**Course 4 - SQL**

**Final Project Submitted by – Ashish Kumar Jha**

**Task 1 - Identifying the Top Branch by Sales Growth Rate.**

WITH SalesGrowth AS (

SELECT

Branch,

DATE\_FORMAT(STR\_TO\_DATE(Date, '%d-%m-%Y'), '%Y-%m') AS Month,

SUM(Total) AS TotalSales,

LAG(SUM(Total)) OVER (PARTITION BY Branch ORDER BY DATE\_FORMAT(STR\_TO\_DATE(Date, '%d-%m-%Y'), '%Y-%m')) AS PreviousMonthSales

FROM

walmartsales

GROUP BY

Branch, DATE\_FORMAT(STR\_TO\_DATE(Date, '%d-%m-%Y'), '%Y-%m')

)

SELECT

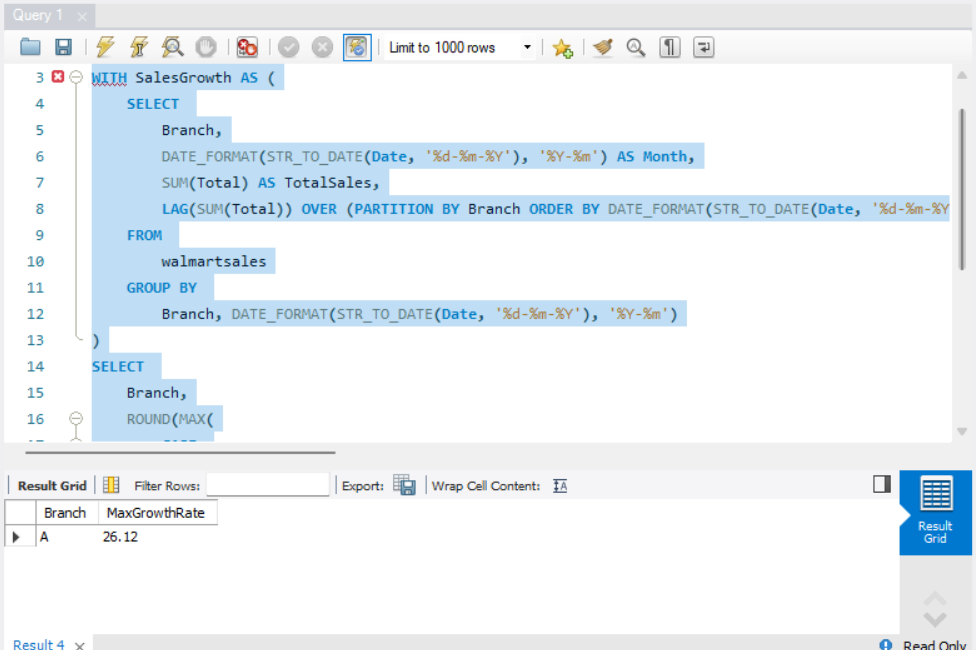
Branch,

ROUND(MAX(

CASE

WHEN PreviousMonthSales IS NOT NULL AND PreviousMonthSales > 0 THEN

(TotalSales - PreviousMonthSales) \* 100.0 / PreviousMonthSales

 ELSE 0

END

), 2) AS MaxGrowthRate

FROM

SalesGrowth

GROUP BY

Branch

ORDER BY

MaxGrowthRate DESC

LIMIT 1;

**Task 2 - Finding the Most Profitable Product Line for Each Branch.**

WITH RankedProductLines AS (

SELECT Branch, `Product line`, `gross income`,

RANK() OVER (PARTITION BY Branch ORDER BY `gross income` DESC) AS Ranked

FROM

WalmartSales

)

SELECT

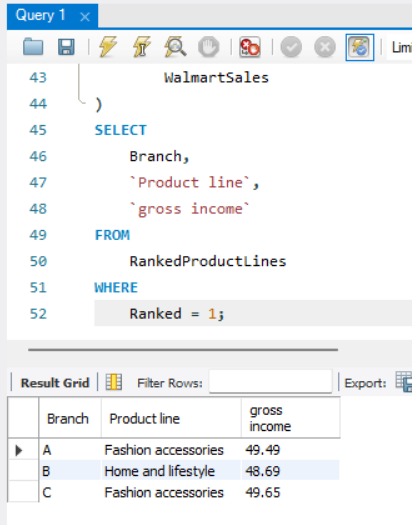
Branch, `Product line`, `gross income`

FROM

RankedProductLines

WHERE

Ranked = 1;



**Task 3 - Analyzing Customer Segmentation Based on Spending.**

WITH CustomerSpending AS (

SELECT `Customer ID`, `Customer type`, ROUND(SUM(Total),2) AS Total\_Spent

FROM WalmartSales

GROUP BY `Customer ID`, `Customer type`

),

CustomerClassification AS (

SELECT `Customer ID`, `Customer type`, Total\_Spent,

CASE

WHEN Total\_Spent >= (SELECT MAX(Total\_Spent) \* 0.66 FROM CustomerSpending) THEN 'High'

