**Day 7**

1. Rank employees by their total sales

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

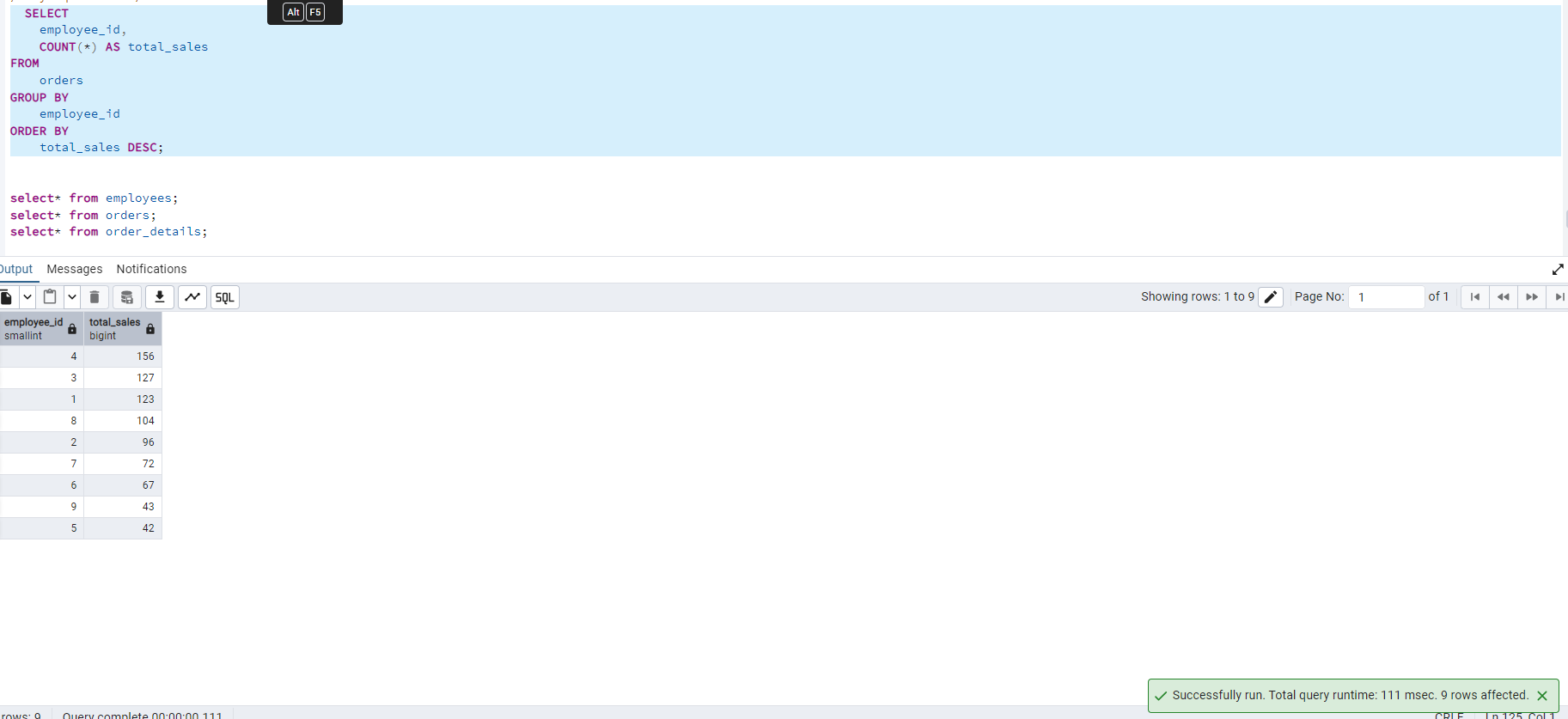
SELECT employee\_id, COUNT(\*) AS total\_sales

FROM orders

GROUP BY employee\_id

ORDER BY

total\_sales DESC;



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).

