BLINKIT DATA SQL PRESENTATION

AYUSH DHASMANA

INTRODUCTION TO BLINKIT DATA

OVERVIEW

THIS DATASET CONTAINS STRUCTURED DATA ON BLINKIT'S OPERATIONS, INCLUDING ORDERS, CUSTOMERS, PRODUCTS, INVENTORY, MARKETING, AND DELIVERY PERFORMANCE. IT HELPS ANALYZE CUSTOMER BEHAVIOR, SALES TRENDS, AND OPERATIONAL EFFICIENCY.

OBJECTIVE

THE GOAL IS TO EXTRACT INSIGHTS FOR BETTER DECISION-MAKING IN:

- *CUSTOMER PURCHASING PATTERNS*
- *MARKETING CAMPAIGN EFFECTIVENESS*
- *INVENTORY AND SUPPLY CHAIN MANAGEMENT*
- *DELIVERY PERFORMANCE OPTIMIZATION*

KEY COMPONENTS

- *ORDERS & CUSTOMERS: PURCHASE HISTORY AND DEMOGRAPHICS*
- *PRODUCTS & INVENTORY: STOCK LEVELS AND AVAILABILITY*
- *MARKETING & DELIVERY: CAMPAIGN IMPACT AND LOGISTICS PERFORMANCE*
- *CUSTOMER FEEDBACK: REVIEWS AND SATISFACTION RATINGS*

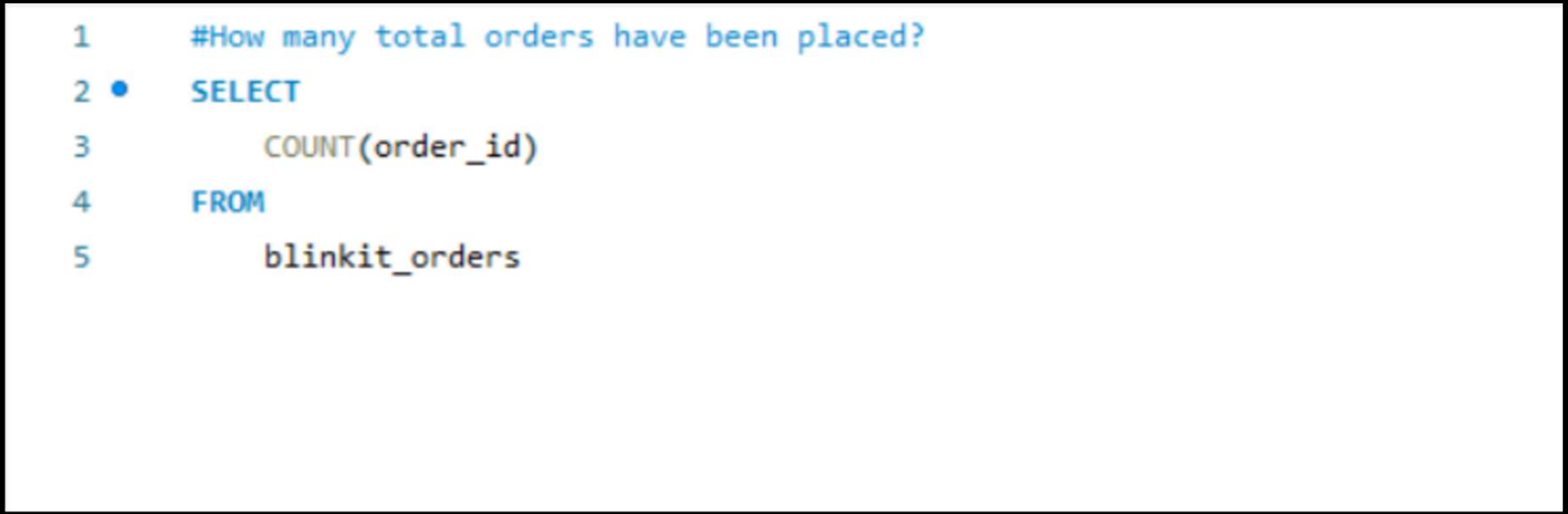
USE CASES

- *SALES FORECASTING*
- *CUSTOMER RETENTION STRATEGIES*
- *LOGISTICS OPTIMIZATION*
- *INVENTORY MANAGEMENT*

THIS ANALYSIS WILL PROVIDE DATA-DRIVEN RECOMMENDATIONS TO IMPROVE BLINKIT'S OPERATIONS.

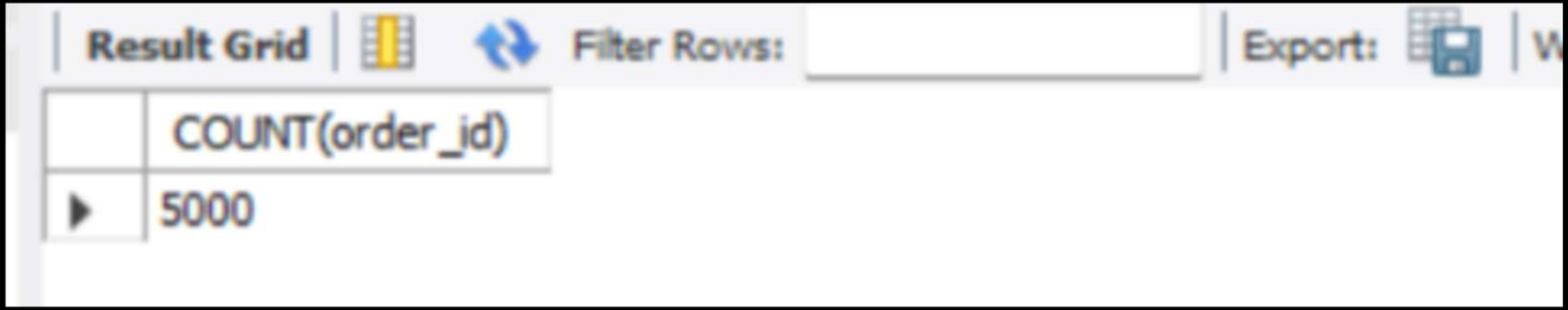
HOW MANY TOTAL ORDERS HAVE BEEN PLACED?

QUERY-:



```sql
#How many total orders have been placed?
SELECT
    COUNT(order_id)
FROM
    blinkit_orders
```

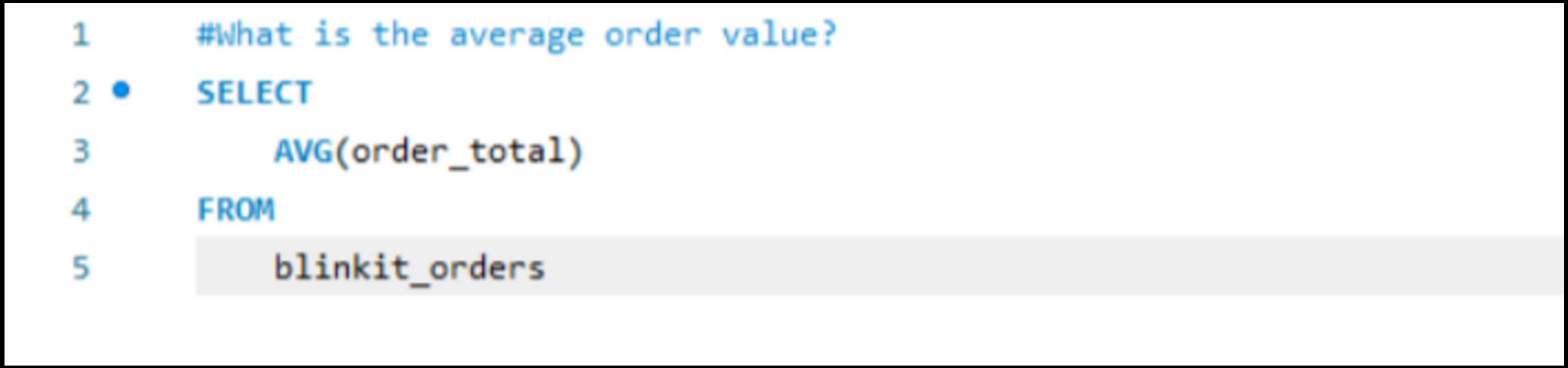
RESULT-:



COUNT(order_id)
5000

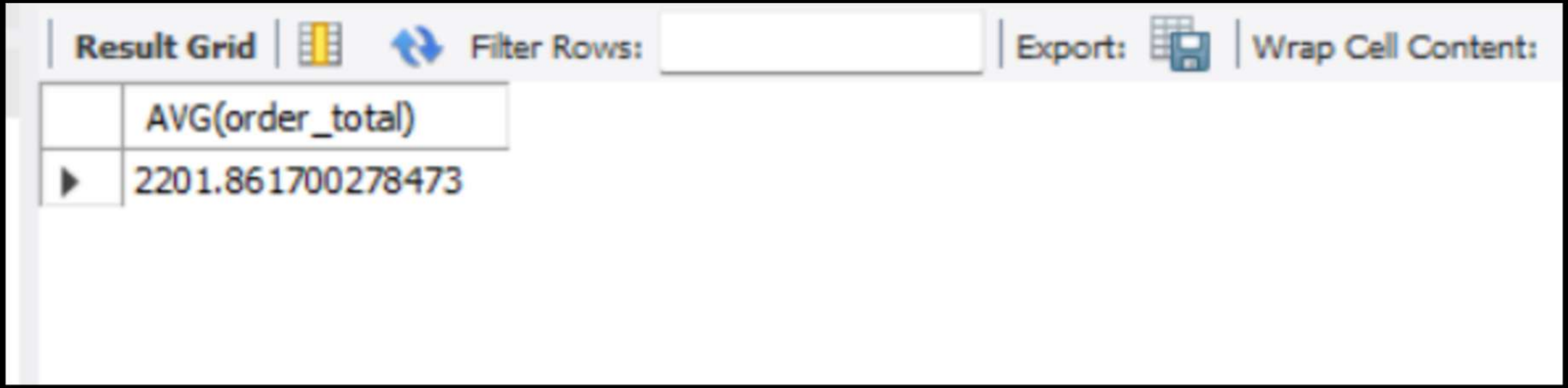
WHAT IS THE AVERAGE ORDER VALUE?

QUERY-:



```
#What is the average order value?
SELECT
    AVG(order_total)
FROM
    blinkit_orders
```

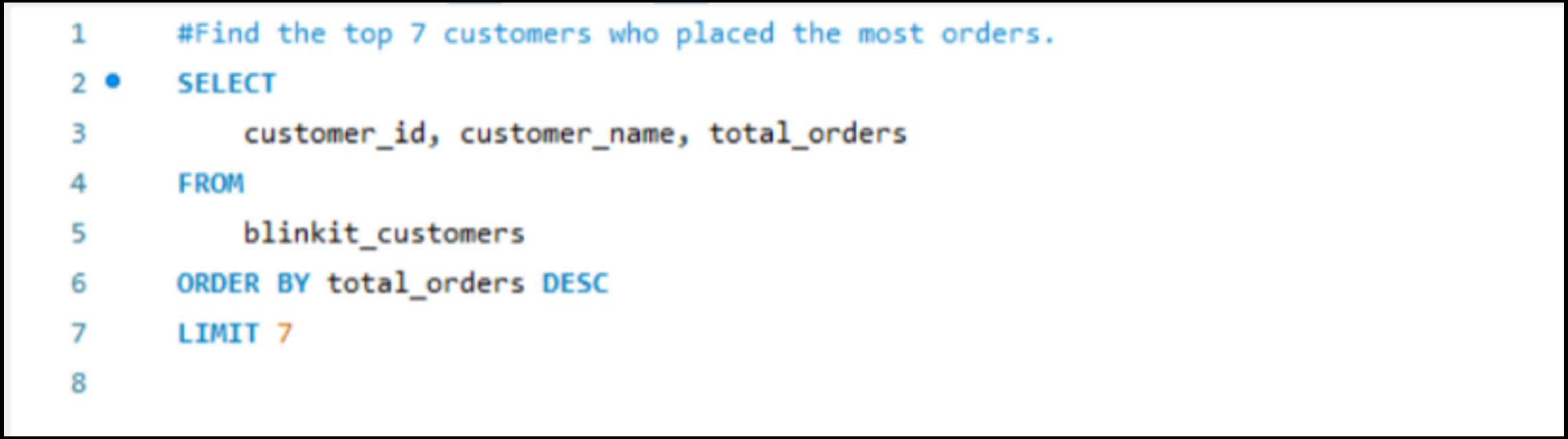
RESULT-:



```
AVG(order_total)
2201.861700278473
```

FIND THE TOP 7 CUSTOMERS WHO PLACED THE MOST ORDERS.

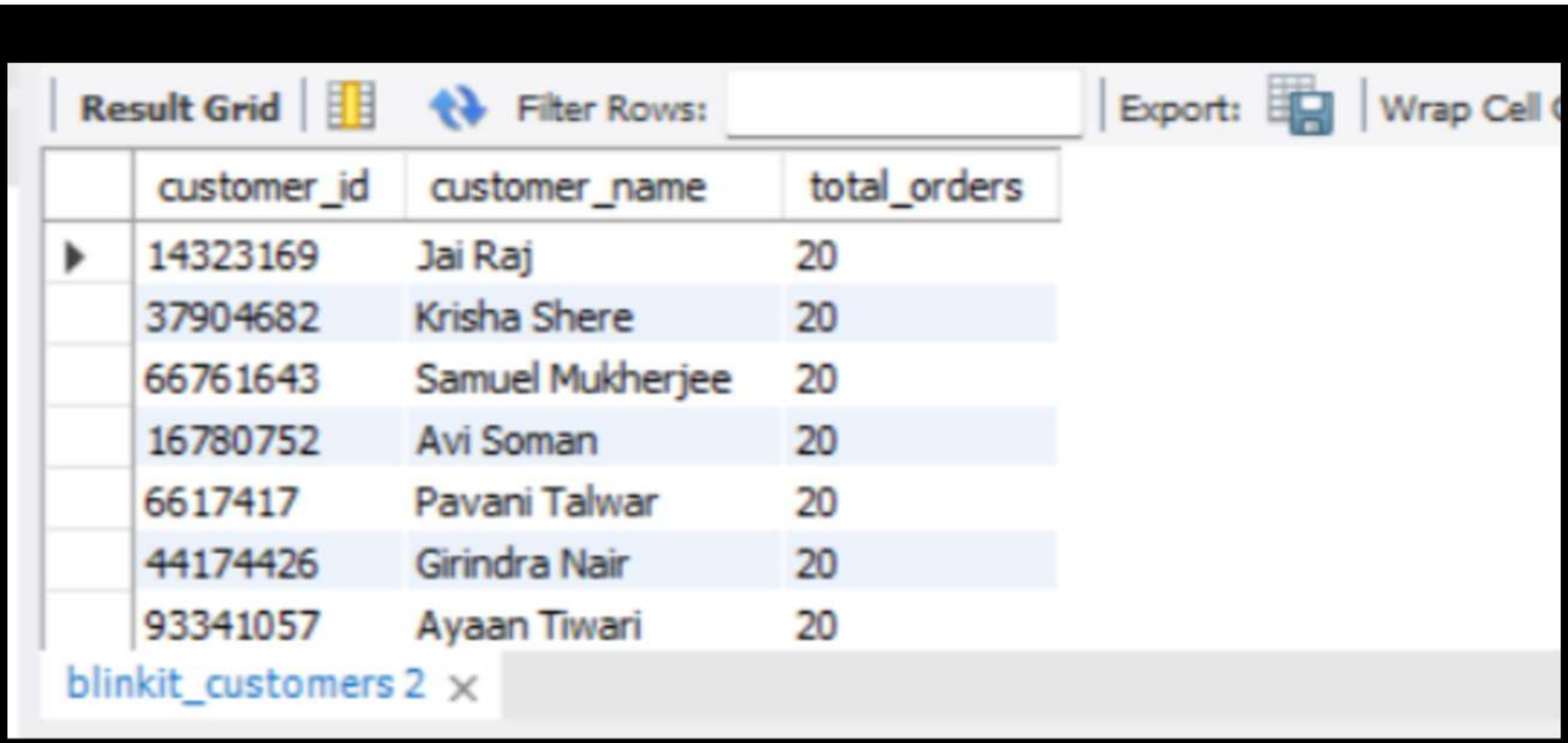
QUERY-:



```sql
#Find the top 7 customers who placed the most orders.
SELECT
    customer_id, customer_name, total_orders
FROM
    blinkit_customers
ORDER BY total_orders DESC
LIMIT 7

