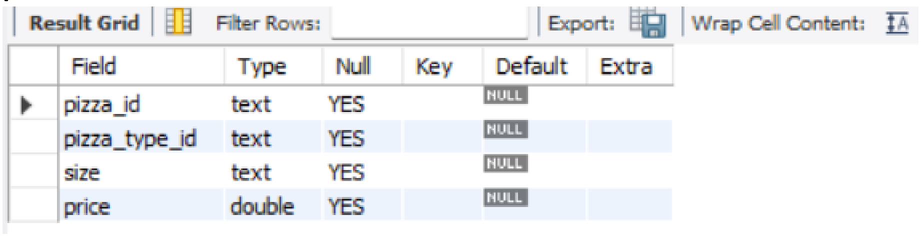
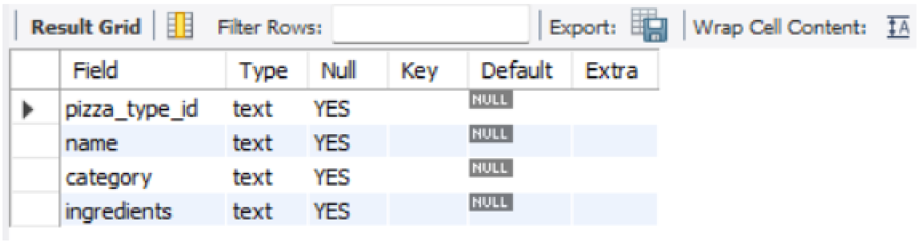
## *Table Structures –*

1. **pizzas**



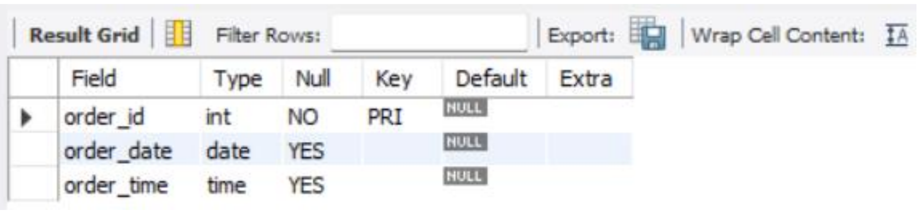
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| pizza_id | text | YES | | NULL | |
| pizza_type_id | text | YES | | NULL | |
| size | text | YES | | NULL | |
| price | double | YES | | NULL | |

2. **pizza_types**



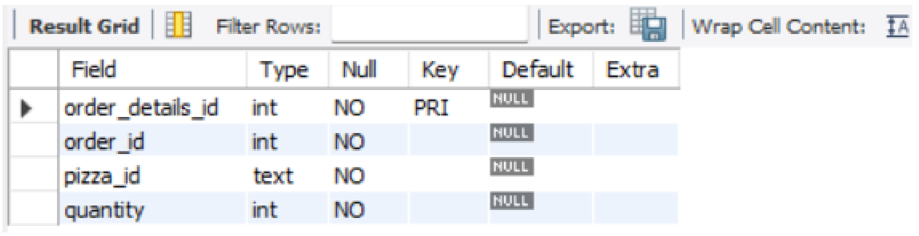
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| pizza_type_id | text | YES | | NULL | |
| name | text | YES | | NULL | |
| category | text | YES | | NULL | |
| ingredients | text | YES | | NULL | |

3. **orders**



| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| order_id | int | NO | PRI | NULL | |
| order_date | date | YES | | NULL | |
| order_time | time | YES | | NULL | |

4. **order_details**



| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| order_details_id | int | NO | PRI | NULL | |
| order_id | int | NO | | NULL | |
| pizza_id | text | NO | | NULL | |
| quantity | int | NO | | NULL | |

