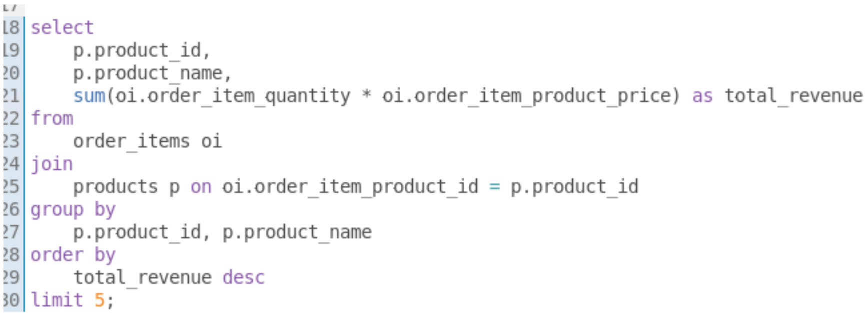
Sean McLean

ALY 6110

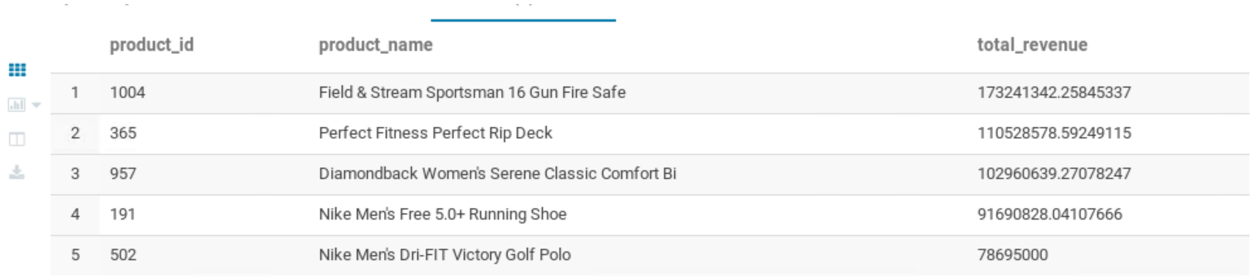
Module 3 Lab – Individual Lab #1

1. What are the top 5 revenue generating products for all years?

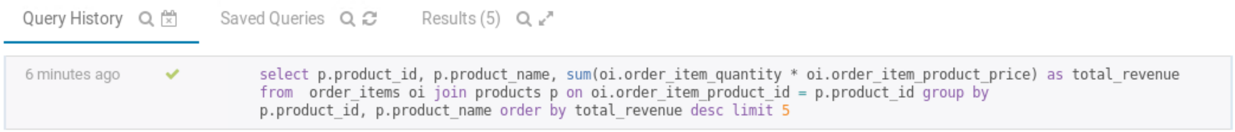
This query calculates the total revenue generated by each product across all years by summing the product of order_item_quantity and order_item_product_price for each product. It then returns the top 5 products with the highest total revenue. Despite reading the tutorials for the assignment the code was derived from Open AI to help with this question.



```sql
select
    p.product_id,
    p.product_name,
    sum(oi.order_item_quantity * oi.order_item_product_price) as total_revenue
from
    order_items oi
join
    products p on oi.order_item_product_id = p.product_id
group by
    p.product_id, p.product_name
order by
    total_revenue desc
limit 5;
```



	product_id	product_name	total_revenue
1	1004	Field & Stream Sportsman 16 Gun Fire Safe	173241342.25845337
2	365	Perfect Fitness Perfect Rip Deck	110528578.59249115
3	957	Diamondback Women's Serene Classic Comfort Bi	102960639.27078247
4	191	Nike Men's Free 5.0+ Running Shoe	91690828.04107666
5	502	Nike Men's Dri-FIT Victory Golf Polo	78695000

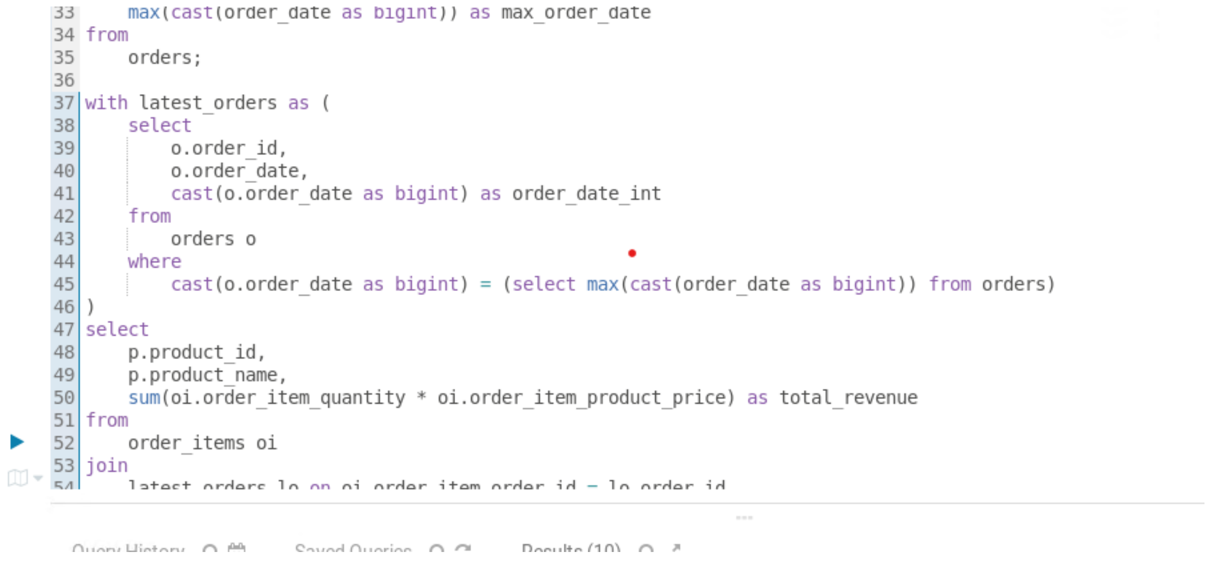


6 minutes ago ✓

```sql
select p.product_id, p.product_name, sum(oi.order_item_quantity * oi.order_item_product_price) as total_revenue
from  order_items oi join products p on oi.order_item_product_id = p.product_id group by
p.product_id, p.product_name order by total_revenue desc limit 5
```

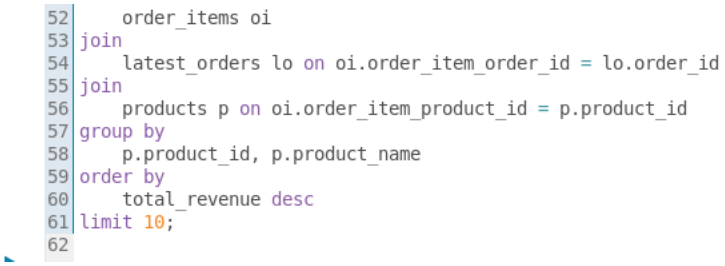
2. What are the top 5 revenue generating products for latest year?

This query first identifies the latest year by finding the maximum value in the order_date field, converting it to a bigint. It then calculates the total revenue for each product within that year by summing the product of order_item_quantity and order_item_product_price, and returns the top 5 products with the highest total revenue. Despite reading the tutorials for the assignment the code was derived from Open AI to help with this question.

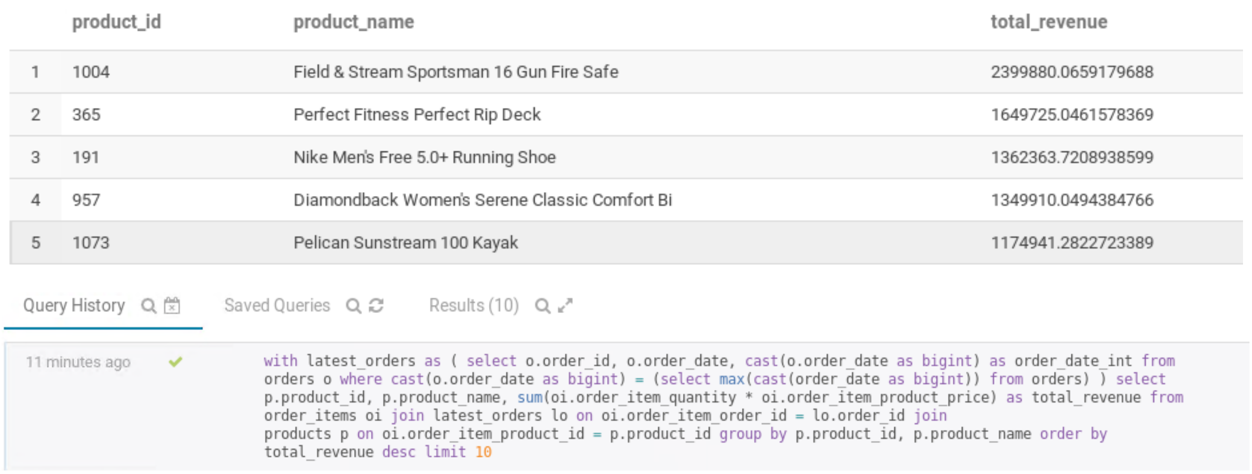


```
    max(cast(order_date as bigint)) as max_order_date
from
    orders;

with latest_orders as (
    select
        o.order_id,
        o.order_date,
        cast(o.order_date as bigint) as order_date_int
    from
        orders o
    where
        cast(o.order_date as bigint) = (select max(cast(order_date as bigint)) from orders)
)
select
    p.product_id,
    p.product_name,
    sum(oi.order_item_quantity * oi.order_item_product_price) as total_revenue
from
    order_items oi
join
    latest_orders lo on oi.order_item_order_id = lo.order_id
```



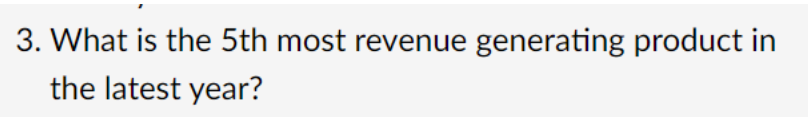
```
    order_items oi
join
    latest_orders lo on oi.order_item_order_id = lo.order_id
join
    products p on oi.order_item_product_id = p.product_id
group by
    p.product_id, p.product_name
order by
    total_revenue desc
limit 10;
```



	product_id	product_name	total_revenue
1	1004	Field & Stream Sportsman 16 Gun Fire Safe	2399880.0659179688
2	365	Perfect Fitness Perfect Rip Deck	1649725.0461578369
3	191	Nike Men's Free 5.0+ Running Shoe	1362363.7208938599
4	957	Diamondback Women's Serene Classic Comfort Bi	1349910.0494384766
5	1073	Pelican Sunstream 100 Kayak	1174941.2822723389