```

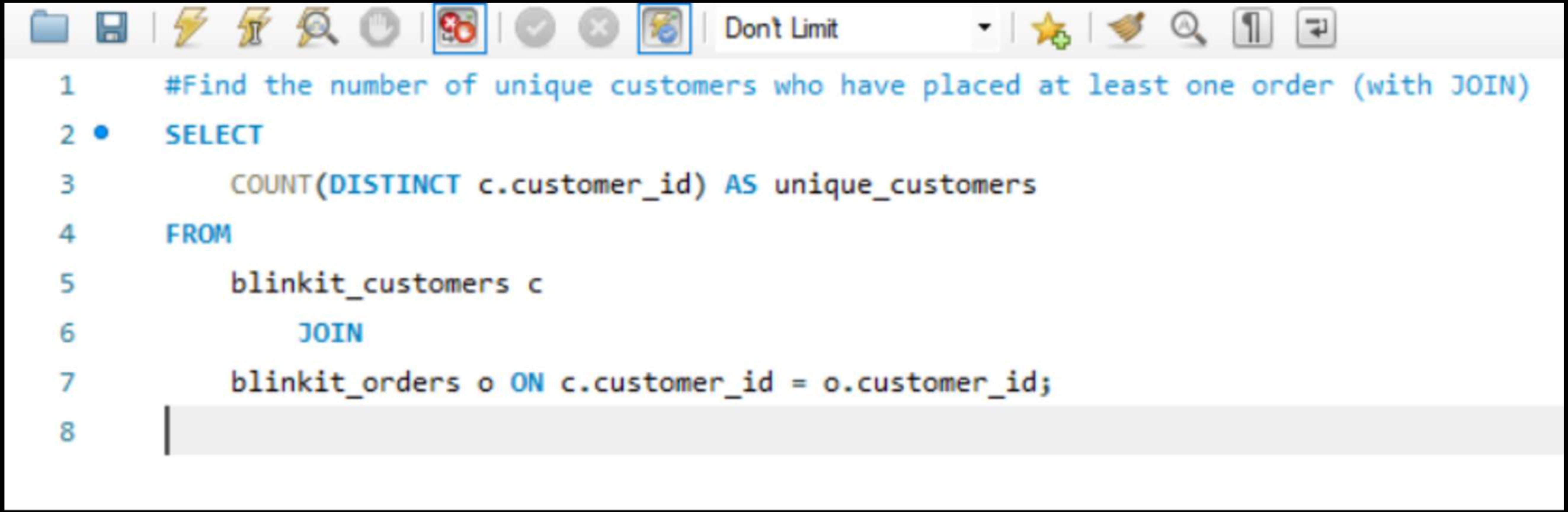
RESULT-:



customer_id	customer_name	total_orders
14323169	Jai Raj	20
37904682	Krisha Shere	20
66761643	Samuel Mukherjee	20
16780752	Avi Soman	20
6617417	Pavani Talwar	20
44174426	Girindra Nair	20
93341057	Ayaan Tiwari	20

FIND THE NUMBER OF UNIQUE CUSTOMERS WHO HAVE PLACED AT LEAST ONE ORDER (WITH JOIN)

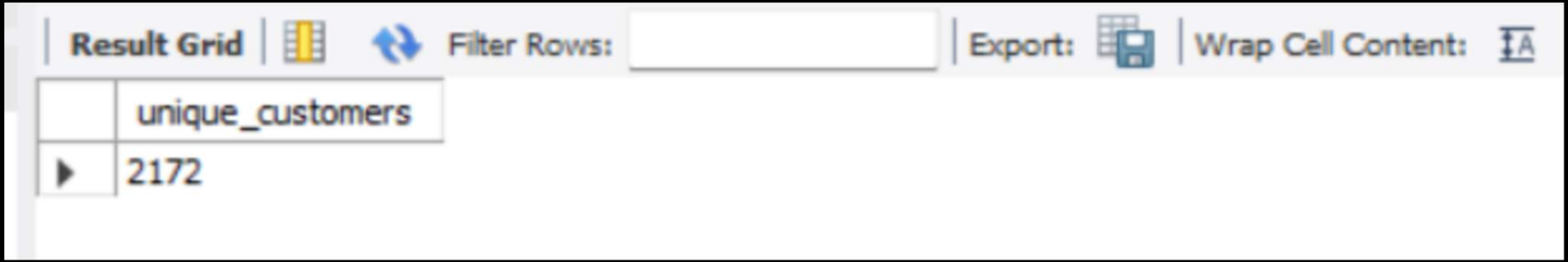
QUERY-:



```
#Find the number of unique customers who have placed at least one order (with JOIN)
SELECT
    COUNT(DISTINCT c.customer_id) AS unique_customers
FROM
    blinkit_customers c
        JOIN
    blinkit_orders o ON c.customer_id = o.customer_id;

```

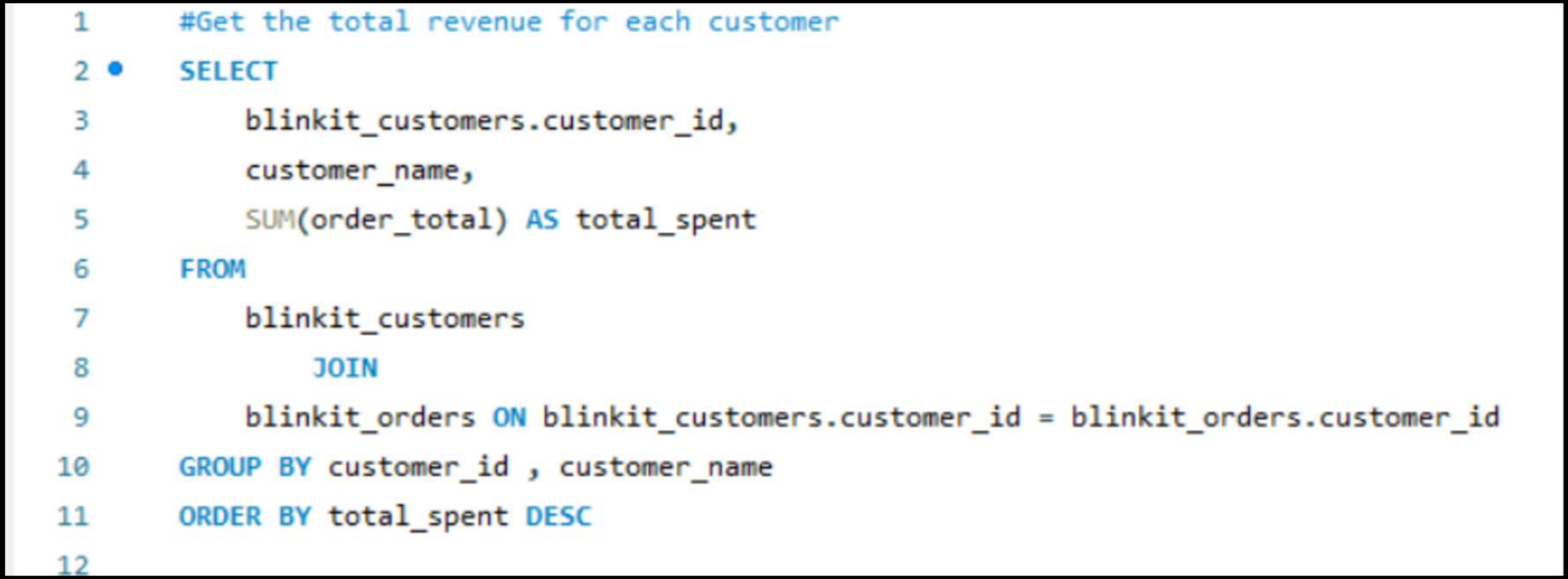
RESULT-:



unique_customers
2172

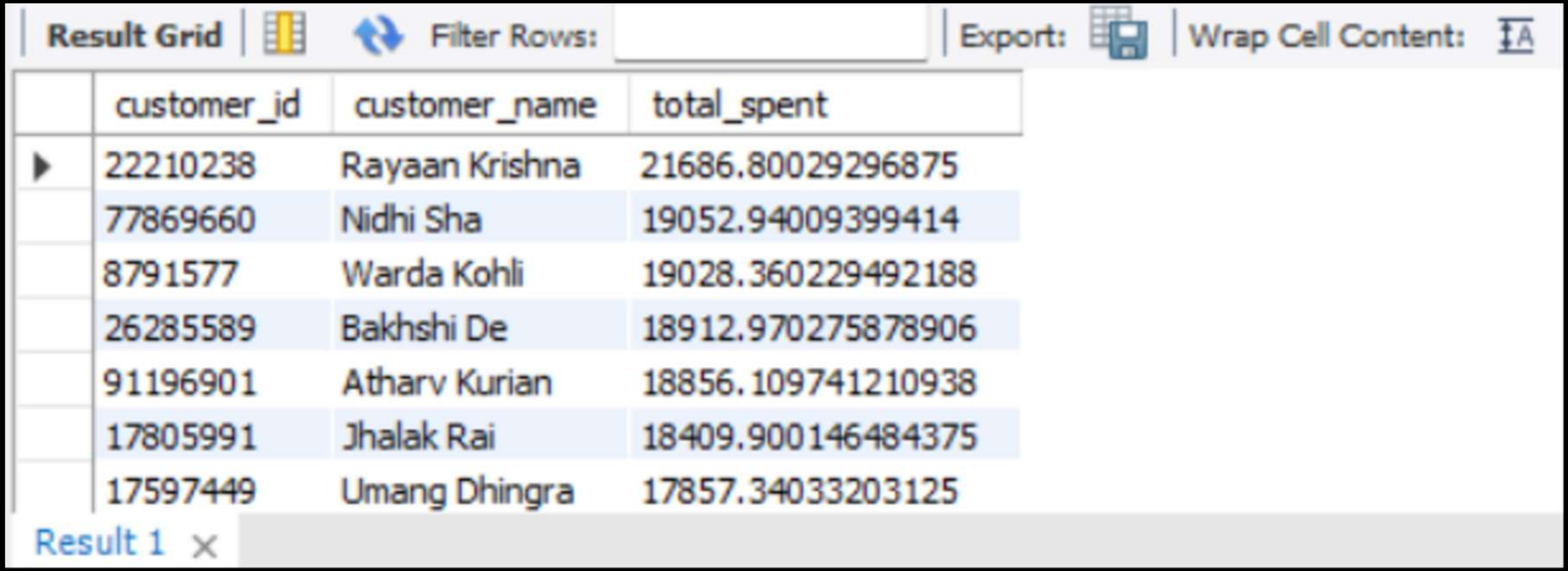
GET THE TOTAL REVENUE FOR EACH CUSTOMER

QUERY-:



```
#Get the total revenue for each customer
SELECT
    blinkit_customers.customer_id,
    customer_name,
    SUM(order_total) AS total_spent
FROM
    blinkit_customers
        JOIN
    blinkit_orders ON blinkit_customers.customer_id = blinkit_orders.customer_id
GROUP BY customer_id , customer_name
ORDER BY total_spent DESC
```

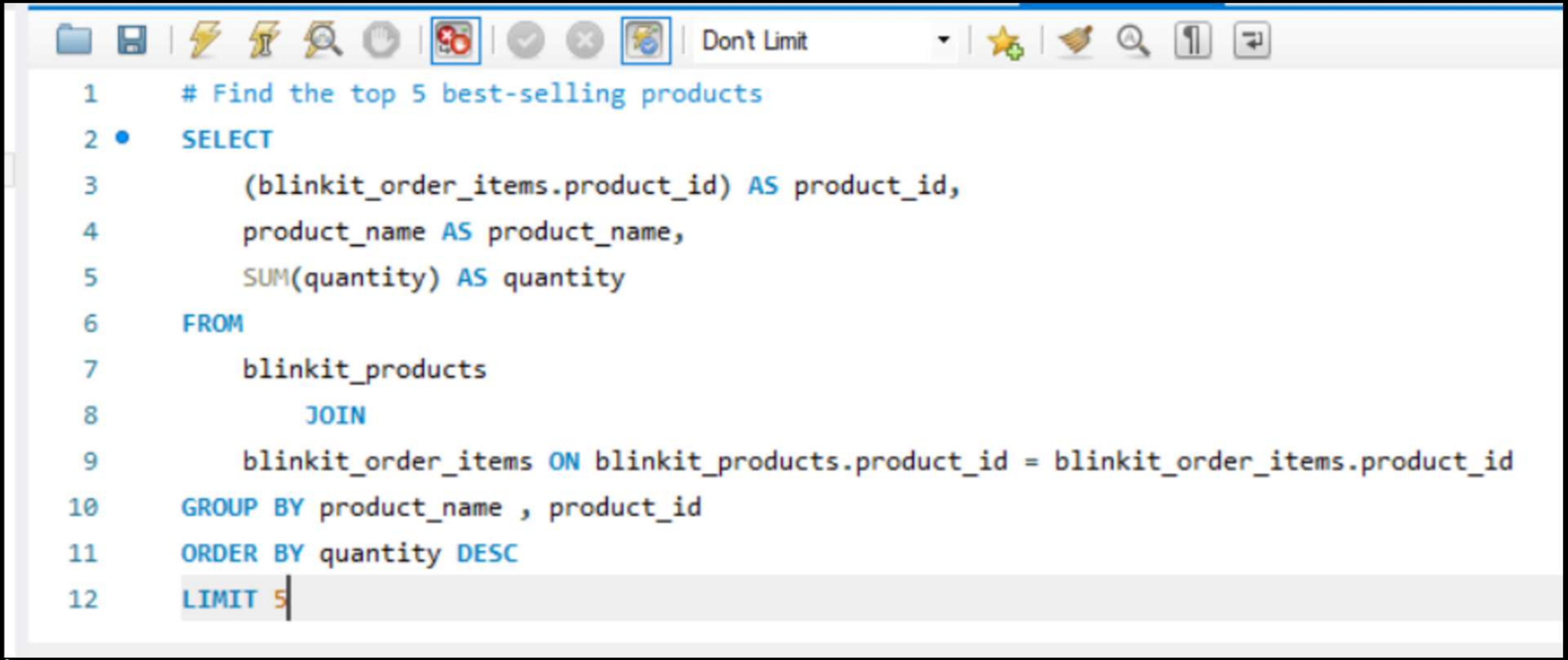
RESULT-:



customer_id	customer_name	total_spent
22210238	Rayaan Krishna	21686.80029296875
77869660	Nidhi Sha	19052.94009399414
8791577	Warda Kohli	19028.360229492188
26285589	Bakhshi De	18912.970275878906
91196901	Atharv Kurian	18856.109741210938
17805991	Jhalak Rai	18409.900146484375
17597449	Umang Dhingra	17857.34033203125

