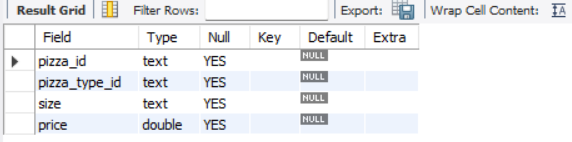
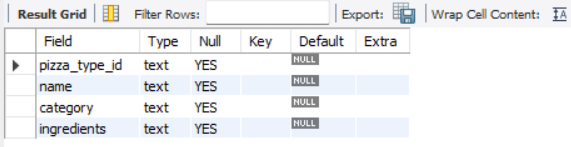
***Table Structures –***

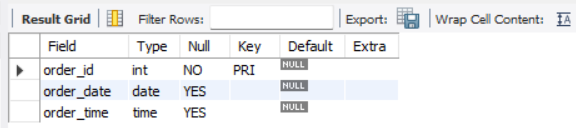
1. **pizzas**



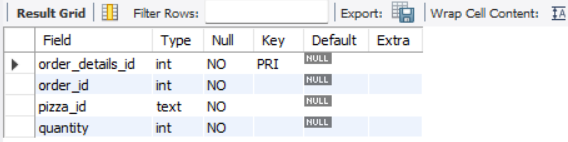
1. **pizza\_types**



1. **orders**



1. **order\_details**



***Business Statement –***

The database represents a pizza business operation, capturing details about pizzas, their types, customer orders, and the specifics of each order. By analyzing this data, the business can uncover valuable insights to optimize operations, understand customer preferences, and drive growth. The data is structured to enable detailed analysis across multiple dimensions, including sales trends, product performance, and operational efficiency.

**create database project;**

**use project;**

**select \* from pizzas;**

**select \* from pizza\_types;**

**select \* from orders;**

**select \* from order\_details;**

1. Retrieve the total number of orders placed.

Query-

**select count(\*) as total\_number from orders;**

Output-



1. Calculate the total revenue generated from pizza sales.

Query-

**select round(sum(od.quantity \* p.price), 2)**

**from order\_details od join pizzas p**

**on od.pizza\_id = p.pizza\_id;**

Output-



1. Identify the highest-priced pizza.

Query-

**select pt.name, p.price**

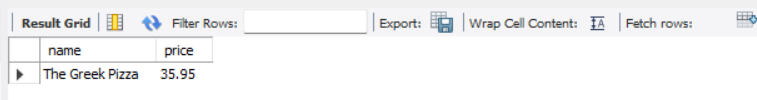
**from pizzas p join pizza\_types pt**

**on p.pizza\_type\_id = pt.pizza\_type\_id**

**order by p.price**

**desc limit 1;**

Output-



1. Identify the most common pizza size ordered.

Query-

**select p.size, count(od.order\_details\_id) as order\_count**

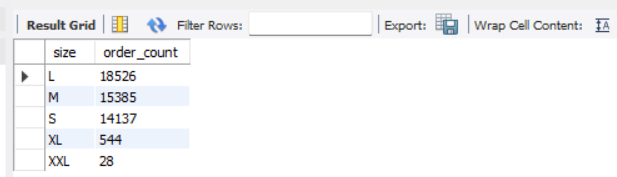
**from pizzas p join order\_details od**

**on p.pizza\_id = od.pizza\_id**

**group by p.size**

**order by order\_count desc;**

Output-



1. List the top 5 most ordered pizza types along with their quantities.

Query-

**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**

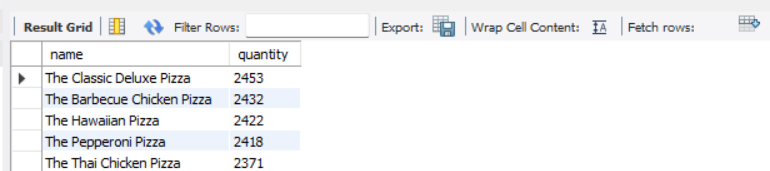
**on od.pizza\_id = p.pizza\_id**

**group by pt.name**

**order by quantity**

**desc limit 5;**

Output-



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

Query-

**select pt.category, sum(od.quantity) as quantity**

**from pizza\_types pt join pizzas p**

**on pt.pizza\_type\_id = p.pizza\_type\_id**

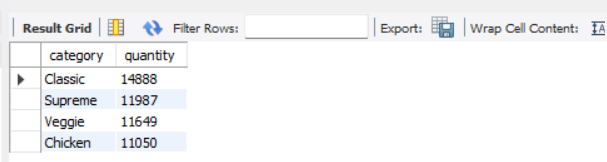
**join order\_details od**

**on od.pizza\_id = p.pizza\_id**

**group by pt.category**

**order by quantity desc;**

Output-



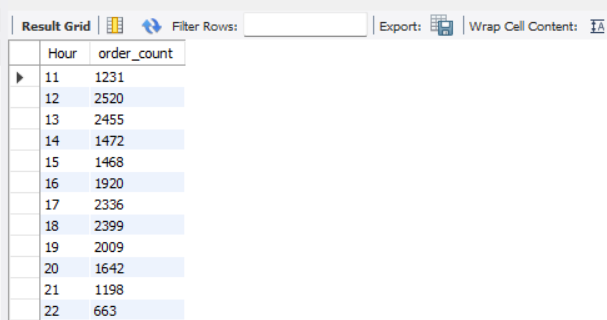
1. Determine the distribution of orders by hour of the day.

Query-

**select hour(order\_time) as Hour, count(order\_id) as order\_count**

**from orders group by hour(order\_time);**

Output-



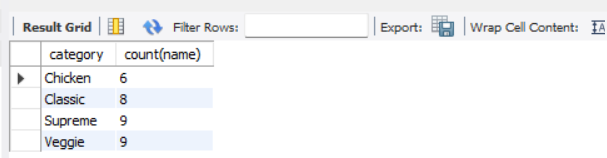
1. Join relevant tables to find the category-wise distribution of pizzas.

Query-

**select category, count(name)**

**from pizza\_types group by category;**

Output-



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

Query-

**select round(avg(quantity),0) as avg\_pizzas\_per\_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;**

Output-



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

Query-

**select 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**

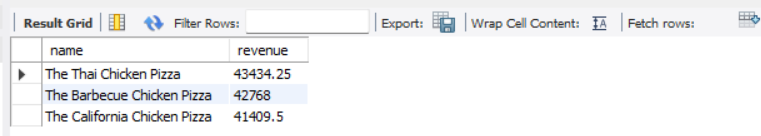
**on od.pizza\_id = p.pizza\_id**

**group by pt.name**

**order by revenue**

**desc limit 3;**

Output-



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

Query-

**select pt.category, sum(od.quantity \* p.price)/ (select round(sum(od.quantity \* p.price), 2)**

**from order\_details od join pizzas p**

**on od.pizza\_id = p.pizza\_id) \* 100 as revenue**

**from pizza\_types pt join pizzas p**

**on pt.pizza\_type\_id = p.pizza\_type\_id**

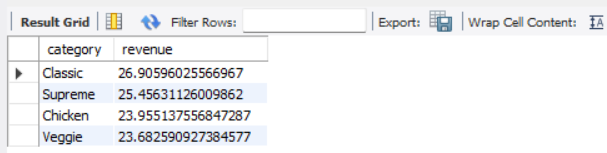
**join order\_details od**

**on od.pizza\_id = p.pizza\_id**

**group by pt.category**

**order by revenue desc;**

Output-



1. Analyze the cumulative revenue generated over time.

Query-

**select order\_date, sum(revenue) over(order by order\_date) as cummalative\_revenue from**

**(select o.order\_date, sum(od.quantity \* p.price) as revenue**

**from order\_details od join pizzas p**

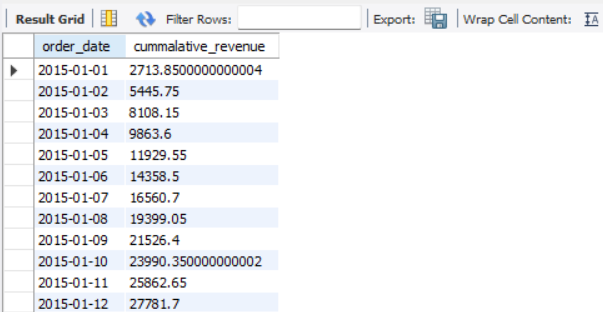
**on od.pizza\_id = p.pizza\_id**

**join orders o**

**on o.order\_id = od.order\_id**

**group by o.order\_date) as sales;**

Output-



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

Query-

**select name, revenue from**

**(select category, name, revenue,**

**rank() over(partition by category order by revenue desc) as rn from**

**(select pt.category, 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**

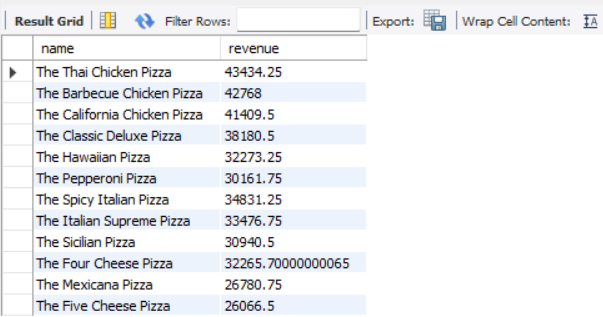
**join order\_details od**

**on od.pizza\_id = p.pizza\_id**

**group by pt.category, pt.name) as a) as b**

**where rn<=3;**

Output-



***Conclusion -***

**Comprehensive Analysis Possibilities:**

1. **Sales Trends:**  
   Analyze orders and order\_details to determine sales trends, peak times, and seasonal patterns.
2. **Customer Preferences:**  
   Join order\_details with pizzas and pizza\_types to identify which pizza types are most popular and whether specific types dominate certain times of the year.
3. **Revenue Insights:**  
   Combine pizzas pricing (if present) with order\_details to estimate revenue and identify high-performing products.
4. **Operational Optimization:**  
   Use order timestamps to evaluate order processing efficiency and adjust staffing for peak periods.
5. **Product Development:**  
   Assess the performance of pizza types in the pizza\_types table to inform decisions about introducing or discontinuing pizzas.