Query Analysis

# PIZZA SALES SQL PROJECT

MAY-2024

# PROJECT OVERVIEW:

This project aims to analyze pizza sales data using SQL to extract actionable business insights. The data includes various aspects such as order information, pizza types, sizes, prices, and the corresponding times the orders were placed. By crafting a series of SQL queries, I delved into customer behavior, sales trends, and product performance, helping to uncover valuable insights that could guide business decisions.

# INSIGHTS AND ANALYSIS

01 Total Orders

02 Total Revenue

03 Highest-Priced Pizza

04 Top 5 Most Ordered Pizza Types

05 Order Distribution by Hour

06 Category-Wise Pizza Distribution

07 Orders Grouped by Date

08 Top 3 Most Ordered Pizzas by Revenue

09 Percentage Contribution to Revenue

10 Cumulative Revenue Analysis

11 Top 3 Pizzas by Revenue for Each Category

# BASIC ANALYSIS

01 Total Orders

02 Total Revenue

03 Highest-Priced Pizza

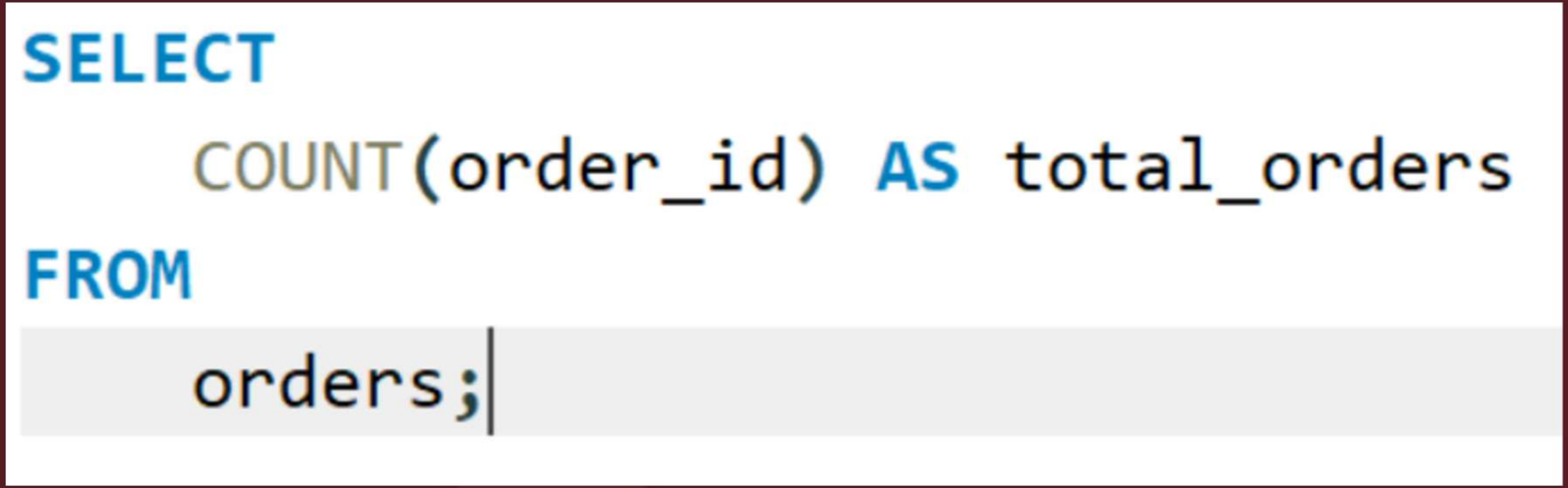
04 Top 5 Most Ordered Pizza Types

# SQL TECHNIQUES EMPLOYED

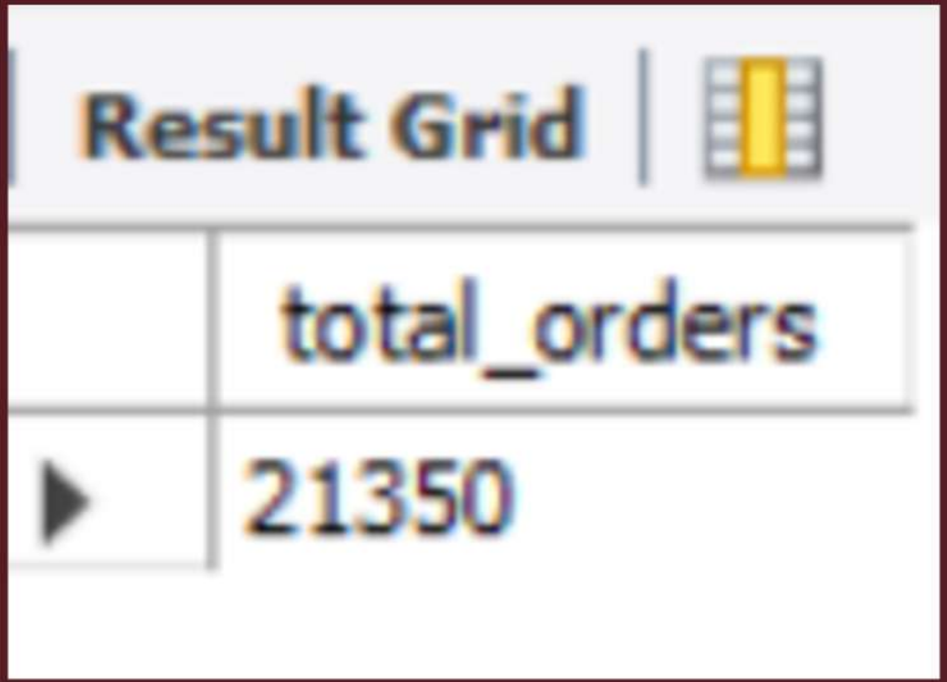
- Aggregation Functions: I used SUM(), COUNT(), AVG(), and other aggregate functions to summarize data effectively.
- JOINS: By performing various types of joins between tables, I was able to combine data across different dimensions such as order details, pizza details, and categories.
- Group By and Date Functions: Grouping data by date and time helped to uncover trends and patterns in sales activity.
- Window Functions: I utilized window functions for more advanced analysis, including ranking pizzas based on revenue and calculating cumulative totals.

# TOTAL ORDERS:

A query was used to retrieve the total number of orders placed over the observed period, providing a general measure of business volume.



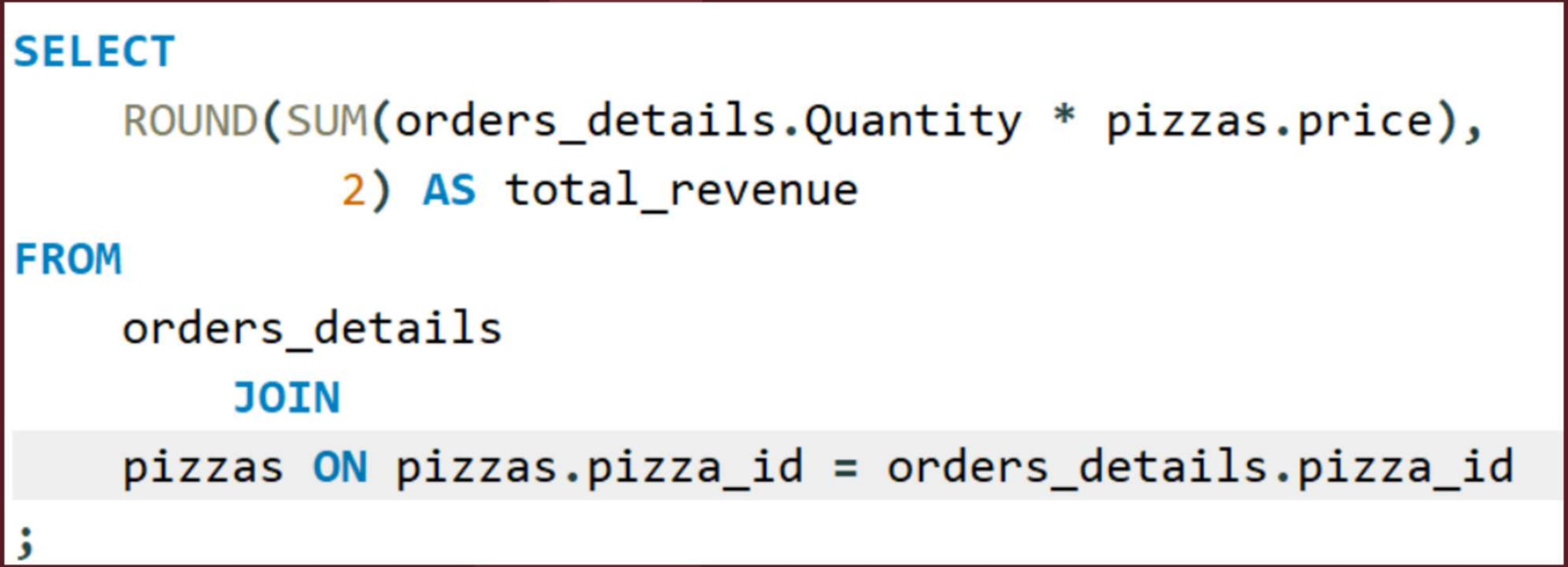
```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```



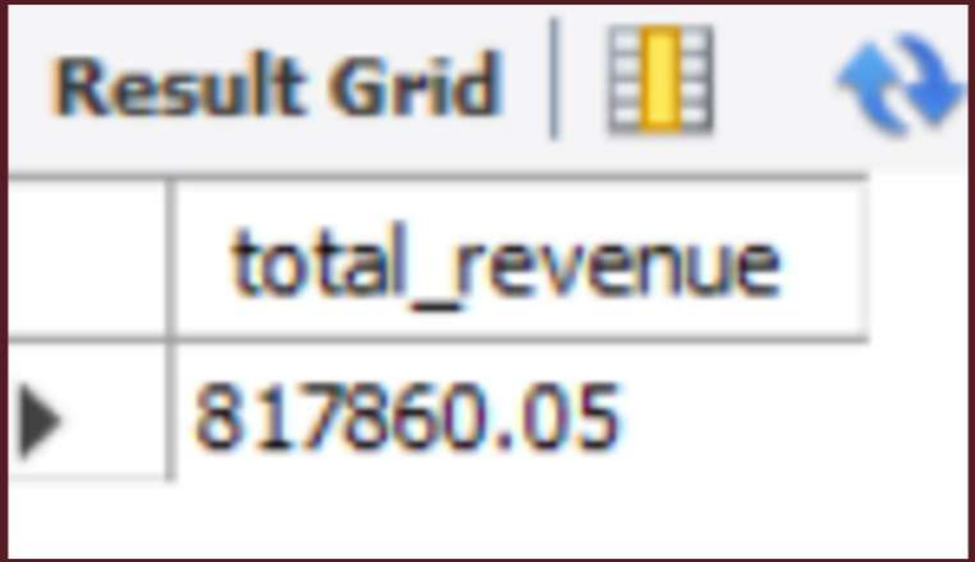
| total_orders |
|---|
| 21350 |

# TOTAL REVENUE:

By summing up the revenue generated from all pizza sales, I calculated the total earnings. This metric serves as a key indicator of business performance.



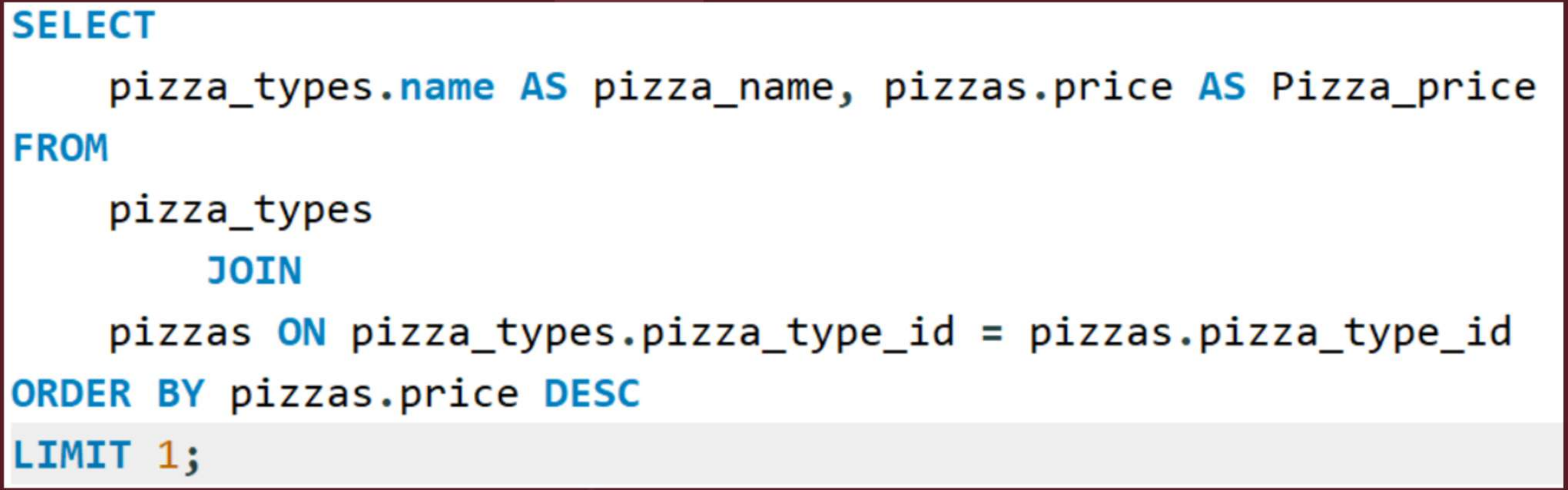
```
SELECT
    ROUND(SUM(orders_details.Quantity * pizzas.price),
            2) AS total_revenue
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
;
```



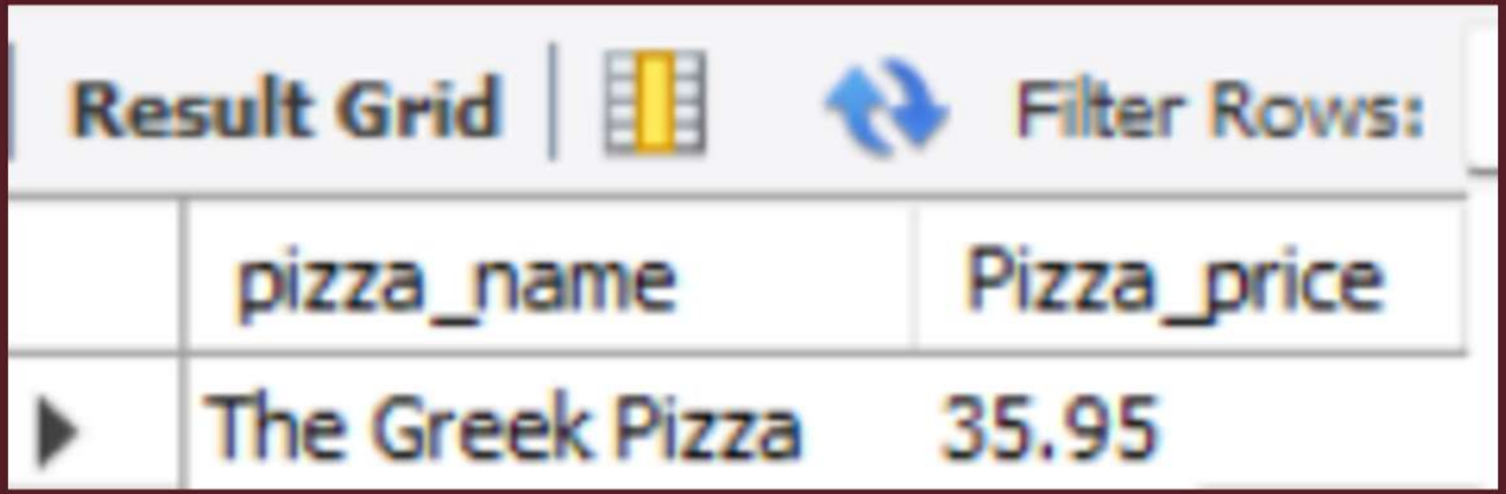
| total_revenue |
|---|
| 817860.05 |

# HIGHEST-PRICED PIZZA:

Identifying the most expensive pizza helped highlight premium offerings and potential profit drivers within the product lineup.



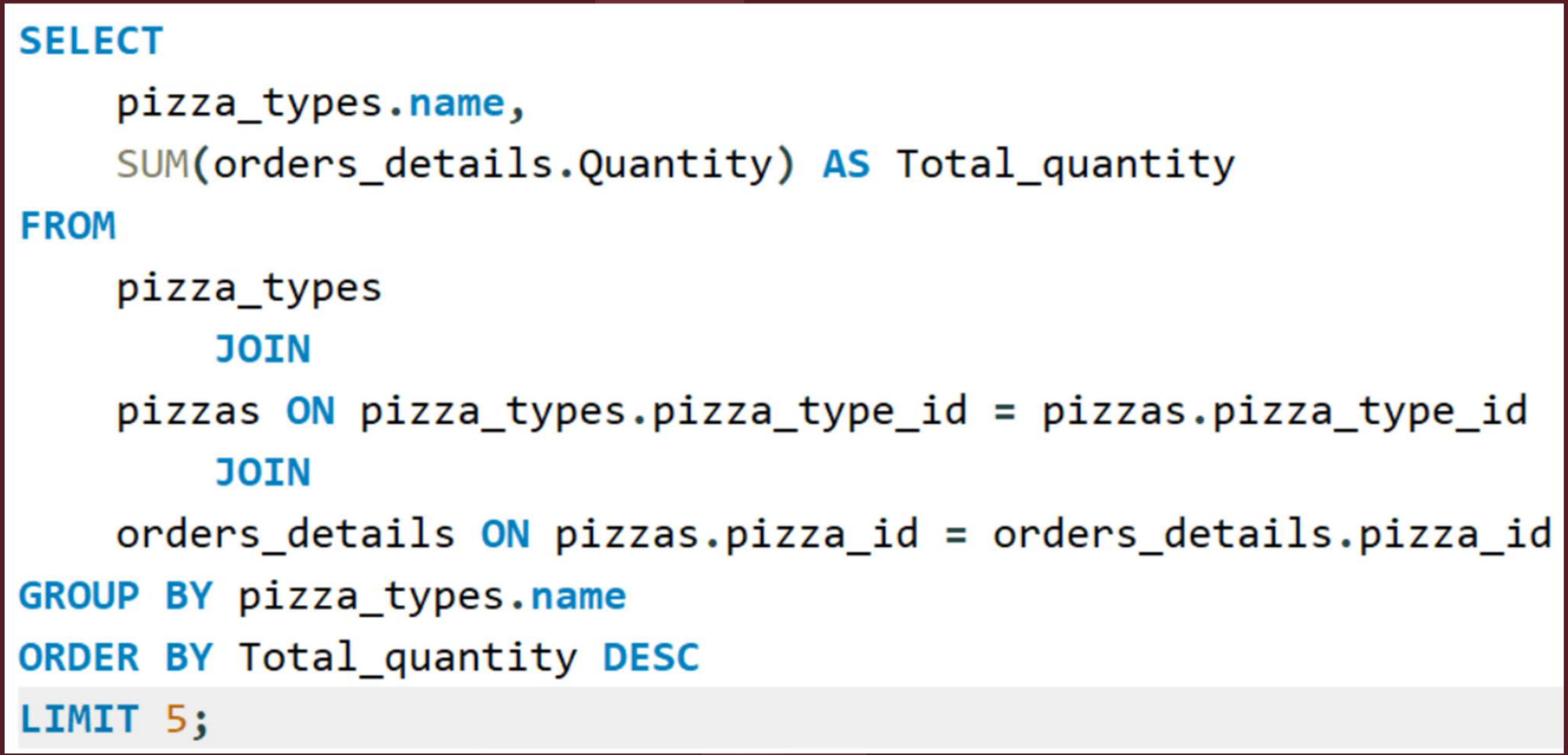
```
SELECT
    pizza_types.name AS pizza_name, pizzas.price AS Pizza_price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```



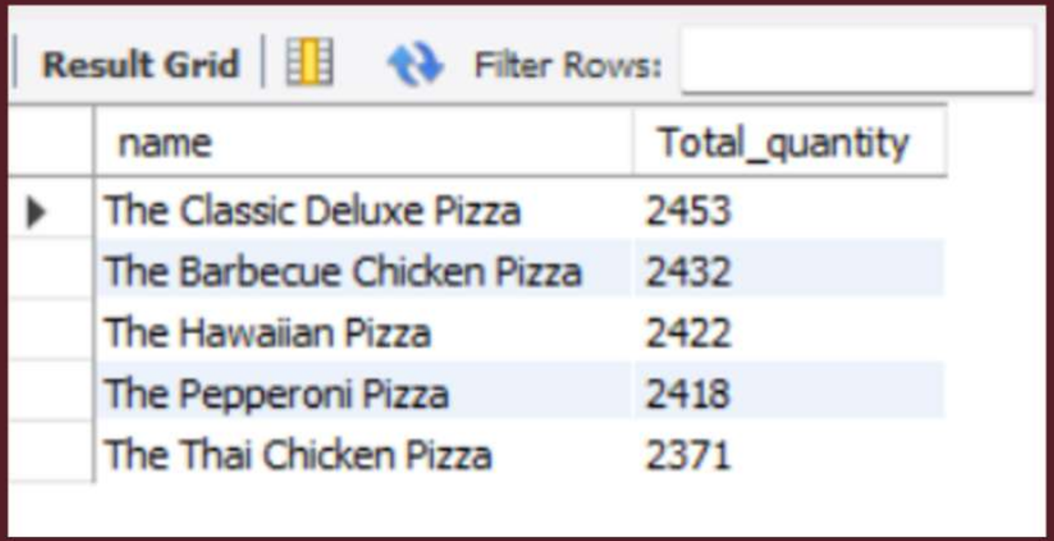
| pizza_name | Pizza_price |
|---|---|
| The Greek Pizza | 35.95 |

# TOP 5 MOST ORDERED PIZZA TYPES:

The top five most ordered pizzas, along with their quantities, were identified, showcasing the business's best-selling items and helping to assess product popularity.



```sql
SELECT
    pizza_types.name,
    SUM(orders_details.Quantity) AS Total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.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 |

# INTERMEDIATE ANALYSIS

05 Order Distribution by Hour

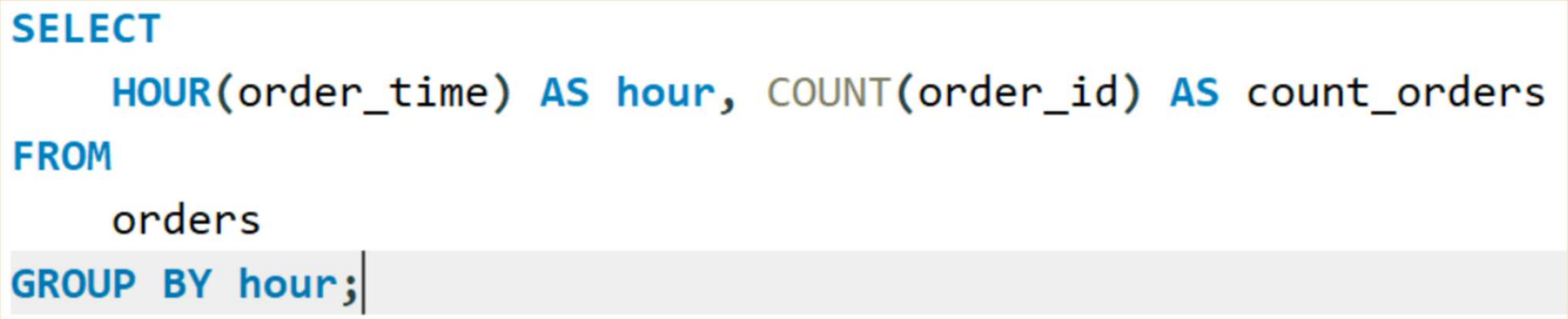
06 Category-Wise Pizza Distribution

07 Orders Grouped by Date

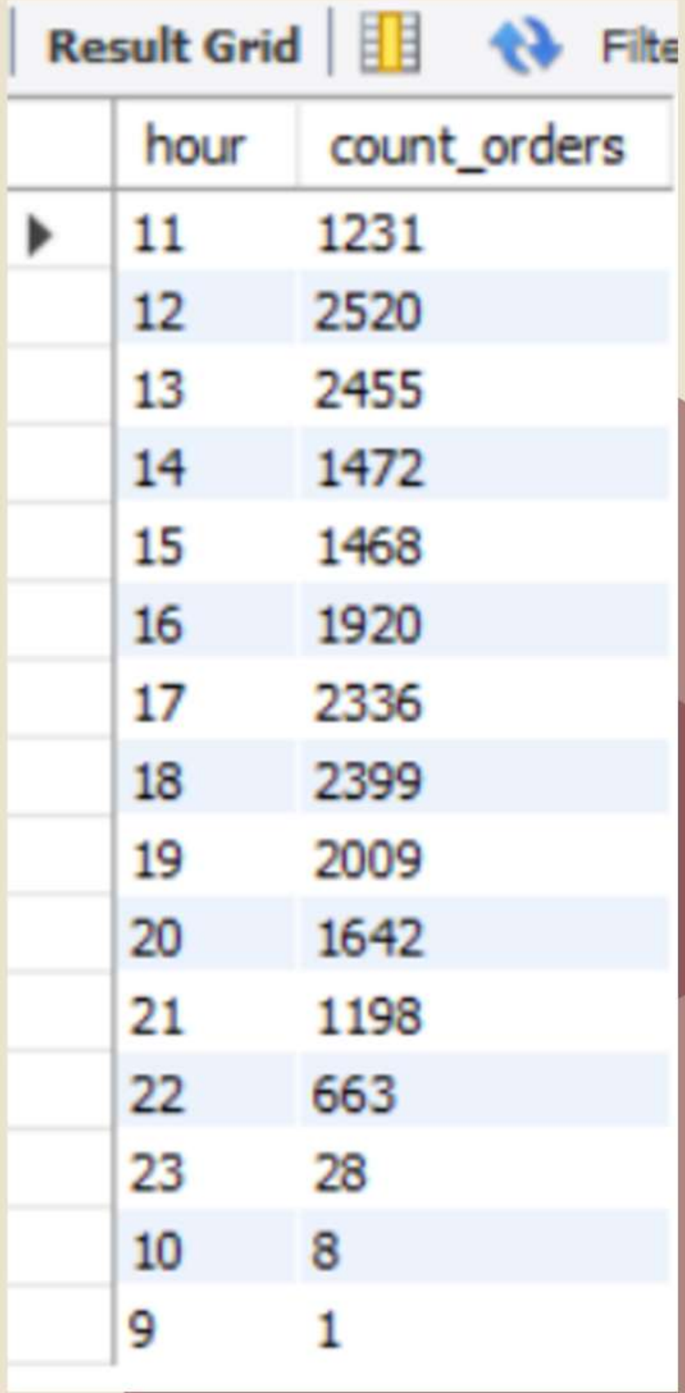
08 Top 3 Most Ordered Pizzas by Revenue

# ORDER DISTRIBUTION BY HOUR:

An analysis of orders placed by hour of the day revealed peak times for pizza purchases
This insight helps businesses optimize staffing and operations during busy periods



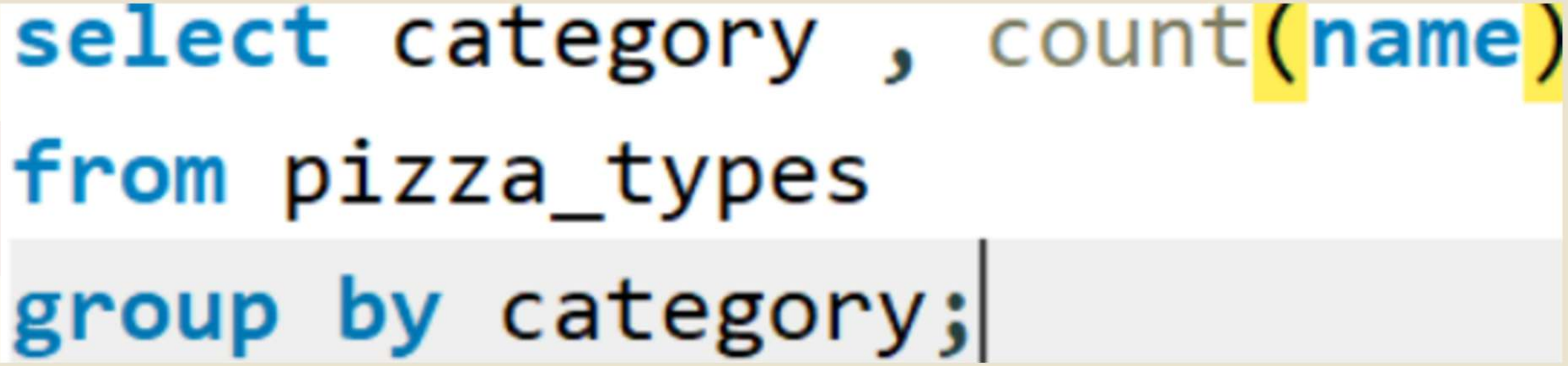
```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS count_orders
FROM
    orders
GROUP BY hour;
```



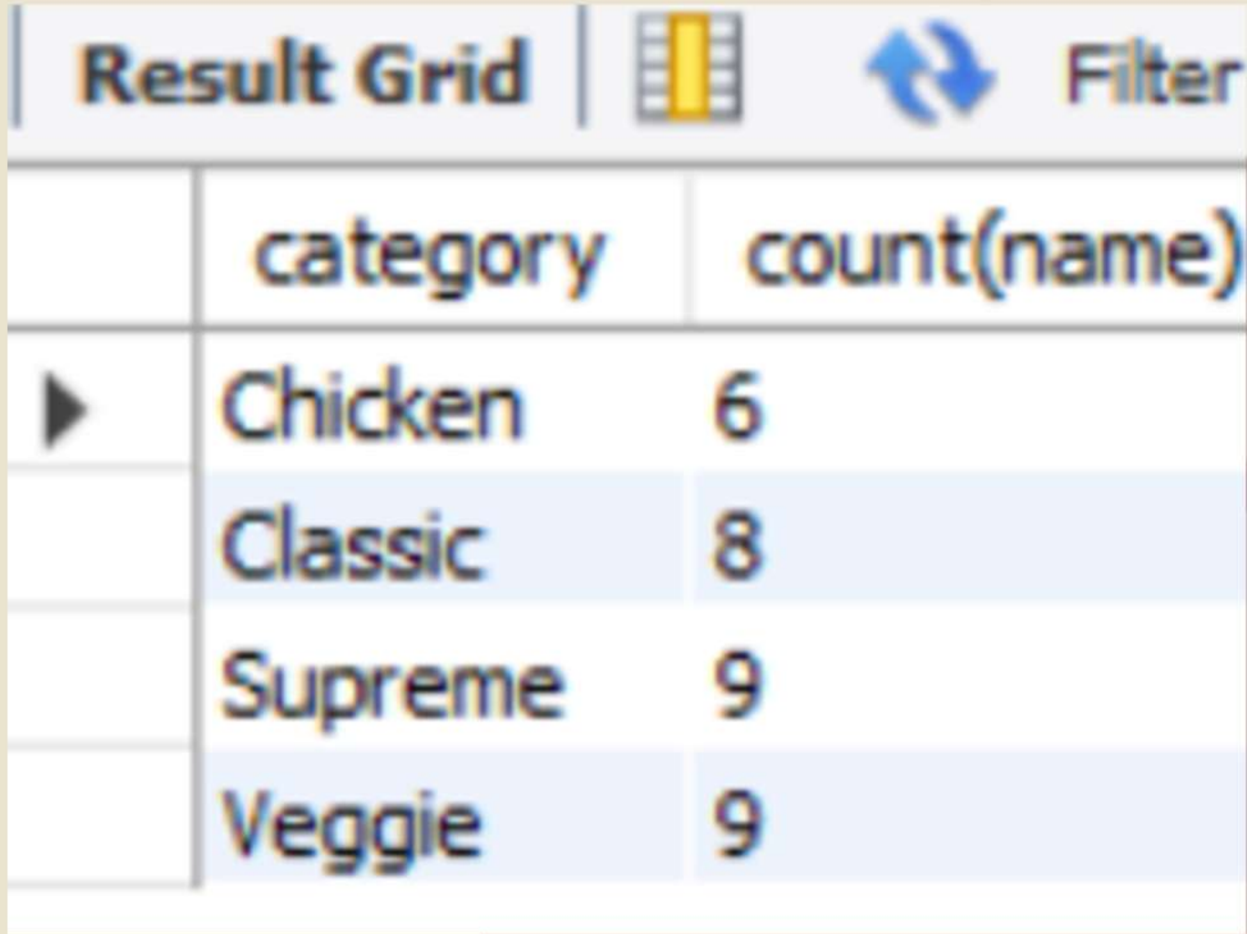
| hour | count_orders |
|---|---|
| 11 | 1231 |
| 12 | 2520 |
| 13 | 2455 |
| 14 | 1472 |
| 15 | 1468 |
| 16 | 1920 |
| 17 | 2336 |
| 18 | 2399 |
| 19 | 2009 |
| 20 | 1642 |
| 21 | 1198 |
| 22 | 663 |
| 23 | 28 |
| 10 | 8 |
| 9 | 1 |

# CATEGORY-WISE PIZZA DISTRIBUTION:

By joining relevant tables, I derived the distribution of pizza sales across different categories (e.g., Veg, Non-Veg), giving insights into customer preferences within product segments.



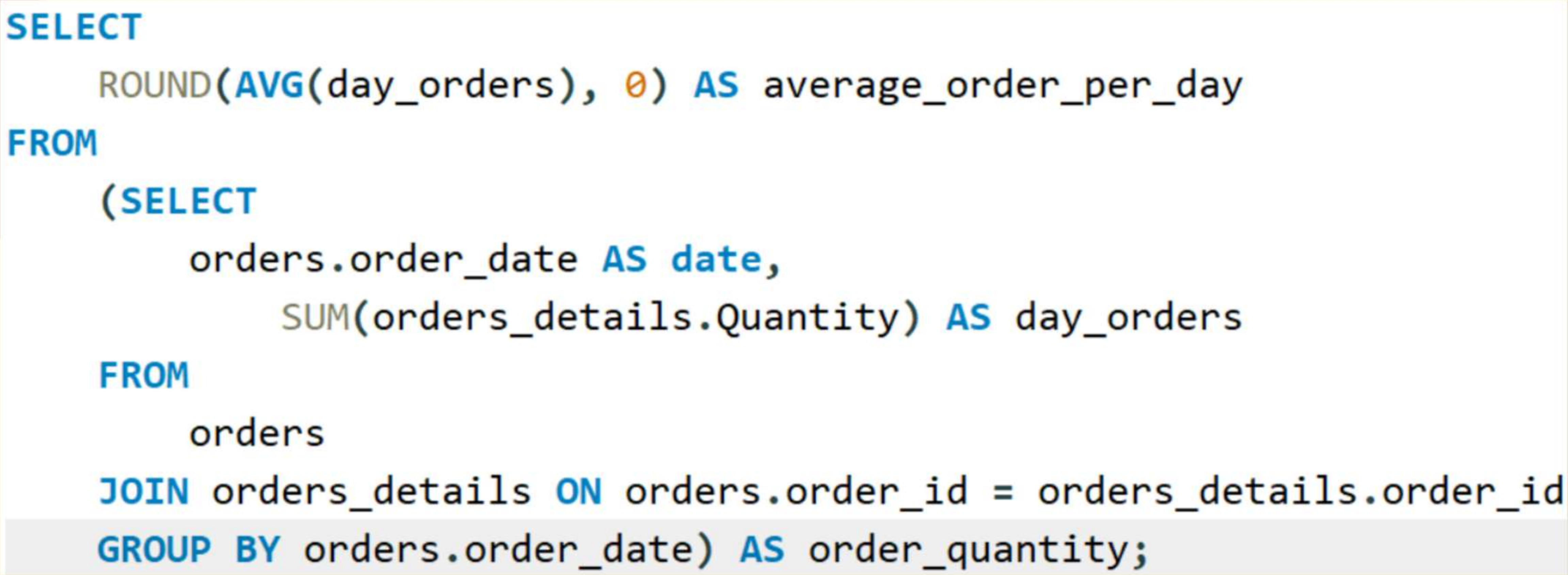
```sql
select category , count(name)
from pizza_types
group by category;
```



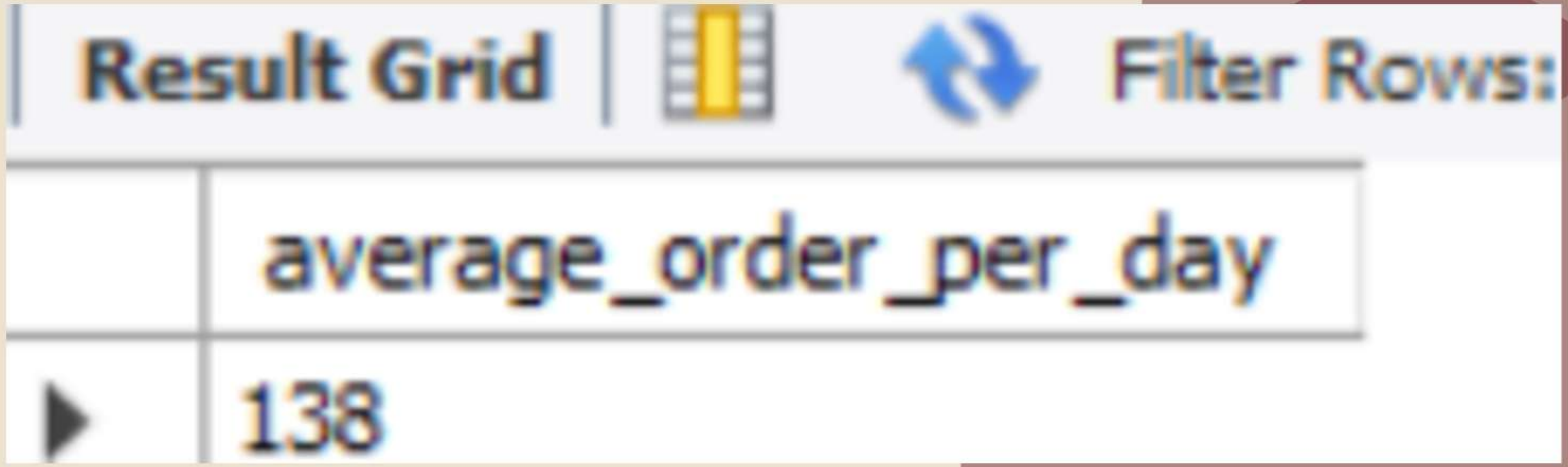
| category | count(name) |
|---|---|
| Chicken | 6 |
| Classic | 8 |
| Supreme | 9 |
| Veggie | 9 |

# ORDERS GROUPED BY DATE:

Grouping the data by date allowed me to calculate the average number of pizzas ordered per day. This analysis provided insight into daily sales patterns and variability over time.



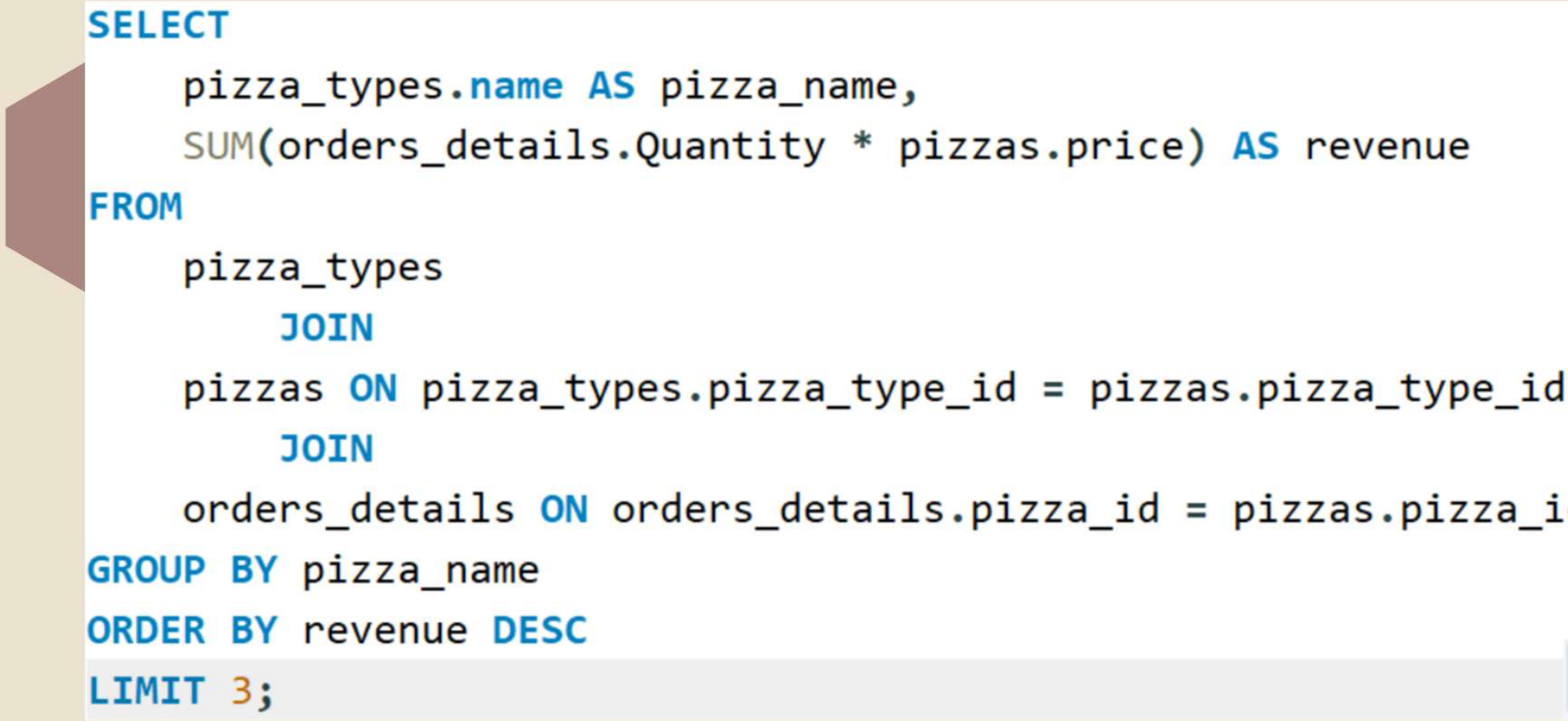
```sql
SELECT
    ROUND(AVG(day_orders), 0) AS average_order_per_day
FROM
    (SELECT
        orders.order_date AS date,
            SUM(orders_details.Quantity) AS day_orders
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```



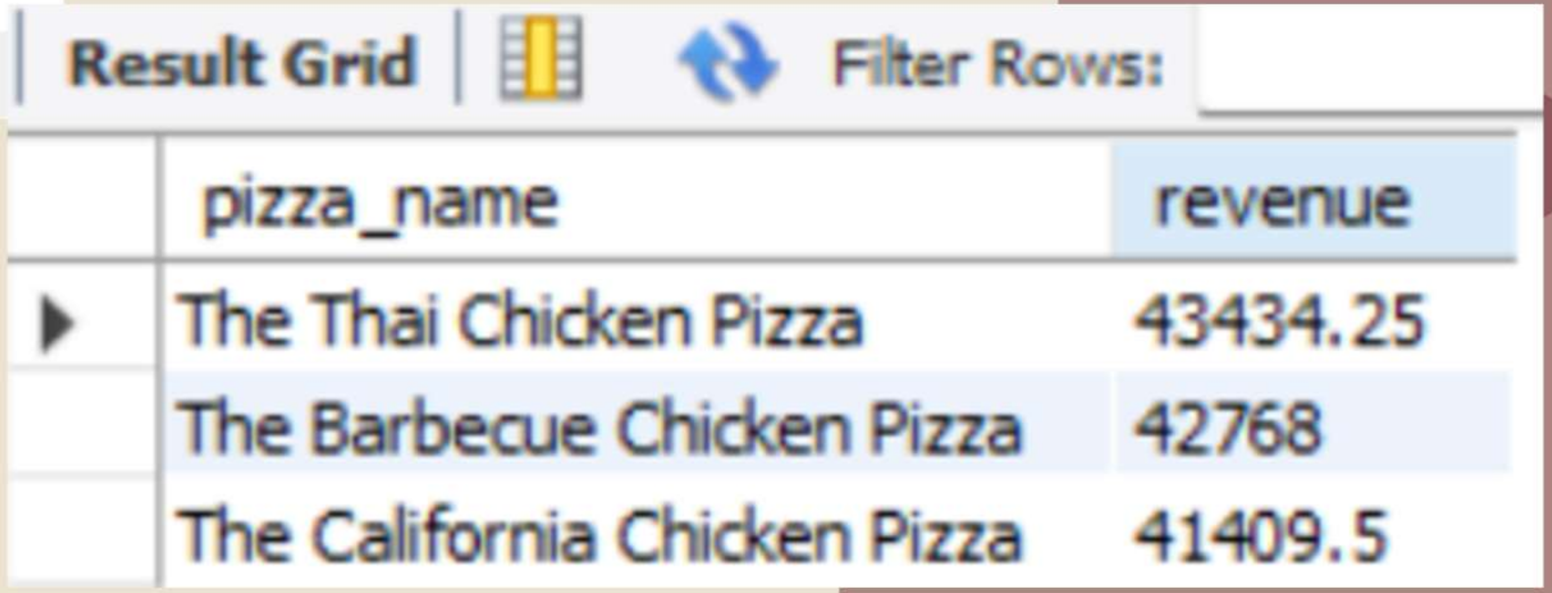
```
average_order_per_day
138
```

# TOP 3 MOST ORDERED PIZZAS BY REVENUE:

I identified the top three most ordered pizza types based on total revenue generated, highlighting which pizzas are not only popular but also profitable.



```sql
SELECT
    pizza_types.name AS pizza_name,
    SUM(orders_details.Quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_name
ORDER BY revenue DESC
LIMIT 3;
```



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

# ADVANCED ANALYSIS

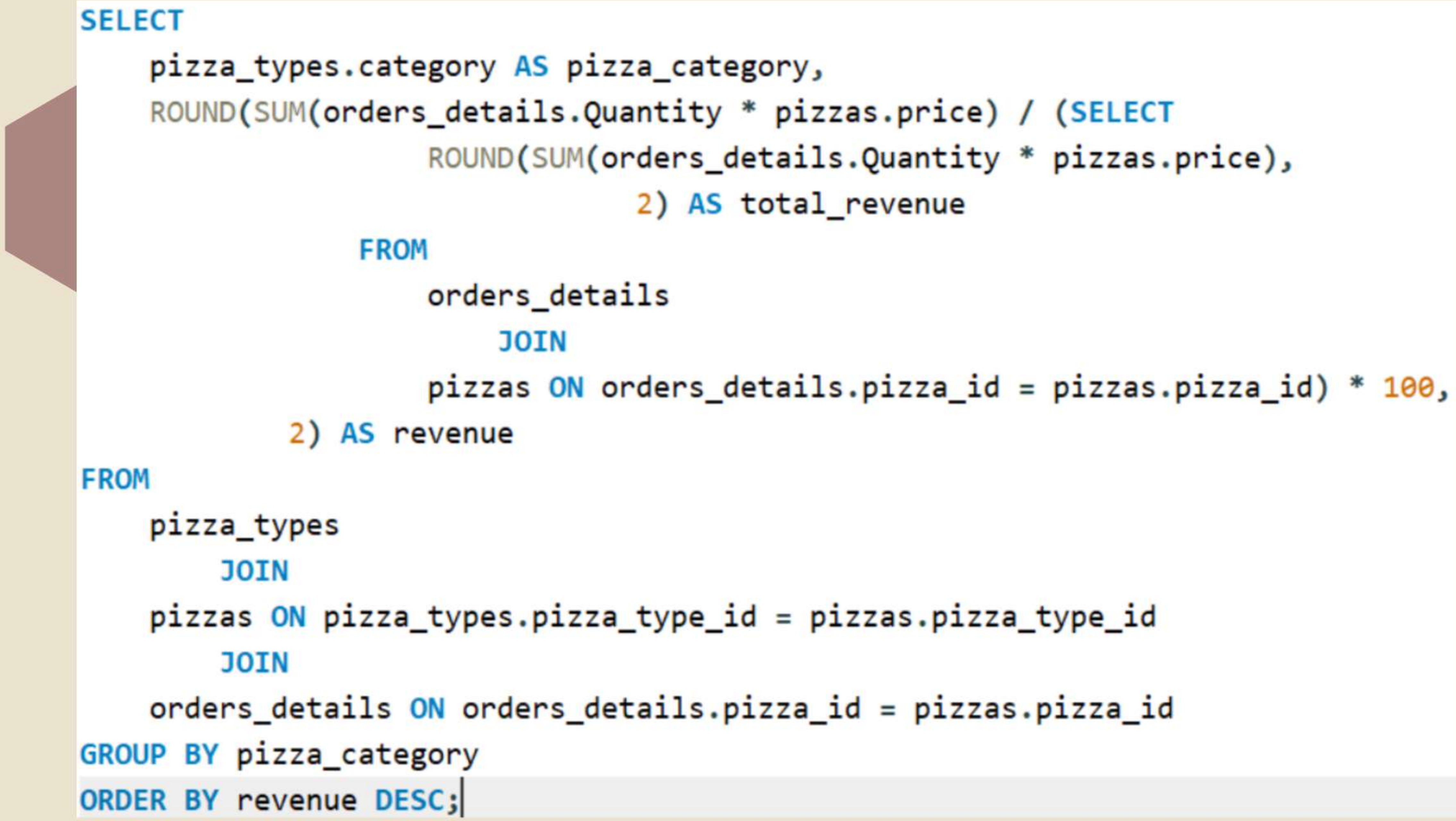
09 Percentage Contribution to Revenue

10 Cumulative Revenue Analysis

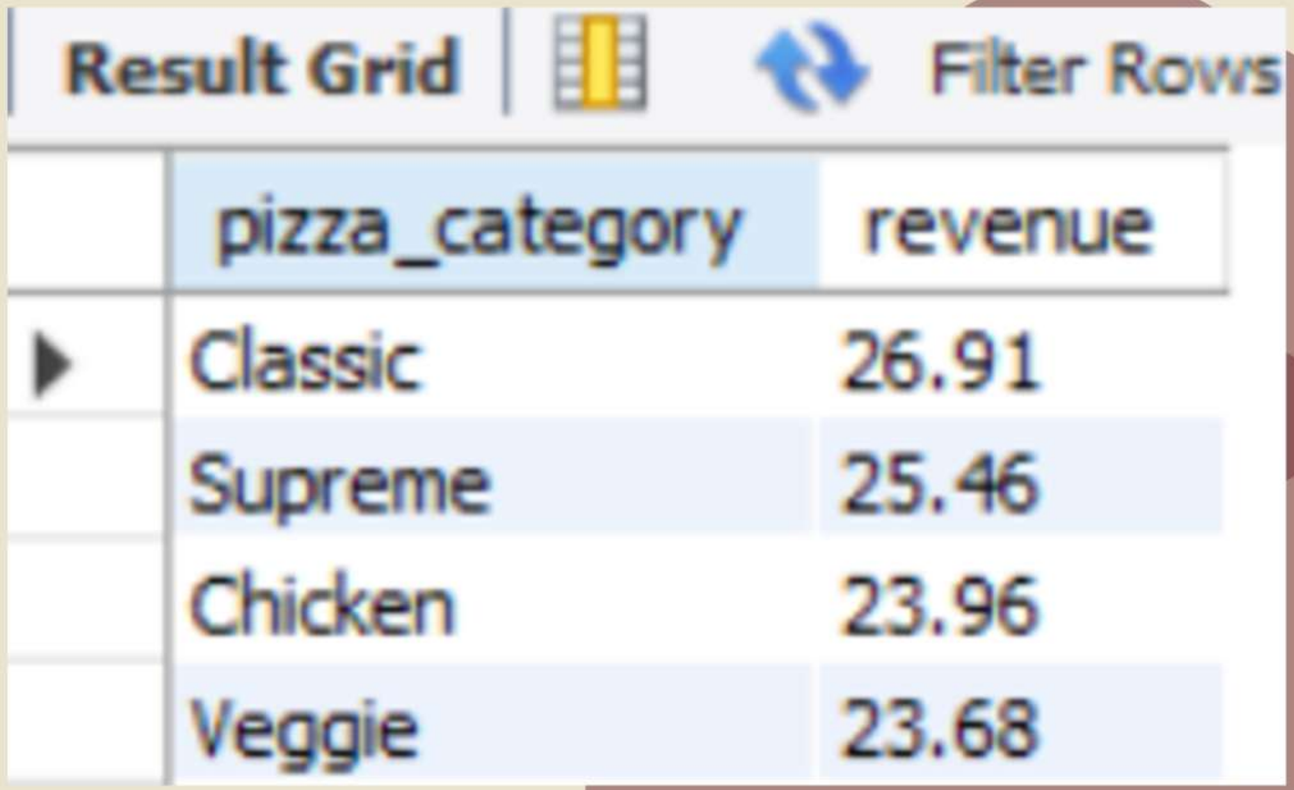
11 Top 3 Pizzas by Revenue for Each Category

# PERCENTAGE CONTRIBUTION TO REVENUE:

I calculated the percentage contribution of each pizza type to the total revenue, allowing for a better understanding of how each product influences overall sales.



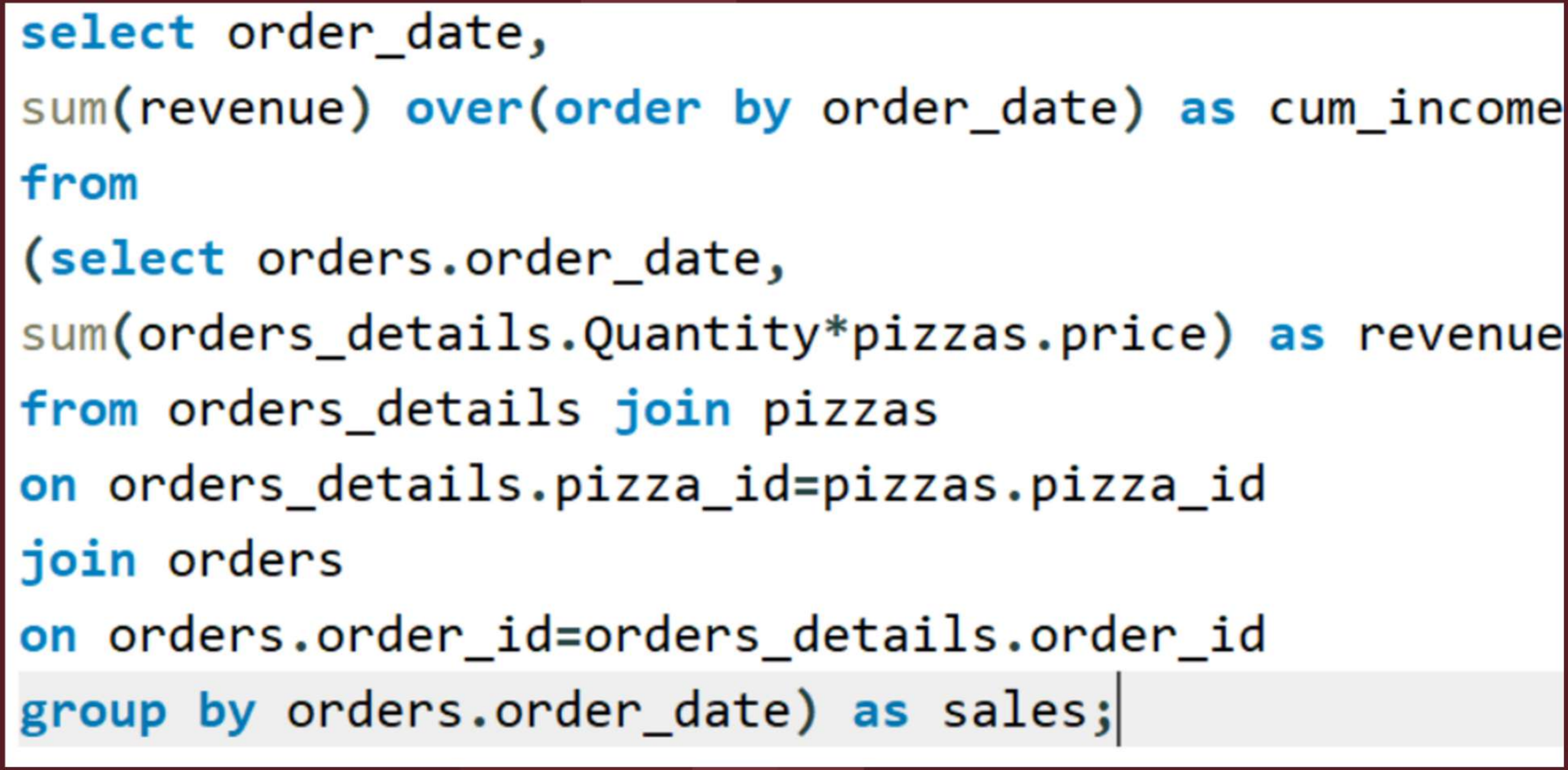
```sql
SELECT
    pizza_types.category AS pizza_category,
    ROUND(SUM(orders_details.Quantity * pizzas.price) / (SELECT
                    ROUND(SUM(orders_details.Quantity * pizzas.price),
                                2) AS total_revenue
                FROM
                    orders_details
                        JOIN
                    pizzas ON orders_details.pizza_id = pizzas.pizza_id) * 100,
            2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_category
ORDER BY revenue DESC;
```



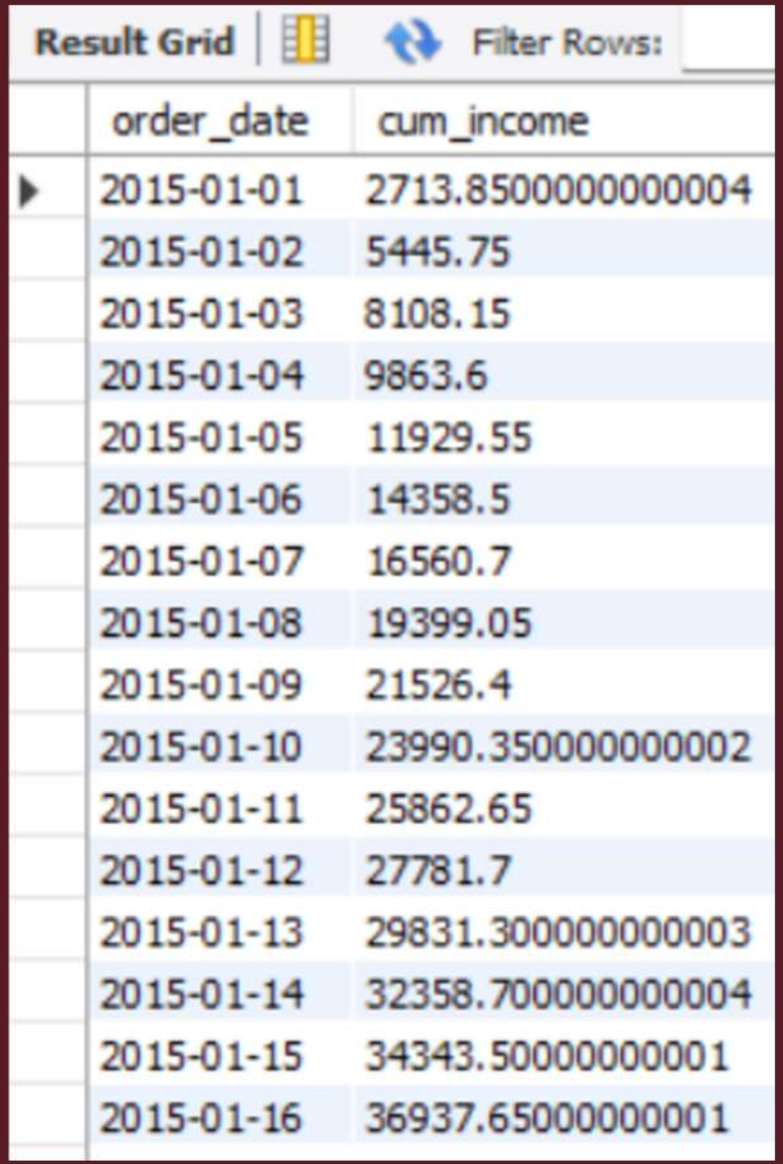
| pizza_category | revenue |
|---|---|
| Classic | 26.91 |
| Supreme | 25.46 |
| Chicken | 23.96 |
| Veggie | 23.68 |

# CUMULATIVE REVENUE ANALYSIS:

By analyzing cumulative revenue over time, I gained insights into the growth of sales throughout the observed period, helping identify trends or anomalies in performance.



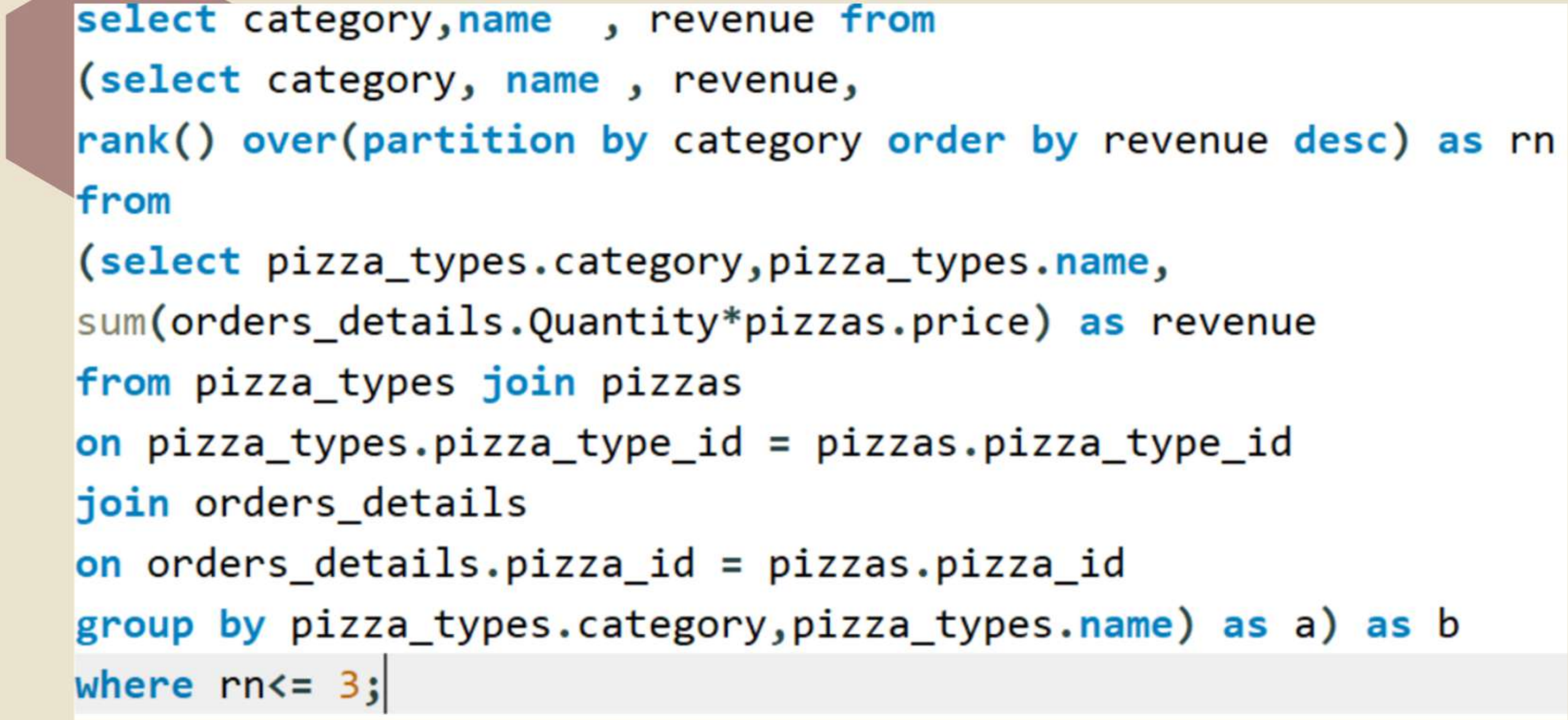
```
select order_date,
sum(revenue) over(order by order_date) as cum_income
from
(select orders.order_date,
sum(orders_details.Quantity*pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id=pizzas.pizza_id
join orders
on orders.order_id=orders_details.order_id
group by orders.order_date) as sales;
```



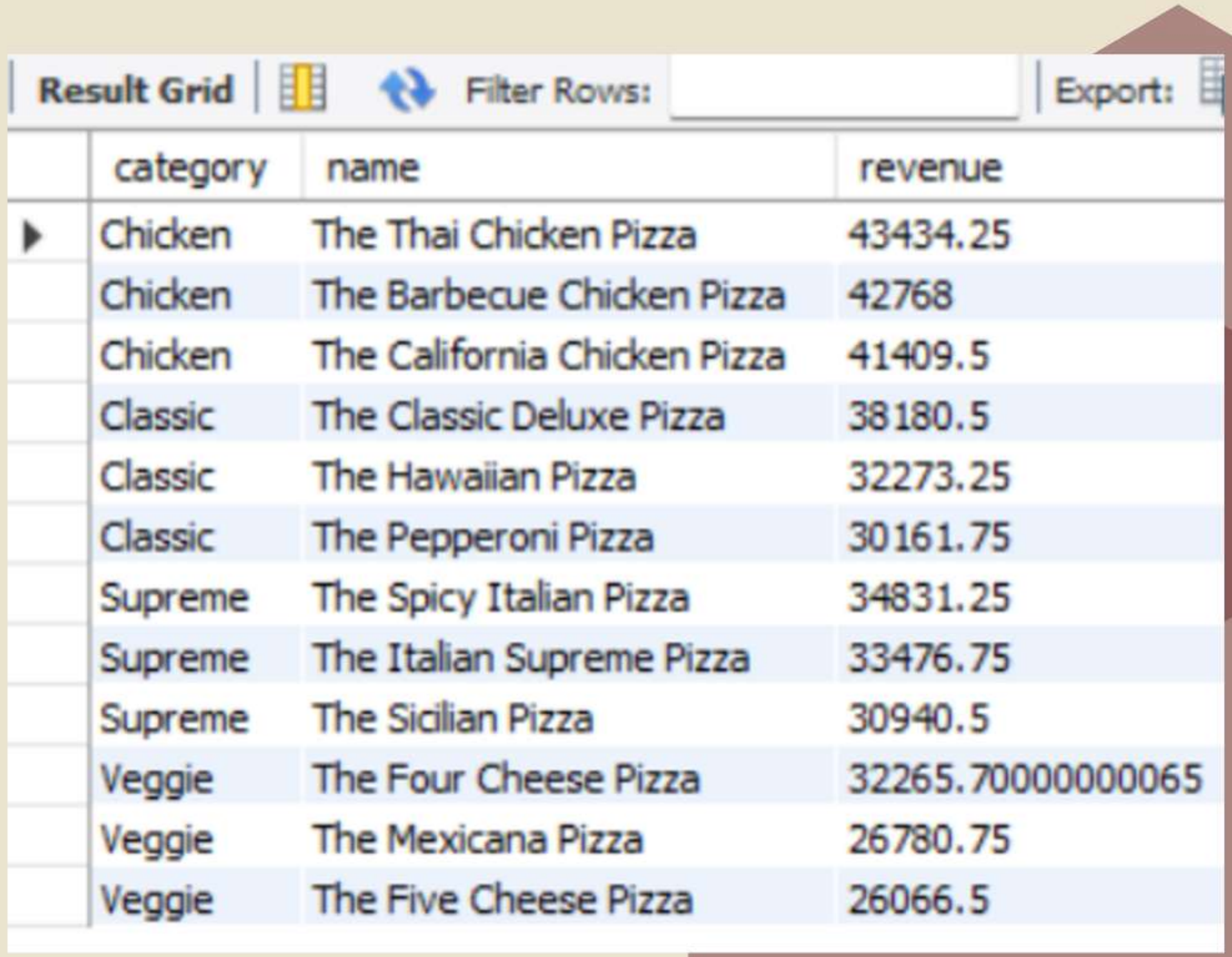
| order_date | cum_income |
|---|---|
| 2015-01-01 | 2713.8500000000004 |
| 2015-01-02 | 5445.75 |
| 2015-01-03 | 8108.15 |
| 2015-01-04 | 9863.6 |
| 2015-01-05 | 11929.55 |
| 2015-01-06 | 14358.5 |
| 2015-01-07 | 16560.7 |
| 2015-01-08 | 19399.05 |
| 2015-01-09 | 21526.4 |
| 2015-01-10 | 23990.350000000002 |
| 2015-01-11 | 25862.65 |
| 2015-01-12 | 27781.7 |
| 2015-01-13 | 29831.300000000003 |
| 2015-01-14 | 32358.700000000004 |
| 2015-01-15 | 34343.50000000001 |
| 2015-01-16 | 36937.65000000001 |

# TOP 3 PIZZAS BY REVENUE FOR EACH CATEGORY:

The analysis went further to determine the top three most ordered pizza types based on revenue within each category (e.g., Veg, Non-Veg). This deeper insight helps understand which pizzas are the most successful within their specific segments.



```sql
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(orders_details.Quantity*pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.category,pizza_types.name) as a) as b
where rn<= 3;
```



| 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 |

# CONCLUSION:

This project highlights the importance of SQL in extracting actionable insights from raw data. Through thorough analysis of pizza sales data, I was able to uncover key performance metrics, customer preferences, and sales trends. These insights can help a pizza business optimize its offerings, manage operations more effectively, and increase profitability.