



CRUST TO CRUMB: UNDERSTANDING PIZZA SALES METRICS

P I Z Z A S P O T
R E S T A U R A N T



ABOUT US



Hello, I'm Akshay Bhujbal, and I spearheaded a comprehensive pizza sales analysis project utilizing SQL. With a keen eye for data-driven insights and a passion for uncovering trends, I delved into the vast dataset to extract valuable information about pizza sales dynamics.

Through meticulous SQL queries and analysis, we dissected sales patterns, customer preferences, and market trends to provide actionable insights for decision-makers in the pizza industry. Our project aimed to not only understand current sales performance but also to forecast future trends and opportunities.

Join me on this journey through the world of pizza sales analysis, where we blend the art of SQL with the science of data to deliver meaningful insights and drive strategic decision-making in the food industry.

ABOUT DATA

"In our pizza sales analysis project, we delved into four datasets: orders, order_details, pizza_types, and pizzas. These datasets, sourced from Kaggle, offered rich insights into various facets of pizza sales. Leveraging SQL's join and subquery functionalities, we tackled a spectrum of questions, from basic inquiries to advanced analytics. Our approach enabled us to uncover valuable trends and patterns, empowering data-driven decision-making in the pizza industry."





"ANALYTICS SHOWCASE: UNVEILING INSIGHTS FROM PIZZA SALES DATA"

Retrieve the total number of orders placed

```
select count(order_id) as total_orders from orders;
```



Calculate the total revenue generated from pizza sales.

```
SELECT  
    ROUND(SUM(order_details.quantity * pizzas.price),  
        2) AS total_sales  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

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;
```

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

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

Determine the distribution of orders by hour of the day.

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

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

```
SELECT
    ROUND( AVG(quantity), 0)as avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

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

```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
        2) AS total_sales
    FROM
        order_details
        JOIN
            pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
    2) AS revenue
FROM
    pizza_types
    JOIN
        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
        order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

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(order_details.quantity * pizzas.price) as revenue
      from order_details join pizzas
        on order_details.pizza_id= pizzas.pizza_id
     join orders
        on orders.order_id=order_details.order_id
   group by orders.order_date)as sales;
```

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((order_details.quantity )* pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b;
where rn <=3;
```



RECOMMENDATIONS

- 1.Optimize Pricing Strategy: Consider adjusting pricing strategies for pizzas based on their demand and popularity to maximize revenue.
- 2.Promotional Campaigns: Launch targeted promotional campaigns for popular pizza types to increase sales and customer engagement.
- 3.Inventory Management: Ensure adequate inventory levels for ingredients used in the most ordered pizzas to avoid stockouts and meet customer demand.
- 4.Operational Efficiency: Streamline operations during peak hours by staffing accordingly to handle the distribution of orders effectively.
- 5.Diversification: Explore introducing new pizza varieties or customization options based on customer preferences to expand the product range and attract a wider customer base.

CONCLUSION:

The analysis of pizza sales data has provided valuable insights into customer preferences, revenue generation, and operational patterns. By leveraging these insights, the pizza business can make informed decisions to enhance sales, improve customer satisfaction, and optimize operational efficiency. Moving forward, continued monitoring of sales trends and customer behavior will be essential to adapt strategies and stay competitive in the market.





PIZZA SPOT
RESTAURANT

THANK YOU

