SQL final project

Project video

Note: I use CTE for most of my solutions as it is easier to understand and can be reused down the level of queries.

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.

Ans:

```
-- get the initial required data

WITH monthly_sales AS (
  SELECT
    Branch,
    DATE_FORMAT(STR_TO_DATE(Date, '%d-%m-%Y'), '%Y-%m') AS month,
    SUM(Total) AS total_sales
  FROM walmartsales_dataset
  GROUP BY Branch, month
),

-- Aggregate total sales per branch per month

growth_calc AS (
  SELECT
    Branch,
    month,
    total_sales,
    LAG(total_sales) OVER (PARTITION BY Branch ORDER BY month) AS prev_month_sales
  FROM monthly_sales
),

-- Calculate the month-over-month growth rate

growth_rate_per_month AS (
  SELECT
    Branch,
    month,
    ROUND(((total_sales - prev_month_sales) / prev_month_sales) * 100, 2) AS monthly_growth
  FROM growth_calc
  WHERE prev_month_sales IS NOT NULL
),
```

```
-- Average (or sum) the growth rates per branch

branch_avg_growth AS (
  SELECT
    Branch,
    ROUND(AVG(monthly_growth), 2) AS avg_monthly_growth
  FROM growth_rate_per_month
  GROUP BY Branch
)

-- Find the branch with the highest average/sum of growth rates

SELECT
  Branch,
  avg_monthly_growth
FROM branch_avg_growth
ORDER BY avg_monthly_growth DESC
LIMIT 1;
```

We find <u>Branch - B</u> is the top branch by sales growth

Result Grid | Filter Rows:

Branch	avg_monthly_growth
B	12.24
C	9.54
A	-9

Limit to 500 rows

```
48    )
49
50    -- Find the branch with the highest average/sum of growth rates
51
52    SELECT
53        Branch,
54        avg_monthly_growth
55    FROM branch_avg_growth
56    ORDER BY avg_monthly_growth DESC
57    LIMIT 1;
58
59
```

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

Branch	avg_monthly_growth
B	12.24

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.

```
WITH product_profit AS (
   SELECT
      Branch,
      `Product line` AS product_line,
      SUM(`gross income`) AS total_profit
   FROM walmartsales_dataset
   GROUP BY Branch, product_line
),

-- ranking every product line in term of total_profit in every branch and giving it a rank

ranked_profit AS (
   SELECT *,
      RANK() OVER (PARTITION BY Branch ORDER BY total_profit DESC) AS rnk
   FROM product_profit
)

-- Selecting all branches top 1 rank product line
SELECT
   Branch,
   product_line,
   ROUND(total_profit, 2) AS total_profit
FROM ranked_profit
WHERE rnk = 1;
```

query result

	Branch	product_line	total_profit
▶	A	Home and lifestyle	1067.49
	B	Sports and travel	951.82
	C	Food and beverages	1131.75

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.

Ans:

```
WITH customer_avg AS (
   SELECT
      `Customer ID`,
      AVG(Total) AS avg_spend
   FROM walmartsales_dataset
   GROUP BY `Customer ID`
),

min_max_vals AS (
   SELECT
      MIN(avg_spend) AS min_val,
      MAX(avg_spend) AS max_val
   FROM customer_avg
),

/* 1/3, 2/3  divides the range in equal parts for every case */

classified_customers AS (
   SELECT
      c.`Customer ID`,
      c.avg_spend,
      m.min_val,
      m.max_val,
      (m.max_val - m.min_val) AS range_val,
      CASE
         WHEN c.avg_spend <= m.min_val + (m.max_val - m.min_val) * 1/3 THEN 'Low'
         WHEN c.avg_spend <= m.min_val + (m.max_val - m.min_val) * 2/3 THEN 'Medium'
         ELSE 'High'
      END AS spending_tier
   FROM customer_avg c
   JOIN min_max_vals m
)

--- list all the records
```

```
SELECT `Customer ID`, ROUND(avg_spend,2) AS avg_spend, spending_tier
FROM classified_customers
ORDER BY avg_spend DESC;
```

Result Grid | Filter Rows:

Customer ID	avg_spend	spending_tier
8	397.53	High
3	349.29	Medium
2	349.14	Medium
15	343.55	Medium
1	337.83	Medium
12	329.1	Medium
11	324.22	Medium
13	319.15	Medium
14	318.93	Medium
10	309.31	Medium
6	308.87	Medium
7	307.88	Low
9	293.46	Low
5	293.02	Low

Result 2

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.

Ans:

```
select
   `Invoice ID`,
   `Product line`,
   Total,
   (select avg(Total) from walmartsales_dataset as w2 where w2.`Product line` = w1.`Product line`) as avg_total
FROM walmartsales_dataset as w1
WHERE
   Total > (SELECT avg(Total) from walmartsales_dataset as w2 where w2.`Product line` = w1.`Product line`) * 2
   or
   Total < (SELECT avg(Total) from walmartsales_dataset as w2 where w2.`Product line` = w1.`Product line`) * 0.01;
```

If deviation is almost 100% from average we classify it as an anomalie.

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

Invoice ID	Product line	Total	avg_total
750-67-8428	Health and beauty	548.9715	323.64301973684223
226-31-3081	Electronic accessories	80.22	319.63253823529413
123-19-1176	Health and beauty	489.048	323.64301973684223
373-73-7910	Sports and travel	634.3785	332.06521987951805
699-14-3026	Electronic accessories	627.6165	319.63253823529413
315-22-5665	Home and lifestyle	772.38	336.63695625
665-32-9167	Health and beauty	76.146	323.64301973684223
351-62-0822	Fashion accessories	60.816	305.089297752809
529-56-3974	Electronic accessories	107.142	319.63253823529413
829-34-3910	Health and beauty	749.49	323.64301973684223
299-46-1805	Sports and travel	590.436	332.06521987951805
656-95-9349	Health and beauty	506.6355	323.64301973684223

336.63695625

Result 7

Task 5: Most Popular Payment Method by City

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

Ans:

```
with payment_counts as (
   select
      City,
      Payment,
      COUNT(*) as payment_count
   from walmartsales_dataset
   group by City, Payment
),
ranked_payments as (
   select
      *,
      rank() OVER (partition by City order by payment_count desc) as payment_rank
   from payment_counts
)
select
   City,
   Payment as most_popular_payment,
   payment_count
from ranked_payments
where payment_rank = 1
order by City;
```

query result

Result Grid | Filter Rows: | Export: | W

City	most_popular_payment	payment_count
Mandalay	Ewallet	113
Naypyitaw	Cash	124
Yangon	Ewallet	126

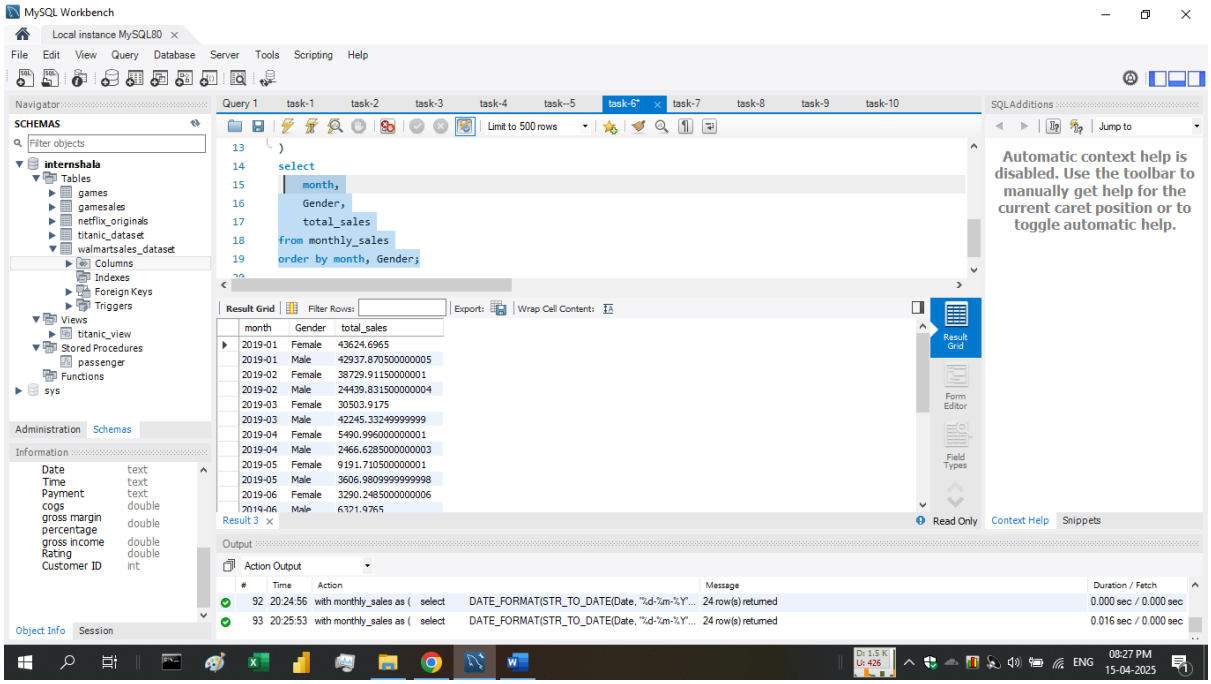
Result 2 ×

Task 6: Monthly Sales Distribution by Gender

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

Ans:

```
with monthly_sales as (
  select
      DATE_FORMAT(STR_TO_DATE(Date, '%d-%m-%Y'), '%Y-%m') as month,
      Gender,
      SUM(Total) as total_sales
   from walmartsales_dataset
   group by month, Gender
)
select
   month,
   Gender,
   total_sales
from monthly_sales
order by month, Gender;
```



