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

JOIN

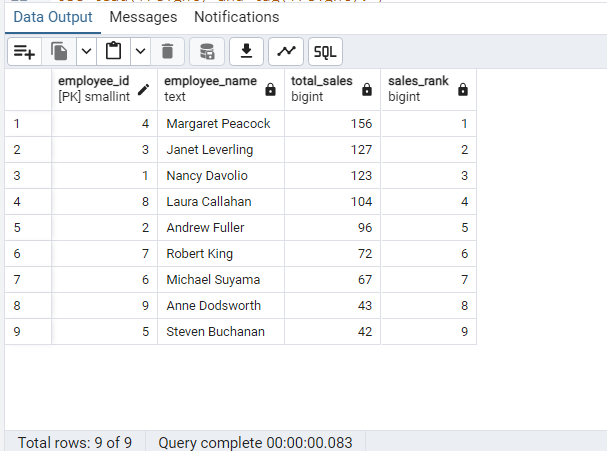
orders o ON e.employee\_id = o.employee\_id

GROUP BY

e.employee\_id

ORDER BY

sales\_rank;



/\*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).\*/

SELECT

order\_id,

customer\_id,

order\_date,

freight,

LAG(freight) OVER (PARTITION BY customer\_id ORDER BY order\_date) AS prev\_freight,

LEAD(freight) OVER (PARTITION BY customer\_id ORDER BY order\_date) AS next\_freight,

ROUND((freight - LAG(freight) OVER (PARTITION BY customer\_id ORDER BY order\_date))::numeric,2) AS compare\_from\_prev,

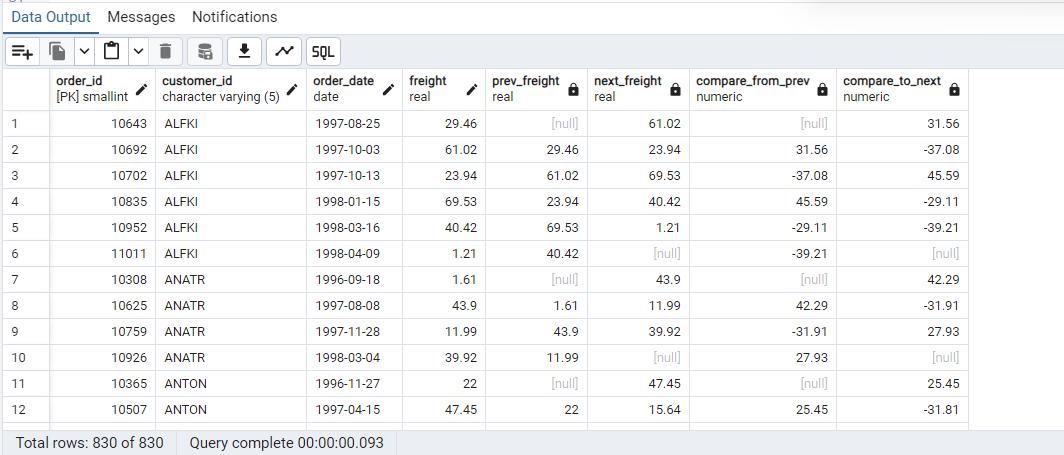
ROUND((LEAD(freight) OVER (PARTITION BY customer\_id ORDER BY order\_date) - freight)::numeric,2) AS compare\_to\_next

FROM

orders

ORDER BY

customer\_id, order\_date;



/\*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 PriceCategories AS (

SELECT

product\_id,

CASE

WHEN unit\_price < 20 THEN 'Low Price'

WHEN unit\_price < 50 THEN 'Medium Price'

ELSE 'High Price'

END AS price\_category, unit\_price

FROM products

)

SELECT

price\_category,

COUNT(product\_id) AS product\_count,

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

FROM PriceCategories

GROUP BY price\_category;

