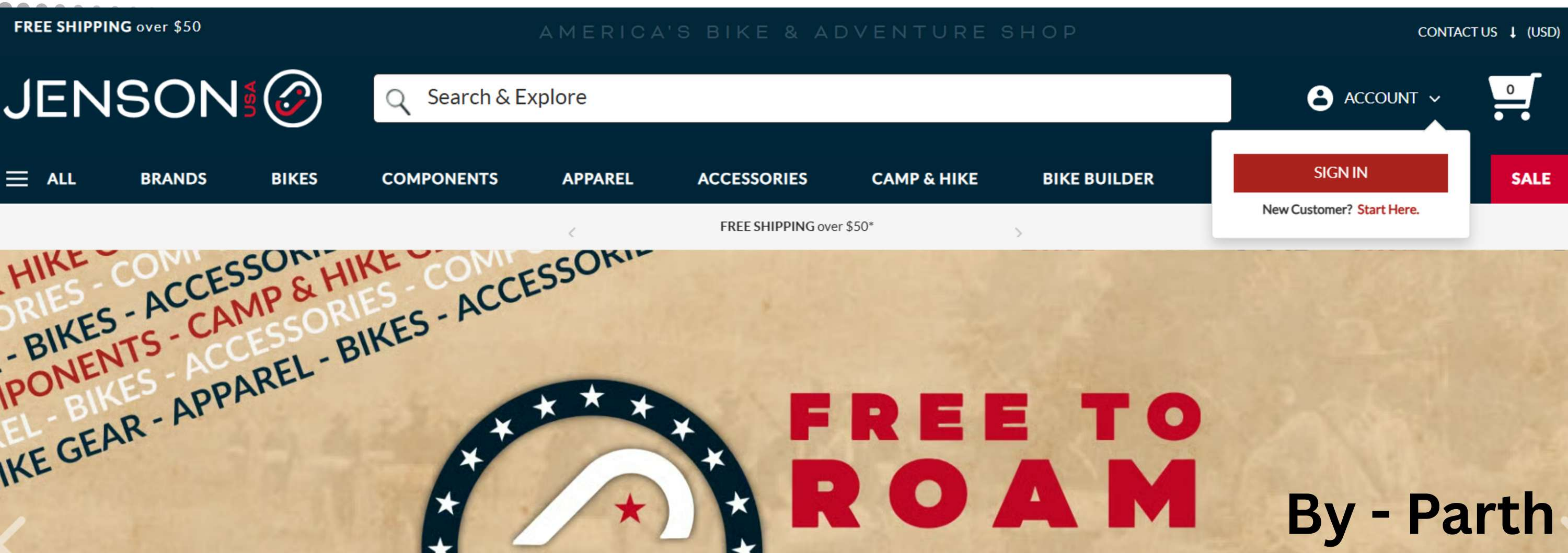
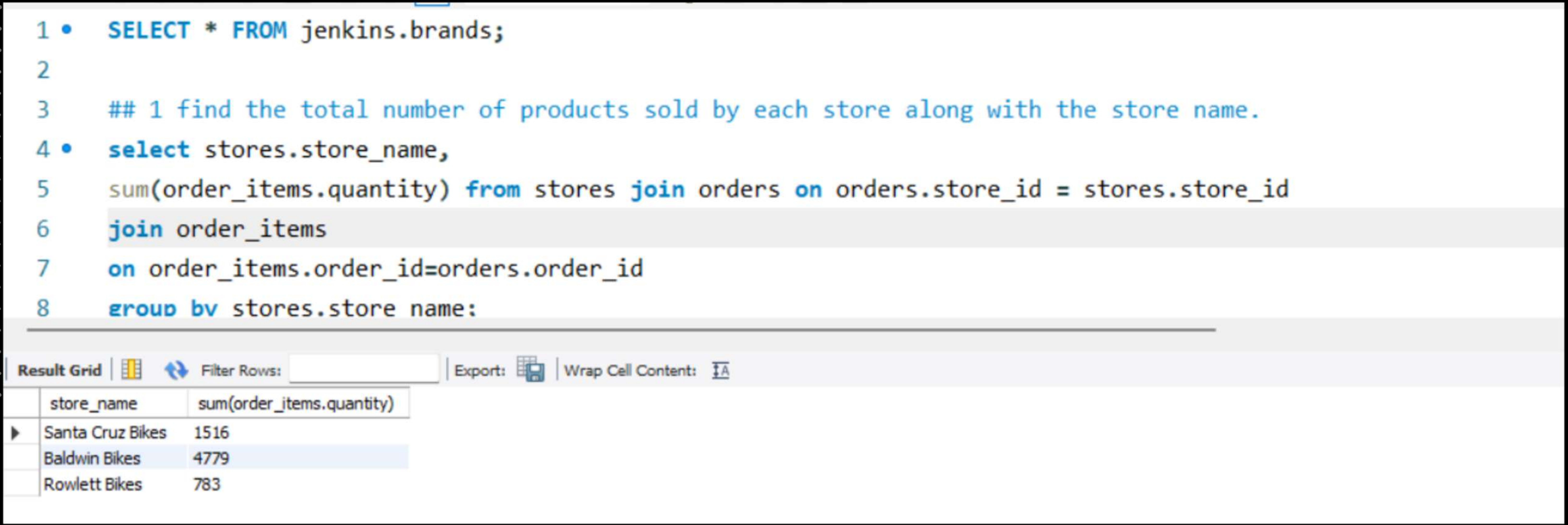
In this project we can handle the data of Jenson store and find multiple outcomes using SQL



