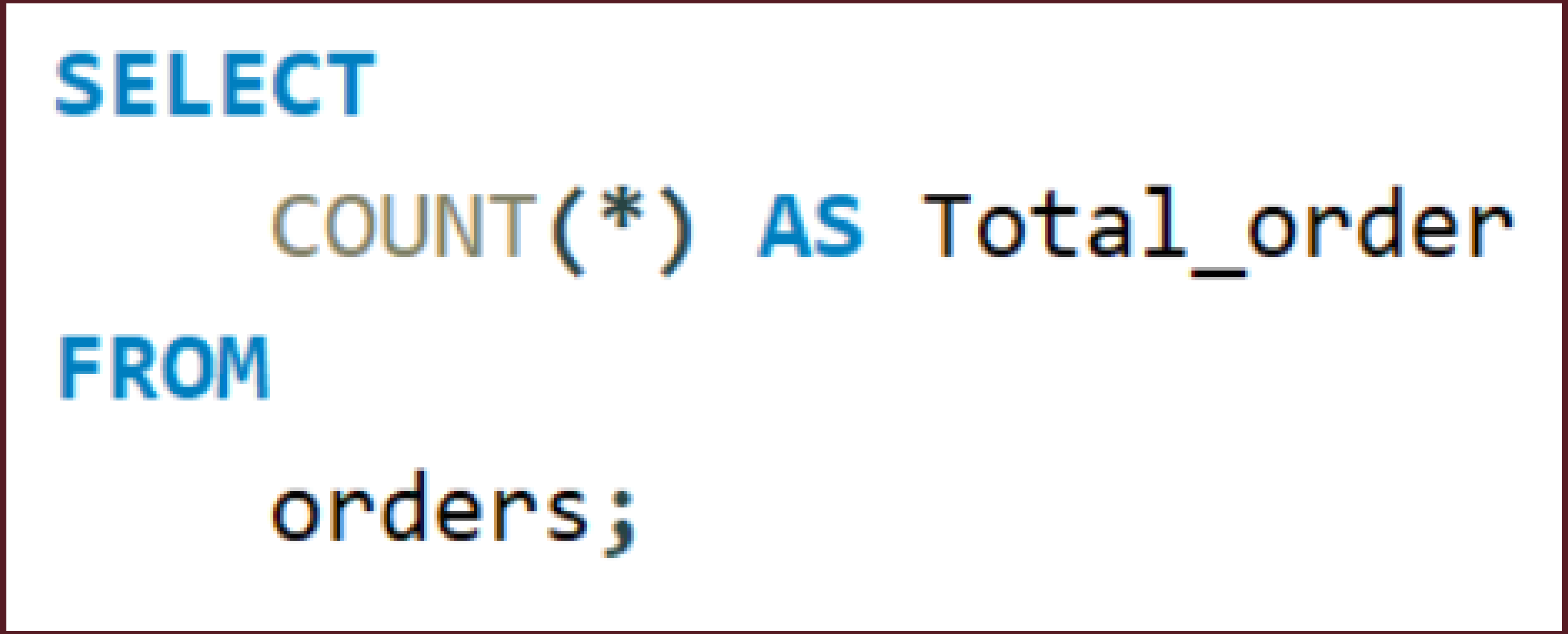
PIZZA SALES ANALYSIS USING SQL

PROJECT OVERVIEW

- **Objective**: Analyze pizza sales data to gain insights into sales performance, customer preferences, and profitability.
- **Data Source**: The data used for the analysis was collected from a fictional pizza store's database.
- **Tools Used:** SQL for data extraction, transformation, and analysis.
- **Key Metrics Analyzed**: Total Sales, Quantity Sold, Popular Pizza Types, Peak Sales Times, and Customer Segmentation.
- **Outcome**: The project provides actionable insights that can help in making data-driven decisions to improve sales and optimize operations.

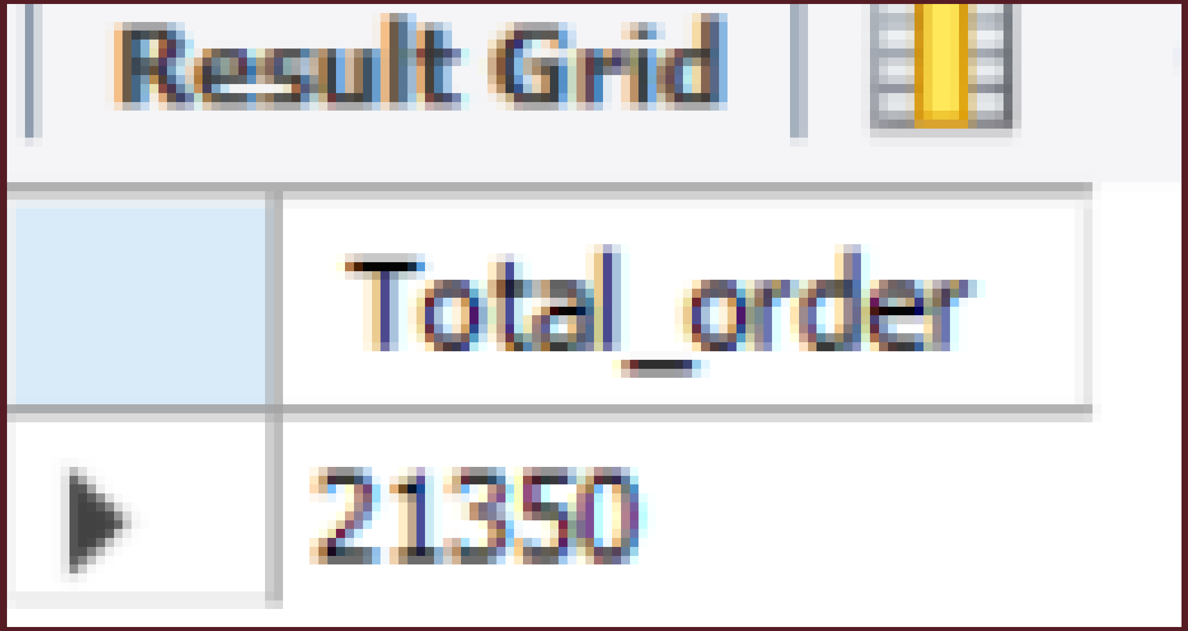
Retrieve the total number of orders placed.

Query



```
SELECT
    COUNT(*) AS Total_order
FROM
    orders;
```

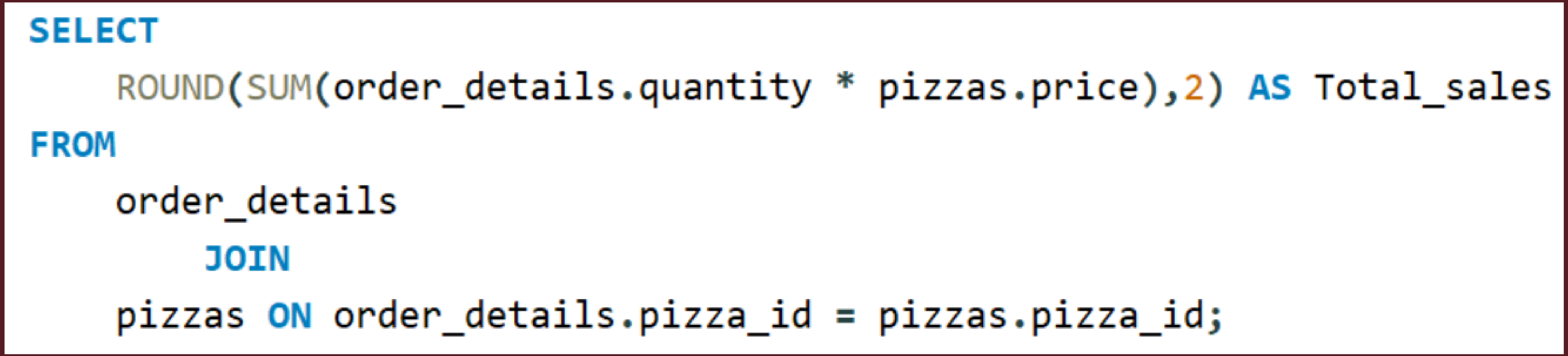
Output



```
Total_order
21350
```

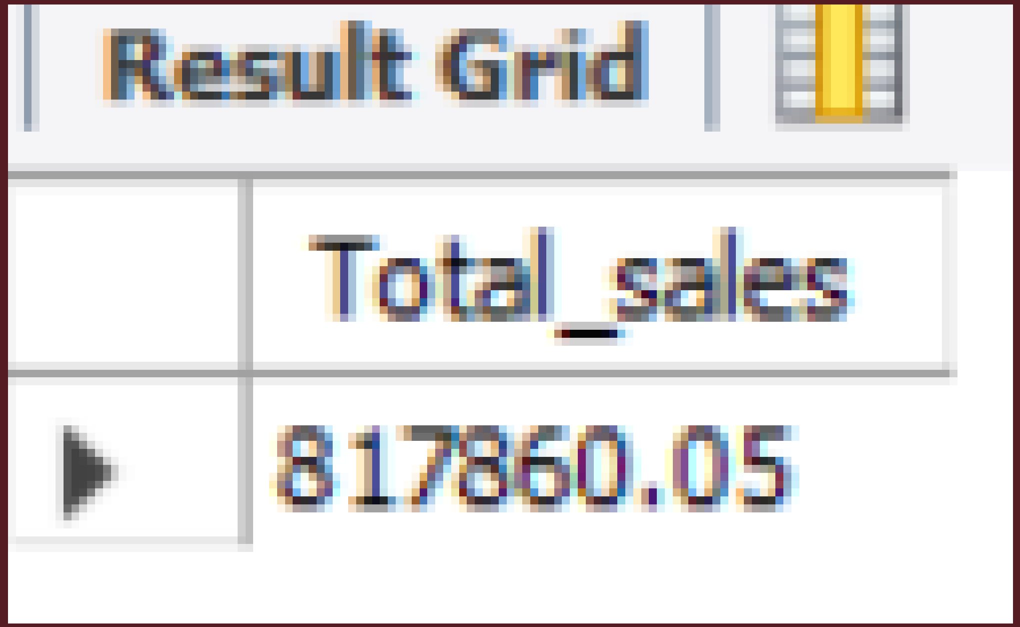
Calculate the total revenue generated from pizza sales.