## ***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;**

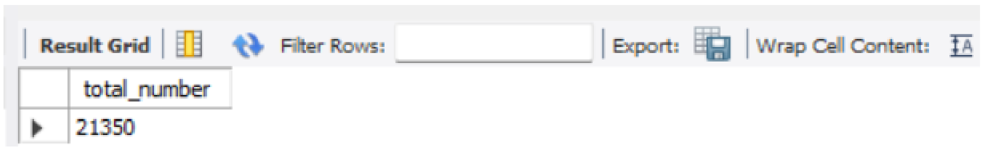
**select * from pizzas;**

**select * from pizza_types;**

**select * from orders;**

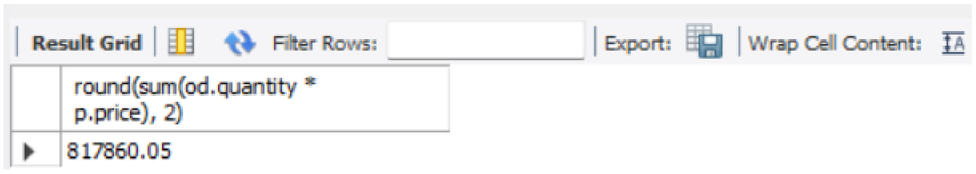
**select * from order_details;**

1. Retrieve the total number of orders placed.
   Query-
   **select count(*) as total_number from orders;**
   Output-



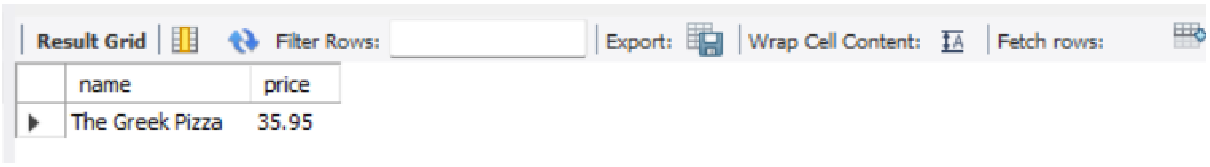
```
total_number
21350
```

2. 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-



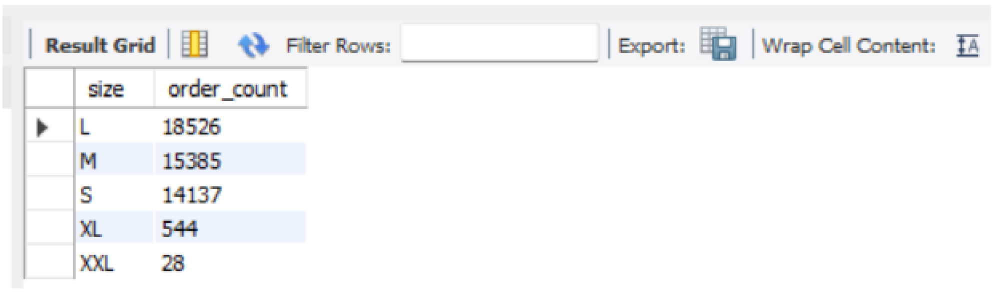
```
round(sum(od.quantity *
p.price), 2)
817860.05
```

3. 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-


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

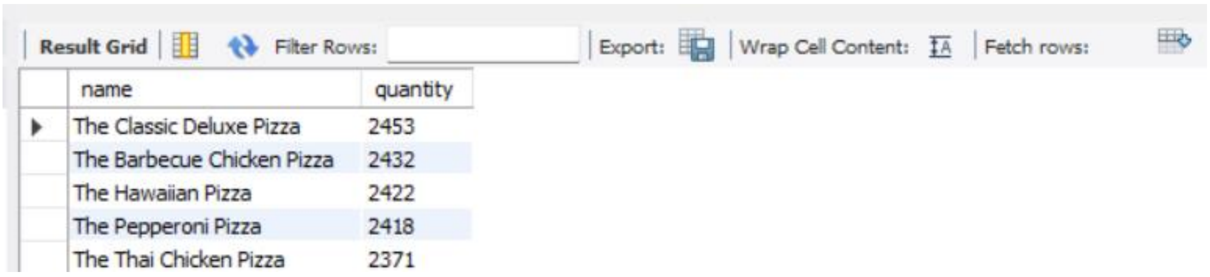
| name | price |
|---|---|
| The Greek Pizza | 35.95 |

4. 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-


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

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

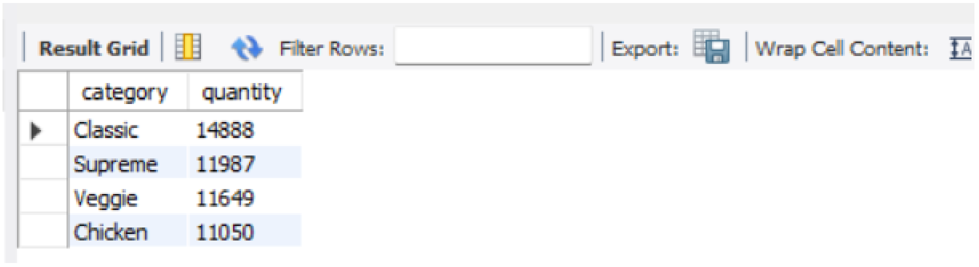
5. List the top 5 most ordered pizza types along with their quantities.
   Query-
   **select pt.name, sum(od.quantity) as quantity**
   **from pizza_types pt join pizzas p**
   **on pt.pizza_type_id = p.pizza_type_id**
   **join order_details od**
   **on od.pizza_id = p.pizza_id**
   **group by pt.name**
   **order by quantity**
   **desc limit 5;**
   Output-


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 |

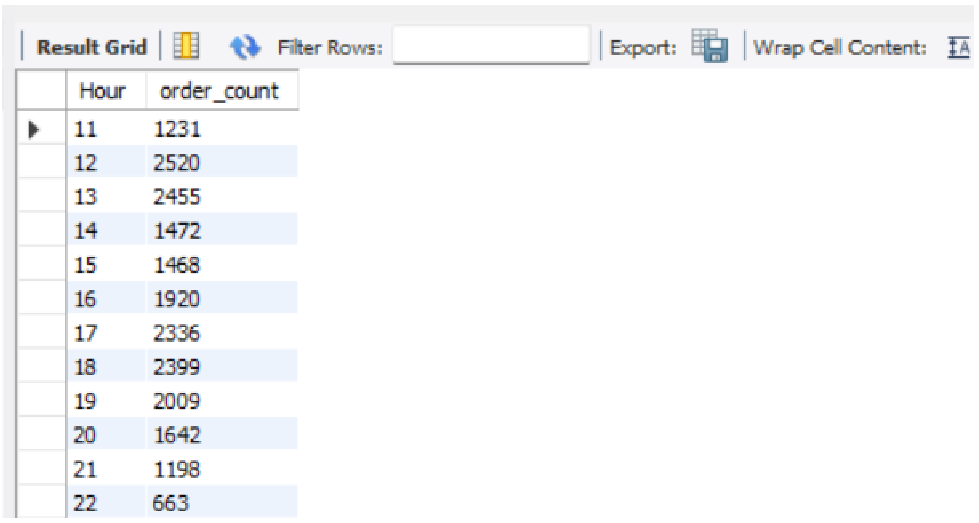
6. Join the necessary tables to find the total quantity of each pizza category ordered.
   Query-
   **select pt.category, sum(od.quantity) as quantity**
   **from pizza_types pt join pizzas p**
   **on pt.pizza_type_id = p.pizza_type_id**
   **join order_details od**
   **on od.pizza_id = p.pizza_id**

**group by pt.category**
**order by quantity desc;**
Output-


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

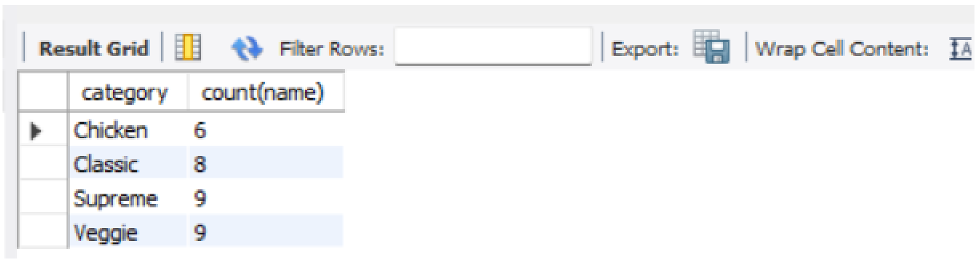
| category | quantity |
|---|---|
| Classic | 14888 |
| Supreme | 11987 |
| Veggie | 11649 |
| Chicken | 11050 |

7. 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-


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

| 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 |

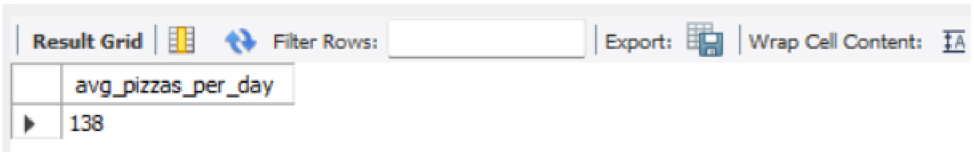
8. Join relevant tables to find the category-wise distribution of pizzas.
   Query-
   **select category, count(name)**
   **from pizza_types group by category;**
   Output-


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

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

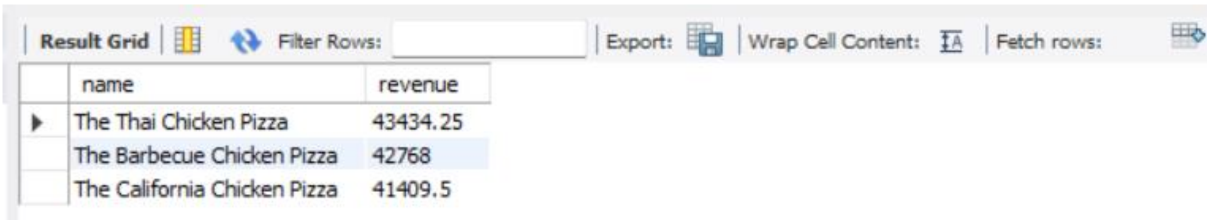
9. Group the orders by date and calculate the average number of pizzas ordered per day.
   Query-
   **select round(avg(quantity),0) as avg_pizzas_per_day from**
   **(select o.order_date, sum(od.quantity) as quantity**
   **from orders o join order_details od**

**on o.order_id = od.order_id**
**group by o.order_date) as order_quantity;**
Output-


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

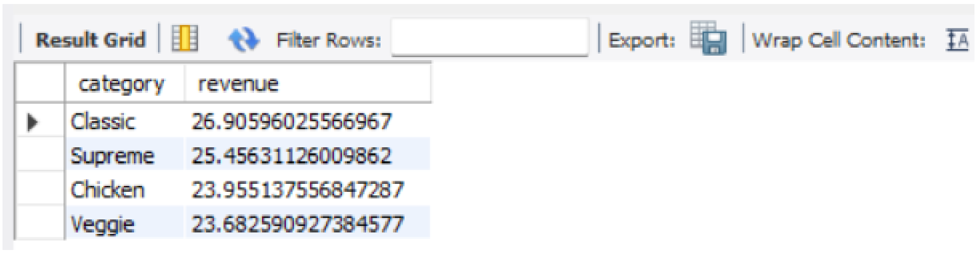
```
avg_pizzas_per_day
138
```

10. 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-


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

```
name                         revenue
The Thai Chicken Pizza       43434.25
The Barbecue Chicken Pizza   42768
The California Chicken Pizza 41409.5
```

11. 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-

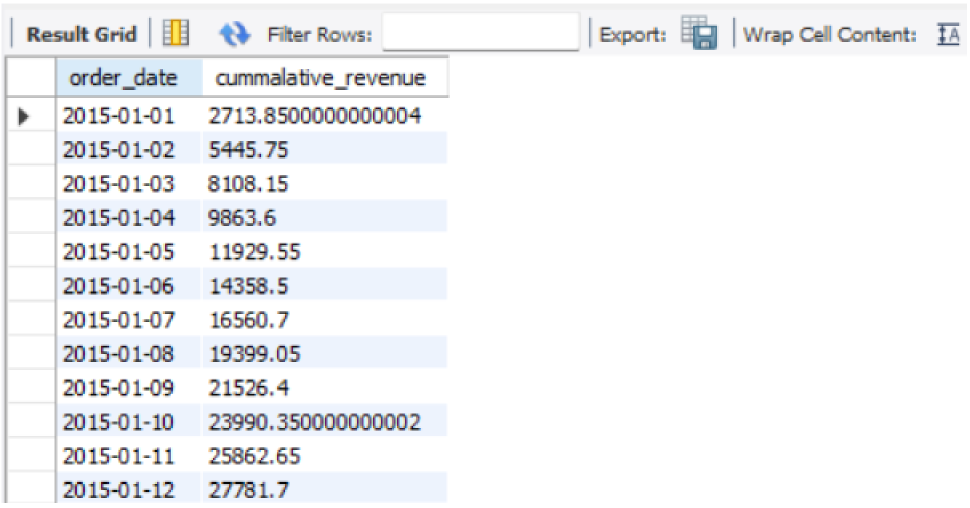

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

```
category  revenue
Classic   26.90596025566967
Supreme   25.45631126009862
Chicken   23.955137556847287
Veggie    23.682590927384577
```

12. 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-



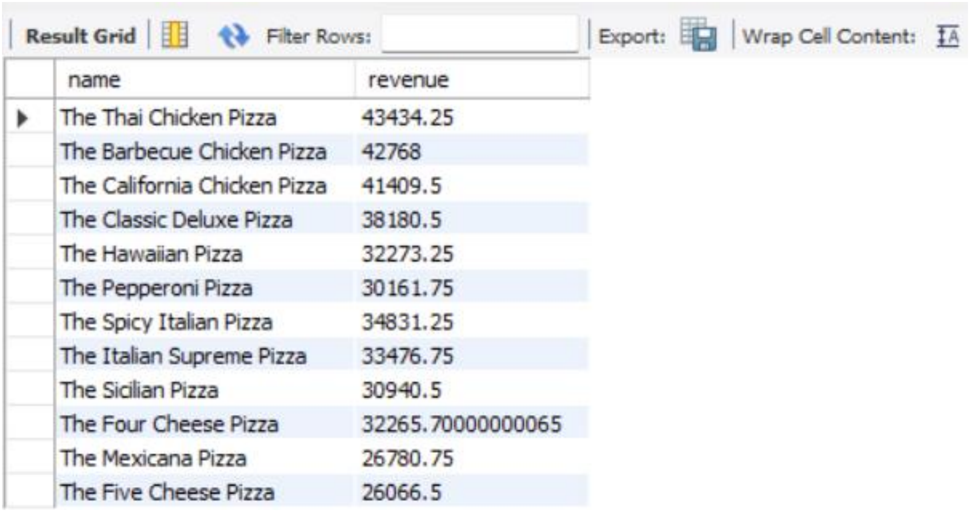
| order_date | cummalative_revenue |
|---|---|
| 2015-01-01 | 2713.8500000000004 |
| 2015-01-02 | 5445.75 |
| 2015-01-03 | 8108.15 |
| 2015-01-04 | 9863.6 |
| 2015-01-05 | 11929.55 |
| 2015-01-06 | 14358.5 |
| 2015-01-07 | 16560.7 |
| 2015-01-08 | 19399.05 |
| 2015-01-09 | 21526.4 |
| 2015-01-10 | 23990.350000000002 |
| 2015-01-11 | 25862.65 |
| 2015-01-12 | 27781.7 |

13. 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-



| 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 |
| The Italian Supreme Pizza | 33476.75 |
| The Sicilian Pizza | 30940.5 |
| The Four Cheese Pizza | 32265.70000000065 |
| The Mexicana Pizza | 26780.75 |
| The Five Cheese Pizza | 26066.5 |

## *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.