**1./\* Rank employees by their total sales**

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

select \* from orders;

Query:

select

employee\_id,

count(\*)as total\_sales,

rank()OVER

(order by count(\*)DESC) as sales\_rank

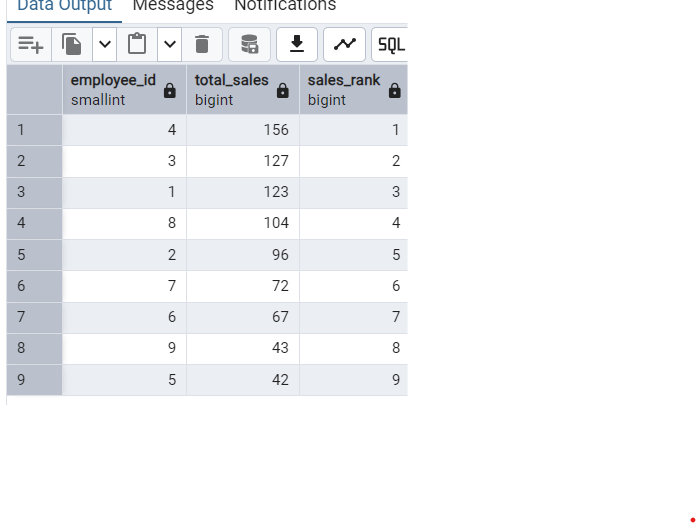
from

orders

group by

Employee\_id;

Output:



**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).\*/

Query:

select

order\_id,

customer\_id,

order\_date,

freight,

lag(freight)over (partition by customer\_id order by freight)as previous\_order\_freight,

freight-lag(freight)over(partition by customer\_id order by freight)as next\_order\_freight

from orders;

select

order\_id,

customer\_id,

order\_date,

freight,

lead(freight)over (partition by customer\_id order by freight)as previous\_order\_freight,

freight-lead(freight)over(partition by customer\_id order by freight)as next\_order\_freight

from orders;

select

order\_id,

customer\_id,

order\_date,

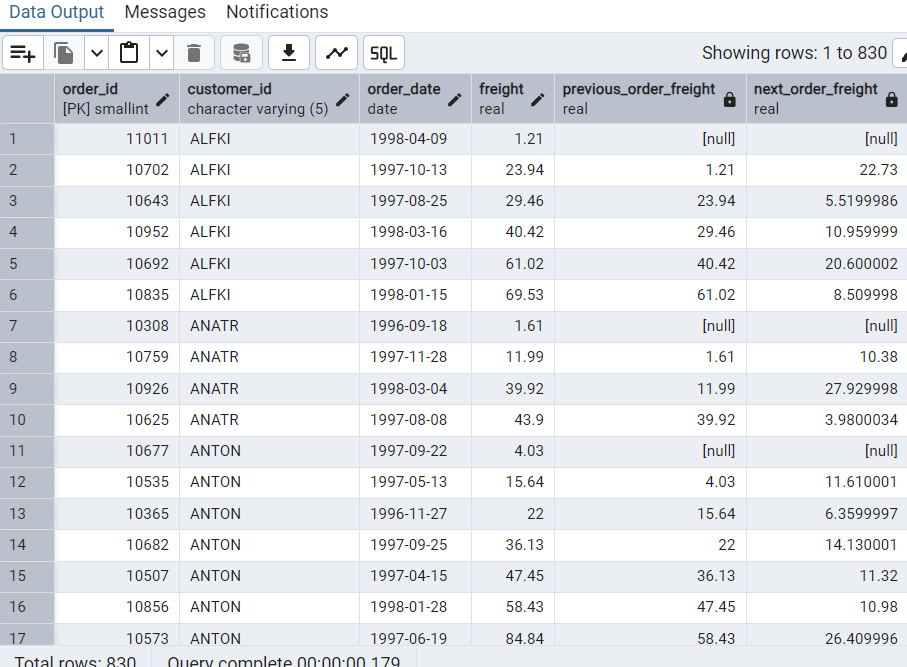
freight,

lead(freight)over (partition by customer\_id order by freight)as previous\_order\_freight,

freight-lead(freight)over(partition by customer\_id order by freight)as next\_order\_freight

from orders;

Output:



**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)\*/

QUERY:

with

price\_category as(

select

category\_id,

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\_category

group by price\_category;

OUTPUT:

