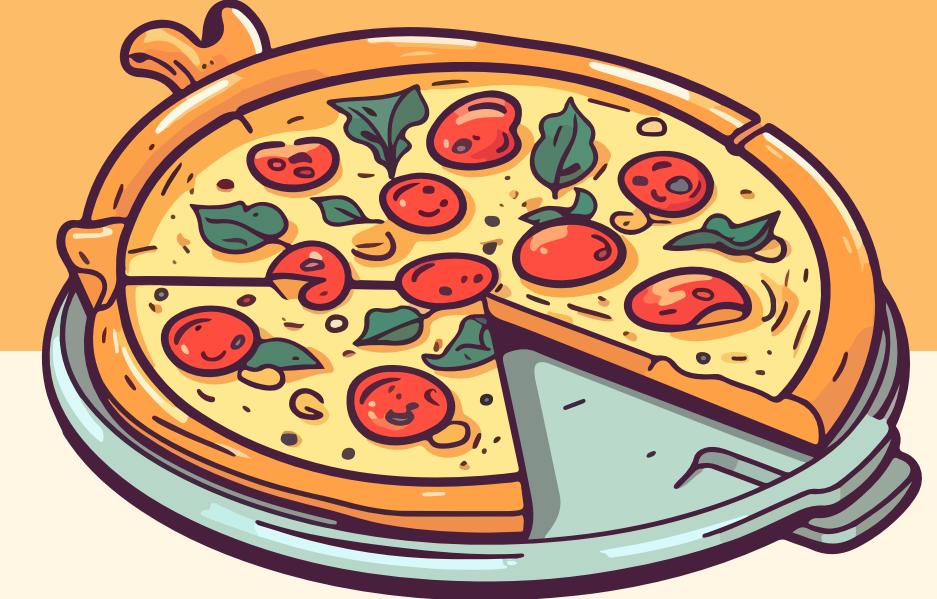




PIZZA SALES

CASE STUDY

OVERVIEW



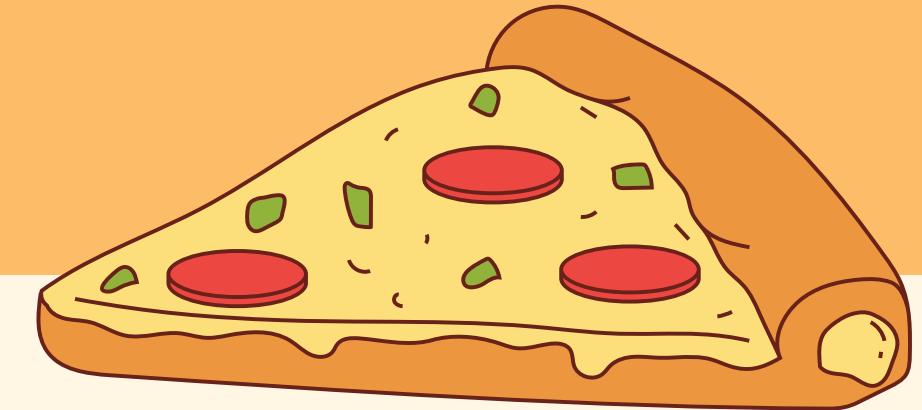
We have a dataset containing a year's worth of sales data from a pizza place. This dataset includes detailed information about each order, such as the date and time, the pizzas served (type, size, quantity, price, and ingredients), and additional details. Our goal is to perform comprehensive analysis and extract actionable insights to optimize business operations and maximize revenue.

BUSINESS UNDERSTANDING



PIZZERIA is the business of selling delicious, freshly made pizzas to a diverse customer base. Our focus is on quality, variety, and exceptional customer service to satisfy cravings and build loyalty. We prioritize efficient operations, menu innovation, and data-driven decision-making to stay competitive and drive growth in the dynamic food industry.

BUSINESS OBJECTIVE



1. Gain insights into customer behavior and preferences.
2. Identify peak hours and busiest days to optimize staffing and resources.
3. Determine the average number of pizzas per order and identify bestsellers.
4. Analyze overall revenue and identify any seasonality in sales.
5. Evaluate the performance of individual pizzas and categories to optimize the menu.
6. Develop targeted promotions or menu adjustments to increase sales and customer satisfaction.

