PIZZA SALES ANALYSIS USING SQL

Analyzing sales data to gain insights

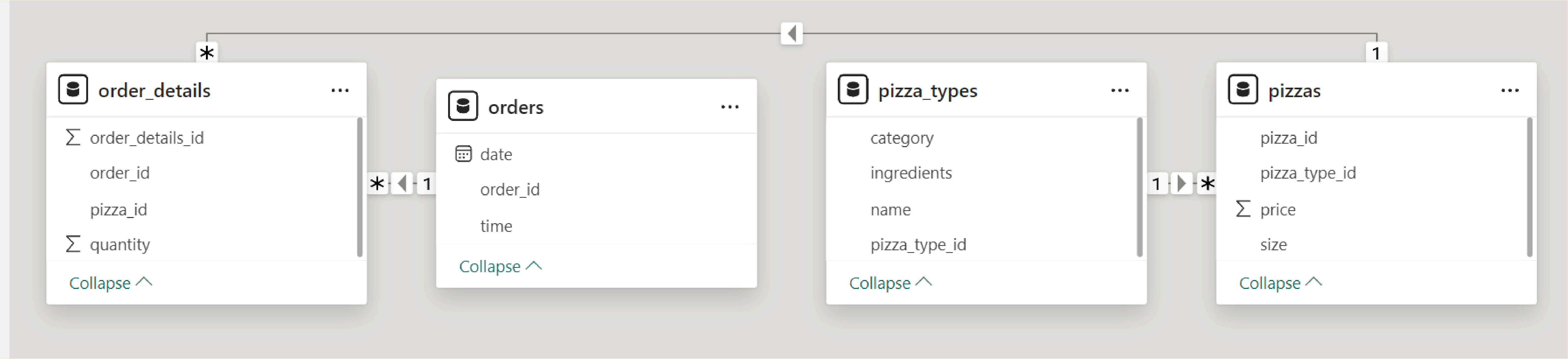
By Vrushabh Mahajan 27-05-2024

OVERVIEW OF THE PROJECT

OBJECTIVE

TO ANALYZE PIZZA SALES DATA USING SQL.

DATA SOURCE



order_details
- ∑ order_details_id
- order_id
- pizza_id
- ∑ quantity

Collapse

orders
- date
- order_id
- time

Collapse

pizza_types
- category
- ingredients
- name
- pizza_type_id

Collapse

pizzas
- pizza_id
- pizza_type_id
- ∑ price
- size

Collapse

SOLVE THE QUESTION BY USING SQL QUERIES

Basic:

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

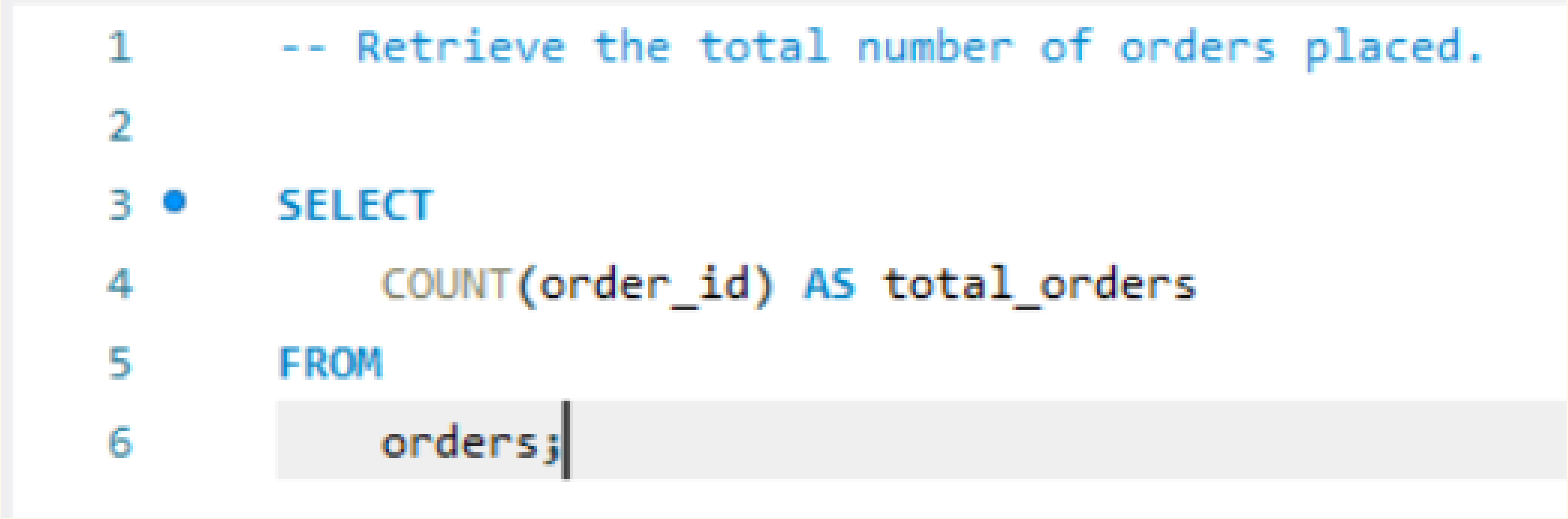
Intermediate:

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

Advanced:

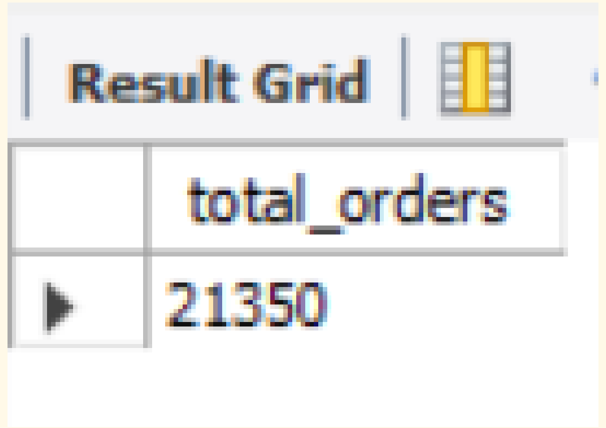
- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



