Sales Performance Analysis of Walmart Stores Using MySQL

By : THENNARASU M

Walmart

INTRODUCTION

Walmart, a major retail chain, operates across several cities, offering a wide range of products. This dataset contains detailed transaction data, including customer demographics, product lines, sales figures and payment methods. This project used advanced SQL techniques to uncover actionable insights into sales performance, customer behaviour, and operational efficiencies

BUSINESS PROBLEMS

Walmart wants to optimize its sales strategies by analyzing historical transaction data across branches, customer types, payment methods, and product lines. To achieve this, advanced MySQL queries will be employed to answer challenging business questions related to sales performance, customer segmentation, and product trends

Walmart

BASIC SETUP

```sql
create database walmart;
use walmart;
show tables;
select * from wmt1;
desc wmt1;
set sql_safe_updates=0;
update wmt1 set date = str_to_date(date,'%d-%m-%Y');
alter table wmt1 modify column date date;
update wmt1 set time = str_to_date(time,'%H:%i:%s');
alter table wmt1 modify column time time;
select count(total) from wmt1 where total is null;
```

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

```
13    # KPI_01 IDENTIFYING TOP BRANCH BY SALES GROWTH RATE
14
15    with growth as(select branch,date_format(date,'%Y-%m') as month,sum(total) as current_sale,
16    lag(sum(total))
17    over(partition by branch order by date_format(date,'%Y-%m')) as last_sale
18    from wmt1 group by branch,month)
19    select branch,round(avg(((last_sale - current_sale)/(last_sale))*100),2) as avg_growth_rate
20    from growth group by branch
21    order by avg_growth_rate desc limit 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

branch	avg_growth_rate
B	3.45

TASK 2: FINDING THE MOST PROFITABLE PRODUCT LINE FOR EACH BRANCH

WALMART NEEDS TO DETERMINE WHICH PRODUCT LINE CONTRIBUTES THE HIGHEST ROFIT TO EACH BRANCH.THE PROFIT MARGIN SHOULD BE CALCULATED BASED ON THE DIFFERENCE BETWEEN THE GROSS INCOME AND COST OF GOODS SOLD

```
23      # KPI_02 FINDING THE MOST PROFITABLE PRODUCT LINE FOR EACH BRANCH
24
25  •   with p_line as(select branch,`product line`,sum(`gross income`) as income,
26      rank() over (partition by branch order by sum(`gross income`) desc) as rank_order
27      from wmt1 group by branch,`product line`)
28      select branch,`product line` from p_line where rank_order=1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

branch	product line
A	Home and lifestyle
B	Sports and travel
C	Food and beverages

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

```
30      # KPI_03 ANALYZING CUSTOMER SEGMENTATION BASED ON SPENDING
31      select distinct(`customer ID`) from wmt1;
32
33      with customertype as(select `customer id`,sum(total) as spending,
34      ntile(3) over(order by sum(total)desc) as spending_type
35      from wmt1 group by `customer id`)
36      select `customer id`,spending,
37      case
38      when spending_type=1 then 'High'
39      when spending_type=2 then 'medium'
40      else 'low'
41      end as spending_category
42      from customertype;
```

customer id	spending	spending_category
8	26634.34199999997	High
3	23402.26349999997	High
2	23392.27999999995	High
15	22674.4599999999	High
1	22634.5454999999	High
12	21720.646500000003	medium

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

```sql
# KPI_04 DETECTING ANOMALIES IN SALES TRANSACTIONS
#company detects some flagged transaction
#which is 30% above or below the averge productline sales
with productline_avg  as (
select `product line`,avg(total) as average_total from wmt1 group by `product line`)
select w.`invoice id`,w.branch,w.`product line`,w.total,p.average_total,
case
when p.average_total*1.3 < w.total then 'anomaly'
when p.average_total*0.7 > w.total then 'anomaly'
else 'normal'
end as status
from wmt1 as w join productline_avg as p on p.`product line`=w.`product line`
where p.average_total*1.3 < w.total or p.average_total*0.7 > w.total ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

invoice id	branch	product line	total	average_total	status
362-58-8315	C	Fashion accessories	401.73	305.089297752809	anomaly
381-20-0914	A	Fashion accessories	189.0945	305.089297752809	anomaly
263-10-3913	C	Fashion accessories	463.428	305.089297752809	anomaly
796-12-2025	C	Fashion accessories	652.26	305.089297752809	anomaly
109-28-2512	B	Fashion accessories	614.943	305.089297752809	anomaly
778-71-5554	C	Fashion accessories	16.2015	305.089297752809	anomaly
616-24-2851	B	Fashion accessories	75.054	305.089297752809	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

