



SQL Project

Chohort 3

Presented By
Sagar Kumar



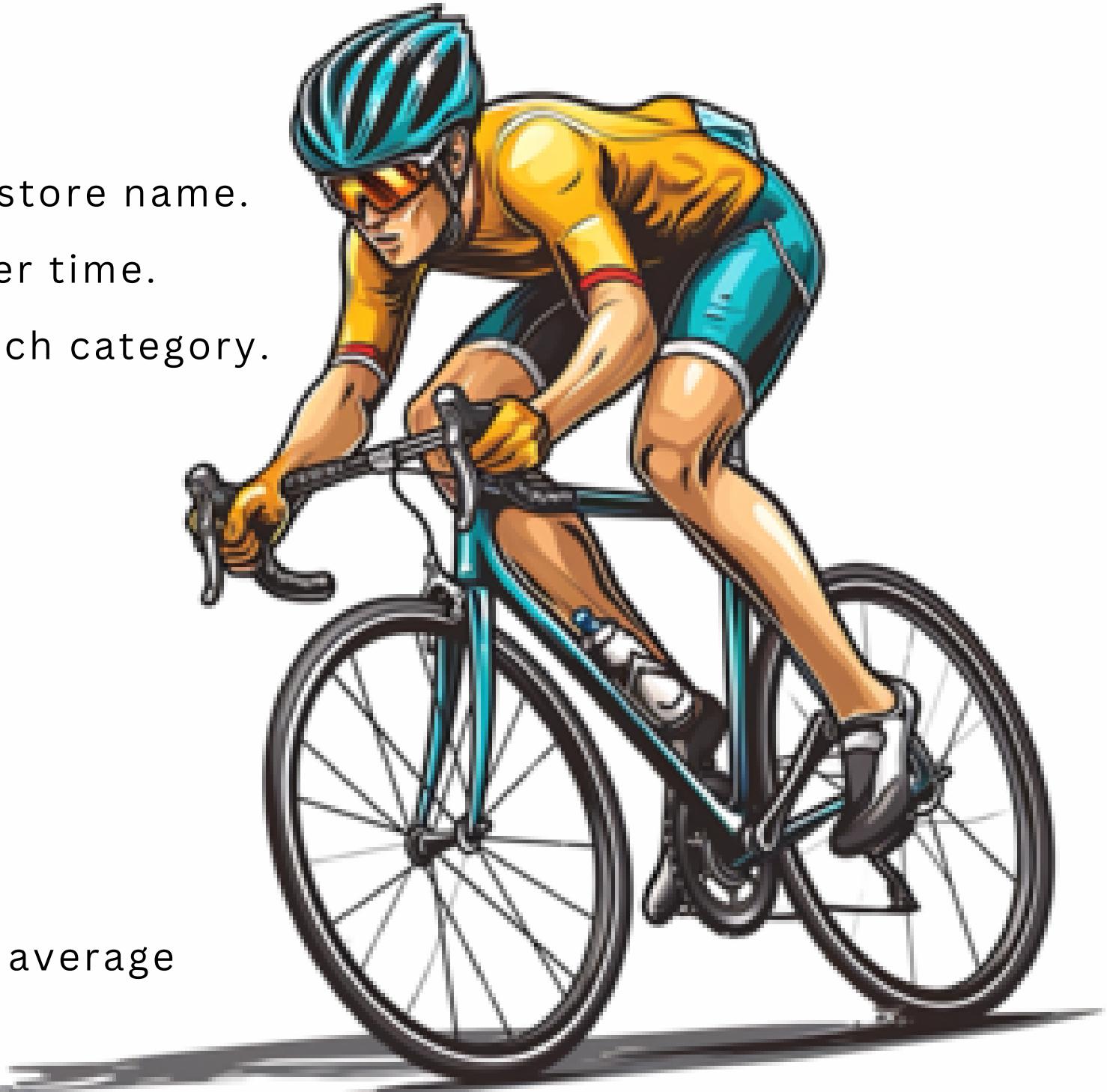
About Jensons USA

- Founded: 1994 by Mike Cachat in Brea, California.
- Origins: Started as a small bike tune-up business, evolved into selling parts and accessories.
- Growth: Relocated multiple times; in 2010 moved to a 74,000 sq. ft. headquarters in Riverside, California.
- Employees: Around 102 employees across North America, Asia, and Europe.
- Revenue: Approximately \$35 million annually.
- Headquarters: 1615 Eastridge Ave, Riverside, California.
- Community Engagement: Sponsors NICA mountain bike teams, participates in cycling events, and runs the "Kommute for Kids" initiative.
- Partnerships: Collaborates with World Bicycle Relief and IMBA to promote global cycling access.
- Mission: Aims to be the best for customers, vendors, and employees, maintaining a strong reputation in the cycling industry.

