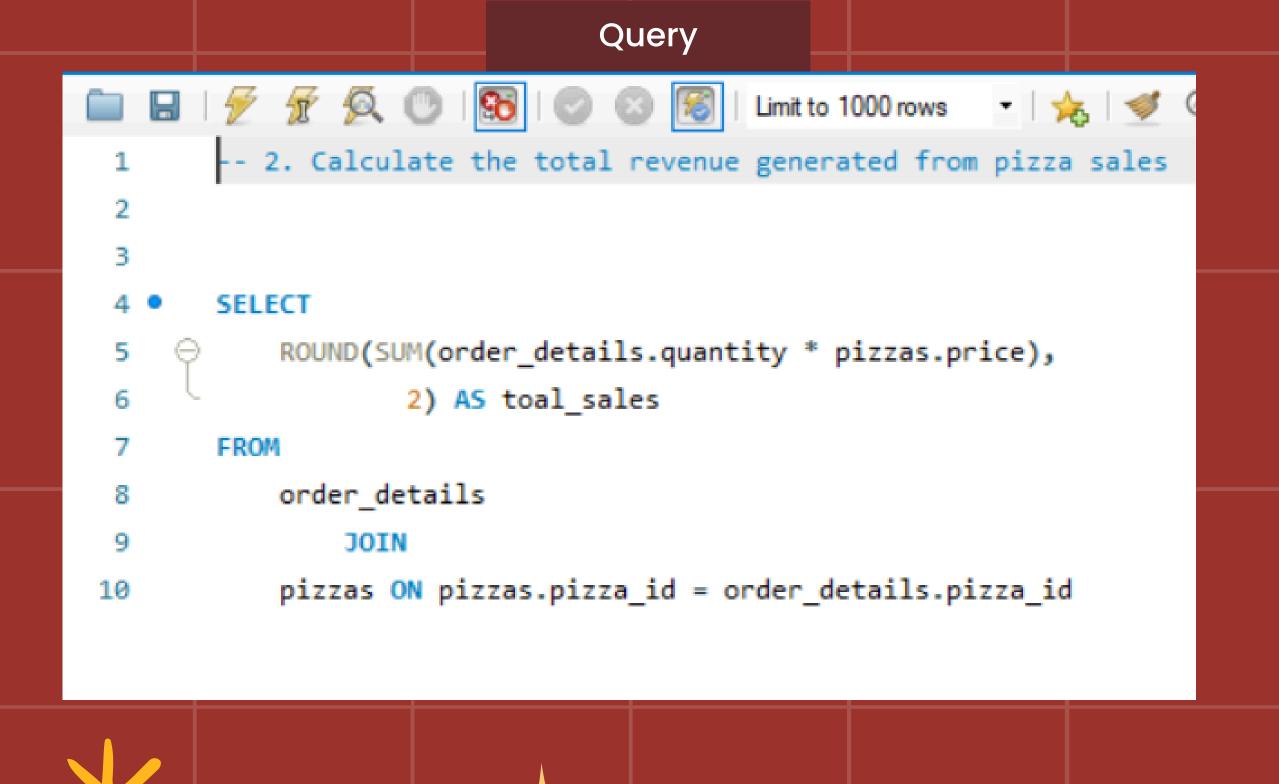






Q.1 Retrieve the total number of orders placed. Query Limit to 1000 rows -- -- 1. Retrieve the total number of orders placed. 3 select count(order_id) as total_orders from orders; Result 5 Result Grid total_orders 21350

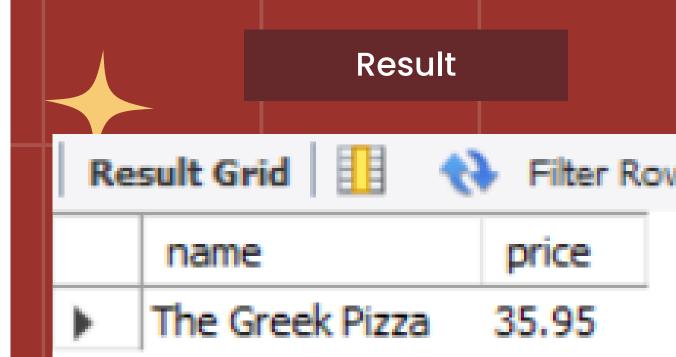
Q.2 Calculate the total revenue generated from pizza sales.