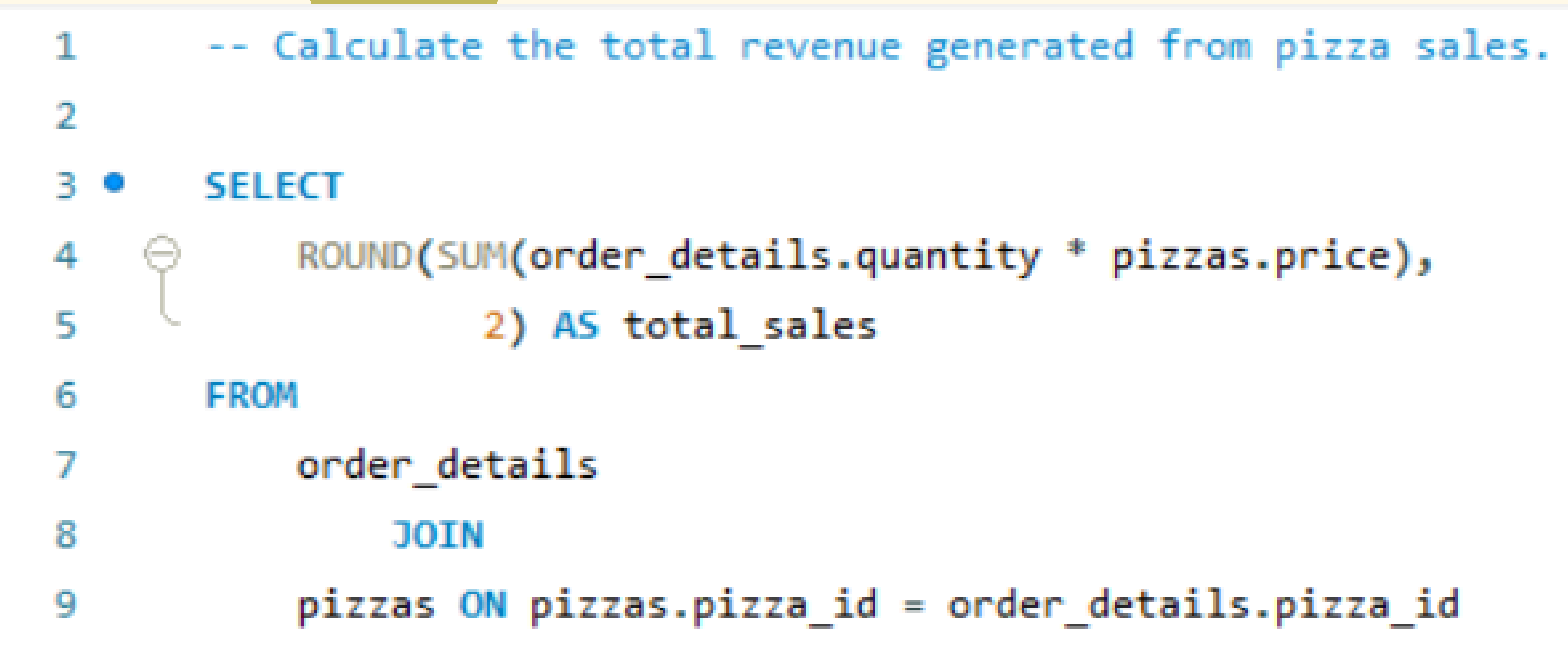
```
-- Retrieve the total number of orders placed.

SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

OUTPUT



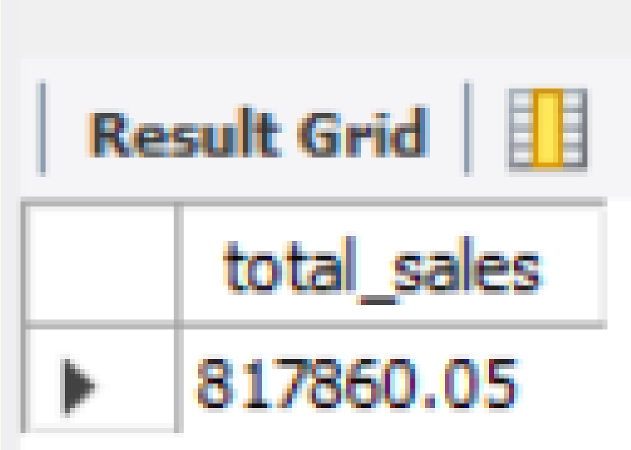
total_orders
21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.



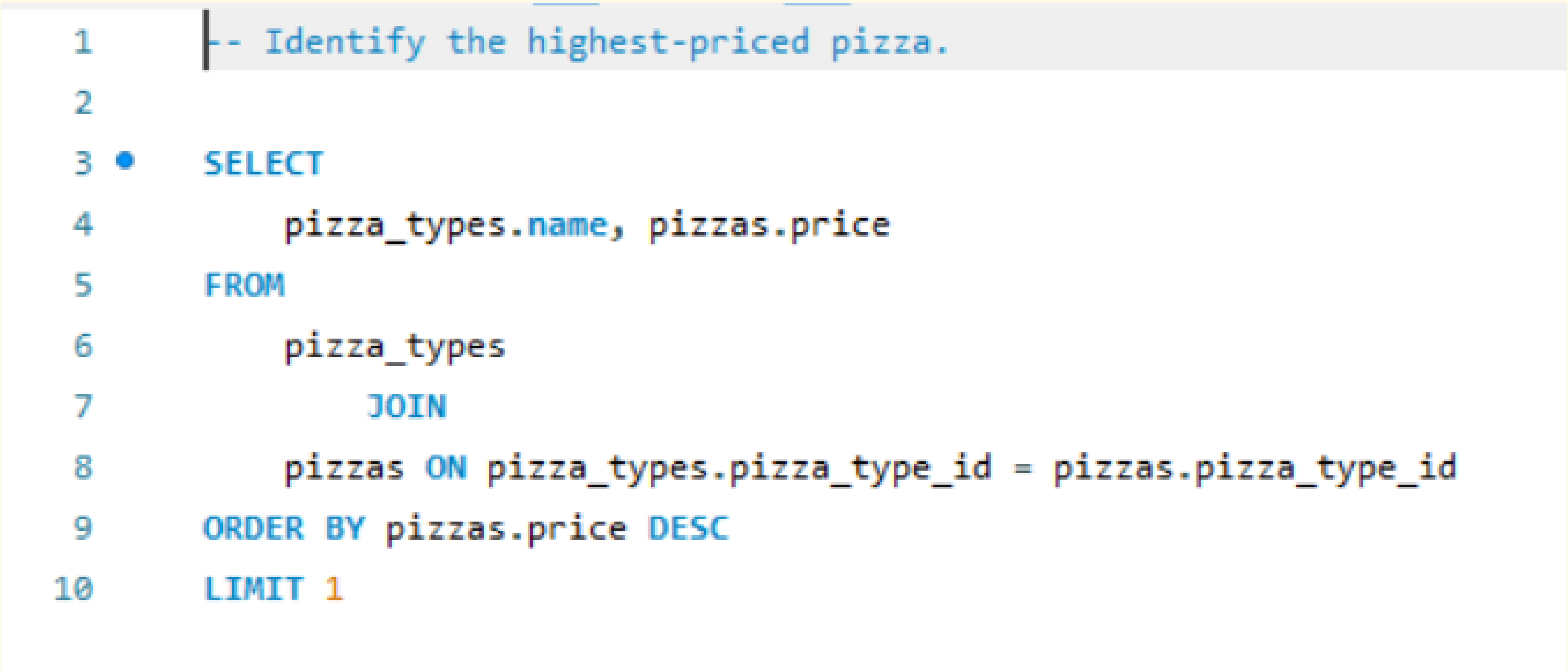
```sql
-- Calculate the total revenue generated from pizza sales.

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
```

OUTPUT



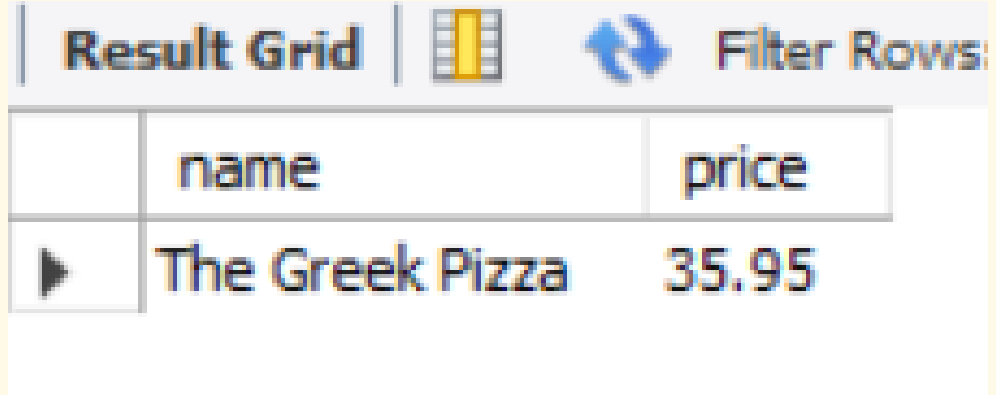
```
  total_sales
▶ 817860.05
```

IDENTIFY THE HIGHEST-PRICED PIZZA.

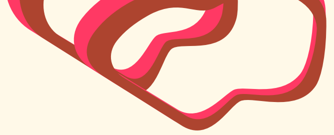


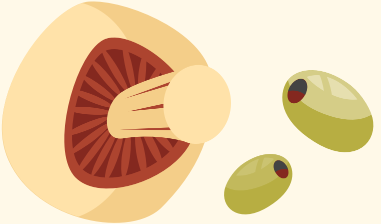
```sql
-- Identify the highest-priced pizza.

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
```

Output

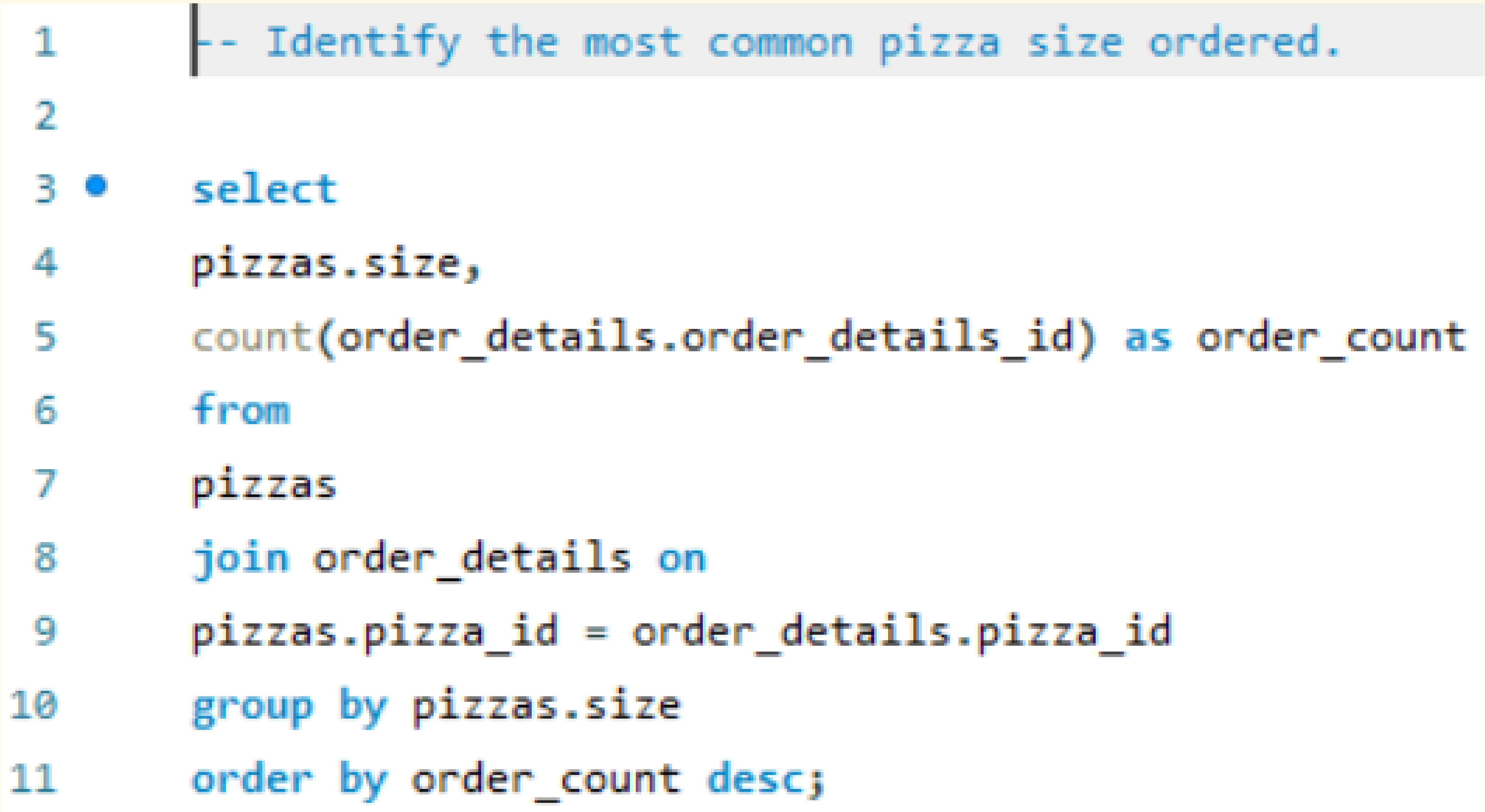


name	price
The Greek Pizza	35.95



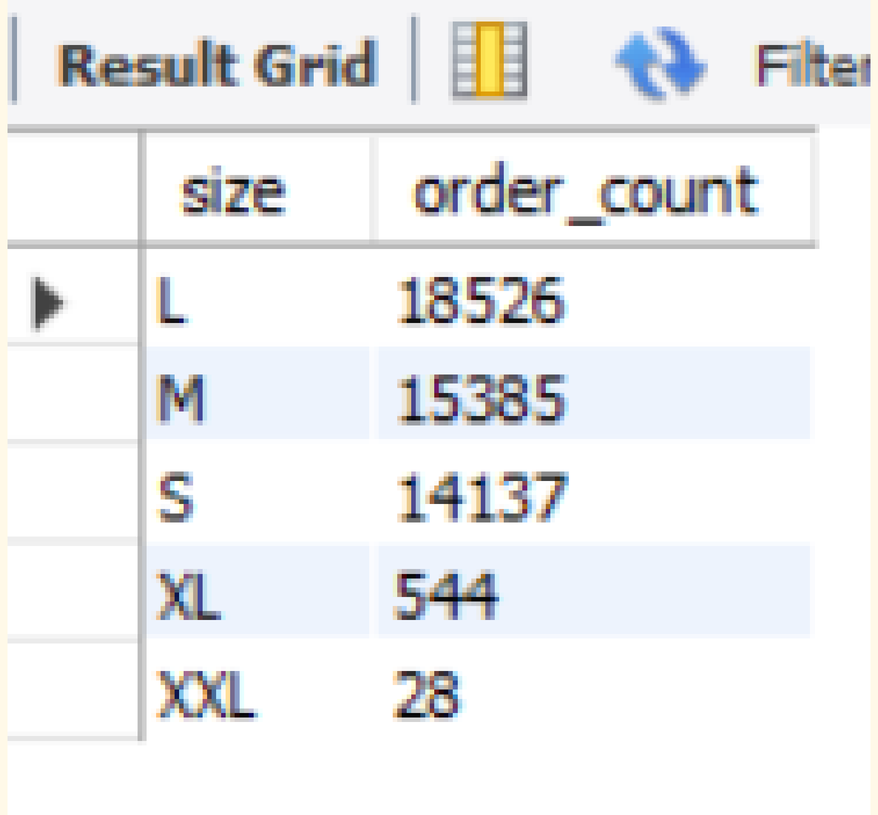


IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

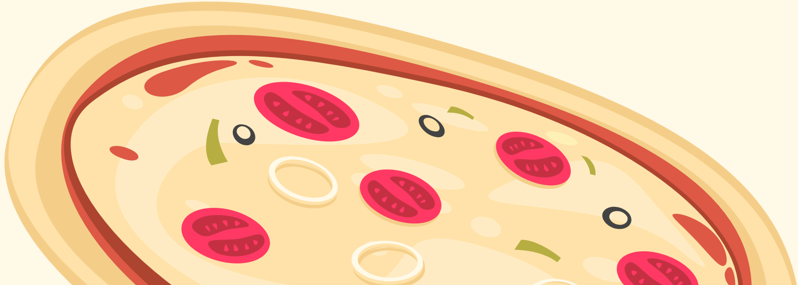


```
-- Identify the most common pizza size ordered.

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;
```

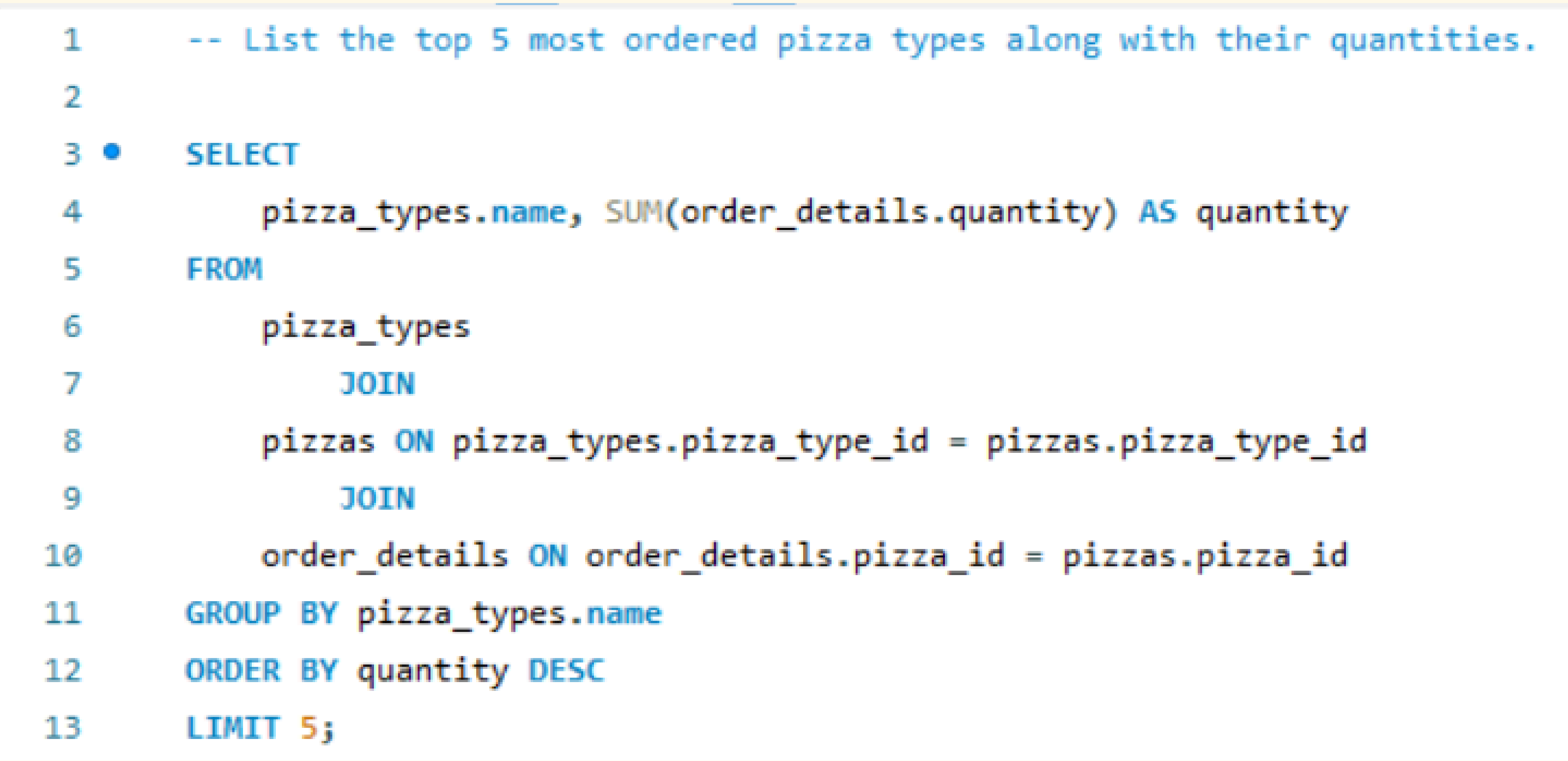
OUTPUT



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

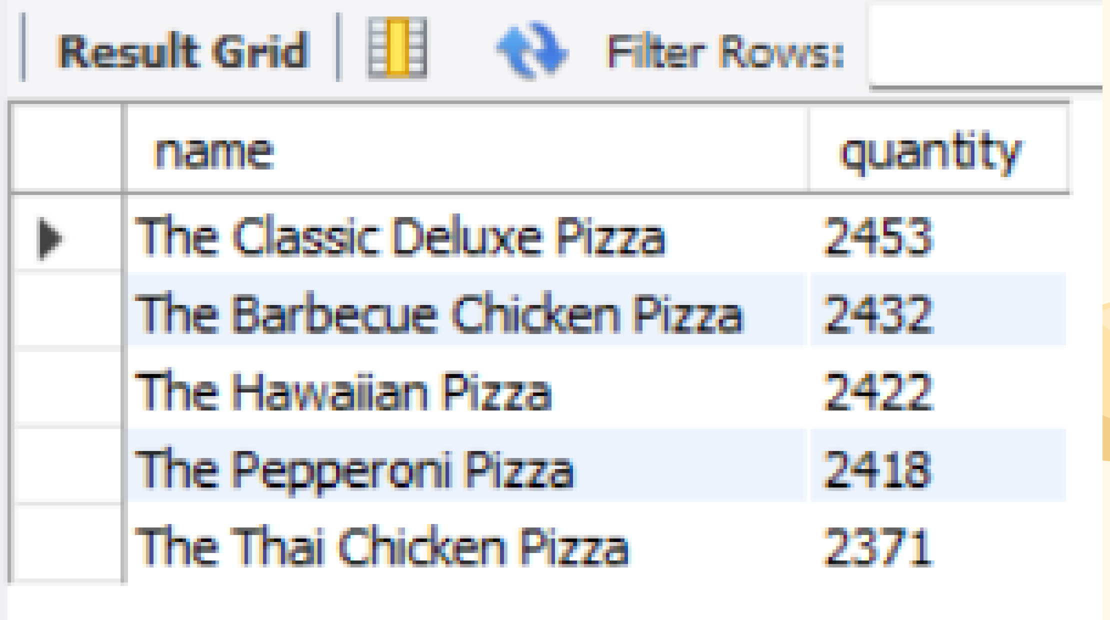


LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.



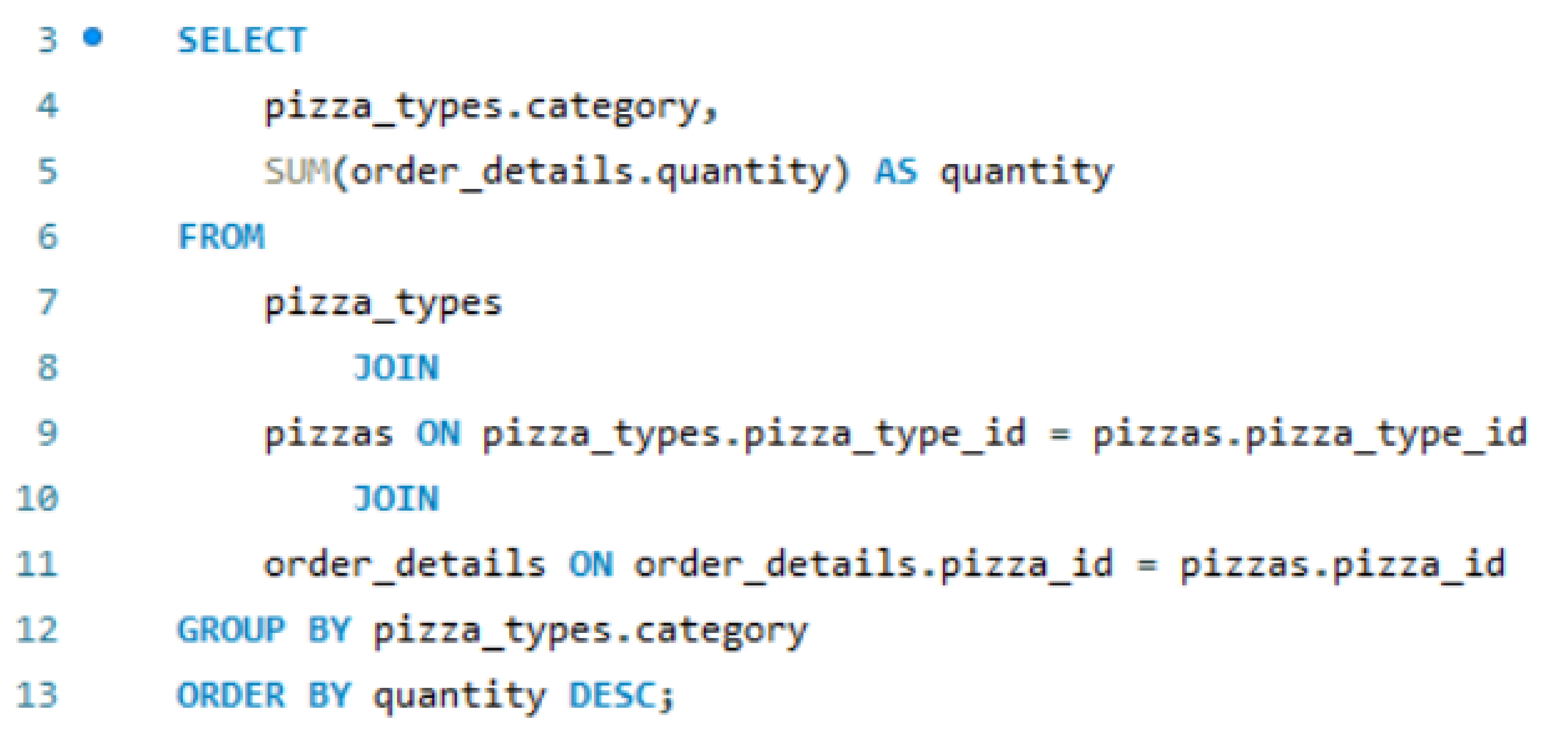
```
-- List the top 5 most ordered pizza types along with their quantities.

SELECT
    pizza_types.name, 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.name
ORDER BY quantity DESC
LIMIT 5;
```

OUTPUT



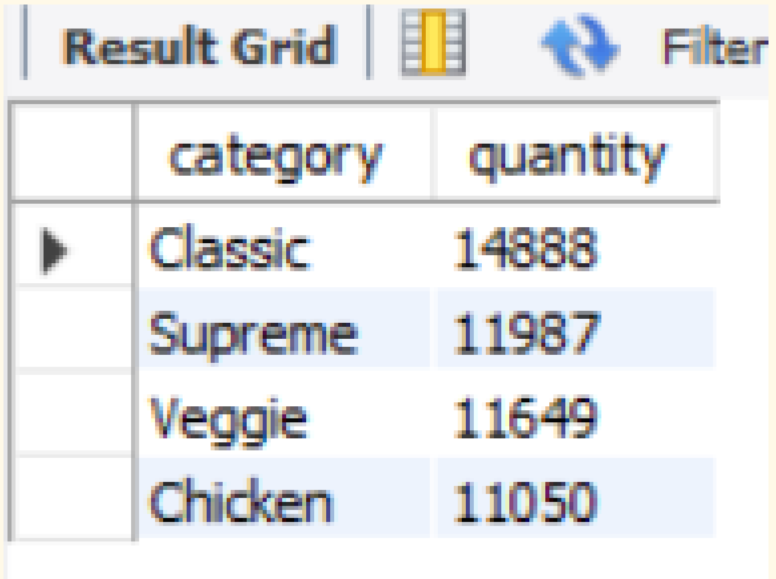
name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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



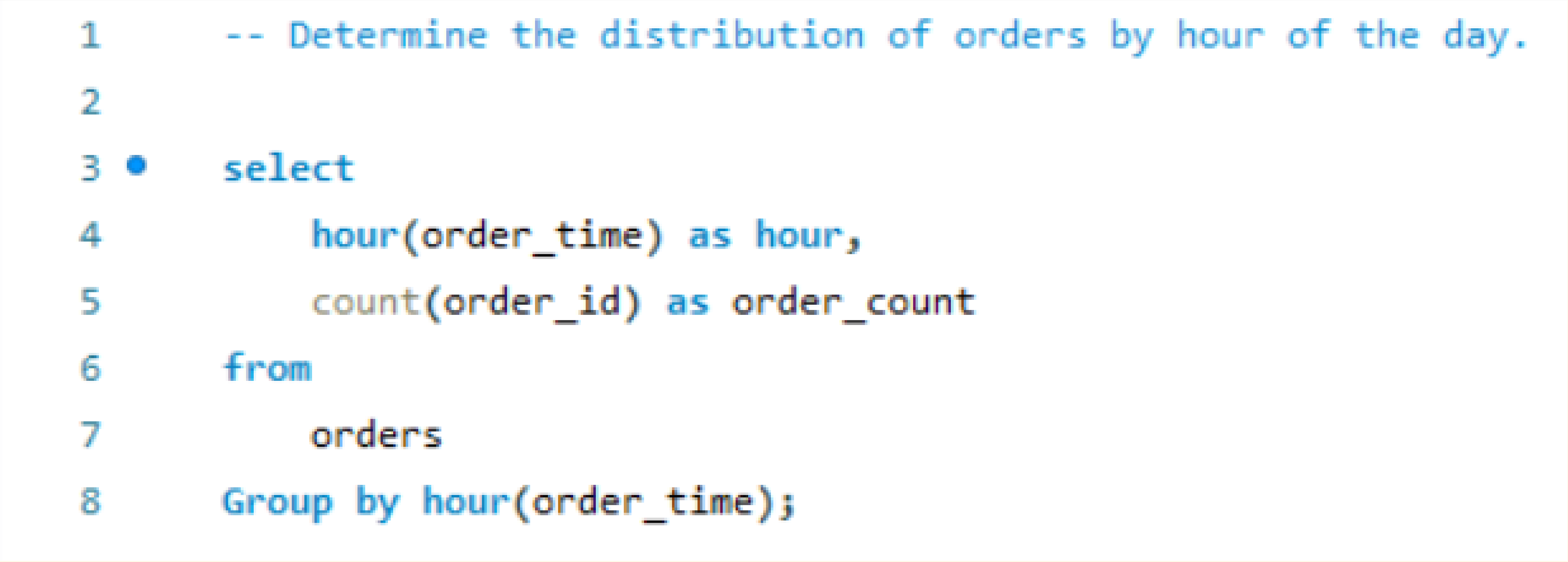
```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity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

OUTPUT



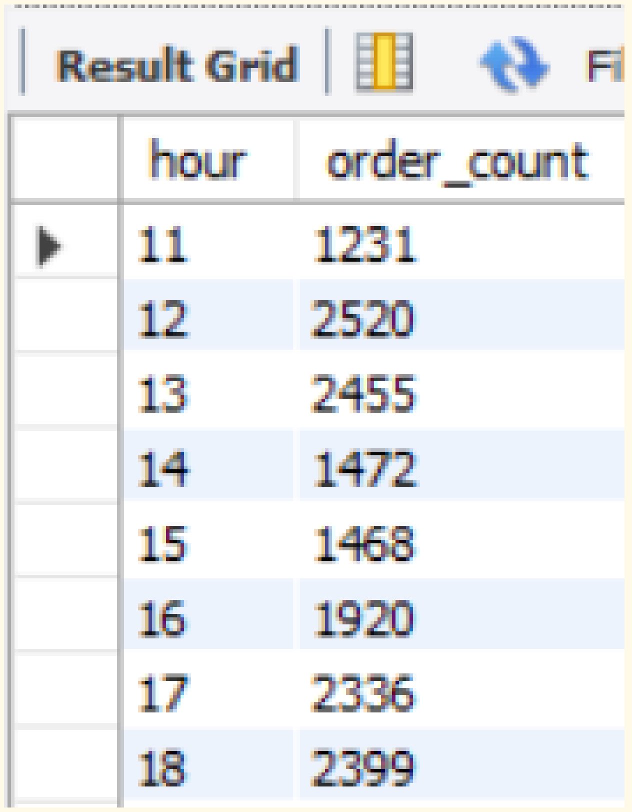
category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.



