E-Commerce (Target) Sales Dataset

Presented by: MAYANK GOYAL

Dataset Overview

Leveraged a real-world eCommerce dataset to perform structured SQL-based analysis on sales performance, customer trends, order behavior, and revenue insights.

Tables Used:

order_item

orders

products

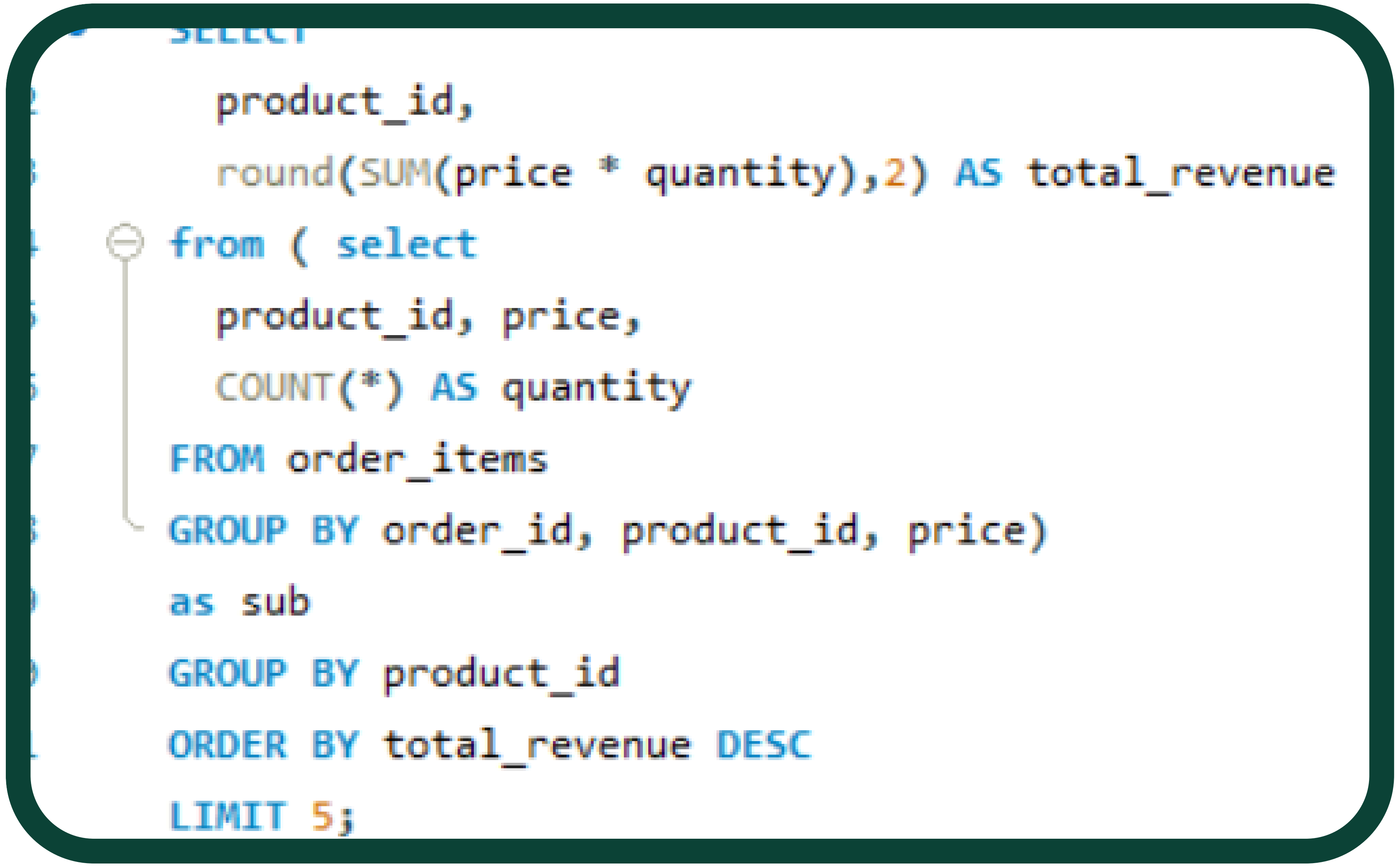
customers

geolocation

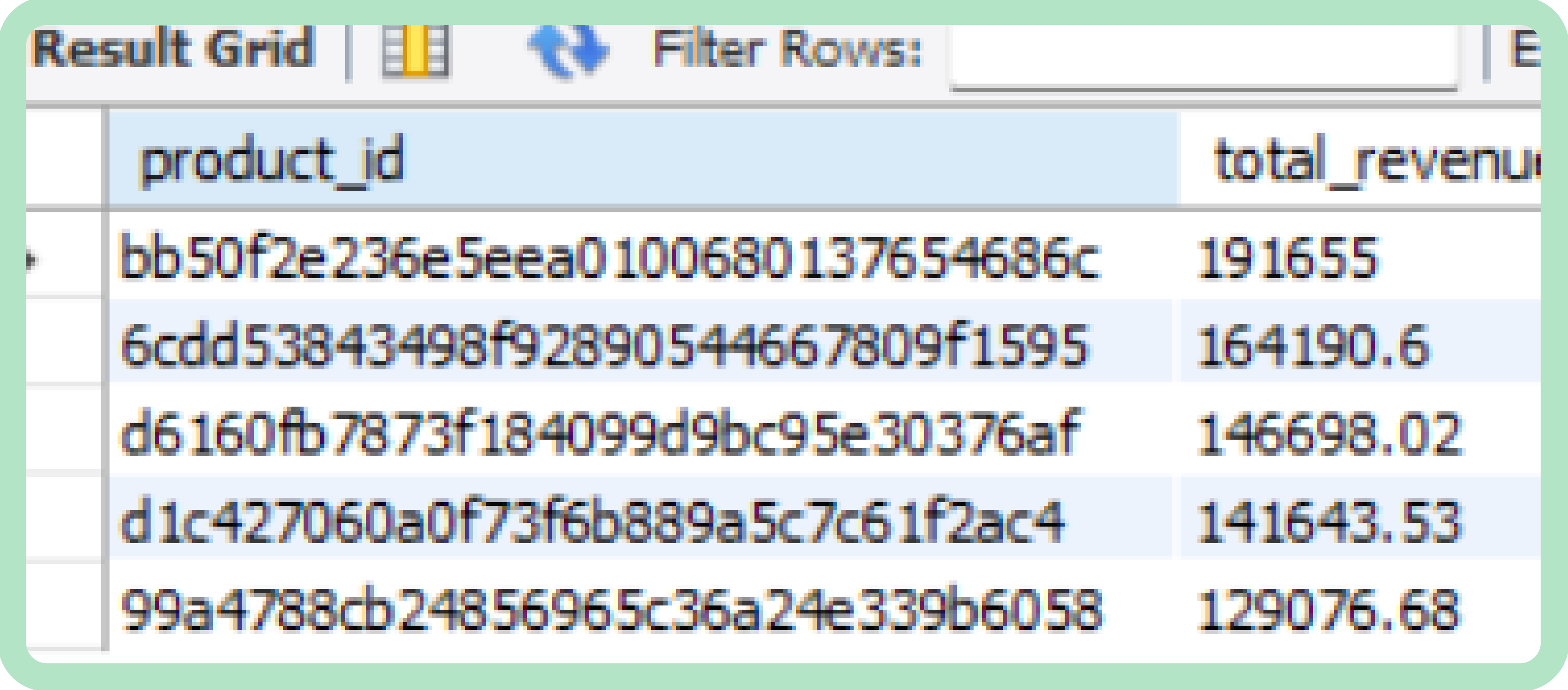
sellers

order_payments

Identify top 5 products that made the most revenue.

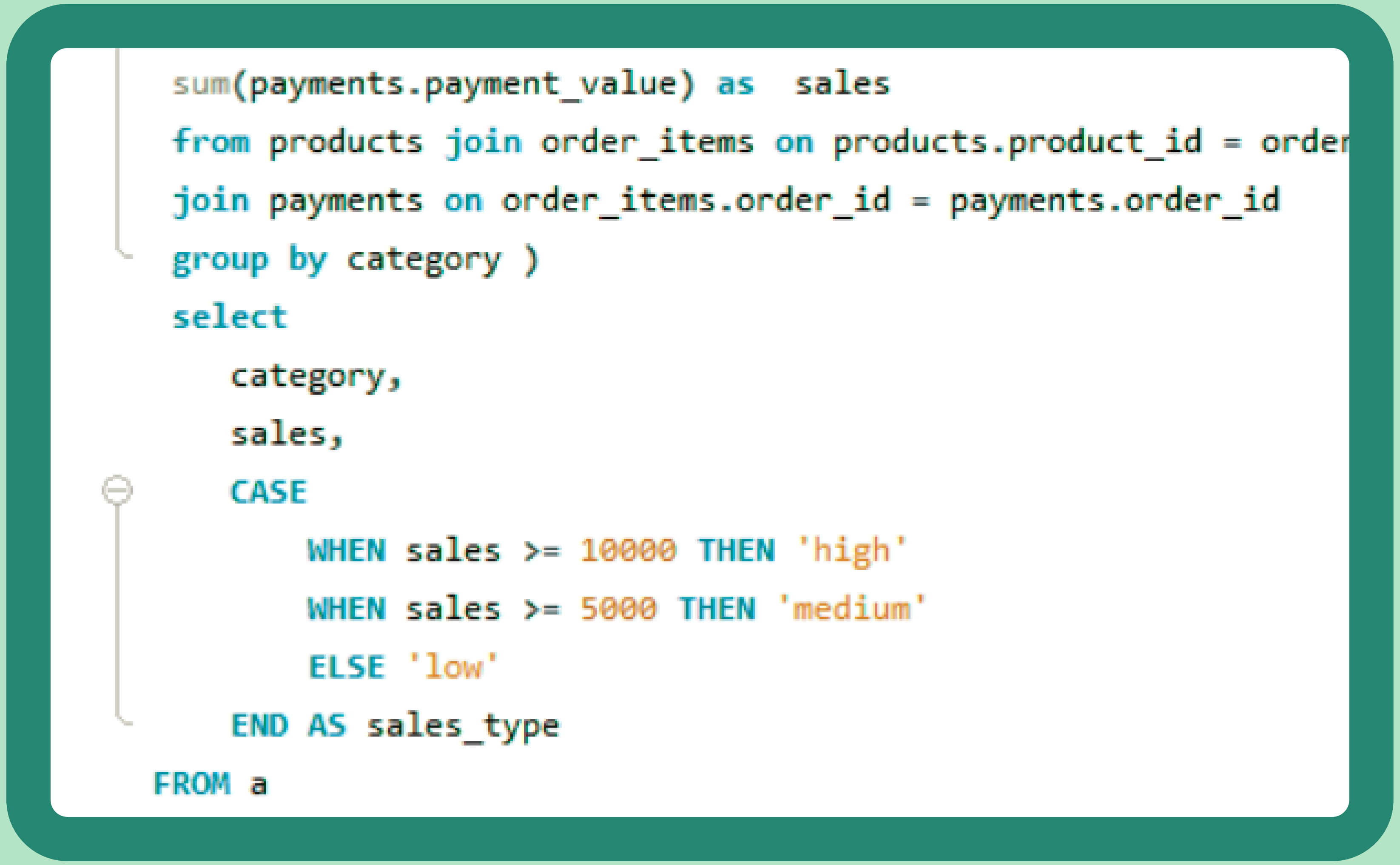


```
SELECT
    product_id,
    round(SUM(price * quantity),2) AS total_revenue
from ( select
    product_id, price,
    COUNT(*) AS quantity
FROM order_items
GROUP BY order_id, product_id, price)
as sub
GROUP BY product_id
ORDER BY total_revenue DESC
LIMIT 5;
```

