Pizza Sales Analysis using SQL

Exploring sales, revenue, and customer trends with SQL queries

By

Amir Khan

Introduction

This project analyzes a pizza sales dataset using SQL to uncover business insights. The analysis covers sales performance, revenue generation, customer ordering patterns, and product popularity

Objective:

- Understand sales trends
- Identify best-selling products
- Analyze revenue distribution

About the Dataset

This project uses four CSV files:

- 🧾 orders.csv → order_id, order_date, order_time
- 📊 order_details.csv → order_id, pizza_id, quantity
- 🍕 pizzas.csv → pizza_id, pizza_type_id, size, price
- 🧂 pizza_types.csv → pizza_type_id, name, category, ingredients

Dataset Size:

- orders.csv → 21,350 rows, 3 columns
- order_details.csv → 48,620 rows, 3 columns
- pizzas.csv → 97 rows, 4 columns
- pizza_types.csv → 32 rows, 4 columns

Together, these datasets provide details about orders, pizza types, prices, sizes, and ingredients, enabling a complete sales analysis.

Skills & Tools Used

SQL Concepts Applied

- Joins: INNER JOIN to combine multiple tables
- Aggregations: SUM, COUNT, AVG, ROUND
- Grouping & Ordering: GROUP BY, ORDER BY, LIMIT
- Window Functions: ROW_NUMBER, cumulative SUM
- Subqueries & CTEs for complex analysis

Tools Used

- 🖥️ MySQL Workbench → Querying & data analysis
- 🎨 Canva → Presentation & visualization

Project Workflow

Data Exploration

- Reviewed dataset structure (orders, pizzas, details, types)
- Understood relationships between tables

Writing SQL Queries

- Basic analysis (orders, revenue, top pizzas)
- Intermediate joins & groupings (categories, time-based analysis)
- Advanced analysis (revenue contribution, ranking with window functions)

Insights & Analysis

Identified sales trends, popular pizzas, and revenue drivers

Presentation

- Summarized results in a structured format
- Used Canva for visualization and storytelling

Query & Result section

Total number of orders placed

```sql
-- Retrieve the total number of orders placed.
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

MYSQL Query

Result

Result Grid | Filter Rows:

total_orders
21350

Total revenue generated from pizza sales

```sql
-- Calculate the total revenue generated from pizza sales.
SELECT
    ROUND(SUM(od.quantity * p.price), 2) AS total_sales
FROM
    order_details od
        JOIN
    pizzas p ON p.pizza_id = od.pizza_id;
```

MYSQL Query

Result

Result Grid

total_sales
817860.05

Highest-priced pizza

```sql
-- Identify the highest-priced pizza.
SELECT
    pt.name, p.price
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
ORDER BY p.price DESC
LIMIT 1;
```

MYSQL Query

Result

Result Grid Filter Rows:

name	price
The Greek Pizza	35.95

Most common pizza size ordered

```sql
-- Identify the most common pizza size ordered.
SELECT
    p.size, COUNT(od.order_details_id) AS order_count
FROM
    pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY p.size
ORDER BY order_count DESC;
```

MYSQL Query

Result

Result Grid Filter Rows:

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

Top 5 most ordered pizza types along with their quantities

```sql
-- List the top 5 most ordered pizza types along with their quantities.
SELECT
    pt.name, SUM(od.quantity) AS quantity
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY quantity DESC
LIMIT 5;
```

MYSQL Query

Result

Result Grid | Filter Rows: Search

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

The total quantity of each pizza category ordered

```mysql
SELECT
    pt.category, SUM(od.quantity) AS quantity
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.category
ORDER BY quantity DESC;
```

MYSQL Query

Result

Result Grid   Filte

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Distribution of orders by hour of the day

```mysql
-- Determine the distribution of orders by hour of the day.
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time)
ORDER BY hour;
```

MYSQL Query

Result

Result Grid Filter Ro

hour	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

Category-wise distribution of pizzas

```
2 •   SELECT
3         category, COUNT(name)
4     FROM
5         pizza_types
6     GROUP BY category;
```

MYSQL Query

Result

Result Grid | Filter

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

Calculate the average number of pizzas ordered per day

```mysql
SELECT
    round(AVG(quantity),0) "avg_no_pizzas_ord/day"
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS quantity
    FROM
        orders o
    JOIN order_details od ON o.order_id = od.order_id
    GROUP BY o.order_date) AS order_quantity
```

MYSQL Query

Result

Result Grid   Filter Rows:

avg_no_pizzas_ord/day
138

Top 3 most ordered pizza types based on revenue

```sql
-- Determine the top 3 most ordered pizza types based on revenue.
SELECT
    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
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY revenue DESC
LIMIT 3;
```

MYSQL Query

Result

Result Grid | Filter Rows: Search

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 percentage contribution of each pizza type to total revenue.
SELECT
    pt.category,
    ROUND(SUM(od.quantity * p.price) / (SELECT
                    ROUND(SUM(od.quantity * p.price), 2) AS total_sales
                FROM
                    order_details od
                        JOIN
                    pizzas p ON p.pizza_id = od.pizza_id) * 100,
            2) AS revenue_pct
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.category
ORDER BY revenue_pct DESC;
```

Result

Result Grid Filter Rows:

category	revenue_pct
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

MYSQL Query

Cumulative revenue generated over time

```sql
-- Analyze the cumulative revenue generated over time.
select order_date, sum(revenue) over(order by order_date) from
(select o.order_date,
sum(od.quantity*p.price) as revenue
from order_details od
join pizzas p
on od.pizza_id=p.pizza_id
join orders o
on od.order_id=o.order_id
group by o.order_date) as sales;
```

MYSQL Query

Result

Result Grid Filter Rows: Search

order_date	sum(revenue) over(order by order_d...
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

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

```sql
select name, revenue, rnk from
(select category, name, revenue, rank()
over(partition by category order by revenue desc)
as rnk from
(SELECT
    pt.category, pt.name, SUM(od.quantity * p.price) AS revenue
FROM
    pizza_types pt
        JOIN
    pizzas p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category , pt.name) as a) as b
where rnk <=3 ;
```

Result

Result Grid | Filter Rows: Search | Expo

name	revenue	rnk
The Thai Chicken Pizza	43434.25	1
The Barbecue Chicken Pizza	42768	2
The California Chicken Pizza	41409.5	3
The Classic Deluxe Pizza	38180.5	1
The Hawaiian Pizza	32273.25	2
The Pepperoni Pizza	30161.75	3
The Spicy Italian Pizza	34831.25	1
The Italian Supreme Pizza	33476.75	2
The Sicilian Pizza	30940.5	3
The Four Cheese Pizza	32265.70000...	1
The Mexicana Pizza	26780.75	2
The Five Cheese Pizza	26066.5	3

MYSQL Query

Conclusion

📊 SQL enabled efficient analysis of large sales data.
🍕 Identified top-selling pizzas and most profitable categories.
💰 Clear view of revenue distribution across products and time.
🕒 Ordering patterns reveal customer behavior and peak hours.
✅ Insights can help optimize menu, pricing, and inventory planning.

Overall:
This project highlights how SQL-driven insights can support data-driven decision making in the food & beverage industry.

Thank You