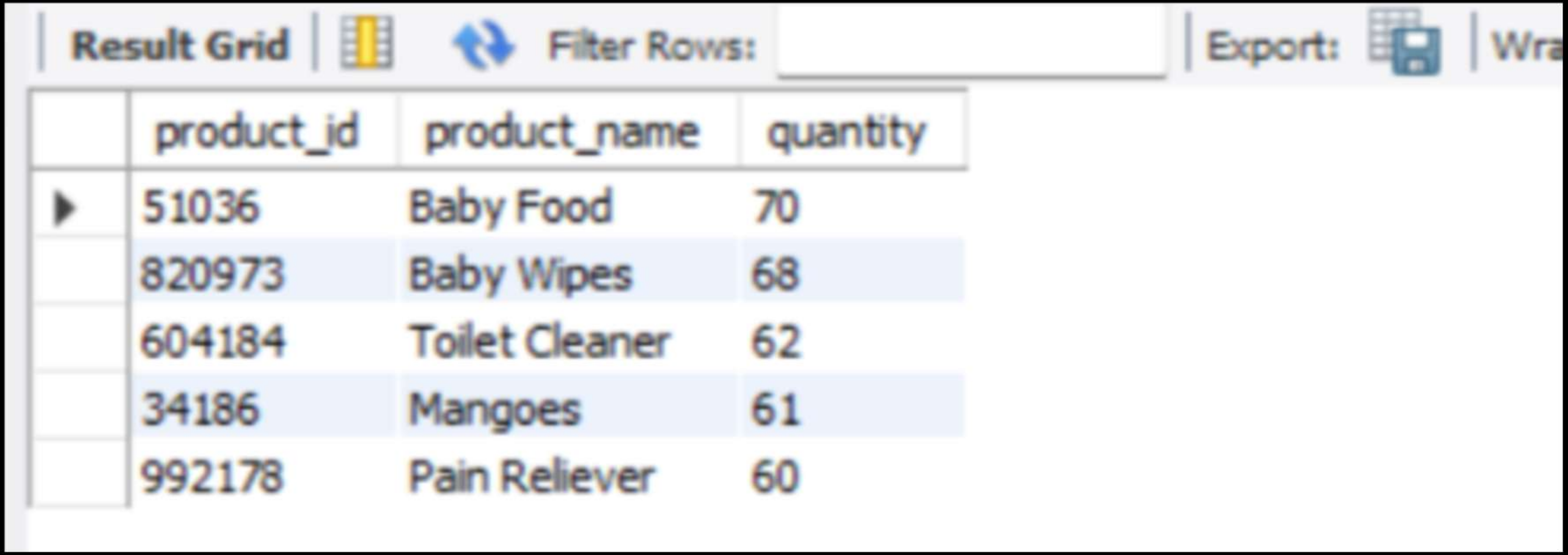
FIND THE TOP 5 BEST-SELLING PRODUCTS

QUERY-:



```sql
# Find the top 5 best-selling products
SELECT
    (blinkit_order_items.product_id) AS product_id,
    product_name AS product_name,
    SUM(quantity) AS quantity
FROM
    blinkit_products
        JOIN
    blinkit_order_items ON blinkit_products.product_id = blinkit_order_items.product_id
GROUP BY product_name , product_id
ORDER BY quantity DESC
LIMIT 5
```

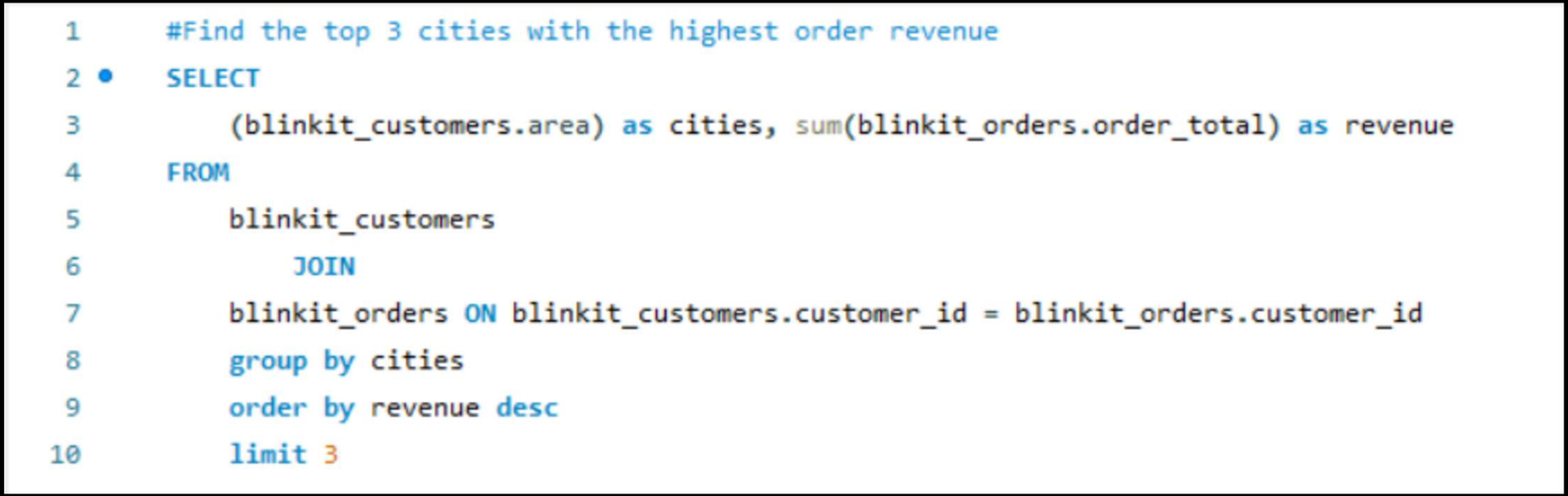
RESULT-:



product_id	product_name	quantity
51036	Baby Food	70
820973	Baby Wipes	68
604184	Toilet Cleaner	62
34186	Mangoes	61
992178	Pain Reliever	60

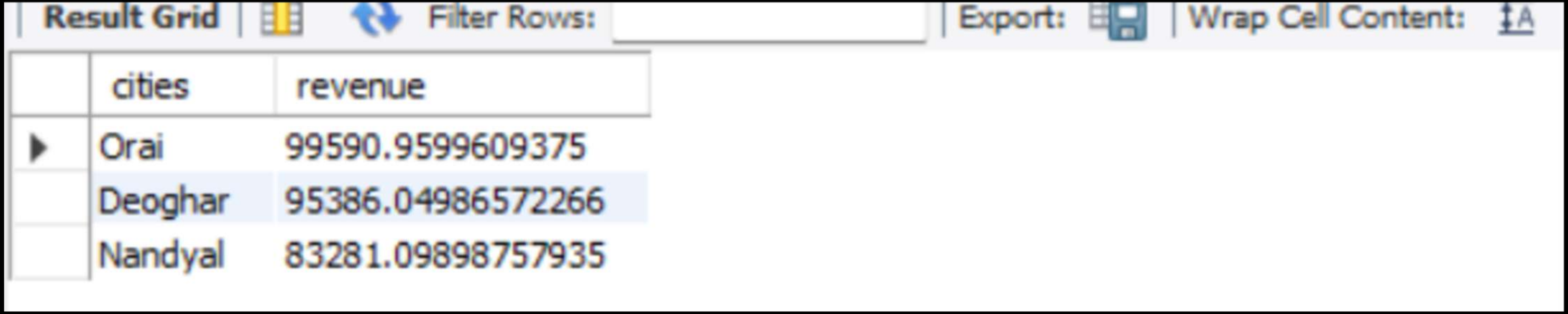
FIND THE TOP 3 CITIES WITH THE HIGHEST ORDER REVENUE

QUERY-:



```
#Find the top 3 cities with the highest order revenue
SELECT
    (blinkit_customers.area) as cities, sum(blinkit_orders.order_total) as revenue
FROM
    blinkit_customers
        JOIN
    blinkit_orders ON blinkit_customers.customer_id = blinkit_orders.customer_id
    group by cities
    order by revenue desc
    limit 3
```