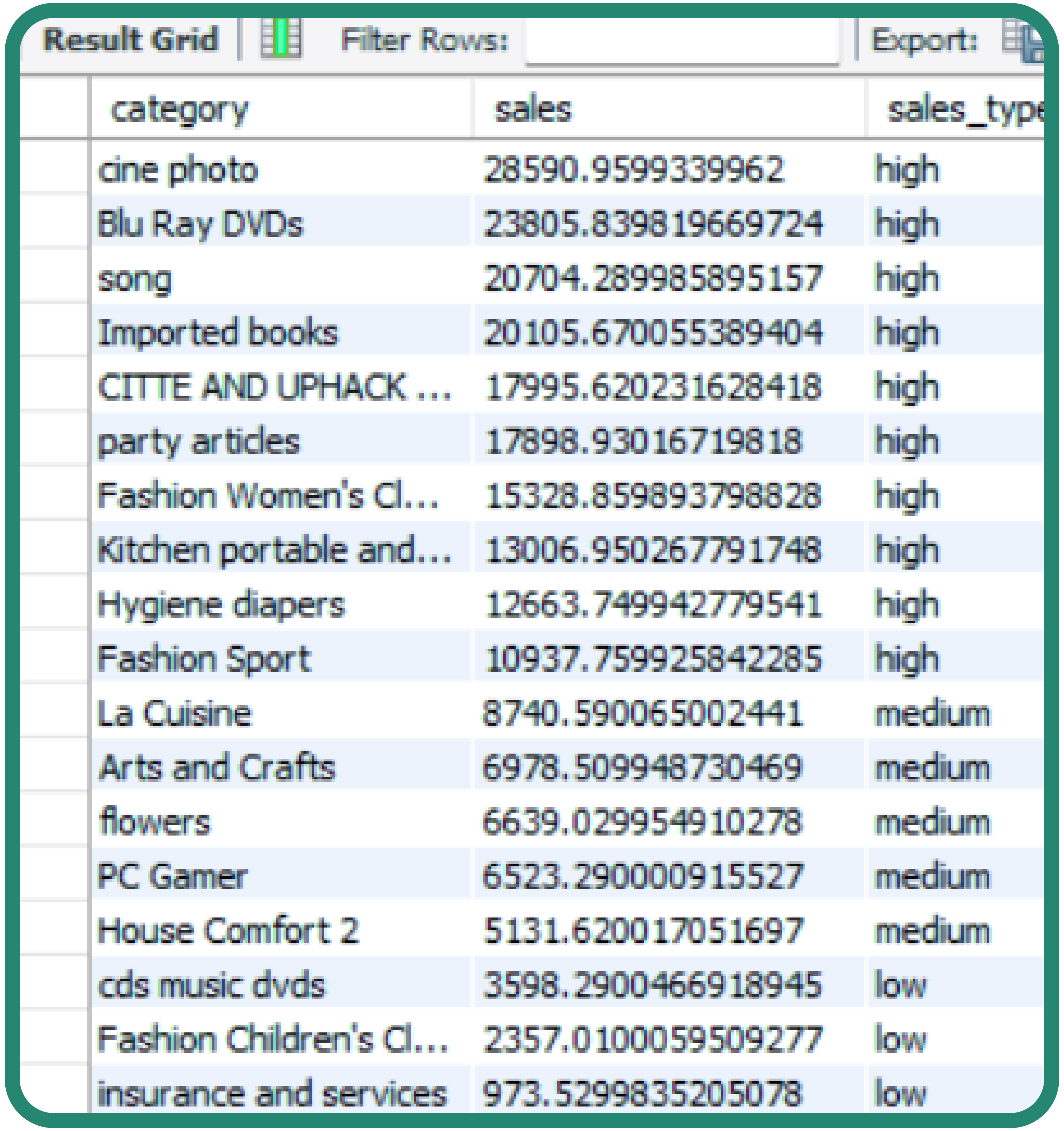


product_id	total_revenu
bb50f2e236e5eea0100680137654686c	191655
6cdd53843498f92890544667809f1595	164190.6
d6160fb7873f184099d9bc95e30376af	146698.02
d1c427060a0f73f6b889a5c7c61f2ac4	141643.53
99a4788cb24856965c36a24e339b6058	129076.68

Classify Product Categories by Sales Volume using Conditional Logic in SQL.

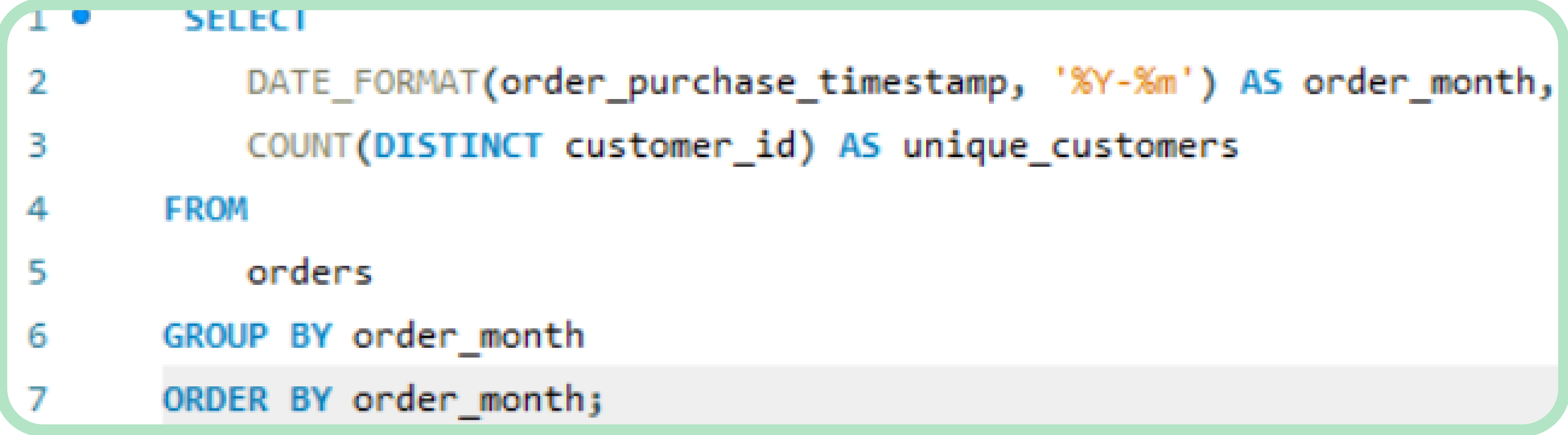


```
    sum(payments.payment_value) as  sales
    from products join order_items on products.product_id = order
    join payments on order_items.order_id = payments.order_id
    group by category )
    select
       category,
       sales,
       CASE
           WHEN sales >= 10000 THEN 'high'
           WHEN sales >= 5000 THEN 'medium'
           ELSE 'low'
       END AS sales_type
   FROM a
```

