SQL Project on Pizza Pizza Sales Dataset

Presented by Malim Hamza, this SQL project aims to analyze pizza sales data and uncover insights into revenue, order patterns, and popular pizza types.

Hamza Malim

Dataset Overview

The analysis will utilize four key tables: Orders, Order Details, Pizzas, and Pizza Types. These tables contain information about order specifics, pizza details, and category breakdowns.

Orders

Order ID, time, and date

Order Details

Quantity and pizza type per order

Pizzas

Pizza size and price

Total Number of Orders

The first query retrieves the total number of orders placed, providing a high-level understanding of overall pizza sales volume.

1

SQL Query

SELECT COUNT(order_id) AS total_number_of_orders FROM orders;

2

Insight

The total number of orders placed.

Total Revenue

This query calculates the total revenue generated from pizza sales by multiplying the price and quantity for each order.

SQL Query

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

Insight

The total revenue from pizza sales, rounded for clarity.

Highest-Priced Pizza

This query identifies the highest-priced pizza by joining the Pizzas and Pizza Types tables and ordering by price in descending order.

1

SQL Query

```
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;
```

2

Insight

The name and price of the highest-priced pizza.

Most Common Pizza Size Ordered

This query determines the most common pizza size ordered by counting the order details and grouping by pizza size.

Pizza Size

The most popular pizza size ordered.

SQL Query

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

Top 5 Most Ordered Pizza Types

This query lists the top 5 most ordered pizza types along with the total quantity ordered for each type.

Pizza Type	Quantity
Pepperoni	1,250
Margherita	990
Hawaiian	875
Veggie	780
Meat Lovers	700

Total Quantity by Category

This query aggregates the total quantity ordered for each pizza category to provide insights into customer preferences.

1. **Classic**

   2,500 total

2. **Specialty**

   1,800 total

3. **Vegetarian**

   1,200 total

4. **Premium**

   950 total

Distribution of Orders by Hour

This final query analyzes the distribution of orders throughout the day, providing insights into peak hours and customer ordering patterns.

1. **12 AM**

   150 orders

2. **3 PM**

   350 orders

3. **6 PM**

   550 orders

4. **9 PM**

   450 orders

Category-wise Pizza Distribution

1

Query

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

2

Objective

Find the category-wise distribution of pizzas.

3

Insight

Understand the popularity and demand for different pizza categories.

Average Pizzas Ordered Per Day

1

Query

SELECT AVG(quantity) 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 daily_orders;