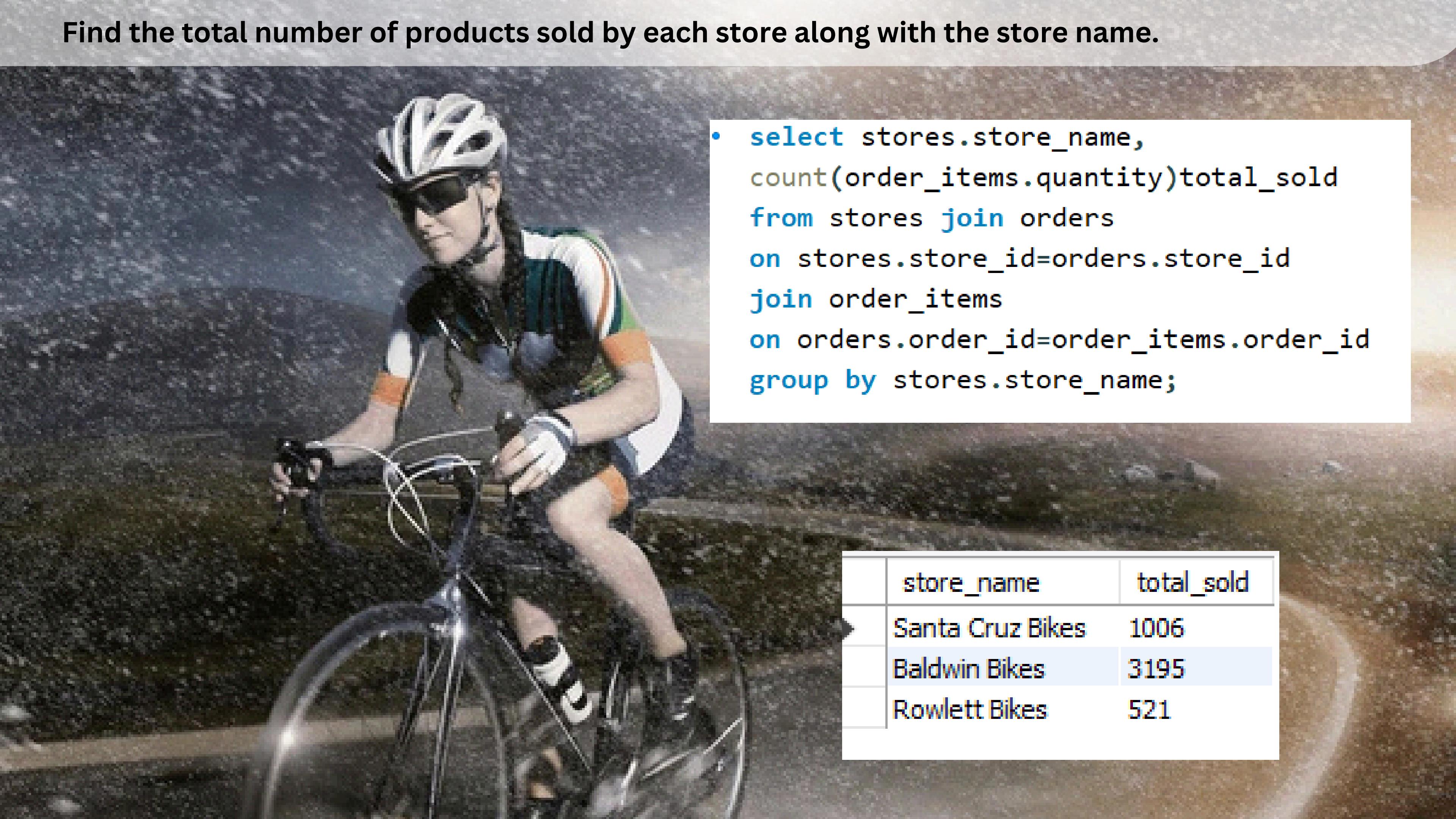


Questions

1. Find the total number of products sold by each store along with the store name.
2. Calculate the cumulative sum of quantities sold for each product over time.
3. Find the product with the highest total sales (quantity * price) for each category.
4. Find the customer who spent the most money on orders.
5. Find the highest-priced product for each category name.
6. Find the total number of orders placed by each customer per store.
7. Find the names of staff members who have not made any sales.
8. Find the top 3 most sold products in terms of quantity.
9. Find the median value of the price list.
10. List all products that have never been ordered.(use Exists)
11. List the names of staff members who have made more sales than the average number of sales by all staff members.
12. Identify the customers who have ordered all types of products (i.e., from every category).



