PIZZA BITE

CHEESE THE DAY WITH OUR PIZZAS

SINCE 1976

Introduction to Pizza Sales SQL Project

This presentation offers an **analytical overview** of the **Pizza Sales SQL Project**. We will explore the **strategies** implemented and the **insights** gained through data analysis. Understanding these components is crucial for enhancing sales performance and customer satisfaction.

Project Objectives

The primary **objectives** of this project include analyzing sales trends, identifying customer preferences, and optimizing inventory management. By achieving these goals, we aim to improve **operational efficiency** and **increase profitability**.

Data Collection Methods

KAGGLE DATASET that we gathered **reliable** and **relevant** data for our SQL analysis.

SQL Queries and Analysis

The project employed complex **SQL queries** to extract meaningful insights from the dataset. Key analyses included sales performance by **location**, customer demographics, and seasonal trends, which were crucial for strategic decision-making.

Retrieve the total number of orders placed.

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

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.

```
select
round(sum(order_details.quantity*pizzas.price),2) as
Totalsales from order_details join pizzas on
pizzas.pizza_id = order_details.pizza_id
```

-- identify the Highest price 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;
```

- identify the most common pizza size orderd

```
SELECT
pizzas.size,COUNT(order_details.order_id) FROM pizzas JOIN
order_details ON pizzas.pizza_id = order_details.pizza_id GROUP
BY pizzas.size
```

- - list the top 5 most ordered pizza types along with their types

```
  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;
```

-- JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

```
 SELECT
pizza_types.category(SUM(order_details.quantity))
FROM pizza_types
JOIN pizzas ON pizza_types.pizza_types_id = pizzas.pizza_types_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),
count(order_id) as order_count from
orders group by hour( order_time);
```

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

```
select category, count(name)
from pizza_types
 group by category
```

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

```
SELECT round( AVG(quantity),2) FROM
 (SELECT orders.orders_date,
SUM(order_details.quantity)
 FROM orders
JOIN  order_details  ON  orders.orders_id  =  order_details.order_details_id
GROUP BY orders.orders_date)
```

-- Determine the top 3 most ordered 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
JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

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

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

-- Analyze the cumulative revenue generated over time.

```
select order_date, sum(revenue)
over (order by order_date)
as cum_revenue
from
(select orders.orders_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, SUM(order_details.quantity * pizzas.price)
AS revenue, RANK()
OVER (PARTITION BY category
ORDER BY SUM(order_details.quantity * pizzas.price) DESC) AS rn
 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 ranked_data
WHERE rn <= 3;
```

Insights Gained

Through our analysis, we uncovered several **insights** such as peak sales periods, popular pizza toppings, and customer buying patterns. These findings are essential for tailoring marketing strategies and enhancing customer engagement.

Strategic Recommendations

Based on our findings, we recommend implementing targeted **marketing campaigns**, optimizing the menu according to customer preferences, and improving **inventory management** to reduce waste. These strategies will help maximize revenue and customer satisfaction.

Conclusion and Future Work

In conclusion, the **Pizza Sales SQL Project** has provided valuable insights and strategic recommendations. Future work will focus on continuous data monitoring, refining analysis techniques, and exploring new technologies to further enhance decision-making processes.

Thanks!