**Q1 : Who are the top and bottom performing sales employees in each country based on their total sales?**

The business Need :

This data can help in identifying high performers who may deserve recognition or additional incentives, as well as employees who may need further support or training , and identifying top and bottom performers provides benchmarks for setting realistic sales targets and goals for employees in each country.

And from another aspect that can be a strong base for analyzing sales performance by country allows businesses to allocate resources such as marketing budgets, sales support, and inventory strategically. High-performing countries may warrant increased investment to further capitalize on opportunities, while underperforming regions may require additional support or adjustments to improve results.

WITH Employee\_sales AS (

    SELECT e.emp\_id, e.first\_name, e.last\_name, e.country, o.Unitprice, o.quantity,

           ROUND(SUM(o.Unitprice \* o.quantity) OVER(PARTITION BY e.country, e.emp\_id)) AS emp\_sales

    FROM employees\_dim AS e

    INNER JOIN orders\_fact AS o ON e.emp\_sur = o.emp\_sur

) ,

Ranked\_employee\_sales\_tab AS (

    SELECT country, first\_name, last\_name, emp\_sales,

           ROW\_NUMBER() OVER (PARTITION BY country ORDER BY emp\_sales DESC) AS sales\_rank\_max,

           ROW\_NUMBER() OVER (PARTITION BY country ORDER BY emp\_sales ASC) AS sales\_rank\_min

    FROM Employee\_sales

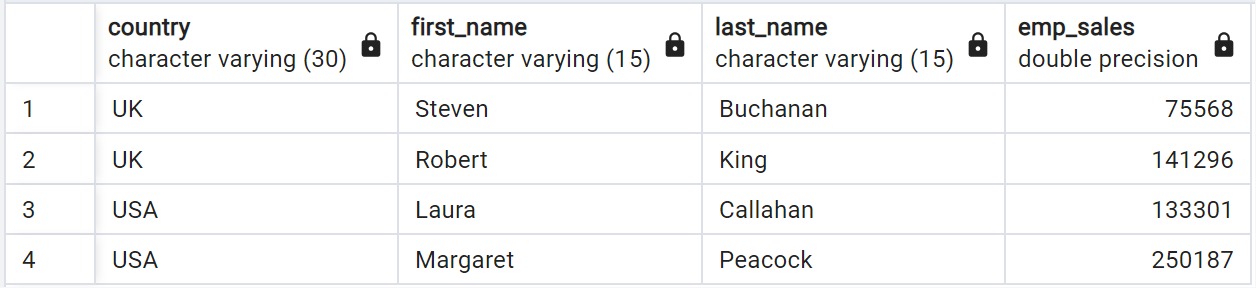
)

SELECT country, first\_name, last\_name, emp\_sales

FROM Ranked\_employee\_sales\_tab

WHERE sales\_rank\_max = 1 OR sales\_rank\_min = 1;

Output:



**Q2 : Who is the supplier whose products represent the highest and lowest sales, and what are their respective sales figures?**

The business Need :

Analyzing supplier sales performance is essential for optimizing sourcing strategies, managing costs, mitigating risks, and fostering mutually beneficial relationships with suppliers , through Knowing which suppliers generate the highest and lowest sales enables businesses to make informed strategic sourcing decisions. They can prioritize relationships with top-performing suppliers, potentially negotiating better terms, volume discounts, or other favorable arrangements. Conversely, businesses can reevaluate or renegotiate contracts with underperforming suppliers or explore alternative sourcing options to mitigate risks and ensure continuity of supply.

