

PIZZA SALES SQL ANALYSIS PROJECT

END-TO-END PORTFOLIO PROJECT USING MYSQL WORKBENCH



By Ayushi Bhati

-  Tools Used: MySQL Workbench, SQL, Canva
-  GitHub: github.com/codeWithAyushi-collab

PROJECT OVERVIEW



Dataset:

- *Used a pizza sales dataset with 4 CSV files:*
- *orders.csv, order_details.csv, pizzas.csv, pizza_types.csv*
- *Created tables in MySQL Workbench & imported data*
- *Structured relational database using appropriate keys*



Objective:

To perform end-to-end SQL-based data analysis on pizza sales to uncover trends, top-selling items, customer behavior, and revenue insights.



Approach:

- *Built database schema manually + imported data*
- *Executed 13 SQL queries (Basic, Intermediate & Advanced levels)*
- *Answered key business questions using joins, groupings, aggregation & window functions*
- *Results were compiled and visualized in a presentation-ready format*



UNDERSTANDING THE PIZZA SALES DATASET

The Dataset Includes 4 CSV Files:

orders.csv — order ID, date, time

order_details.csv — order ID, pizza ID, quantity

pizzas.csv — pizza ID, type ID, size, price

pizza_types.csv — type ID, name, category, ingredients

Tables Created in MySQL:

orders (primary key: *order_id*)

order_details (primary key: *order_details_id*, foreign key: *order_id*)

pizzas (primary key: *pizza_id*, foreign key: *pizza_type_id*)

pizza_types (primary key: *pizza_type_id*)

Schema Relations:

Each order can contain multiple pizzas

Pizza IDs link between order_details and pizzas

Pizza types provide info like ingredients & category

orders → order_details → pizzas → pizza_types



BASIC SQL ANALYSIS – BUSINESS INSIGHTS



Purpose:

To understand order volume, revenue, and popular choices using simple aggregate and sort queries.



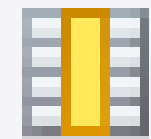
Questions Covered:

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.

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_sales
▶	817860.05

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

```
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: <input type="text"/>		
	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



INTERMEDIATE SQL ANALYSIS – DEEPER BUSINESS INSIGHTS



Objective:

To gain deeper insights into time-based trends, category performance, and daily order behaviors using grouping, joins, and date functions.



Questions Covered:

Join the necessary tables to find the total quantity of each pizza category ordered.

Determine the distribution of orders by hour of the day.

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

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

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

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```
• 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
	category	quantity	
▶	Classic	14888	
	Supreme	11987	
	Veggie	11649	
	Chicken	11050	

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);
```

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	

● ADVANCED SQL ANALYSIS – BUSINESS INSIGHTS

Objective

To derive strategic-level insights by analyzing revenue distribution and trends over time using cumulative functions, percentages, and advanced groupings.

Key Questions Explored

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

Analyze the cumulative revenue generated over time.

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

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

```
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
        SUM(order_details.quantity * pizzas.price)
        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
	category	revenue	
▶	Classic	26.91	
	Supreme	25.46	
	Chicken	23.96	
	Veggie	23.68	

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
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	2713.85000000000004	
	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	23990.3500000000002	
	2015-01-11	25862.65	
	2015-01-12	27781.7	
	2015-01-13	29831.3000000000003	
	2015-01-14	32358.7000000000004	
	2015-01-15	34343.500000000001	
	2015-01-16	36937.650000000001	
	2015-01-17	39001.750000000001	
	2015-01-18	40978.6000000000006	
	2015-01-19	43365.750000000001	
	2015-01-20	45763.650000000001	

THANK YOU FOR VIEWING MY SQL PROJECT!

✨ Project Summary

- ◆ *Project Title: Pizza Sales Analysis using SQL*
 - ◆ *Tools Used: MySQL Workbench, Canva*
 - ◆ *Dataset: Pizza Sales Dataset (from GitHub)*
- ◆ *Queries Written: 13 (Basic, Intermediate, Advanced)*

◆ Skills Showcased:

Data Extraction & Filtering
Joins & Aggregations
Analytical Query Writing
Business Insight Generation
Data Presentation