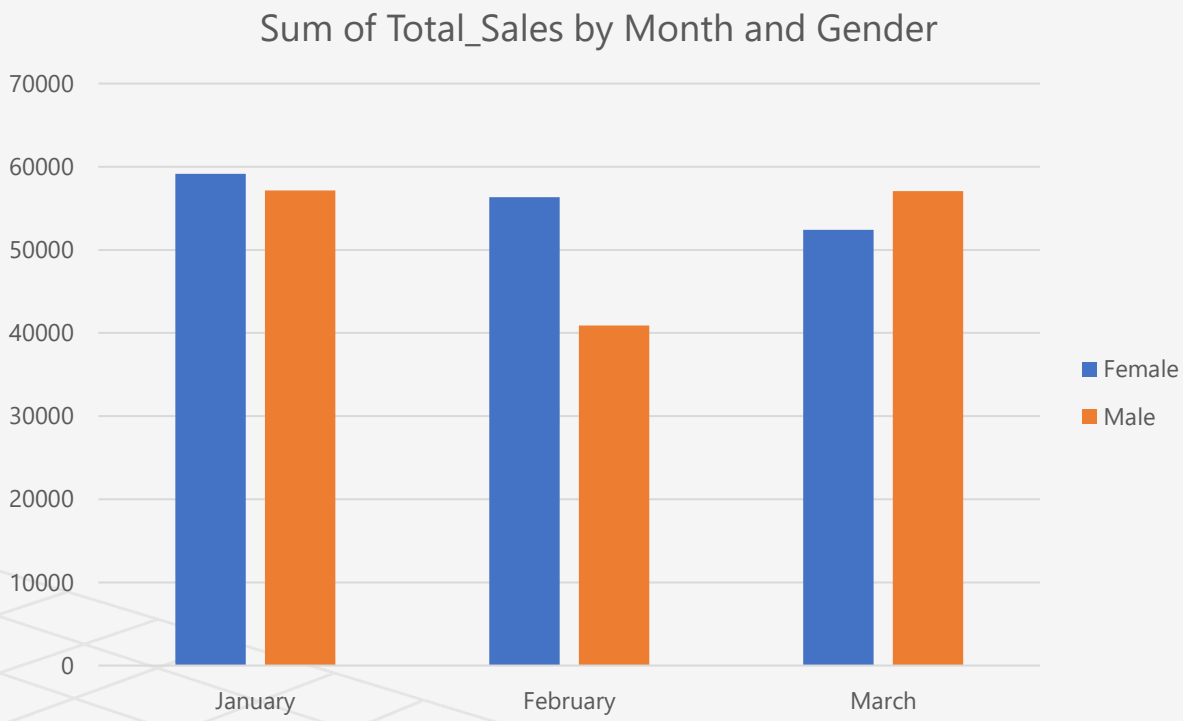
```
1 SELECT
2     DATE_FORMAT(Date, '%M') AS Month,
3     Gender,
4     SUM(Total) AS Total_Sales
5 FROM walmartsales
6 GROUP BY Month, Gender
7 ORDER BY Month, Gender;
```


Task 6: Monthly Sales Distribution by Gender.

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

Step 1: Calculate total sales for each gender on a monthly basis.