WITH supplier\_sales AS (

SELECT s.supplier\_id , s.company\_name ,ROUND(SUM(o.Unitprice \* o.quantity) OVER(PARTITION BY supplier\_id)) AS Sup\_sales

FROM suppliers\_dim AS s , orders\_fact AS o

WHERE s.supplier\_sur = o.supplier\_sur),

Ranked\_suppliers\_sales\_tab AS (

    SELECT supplier\_id, company\_name,  Sup\_sales,

           ROW\_NUMBER() OVER (ORDER BY Sup\_sales DESC) AS sales\_rank\_max,

           ROW\_NUMBER() OVER (ORDER BY Sup\_sales ASC) AS sales\_rank\_min

    FROM supplier\_sales

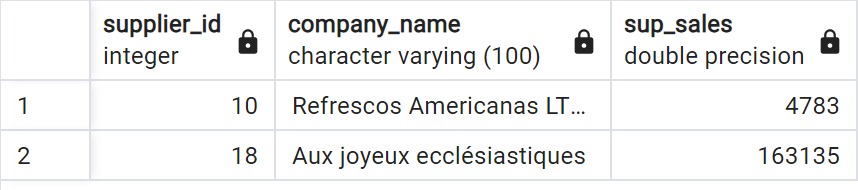
)

SELECT supplier\_id, company\_name,  Sup\_sales

FROM Ranked\_suppliers\_sales\_tab

WHERE sales\_rank\_max = 1 OR sales\_rank\_min = 1;

Output:



**Q3 : How can we categorize our customers based on their purchasing behavior ?**

The business Need :

The categories derived from this query, such as “At Risk”, “Customers Needing Attention”, “Promising”, “Potential Loyalists”, and “Loyal Customers”, reflect different stages of the customer lifecycle and provide a framework for understanding and addressing the needs of each customer group effectively.

The categorized groups can help prioritize resources, identify opportunities for upselling or cross-selling, and implement targeted marketing or service strategies.

WITH customer\_Purchases AS (

SELECT cust.customer\_id , cust.company\_name

       ,ROUND(SUM(o.Unitprice \* o.quantity) OVER(PARTITION BY customer\_id)) AS cust\_sales

FROM customer\_dim AS cust , orders\_fact AS o

WHERE cust.cust\_sur = o.cust\_sur),

cust\_category\_tab AS(

SELECT customer\_id , company\_name ,cust\_sales , NTILE(5) OVER(ORDER BY cust\_sales) AS customer\_categories

FROM customer\_Purchases )

SELECT DISTINCT customer\_id , company\_name ,cust\_sales ,customer\_categories ,

          CASE

                    WHEN customer\_categories = 1 THEN 'At Risk'

                       WHEN customer\_categories = 2 THEN 'Customers Needing Attention'

                       WHEN customer\_categories = 3 THEN 'promising'

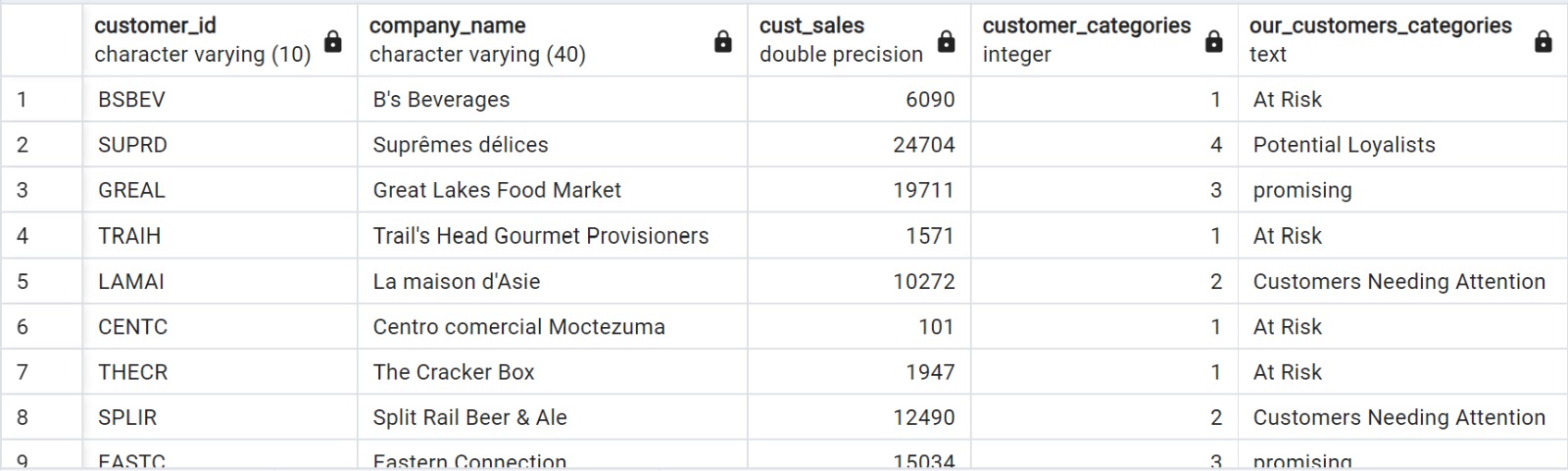
                       WHEN customer\_categories = 4 THEN 'Potential Loyalists'