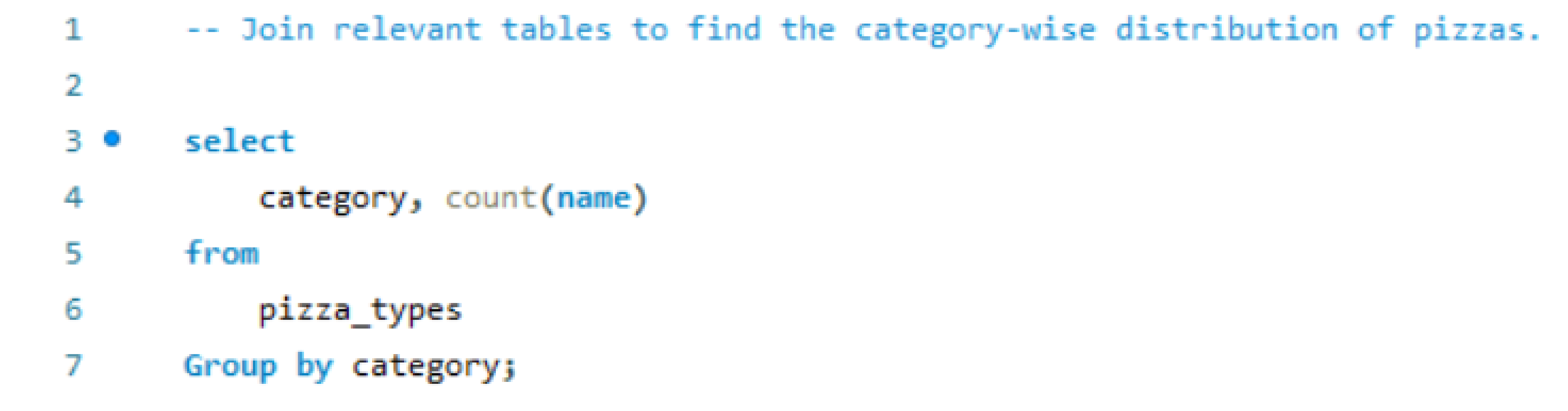
```sql
-- Determine the distribution of orders by hour of the day.

select
    hour(order_time) as hour,
    count(order_id) as order_count
from
    orders
Group by hour(order_time);
```

OUTPUT



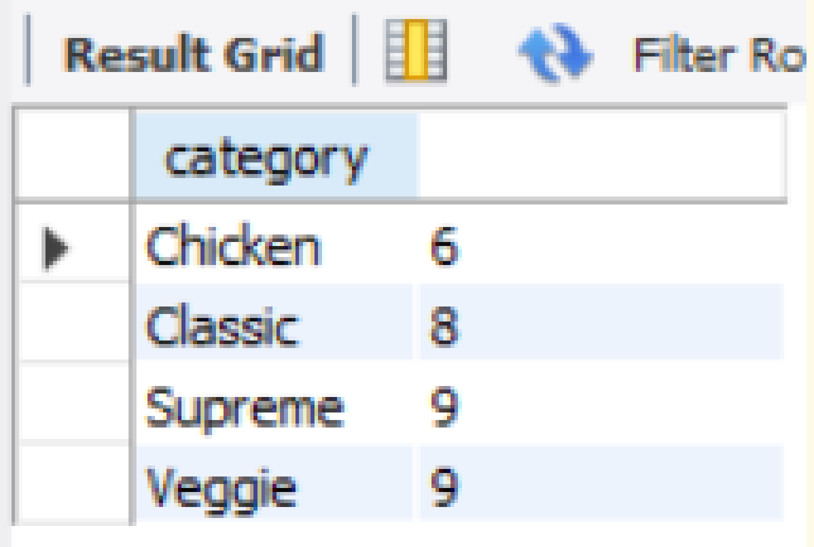
hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399

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



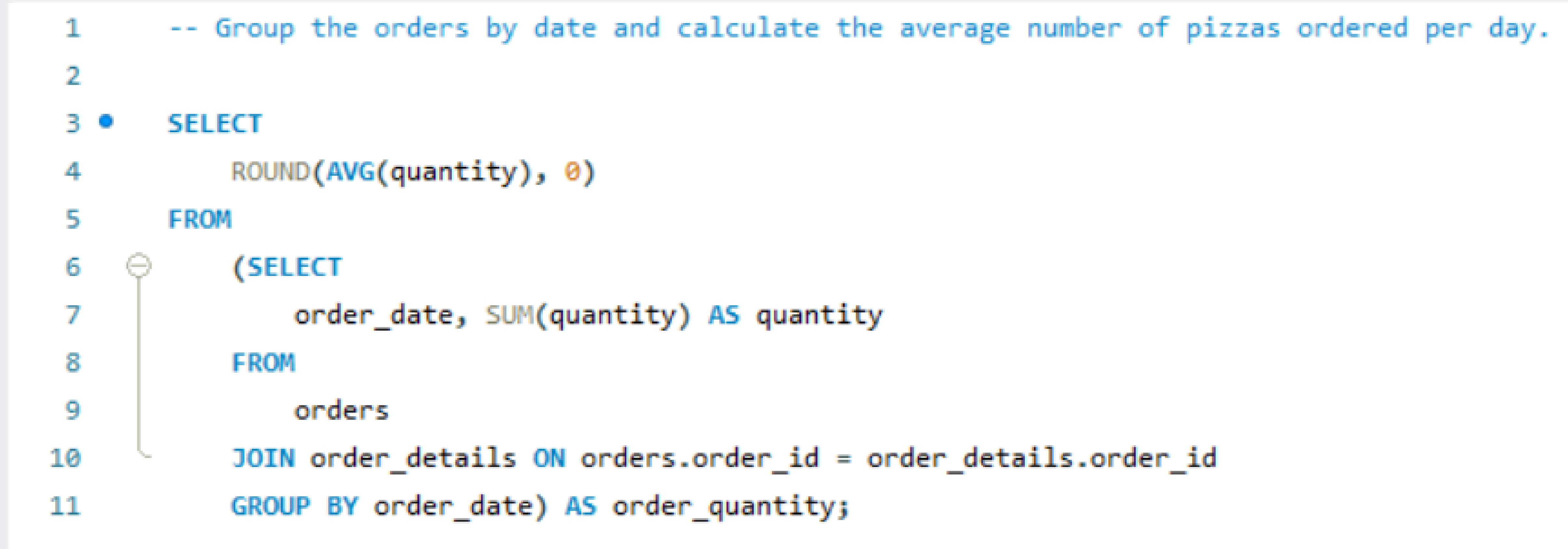
```
-- Join relevant tables to find the category-wise distribution of pizzas.

select
    category, count(name)
from
    pizza_types
Group by category;
```

OUTPUT



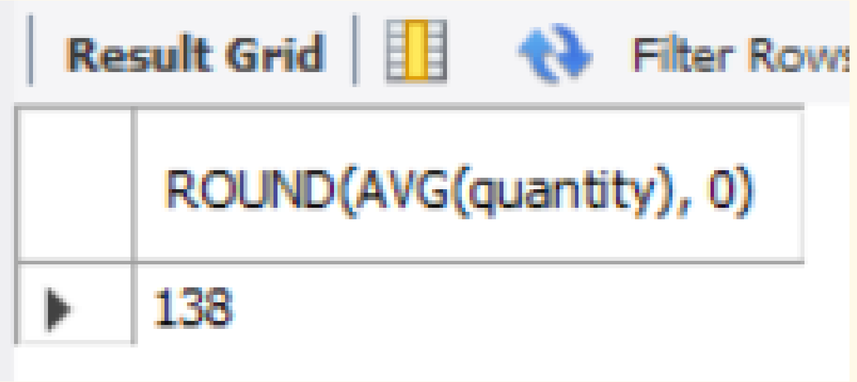
category	
Chicken	6
Classic	8
Supreme	9
Veggie	9

