SQL final project

## 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

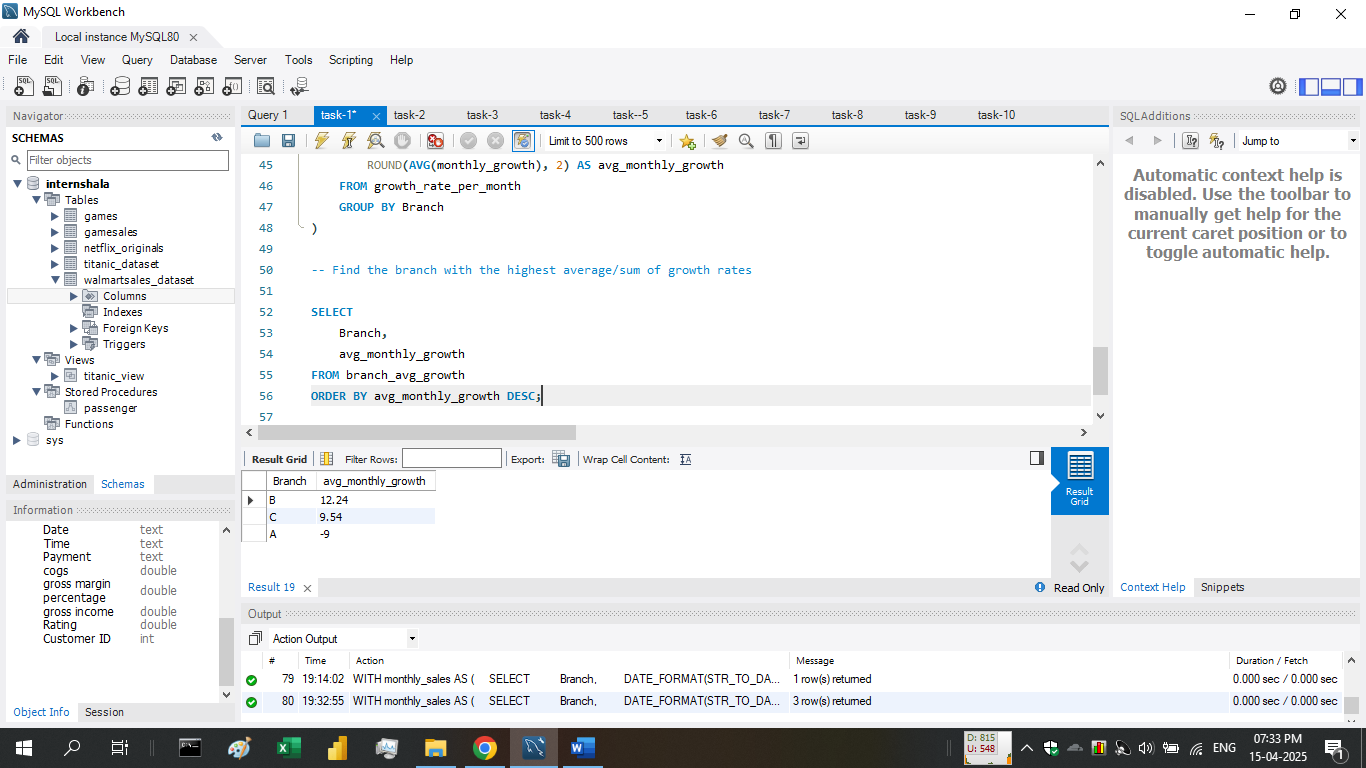
SELECT

Branch,

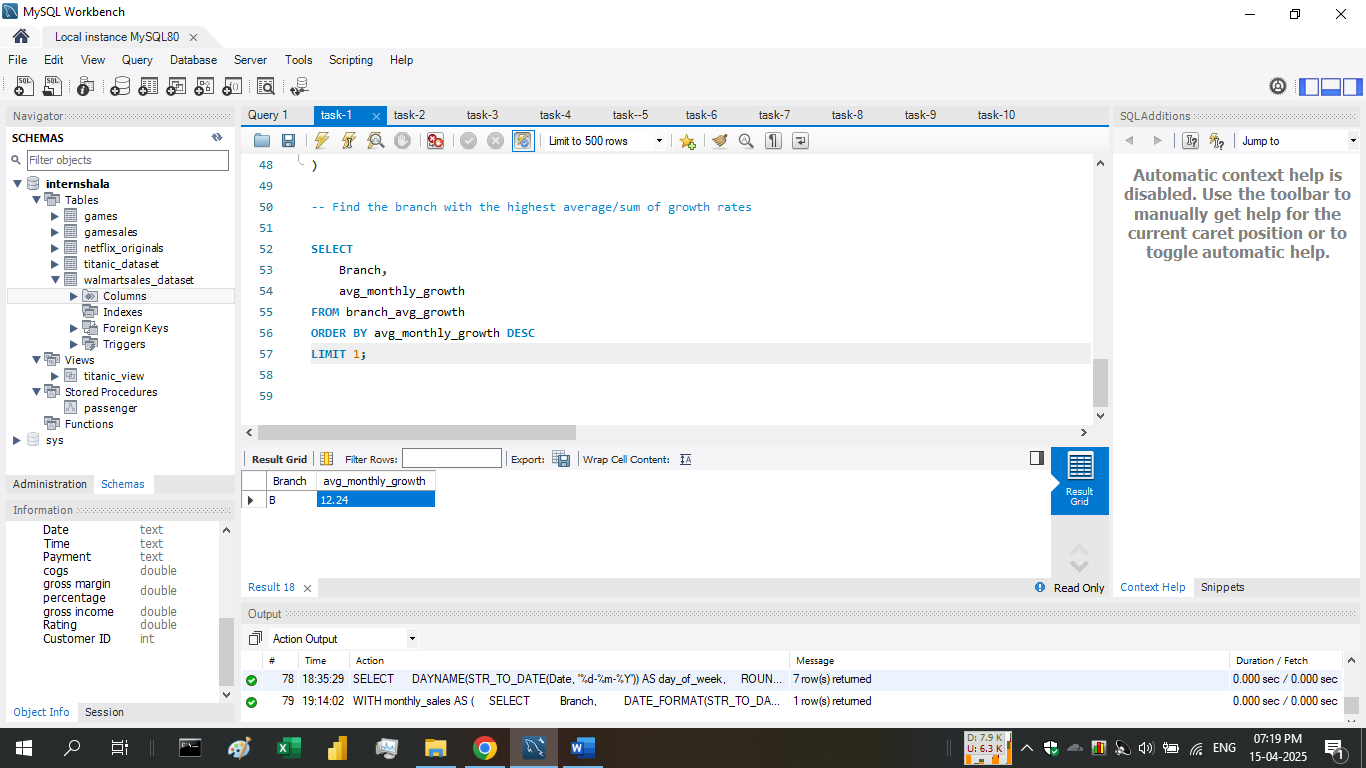
avg\_monthly\_growth

FROM branch\_avg\_growth

ORDER BY avg\_monthly\_growth DESC

LIMIT 1;