Query



```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),2) AS Total_sales
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

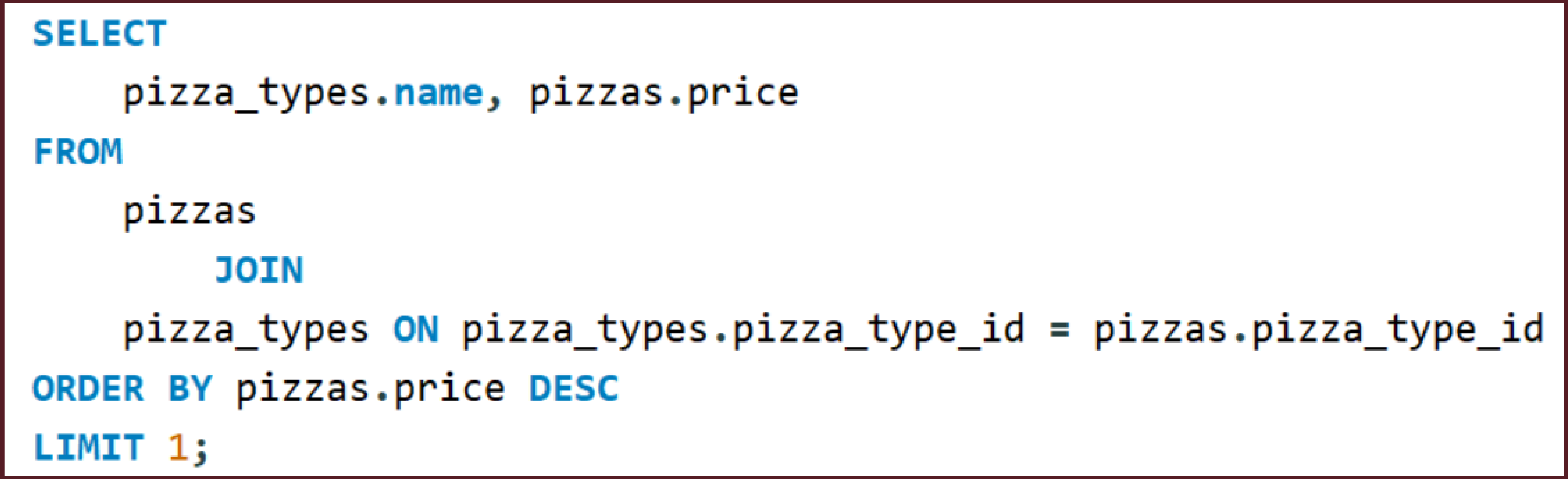
Output



```
Result Grid
  Total_sales
▶ 817860.05
```

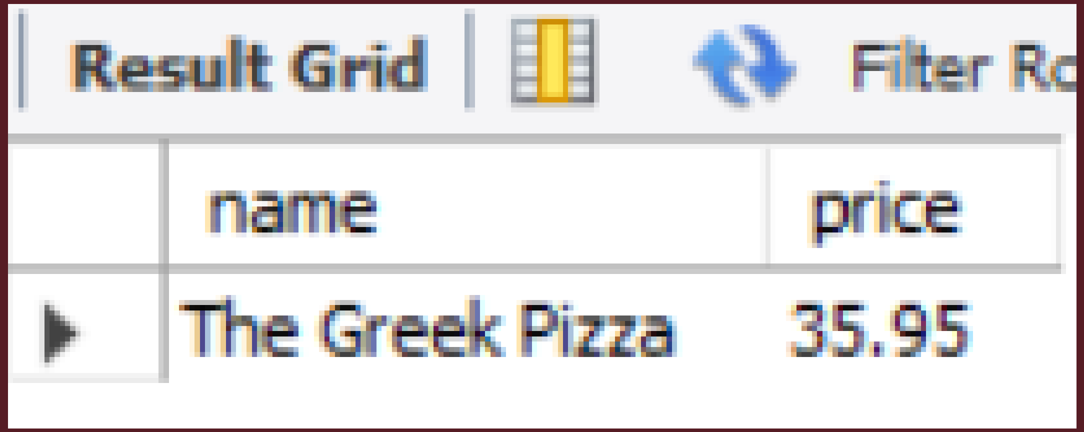
Identify the highest-priced pizza.

Query



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

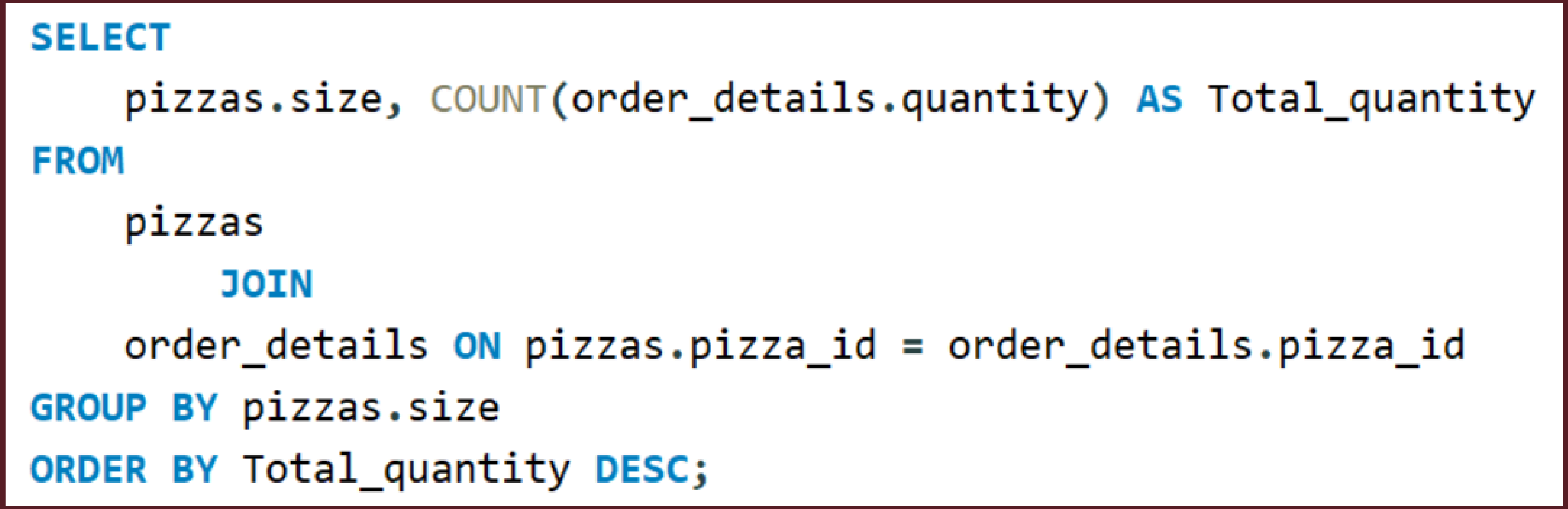
Output



name	price
The Greek Pizza	35.95

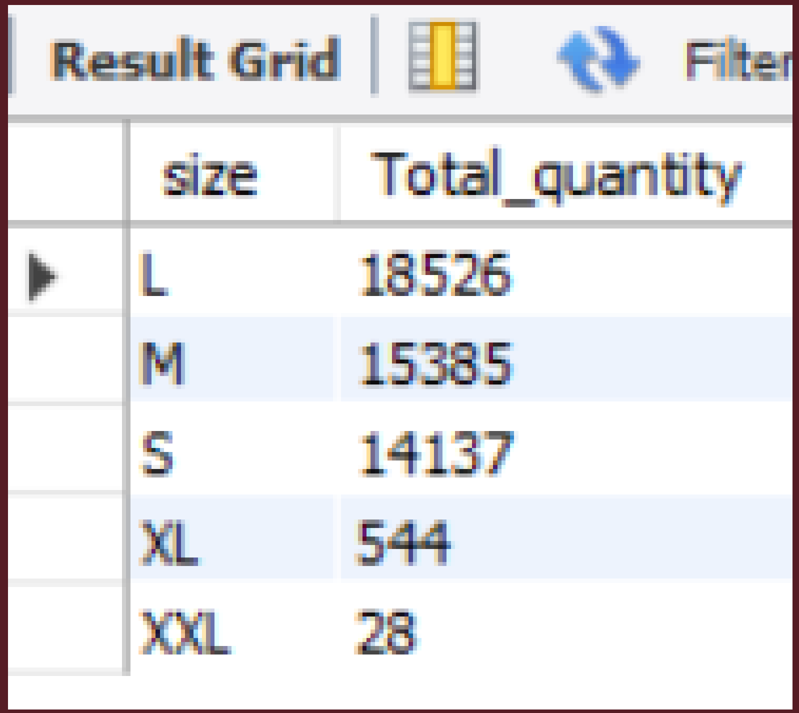
Identify the most common pizza size ordered.

Query



```
SELECT
    pizzas.size, COUNT(order_details.quantity) AS Total_quantity
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY Total_quantity DESC;
```

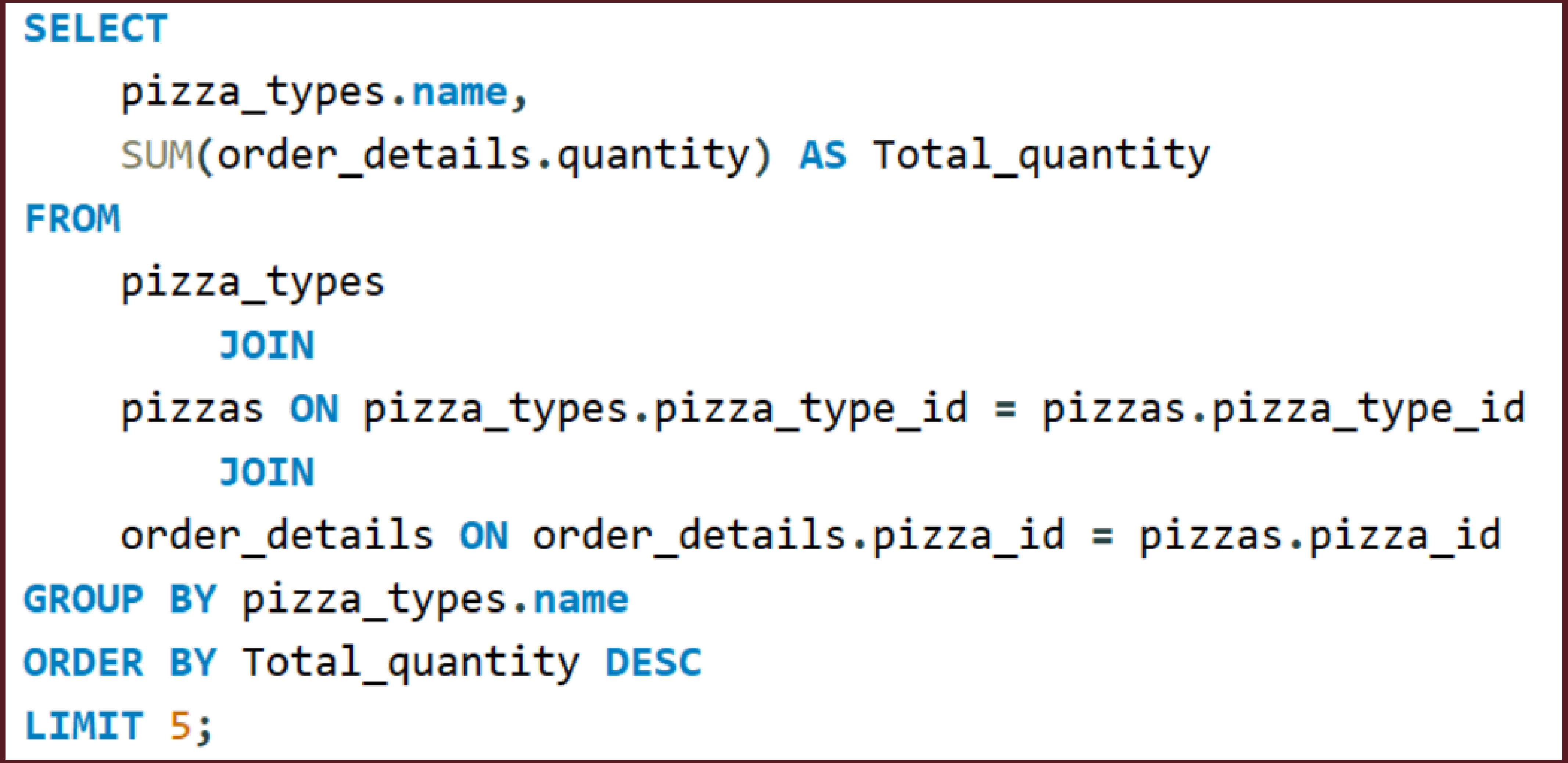
Output



size	Total_quantity
L	18526
M	15385
S	14137
XL	544
XXL	28

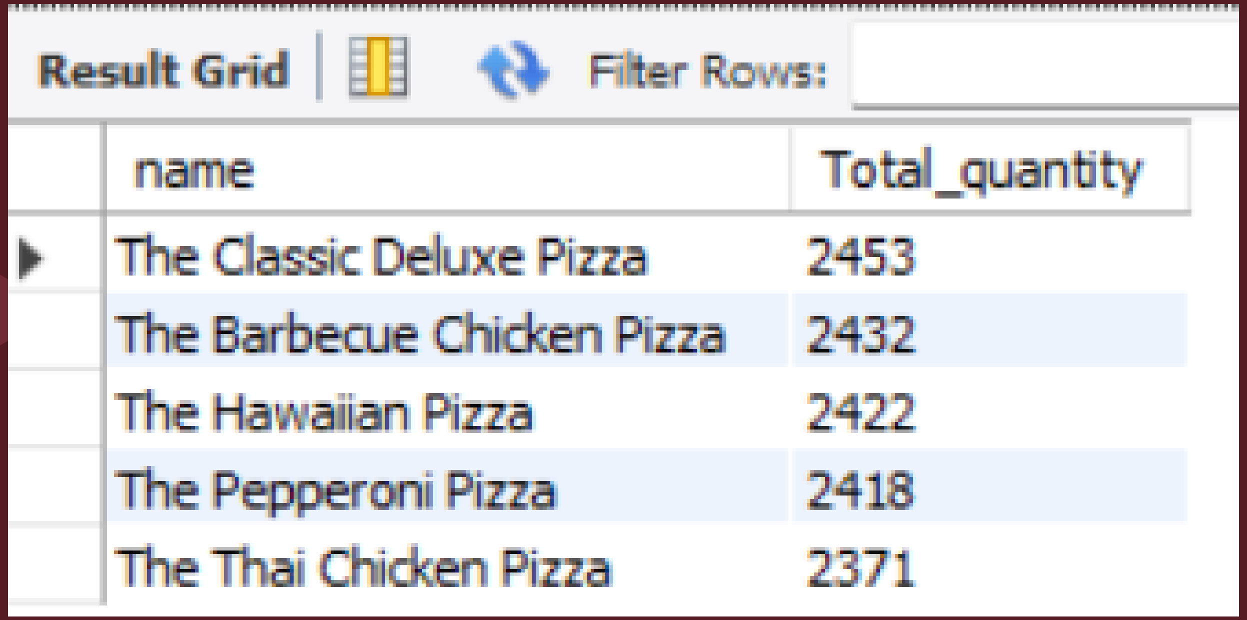
List the top 5 most ordered pizza types along with their quantities.

Query



```
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS Total_quantity
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.name
ORDER BY Total_quantity DESC
LIMIT 5;
```

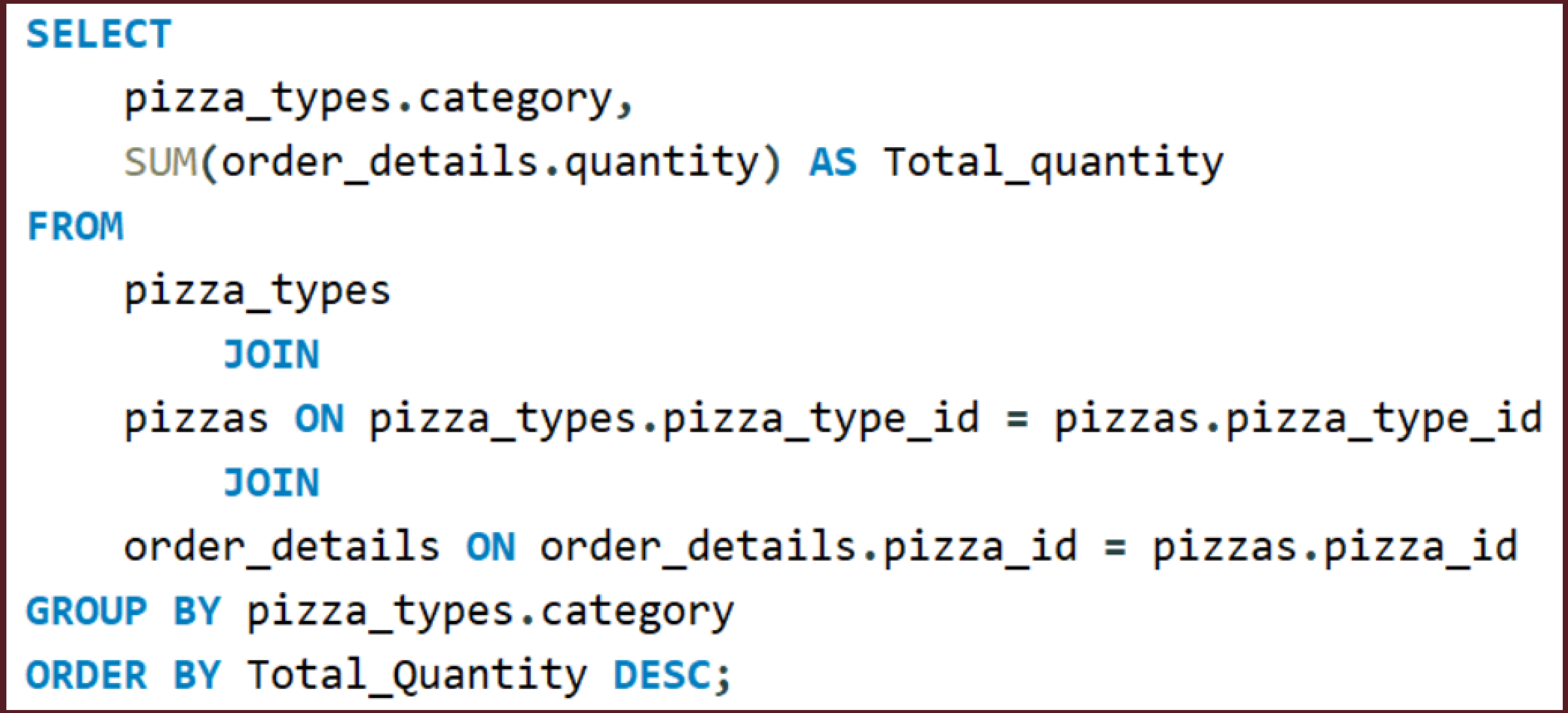
Output



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

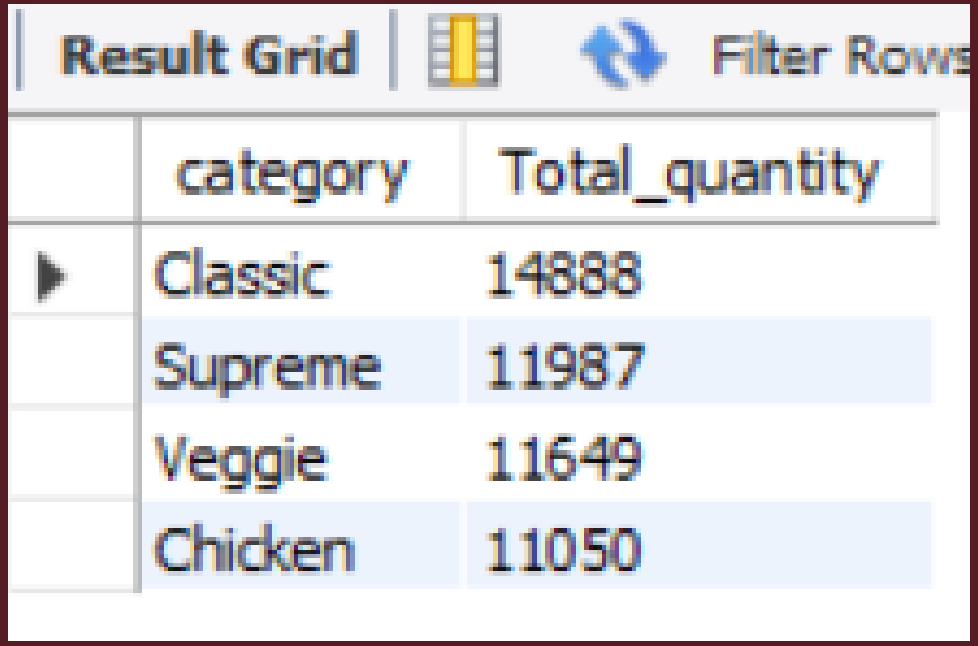
Join the necessary tables to find the total quantity of each pizza category ordered.

Query



```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS Total_quantity
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
ORDER BY Total_Quantity DESC;
```

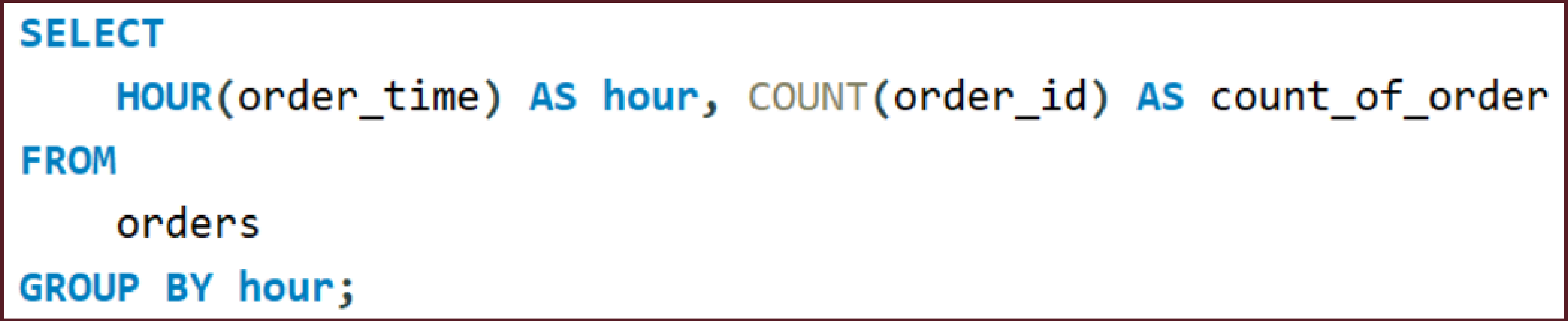
Output



category	Total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

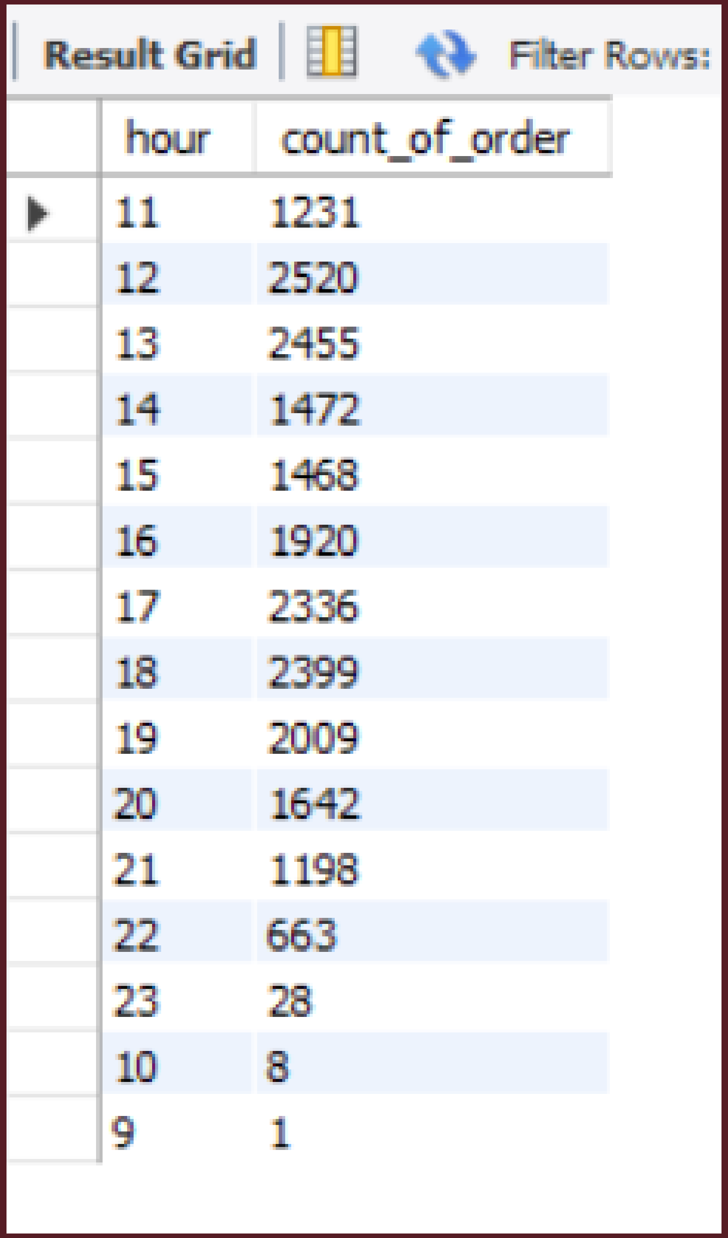
Determine the distribution of orders by hour of the day.

Query



```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS count_of_order
FROM
    orders
GROUP BY hour;
```

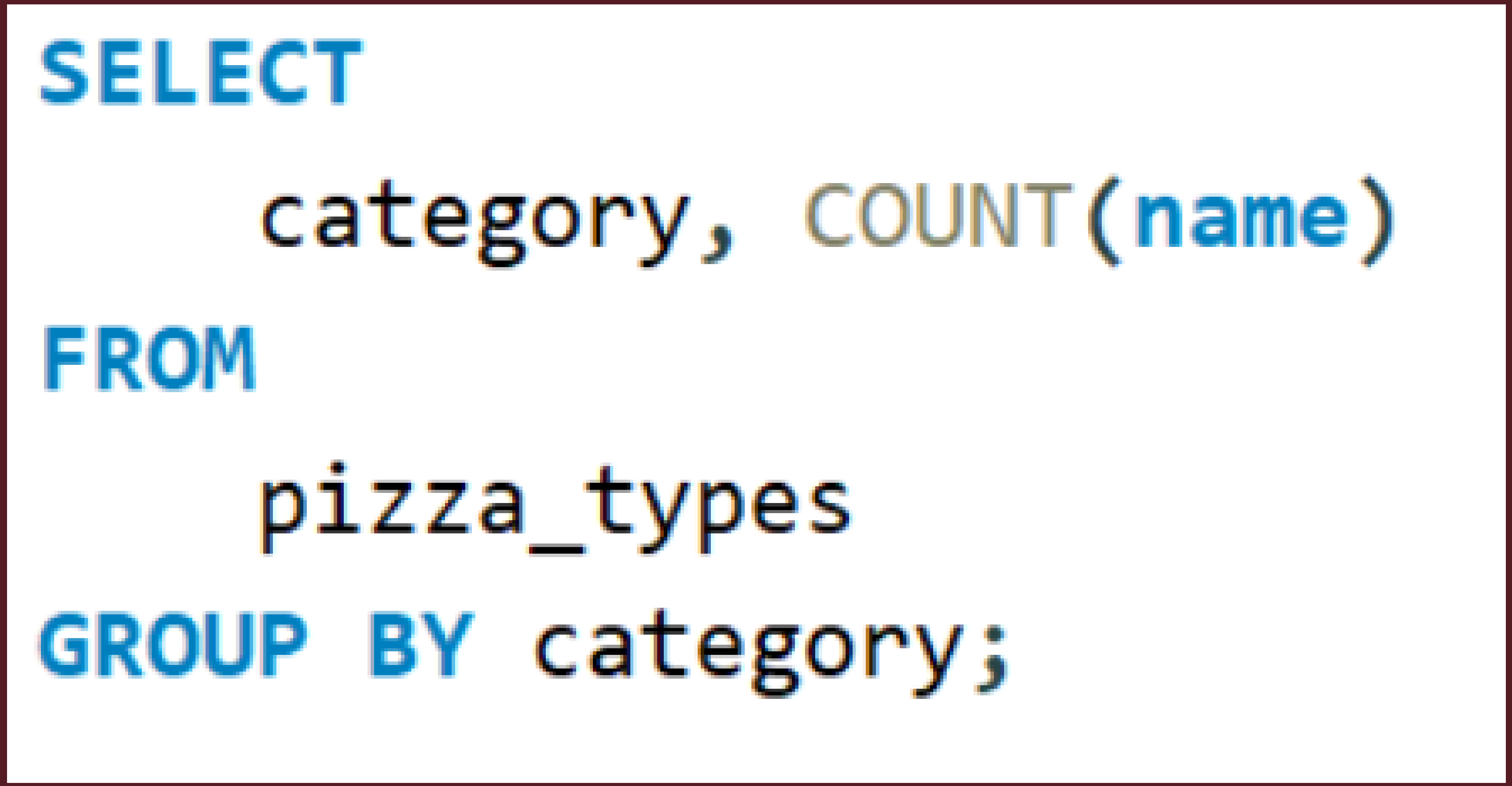
Output



hour	count_of_order
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

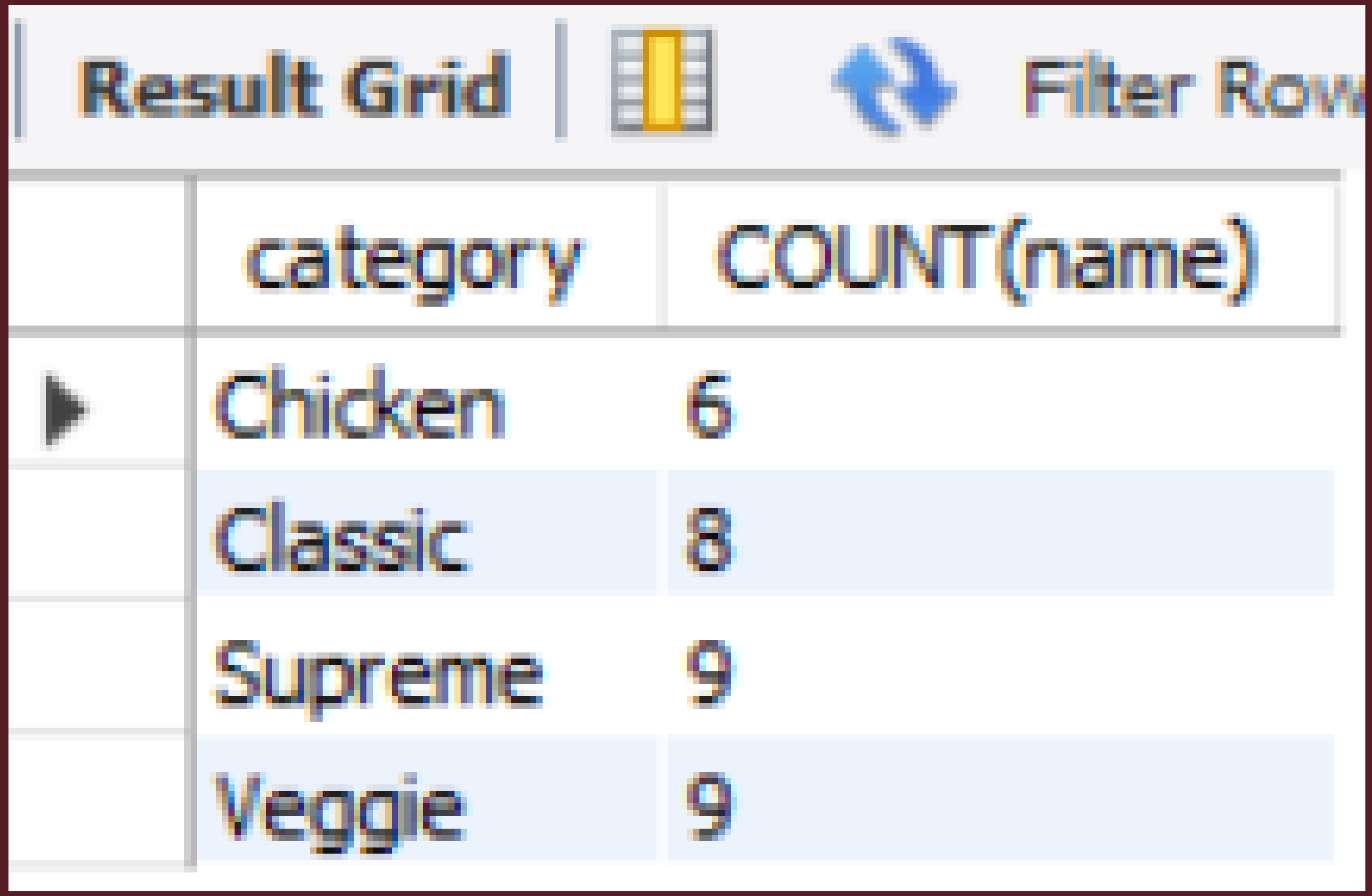
Join relevant tables to find the category-wise distribution of pizzas.

Query



```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

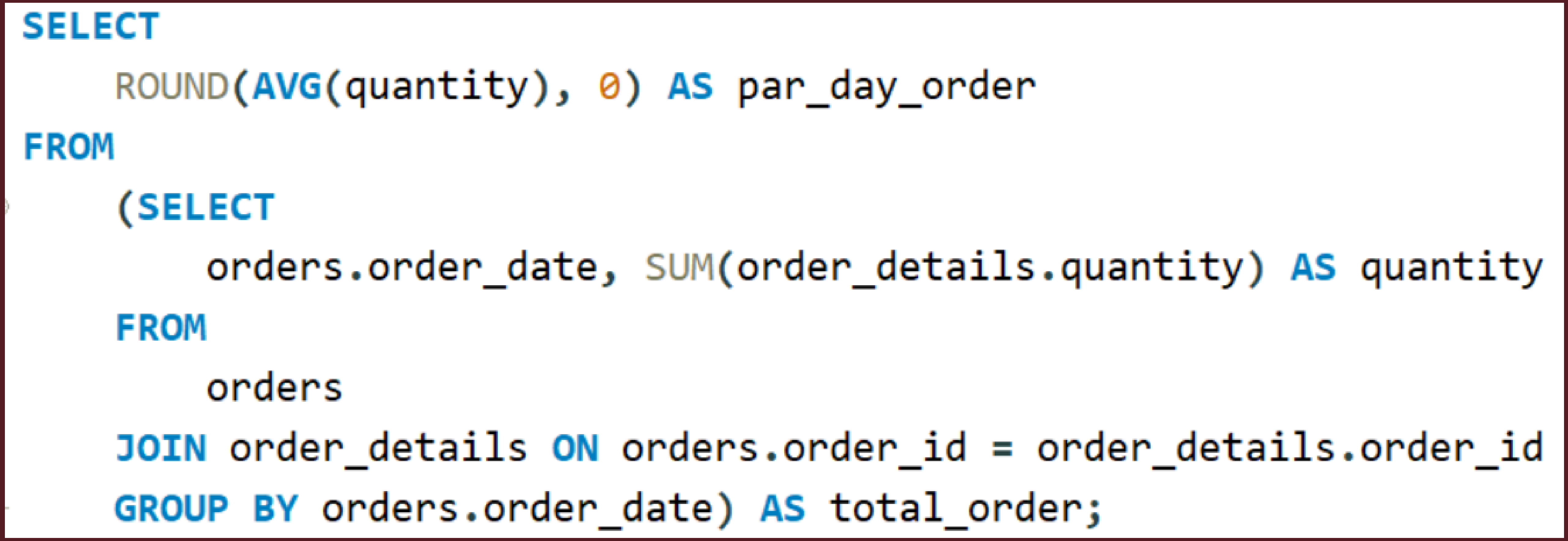
Output



category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

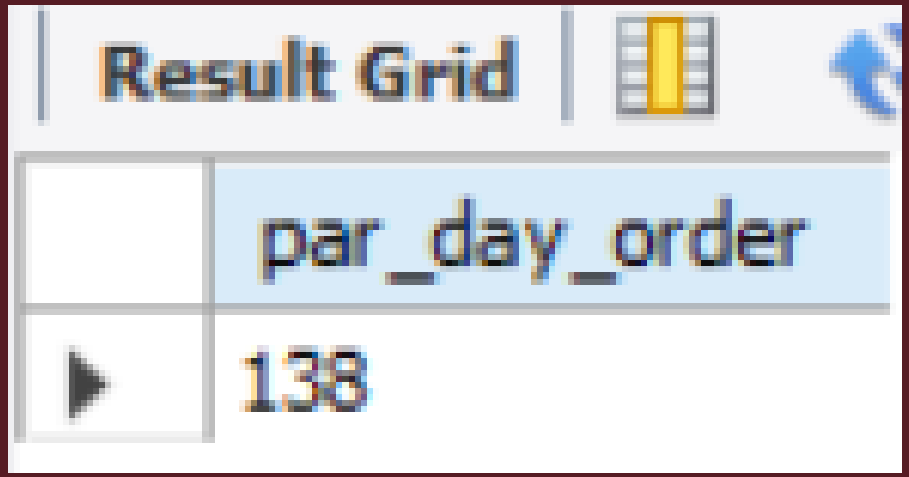
Group the orders by date and calculate the average number of pizzas ordered per day.

Query



```
SELECT
    ROUND(AVG(quantity), 0) AS par_day_order
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS total_order;
```

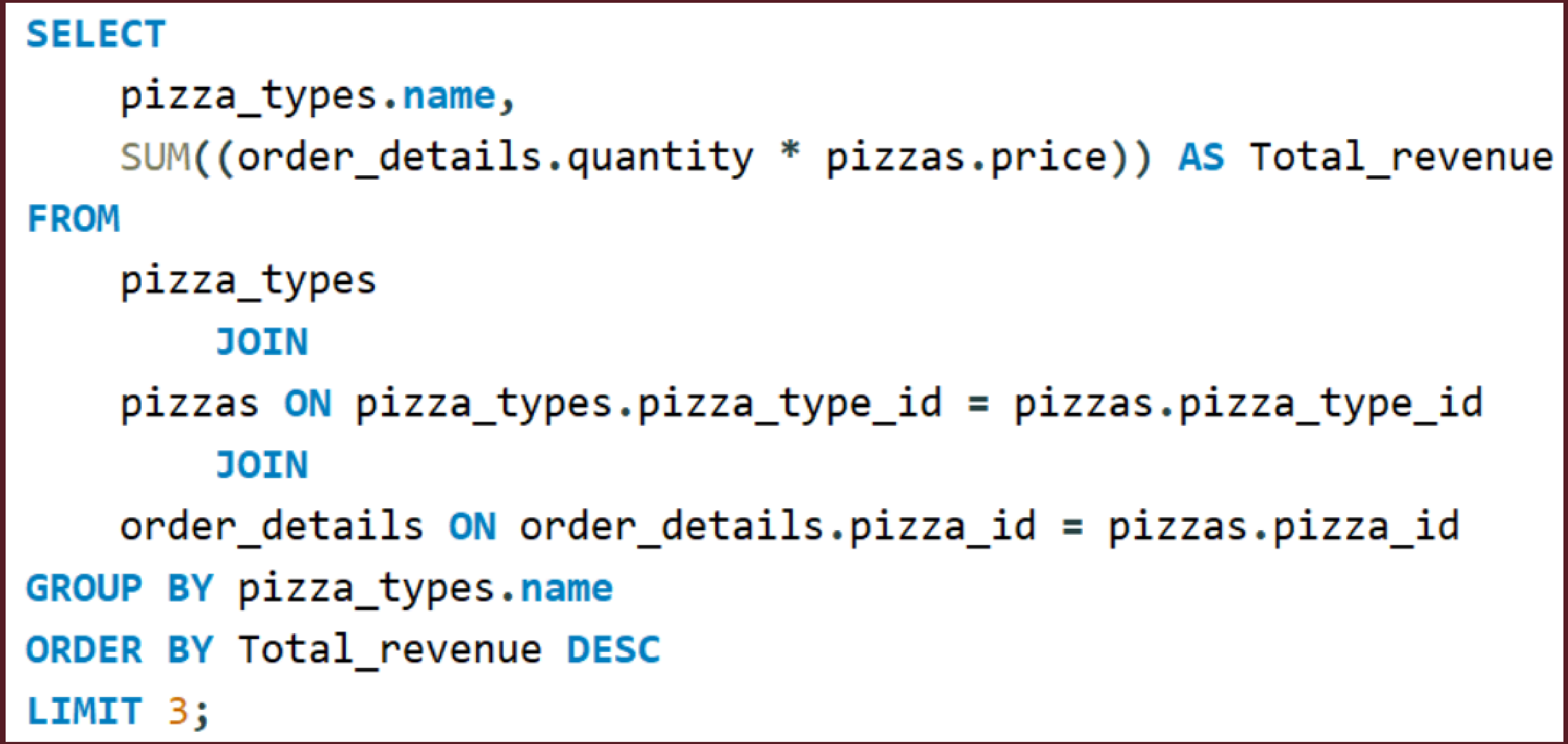
Output



	par_day_order
▶	138

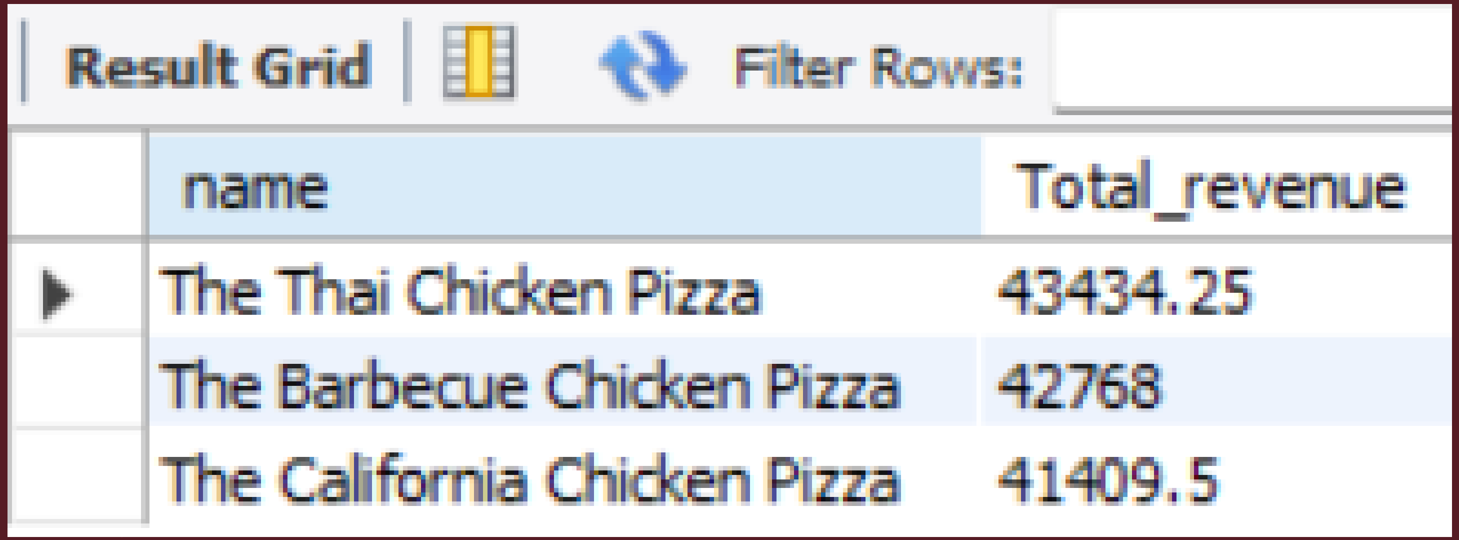
Determine the top 3 most ordered pizza types based on revenue.

Query



```
SELECT
    pizza_types.name,
    SUM((order_details.quantity * pizzas.price)) AS Total_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.name
ORDER BY Total_revenue DESC
LIMIT 3;
```

Output

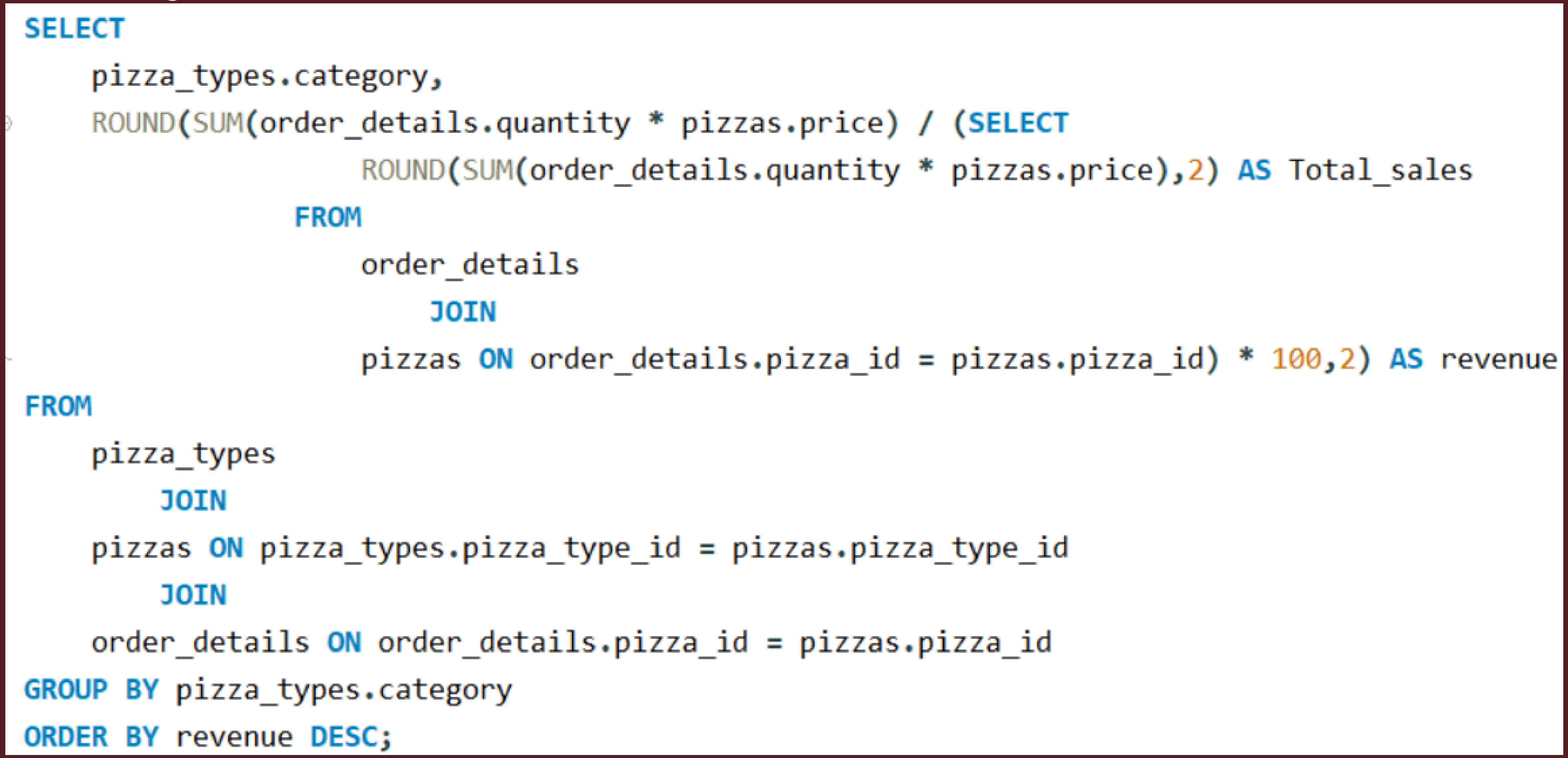


Result Grid | Filter Rows:

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

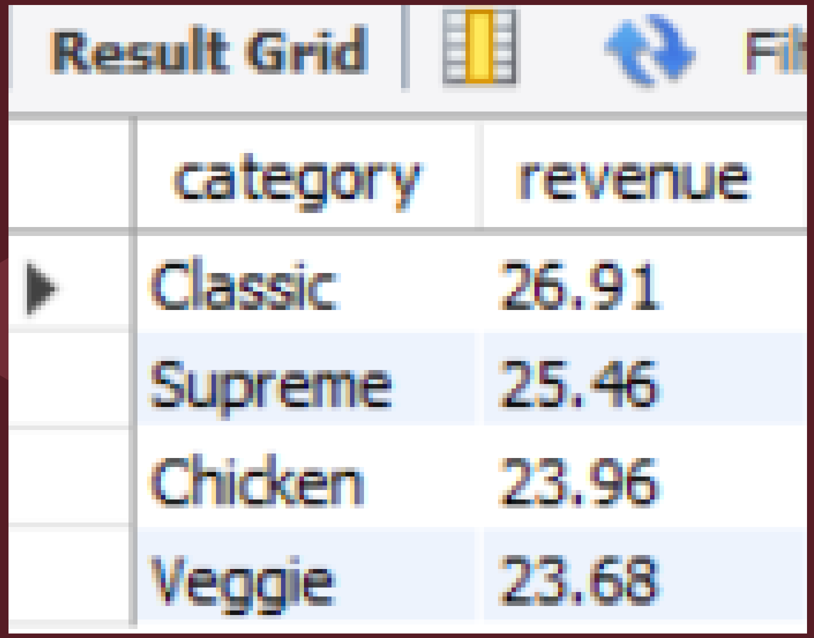
Calculate the percentage contribution of each pizza type to total revenue.

Query



```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                ROUND(SUM(order_details.quantity * pizzas.price),2) AS Total_sales
            FROM
                order_details
                    JOIN
                pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,2) 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
ORDER BY revenue DESC;
```

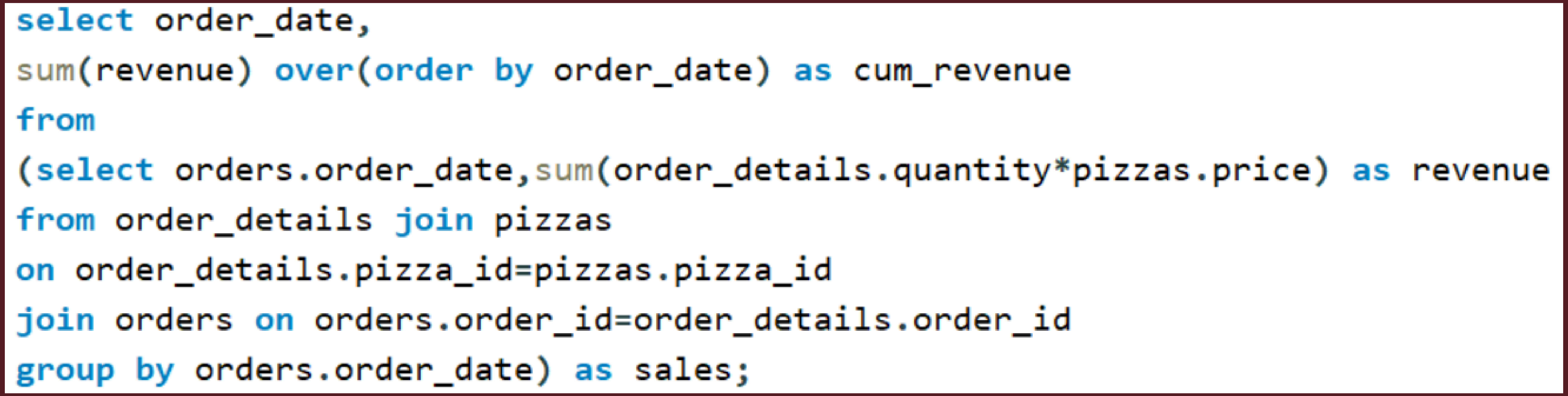
Output



category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

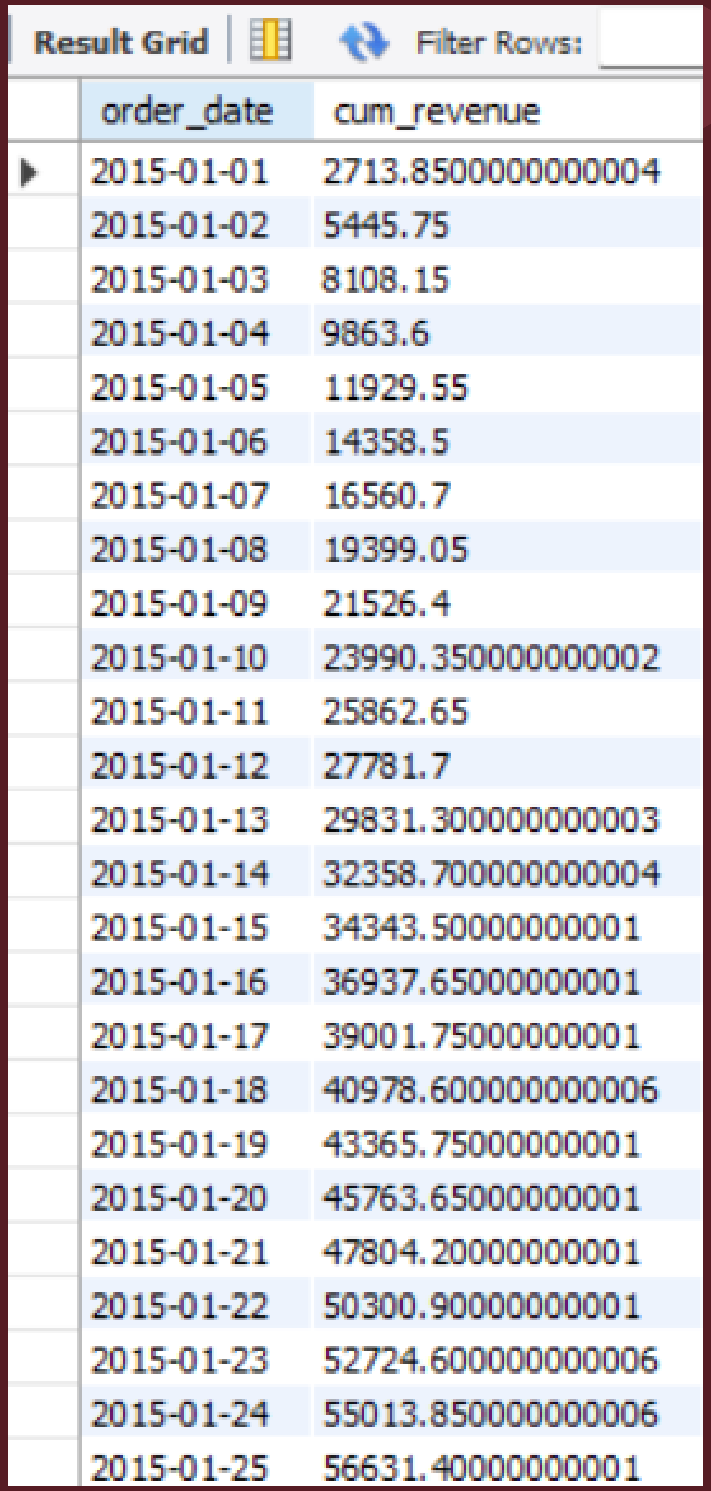
Analyze the cumulative revenue generated over time.

Query



```
select order_date,
sum(revenue) over(order by order_date) 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 on orders.order_id=order_details.order_id
group by orders.order_date) as sales;
```

Output

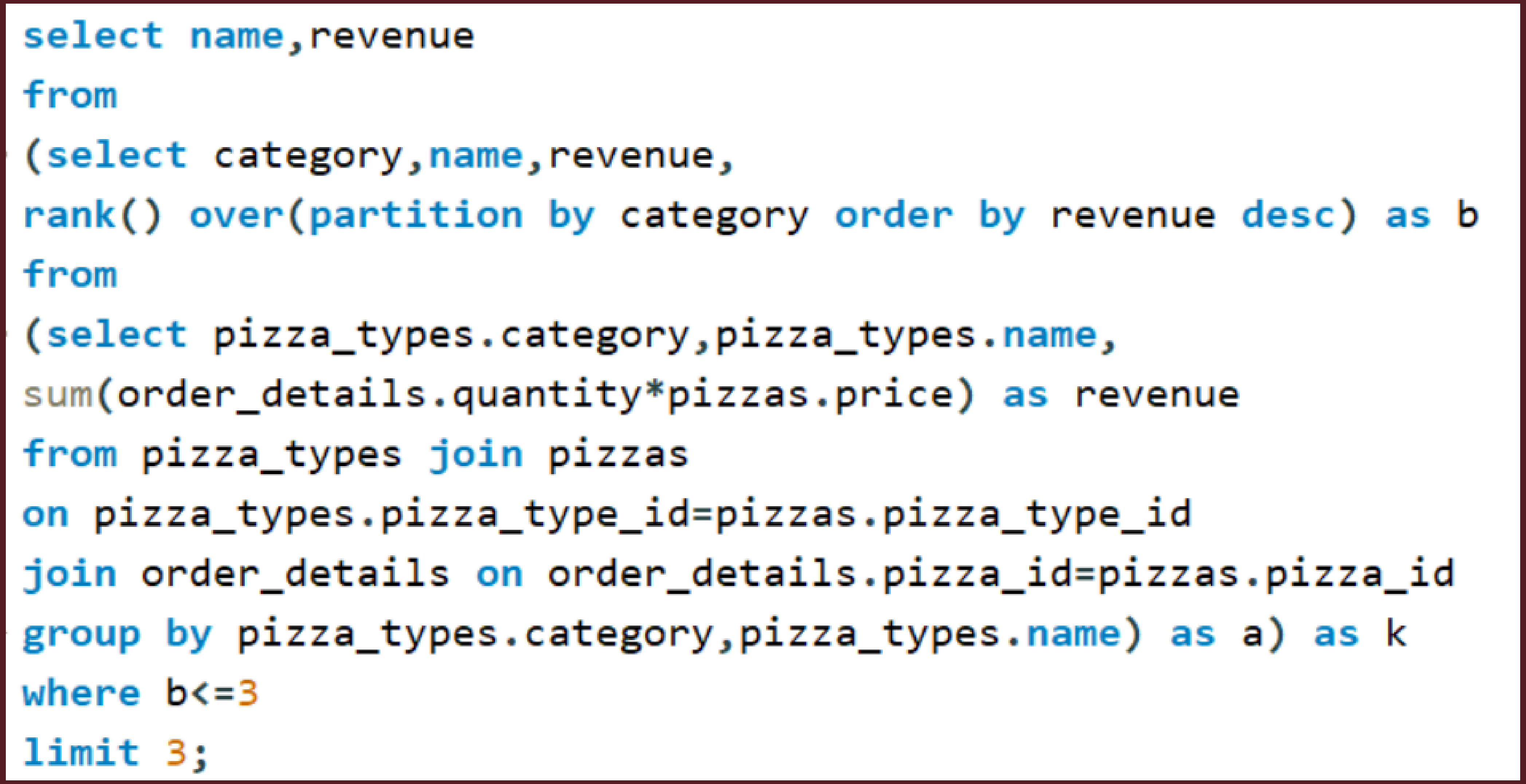


Result Grid | Filter Rows:

order_date	cum_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
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.600000000006
2015-01-24	55013.850000000006
2015-01-25	56631.40000000001

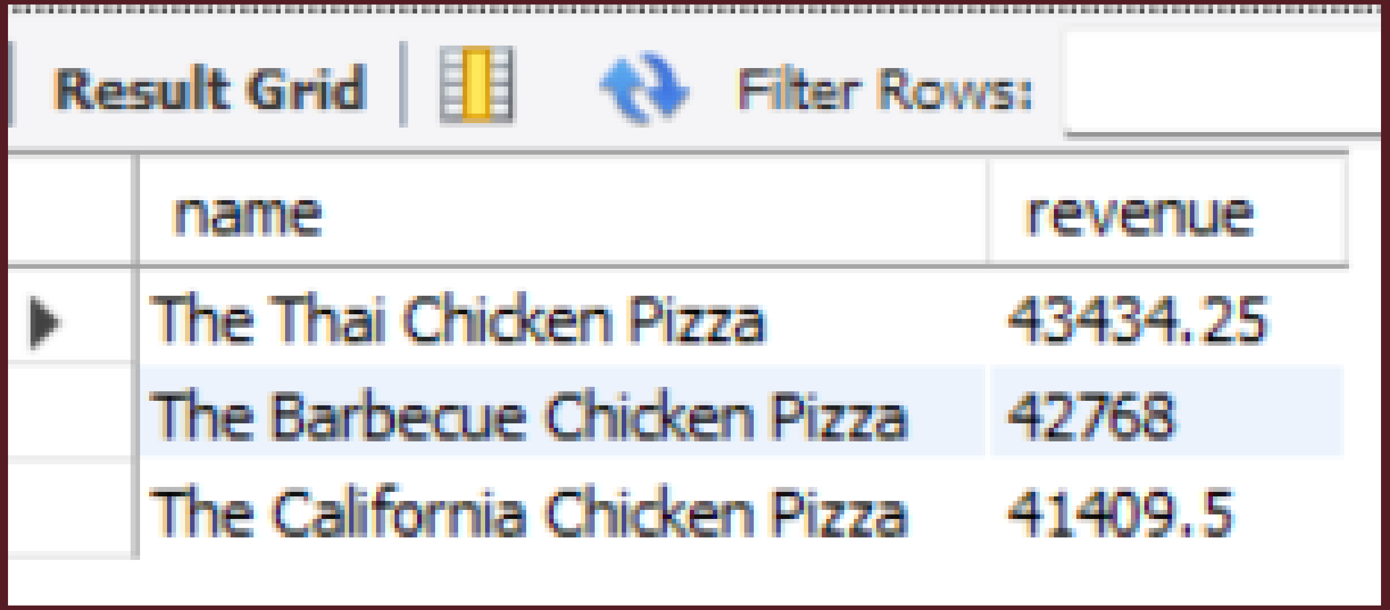
Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Query



```
select name,revenue
from
(select category,name,revenue,
rank() over(partition by category order by revenue desc) as b
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 k
where b<=3
limit 3;
```

Output



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

THANK YOU