**SuperMarketAnalysis**

**SQL Queries**

1. **To create and use a database:**

**SQL Query:**

CREATE DATABASE SalesAnalysis;

USE SalesAnalysis;

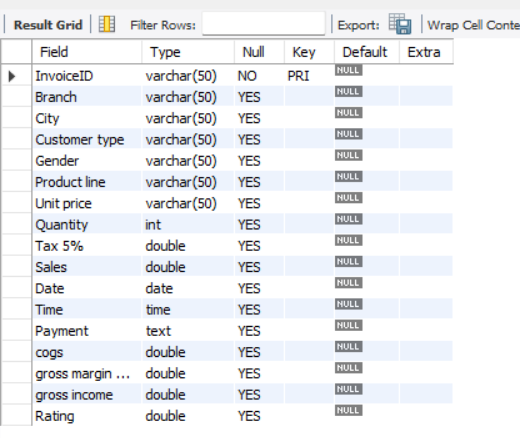
1. **Describe the schema:**

Describe command is used to display the structure of a table. It provides metadata about the table.

**SQL Query:**

DESCRIBE salesanalysis.supermarketanalysis;

**Output:**

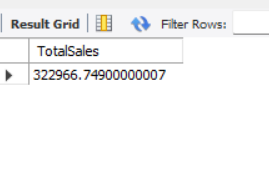


1. **Total Sales:**

**SQL Query:**

SELECT SUM(Sales) TotalSales FROM supermarketanalysis;

**Output:**



1. **Total Sales by City:**

**SQL Query:**

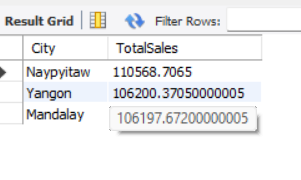
SELECT City, SUM(Sales) TotalSales

FROM supermarketanalysis

GROUP BY City

ORDER BY TotalSales DESC;

**Output:**



1. **Calculate total sales for each branch:**

**SQL Query:**

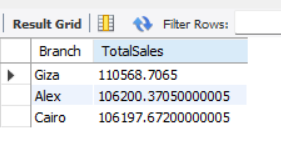
SELECT Branch, SUM(Sales) TotalSales

FROM supermarketanalysis

GROUP BY Branch

ORDER BY TotalSales DESC;

**Output:**



1. **Get the average customer rating for each product line.**

**SQL Query:**

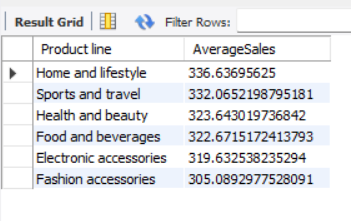
SELECT `Product line`, avg(sales) AverageSales

FROM supermarketanalysis

GROUP BY `Product line`

ORDER BY AverageSales DESC;

**Output:**



1. **Calculate the total sales and total quantity sold for each product line.**

**SQL Query:**

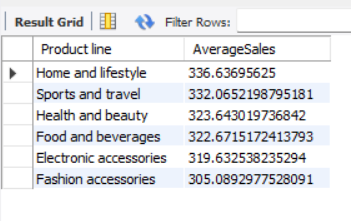
SELECT `Product line`, SUM(sales) TotalSales, SUM(quantity) TotalQuantity

FROM supermarketanalysis

GROUP BY `Product line`

ORDER BY TotalSales DESC;

**Output:**



1. **Show sales trend over month.**

**SQL Query:**

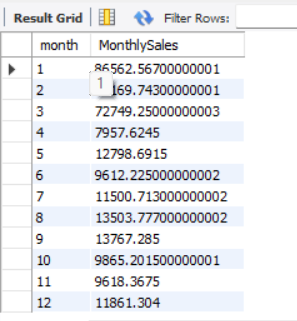
SELECT MONTH(Date) month, SUM(Sales) MonthlySales

FROM supermarketanalysis

GROUP BY MONTH

ORDER BY month;

**Output:**



1. **Calculate gross income and gross margin percentage for each payment type.**

**SQL Query:**

SELECT `Payment`,

SUM(`gross income`) AS totalgrossincome,

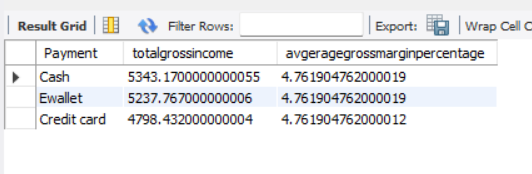
AVG(`gross margin percentage`) AS avgeragegrossmarginpercentage

FROM supermarketanalysis

GROUP BY `Payment`

ORDER BY totalgrossincome DESC;

**Output:**



1. **Analyze total sales by gender.**

**SQL Query:**

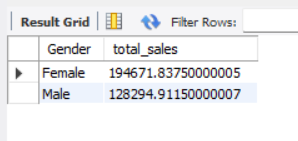
SELECT Gender, SUM(`Sales`) AS total\_sales

FROM supermarketanalysis

GROUP BY Gender

ORDER BY total\_sales DESC;

**Output:**



1. **Analyze sales by time of day.**

**SQL Query:**

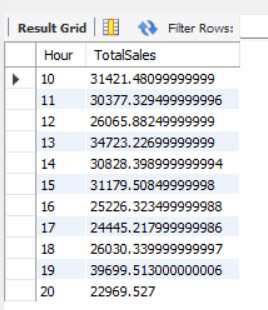
SELECT HOUR(time) Hour, SUM(sales) TotalSales