By - Parth

PROJECT INTRODUCTION

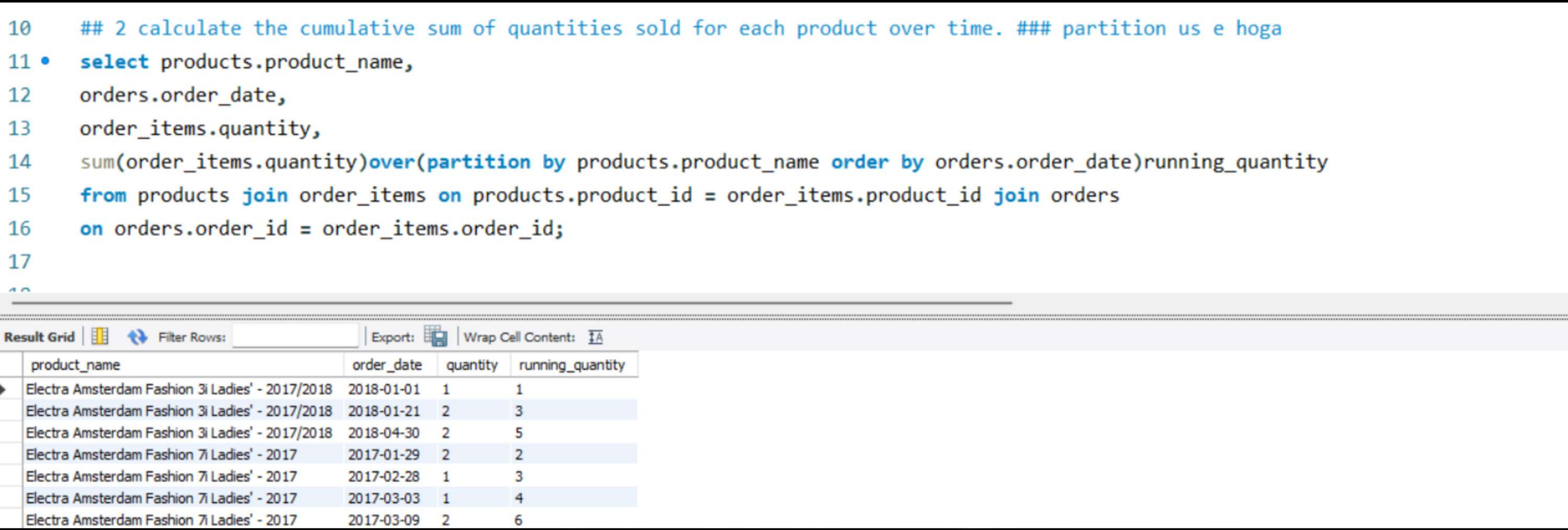
As a Data Analyst, I worked on the Milestone-2 project using SQL to analyze business operations for Jenson USA, a retail company. The goal was to derive actionable insights from their database covering customer behavior, staff performance, inventory levels, and store sales. By writing SQL queries, I explored key metrics that help optimize decision-making across different departments of the organization.



```
SELECT * FROM jenkins.brands;

## 1 find the total number of products sold by each store along with the store name.
select stores.store_name,
sum(order_items.quantity) from stores join orders on orders.store_id = stores.store_id
join order_items
on order_items.order_id=orders.order_id
group by stores.store_name;
```