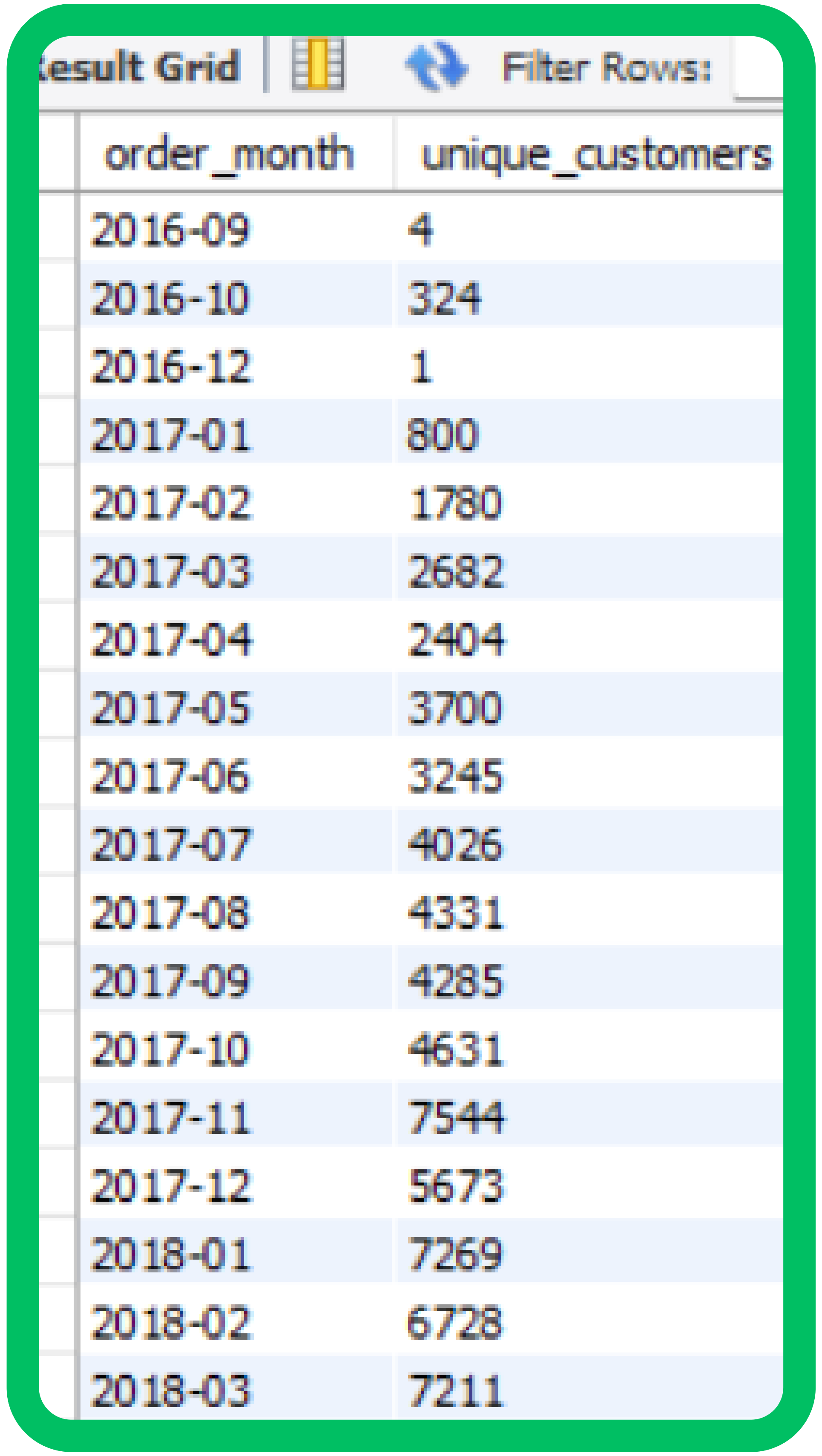


category	sales	sales_type
cine photo	28590.9599339962	high
Blu Ray DVDs	23805.839819669724	high
song	20704.289985895157	high
Imported books	20105.670055389404	high
CITTE AND UPHACK ...	17995.620231628418	high
party articles	17898.93016719818	high
Fashion Women's Cl...	15328.859893798828	high
Kitchen portable and...	13006.950267791748	high
Hygiene diapers	12663.749942779541	high
Fashion Sport	10937.759925842285	high
La Cuisine	8740.590065002441	medium
Arts and Crafts	6978.509948730469	medium
flowers	6639.029954910278	medium
PC Gamer	6523.290000915527	medium
House Comfort 2	5131.620017051697	medium
cds music dvds	3598.2900466918945	low
Fashion Children's Cl...	2357.0100059509277	low
insurance and services	973.5299835205078	low

Count how many unique customers ordered each month.

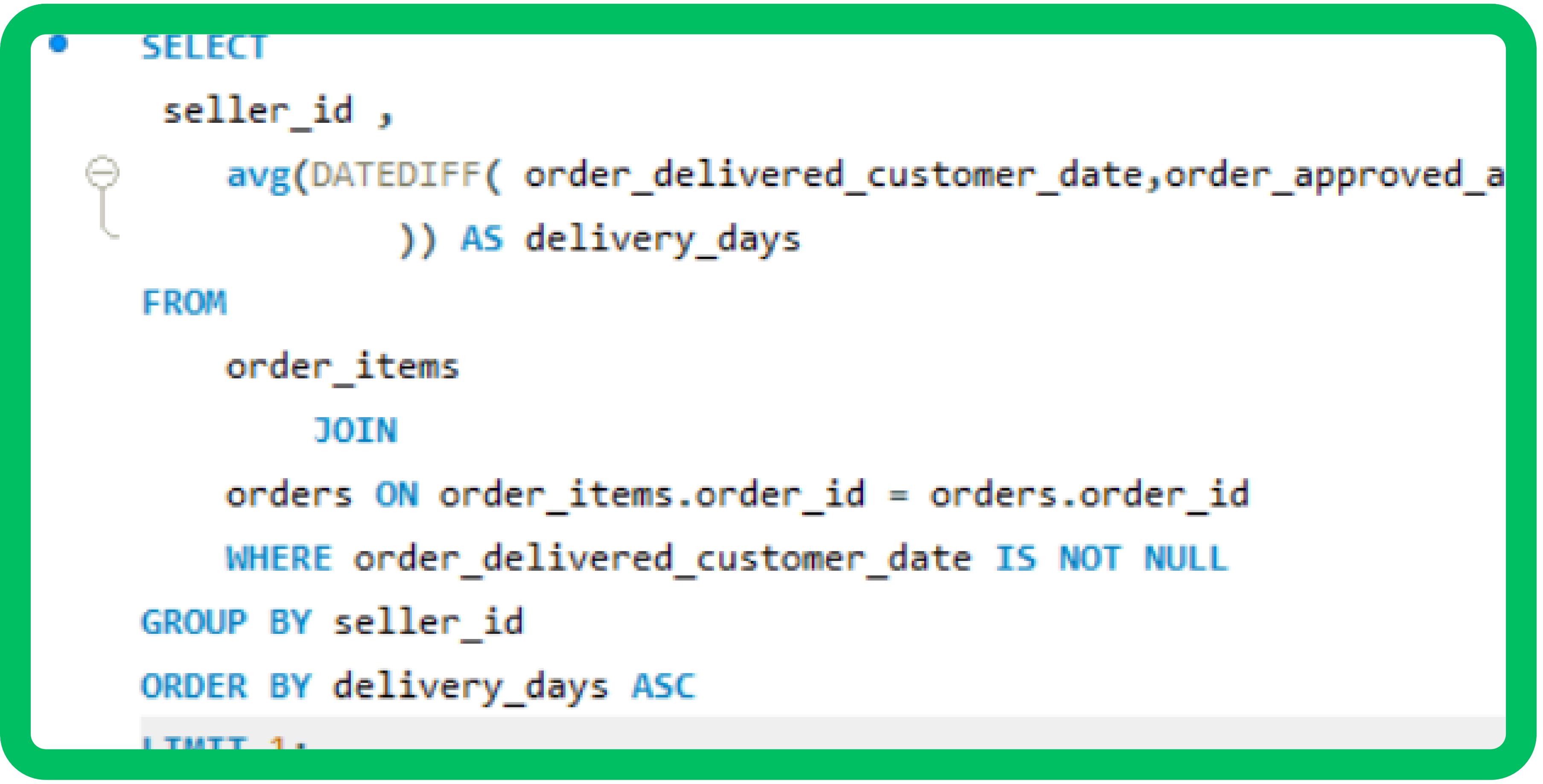


```sql
SELECT
    DATE_FORMAT(order_purchase_timestamp, '%Y-%m') AS order_month,
    COUNT(DISTINCT customer_id) AS unique_customers
FROM
    orders
GROUP BY order_month
ORDER BY order_month;
```

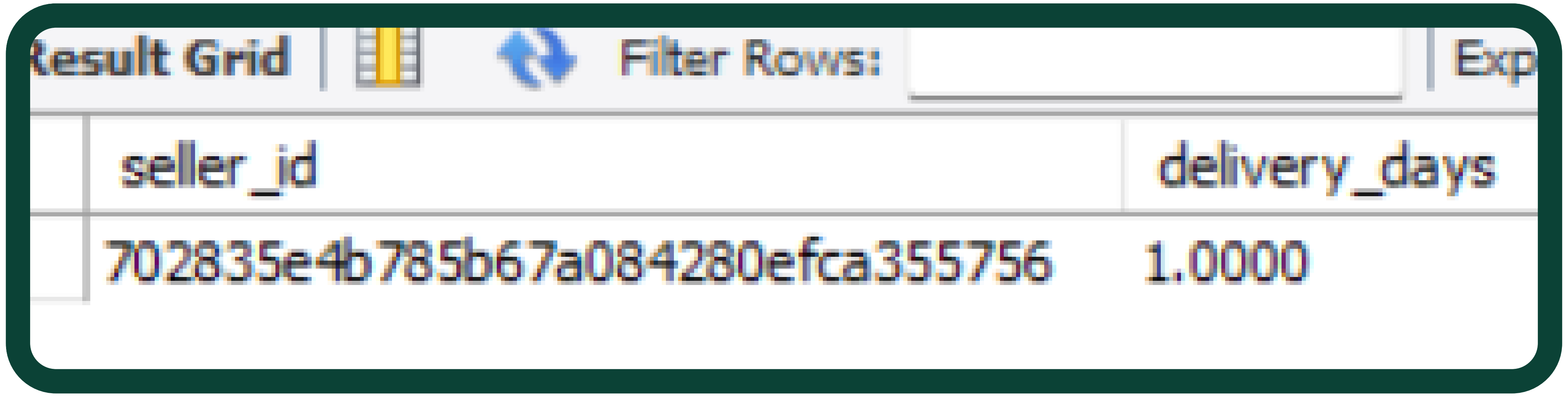


order_month	unique_customers
2016-09	4
2016-10	324
2016-12	1
2017-01	800
2017-02	1780
2017-03	2682
2017-04	2404
2017-05	3700
2017-06	3245
2017-07	4026
2017-08	4331
2017-09	4285
2017-10	4631
2017-11	7544
2017-12	5673
2018-01	7269
2018-02	6728
2018-03	7211

Find which seller delivers the fastest on average.

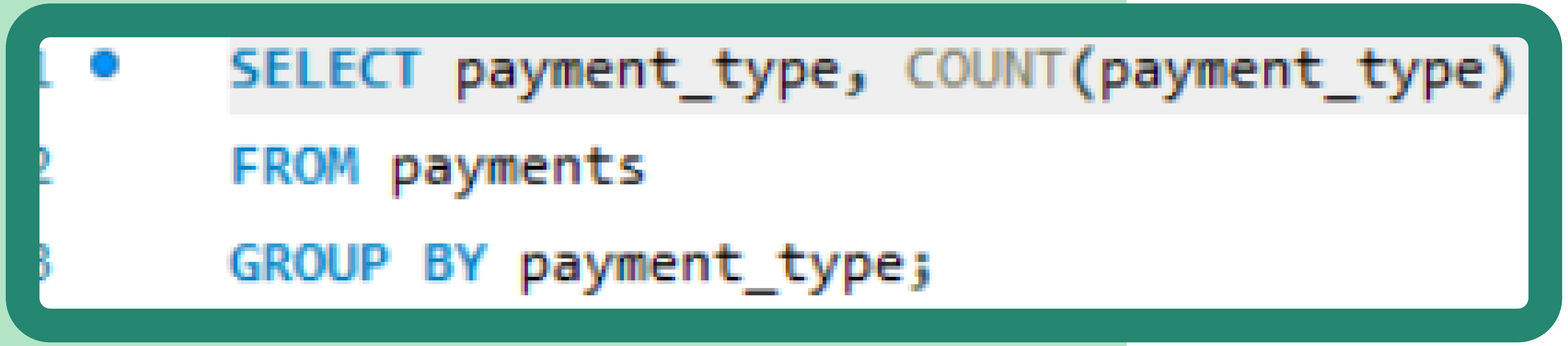


```
SELECT
 seller_id ,
    avg(DATEDIFF( order_delivered_customer_date,order_approved_a
            )) AS delivery_days
FROM
    order_items
        JOIN
    orders ON order_items.order_id = orders.order_id
    WHERE order_delivered_customer_date IS NOT NULL
GROUP BY seller_id
ORDER BY delivery_days ASC
LIMIT 1;
```

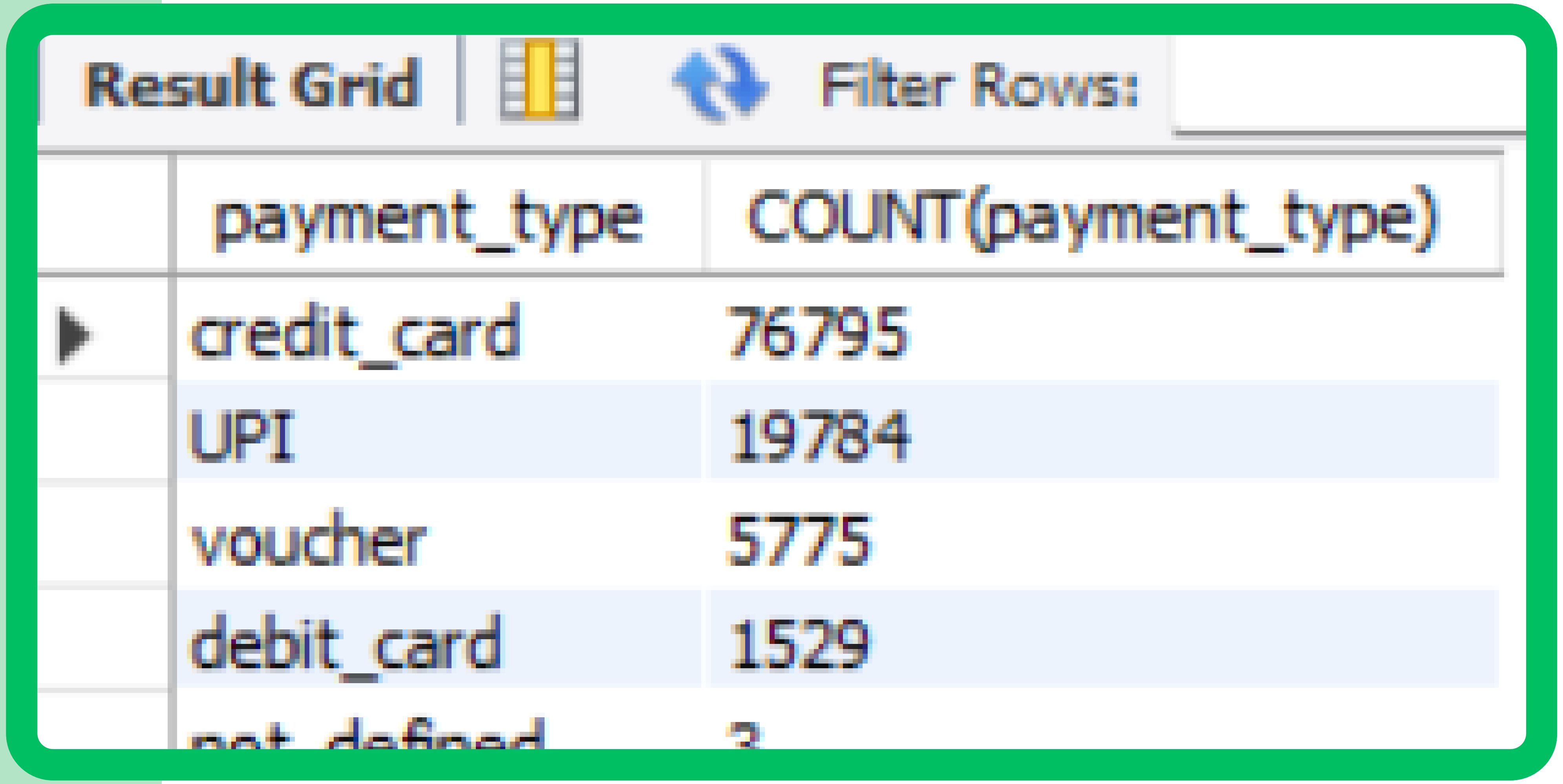


seller_id	delivery_days
702835e4b785b67a084280efca355756	1.0000

Find which payment type is used the most.

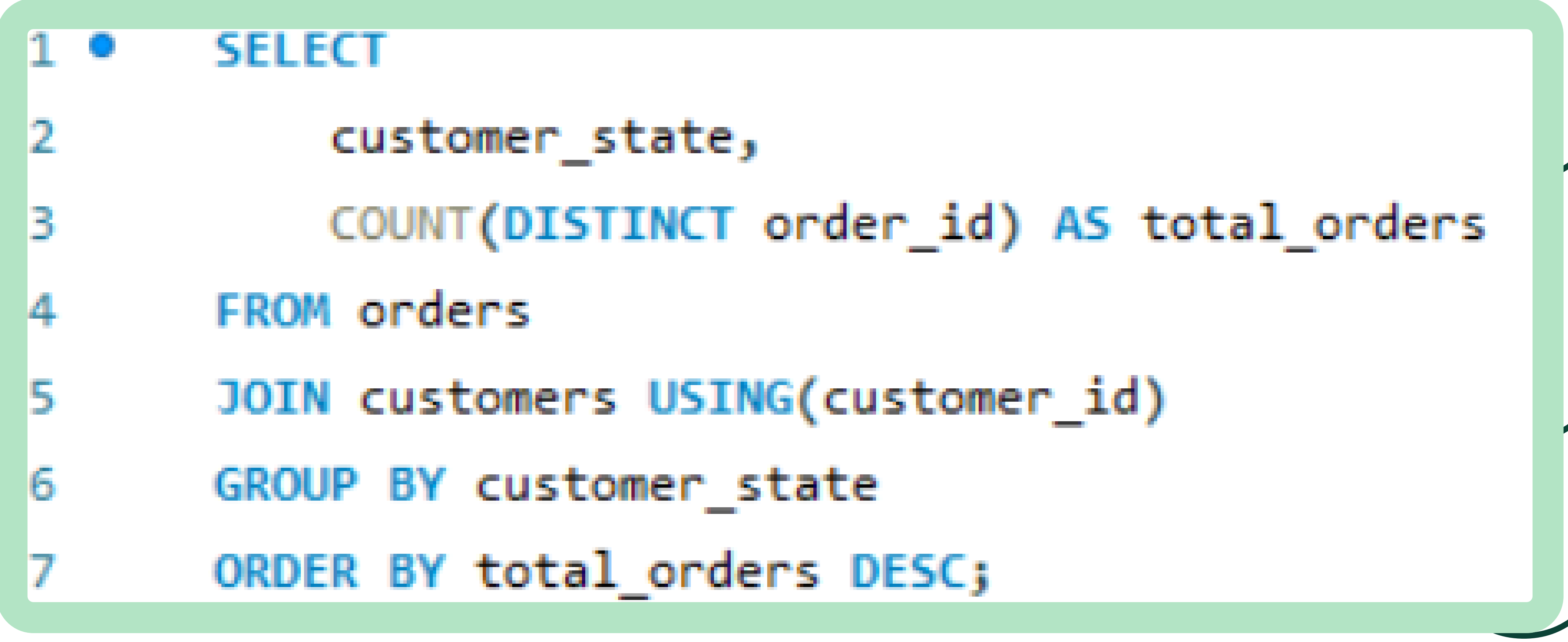


```
SELECT payment_type, COUNT(payment_type)
FROM payments
GROUP BY payment_type;
```

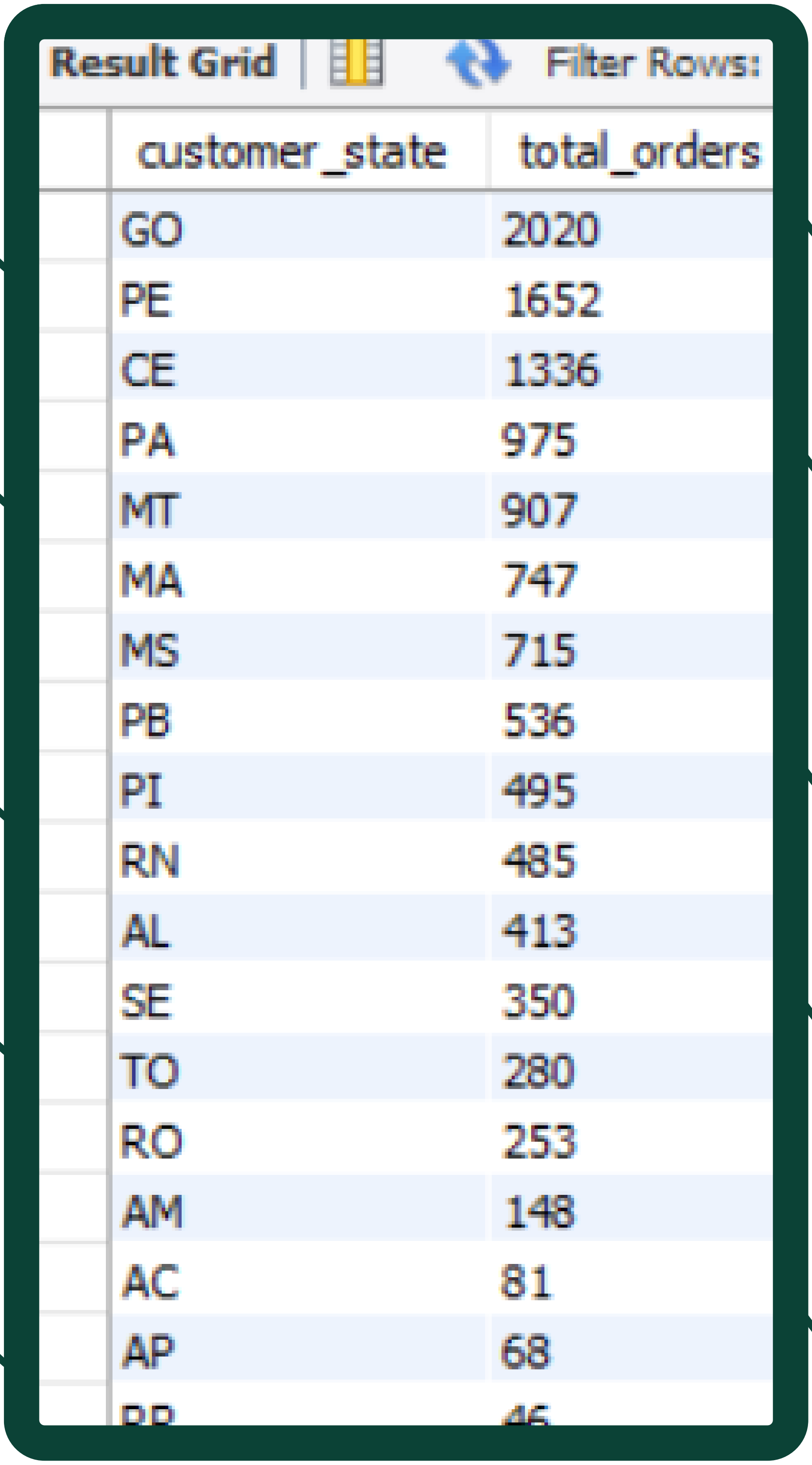


payment_type	COUNT(payment_type)
credit_card	76795
UPI	19784
voucher	5775
debit_card	1529
not_defined	3

Count orders coming from each state.

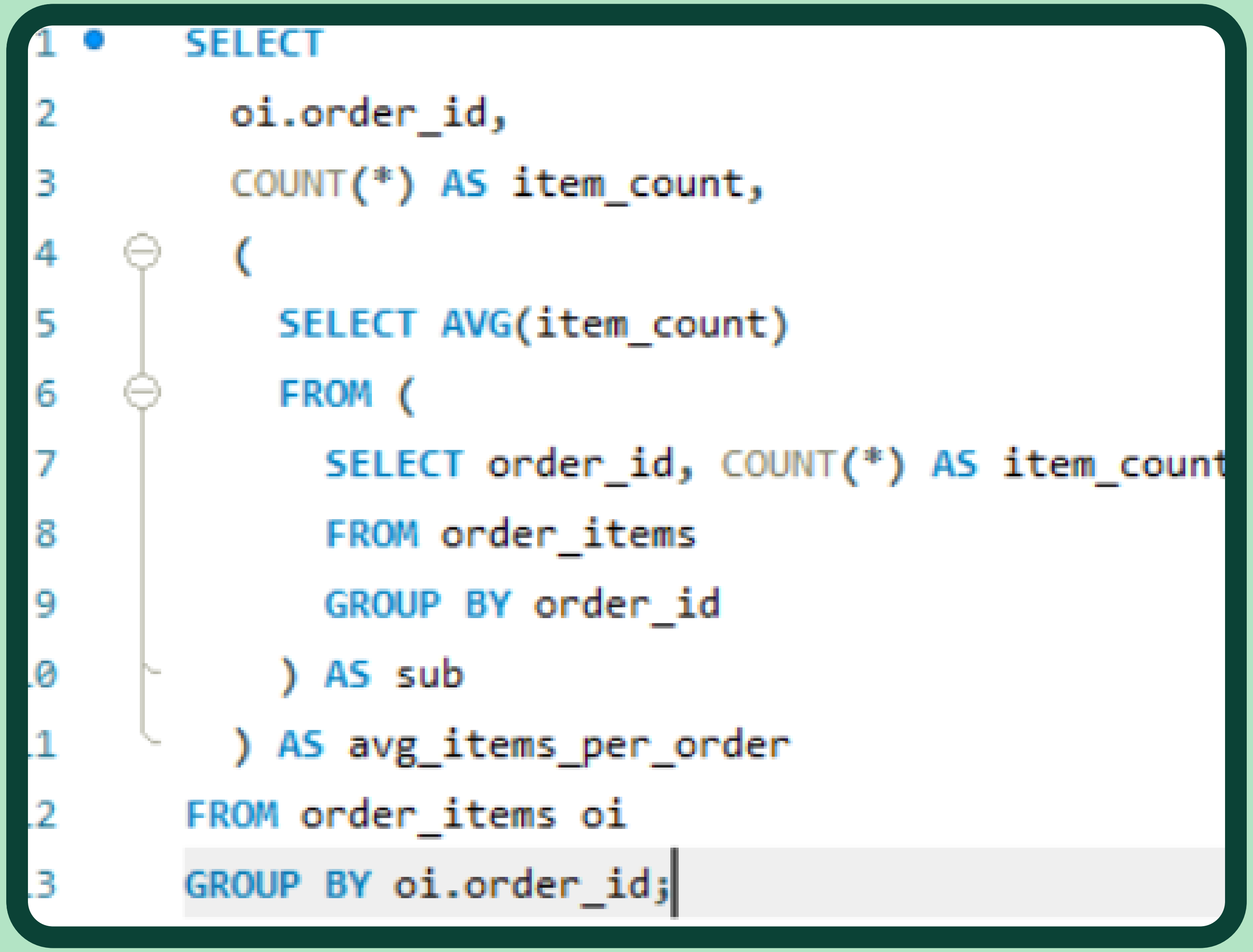


```sql
SELECT
    customer_state,
    COUNT(DISTINCT order_id) AS total_orders
FROM orders
JOIN customers USING(customer_id)
GROUP BY customer_state
ORDER BY total_orders DESC;
```

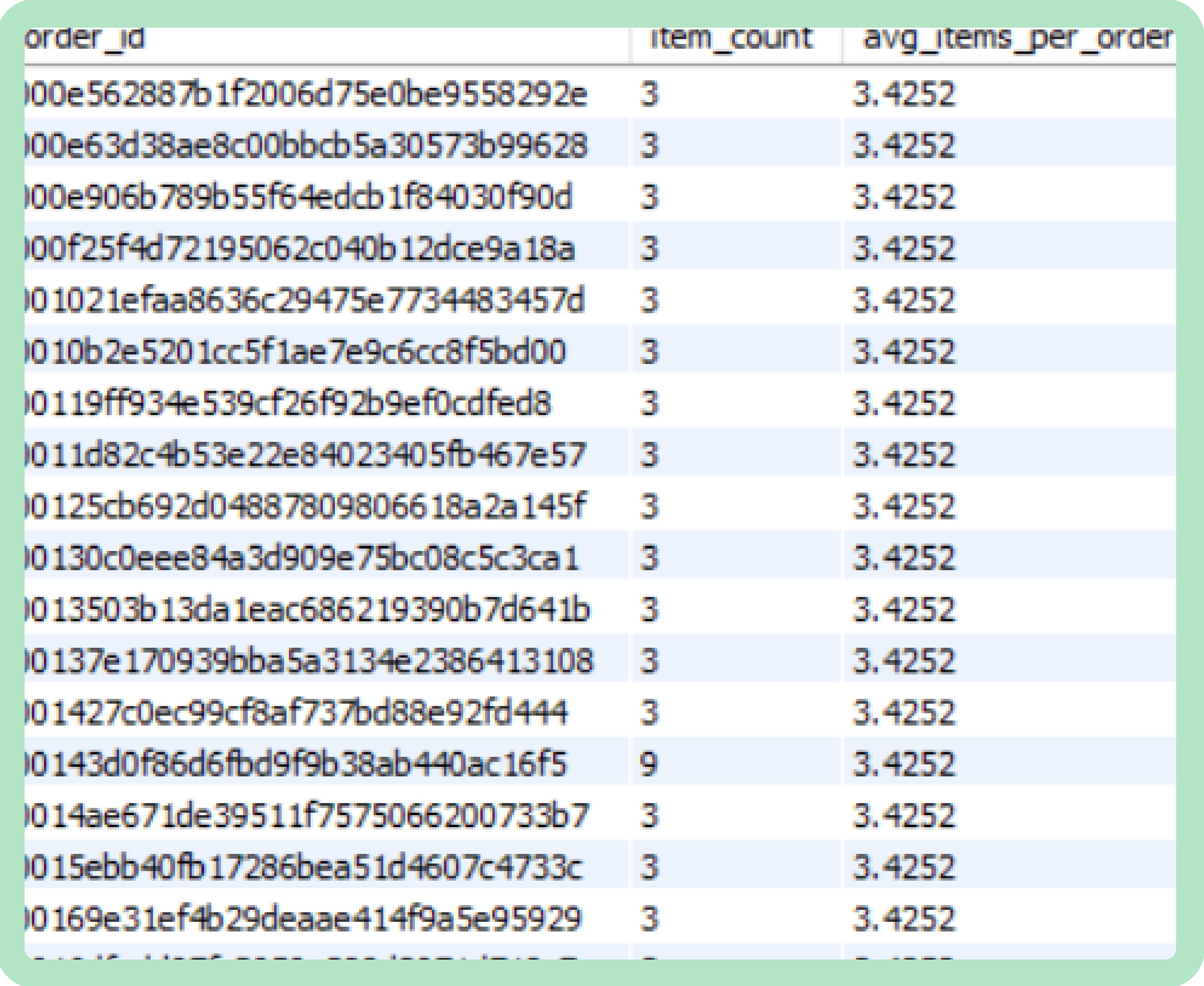


customer_state	total_orders
GO	2020
PE	1652
CE	1336
PA	975
MT	907
MA	747
MS	715
PB	536
PI	495
RN	485
AL	413
SE	350
TO	280
RO	253
AM	148
AC	81
AP	68
RR	46

find the average quantity of items per order.

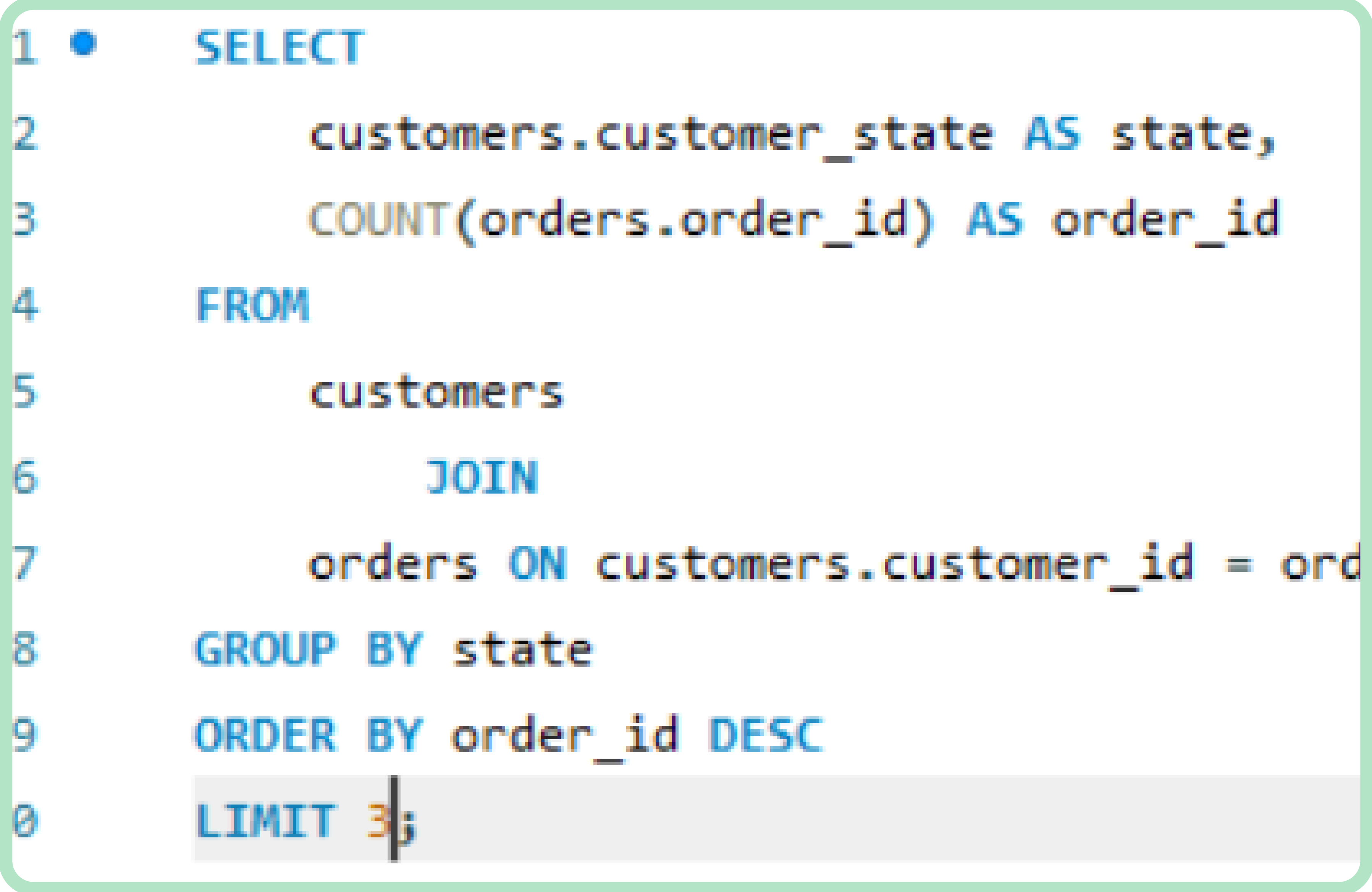


```
SELECT
  oi.order_id,
  COUNT(*) AS item_count,
  (
    SELECT AVG(item_count)
    FROM (
      SELECT order_id, COUNT(*) AS item_count
      FROM order_items
      GROUP BY order_id
    ) AS sub
  ) AS avg_items_per_order
FROM order_items oi
GROUP BY oi.order_id;
```

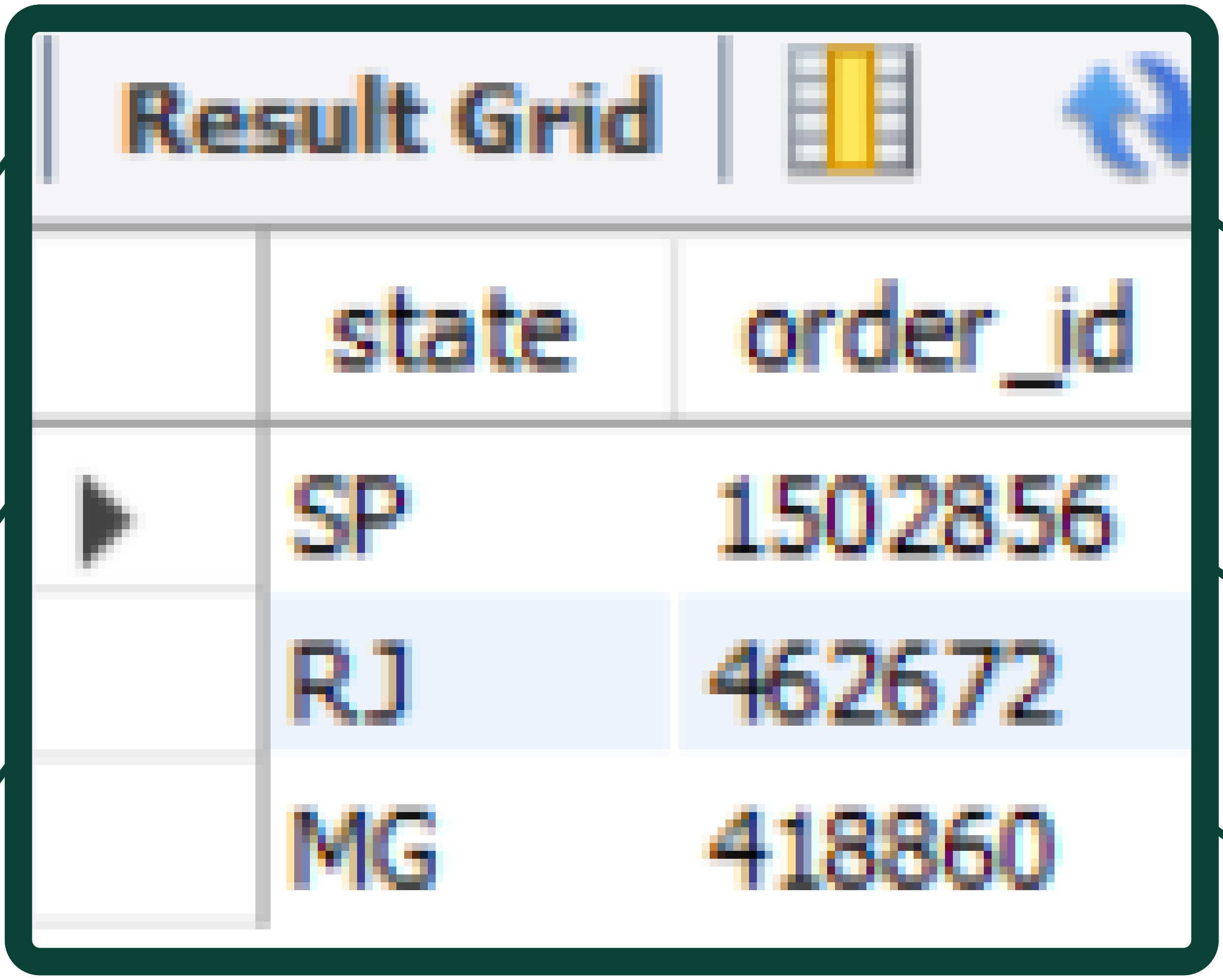


order_id	item_count	avg_items_per_order
000e562887b1f2006d75e0be9558292e	3	3.4252
000e63d38ae8c00bbcb5a30573b99628	3	3.4252
000e906b789b55f64edcb1f84030f90d	3	3.4252
000f25f4d72195062c040b12dce9a18a	3	3.4252
001021efaa8636c29475e7734483457d	3	3.4252
0010b2e5201cc5f1ae7e9c6cc8f5bd00	3	3.4252
00119ff934e539cf26f92b9ef0cdfed8	3	3.4252
0011d82c4b53e22e84023405fb467e57	3	3.4252
00125cb692d04887809806618a2a145f	3	3.4252
00130c0eee84a3d909e75bc08c5c3ca1	3	3.4252
0013503b13da1eac686219390b7d641b	3	3.4252
00137e170939bba5a3134e2386413108	3	3.4252
001427c0ec99cf8af737bd88e92fd444	3	3.4252
00143d0f86d6fbd9f9b38ab440ac16f5	9	3.4252
0014ae671de39511f7575066200733b7	3	3.4252
0015ebb40fb17286bea51d4607c4733c	3	3.4252
00169e31ef4b29deaae414f9a5e95929	3	3.4252

Find the 3 states with the most customer orders.

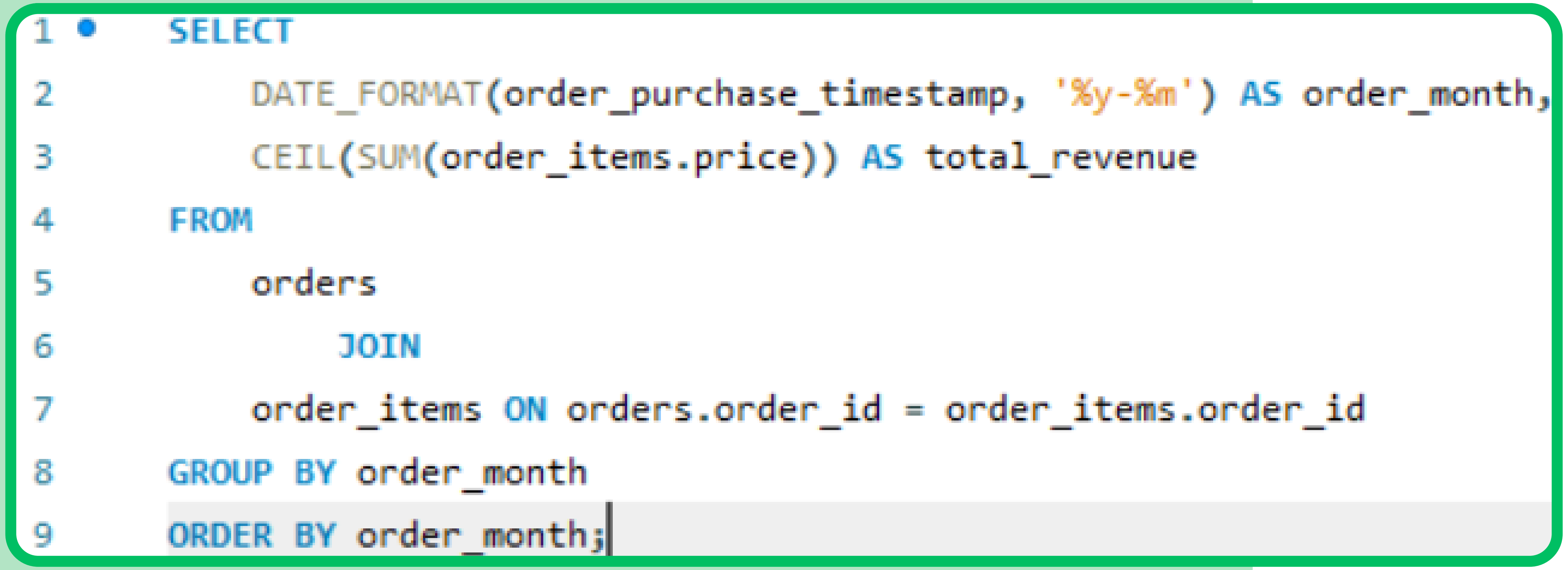


```sql
SELECT
    customers.customer_state AS state,
    COUNT(orders.order_id) AS order_id
FROM
    customers
        JOIN
    orders ON customers.customer_id = ord
GROUP BY state
ORDER BY order_id DESC
LIMIT 3;
```