FROM supermarketanalysis

GROUP BY Hour

ORDER BY Hour;

**Output:**



1. **Analyze gross income based on customer type.**

**SQL Query:**

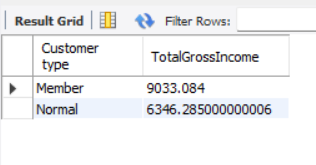
SELECT `Customer type`, SUM(`gross income`) TotalGrossIncome

FROM supermarketanalysis

GROUP BY `Customer type`

ORDER BY TotalGrossIncome DESC;

**Output:**



1. **The top 3 total sales of each product line within each customer type.**

**Sql Query:**

SELECT `product line`, `customer type`, sum(sales) TotalSales,

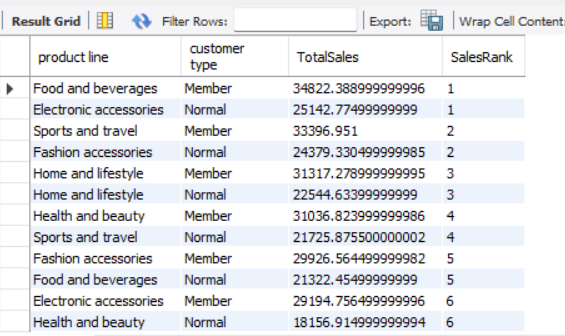
RANK() OVER(PARTITION BY `customer type` ORDER BY sum(sales) DESC) SalesRank

FROM supermarketanalysis

GROUP BY `product line`, `customer type`

ORDER BY SalesRank;

**Output:**



1. **Calculates monthly sales and the percentage change from the previous month.**

**Sql Query:**

WITH MonthlySales as (

SELECT DATE\_FORMAT(Date,'%Y-%m') Month, sum(sales) TotalSales

FROM supermarketanalysis

GROUP BY Month),

PreviousMonthSales as(

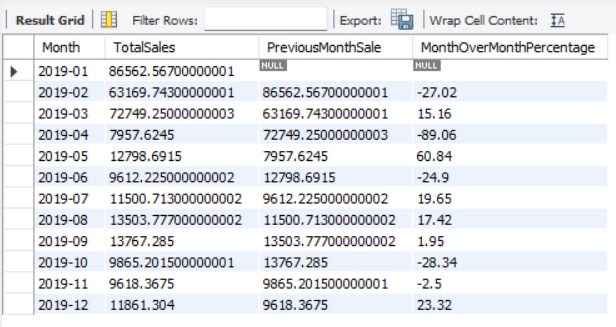
SELECT Month, TotalSales, LAG(TotalSales) OVER(ORDER BY Month ) PreviousMonthSale

FROM MonthlySales)

SELECT Month, TotalSales,PreviousMonthSale, ROUND(((TotalSales-PreviousMonthSale)/PreviousMonthSale)\*100,2) AS MonthOverMonthPercentage

FROM PreviousMonthSales;

**Output:**



1. **Find the top-selling product line in each city.**

**SQL Query:**

WITH CitySales AS(

SELECT City, `product line`, sum(sales) TotalSales

FROM supermarketanalysis

GROUP BY City, `product line`

ORDER BY City),

CitySalesRank AS(

SELECT City, `product line`, TotalSales, RANK() OVER(PARTITION BY City ORDER BY TotalSales) CityRank

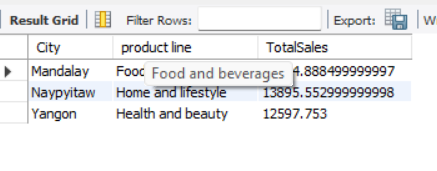
FROM CitySales

)

SELECT City, `product line`, TotalSales FROM CitySalesRank

WHERE CityRank=1;

**Output:**



1. **Analyze the contribution of each customer type to the total sales, both in absolute values and percentages.**

**SQL Query:**

WITH OverallSales as(

SELECT SUM(sales) TotalSales

FROM supermarketanalysis

),

CustomerContribution as(

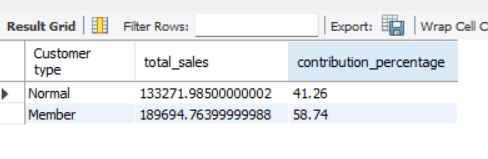
SELECT `Customer type`, SUM(`Sales`) AS total\_sales, ROUND((SUM(`Sales`)/(SELECT TotalSales FROM OverallSales))\*100,2) AS contribution\_percentage

FROM supermarketanalysis

GROUP BY `Customer type`)

SELECT \* FROM CustomerContribution;

**Output:**



1. **calculates the cumulative sales over time.**

**SQL Query:**

WITH DailySales as(

SELECT Date, SUM(Sales) AS TotalSale

FROM supermarketanalysis

GROUP BY Date

ORDER BY Date),

CumulativeSales as(

SELECT Date, SUM(Sales) OVER( ORDER BY Date) CumulativeSales

FROM supermarketanalysis)

SELECT ds.Date, ds.TotalSale, cs.CumulativeSales FROM CumulativeSales cs

INNER JOIN DailySales ds ON cs.Date=ds.Date;

**Output:**