DETAILED ANALYSIS PLAN



1. Customer Behavior Analysis:

- Count total orders and analyze distribution by day and hour.
- Identify peak hours and busiest days.

2. Order Analysis:

- Calculate the average order value.
- Determine the typical number of pizzas per order.
- Identify the top-selling pizzas and categories.

3. Revenue Analysis:

- Calculate total revenue and analyze revenue distribution over time (by quarter, month, and day).
- Identify any seasonality in sales.

4. Menu Optimization:

- Analyze the performance of individual pizzas and categories.
- Determine which pizzas to keep, remove, or promote based on sales volume and revenue contribution.

5. Actionable Insights:

- Develop strategies to optimize staffing and resources based on peak hours.
- Create promotions or menu adjustments to capitalize on bestsellers and increase revenue.
- Identify opportunities for business growth and improvement.

5. Actionable Insights:

- Develop strategies to optimize staffing and resources based on peak hours.
- Create promotions or menu adjustments to capitalize on bestsellers and increase revenue.
- Identify opportunities for business growth and improvement.

DATABASE

```
use pizza_sales_analysis ;  
SHOW TABLES;
```

There are 4 tables in the database-
order_details,orders,pizza_types,pizzas
and one is derived table which is
date_info.

Tables_in_pizza_sales_analysis

| |
|---------------|
| date_info |
| order_details |
| orders |
| pizza_types |
| pizzas |

Table columns

order_details

| | order_details_id | order_id | pizza_id | quantity |
|--|------------------|----------|---------------|----------|
| | 1 | 1 | hawaiian_m | 1 |
| | 2 | 2 | classic_dlx_m | 1 |
| | 3 | 2 | five_cheese_l | 1 |
| | 4 | 2 | ital_supr_l | 1 |
| | 5 | 2 | mexicana_m | 1 |
| | 6 | 2 | thai_ckn_l | 1 |
| | 7 | 3 | ital_supr_m | 1 |
| | 8 | 3 | prsc_argla_l | 1 |
| | 9 | 4 | ital_supr_m | 1 |
| | 10 | 5 | ital_supr_m | 1 |

pizza_types

| pizza_type_id | name | category | ingredients |
|---------------|------------------------------|----------|--|
| bbq_ckn | The Barbecue Chicken Pizza | Chicken | Barbecued Chicken, Red Peppers, Green Peppers, Onions, Lettuce, Tomato, Ranch Dressing |
| big_meat | The Big Meat Pizza | Classic | Bacon, Pepperoni, Italian Sausage, Chorizo Sausage, Ground Beef, Mozzarella Cheese |
| brie_carre | The Brie Carre Pizza | Supreme | Brie Carre Cheese, Prosciutto, Caramelized Onion, Fresh Mozzarella, Arugula, Lettuce, Tomato, Basil, Olive Oil, Salt, Pepper |
| calabrese | The Calabrese Pizza | Supreme | 'Nduja Salami, Pancetta, Tomatoes, Red Onions, Fresh Mozzarella, Arugula, Lettuce, Tomato, Basil, Olive Oil, Salt, Pepper |
| cali_ckn | The California Chicken Pizza | Chicken | Chicken, Artichoke, Spinach, Garlic, Jalapeno Peppers, Mozzarella Cheese, Lettuce, Tomato, Olive Oil, Salt, Pepper |
| ckn_alfredo | The Chicken Alfredo Pizza | Chicken | Chicken, Red Onions, Red Peppers, Mushrooms, Alfredo Sauce, Mozzarella Cheese, Lettuce, Tomato, Olive Oil, Salt, Pepper |
| ckn_pesto | The Chicken Pesto Pizza | Chicken | Chicken, Tomatoes, Red Peppers, Spinach, Garlic, Mozzarella Cheese, Lettuce, Tomato, Olive Oil, Salt, Pepper |

orders

| | order_id | date | time |
|--|----------|------------|----------|
| | 1 | 2015-01-01 | 11:38:36 |
| | 2 | 2015-01-01 | 11:57:40 |
| | 3 | 2015-01-01 | 12:12:28 |
| | 4 | 2015-01-01 | 12:16:31 |
| | 5 | 2015-01-01 | 12:21:30 |
| | 6 | 2015-01-01 | 12:29:36 |
| | 7 | 2015-01-01 | 12:50:37 |
| | 8 | 2015-01-01 | 12:51:37 |
| | 9 | 2015-01-01 | 12:52:01 |

pizzas

| pizza_id | pizza_type_id | size | price |
|--------------|---------------|-------|-------|
| bbq_ckn_l | bbq_ckn | L | 20.75 |
| bbq_ckn_m | bbq_ckn | M | 16.75 |
| bbq_ckn_s | bbq_ckn | S | 12.75 |
| big_meat_l | big_meat | L | 20.5 |
| big_meat_m | big_meat | M | 16 |
| big_meat_s | big_meat | S | 12 |
| brie_carre_s | brie_carre | S | 23.65 |
| | | | |