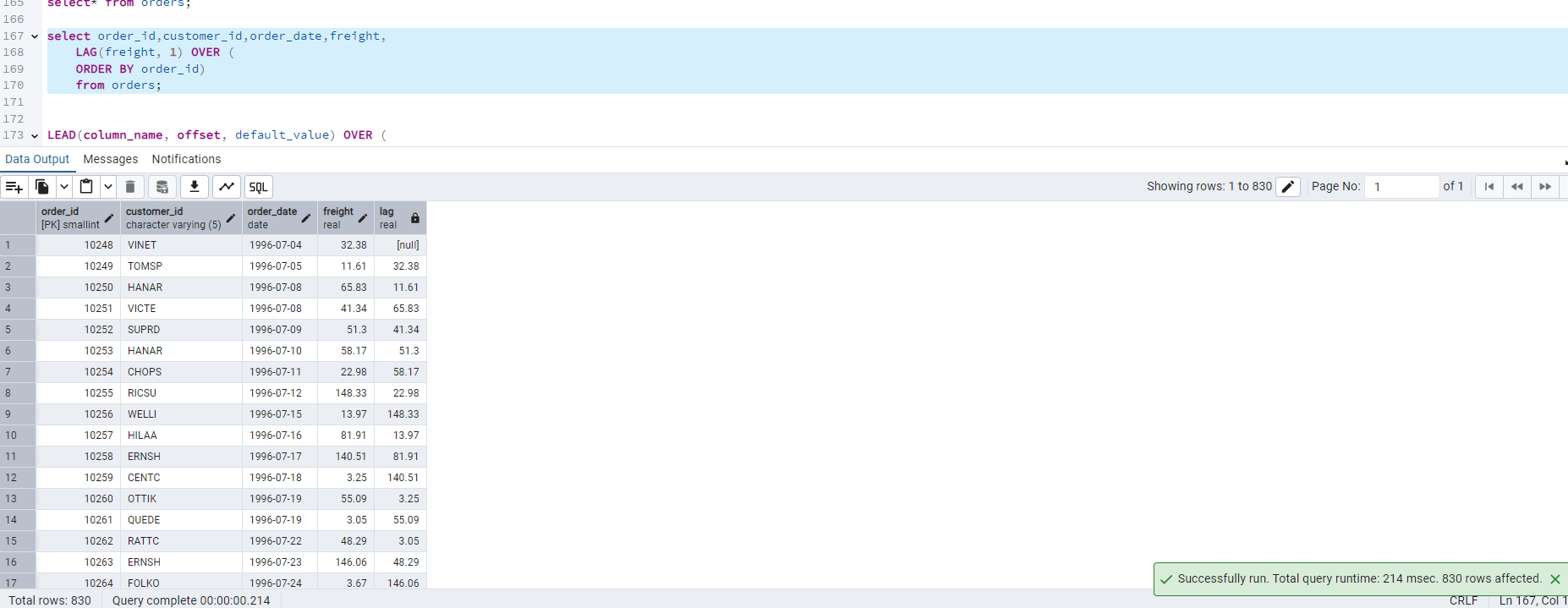
lag(freight)

select order\_id,customer\_id,order\_date,freight,

LAG(freight, 1) OVER (

ORDER BY order\_id)

from orders;



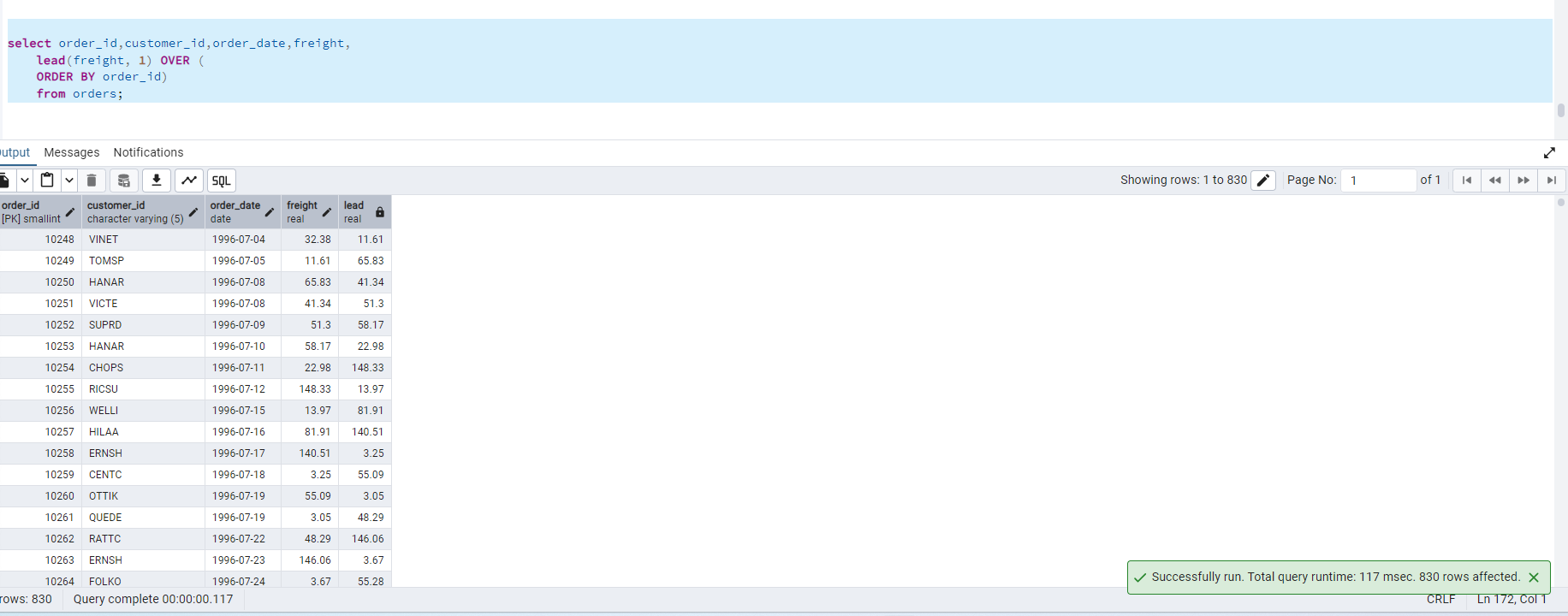
Use lead(freight)

select order\_id,customer\_id,order\_date,freight,

lead(freight, 1) OVER (

ORDER BY order\_id)

from orders;



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\_cte AS (

SELECT

product\_name,

unit\_price,

CASE

WHEN unit\_price < 20 THEN 'Low Price'

WHEN unit\_price < 50 THEN 'Medium Price'

ELSE 'High Price'

END AS price\_category

FROM products

)

SELECT

price\_category,

COUNT(\*) AS product\_count,

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

FROM price\_cte

GROUP BY price\_category;

