PIZZA $ALES ANALYSIS

USING SQL

By: Daipayan Maitra

TABLE OF CONTENTS

01

INTRODUCTION

Dataset description and business problems

02

SALES PERFORMANCE

Analysis of total orders, revenue, and average order value

03

MENU-ITEM PERFORMANCE

Examination of top-selling and worst-selling pizzas

04

ORDERING PATTERNS & TRENDS

Analysis of hourly and daily ordering trends

05

CUSTOMER BEHAVIOUR AND PREFERENCES

Insights into pizza category distribution and customer preferences

06

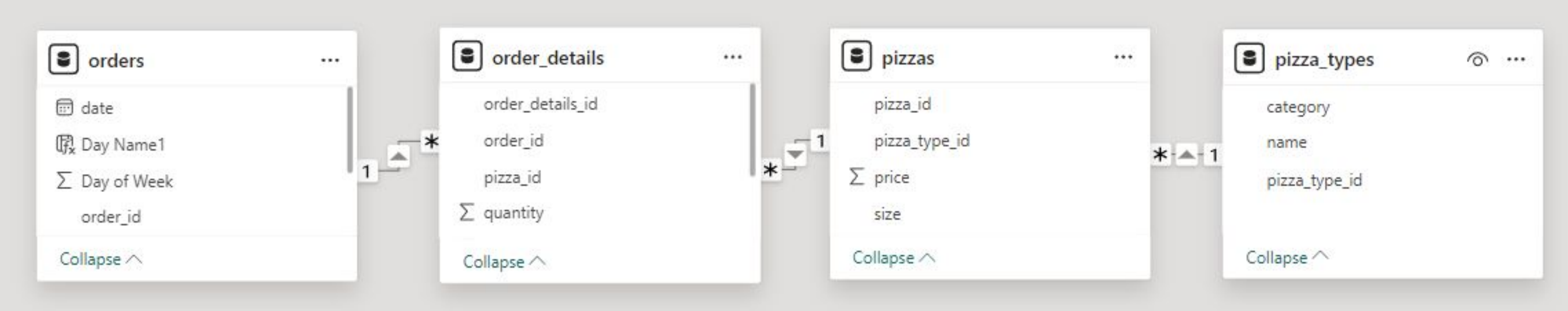
CONCLUSION & RECOMMENDATIONS

Summary of key findings and recommendations for business growth

01

INTRODUCTION

DATASET DESCRIPTION



orders
- date
- Day Name1
- Day of Week
- order_id

Collapse

order_details
- order_details_id
- order_id
- pizza_id
- quantity

Collapse

pizzas
- pizza_id
- pizza_type_id
- price
- size

Collapse

pizza_types
- category
- name
- pizza_type_id

Collapse

We have 4 tables: **orders, order_details, pizzas and pizza_types**

BUSINESS OBJECTIVE

Plato's Pizza Restaurant aims to improve sales performance, customer satisfaction, and operational efficiency by analyzing sales data and identifying areas for improvement.

- Identify **peak ordering periods** and optimize staffing and inventory accordingly
- Determine **top-selling menu items** and opportunities for growth
- Understand **customer ordering trends** and preferences
- Enhance **customer experience** and drive business growth

BUSINESS QUESTIONS

- The total number of orders placed
- The total revenue generated from pizza sales
- The highest priced pizza.
- The most common pizza size ordered.
- The top 5 most ordered pizza types along their quantities.
- The quantity of each pizza categories ordered.
- The distribution of orders by hours of the day.
- The category-wise distribution of pizzas.
- The average number of pizzas ordered per day.
- Top 3 most ordered pizza type base on revenue.
- The percentage contribution of each pizza type to revenue.
- The cumulative revenue generated over time.
- The top 3 most ordered pizza type based on revenue for each pizza category.

02

SALES PERFORMANCE

THE BIG PICTURE

48,620

Total Orders

```
SELECT COUNT(DISTINCT
order_id) AS
Total_orders_placed
FROM order_details;
```

$817,860

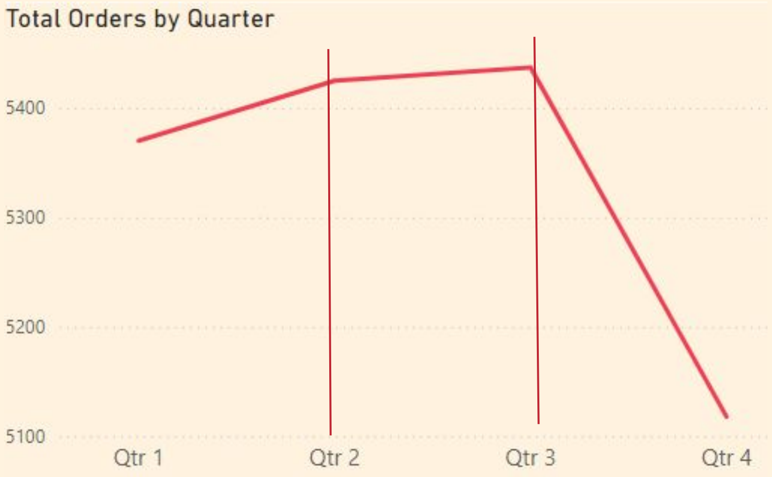
Total Revenue

```
SELECT ROUND(SUM(p.price *
o.quantity),0) AS
Total_revenue
FROM order_details o
JOIN pizzas p ON o.pizza_id =
p.pizza_id;
```

$17

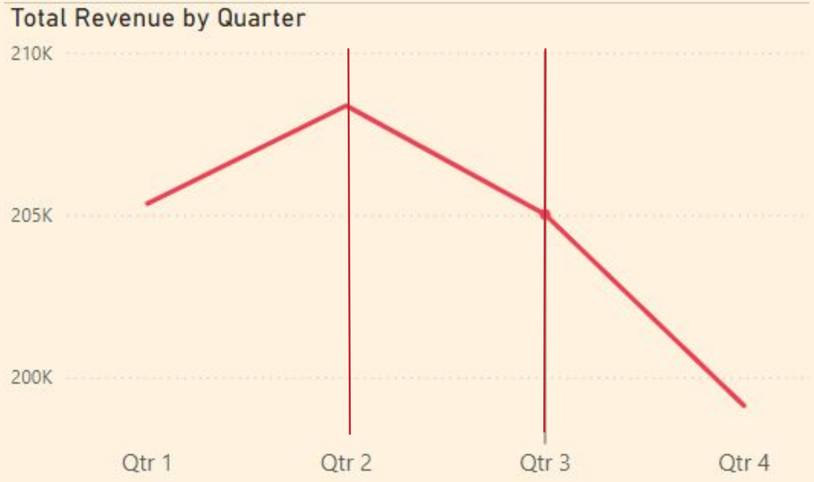
Average Order Value

MONTHLY ORDER TREND



In spite of same order values, revenue falls in second quarter

MONTHLY REVENUE TREND



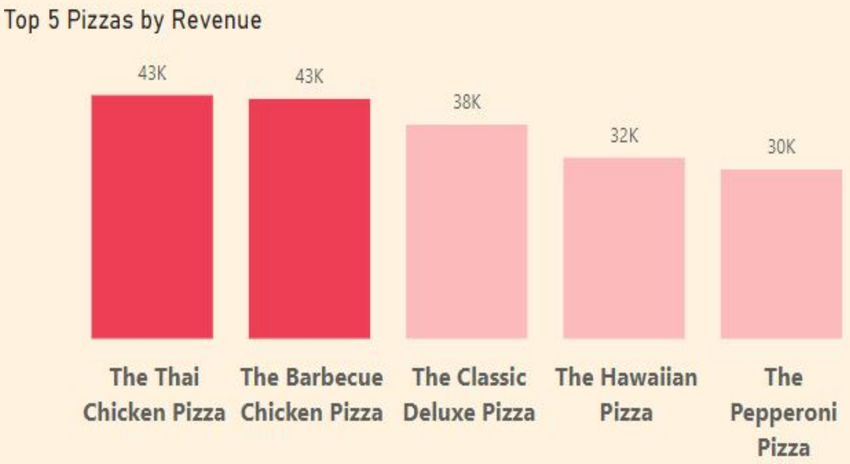
Revenue is higher in the first quarter of the year and lower in the subsequent quarters

Revenue is relatively flat suggests that the restaurant may be struggling to grow

