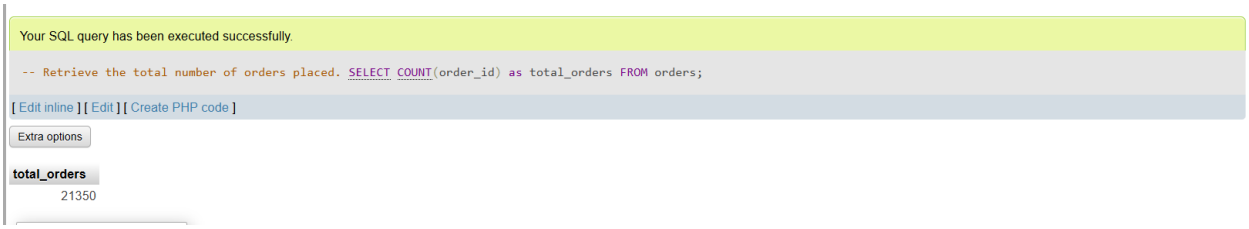
Basic:

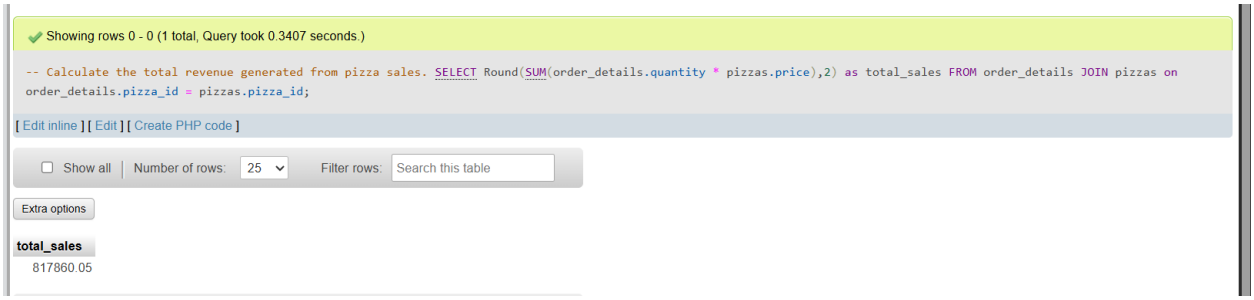
Retrieve the total number of orders placed.

```
SELECT COUNT (order_id) as total_orders FROM orders;
```



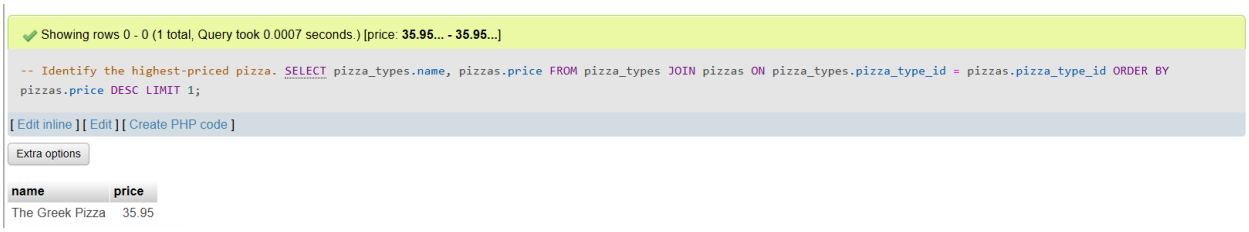
Calculate the total revenue generated from pizza sales.

```
SELECT Round(SUM(order_details.quantity * pizzas.price),2) as total_sales FRO
M order_details JOIN pizzas on order_details.pizza_id = pizzas.pizza_id;
```



