CDACL - 006 - Market Analysis

Datamites

Global Institute for Data Science

Team ID: PTID-CDA-DEC-24-283

Project ID: CDACL-006

Project Name: Market Analysis

By

Team member 1: -

Name: Raasi P

Email: rashireddy981212@gmail.com

Team member 2: -

Name: Neetu Sharma

Email: neets.2012@gmail.com

Perform the Problem Queries:

1. What are the top 10 aisles with the highest number of products?
2. How many unique departments are there in the dataset?
3. What is the distribution of products across departments?
4. What are the top 10 products with the highest reorder rates?
5. How many unique users have placed orders in the dataset?
6. What is the average number of days between orders for each user?
7. What are the peak hours of order placement during the day?
8. How does order volume vary by day of the week?
9. What are the top 10 most ordered products?
10. How many users have placed orders in each department?
11. What is the average number of products per order?
12. What are the most reordered products in each department?
13. How many products have been reordered more than once?
14. What is the average number of products added to the cart per order?
15. How does the number of orders vary by hour of the day?
16. What is the distribution of order sizes (number of products per order)?
17. What is the average reorder rate for products in each aisle?
18. How does the average order size vary by day of the week?
19. What are the top 10 users with the highest number of orders?
20. How many products belong to each aisle and department?

Also, analyse customer purchasing behaviour and product performance to optimize marketing strategies and improve customer satisfaction.

Attributes information:

1. **Aisles Dataset:**
   - aisle_id: Unique identifier for the aisle. ***(Primary Keys)***
   - aisle: Name of the aisle.

2. **Departments Dataset:**
   - department_id: Unique identifier for the department. ***(Primary Keys)***
   - department: Name of the department.

3. **Order Products Prior Dataset:**
   - order_id: Unique identifier for the order. ***(Primary Keys)***
   - product_id: Unique identifier for the product.
   - add_to_cart_order: Order in which the product was added to the cart.
   - reordered: Indicates whether the product was reordered in this order (1 for reordered, 0 for not reordered).

4. **Orders Dataset:**
   - order_id: Unique identifier for the order. ***(Primary Keys)***
   - user_id: Unique identifier for the user.
   - eval_set: Evaluation set (prior, train, test).
   - order_number: Order sequence number for the user.
   - order_dow: Day of the week the order was placed.
   - order_hour_of_day: Hour of the day the order was placed.
   - days_since_prior_order: Number of days since the last order.

5. **Products Dataset:**
   - product_id: Unique identifier for the product.
   - product_name: Name of the product.
   - aisle_id: Identifier for the aisle the product belongs to. ***(Primary Keys)***
   - department_id: Identifier for the department the product belongs to.

QUERIES WITH ANSWERS

PERFORM THE PROBLEM QUERIES

1) What are the top 10 aisles with the highest number of products?

```
SELECT count(*) AS Product_count, aisle_id
FROM products GROUP BY aisle_id
ORDER BY aisle_id DESC LIMIT 10;
```

2) How many unique departments are there in the dataset?

```
SELECT COUNT(DISTINCT department) AS Unique_departments
FROM departments;
```

3) What is the distribution of products across departments?

```
SELECT department_id, count(*) AS Product_count
FROM products GROUP BY department_id
ORDER BY department_id;
```

4) What are the top 10 products with the highest reorder rates?

```
SELECT reordered, product_id
FROM order_products_train
ORDER BY reordered
DESC LIMIT 10;
```

5) How many unique users have placed orders in the dataset?

```
SELECT COUNT(DISTINCT user_id) AS Unique_users
FROM orders;
```

6) What is the average number of days between orders for each user?

```
SELECT user_id, AVG(days_since_prior_order)
AS Avg_days_between_orders
FROM orders GROUP BY user_id;
```

7) What are the peak hours of order placement during the day?

```
SELECT order_hour_of_day, count(*) as order_count
FROM orders
GROUP BY order_hour_of_day
ORDER BY order_count DESC;
```

8) How does order volume vary by day of the week?

```
SELECT order_dow, count(*) AS order_count
FROM orders
GROUP BY order_dow
ORDER BY order_count DESC;
```

9) What are the top 10 most ordered products?

```
SELECT product_id, COUNT(*) AS product_count
FROM order_products_train
GROUP BY product_id
ORDER BY product_count DESC LIMIT 10;
```

10) How many users have placed orders in each department?

```
SELECT p.department_id, count(DISTINCT o.user_id) AS user_count
FROM products p JOIN order_products_train opt
ON p.product_id = opt.product_id
JOIN orders o ON opt.order_id = o.order_id
GROUP BY p.department_id;
```

11) What is the average number of products per order?

```
SELECT order_id, AVG(product_id) AS avg_products_per_order
FROM order_products_train
GROUP BY order_id;
```

12) What are the most reordered products in each department?

```
SELECT p.department_id, opt.product_id,
SUM(opt.reordered) AS reordered_products
FROM products p JOIN order_products_train opt
ON p.product_id = opt.product_id
GROUP BY opt.product_id
ORDER BY reordered_products DESC;
```

13) How many products have been reordered more than once?

```
SELECT COUNT(*) AS products_reordered_more_than_once
FROM
(SELECT product_id, SUM(reordered) AS reorder_count
FROM order_products_train
GROUP BY product_id HAVING reorder_count > 1)
AS subquery;
```

14) What is the average number of products added to the cart per order?

```
SELECT order_id, AVG(product_id) AS avg_product_count
FROM order_products_train
GROUP BY order_id;
```

15) What is the average number of products added to the cart per order?

```
SELECT order_hour_of_day, COUNT(*) AS order_number_count
FROM orders
GROUP BY order_hour_of_day
ORDER BY order_number_count DESC;
```

16) What is the distribution of order sizes (number of products per order)?

```
SELECT order_id, COUNT(*) AS no_of_products_per_order
FROM order_products_train
GROUP BY order_id;
```

17) What is the average reorder rate for products in each aisle?

```
SELECT p.aisle_id, AVG(opt.reordered) AS avg_reorder_rate
FROM order_products_train opt
JOIN products p ON opt.product_id = p.product_id
GROUP BY p.aisle_id;
```

18) How does the average order size vary by day of the week?

```
SELECT o.order_dow AS order_day, AVG(order_number) AS avg_order_size
FROM order_products_train opt
JOIN orders o ON opt.order_id = o.order_id
GROUP BY o.order_dow;
```

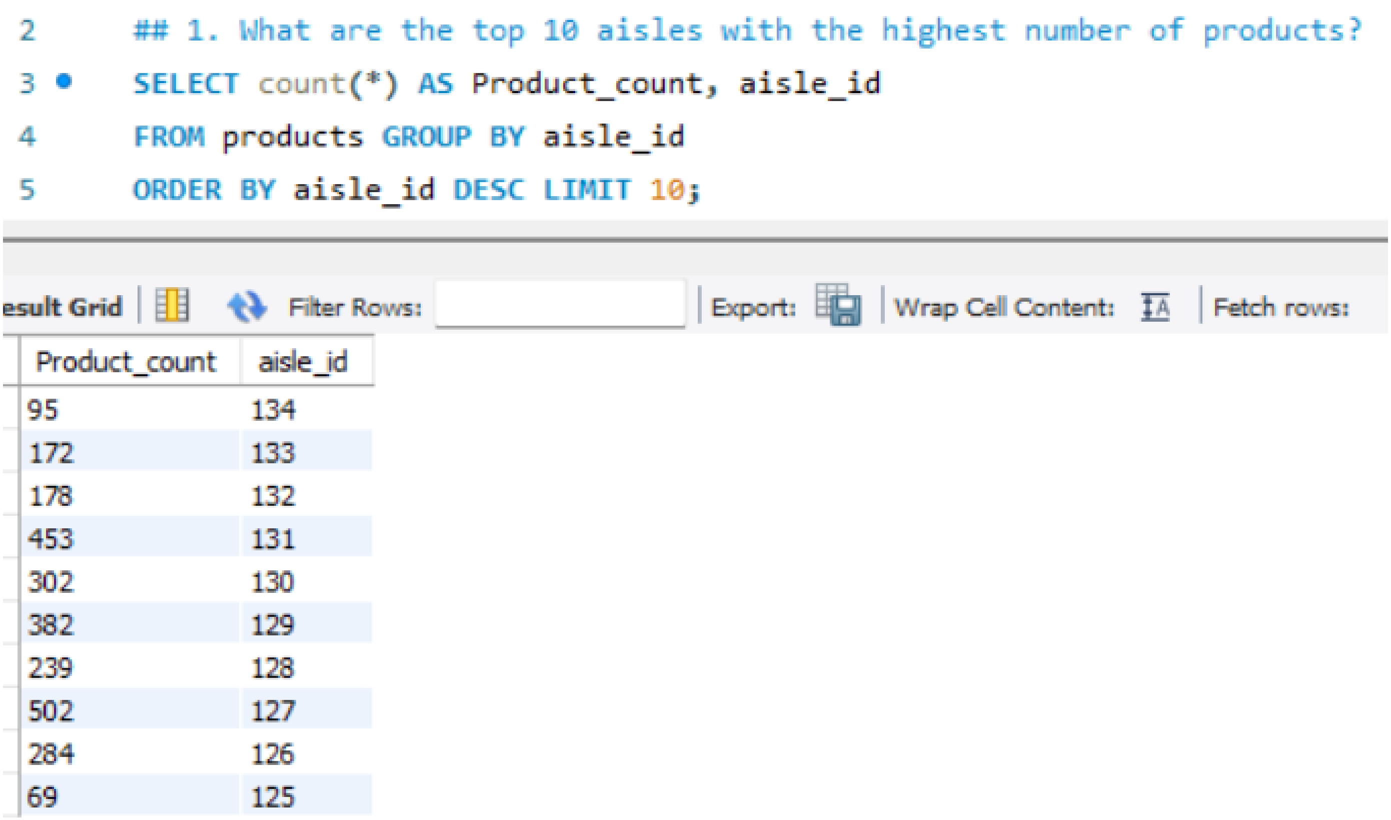
19) What are the top 10 users with the highest number of orders?

```
SELECT user_id, COUNT(order_id) AS order_count
FROM orders
GROUP BY user_id
ORDER BY order_count DESC
LIMIT 10;
```

20) How many products belong to each aisle and department?

```
SELECT aisle_id, department_id, COUNT(*) AS product_count
FROM products
GROUP BY aisle_id, department_id;
```

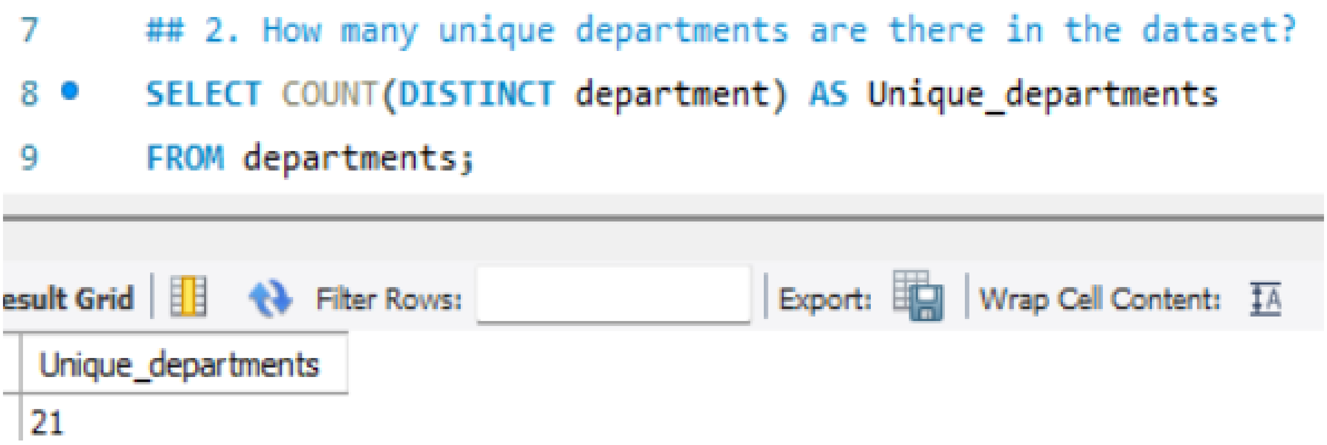
MySQL Execution with Explanation:



```
## 1. What are the top 10 aisles with the highest number of products?
SELECT count(*) AS Product_count, aisle_id
FROM products GROUP BY aisle_id
ORDER BY aisle_id DESC LIMIT 10;
```

Product_count	aisle_id
95	134
172	133
178	132
453	131
302	130
382	129
239	128
502	127
284	126
69	125

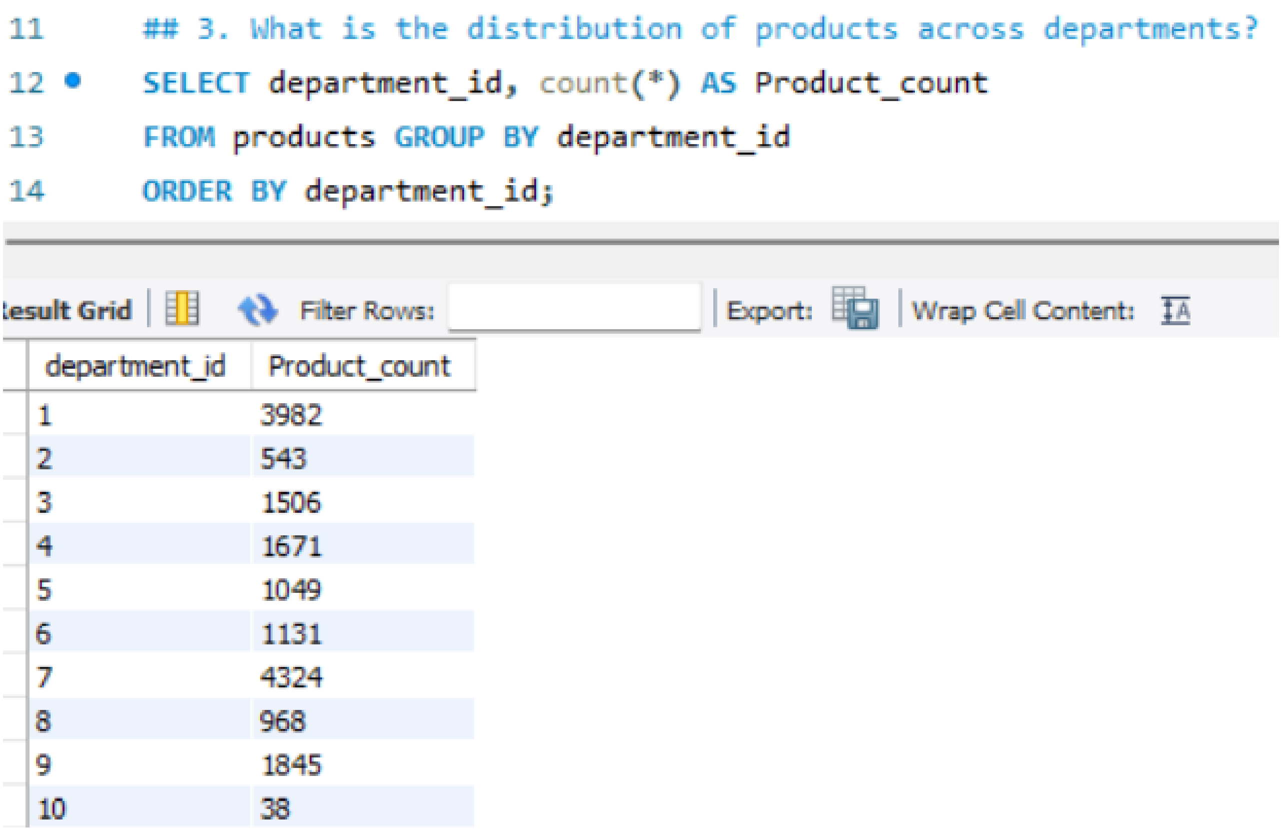
Statement Explanation: The above SELECT Statement retrieves count products (COUNT(*)) grouped by aisle_id. Sort by product count (ORDER BY Product_count DESC) and limit to the top 10 aisles (LIMIT 10) as shown in the result.



```
## 2. How many unique departments are there in the dataset?
SELECT COUNT(DISTINCT department) AS Unique_departments
FROM departments;
```

Unique_departments
21

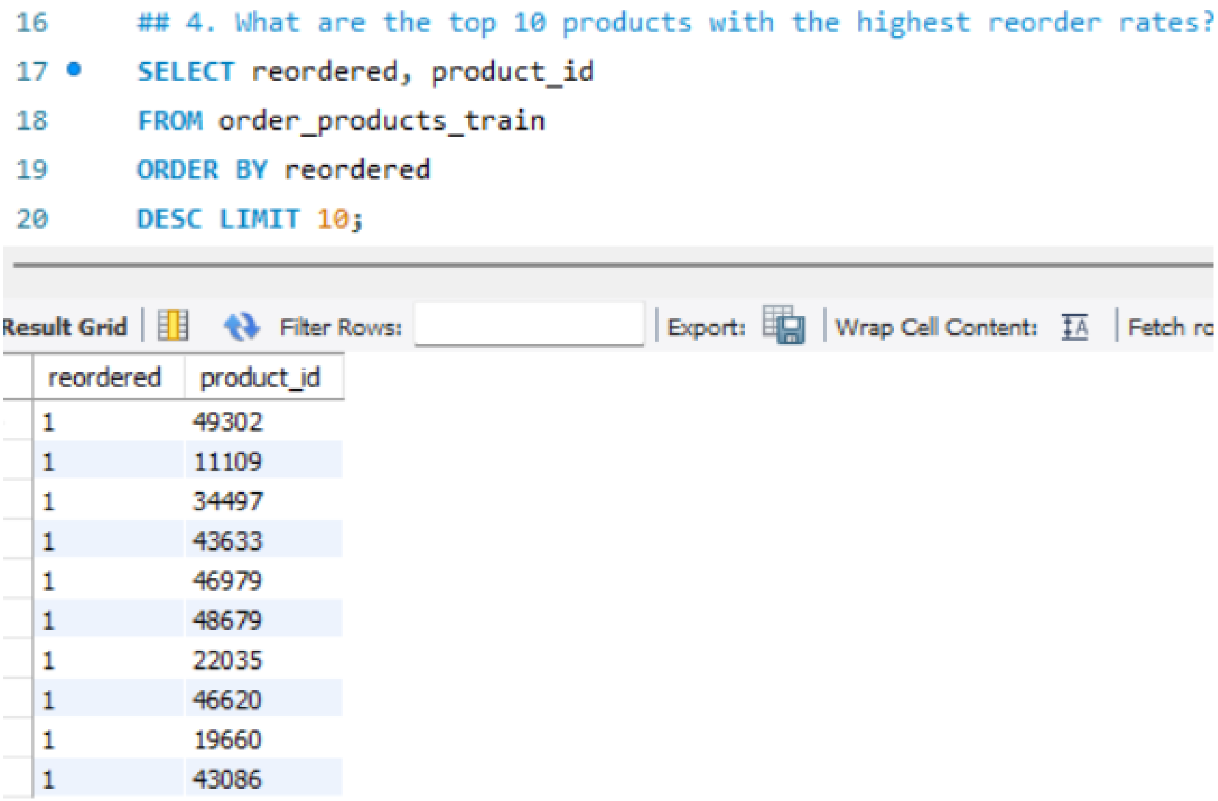
Statement Explanation: The above SELECT Statement retrieves COUNT(DISTINCT department_id) to count unique department IDs in the departments table as shown in the result.



```
## 3. What is the distribution of products across departments?
SELECT department_id, count(*) AS Product_count
FROM products GROUP BY department_id
ORDER BY department_id;
```

department_id	Product_count
1	3982
2	543
3	1506
4	1671
5	1049
6	1131
7	4324
8	968
9	1845
10	38

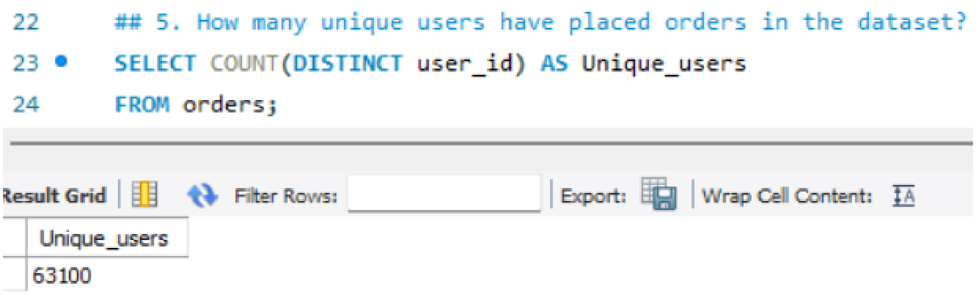
Statement Explanation: The above SELECT Statement retrieves count products grouped by department_id. Sort by department_id as shown in the result.



```
## 4. What are the top 10 products with the highest reorder rates?
SELECT reordered, product_id
FROM order_products_train
ORDER BY reordered
DESC LIMIT 10;
```

reordered	product_id
1	49302
1	11109
1	34497
1	43633
1	46979
1	48679
1	22035
1	46620
1	19660
1	43086

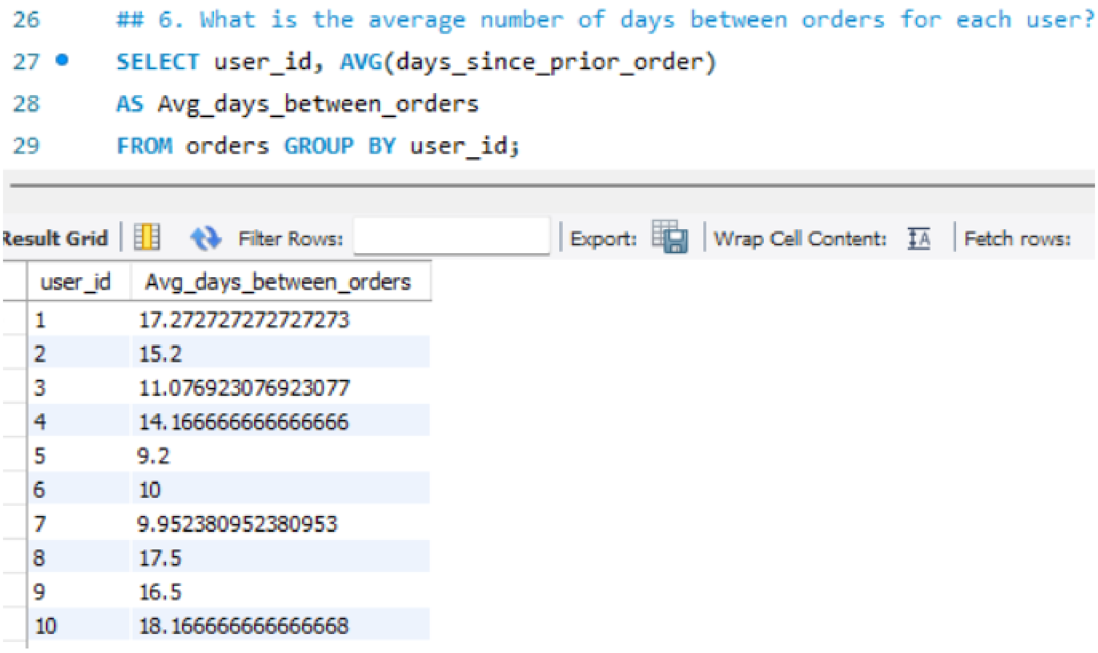
Statement Explanation: The above SELECT Statement retrieves calculated average reorder rate (AVG(reordered)) for each product_id. Sort by reorder rate and take the top 10 as shown in the result.



```sql
## 5. How many unique users have placed orders in the dataset?
SELECT COUNT(DISTINCT user_id) AS Unique_users
FROM orders;
```

Unique_users
63100

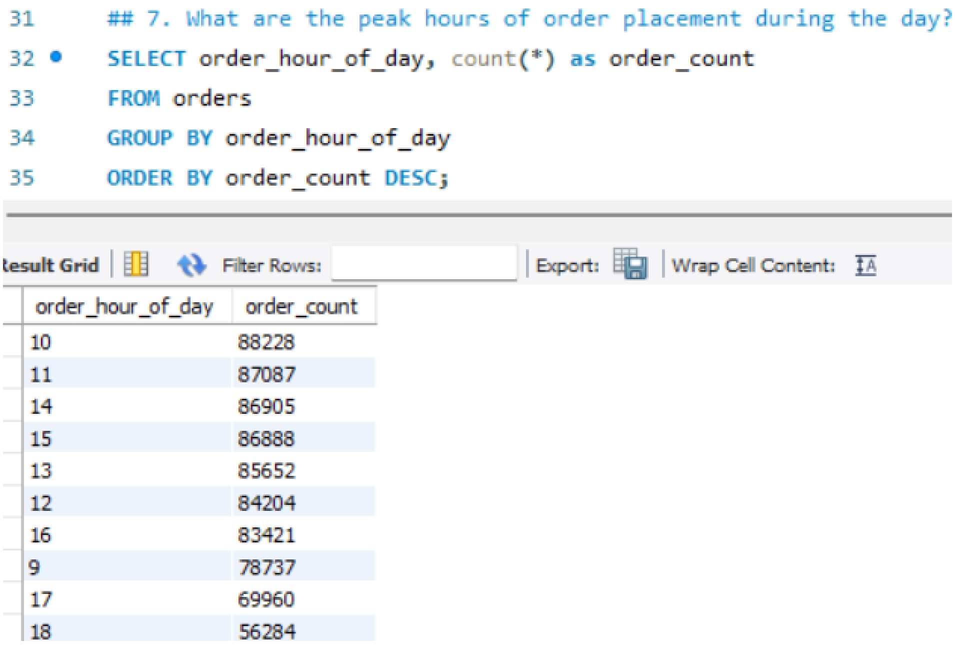
Statement Explanation: The above SELECT Statement retrieves COUNT(DISTINCT user_id) to count unique users in the orders table as shown in the result.



```sql
## 6. What is the average number of days between orders for each user?
SELECT user_id, AVG(days_since_prior_order)
AS Avg_days_between_orders
FROM orders GROUP BY user_id;
```

user_id	Avg_days_between_orders
1	17.272727272727273
2	15.2
3	11.076923076923077
4	14.166666666666666
5	9.2
6	10
7	9.952380952380953
8	17.5
9	16.5
10	18.166666666666668

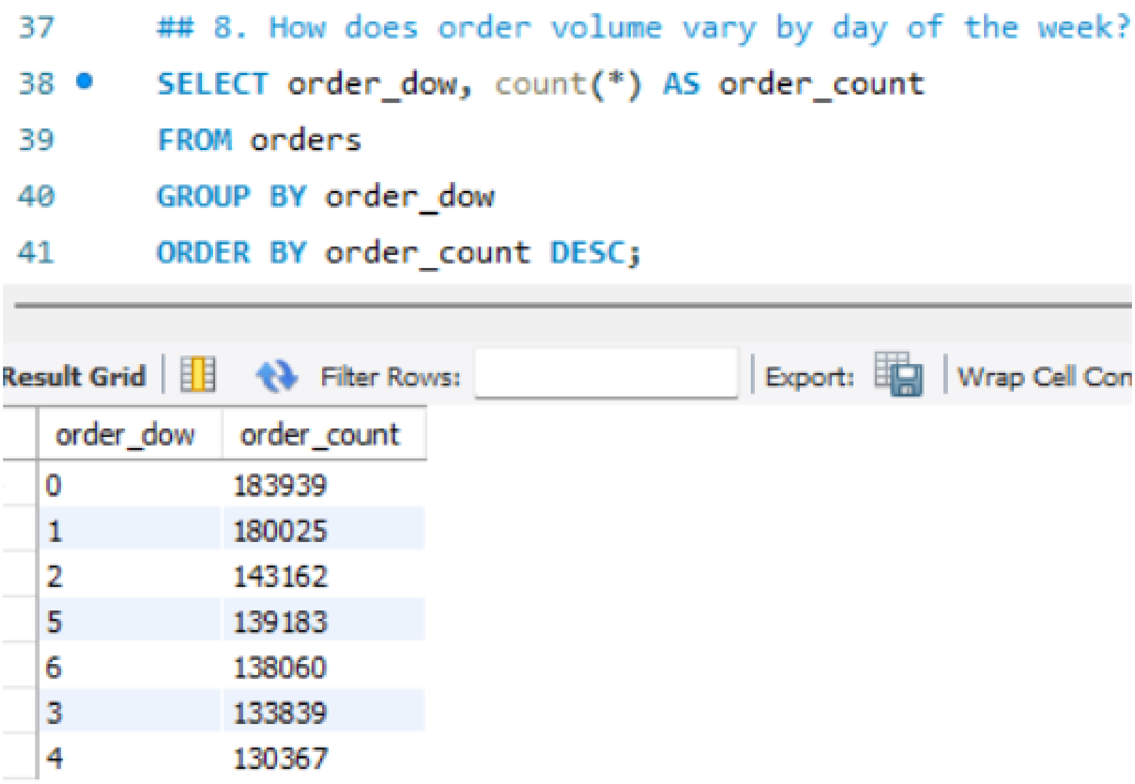
Statement Explanation: The above SELECT Statement retrieves group by user_id and compute the average (AVG(days_since_prior_order)) as shown in the result.



```sql
## 7. What are the peak hours of order placement during the day?
SELECT order_hour_of_day, count(*) as order_count
FROM orders
GROUP BY order_hour_of_day
ORDER BY order_count DESC;
```

order_hour_of_day	order_count
10	88228
11	87087
14	86905
15	86888
13	85652
12	84204
16	83421
9	78737
17	69960
18	56284

