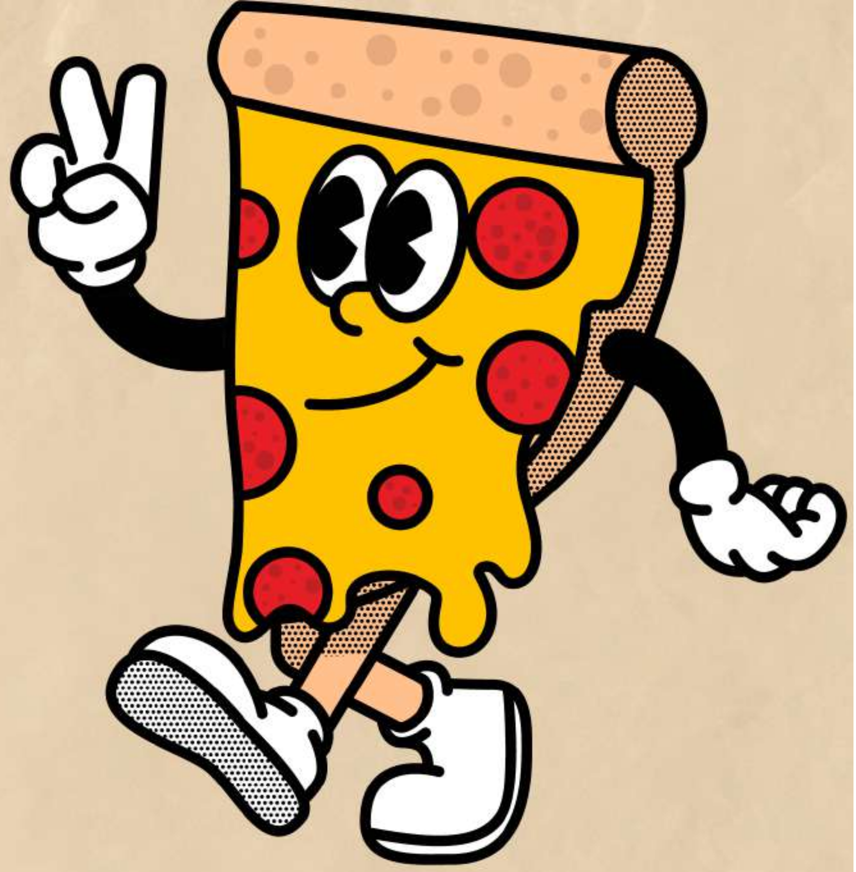
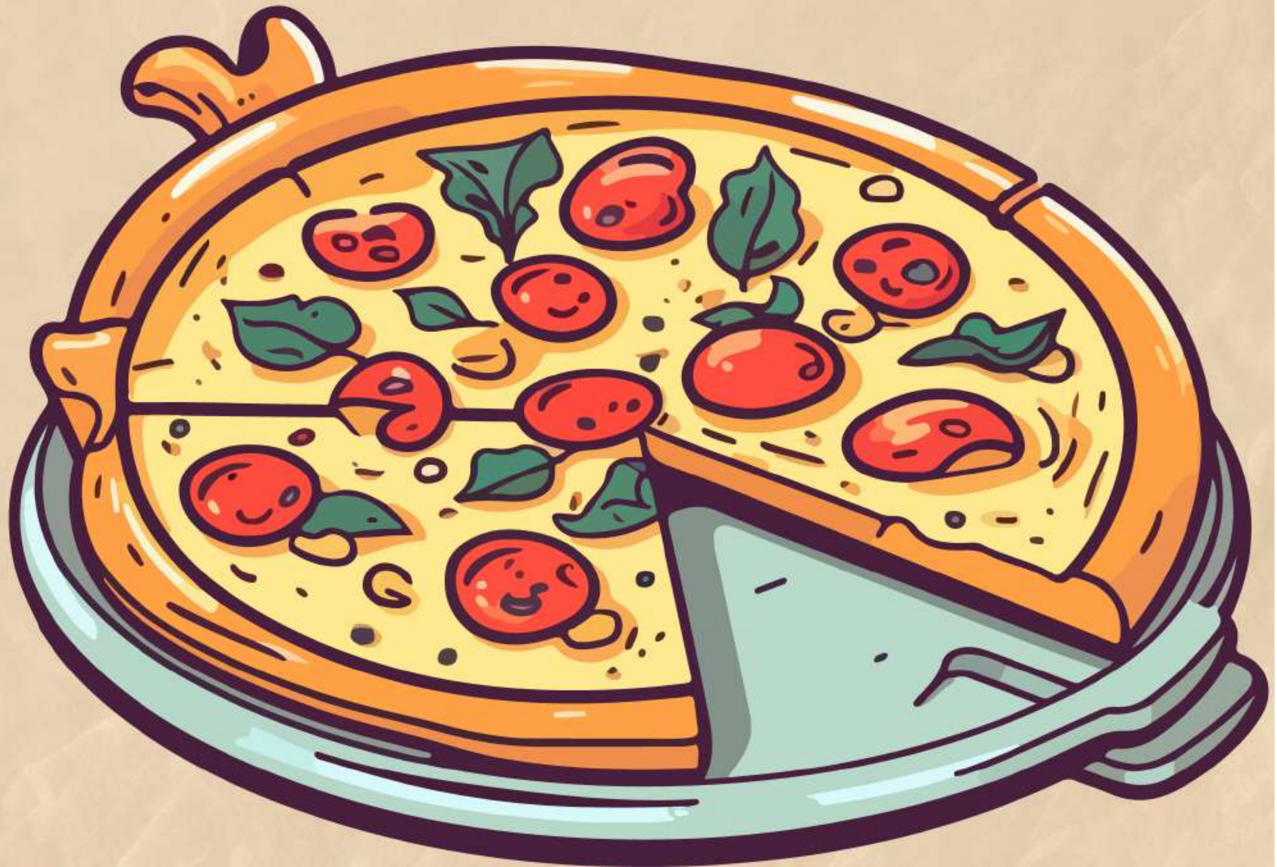
# Presentation

By Pratik Mishra





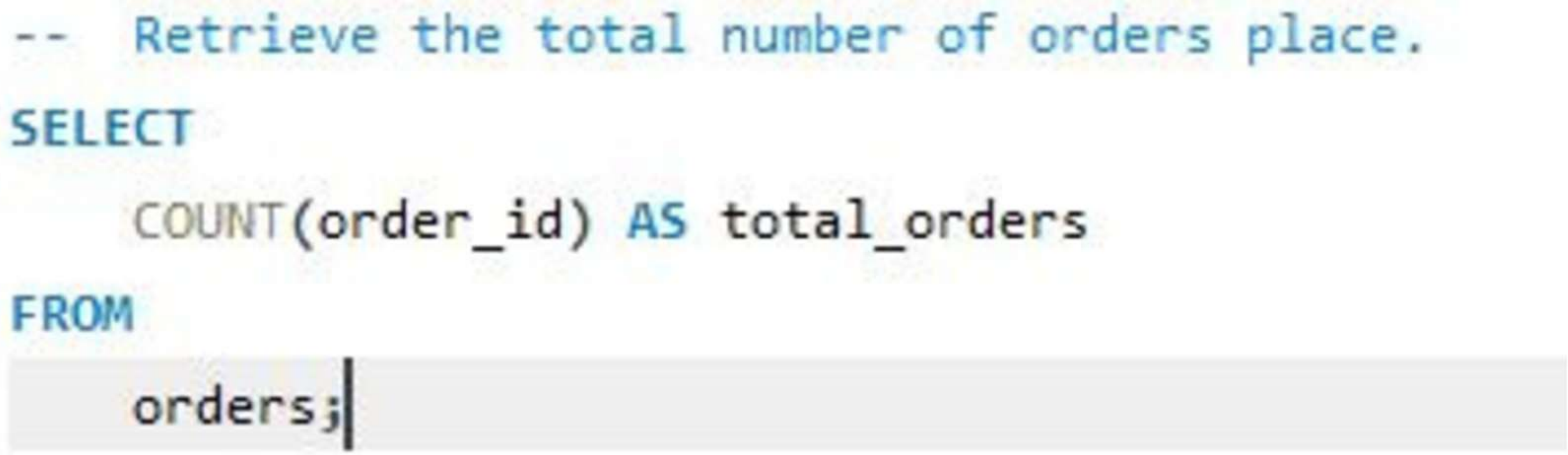
# ANALYZING PIZZA SALES DATA WITH MYSQL

**Welcome to our data**
**analysis project, where we explore pizza sales using MySQL to uncover key insights. This project aims to help us better understand customer preferences, sales patterns, and operational efficiency.**

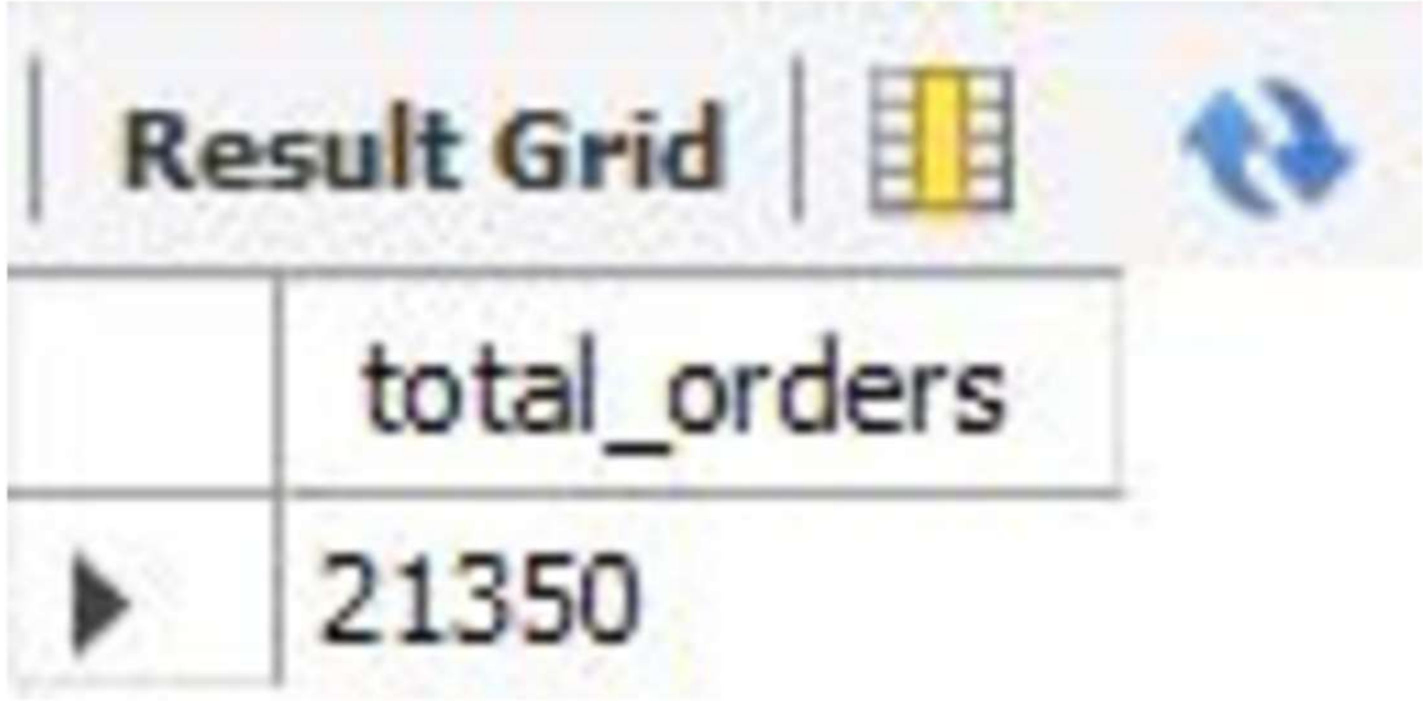
# PROJECT GOALS

Our goal is to leverage MySQL to analyze a dataset of pizza sales, providing valuable insights into sales trends, customer behavior, and product performance, which will guide data-driven decision-making for the business.