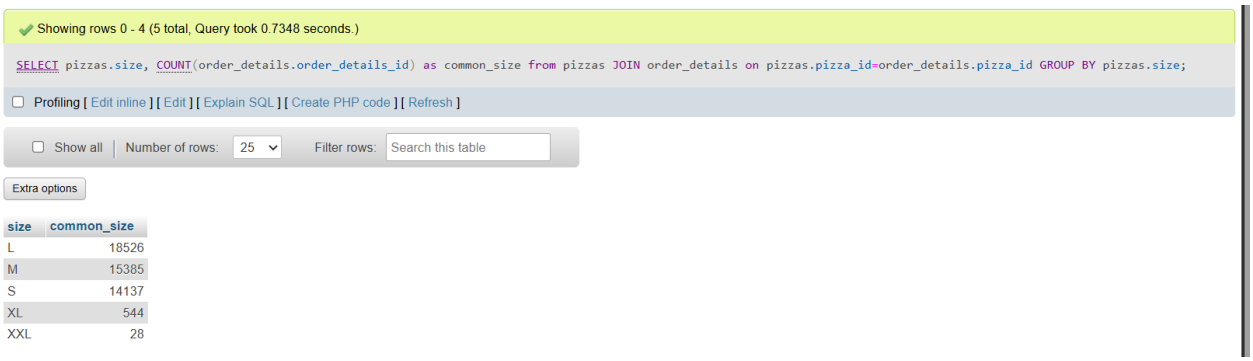
Identify the highest-priced pizza.

```
SELECT pizza_types.name, pizzas.price FROM pizza_types JOIN pizzas ON pizza_t
ypes.pizza_type_id = pizzas.pizza_type_id ORDER BY pizzas.price DESC LIMIT 1;
```



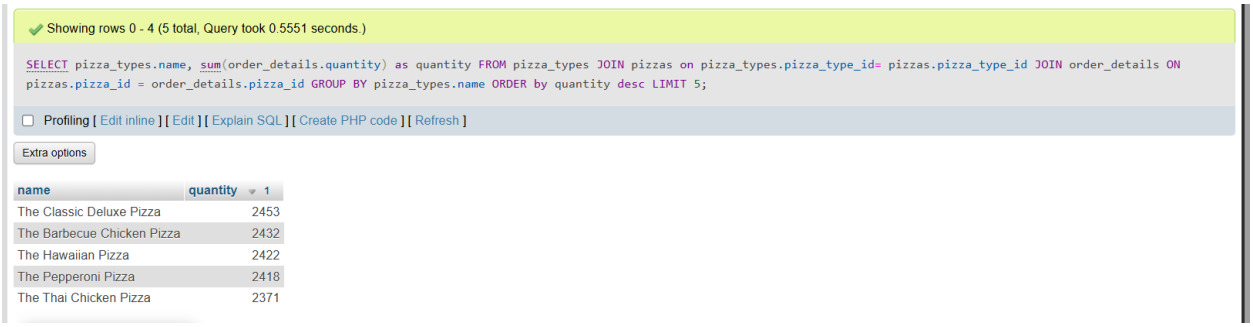
Identify the most common pizza size ordered.

```
SELECT pizzas.size, COUNT(order_details.order_details_id) as common_size from
 pizzas JOIN order_details on pizzas.pizza_id=order_details.pizza_id GROUP BY
 pizzas.size;
```



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 order_details ON pizzas.pizza_id = order_details.pizza_id GROUP BY pizza_types.name ORDER by quantity desc LIMIT 5;
```



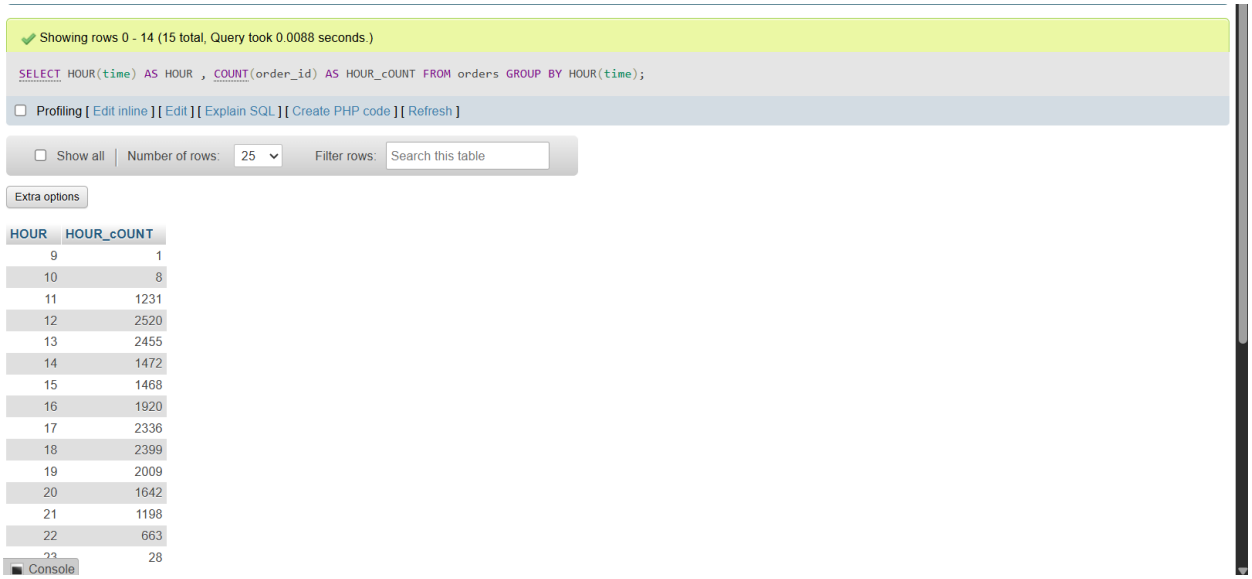
Intermediate:

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

```
SELECT pizza_types.category, SUM(order_details.quantity) AS totalprice_category FROM pizza_types JOIN pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id JOIN order_details ON pizzas.pizza_id = order_details.pizza_id GROUP BY pizza_types.category ORDER BY quantity DESC;
```

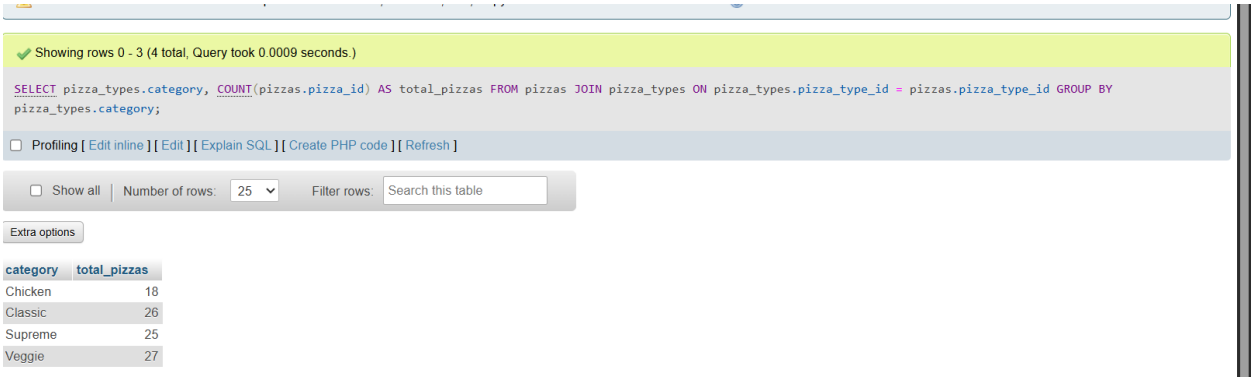
Determine the distribution of orders by hour of the day.

```
SELECT HOUR(time) AS HOUR , COUNT(order_id) AS HOUR_cOUNT FROM orders GROUP BY HOUR(time);
```



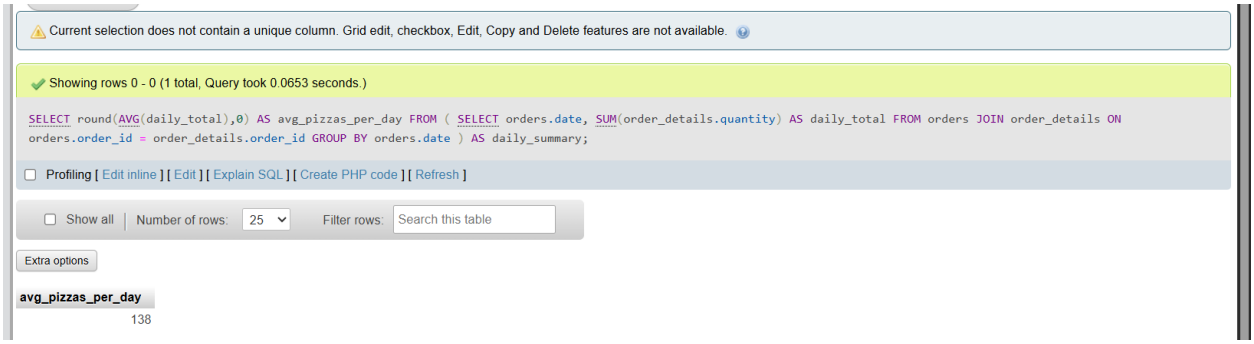
Join relevant tables to find the category-wise distribution of pizzas.

```
SELECT pizza_types.category, COUNT(pizzas.pizza_id) AS total_pizzas FROM pizzas JOIN pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id GROUP BY pizza_types.category;
```



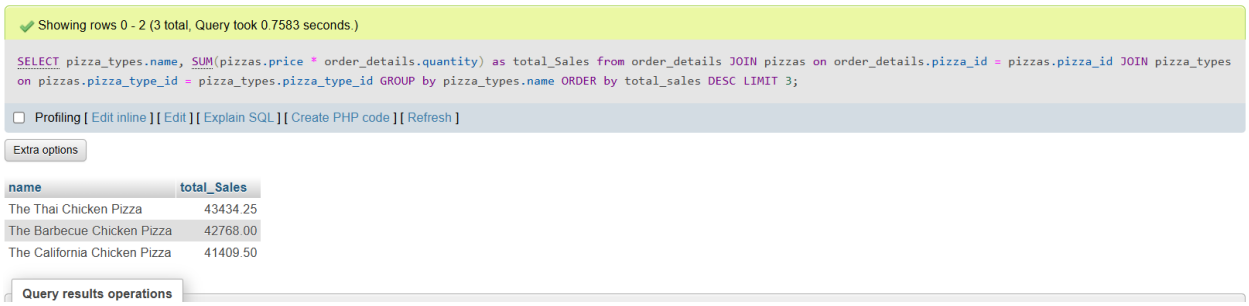
Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT round(AVG(daily_total),0) AS avg_pizzas_per_day FROM ( SELECT orders.date, SUM(order_details.quantity) AS daily_total FROM orders JOIN order_details ON orders.order_id = order_details.order_id GROUP BY orders.date ) AS daily_summary;
```



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

```
SELECT pizza_types.name, SUM(pizzas.price * order_details.quantity) as total_Sales from order_details JOIN pizzas on order_details.pizza_id = pizzas.pizza_id JOIN pizza_types on pizzas.pizza_type_id = pizza_types.pizza_type_id GROUP by pizza_types.name ORDER by total_sales DESC LIMIT 3;
```



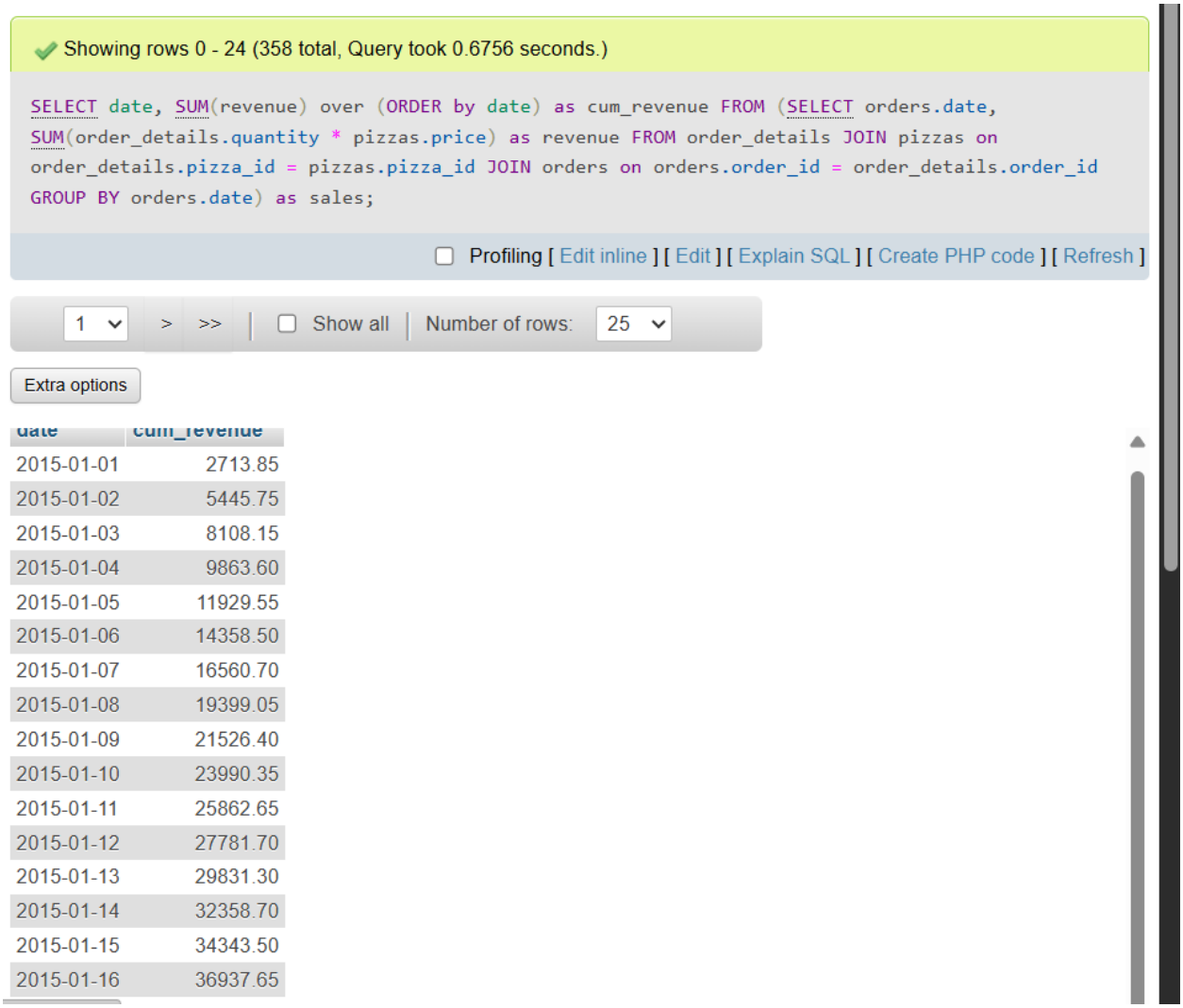
Advanced:

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

```
SELECT pizza_types.category, ROUND(SUM(pizzas.price * order_details.quantity), 2) AS category_sales, ROUND( 100.0 * SUM(pizzas.price * order_details.quantity) / (SELECT SUM(order_details.quantity * pizzas.price) FROM order_details JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id), 2) AS percentage_of_total_sales FROM order_details JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id JOIN pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id GROUP BY pizza_types.category ORDER BY percentage_of_total_sales DESC;
```

Analyze the cumulative revenue generated over time.

```
SELECT date, SUM(revenue) over (ORDER by date) as cum_revenue FROM (SELECT orders.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.date) as sales;
```



Showing rows 0 - 24 (358 total, Query took 0.6756 seconds.)

```
SELECT date, SUM(revenue) over (ORDER by date) as cum_revenue FROM (SELECT orders.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.date) as sales;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> Show all Number of rows: 25

Extra options

date	cum_revenue
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.60
2015-01-05	11929.55
2015-01-06	14358.50
2015-01-07	16560.70
2015-01-08	19399.05
2015-01-09	21526.40
2015-01-10	23990.35
2015-01-11	25862.65
2015-01-12	27781.70
2015-01-13	29831.30
2015-01-14	32358.70
2015-01-15	34343.50
2015-01-16	36937.65

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 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, pizza_types.name ) AS a ) AS b WHERE rn <= 3;
```

Showing rows 0 - 11 (12 total, Query took 0.7174 seconds.)

```
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
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, pizza_types.name ) AS a ) AS b WHERE rn <= 3;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25

Extra options

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768.00
The California Chicken Pizza	41409.50
The Classic Deluxe Pizza	38180.50
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	30940.50
The Four Cheese Pizza	32265.70
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.50