Find the total number of products sold by each store along with the store name.



- ```
select stores.store_name,
count(order_items.quantity)total_sold
from stores join orders
on stores.store_id=orders.store_id
join order_items
on orders.order_id=order_items.order_id
group by stores.store_name;
```

|                  | store_name | total_sold |
|------------------|------------|------------|
| Santa Cruz Bikes | 1006       |            |
| Baldwin Bikes    | 3195       |            |
| Rowlett Bikes    | 521        |            |

# Calculate the cumulative sum of quantities sold for each product over time.

```
• 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) cumulative_quantity
from products
join order_items
join orders
on products.product_id = order_items.product_id
and
orders.order_id=order_items.order_id;
```

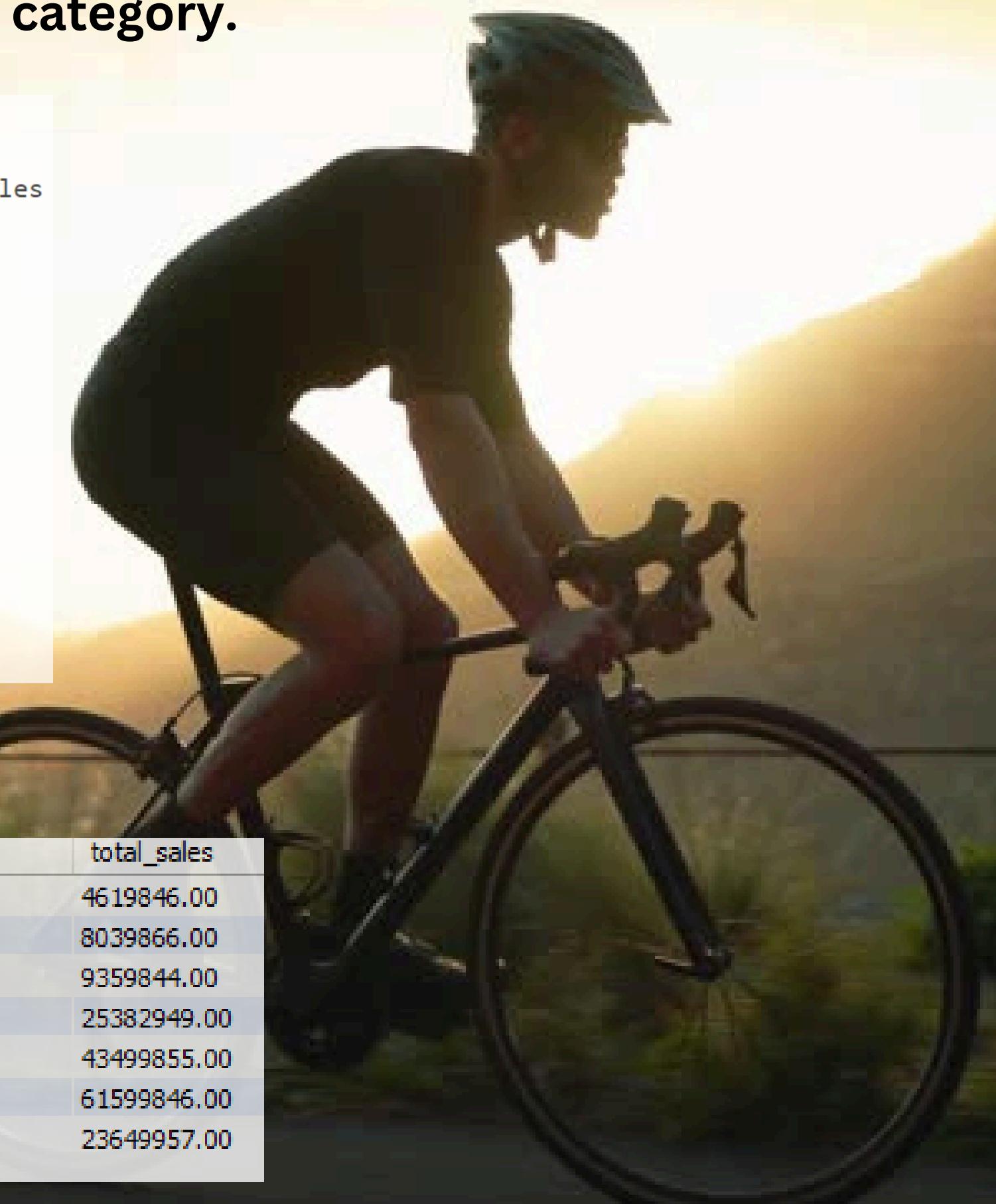
Result Grid | Filter Rows:  Export: Wrap Cell Content: Fetch rows:

|   | product_name                                     | order_date | quantity | cumulative_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                   |
|   | Electra Amsterdam Fashion 7i Ladies' - 2017      | 2017-04-06 | 1        | 7                   |

# 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) total_sales
 from products
 join categories
 join order_items
 on products.category_id = categories.category_id
 and products.product_id = order_items.product_id
 group by products.product_name,categories.category_name)

 select category_name,product_name,total_sales from
 (select * ,
 rank () over (partition by category_name
 order by total_sales desc)rnk from a)b
 where rnk = 1;
```



| category_name       | product_name                                  | total_sales |
|---------------------|-----------------------------------------------|-------------|
| Children Bicycles   | Electra Girl's Hawaii 1 (20-inch) - 2015/2016 | 4619846.00  |
| Comfort Bicycles    | Electra Townie Original 7D EQ - 2016          | 8039866.00  |
| Cruisers Bicycles   | Electra Townie Original 7D EQ - 2016          | 9359844.00  |
| Cyclocross Bicycles | Surly Straggler 650b - 2016                   | 25382949.00 |
| Electric Bikes      | Trek Conduit+ - 2016                          | 43499855.00 |
| Mountain Bikes      | Trek Slash 8 275 - 2016                       | 61599846.00 |
| Road Bikes          | Trek Domane SLR 6 Disc - 2017                 | 23649957.00 |

# Find the customer who spent the most money on orders.