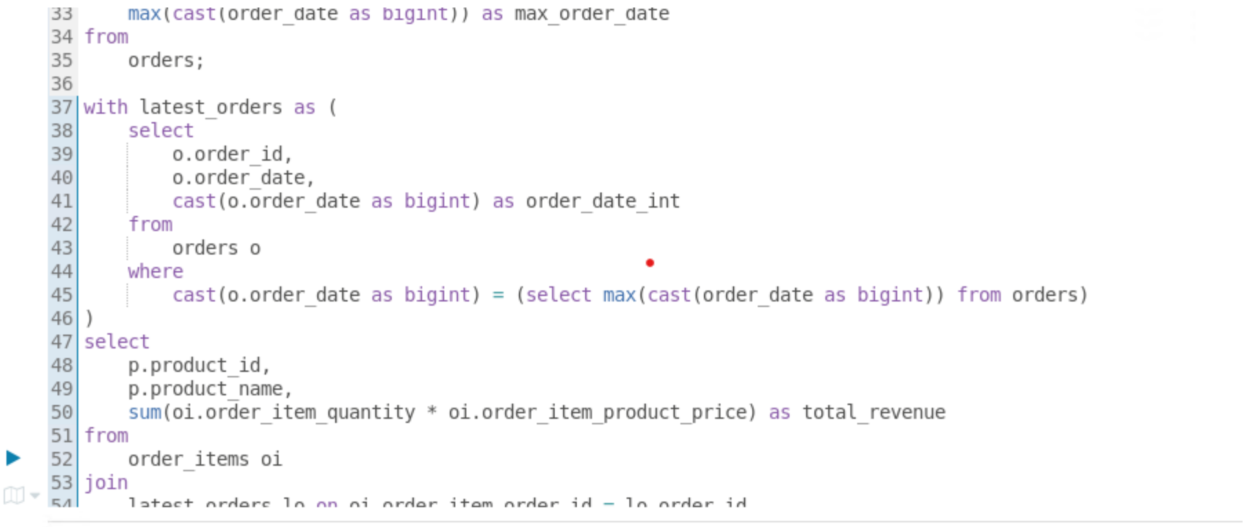
11 minutes ago ✓

```
with latest_orders as ( select o.order_id, o.order_date, cast(o.order_date as bigint) as order_date_int from
orders o where cast(o.order_date as bigint) = (select max(cast(order_date as bigint)) from orders) ) select
p.product_id, p.product_name, sum(oi.order_item_quantity * oi.order_item_product_price) as total_revenue from
order_items oi join latest_orders lo on oi.order_item_order_id = lo.order_id join
products p on oi.order_item_product_id = p.product_id group by p.product_id, p.product_name order by
total_revenue desc limit 10
```



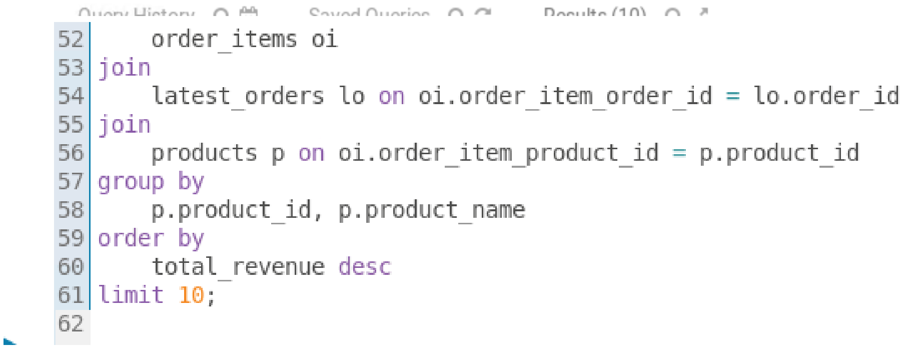
3. What is the 5th most revenue generating product in the latest year?

Using the latest year identified in the previous query, this query ranks products by total revenue using the ROW_NUMBER function. It calculates the revenue by summing the product of order_item_quantity and order_item_product_price for each product and then selects the product with the 5th highest revenue for that year. Despite reading the tutorials for the assignment the code was derived from Open AI to help with this question.

