Day 7 Assignment

/\*1. Rank employees by their total sales

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

SELECT

E.EMPLOYEE\_ID,

CONCAT(E.FIRST\_NAME, ' ', E.LAST\_NAME) AS EMPLOYEE\_NAME,

COUNT(O.ORDER\_ID) as Total\_orders,

RANK() OVER (

ORDER BY COUNT(o.order\_id)

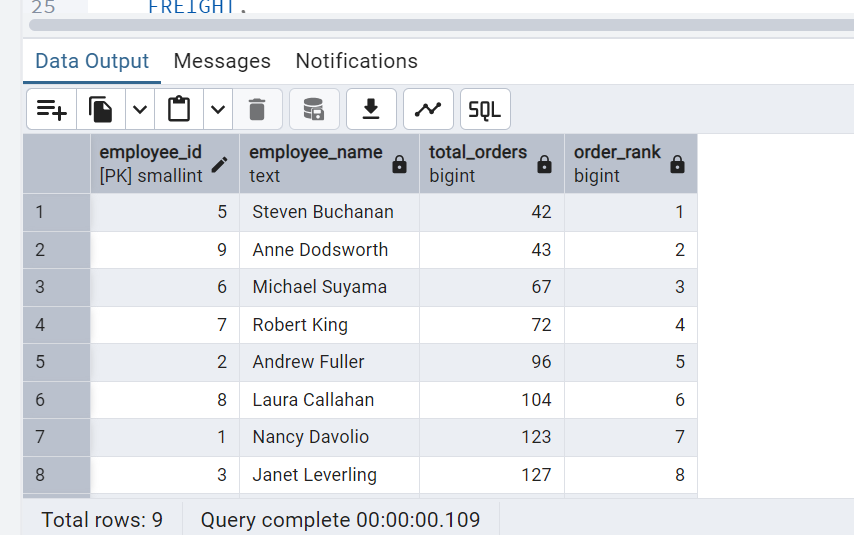
) AS order\_rank

FROM EMPLOYEES E

RIGHT JOIN ORDERS O ON E.EMPLOYEE\_ID = O.EMPLOYEE\_ID

GROUP BY E.EMPLOYEE\_ID, EMPLOYEE\_NAME

order by order\_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 (

ORDER BY ORDER\_ID

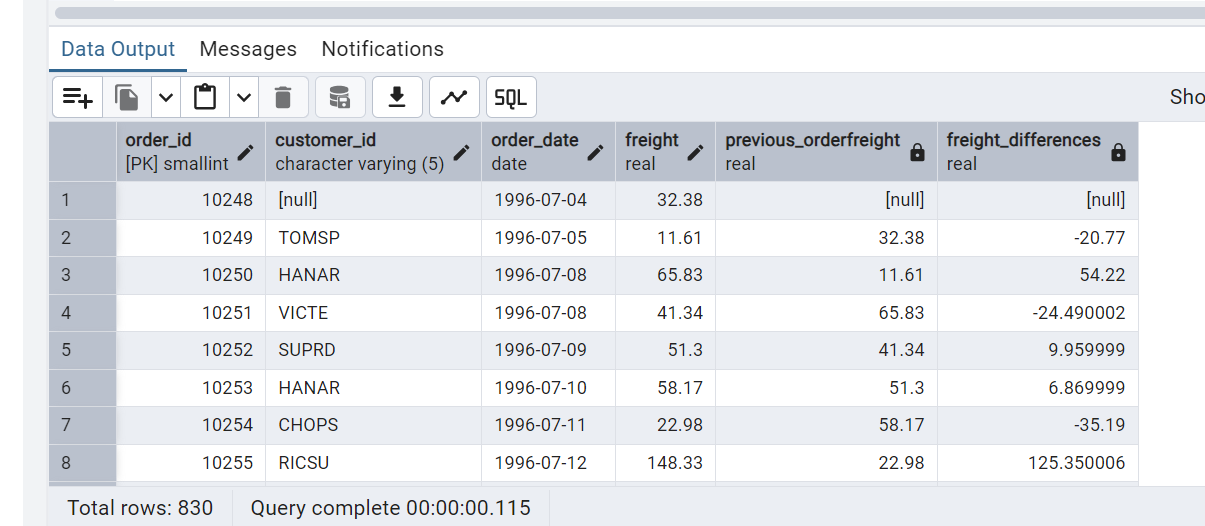
) AS PREVIOUS\_ORDERFREIGHT,

FREIGHT - LAG(FREIGHT) OVER (

ORDER BY ORDER\_ID

) AS FREIGHT\_DIFFERENCES

FROM ORDERS;



---- Lead ----

SELECT

ORDER\_ID,

CUSTOMER\_ID,

ORDER\_DATE,

FREIGHT,

Lead(FREIGHT) OVER (

ORDER BY ORDER\_ID

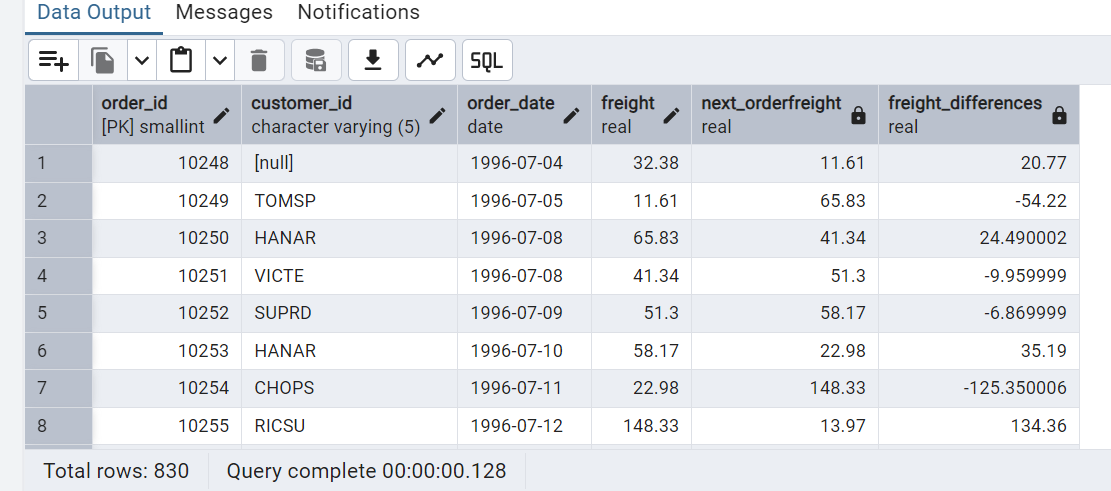
) AS Next\_orderFreight,

FREIGHT -LEAD(FREIGHT) OVER (

ORDER BY ORDER\_ID

) AS FREIGHT\_DIFFERENCES

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)\*/

SELECT

CASE

WHEN UNIT\_PRICE < 20 THEN 'Low Price'

WHEN UNIT\_PRICE < 50 THEN 'Medium Price'

ELSE 'HIGH price'

END AS PRICE\_CATEGORY,

COUNT(CATEGORY\_ID) AS PRODUCT\_COUNT,

ROUND(AVG(UNIT\_PRICE)::NUMERIC, 2) AS AVG\_PRICE

FROM PRODUCTS

GROUP BY PRICE\_CATEGORY;

