Pizza Sales

Analysis

PROJECT SUMMARY

Analysis of Pizza Sales Data

This project involves a comprehensive analysis of a dataset encompassing pizza sales. The dataset is structured across four key tables:

- Orders Table: Captures the date and time of each pizza order, serving as the primary record for tracking sales.
- Order Details Table: Includes specific details like order ID, pizza ID, and the quantity of pizzas ordered, linking individual orders to their contents.
- Pizza Types Table: Details each type of pizza available, categorized by a unique type ID, the name of the pizza, its category, and the ingredients used.
- Pizza Table: Describes each pizza with attributes such as pizza ID, type ID, size, and price.

The analysis aims to derive insights into sales patterns, customer preferences, and pricing strategies. Key metrics evaluated include sales volume by date and time, popularity of different pizza types, and revenue generation based on pizza sizes and pricing. These insights are crucial for optimizing the menu and pricing strategies to enhance customer satisfaction and profitability.

Retrieve the total number of orders placed

```sql
SELECT
    COUNT(*) As total_orders
FROM orders;
```

total_orders
21350

Total revenue generated from pizza sales

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

total_sales
817860.05

Highest-priced pizza

```sql
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;
```

name	price
The Greek Pizza	35.95

The most common pizza size ordered

```sql
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS pizza_count
FROM pizzas
JOIN orders_details
ON pizzas.pizza_id=orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY pizza_count DESC;
```

size	pizza_count
L	18526
M	15385
S	14137
XL	544
XXL	28

Top 5 most ordered pizza types along with their quantities

```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quantity) AS quantity
FROM
    orders_details
JOIN pizzas
ON pizzas.pizza_id= orders_details.pizza_id
JOIN pizza_types
ON pizza_types.pizza_type_id= pizzas.pizza_type_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

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

Total quantity of each pizza category ordered

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

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Distribution of orders by hour of the day

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

HOUR(order_time)	COUNT(order_id)
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
10	8
9	1

Category-wise distribution of pizzas

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

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

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

```sql
    WITH DailyOrderQuantities AS (
    SELECT
        o.order_date,
        SUM(od.quantity) AS quantity
    FROM
        orders o
    JOIN
        orders_details od
    ON
        o.order_id = od.order_id
    GROUP BY
        o.order_date
)
SELECT
    ROUND(AVG(quantity), 0) AS average_quantity
FROM
    DailyOrderQuantities;
```

average_quantity
138

Top 3 most ordered pizza types based on revenue.

```sql
SELECT
    pizza_types.name,
    ROUND(SUM(orders_details.quantity * pizzas.price),2) AS revenue
FROM
    orders_details
JOIN pizzas
ON pizzas.pizza_id= orders_details.pizza_id
JOIN pizza_types
ON pizzas.pizza_type_id=pizza_types.pizza_type_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

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

Percentage contribution of each pizza type to total revenue.

```sql
-- Calculate the total sales from all orders first
WITH TotalSales AS (
    SELECT
        ROUND(SUM(od.quantity * p.price), 2) AS total_sales
    FROM
        orders_details od
        JOIN pizzas p ON p.pizza_id = od.pizza_id
)
-- Calculate the revenue share of each pizza category
SELECT
    pt.category,
    ROUND(SUM(od.quantity * p.price) / (SELECT total_sales FROM TotalSales) * 100, 2) AS revenue_
FROM
    pizza_types pt
    JOIN pizzas p ON p.pizza_type_id = pt.pizza_type_id
    JOIN orders_details od ON od.pizza_id = p.pizza_id
GROUP BY
    pt.category
ORDER BY
    revenue_percentage DESC;
```

category	revenue_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Cumulative revenue generated over time

```sql
WITH RevenueByDate AS (
    SELECT
        o.order_date,
        SUM(od.quantity * p.price) AS revenue
    FROM
        orders_details od
        JOIN pizzas p ON od.pizza_id = p.pizza_id
        JOIN orders o ON o.order_id = od.order_id
    GROUP BY
        o.order_date
)
SELECT
    order_date,
    SUM(revenue) OVER (ORDER BY order_date) AS cum_revenue
FROM
    RevenueByDate
ORDER BY
    order_date;
```

order_date	cum_revenue
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

Top 3 most ordered pizza types based on revenue for each pizza category

```sql
WITH RevenueCTE AS (
    SELECT
        pt.category,
        pt.name,
        SUM(od.quantity * p.price) AS revenue
    FROM
        pizza_types pt
        JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN orders_details od ON od.pizza_id = p.pizza_id
    GROUP BY
        pt.category,
        pt.name
)
SELECT
    category,
    name,
    revenue,
    RANK() OVER (PARTITION BY category ORDER BY revenue DESC) AS rn
FROM
    RevenueCTE
ORDER BY
    category, rn;
```

category	name	revenue	rn
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Chicken	The Southwest Chicken Pizza	34705.75	4
Chicken	The Chicken Alfredo Pizza	16900.25	5
Chicken	The Chicken Pesto Pizza	16701.75	6
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Classic	The Greek Pizza	28454.100000000013	4
Classic	The Italian Capocollo Pizza	25094	5
Classic	The Napolitana Pizza	24087	6

Conclusion

Key Insights from Our Pizza Database Analysis:

Sales Performance:

- Identified the total revenue generated from pizza sales, providing insights into overall business health and customer spending patterns.

Customer Preferences:

- The most popular pizza sizes and types were highlighted, indicating customer favorites and potential areas for menu expansion.
- The highest-priced pizza and its demand offered insights into the pricing strategy’s effectiveness.

Operational Efficiency:

- Analysis of order distributions by time and day provided valuable data to optimize staffing and operational hours.
- Average daily orders helped assess daily performance and customer flow.

Strategic Recommendations

- Focus on high-revenue generating pizzas for promotional activities.
- Adjust inventory and staffing according to peak order times to enhance service efficiency.
- Consider menu adjustments based on popular pizza sizes and types to better meet customer demand.

Moving Forward:

- Continuously track and analyze customer preferences and sales data to stay responsive to market changes.
- Utilize these insights to drive marketing strategies, menu design, and business operations.