# RETRIEVE THE TOTAL NUMBER OF ORDERS PLACE.

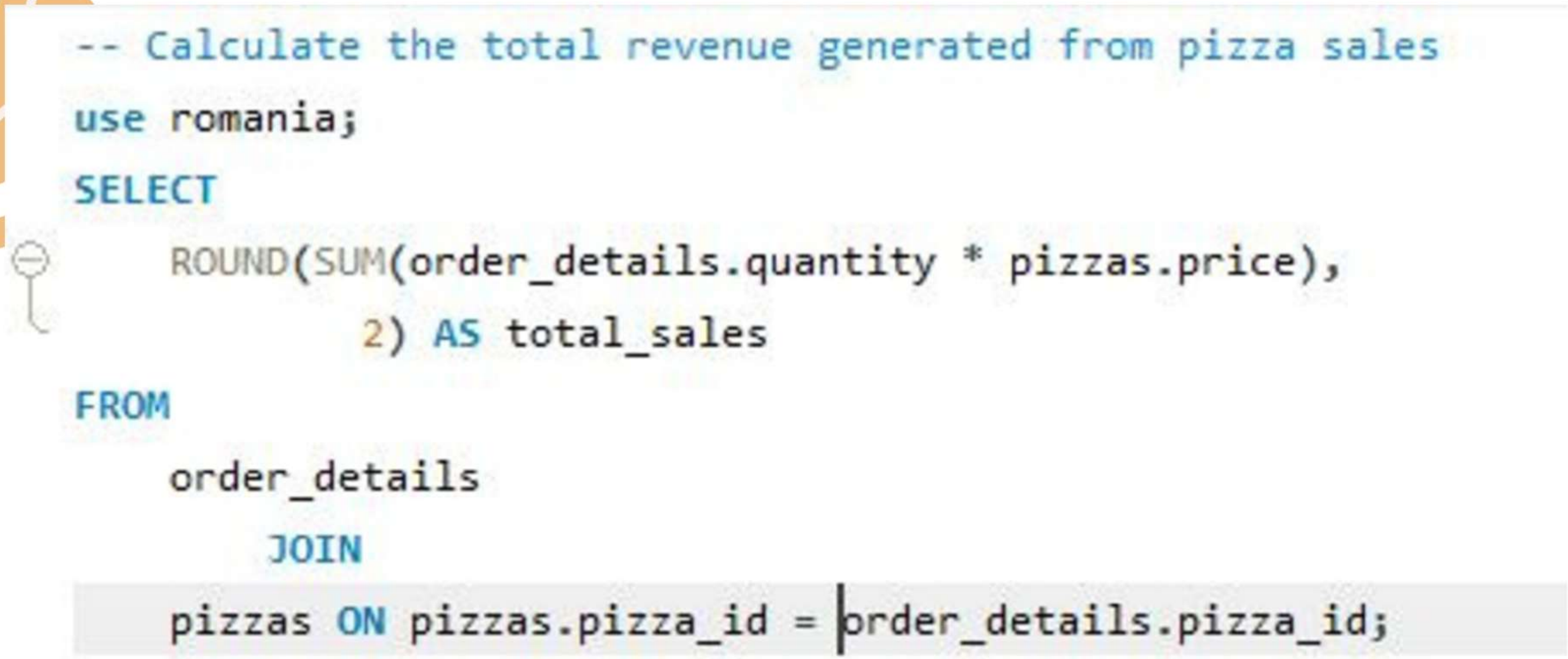


```
--  Retrieve the total number of orders place.
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

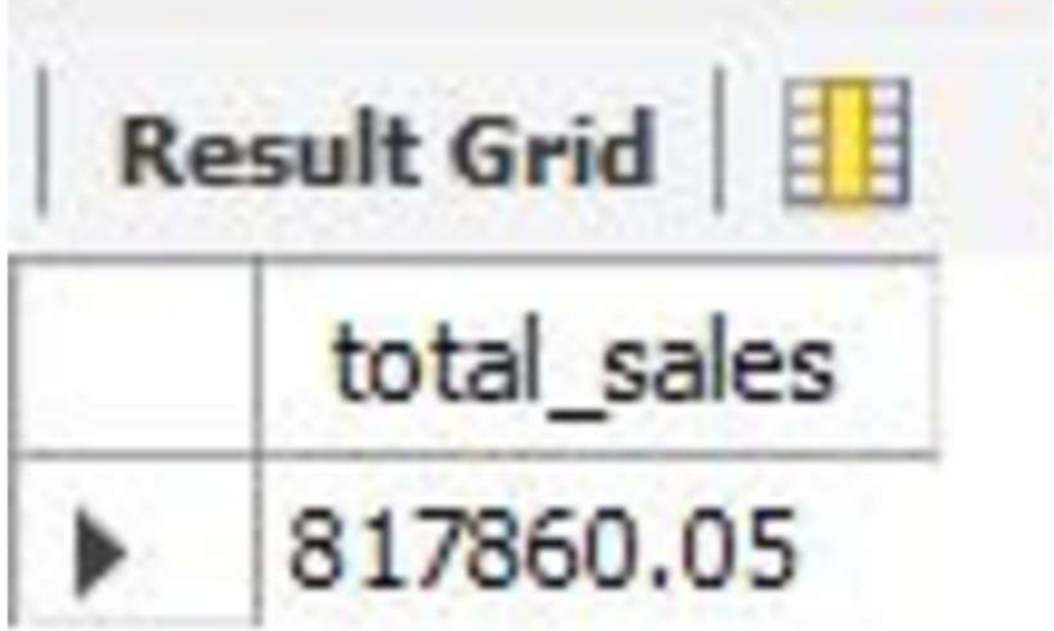


```
total_orders
21350
```

# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

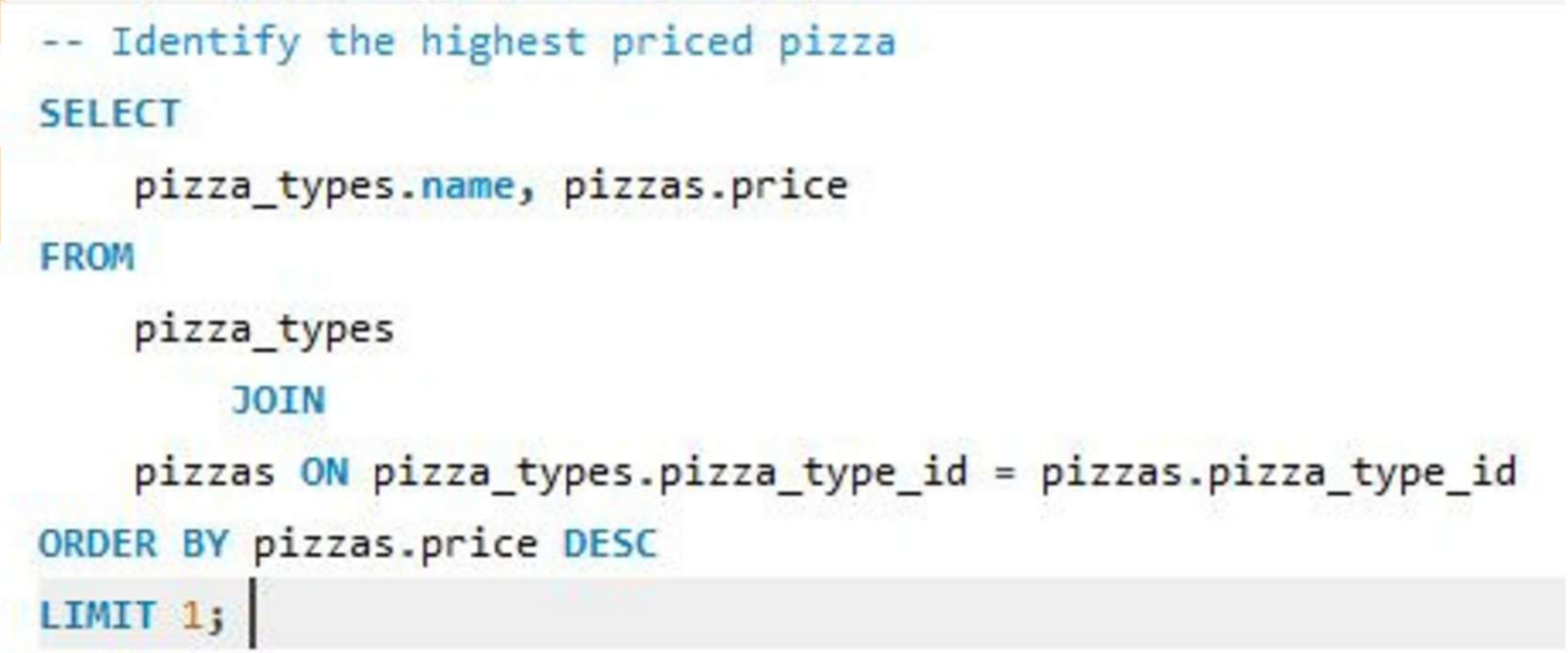


```
-- Calculate the total revenue generated from pizza sales
use romania;
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