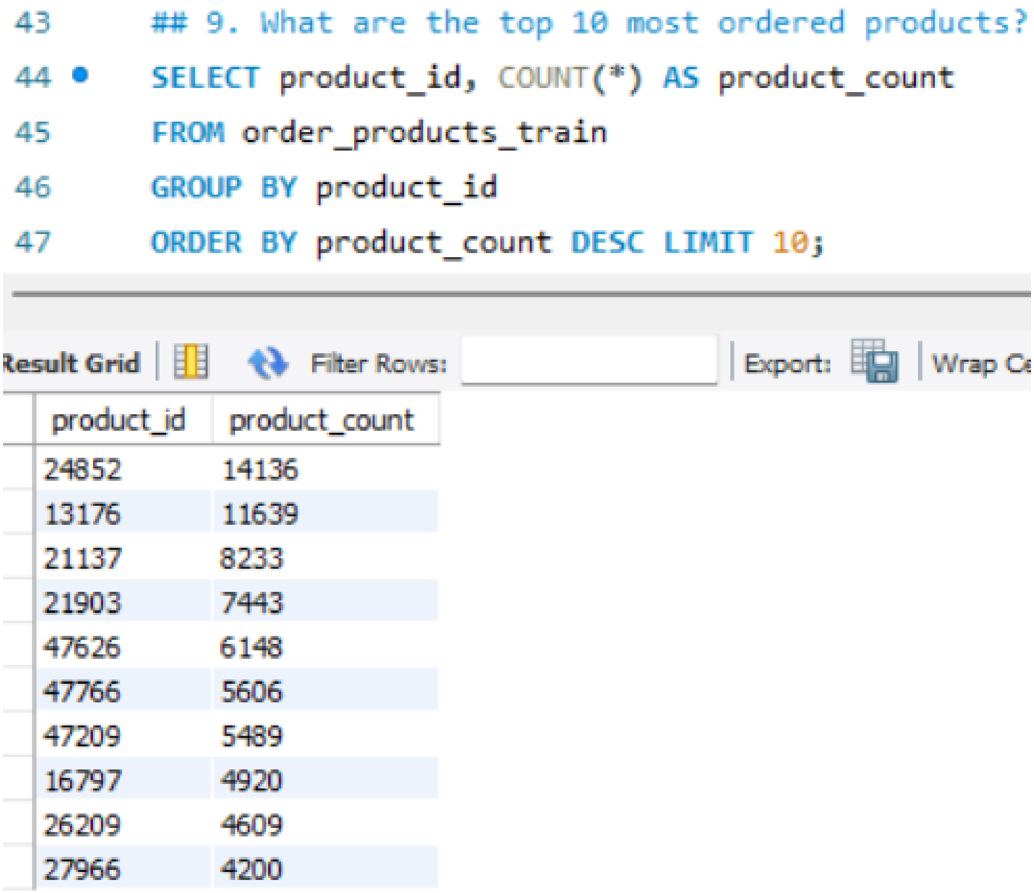
Statement Explanation: The above SELECT Statement retrieves count orders for each hour (order_hour_of_day). Sort by order count in descending order as shown in the result.



```sql
## 8. How does order volume vary by day of the week?
SELECT order_dow, count(*) AS order_count
FROM orders
GROUP BY order_dow
ORDER BY order_count DESC;
```

order_dow	order_count
0	183939
1	180025
2	143162
5	139183
6	138060
3	133839
4	130367

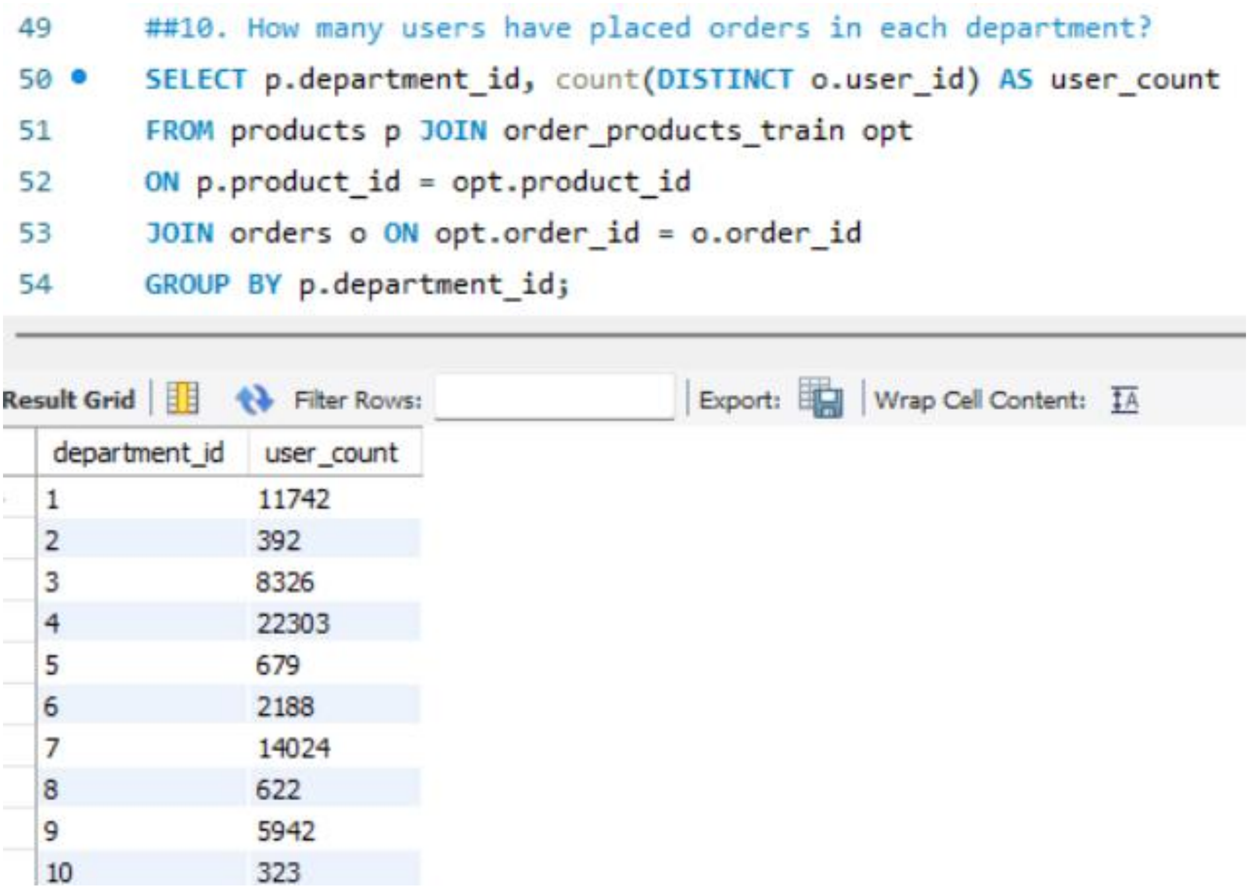
Statement Explanation: The above SELECT Statement retrieves count orders for each order_dow. Sort by order count in descending order as shown in the result.



```sql
## 9. What are the top 10 most ordered products?
SELECT product_id, COUNT(*) AS product_count
FROM order_products_train
GROUP BY product_id
ORDER BY product_count DESC LIMIT 10;
```

product_id	product_count
24852	14136
13176	11639
21137	8233
21903	7443
47626	6148
47766	5606
47209	5489
16797	4920
26209	4609
27966	4200