03

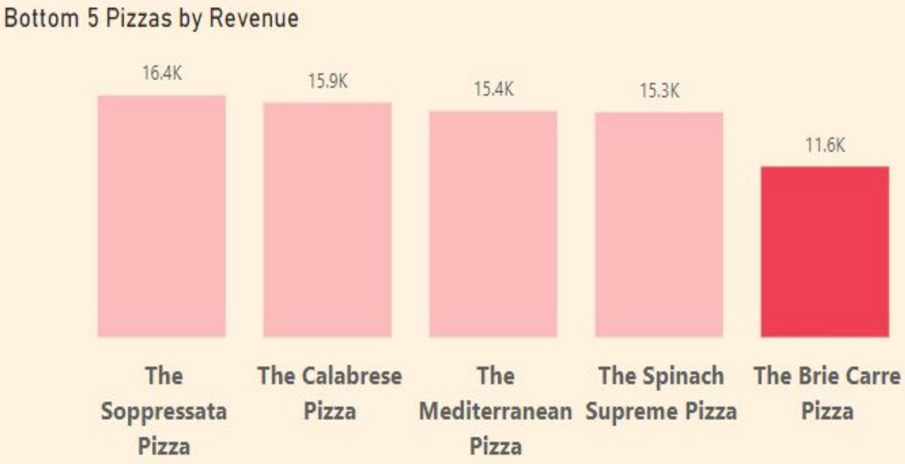
MENU-ITEM PERFORMANCE

BEST SELLING PIZZAS



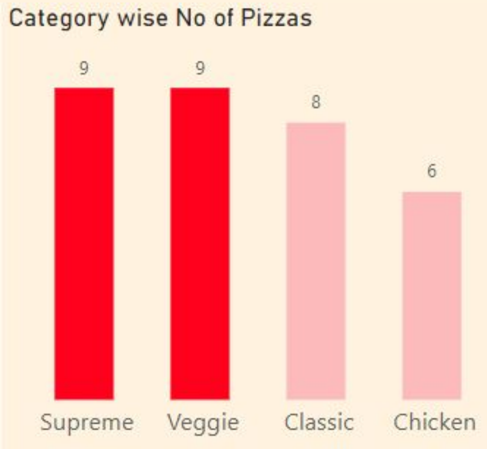
```
SELECT pt.name, SUM(p.price * o.quantity) AS
Total_revenue
FROM order_details o
JOIN pizzas p ON o.pizza_id = p.pizza_id
JOIN pizza_type pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.name
ORDER BY Total_revenue DESC
LIMIT 3;
```

WORST SELLING PIZZAS



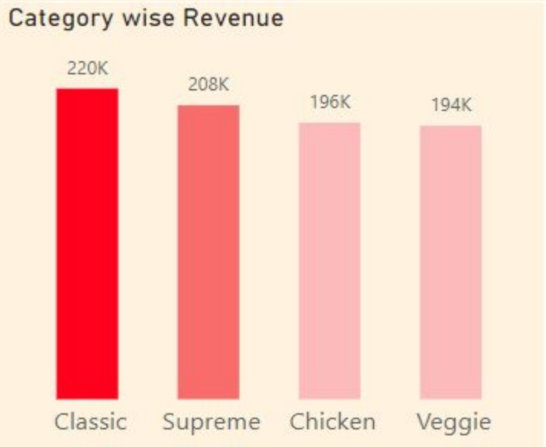
```
SELECT pt.name, SUM(p.price * o.quantity) AS
Total_revenue
FROM order_details o
JOIN pizzas p ON o.pizza_id = p.pizza_id
JOIN pizza_type pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.name
ORDER BY Total_revenue ASC
LIMIT 3;
```

CATEGORY WISE PIZZA DISTRIBUTION


Category wise No of Pizzas

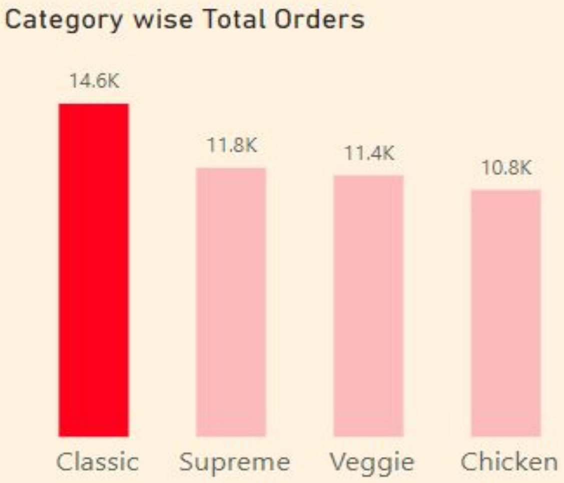
```sql
SELECT pt.category, COUNT(*) AS
pizza_count
FROM pizzas p
JOIN pizza_type pt ON
p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category
ORDER BY pizza_count DESC;
```