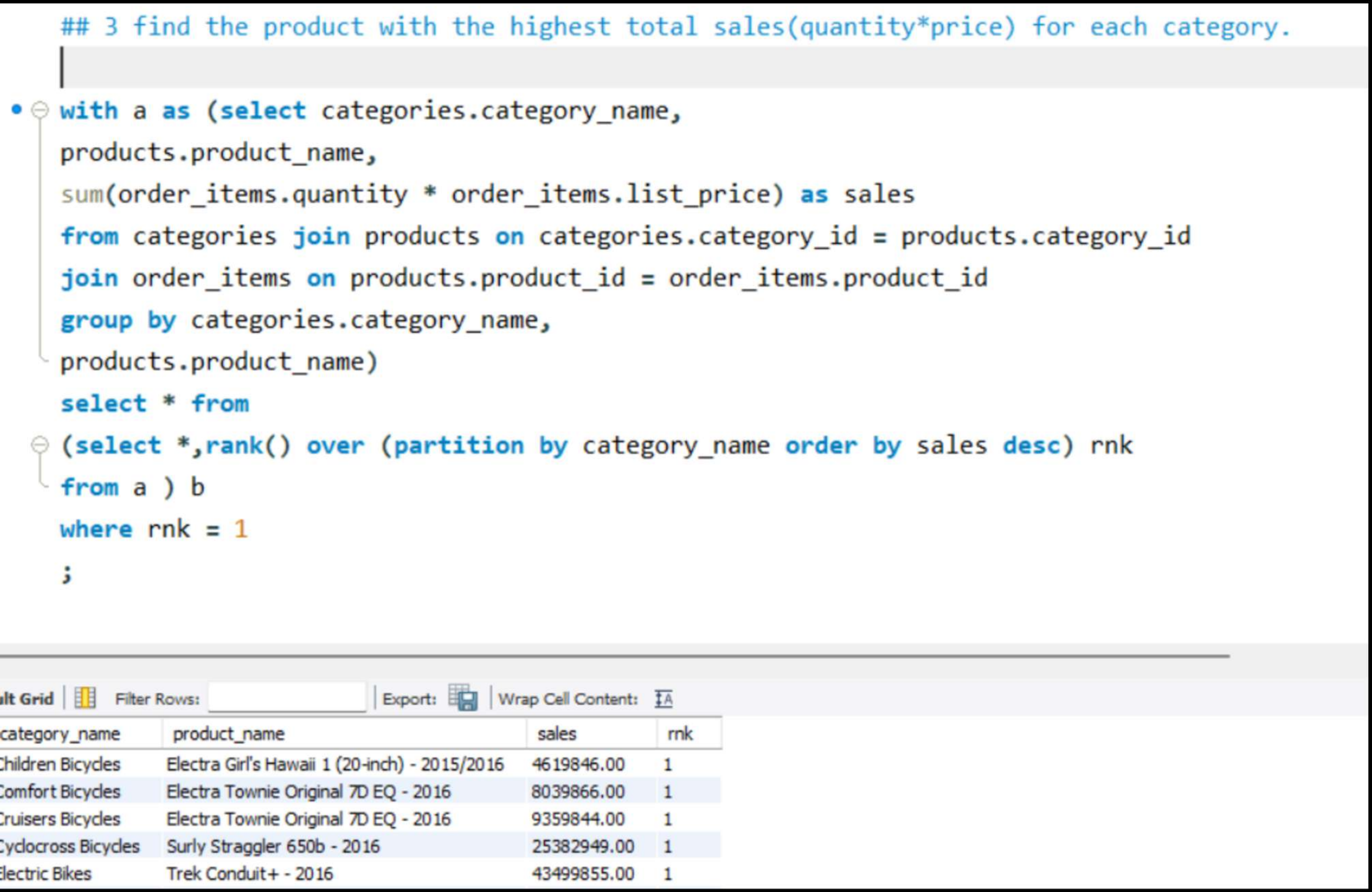
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

```
store_name        sum(order_items.quantity)
Santa Cruz Bikes  1516
Baldwin Bikes     4779
Rowlett Bikes     783
```


```
## 2 calculate the cumulative sum of quantities sold for each product over time. ### partition us e hoga
select products.product_name,
orders.order_date,
order_items.quantity,
sum(order_items.quantity)over(partition by products.product_name order by orders.order_date)running_quantity
from products join order_items on products.product_id = order_items.product_id join orders
on orders.order_id = order_items.order_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

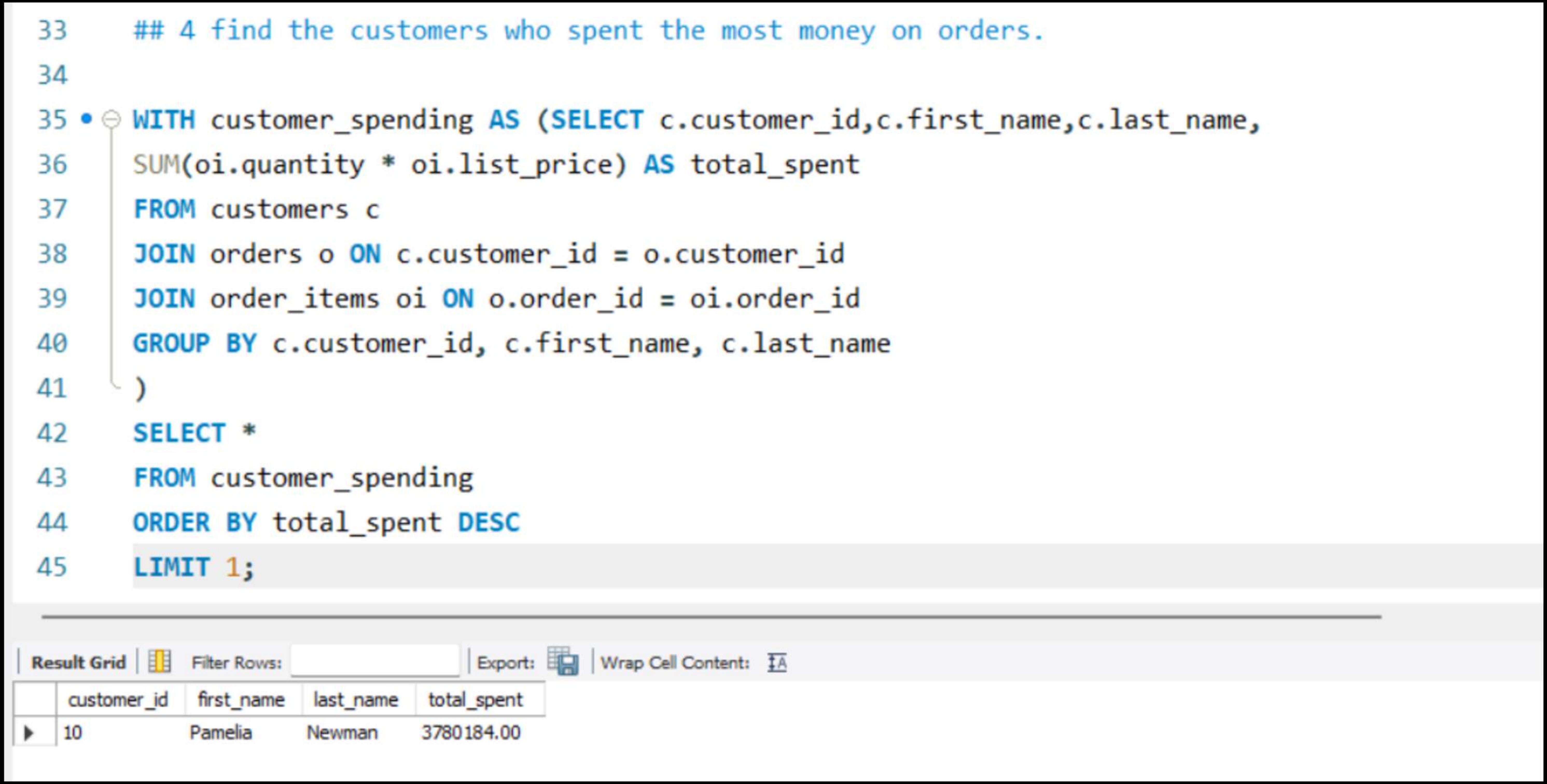
product_name	order_date	quantity	running_quantity
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	2018-01-01	1	1
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	2018-01-21	2	3
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	2018-04-30	2	5
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-01-29	2	2
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-02-28	1	3
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-03-03	1	4
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-03-09	2	6



```
## 3 find the product with the highest total sales(quantity*price) for each category.