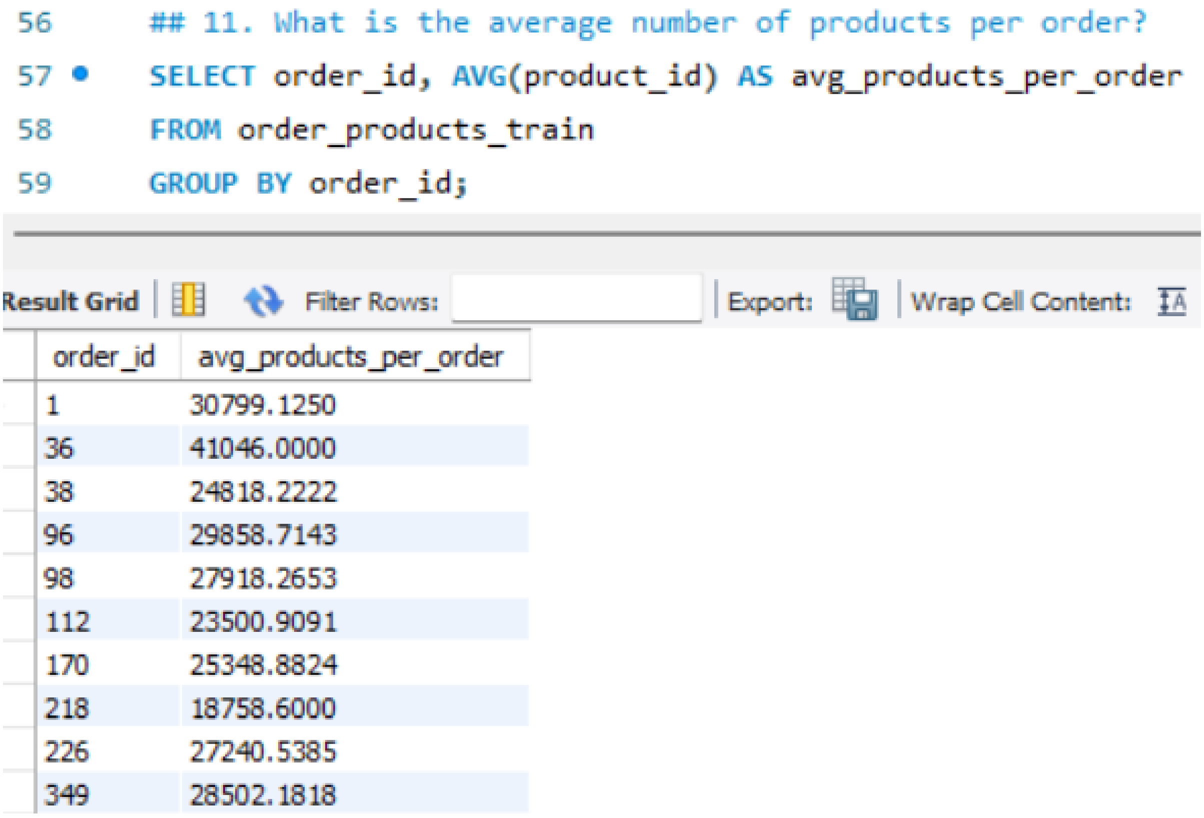
Statement Explanation: The above SELECT Statement retrieves count orders for each product_id. Sort by order count and take the top 10 as shown in the result.



```sql
##10. How many users have placed orders in each department?
SELECT p.department_id, count(DISTINCT o.user_id) AS user_count
FROM products p JOIN order_products_train opt
ON p.product_id = opt.product_id
JOIN orders o ON opt.order_id = o.order_id
GROUP BY p.department_id;
```

department_id	user_count
1	11742
2	392
3	8326
4	22303
5	679
6	2188
7	14024
8	622
9	5942
10	323

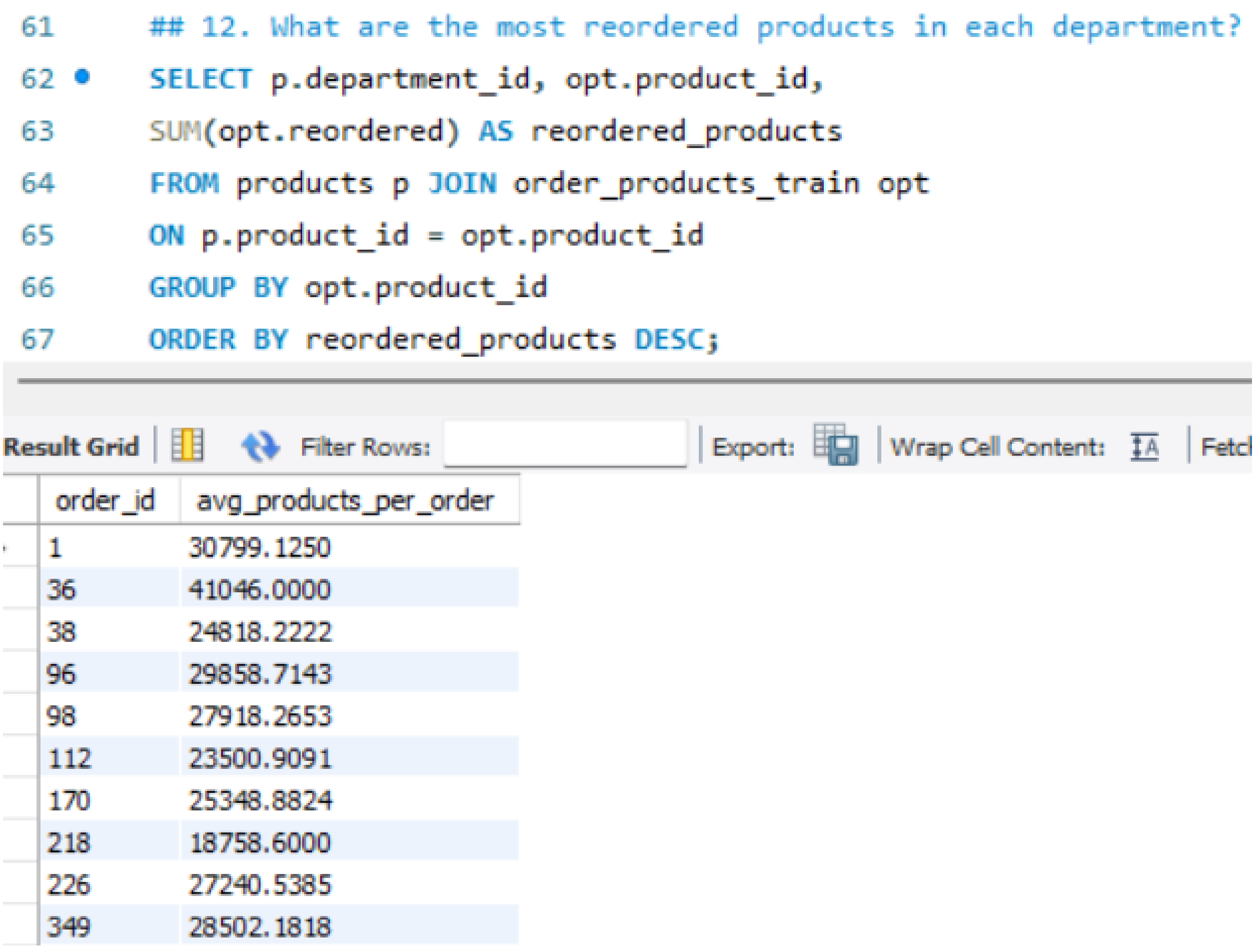
Statement Explanation: The above SELECT Statement retrieves join of products from order_products_train, and orders tables. Group by department_id and count unique user_id as shown in the result.



```sql
## 11. What is the average number of products per order?
SELECT order_id, AVG(product_id) AS avg_products_per_order
FROM order_products_train
GROUP BY order_id;
```

order_id	avg_products_per_order
1	30799.1250
36	41046.0000
38	24818.2222
96	29858.7143
98	27918.2653
112	23500.9091
170	25348.8824
218	18758.6000
226	27240.5385
349	28502.1818

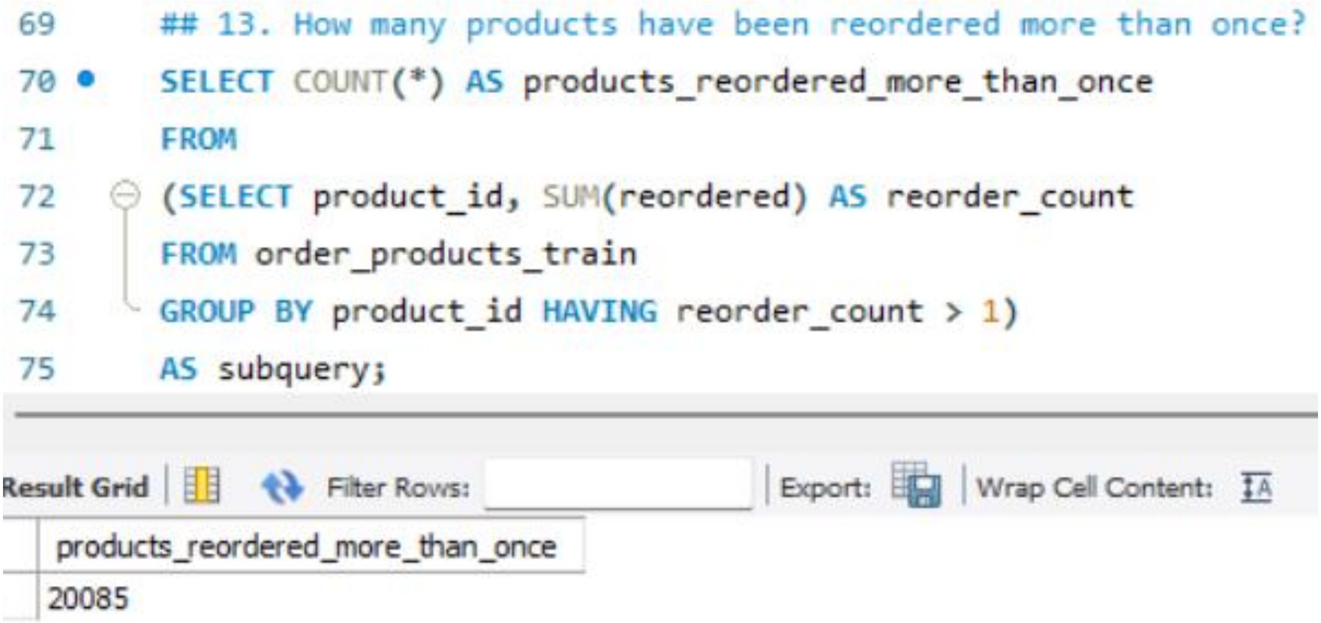
Statement Explanation: The above SELECT Statement retrieves group by order_id and compute the average number of products as shown in the result.



```sql
## 12. What are the most reordered products in each department?
SELECT p.department_id, opt.product_id,
SUM(opt.reordered) AS reordered_products
FROM products p JOIN order_products_train opt
ON p.product_id = opt.product_id
GROUP BY opt.product_id
ORDER BY reordered_products DESC;
```

order_id	avg_products_per_order
1	30799.1250
36	41046.0000
38	24818.2222
96	29858.7143
98	27918.2653
112	23500.9091
170	25348.8824
218	18758.6000
226	27240.5385
349	28502.1818