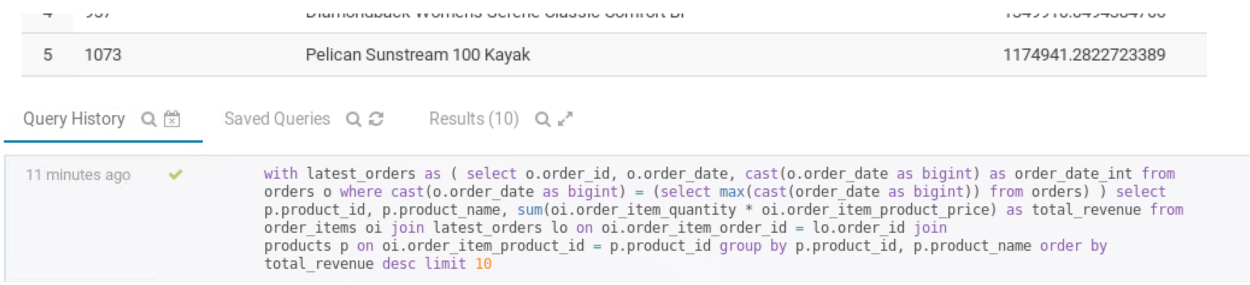


```
    max(cast(order_date as bigint)) as max_order_date
from
    orders;

with latest_orders as (
    select
        o.order_id,
        o.order_date,
        cast(o.order_date as bigint) as order_date_int
    from
        orders o
    where
        cast(o.order_date as bigint) = (select max(cast(order_date as bigint)) from orders)
)
select
    p.product_id,
    p.product_name,
    sum(oi.order_item_quantity * oi.order_item_product_price) as total_revenue
from
    order_items oi
join
    latest_orders lo on oi.order_item_order_id = lo.order_id
```



```
    order_items oi
join
    latest_orders lo on oi.order_item_order_id = lo.order_id
join
    products p on oi.order_item_product_id = p.product_id
group by
    p.product_id, p.product_name
order by
    total_revenue desc
limit 10;
```

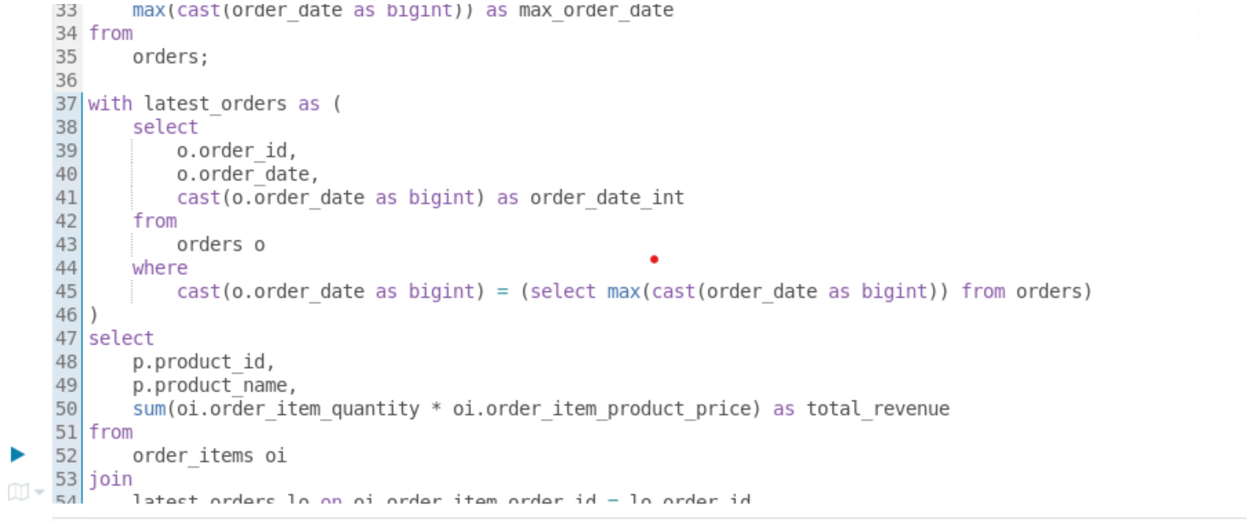


5	1073	Pelican Sunstream 100 Kayak	1174941.2822723389

```
with latest_orders as ( select o.order_id, o.order_date, cast(o.order_date as bigint) as order_date_int from
orders o where cast(o.order_date as bigint) = (select max(cast(order_date as bigint)) from orders) ) select
p.product_id, p.product_name, sum(oi.order_item_quantity * oi.order_item_product_price) as total_revenue from
order_items oi join latest_orders lo on oi.order_item_order_id = lo.order_id join
products p on oi.order_item_product_id = p.product_id group by p.product_id, p.product_name order by
total_revenue desc limit 10
```