**We find Branch - B is the top branch by sales growth**



## 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,

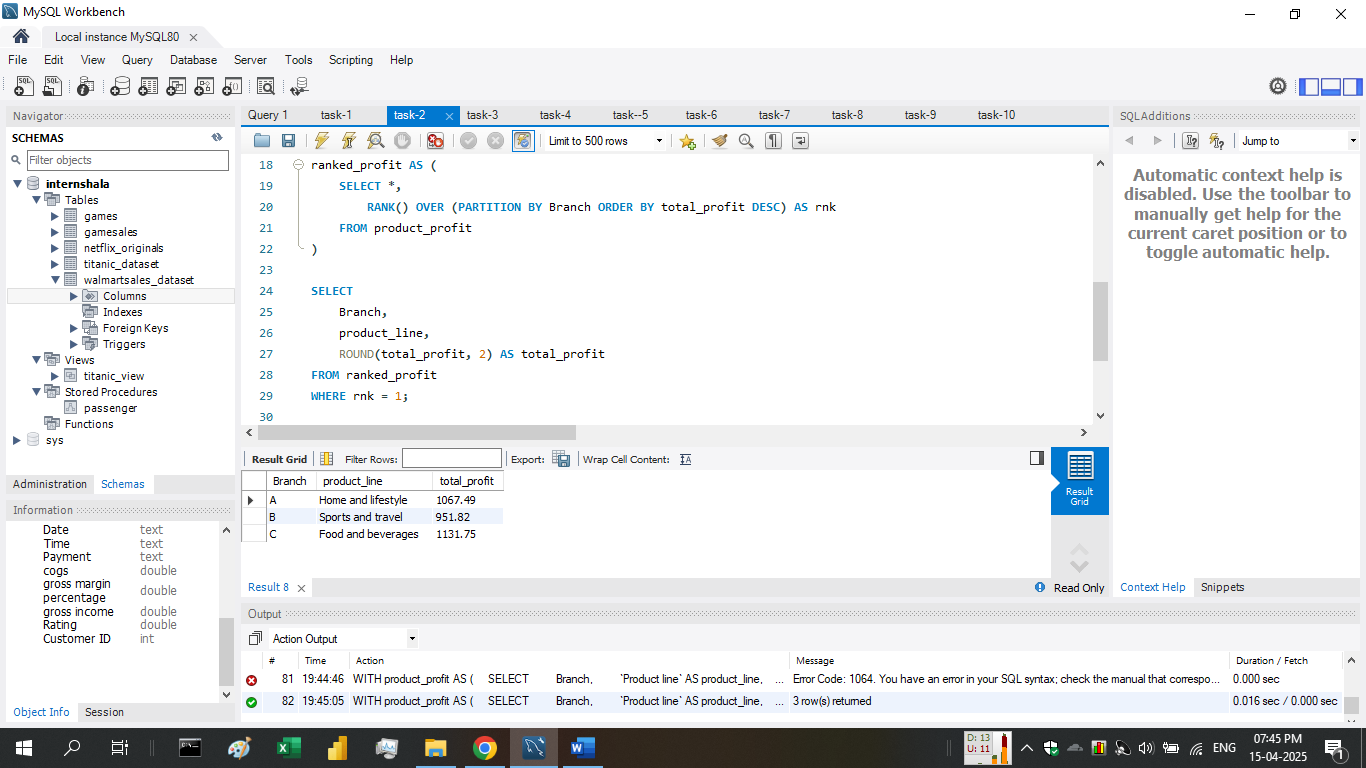
product\_line,

ROUND(total\_profit, 2) AS total\_profit

FROM ranked\_profit

WHERE rnk = 1;

query result



