PIZZA SALE SQL

Projet Title: SQL PIZZA SALES ANALYSIS PROJECT

Tool Used : My Sql

Dataset Source : Mock Data

Key Objectives : Sales Analysis

Main SQL Concepts Used:

- **Joins**
- **Subqueries**

Top Insight :

Analysed Pizza Sales Data Using SQL To Uncover key BusinessInsights LikeTop-Selling Pizzas, Pick Order Times And Revenue Contribution

PIZZA SALE- SQL

This dataset represents pizza sales transactions of a pizza store.

It captures orders, order details, pizza types, sizes, and prices, allowing analysis of:

Sales trends

Revenue

Customer ordering behavior

Popular pizzas & categories

The dataset consists of 4 tables.

orders Table

Purpose: Stores high-level order information.

order_details Table

Purpose: Stores details of pizzas ordered in each order.

pizzas Table

Purpose: Stores pizza size and pricing information.

pizza_types Table

Purpose: Stores descriptive details of pizzas.

PIZZA SALE- SQL

Example Business Questions You Can Answer

Total revenue generated

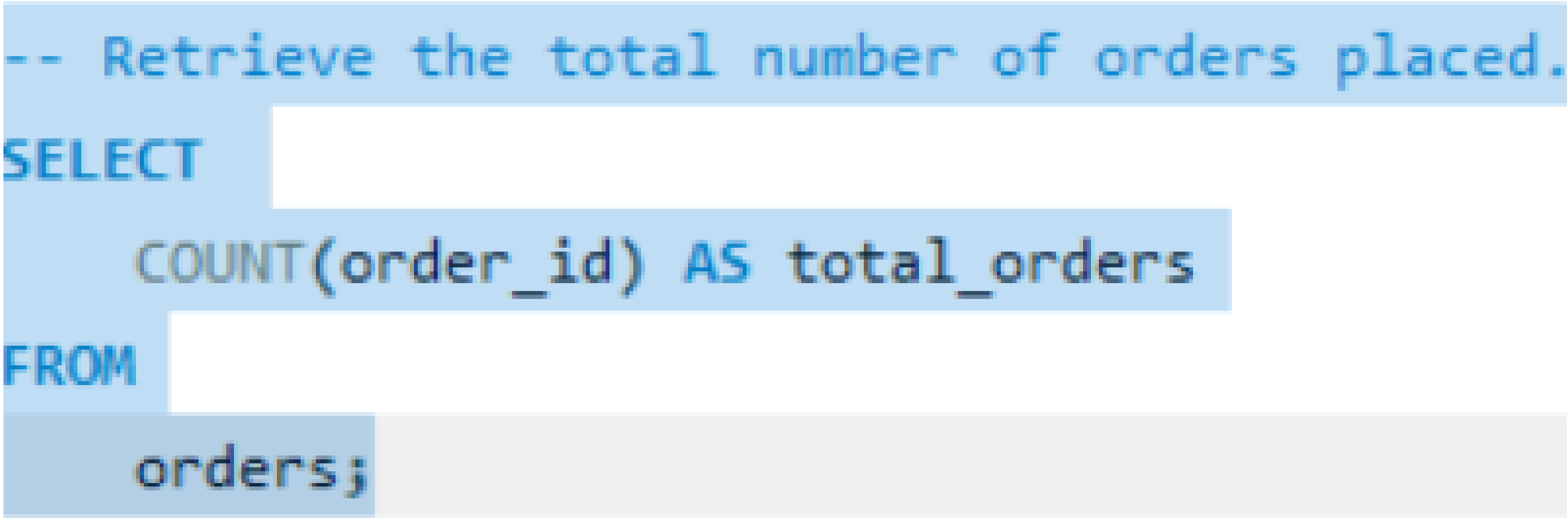
Most popular pizza type & size

Sales by category

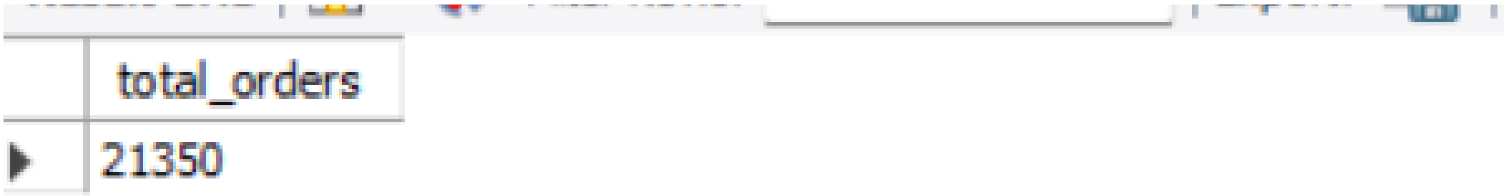
Peak order hours

Top-selling pizzas

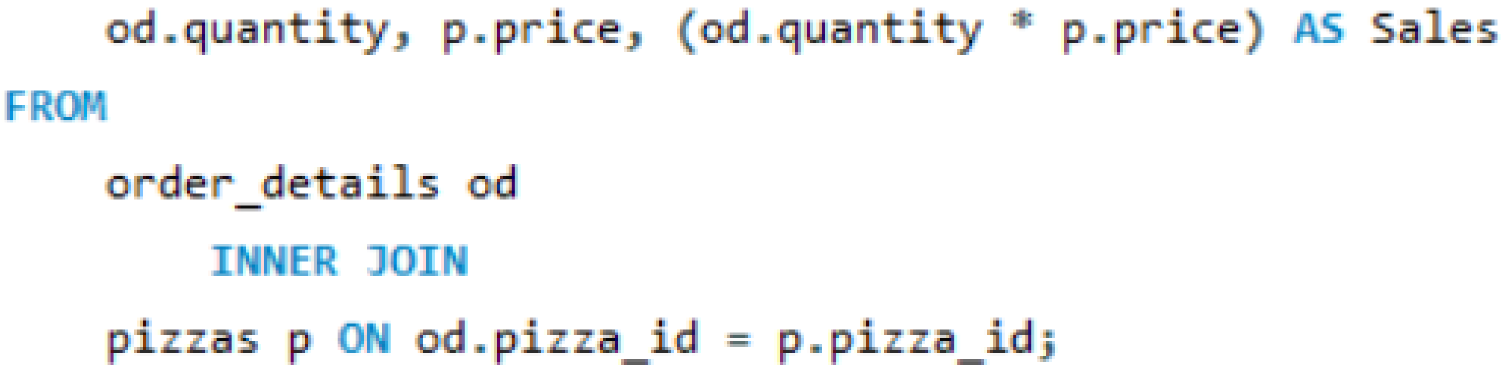
Average order value



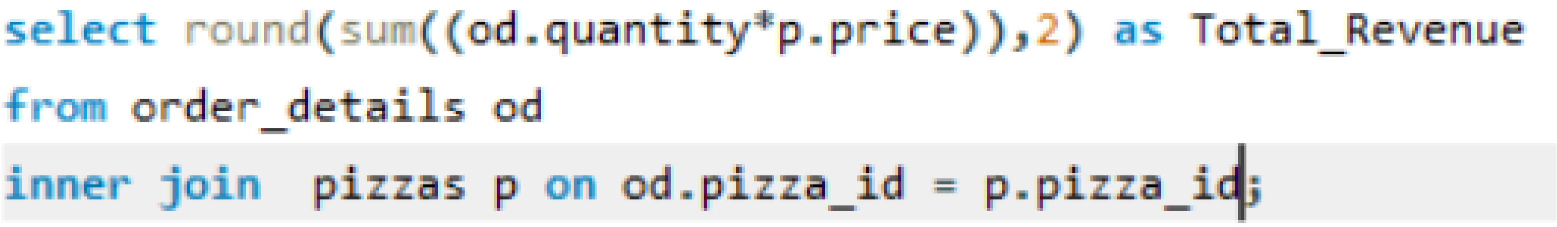
```sql
-- Retrieve the total number of orders placed.
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```



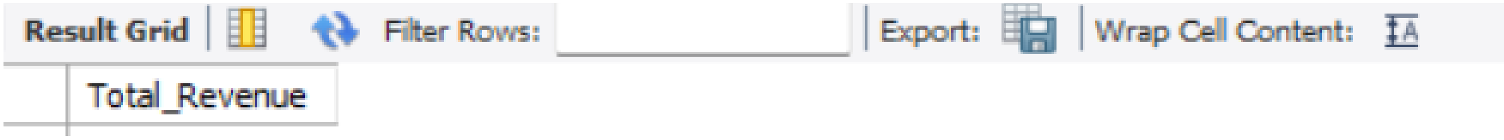
```
total_orders
21350
```



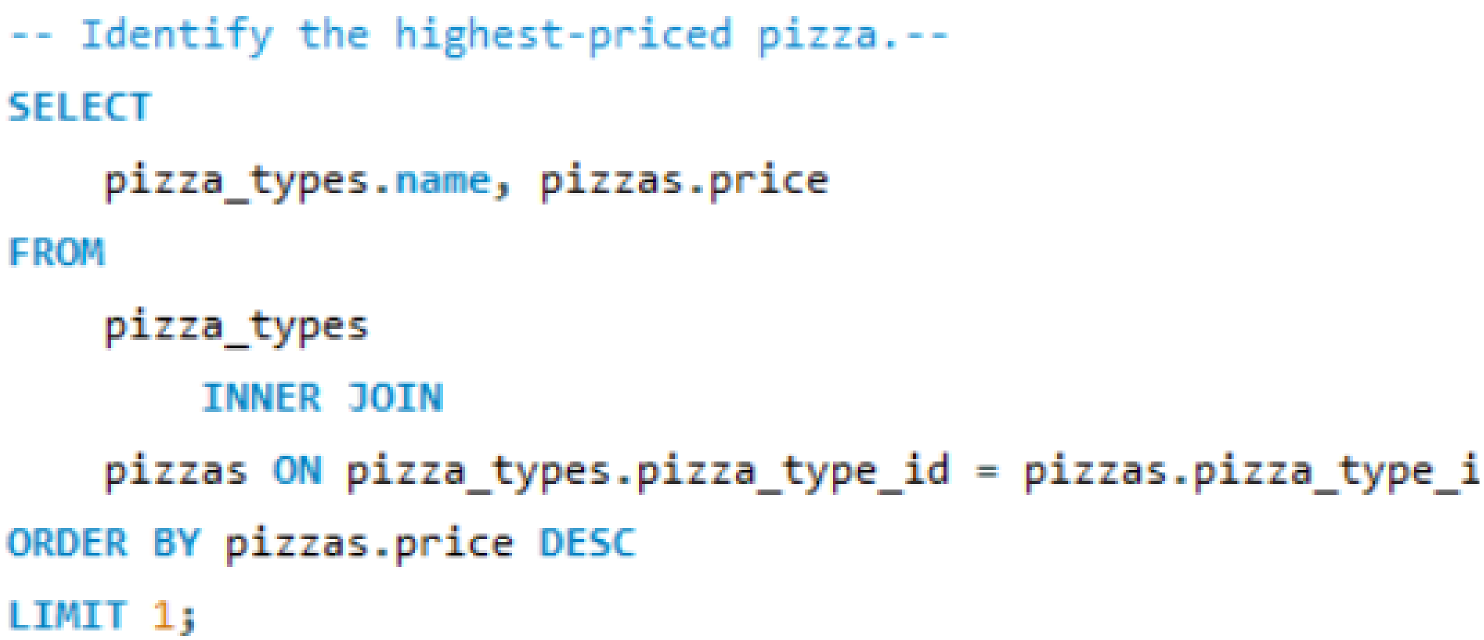
```sql
