

# Pizza Sales Data Analysis Using SQL

- Name: Gnanasurya R
- Tools: MySQL Workbench, SQL
- Project Type : SQL Data Analysis

# Objective

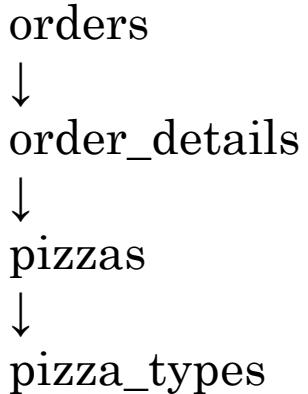
- Analyze sales performance
- Identify revenue drivers
- Understand customer ordering patterns
- Provide data-driven insights

# Dataset Overview

## Tables:

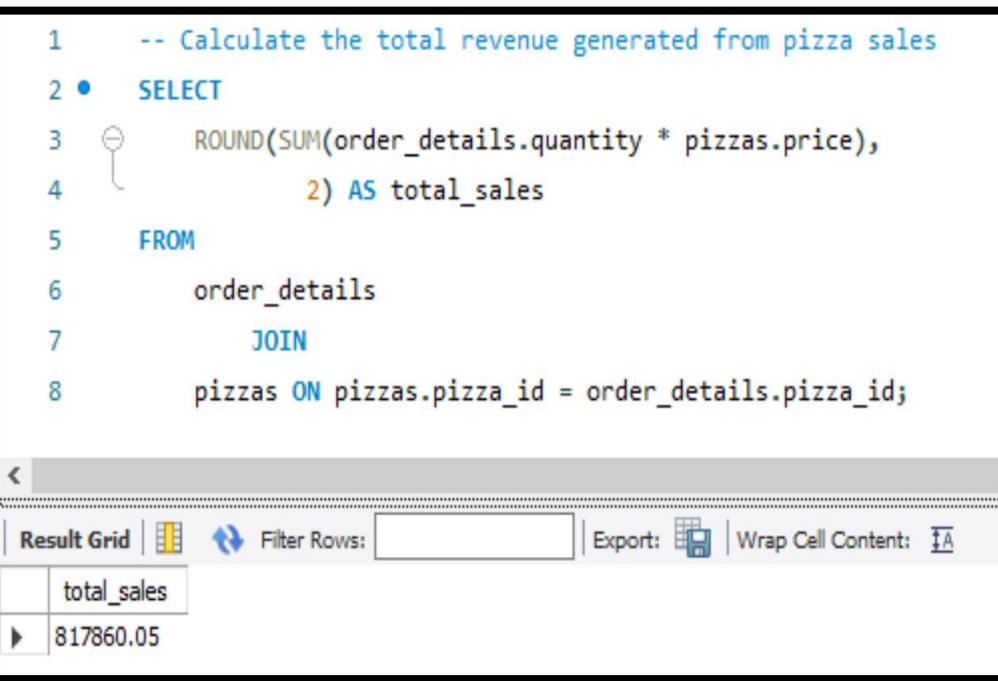
- Orders
- order\_details
- pizzas
- pizza\_types

## Database Relationships:



# Total Revenue

```
1  -- Calculate the total revenue generated from pizza sales
2 •  SELECT
3   ROUND(SUM(order_details.quantity * pizzas.price),
4         2) AS total_sales
5  FROM
6    order_details
7  JOIN
8    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```



The screenshot shows the SQL Workbench interface with the query results. The results are displayed in a table with one row, labeled 'total\_sales' in the first column and '817860.05' in the second column.

total_sales	817860.05

- Revenue calculated as quantity  $\times$  price
- Multi-table join used
- Total revenue: ₹X
- Indicates overall business performance

# Product Performance

```
1  -- List the top 5 most ordered pizza types along with their quantities.
2 • SELECT
3      pizza_types.name, SUM(order_details.quantity) AS quantity
4  FROM
5      pizza_types
6      JOIN
7          pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8      JOIN
9          order_details ON order_details.pizza_id = pizzas.pizza_id
10     GROUP BY pizza_types.name
11     ORDER BY quantity DESC
12     LIMIT 5;
13
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:  Fetch 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

- Aggregated quantity using SUM
- Ranked products
- Top performers drive majority of sales

## Insight:

Focus marketing on top-selling items, and find any improvements needed in other category products

# Time-Based Analysis

```
1 -- Determine the distribution of orders by hour of the day.  
2 • select hour(order_time) , count(order_id) as order_count from orders  
3 group by hour(order_time);
```

The screenshot shows a SQL workbench interface with a results grid. The grid has two columns: 'hour(order\_time)' and 'order\_count'. The data is as follows:

	hour(order_time)	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642

- Extracted hour from order\_time
- Identified peak demand window
- Useful for staffing & promotions

# Category Analysis

```
1 -- Calculate the percentage contribution of each pizza type to total revenue.
2 • select pizza_types.category , round(sum(
3     order_details.quantity* pizzas.price)/(SELECT
4         ROUND(SUM(order_details.quantity * pizzas.price),
5             2) AS total_sales FROM
6         order_details JOIN
7             pizzas ON pizzas.pizza_id = order_details.pizza_id)*100 ,2)as revinue
8     from pizza_types join pizzas
9     on pizza_types.pizza_type_id = pizzas.pizza_type_id
10    join order_details |
11        on order_details.pizza_id = pizzas.pizza_id
12    group by pizza_types.category order by revinue desc ;
13
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

category	revinue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

- Revenue segmented by category
- Identified dominant category
- Shows customer preference trend

# Top 3 Pizza Types per Category, Based on revenue

```
1  -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2 • select name, revenue from
3   (select category, name , revenue, rank() over(partition by category order by revenue desc) as rn from
4   (select pizza_types.category, pizza_types.name,
5    sum((order_details.quantity) * pizzas.price) as revenue
6    from pizza_types join pizzas
7    on pizza_types.pizza_type_id = pizzas.pizza_type_id
8    join order_details
9    on order_details.pizza_id = pizzas.pizza_id
10   group by pizza_types.category, pizza_types.name) as a) as b
11  where rn<=3;
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

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 Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25

Result 3 ×

- Identify highest revenue-generating pizzas within each category
- Perform intra-category performance comparison

# Key Takeaways & Skills Demonstrated

- **Analytical Insights**
- Identified high-revenue products and categories
- Determined peak sales periods using time-based aggregation
- Measured product-level revenue contribution
- Performed intra-category ranking using window functions

## Technical Competencies

- Multi-table INNER JOIN operations
- Advanced aggregation (SUM, COUNT, AVG)
- GROUP BY with dimensional analysis
- Window functions (RANK, PARTITION BY, OVER)
- Revenue metric modeling

## Business Impact

- Enables data-driven product prioritization
- Supports staffing optimization during peak hours
- Assists in revenue-focused marketing strategies