PERFORMING ANALYSIS USING SQL QUERIES



CUSTOMER BEHAVIOUR ANALYSIS

1.WHAT DAY AND HOUR TO BE BUSIEST?

SELECT

```
hour_of_day,  
SUM(CASE WHEN day_of_week = 'Monday' THEN total_orders ELSE 0 END) AS Monday,  
SUM(CASE WHEN day_of_week = 'Tuesday' THEN total_orders ELSE 0 END) AS Tuesday,  
SUM(CASE WHEN day_of_week = 'Wednesday' THEN total_orders ELSE 0 END) AS Wednesday,  
SUM(CASE WHEN day_of_week = 'Thursday' THEN total_orders ELSE 0 END) AS Thursday,  
SUM(CASE WHEN day_of_week = 'Friday' THEN total_orders ELSE 0 END) AS Friday,  
SUM(CASE WHEN day_of_week = 'Saturday' THEN total_orders ELSE 0 END) AS Saturday,  
SUM(CASE WHEN day_of_week = 'Sunday' THEN total_orders ELSE 0 END) AS Sunday
```

FROM (

SELECT

```
HOUR(o.time) AS hour_of_day,  
dayname(di.date) AS day_of_week,  
COUNT(distinct o.ORDER_ID) AS total_orders
```

FROM

```
ORDERS o  
INNER JOIN date_info di ON o.date = di.date  
GROUP BY HOUR(o.time), DAYNAME(di.date)
```

) AS t

GROUP By

hour_of_day

ORDER BY

hour_of_day;

| hour_of_day | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday |
|-------------|--------|---------|-----------|----------|--------|----------|--------|
| 9 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 10 | 1 | 0 | 2 | 3 | 0 | 1 | 1 |
| 11 | 226 | 188 | 231 | 214 | 192 | 85 | 95 |
| 12 | 398 | 421 | 420 | 434 | 415 | 222 | 210 |
| 13 | 331 | 371 | 381 | 438 | 413 | 260 | 261 |
| 14 | 182 | 196 | 201 | 233 | 221 | 225 | 214 |
| 15 | 192 | 191 | 210 | 228 | 190 | 235 | 222 |
| 16 | 247 | 295 | 259 | 289 | 258 | 294 | 278 |
| 17 | 317 | 309 | 343 | 376 | 344 | 325 | 322 |
| 18 | 303 | 301 | 335 | 361 | 387 | 388 | 324 |
| 19 | 235 | 274 | 238 | 270 | 340 | 371 | 281 |
| 20 | 174 | 214 | 193 | 217 | 319 | 328 | 197 |
| 21 | 134 | 131 | 139 | 114 | 268 | 265 | 147 |
| 22 | 53 | 81 | 72 | 60 | 180 | 147 | 70 |
| 23 | 1 | 0 | 0 | 2 | 11 | 12 | 2 |

ORDER ANALYSIS

1. CALCULATE THE AVERAGE ORDER VALUE

```
SELECT round((sum(p.price * o.quantity) / (SELECT count(order_id) FROM orders)),  
2) AS AVERAGE_ORDER_VALUE  
FROM pizzas p  
INNER JOIN order_details o ON p.pizza_id = o.pizza_id;
```

| AVERAGE_ORDER_VALUE |
|---------------------|
| 38.31 |

2. Calculate the average order value.

```
SELECT round((sum(p.price * o.quantity) / (SELECT count(order_id) FROM orders)),  
2) AS AVERAGE_ORDER_VALUE  
FROM pizzas p  
INNER JOIN order_details o ON p.pizza_id = o.pizza_id;
```

