# 🍕 SQL DATA ANALYSIS PROJECT – PIZZA SALES INSIGHTS USING MYSQL

Presented by Tanay Jujarao

# 🧩 PROJECT OVERVIEW

This project is an in-depth analysis of a year's worth of sales data from a fictional pizza restaurant. The primary goal is to use SQL to query the database and extract actionable insights.
The analysis is structured around 13 key business questions, which progress in difficulty from Basic (simple aggregations) to Intermediate (joins and groupings) and finally to Advanced (window functions, cumulative calculations, and complex cohort analysis). This project demonstrates a strong understanding of relational database querying, data aggregation, and business-focused problem-solving. I’ve attached 3 advanced and 3 intermediate questions to showcase the project results.

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

```sql
SELECT category,name,revenue,ranking FROM
(SELECT category,name,revenue,RANK() OVER (PARTITION BY category ORDER BY revenue DESC ) AS ranking
FROM
(SELECT pizza_types.category,pizza_types.name,sum(orders_details.quantity*pizzas.price) AS revenue
FROM pizza_types
JOIN pizzas
ON pizza_types.pizza_type_id=pizzas.pizza_type_id
JOIN orders_details
ON orders_details.pizza_id=pizzas.pizza_id
GROUP BY pizza_types.category,pizza_types.name) AS a) AS b
WHERE ranking<4
;
```

Result Grid | Filter Rows: | Export: | Wrap C

| category | name | revenue | ranking |
|---|---|---|---|
| Chicken | The Thai Chicken Pizza | 43434.25 | 1 |
| Chicken | The Barbecue Chicken Pizza | 42768 | 2 |
| Chicken | The California Chicken Pizza | 41409.5 | 3 |
| Classic | The Classic Deluxe Pizza | 38180.5 | 1 |
| Classic | The Hawaiian Pizza | 32273.25 | 2 |

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

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(orders_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(orders_details.quantity * pizzas.price),
                                2) AS total_sales
                FROM
                    orders_details
                        JOIN
                    pizzas ON orders_details.pizza_id = pizzas.pizza_id) * 100,
            2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

Result Grid | Filt

| category | revenue |
|---|---|
| Classic | 26.91 |
| Supreme | 25.46 |
| Chicken | 23.96 |
| Veggie | 23.68 |

# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

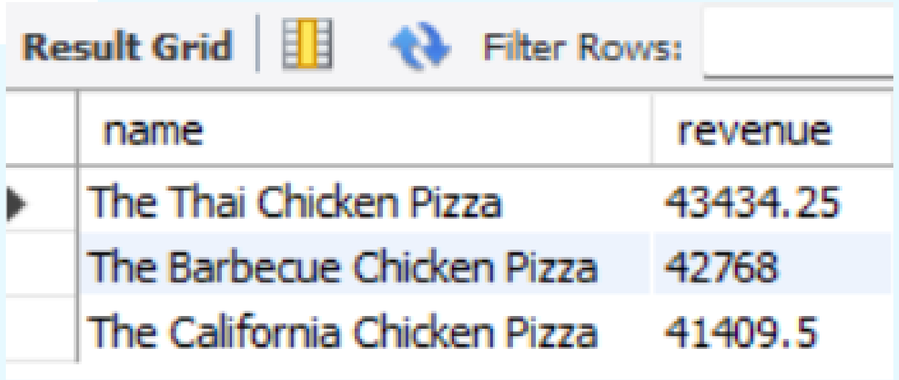
```
SELECT order_date,revenue,
SUM(revenue) OVER (ORDER BY order_date) AS cumulative_revenue FROM
(SELECT orders.order_date,round(sum(orders_details.quantity*pizzas.price),2) AS revenue
FROM orders
JOIN orders_details
ON orders.order_id=orders_details.order_id
JOIN pizzas
ON pizzas.pizza_id=orders_details.pizza_id
GROUP BY orders.order_date) AS sales;
```

Result Grid | Filter Rows:

| order_date | revenue | cumulative_revenue |
|---|---|---|
| 2015-01-01 | 2713.85 | 2713.85 |
| 2015-01-02 | 2731.9 | 5445.75 |
| 2015-01-03 | 2662.4 | 8108.15 |
| 2015-01-04 | 1755.45 | 9863.6 |
| 2015-01-05 | 2065.95 | 11929.55 |

# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_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 |

# DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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

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 |

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

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

Result Grid | Filter Rows:

| category | SUM(orders_details.quantity) |
|---|---|
| Classic | 14888 |
| Veggie | 11649 |
| Supreme | 11987 |
| Chicken | 11050 |

# 🧩 KEY INSIGHTS

- 🍕 Most Ordered Pizza Type: The Classic Deluxe Pizza
- 💰 Total Revenue: 817860.05 $
- 📈 Peak Ordering Hours: 12-13 and 16-19
- 🏆 Top Revenue-Generating Pizzas: The Thai Chicken Pizza and The Barbecue Chicken Pizza
- 🔍 Most Popular Category: Classic

# THANK YOU