with a as (select categories.category_name,
products.product_name,
sum(order_items.quantity * order_items.list_price) as sales
from categories join products on categories.category_id = products.category_id
join order_items on products.product_id = order_items.product_id
group by categories.category_name,
products.product_name)
select * from
(select *,rank() over (partition by category_name order by sales desc) rnk
from a ) b
where rnk = 1
;
```

lt Grid | Filter Rows: | Export: | Wrap Cell Content:

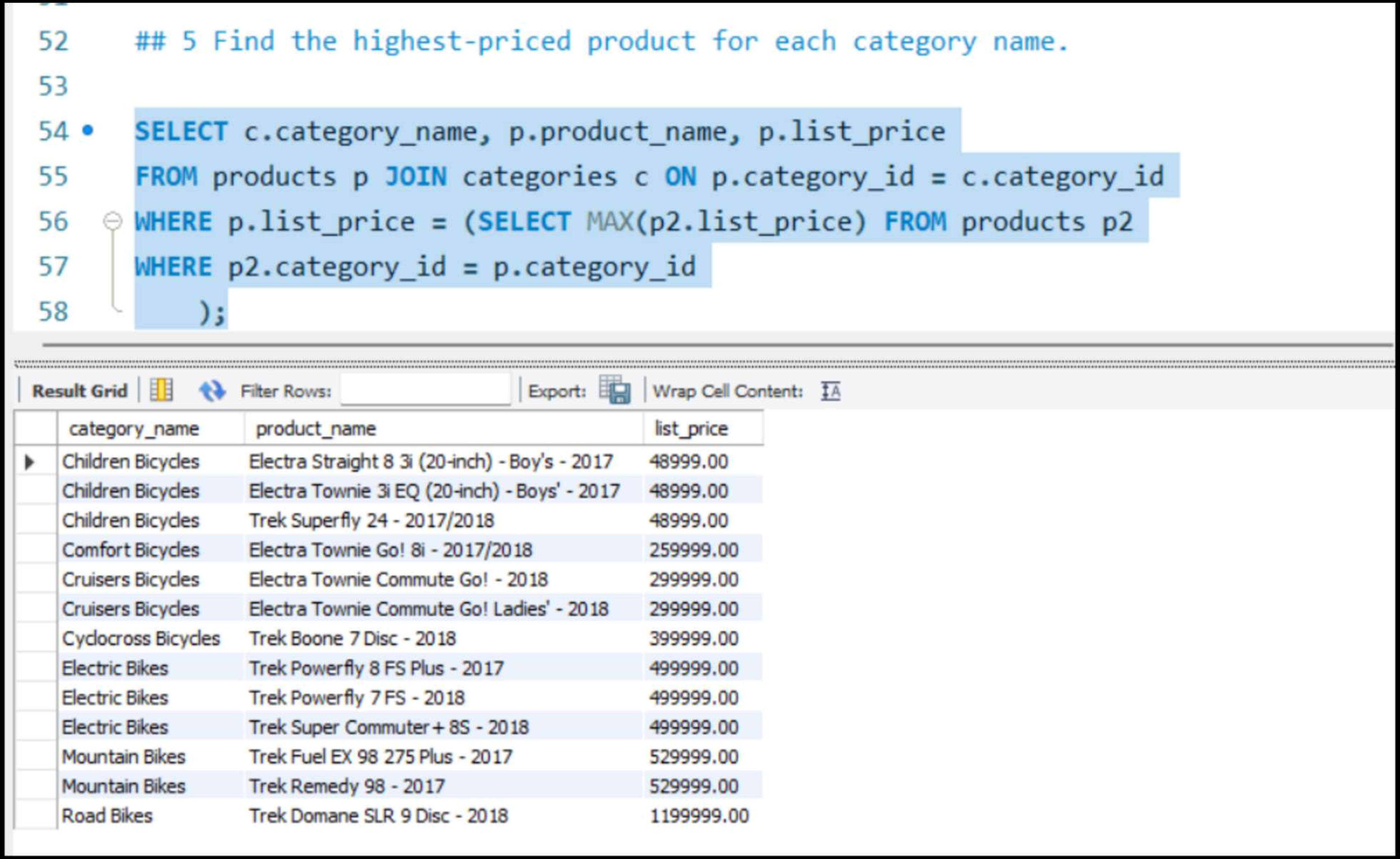
category_name	product_name	sales	rnk
hildren Bicycles	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	4619846.00	1
omfort Bicycles	Electra Townie Original 7D EQ - 2016	8039866.00	1
ruisers Bicycles	Electra Townie Original 7D EQ - 2016	9359844.00	1
yclocross Bicycles	Surly Straggler 650b - 2016	25382949.00	1
lectric Bikes	Trek Conduit+ - 2016	43499855.00	1



```sql
## 4 find the customers who spent the most money on orders.

WITH customer_spending AS (SELECT c.customer_id,c.first_name,c.last_name,
SUM(oi.quantity * oi.list_price) AS total_spent
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
JOIN order_items oi ON o.order_id = oi.order_id
GROUP BY c.customer_id, c.first_name, c.last_name
)
SELECT *
FROM customer_spending
ORDER BY total_spent DESC
LIMIT 1;
```

customer_id	first_name	last_name	total_spent
10	Pamelia	Newman	3780184.00

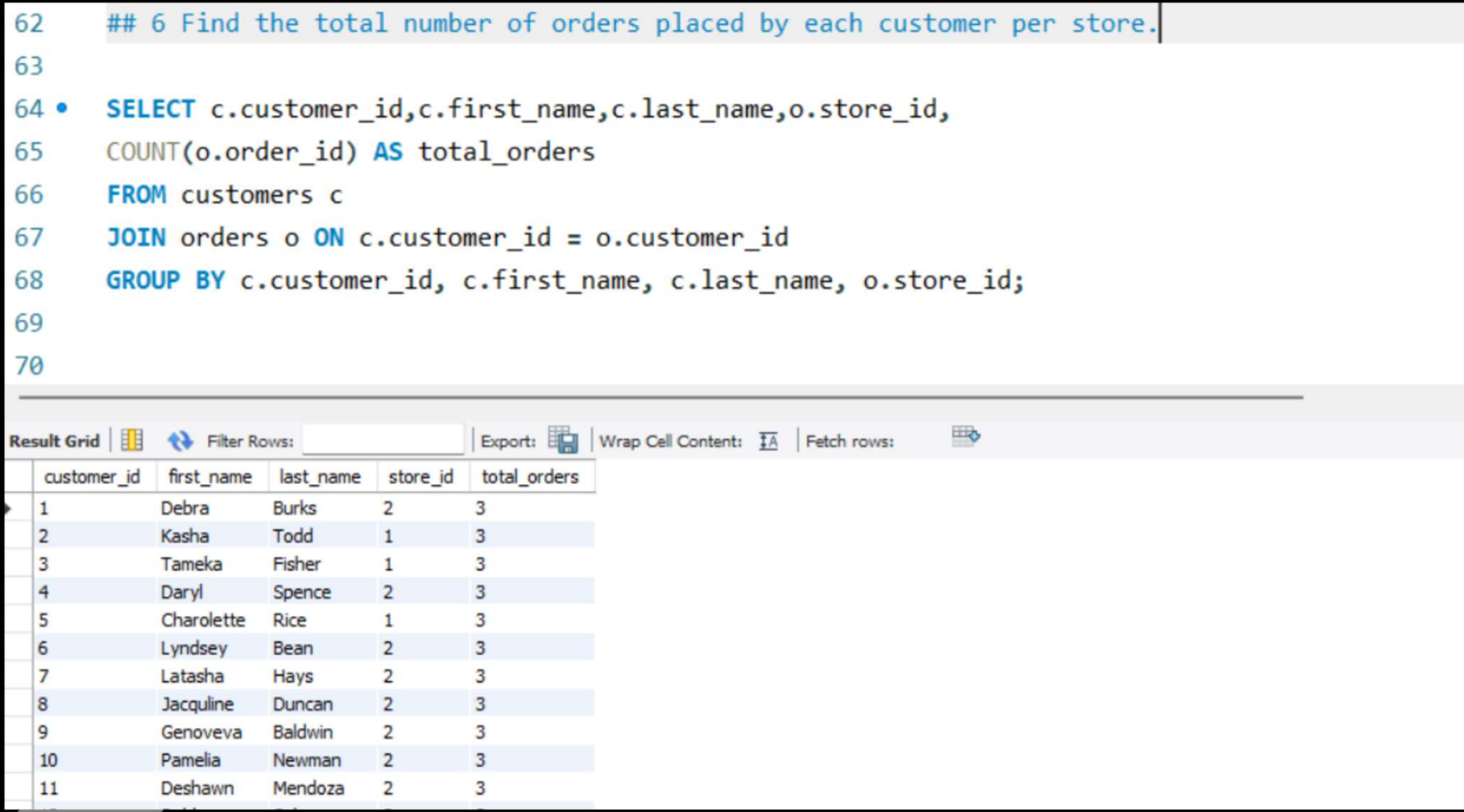
JENSON USA



```
## 5 Find the highest-priced product for each category name.

