Where Every Slice is a Taste of Perfection

PIZZA SALES

ORDER NOW

SQL Project

INTRODUCTION

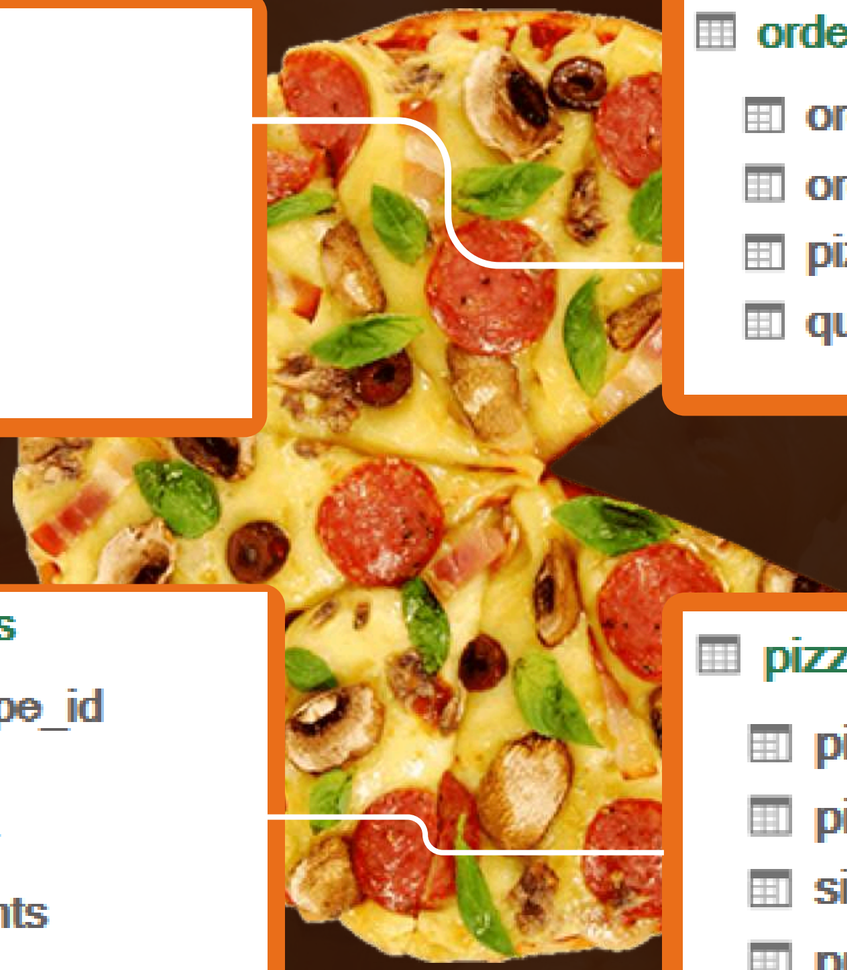
In the highly competitive food industry, data plays a crucial role in understanding customer preferences and optimizing business operations. This project focuses on a pizza sales dataset and utilizes SQL to perform in-depth data analysis.

By querying and visualizing key metrics such as revenue, order frequency, and product performance, the project aims to uncover actionable insights that can help improve sales strategies, enhance customer satisfaction, and streamline operations for a pizza business.

SCHEMA

orders
- order_id
- date
- time

order_details
- order_details_id
- order_id
- pizza_id
- quantity



pizza_types
- pizza_type_id
- name
- category
- ingredients

pizzas
- pizza_id
- pizza_type_id
- size
- price

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

ORDER NOW

```sql
SELECT
    COUNT(order_id) AS Total_Orders
FROM
    orders;
```

Total_Orders
21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```sql
SELECT
    ROUND(SUM(pizzas.price * order_details.Quantity),
            2) AS Total_Revenue
FROM
    pizzas
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id;
```

Total_Revenue
817860.05

Pepperoni Pizza

IDENTIFY THE HIGHEST-PRICED PIZZA WITH PIZZA NAME.

```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

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed at ipsum vitae lacus lobortis lacinia. Donec tristique arcu massa, at pharetra tortor feugiat non.

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

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

Olivia Wilson

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed at ipsum vitae lacus lobortis lacinia. Donec tristique arcu massa, at pharetra tortor feugiat non.

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```sql
SELECT pizza_types.name,
SUM(order_details.Quantity) AS qty
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 qty DESC LIMIT 5;
```

name	qty
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

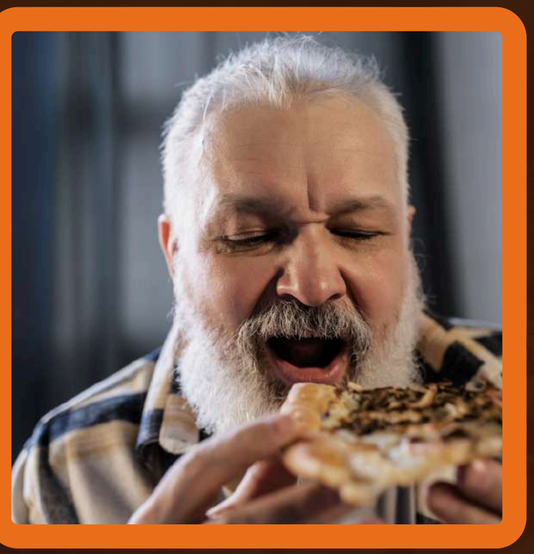
DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    HOUR(order_time) AS order_hours,
    COUNT(order_id) AS order_count
FROM
    orders
GROUP BY order_hours
ORDER BY order_hours;
```

order_hours	order_count
9	1
10	8
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

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE OF TOTAL NUMBER OF PIZZAS ORDERED PER DAY.

Avery Davis



★★★★☆

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed at ipsum vitae lacus lobortis lacinia. Donec tristique arcu massa, at pharetra tortor feugiat non.

```sql
SELECT
FLOOR(AVG(total_qty)) AS avg_order_qty
FROM
    (SELECT
        orders.order_date, SUM(order_details.Quantity) AS total_qty
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS qty;
```

avg_order_qty
138

DETERMINE THE TO 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
SELECT pizza_types.name,
    SUM(pizzas.price * order_details.Quantity) 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;
```

Veggie Delight

$30

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

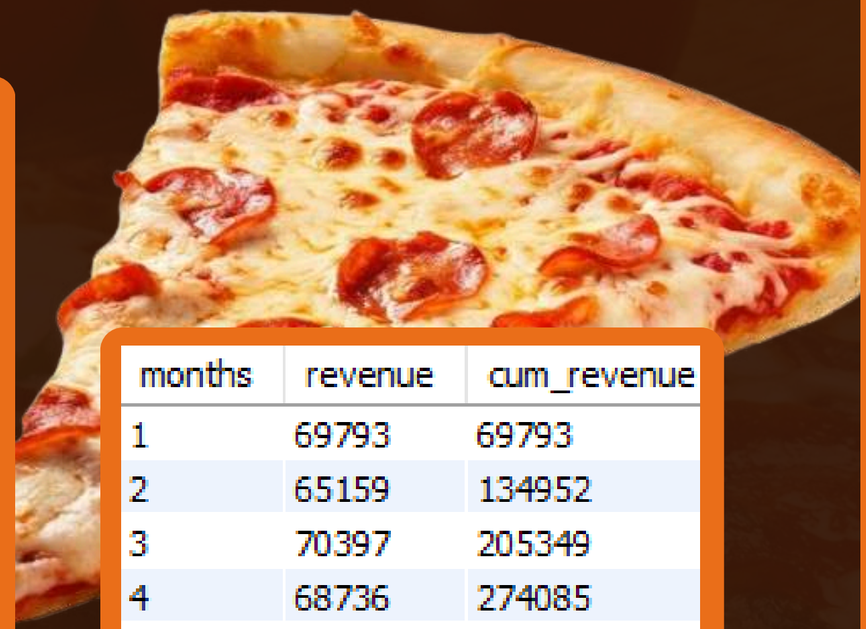
CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA CATEGORY TYPE TO TOTAL REVENUE.

```sql
SELECT
    pizza_types.category,
    CONCAT(ROUND(((SUM(pizzas.price * order_details.Quantity) * 100) / (SELECT
                            SUM(pizzas.price * order_details.Quantity)
                        FROM
                            pizzas
                                JOIN
                            order_details ON pizzas.pizza_id = order_details.pizza_id)),
                    2),
            '%') AS Percentage_contribution
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.category;
```

category	Percentage
Classic	26.91%
Veggie	23.68%
Supreme	25.46%
Chicken	23.96%

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER MONTHLY.

```
SELECT months,revenue,SUM(revenue)
OVER(ORDER BY months) AS cum_revenue
FROM
(SELECT MONTH(orders.order_date) AS months,
FLOOR(SUM(price*order_details.Quantity)) AS revenue
FROM orders
JOIN order_details
ON orders.order_id = order_details.order_id
JOIN pizzas
ON order_details.pizza_id = pizzas.pizza_id
GROUP BY months) AS revenue_collect;
```



months	revenue	cum_revenue
1	69793	69793
2	65159	134952
3	70397	205349
4	68736	274085
5	71402	345487
6	68230	413717
7	72557	486274
8	68278	554552
9	64180	618732
10	64027	682759
11	70395	753154
12	64701	817855

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```sql
SELECT ranks,category,name,revenue
FROM
(SELECT category,name,revenue,
RANK() over(PARTITION BY category ORDER BY revenue DESC) AS ranks
FROM
    (SELECT pizza_types.category, pizza_types.name,
FLOOR(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.category,pizza_types.name) AS t1) AS t2
WHERE ranks <=3;
```

category	name	revenue
Chicken	The Thai Chicken Pizza	43434
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409
Classic	The Classic Deluxe Pizza	38180
Classic	The Hawaiian Pizza	32273
Classic	The Pepperoni Pizza	30161
Supreme	The Spicy Italian Pizza	34831
Supreme	The Italian Supreme Pizza	33476
Supreme	The Sicilian Pizza	30940
Veggie	The Four Cheese Pizza	32265
Veggie	The Mexicana Pizza	26780
Veggie	The Five Cheese Pizza	26066

CONCLUSION

Through this project, SQL was effectively used to explore and analyze a comprehensive pizza sales dataset. The analysis revealed key insights such as the most popular pizza types, high-performing categories, peak sales periods, and revenue patterns across different sizes and ingredients. These findings can support data-driven strategies to optimize the menu, enhance customer satisfaction, and boost overall sales performance.

This project demonstrates how SQL can transform raw sales data into actionable insights, empowering businesses in the food industry to make smarter, more informed decisions.

THANK YOU