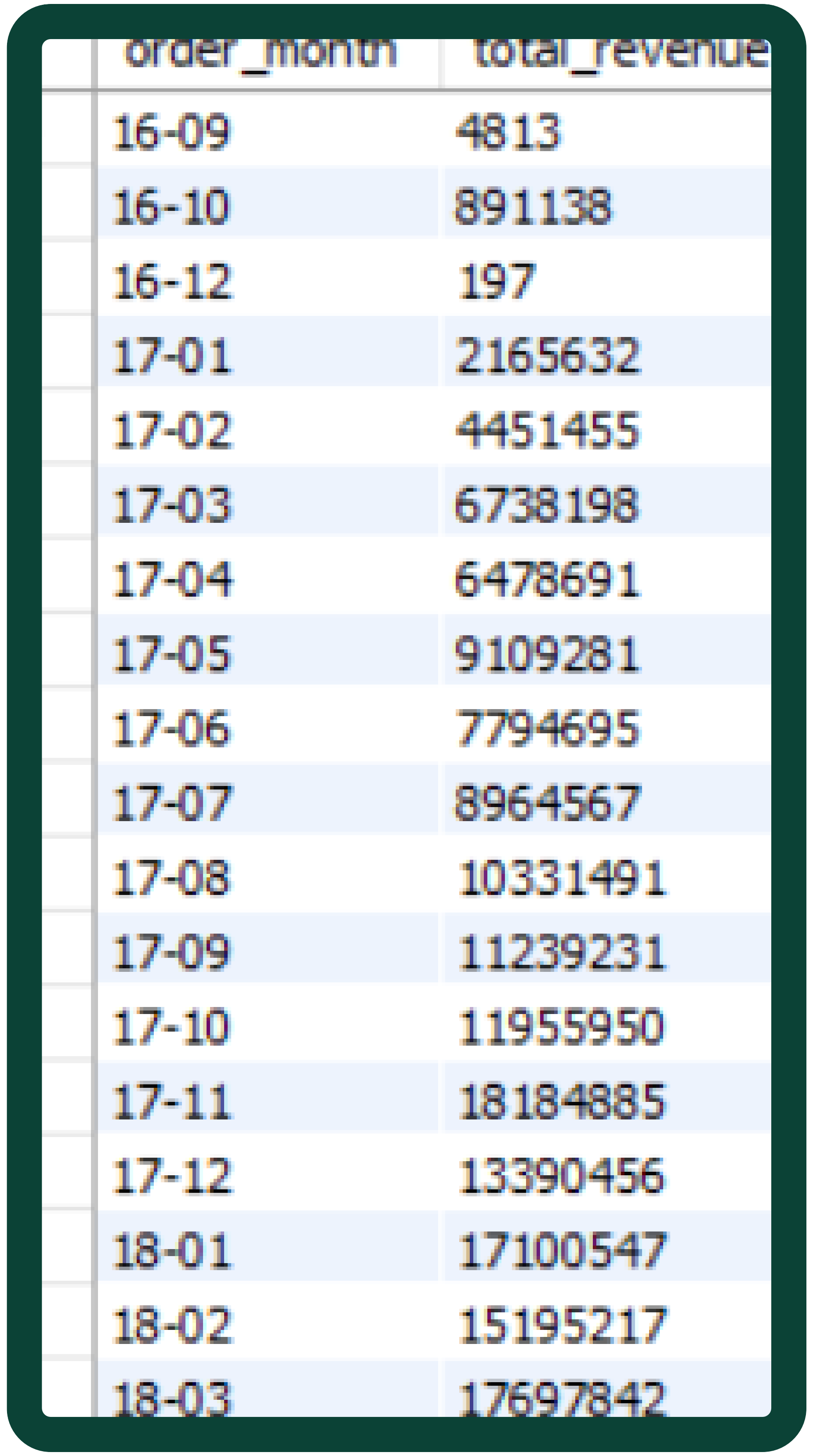


state	order_id
SP	1502856
RJ	462672
MG	418860

See how total revenue changes each month.

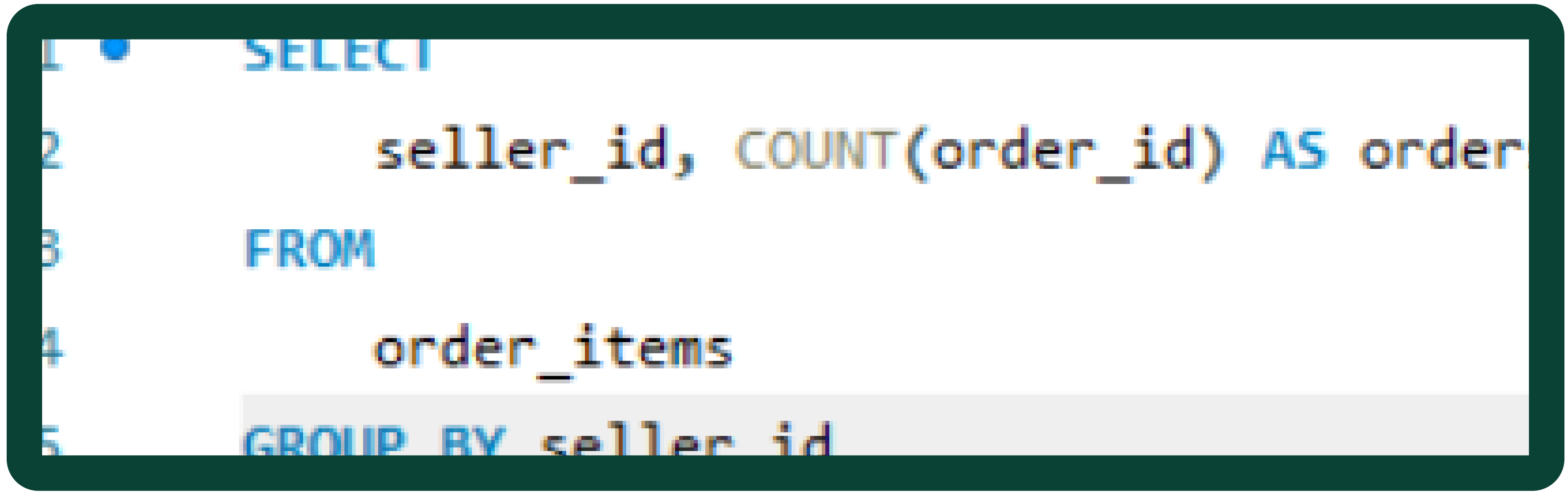


```sql
SELECT
    DATE_FORMAT(order_purchase_timestamp, '%y-%m') AS order_month,
    CEIL(SUM(order_items.price)) AS total_revenue
FROM
    orders
        JOIN
    order_items ON orders.order_id = order_items.order_id
GROUP BY order_month
ORDER BY order_month;
```

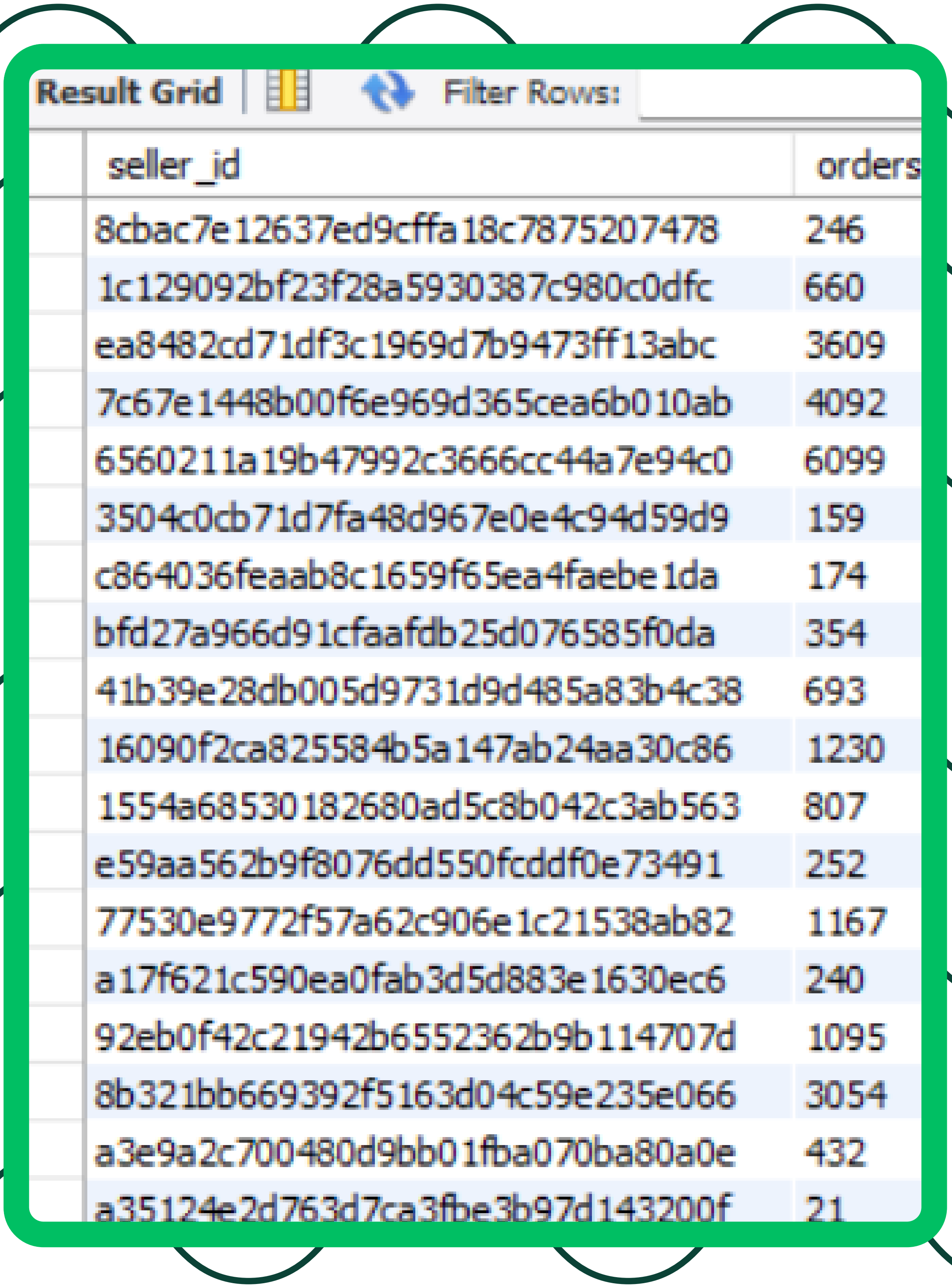


order_month	total_revenue
16-09	4813
16-10	891138
16-12	197
17-01	2165632
17-02	4451455
17-03	6738198
17-04	6478691
17-05	9109281
17-06	7794695
17-07	8964567
17-08	10331491
17-09	11239231
17-10	11955950
17-11	18184885
17-12	13390456
18-01	17100547
18-02	15195217
18-03	17697842

Find sellers with the most number of distinct orders.

