PIZZA

PIZZA HUT SALES ANALYSIS USING SQL

By Saurav Payal

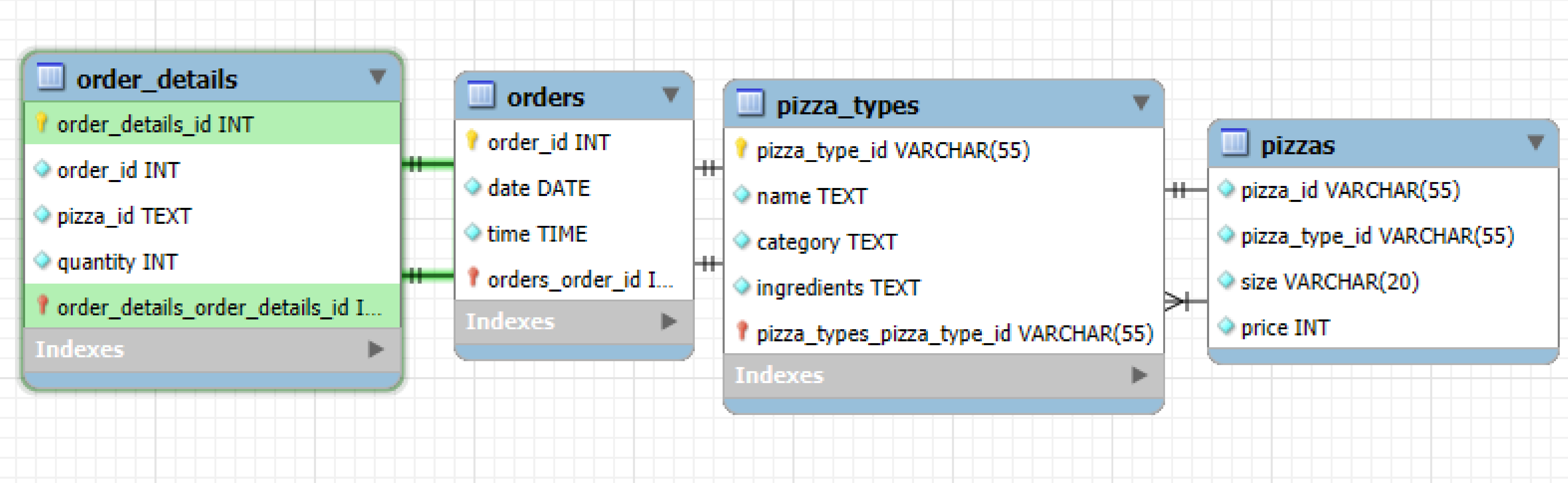
ABOUT ME

- I am a recent Mechanical Engineering graduate from Manipal University.
- I am skilled in tools such as MS Excel and MySQL,PowerBi and Python with expertise in data visualization, exploratory data analysis, and creating automated dashboards.
- My passion lies in leveraging data-driven insights to solve problems and drive business growth.
- I am now seeking opportunities in data analysis and business analysis to apply my analytical and technical skills to impactful projects.

ABOUT THE PROJECT:

- Objective: "To analyze PizzaHut sales data and derive meaningful insights using SQL."
- Key Skills: SQL Queries, Data Analysis, Problem Solving.
- Key Focus: Exploring sales performance, customer preferences, and revenue trends.

DATABASE SCHEMA OVERVIEW



order_details
order_details_id INT
order_id INT
pizza_id TEXT
quantity INT
order_details_order_details_id I...
Indexes

orders
order_id INT
date DATE
time TIME
orders_order_id I...
Indexes

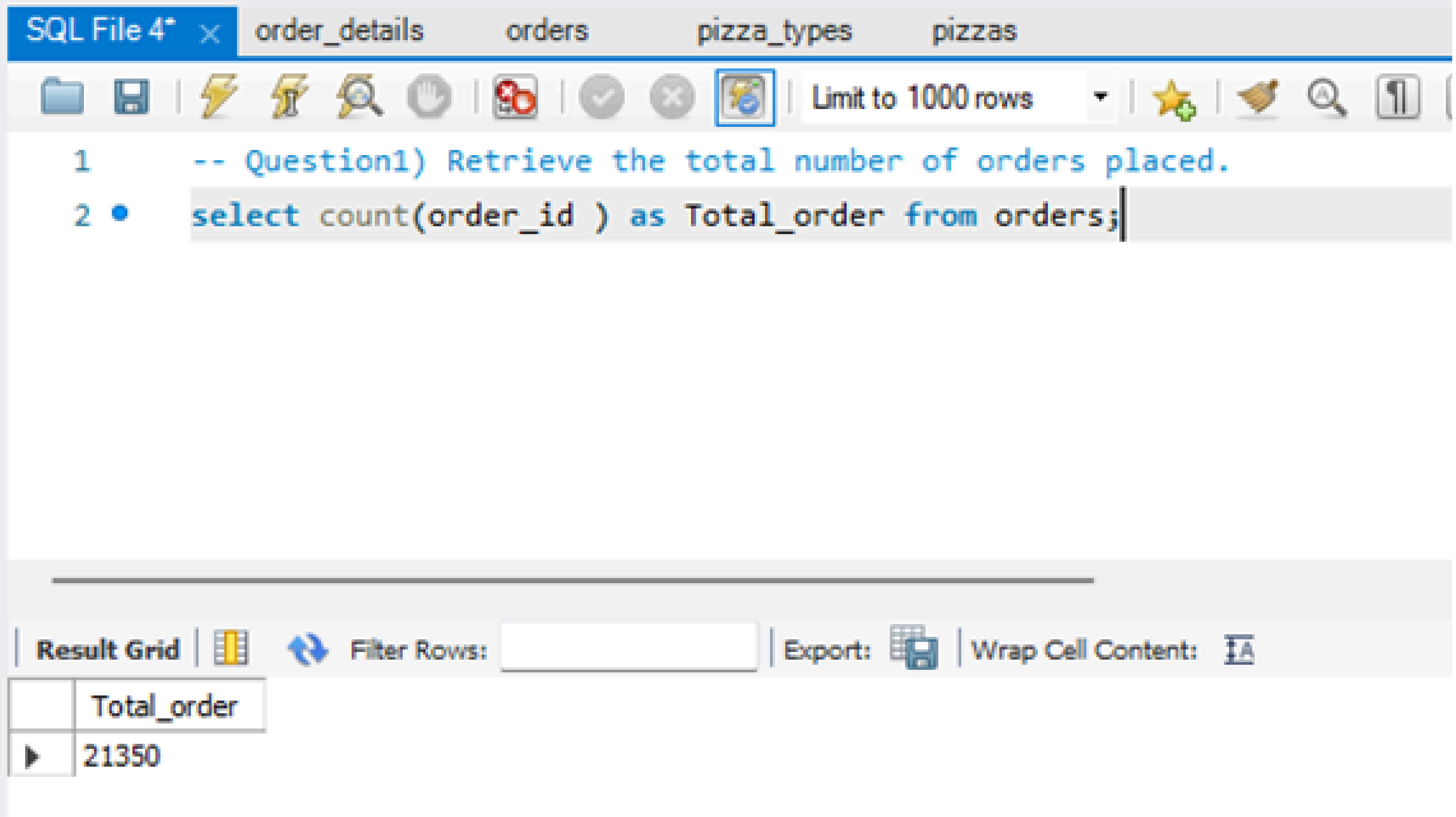
pizza_types
pizza_type_id VARCHAR(55)
name TEXT
category TEXT
ingredients TEXT
pizza_types_pizza_type_id VARCHAR(55)
Indexes

pizzas
pizza_id VARCHAR(55)
pizza_type_id VARCHAR(55)
size VARCHAR(20)
price INT

Order_details

Contains data on customer orders, including order ID, customer ID, and order date

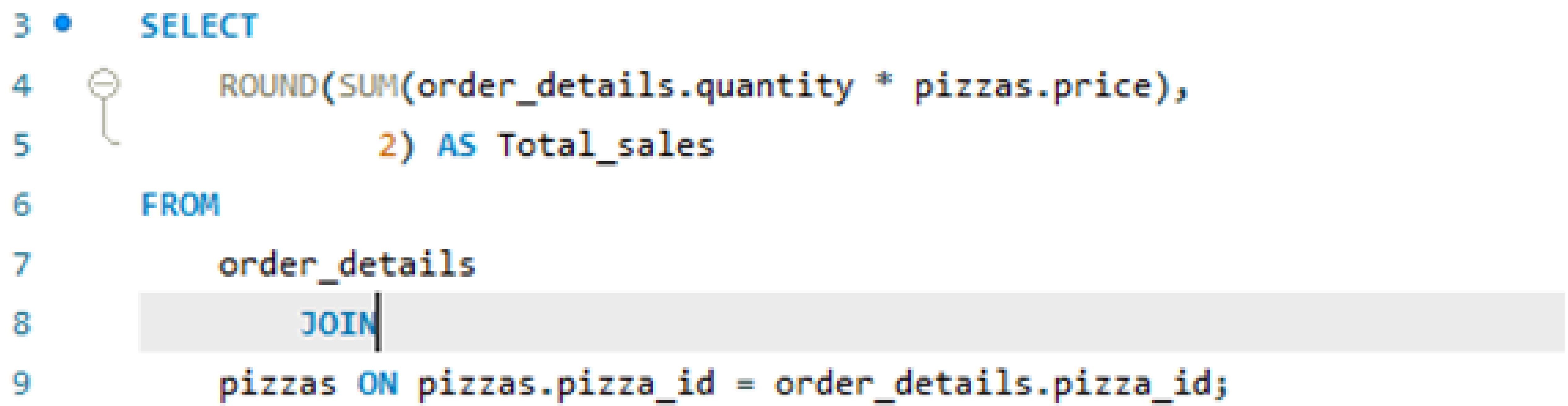
RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED



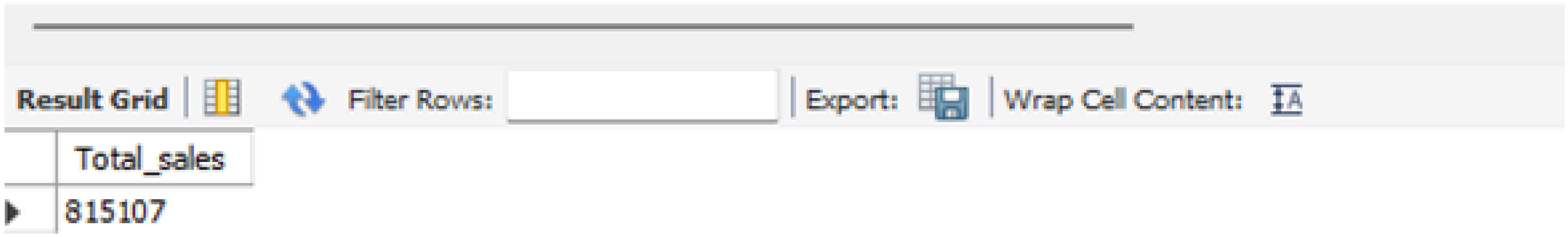
```sql
-- Question1) Retrieve the total number of orders placed.
select count(order_id ) as Total_order from orders;
```

```
Total_order
21350
```

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

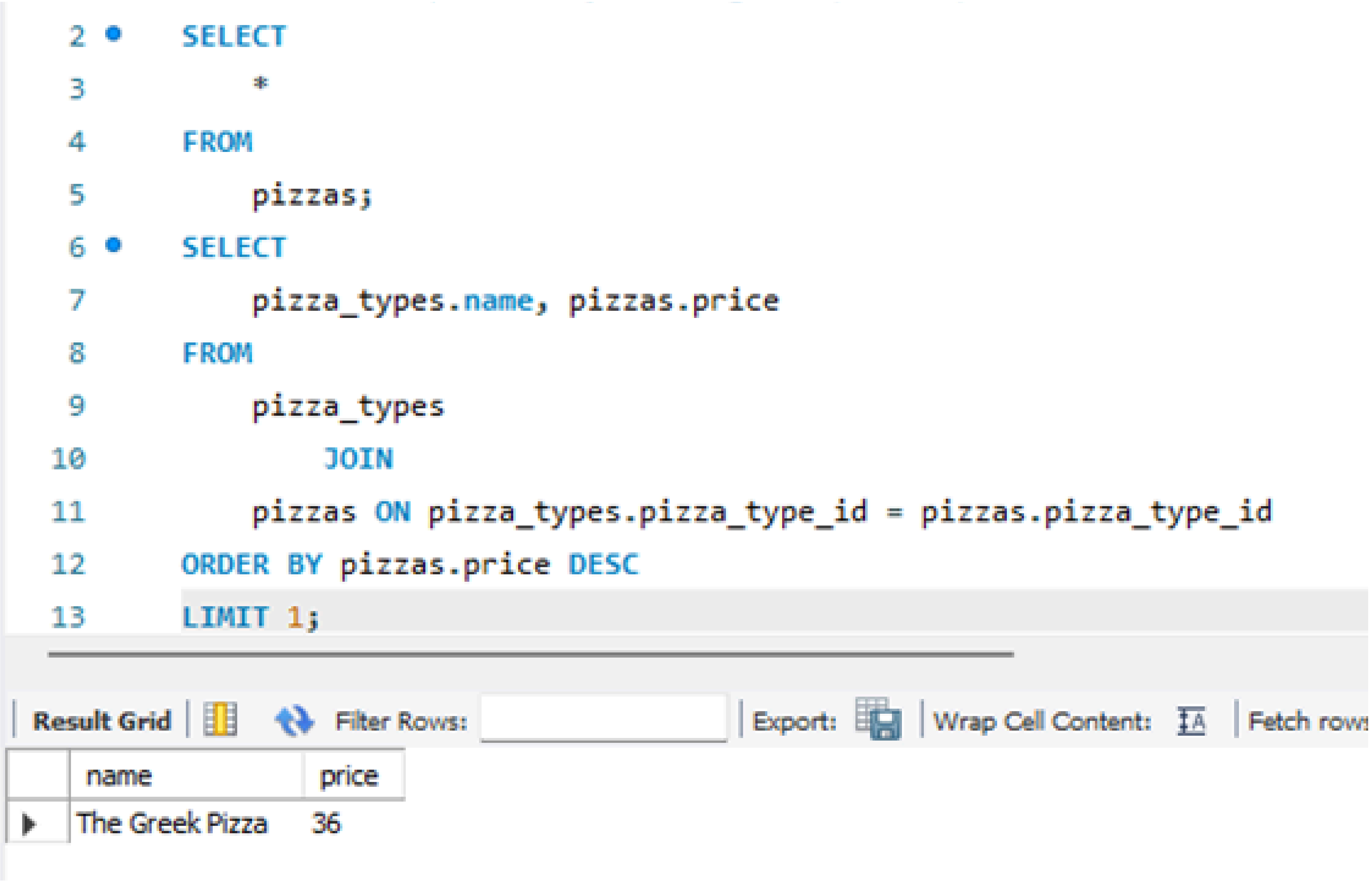


```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS Total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```



```
Total_sales
815107
```

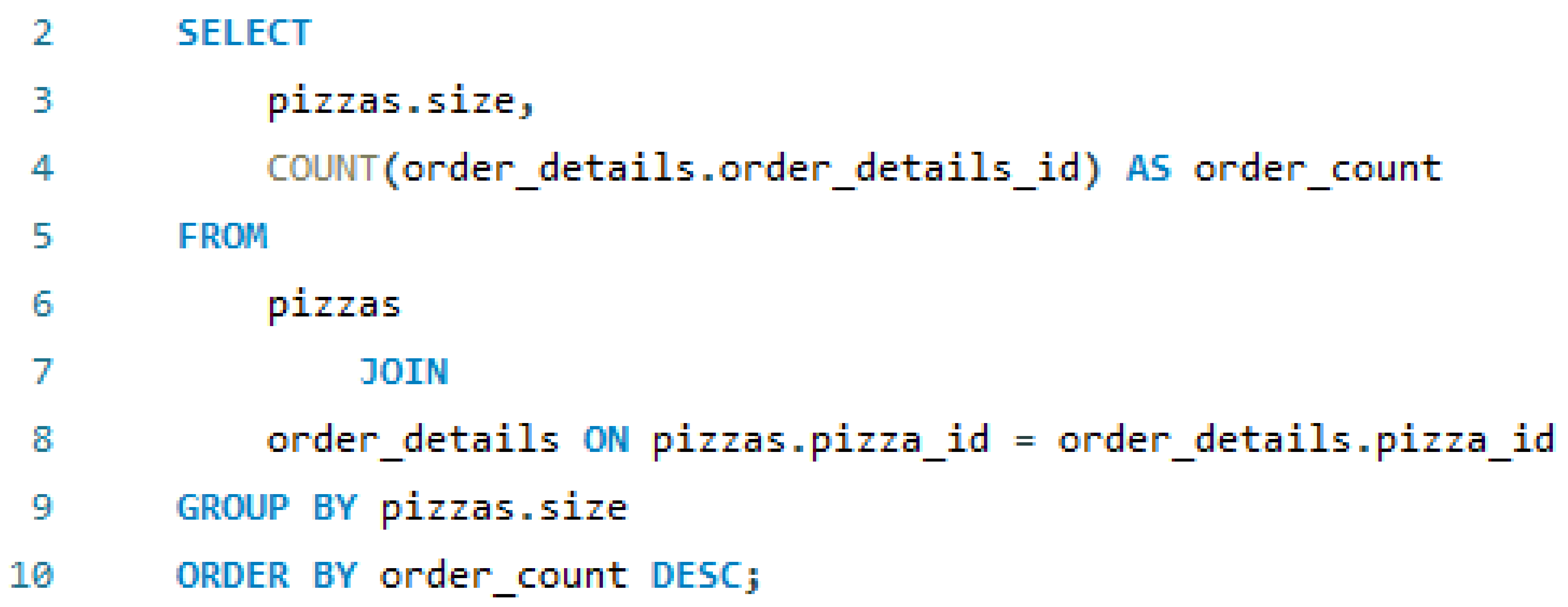
IDENTIFY THE HIGHEST-PRICED PIZZA



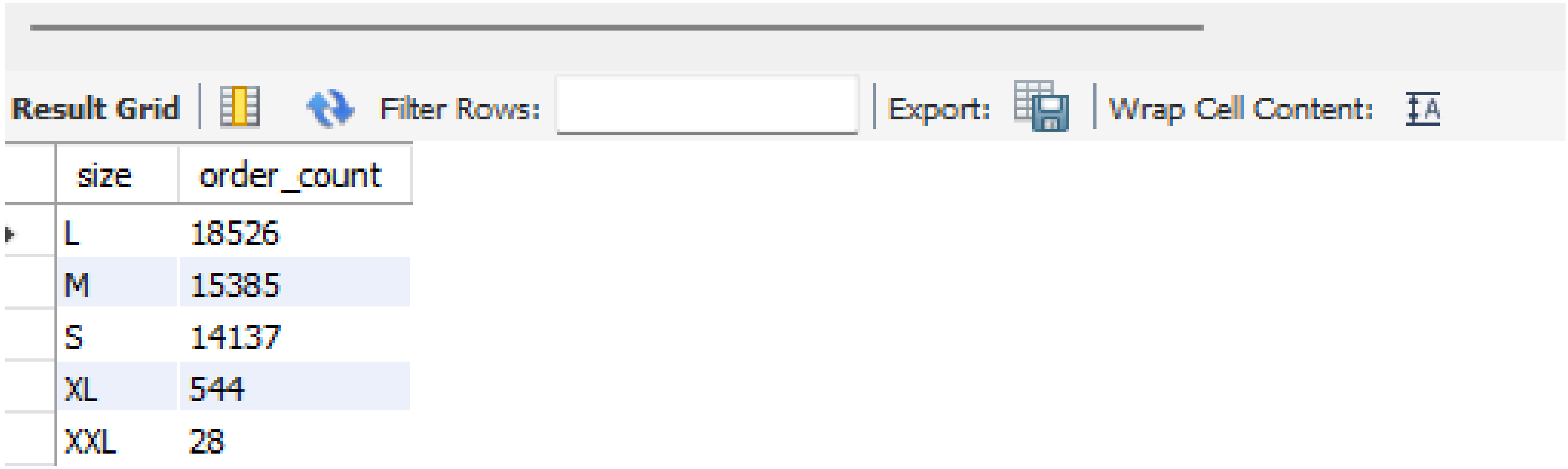
```
SELECT
    *
FROM
    pizzas;
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

name	price
The Greek Pizza	36

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

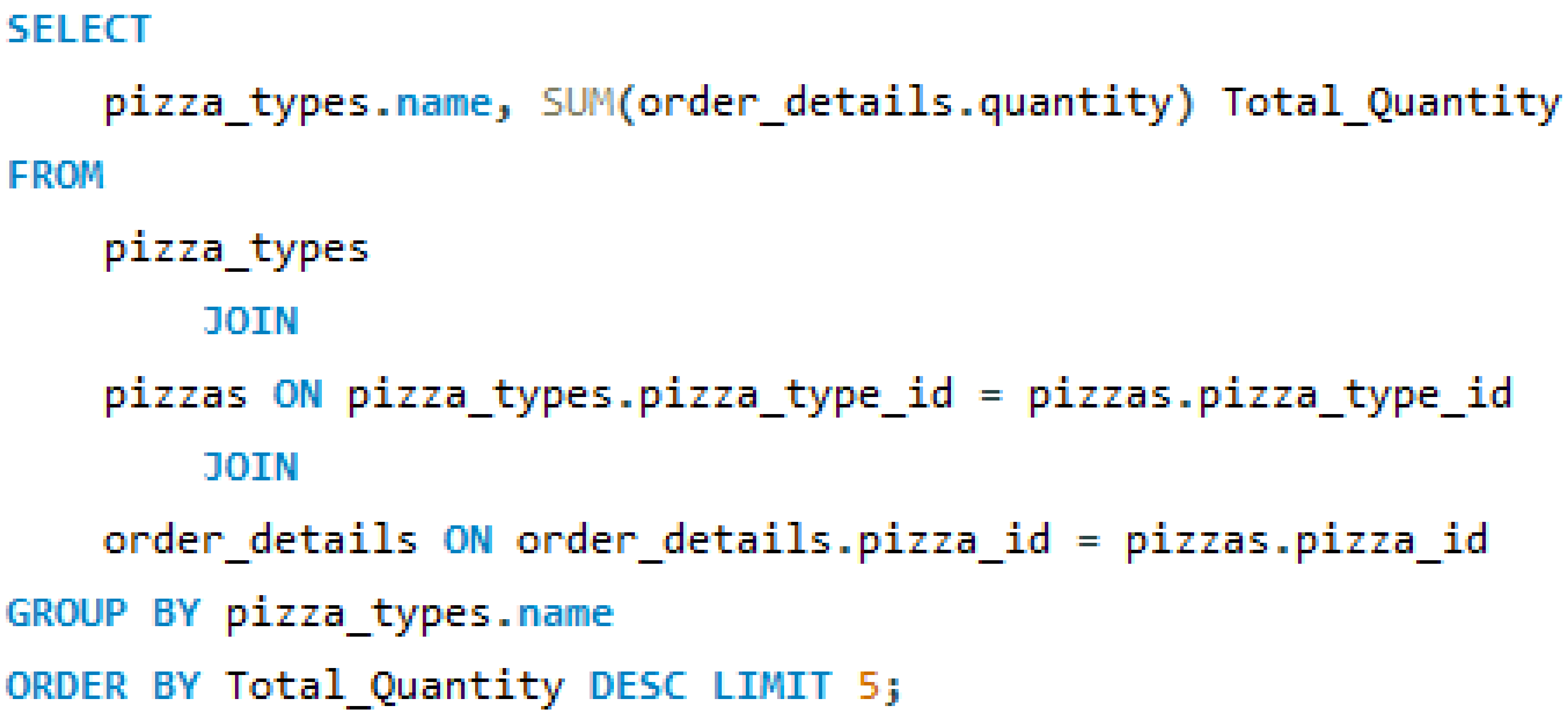


```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

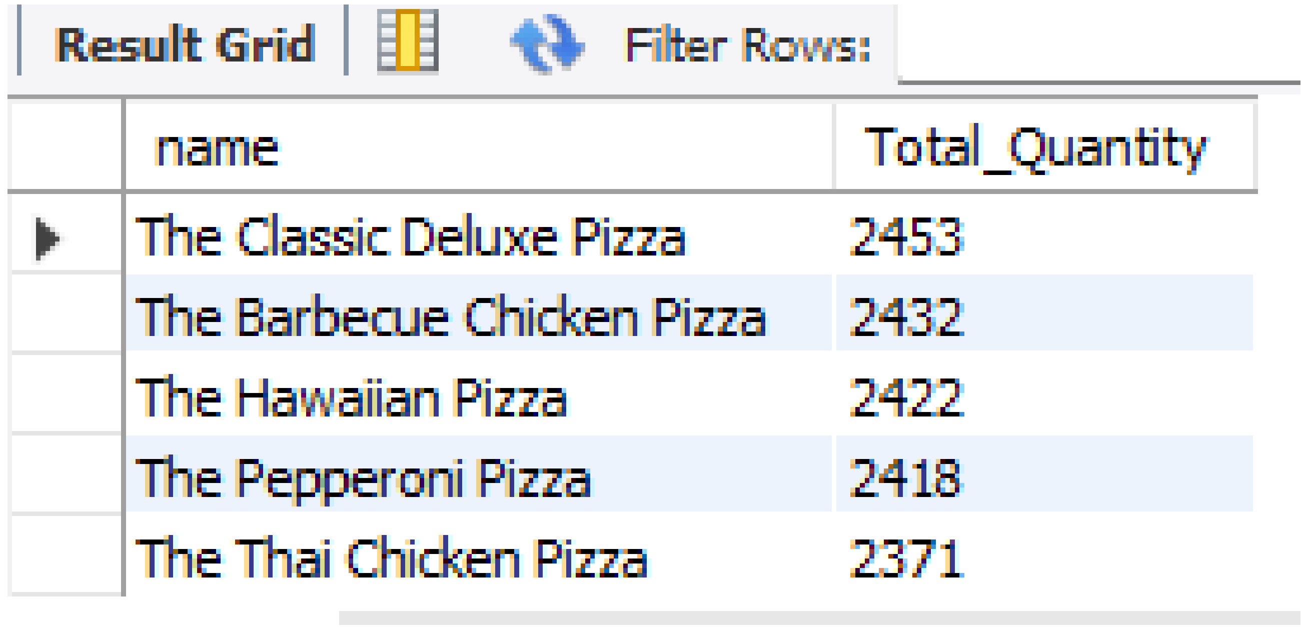


size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES

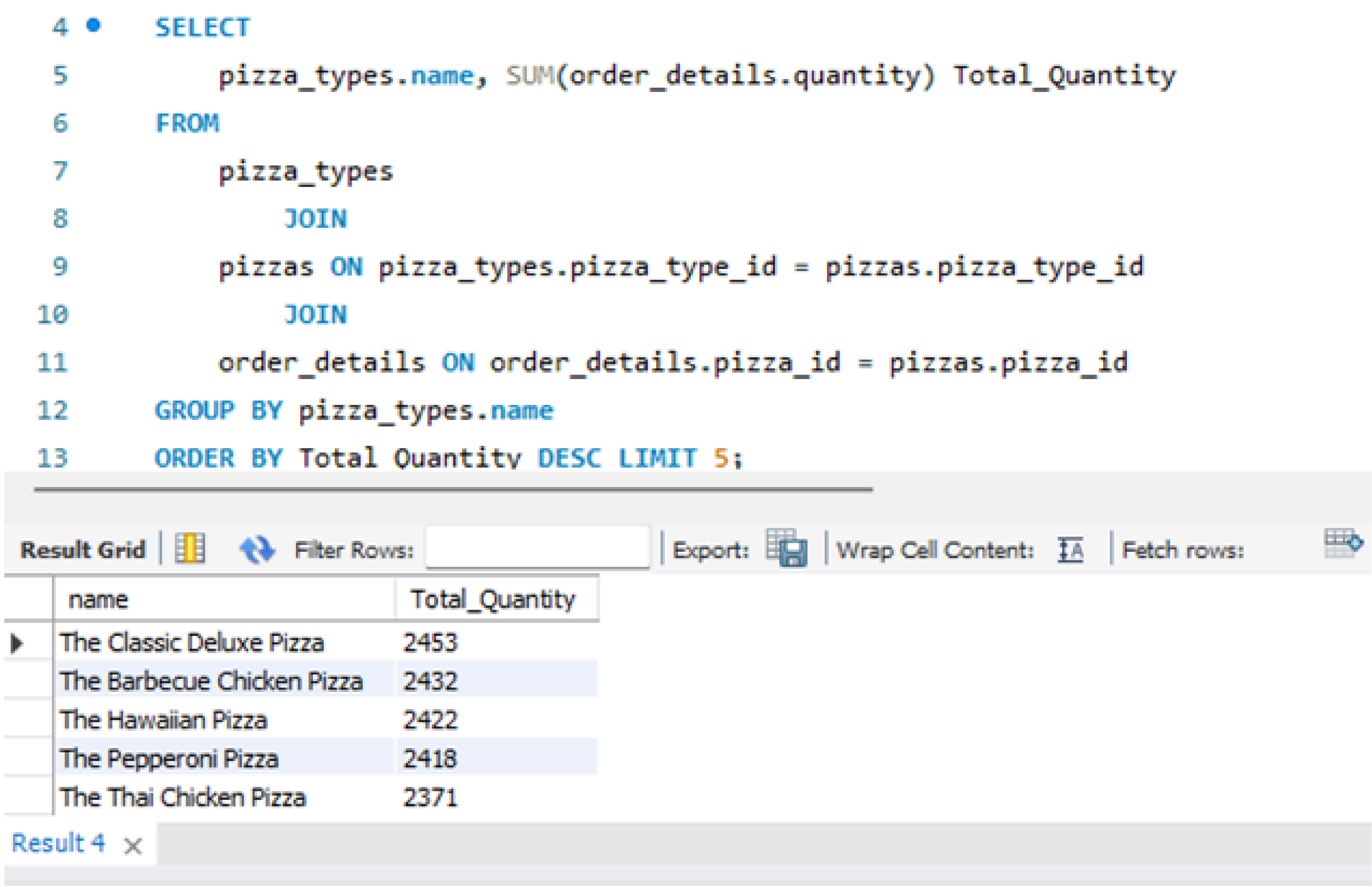


```
SELECT
    pizza_types.name, SUM(order_details.quantity) Total_Quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY Total_Quantity DESC LIMIT 5;
```



name	Total_Quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

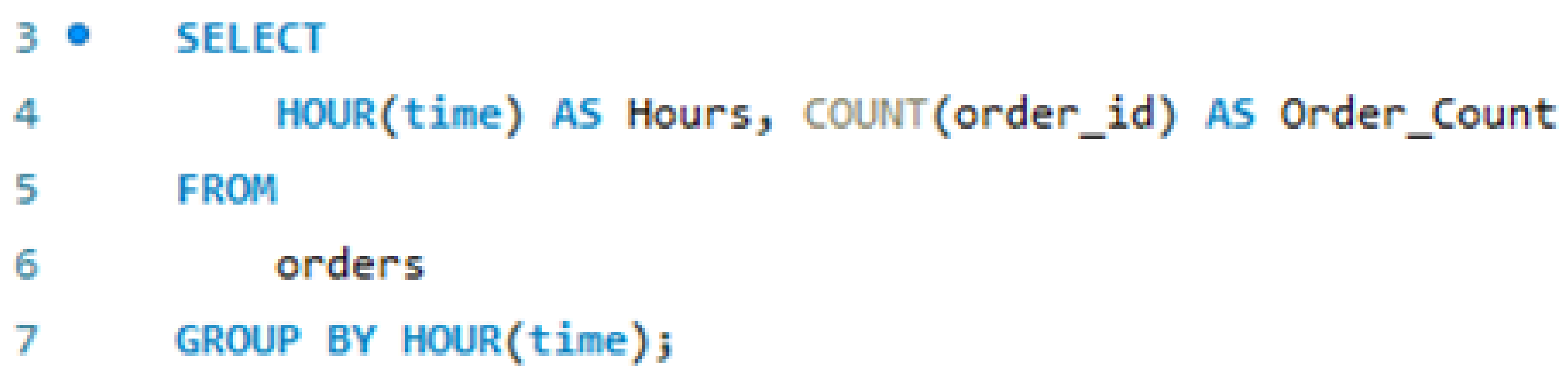
JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED



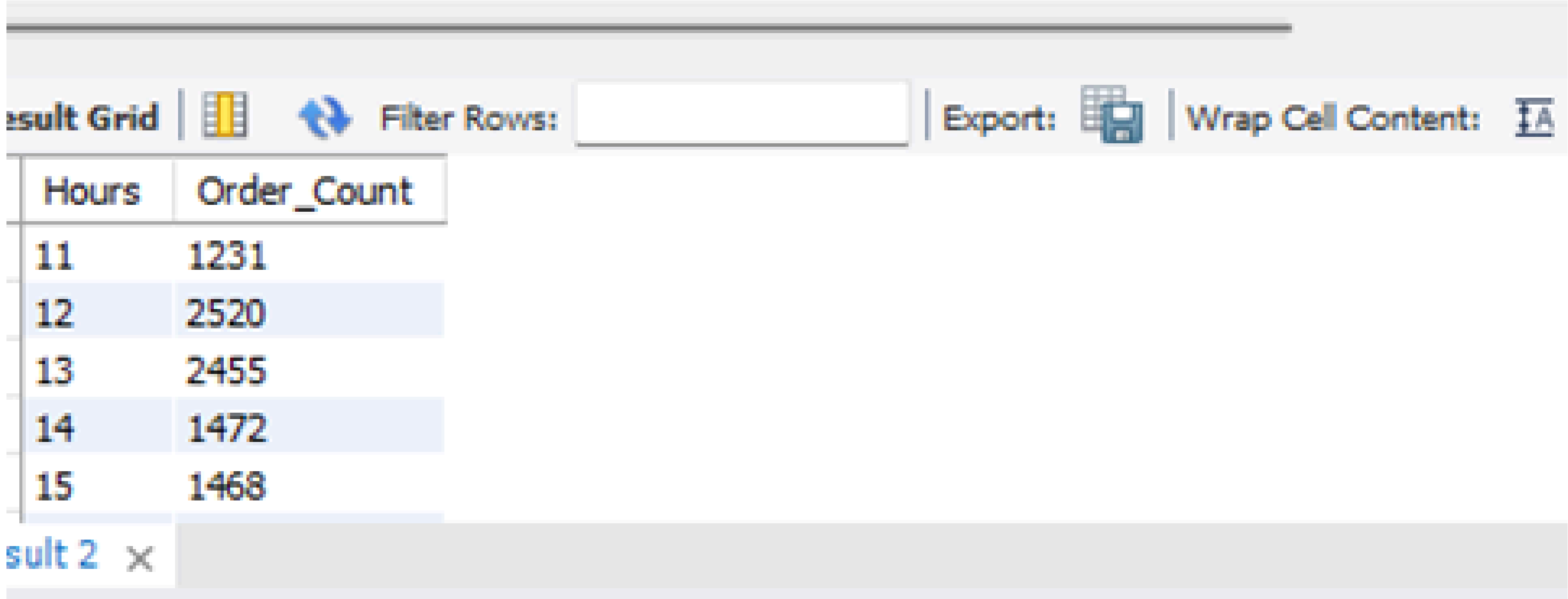
```
SELECT
    pizza_types.name, SUM(order_details.quantity) Total_Quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY Total_Quantity DESC LIMIT 5;
```

name	Total_Quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

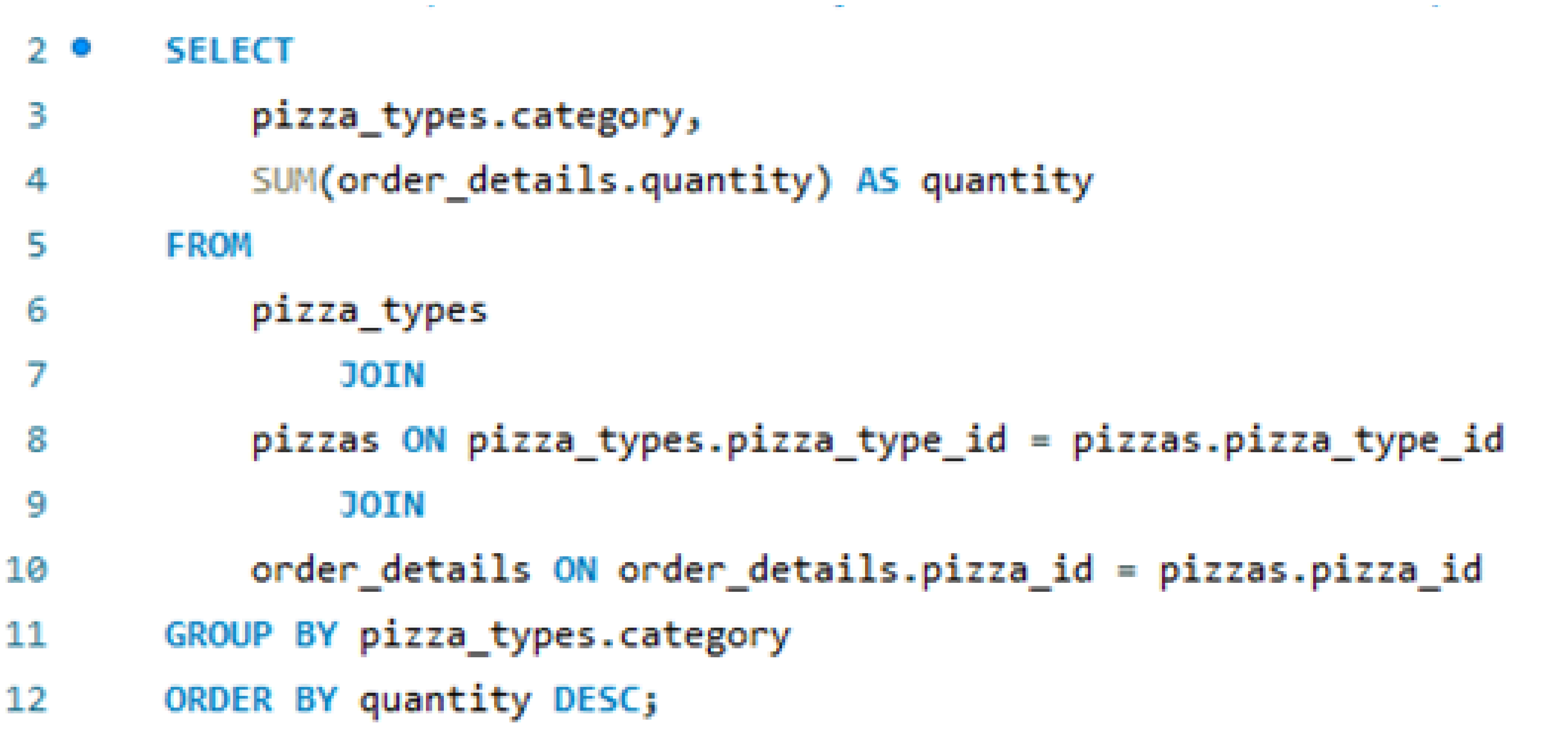


```sql
SELECT
    HOUR(time) AS Hours, COUNT(order_id) AS Order_Count
FROM
    orders
GROUP BY HOUR(time);
```

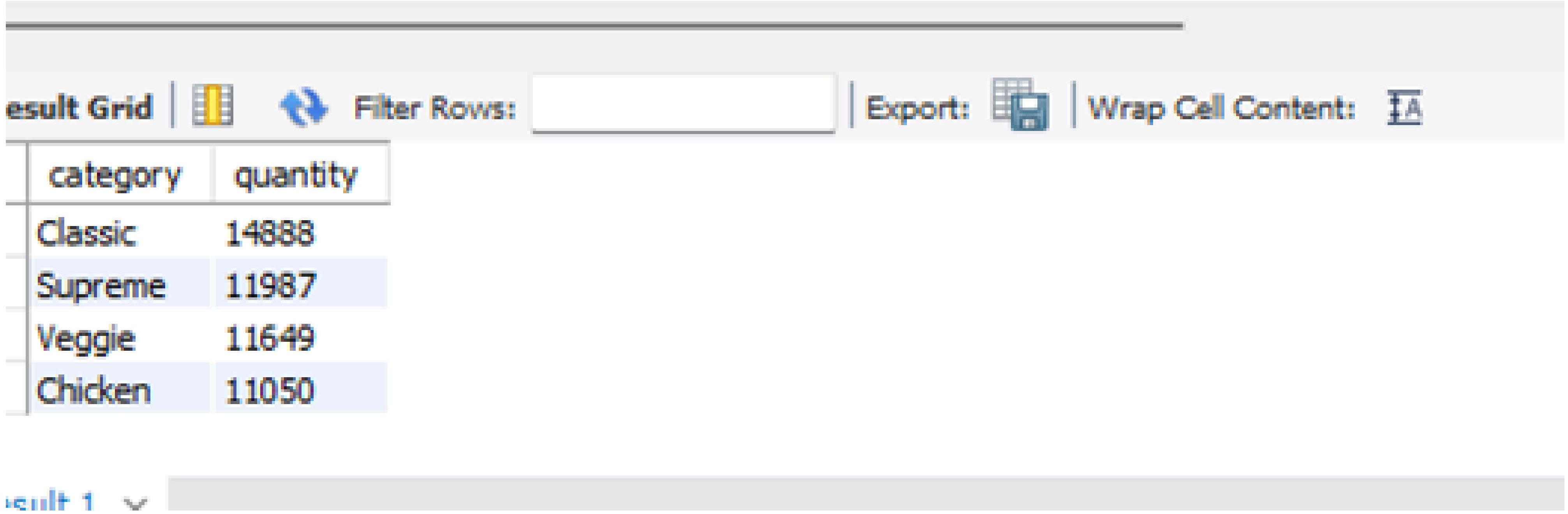


Hours	Order_Count
11	1231
12	2520
13	2455
14	1472
15	1468

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

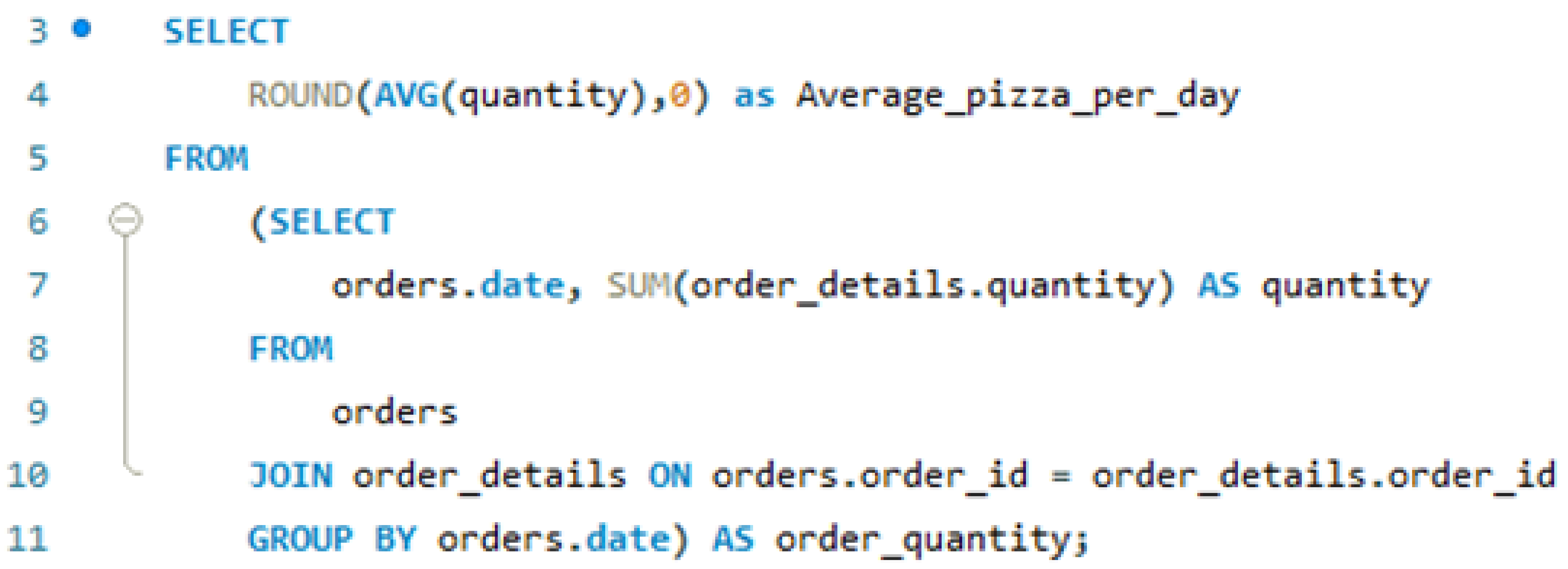


```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

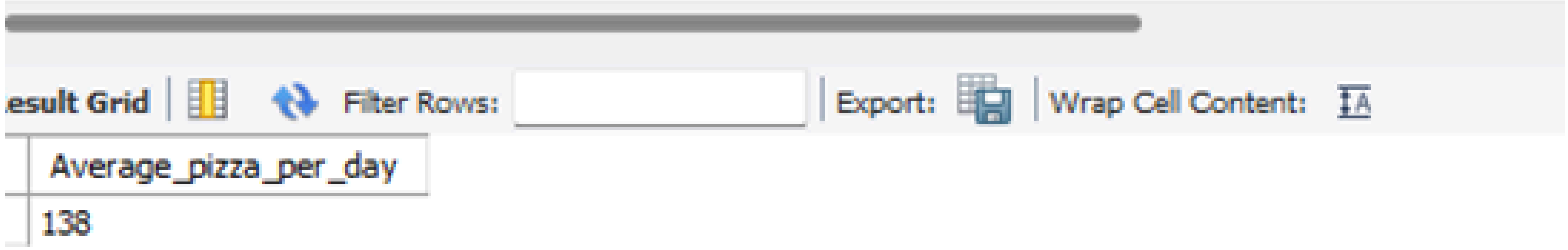


category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY

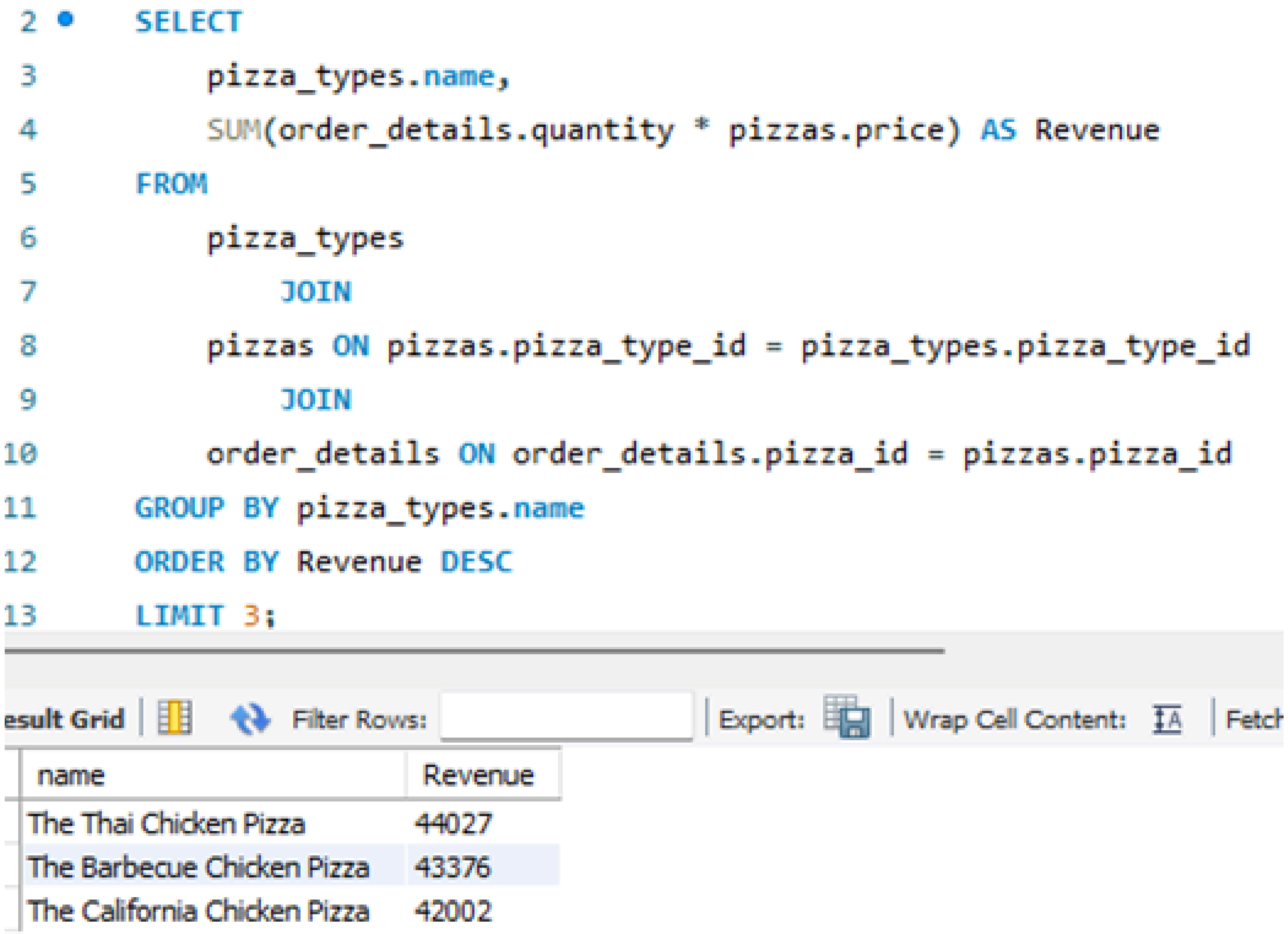


```
SELECT
    ROUND(AVG(quantity),0) as Average_pizza_per_day
FROM
    (SELECT
        orders.date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.date) AS order_quantity;
```



Average_pizza_per_day
138

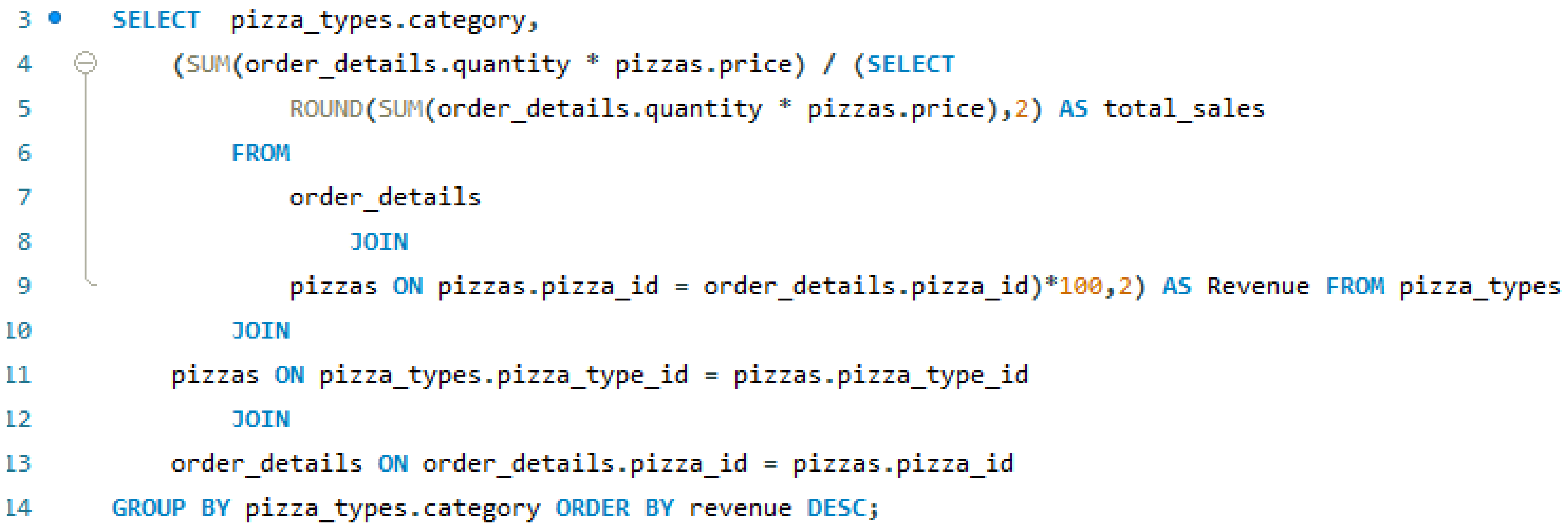
DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE



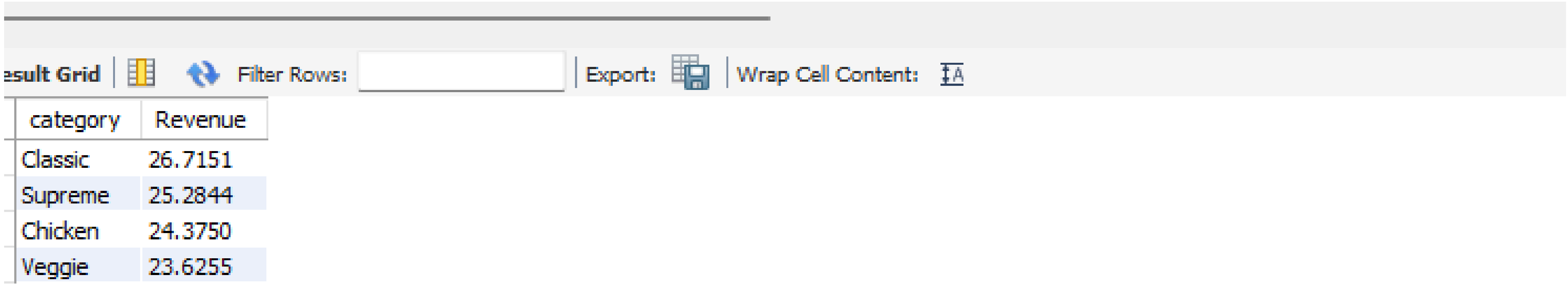
```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS Revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY Revenue DESC
LIMIT 3;
```

name	Revenue
The Thai Chicken Pizza	44027
The Barbecue Chicken Pizza	43376
The California Chicken Pizza	42002

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

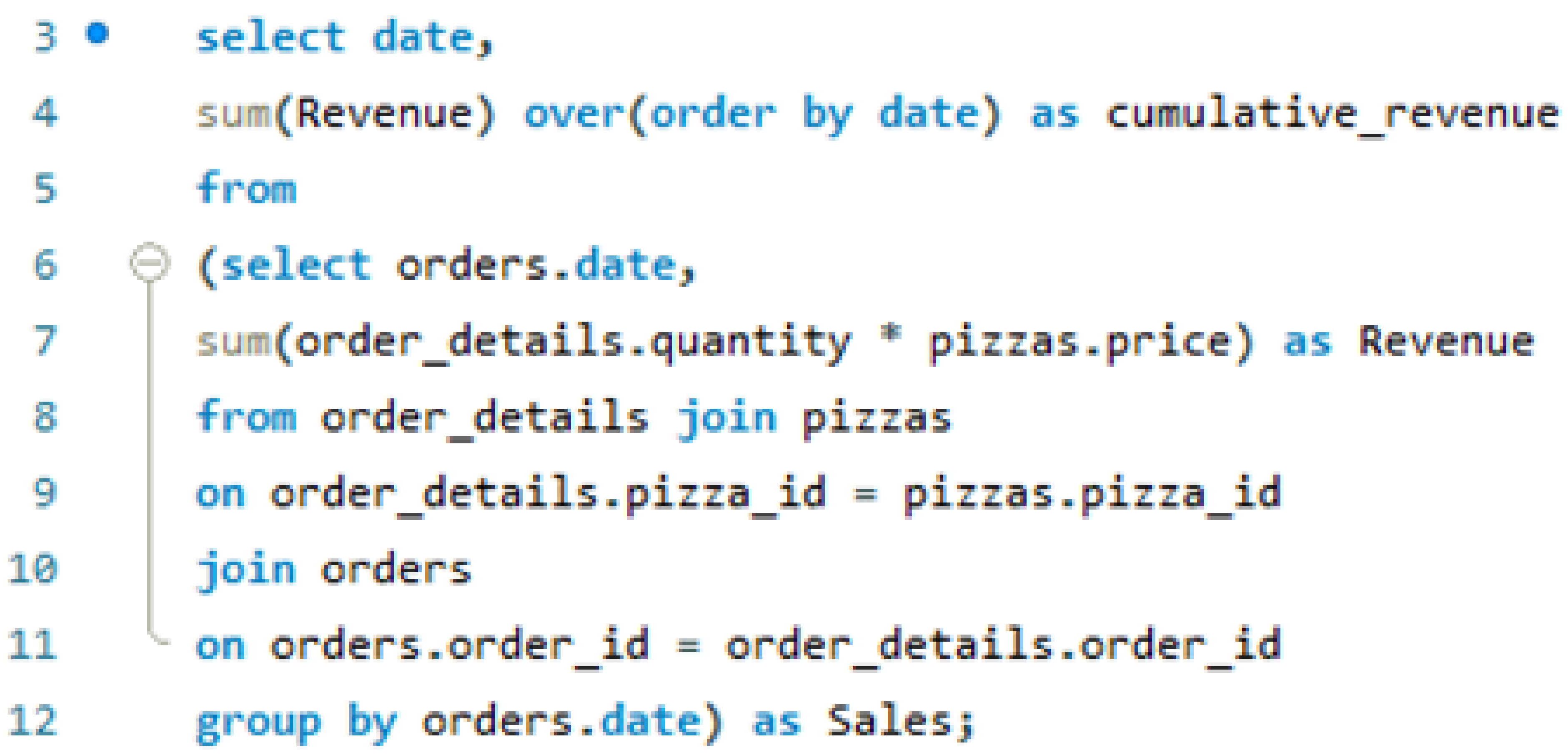


```sql
SELECT  pizza_types.category,
    (SUM(order_details.quantity * pizzas.price) / (SELECT
            ROUND(SUM(order_details.quantity * pizzas.price),2) AS total_sales
        FROM
            order_details
                JOIN
            pizzas ON pizzas.pizza_id = order_details.pizza_id)*100,2) AS Revenue FROM pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category ORDER BY revenue DESC;
```

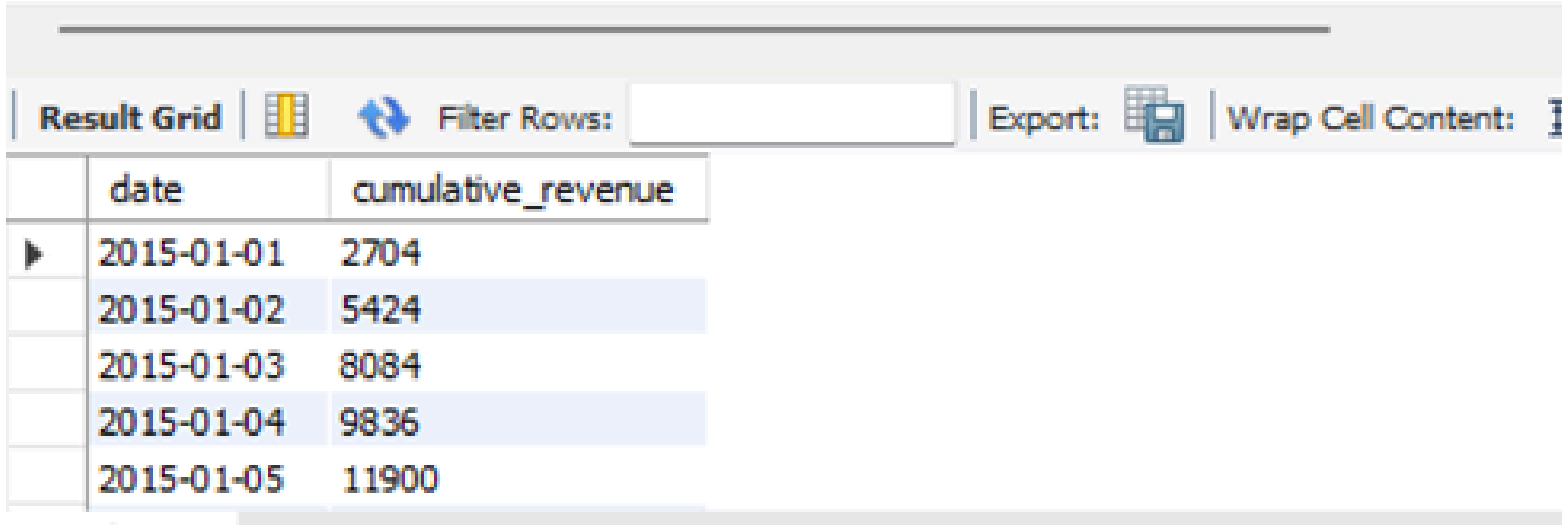


category	Revenue
Classic	26.7151
Supreme	25.2844
Chicken	24.3750
Veggie	23.6255

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

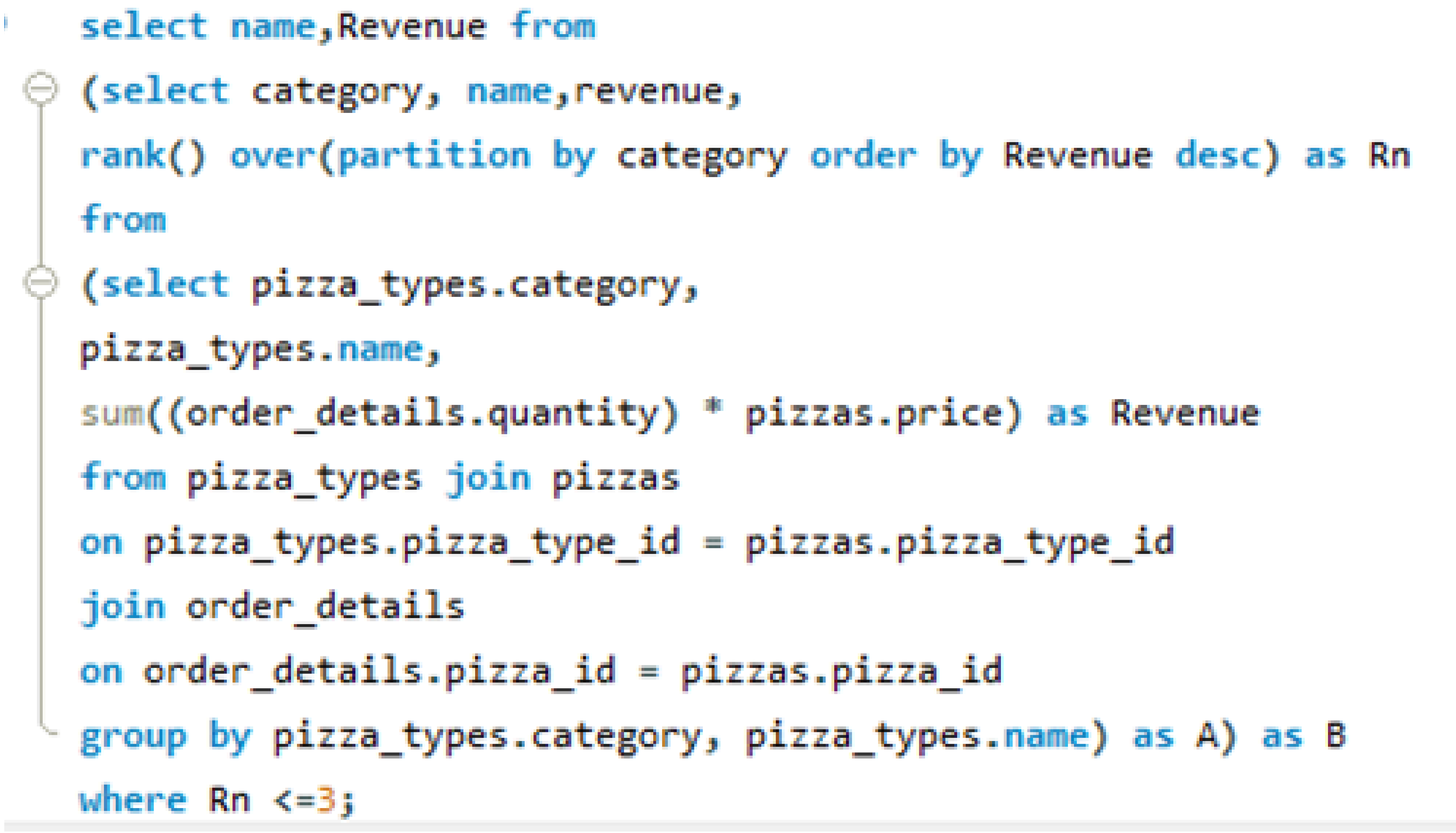


```
select date,
sum(Revenue) over(order by date) as cumulative_revenue
from
(select orders.date,
sum(order_details.quantity * pizzas.price) as Revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.date) as Sales;
```

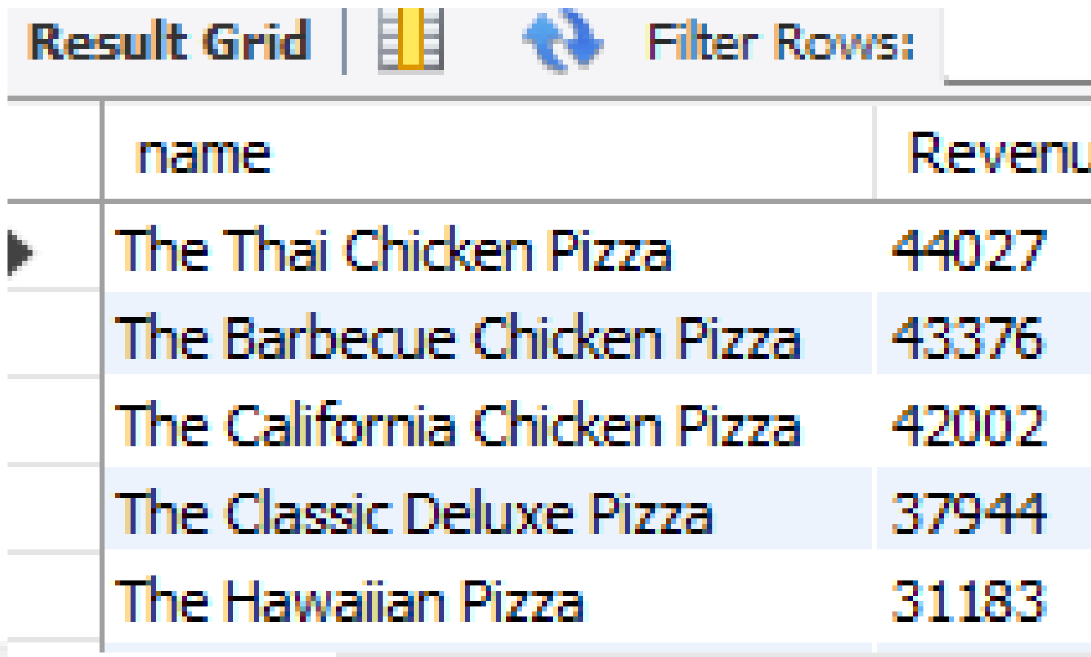


date	cumulative_revenue
2015-01-01	2704
2015-01-02	5424
2015-01-03	8084
2015-01-04	9836
2015-01-05	11900

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY



```
select name,Revenue from
(select category, name,revenue,
rank() over(partition by category order by Revenue desc) as Rn
from
(select pizza_types.category,
pizza_types.name,
sum((order_details.quantity) * pizzas.price) as Revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as A) as B
where Rn <=3;
```



name	Revenu
The Thai Chicken Pizza	44027
The Barbecue Chicken Pizza	43376
The California Chicken Pizza	42002
The Classic Deluxe Pizza	37944
The Hawaiian Pizza	31183

LEARNING OUTCOMES

- Gained expertise in SQL for data analysis.
- Improved understanding of database management and schema design.
- Learned to generate insights from raw data.
- Enhanced problem-solving and query optimization skills.
- Practical experience in revenue and performance analysis.

THANK YOU

Questions? Feedback?

Feel Free to Connect!