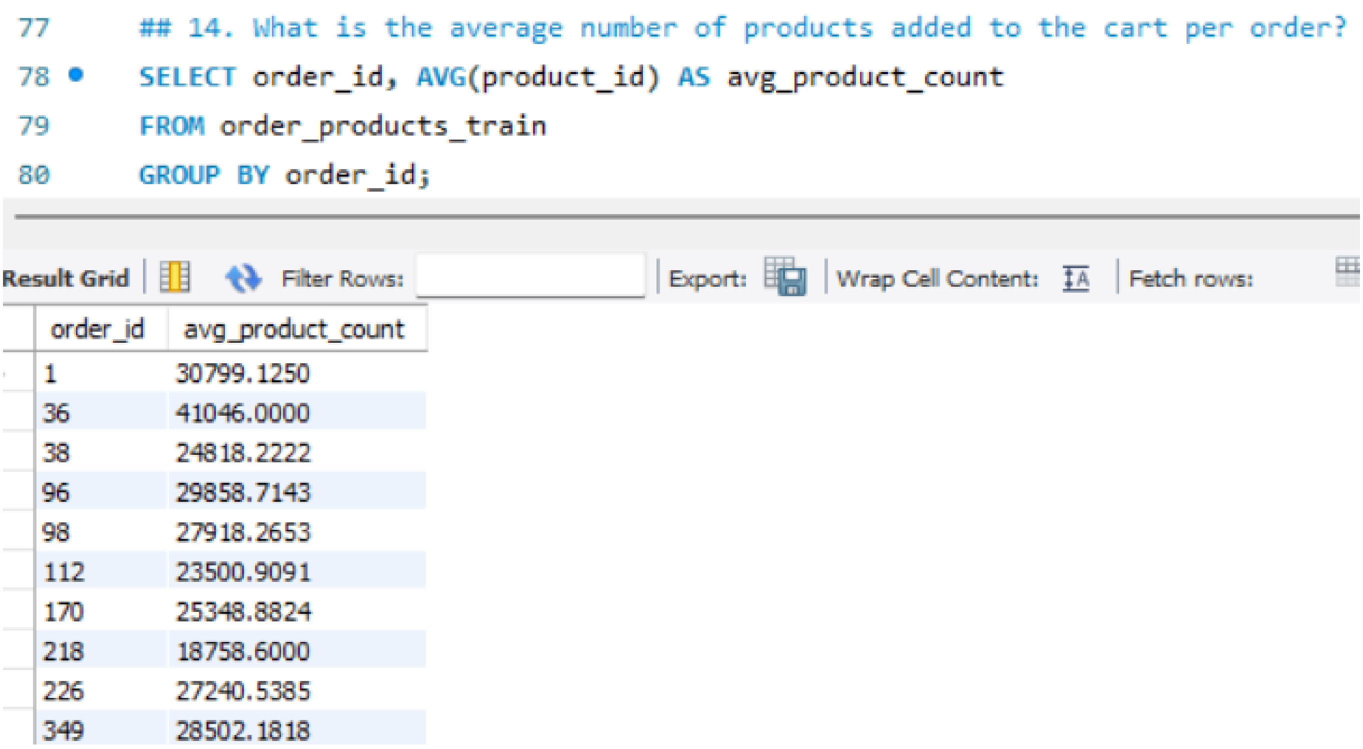
Statement Explanation: The above SELECT Statement retrieves join of products and order_products_train, group by department_id and product_id, and sum reorders (SUM(opt.reordered)) as shown in the result.



```sql
## 13. How many products have been reordered more than once?
SELECT COUNT(*) AS products_reordered_more_than_once
FROM
(SELECT product_id, SUM(reordered) AS reorder_count
FROM order_products_train
GROUP BY product_id HAVING reorder_count > 1)
AS subquery;
```

products_reordered_more_than_once
20085

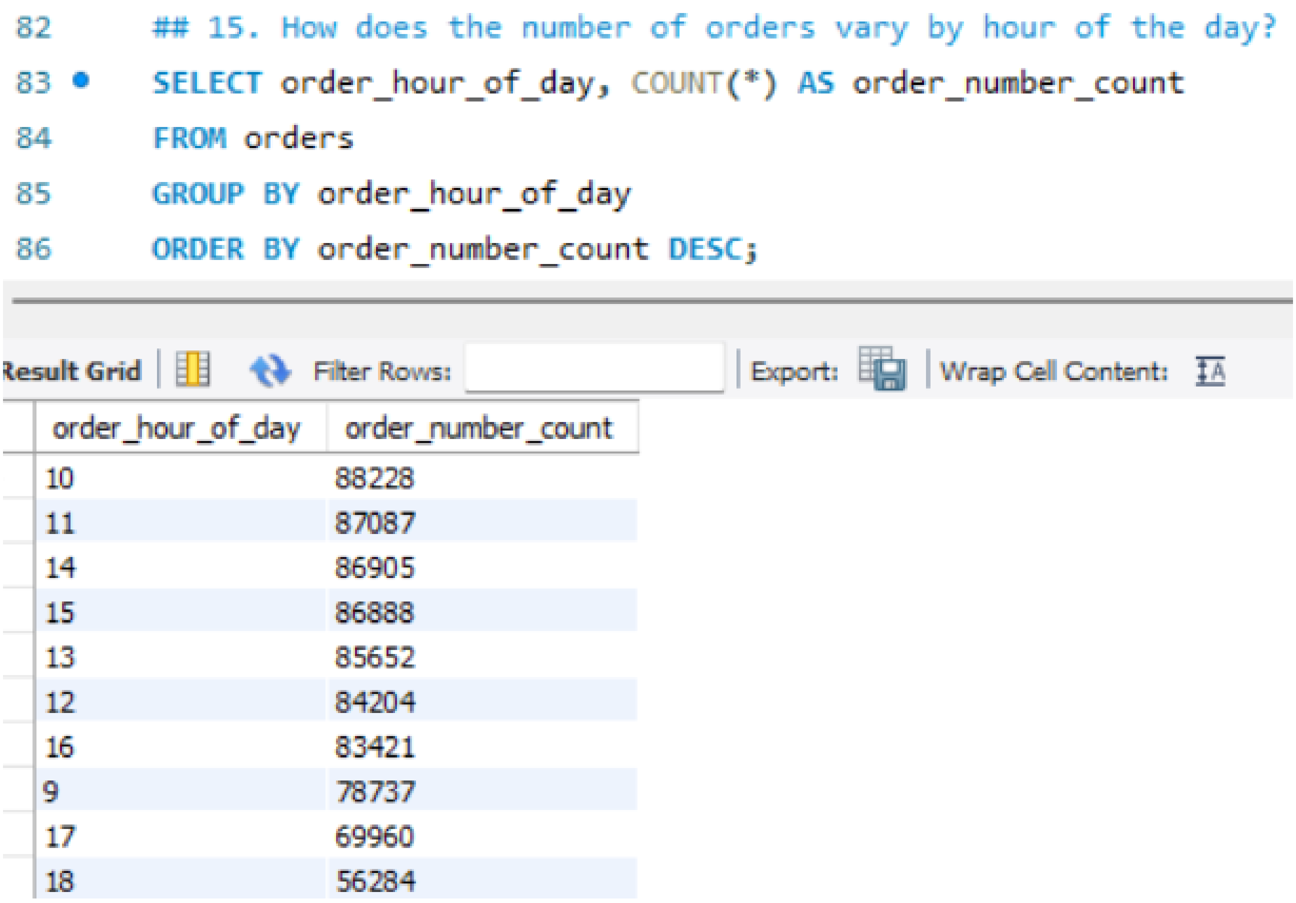
Statement Explanation: The above SELECT Statement retrieves filtered products with SUM(reordered) > 1 and count them as shown in the result.



```sql
## 14. What is the average number of products added to the cart per order?
SELECT order_id, AVG(product_id) AS avg_product_count
FROM order_products_train
GROUP BY order_id;
```

order_id	avg_product_count
1	30799.1250
36	41046.0000
38	24818.2222
96	29858.7143
98	27918.2653
112	23500.9091
170	25348.8824
218	18758.6000
226	27240.5385
349	28502.1818

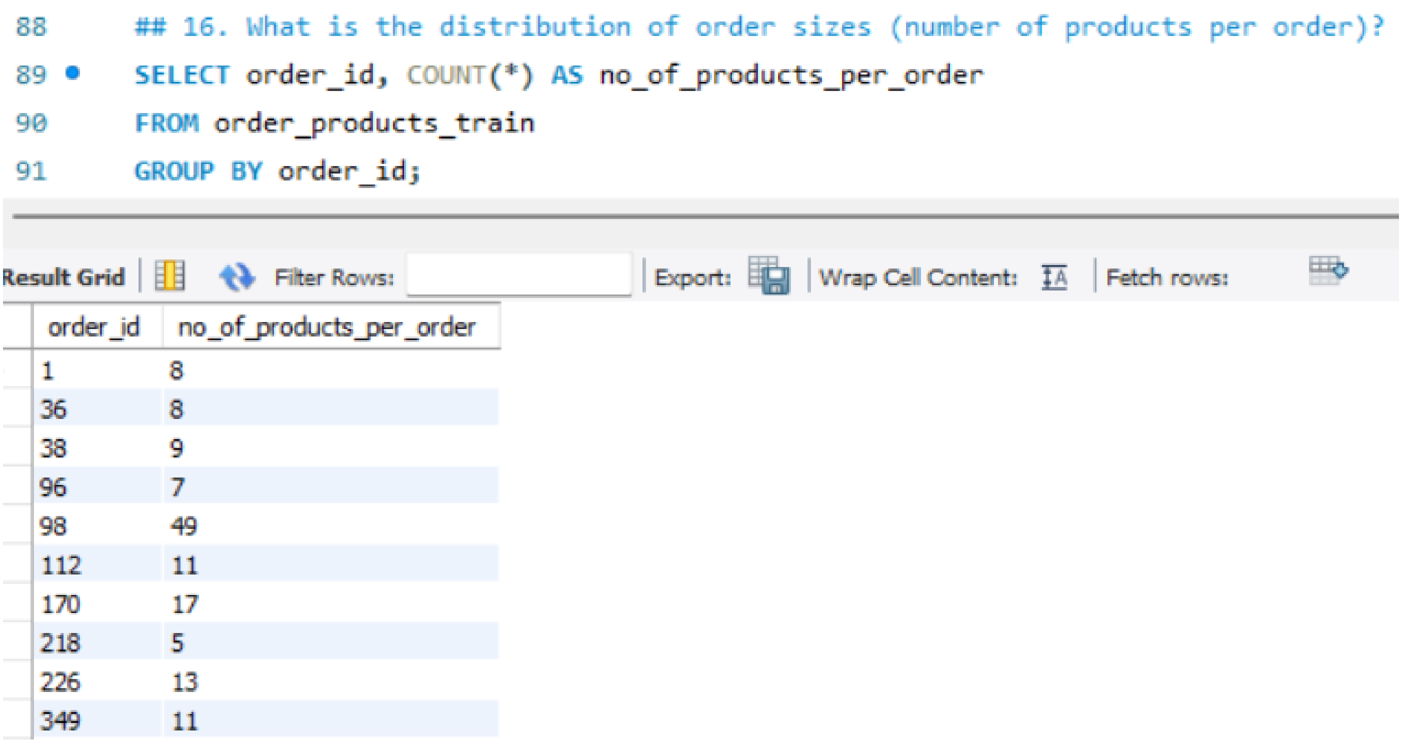
Statement Explanation: The above SELECT Statement retrieves group by order_id and compute the average number of products as shown in the result.



```sql
## 15. How does the number of orders vary by hour of the day?
SELECT order_hour_of_day, COUNT(*) AS order_number_count
FROM orders
GROUP BY order_hour_of_day
ORDER BY order_number_count DESC;
```