SELECT c.category_name, p.product_name, p.list_price
FROM products p JOIN categories c ON p.category_id = c.category_id
WHERE p.list_price = (SELECT MAX(p2.list_price) FROM products p2
WHERE p2.category_id = p.category_id
    );
```

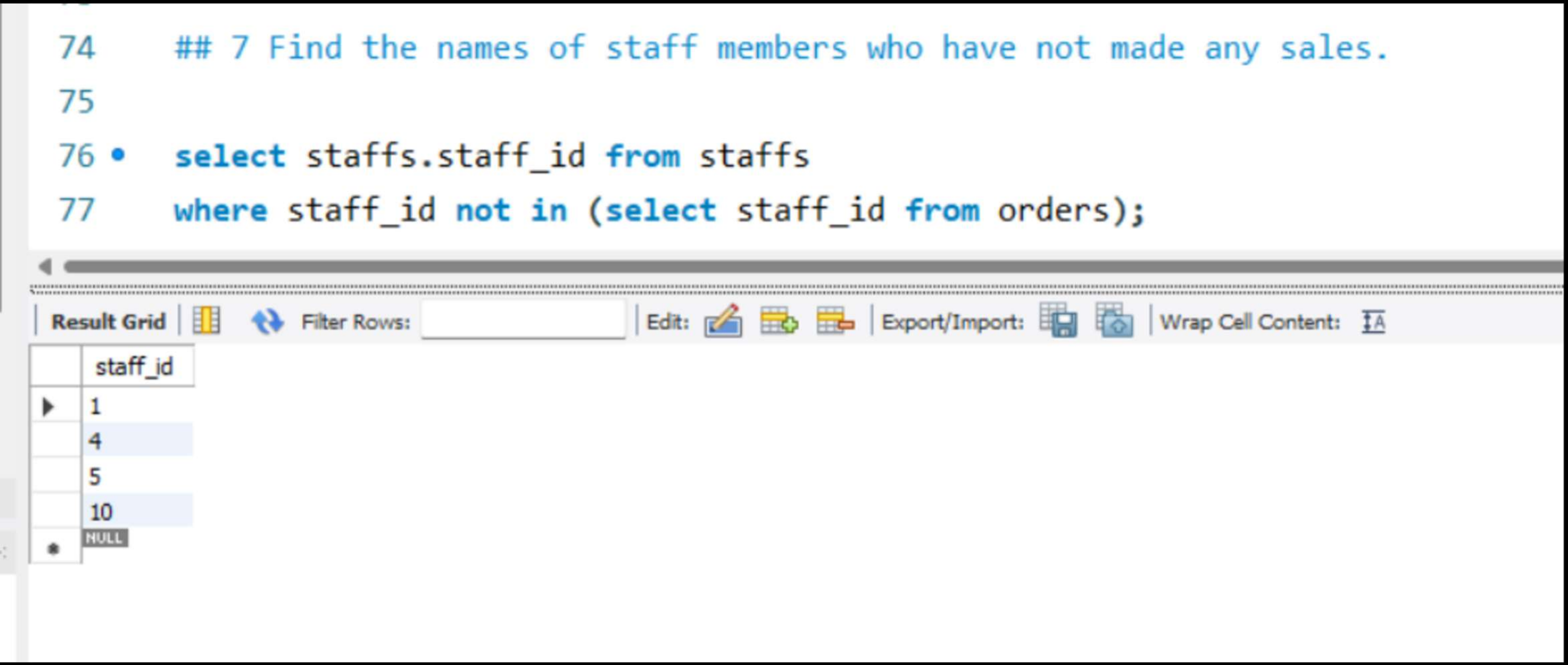
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category_name	product_name	list_price
Children Bicycles	Electra Straight 8 3i (20-inch) - Boy's - 2017	48999.00
Children Bicycles	Electra Townie 3i EQ (20-inch) - Boys' - 2017	48999.00
Children Bicycles	Trek Superfly 24 - 2017/2018	48999.00
Comfort Bicycles	Electra Townie Go! 8i - 2017/2018	259999.00
Cruisers Bicycles	Electra Townie Commute Go! - 2018	299999.00
Cruisers Bicycles	Electra Townie Commute Go! Ladies' - 2018	299999.00
Cyclocross Bicycles	Trek Boone 7 Disc - 2018	399999.00
Electric Bikes	Trek Powerfly 8 FS Plus - 2017	499999.00
Electric Bikes	Trek Powerfly 7 FS - 2018	499999.00
Electric Bikes	Trek Super Commuter + 8S - 2018	499999.00
Mountain Bikes	Trek Fuel EX 98 275 Plus - 2017	529999.00
Mountain Bikes	Trek Remedy 98 - 2017	529999.00
Road Bikes	Trek Domane SLR 9 Disc - 2018	1199999.00



```
## 6 Find the total number of orders placed by each customer per store.