```
58      # KPI_05 MOST POPULAR PAYMENT METHOD BY CITY
59
60      with popular as (select city,payment,count(*) as popularity,
61      rank() over (partition by city order by count(*) desc) as rnk
62      from wmt1 group by city,payment order by city,payment )
63      select city,payment,popularity from popular where rnk=1;
64
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

city	payment	popularity
Mandalay	Ewallet	113
Naypyitaw	Cash	124
Yangon	Ewallet	126

TASK 6: MONTHLY SALES DISTRIBUTION BY GENDER (6 MARKS) WALMART WANTS TO UNDERSTAND THE SALES DISTRIBUTION BETWEEN MALE AND FEMALE CUSTOMERS ON A MONTHLY BASIS

```
65      # KPI_06 MONTHLY SALES DISTRIBUTION BY GENDER
66
67  •   select year(date) as year,month(date) as month,gender,round(sum(total),2) as distribution_of_sales from wmt
68      group by year(date),month(date),gender order by year(date),month(date);
69
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

year	month	gender	distribution_of_sales
2019	1	Female	59138.98
2019	1	Male	57152.89
2019	2	Female	56335.56
2019	2	Male	40883.82
2019	3	Female	52408.39
2019	3	Male	57047.12

TASK 7: BEST PRODUCT LINE BY CUSTOMER TYPE (6 MARKS) WALMART WANTS TO KNOW WHICH PRODUCT LINES ARE PREFERRED BY DIFFERENT CUSTOMER TYPES(MEMBER VS. NORMAL)

```
70      # KPI_07 BEST PRODUCT LINE BY CUSTOMER TYPE
71
72  •   select `customer type`,`product line`,purchase_frequency
73      from (select `customer type`,`product line`,count(*) as purchase_frequency,
74      rank() over (partition by `customer type` order by count(*) desc) as rnk
75      from wmt1 group by `customer type`,`product line`)as ranked_data where rnk=1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer type	product line	purchase_frequency
Member	Food and beverages	94
Normal	Electronic accessories	92
Normal	Fashion accessories	92

TASK 8: IDENTIFYING REPEAT CUSTOMERS WALMART NEEDS TO IDENTIFY CUSTOMERS WHO MADE REPEAT PURCHASES WITHIN A SPECIFIC TIME FRAME (E.G., WITHIN 30 DAYS)

```sql
# KPI_08 IDENTIFYING REPEAT CUSTOMERS
SELECT * FROM (
    SELECT
        `customer id`,
        `invoice id`,
        date AS current_purchase_date,
        LAG(date) OVER (PARTITION BY `customer id` ORDER BY date) AS previous_purchase_date,
        DATEDIFF(date, LAG(date) OVER (PARTITION BY `customer id` ORDER BY date)) AS days_between_purchases,
        CASE
            WHEN LAG(date) OVER (PARTITION BY `customer id` ORDER BY date) IS NULL THEN 'First-time Customer'
            WHEN DATEDIFF(date, LAG(date) OVER (PARTITION BY `customer id` ORDER BY date)) <= 30 THEN 'Frequent Buyer'
            ELSE 'Occasional Buyer'
        END AS customer_type
    FROM wmt1
) subquery
WHERE customer_type != 'First-time Customer';
```

customer id	invoice id	current_purchase_date	previous_purchase_date	days_between_purchases	customer_type
1	662-72-2873	2019-01-06	2019-01-05	1	Frequent Buyer
1	665-32-9167	2019-01-10	2019-01-06	4	Frequent Buyer
1	744-02-5987	2019-01-10	2019-01-10	0	Frequent Buyer
1	848-07-1692	2019-01-12	2019-01-10	2	Frequent Buyer
1	751-41-9720	2019-01-12	2019-01-12	0	Frequent Buyer

TASK 9: FINDING TOP 5 CUSTOMERS BY SALES VOLUME

WALMART WANTS TO REWARD ITS TOP 5 CUSTOMERS WHO HAVE GENERATED THE MOST SALES REVENUE

```
94      # KPI_09 FINDING TOP 5 CUSTOMERS BY SALES VOLUME
95
96  •   select `customer id`,round(sum(cogs),2) as sales_volume
97      from wmt1 group by `customer id` order by sum(cogs) desc limit 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch r

customer id	sales_volume
8	25366.04
3	22287.87
2	22278.36
15	21594.72
1	21556.71

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.

```
 99      # KPI_10 ANALYZING THE SALES TRENDS BY THE DAYS OF A WEEK
100
101 •   select dayname(date) as Days,round(sum(cogs),2) as Total_sales
102     from wmt1 group by days order by Total_sales desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Days	Total_sales
Saturday	53448.39
Tuesday	49030.71
Thursday	43189.76
Sunday	42340.85
Friday	41834.61
Wednesday	41648.7