CATEGORY WISE REVENUE DISTRIBUTION


Category wise Revenue

```sql
SELECT pt.category, SUM(p.price * o.quantity)
AS Total_revenue
FROM order_details o
JOIN pizzas p ON o.pizza_id = p.pizza_id
JOIN pizza_type pt ON p.pizza_type_id =
pt.pizza_type_id
GROUP BY pt.category
ORDER BY Total_revenue DESC;
```

CATEGORY WISE ORDER DISTRIBUTION

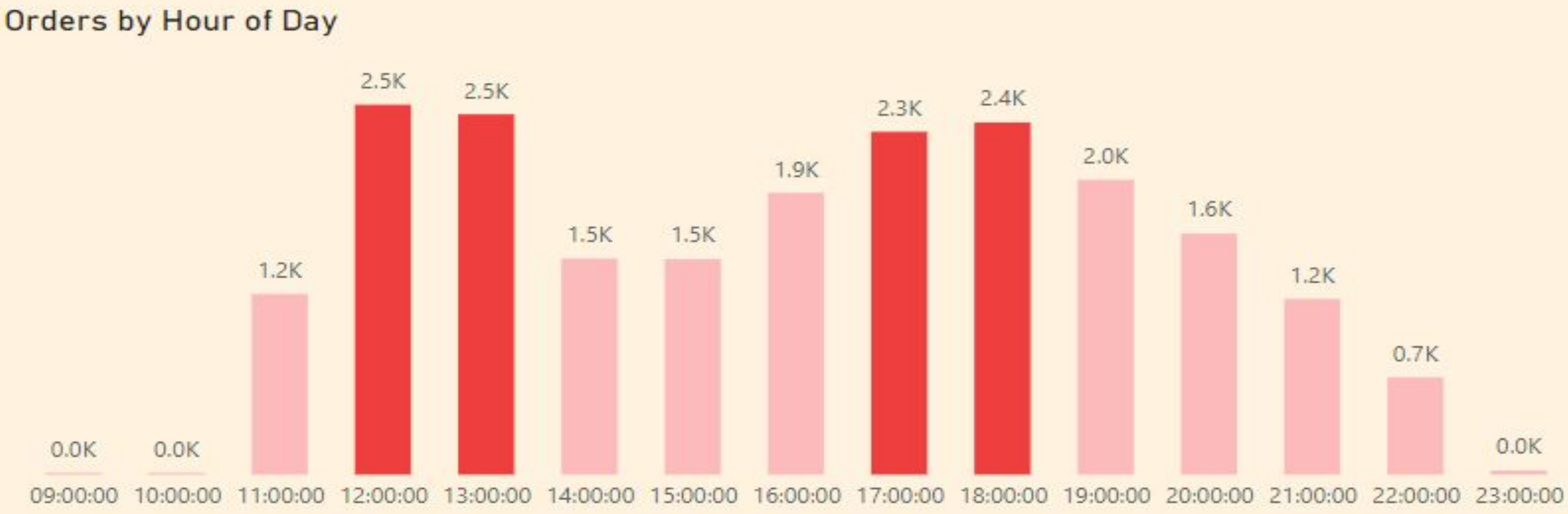

Category wise Total Orders

```sql
SELECT pt.category, Count(o.order_id) AS
total_orders
FROM pizza_type pt
JOIN pizzas p ON pt.pizza_type_id =
p.pizza_type_id
JOIN order_details o ON p.pizza_id = o.pizza_id
GROUP BY pt.category
ORDER BY total_orders DESC;
```