## 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

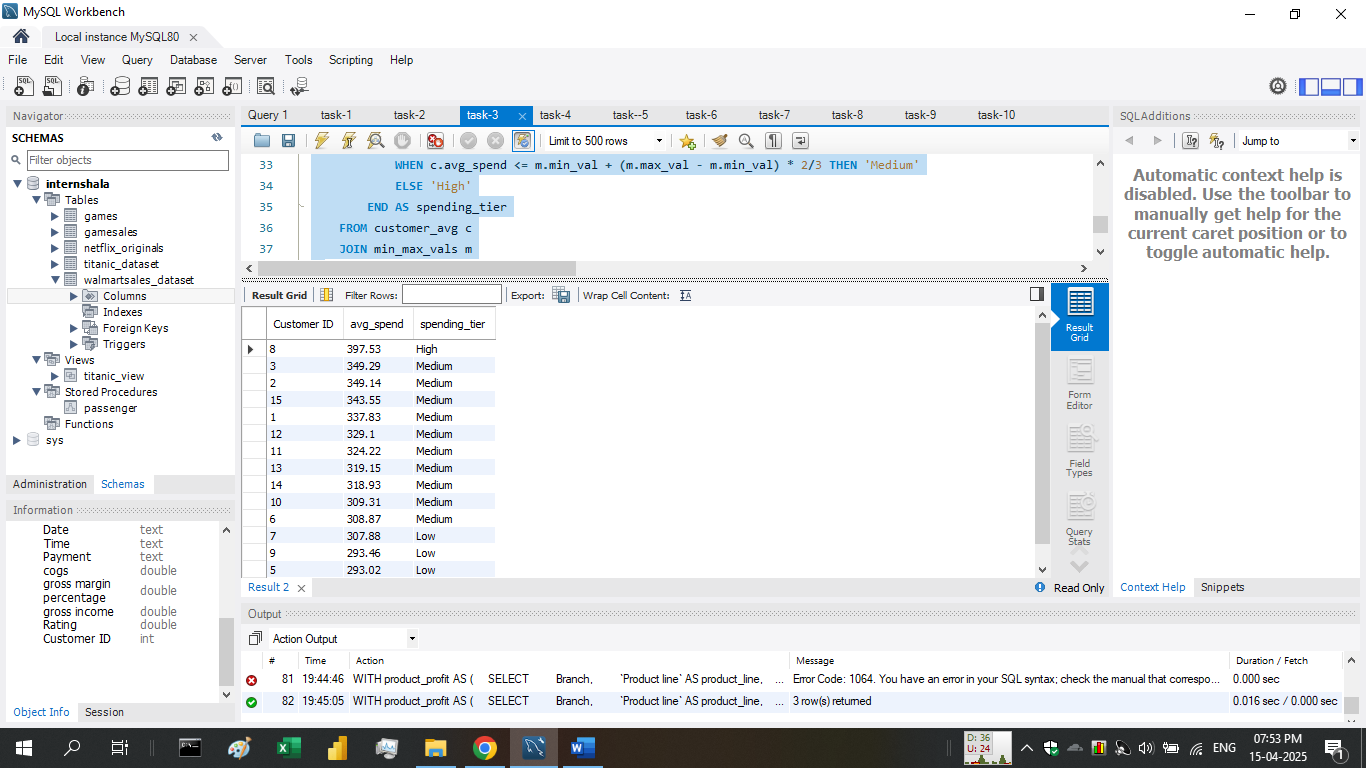
)

--- list all the records

SELECT `Customer ID`, ROUND(avg\_spend,2) AS avg\_spend, spending\_tier

FROM classified\_customers

ORDER BY avg\_spend DESC;