    od.quantity, p.price, (od.quantity * p.price) AS Sales
FROM
    order_details od
        INNER JOIN
    pizzas p ON od.pizza_id = p.pizza_id;
```



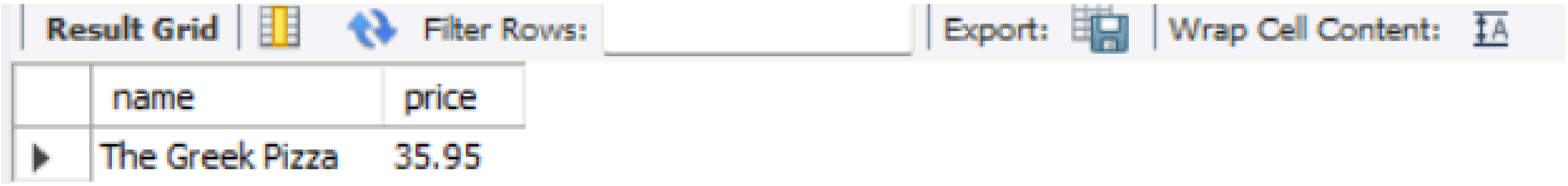
```sql
select round(sum((od.quantity*p.price)),2) as Total_Revenue
from order_details od
inner join  pizzas p on od.pizza_id = p.pizza_id;
```



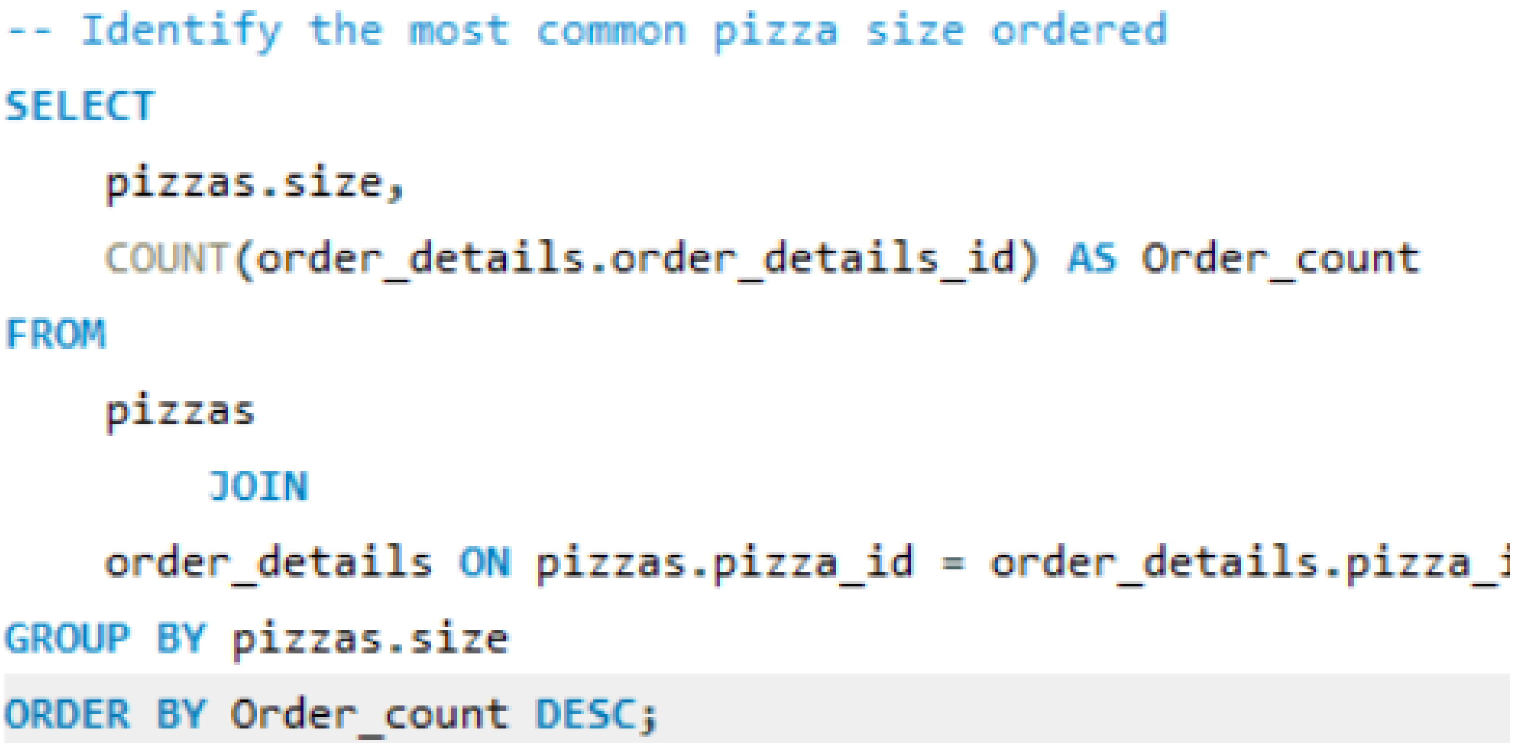
```
Total_Revenue
```



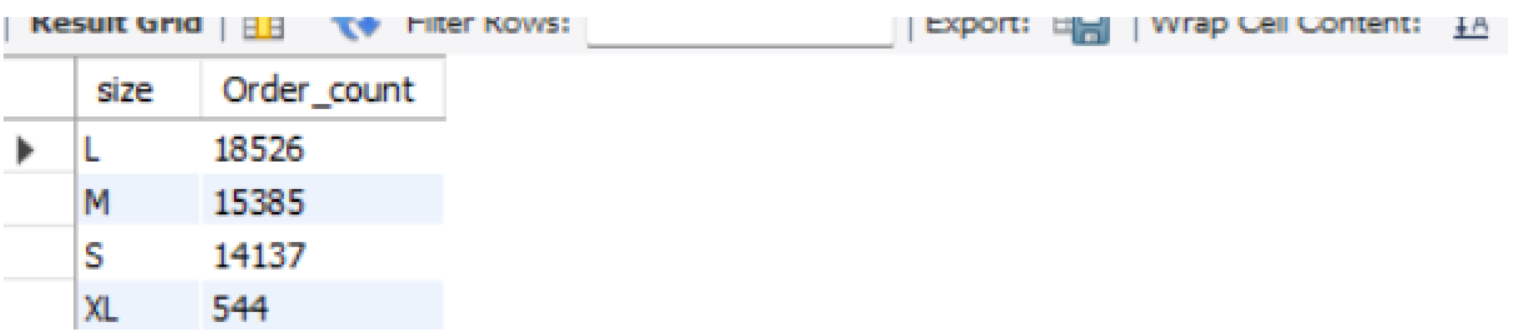
```
-- Identify the highest-priced pizza.--
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        INNER JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_i
ORDER BY pizzas.price DESC
LIMIT 1;
```



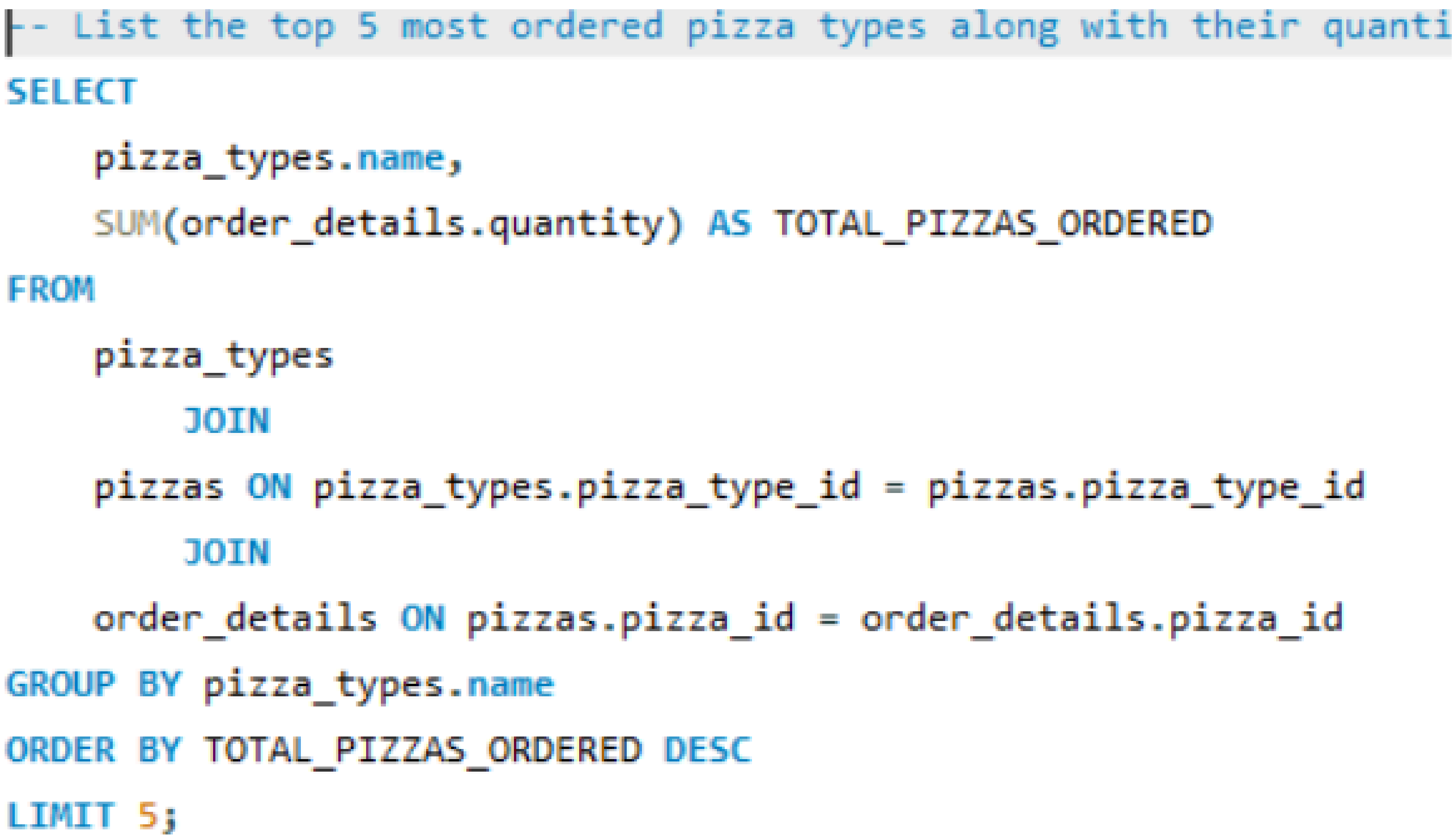
name	price
The Greek Pizza	35.95



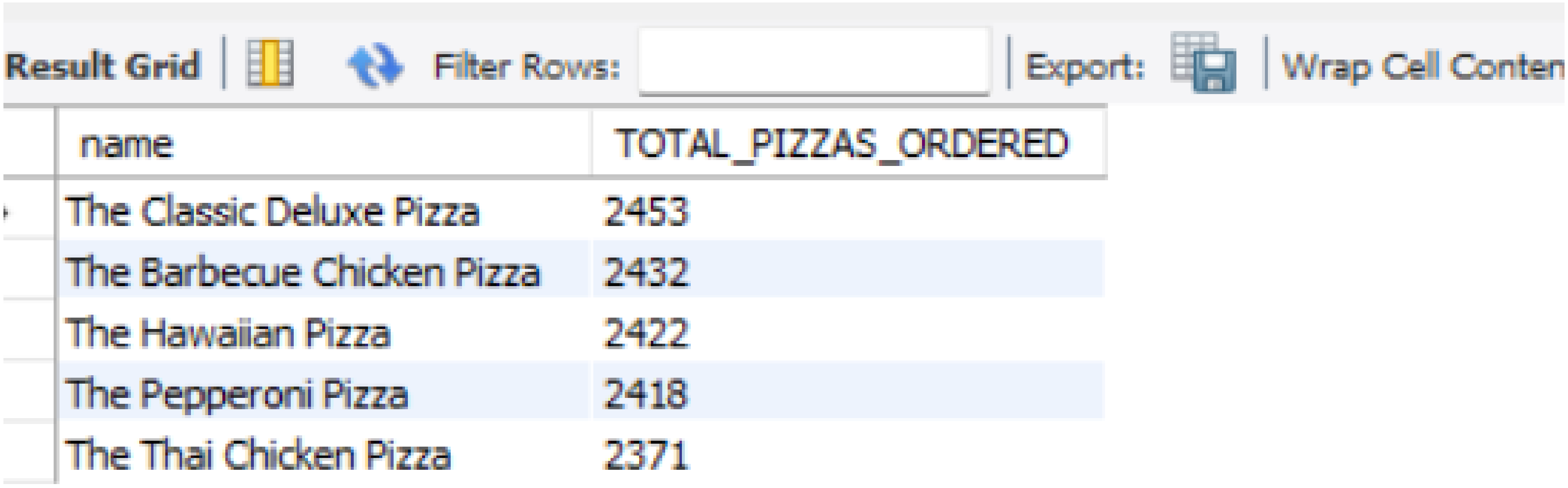
```
-- Identify the most common pizza size ordered
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS Order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_i
GROUP BY pizzas.size
ORDER BY Order_count DESC;
```



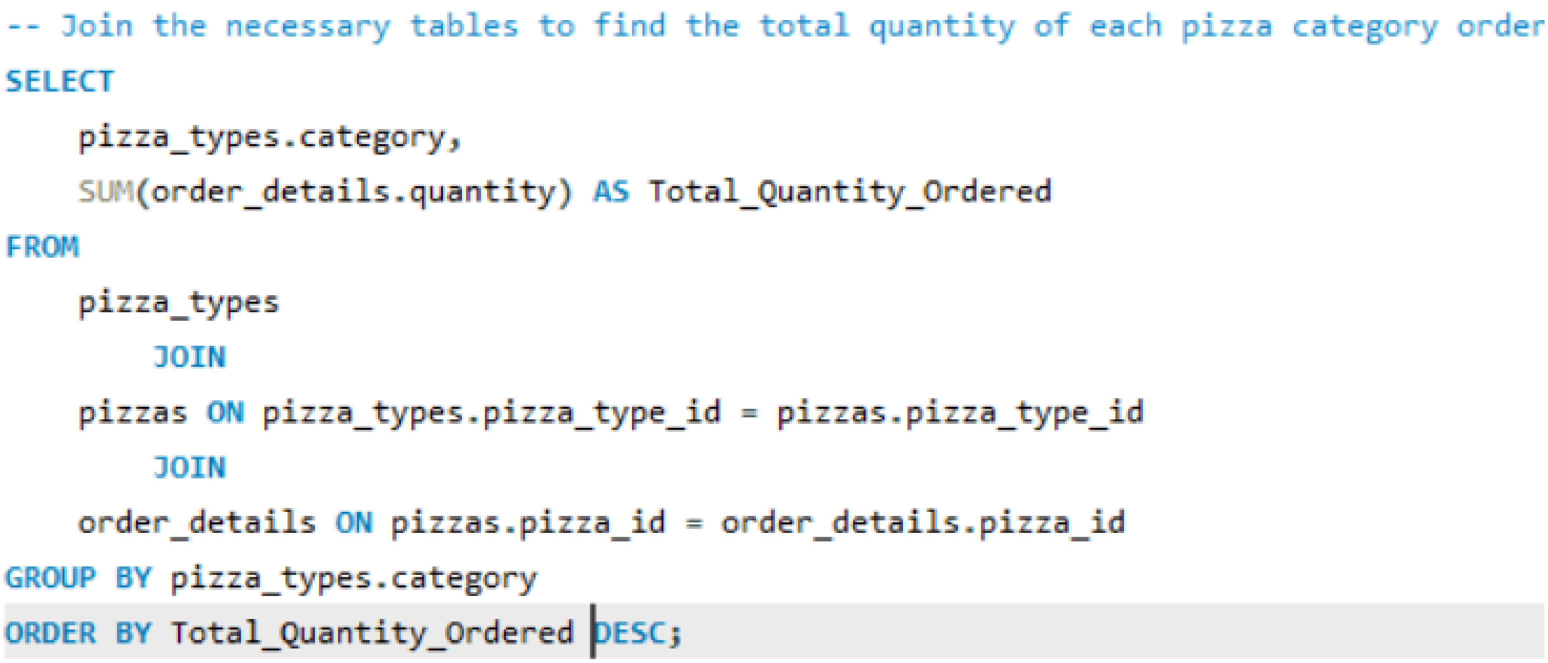
size	Order_count
L	18526
M	15385
S	14137
XL	544



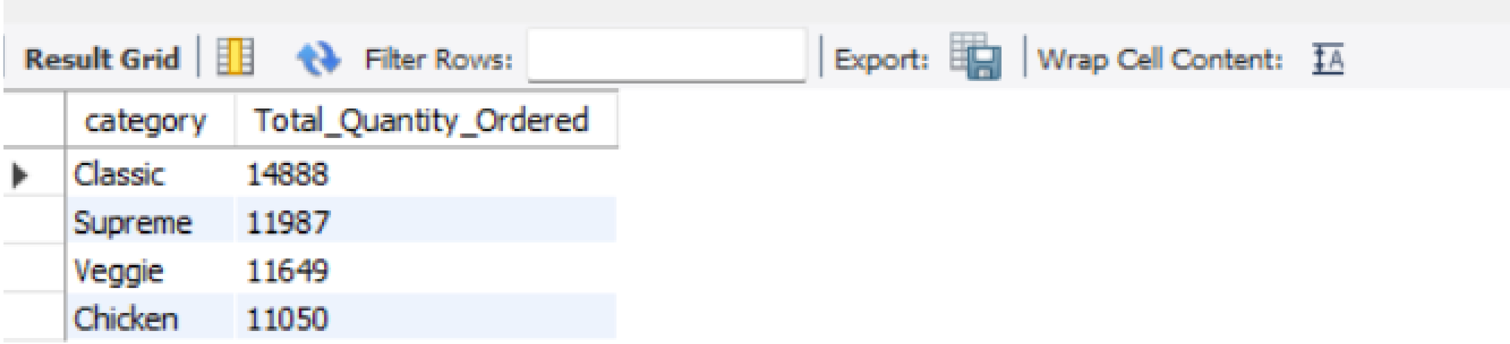
```sql
-- List the top 5 most ordered pizza types along with their quanti
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS TOTAL_PIZZAS_ORDERED
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY TOTAL_PIZZAS_ORDERED DESC
LIMIT 5;
```



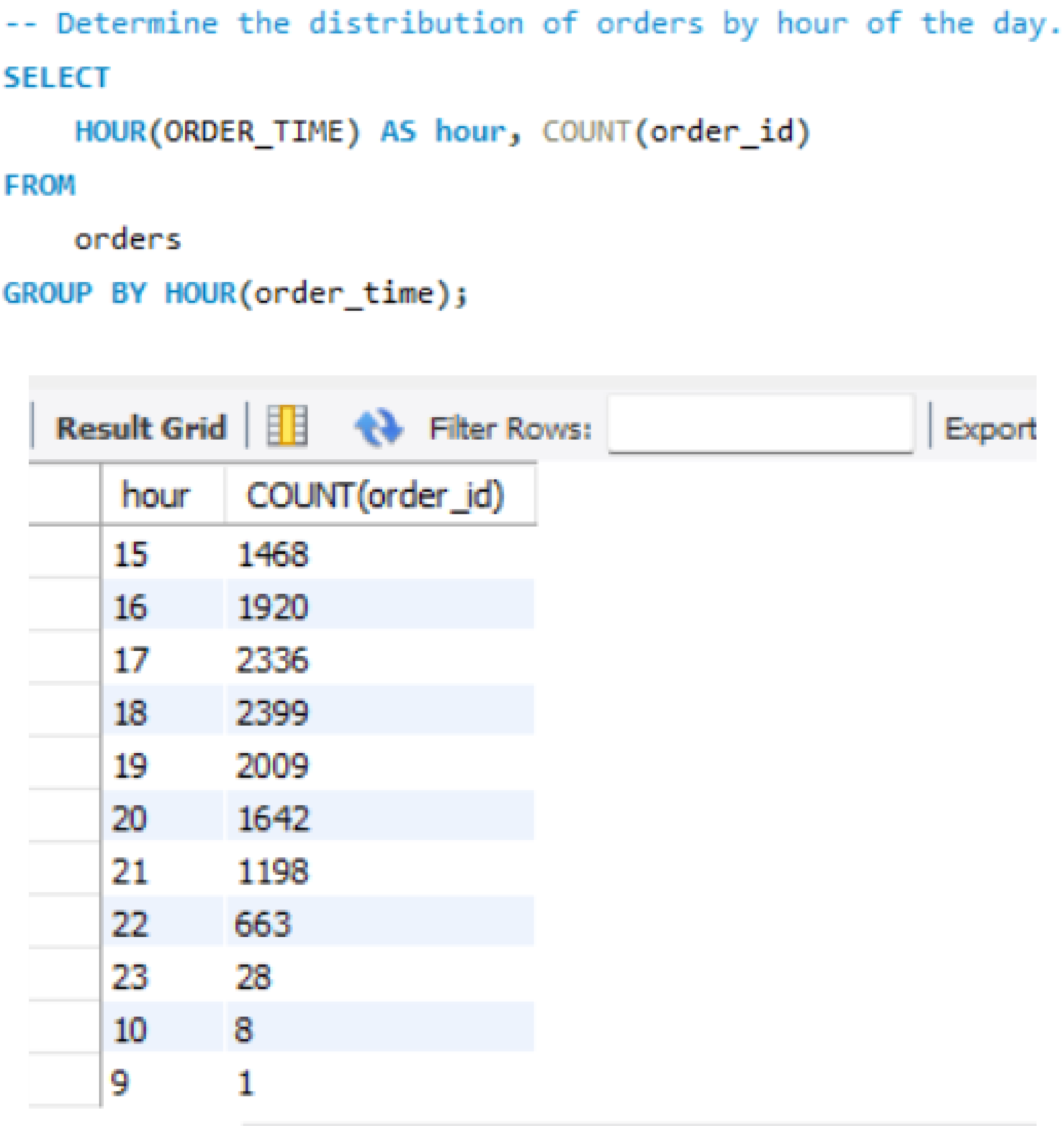
name	TOTAL_PIZZAS_ORDERED
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371



```sql
-- Join the necessary tables to find the total quantity of each pizza category order
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS Total_Quantity_Ordered
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 Total_Quantity_Ordered DESC;
```

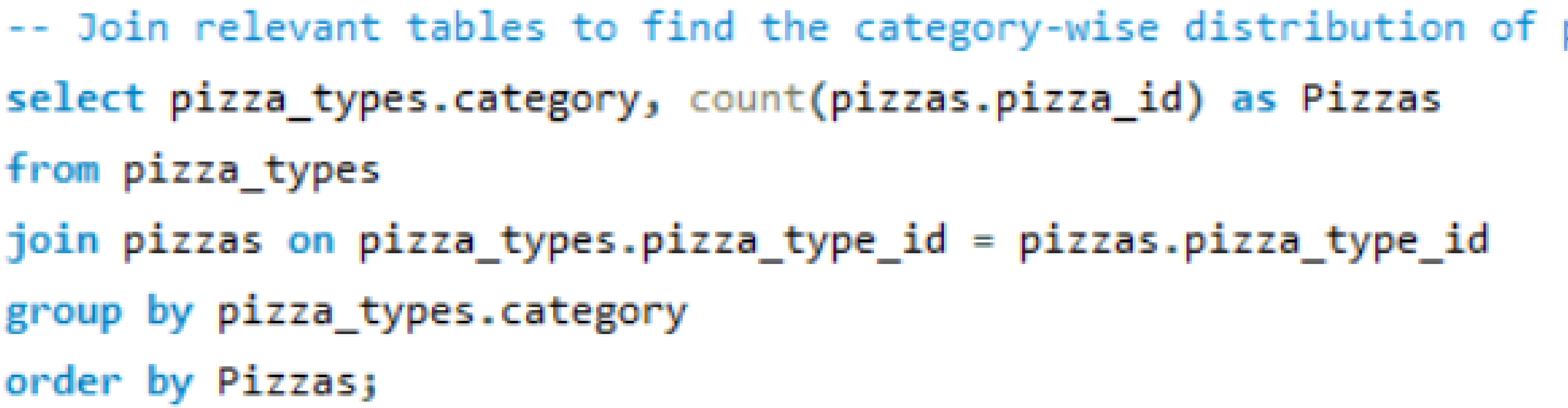


category	Total_Quantity_Ordered
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

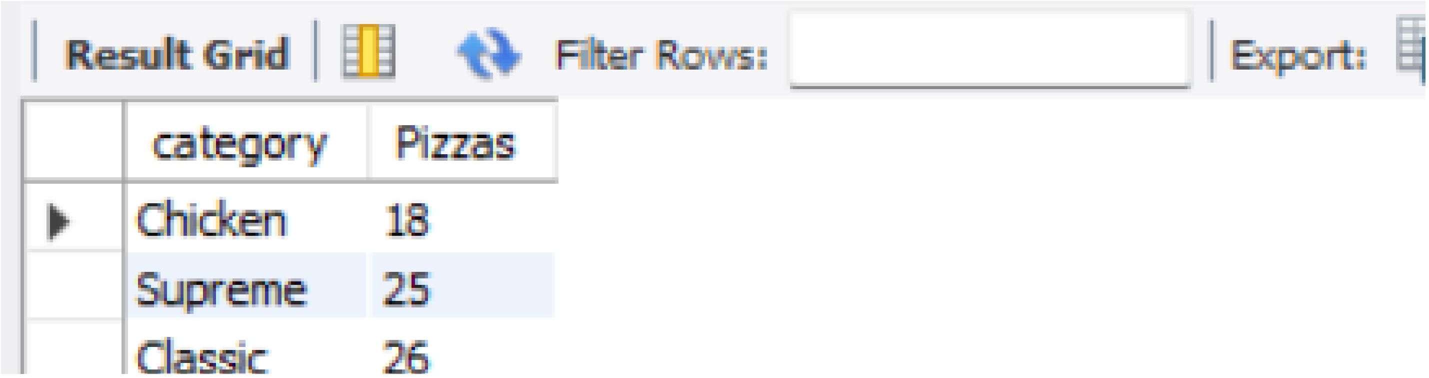


```sql
-- Determine the distribution of orders by hour of the day.
SELECT
    HOUR(ORDER_TIME) AS hour, COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time);
```

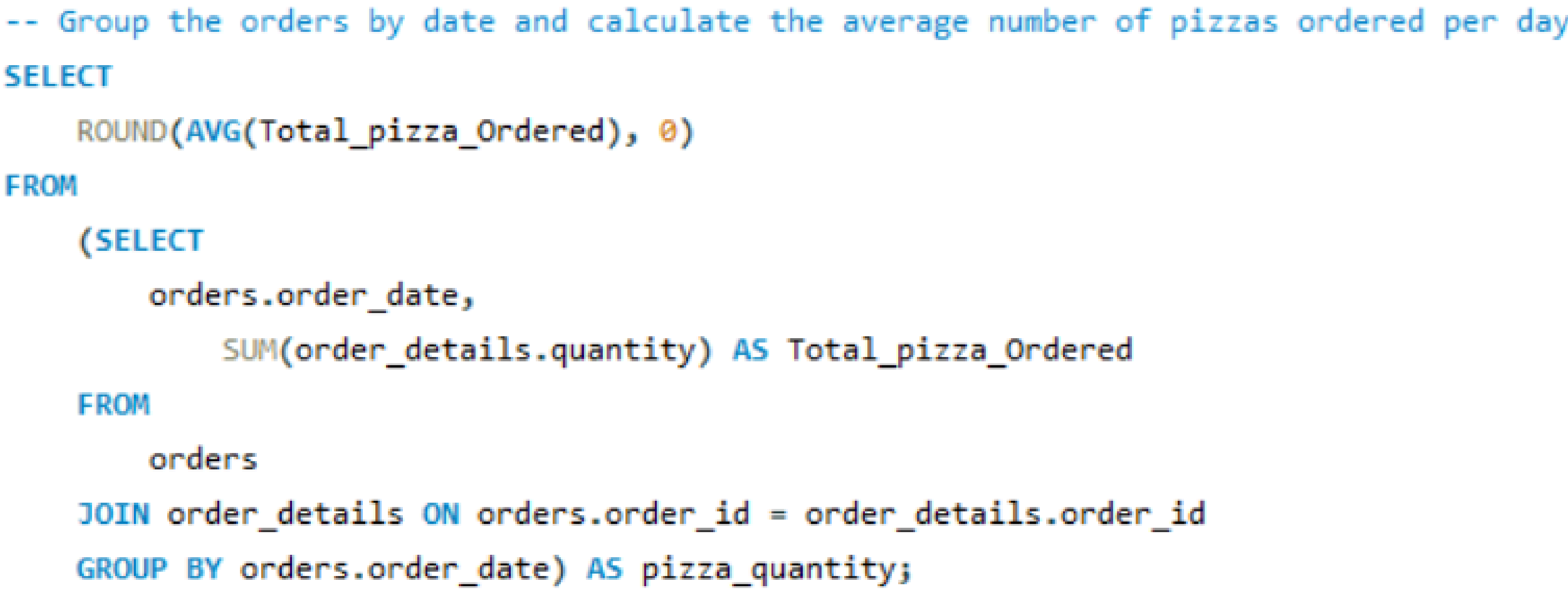
hour	COUNT(order_id)
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1



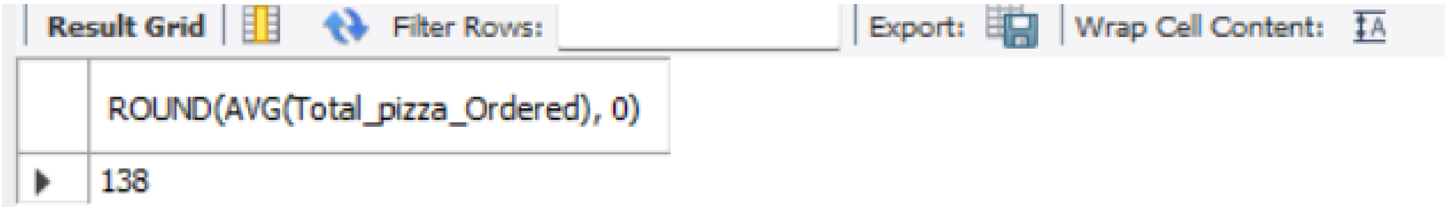
```
-- Join relevant tables to find the category-wise distribution of
select pizza_types.category, count(pizzas.pizza_id) as Pizzas
from pizza_types
join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
group by pizza_types.category
order by Pizzas;
```



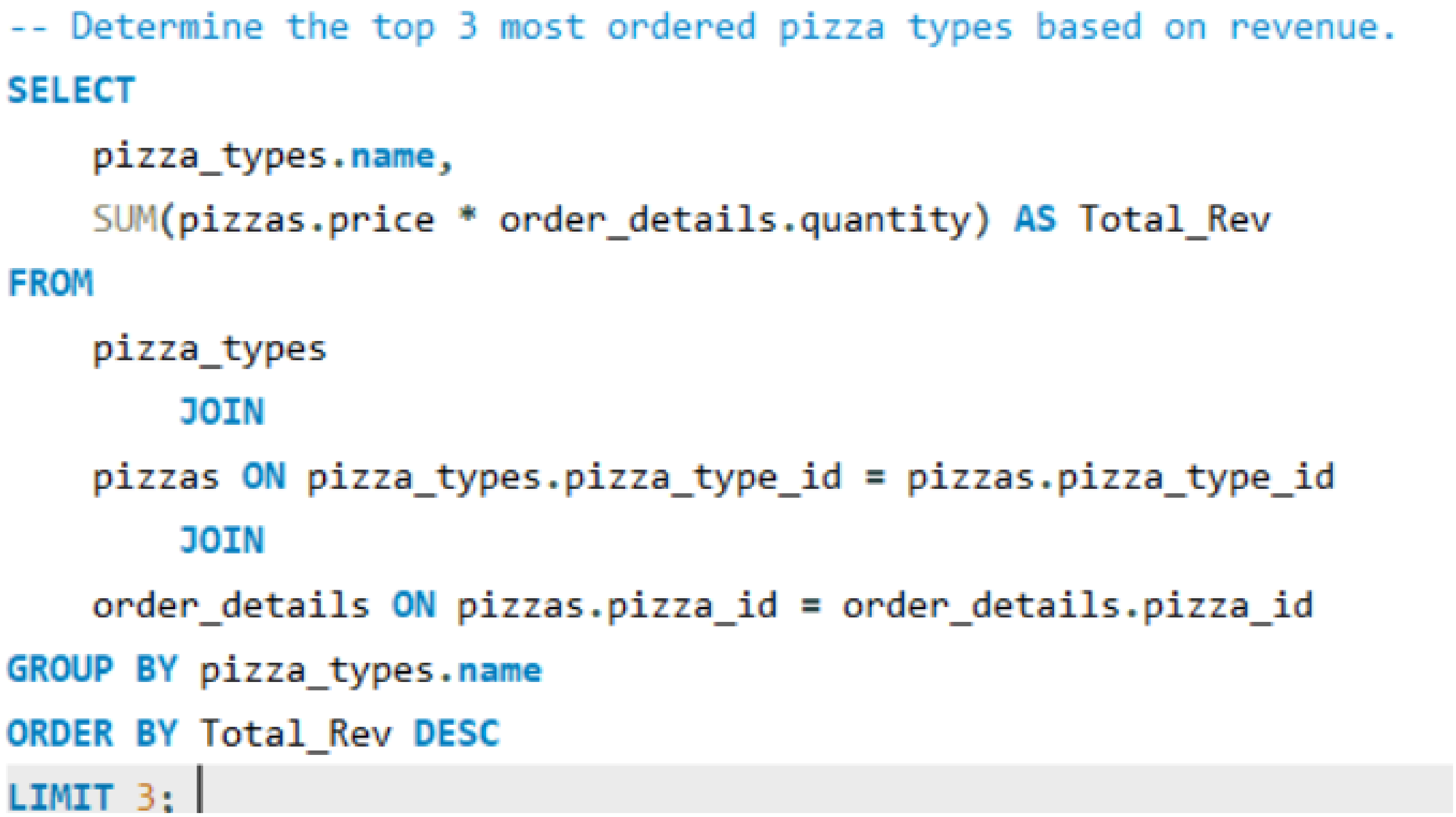
category	Pizzas
Chicken	18
Supreme	25
Classic	26



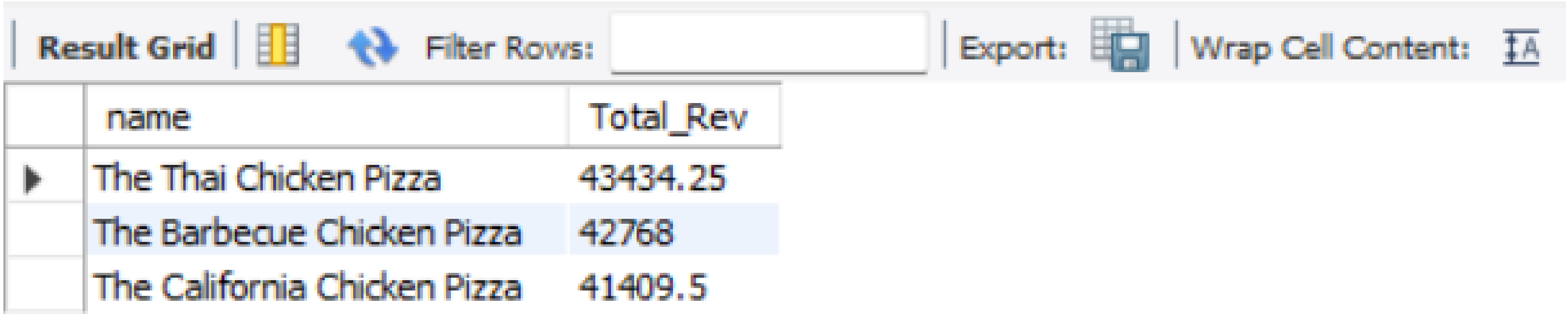
```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day
SELECT
    ROUND(AVG(Total_pizza_Ordered), 0)
FROM
    (SELECT
        orders.order_date,
            SUM(order_details.quantity) AS Total_pizza_Ordered
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS pizza_quantity;
```



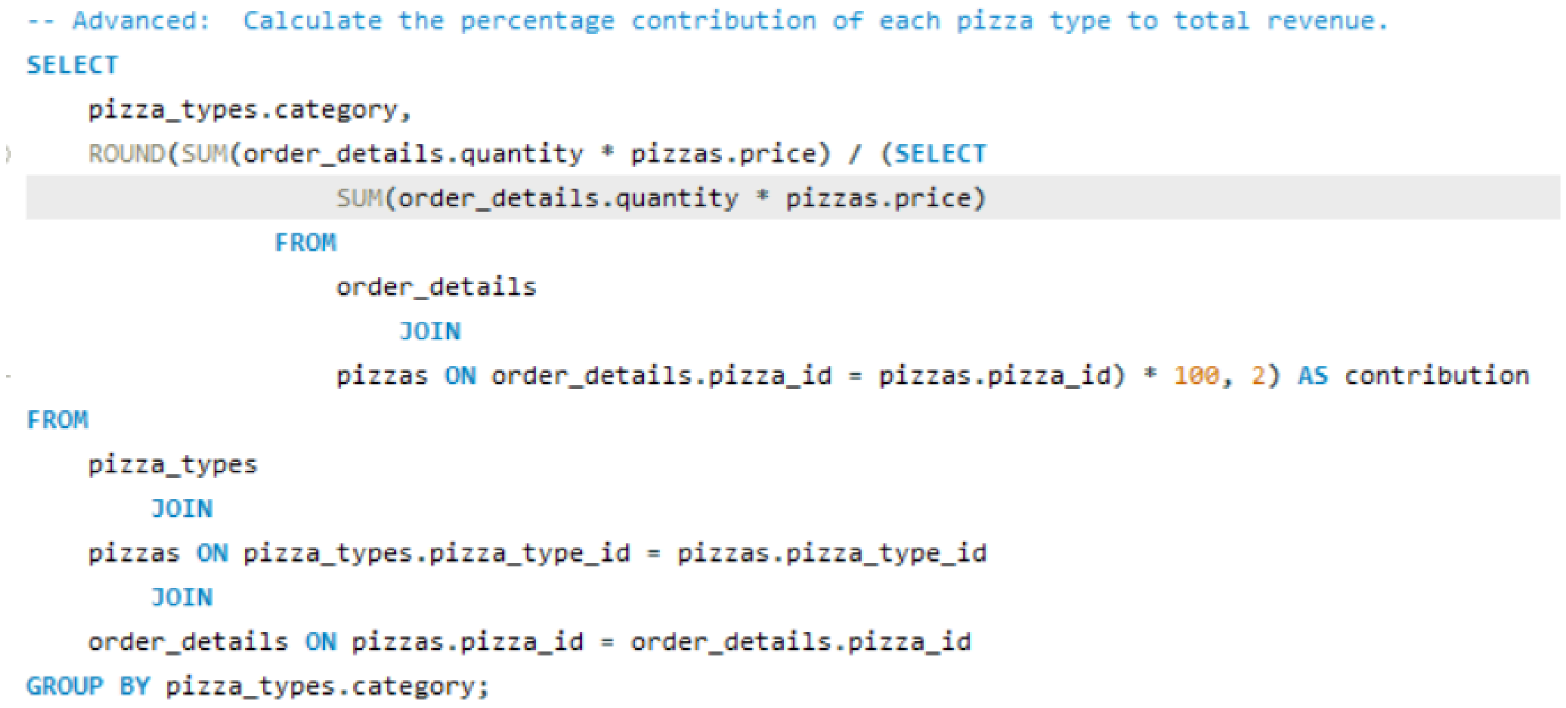
```
ROUND(AVG(Total_pizza_Ordered), 0)
138
```



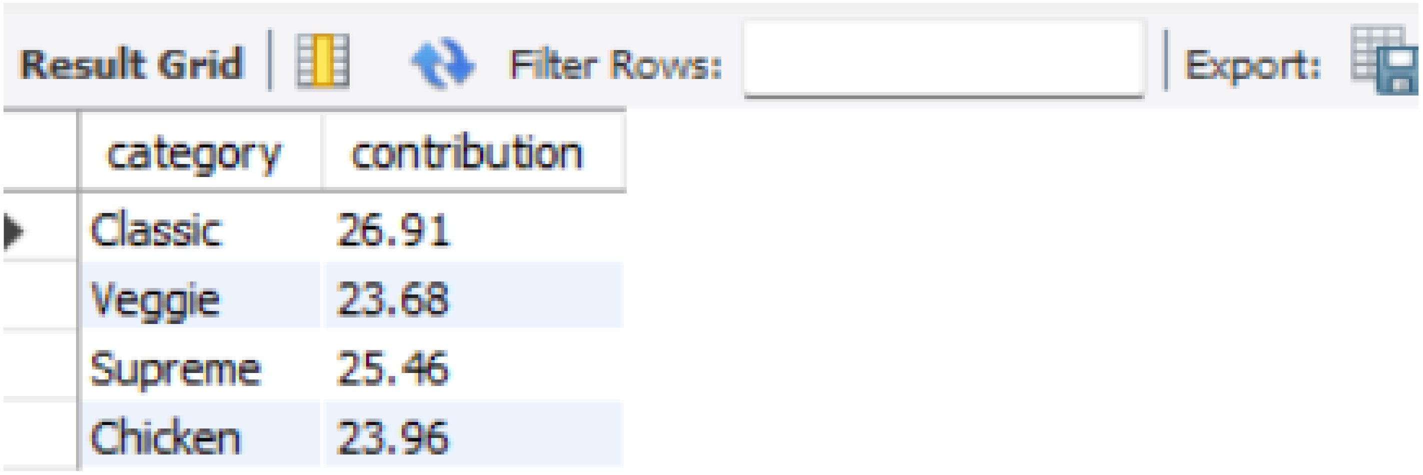
```sql
-- Determine the top 3 most ordered pizza types based on revenue.
SELECT
    pizza_types.name,
    SUM(pizzas.price * order_details.quantity) AS Total_Rev
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY Total_Rev DESC
LIMIT 3;
```



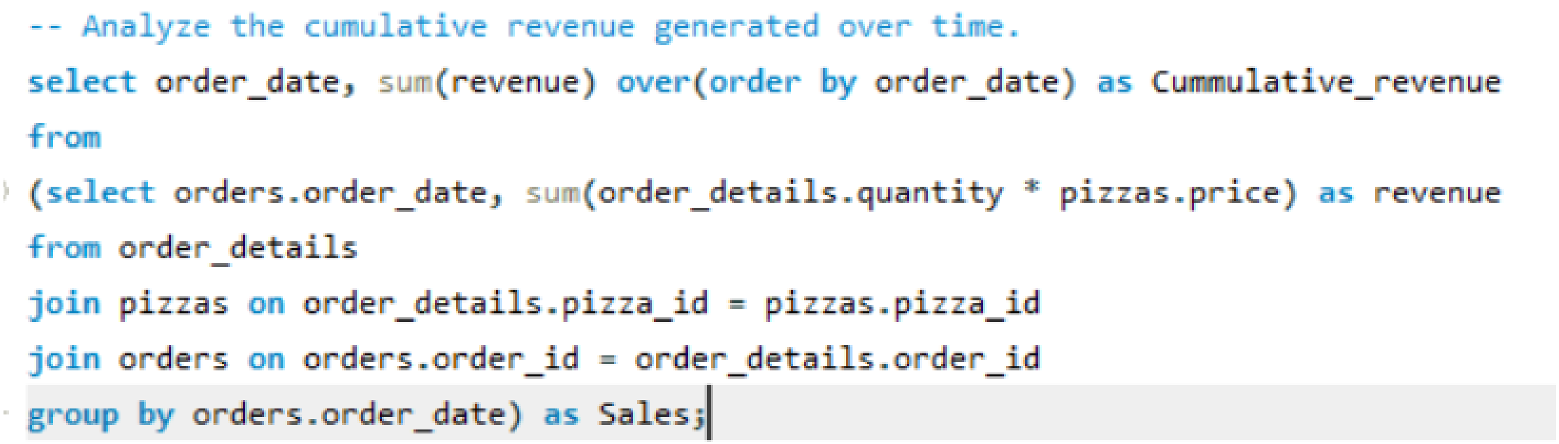
name	Total_Rev
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5



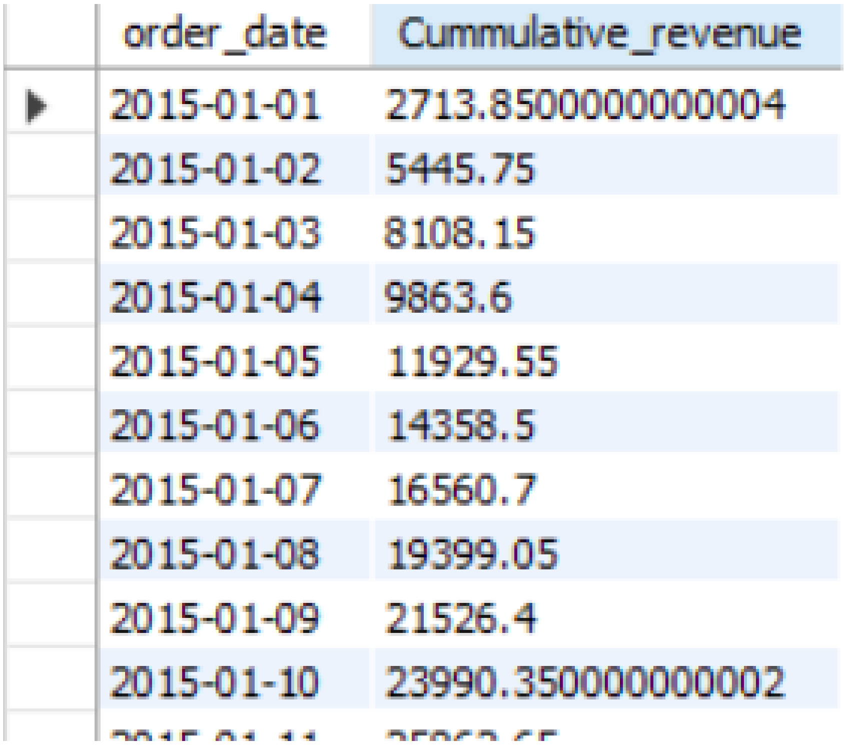
```
-- Advanced:  Calculate the percentage contribution of each pizza type to total revenue.
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    SUM(order_details.quantity * pizzas.price)
                FROM
                    order_details
                        JOIN
                    pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100, 2) AS contribution
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;
```



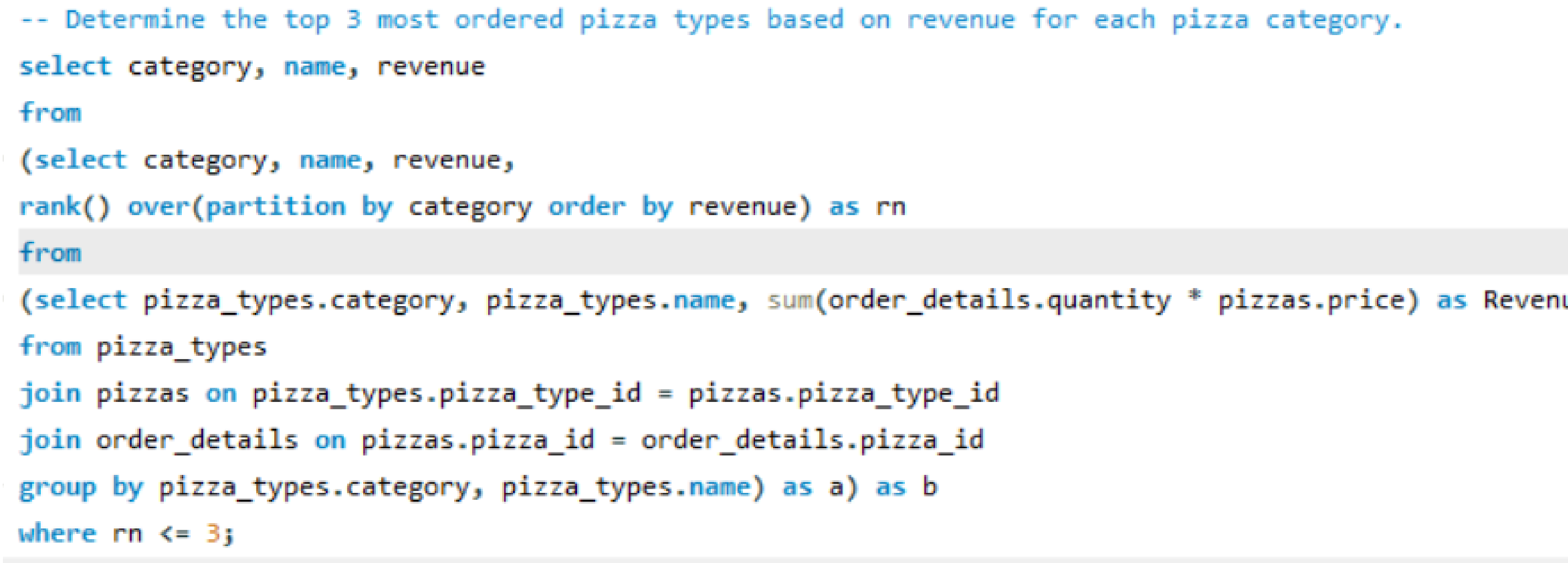
category	contribution
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96



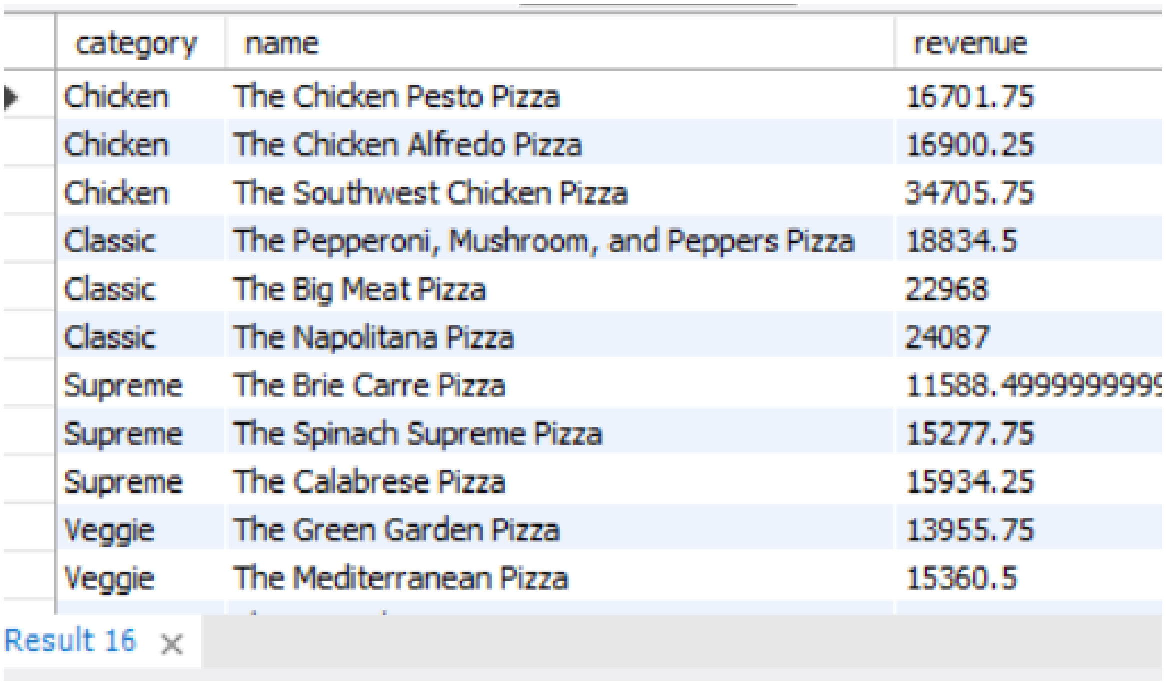
```
-- Analyze the cumulative revenue generated over time.
select order_date, sum(revenue) over(order by order_date) as Cummulative_revenue
from
(select orders.order_date, sum(order_details.quantity * pizzas.price) as revenue
from order_details
join pizzas on order_details.pizza_id = pizzas.pizza_id
join orders on orders.order_id = order_details.order_id
group by orders.order_date) as Sales;
```



order_date	Cummulative_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



```
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
select category, name, revenue
from
(select category, name, revenue,
rank() over(partition by category order by revenue) as rn
from
(select pizza_types.category, pizza_types.name, sum(order_details.quantity * pizzas.price) as Revenu
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, pizza_types.name) as a) as b
where rn <= 3;
```



category	name	revenue
Chicken	The Chicken Pesto Pizza	16701.75
Chicken	The Chicken Alfredo Pizza	16900.25
Chicken	The Southwest Chicken Pizza	34705.75
Classic	The Pepperoni, Mushroom, and Peppers Pizza	18834.5
Classic	The Big Meat Pizza	22968
Classic	The Napolitana Pizza	24087
Supreme	The Brie Carre Pizza	11588.499999999
Supreme	The Spinach Supreme Pizza	15277.75
Supreme	The Calabrese Pizza	15934.25
Veggie	The Green Garden Pizza	13955.75
Veggie	The Mediterranean Pizza	15360.5

PIZZA SALE- SQL

THANK YOU