04

ORDERING PATTERNS AND TRENDS

HOURLY ORDERS

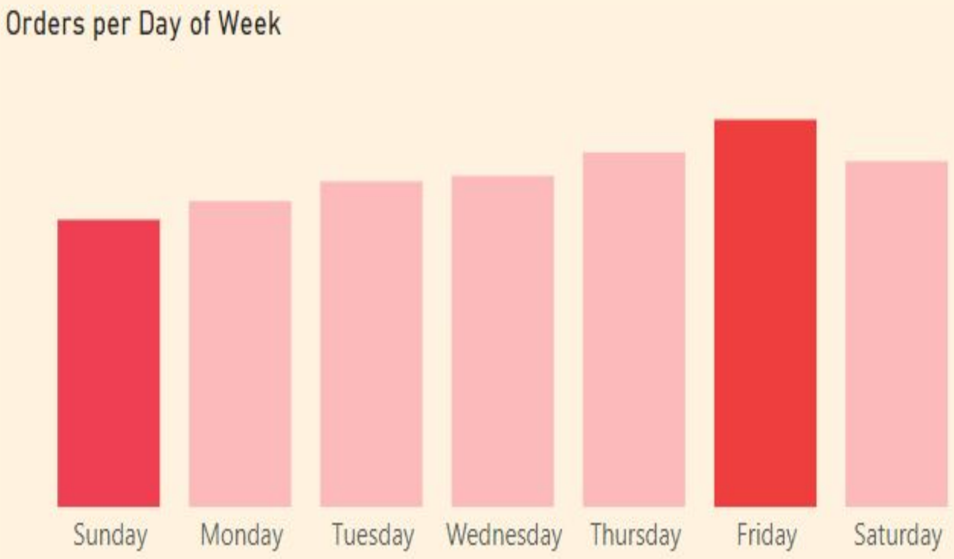


Orders by Hour of Day

Peak hours: **12 pm – 2 pm** (6,543 and 6,203 orders)
Off-peak hours: **9 pm – 11 pm** (1,370 and 68 orders)

```
SELECT LEFT(o.time, 2) AS hour_of_day, COUNT(*) AS
total_orders
FROM orders o
JOIN order_details od ON o.order_id = od.order_id
GROUP BY LEFT(o.time, 2);
```

DAILY AVERAGES

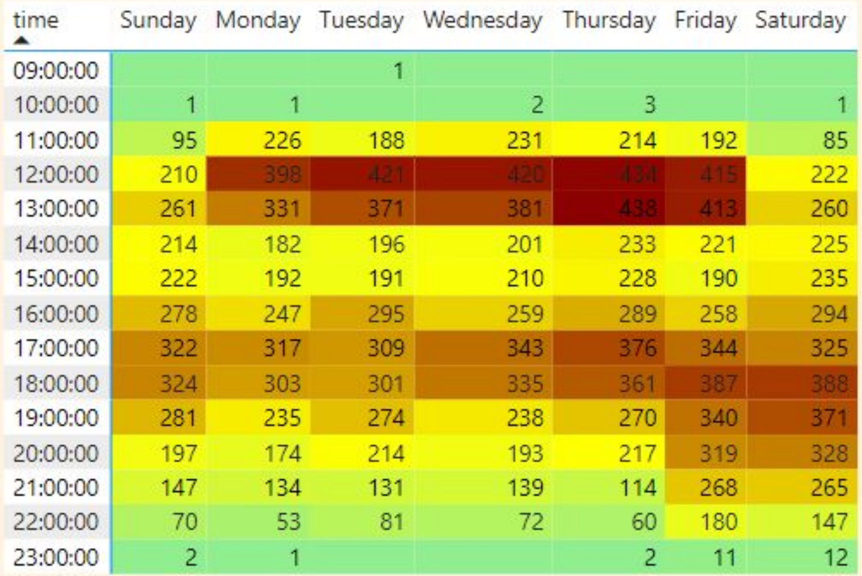


Highest average orders: **Friday (70.76)**
Lowest average orders: **Sunday (50.46)**

```
WITH day_names AS (
  SELECT 1 AS day_num, 'Sunday' AS day_name
  UNION SELECT 2, 'Monday'
  UNION SELECT 3, 'Tuesday'
  UNION SELECT 4, 'Wednesday'
  UNION SELECT 5, 'Thursday'
  UNION SELECT 6, 'Friday'
  UNION SELECT 7, 'Saturday'
)
SELECT
  dn.day_num,
  dn.day_name,
  AVG(daily_orders.orders_per_day) AS
avg_orders_per_day
FROM (
  SELECT
    DAYOFWEEK(date) AS day_of_week,
    COUNT(*) AS orders_per_day
  FROM orders
  GROUP BY date
) daily_orders
JOIN day_names dn ON daily_orders.day_of_week =
dn.day_num
GROUP BY dn.day_num, dn.day_name
ORDER BY dn.day_num;
```