## 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

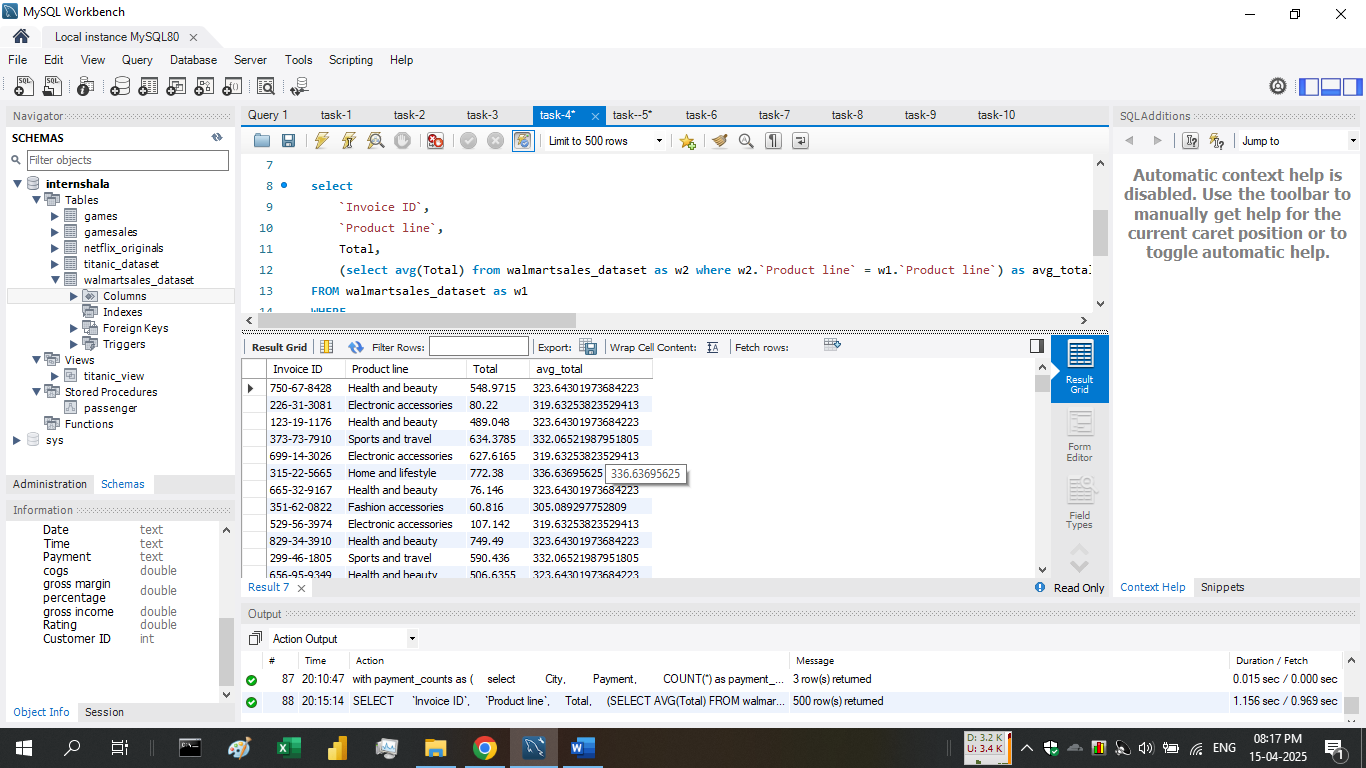
WHERE

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.