SELECT c.customer_id,c.first_name,c.last_name,o.store_id,
COUNT(o.order_id) AS total_orders
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.first_name, c.last_name, o.store_id;
```

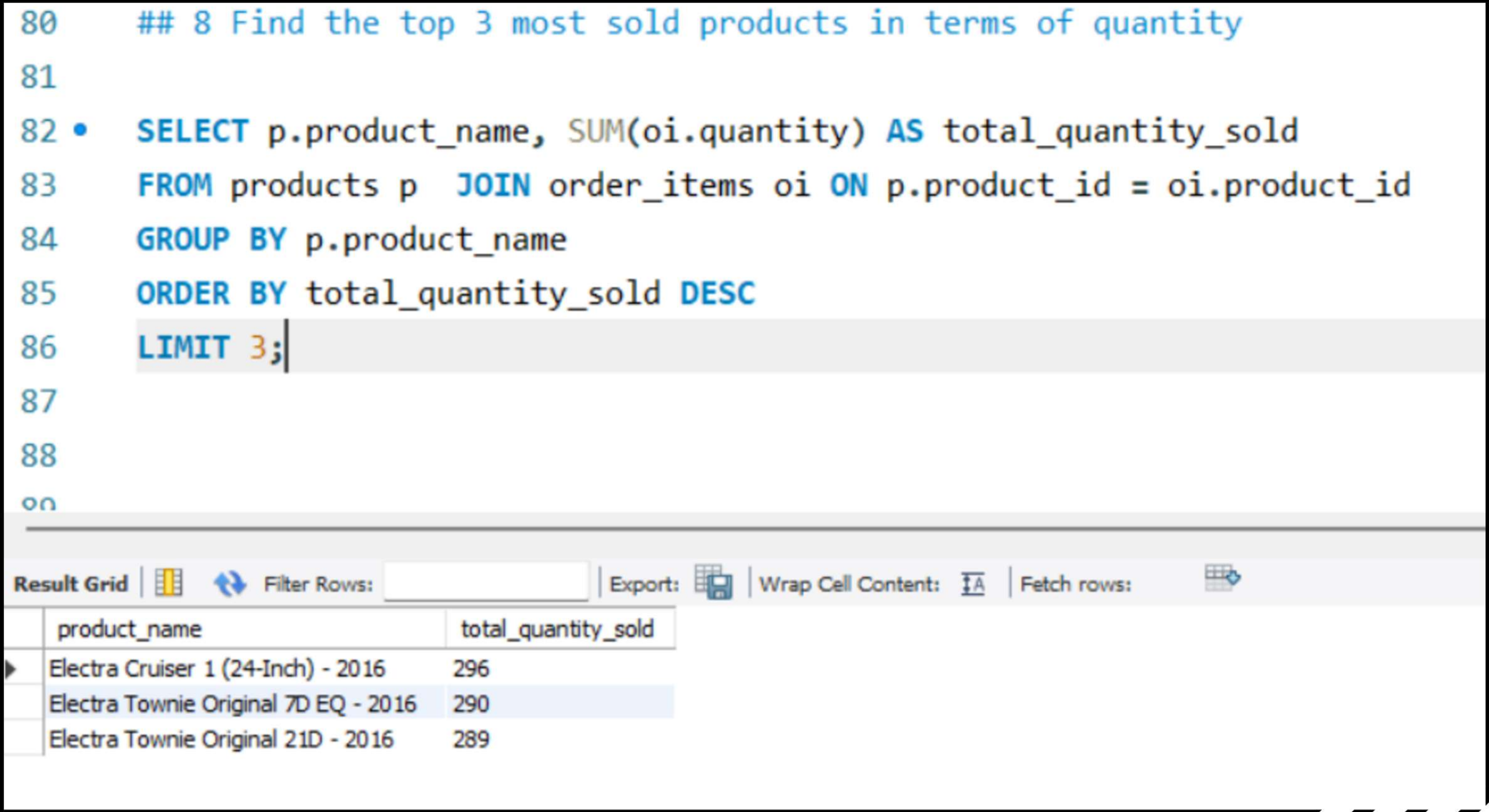
customer_id	first_name	last_name	store_id	total_orders
1	Debra	Burks	2	3
2	Kasha	Todd	1	3
3	Tameka	Fisher	1	3
4	Daryl	Spence	2	3
5	Charolette	Rice	1	3
6	Lyndsey	Bean	2	3
7	Latasha	Hays	2	3
8	Jacquline	Duncan	2	3
9	Genoveva	Baldwin	2	3
10	Pamelia	Newman	2	3
11	Deshawn	Mendoza	2	3



```
## 7 Find the names of staff members who have not made any sales.

select staffs.staff_id from staffs
where staff_id not in (select staff_id from orders);
```

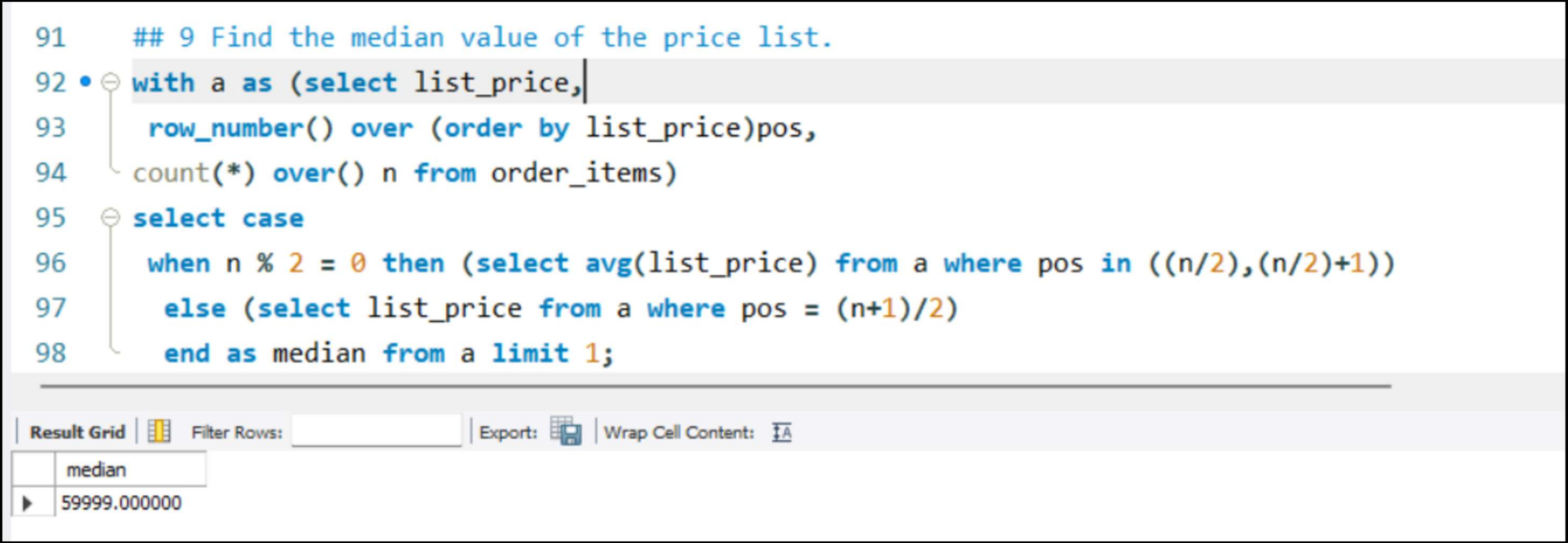
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