```
• SELECT
 CONCAT(customers.first_name,
 ' ',
 customers.last_name) full_name,
 SUM(order_items.quantity * order_items.list_price) spent
FROM
 customers
 JOIN
 orders
 JOIN
 order_items
 ON customers.customer_id = orders.customer_id
 AND orders.order_id = order_items.order_id
GROUP BY full_name
ORDER BY spent DESC
LIMIT 1;
```

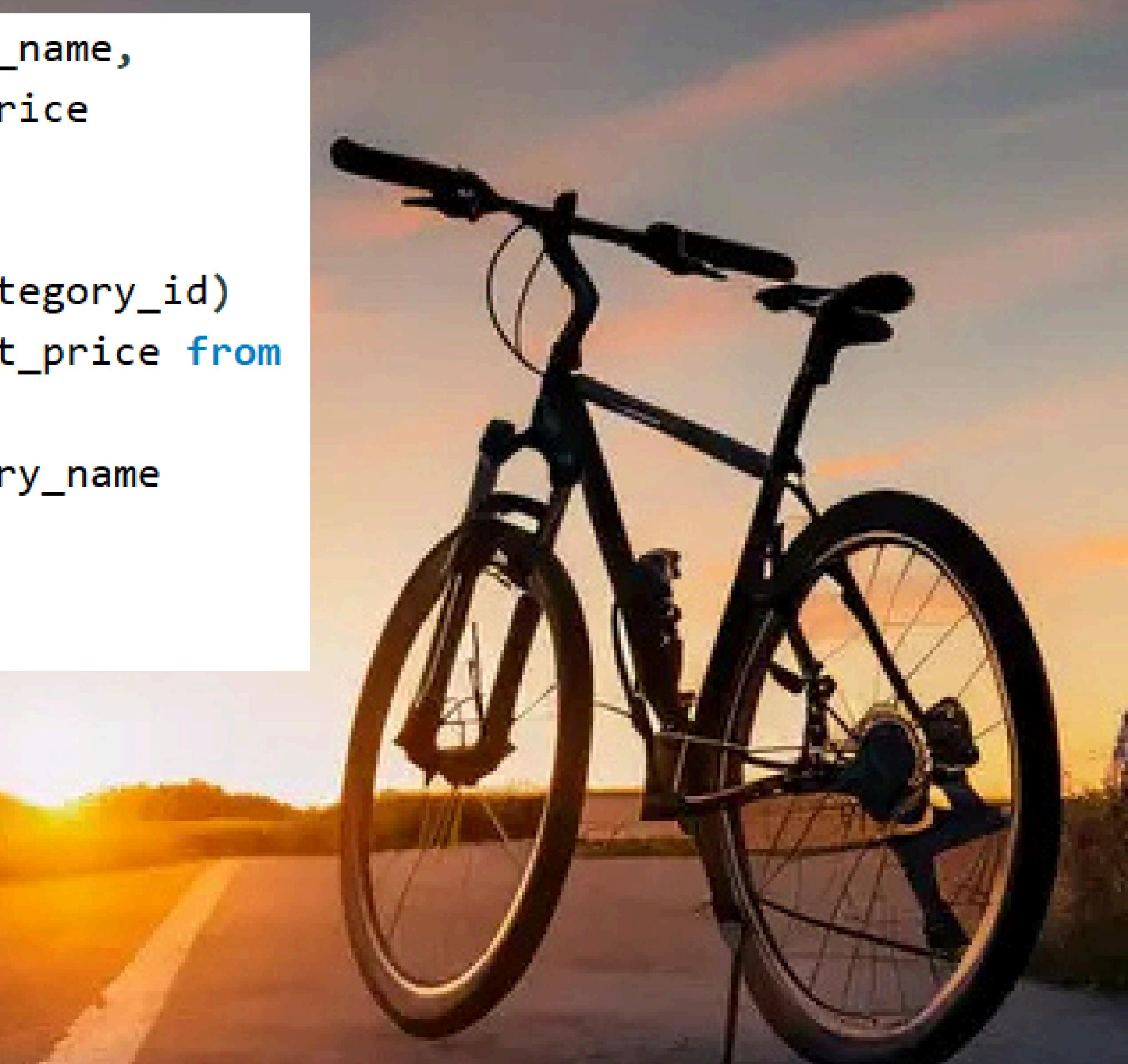
Result Grid | Filter Rows:

|   | full_name      | spent      |
|---|----------------|------------|
| ▶ | Pamelia Newman | 3780184.00 |

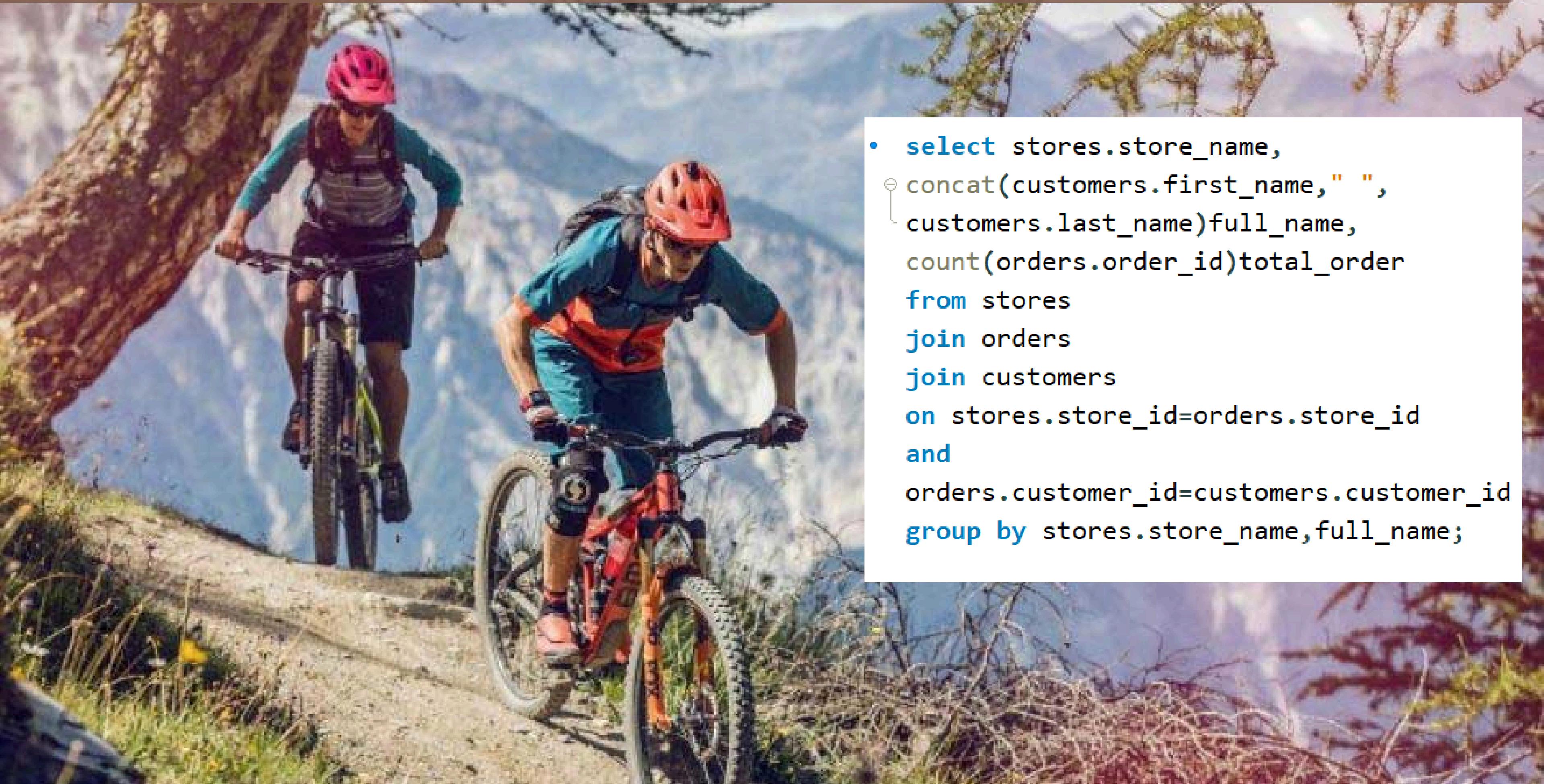
Find the highest-priced product for each category name.