PEAK PERIOD ANALYSIS

Combining Hourly and Daily Trends



time ▲	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
09:00:00			1				
10:00:00	1	1		2	3		1
11:00:00	95	226	188	231	214	192	85
12:00:00	210	398	421	420	434	415	222
13:00:00	261	331	371	381	438	413	260
14:00:00	214	182	196	201	233	221	225
15:00:00	222	192	191	210	228	190	235
16:00:00	278	247	295	259	289	258	294
17:00:00	322	317	309	343	376	344	325
18:00:00	324	303	301	335	361	387	388
19:00:00	281	235	274	238	270	340	371
20:00:00	197	174	214	193	217	319	328
21:00:00	147	134	131	139	114	268	265
22:00:00	70	53	81	72	60	180	147
23:00:00	2	1			2	11	12

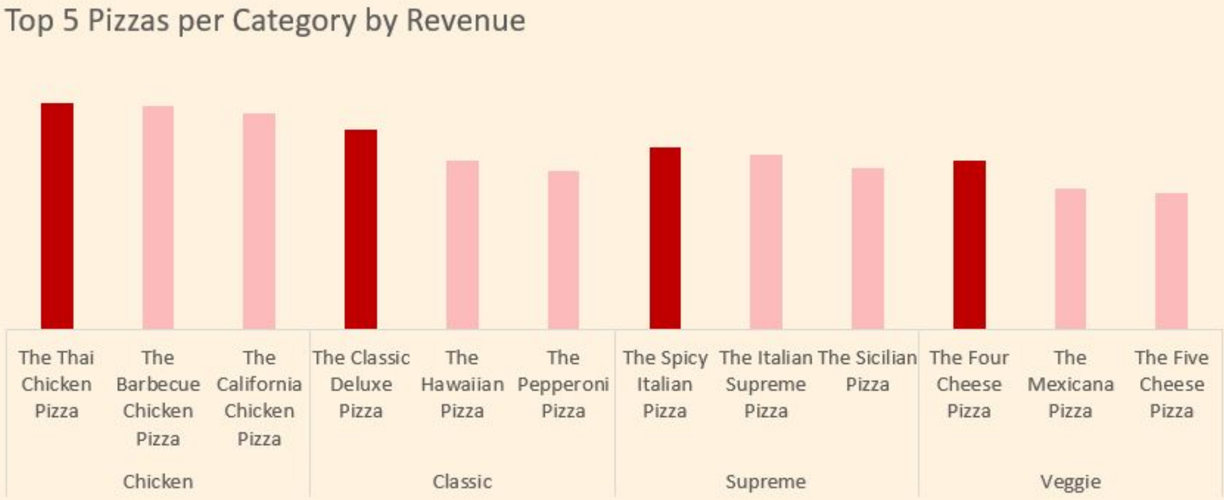
Peak periods: Everyday lunch (12 pm - 2 pm) and weekend dinner (5 pm - 7 pm)

