WALMART
SALES ANALYSIS

Using MS SQL

Presented by
Pooja Sarwar

CONTENTS

INTRODUCTION

Walmart Inc. is a global retail giant founded in 1962 by Sam Walton. Headquartered in Bentonville, Arkansas, Walmart operates a chain of hypermarkets, discount department stores, and grocery stores. It is one of the largest and most recognized retail brands in the world, with over 10,000 stores across more than 20 countries and a workforce exceeding 2 million associates.

Walmart is renowned for its commitment to providing a wide range of products at competitive prices, serving millions of customers daily. The company's extensive supply chain and innovative use of technology enable it to maintain its position as a leader in the retail sector. Walmart's mission is to save people money and help them live better by offering high-quality products and services at affordable prices.

PROJECT OVERVIEW

The Walmart Sales Analysis project aims to leverage SQL to extract and analyze sales data, uncovering valuable insights into the company's performance. By focusing on key metrics such as sales trends, product wise performance, and payment methods, this project seeks to provide a comprehensive understanding of sales dynamics. The analysis involves querying data from various sources to identify patterns and trends over time. This approach enables Walmart to make informed decisions about inventory management, marketing strategies, and store operations.

Through the use of SQL, the project will generate reports on monthly sales trends, departmental contributions, and store performance. The insights derived from these analyses will help Walmart optimize its inventory, tailor promotions to specific departments and stores, and strategically plan store placements. Ultimately, the goal is to support data-driven decision-making that enhances overall business performance and customer satisfaction.

DATASET
OVERVIEW

- **Invoice_ID**: Unique identifier for each invoice.
- **Branch**: Branch of the store where the sale took place.
- **City**: City where the store branch is located.
- **Customer_type**: Type of customer (e.g., Member, Normal).
- **Gender**: Gender of the customer.
- **gross_income**: Gross income from the sale.
- **Rating**: Customer rating for the purchase.

- **Product_line**: Category of the product sold.
- **Unit_price**: Price per unit of the product.
- **Quantity**: Quantity of the product sold.
- **Tax_5%**: Tax applied on the sale.
- **Total**: Total amount of the sale.
- **Date**: Date of the transaction.
- **Time**: Time of the transaction.
- **Payment**: Payment method used (e.g., Cash, Credit Card).
- **cogs**: Cost of goods sold.
- **gross_margin_percentage**: Gross margin percentage.

ANALYSIS
USING SQL

1. Retrieve all columns for sales made in a specific branch (e.g., Branch 'A')

```
Select * ,
 format (unit_price * quantity, 'N2' ) as Sales from walmartdata  --(N2' specifies numeric value with two decimal places)
 where branch = 'A'
 order by sales desc
```

	Branch	Invoice_ID	City	Customer_type	Gender	Product_line	Unit_price	Quantity	Tax_5%	Total	Date	Time	Payment	cogs	gross_ma
1	A	687-47-8271	Yangon	Normal	Male	Fashion accessories	98.98	10.00	49.49	1039.29	02/08/19	16:20:00	Credit card	989.80	4.76
2	A	634-97-8956	Yangon	Normal	Male	Food and beverages	32.90	3.00	4.94	103.635	02/17/19	17:27:00	Credit card	98.70	4.76
3	A	643-38-7867	Yangon	Normal	Male	Home and lifestyle	97.94	1.00	4.9	102.837	03/07/19	11:44:00	Ewallet	97.94	4.76
4	A	594-34-4444	Yangon	Normal	Male	Electronic accessories	97.16	1.00	4.86	102.018	03/08/19	20:38:00	Ewallet	97.16	4.76
5	A	186-43-8965	Yangon	Member	Female	Home and lifestyle	47.68	2.00	4.77	100.128	02/24/19	10:10:00	Credit card	95.36	4.76

2. Find the total sales for each product line

```
select Product_line,
 format(sum(unit_price * quantity ),'N2') as Total_Sales from walmartdata
 group by product_line
 order by Total_sales desc
```

Results | Messages

	Product_line	Total_Sales
1	Food and beverages	53,471.28
2	Sports and travel	52,497.93
3	Electronic accessories	51,750.03
4	Fashion accessories	51,719.90
5	Home and lifestyle	51,297.06
6	Health and beauty	46,851.18