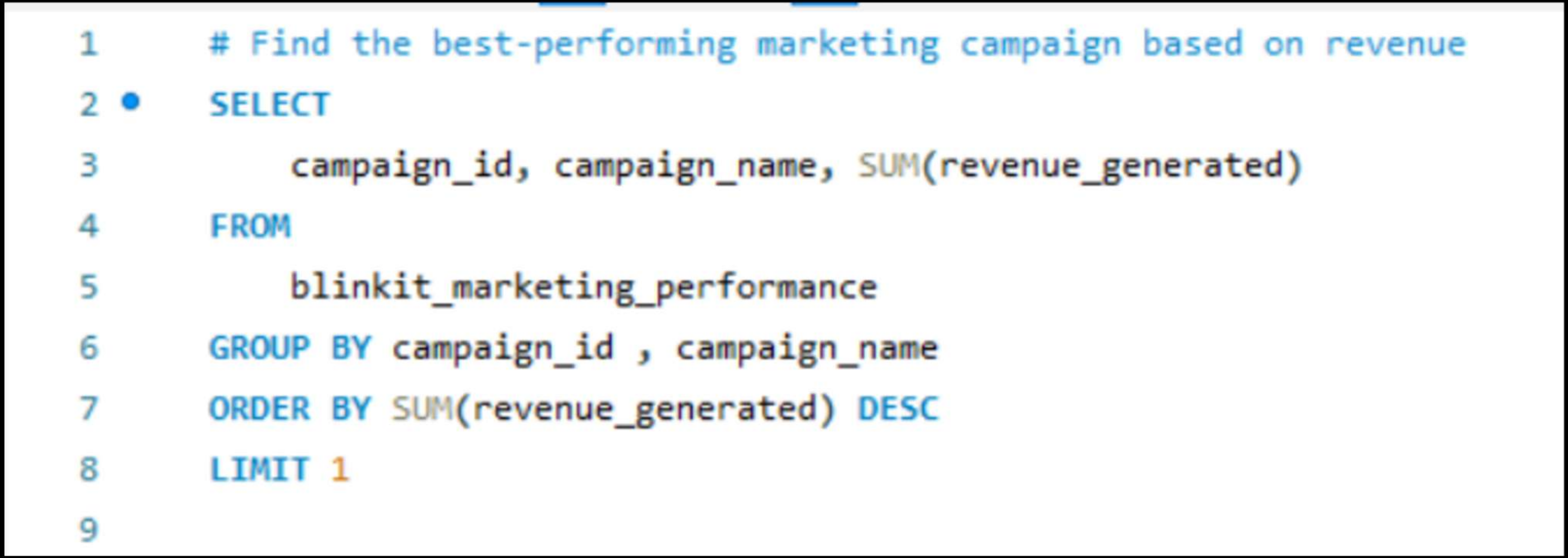
RESULT-:



cities	revenue
Orai	99590.9599609375
Deoghar	95386.04986572266
Nandyal	83281.09898757935

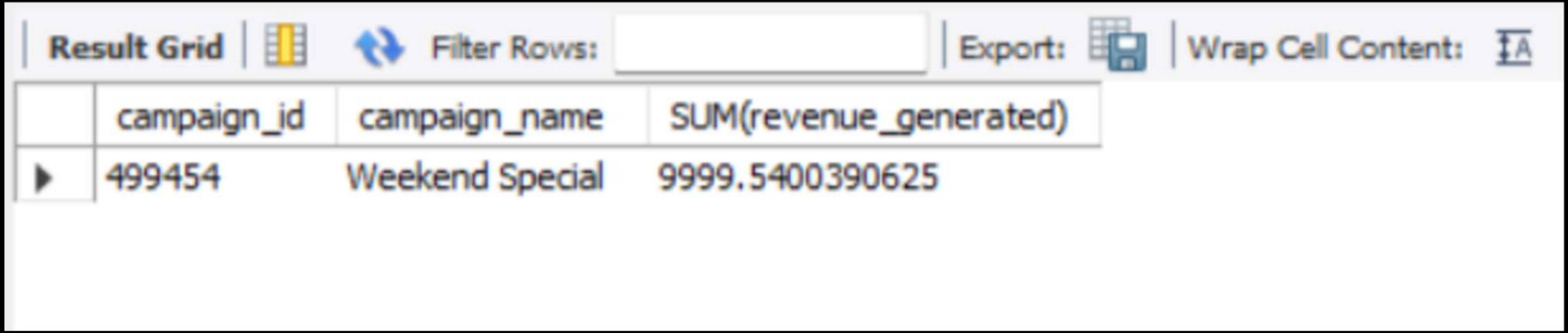
FIND THE BEST-PERFORMING MARKETING CAMPAIGN BASED ON REVENUE

QUERY-:



```sql
# Find the best-performing marketing campaign based on revenue
SELECT
    campaign_id, campaign_name, SUM(revenue_generated)
FROM
    blinkit_marketing_performance
GROUP BY campaign_id , campaign_name
ORDER BY SUM(revenue_generated) DESC
LIMIT 1
```