| AVERAGE_ORDER_VALUE |
|---------------------|
| 38.31 |

3. Determine the typical number of pizzas per order.

```
SELECT AVG(pizzas_per_order) AS average_pizzas_per_order  
FROM (  
    SELECT order_id, COUNT(*) AS pizzas_per_order  
    FROM order_details  
    GROUP BY order_id  
) AS order_pizza_counts;
```

| average_pizzas_per_order |
|--------------------------|
| 2.2773 |

4. Identify the top-selling pizzas and categories.

```
SELECT
    p.pizza_id, sum(quantity) AS total_quantity
FROM
    order_details od join pizzas p on
    od.pizza_id= p.pizza_id
GROUP BY p.pizza_id
ORDER BY total_quantity DESC
LIMIT 5;
```

| pizza_id | total_quantity |
|---------------|----------------|
| big_meat_s | 1914 |
| thai_dn_l | 1410 |
| five_cheese_l | 1409 |
| four_cheese_l | 1316 |
| dassic_dx_m | 1181 |

```
SELECT
    Pt.category, SUM(od.quantity) AS quantity
FROM
    pizza_types pt
        INNER JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        INNER JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.category
ORDER BY quantity DESC;
```

| category | quantity |
|----------|----------|
| Classic | 14888 |
| Supreme | 11987 |
| Veggie | 11649 |
| Chicken | 11050 |

5. Find the top 5 most ordered pizza types along with their quantities.

```
SELECT Pt.name,sum( od.quantity) as total_orders  
FROM pizza_types pt inner join  
pizzas p on  
pt.pizza_type_id= p.pizza_type_id  
inner join order_details od  
on p.pizza_id = od.pizza_id  
group by pt.name  
order by total_orders desc  
limit 5;
```

| name | total_orders |
|----------------------------|---------------------|
| The Classic Deluxe Pizza | 2453 |
| The Barbecue Chicken Pizza | 2432 |
| The Hawaiian Pizza | 2422 |
| The Pepperoni Pizza | 2418 |
| The Thai Chicken Pizza | 2371 |

REVENUE ANALYSIS

1. FIND THE TOTAL REVENUE

```
SELECT  
    ROUND(SUM(p.price * o.quantity), 2) AS TOTAL_REVENUE  
FROM  
    pizzas P  
    INNER JOIN  
    order_details O ON P.pizza_id = O.pizza_id;
```

| TOTAL_REVENUE |
|---------------|
| 817860.05 |

2. Find the total revenue for each quarter

```
WITH OrderDetailsRevenue AS (
    SELECT o.date,
    SUM(p.price * od.quantity) AS revenue
    FROM orders o
    JOIN order_details od
    ON o.order_id = od.order_id
    JOIN pizzas p
    ON od.pizza_id = p.pizza_id
    GROUP BY
    o.date
),
DateInfoQuarter AS (
    SELECT
        date, QUARTER(date) AS quarter
    FROM date_info
    WHERE
        quarter IS NOT NULL
)
SELECT diq.quarter,
    round(SUM(ODR.revenue),2) AS Total_Revenue
    FROM DateInfoQuarter diq
    JOIN OrderDetailsRevenue ODR ON
    diq.date = ODR.date
    GROUP BY
    diq.quarter;
```

| quarter | Total_Revenue |
|---------|---------------|
| 1 | 12442770.4 |
| 2 | 12638079.4 |
| 3 | 12671840.3 |
| 4 | 12309252.6 |

3. Find the total revenue for each month

```
WITH OrderDetailsRevenue AS (
    SELECT o.date,
    SUM(p.price * od.quantity) AS revenue
    FROM orders o
    JOIN order_details od
    ON o.order_id = od.order_id
    JOIN pizzas p
    ON od.pizza_id = p.pizza_id
    GROUP BY
    o.date
),
```

```
DateInfoQuarter AS (
    SELECT
        date,month(date) AS month
    FROM date_info
    WHERE
        month IS NOT NULL
)
SELECT diq.month,
    round(SUM(ODR.revenue),2) AS Total_Revenue
FROM DateInfoQuarter diq
JOIN OrderDetailsRevenue ODR ON
    diq.date = ODR.date
GROUP BY
    diq.month;
```

| month | Total_Revenue |
|-------|---------------|
| 1 | 4211337.55 |
| 2 | 3994483.95 |
| 3 | 4236948.9 |
| 4 | 4166161.95 |
| 5 | 4367176.05 |
| 6 | 4104741.4 |
| 7 | 4693033.85 |
| 9 | 3856639.65 |
| 11 | 4503324.9 |
| 8 | 4122166.8 |
| 10 | 4050775.95 |
| 12 | 3755151.75 |