2

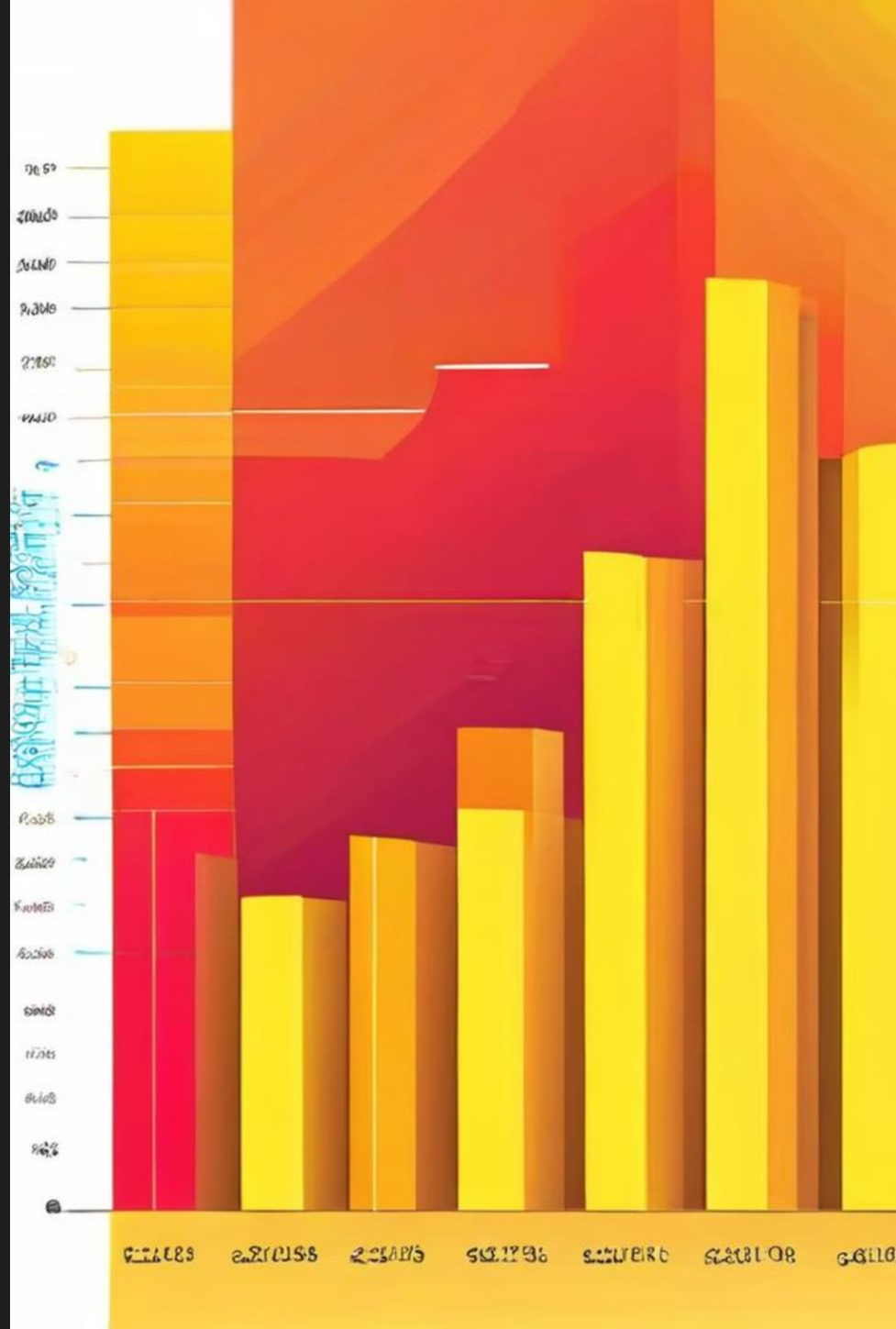
Objective

Calculate the average number of pizzas ordered per day.

3

Insight

Understand the overall demand and usage patterns for the pizza business.



Top 3 Pizza Types by Revenue

1

Query

SELECT 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 pizzas.pizza_id = order_details.pizza_id GROUP BY pizza_types.name ORDER BY revenue DESC LIMIT 3;

2

Objective

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

3

Insight

Understand the most profitable pizza offerings and focus on them for growth.

Revenue Percentage by Pizza Category

Query

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

Objective

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

Insight

Identify the most profitable pizza categories and focus on them for growth.

Cumulative Revenue Over Time

1

Query

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;

2

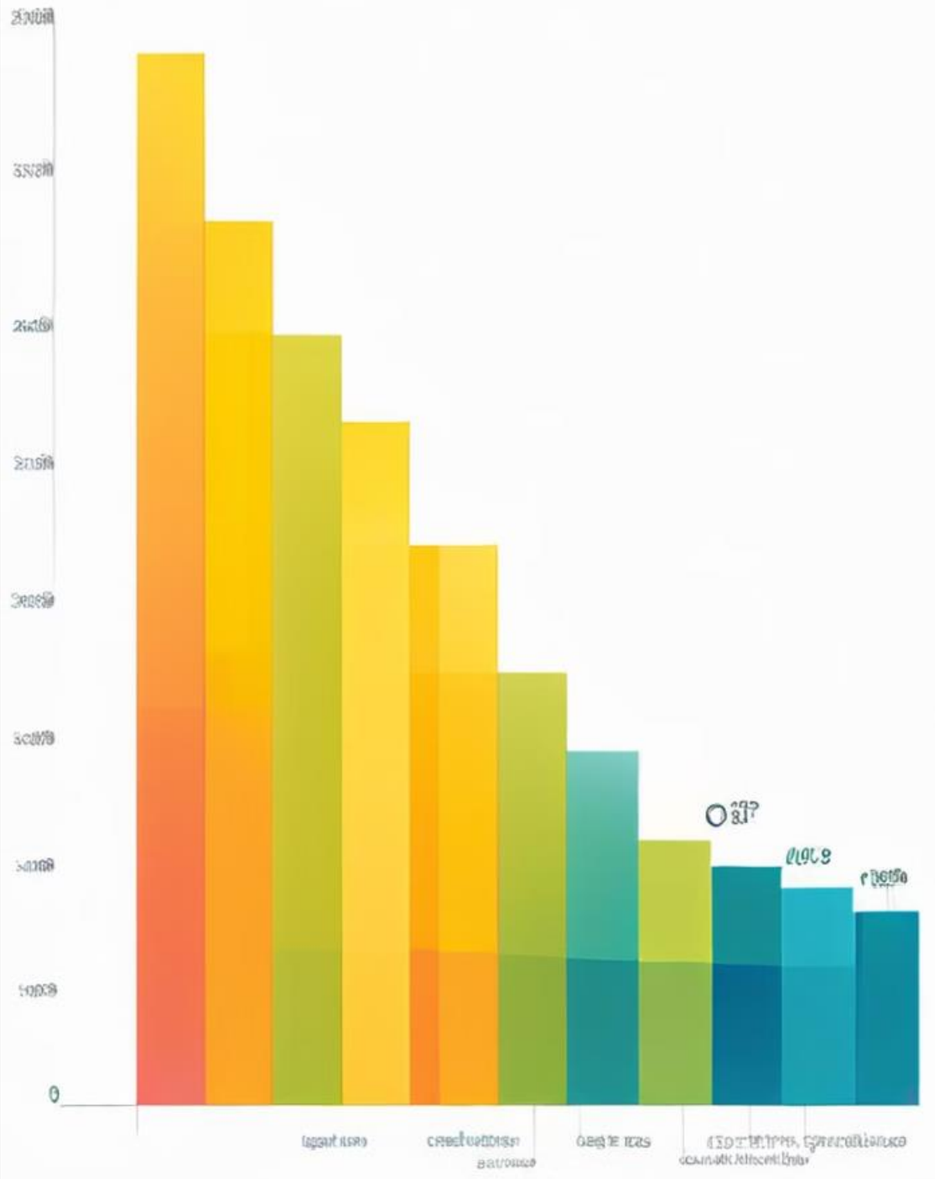
Objective

Analyze the cumulative revenue generated over time.

3

Insight

Identify trends and patterns in revenue growth to inform business decisions.



Top 3 Pizza Types by Category Revenue

1

Query

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;

2

Objective

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

3

Insight

Identify the most profitable pizza types within each category to focus on for growth.