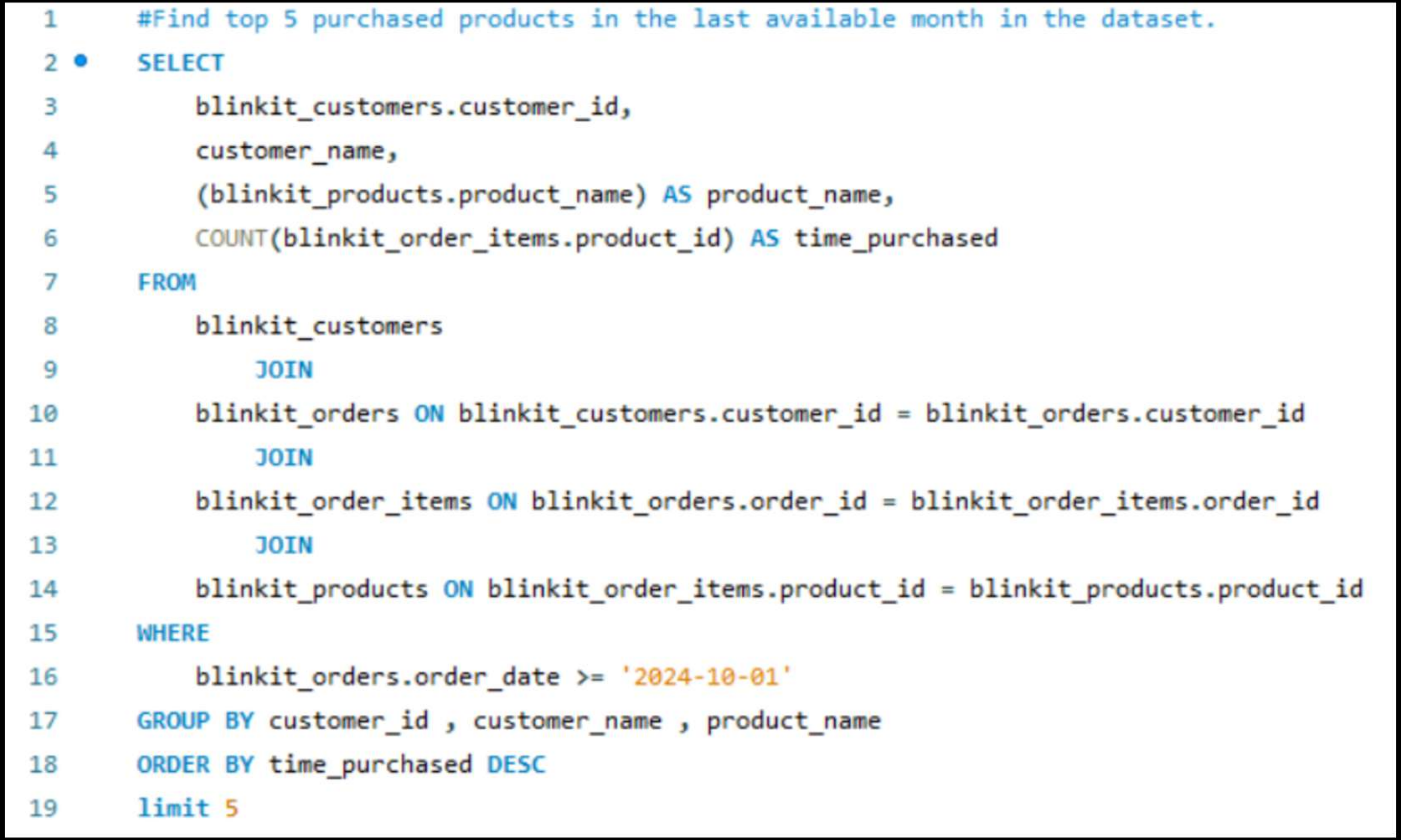
RESULT-:



```
campaign_id   campaign_name     SUM(revenue_generated)
499454        Weekend Special   9999.5400390625
```

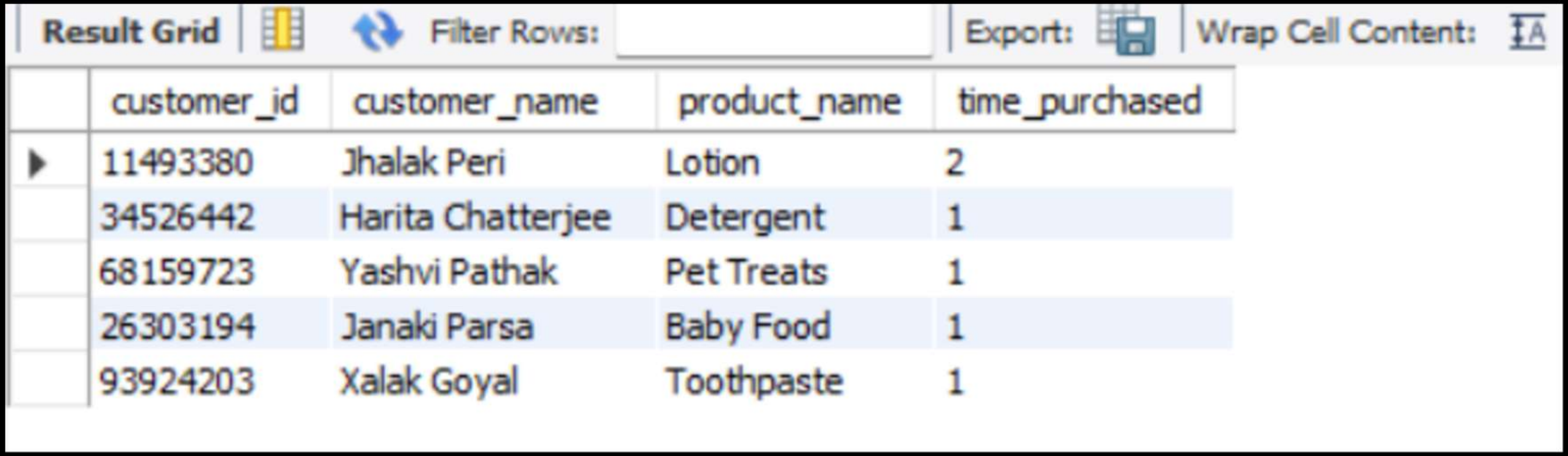
FIND TOP 5 PURCHASED PRODUCTS IN THE LAST AVAILABLE MONTH IN THE DATASET.

QUERY-:



```sql
#Find top 5 purchased products in the last available month in the dataset.
SELECT
    blinkit_customers.customer_id,
    customer_name,
    (blinkit_products.product_name) AS product_name,
    COUNT(blinkit_order_items.product_id) AS time_purchased
FROM
    blinkit_customers
        JOIN
    blinkit_orders ON blinkit_customers.customer_id = blinkit_orders.customer_id
        JOIN
    blinkit_order_items ON blinkit_orders.order_id = blinkit_order_items.order_id
        JOIN
    blinkit_products ON blinkit_order_items.product_id = blinkit_products.product_id
WHERE
    blinkit_orders.order_date >= '2024-10-01'
GROUP BY customer_id , customer_name , product_name
ORDER BY time_purchased DESC
limit 5
```

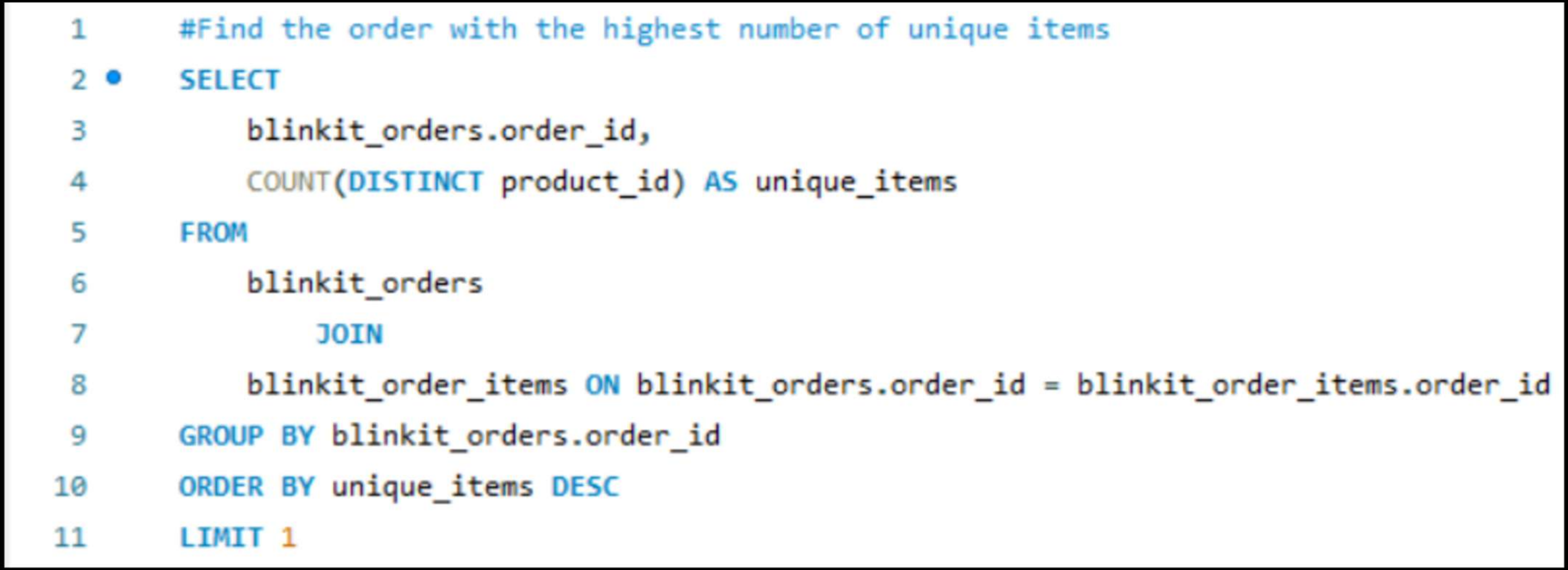
RESULT-:



customer_id	customer_name	product_name	time_purchased
11493380	Jhalak Peri	Lotion	2
34526442	Harita Chatterjee	Detergent	1
68159723	Yashvi Pathak	Pet Treats	1
26303194	Janaki Parsa	Baby Food	1
93924203	Xalak Goyal	Toothpaste	1