WHEN Total\_Spent >= (SELECT MAX(Total\_Spent) \* 0.33 FROM CustomerSpending) THEN 'Medium'

ELSE 'Low'

END AS Spending\_Tier

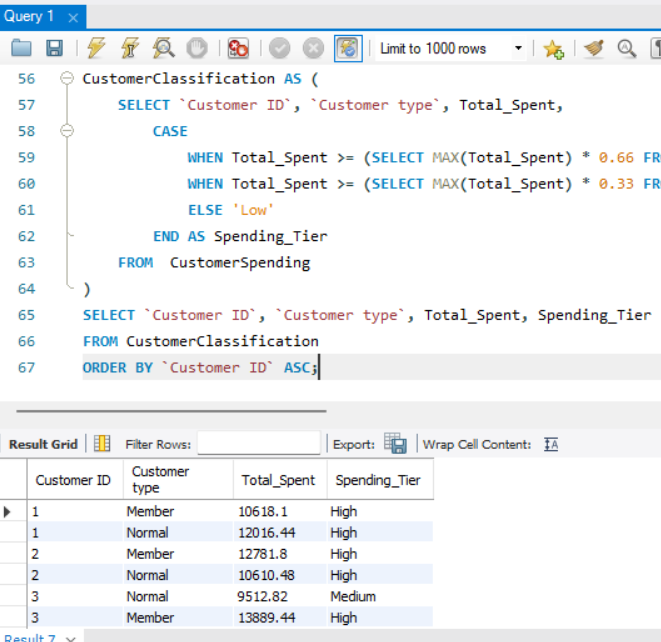
FROM CustomerSpending

)

SELECT `Customer ID`, `Customer type`, Total\_Spent, Spending\_Tier

FROM CustomerClassification

ORDER BY `Customer ID` ASC;



**Task 4 - Detecting Anomalies in Sales Transactions.**

SELECT AVG(Total) AS Average\_Total, STDDEV(Total) AS Standard\_Deviation

FROM WalmartSales;

SELECT `Invoice ID`, `Product line`, Total

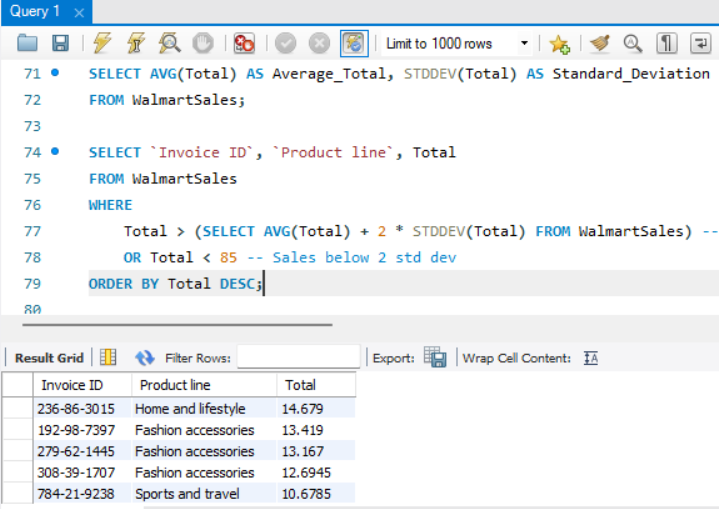
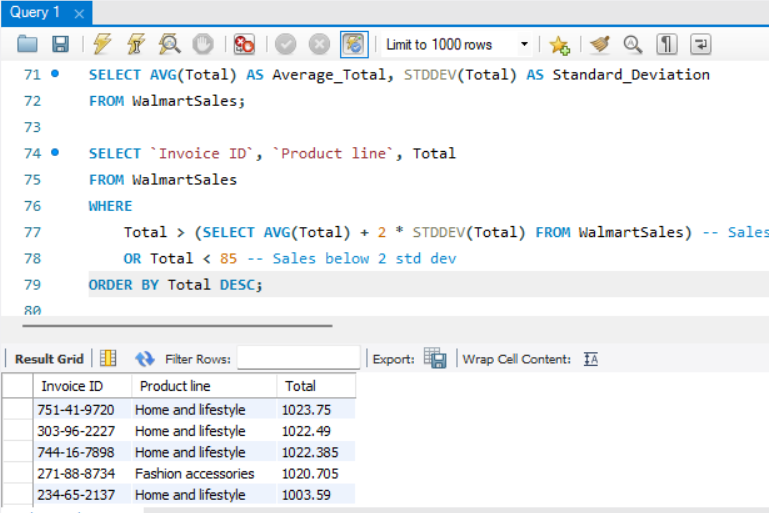
FROM WalmartSales