```
• with a as (select categories.category_name,
products.product_name,products.list_price
from categories
join products
on categories.category_id=products.category_id)
select category_name,product_name,list_price from
(select * ,
dense_rank() over(partition by category_name
order by list_price)rnk from a)b
where rnk =1;
```

|   | category_name       | product_name                           | list_price |
|---|---------------------|----------------------------------------|------------|
| ▶ | Children Bicycles   | Strider Classic 12 Balance Bike - 2018 | 8999.00    |
|   | Comfort Bicycles    | Sun Bicycles Cruz 7 - Women's - 2017   | 41699.00   |
|   | Comfort Bicycles    | Sun Bicycles Cruz 7 - 2017             | 41699.00   |
|   | Cruisers Bicycles   | Pure Cycles Vine 8-Speed - 2016        | 429.00     |
|   | Cyclocross Bicycles | Surly Straggler 650b - 2018            | 1549.00    |
|   | Cyclocross Bicycles | Surly Straggler - 2018                 | 1549.00    |
|   | Cyclocross Bicycles | Surly Straggler - 2016                 | 1549.00    |
|   | Electric Bikes      | Sun Bicycles ElectroLite - 2017        | 155999.00  |
|   | Mountain Bikes      | Surly Krampus - 2018                   | 1499.00    |
|   | Road Bikes          | Surly Pack Rat - 2018                  | 1349.00    |



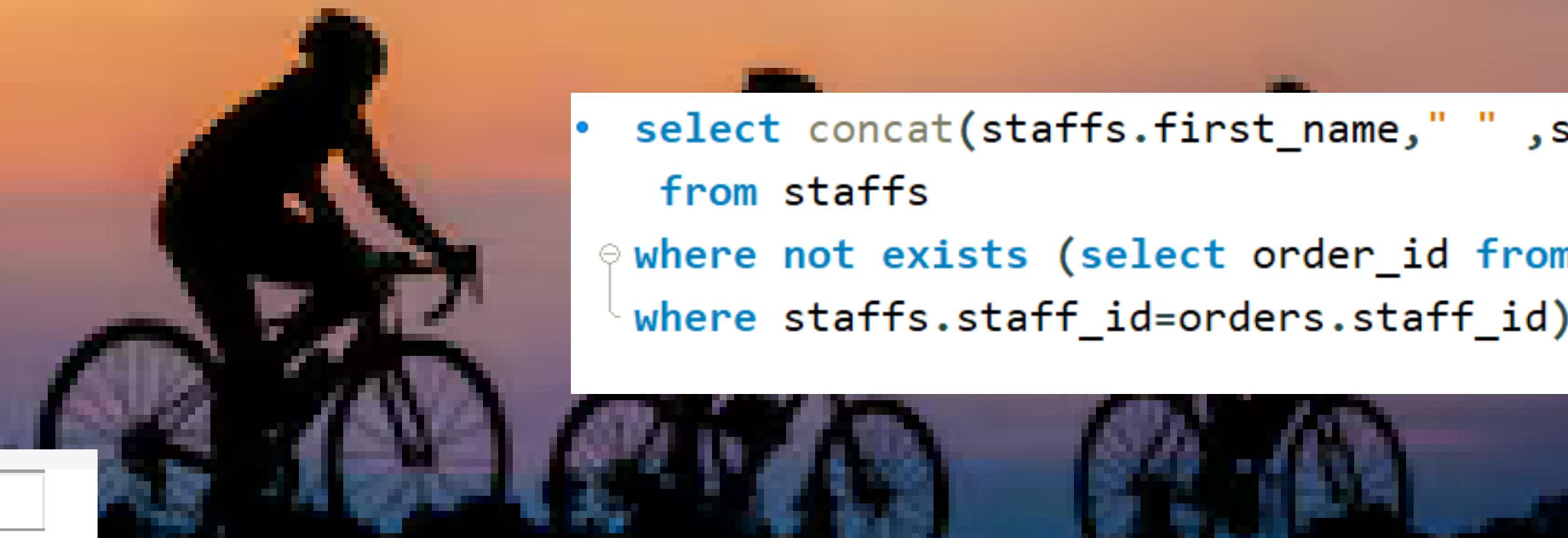
# Find the total number of orders placed by each customer per store.



- `select stores.store_name,  
concat(customers.first_name, " ",  
customers.last_name)full_name,  
count(orders.order_id)total_order  
from stores  
join orders  
join customers  
on stores.store_id=orders.store_id  
and  
orders.customer_id=customers.customer_id  
group by stores.store_name,full_name;`

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