Off-peak periods: Monday and Wednesday evenings

05

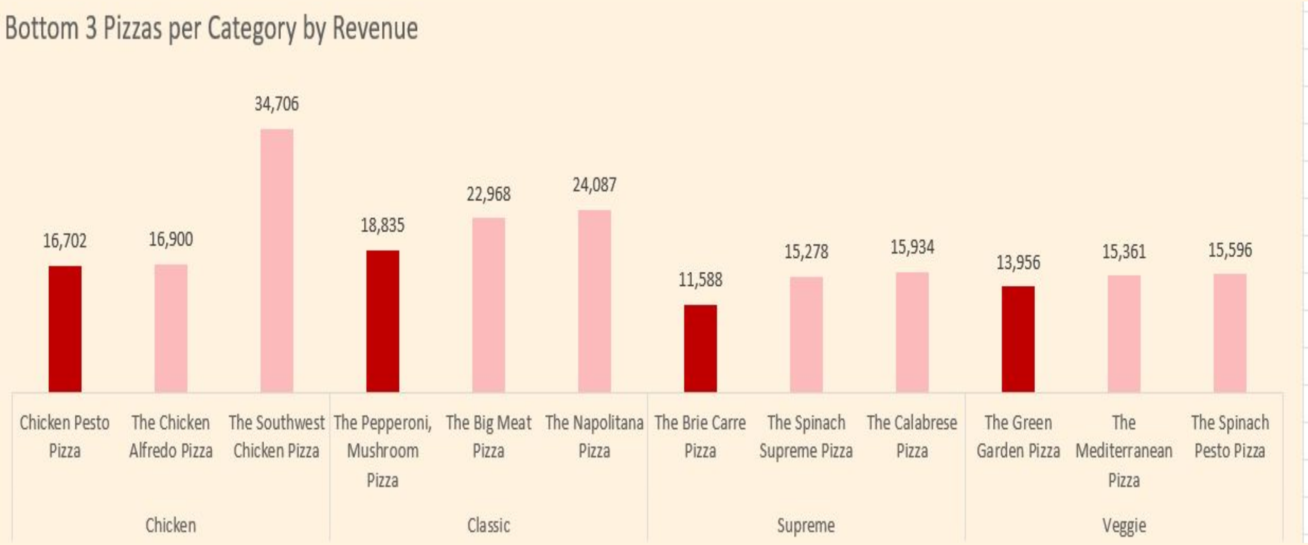
CUSTOMER BEHAVIOUR & PREFERENCES

BEST PERFORMING PIZZAS BY CATEGORY



```
WITH ranked_pizzas AS (
   SELECT
      pt.category,
      pt.name,
      SUM(p.price * o.quantity) AS Total_revenue,
      ROW_NUMBER() OVER (PARTITION BY
pt.category ORDER BY SUM(p.price * o.quantity)
DESC) AS rank1
   FROM
      order_details o
      JOIN pizzas p ON o.pizza_id = p.pizza_id
      JOIN pizza_type pt ON p.pizza_type_id =
pt.pizza_type_id
   GROUP BY
      pt.category, pt.name
)
SELECT
   category,
   name AS pizza_type,
   Total_revenue
FROM
   ranked_pizzas
WHERE
   rank1 <= 3
ORDER BY
   category, Total_revenue DESC;
```

WORST PERFORMING PIZZAS BY CATEGORY



```
WITH ranked_pizzas AS (
   SELECT
      pt.category,
      pt.name,
      SUM(p.price * o.quantity) AS Total_revenue,
      ROW_NUMBER() OVER (PARTITION BY
pt.category ORDER BY SUM(p.price * o.quantity)
ASC) AS rank1
   FROM
      order_details o
      JOIN pizzas p ON o.pizza_id = p.pizza_id
      JOIN pizza_type pt ON p.pizza_type_id =
pt.pizza_type_id
   GROUP BY
      pt.category, pt.name
)
SELECT
   category,
   name AS pizza_type,
   Total_revenue
FROM
   ranked_pizzas
WHERE
   rank1 <= 3
ORDER BY
   category, Total_revenue ASC;
```

06

INSIGHTS & RECOMMENDATIONS

INSIGHTS AND ANALYSIS

The data suggests that there is a mismatch between menu offerings and customer preferences. Veggie pizzas, despite being a large portion of the menu, are underperforming, while chicken pizzas drive a significant portion of orders.

Opportunity to upsell/cross-sell: With an average order value of $17, there's an opportunity to increase revenue by offering upsells or cross-sells, such as drinks, sides, or desserts.

Popular pizzas might be hurting sales of other items: Think about changing prices or promotions to help other menu items sell better

RECOMMENDATIONS FOR BUSINESS GROWTH AND IMPROVEMENT

Optimize menu engineering:

- Consider removing or modifying underperforming veggie pizzas
- Add more chicken pizza options to capitalize on demand

Enhance marketing efforts:

- Target promotions and ads towards peak ordering periods (Fridays and lunch hours)
- Highlight top-selling pizzas and new menu items

Improve operational efficiency:

- Adjust staffing and inventory accordingly during peak periods
- Streamline order processing and delivery times