4.Find the total revenue by pizza category

```
SELECT
    Pt.category,
    ROUND(SUM(p.price * od.quantity), 2) AS total_revenue
FROM
    pizza_types pt
        INNER JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        INNER JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.category
ORDER BY total_revenue DESC;
```

| category | total_revenue |
|----------|---------------|
| Classic | 220053.1 |
| Supreme | 208197 |
| Chicken | 195919.5 |
| Veggie | 193690.45 |

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

SELECT

```
Pt.category,  
CONCAT(ROUND(SUM(p.price * od.quantity) / (SELECT  
    ROUND(SUM(p.price * o.quantity), 2) AS total_sales  
FROM  
    pizzas P  
    INNER JOIN  
    order_details O ON P.pizza_id = O.pizza_id) * 100,  
    2),  
    '%') AS revenue  
FROM  
    pizza_types pt  
    INNER JOIN  
    pizzas p ON pt.pizza_type_id = p.pizza_type_id  
    INNER JOIN  
    order_details od ON p.pizza_id = od.pizza_id  
GROUP BY pt.category  
ORDER BY revenue DESC;
```

| category | revenue |
|----------|---------|
| Classic | 26.91% |
| Supreme | 25.46% |
| Chicken | 23.96% |
| Veggie | 23.68% |

6.Analyze the cumulative revenue generated over time

```
select date,sum(revenue) over(order by date) as cum_revenue  
from  
(select o.date,  
sum(od.quantity*p.price) as revenue  
from order_details od join pizzas p  
on od.pizza_id=p.pizza_id  
join orders o on  
o.order_id= od.order_id  
group by o.date) as sales;
```

| | |
|------------|-------------------|
| 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 | 32259.0000000004 |

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

```
select category,name ,revenue from  
(select category,name,revenue,  
rank() over(partition by category order by revenue desc) as rn  
from  
(select pt.category ,pt.name,  
sum((od.quantity)*p.price) as revenue  
from pizza_types pt join pizzas p  
on pt.pizza_type_id= p.pizza_type_id  
join order_details od on  
od.pizza_id=p.pizza_id  
group by pt.category, pt.name) as a ) as b  
where rn<=3;
```

| category | name | revenue |
|----------|------------------------------|-------------------|
| Chicken | The Thai Chicken Pizza | 43434.25 |
| Chicken | The Barbecue Chicken Pizza | 42768 |
| Chicken | The California Chicken Pizza | 41409.5 |
| Classic | The Classic Deluxe Pizza | 38180.5 |
| Classic | The Hawaiian Pizza | 32273.25 |
| Classic | The Pepperoni Pizza | 30161.75 |
| Supreme | The Spicy Italian Pizza | 34831.25 |
| Supreme | The Italian Supreme Pizza | 33476.75 |
| Supreme | The Sicilian Pizza | 30940.5 |
| Veggie | The Four Cheese Pizza | 32265.70000000554 |
| Veggie | The Mexicana Pizza | 26780.75 |
| Veggie | The Five Cheese Pizza | 26066.5 |

MENU OPTIMIZATION

1. Categorize on the basis of sales volume of pizza

```
SELECT  
    pizza_id,  
    sales_category  
FROM (  
    SELECT  
        pizza_id,  
        CASE  
            WHEN total_orders >= 1200 THEN 'Good'  
            WHEN total_orders >=500 and total_orders<=1200 THEN 'Medium'  
            ELSE 'Low'  
        END AS sales_category
```

```
    FROM (  
        SELECT  
            pizza_id,  
            COUNT(*) AS total_orders  
        FROM  
            order_details  
        GROUP BY  
            pizza_id  
    ) AS pizza_sales  
) AS categorized_sales  
WHERE  
    sales_category IN ('Medium', 'Low');
```

| pizza_id | sales_category |
|---------------|----------------|
| bbq_dkn_l | Medium |
| bbq_dkn_m | Medium |
| bbq_dkn_s | Low |
| brie_carre_s | Low |
| calabrese_l | Low |
| calabrese_m | Medium |
| calabrese_s | Low |
| cali_dkn_l | Medium |
| cali_dkn_m | Medium |
| cali_dkn_s | Low |
| dkn_alfredo_l | Low |
| dkn_alfredo_m | Medium |
| dkn_alfredo_s | Low |
| dkn pesto_l | Low |
| dkn pesto_m | Low |
| dkn pesto_s | Low |

There are several pizzas which fall into medium and low category .Here is the scope for promotion and removal of pizzas decision can take place. The best way to do is by comparing ordered quantity and revenue along with customer feedback.

