Pizza Sales Analysis

using SQL and Excel

~Owaise

Introduction

This project demonstrates the application of SQL concepts to analyze a pizza sales dataset. It was undertaken as a practice exercise to reinforce my SQL skills, following a guided tutorial from WS Cube Tech.

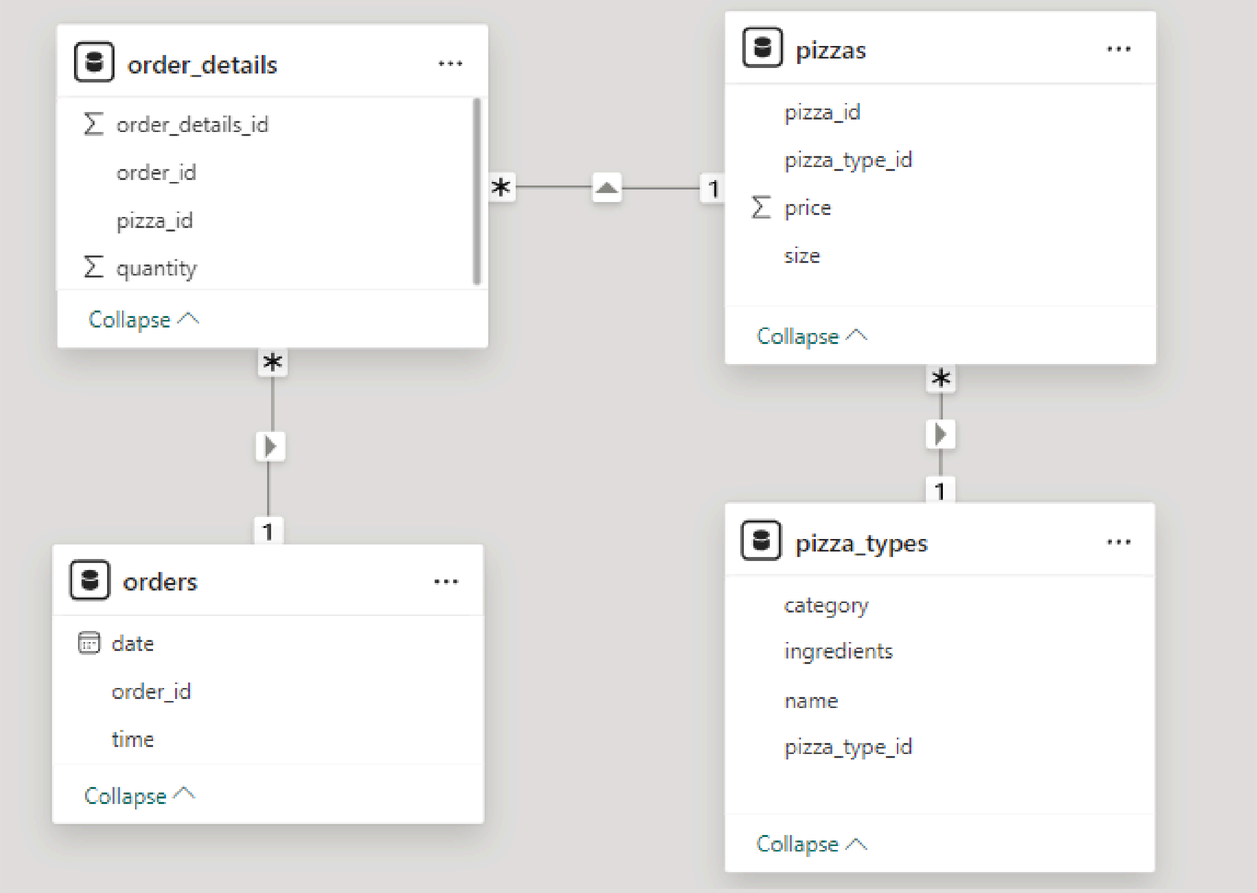
The dataset includes four interrelated CSV files:

1. Orders
2. Order Details
3. Pizzas
4. Pizza Types

Through this project, I solved 13 queries covering sales insights, revenue generation, and product performance. These queries helped uncover valuable trends, such as the most ordered pizzas, revenue distribution, and cumulative growth over time.

This project showcases my ability to write complex SQL queries, understand relational database concepts, and derive actionable insights from structured data. The visuals are made using Excel are added to interpret the results of the queries.

Dataset Overview



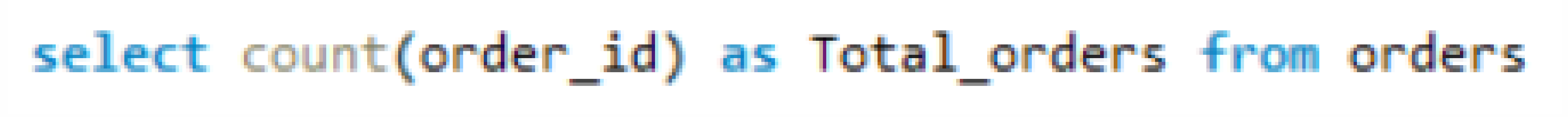
The dataset contains four CSV files representing relational data for pizza sales:

- **Orders :** Contains order IDs, order dates and Time.
- **Orders Details :** Links orders to pizzas with pizza_id and order_id.
- **Pizzas :** Lists pizzas with their sizes and base prices. Links Pizza Types to Order details with pizza_type_id and pizza_id.
- **Pizza Types :** Provides metadata about pizzas, including categories and ingredients.

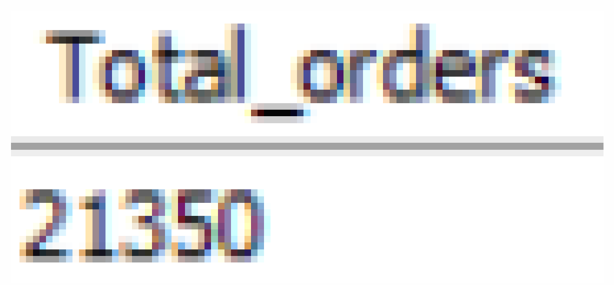
Queries

1. Total Number of Orders Placed
2. Total Revenue Generated from Pizza Sales.
3. Most Expensive Pizza
4. Pizza Ordered Quantity based on Size
5. Most Ordered Top 5 Pizzas
6. Total Pizza Ordered Quantity by Category
7. Total Orders by Hour
8. Category-wise Distribution of Pizzas
9. Average Pizzas Ordered per Day
10. Top 3 Ordered Pizzas based on Revenue
11. Contribution of Revenue by Each Pizza Category in Percentage
12. Cumulative Revenue generated over Time
13. Category-wise Top 3 Revenue generated Pizzas

1. Total Number of Orders Placed

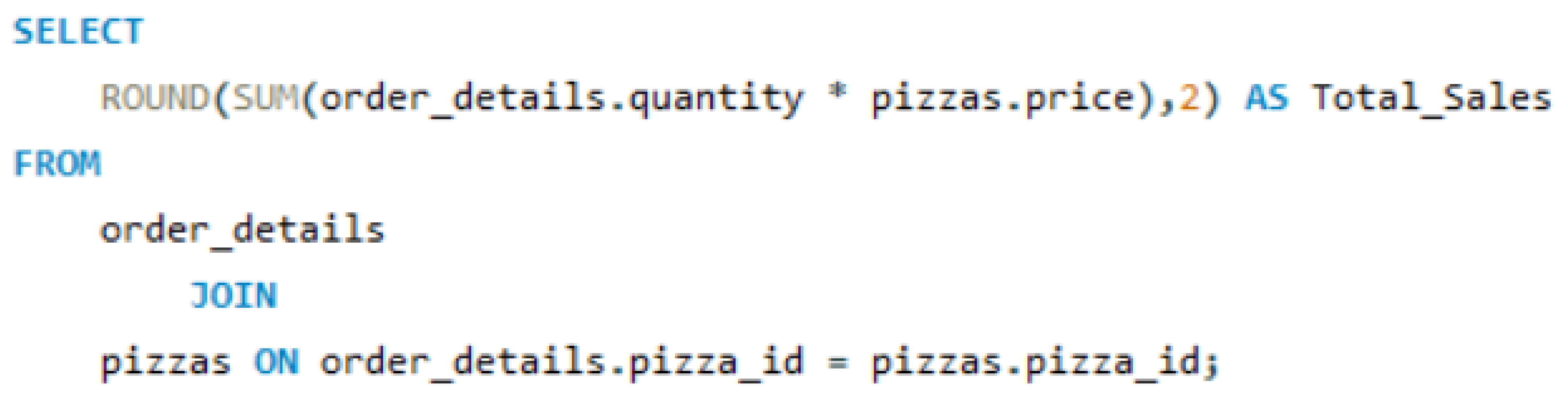


```sql
select count(order_id) as Total_orders from orders
```

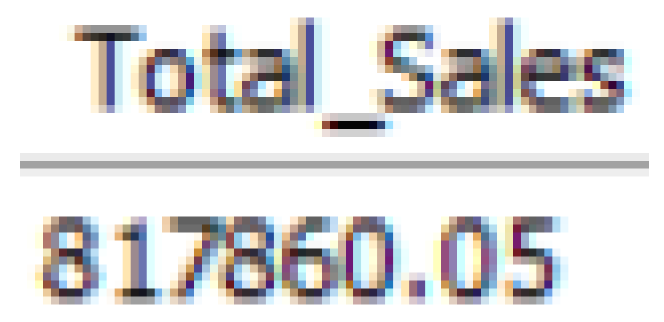


Total_orders
21350

2. Total Revenue Generated from Pizza Sales.

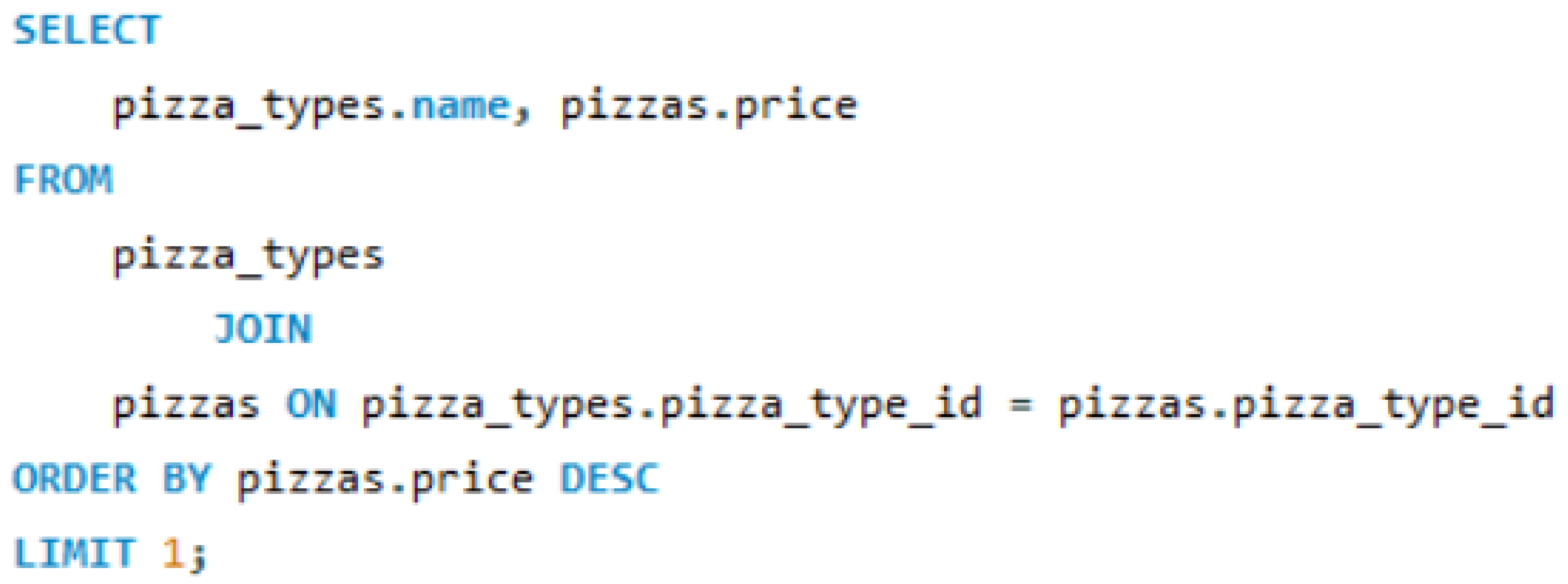


```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),2) AS Total_Sales
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

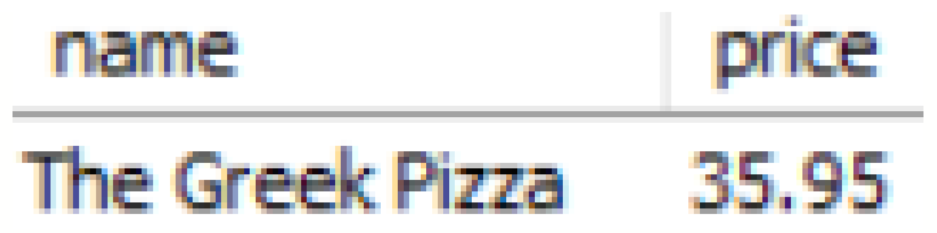


```
Total_Sales
817860.05
```

3. Most Expensive Pizza

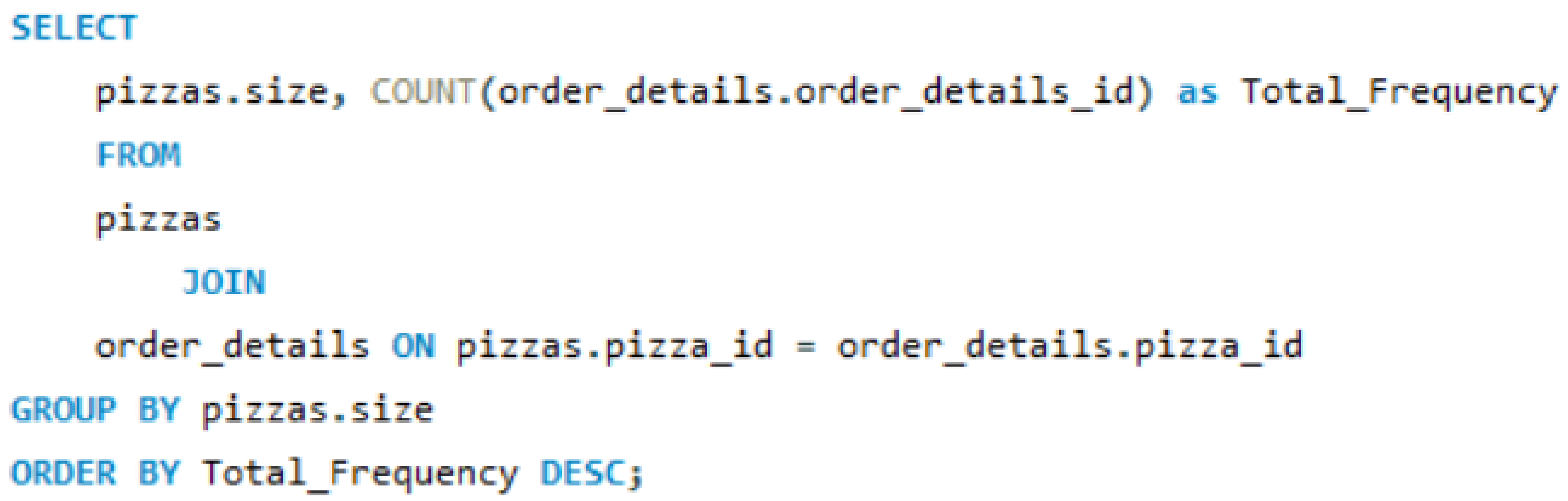


```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

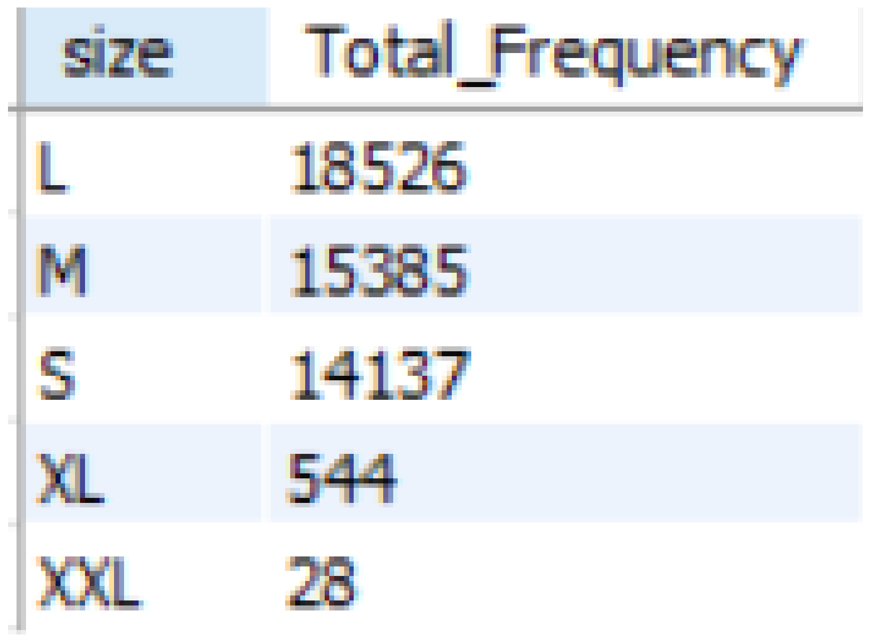


name	price
The Greek Pizza	35.95

4. Pizza Ordered Quantity based on Size



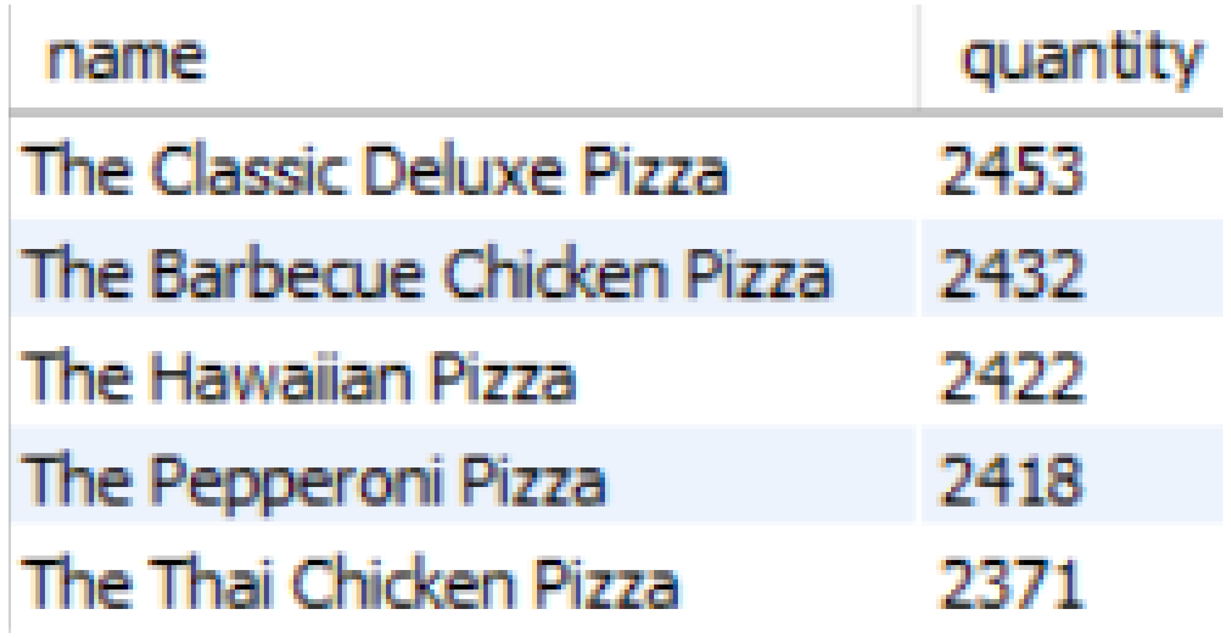
```
SELECT
    pizzas.size, COUNT(order_details.order_details_id) as Total_Frequency
    FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY Total_Frequency DESC;
```



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

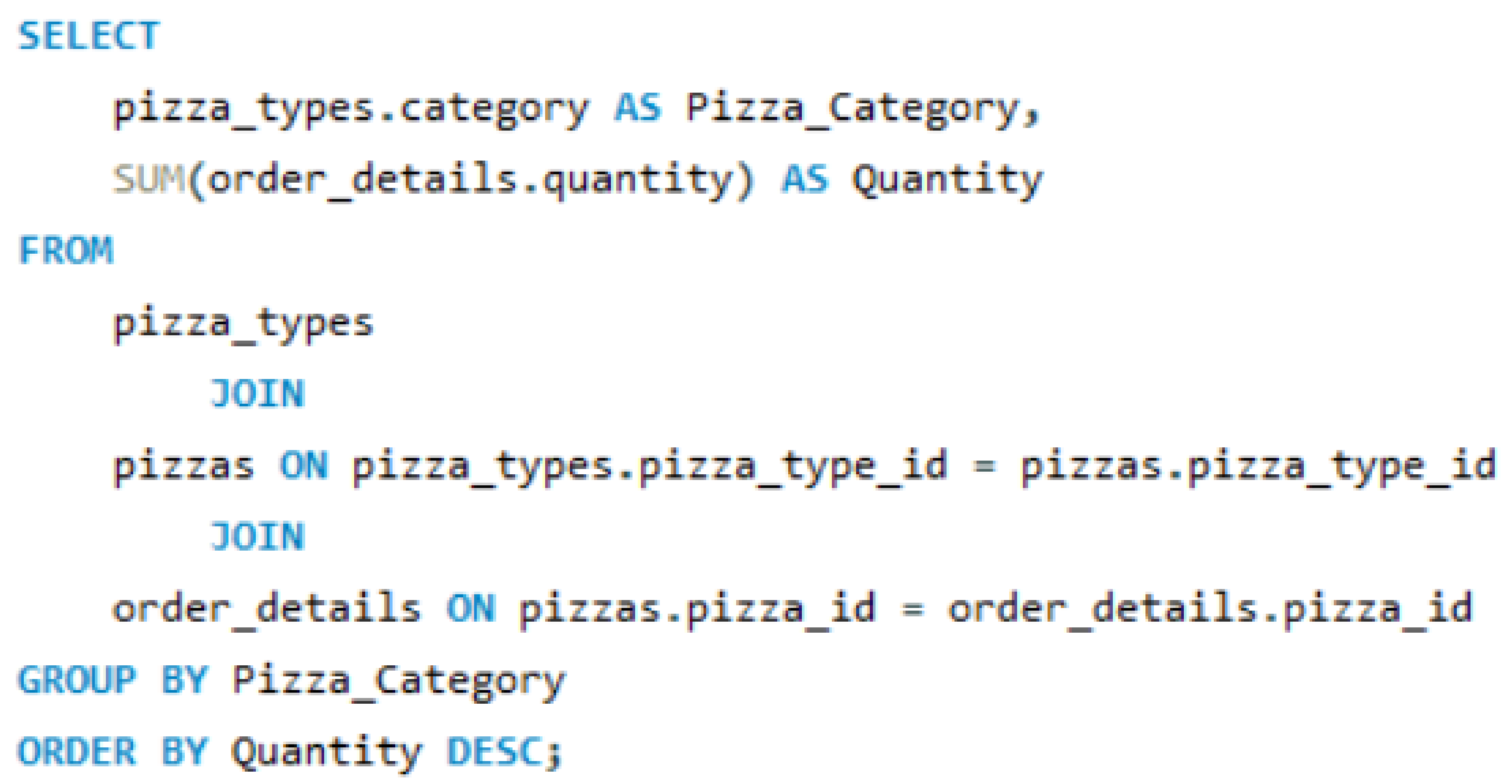
5. Most Ordered Top 5 Pizzas

```
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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

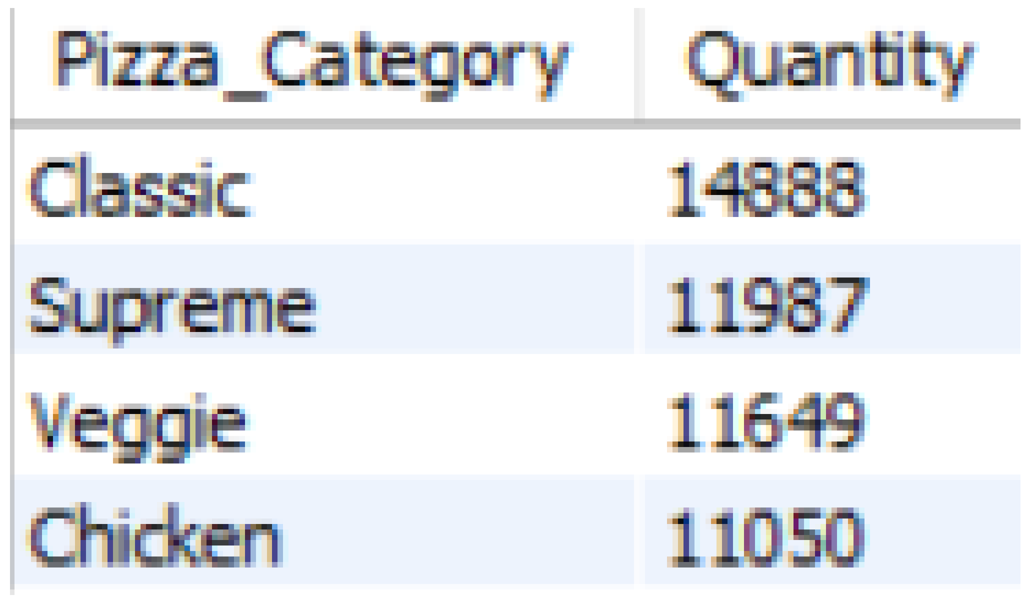


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

6. Total Pizza Ordered Quantity by Category



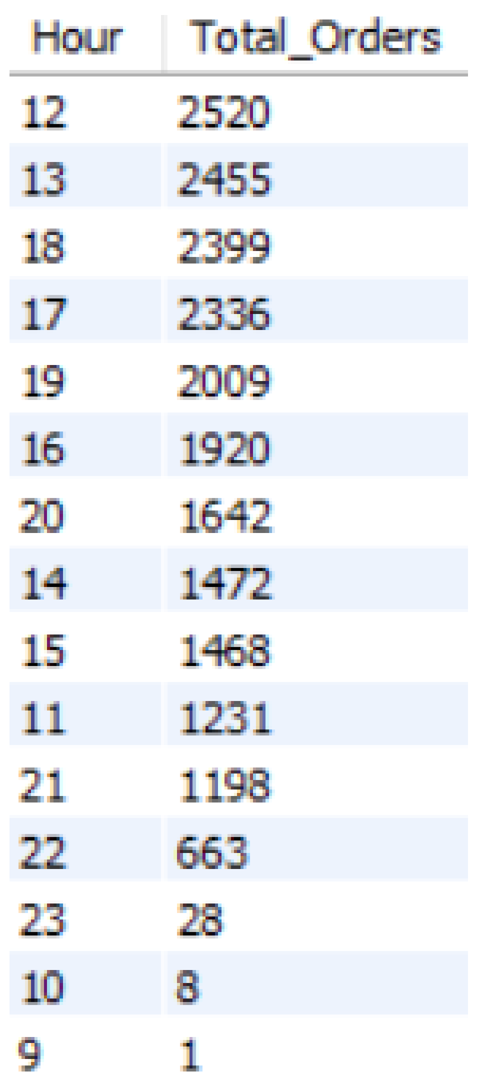
```
SELECT
    pizza_types.category AS Pizza_Category,
    SUM(order_details.quantity) AS Quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY Pizza_Category
ORDER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



Pizza_Category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

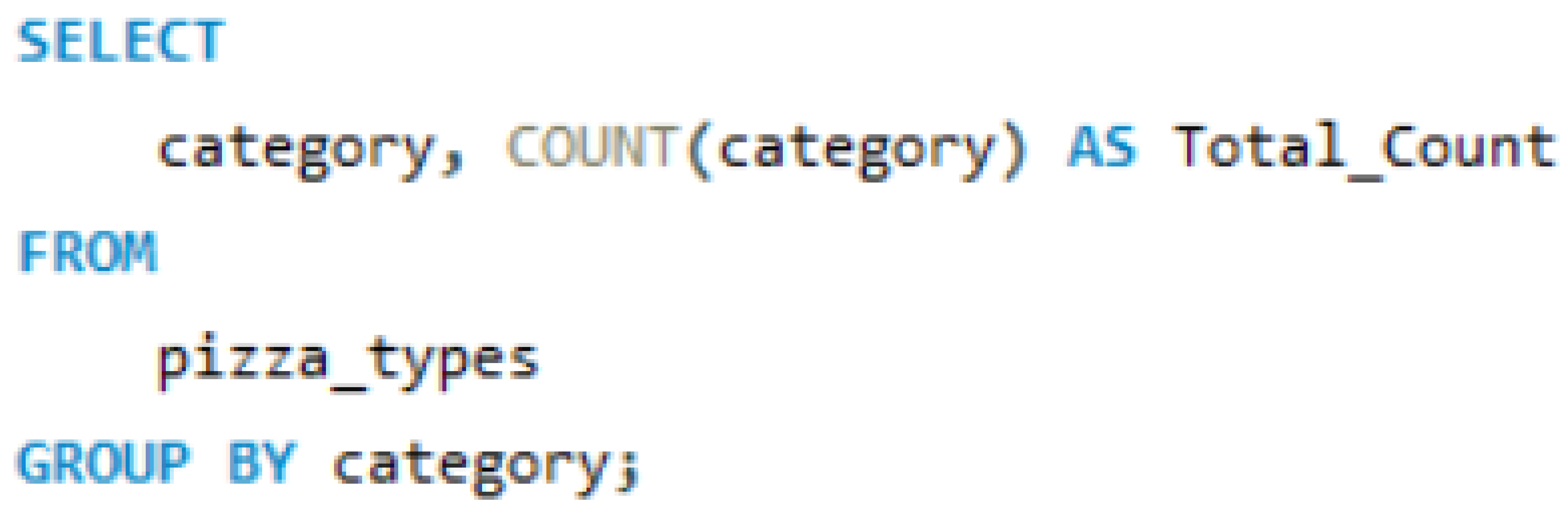
7. Total Orders by Hour

```
SELECT
    HOUR(order_time) AS Hour, COUNT(order_id) AS Total_Orders
FROM
    orders
GROUP BY Hour
ORDER BY Total_Orders DESC;
```

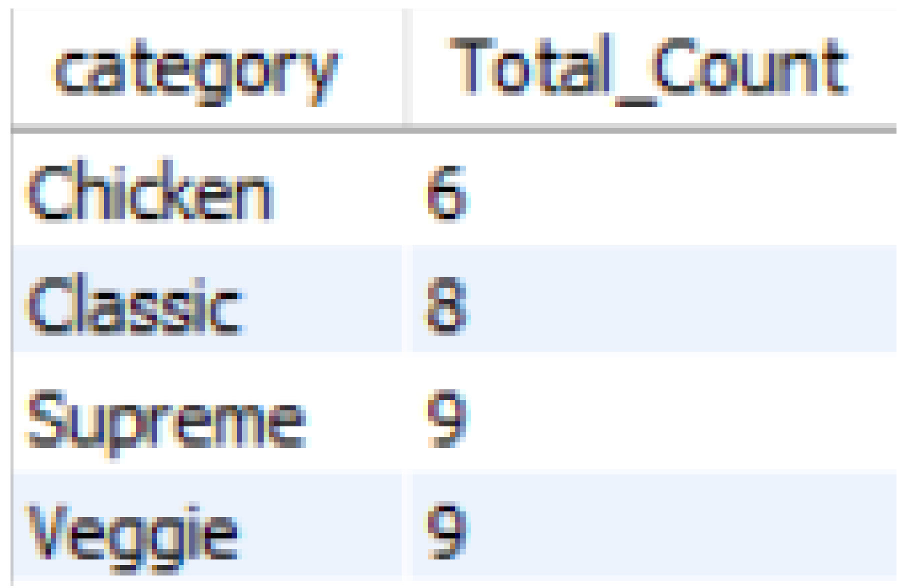


Hour	Total_Orders
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231
21	1198
22	663
23	28
10	8
9	1

8. Category-wise Distribution of Pizzas

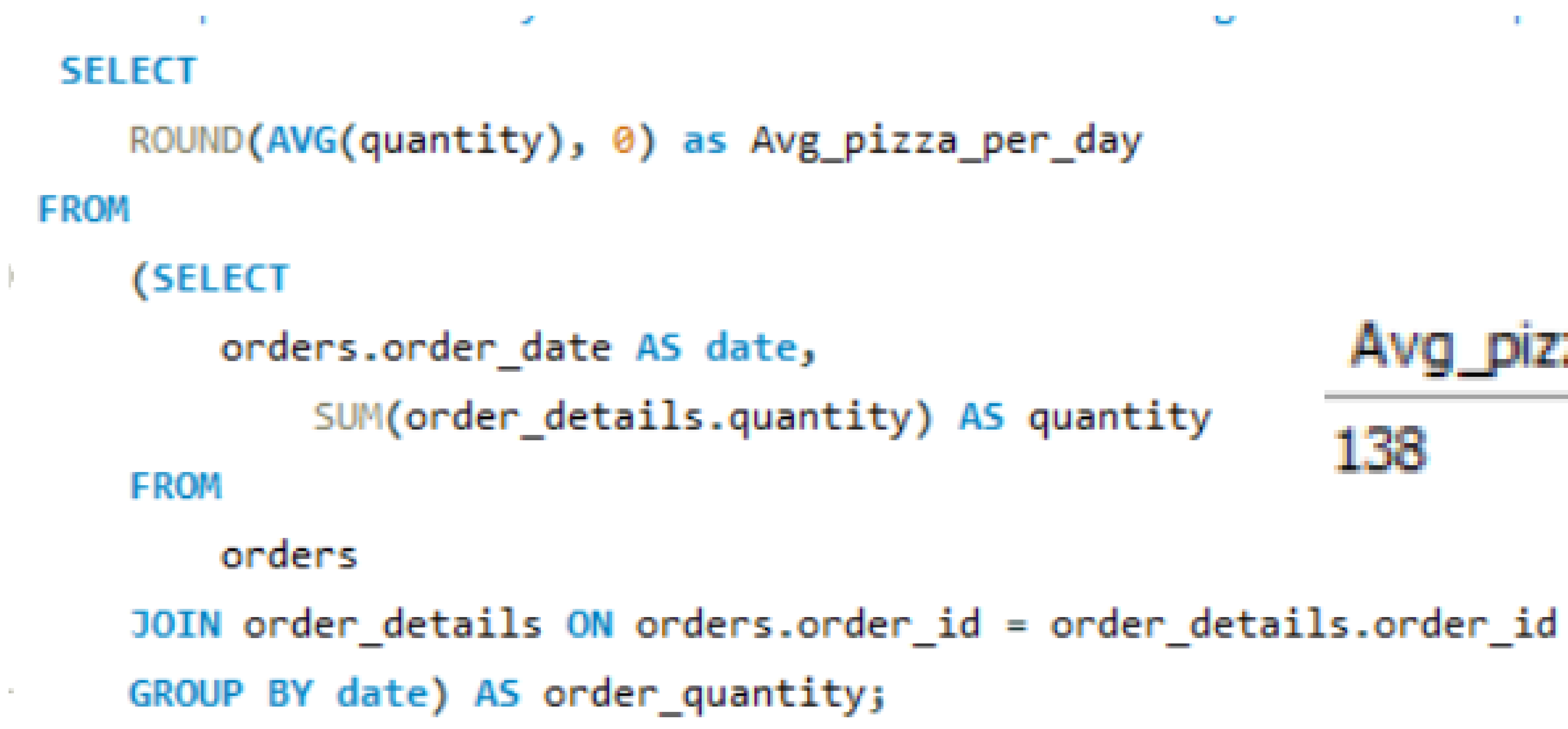


```
SELECT
    category, COUNT(category) AS Total_Count
FROM
    pizza_types
GROUP BY category;
```

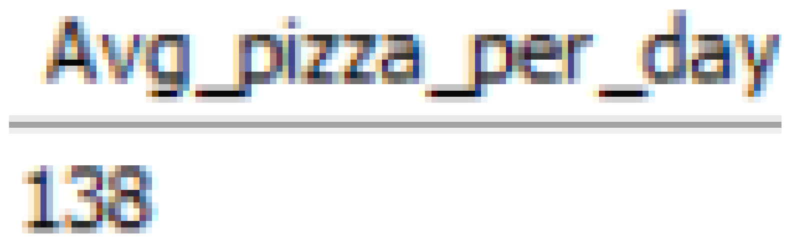


category	Total_Count
Chicken	6
Classic	8
Supreme	9
Veggie	9

9. Average Pizzas Ordered per Day

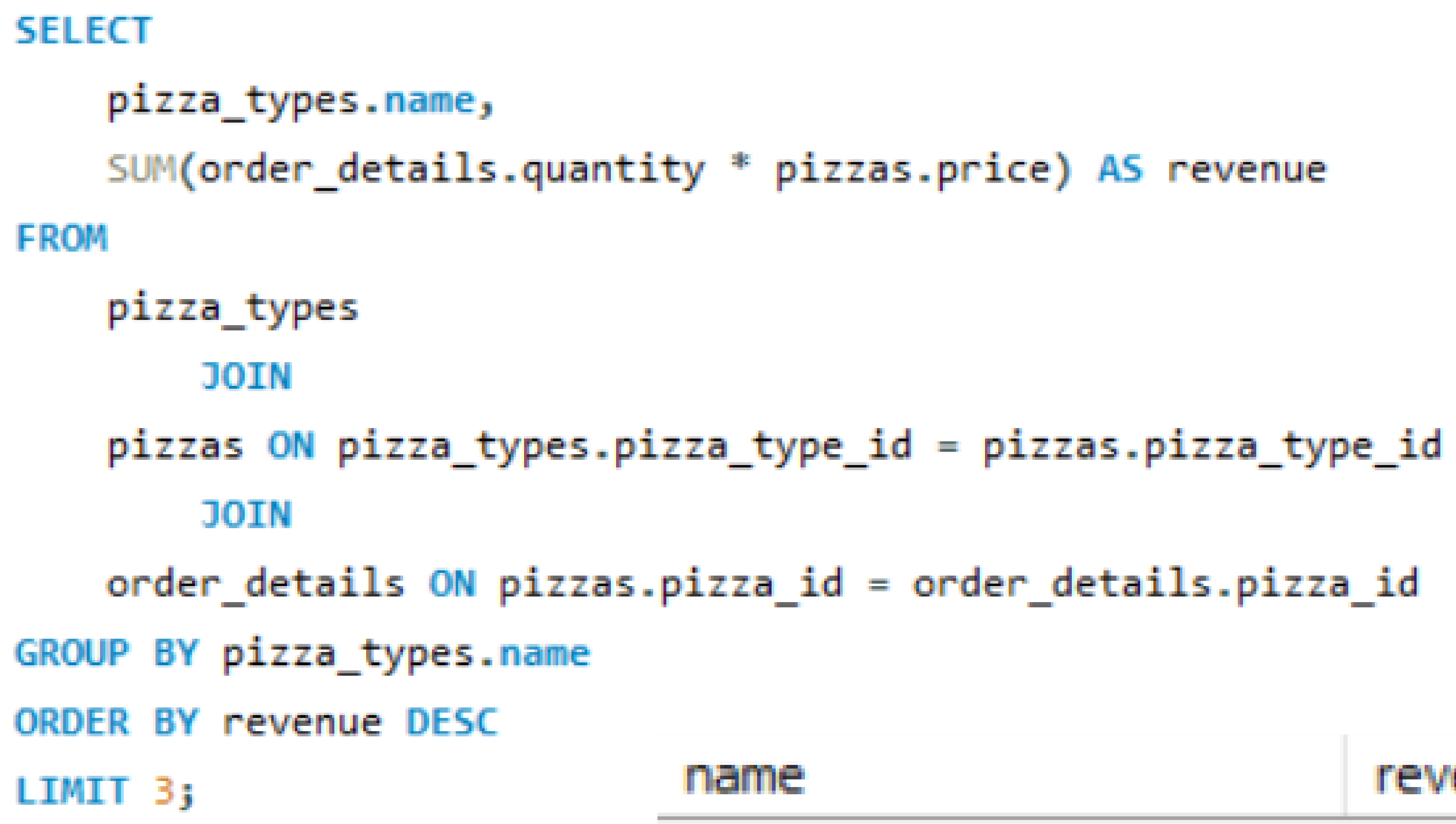


```sql
 SELECT
    ROUND(AVG(quantity), 0) as Avg_pizza_per_day
FROM
    (SELECT
        orders.order_date AS date,
            SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY date) AS order_quantity;
```

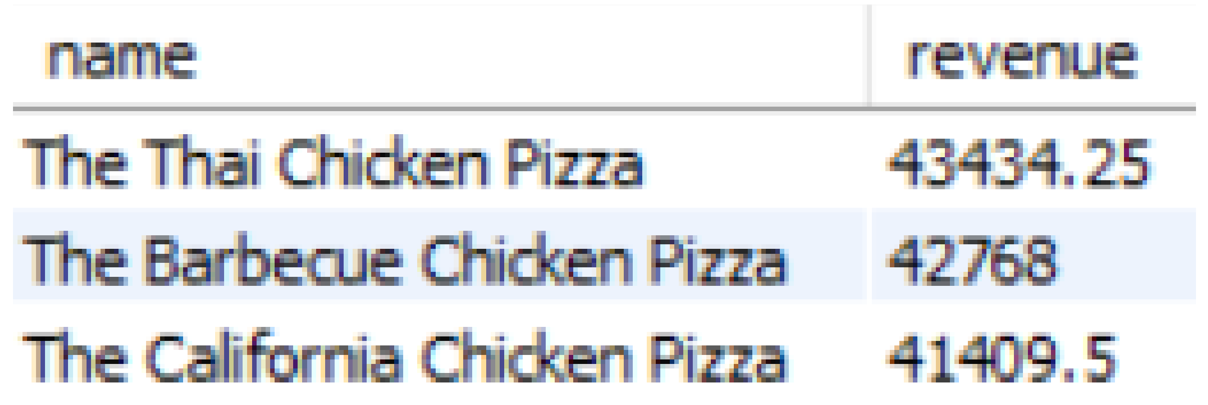


Avg_pizza_per_day
138

10. Top 3 Ordered Pizzas based on Revenue

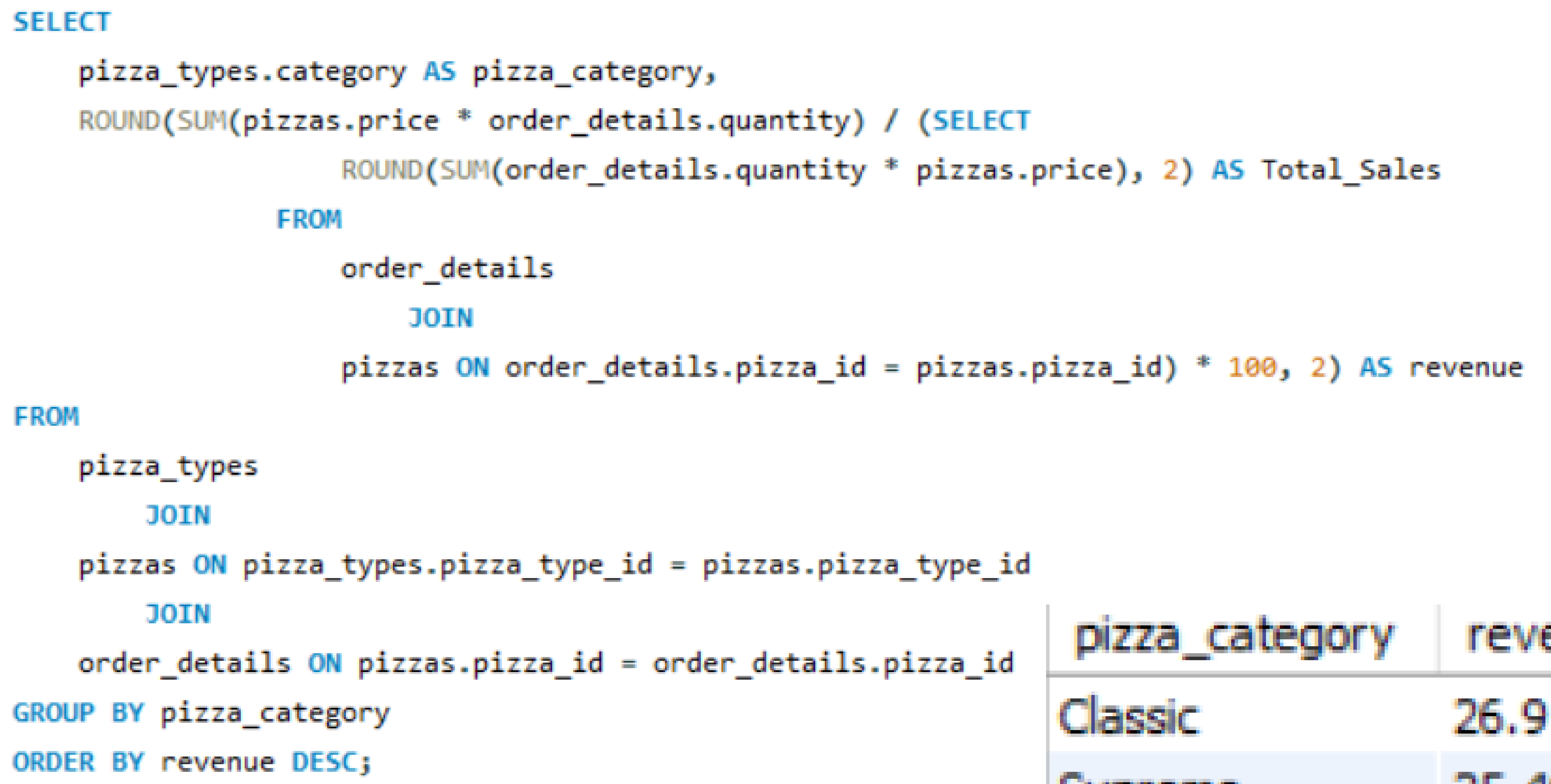


```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue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

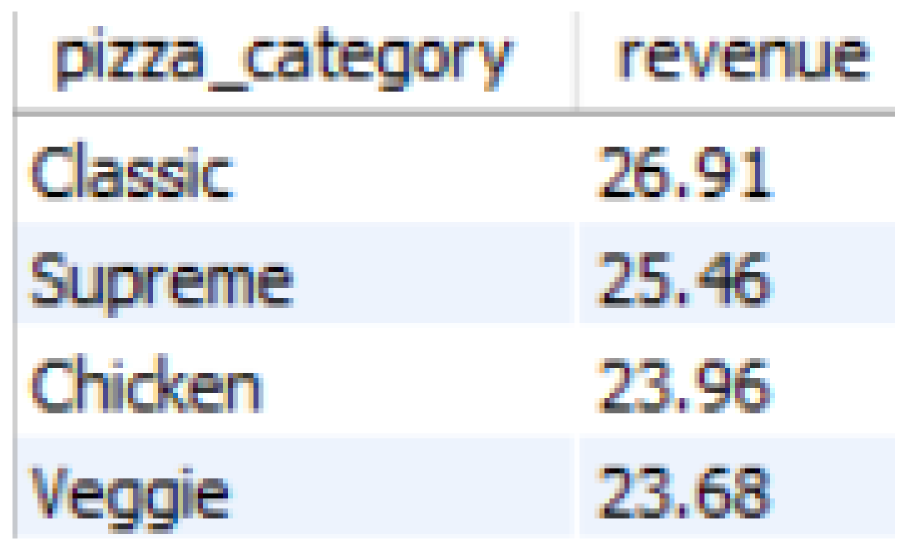


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

11. Contribution of Revenue by Each Pizza Category in Percentage



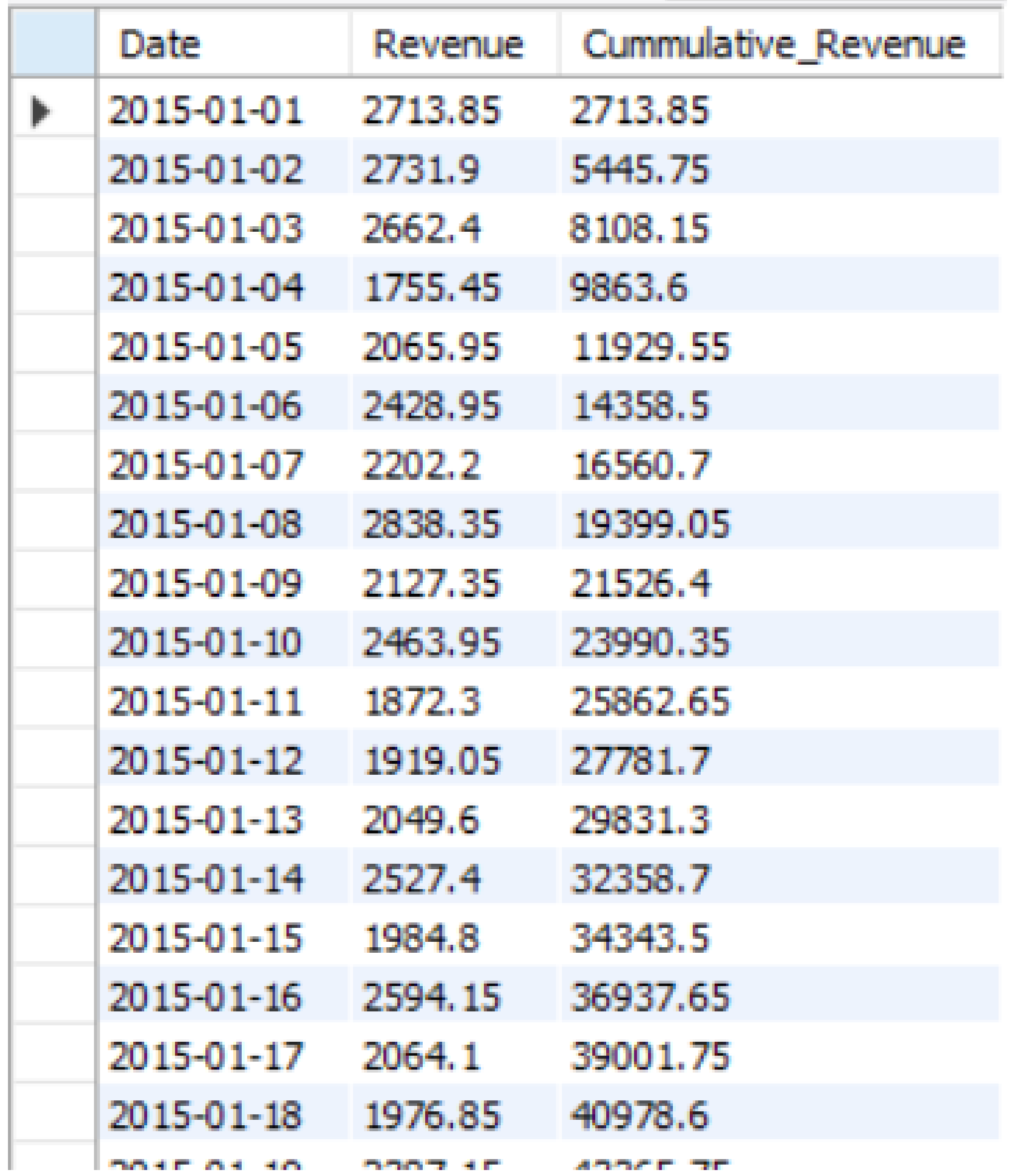
```sql
SELECT
    pizza_types.category AS pizza_category,
    ROUND(SUM(pizzas.price * order_details.quantity) / (SELECT
                ROUND(SUM(order_details.quantity * pizzas.price), 2) AS Total_Sales
            FROM
                order_details
                    JOIN
                pizzas ON order_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
    order_details ON pizzas.pizza_id = order_details.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

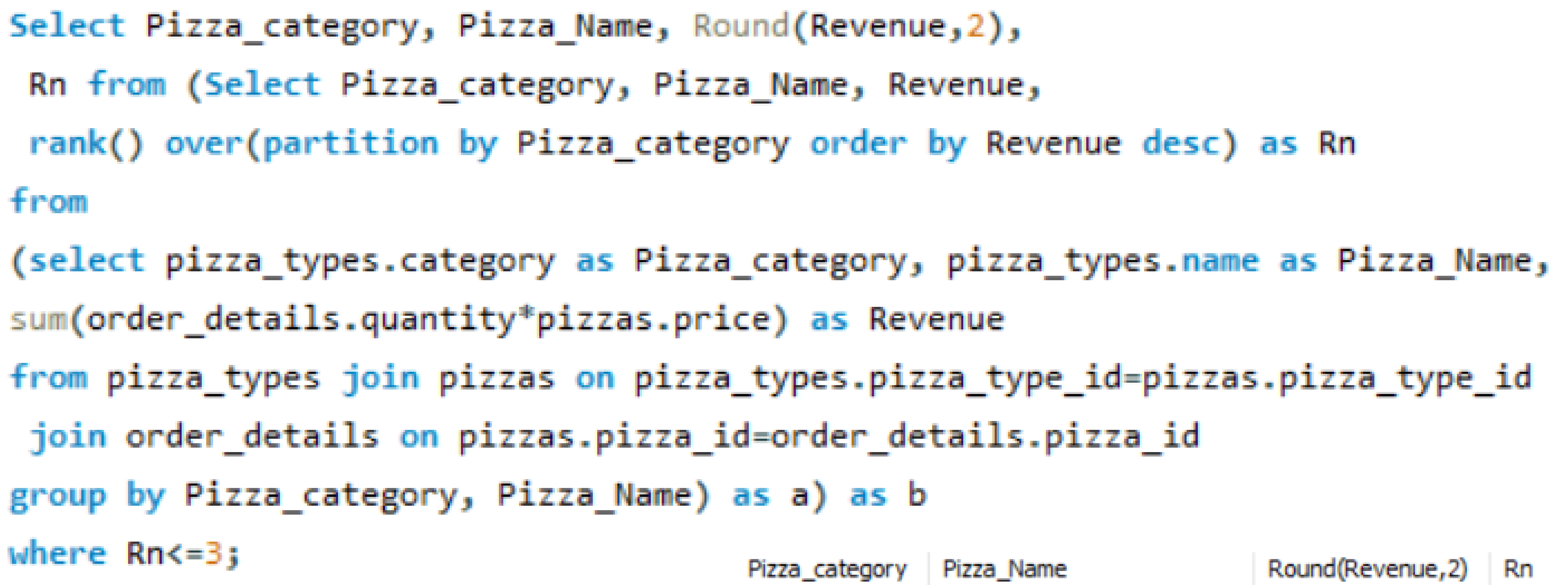
12. Cumulative revenue generated over Time

```sql
Select Date, Revenue, Round(sum(Revenue) over(order by Date),2) as Cummulative_Revenue from
(select orders.order_date as Date, round(sum(pizzas.price*order_details.quantity),2) as Revenue
from orders join order_details on orders.order_id=order_details.order_id join pizzas on
pizzas.pizza_id=order_details.pizza_id
group by Date) as rev_per_day;
```

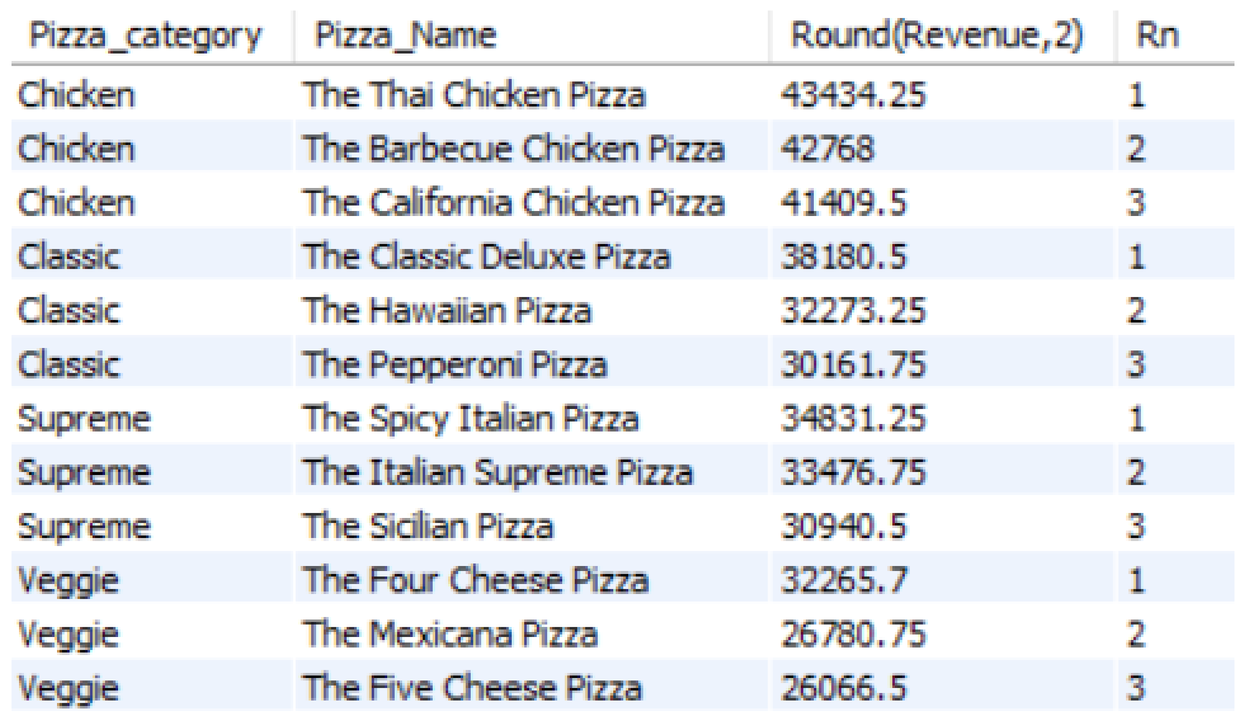


Date	Revenue	Cummulative_Revenue
2015-01-01	2713.85	2713.85
2015-01-02	2731.9	5445.75
2015-01-03	2662.4	8108.15
2015-01-04	1755.45	9863.6
2015-01-05	2065.95	11929.55
2015-01-06	2428.95	14358.5
2015-01-07	2202.2	16560.7
2015-01-08	2838.35	19399.05
2015-01-09	2127.35	21526.4
2015-01-10	2463.95	23990.35
2015-01-11	1872.3	25862.65
2015-01-12	1919.05	27781.7
2015-01-13	2049.6	29831.3
2015-01-14	2527.4	32358.7
2015-01-15	1984.8	34343.5
2015-01-16	2594.15	36937.65
2015-01-17	2064.1	39001.75
2015-01-18	1976.85	40978.6
[illegible]	[illegible]	[illegible]

13.Category-wise Top 3 Revenue Generated Pizzas

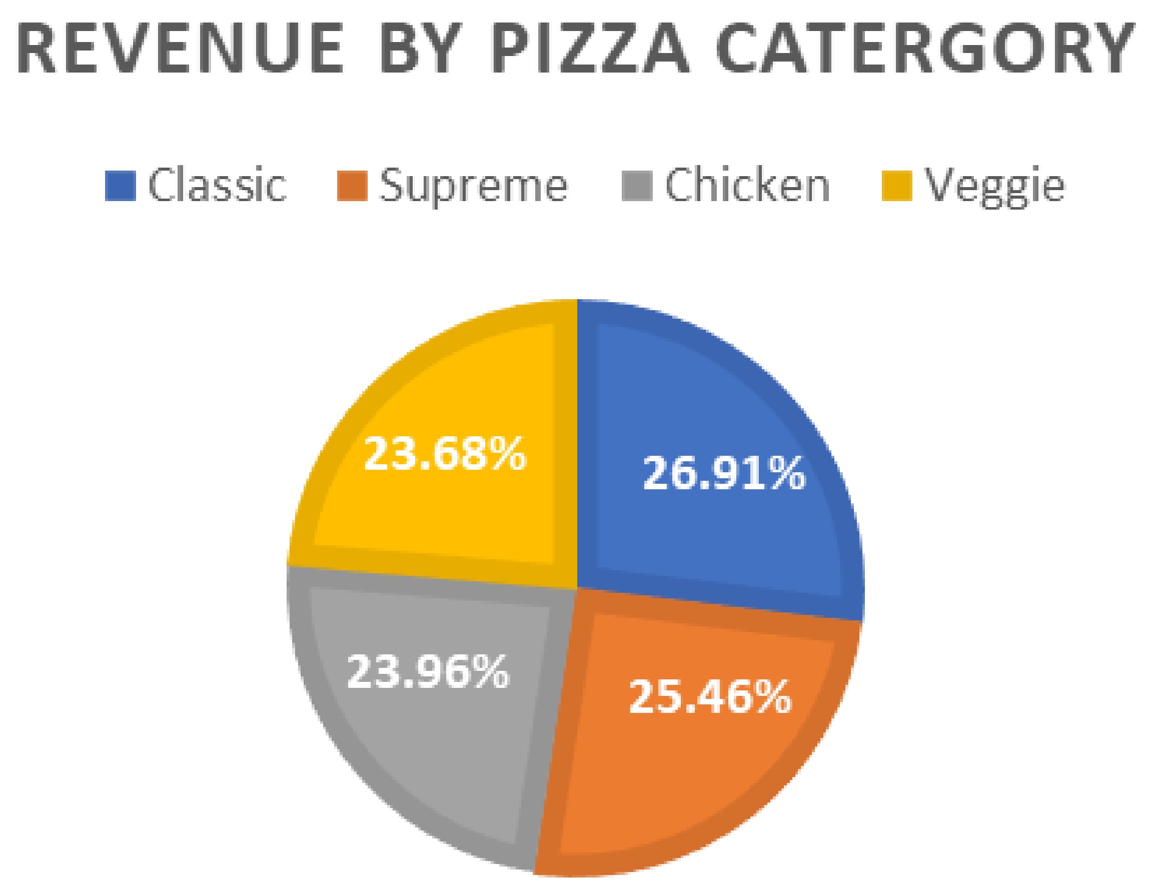


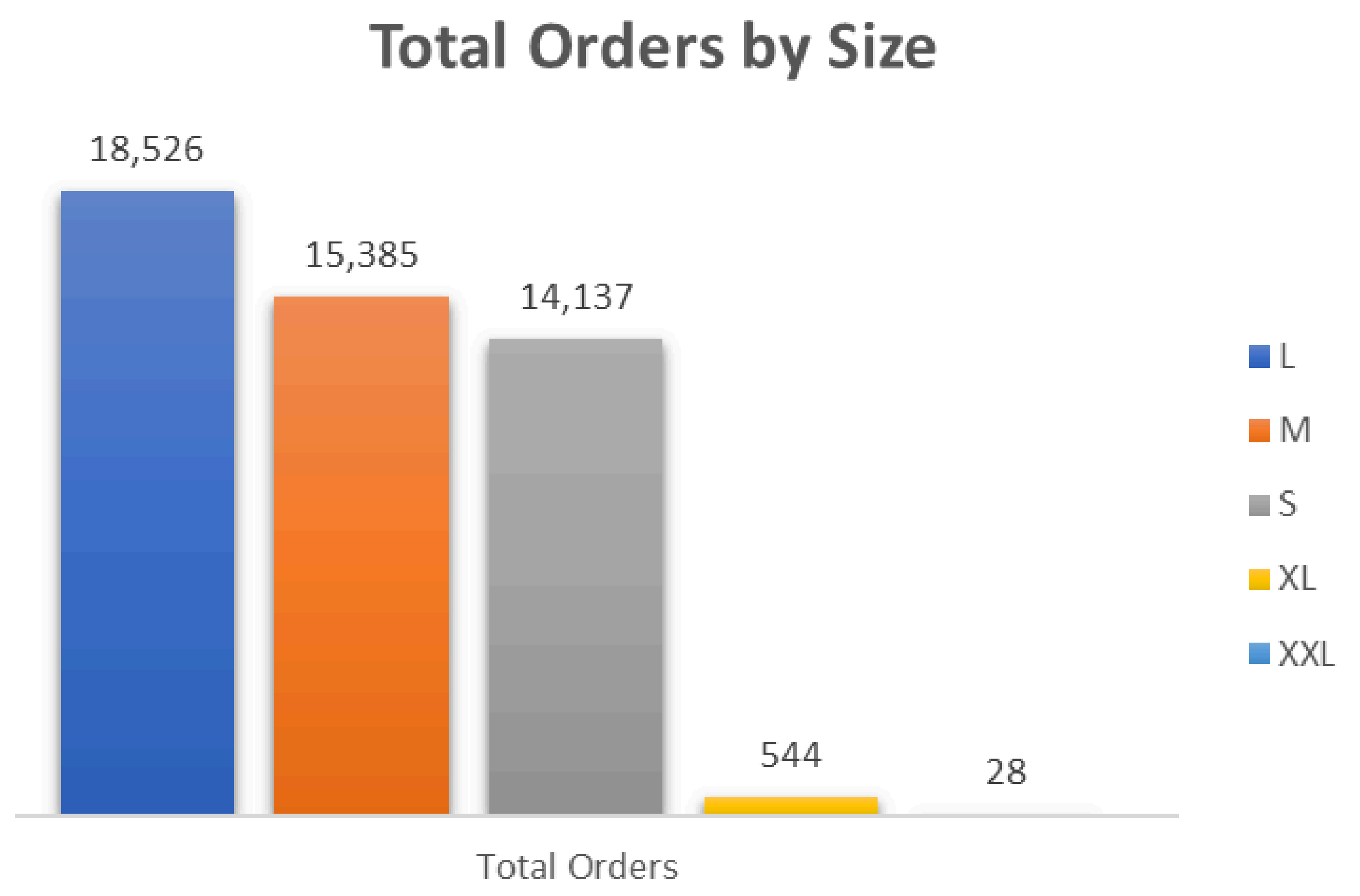
```
Select Pizza_category, Pizza_Name, Round(Revenue,2),
 Rn from (Select Pizza_category, Pizza_Name, Revenue,
 rank() over(partition by Pizza_category order by Revenue desc) as Rn
from
(select pizza_types.category as Pizza_category, pizza_types.name as Pizza_Name,
sum(order_details.quantity*pizzas.price) as Revenue
from pizza_types join pizzas on pizza_types.pizza_type_id=pizzas.pizza_type_id
 join order_details on pizzas.pizza_id=order_details.pizza_id
group by Pizza_category, Pizza_Name) as a) as b
where Rn<=3;
```

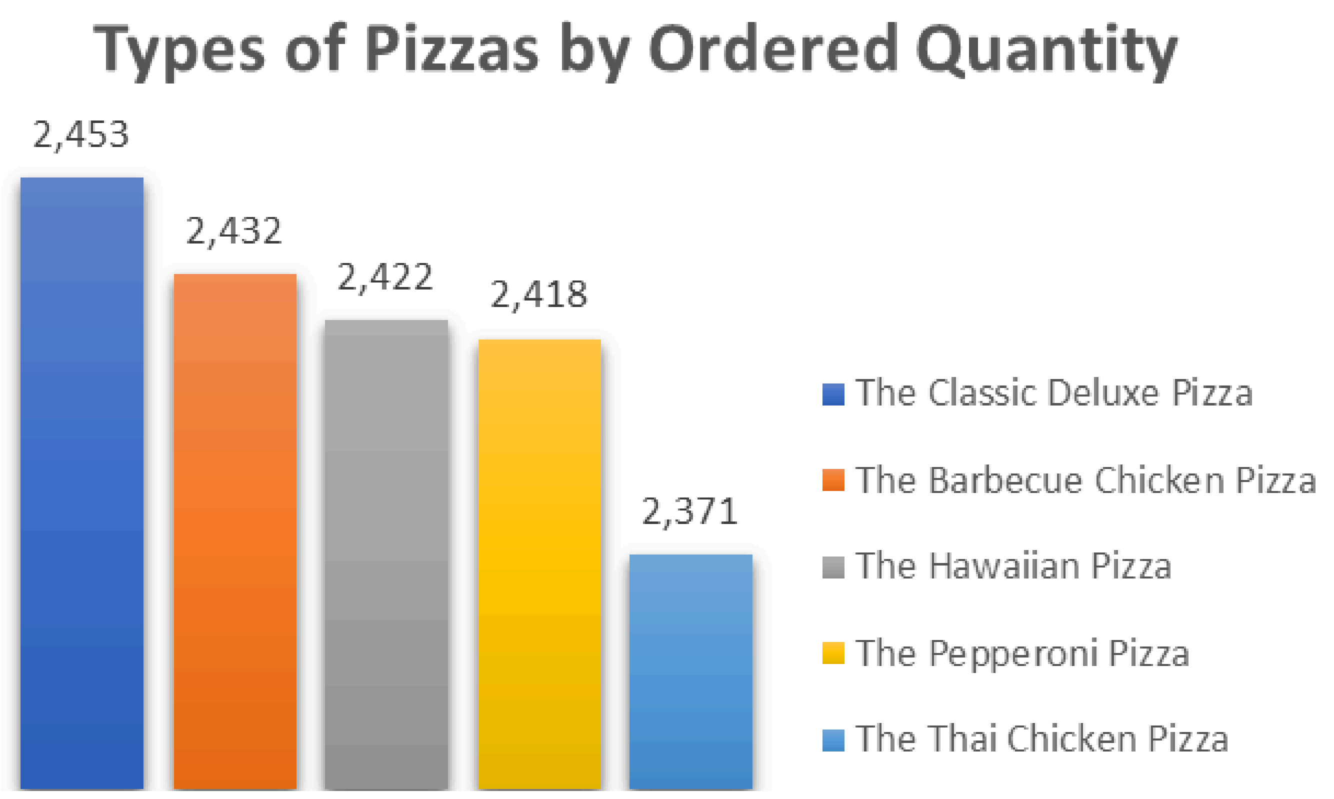


Pizza_category	Pizza_Name	Round(Revenue,2)	Rn
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Spicy Italian Pizza	34831.25	1
Supreme	The Italian Supreme Pizza	33476.75	2
Supreme	The Sicilian Pizza	30940.5	3
Veggie	The Four Cheese Pizza	32265.7	1
Veggie	The Mexicana Pizza	26780.75	2
Veggie	The Five Cheese Pizza	26066.5	3

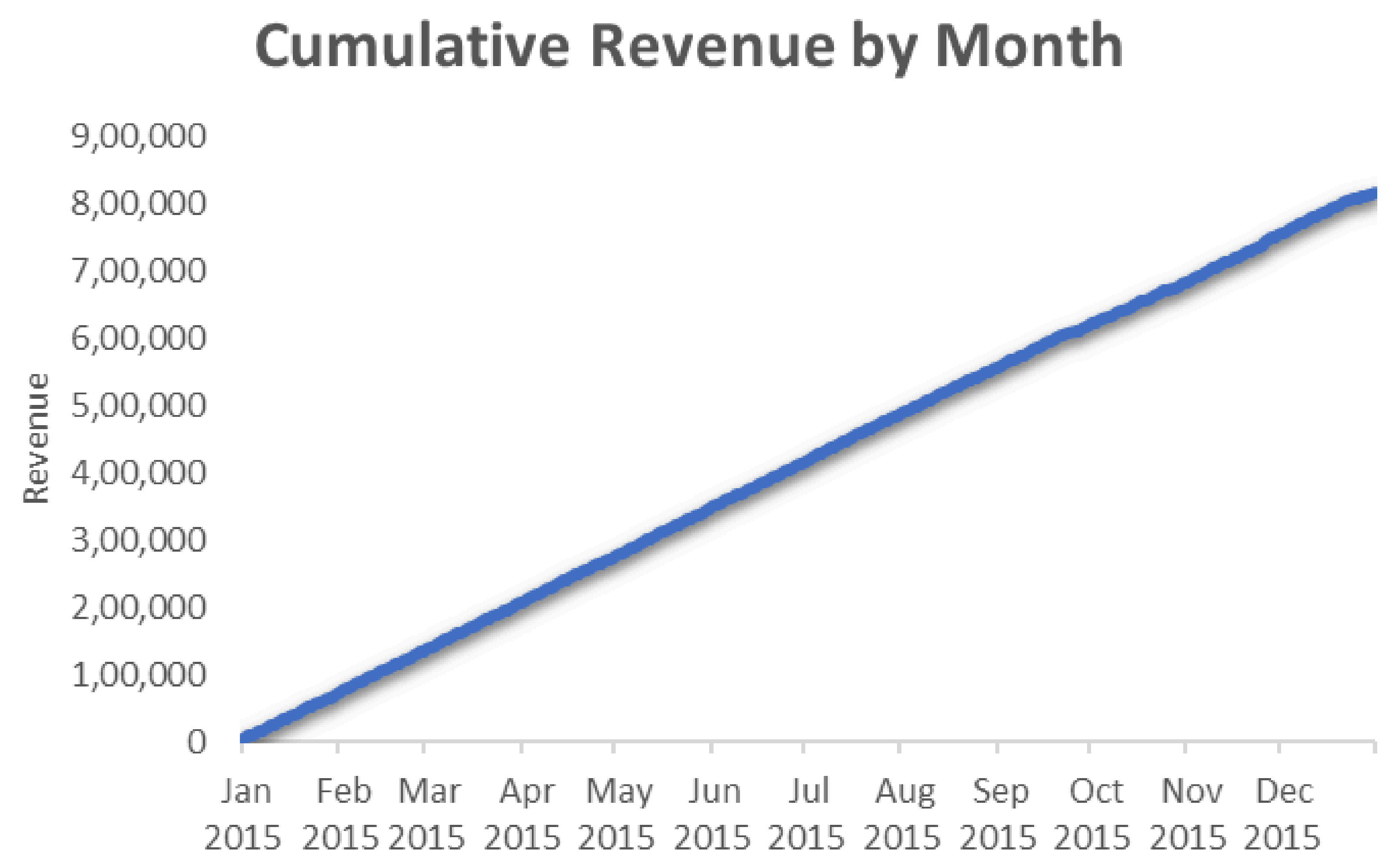
Visualization

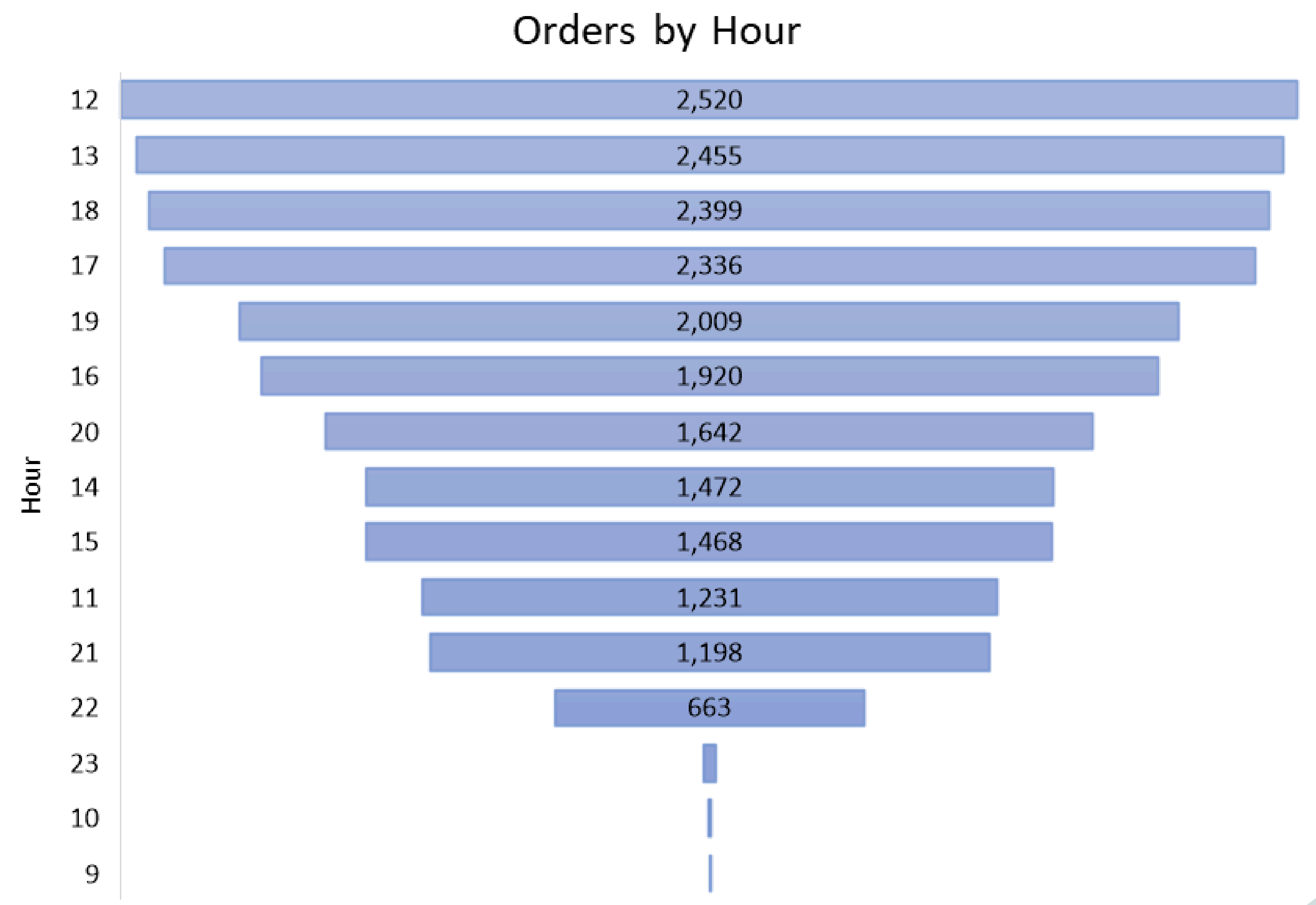






Visualization





Key Learnings

Enhanced SQL proficiency, including:

- Writing complex JOINs across multiple tables.
- Using aggregate functions for summarizing data.
- Applying filters and grouping for deeper insights.

Developed a structured approach to answering business questions through queries.

Gained hands-on experience with real-world relational datasets.

Conclusion

This SQL project provided a comprehensive analysis of pizza sales data, demonstrating the power of SQL in extracting actionable insights from structured datasets. Key findings include:

- Sales Performance: Identified the most ordered pizzas and their revenue contribution, highlighting customer preferences.
- Category Insights: Revealed the distribution of pizza sales across categories and sizes, aiding in inventory planning.
- Revenue Trends: Analyzed cumulative revenue growth over time, showcasing seasonal patterns and sales peaks.
- Operational Efficiency: Determined peak order hours, enabling better resource allocation during high-demand periods.

Overall, this project reinforced SQL concepts such as joins, aggregations, and data filtering while providing real-world insights into operational and sales metrics.