## 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,

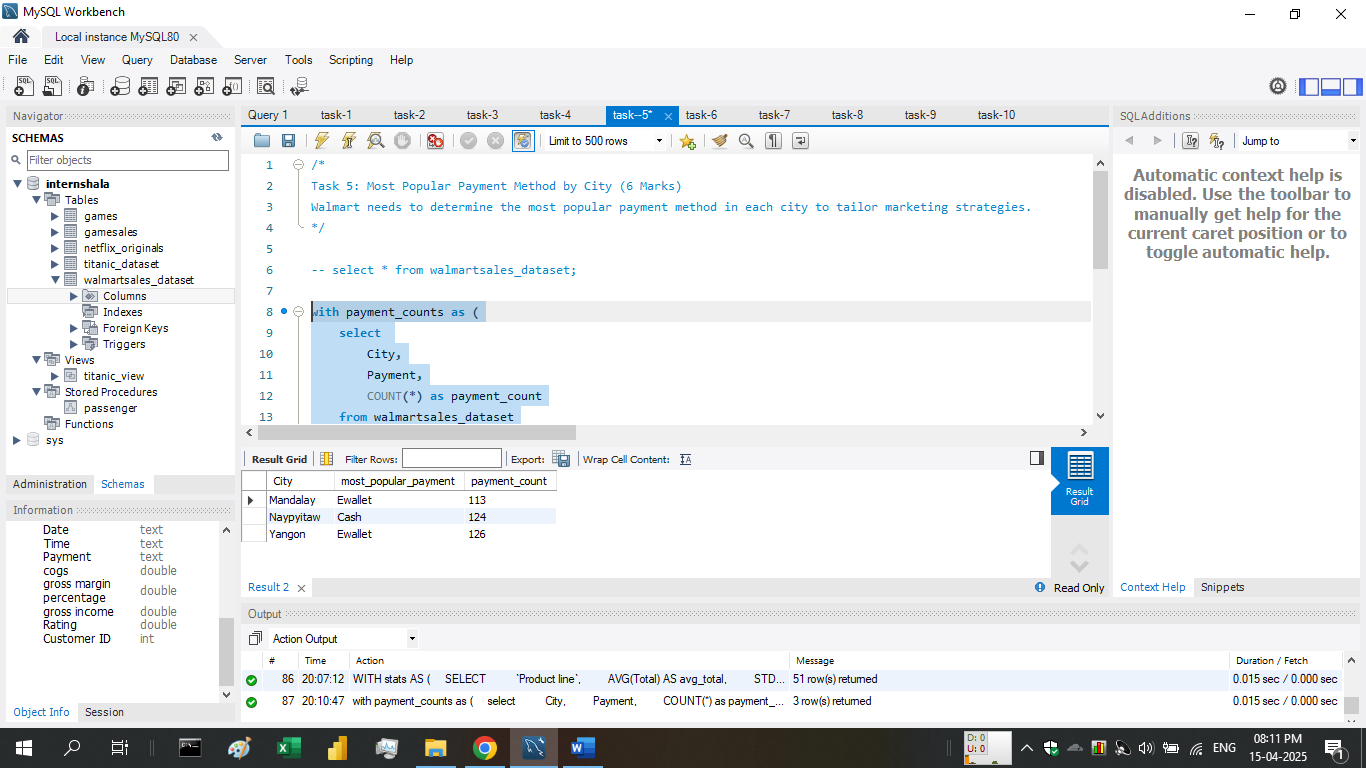
payment\_count

from ranked\_payments

where payment\_rank = 1

order by City;

