**Assignment Day 7**

1. Rank employees by their total sales

(Total sales = Total no of orders handled, JOIN employees and orders table)

SELECT

e.employee\_id,

e.first\_name || ' ' || e.last\_name AS employee\_name,

COUNT(o.order\_id) AS total\_sales,

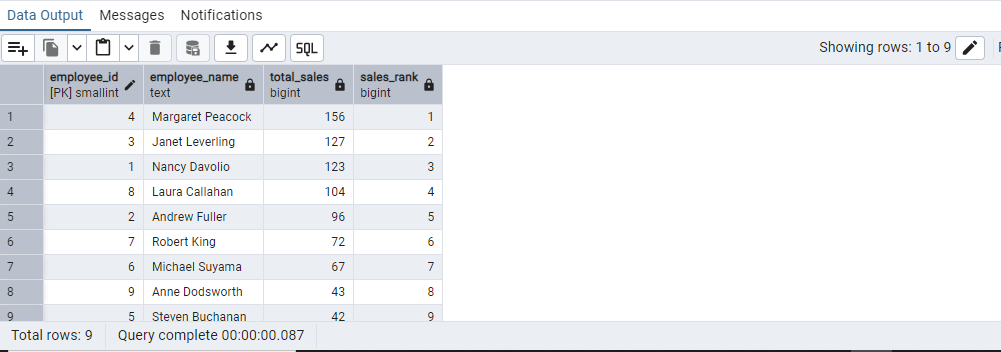
RANK() OVER (ORDER BY COUNT(o.order\_id) DESC) AS sales\_rank

FROM employees e

LEFT JOIN orders o ON e.employee\_id = o.employee\_id

GROUP BY e.employee\_id, e.first\_name, e.last\_name

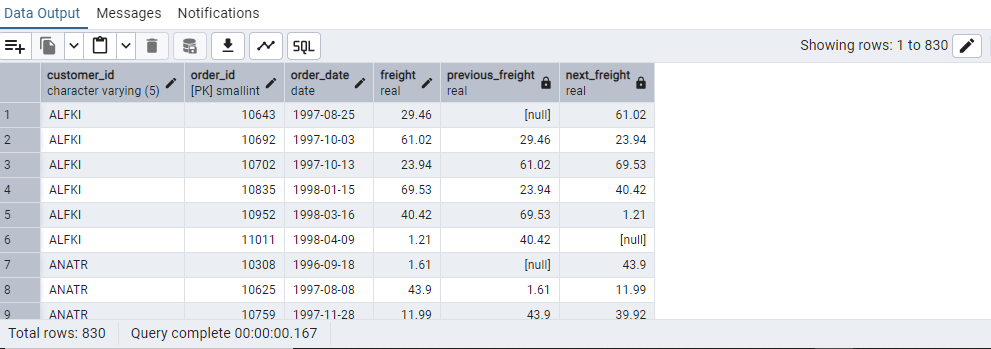
order by sales\_rank ASC



2. Compare current order's freight with previous and next order for each customer.

(Display order\_id, customer\_id, order\_date, freight,

Use lead(freight) and lag(freight).



3. Show products and their price categories, product count in each category, avg price:

(HINT:· Create a CTE which should have price\_category definition:

WHEN unit\_price < 20 THEN 'Low Price'

WHEN unit\_price < 50 THEN 'Medium Price'

ELSE 'High Price'

· In the main query display: price\_category, product\_count in each price\_category, ROUND(AVG(unit\_price)::numeric, 2) as avg\_price)

WITH price\_category AS (

SELECT

product\_id,

product\_name,

unit\_price,

CASE

WHEN unit\_price < 20 THEN 'Low Price'

WHEN unit\_price < 50 THEN 'Medium Price'

ELSE 'High Price'

END AS unit\_price\_categories

FROM products

)

SELECT

unit\_price\_categories,

COUNT(product\_id) AS product\_count,

ROUND(AVG(unit\_price)::numeric, 2) AS avg\_price

FROM price\_category

GROUP BY unit\_price\_categories

ORDER BY avg\_price;

