

# SQL QUERIES ON PIZZA SALES DATA

ARSHAD ALI

# Questions:

## ❖ Basic:

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.

## ❖ Intermediate:

1. Join the necessary tables to find the total quantity of each pizza category category.
2. Determine the distribution of orders by hour of the day.
3. Join relevant tables to find the category-wise distribution of pizzas.
4. Group the orders by date and calculate the average number of pizzas ordered per day.
5. Determine the top 3 most ordered pizza types based on revenue.

## ❖ Advanced:

1. Calculate the percentage contribution of each pizza type to total revenue.
2. Analyse the cumulative revenue generated over time.
3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

# Introduction

- ❖ In the realm of data analysis, SQL (Structured Query Language) stands out as a fundamental tool for querying and managing data stored in relational databases. This project focuses on analyzing pizza sales data using SQL queries to extract, transform, and analyze the information contained within four CSV files.
- ❖ The primary goal of this project is to gain insights into various aspects of pizza sales, such as customer preferences, sales trends, and product performance. By utilizing SQL, we can efficiently query the data to answer critical business questions and identify patterns that can inform decision-making.
- ❖ I used different type of sql queries to extract meaningful data. The difficulty level of questions were basic to advance.

Q1. Retrieve the total number of orders placed.

```
select count(order_id) as total_orders from orders;
```

| Result Grid |              |
|-------------|--------------|
|             | total_orders |
| ▶           | 21350        |

Q2. 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   |

Q3. Identify the highest-priced pizza.

```
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;
```

Result Grid | Filter Rows

|   | name            | price |
|---|-----------------|-------|
| ▶ | The Greek Pizza | 35.95 |

Q4. Identify the most common pizza size ordered.

```
select pizzas.size, count(order_details.order_details_id) as order_count
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size order by order_count desc;
```

Result Grid | Filter Rows:

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

Q5. 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 pizzas.pizza_type_id = pizza_types.pizza_type_id  
   join order_details  
    on order_details.pizza_id = pizzas.pizza_id  
 group by pizza_types.names order by quantity desc limit 5;
```

Q6. Join the necessary tables to find the total quantity of each pizza category .

```
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 Rows:

|   | category | quantity |
|---|----------|----------|
| ▶ | Classic  | 14888    |
|   | Supreme  | 11987    |
|   | Veggie   | 11649    |
|   | Chicken  | 11050    |

Q7. Determine the distribution of orders by hour of the day.

```
select hour(order_time), count(order_id) as order_count from orders  
group by hour(order_time);
```

Result Grid | Filter Rows:

|   | hour(order_time) | order_count |
|---|------------------|-------------|
| ▶ | 11               | 1231        |
|   | 12               | 2520        |
|   | 13               | 2455        |
|   | 14               | 1472        |
|   | 15               | 1468        |
|   | 16               | 1920        |
|   | 17               | 2336        |

Q8. Join relevant tables to find the category-wise distribution of pizzas.

```
select category, count(name) from pizza_types  
group by category;
```

Result Grid | Filter Rows:

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

Q9. Group the orders by date and calculate the average number of pizzas ordered per day.

```
• select round(avg(quantity), 0) as avg_pizza_ordered_per_day  
  from  
    (select orders.order_date, count(order_details.order_id) as quantity  
     from orders join order_details  
     on orders.order_id = order_details.order_id  
     group by orders.order_date) as order_quantity;
```

| Result Grid |                           |
|-------------|---------------------------|
|             | avg_pizza_ordered_per_day |
| ▶           | 136                       |

Q10. Determine the top 3 most ordered pizza types based on revenue.

```
select pizza_types.name,  
sum(order_details.quantity * pizzas.price) as revenue  
from pizza_types join pizzas  
on pizzas.pizza_type_id = pizza_types.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;
```

|   | name                         | revenue  |
|---|------------------------------|----------|
| ▶ | The Thai Chicken Pizza       | 43434.25 |
|   | The Barbecue Chicken Pizza   | 42768    |
|   | The California Chicken Pizza | 41409.5  |

Q11. Calculate the percentage contribution of each pizza type to total revenue.

```
• select pizza_types.category,  
    round(sum(order_details.quantity * pizzas.price) / (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) * 100, 2) AS revenue  
from pizzas_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;
```

## Q12. Analyse 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 | 2713.8500000000004 |
|   | 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           |

Q13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
select name, revenue
from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn
from
(select pizza_types.category, pizza_types.name,
sum(order_details.quantity * pizzas.price) as revenue
from pizza_types join pizzas
on pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn <= 3;
```