                       WHEN customer\_categories = 5 THEN 'Loyal Customers'

                END AS Our\_Customers\_categories

FROM cust\_category\_tab ;

Output:



**Q4 : How does the monthly sales performance change over time, and what is the growth rate of sales from one month to the next?**

The business Need :

Monthly sales data allows businesses to identify seasonal variations in demand for their products or services. Recognizing these patterns enables companies to adjust their marketing strategies, inventory levels, and staffing accordingly to capitalize on peak periods and mitigate lulls in sales.

WITH MonthlySales AS (

SELECT TO\_TIMESTAMP(order\_date::text, 'YYYYMMDD') AS actual\_date,

       quantity,

       Unitprice

FROM orders\_fact

)

SELECT EXTRACT(YEAR FROM actual\_date) AS year,

 EXTRACT(MONTH FROM actual\_date) AS month,

  ROUND(SUM(Unitprice \* quantity)) AS total\_sales,

  ROUND(LAG(SUM(Unitprice \* quantity)) OVER (ORDER BY EXTRACT(YEAR FROM actual\_date),

                                                 EXTRACT(MONTH FROM actual\_date))) AS prev\_month\_sales,

  ROUND((SUM(Unitprice \* quantity) - LAG(SUM(Unitprice \* quantity)) OVER (ORDER BY EXTRACT(YEAR FROM actual\_date), EXTRACT(MONTH FROM actual\_date))) / LAG(SUM(Unitprice \* quantity)) OVER (ORDER BY EXTRACT(YEAR FROM actual\_date), EXTRACT(MONTH FROM actual\_date)) \* 100) AS growth\_rate,

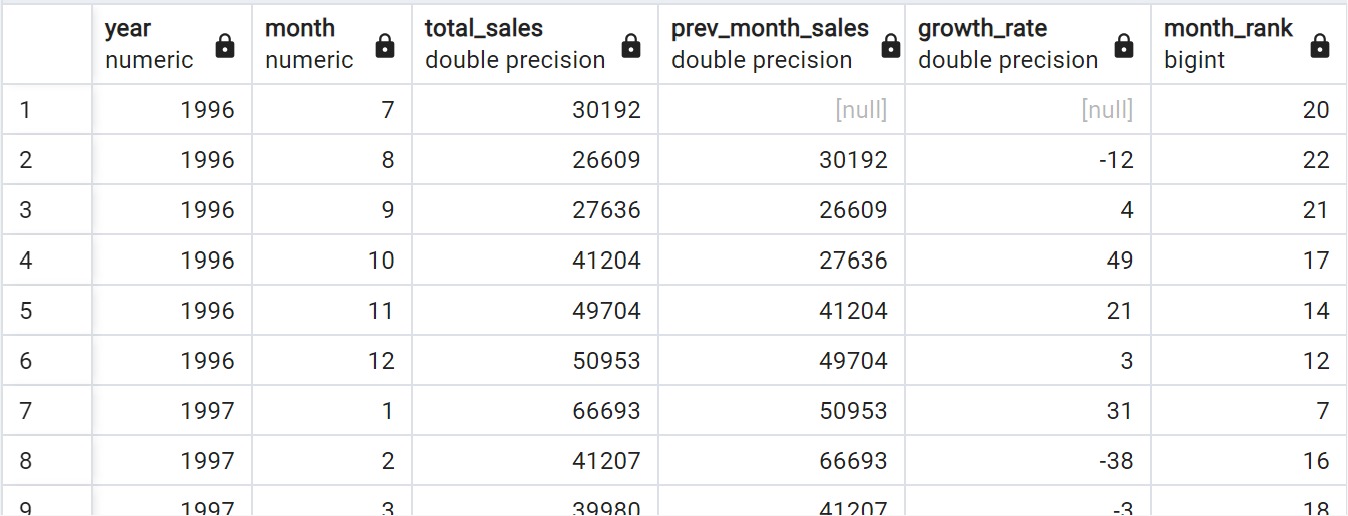
       RANK() OVER(ORDER BY SUM(QUANTITY \* Unitprice) DESC) AS MONTH\_RANK