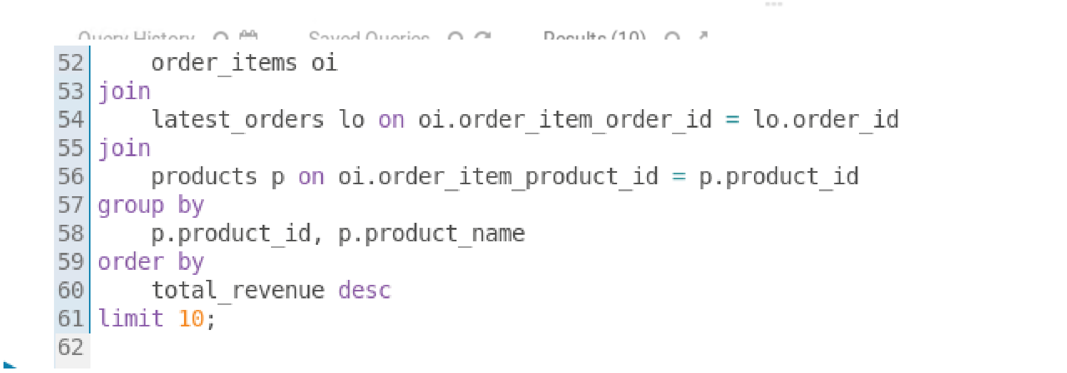
4. How much revenue was generated by the product of your choice not in the top 5?

This query calculates the total revenue generated by a specific product, identified by its product ID, that is not in the top 5 revenue-generating products. It sums the product of order_item_quantity and order_item_product_price for the specified product ID. Despite reading the tutorials for the assignment the code was derived from Open AI to help with this question.

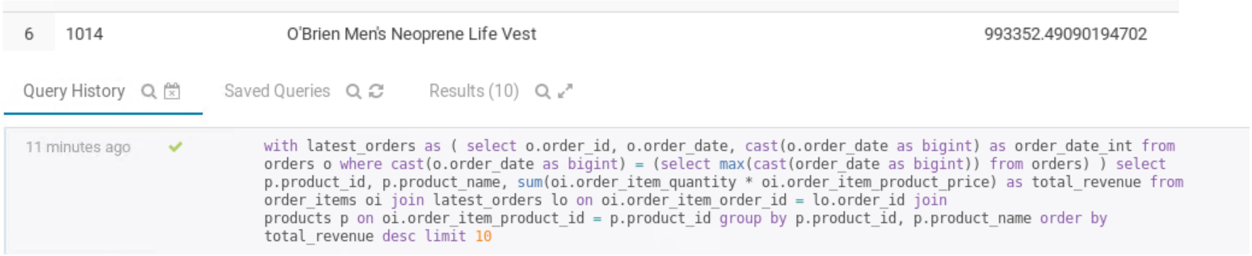


```
    max(cast(order_date as bigint)) as max_order_date
from
    orders;

with latest_orders as (
    select
        o.order_id,
        o.order_date,
        cast(o.order_date as bigint) as order_date_int
    from
        orders o
    where
        cast(o.order_date as bigint) = (select max(cast(order_date as bigint)) from orders)
)
select
    p.product_id,
    p.product_name,
    sum(oi.order_item_quantity * oi.order_item_product_price) as total_revenue
from
    order_items oi
join
    latest_orders lo on oi.order_item_order_id = lo.order_id
```



```
    order_items oi
join
    latest_orders lo on oi.order_item_order_id = lo.order_id
join
    products p on oi.order_item_product_id = p.product_id
group by
    p.product_id, p.product_name
order by
    total_revenue desc
limit 10;
```