query result



## 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

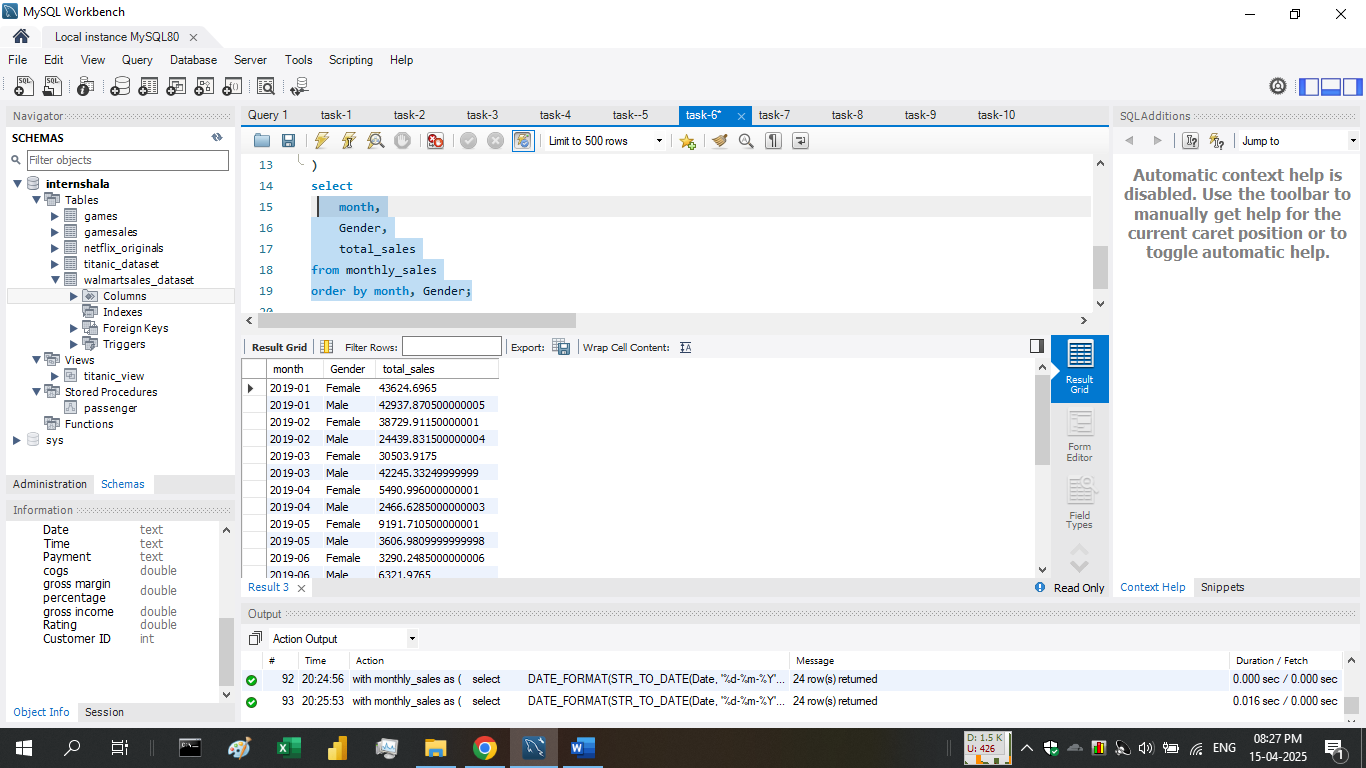
month,

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`,

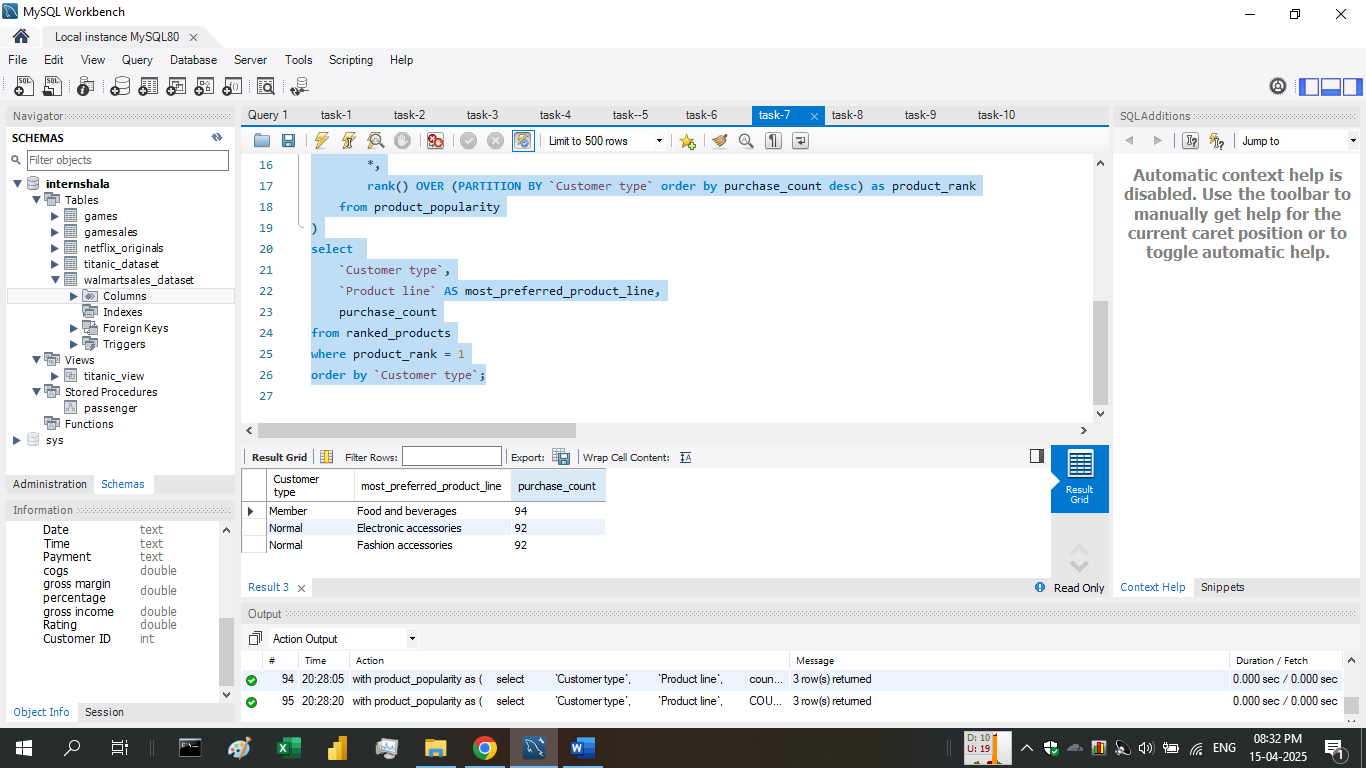
`Product line` AS most\_preferred\_product\_line,

purchase\_count

from ranked\_products

where product\_rank = 1

order by `Customer type`;



## 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`

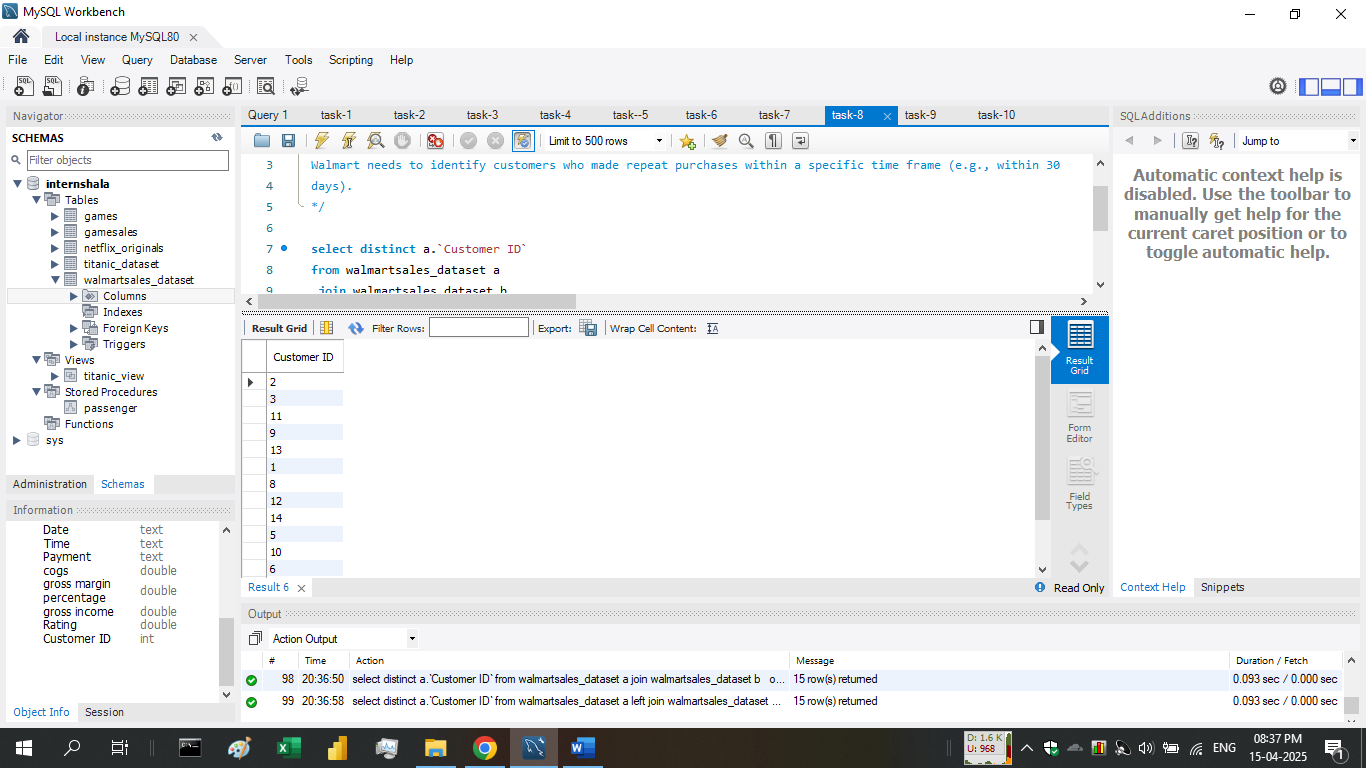
from walmartsales\_dataset a

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`,

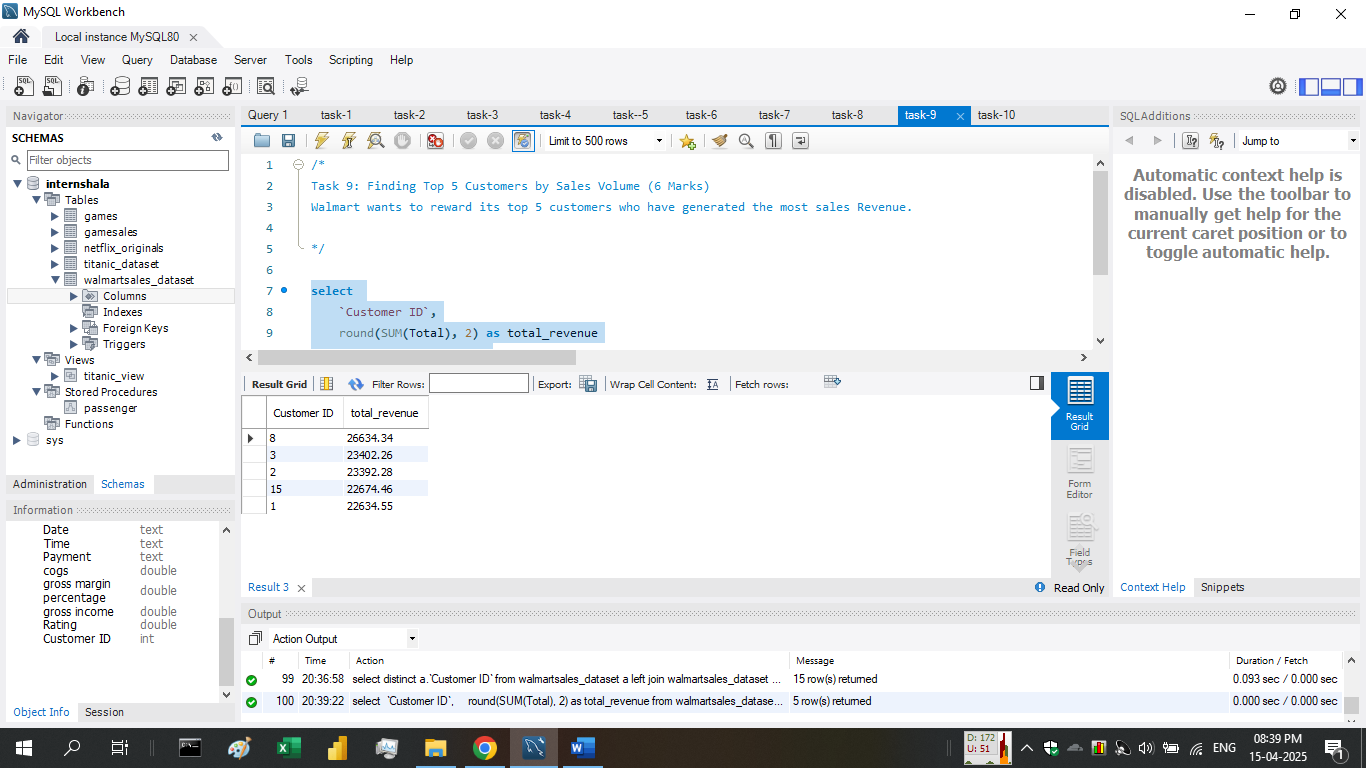
round(SUM(Total), 2) as total\_revenue

from walmartsales\_dataset

group by `Customer ID`

order by total\_revenue DESC

limit 5;



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

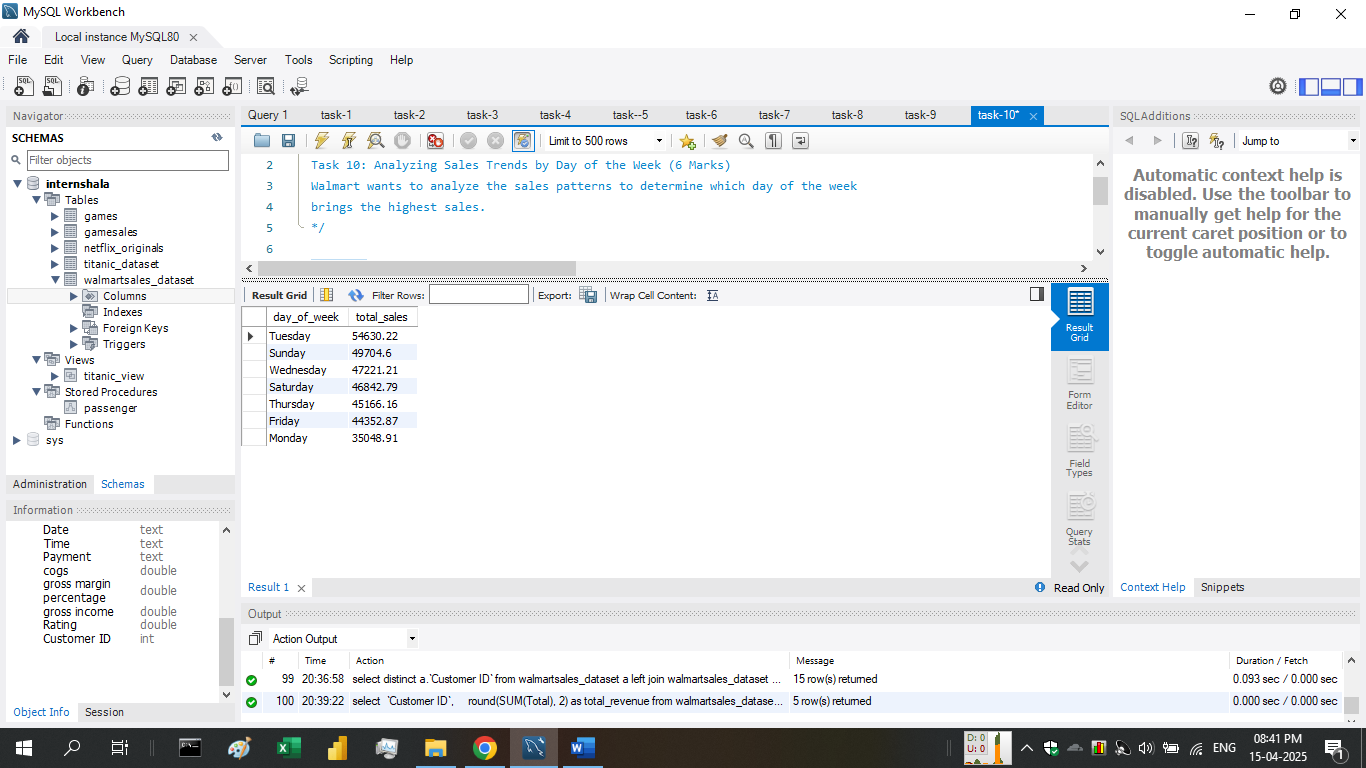
Walmart wants to analyze the sales patterns to determine which day of the week brings the highest sales.

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