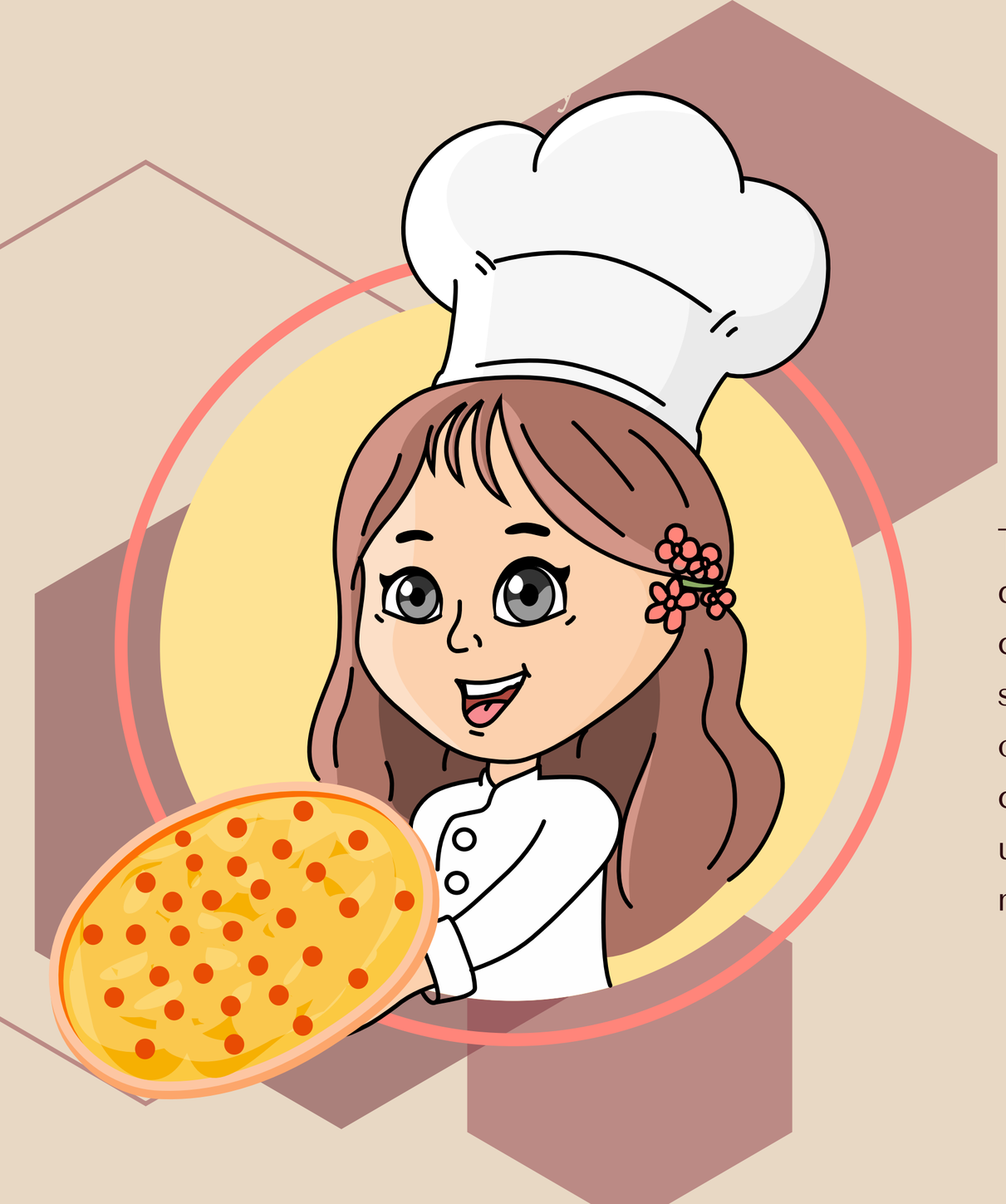
SQL Data Analysis Project: Pizza Sales

Khushi Kumari

INTRODUCTION

In today's data-driven business environment, the ability to analyze and interpret data is crucial for success. This project aims to leverage SQL to conduct a comprehensive analysis of pizza sales data, providing actionable insights into order patterns, revenue streams, and customer preferences. By writing a series of SQL queries, we explore different facets of the pizza business, such as identifying the top-selling pizza types, analyzing order distribution by time, and calculating the contribution of individual pizzas to total revenue. The results of this analysis can be used to enhance business strategies, drive sales growth, and improve operational efficiency.



OBJECTIVE

The objective of this project is to analyze pizza sales data using SQL queries to extract valuable insights that can help optimize business operations, improve customer satisfaction, and maximize revenue. The project focuses on key aspects such as order volume, revenue analysis, customer preferences, and time-based ordering patterns, ultimately guiding decision-making for inventory management, pricing strategies, and product promotion.