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



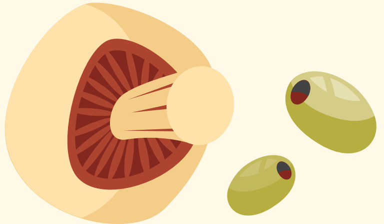
```
-- Group the orders by date and calculate the average number of pizzas ordered per day.

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

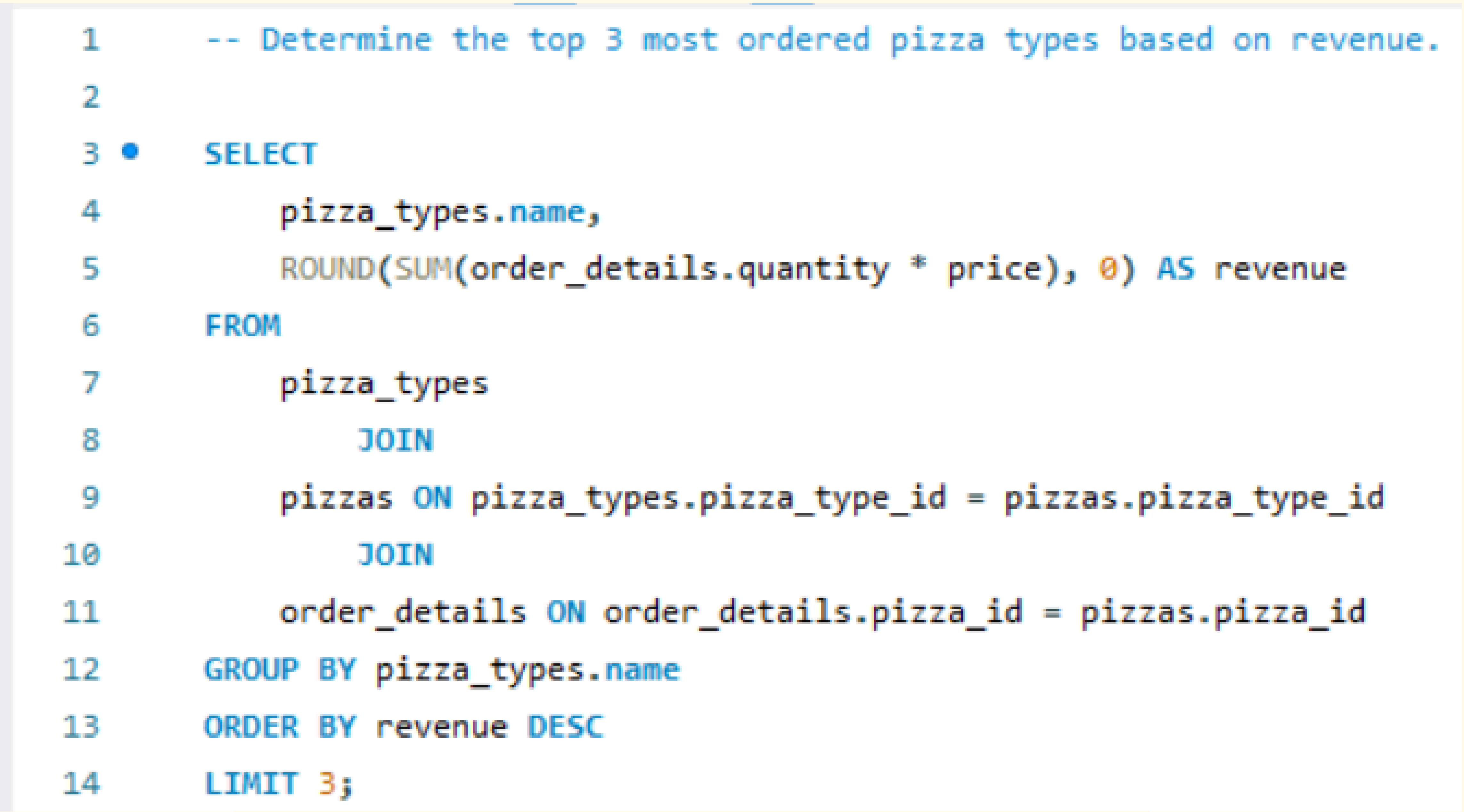
OUTPUT



ROUND(AVG(quantity), 0)
138

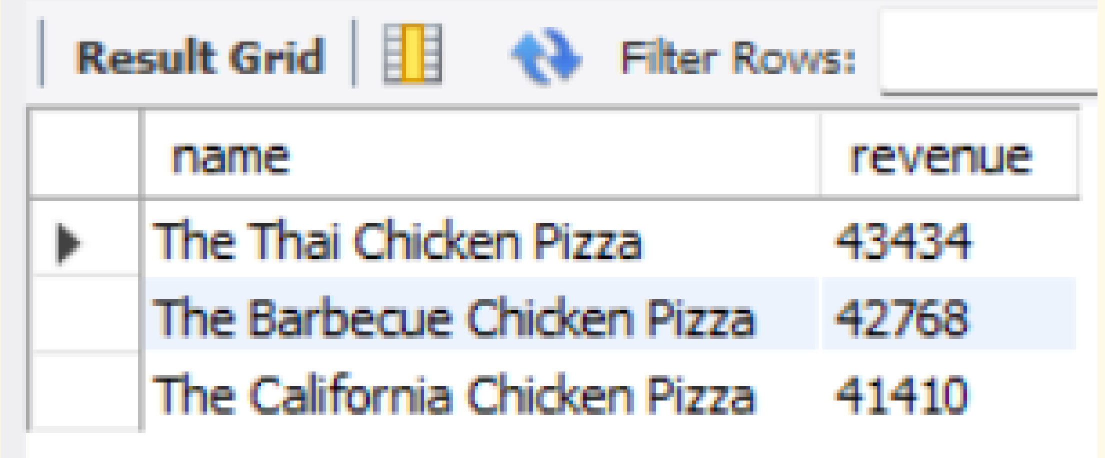
DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.





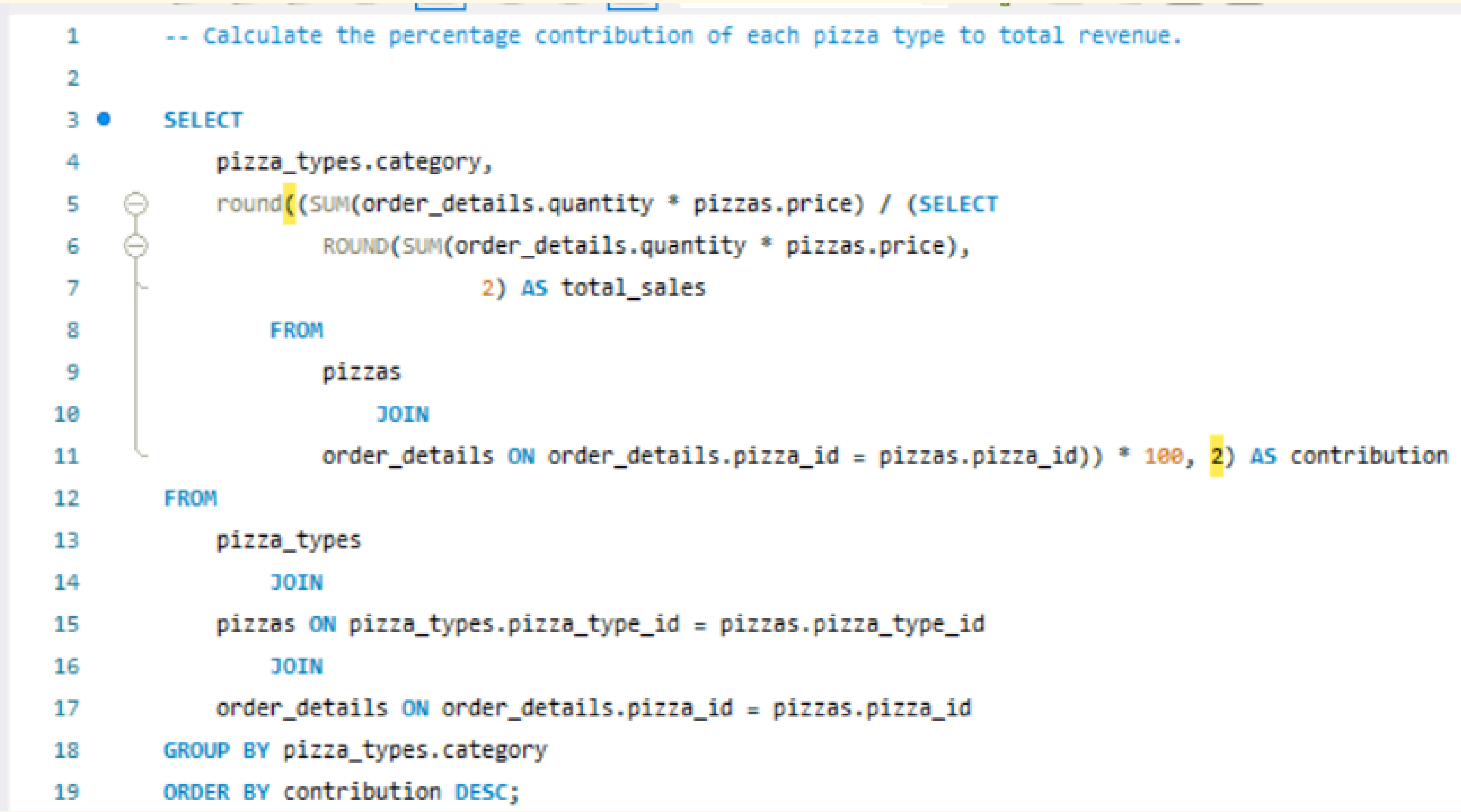
```
-- Determine the top 3 most ordered pizza types based on revenue.

SELECT
    pizza_types.name,
    ROUND(SUM(order_details.quantity * price), 0) 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.name
ORDER BY revenue DESC
LIMIT 3;
```

OUTPUT



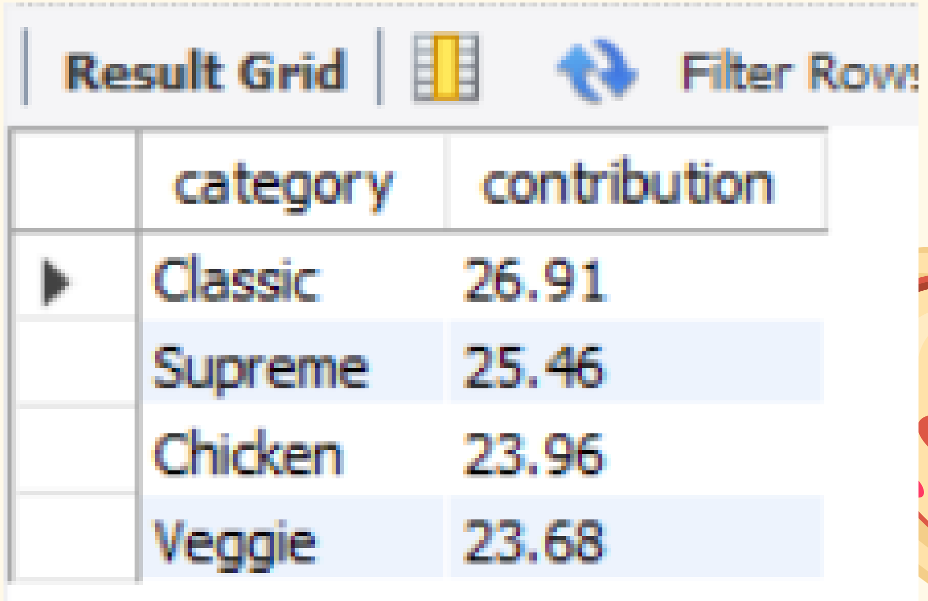
name	revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.



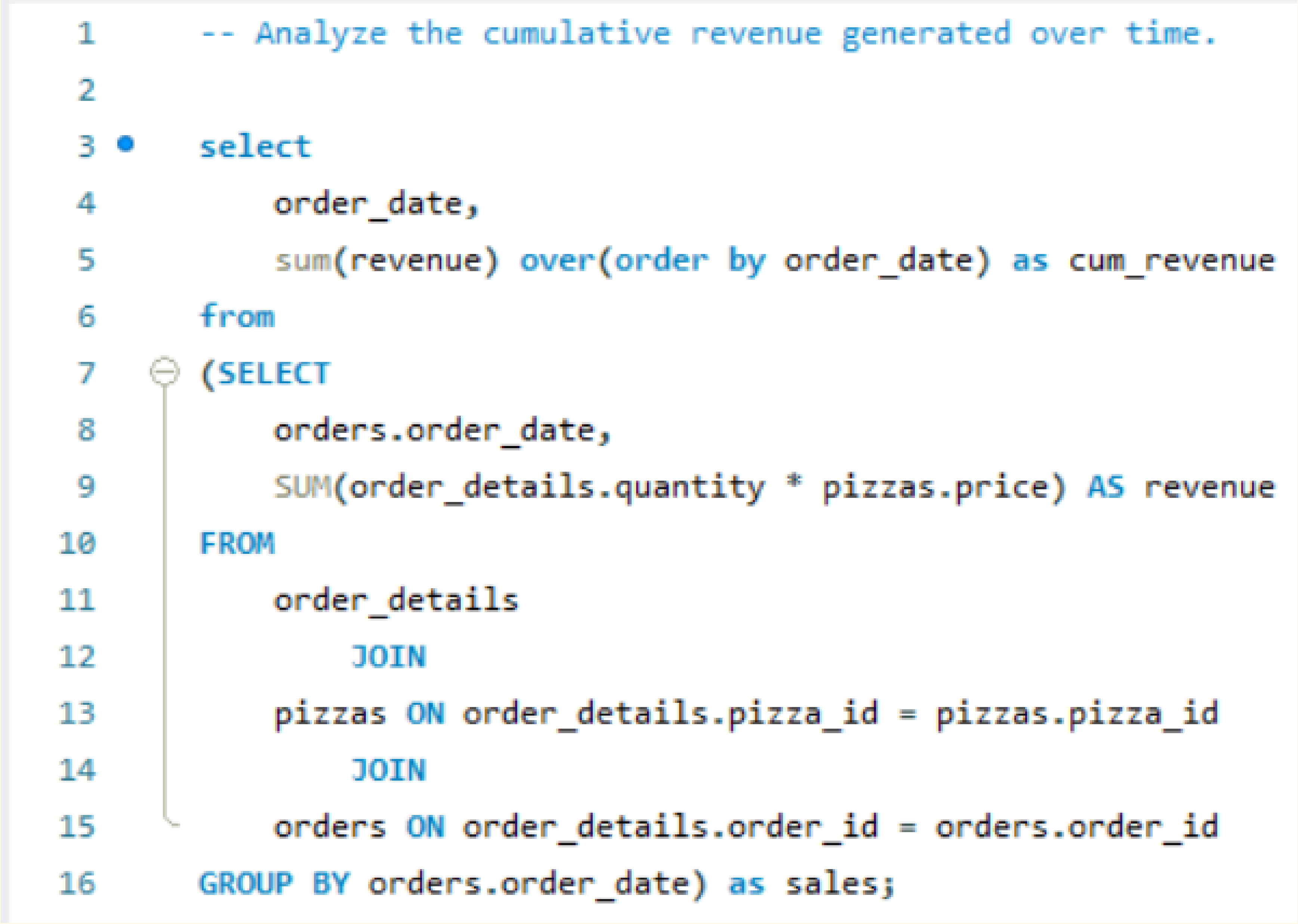
```
-- Calculate the percentage contribution of each pizza type to total revenue.

SELECT
    pizza_types.category,
    round((SUM(order_details.quantity * pizzas.price) / (SELECT
            ROUND(SUM(order_details.quantity * pizzas.price),
                        2) AS total_sales
        FROM
            pizzas
                JOIN
            order_details ON order_details.pizza_id = pizzas.pizza_id)) * 100, 2) AS contribution
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 contribution DESC;
```

OUTPUT



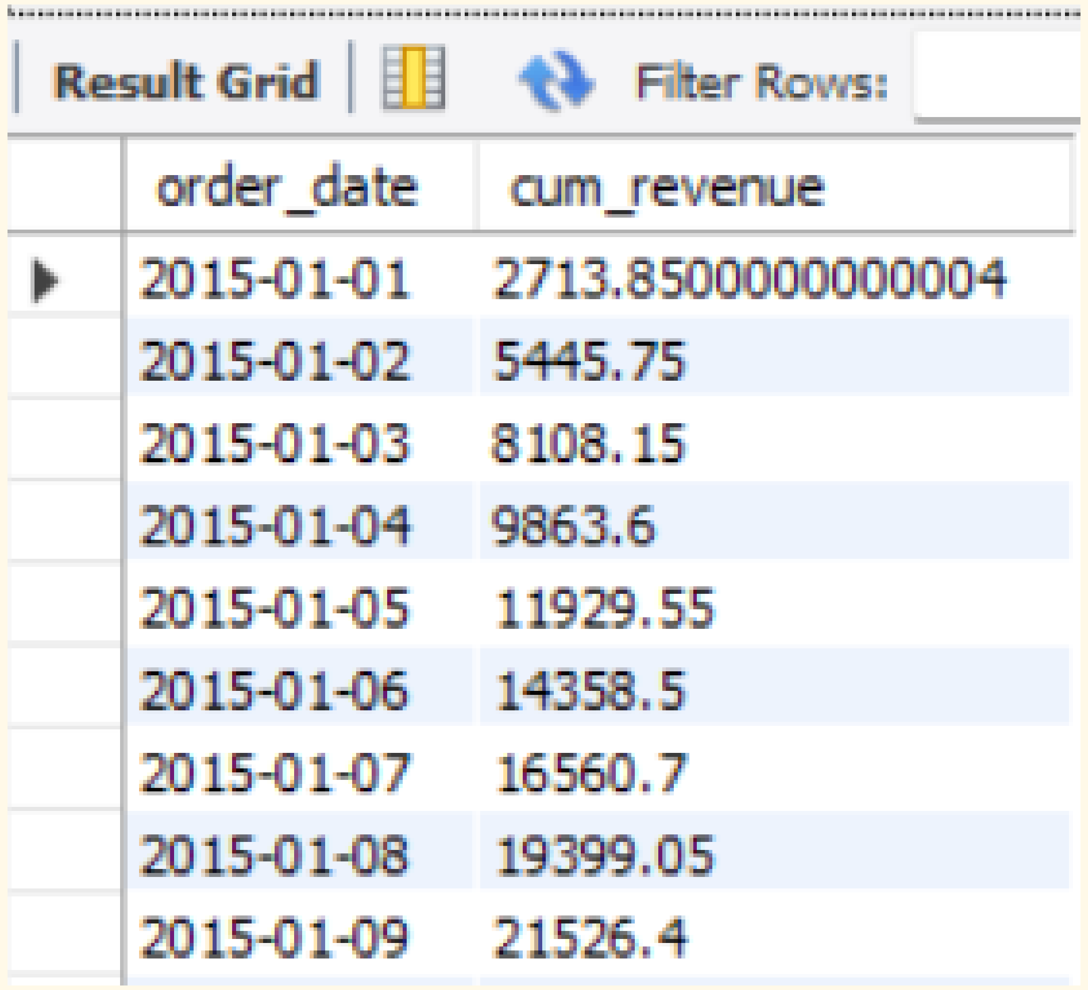
category	contribution
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



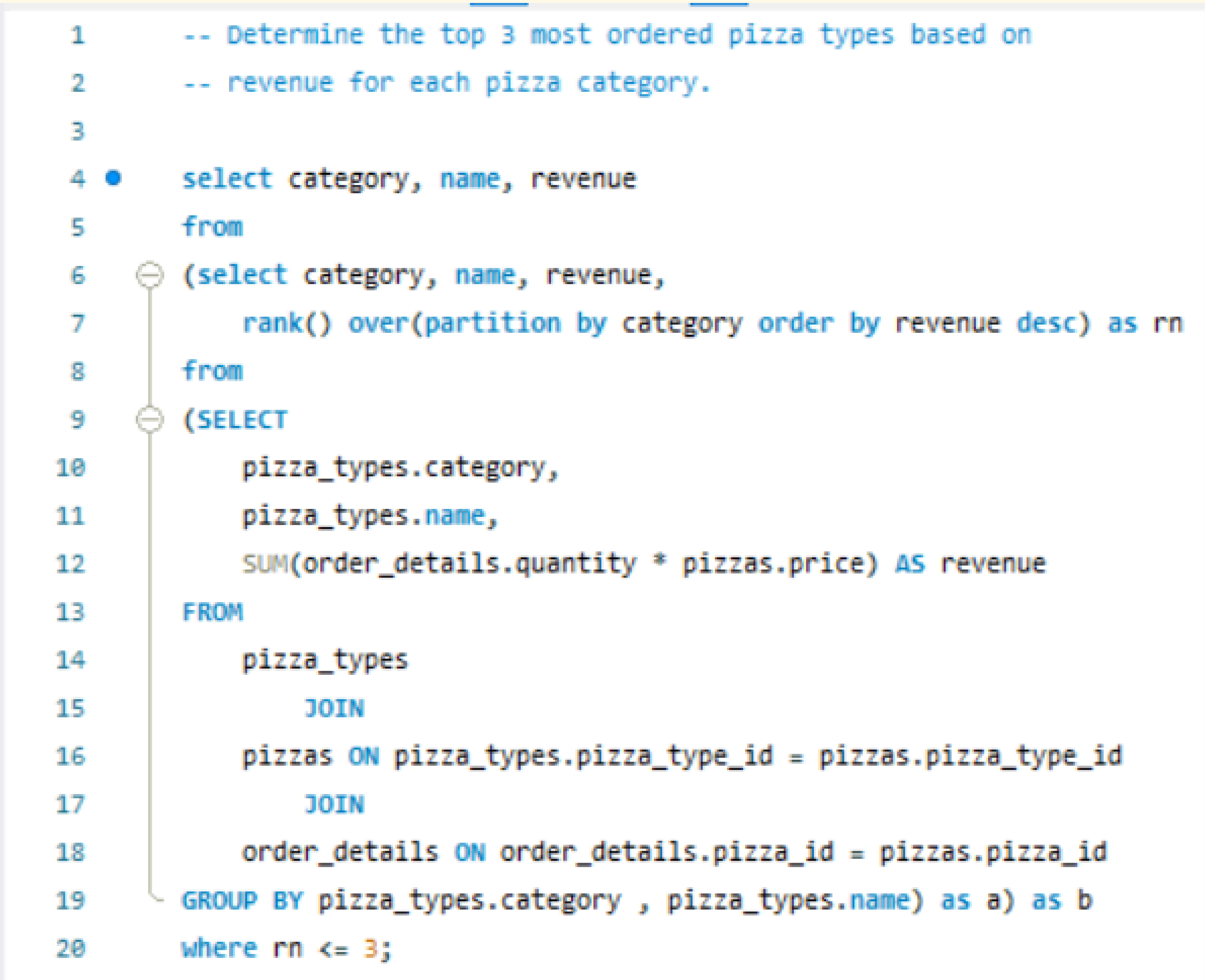
```sql
-- Analyze the cumulative revenue generated over time.

select
    order_date,
    sum(revenue) over(order by order_date) as cum_revenue
from
(SELECT
    orders.order_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 order_details.order_id = orders.order_id
GROUP BY orders.order_date) as sales;
```

OUTPUT



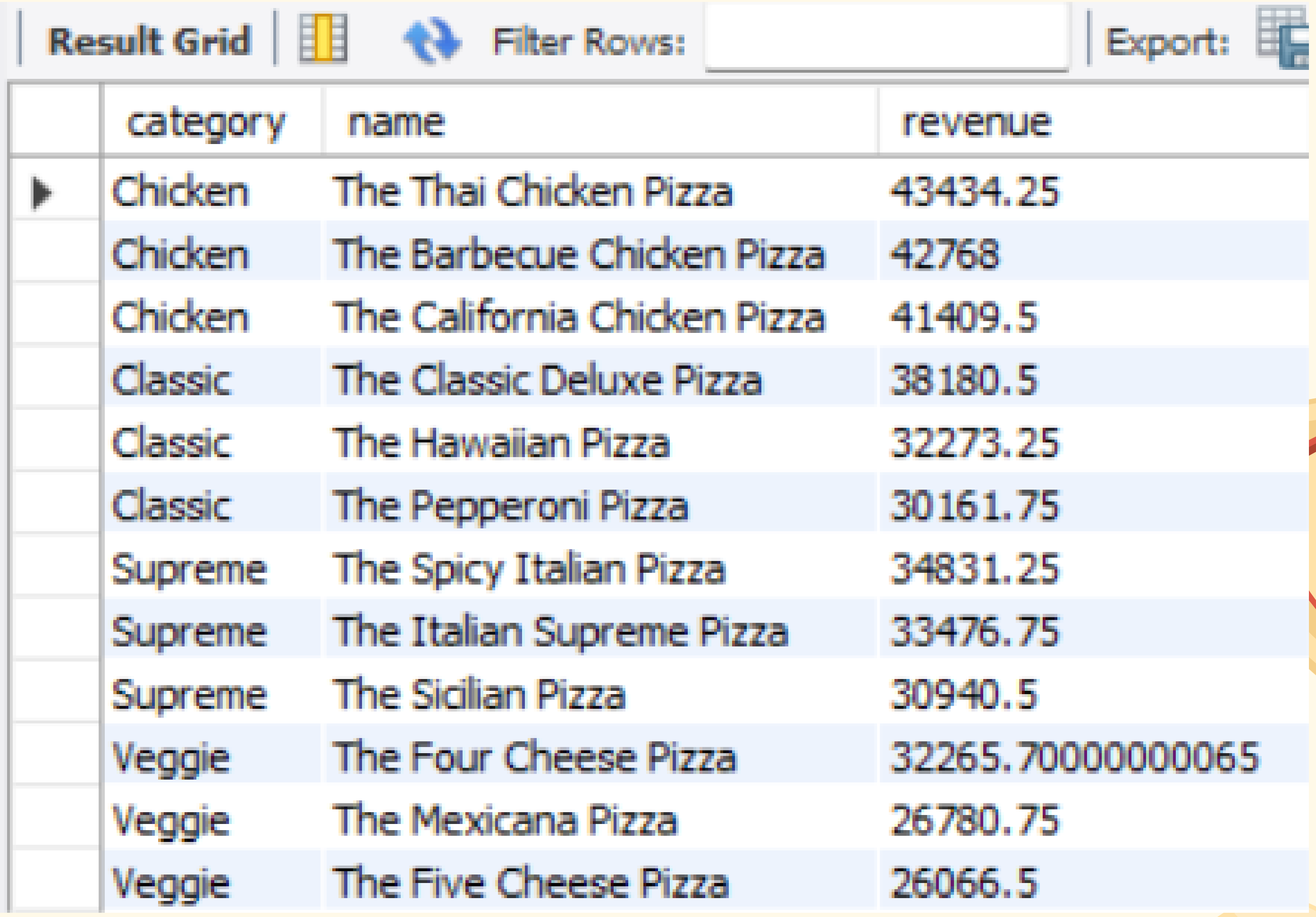
order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4

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



```
-- Determine the top 3 most ordered pizza types based on
-- revenue for each pizza category.

select category, 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;
```

OUTPUT



category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

SUMMARY OF FINDINGS

1. **Total Orders: Calculated the total number of orders, showing the overall sales activity.**
2. **Total Revenue: Determined the total revenue from pizza sales, highlighting financial performance.**
3. **Highest-Priced Pizza: Identified the highest-priced pizza, offering insights into premium product performance.**
4. **Most Common Pizza Size: Found the most frequently ordered pizza size, indicating customer preference.**
5. **Top 5 Pizzas: Listed the top 5 most ordered pizzas, aiding in menu planning and promotions.**
6. **Pizza Category Quantities: Analyzed the total quantities ordered in each pizza category to understand customer preferences.**
7. **Order Distribution by Hour: Revealed peak hours for orders, useful for optimizing staffing and operations.**
8. **Category-Wise Distribution: Examined the popularity of different pizza types by category.**
9. **Daily Average Orders: Calculated the average number of pizzas ordered per day, showing daily sales trends.**
10. **Top Revenue Pizzas: Identified the top 3 pizzas based on revenue, helping focus on high-revenue items.**
11. **Revenue Contribution: Analyzed the percentage contribution of each pizza type to total revenue.**
12. **Cumulative Revenue: Tracked revenue growth over time for forecasting and strategy.**
13. **Top Pizzas by Category: Identified the top 3 most ordered pizzas within each category, aiding targeted marketing.**

FUTURE WORK
ADDITIONAL ANALYSES THAT COULD BE CONDUCTED

1. **Customer Segmentation Analysis:**
   - Objective: Identify different customer segments based on ordering behavior and preferences.
   - Approach: Use clustering techniques (e.g., k-means clustering) to group customers into segments such as frequent buyers, high spenders, or occasional buyers. Analyze each segment's characteristics and preferences.
2. **Sales Forecasting:**
   - Objective: Predict future sales trends to help with inventory and staffing decisions.
   - Approach: Utilize time series analysis and machine learning models (e.g., ARIMA, Prophet) to forecast future sales based on historical data.
3. **Promotion Effectiveness:**
   - Objective: Measure the impact of various promotions and discounts on sales.
   - Approach: Perform A/B testing and regression analysis to determine the effectiveness of different promotional campaigns. Compare sales data during promotion periods with non-promotion periods.
4. **Customer Lifetime Value (CLV):**
   - Objective: Calculate the long-term value of customers to the business.
   - Approach: Use RFM (Recency, Frequency, Monetary) analysis and predictive modeling to estimate the CLV of different customer segments. This helps in identifying valuable customers and tailoring marketing efforts.
5. **Inventory Optimization:**
   - Objective: Optimize inventory levels to reduce waste and improve turnover.
   - Approach: Analyze sales data to identify trends and use inventory management techniques (e.g., just-in-time inventory) to optimize stock levels based on demand forecasts.

“THE BEST WAY TO PREDICT YOUR FUTURE IS TO CREATE IT.”

—Abraham Lincoln

THANKS!

Do you have any questions?

vrushabhmahajan21@gmail.com

+91 727 663 4577

linkedin.com/in/vrushabh-mahajan/