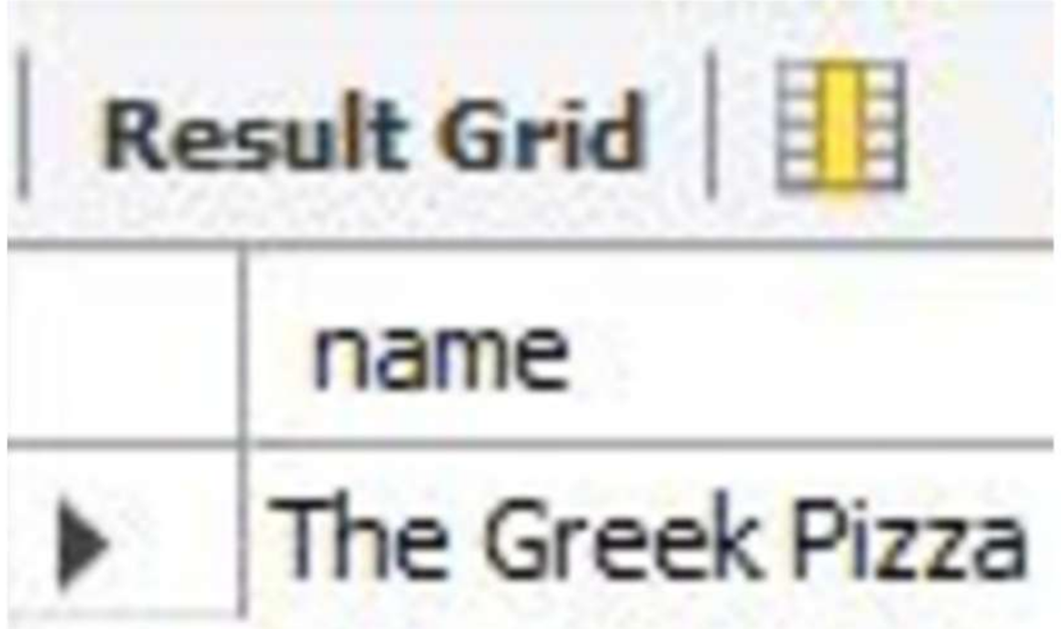


| total_sales |
|---|
| 817860.05 |

# IDENTIFY THE HIGHEST PRICED PIZZA

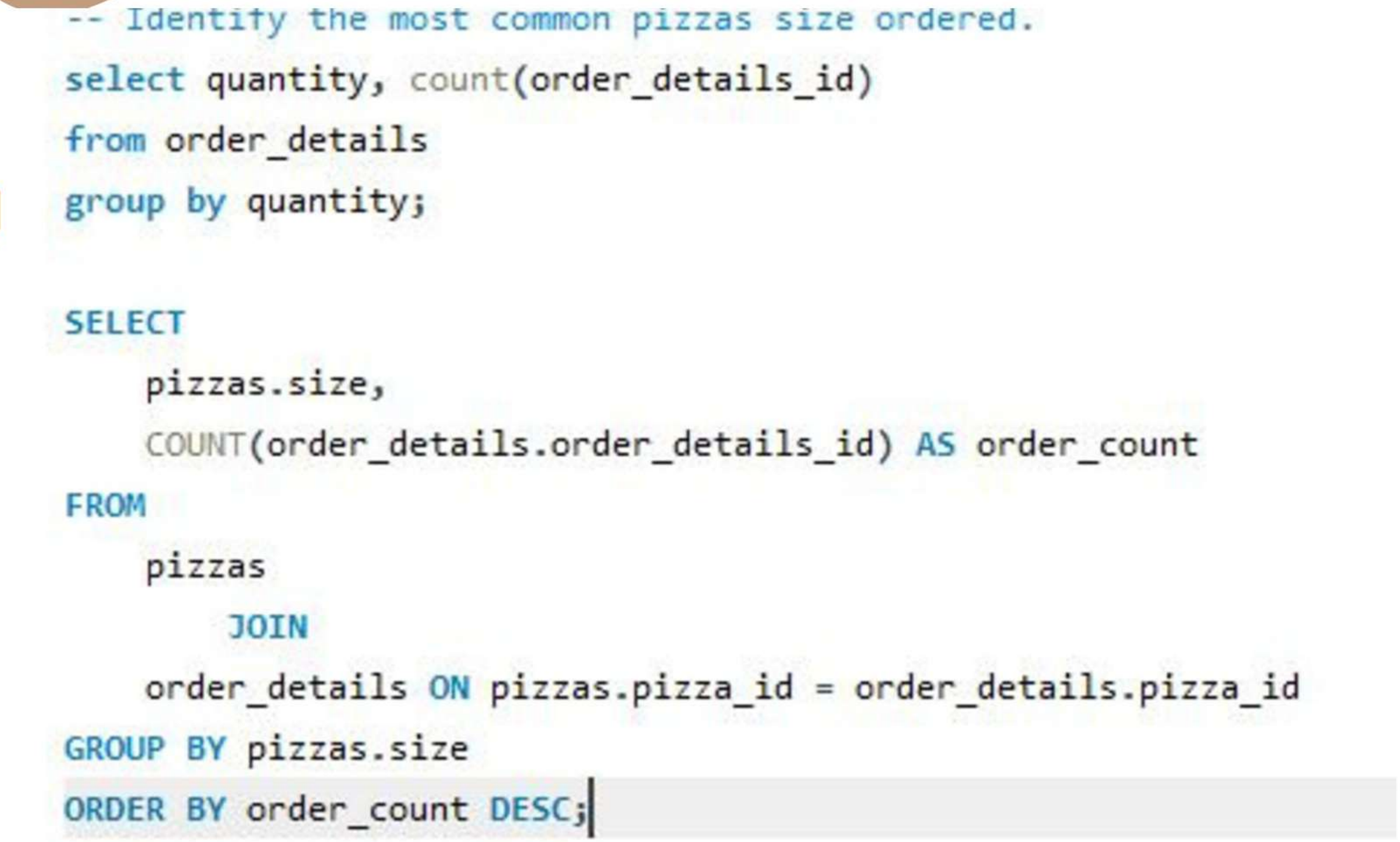


```
-- Identify the highest priced pizza
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```



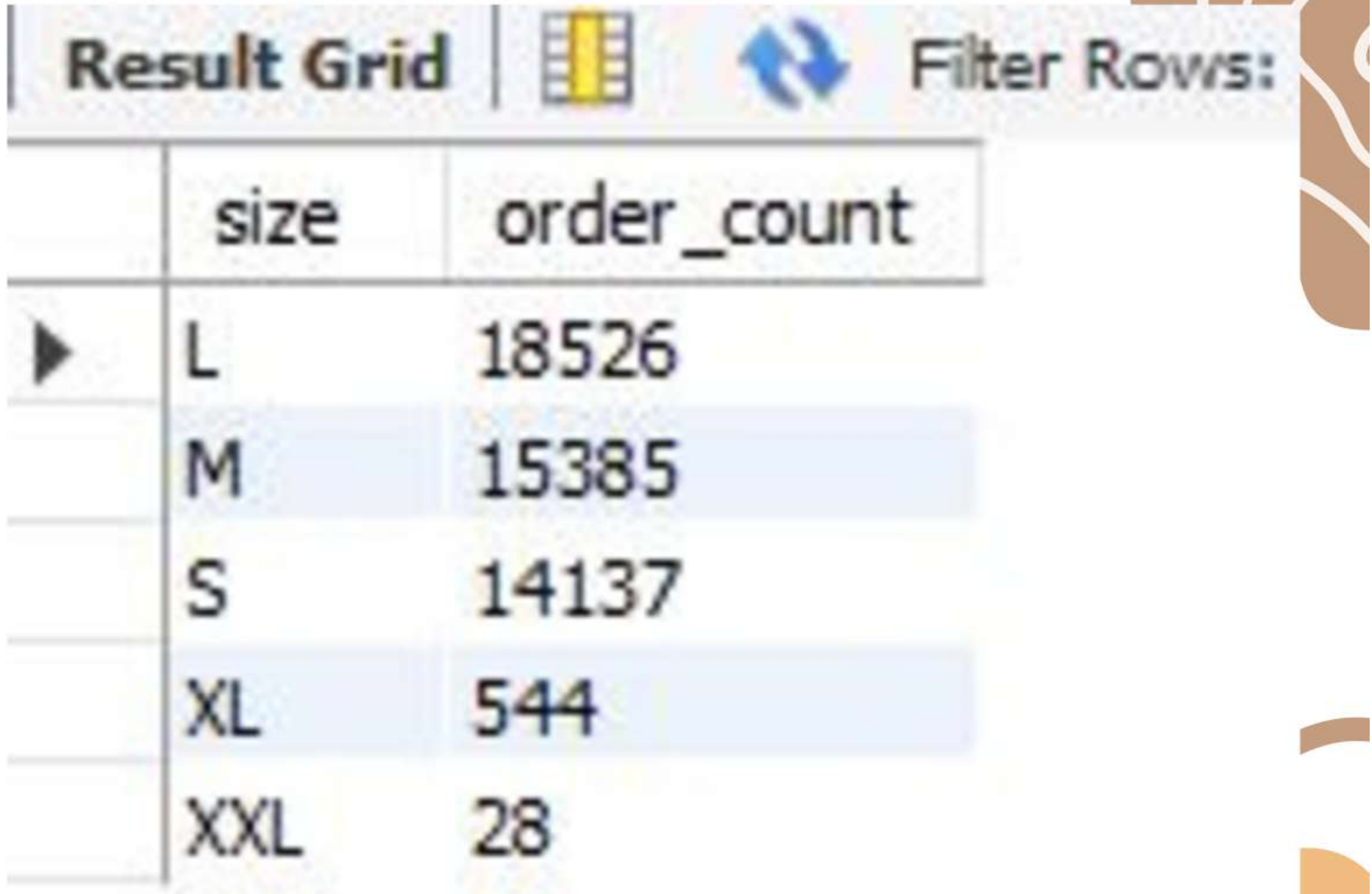
```
name
The Greek Pizza
```

# IDENTIFY THE MOST COMMON PIZZAS SIZE ORDERED.



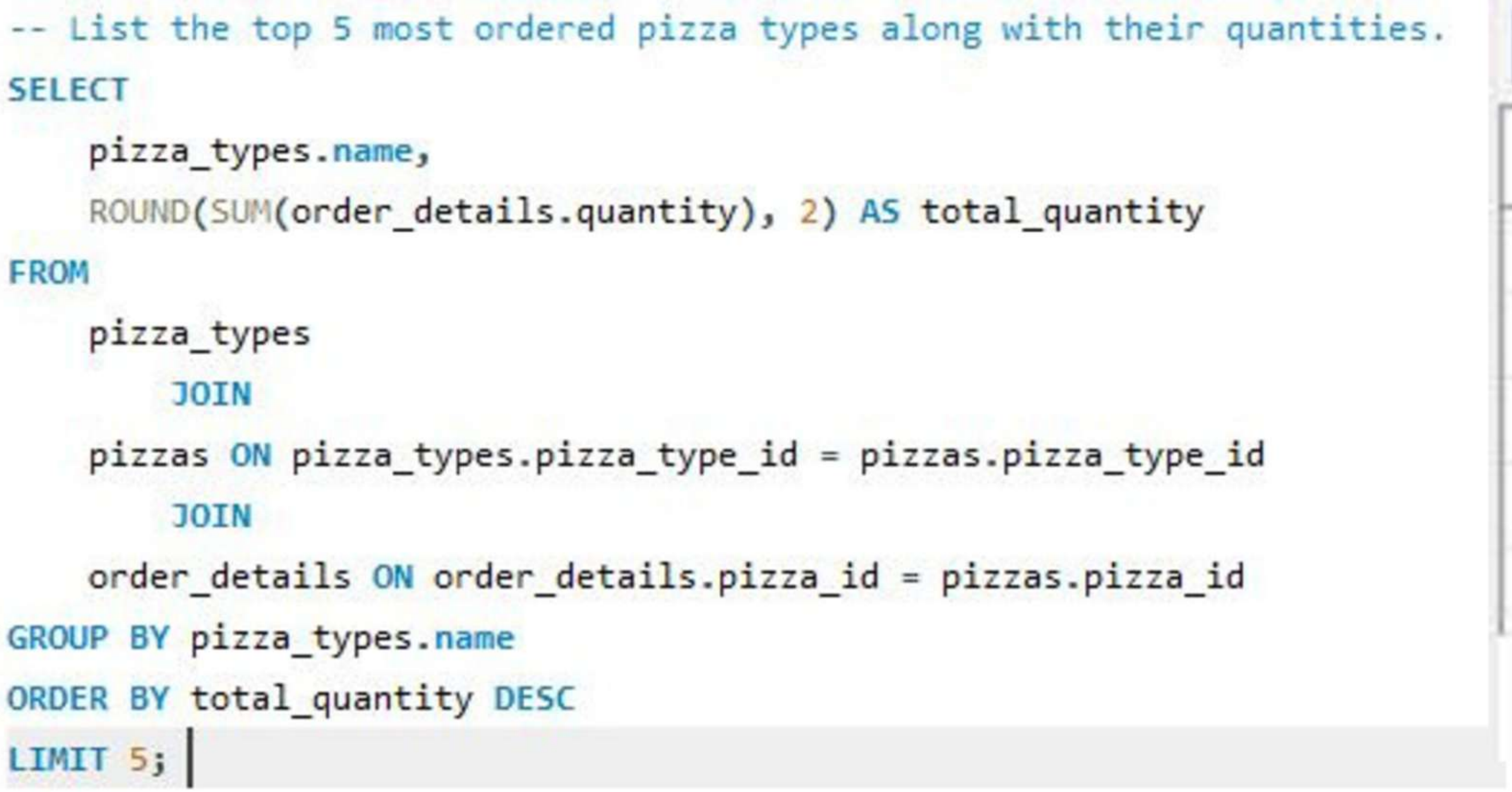
```
-- Identify the most common pizzas size ordered.
select quantity, count(order_details_id)
from order_details
group by quantity;

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;
```

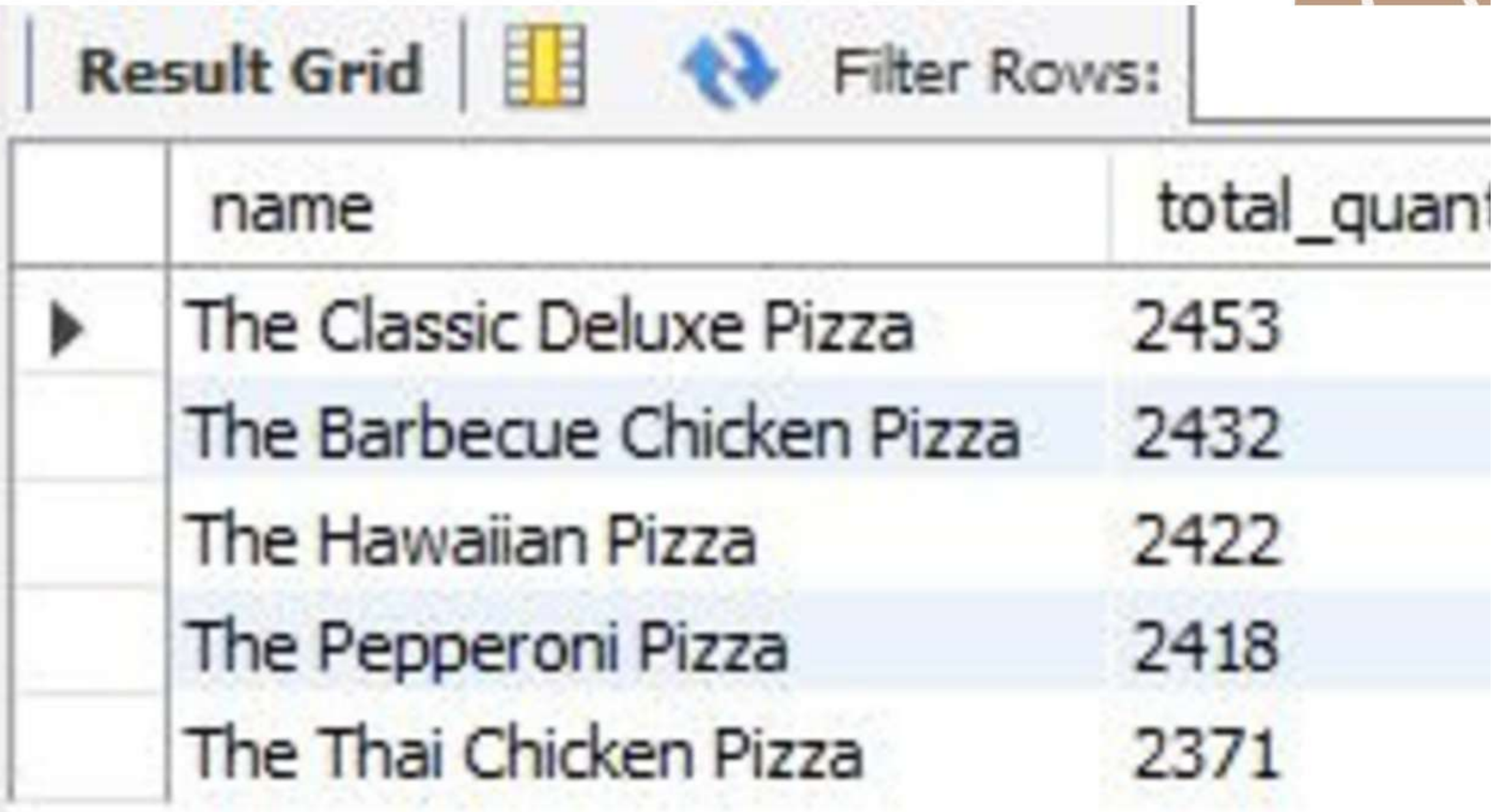


| size | order_count |
|---|---|
| L | 18526 |
| M | 15385 |
| S | 14137 |
| XL | 544 |
| XXL | 28 |

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

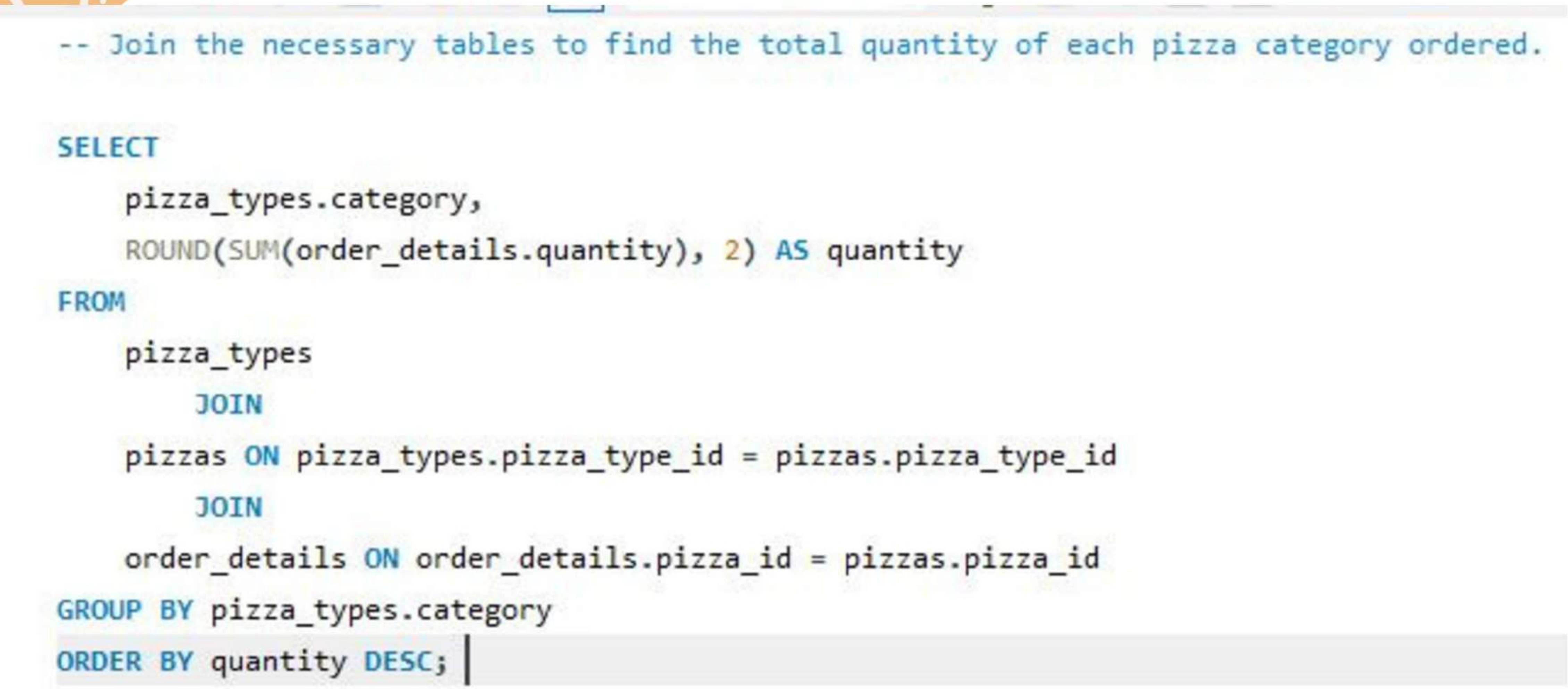


```
-- List the top 5 most ordered pizza types along with their quantities.
SELECT
    pizza_types.name,
    ROUND(SUM(order_details.quantity), 2) AS total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY total_quantity DESC
LIMIT 5;
```



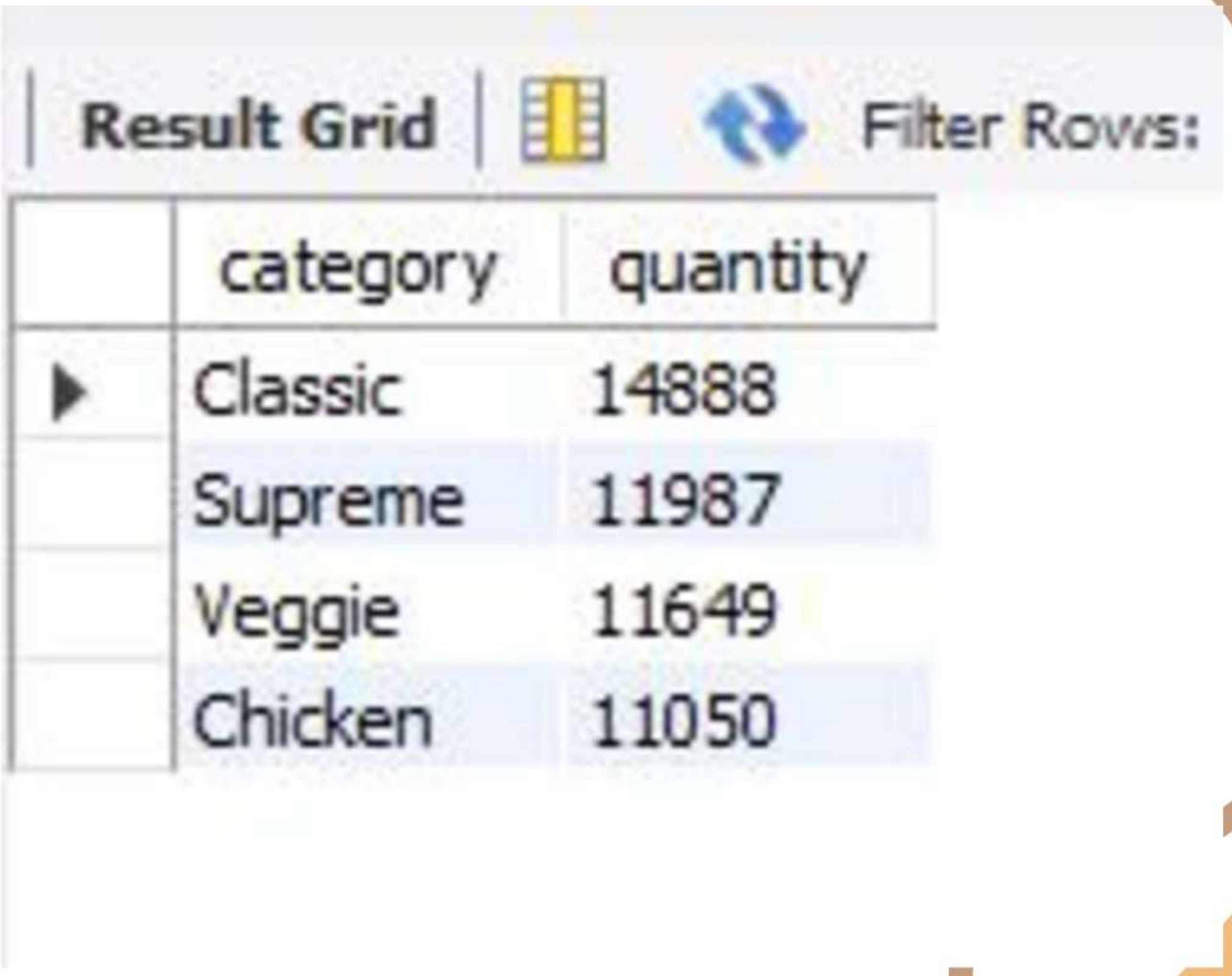
| name | total_quant |
|---|---|
| The Classic Deluxe Pizza | 2453 |
| The Barbecue Chicken Pizza | 2432 |
| The Hawaiian Pizza | 2422 |
| The Pepperoni Pizza | 2418 |
| The Thai Chicken Pizza | 2371 |

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



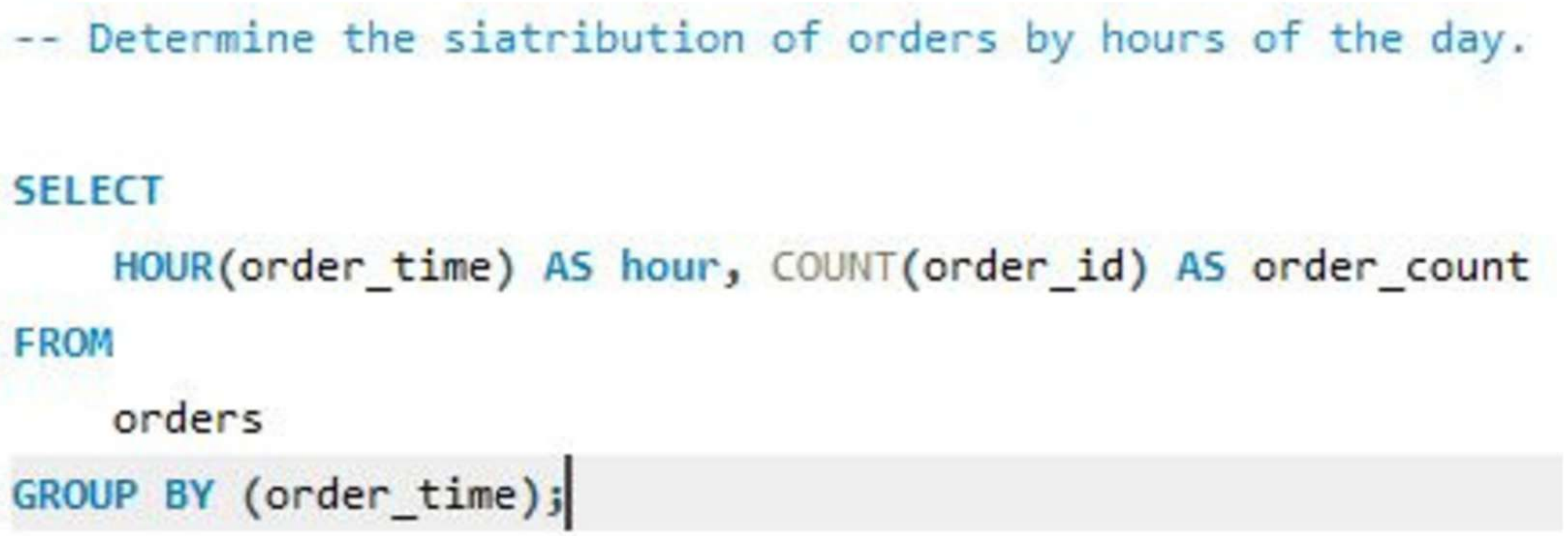
```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered.

SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity), 2) 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;
```



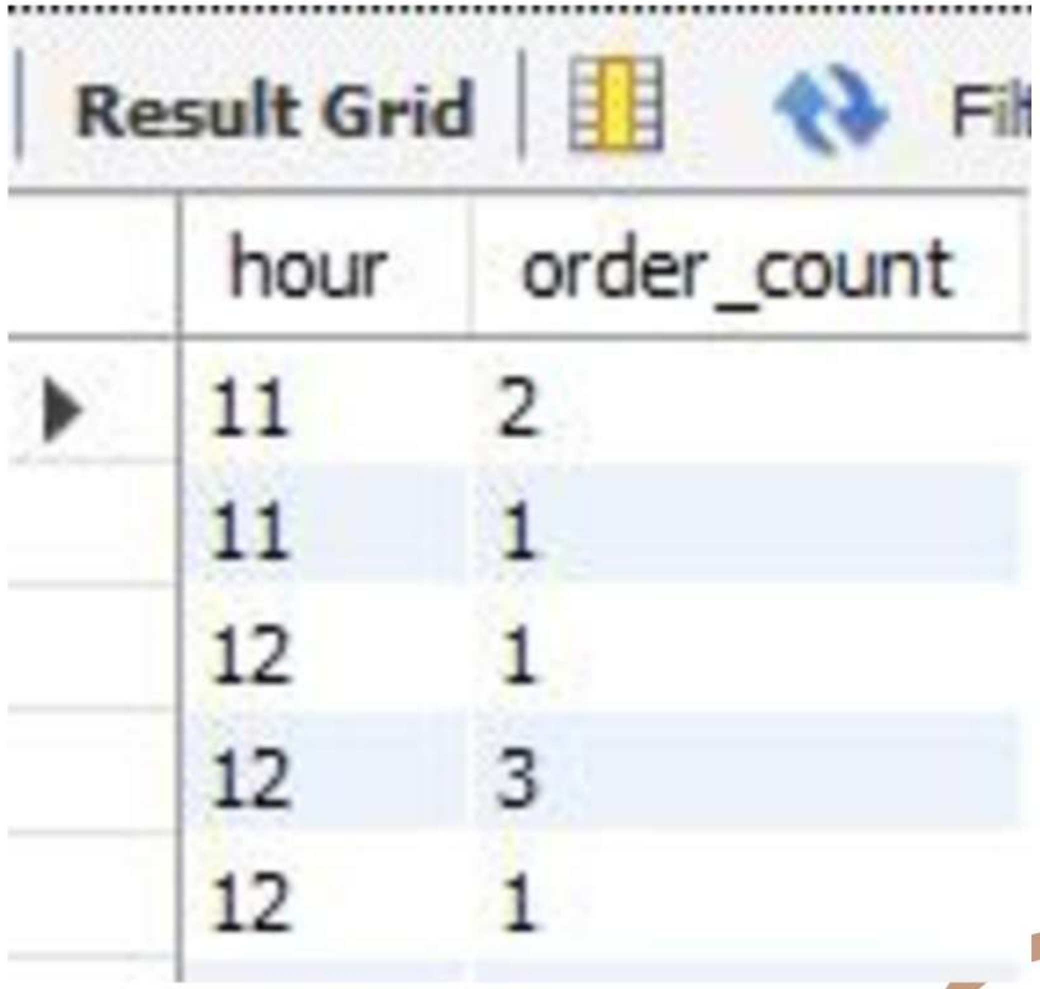
| | category | quantity |
|---|---|---|
| ▶ | Classic | 14888 |
| | Supreme | 11987 |
| | Veggie | 11649 |
| | Chicken | 11050 |

# DETERMINE THE SIATRIBUTION OF ORDERS BY HOURS OF THE DAY.



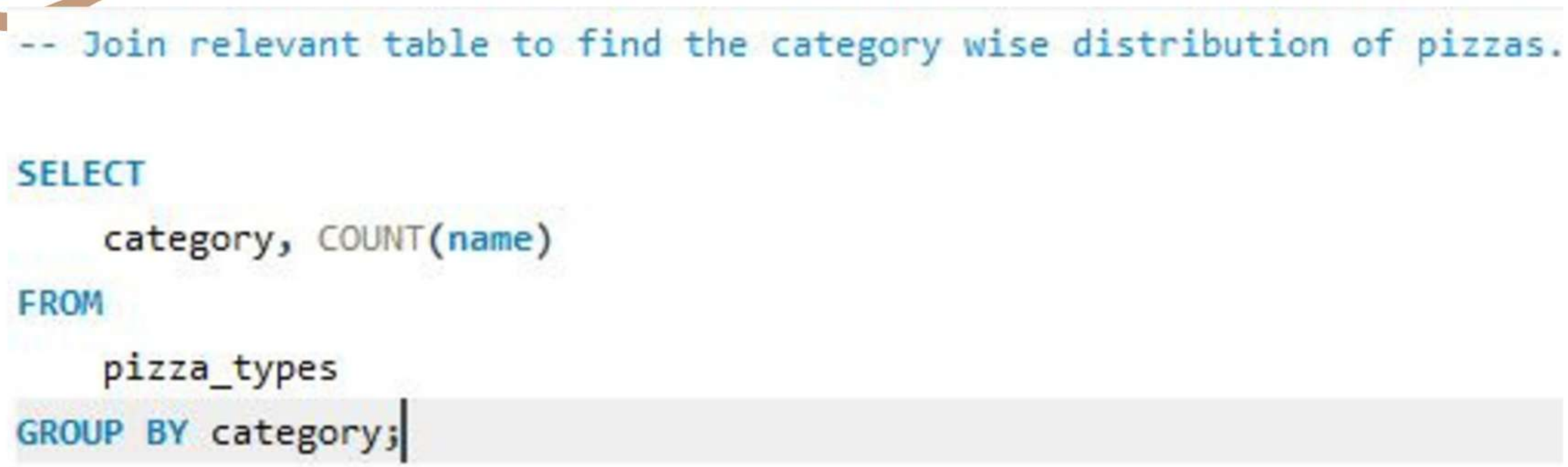
```sql
-- Determine the siatribution of orders by hours of the day.

SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY (order_time);
```



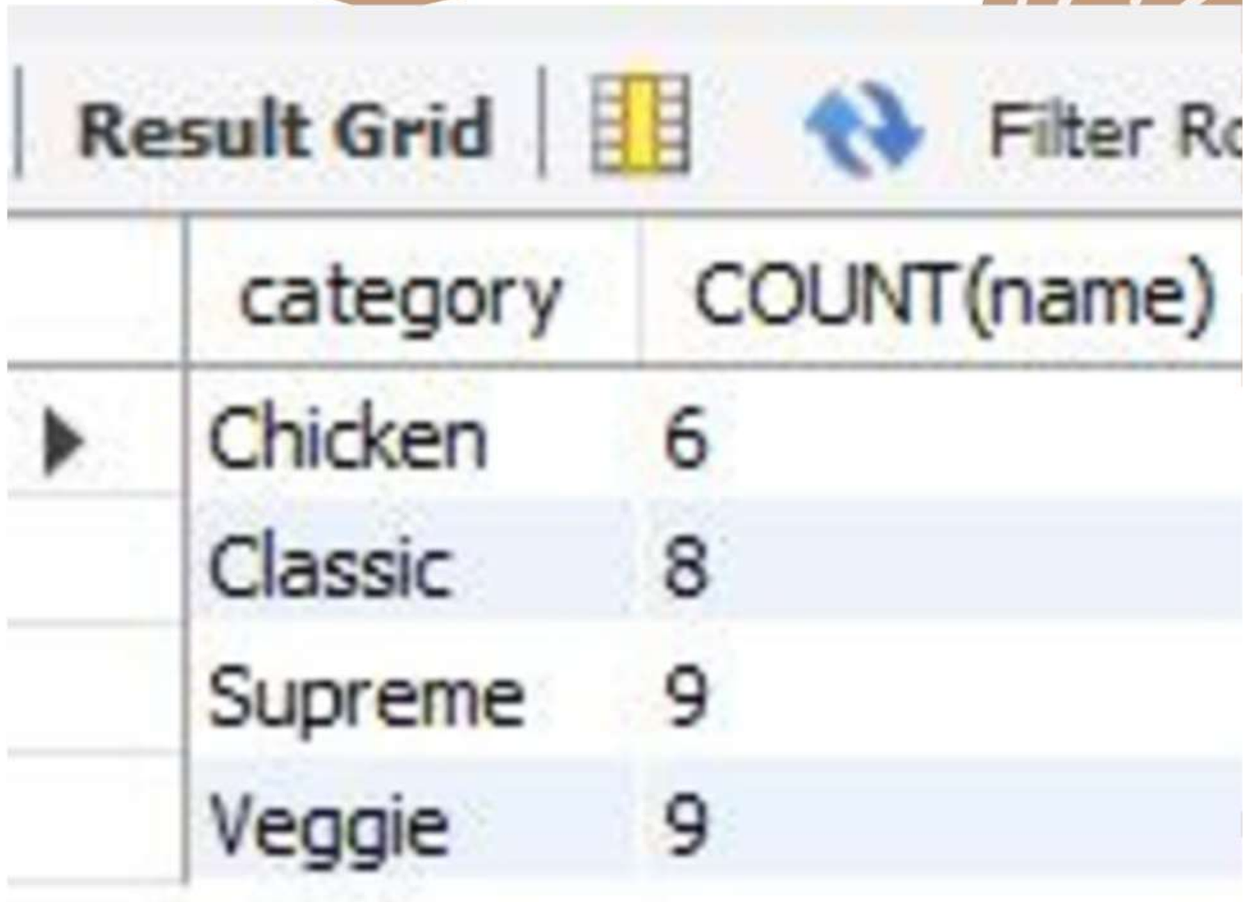
| hour | order_count |
|---|---|
| 11 | 2 |
| 11 | 1 |
| 12 | 1 |
| 12 | 3 |
| 12 | 1 |

# JOIN RELEVANT TABLE TO FIND THE CATEGORY WISE DISTRIBUTION OF PIZZAS.



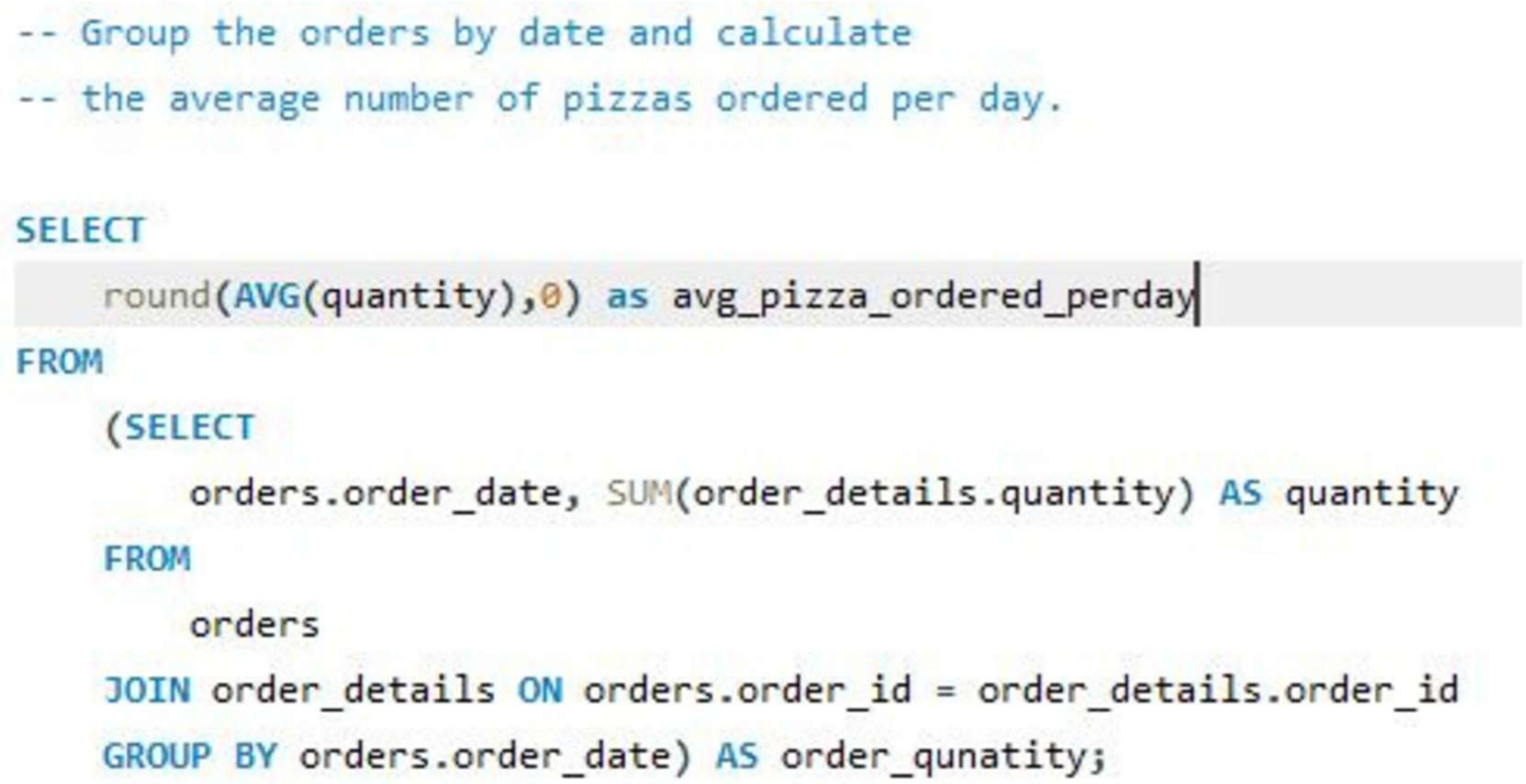
```
-- Join relevant table to find the category wise distribution of pizzas.

SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```



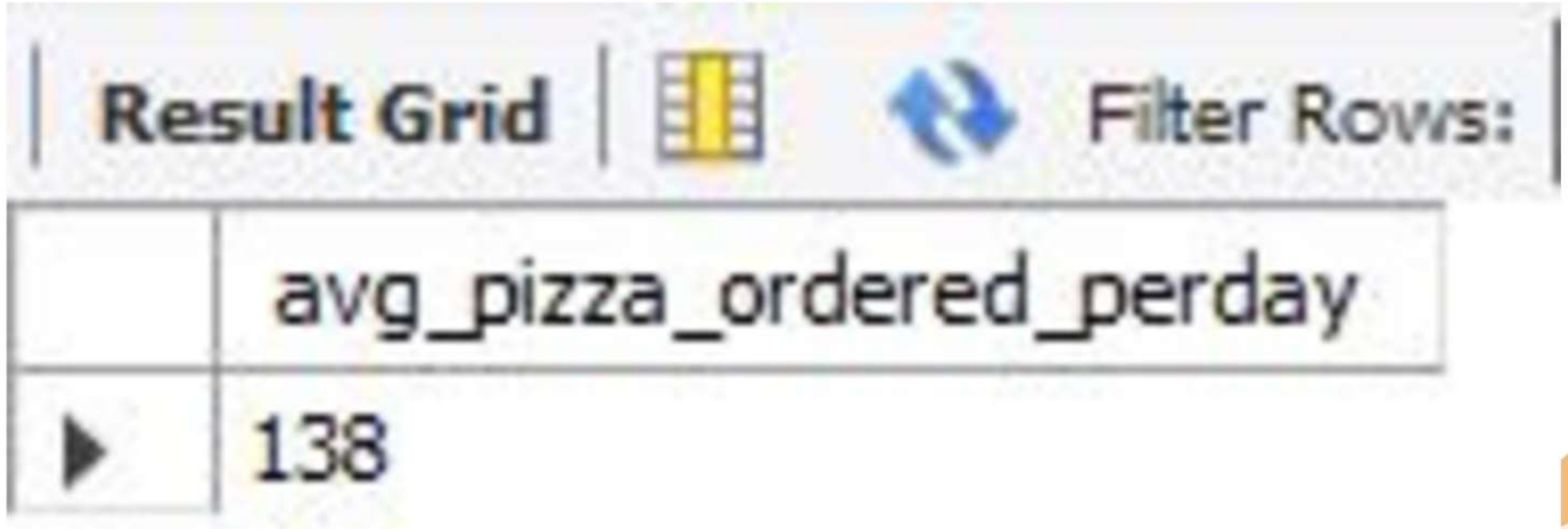
| category | COUNT(name) |
|---|---|
| Chicken | 6 |
| Classic | 8 |
| Supreme | 9 |
| Veggie | 9 |

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



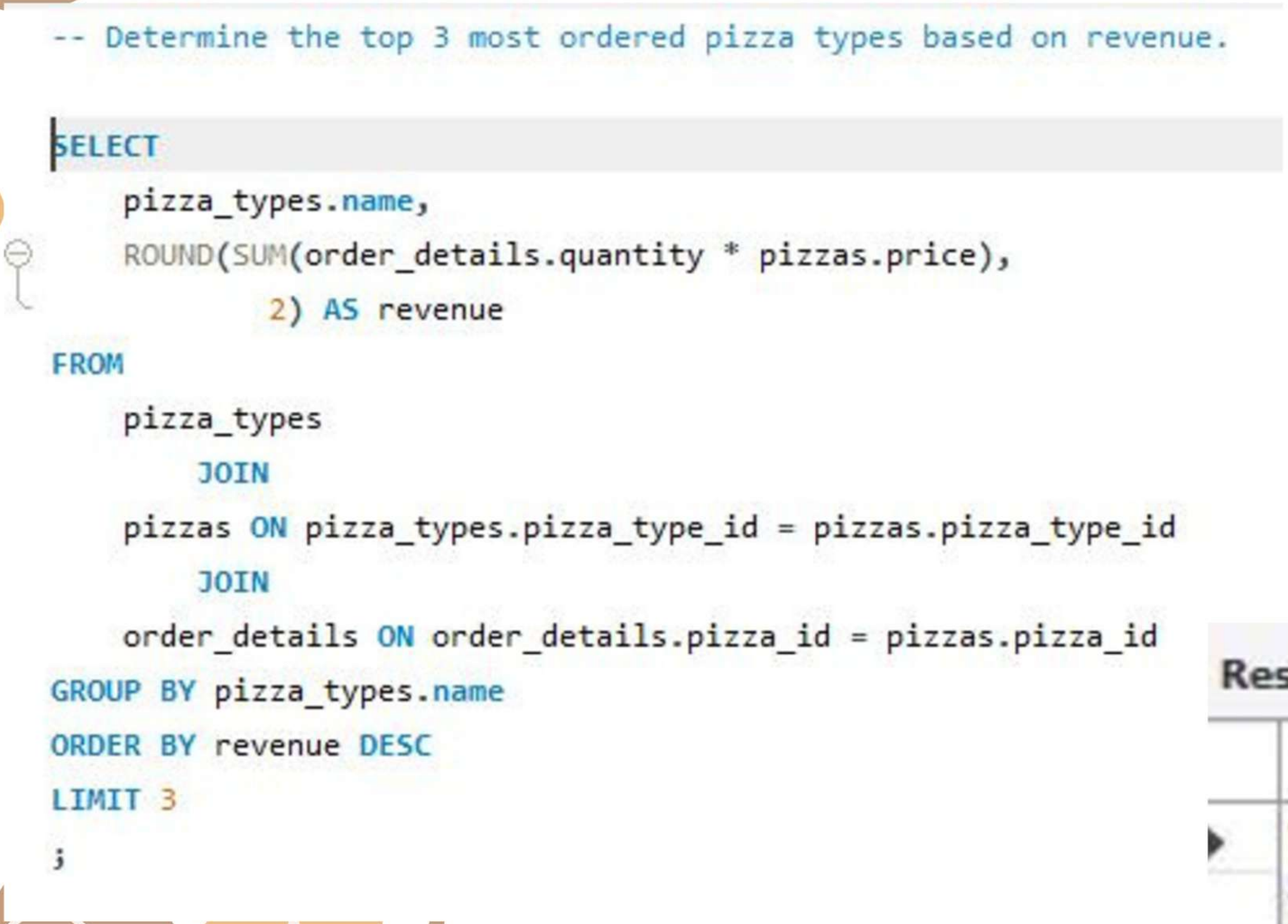
```sql
-- Group the orders by date and calculate
-- the average number of pizzas ordered per day.

SELECT
    round(AVG(quantity),0) as avg_pizza_ordered_perday
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_qunatity;
```