```
staff_id
1
4
5
10
NULL
```



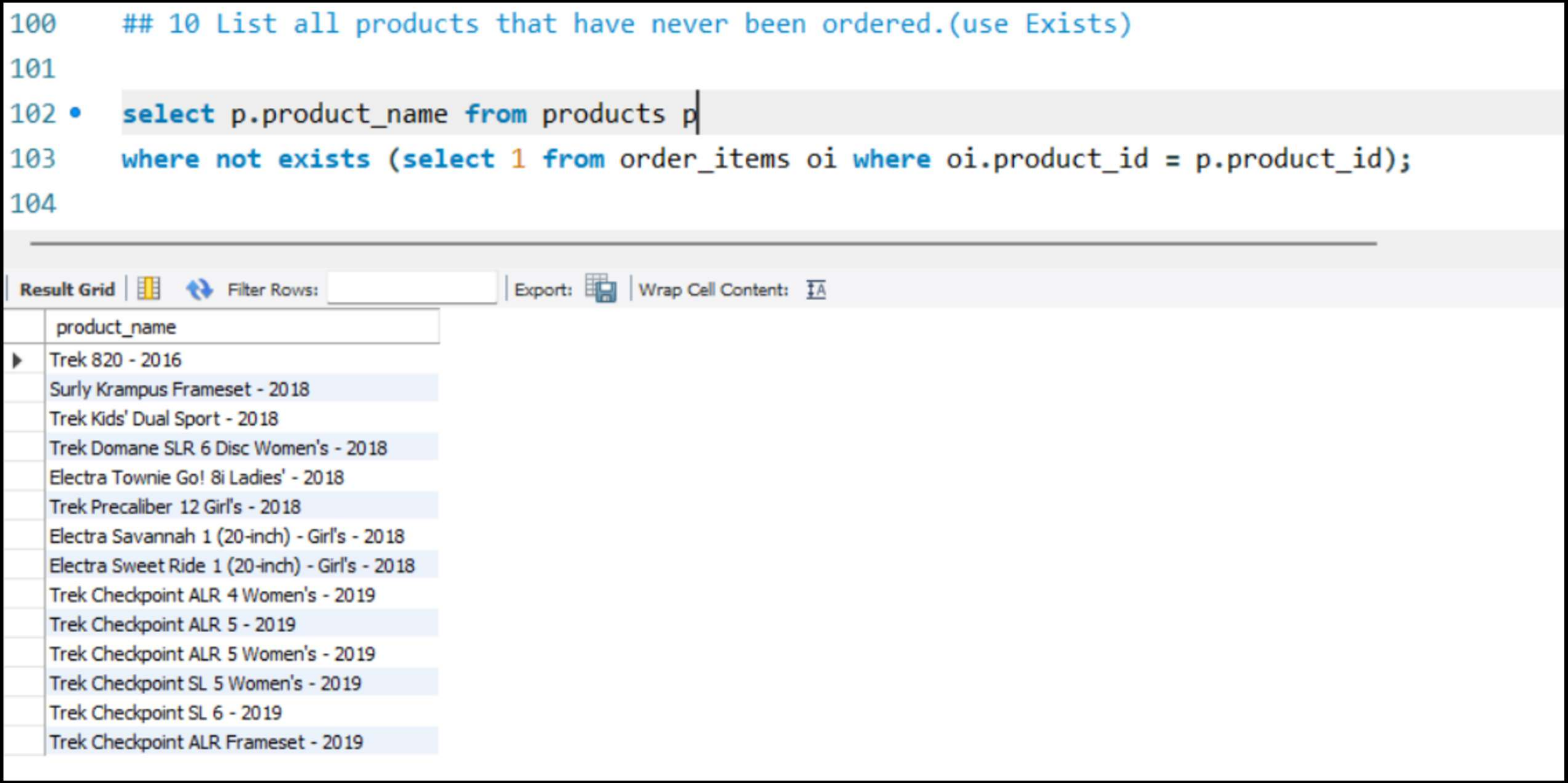
```
## 8 Find the top 3 most sold products in terms of quantity

SELECT p.product_name, SUM(oi.quantity) AS total_quantity_sold
FROM products p  JOIN order_items oi ON p.product_id = oi.product_id
GROUP BY p.product_name
ORDER BY total_quantity_sold DESC
LIMIT 3;
```

product_name	total_quantity_sold
Electra Cruiser 1 (24-Inch) - 2016	296
Electra Townie Original 7D EQ - 2016	290
Electra Townie Original 21D - 2016	289



```
## 9 Find the median value of the price list.
with a as (select list_price,
 row_number() over (order by list_price)pos,
count(*) over() n from order_items)
select case
 when n % 2 = 0 then (select avg(list_price) from a where pos in ((n/2),(n/2)+1))
  else (select list_price from a where pos = (n+1)/2)
  end as median from a limit 1;
```

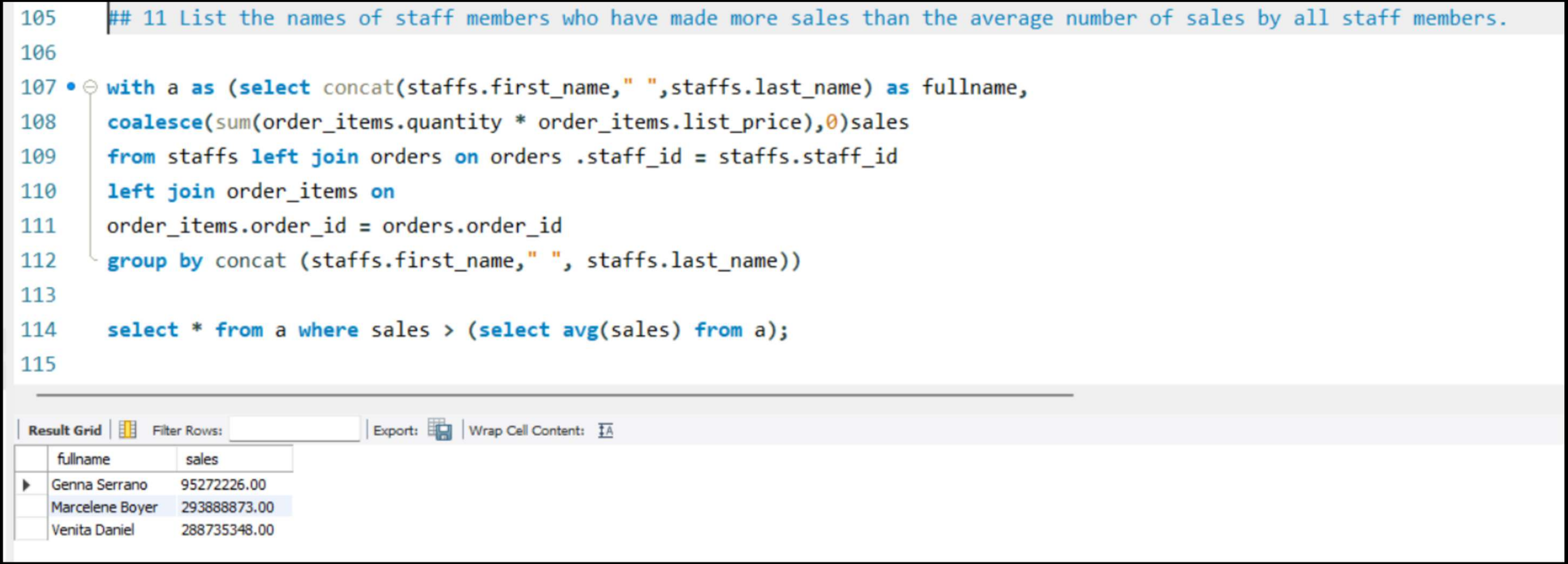
```
median
59999.000000
```