Q.3 Identify the highest-priced pizza.

```
-- 3.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
9
       LIMIT 1;
10
```





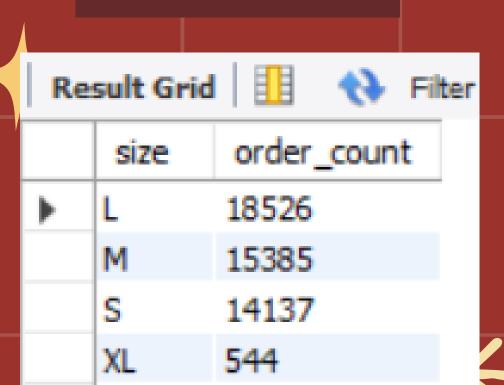




Q.4 Identify the most common pizza size ordered.

Query

```
-- 4.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
10
      ORDER BY order_count DESC;
11
```



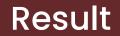
28

XXL

Result

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

```
-- 5.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 pizza_types.pizza_type_id = pizzas.pizza_type_id
               JOIN
10
           order_details ON order_details.pizza_id = pizzas.pizza_id
11
       GROUP BY pizza_types.name
12
       ORDER BY quantity DESC
13
       LIMIT 5;
14
```



Result Grid			
	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	





Q.6 Join the necessary tables to find the total quantity of each pizza category ordered.