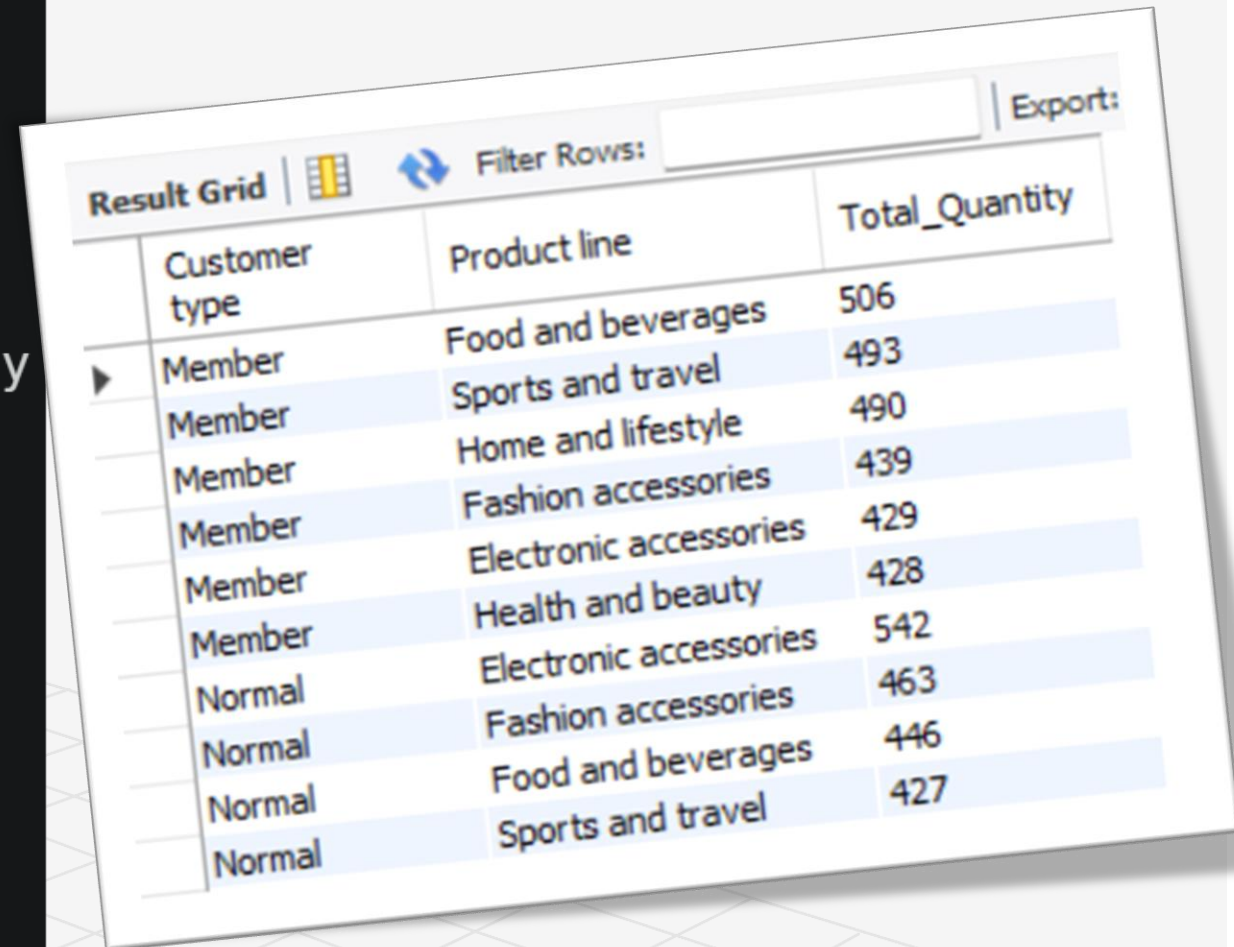
Result Grid			
Filter Rows:			
	Month	Gender	Total_Sales
▶	February	Female	56335.55549999999
	February	Male	40883.81849999999
	January	Female	59138.98200000001
	January	Male	57152.88599999999
	March	Female	52408.387500000004
	March	Male	57047.11949999997



Task 7: Best Product Line by Customer Type.

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

```
1 SELECT
2     `Customer type`,
3     `Product line`,
4     SUM(Quantity) AS Total_Quantity
5 FROM
6     walmartsales
7 GROUP BY
8     `Customer type`,
9     `Product line`
10 ORDER BY
11     `Customer type`,
12     Total_Quantity DESC;
13
```