```
## 10 List all products that have never been ordered.(use Exists)

select p.product_name from products p
where not exists (select 1 from order_items oi where oi.product_id = p.product_id);
```

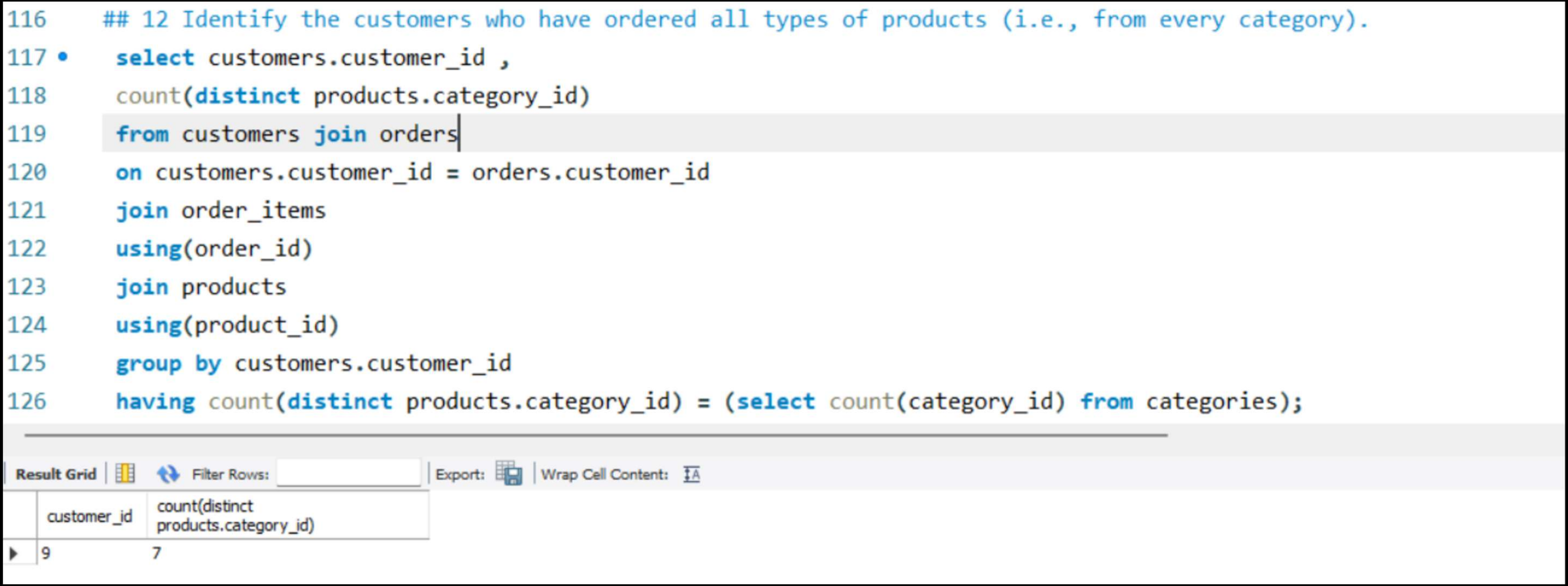
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_name
Trek 820 - 2016
Surly Krampus Frameset - 2018
Trek Kids' Dual Sport - 2018
Trek Domane SLR 6 Disc Women's - 2018
Electra Townie Go! 8i Ladies' - 2018
Trek Precaliber 12 Girl's - 2018
Electra Savannah 1 (20-inch) - Girl's - 2018
Electra Sweet Ride 1 (20-inch) - Girl's - 2018
Trek Checkpoint ALR 4 Women's - 2019
Trek Checkpoint ALR 5 - 2019
Trek Checkpoint ALR 5 Women's - 2019
Trek Checkpoint SL 5 Women's - 2019
Trek Checkpoint SL 6 - 2019
Trek Checkpoint ALR Frameset - 2019



```
## 11 List the names of staff members who have made more sales than the average number of sales by all staff members.

with a as (select concat(staffs.first_name," ",staffs.last_name) as fullname,
coalesce(sum(order_items.quantity * order_items.list_price),0)sales
from staffs left join orders on orders .staff_id = staffs.staff_id
left join order_items on
order_items.order_id = orders.order_id
group by concat (staffs.first_name," ", staffs.last_name))

select * from a where sales > (select avg(sales) from a);
```

fullname	sales
Genna Serrano	95272226.00
Marcelene Boyer	293888873.00
Venita Daniel	288735348.00



```
## 12 Identify the customers who have ordered all types of products (i.e., from every category).
select customers.customer_id ,
count(distinct products.category_id)
from customers join orders
on customers.customer_id = orders.customer_id
join order_items
using(order_id)
join products
using(product_id)
group by customers.customer_id
having count(distinct products.category_id) = (select count(category_id) from categories);
```

customer_id	count(distinct products.category_id)
9	7

Summary of SQL Tasks Completed

Through this project, I developed and applied core SQL skills to solve real-world business problems. Below is a summary of the insights generated through structured queries

- *Store-level Sales Overview* – Identified how many products each store sold, helping evaluate store performance.
- *Product Demand Analysis* – Calculated the cumulative quantity sold for each product to track demand over time.
- *Top-selling Products by Category* – Found the highest revenue-generating products in each category using total sales (price × quantity).
- *Customer Spending Behavior* – Discovered which customer spent the most on purchases, highlighting top buyers.

These insights enabled a comprehensive understanding of the sales dynamics and customer behavior at Jenson USA, demonstrating how data can support better business strategies.

Submitted by Parth Mishra