FROM MonthlySales

GROUP BY EXTRACT(YEAR FROM actual\_date), EXTRACT(MONTH FROM actual\_date)

ORDER BY EXTRACT(YEAR FROM actual\_date), EXTRACT(MONTH FROM actual\_date);

Output:



**Q5 : How do the total sales for each month distribute across three 10-day periods within that month?**

The business Need :

Knowing when sales tend to peak or dip throughout the month enables businesses to manage their inventory levels more efficiently.

Analyzing sales distribution can inform marketing strategies and promotional campaigns. Businesses can tailor their marketing efforts to target specific periods within the month when sales are typically higher or lower. For instance, they may focus on running promotions or launching new product releases during periods of lower sales to stimulate demand.

WITH MonthSales AS (

SELECT

  TO\_CHAR(TO\_TIMESTAMP(order\_date::text, 'YYYYMMDD'), 'DD') AS actual\_date,

  TO\_CHAR(TO\_TIMESTAMP(order\_date::text, 'YYYYMMDD'), 'Mon YYYY') AS monthh,

  quantity,

  Unitprice

FROM orders\_fact

),

avgg AS (

SELECT

  monthh,

  SUM(CASE WHEN EXTRACT(DAY FROM TO\_TIMESTAMP(actual\_date, 'DD')) <= 10 THEN Unitprice \* Quantity ELSE 0 END) AS total\_sales\_of\_first\_10\_days,

  SUM(CASE WHEN EXTRACT(DAY FROM TO\_TIMESTAMP(actual\_date, 'DD')) BETWEEN 11 AND 20 THEN Unitprice \* Quantity ELSE 0 END) AS total\_sales\_of\_second\_10\_days,

  SUM(CASE WHEN EXTRACT(DAY FROM TO\_TIMESTAMP(actual\_date, 'DD')) BETWEEN 21 AND 31 THEN Unitprice \* Quantity ELSE 0 END) AS total\_sales\_of\_third\_10\_days

FROM MonthSales

GROUP BY monthh

)

SELECT

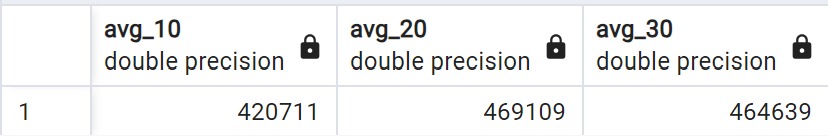
 ROUND(sum(total\_sales\_of\_first\_10\_days)) AS avg\_10,

 ROUND(sum(total\_sales\_of\_second\_10\_days)) AS avg\_20,

 ROUND(sum(total\_sales\_of\_third\_10\_days)) AS avg\_30

FROM avgg;

Output:



**Q6 : What are the top-selling products based on their category?**

The business Need :

By analyzing sales data by category, businesses can gain insights into which product categories are the most popular and generate the highest revenue. This information allows them to understand customer preferences and purchasing behavior within different product categories.

WITH product\_sales AS (

SELECT DISTINCT

  prod\_name,

  ROUND(SUM(OD.quantity \* p.Unitprice) OVER (PARTITION BY prod\_name)) AS stock\_sales

FROM products\_dim AS P

inner join public.orders\_fact AS OD

ON OD.product\_sur = p.prod\_sur

)

SELECT

 prod\_name,

 stock\_sales,

 RANK() OVER (ORDER BY stock\_sales DESC) AS product\_rnk

FROM product\_sales

ORDER BY product\_rnk;

Output:

