

A Slice of SQL

Pizza Sales Data Analysis

By : Saisha Raut



Project Summary

01

OBJECTIVE

To analyze pizza sales data to :

- provide actionable insights on sales performance,
- customer preferences, and revenue generation.

02

FEATURES OF THE TOPIC

- Sub Query
- Group By
- Cumulative Sum, Rank
- Visualizations

03

ABOUT THE DATASET

- Analyzing sales records from a pizza restaurant, spanning one year.
- Dataset has 48,620 sales transactions to analyze.

04

ABOUT THE TOPIC

- **MySQL** for data querying and manipulation.
- **TableauPublic** for data visualization.

INTRODUCTION

This project analyzes pizza sales data to uncover valuable business insights, including customer ordering patterns, popular pizza types, revenue distribution, and time-based sales trends. This analysis highlights the top contributors to revenue, peak order hours, and category-wise performance. The study also tracks cumulative revenue growth over time, providing a clear view of business trends. Visualizations in Tableau and data analysis in MySQL showcase how data-driven decision-making can optimize product strategy and improve overall sales outcomes in the food industry.

DATA SCHEMA

orders	orders_details	pizza_types	pizzas
order_id <small>PK int</small>	order_details_id <small>PK int</small>	pizza_type_id <small>PK text</small>	pizza_id <small>PK text</small>
order_date <small>date</small>	order_id <small>int</small>	name <small>text</small>	pizza_type_id <small>text</small>
order_time <small>time</small>	pizza_id <small>text</small>	category <small>text</small>	size <small>text</small>

BASIC QUESTIONS

01

Retrieve the total number of orders placed

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

total_orders
21350

02

Calculate the total revenue generated from pizza sales

SELECT

```
ROUND(SUM(orders_details.quantity * pizzas.price),  
2) AS total_sales
```

FROM

orders_details

JOIN

```
pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

total_sales

817860.05

03

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	price
The Greek Pizza	35.95

04

Identify the most common pizza size ordered

```
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS orders_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY orders_count DESC
LIMIT 1;
```

size	orders_count
L	18526

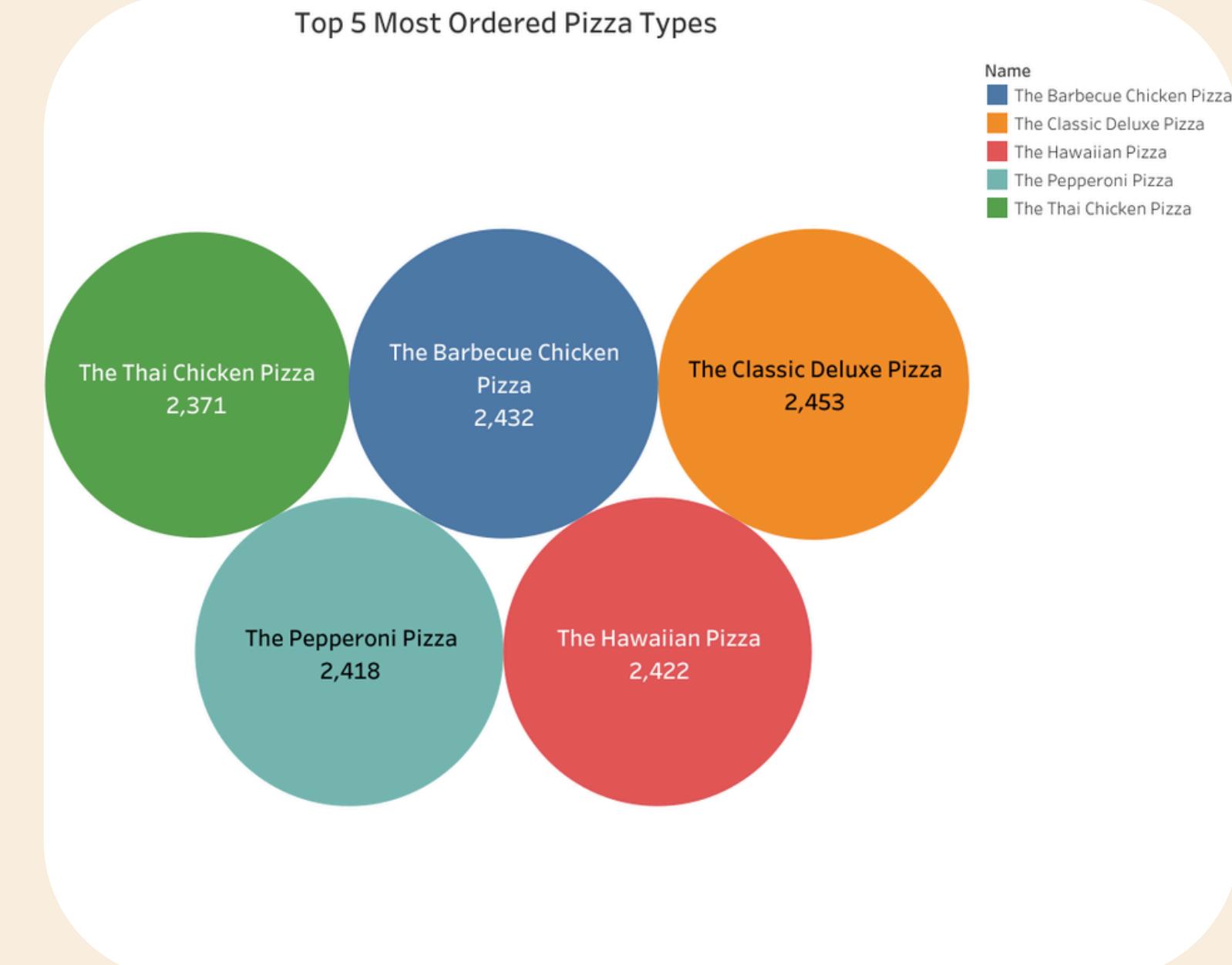
05

List the top 5 most ordered pizza types along with their quantities

```
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS quantity
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.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

Top 5 Most Ordered Pizza Types



INTERMEDIATE QUESTIONS

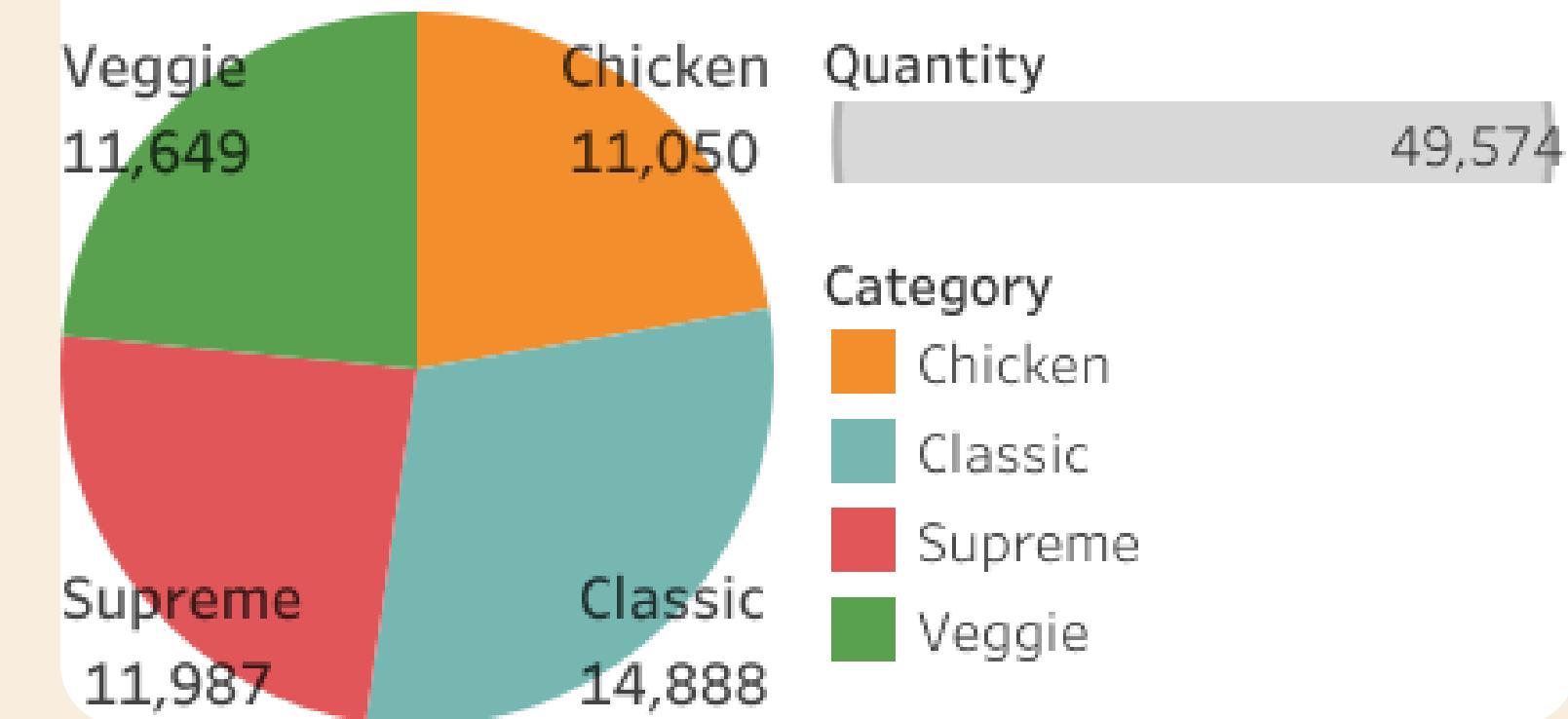
06

Join the necessary tables to find the total quantity of each pizza category ordered

```
SELECT  
    pizza_types.category,  
    SUM(orders_details.quantity) AS quantity  
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  
