

1-----

```
FROM orders O
JOIN employees E on O.employee_id = E.employee_id
GROUP BY E.employee_id
ORDER BY 2 DESC;
```

Data Output	Messages	Notifications
<div> <div>SQL</div> </div>		
employee_id [PK] smallint	total_sales bigint	sales_rank bigint
4	156	1
3	127	2
1	123	3
8	104	4
2	96	5
7	72	6
6	67	7
9	43	8
5	42	9

2-----

```
SELECT order_id, customer_id, order_date, freight,
LAG(freight) OVER (partition by customer_id order by freight) AS previous_order_freight,
LEAD(freight) OVER (partition by customer_id order by freight) AS next_order_freight
FROM orders;
```

Data Output

Messages

Notifications

+

📄

▼

📄

▼

📄

📄

📄

📄

SQL

	order_id [PK] smallint	customer_id character varying (5)	order_date date	freight real	previous_order_freight real	next_order_freight real
1	11011	ALFKI	1998-04-09	1.21	[null]	23.94
2	10702	ALFKI	1997-10-13	23.94	1.21	29.46
3	10643	ALFKI	1997-08-25	29.46	23.94	40.42
4	10952	ALFKI	1998-03-16	40.42	29.46	61.02
5	10592	ALFKI	1997-10-03	61.02	40.42	69.53
6	10835	ALFKI	1998-01-15	69.53	61.02	[null]
7	10308	ANATR	1996-09-18	1.61	[null]	11.99
8	10759	ANATR	1997-11-28	11.99	1.61	39.92
9	10926	ANATR	1998-03-04	39.92	11.99	43.9
10	10625	ANATR	1997-08-08	43.9	39.92	[null]
11	10577	ANTON	1997-09-22	4.03	[null]	15.64
12	10705	ANTON	1997-05-10	15.64	4.03	89

Total rows: 830

Query complete 00:00:00.126

Total rows: 830 Query complete 00:00:00.126

3-----

```
42 --main query
43 SELECT price_category,
44 COUNT (*) AS product_count,
45 ROUND(AVG(unit_price)::numeric, 2) as avg_price
46 FROM cte_price_category
47 GROUP BY price_category
48 ORDER BY price category;
```

Data Output Messages Notifications

	price_category text	product_count bigint	avg_price numeric
1	High Price	7	105.11
2	Low Price	39	12.95
3	Medium Price	31	31.59