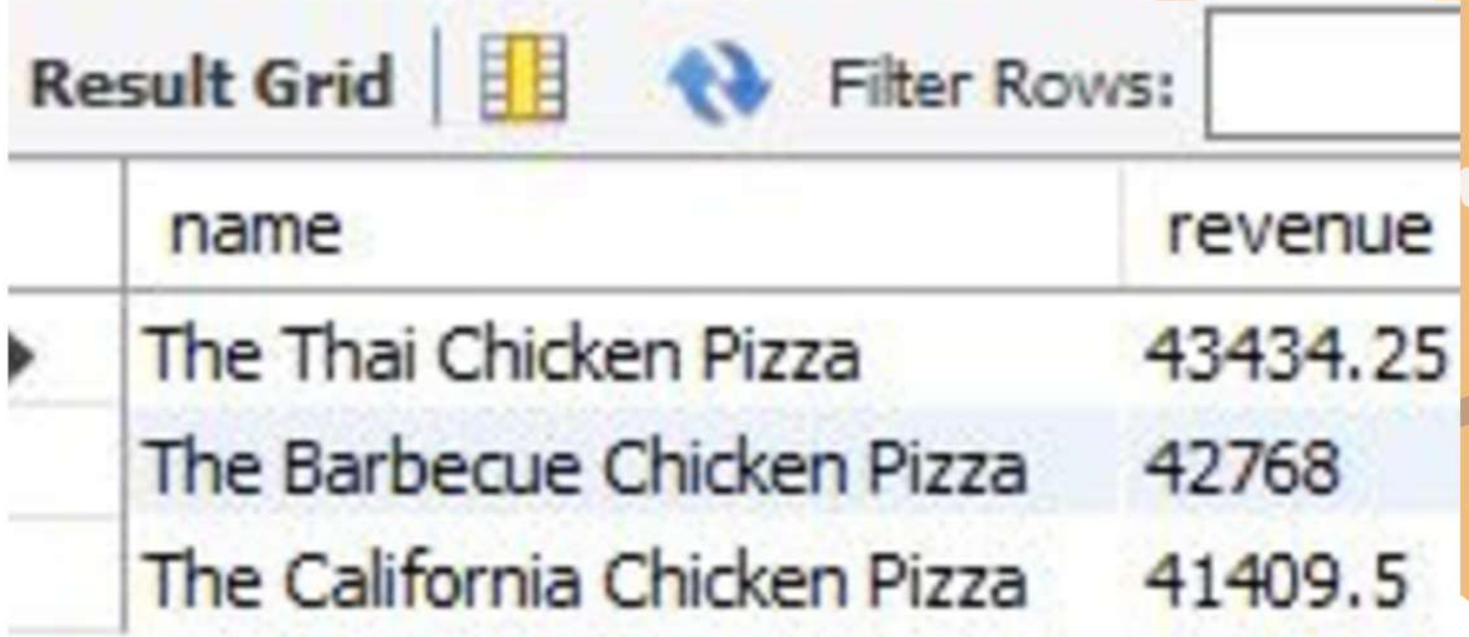
```
avg_pizza_ordered_perday
138
```

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



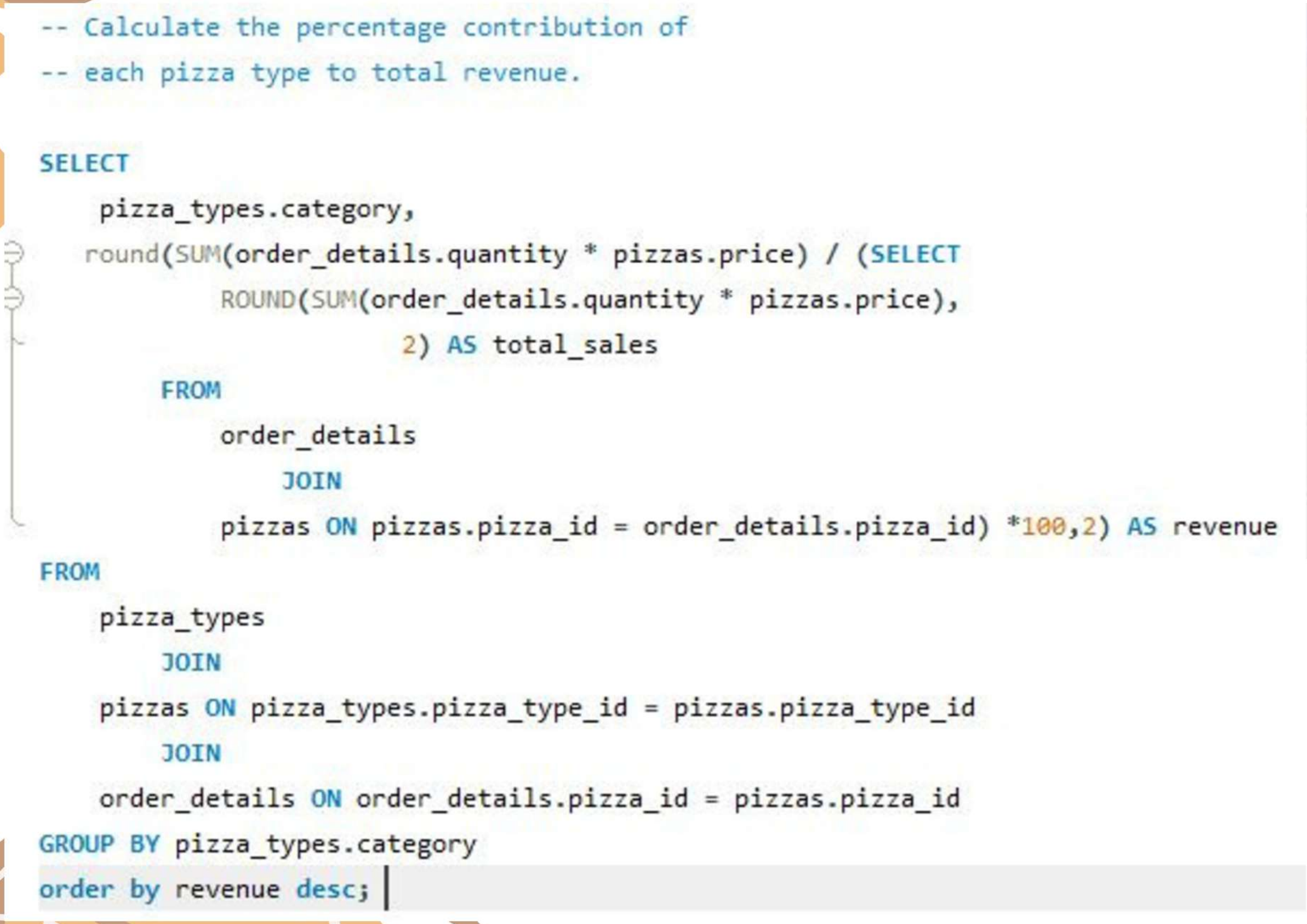
```
-- Determine the top 3 most ordered pizza types based on revenue.

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



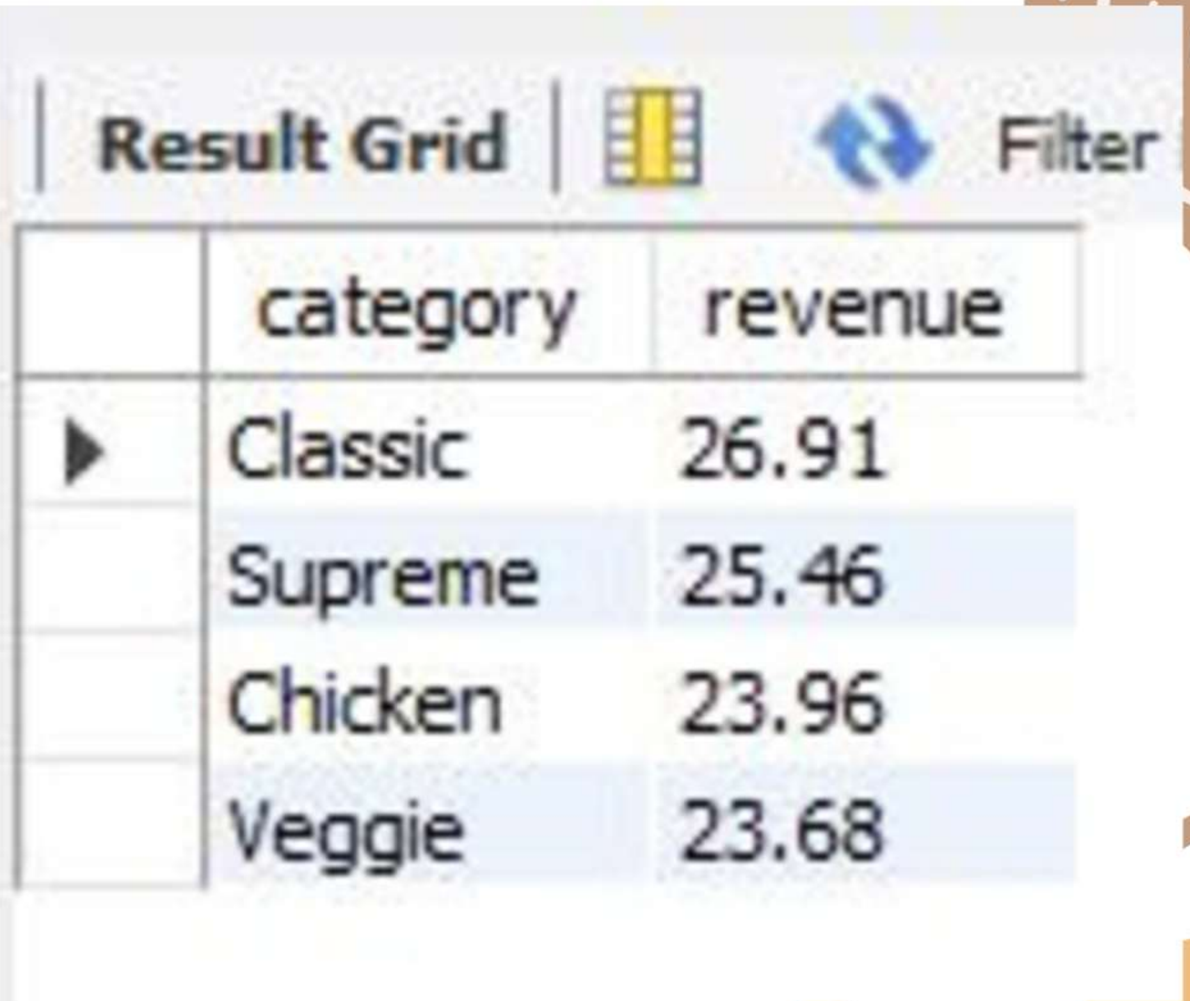
| name | revenue |
|---|---|
| The Thai Chicken Pizza | 43434.25 |
| The Barbecue Chicken Pizza | 42768 |
| The California Chicken Pizza | 41409.5 |

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



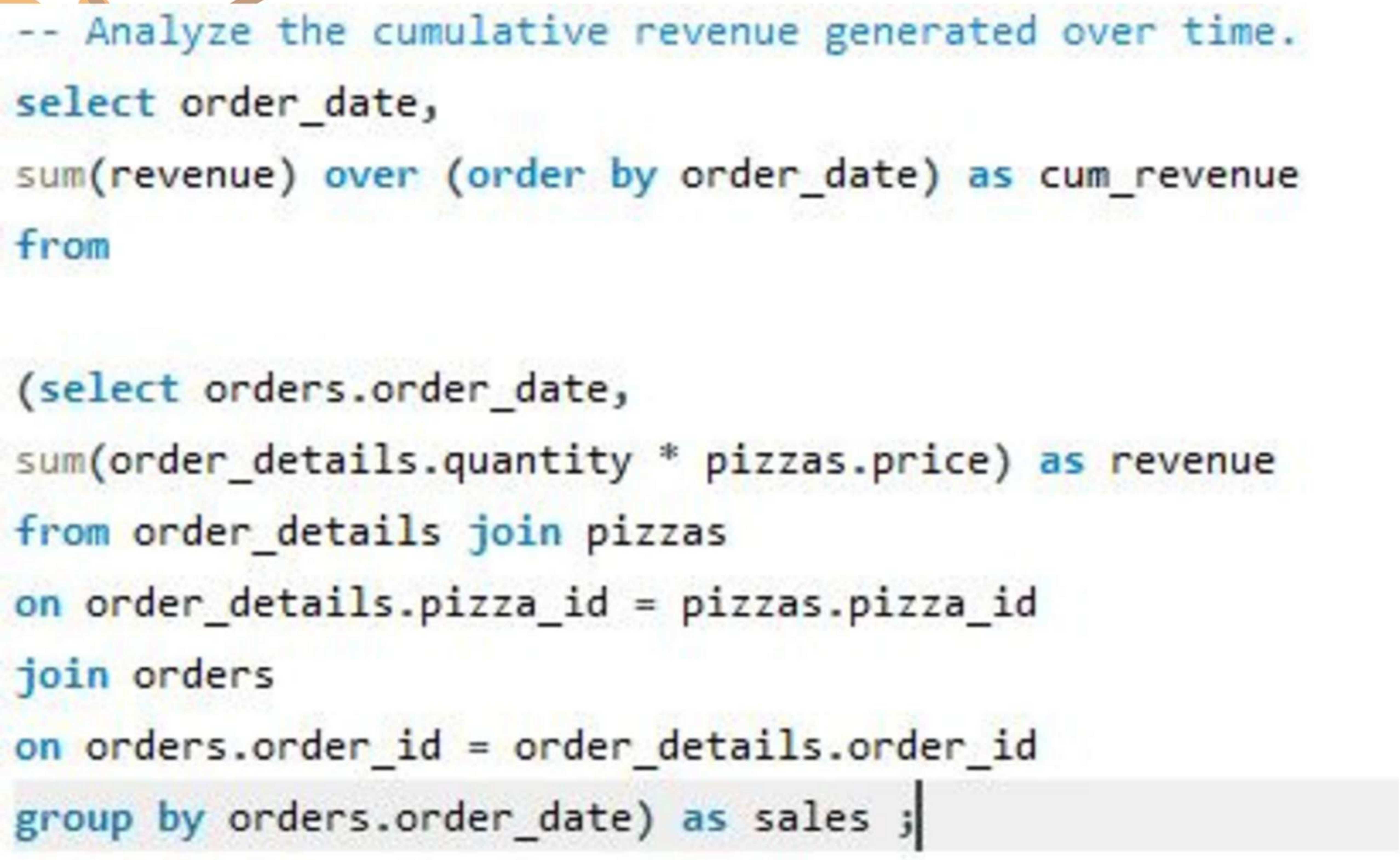
```
-- Calculate the percentage contribution of
-- each pizza type to total revenue.

