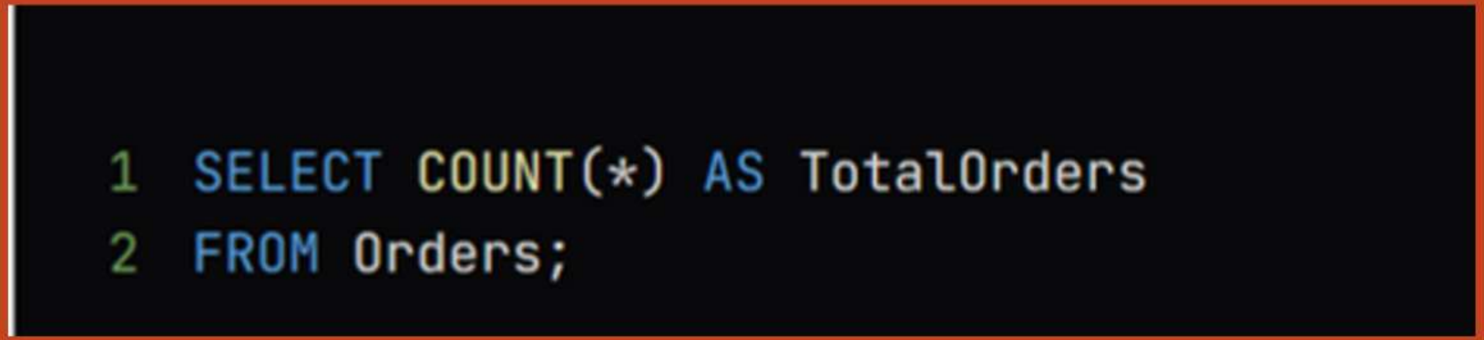
Pizza Analytics

SQL Queries Overview

PRESENTED BY

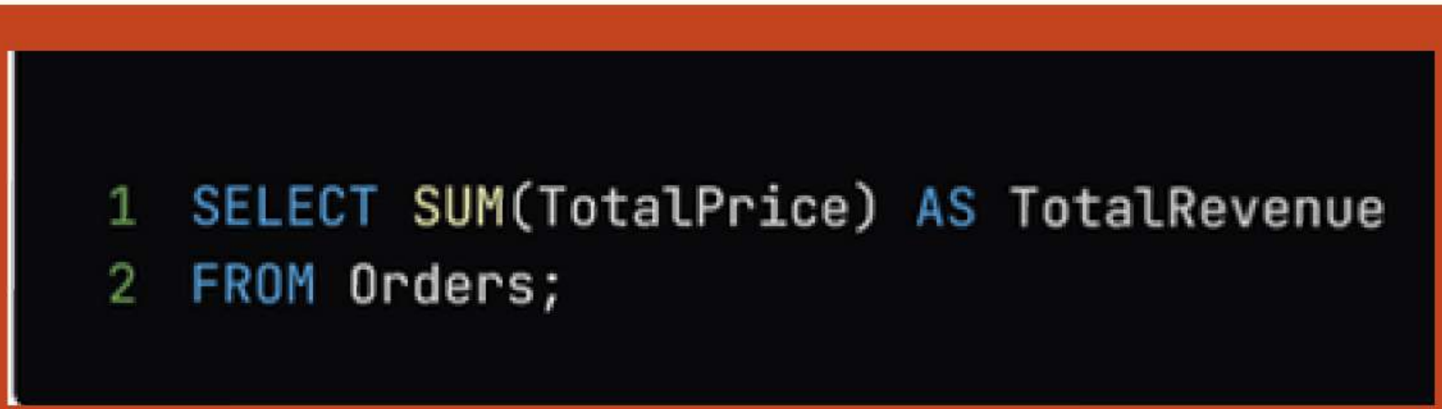
KUMAR GAUTAM SINGH

Total Orders



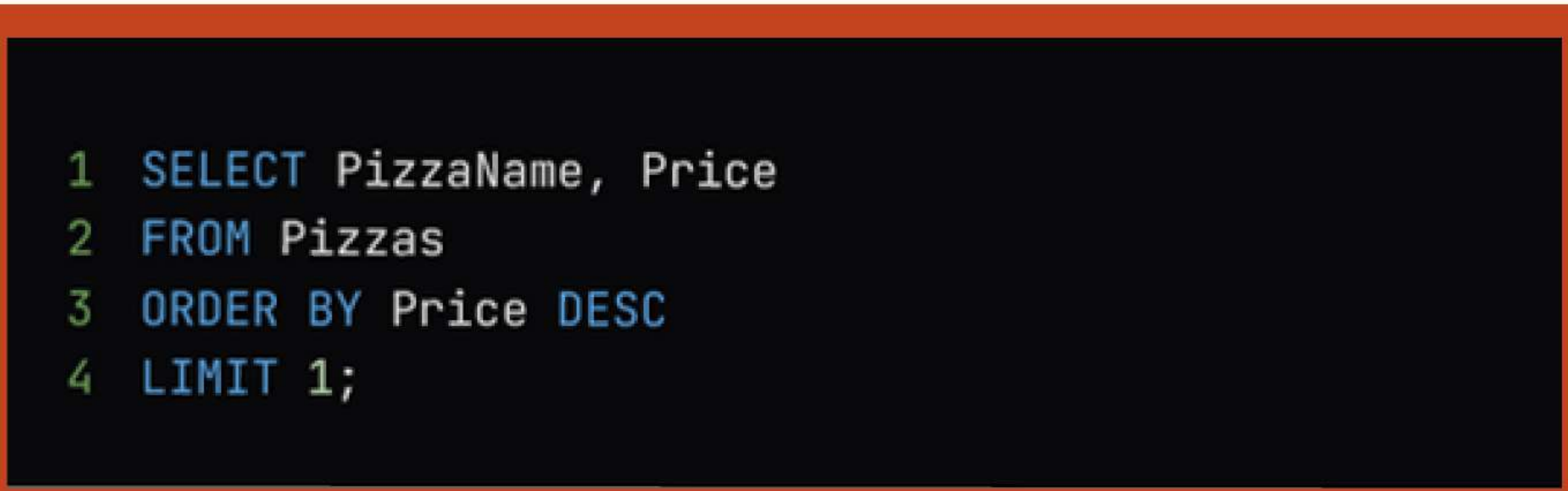
```sql
SELECT COUNT(*) AS TotalOrders
FROM Orders;
```

Total Pizza Revenue



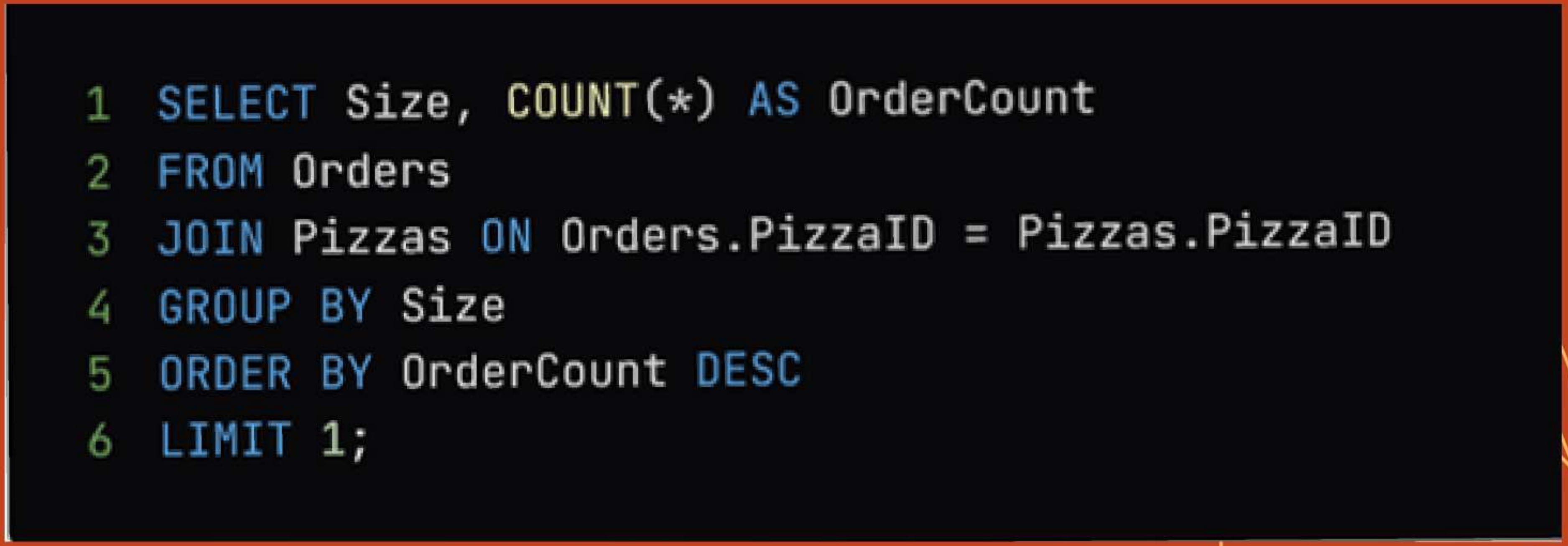
```sql
SELECT SUM(TotalPrice) AS TotalRevenue
FROM Orders;
```

Highest-Priced Pizza



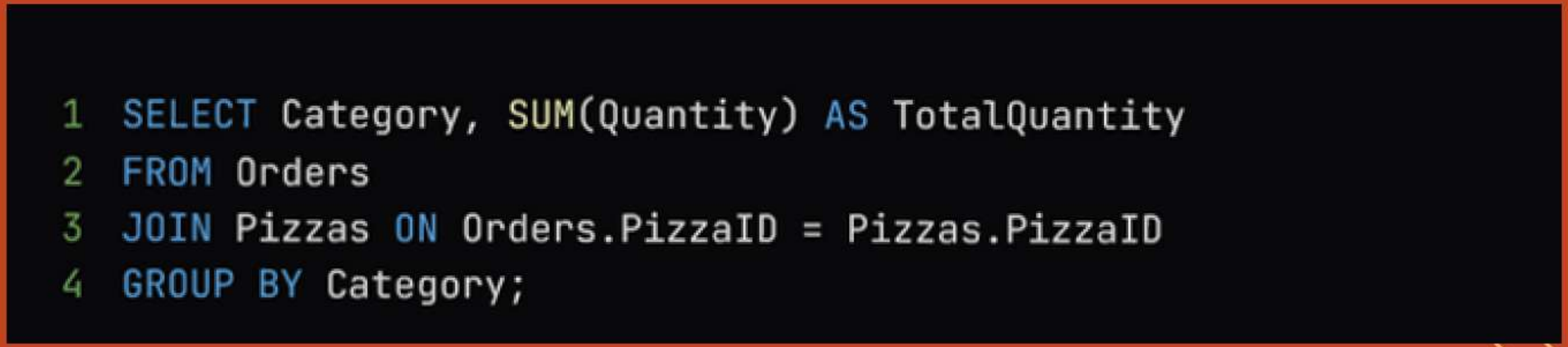
```sql
SELECT PizzaName, Price
FROM Pizzas
ORDER BY Price DESC
LIMIT 1;
```

Most Common Size



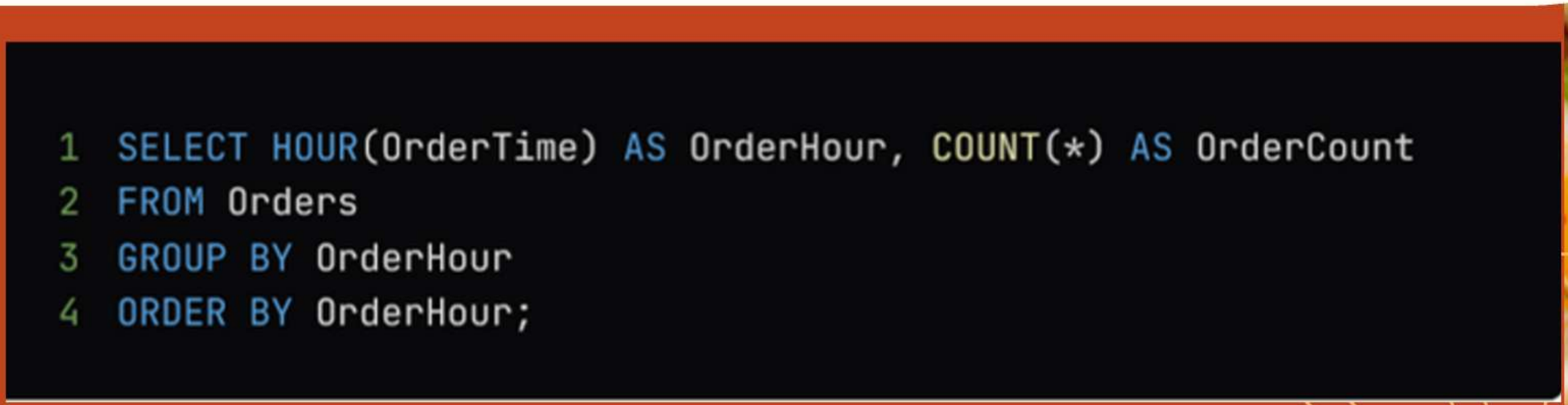
```sql
SELECT Size, COUNT(*) AS OrderCount
FROM Orders
JOIN Pizzas ON Orders.PizzaID = Pizzas.PizzaID
GROUP BY Size
ORDER BY OrderCount DESC
LIMIT 1;
```

Top 5 Pizzas



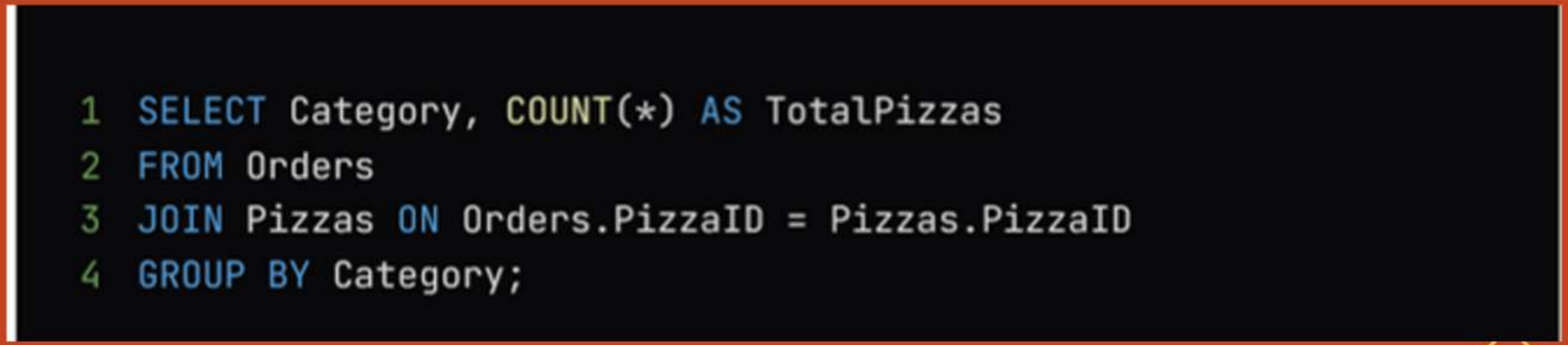
```sql
SELECT Category, SUM(Quantity) AS TotalQuantity
FROM Orders
JOIN Pizzas ON Orders.PizzaID = Pizzas.PizzaID
GROUP BY Category;
```

Determine the distribution of orders by hour of the day.



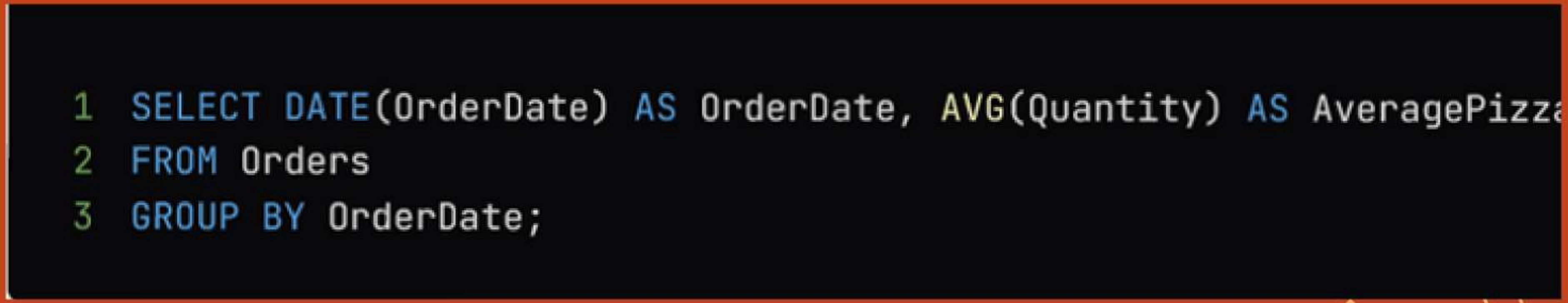
```sql
SELECT HOUR(OrderTime) AS OrderHour, COUNT(*) AS OrderCount
FROM Orders
GROUP BY OrderHour
ORDER BY OrderHour;
```

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



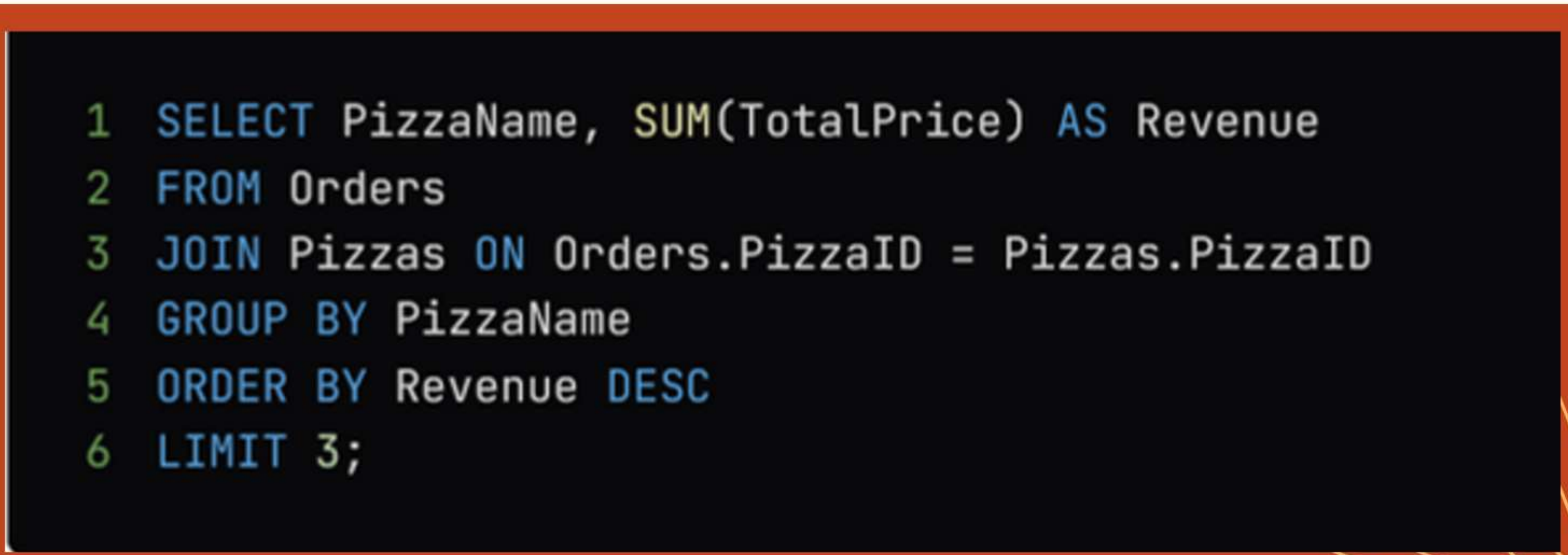
```sql
SELECT Category, COUNT(*) AS TotalPizzas
FROM Orders
JOIN Pizzas ON Orders.PizzaID = Pizzas.PizzaID
GROUP BY Category;
```

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



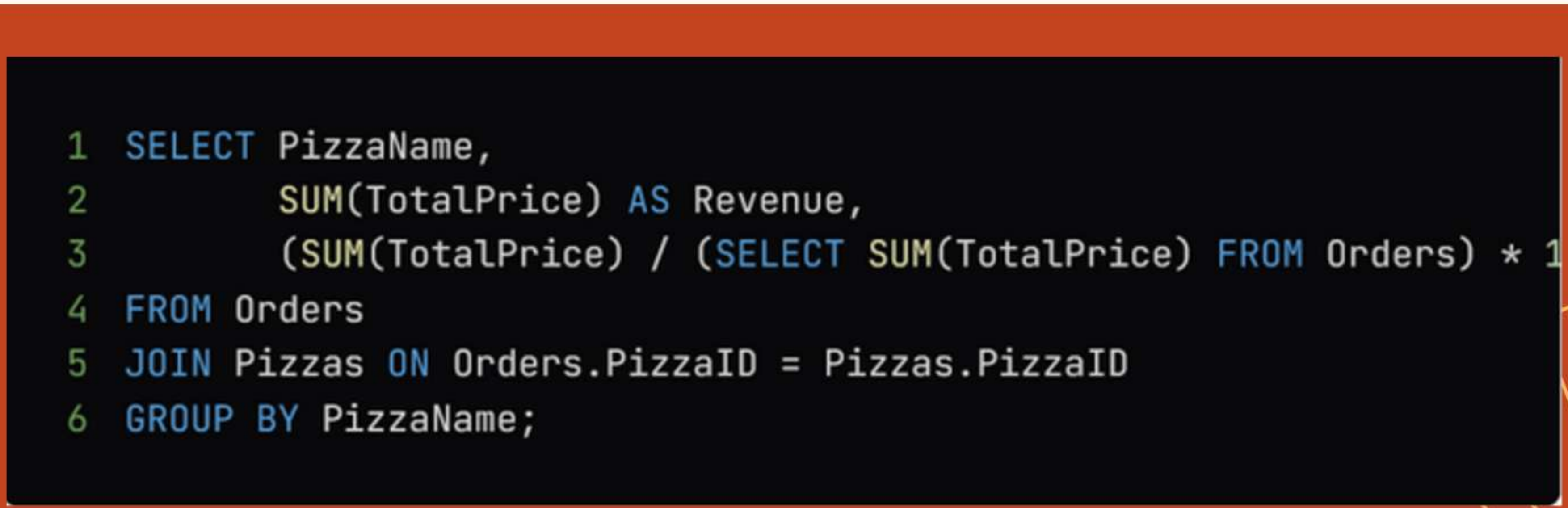
```sql
SELECT DATE(OrderDate) AS OrderDate, AVG(Quantity) AS AveragePizza
FROM Orders
GROUP BY OrderDate;
```

Determine the top 3 most ordered pizza types based on revenue.Top 5 Pizzas



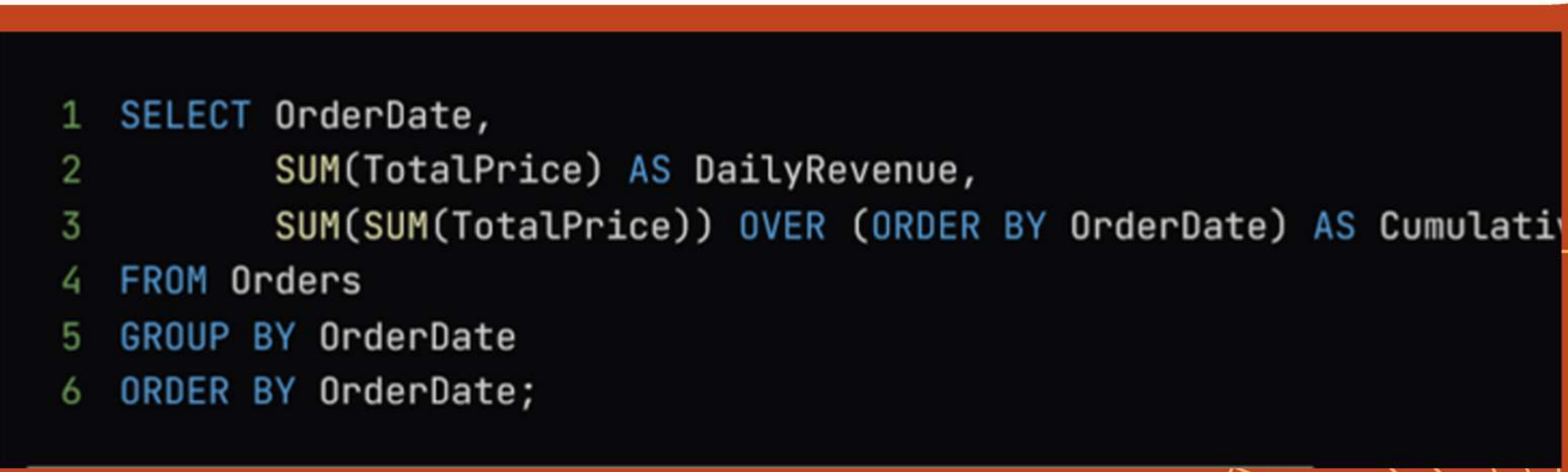
```sql
SELECT PizzaName, SUM(TotalPrice) AS Revenue
FROM Orders
JOIN Pizzas ON Orders.PizzaID = Pizzas.PizzaID
GROUP BY PizzaName
ORDER BY Revenue DESC
LIMIT 3;
```

Calculate the percentage contribution of each pizza type to total revenue



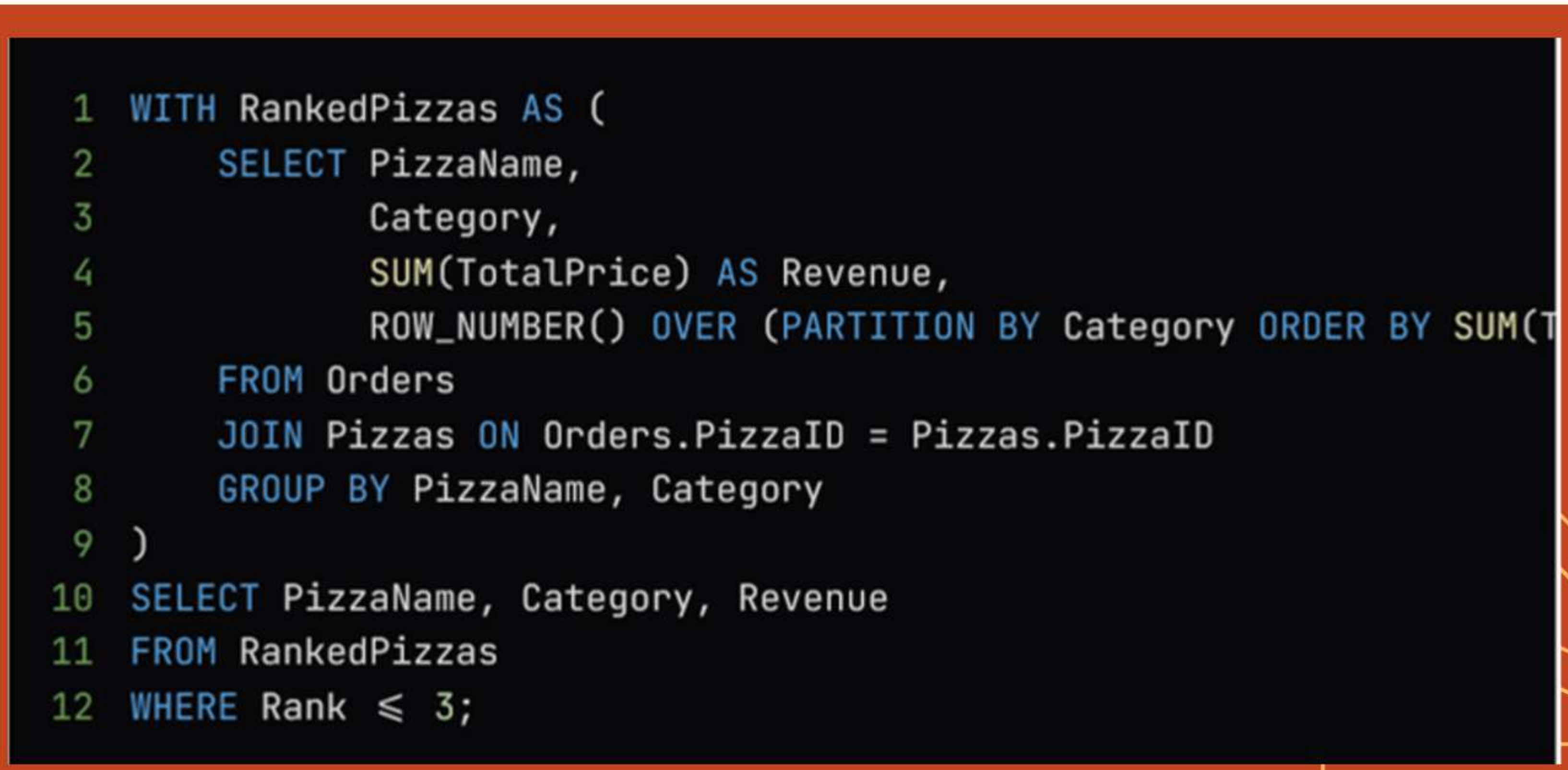
```sql
SELECT PizzaName,
       SUM(TotalPrice) AS Revenue,
       (SUM(TotalPrice) / (SELECT SUM(TotalPrice) FROM Orders) * 1
FROM Orders
JOIN Pizzas ON Orders.PizzaID = Pizzas.PizzaID
GROUP BY PizzaName;
```

Analyze the cumulative revenue generated over time.



```
SELECT OrderDate,
       SUM(TotalPrice) AS DailyRevenue,
       SUM(SUM(TotalPrice)) OVER (ORDER BY OrderDate) AS Cumulativ
FROM Orders
GROUP BY OrderDate
ORDER BY OrderDate;
```

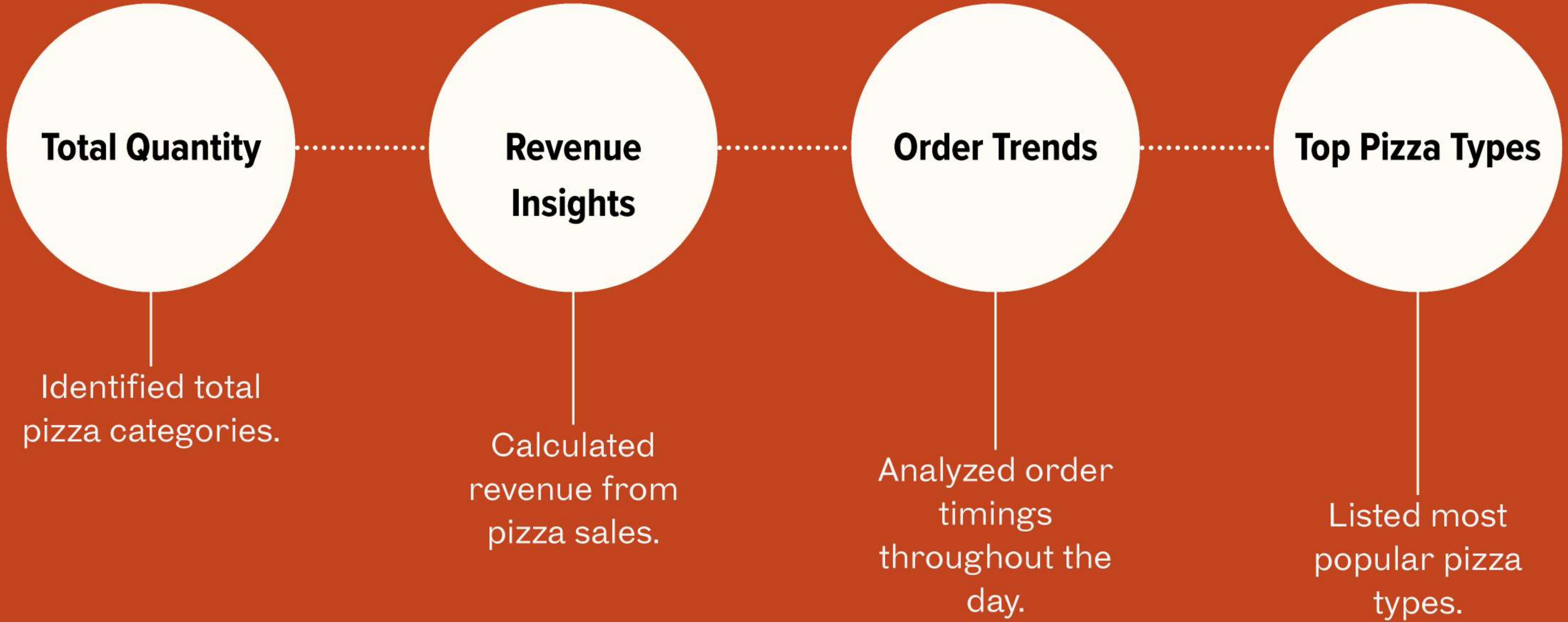
Determine the top 3 most ordered pizza types based on revenue for each pizza category.



```sql
WITH RankedPizzas AS (
    SELECT PizzaName,
           Category,
           SUM(TotalPrice) AS Revenue,
           ROW_NUMBER() OVER (PARTITION BY Category ORDER BY SUM(T
    FROM Orders
    JOIN Pizzas ON Orders.PizzaID = Pizzas.PizzaID
    GROUP BY PizzaName, Category
)
SELECT PizzaName, Category, Revenue
FROM RankedPizzas
WHERE Rank <= 3;
```

Pizza Orders Analysis

Overview of important SQL queries



Total Quantity

Identified total pizza categories.

Revenue Insights

Calculated revenue from pizza sales.

Order Trends

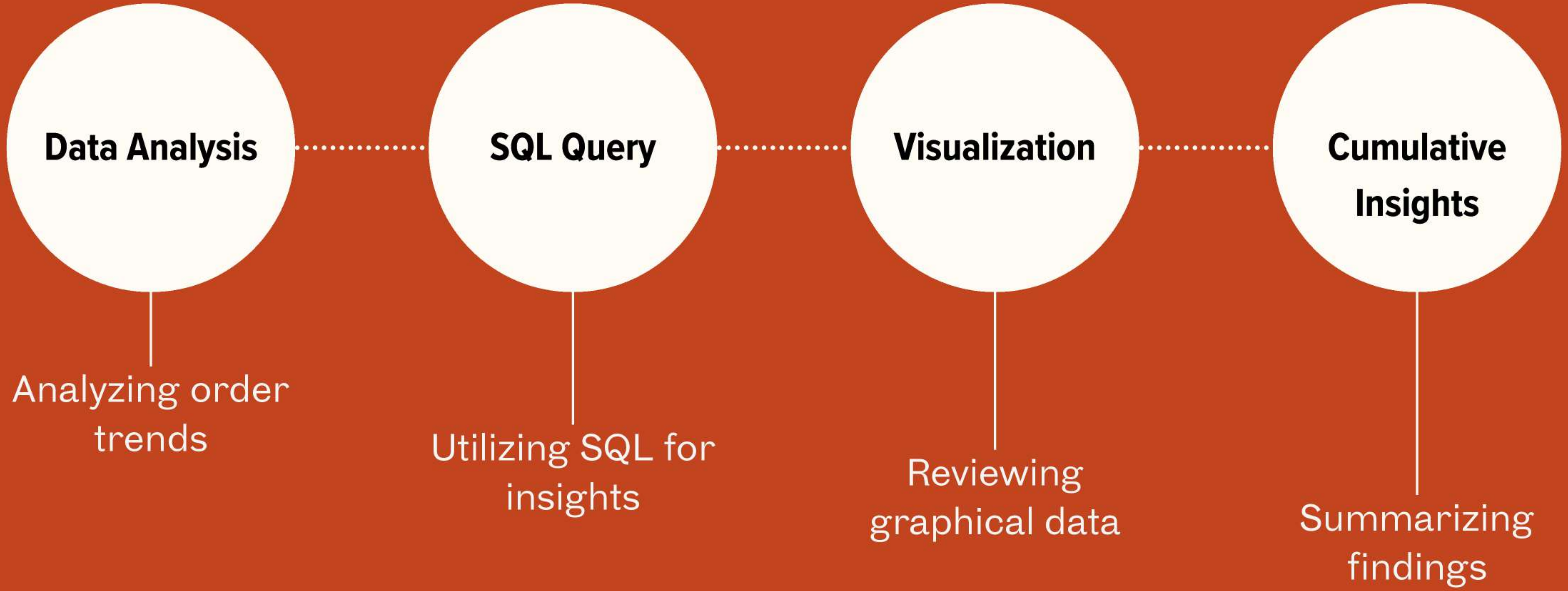
Analyzed order timings throughout the day.

Top Pizza Types

Listed most popular pizza types.

Order Distribution

Understanding orders by time of day



Data Analysis

Analyzing order trends

SQL Query

Utilizing SQL for insights

Visualization

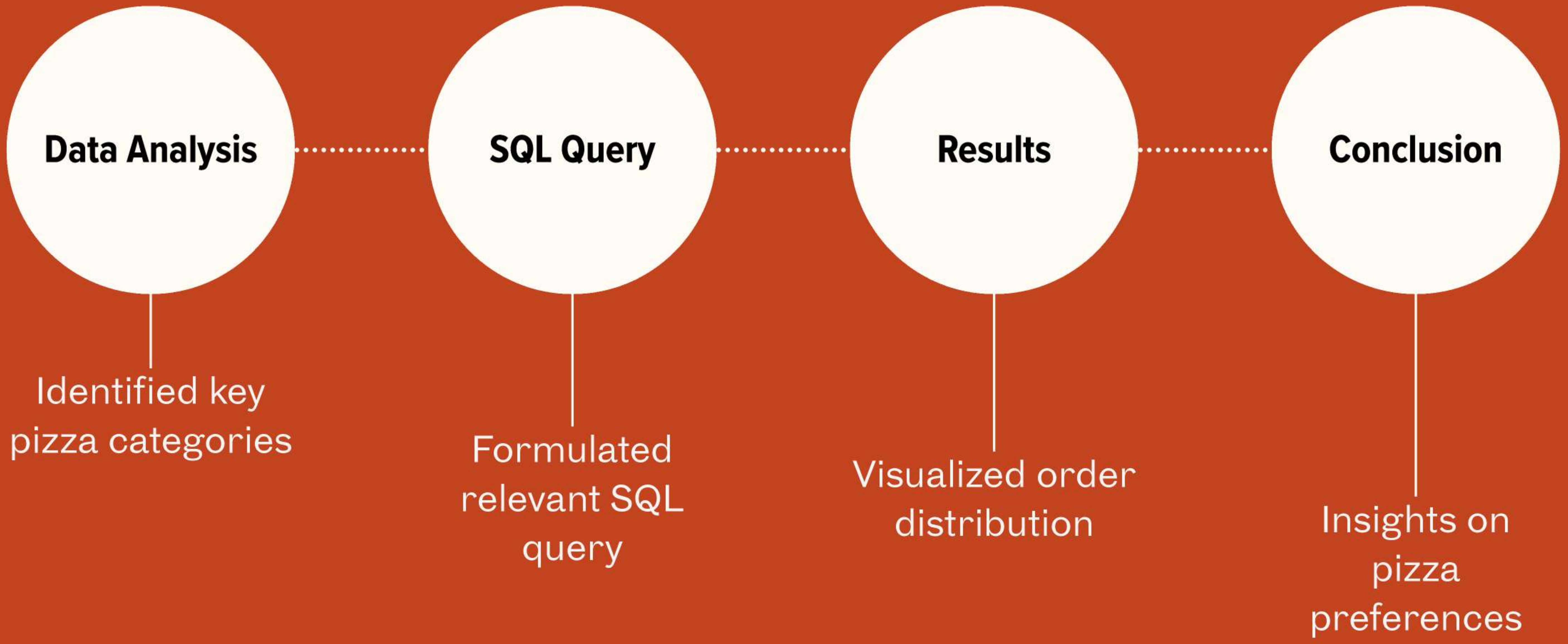
Reviewing graphical data

Cumulative Insights

Summarizing findings

Pizza Category Distribution

Overview of pizza order categories



Data Analysis

Identified key pizza categories

SQL Query

Formulated relevant SQL query

Results

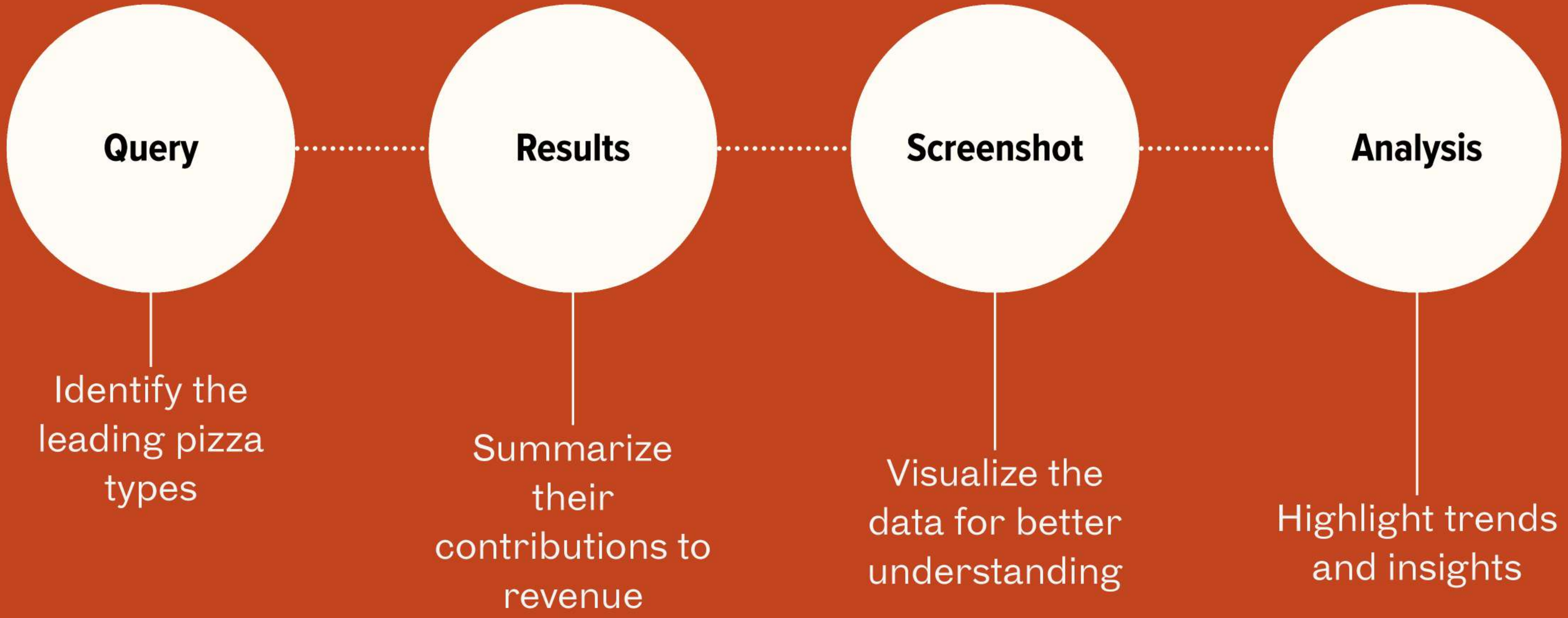
Visualized order distribution

Conclusion

Insights on pizza preferences

Top 3 Pizzas

Based on revenue analysis



Query

Identify the leading pizza types

Results

Summarize their contributions to revenue

Screenshot

Visualize the data for better understanding

Analysis

Highlight trends and insights

THANK YOU

Thank you for taking the time to review your findings