Retrieve the total number of orders placed.

```sql
-- Retrieve the total number of orders placed.

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

Result Grid

total_orders
21350

Calculate the total revenue generated from pizza sales.

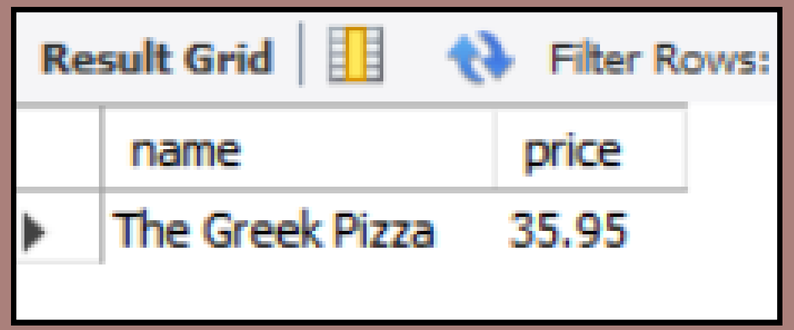
```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

Result Grid

	total_orders
▶	21350

Identify the 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;
```



Identify the most common pizza size ordered.

```sql
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;
```

Result Grid | Filter

size	Order_count
L	18526

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

```sql
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;
```

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 R

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

Result Grid | Filter

hour	order_count
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

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

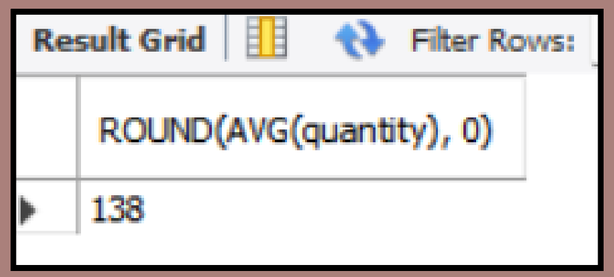
```sql
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Row

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
SELECT
    ROUND(AVG(quantity), 0)
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;
```



Determine the top 5 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 5;
```

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Spicy Italian Pizza	34831.25

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_sale
                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 Rows:

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

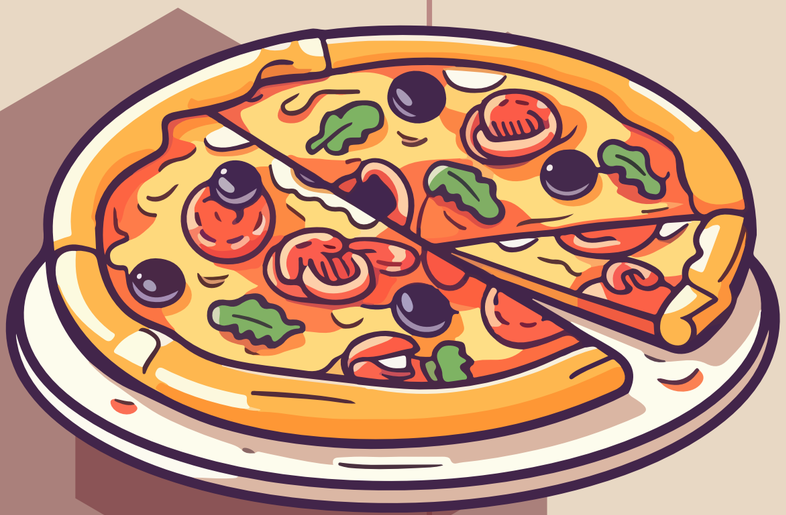
Analyze the cumulative revenue generated over time.

```sql
select order_date,
sum(revenue) over(order by order_date) as cum_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;
```

Result Grid | Filter Rows:

order_date	cum_revenue
2015-01-01	2713.850000000004
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
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.600000000006
2015-01-24	55013.850000000006
2015-01-25	56631.40000000001

KEY INSIGHTS



Total Orders & Revenue:

A clear understanding of total orders placed and revenue generated from pizza sales.

Category Performance:

Revealed which pizza categories (e.g., vegetarian, meat) have the highest order volumes.

Top Pizzas by Revenue:

Highlighted the top 5 pizzas generating the most revenue.

Cumulative Growth:

Tracked how revenue accumulates over time, giving insight into long-term performance trends.

Customer Preferences:

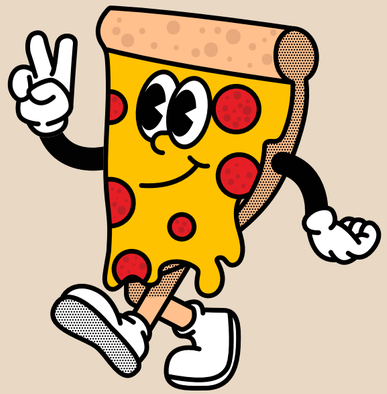
Identified the most common pizza size and the top 5 most ordered pizza types.

Order Patterns:

:Found peak ordering times by analyzing order distribution by hour.

Revenue Contribution:

Calculated each pizza's percentage contribution to total revenue, identifying key revenue drivers.



CONCLUSION

This SQL data analysis project provides a detailed understanding of pizza sales trends, customer preferences, and revenue generation. By analyzing key metrics such as total orders, revenue, and the popularity of different pizza types and sizes, we gained valuable insights into the business's performance. The intermediate and advanced analyses further deepened our understanding by revealing order patterns across time, category-wise distributions, and revenue contributions of specific pizzas.

These insights can be used to make data-driven decisions to improve inventory management, optimize staffing schedules, adjust pricing strategies, and target marketing efforts for high-revenue products. Ultimately, this project demonstrates the power of SQL in transforming raw data into meaningful information that can drive business growth and operational efficiency.

THANK YOU