order_hour_of_day	order_number_count
10	88228
11	87087
14	86905
15	86888
13	85652
12	84204
16	83421
9	78737
17	69960
18	56284

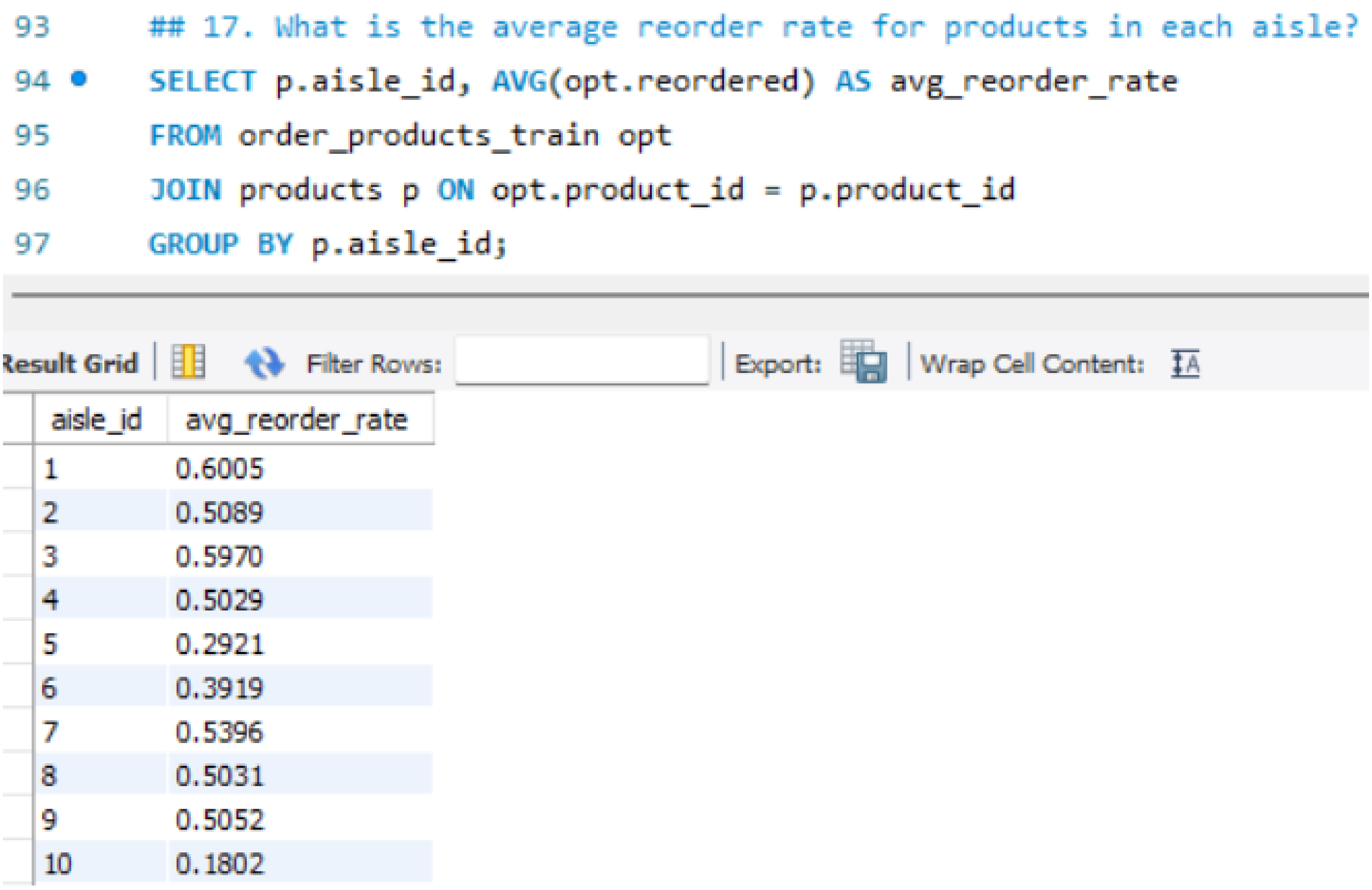
Statement Explanation: The above SELECT Statement retrieves count orders grouped by order_hour_of_day as shown in the result.



```
## 16. What is the distribution of order sizes (number of products per order)?
SELECT order_id, COUNT(*) AS no_of_products_per_order
FROM order_products_train
GROUP BY order_id;
```

order_id	no_of_products_per_order
1	8
36	8
38	9
96	7
98	49
112	11
170	17
218	5
226	13
349	11

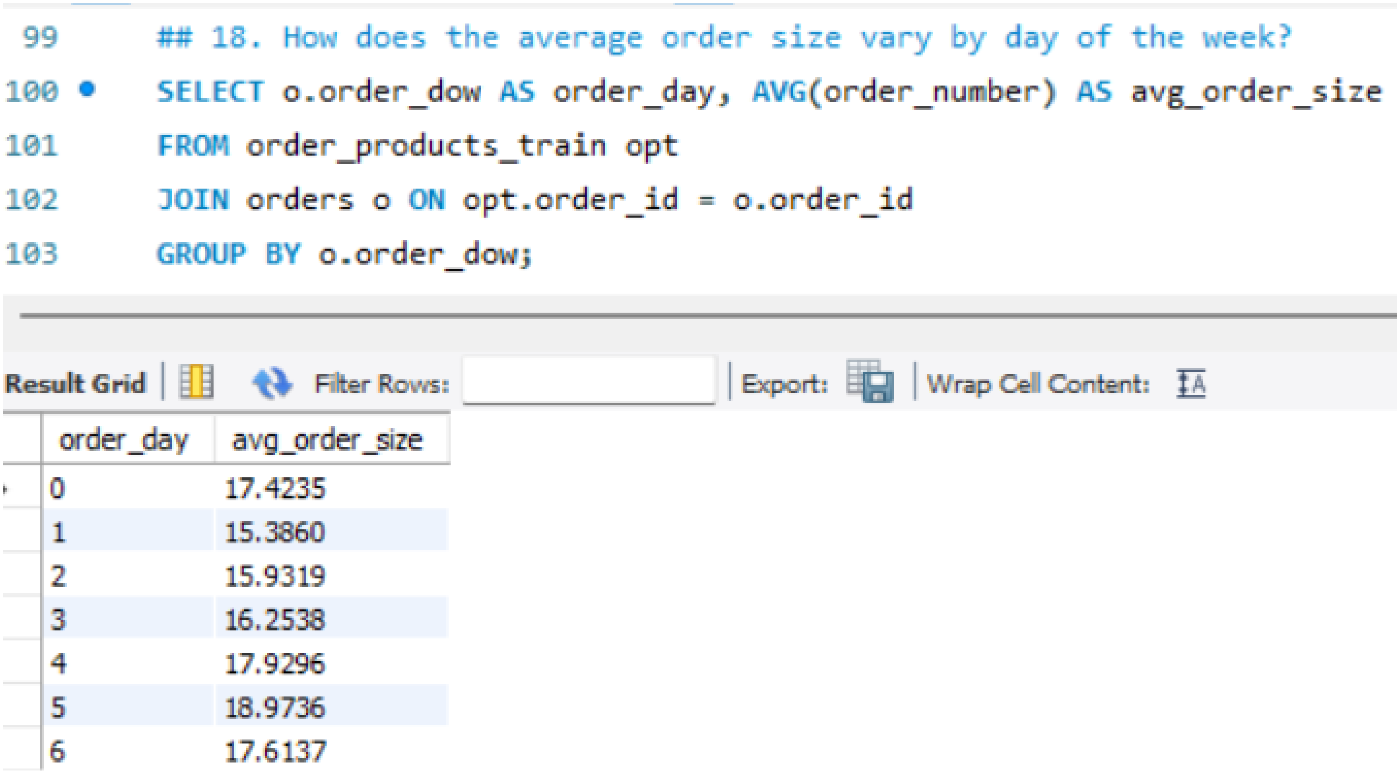
Statement Explanation: The above SELECT Statement retrieves count products grouped by order_id as shown in the result.



```
## 17. What is the average reorder rate for products in each aisle?
SELECT p.aisle_id, AVG(opt.reordered) AS avg_reorder_rate
FROM order_products_train opt
JOIN products p ON opt.product_id = p.product_id
GROUP BY p.aisle_id;
```

aisle_id	avg_reorder_rate
1	0.6005
2	0.5089
3	0.5970
4	0.5029
5	0.2921
6	0.3919
7	0.5396
8	0.5031
9	0.5052
10	0.1802