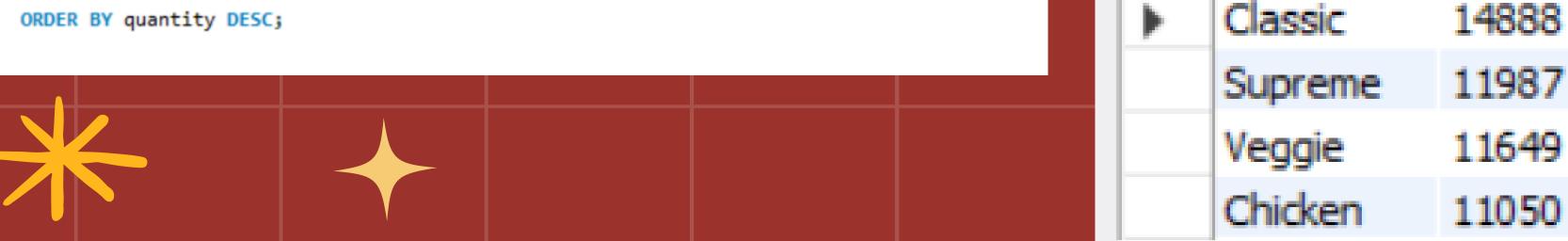
Result

quantity

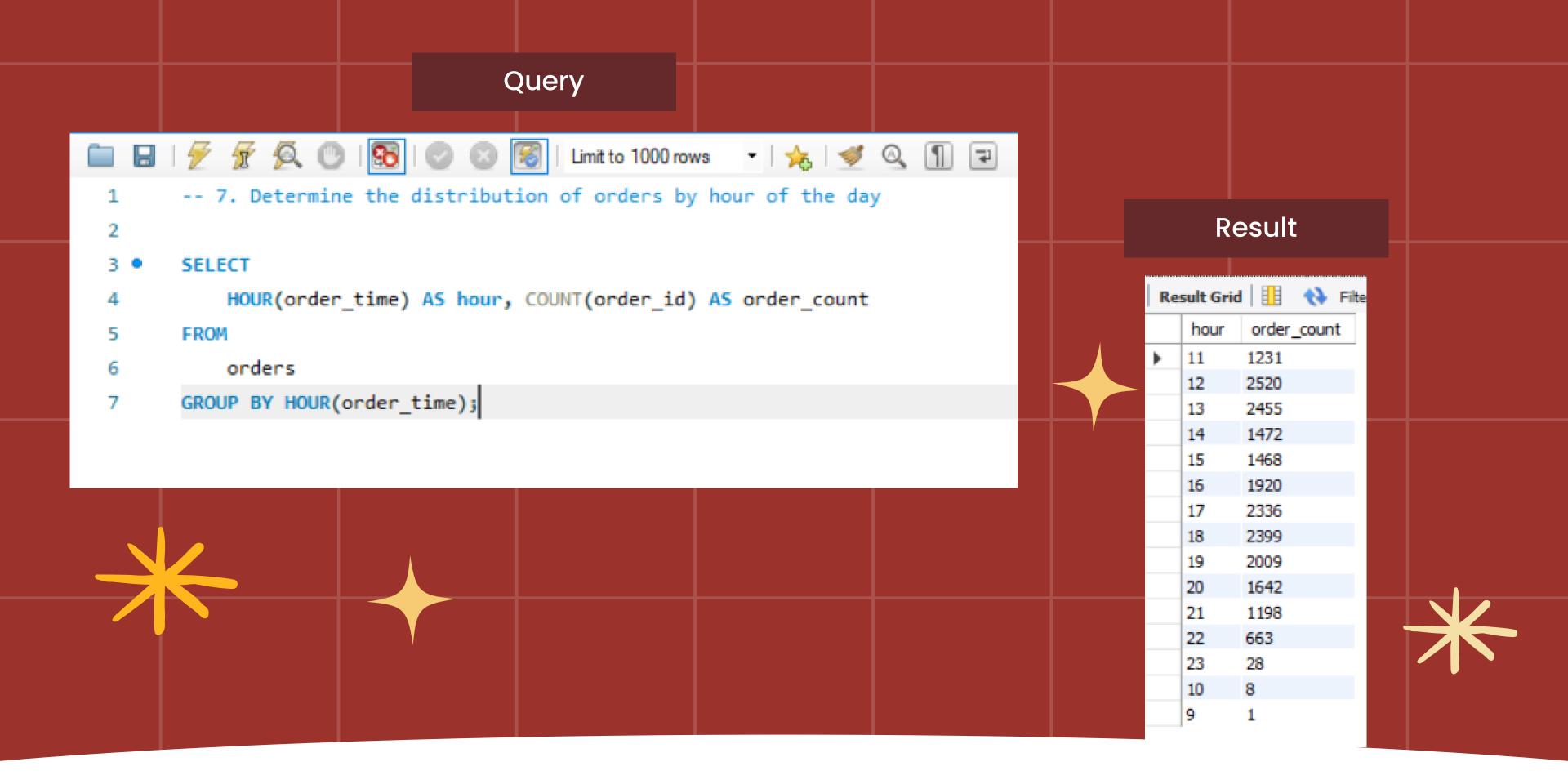
Result Grid

category

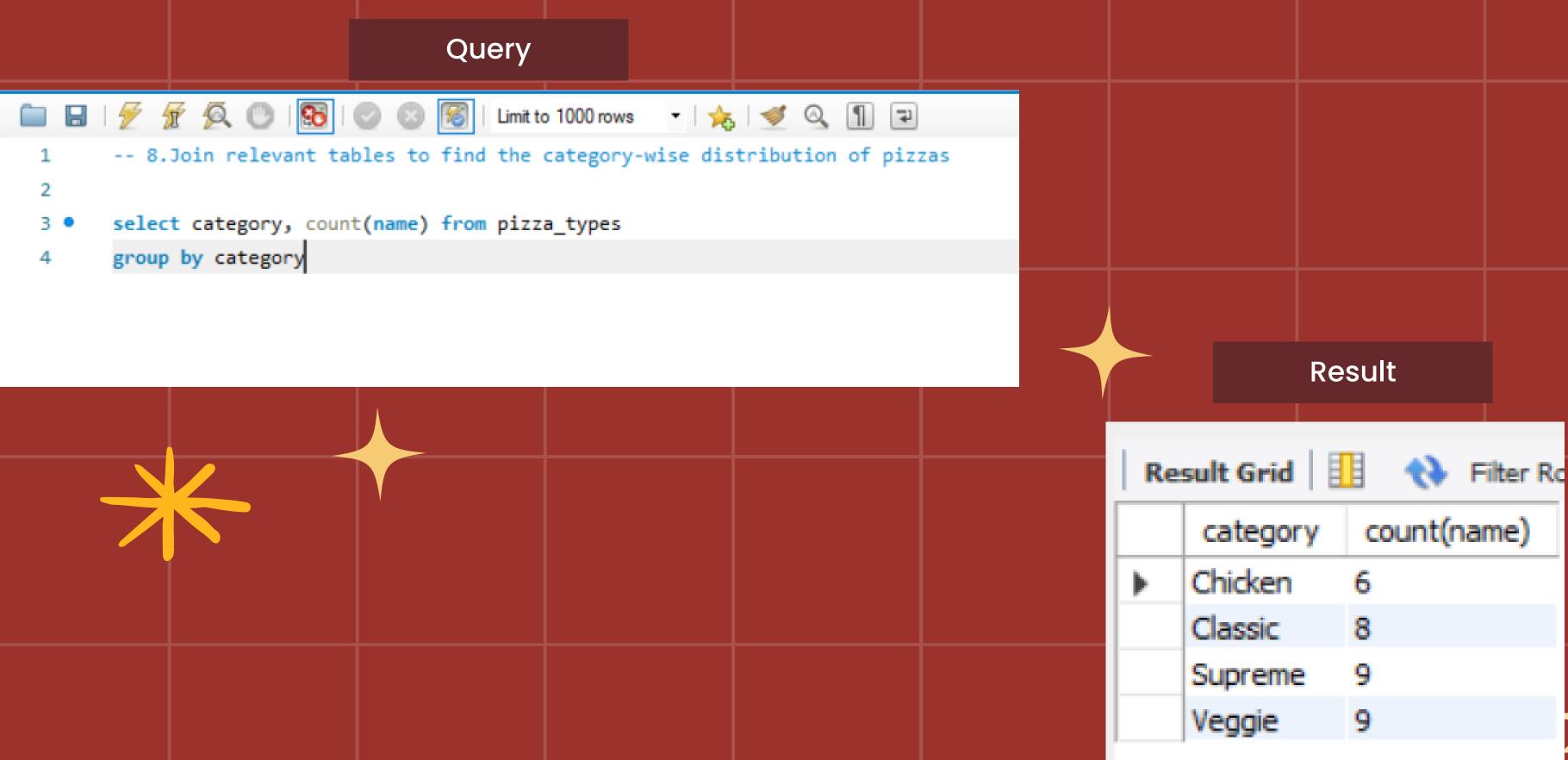
```
-- 6.Join the necessary tables to find the total quantity of each pizza category ordered.
 1
       SELECT
           pizza_types.category,
           SUM(order_details.quantity) AS quantity
       FROM
           pizza types
               JOIN
           pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10
               JOIN
           order_details ON order_details.pizza_id = pizzas.pizza_id
11
       GROUP BY pizza_types.category
12
       ORDER BY quantity DESC;
13
```



Q. 7 Determine the distribution of orders by hour of the day.



Q.8 Join relevant tables to find the category-wise distribution of pizzas.



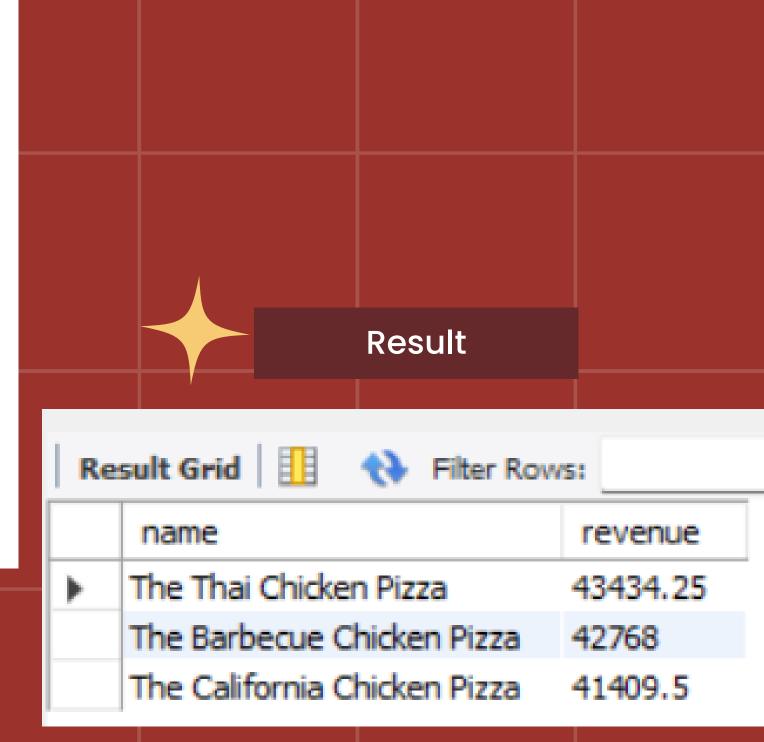
Q.9 Group the orders by date and calculate the average number of pizzas ordered per day.

Query

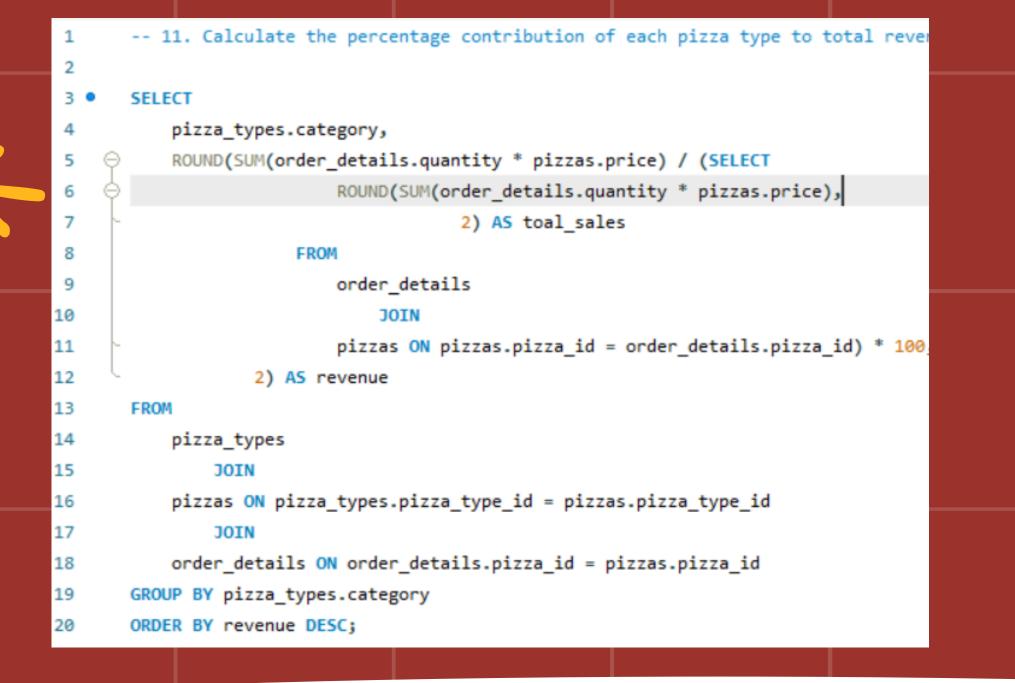
```
-- 9.Group the orders by date and calculate the average number of pizzas ordered per day.
      SELECT
         ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
      FROM
         (SELECT
             orders.order_date, SUM(order_details.quantity) AS quantity
         FROM
             orders
10
         JOIN order_details ON orders.order_id = order_details.order_id
11
                                                                                                         Result
         GROUP BY orders.order_date) AS order_quantity;
12
                                                                                 Result Grid
                                                                                        avg_pizza_ordered_per_day
```

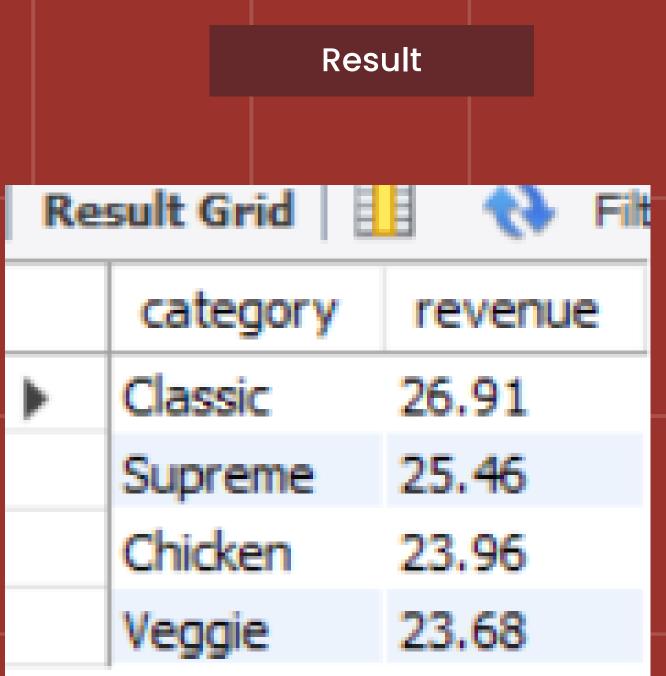
Q.10 Determine the top 3 most ordered pizza types based on revenue.

```
- 10.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
10
           order_details ON order_details.pizza_id = pizzas.pizza_id
11
       GROUP BY pizza_types.name
12
13
       ORDER BY revenue DESC
       LIMIT 3;
14
```



Q.11 Calculate the percentage contribution of each pizza type to total revenue.







Q.12 Analyze the cumulative revenue generated over time.

Query

```
-- 12.Analyze the cumulative revenue generated over time.
       select order_date,
       round(sum(revenue) over (order by order_date), 2) 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
10
       on orders.order_id = order_details.order_id
11
       group by orders.order_date) as sales;
12
```

Result

Result Grid		Filter Ro
	order_date	cum_revenu
	2015-01-01	2713.85
	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	23990.35
	2015-01-11	25862.65

Q.13 Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
-- 13.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
8
       from pizza_types join pizzas
       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, pizza_types.name) as a) as b
13
14
       where rn <= 3;
```



