**SQL Queries**

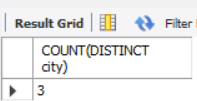
**Business Questions:**

1. **What is the count of distinct cities in the dataset?**

**SQL Querie:**

SELECT COUNT(DISTINCT city) FROM salesdata;

**Output:**

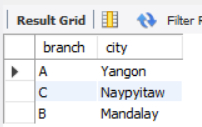


1. **For each branch, what is the corresponding city?**

**SQL Query:**

SELECT DISTINCT branch, city FROM salesdata;

**Output:**

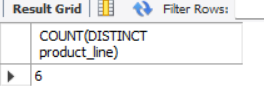


1. **What is the count of distinct product lines in the dataset?**

**SQL Query:**

SELECT DISTINCT product\_line FROM salesdata;

**Output:**



1. **Which payment method occurs most frequently?**

**SQL Query:**

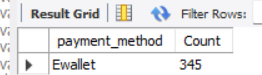
SELECT payment\_method, COUNT(\*) Count FROM salesdata

GROUP BY payment\_method

ORDER BY Count DESC

LIMIT 1;

**Output:**



1. **Which product line has the highest sales?**

**SQL Query:**

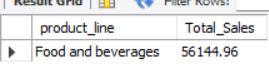
SELECT product\_line, SUM(total) Total\_Sales FROM salesdata

GROUP BY product\_line

ORDER BY Total\_Sales DESC

LIMIT 1;

**Output:**



1. **How much revenue is generated each month?**

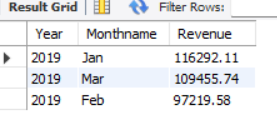
**SQL Query:**

SELECT YEAR(date) Year, Monthname, SUM(total) Revenue FROM salesdata

GROUP BY Year,Monthname

ORDER BY Revenue DESC;

**Output:**



1. **In which month did the cost of goods sold reach its peak?**

**SQL Query:**

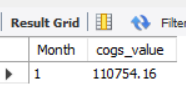
SELECT MONTH(date) Month, SUM(cogs) cogs\_value FROM salesdata

GROUP BY Month

ORDER BY cogs\_value DESC

LIMIT 1;

**Output:**



1. **Which product line generated the highest revenue?**

**SQL Query:**

SELECT product\_line, SUM(total) Revenue FROM salesdata

GROUP BY product\_line

ORDER BY Revenue DESC

LIMIT 1;

**Output:**



1. **In which city was the highest revenue recorded?**

**SQL Query:**

SELECT city, SUM(total) Revenue FROM salesdata

GROUP BY city

ORDER BY Revenue DESC

LIMIT 1;

**Ouput:**



1. **Which product line incurred the highest Value Added Tax?**

**SQL Query:**

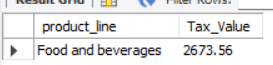
SELECT product\_line, ROUND(SUM(VAT),2) Tax\_Value FROM salesdata

GROUP BY product\_line

ORDER BY Tax\_Value DESC

LIMIT 1;

**Output:**

****

1. **For each product line, add a column indicating "Good" if its sales are above average, otherwise "Bad."**

**SQL Query:**

WITH Sales AS(

SELECT product\_line, SUM(total) Total\_sales

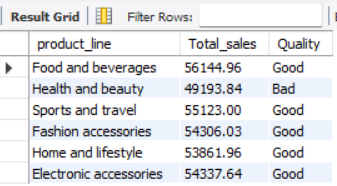
FROM salesdata

GROUP BY product\_line)

SELECT \*, CASE WHEN Total\_sales>(SELECT AVG(Total\_sales) FROM Sales) THEN "Good" ELSE "Bad" END AS Quality

FROM Sales;b

**Output:**



1. **Identify the branch that exceeded the average number of products sold.**

**SQL Query:**

WITH Branch\_sales\_count AS (

SELECT branch, count(\*) product\_count FROM salesdata

GROUP BY branch)

SELECT branch, product\_count FROM Branch\_sales\_count

WHERE product\_count> (SELECT AVG(product\_count) FROM Branch\_sales\_count);

**Output:**

****

1. **Which product line is most frequently associated with each gender?**

**SQL Query:**

WITH gender\_rank AS(

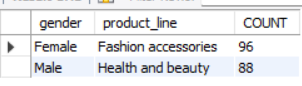
SELECT gender, product\_line, COUNT(\*) COUNT, RANK() OVER(PARTITION BY gender ORDER BY COUNT(\*) DESC) RANKP FROM salesdata

GROUP BY gender, product\_line)

SELECT gender, product\_line,COUNT FROM gender\_rank

WHERE RANKP=1;

**Output:**

****

1. **Calculate the average rating for each product line.**

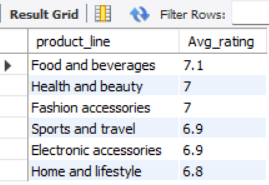
**SQL Query:**

SELECT product\_line, ROUND(AVG(rating),1) Avg\_rating FROM salesdata

GROUP BY product\_line

ORDER BY Avg\_rating DESC;

**Output:**



1. **Count the sales occurrences for each time of day on every weekday.**

**SQL Query:**

SELECT dayname, dayoftime, COUNT(\*) AS sales\_count

FROM salesdata

GROUP BY dayname, dayoftime

ORDER BY

CASE dayname

WHEN 'Mon' THEN 1

WHEN 'Tue' THEN 2

WHEN 'Wed' THEN 3

WHEN 'Thu' THEN 4

WHEN 'Fri' THEN 5

WHEN 'Sat’ THEN 6

WHEN 'Sun' THEN 7

END,

CASE dayoftime

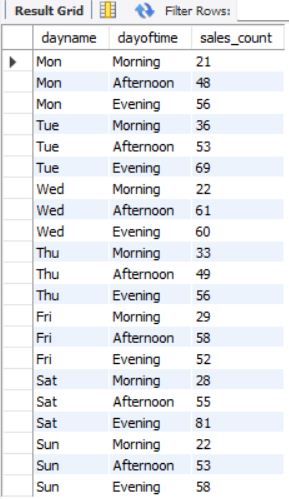
WHEN 'Morning' THEN 1

WHEN 'Afternoon' THEN 2

WHEN 'Evening' THEN 3

END;

**Output:**



1. **Identify the customer type contributing the highest revenue.**

**SQL Query:**

SELECT customer\_type, SUM(total) revenue

FROM salesdata

GROUP BY customer\_type

ORDER BY revenue DESC

LIMIT 1;

**Output:**



1. **Determine the city with the highest VAT percentage.**

**SQL Query:**

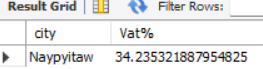
SELECT city, SUM(VAT)/(SELECT SUM(VAT) FROM salesdata)\*100 `Vat%` FROM salesdata

GROUP BY city

ORDER BY `Vat%` DESC

LIMIT 1;

**Output:**

****

1. **Identify the customer type with the highest VAT payments.**

**SQL Query:**

SELECT customer\_type, ROUND(SUM(VAT),2) VAT\_Payment

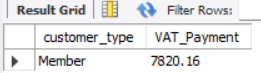
FROM salesdata

GROUP BY customer\_type

ORDER BY VAT\_Payment DESC

LIMIT 1;

**Output:**

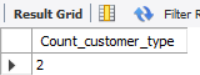


1. **What is the count of distinct customer types in the dataset?**

**SQL Query:**

SELECT COUNT(DISTINCT customer\_type) Count\_customer\_type FROM salesdata;

Output:

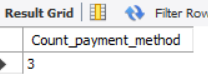


1. **What is the count of distinct payment methods in the dataset?**

**SQL Query:**

SELECT COUNT(DISTINCT payment\_method) Count\_payment\_method FROM salesdata;

**Output:**



1. **Which customer type occurs most frequently?**

**SQL Query:**

SELECT Customer\_type, COUNT(\*) AS Frequency

FROM salesdata

GROUP BY Customer\_type

ORDER BY Frequency DESC

LIMIT 1;

**Output:**



1. **Identify the customer type with the highest purchase frequency.**

**SQL Query:**

SELECT Customer\_type, SUM(quantity) purchase\_frequency

FROM salesdata

GROUP BY Customer\_type

ORDER BY purchase\_frequency DESC

LIMIT 1;

**Output:**

****

1. **Determine the predominant gender among customers.**

**SQL Query:**

SELECT gender, COUNT(\*) gender\_count FROM salesdata

GROUP BY gender

ORDER BY gender\_count DESC

LIMIT 1;

**Output:**



1. **Examine the distribution of genders within each branch.**

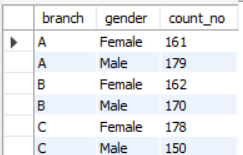
**SQL Query:**

SELECT branch, gender, COUNT(\*) count\_no FROM salesdata

GROUP BY branch, gender

ORDER BY branch ;

**Output:**



1. **Identify the time of day when customers provide the most ratings.**

**SQL Query:**

SELECT dayoftime, COUNT(rating) count\_rating FROM salesdata

GROUP BY dayoftime

ORDER BY count\_rating DESC

LIMIT 1;

Output:



1. **Determine the time of day with the highest customer ratings for each branch.**

**SQL Query:**

WITH Rating\_rank AS(

SELECT branch, dayoftime, MAX(rating) Highest\_Rating, RANK() OVER(PARTITION BY branch ORDER BY MAX(rating) DESC) rank\_branch FROM salesdata

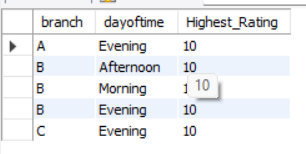
GROUP BY branch, dayoftime

ORDER BY branch, rank\_branch)

SELECT branch, dayoftime, Highest\_Rating FROM Rating\_rank

WHERE rank\_branch=1;

**Output:**

****

1. **Identify the day of the week with the highest average ratings.**

**SQL Query:**

SELECT dayname, ROUND(AVG(rating),2) Avg\_rating FROM salesdata

GROUP BY dayname

ORDER BY Avg\_rating DESC

LIMIT 1;

**Output:**



1. **Determine the day of the week with the highest average ratings for each branch.**

**SQL Query:**

WITH Avg\_ratings AS (

SELECT branch, dayname, ROUND(AVG(rating),2) Avg\_rating FROM salesdata

GROUP BY branch, dayname

ORDER BY branch),

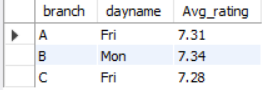
Rank\_Avg\_ratings AS(

SELECT \*, RANK() OVER(PARTITION BY branch ORDER BY Avg\_rating DESC) rank\_rating FROM Avg\_ratings)

SELECT branch, dayname, Avg\_rating FROM Rank\_Avg\_ratings

WHERE rank\_rating =1;

**Output:**

****