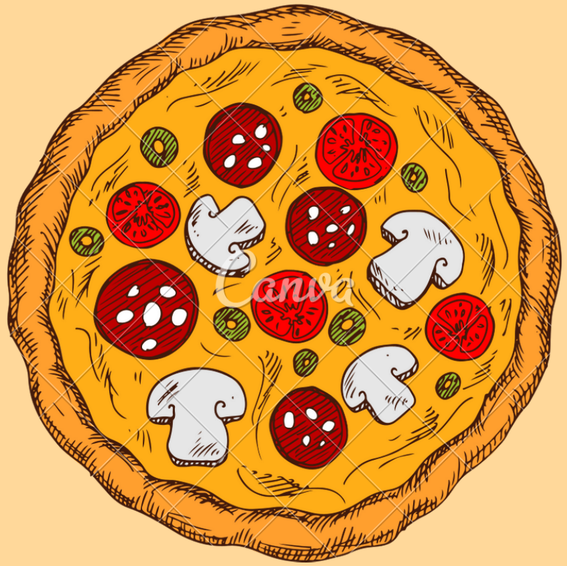


PIZZA SALES ANALYSIS

UNCOVERING INSIGHTS WITH MYSQL

Hello!

Greetings, and welcome to our MySQL project presentation. In an age where consumer preferences shape market landscapes, our MySQL project endeavors to uncover the delicious details that is secretely hidden within the numbers. Join us as we we uncover insights that can reshape the landscape of pizza consumption, elevate business strategies and drive success.

OBJECTIVE

- Explore pizza sales data to discern popular pizza varieties, toppings, and sizes.
- Deep dive into pizza sales trends, including revenue analysis and average order value.
- Enhance inventory management and pricing strategies through data-driven insights and demand forecasting.
- Identify peak hours and periods of high order volume to optimize staffing and operational efficiency.

SCOPE

- Overview of the dataset and our methodology for data preprocessing.
- Detailed analysis of pizza sales trends, revenue breakdown, and average order value.
- Exploration of customer segments based on demographics and purchasing behavior.
- Presentation of key findings through MySQL queries and data visualizations.

Retrieve the total number of orders placed.

```sql
SELECT
    COUNT(order_id) AS Total_Order
FROM
    orders;
```

Result Grid

	Total_Order
▶	21350

Identify the highest-priced pizza.

```sql
SELECT
    pizza_types.name AS Product_Name,
    pizzas.price AS Highest_Price
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result Grid | Filter Rows:

Product_Name	Highest_Price
The Greek Pizza	35.95

Join relevant tables to find the category-wise distribution of pizzas.

```sql
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Rows

category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

Calculate the total revenue generated from pizza sales.

```sql
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;
```

Result Grid

	Total_Sales
▶	817860.05

Identify the most common pizza size ordered.

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

Result Grid

size	Order_count
L	18526

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

```sql
SELECT
    pizza_types.name as Name, 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.name
ORDER BY Quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

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

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

```sql
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;
```

Result Grid | Filter

category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(order_time) AS At_Hour, COUNT(order_id) AS Total_Orders
FROM
    orders
GROUP BY At_hour
ORDER BY At_hour;
```

Result Grid | Filter Rc

At_Hour	Total_Orders
9	1
10	8
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

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

```sql
SELECT
    round(AVG(quantity),0) AS Average_Pizza_Ordered
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;
```

Result Grid | Filter Rov

	Average_Pizza_Ordered
▶	138

Determine the top 3 most ordered pizza types based on revenue.

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity * Pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

```sql
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;
```

Result Grid | Filter

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time.

```sql
select order_date, sum(revenue) over (order by order_date) as Cum_Rev
from
(select orders.order_date, sum(order_details.quantity * pizzas.price) as Revenue
from orders join order_details
on orders.order_id = order_details.order_id
join pizzas
on pizzas.pizza_id = order_details.pizza_id
group by order_date) as Sales;
```

Result Grid | Filter Rows:

order_date	Cum_Rev
2015-01-01	2713.8500000000004
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	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006

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

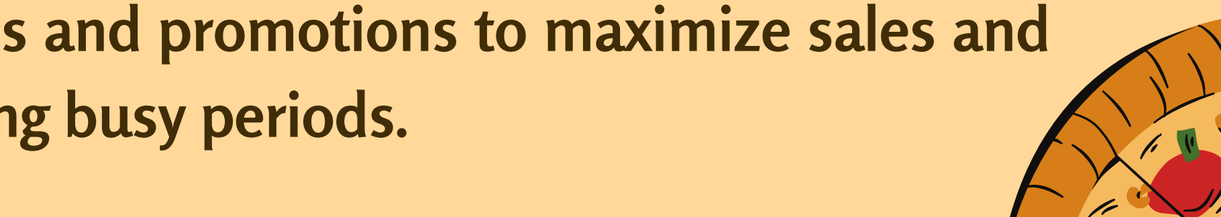
```sql
select category,name, revenue from
(select category, name, revenue, rank() over(partition by category order by revenue desc)
as c
from
(select pizza_types.category, pizza_types.name, sum(order_details.quantity * pizzas.price)
as Revenue
from order_details join pizzas
on order_details.pizza_id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

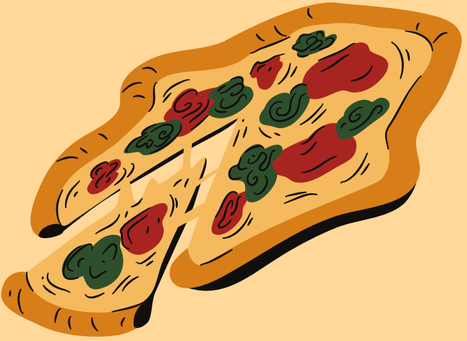
Result Grid | Filter Rows: | Export:

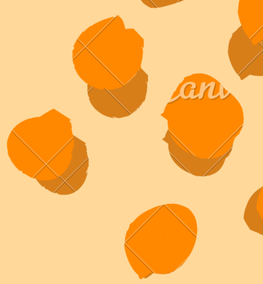
category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

KEY FINDINGS

- Revenue Trends: We identified peak periods of revenue generation, allowing us to optimize pricing strategies and promotional efforts for maximum profitability.
- Sales Volume Analysis: By analyzing sales volume across different pizza varieties, toppings, and sizes, we identified top-selling products and customer preferences, enabling us to streamline inventory management and production planning.
- Order Trends: Our analysis of order trends by day of the week, time of day, and season revealed peak hours and periods of high demand, informing staffing and resource allocation decisions to improve operational efficiency.
- Peak Hours Analysis: Understanding customer behavior during peak hours helped us tailor marketing strategies and promotions to maximize sales and enhance customer satisfaction during busy periods.







THANK YOU !