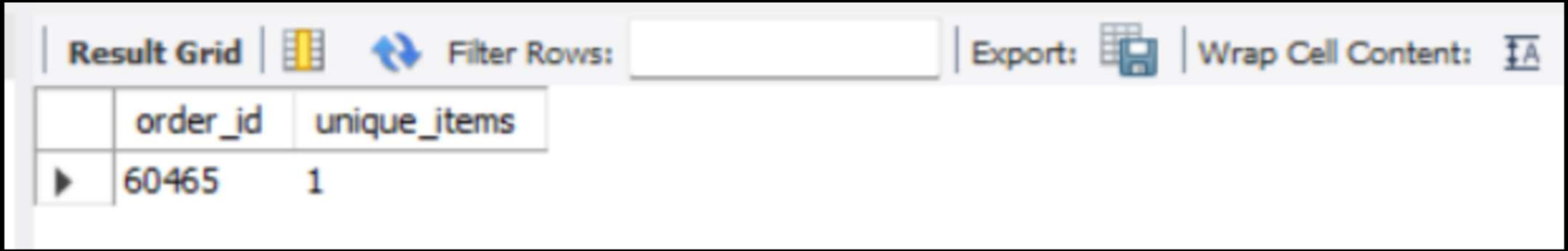
FIND THE ORDER WITH THE HIGHEST NUMBER OF UNIQUE ITEMS

QUERY-:



```sql
#Find the order with the highest number of unique items
SELECT
    blinkit_orders.order_id,
    COUNT(DISTINCT product_id) AS unique_items
FROM
    blinkit_orders
        JOIN
    blinkit_order_items ON blinkit_orders.order_id = blinkit_order_items.order_id
GROUP BY blinkit_orders.order_id
ORDER BY unique_items DESC
LIMIT 1
```

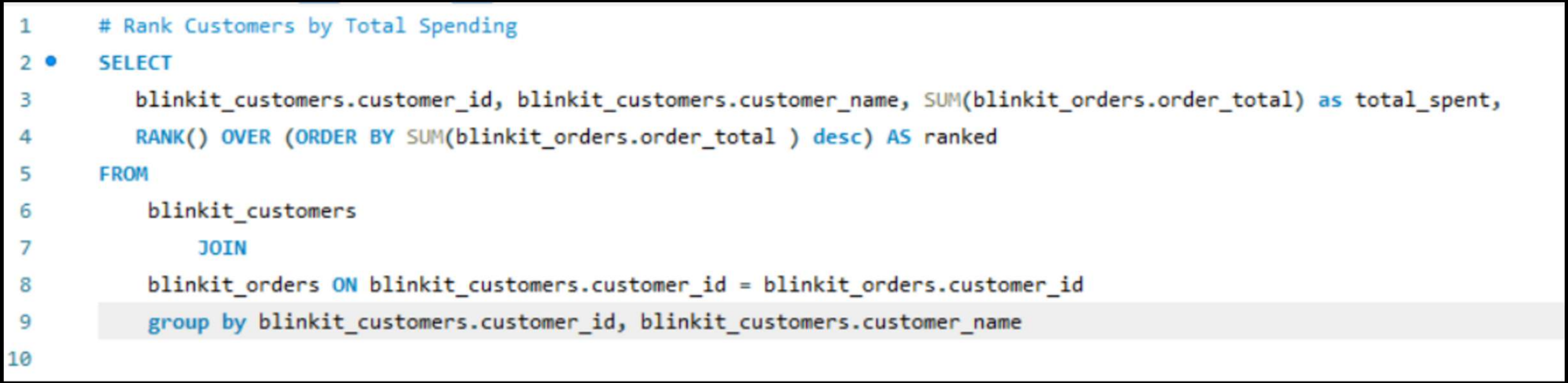
RESULT-:



order_id	unique_items
60465	1

RANK CUSTOMERS BY TOTAL SPENDING

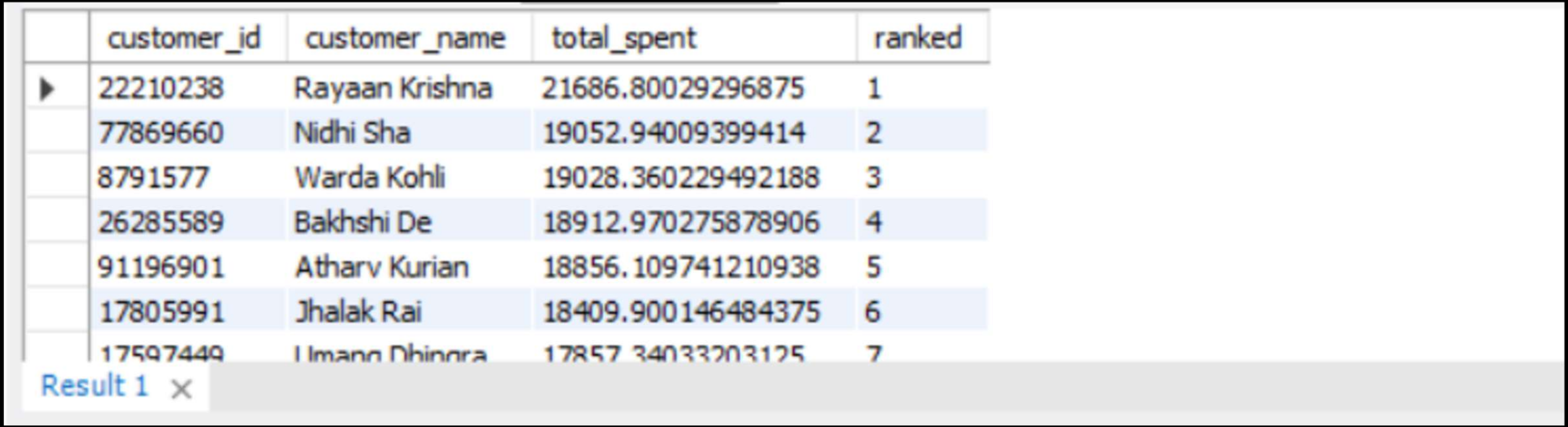
QUERY-:



```
# Rank Customers by Total Spending
SELECT
   blinkit_customers.customer_id, blinkit_customers.customer_name, SUM(blinkit_orders.order_total) as total_spent,
   RANK() OVER (ORDER BY SUM(blinkit_orders.order_total ) desc) AS ranked
FROM
    blinkit_customers
        JOIN
    blinkit_orders ON blinkit_customers.customer_id = blinkit_orders.customer_id
    group by blinkit_customers.customer_id, blinkit_customers.customer_name

```

RESULT-:



customer_id	customer_name	total_spent	ranked
22210238	Rayaan Krishna	21686.80029296875	1
77869660	Nidhi Sha	19052.94009399414	2
8791577	Warda Kohli	19028.360229492188	3
26285589	Bakhshi De	18912.970275878906	4
91196901	Atharv Kurian	18856.109741210938	5
17805991	Jhalak Rai	18409.900146484375	6
17597449	Umang Dhingra	17857.34033203125	7