

PIZZA SALES

SQL PROJECT





INTRODUCTION

In today's dynamic food industry, pizza remains a beloved staple, satisfying cravings and bringing people together for generations. As pizzerias continue to evolve, understanding and analyzing sales data is paramount for optimizing operations, enhancing customer satisfaction, and driving profitability.

This SQL project delves into the realm of pizza sales data analysis. Through structured queries and data manipulation, we aim to uncover valuable insights into sales trends, customer preferences, and operational efficiencies within a fictional pizzeria.

DATASET FILE



orders

order_details

pizza_type

pizza

KEY OBJECTIVES

- Sales Performance Analysis: Understanding overall sales trends, including peak hours, popular menu items, and revenue patterns.
- Customer Segmentation: Segmenting customers based on their ordering behavior, frequency, and preferences.
- Menu Optimization: Identifying underperforming menu items and opportunities for introducing new offerings.
- Operational Efficiency: Analyzing order fulfillment times, delivery performance, and staffing requirements.



QUESTION SET

Basic:

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.



Intermediate:

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

Advanced:

- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

BASIC SET QUESTIONS

```
-- 1.Total number of orders placed.
```

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

```
-- 2.Calculate total revenue generated from pizza sales.
```

```
SELECT round(SUM( order_details.quantity * pizzas.price),2) AS total_revenue  
FROM order_details  
JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id ;
```

```
-- 3.Identify the highest price pizza.
```

```
SELECT pizza_types.name AS pizza_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;
```

BASIC SET QUESTIONS

```
-- 4. Identify the most common pizza size ordered.  
SELECT pizzas.size, COUNT(order_details.order_details_id) AS order_count  
FROM pizzas  
JOIN order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizzas.size  
ORDER BY order_count DESC  
LIMIT 1;
```

```
-- 5. List the top 5 most ordered pizza types along with their quantities.  
SELECT pizza_types.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;
```

INTERMEDIATE SET QUESTIONS

```
/*Q1-Join the necessary tables to find the total quantities of each pizza 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 category
ORDER BY quantity DESC;
```

```
/*Q2-Determine the distribution of orders by hour of the day?*/
```

```
SELECT HOUR(time) , COUNT(order_id)
FROM orders
GROUP BY HOUR(time);
```

INTERMEDIATE SET QUESTIONS

```
/*Q3-Join relevant tables to find the category-wise distribution of pizzas.*/

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

/*Q4-Group the orders by date and calculate the average number of pizzas ordered per day.*/

SELECT ROUND(AVG(quantity),0) AS avg_pizza_per_day
FROM
(SELECT orders.date, SUM(order_details.quantity) AS quantity
FROM orders
JOIN order_details ON orders.order_id = order_Details.order_id
GROUP BY orders.date) AS order_quantity;

/*Q5-Determine the top 3 most pizza types based on Revenue.*/
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
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

ADVANCE SET QUESTIONS

```
/*Q1- Calculate the percentage contribution of each pizza type to total revenue.*/
-- each pizza price / total sales * 100
SELECT pizza_types.category,
(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 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.category
ORDER BY revenue DESC ;
```

ADVANCE SET QUESTIONS

```
/*Q2-Analyse the cumulative(prev+current) revenue generated overtime.*/
SELECT orders.date,
SUM(revenue) OVER(ORDER BY orders.date) AS cum_revenue
FROM
(SELECT orders.date,
SUM(order_details.quantity * pizzas.price) AS revenue
FROM order_details
JOIN order_details 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;

/*Q3-Determine the top 3 pizza typepizza_typeess based on revenue for each pizza category.*/
SELECT name, revenue,
(SELECT category , pizza_types.name , revenue,
RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS rn,
(SELECT pizza_types.category,pizza_types.name,
SUM((order_details.quantity) * pizzas.pizza_type_id) 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;
```

THANK YOU