- ```
select concat(staffs.first_name, " ", staffs.last_name)full_name,
count(orders.order_id)sales
from staffs left join orders
on staffs.staff_id=orders.staff_id
group by full_name
having sales = 0;
```



- ```
select concat(staffs.first_name, " ", staffs.last_name)full_name
from staffs
where not exists (select order_id from orders
where staffs.staff_id=orders.staff_id);
```

| full_name          |
|--------------------|
| Fabiola Jackson    |
| Virgie Wiggins     |
| Jannette David     |
| Bernardine Houston |

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

```
• with a as
 (select products.product_name,
 sum(order_items.quantity)sold_quantity
 from products join order_items
 on products.product_id=order_items.product_id
 group by products.product_name
 order by sold_quantity desc)
 select * from (
 select *,
 dense_rank() over(order by sold_quantity desc)rnk from a)b
 where rnk<=3;
```

| product_name                         | sold_quantity | rnk |
|--------------------------------------|---------------|-----|
| Electra Cruiser 1 (24-Inch) - 2016   | 296           | 1   |
| Electra Townie Original 7D EQ - 2016 | 290           | 2   |
| Electra Townie Original 21D - 2016   | 289           | 3   |

# Find the median value of the price list.

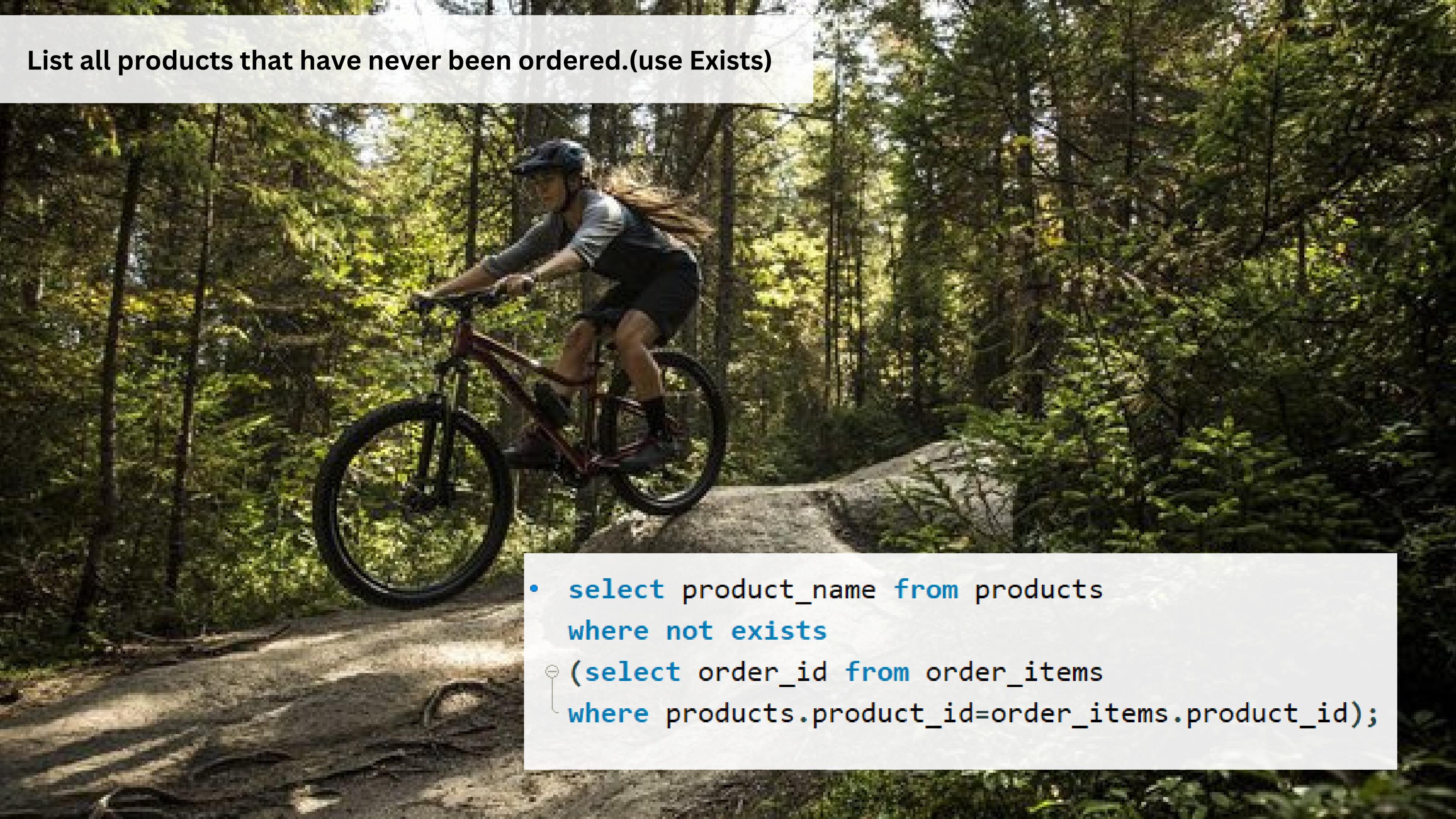
```
• with a as (select list_price,
 row_number() over(order by list_price)rn ,
 count(*)over()n from products)

 select case
 when n % 2 = 0 THEN
 (select avg(list_price) from a where rn in (n/2,n/2+1))
 else (select list_price from a where rn = (n+1)/2)
 end as median from a limit 1;
```

|   |          |
|---|----------|
|   | median   |
| ▶ | 74999.00 |



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



- `select product_name from products  
where not exists  
(select order_id from order_items  
where products.product_id=order_items.product_id);`

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

A photograph of a mountain biker in motion on a rocky trail through a dense forest. The biker is wearing a helmet and protective gear. A small inset window in the bottom right corner shows a screenshot of a database query results table.

|   | full_name       | total_sales  |
|---|-----------------|--------------|
| ▶ | Genna Serrano   | 95272226.00  |
|   | Marcelene Boyer | 293888873.00 |
|   | Venita Daniel   | 288735348.00 |

- with a as
  - (`select concat(staffs.first_name, " ", staffs.last_name)full_name,`  
`coalesce(sum(order_items.quantity*order_items.list_price),0)total_sales`  
`from staffs left join orders`  
`on staffs.staff_id=orders.staff_id`  
`left join order_items`  
`on orders.order_id = order_items.order_id`  
`group by full_name)`
  
`select * from a`  
`where total_sales>(select avg(total_sales) from a);`

## Identify the customers who have ordered all types of products (i.e., from every category).

- ```
select concat(customers.first_name, " ", customers.last_name)full_name,
count(order_items.product_id)
from customers
join orders
join order_items
join products
on customers.customer_id=orders.customer_id
and orders.order_id=order_items.order_id
and order_items.product_id=products.product_id
group by full_name
having count(distinct products.category_id)=
(select count(category_id) from categories);
```



full_name	count(order_items.product_id)
Genoveva Baldwin	9

Thank You



sagarburman49@gmail.com



<https://www.linkedin.com/in/sagar49>