SELECT
    pizza_types.category,
   round(SUM(order_details.quantity * pizzas.price) / (SELECT
            ROUND(SUM(order_details.quantity * pizzas.price),
                        2) AS total_sales
        FROM
            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;
```



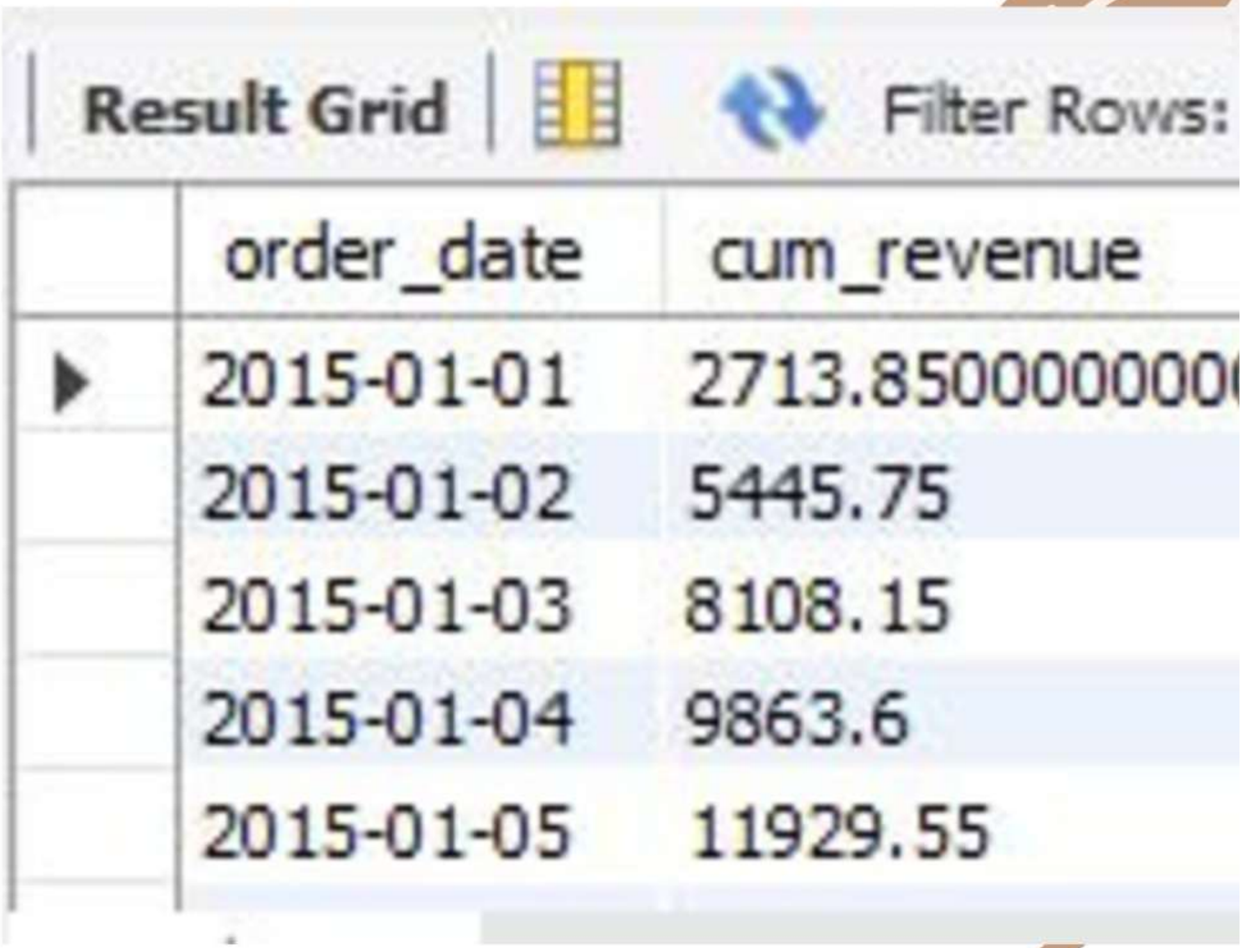
| category | revenue |
|---|---|
| Classic | 26.91 |
| Supreme | 25.46 |
| Chicken | 23.96 |
| Veggie | 23.68 |

# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



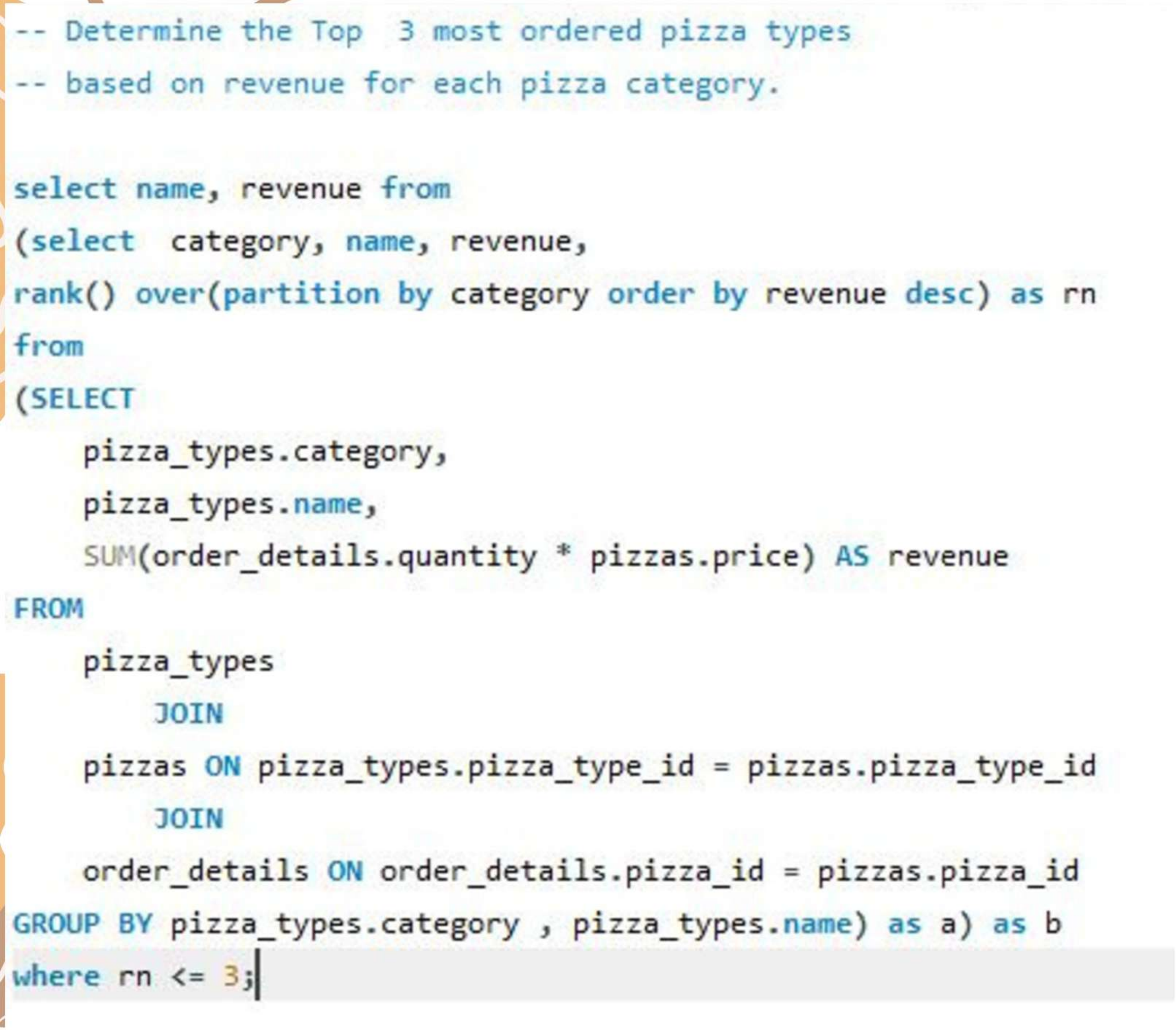
```
-- Analyze the cumulative revenue generated over time.
select order_date,
sum(revenue) over (order by order_date) as cum_revenue
from

(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 orders.order_id = order_details.order_id
group by orders.order_date) as sales ;
```



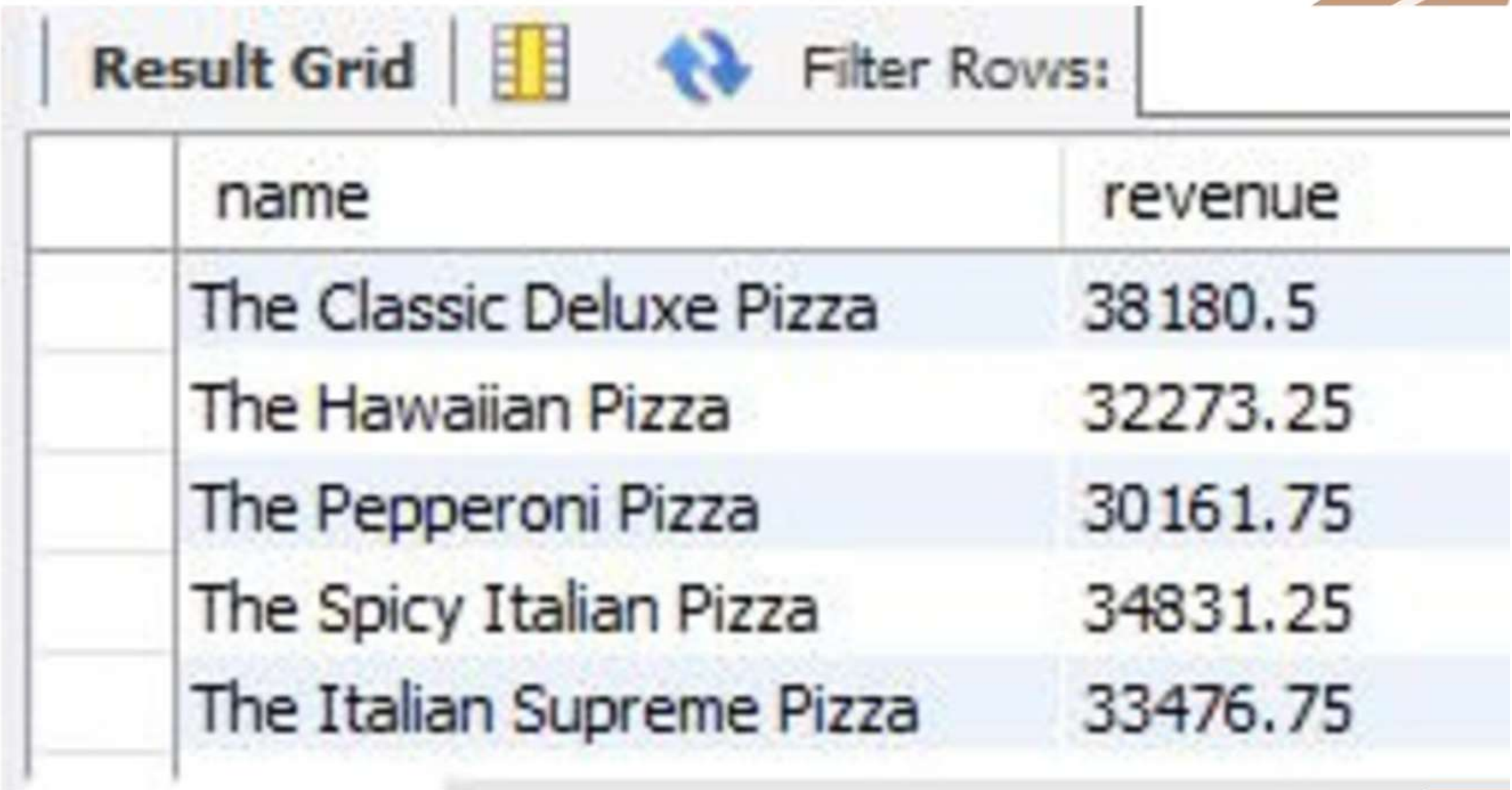
| order_date | cum_revenue |
|---|---|
| 2015-01-01 | 2713.8500000000 |
| 2015-01-02 | 5445.75 |
| 2015-01-03 | 8108.15 |
| 2015-01-04 | 9863.6 |
| 2015-01-05 | 11929.55 |

# 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 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 | revenue |
|---|---|
| The Classic Deluxe Pizza | 38180.5 |
| The Hawaiian Pizza | 32273.25 |
| The Pepperoni Pizza | 30161.75 |
| The Spicy Italian Pizza | 34831.25 |
| The Italian Supreme Pizza | 33476.75 |

# THANK YOU