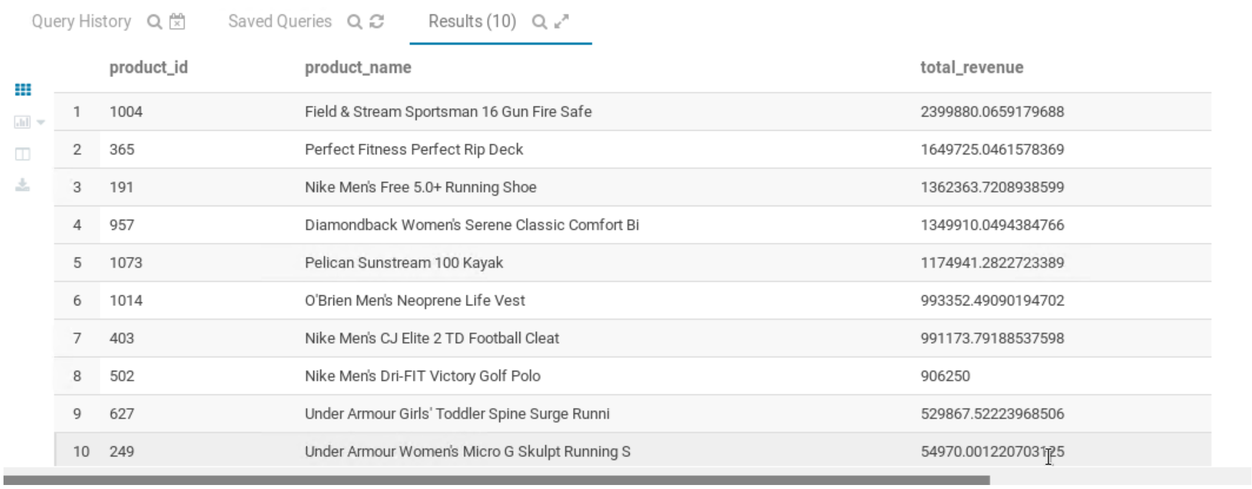
```
6    1014    O'Brien Men's Neoprene Life Vest    993352.49090194702
```

11 minutes ago ✓

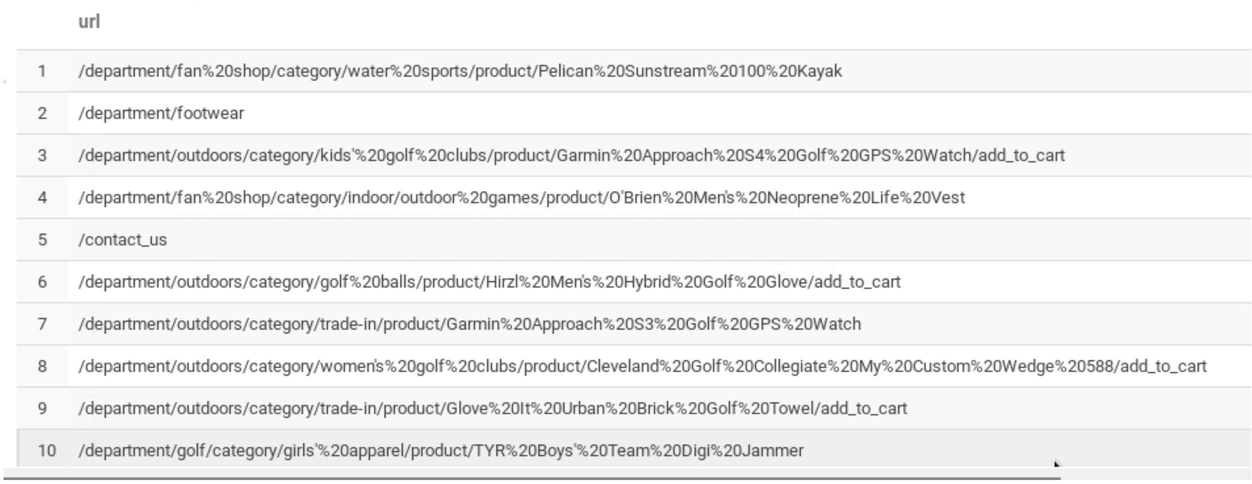
```
with latest_orders as ( select o.order_id, o.order_date, cast(o.order_date as bigint) as order_date_int from
orders o where cast(o.order_date as bigint) = (select max(cast(order_date as bigint)) from orders) ) select
p.product_id, p.product_name, sum(oi.order_item_quantity * oi.order_item_product_price) as total_revenue from
order_items oi join latest_orders lo on oi.order_item_order_id = lo.order_id join
products p on oi.order_item_product_id = p.product_id group by p.product_id, p.product_name order by
total_revenue desc limit 10
```