3. List all sales transactions where the payment method was 'Cash'

```
Select * ,
 format (unit_price * quantity, 'N2' ) as Sales from walmartdata
 where payment = 'cash'
 order by sales desc
```

Results | Messages

	Gender	Product_line	Unit_price	Quantity	Tax_5%	Total	Date	Time	Payment	cogs	gross_margin_percentage	gross_income	Rating	Sales
1	Female	Electronic accessories	98.84	1.00	4.94	103.782	02/15/19	11:21:00	Cash	98.84	4.76	4.94	8.40	98.84
2	Female	Home and lifestyle	16.37	6.00	4.91	103.131	02/08/19	10:58:00	Cash	98.22	4.76	4.91	7.00	98.22
3	Female	Sports and travel	98.13	1.00	4.91	103.0365	01/21/19	17:36:00	Cash	98.13	4.76	4.91	8.90	98.13
4	Male	Home and lifestyle	95.58	10.00	47.79	1003.59	01/16/19	13:32:00	Cash	955.80	4.76	47.79	4.80	955.80
5	Female	Sports and travel	95.44	10.00	47.72	1002.12	01/09/19	13:45:00	Cash	954.40	4.76	47.72	5.20	954.40

4. Calculate the total gross income generated in each city

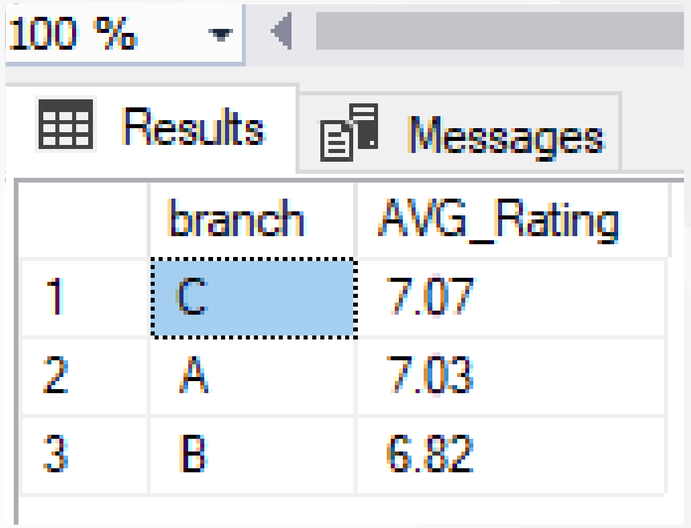
```
select city,
 format(sum(gross_income),'N2') as
Total_Gross_Income from walmartdata
 group by City
 order by Total_Gross_Income desc
```

Results | Messages

	city	Total_Gross_Income
1	Naypyitaw	5,265.33
2	Mandalay	5,057.36
3	Yangon	5,057.36

5. Find the average rating given by customers in each branch

```
Select branch,
 format ( Avg(rating), 'N2') as AVG_Rating from walmartdata
 group by branch
 order by AVG_Rating desc
```



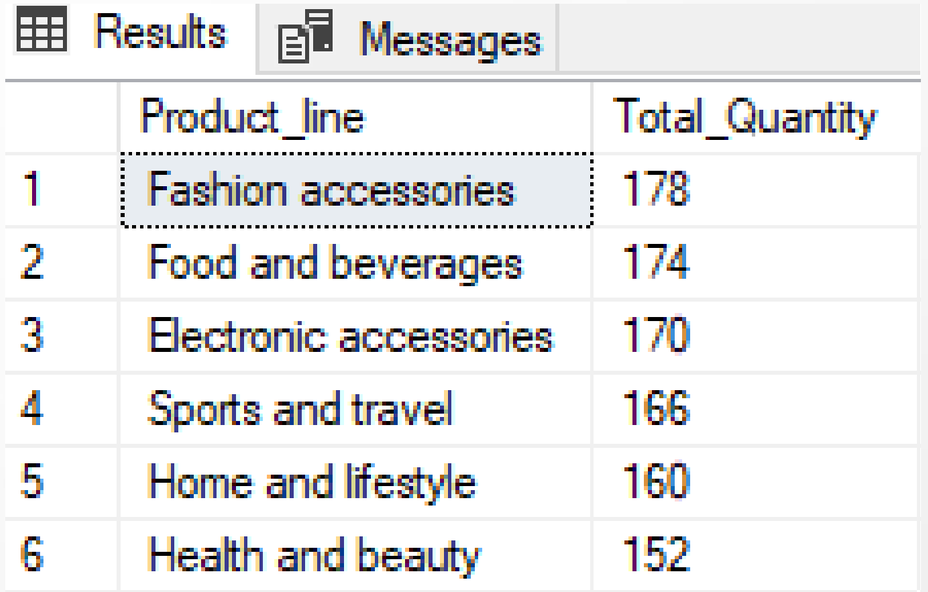
100 %

Results | Messages

	branch	AVG_Rating
1	C	7.07
2	A	7.03
3	B	6.82

6. Determine the total quantity of each product line sold

```
select Product_line,
 count(quantity ) as Total_Quantity from
walmartdata
 group by product_line
 order by Total_quantity desc
```



Results | Messages

	Product_line	Total_Quantity
1	Fashion accessories	178
2	Food and beverages	174
3	Electronic accessories	170
4	Sports and travel	166
5	Home and lifestyle	160
6	Health and beauty	152

7. List the top 5 products by unit price

```
Select top 5 Product_line,
 invoice_id,
 unit_price  from walmartdata
order by unit_price
```

Results | Messages

	Product_line	invoice_id	unit_price
1	Health and beauty	333-23-2632	10.08
2	Food and beverages	239-48-4278	10.13
3	Health and beauty	516-77-6464	10.16
4	Sports and travel	784-21-9238	10.17
5	Fashion accessories	115-38-7388	10.18

8. Find sales transactions with a gross margin percentage greater than 3%

```
Select format((unit_price * quantity), 'N2')
As Sales,
 gross_margin_percentage from
walmartdata
 where gross_margin_percentage > 3
```

Results | Messages

	Sales	gross_margin_percentage
1	522.83	4.76
2	76.40	4.76
3	324.31	4.76
4	465.76	4.76
5	604.17	4.76

9. Retrieve sales transactions that occurred on weekends

```
Select format((unit_price * quantity), 'N2') As Sales,
Date,
 datename(weekday, date) as Week_day from walmartdata
 where datename(weekday, date) in ('saturday','sunday')
```

Results | Messages

	Sales	Date	Week_day
1	522.83	01/05/19	Saturday
2	324.31	03/03/19	Sunday
3	465.76	01/27/19	Sunday
4	735.60	02/24/19	Sunday
5	102.04	03/09/19	Saturday
6	172.80	02/17/19	Sunday

10. Calculate the total sales and gross income for each month

```
Select datename(Month,date) As
Month_Name,
 format(sum((unit_price * quantity)), 'N2')
As Total_Sales,
 sum(gross_income) As
Total_Gross_Income from walmartdata
group by datename(Month,date)
```

Results | Messages

	Month_Name	Total_Sales	Total_Gross_Income
1	February	92,589.88	4629.70
2	January	110,754.16	5537.95
3	March	104,243.34	5212.40

11. Find the number of sales transactions that occurred after 6 PM

```
select count(unit_price * quantity) As Sales_count_after_6PM,
 datename(month,date) as Month_name from walmartdata
 where time > '18:00:00'
group by datename(month,date)
order by sales_count_after_6PM desc
```

Results | Messages

	Sales_count_after_6PM	Month_name
1	105	March
2	94	January
3	80	February

12. List the sales transactions that have a higher total than the average total of all transactions

```
select format((unit_price * quantity), 'N2')
As Sales from walmartdata
 where (unit_price * quantity) > (select
avg(unit_price * quantity) from
walmartdata )
order by sales
```

Results | Messages

	Sales
1	307.68
2	307.76
3	308.85
4	309.88
5	310.72
6	310.88

13. Find customers who made more than 5 purchases in a single month

```
select invoice_id,customer_type,
 datename(month,date) as Month_name,
 quantity from walmartdata
 where quantity > 5
```

Results | Messages

	invoice_id	customer_type	Month_name	quantity
1	750-67-8428	Member	January	7.00
2	631-41-3108	Normal	March	7.00
3	123-19-1176	Member	January	8.00
4	373-73-7910	Normal	February	7.00
5	699-14-3026	Normal	March	7.00
6	355-53-5943	Member	February	6.00

14. Calculate the cumulative gross income for each branch by date

```
select branch,
 date,
 sum(gross_income) over (partition by
branch order by date ) as
cumulative_gross_income from
walmartdata
order by branch, date
```

Results | Messages

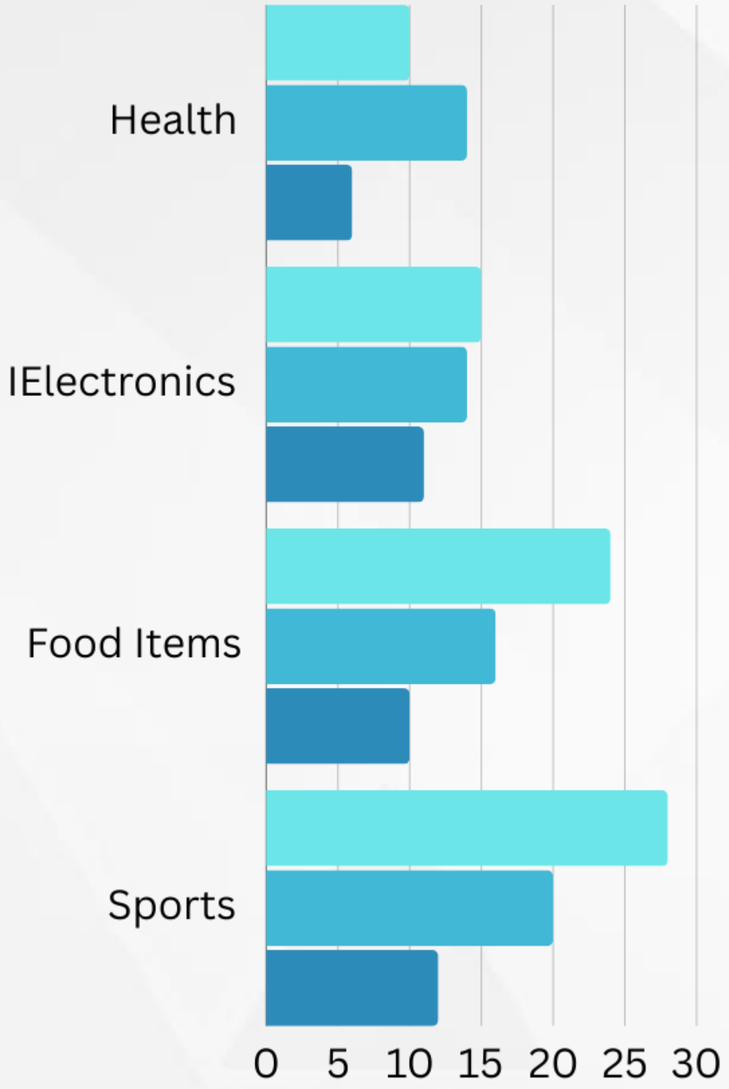
	branch	date	cumulative_gross_income
1	A	01/01/19	112.92
2	A	01/01/19	112.92
3	A	01/01/19	112.92
4	A	01/01/19	112.92
5	A	01/01/19	112.92
6	A	01/02/19	127.54

15. Find the total cogs for each customer type in each city

```
Select city,
 customer_type,
 sum(cogs) as Total_COGS from walmartdata
group by city, customer_type
order by city
```

Results | Messages

	city	customer_type	Total_COGS
1	Mandalay	Member	51147.32
2	Mandalay	Normal	49993.32
3	Naypyitaw	Member	54172.65
4	Naypyitaw	Normal	51130.88
5	Yangon	Member	51083.31
6	Yangon	Normal	50059.90

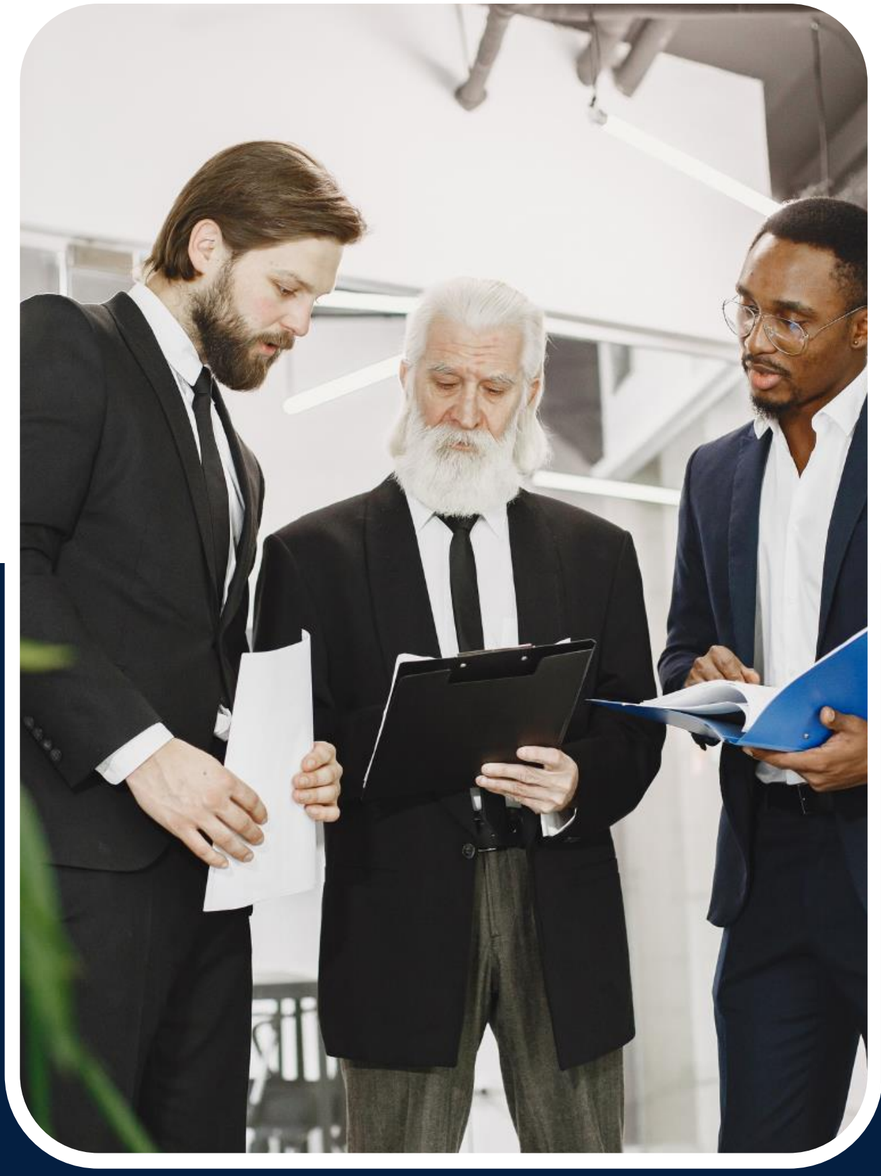


FINDINGS

The SQL queries provided facilitate a comprehensive analysis of Walmart's sales data across various dimensions. By retrieving detailed transaction records for specific branches, calculating total sales for each product line, and examining payment methods, we gain valuable insights into sales performance, customer preferences, and financial metrics. For instance, determining gross income by city and assessing average customer ratings by branch allows Walmart to pinpoint high-performing areas and understand customer satisfaction. Additionally, identifying high-value transactions and weekend sales patterns helps in fine-tuning inventory management and marketing strategies.

CONCLUSION

The comprehensive analysis of Walmart's sales data reveals critical insights into sales performance and operational efficiency. By examining transaction details across branches and cities, evaluating product line performance, and analyzing customer payment methods and ratings, we gain a nuanced understanding of factors driving revenue and customer satisfaction.



The detailed findings, such as the cumulative gross income by branch and the distribution of COGS by customer type and city, further refine our understanding of profitability and cost management. By leveraging these insights, Walmart can implement targeted strategies to enhance sales, optimize resource allocation, and improve overall customer experience. Ultimately, the analysis not only supports data-driven decision-making but also positions Walmart to adapt to evolving market conditions and drive sustained growth.