**2. Total Sales by City**

Calculate total sales per city:

sql

Copy code

SELECT

`City`,

SUM(`Sales`) AS total\_sales

FROM salesAnalysis

GROUP BY `City`

ORDER BY total\_sales DESC;

**3. Total Sales by Branch**

Calculate total sales for each branch:

sql

Copy code

SELECT

`Branch`,

SUM(`Sales`) AS total\_sales

FROM salesAnalysis

GROUP BY `Branch`

ORDER BY total\_sales DESC;

**4. Average Rating by Product Line**

Get the average customer rating for each product line:

sql

Copy code

SELECT

`Product line`,

AVG(`Rating`) AS average\_rating

FROM salesAnalysis

GROUP BY `Product line`

ORDER BY average\_rating DESC;

**5. Total Sales by Customer Type**

Calculate total sales based on customer type:

sql

Copy code

SELECT

`Customer type`,

SUM(`Sales`) AS total\_sales

FROM salesAnalysis

GROUP BY `Customer type`

ORDER BY total\_sales DESC;

**6. Sales and Quantity by Product Line**

Calculate the total sales and total quantity sold for each product line:

sql

Copy code

SELECT

`Product line`,

SUM(`Sales`) AS total\_sales,

SUM(`Quantity`) AS total\_quantity

FROM salesAnalysis

GROUP BY `Product line`

ORDER BY total\_sales DESC;

**7. Sales Trend Over Time (By Date)**

Show sales trend over time (by day):

sql

Copy code

SELECT

`Date`,

SUM(`Sales`) AS total\_sales

FROM salesAnalysis

GROUP BY `Date`

ORDER BY `Date`;

**8. Top 5 Highest Gross Income**

Retrieve the top 5 highest gross income sales:

sql

Copy code

SELECT

`Invoice ID`,

`Branch`,

`City`,

`Sales`,

`gross income`

FROM salesAnalysis

ORDER BY `gross income` DESC

LIMIT 5;

**9. Gross Income and Margin by Payment Type**

Calculate gross income and gross margin percentage for each payment type:

sql

Copy code

SELECT

`Payment`,

SUM(`gross income`) AS total\_gross\_income,

AVG(`gross margin percentage`) AS avg\_gross\_margin\_percentage

FROM salesAnalysis

GROUP BY `Payment`

ORDER BY total\_gross\_income DESC;

**10. Sales by Gender**

Analyze total sales by gender:

sql

Copy code

SELECT

`Gender`,

SUM(`Sales`) AS total\_sales

FROM salesAnalysis

GROUP BY `Gender`

ORDER BY total\_sales DESC;

**11. Sales and Tax Analysis**

Get total sales and tax (5%) analysis:

sql

Copy code

SELECT

SUM(`Sales`) AS total\_sales,

SUM(`Tax 5%`) AS total\_tax

FROM salesAnalysis;

**12. Sales by Time of Day**

Analyze sales by time of day (if Time is stored as a string in hh:mm:ss format):

sql

Copy code

SELECT

HOUR(`Time`) AS hour\_of\_day,

SUM(`Sales`) AS total\_sales

FROM salesAnalysis

GROUP BY HOUR(`Time`)

ORDER BY hour\_of\_day;

**13. Customer Type and Gross Income**

Analyze gross income based on customer type:

sql

Copy code

SELECT

`Customer type`,

SUM(`gross income`) AS total\_gross\_income

FROM salesAnalysis

GROUP BY `Customer type`

ORDER BY total\_gross\_income DESC;

**1. Sales Performance by Product Line and Customer Type**

This query ranks the total sales of each product line within each customer type, using a window function:

sql

Copy code

SELECT

`Product line`,

`Customer type`,

SUM(`Sales`) AS total\_sales,

RANK() OVER (PARTITION BY `Customer type` ORDER BY SUM(`Sales`) DESC) AS sales\_rank

FROM salesAnalysis

GROUP BY `Product line`, `Customer type`

ORDER BY `Customer type`, sales\_rank;

**2. Monthly Sales Trends**

This query summarizes monthly sales and calculates the percentage change from the previous month:

sql

Copy code

WITH MonthlySales AS (

SELECT

DATE\_FORMAT(`Date`, '%Y-%m') AS month,

SUM(`Sales`) AS total\_sales

FROM salesAnalysis

GROUP BY month

),

MonthlyChange AS (

SELECT

month,

total\_sales,

LAG(total\_sales) OVER (ORDER BY month) AS previous\_month\_sales

FROM MonthlySales

)

SELECT

month,

total\_sales,

previous\_month\_sales,

(total\_sales - previous\_month\_sales) / previous\_month\_sales \* 100 AS percentage\_change