Task 7: Best Product Line by Customer Type

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

Ans:

```
-- times each product line was bought by each customer type.

with product_popularity as (
   select
      `Customer type`,
      `Product line`,
      count(*) as purchase_count
   from walmartsales_dataset
   group by `Customer type`, `Product line`
),
-- Rank the product lines by their popularity within each customer type.
ranked_products as (
   select
      *,
      rank() OVER (PARTITION BY `Customer type` order by purchase_count desc) as
product_rank
   from product_popularity
)
select
   `Customer type`,
   `Product line` AS most_preferred_product_line,
   purchase_count
from ranked_products
where product_rank = 1
order by `Customer type`;
```

Result Grid | Filter Rows: | Export: | Wrap Cel

Customer type	most_preferred_product_line	purchase_count
Member	Food and beverages	94
Normal	Electronic accessories	92
Normal	Fashion accessories	92

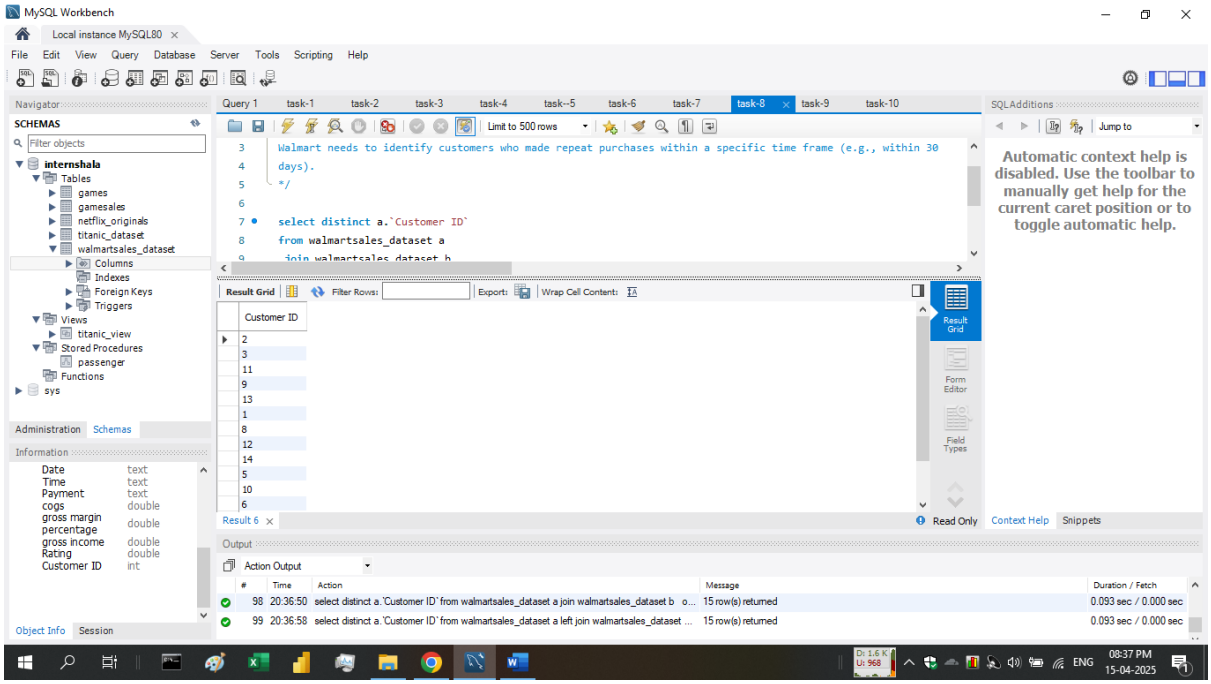
Result 3

Task 8: Identifying Repeat Customers

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

Ans:

```
select distinct a.`Customer ID`
from walmartsales_dataset a
 join walmartsales_dataset b
  on a.`Customer ID` = b.`Customer ID`
  and a.`Invoice ID` != b.`Invoice ID`
  and ABS(DATEDIFF(STR_TO_DATE(a.Date, '%d-%m-%Y'), STR_TO_DATE(b.Date, '%d-%m-%Y'))) <= 30;
```



Task 9: Finding Top 5 Customers by Sales Volume

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

Ans:

```
select
        `Customer ID`,
  round(SUM(Total), 2) as total_revenue
from walmartsales_dataset
group by `Customer ID`
order by total_revenue DESC
limit 5;
```

Result Grid | Filter Rows:

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

Result 3

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.

Ans:

```
select
   dayname(STR_TO_DATE(Date, '%d-%m-%Y')) as day_of_week,
   round(SUM(Total), 2) as total_sales
from walmartsales_dataset
group by day_of_week
order by total_sales desc;
```

Result Grid | Filter Rows:

day_of_week	total_sales
Tuesday	54630.22
Sunday	49704.6
Wednesday	47221.21
Saturday	46842.79
Thursday	45166.16
Friday	44352.87
Monday	35048.91

result shows that Tuesday is most profitable day of the week in most cases.

Result 1