2.Find 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 |

3.Find the highest priced pizza.

```
SELECT category,count(name) from pizza_types  
group by category;
```

| category | count(name) |
|----------|-------------|
| Chicken | 6 |
| Classic | 8 |
| Supreme | 9 |
| Veggie | 9 |

4.Determine the distribution of orders by hour of the day

```
SELECT HOUR(time) as hour, count(order_id) as order_count from orders  
group by hour(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 |

5.Find the size of most common ordered

```
SELECT p.size, count(od.order_details_id) as order_count  
from pizzas p join order_details od  
on p.pizza_id=od.pizza_id  
group by p.size order by order_count desc;
```

| size | order_count |
|------|-------------|
| L | 18526 |
| M | 15385 |
| S | 14137 |
| XL | 544 |
| XXL | 28 |

6.Calculate the average number of pizzas ordered per day.

```
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizzas_perday
FROM
    (SELECT
        o.DATE, SUM(od.quantity) AS quantity
    FROM
        orders o
    JOIN order_details od ON o.order_id = od.order_id
    GROUP BY o.date) AS order_quantity;
```

| avg_pizzas_perday |
|-------------------|
| 138 |

SUMMARY

Total Orders and Revenue: With 21,350 total orders generating \$817.86k in revenue, it's evident that the pizza business is thriving.

Average Order Value: The average order value of \$38.31 suggests that customers tend to spend moderately per order, indicating a balanced pricing strategy.

Price Analysis: The Greek Pizza stands out with the highest price of \$35.95. This indicates potential for premium pricing strategies for specialty pizzas.

Pizza Variety: Veggie and Supreme pizzas lead with 9 varieties each, indicating strong customer demand for these options. Classic and Chicken pizzas also have significant variety.

Busiest Hours and Days: Peak order hours from 12 to 8 pm, particularly on Thursdays and Fridays, suggest targeted marketing and staffing efforts during these periods.

Popular Pizza Sizes: Large pizzas are the most commonly ordered size, followed by medium, indicating customer preferences for larger portions.

Quarterly Performance: While order volumes generally increase with each quarter, the 4th quarter shows a slight decline compared to the 1st quarter. Seasonality factor is there.in holiday seasons people preferred to go out of the town this might be a factor

Monthly Performance: July emerges as the month with the highest number of orders, while November and December have relatively lower order volumes. Seasonal factors and promotional activities may influence these variations.

Weekday Revenue: Friday and Saturday consistently generate the highest revenue, suggesting increased customer engagement during weekends. However, Sunday revenue is relatively lower, indicating potential opportunities for targeted promotions or menu offerings to boost sales on this day.

Bestselling Pizzas: Classic Deluxe and Barbecue Chicken are the most ordered pizzas, indicating strong customer preferences for these flavors. Promotions or special deals on these pizzas can further capitalize on their popularity.

Revenue by Category: Classic pizzas contribute the highest revenue, indicating their popularity among customers. However, it's important to explore opportunities to promote other categories and diversify revenue streams.

SUMMARY

Menu Optimization:

We can use customer feedback and sales data to refine the menu, focusing on bestselling pizzas and introducing new flavors to attract diverse customer preferences. Also we can ask for low and medium range pizza.

Promotional Strategies:

We can leverage peak hours and days to run targeted promotions, discounts, or bundle offers to drive sales during slower periods.

Seasonal Campaigns:

We use tailor marketing campaigns and menu offerings to align with seasonal trends and capitalize on peak demand periods.

Customer Engagement:

By implementing loyalty programs or rewards to incentivize repeat purchases and foster customer loyalty.

Operational Efficiency:

Optimize staffing and operations during peak hours to ensure timely order fulfillment and enhance customer satisfaction.