

Pizza Sales Analysis

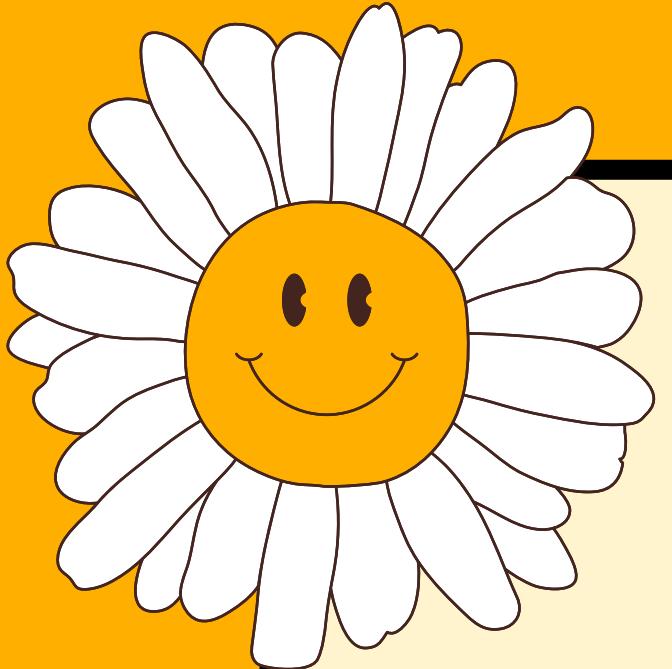
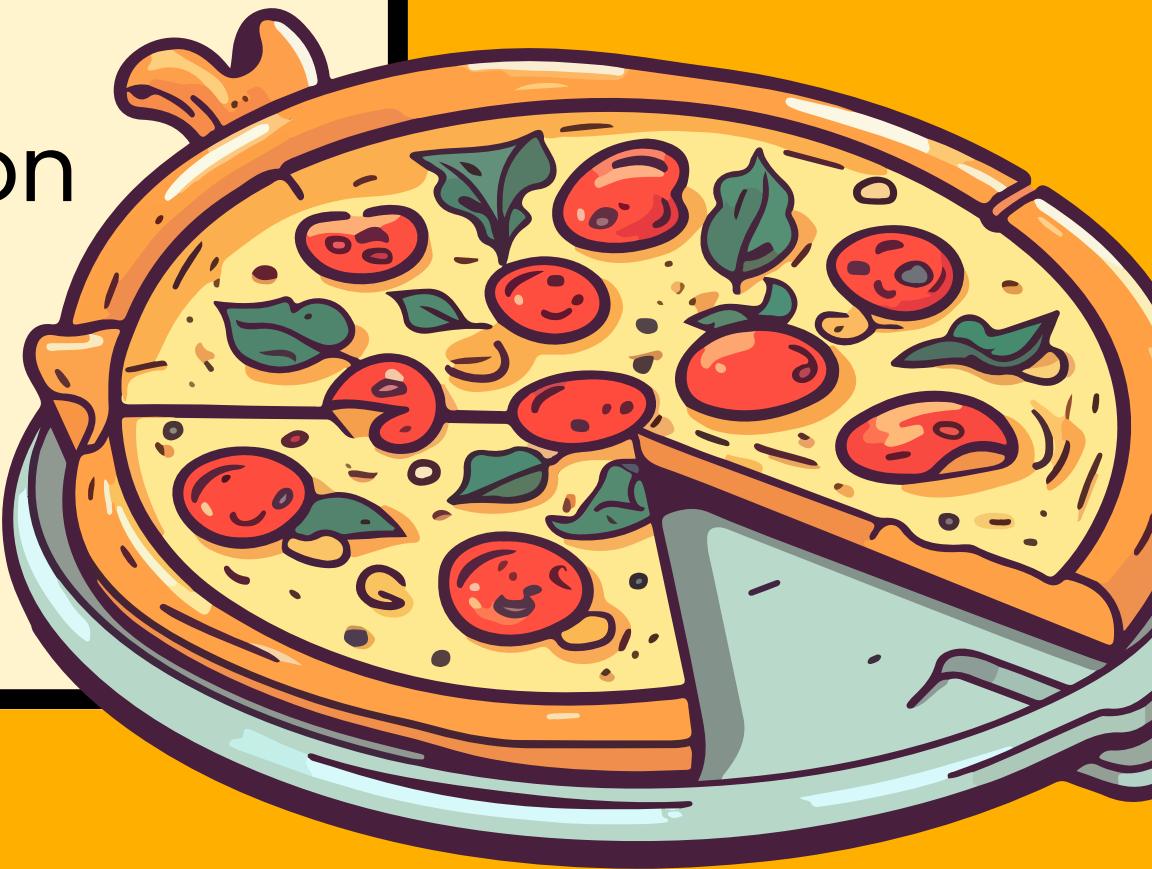
**ENHANCING PRODUCT SALES STRATEGY
USING SQL**

29 August, 2024

Prepared by Rajdeep

Introduction

In this project, I conducted a comprehensive analysis of a pizza sales dataset using SQL. The primary objective was to gain actionable insights into the sales performance, customer preferences, and revenue generation patterns of a pizza business.

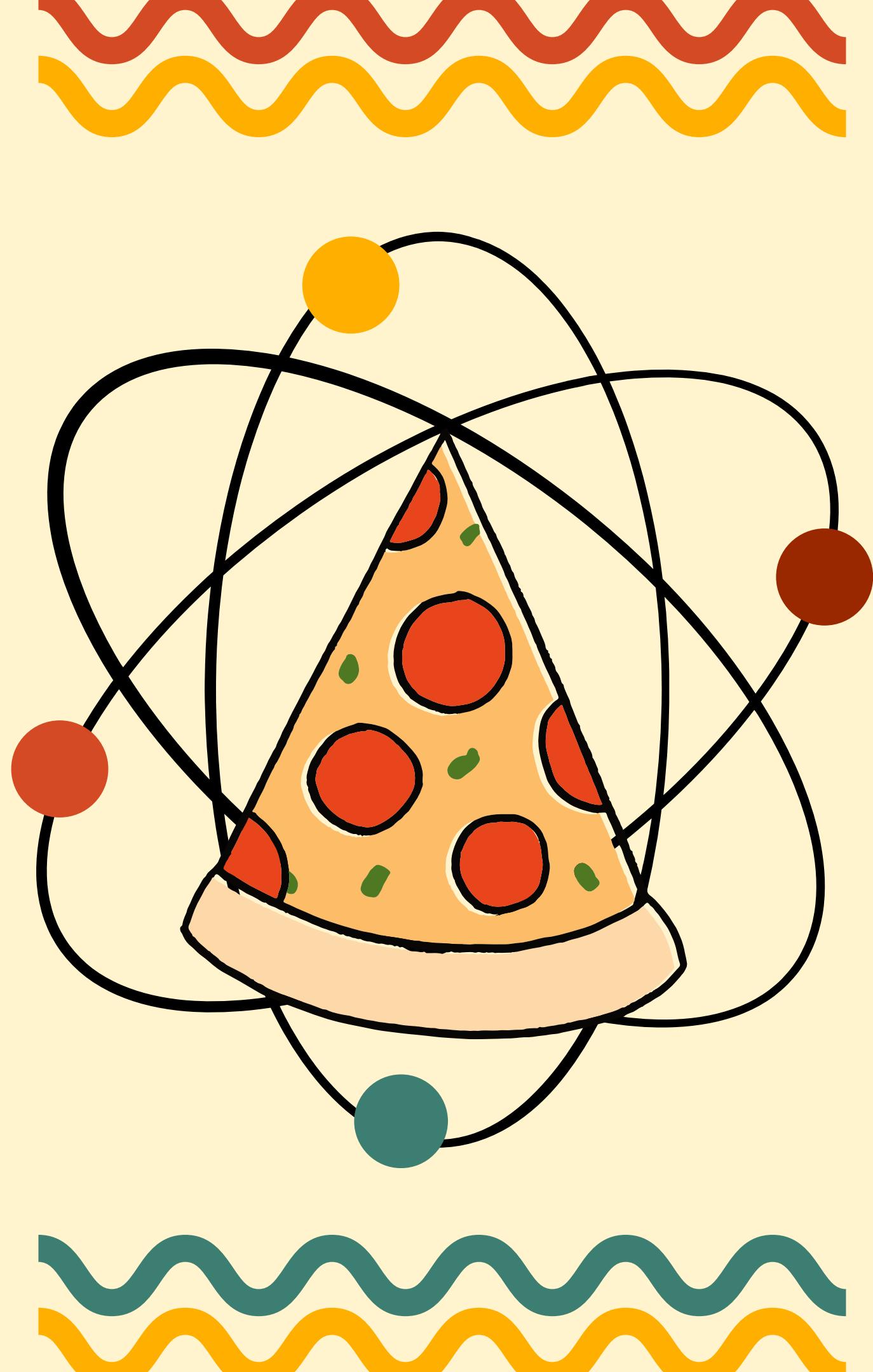


Background

Background project

Current scenario:

In today's competitive pizza industry, understanding sales patterns and customer preferences is crucial for growth. With the rise of online ordering, businesses have access to vast data but often struggle to utilize it effectively. This project addresses this by using SQL to analyze pizza sales data, providing insights into key trends, top-performing products, and revenue streams. The analysis helps businesses optimize their menu and enhance marketing strategies.

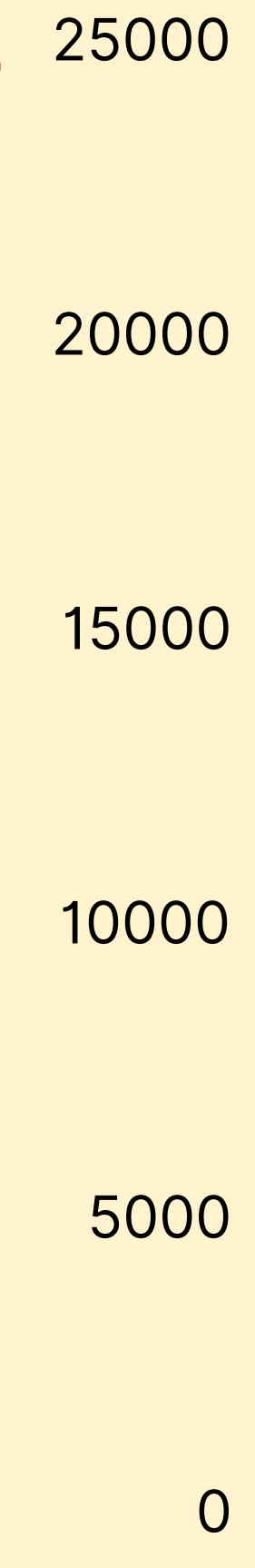


Methodology of Analysis

Data Analysis 1.1

```
SELECT  
distinct count(order_id) as  
total_orders  
FROM orders;
```

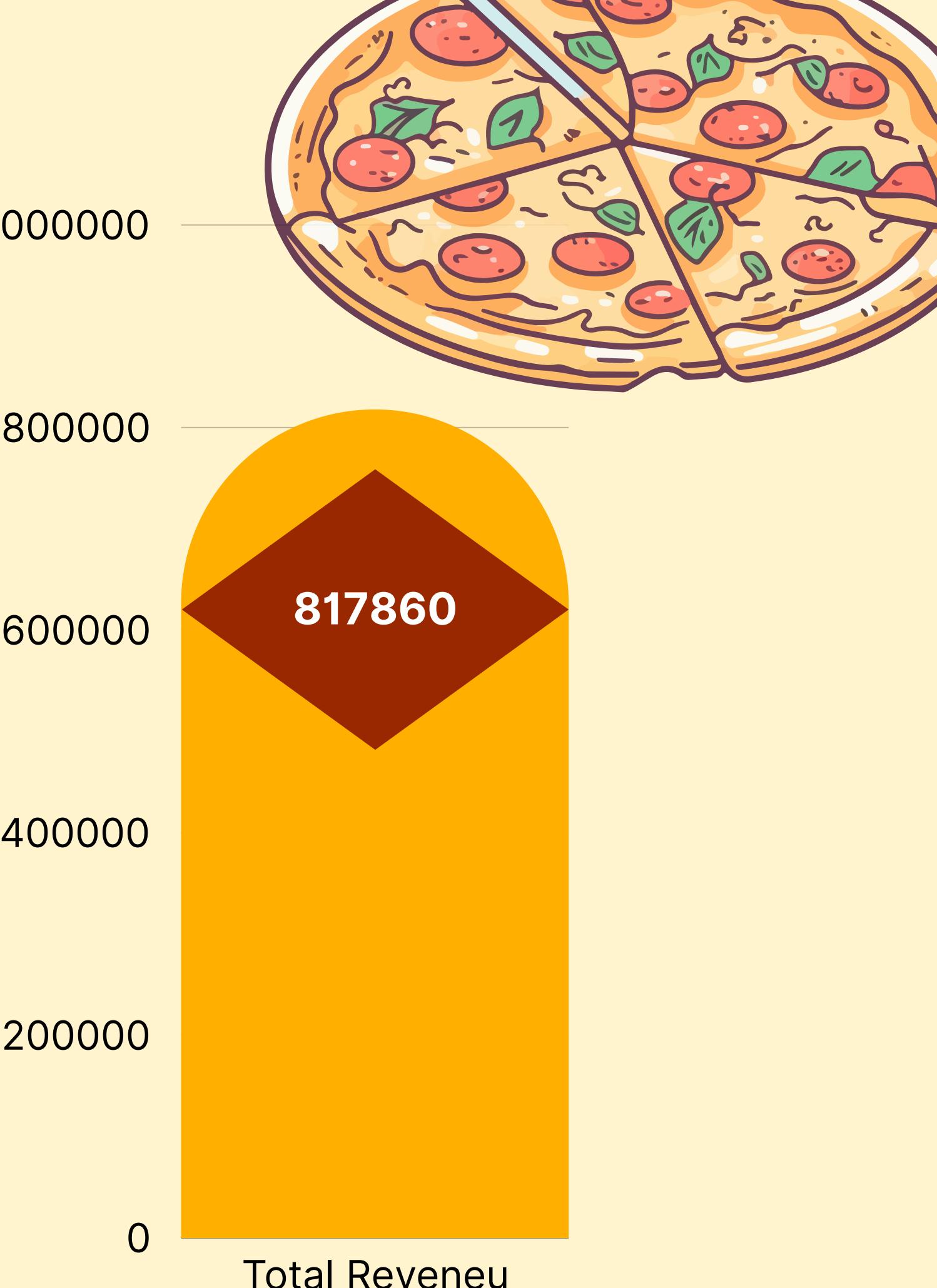
Total number of
orders placed.



Data Analysis 1.2

The total revenue generated from pizza sales.

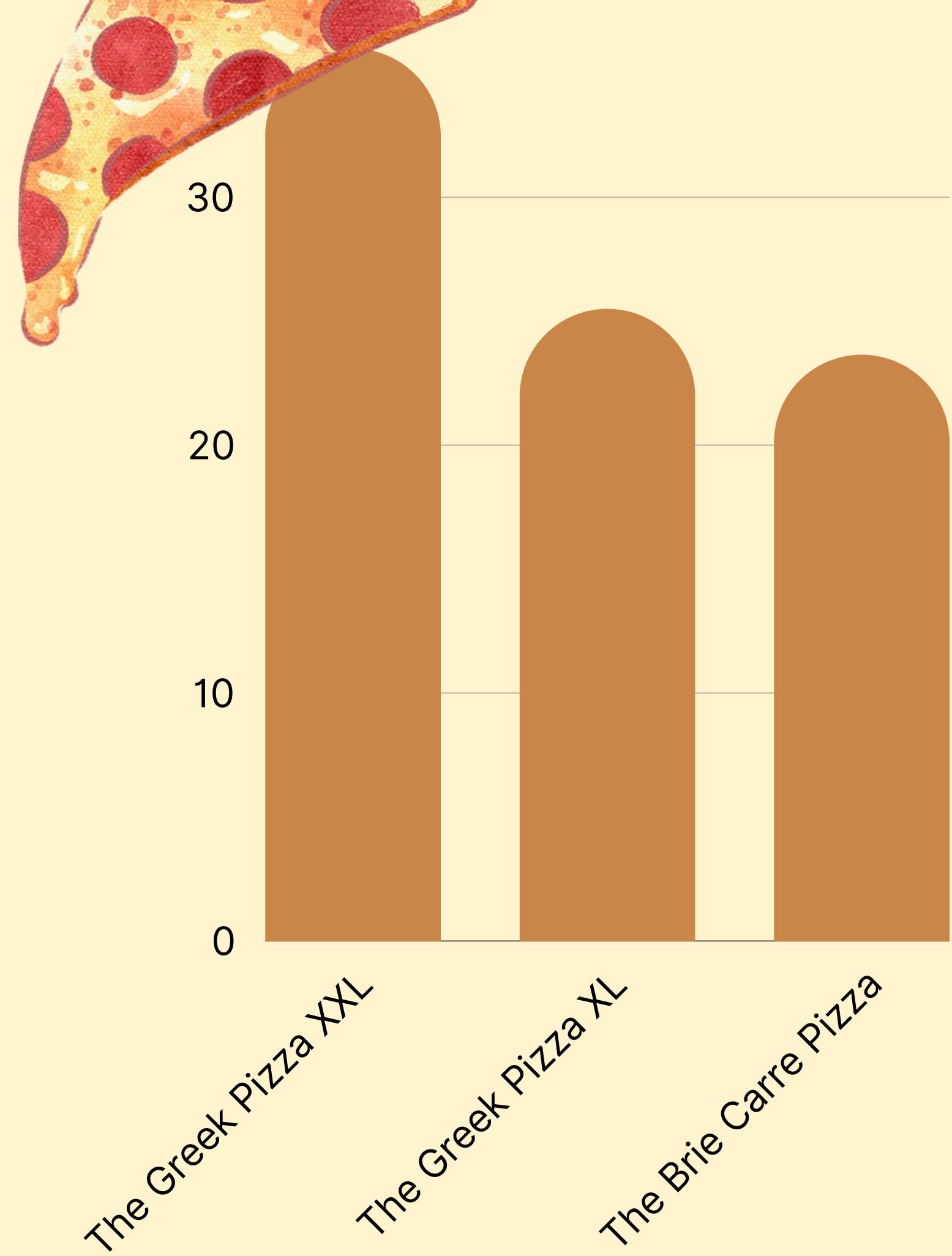
```
SELECT  
ROUND(SUM(A.quantity * B.price),2) as  
Total_price  
from  
order_details as A join pizzas B  
on A.pizza_id = B.pizza_id;
```



Data Analysis 1.3

Identify the highest-priced pizza.

```
select
    A.name as Name,
    B.price as Price
from pizza_types as A join pizzas as B
on A.pizza_type_id = B.pizza_type_id
order by B.price desc
limit 1;
```



Data Analysis 1.4

Identify the most common pizza size ordered.

```
select count(A.order_details_id) as Order_count, B.size as size  
from order_details as A join pizzas as B  
on A.pizza_id = B.pizza_id  
group by B.size  
order by count(A.order_details_id) desc  
LIMIT 1;
```

20000

15000

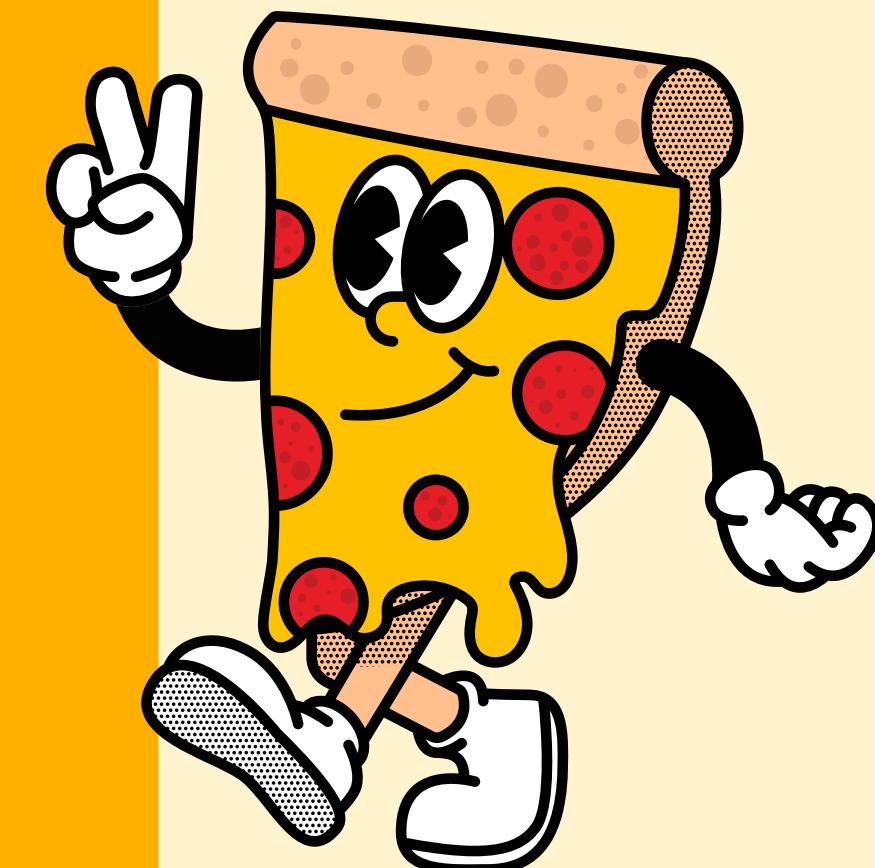
10000

5000

0

18526

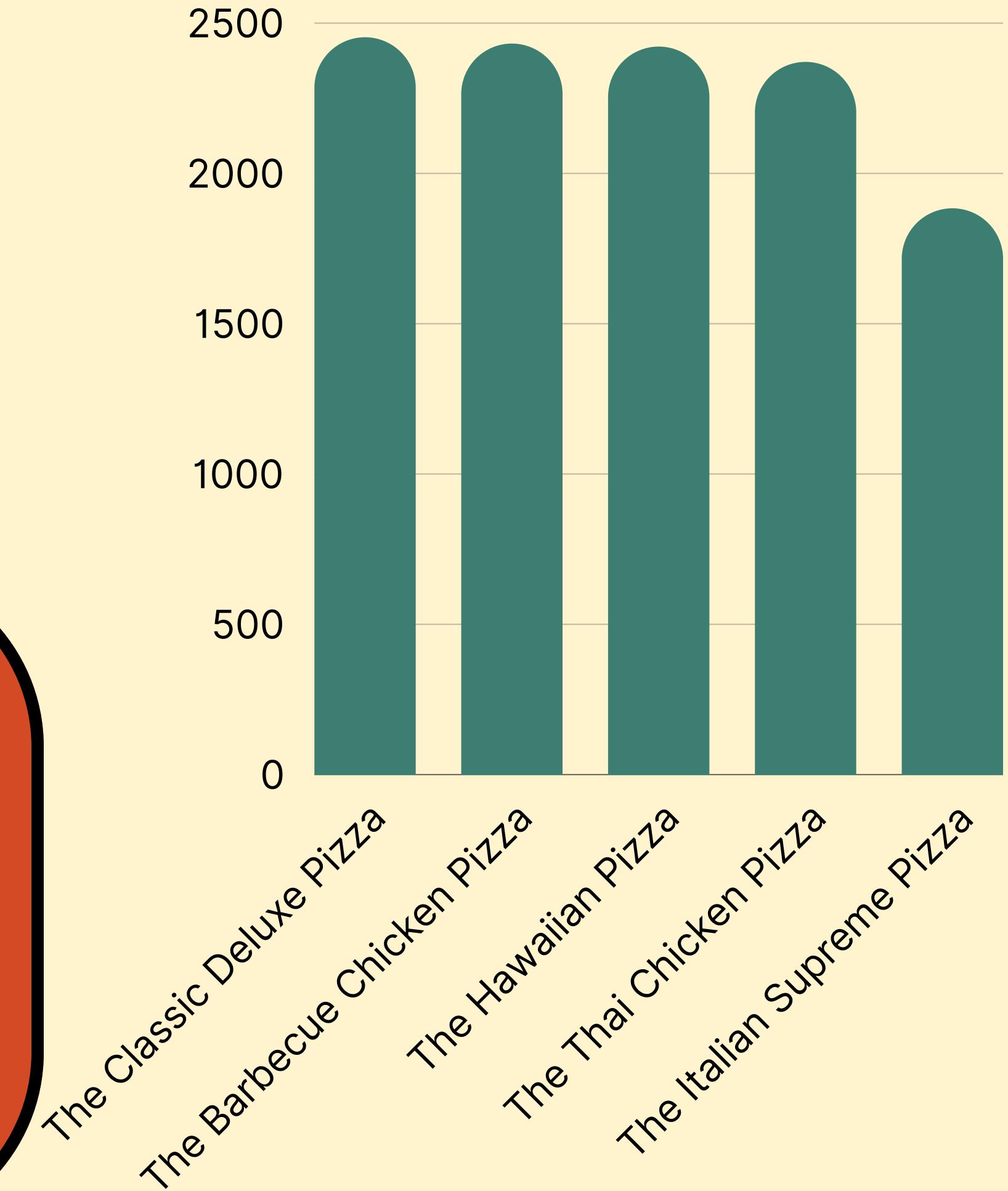
Size L



Data Analysis 1.5

Top 5 most ordered pizza types with quantities.

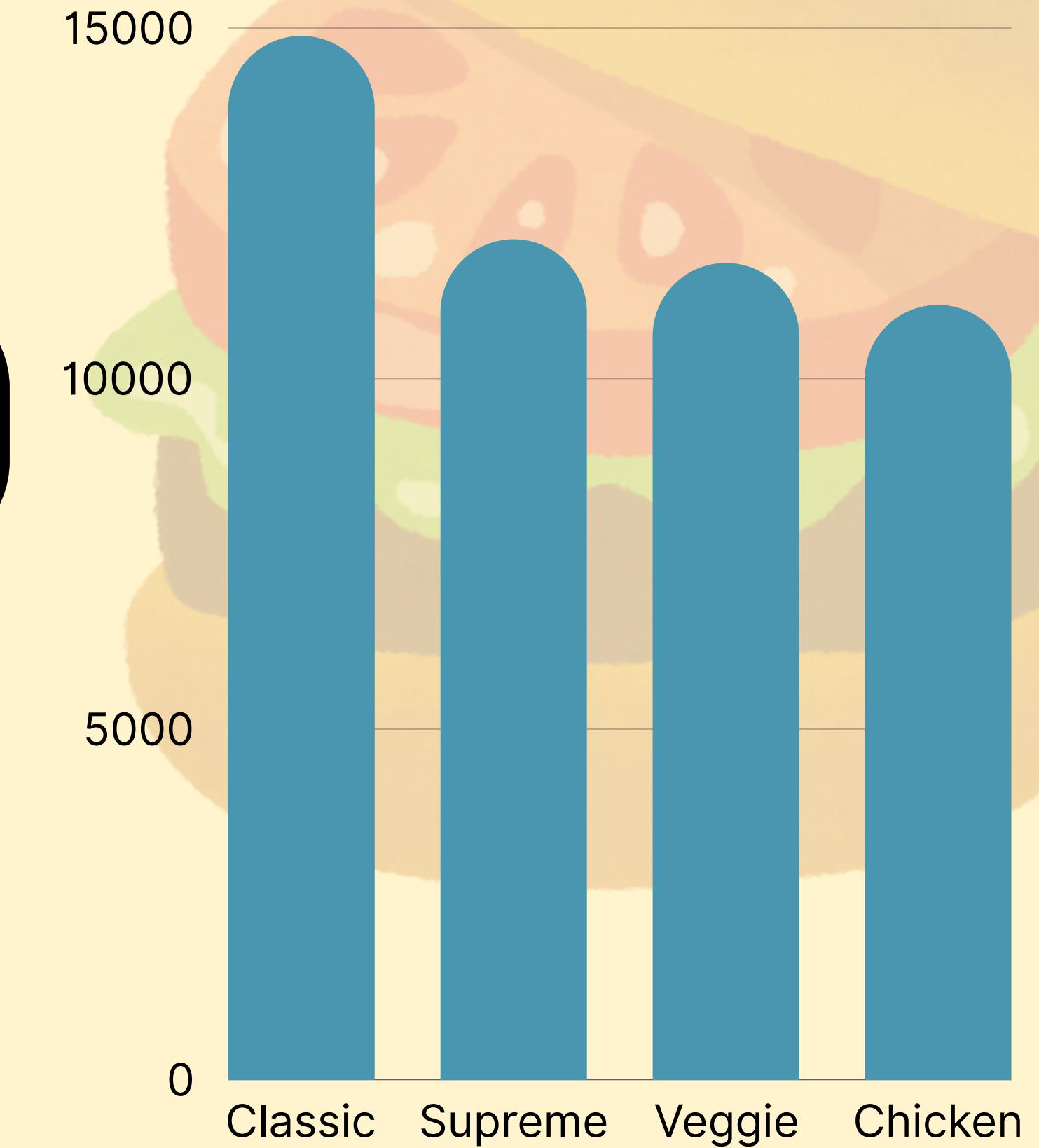
```
SELECT sum(A.quantity) AS Quantity,  
       C.name AS Name  
  FROM order_details AS A JOIN pizzas AS B  
    ON A.pizza_id = B.pizza_id  
   JOIN pizza_types AS C  
    ON B.pizza_type_id = C.pizza_type_id  
 GROUP BY C.name ORDER BY A.quantity desc  
LIMIT 10;
```



Data Analysis 2.1

The total quantity of each pizza category ordered.

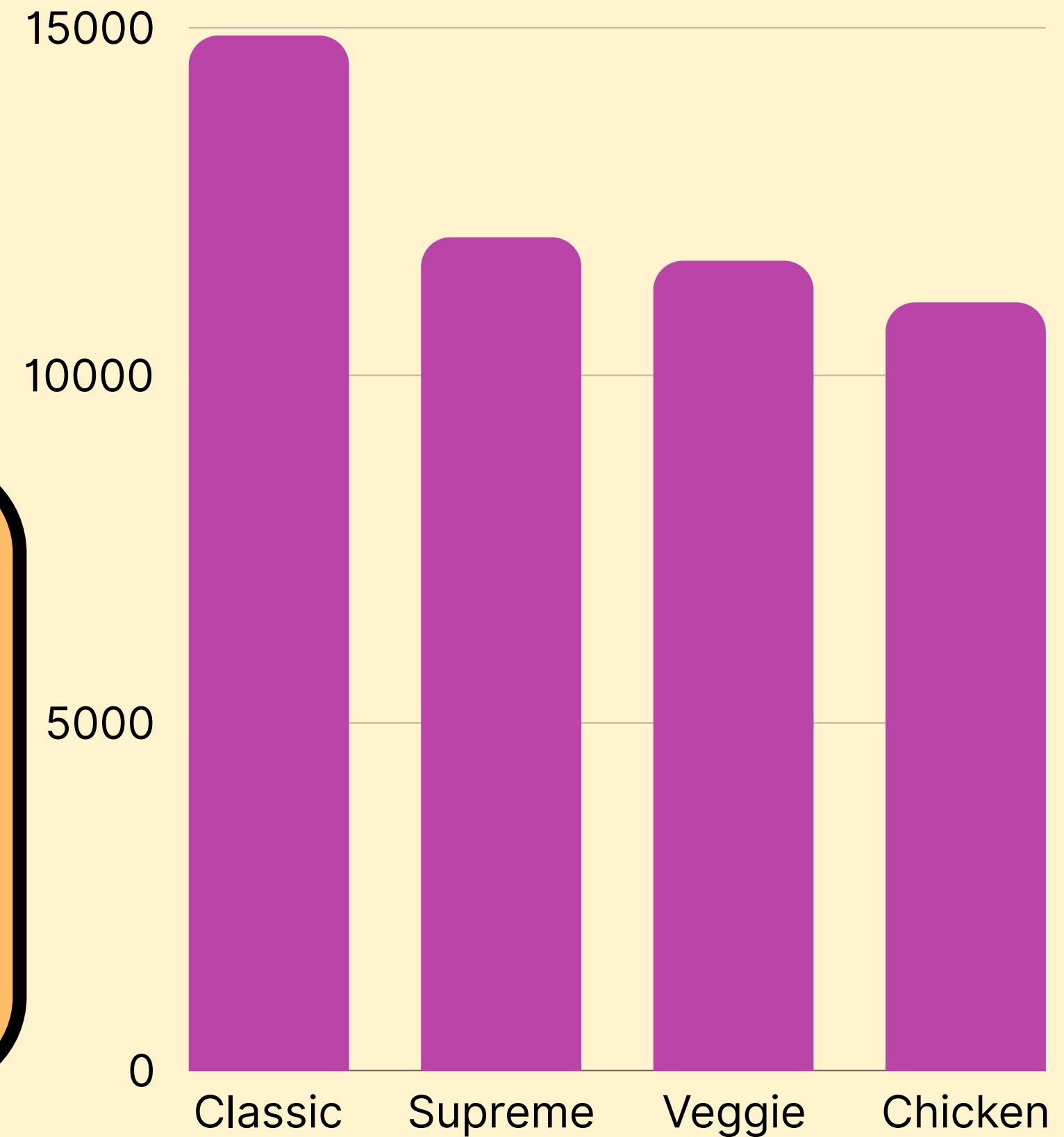
```
SELECT sum(A.quantity) Quantity,
      C.category Category
  FROM order_details AS A JOIN pizzas AS B
  ON A.pizza_id = B.pizza_id
  JOIN pizza_types AS C
  ON B.pizza_type_id = C.pizza_type_id
 GROUP BY C.category
ORDER BY A.quantity desc;
```



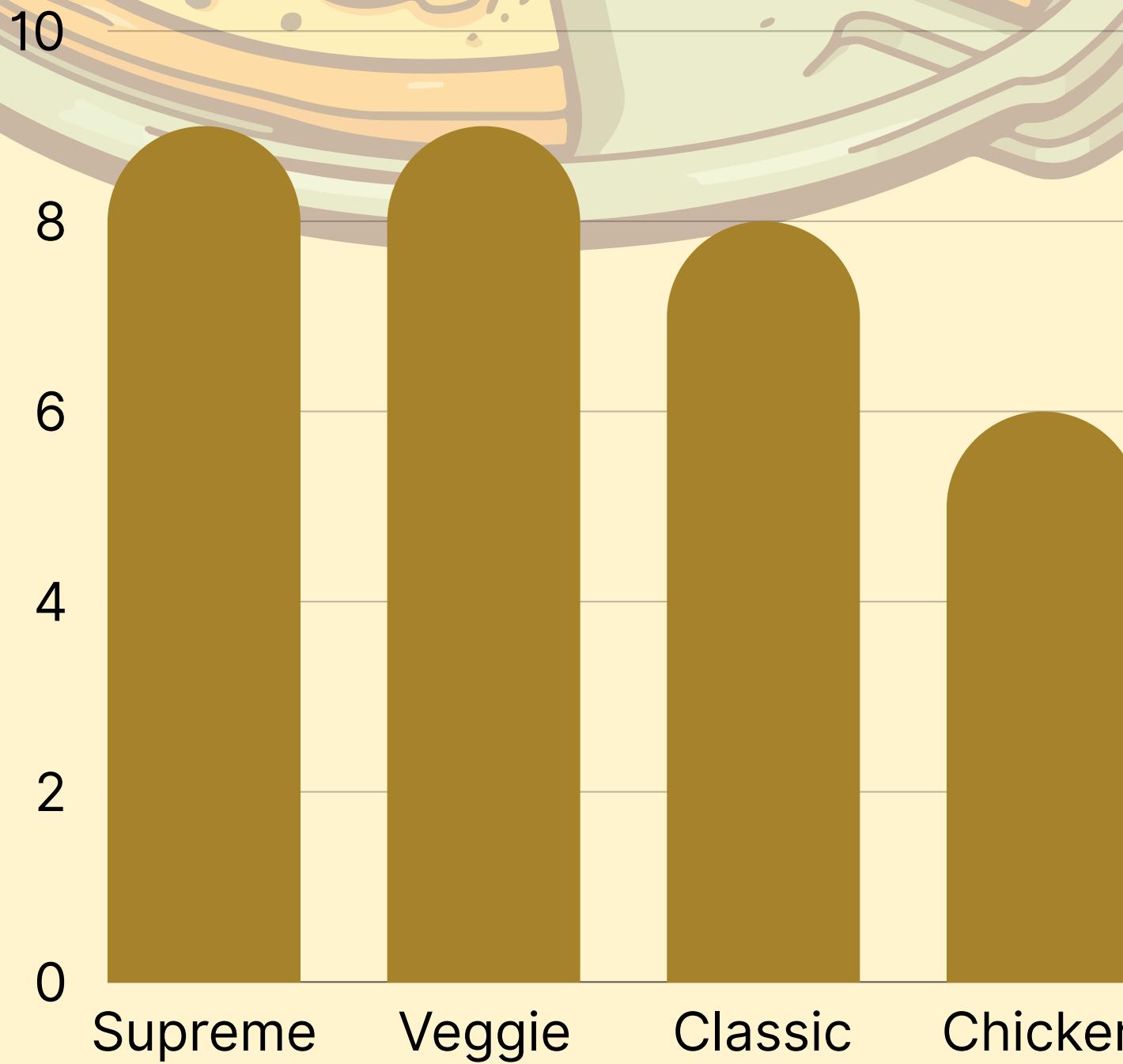
Data Analysis 2.2

The distribution of orders by hour of the day.

```
SELECT
    hour(order_time) Hours,
    count(order_id) Total_Orders
FROM orders
GROUP BY hour(order_time)
ORDER BY count(order_id) DESC;
```



Data Analysis 2.3



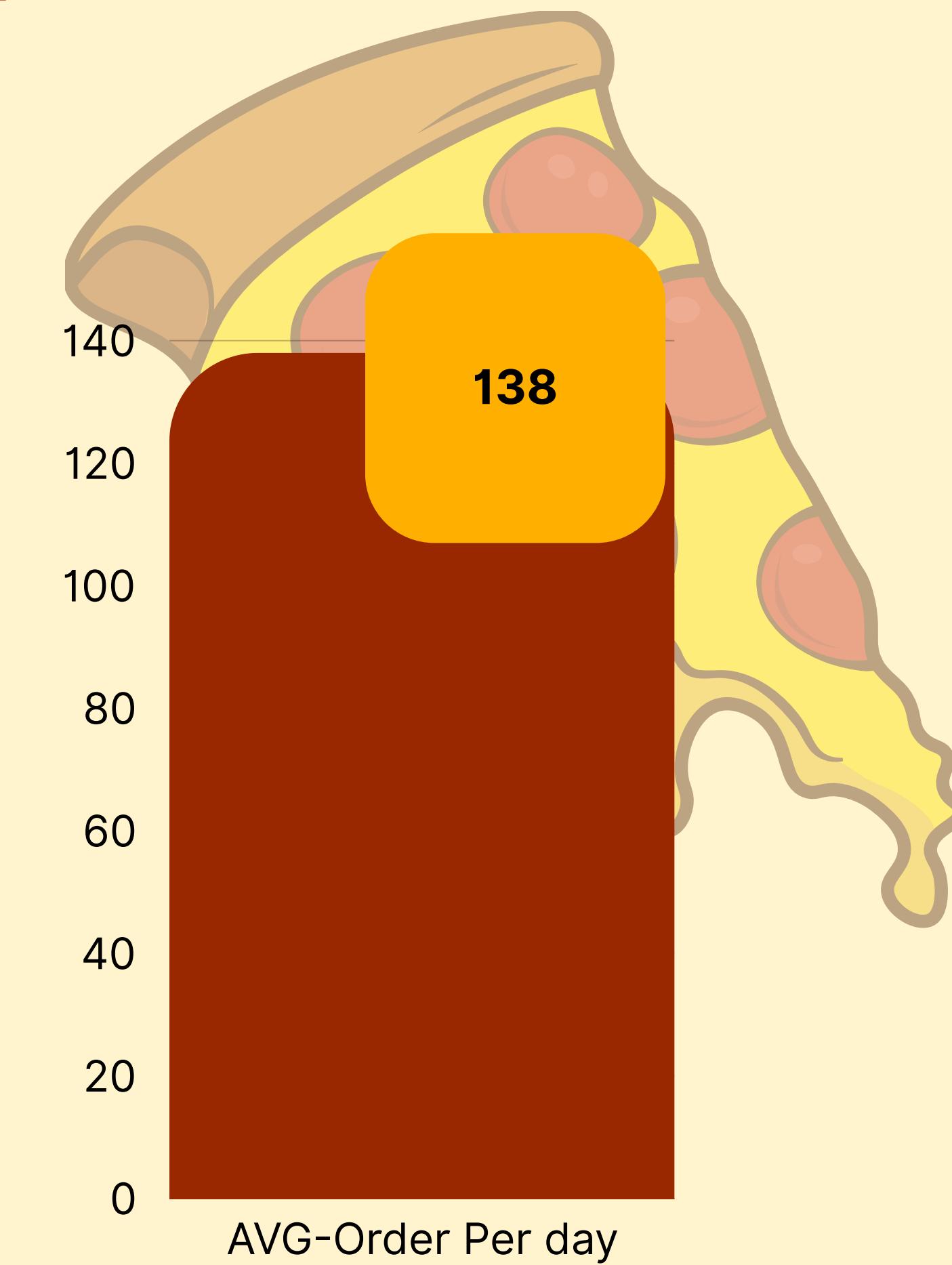
The Category-wise distribution of pizzas.

```
SELECT
    category AS Category_Of_Pizzas,
    count(name) as TOTAL_RECIPIE
FROM pizza_types
group by category
ORDER BY count(name) DESC;
SELECT * FROM pizza_types;
```

Data Analysis 2.4

The average number of pizzas ordered per day.

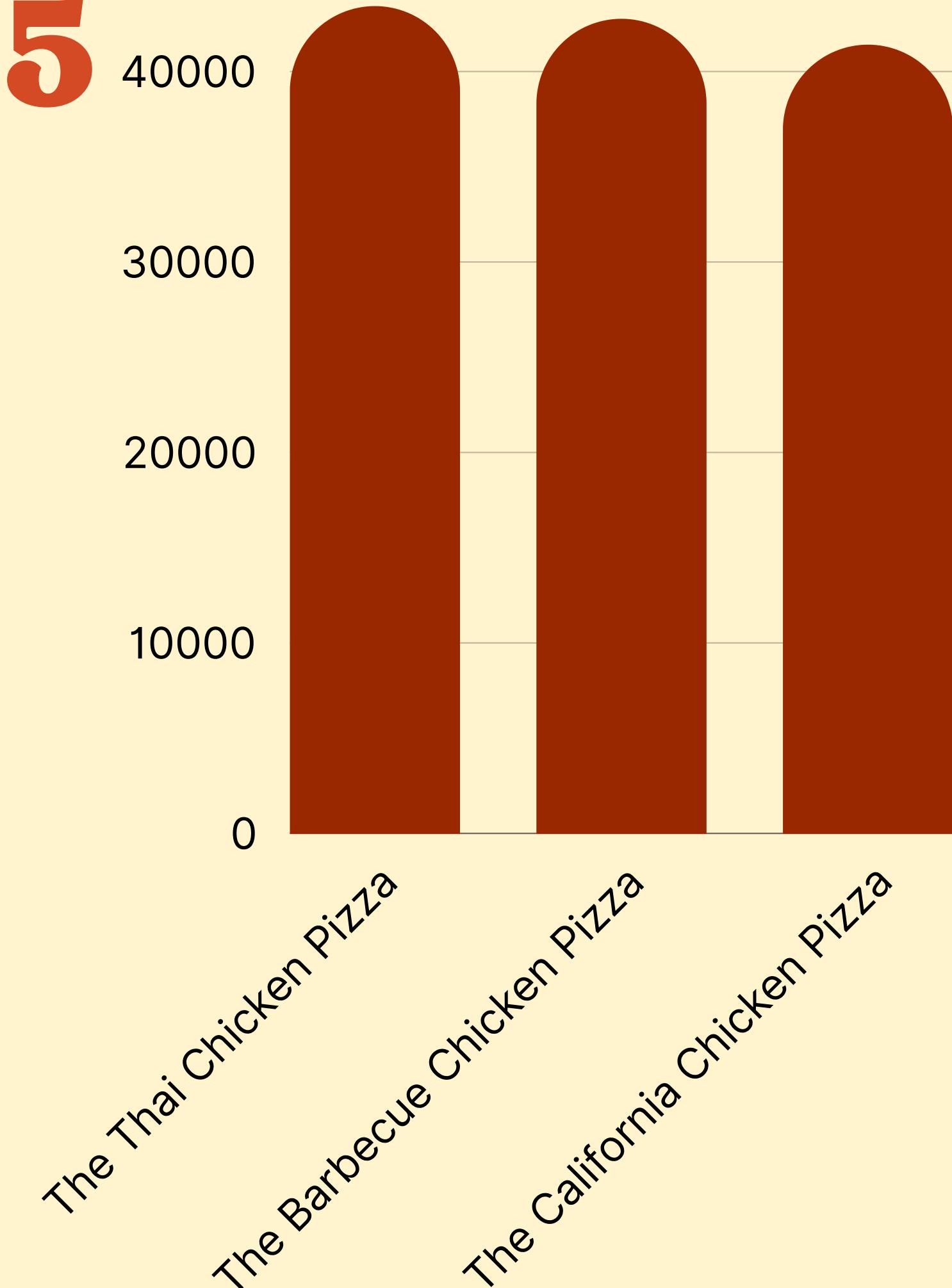
```
SELECT ROUND(AVG(Quantity)) AS  
AVG_per_day_sale  
FROM (  
    SELECT  
        A.order_date, SUM(B.quantity) as Quantity  
    FROM orders A JOIN order_details B  
    ON A.order_id = B.order_id  
    GROUP BY A.order_date  
    ORDER BY B.quantity) AS order_each_day;
```



Data Analysis 2.5

Top 3 most ordered pizza types based on revenue.

```
SELECT
    B.name AS POPULAR_PIZZA,
    SUM(A.PRICE * C.quantity) AS REVENUE
FROM pizzas AS A JOIN pizza_types AS B
ON A.pizza_type_id = B.pizza_type_id
JOIN order_details AS C
ON A.pizza_id = C.pizza_id
GROUP BY B.name
ORDER BY SUM(A.PRICE) DESC
LIMIT 3;
```



Data Analysis 3.1

Percentage contribution of each pizza type to total revenue.



```
SELECT B.category AS PIZZAS_category,  
ROUND(  
  ( SUM(A.PRICE * C.quantity) /  
    ( SELECT  
      SUM(A.PRICE * C.quantity)  
      FROM pizzas AS A JOIN order_details AS C  
      ON A.pizza_id = C.pizza_id  
    )  
  ) *100,2 ) AS REVENUE_IN_PERCENTAGE  
FROM pizzas AS A JOIN pizza_types AS B  
ON A.pizza_type_id = B.pizza_type_id  
JOIN order_details AS C  
ON A.pizza_id = C.pizza_id  
GROUP BY B.category  
ORDER BY SUM(A.PRICE) DESC;
```

Data Analysis 3.2

Analyze the cumulative revenue generated over time..

Date	Revenue	Cumulative Revenue
2015-01-01	2714	2714
2015-01-02	2732	5446
2015-01-03	2662	8108
2015-01-04	1755	9864
2015-01-05	2066	11930
2015-01-06	2429	14358
2015-01-07	2202	16561

```
SELECT  
Order_Date,  
round(REVENUE) Revenue,  
ROUND( SUM(REVENUE) OVER(ORDER BY  
ORDER_DATE) ) AS Cumulative_Revenue  
FROM (  
SELECT A.order_date,  
(SUM(B.quantity * C.price)) AS REVENUE  
FROM orders A JOIN order_details B  
ON A.order_id = B.order_id  
JOIN pizzas C  
ON B.pizza_id = C.pizza_id  
GROUP BY A.order_date  
) AS PER_DAY_REVENEU;
```

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

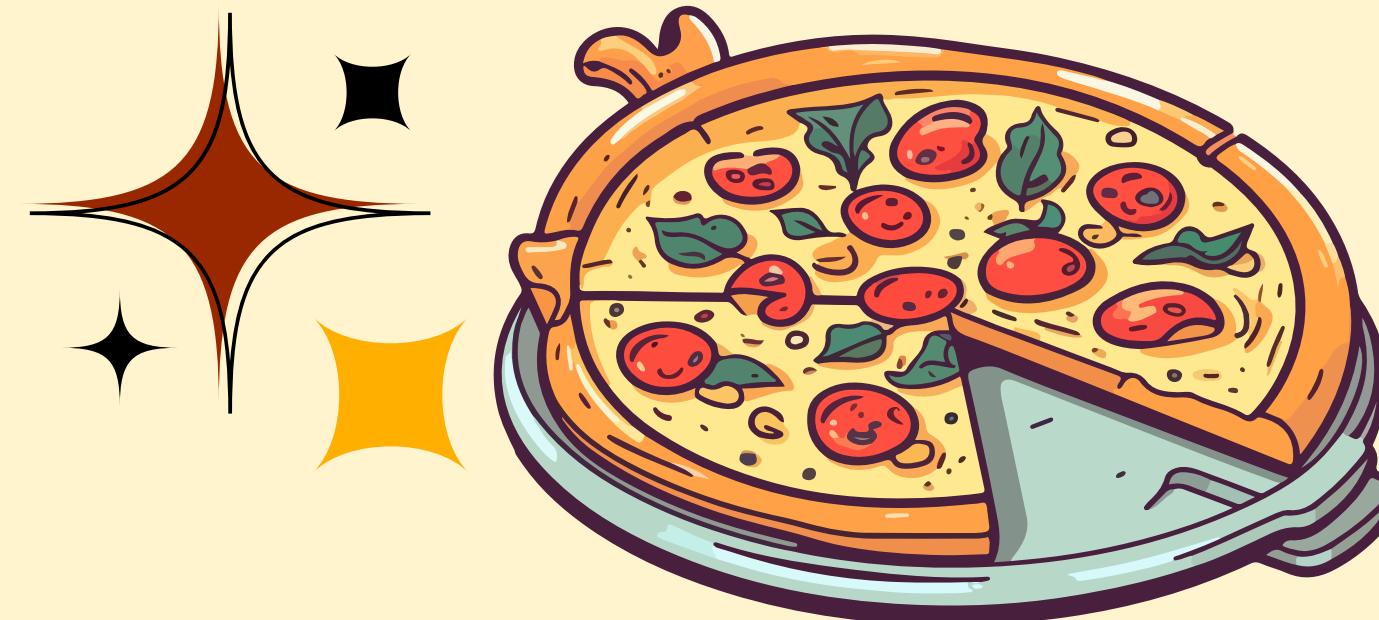
Rank	Category	Name	Revenue
1	Chicken	The Thai Chicken Pizza	43434
2		The Barbecue Chicken Pizza	42768
3		The California Chicken Pizza	41410
1	Classic	The Classic Deluxe Pizza	38180
2		The Hawaiian Pizza	32273
3		The Pepperoni Pizza	30162
1	Supreme	The Spicy Italian Pizza	34831
2		The Italian Supreme Pizza	33477
3		The Sicilian Pizza	30940
1	Veggie	The Four Cheese Pizza	32266
2	Veggie	The Mexicana Pizza	26781
3	Veggie	The Five Cheese Pizza	26066

Data Analysis 3.3

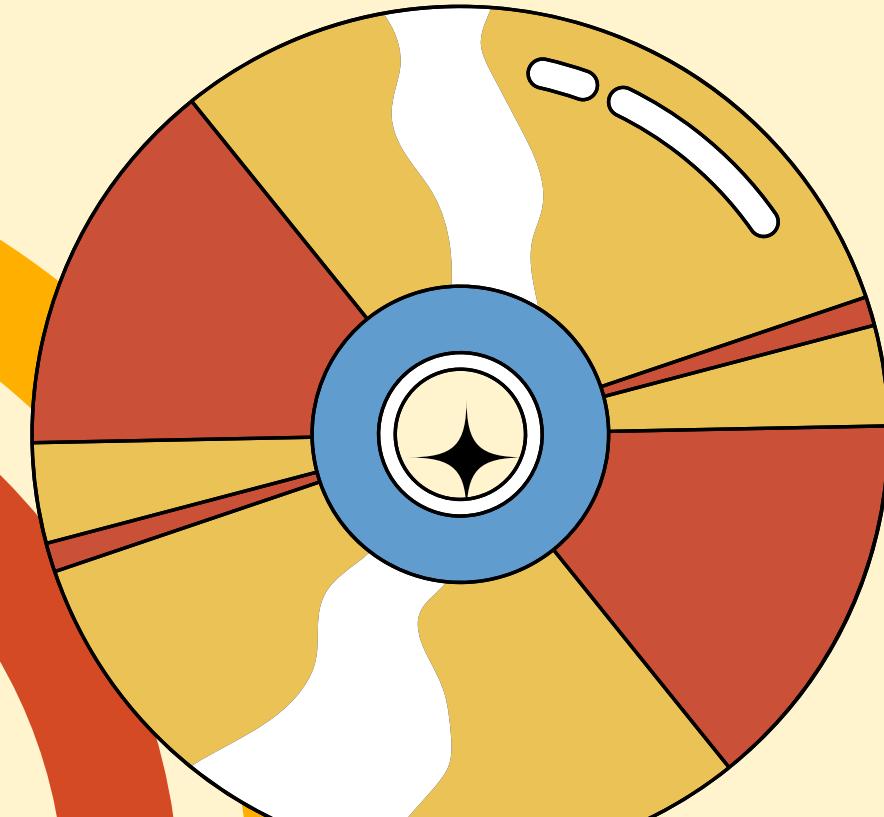
```
SELECT Category, Name, Revenue, SERIAL_NO
FROM (
    SELECT Category, Name, Revenue,
    DENSE_RANK() OVER(PARTITION BY category
    ORDER BY revenue DESC) AS SERIAL_NO
    FROM (
        SELECT C.category, C.name,
        ROUND(SUM(A.quantity * B.price)) AS REVENUE
        FROM order_details A JOIN pizzas B
        ON A.pizza_id = B.pizza_id
        JOIN pizza_types C
        ON B.pizza_type_id = C.pizza_type_id
        group by C.category, C.name
        ORDER BY C.category
    ) AS TABLE_A
) AS TABLE_B
WHERE SERIAL_NO < 4;
```

Objectives





Project Objectives



Analysis Phase

- Conduct a comprehensive analysis of current sales performance, market trends, and consumer behavior.

Strategy Development

- Brainstorm and devise new strategies to leverage strengths and opportunities.
- Develop a multifaceted approach integrating marketing, sales promotions, and customer engagement.

Implementation Plan

- Determine key performance indicators (KPIs) to measure the success of the new strategies.

Expected Outcomes

Improved Sales Insights

- Pinpoint top-performing pizzas by revenue and order frequency.
- Analyze trends in pizza sizes, toppings, and order times to tailor offerings.

Enhanced Operational Efficiency

- Forecast demand for ingredients based on sales patterns to reduce waste.
- Identify peak order hours to improve staffing and delivery times.

Data-Driven Decision Making

- Use sales data to design promotions for underperforming products.
- Adjust the menu by focusing on high-revenue and popular items while phasing out low-performing options.

Conclusion

This project delivers actionable insights into sales performance, customer preferences, and operational efficiency. By leveraging SQL for data analysis, it empowers the business to make informed decisions, optimize inventory, and enhance marketing strategies, ultimately driving growth and improving customer satisfaction in a competitive market.

Thank You!

