select \* from employees;

select \* from orders;

select e.employeeID,

count(\*)as total\_sale,

rank() over (order by count (orderID) desc) as sales\_rank

from orders o

join employees e on O.employeeID= e.employeeID

group by e.employeeID

order by 2 desc;

A screenshot of a computer

AI-generated content may be incorrect.

select orderID,customerID,orderDate,

lag(freight) over(partition by customerID order by orderID)as previous\_freight,

lead(freight)over (partition by customerID order by orderID)as next\_freight

from orders;

A screenshot of a computer

AI-generated content may be incorrect.

with cte\_price\_category as

(select unitPrice,

case

WHEN unitPrice< 20 THEN 'Low Price'

WHEN unitPrice < 50 THEN 'Medium Price'

ELSE 'High Price'

end

from products)

select cte\_price\_category,

count (\*) as product\_count,

ROUND(AVG(unitPrice)::numeric, 2) as avg\_price

from cte\_price\_category

group by cte\_price\_category

order by cte\_price\_category;

A screenshot of a computer

AI-generated content may be incorrect.