FROM MonthlyChange;

**3. Top Selling Products in Each City**

Find the top selling product line in each city:

sql

Copy code

WITH CitySales AS (

SELECT

`City`,

`Product line`,

SUM(`Sales`) AS total\_sales

FROM salesAnalysis

GROUP BY `City`, `Product line`

),

RankedCitySales AS (

SELECT

`City`,

`Product line`,

total\_sales,

RANK() OVER (PARTITION BY `City` ORDER BY total\_sales DESC) AS sales\_rank

FROM CitySales

)

SELECT

`City`,

`Product line`,

total\_sales

FROM RankedCitySales

WHERE sales\_rank = 1; -- Top selling product

**4. Customer Type Contribution to Total Sales**

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

sql

Copy code

WITH TotalSales AS (

SELECT SUM(`Sales`) AS overall\_sales

FROM salesAnalysis

),

CustomerContribution AS (

SELECT

`Customer type`,

SUM(`Sales`) AS total\_sales,

(SUM(`Sales`) / (SELECT overall\_sales FROM TotalSales)) \* 100 AS contribution\_percentage

FROM salesAnalysis

GROUP BY `Customer type`

)

SELECT

`Customer type`,

total\_sales,

contribution\_percentage

FROM CustomerContribution;

**5. Average Sales per Branch with Standard Deviation**

This query calculates the average sales per branch along with the standard deviation of sales:

sql

Copy code

SELECT

`Branch`,

AVG(`Sales`) AS average\_sales,

STDDEV(`Sales`) AS sales\_stddev

FROM salesAnalysis

GROUP BY `Branch`

ORDER BY average\_sales DESC;

**6. Gross Margin Analysis by Product Line**

This query calculates the average gross margin percentage and gross income for each product line, showing how they relate to total sales:

sql

Copy code

SELECT

`Product line`,

AVG(`gross margin percentage`) AS avg\_gross\_margin,

SUM(`gross income`) AS total\_gross\_income,

SUM(`Sales`) AS total\_sales

FROM salesAnalysis

GROUP BY `Product line`

HAVING total\_sales > 0 -- Filter to include only product lines with sales

ORDER BY avg\_gross\_margin DESC;

**7. Cumulative Sales Over Time**

This query calculates the cumulative sales over time:

sql

Copy code

SELECT

`Date`,

SUM(`Sales`) AS daily\_sales,

SUM(SUM(`Sales`)) OVER (ORDER BY `Date` ASC) AS cumulative\_sales

FROM salesAnalysis

GROUP BY `Date`

ORDER BY `Date`;

**8. Sales Comparison: This Year vs Last Year**

Compare the total sales for the current year against the previous year:

sql

Copy code

WITH YearlySales AS (

SELECT

YEAR(`Date`) AS year,

SUM(`Sales`) AS total\_sales

FROM salesAnalysis

GROUP BY year

)

SELECT

current.year AS current\_year,

current.total\_sales AS current\_year\_sales,

previous.total\_sales AS previous\_year\_sales,

(current.total\_sales - previous.total\_sales) / previous.total\_sales \* 100 AS sales\_growth\_percentage

FROM YearlySales AS current

JOIN YearlySales AS previous ON current.year = previous.year + 1

WHERE current.year = YEAR(CURDATE());

**9. Segmented Sales Analysis by Gender and Customer Type**

This query analyzes total sales segmented by gender and customer type, showing how they differ:

sql

Copy code

SELECT

`Gender`,

`Customer type`,

SUM(`Sales`) AS total\_sales

FROM salesAnalysis

GROUP BY `Gender`, `Customer type`

ORDER BY `Gender`, total\_sales DESC;

**10. Sales Outliers Detection**

Identify outliers in sales using Z-score:

sql

Copy code

WITH SalesStats AS (

SELECT

AVG(`Sales`) AS avg\_sales,

STDDEV(`Sales`) AS stddev\_sales

FROM salesAnalysis

),

SalesOutliers AS (

SELECT

\*,

( `Sales` - (SELECT avg\_sales FROM SalesStats) ) / (SELECT stddev\_sales FROM SalesStats) AS z\_score

FROM salesAnalysis

)

SELECT

`Invoice ID`,

`Sales`,

z\_score

FROM SalesOutliers

WHERE ABS(z\_score) > 2; -- Outliers defined as those with a Z-score greater than 2

 salesAnalysis is the table name.

 Sales is derived as Unit price \* Quantity + Tax 5%.

 cogs refers to cost of goods sold, and gross income is derived from the difference between sales and COGS.

 Dates are in a standard date format (YYYY-MM-DD), and time is in hh:mm:ss format.