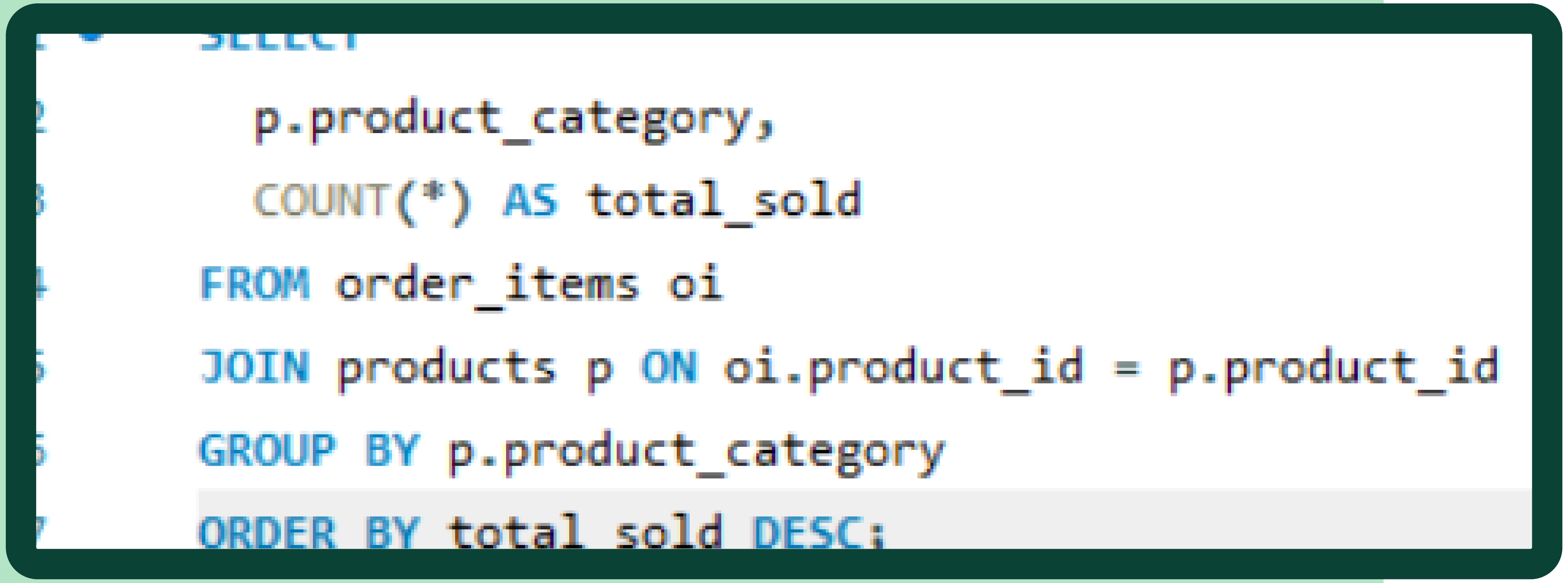


```
SELECT
    seller_id, COUNT(order_id) AS order
FROM
    order_items
GROUP BY seller_id
```

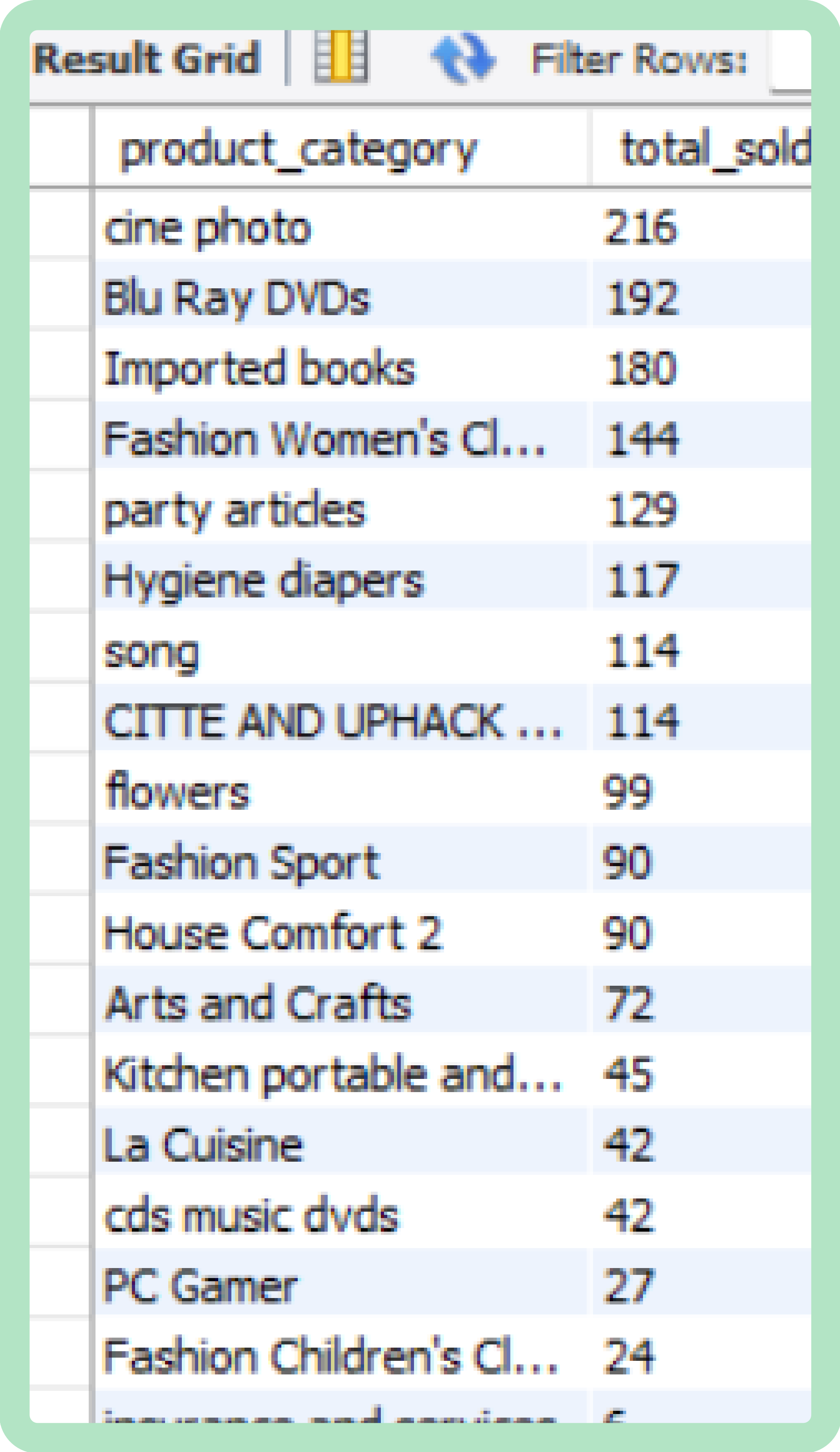


seller_id	orders
8cbac7e12637ed9cffa18c7875207478	246
1c129092bf23f28a5930387c980c0dfc	660
ea8482cd71df3c1969d7b9473ff13abc	3609
7c67e1448b00f6e969d365cea6b010ab	4092
6560211a19b47992c3666cc44a7e94c0	6099
3504c0cb71d7fa48d967e0e4c94d59d9	159
c864036feaab8c1659f65ea4faebe1da	174
bfd27a966d91cfaafdb25d076585f0da	354
41b39e28db005d9731d9d485a83b4c38	693
16090f2ca825584b5a147ab24aa30c86	1230
1554a68530182680ad5c8b042c3ab563	807
e59aa562b9f8076dd550fcddf0e73491	252
77530e9772f57a62c906e1c21538ab82	1167
a17f621c590ea0fab3d5d883e1630ec6	240
92eb0f42c21942b6552362b9b114707d	1095
8b321bb669392f5163d04c59e235e066	3054
a3e9a2c700480d9bb01fba070ba80a0e	432
a35124e2d763d7ca3fbe3b97d143200f	21

Find how many products were sold per category



```
SELECT
    p.product_category,
    COUNT(*) AS total_sold
FROM order_items oi
JOIN products p ON oi.product_id = p.product_id
GROUP BY p.product_category
ORDER BY total_sold DESC;
```



product_category	total_sold
cine photo	216
Blu Ray DVDs	192
Imported books	180
Fashion Women's Cl...	144
party articles	129
Hygiene diapers	117
song	114
CITTE AND UPHACK ...	114
flowers	99
Fashion Sport	90
House Comfort 2	90
Arts and Crafts	72
Kitchen portable and...	45
La Cuisine	42
cds music dvds	42
PC Gamer	27
Fashion Children's Cl...	24
insurance and services	6

Get In Touch

Email

itsmaygal02@gmail.com

LinkedIn

https://www.linkedin.com/in/mayank-goyal-4b8756363/

Github

https://github.com/mayank-goyal09

THANK YOU

This project was built as part of my data analytics learning journey using SQL and real-world datasets