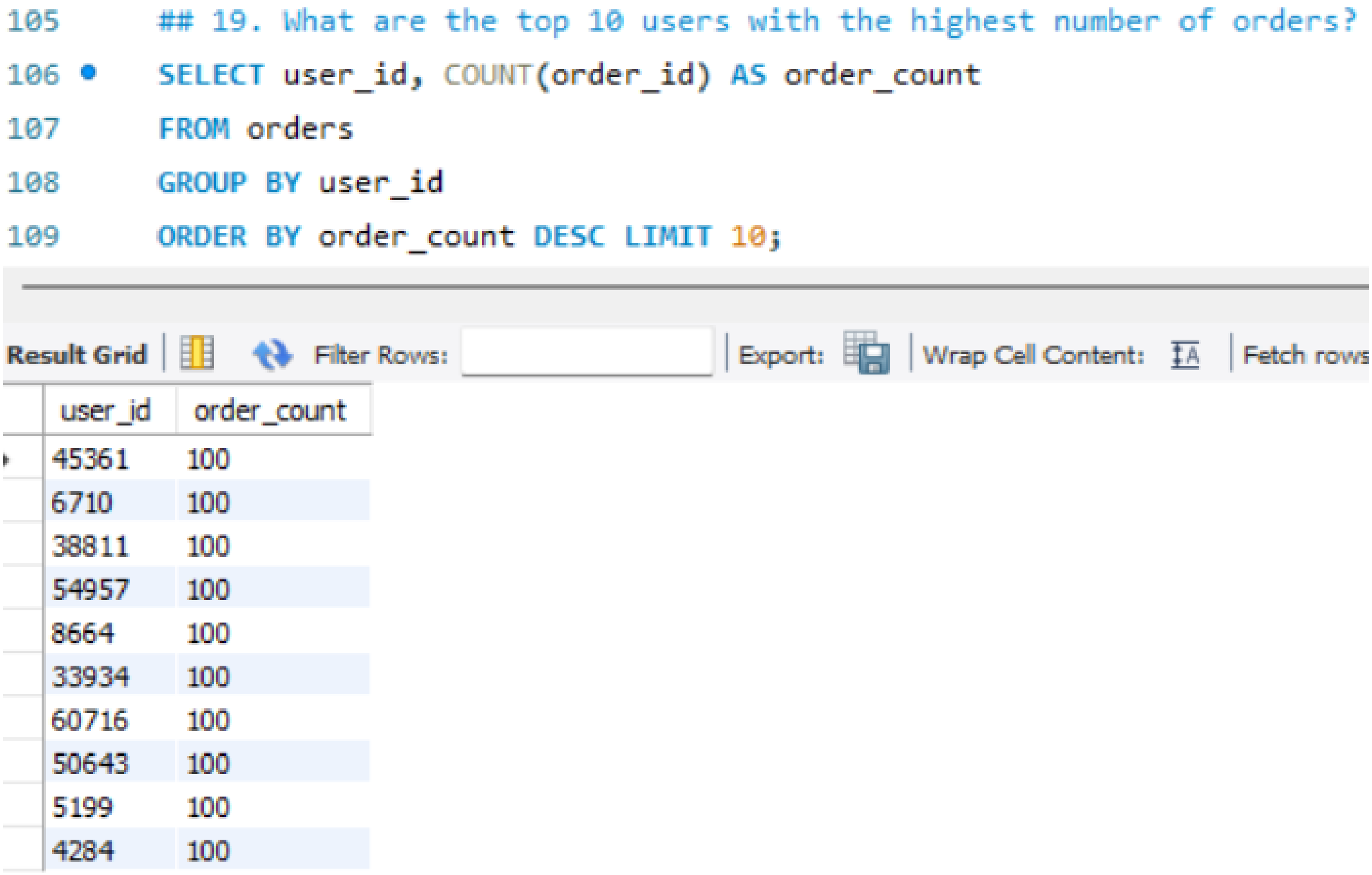
Statement Explanation: The above SELECT Statement retrieves join of products and order_products_train. Group by aisle_id and compute AVG(reordered) as shown in the result.



```sql
## 18. How does the average order size vary by day of the week?
SELECT o.order_dow AS order_day, AVG(order_number) AS avg_order_size
FROM order_products_train opt
JOIN orders o ON opt.order_id = o.order_id
GROUP BY o.order_dow;
```

order_day	avg_order_size
0	17.4235
1	15.3860
2	15.9319
3	16.2538
4	17.9296
5	18.9736
6	17.6137

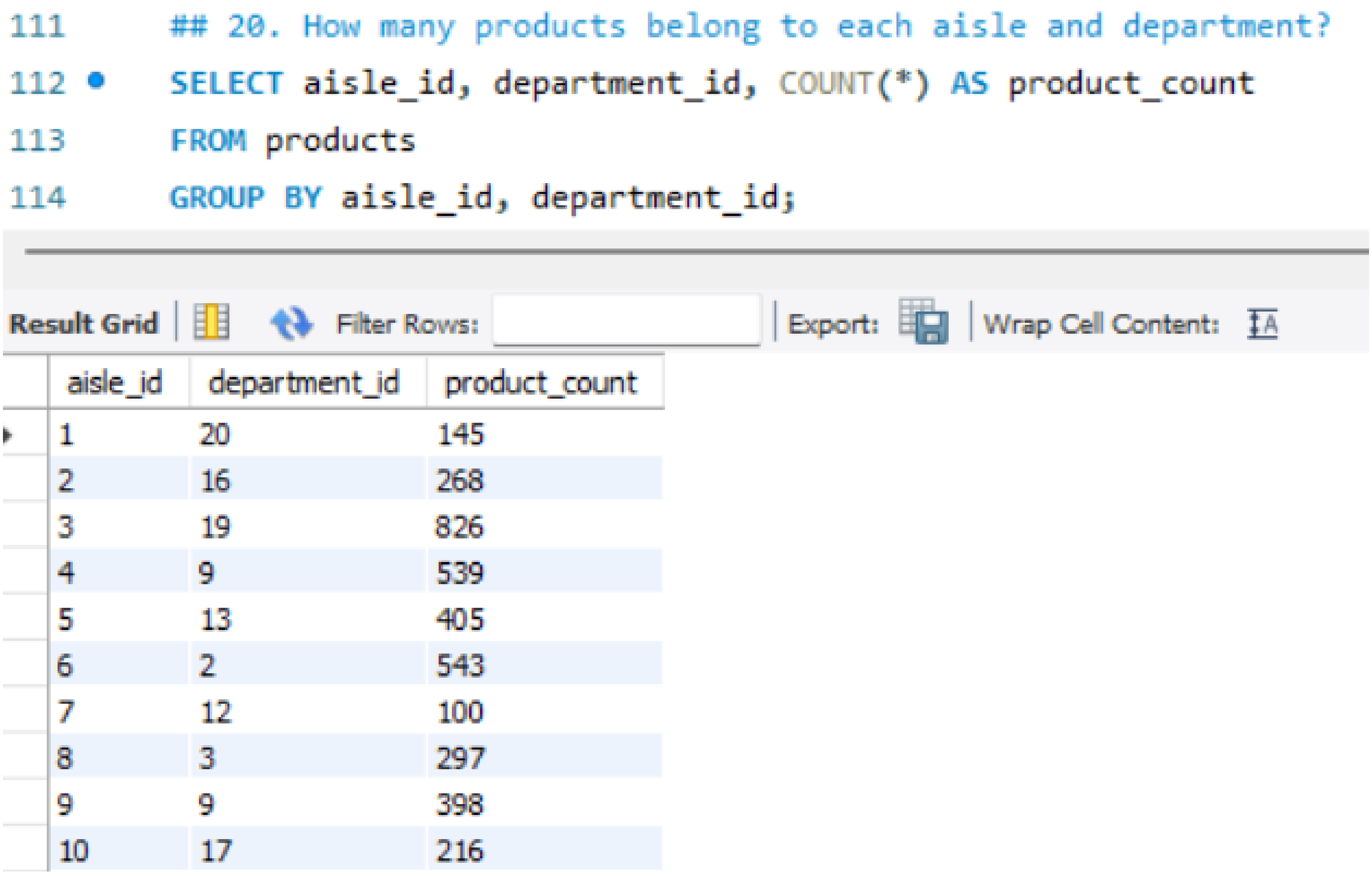
Statement Explanation: The above SELECT Statement retrieves join of orders and order_products_train, group by order_dow, and compute the average order size as shown in the result.



```sql
## 19. What are the top 10 users with the highest number of orders?
SELECT user_id, COUNT(order_id) AS order_count
FROM orders
GROUP BY user_id
ORDER BY order_count DESC LIMIT 10;
```

user_id	order_count
45361	100
6710	100
38811	100
54957	100
8664	100
33934	100
60716	100
50643	100
5199	100
4284	100

Statement Explanation: The above SELECT Statement retrieves group by user_id and count orders. Sort and take the top 10 as shown in the result.



```sql
## 20. How many products belong to each aisle and department?
SELECT aisle_id, department_id, COUNT(*) AS product_count
FROM products
GROUP BY aisle_id, department_id;
```

aisle_id	department_id	product_count
1	20	145
2	16	268
3	19	826
4	9	539
5	13	405
6	2	543
7	12	100
8	3	297
9	9	398
10	17	216

Statement Explanation: The above SELECT Statement retrieves group by aisle_id and department_id and count products as shown in the result.