5. There is one product that did not show up in the previous result. It seems to be viewed a lot, but never purchased. Why?

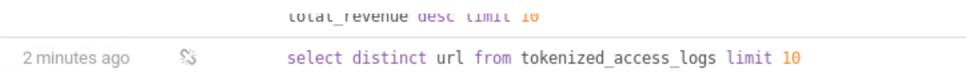
This query identifies products that have been frequently viewed based on URL patterns in the tokenized_access_logs but have never been purchased. It counts the number of views for each product by matching the URL patterns with product IDs. Then, it performs a left join with the purchased products to filter out those that have purchase records. The result shows the most viewed products that have zero purchases, ordered by the number of views. I could not figure out how to solve this problem, but I assume it was viewed a lot because it was attached to another product during the purchasing process and users decided not to buy the products together. Despite reading the tutorials for the assignment the code was derived from Open AI to help with this question.



	product_id	product_name	total_revenue
1	1004	Field & Stream Sportsman 16 Gun Fire Safe	2399880.0659179688
2	365	Perfect Fitness Perfect Rip Deck	1649725.0461578369
3	191	Nike Men's Free 5.0+ Running Shoe	1362363.7208938599
4	957	Diamondback Women's Serene Classic Comfort Bi	1349910.0494384766
5	1073	Pelican Sunstream 100 Kayak	1174941.2822723389
6	1014	O'Brien Men's Neoprene Life Vest	993352.49090194702
7	403	Nike Men's CJ Elite 2 TD Football Cleat	991173.79188537598
8	502	Nike Men's Dri-FIT Victory Golf Polo	906250
9	627	Under Armour Girls' Toddler Spine Surge Runni	529867.52223968506
10	249	Under Armour Women's Micro G Skulpt Running S	54970.001220703125



	url
1	/department/fan%20shop/category/water%20sports/product/Pelican%20Sunstream%20100%20Kayak
2	/department/footwear
3	/department/outdoors/category/kids'%20golf%20clubs/product/Garmin%20Approach%20S4%20Golf%20GPS%20Watch/add_to_cart
4	/department/fan%20shop/category/indoor/outdoor%20games/product/O'Brien%20Men's%20Neoprene%20Life%20Vest
5	/contact_us
6	/department/outdoors/category/golf%20balls/product/Hirzl%20Men's%20Hybrid%20Golf%20Glove/add_to_cart
7	/department/outdoors/category/trade-in/product/Garmin%20Approach%20S3%20Golf%20GPS%20Watch
8	/department/outdoors/category/women's%20golf%20clubs/product/Cleveland%20Golf%20Collegiate%20My%20Custom%20Wedge%20588/add_to_cart
9	/department/outdoors/category/trade-in/product/Glove%20It%20Urban%20Brick%20Golf%20Towel/add_to_cart
10	/department/golf/category/girls'%20apparel/product/TYR%20Boys'%20Team%20Digi%20Jammer



```
                        total_revenue desc limit 10
2 minutes ago           select distinct url from tokenized_access_logs limit 10
```

References

OpenAI. (2024). *ChatGPT* (June 12 version). https://chatgpt.com/