WHERE

Total > (SELECT AVG(Total) + 2 \* STDDEV(Total) FROM WalmartSales) -- Sales above 2 std dev

OR Total < 85 -- Sales below 2 std dev

ORDER BY Total DESC;

****

**Task 5 - Most Popular Payment Method by City.**

SELECT City, Payment, Payment\_Count

FROM (

SELECT City, Payment,

COUNT(\*) AS Payment\_Count,

MAX(COUNT(\*)) OVER (PARTITION BY City) AS Max\_Count

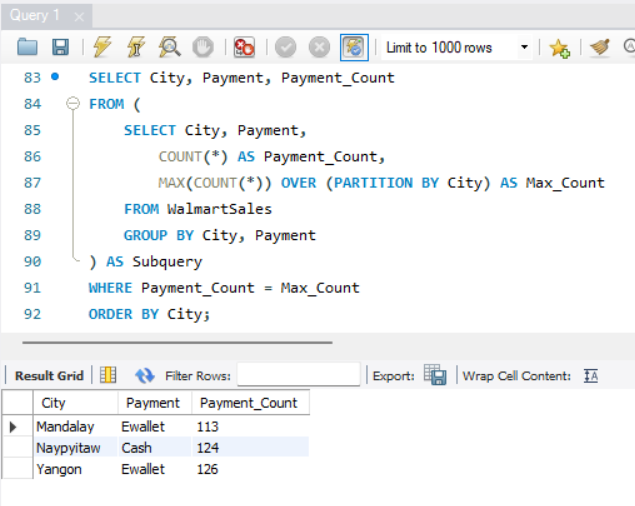
FROM WalmartSales

GROUP BY City, Payment

) AS Subquery

WHERE Payment\_Count = Max\_Count

ORDER BY City;

****

**Task 6 - Monthly Sales Distribution by Gender.**

SELECT

DATE\_FORMAT(STR\_TO\_DATE(Date, '%d-%m-%Y'), '%M') AS MONTHNAME,

Gender,

ROUND(SUM(Total),2) AS Total\_Sales

FROM WalmartSales

GROUP BY

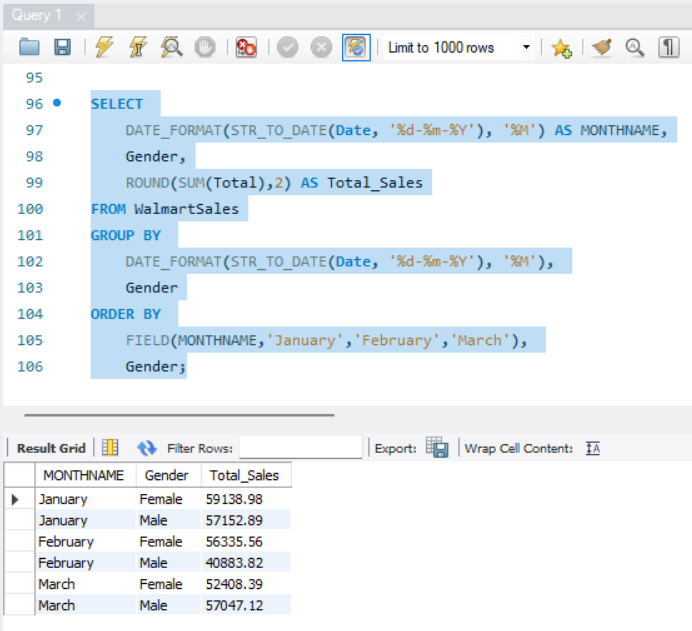
DATE\_FORMAT(STR\_TO\_DATE(Date, '%d-%m-%Y'), '%M'),

Gender

ORDER BY

FIELD(MONTHNAME,'January','February','March'),

Gender;

****

**Task 7 - Best Product Line by Customer Type.**

SELECT ws.`Customer type`,

ws.`Product line`,

ROUND(SUM(ws.Total),2) AS Total\_Sales

FROM WalmartSales ws

GROUP BY ws.`Customer type`, ws.`Product line`

HAVING

SUM(ws.Total) = (

SELECT MAX(Product\_Line\_Sales)