ORDER BY quantity DESC;
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Total Quantity of
Each Pizza Category
Ordered



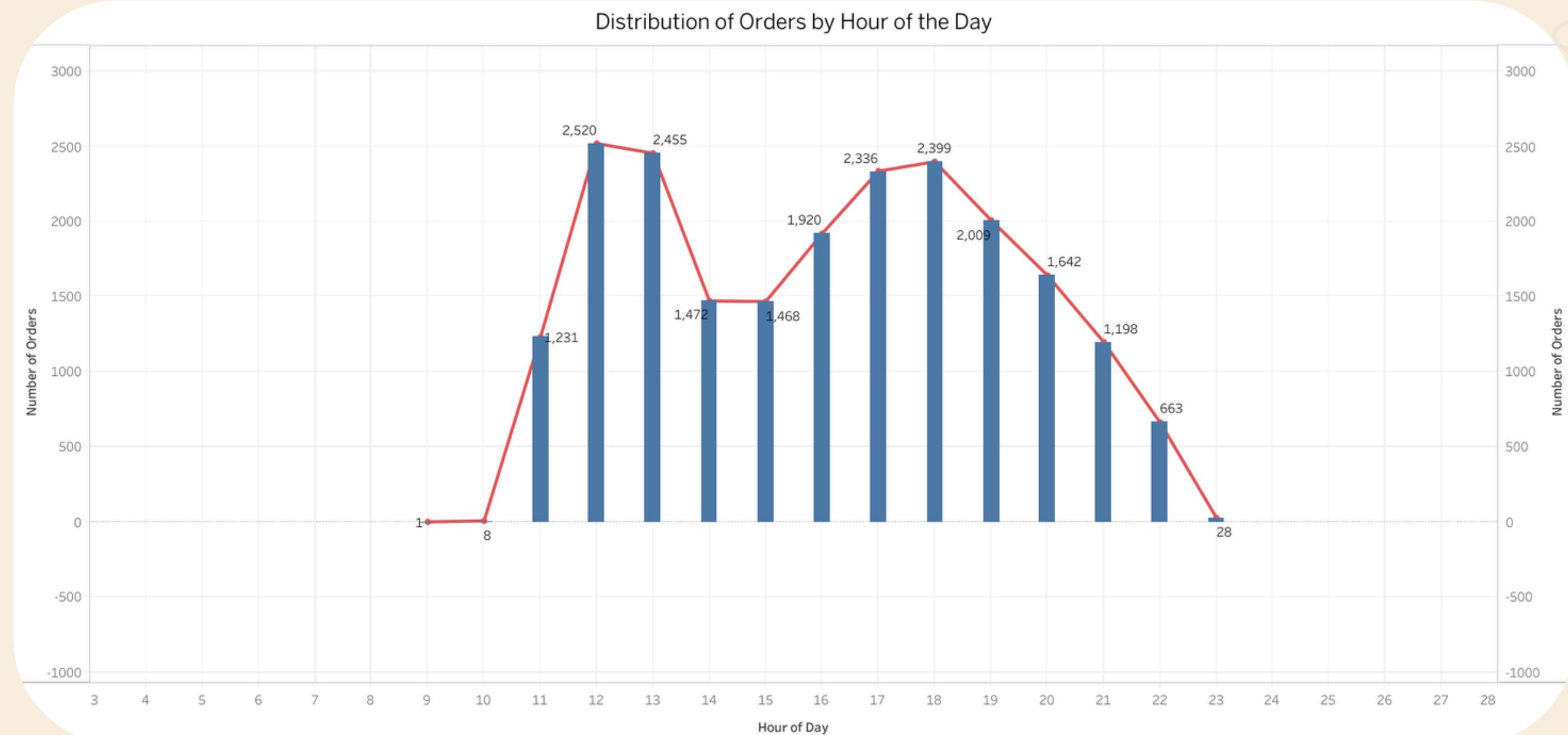
07

Determine the distribution of orders by hour of the day

O

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

hour	order_count
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



08

Join relevant tables 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

09

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

```
SELECT  
    ROUND(AVG(quantity), 0) as avg_pizzas_ordered_per_day  
FROM  
    (SELECT  
        orders.order_date, SUM(orders_details.quantity) AS quantity  
    FROM  
        orders  
    JOIN orders_details ON orders.order_id = orders_details.order_id  
    GROUP BY orders.order_date) AS orders_quantity;
```

avg_pizzas_ordered_per_day

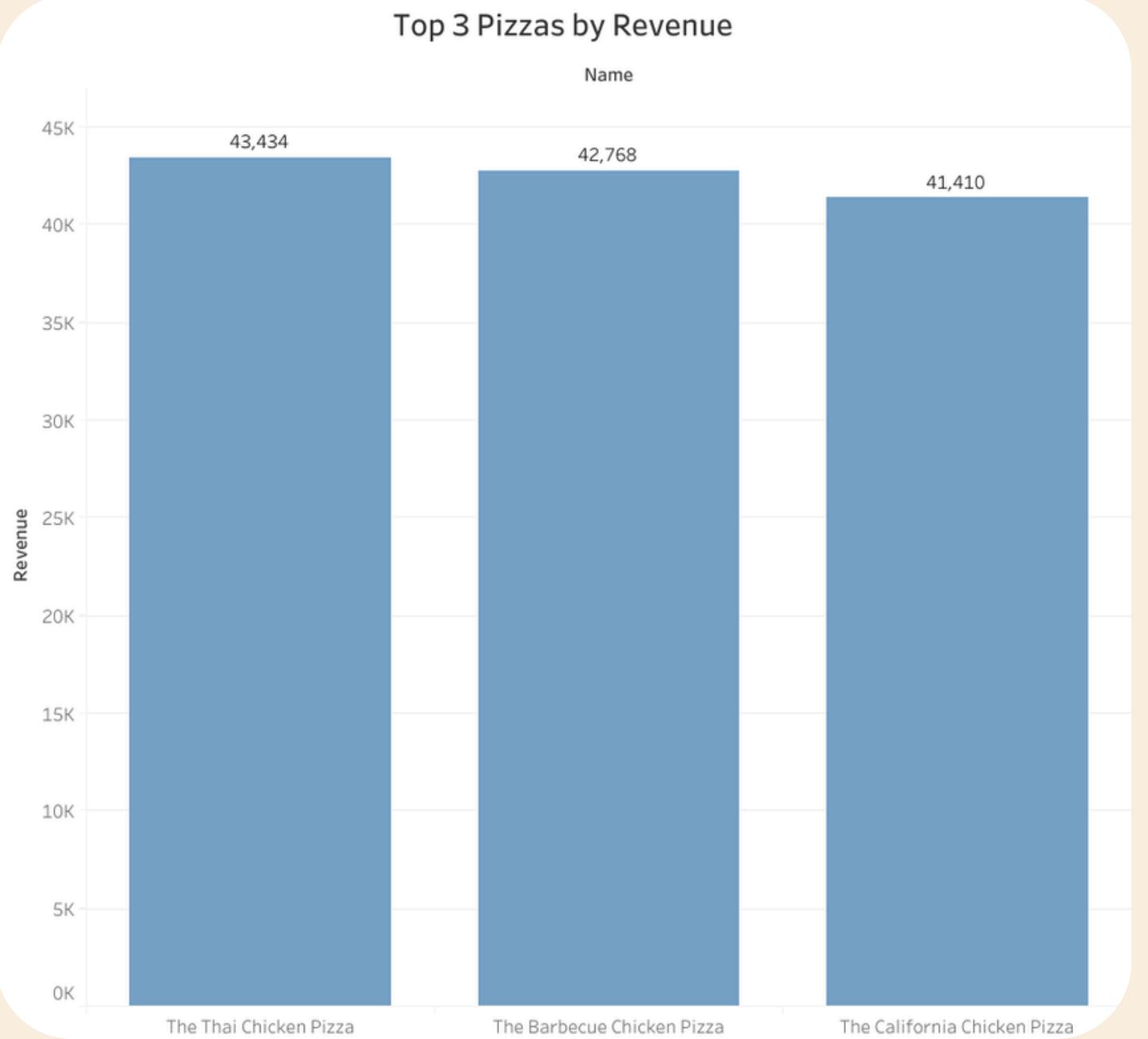
138

10

Determine the top 3 most ordered pizza types based on revenue

```
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_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



ADVANCED QUESTIONS

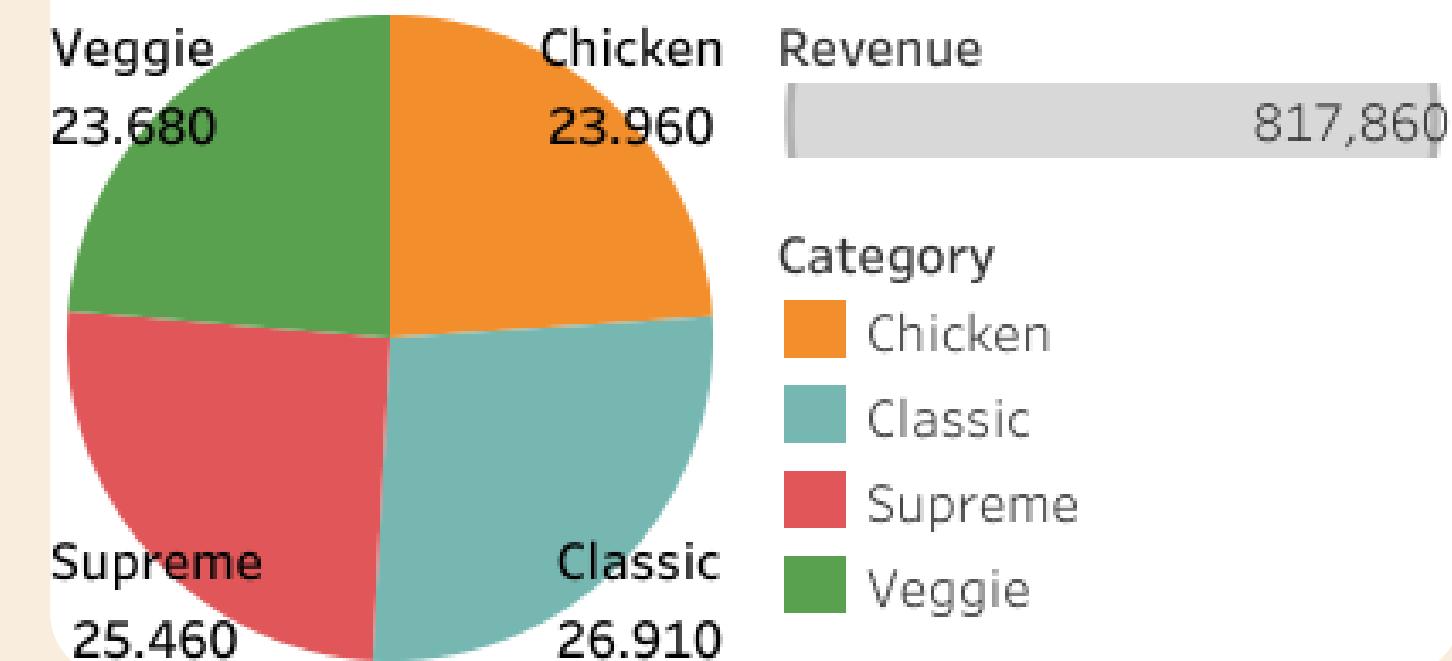
11

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

```
SELECT
    pizza_types.category,
    ROUND(SUM(orders_details.quantity * pizzas.price) / (SELECT
        SUM(orders_details.quantity * pizzas.price)
    FROM
        orders_details
    JOIN
        pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,
    2) AS revenue_percentage
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
ORDER BY revenue_percentage DESC;
```

category	revenue_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Percentage
Contribution of Each
Pizza Category to
Total Revenue

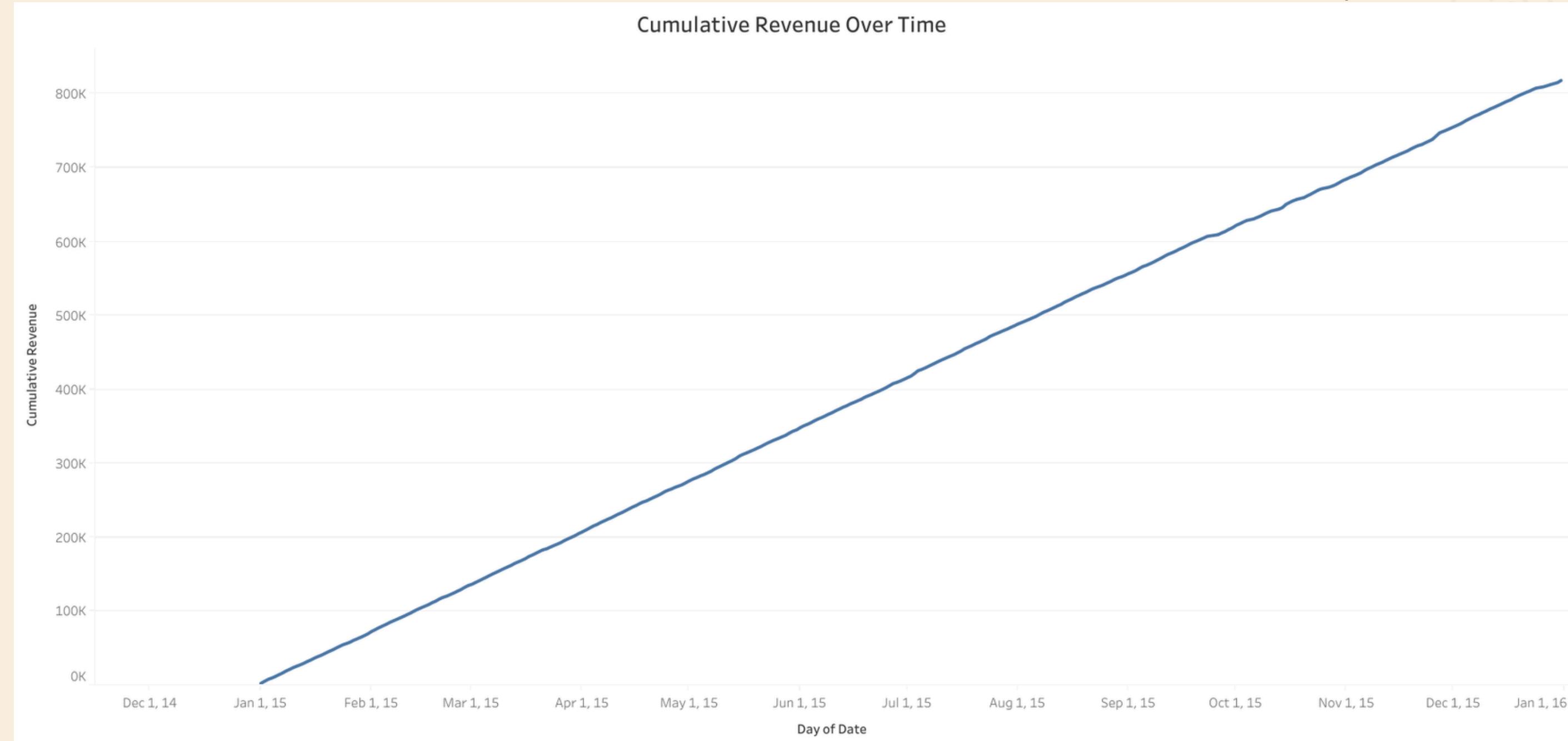


12

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(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_revenue
2015-01-01	2713.850000000004
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	23990.35000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.30000000003
2015-01-14	32358.70000000004



13

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(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;
```

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
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
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5



THANKS!

saisharaut9@gmail.com
+91 9860515990
github.com/saisharaut99