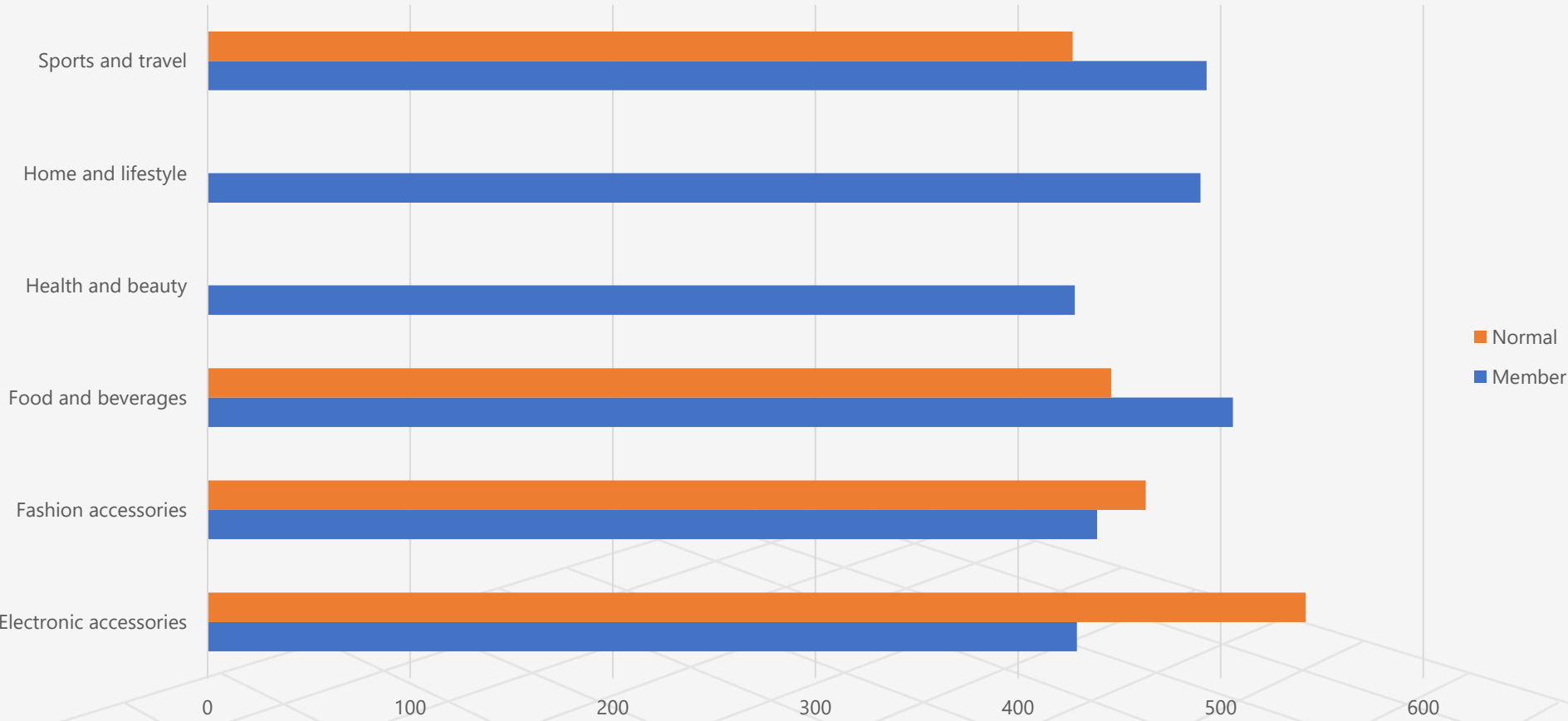
The screenshot shows a 'Result Grid' window with a table of query results. The table has three columns: 'Customer type', 'Product line', and 'Total_Quantity'. The results are sorted by 'Customer type' and then by 'Total_Quantity' in descending order. The 'Member' customer type has five entries, and the 'Normal' customer type has three entries. The 'Total_Quantity' values are displayed in the third column.

Customer type	Product line	Total_Quantity
Member	Food and beverages	506
Member	Sports and travel	493
Member	Home and lifestyle	490
Member	Fashion accessories	439
Member	Electronic accessories	429
Member	Health and beauty	428
Normal	Electronic accessories	542
Normal	Fashion accessories	463
Normal	Food and beverages	446
Normal	Sports and travel	427

Task 7: Best Product Line by Customer Type.

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

Sum of Total_Quantity by Product line and Customer type



Task 8: Identifying Repeat Customers.

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



Step 1: Identify repeat customers within 30 days.

```
1 SELECT a.`Customer ID`, COUNT(DISTINCT b.`Invoice ID`) AS repeat_purchases
2 FROM walmartsales a
3 JOIN walmartsales b
4 ON a.`Customer ID` = b.`Customer ID`
5 AND a.`Invoice ID` <> b.`Invoice ID`
6 AND DATEDIFF(a.Date, b.Date) BETWEEN 1 AND 30
7 GROUP BY a.`Customer ID`
8 HAVING repeat_purchases > 1;
```

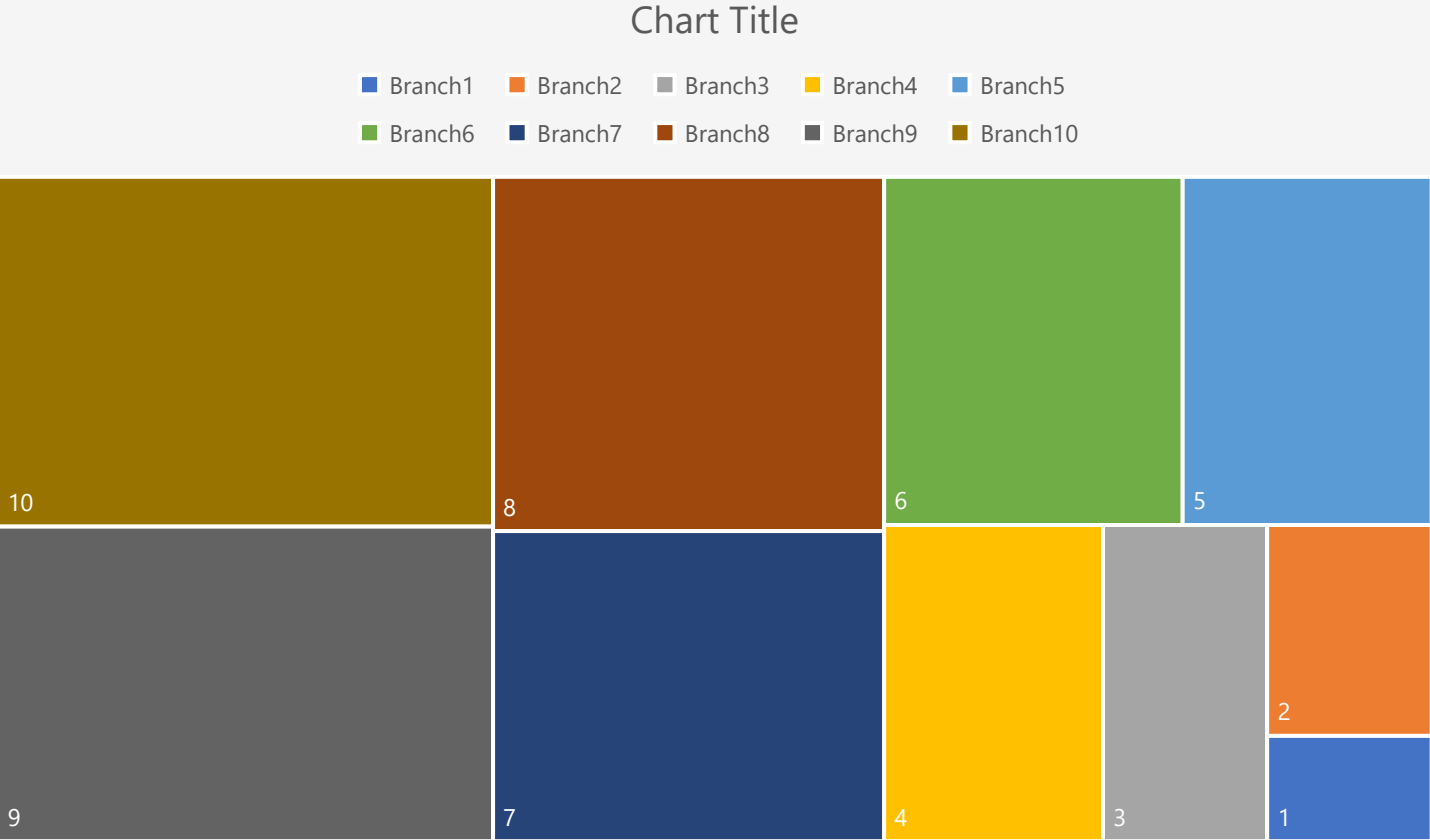
Task 8: Identifying Repeat Customers.

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

Step 1: Identify repeat customers within 30 days.

Result Grid   Filter Rows:

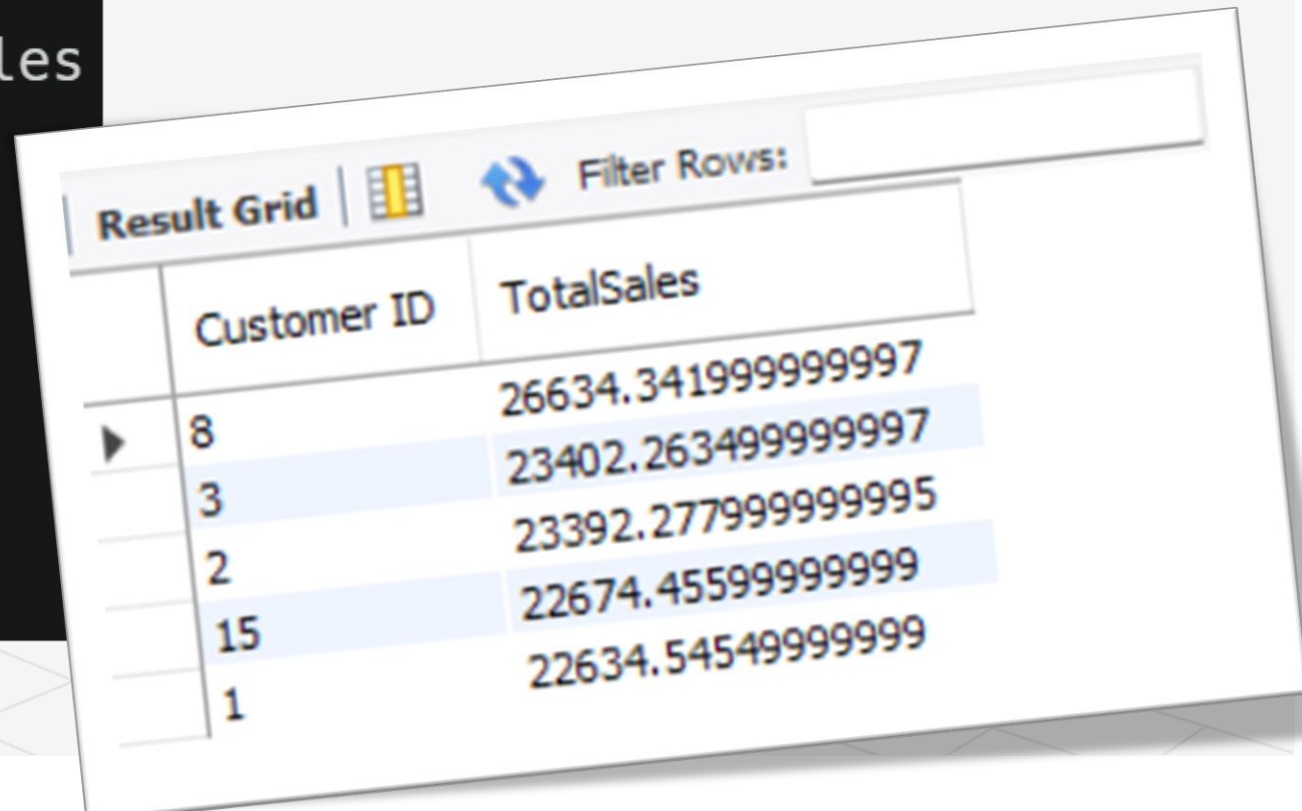
	Customer ID	repeat_purchases
	3	66
	4	66
	5	65
	6	65
	7	66
	8	65
	9	66
	10	66



Task 9: Finding Top 5 Customers by Sales Volume.

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

```
1 SELECT
2     `Customer ID`,
3     SUM(`Total`) AS TotalSales
4 FROM
5     walmartsales
6 GROUP BY
7     `Customer ID`
8 ORDER BY
9     TotalSales DESC
10 LIMIT 5;
```

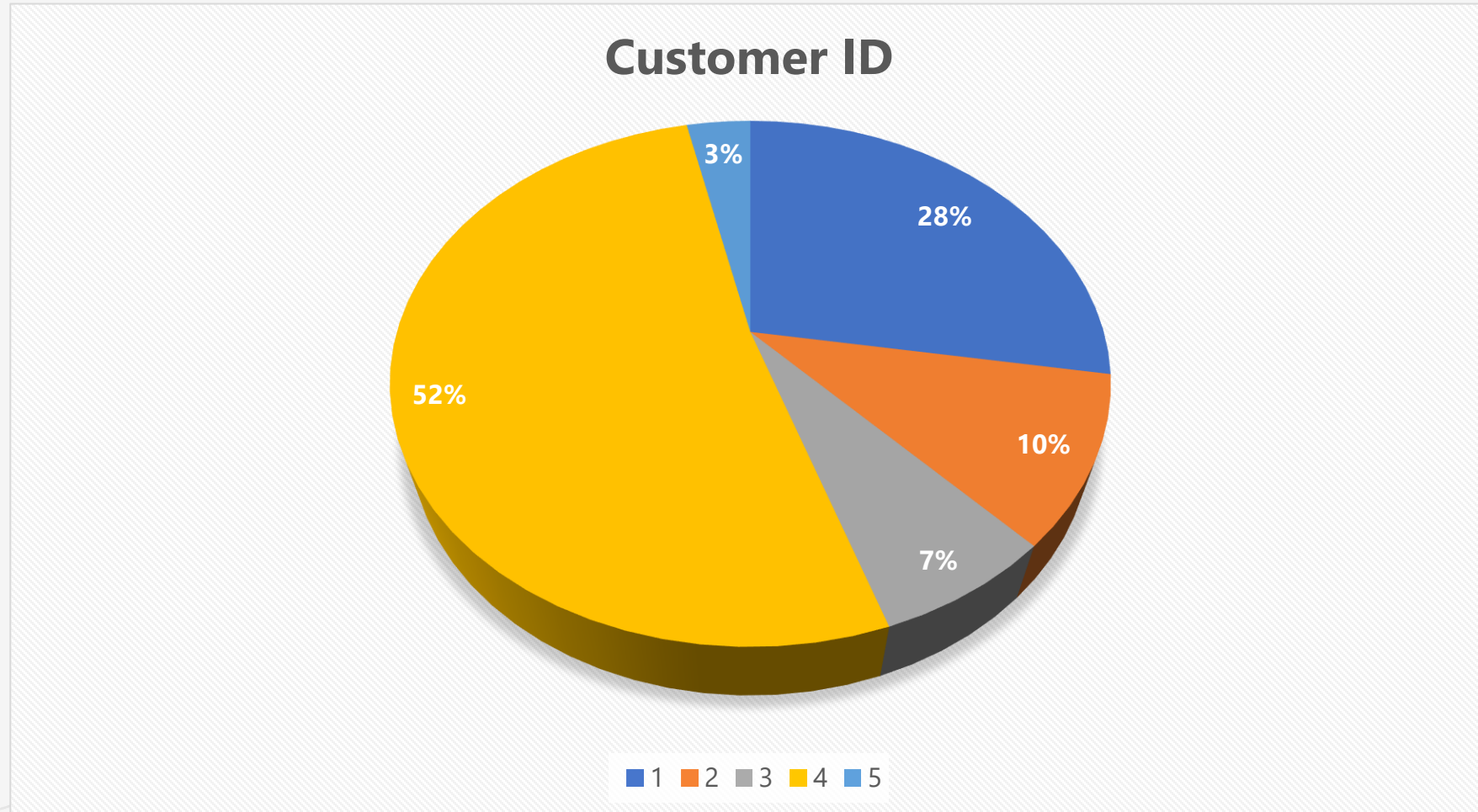


The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid displays the results of the SQL query, listing the top 5 customers by total sales in descending order. The columns are 'Customer ID' and 'TotalSales'. The data is as follows:

	Customer ID	TotalSales
▶	8	26634.341999999997
	3	23402.263499999997
	2	23392.277999999995
	15	22674.455999999999
	1	22634.545499999999

Task 9: Finding Top 5 Customers by Sales Volume.

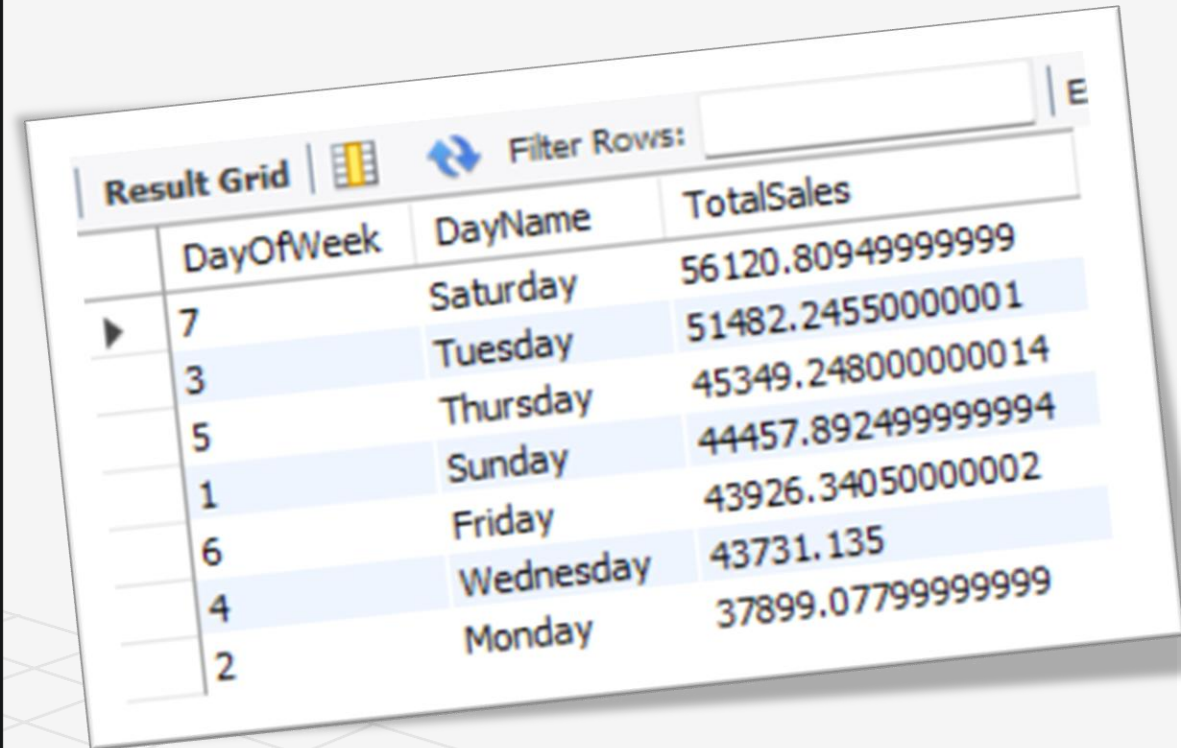
Walmart wants to reward its top 5 customers who have generated the most sales Revenue.



Task 10: 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.

```
1 SELECT
2     DAYOFWEEK(`Date`) AS DayOfWeek,
3     MAX(CASE
4         WHEN DAYOFWEEK(`Date`) = 1 THEN 'Sunday'
5         WHEN DAYOFWEEK(`Date`) = 2 THEN 'Monday'
6         WHEN DAYOFWEEK(`Date`) = 3 THEN 'Tuesday'
7         WHEN DAYOFWEEK(`Date`) = 4 THEN 'Wednesday'
8         WHEN DAYOFWEEK(`Date`) = 5 THEN 'Thursday'
9         WHEN DAYOFWEEK(`Date`) = 6 THEN 'Friday'
10        WHEN DAYOFWEEK(`Date`) = 7 THEN 'Saturday'
11    END) AS DayName,
12     SUM(`Total`) AS TotalSales
13 FROM
14     walmartsales
15 GROUP BY
16     DayOfWeek
17 ORDER BY
18     TotalSales DESC;
```



The screenshot shows a database interface with a 'Result Grid' tab. It features a 'Filter Rows' input field and a table with three columns: 'DayOfWeek', 'DayName', and 'TotalSales'. The table contains seven rows of data, sorted by 'TotalSales' in descending order. The rows are: Saturday (DayOfWeek 7, TotalSales 56120.80949999999), Tuesday (DayOfWeek 3, TotalSales 51482.24550000001), Thursday (DayOfWeek 5, TotalSales 45349.248000000014), Sunday (DayOfWeek 1, TotalSales 44457.892499999994), Friday (DayOfWeek 6, TotalSales 43926.34050000002), Wednesday (DayOfWeek 4, TotalSales 43731.135), and Monday (DayOfWeek 2, TotalSales 37899.07799999999).

DayOfWeek	DayName	TotalSales
7	Saturday	56120.80949999999
3	Tuesday	51482.24550000001
5	Thursday	45349.248000000014
1	Sunday	44457.892499999994
6	Friday	43926.34050000002
4	Wednesday	43731.135
2	Monday	37899.07799999999

Task 10: 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.