FROM (

SELECT SUM(Total) AS Product\_Line\_Sales

FROM WalmartSales

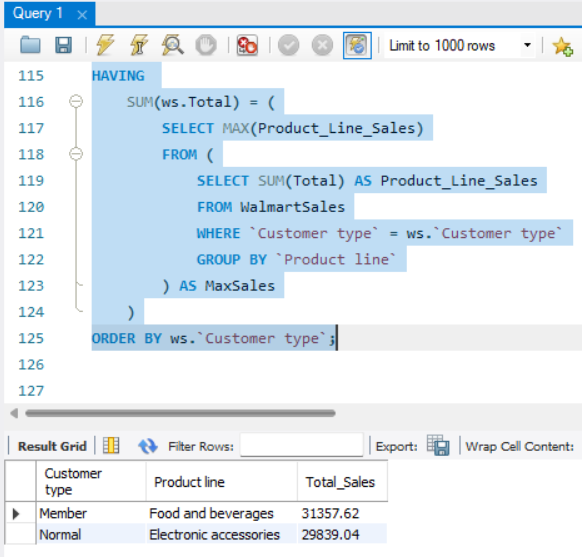
WHERE `Customer type` = ws.`Customer type`

GROUP BY `Product line`

) AS MaxSales

)

ORDER BY ws.`Customer type`;

****

**Task 8 - Identifying Repeat Customers.**

SELECT

w1.`Customer ID`,

MIN(STR\_TO\_DATE(w1.Date, '%d-%m-%Y')) AS First\_Purchase,

MAX(STR\_TO\_DATE(w2.Date, '%d-%m-%Y')) AS Last\_Purchase,

(COUNT(DISTINCT w2.`Invoice ID`) + 1) AS Total\_Repeat\_Buys

FROM walmartsales w1

JOIN

walmartsales w2 ON w1.`Customer ID` = w2.`Customer ID`

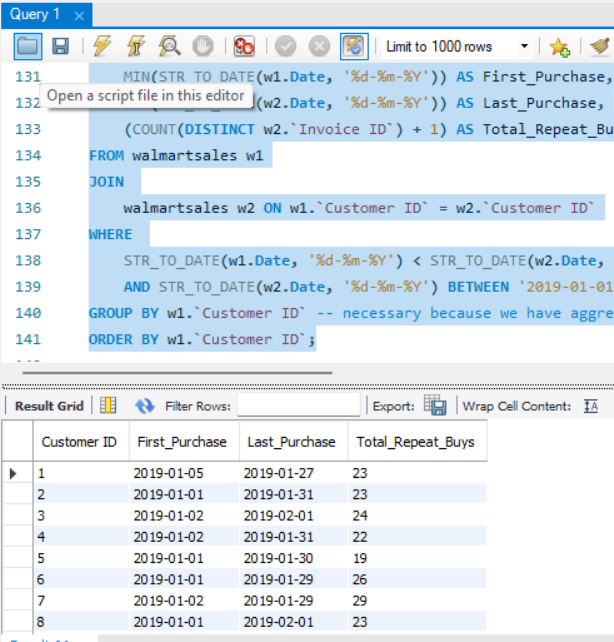
WHERE

STR\_TO\_DATE(w1.Date, '%d-%m-%Y') < STR\_TO\_DATE(w2.Date, '%d-%m-%Y')

AND STR\_TO\_DATE(w2.Date, '%d-%m-%Y') BETWEEN '2019-01-01' AND '2019-02-01'

GROUP BY w1.`Customer ID` -- necessary because we have aggregate function in query

ORDER BY w1.`Customer ID`;



**Task 9 - Finding Top 5 Customers by Sales Volume.**

SELECT

`Customer ID`,

ROUND(SUM(Total),2) AS Total\_Revenue

FROM

walmartsales

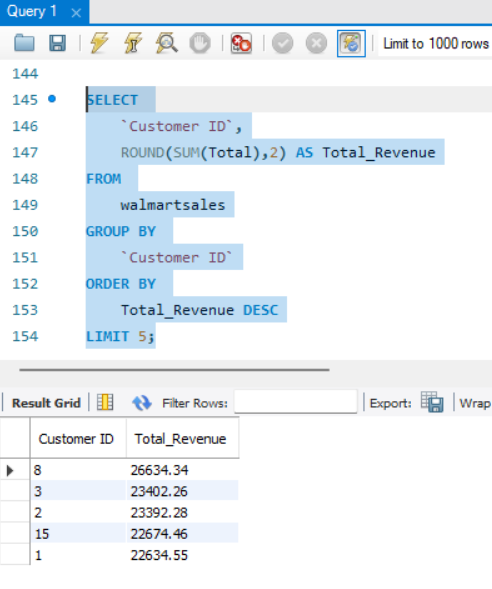
GROUP BY

`Customer ID`

ORDER BY

Total\_Revenue DESC

LIMIT 5;



**Task 10 - Analyzing Sales Trends by Day of the Week.**

SELECT

DAYNAME(STR\_TO\_DATE(Date, '%d-%m-%Y')) AS Day\_of\_Week,

ROUND(SUM(Total),2) AS Total\_Sales

FROM

walmartsales

GROUP BY

DAYNAME(STR\_TO\_DATE(Date, '%d-%m-%Y'))

ORDER BY

Total\_Sales DESC;

