SQL PROJECTIONS ON PIZZA SALES

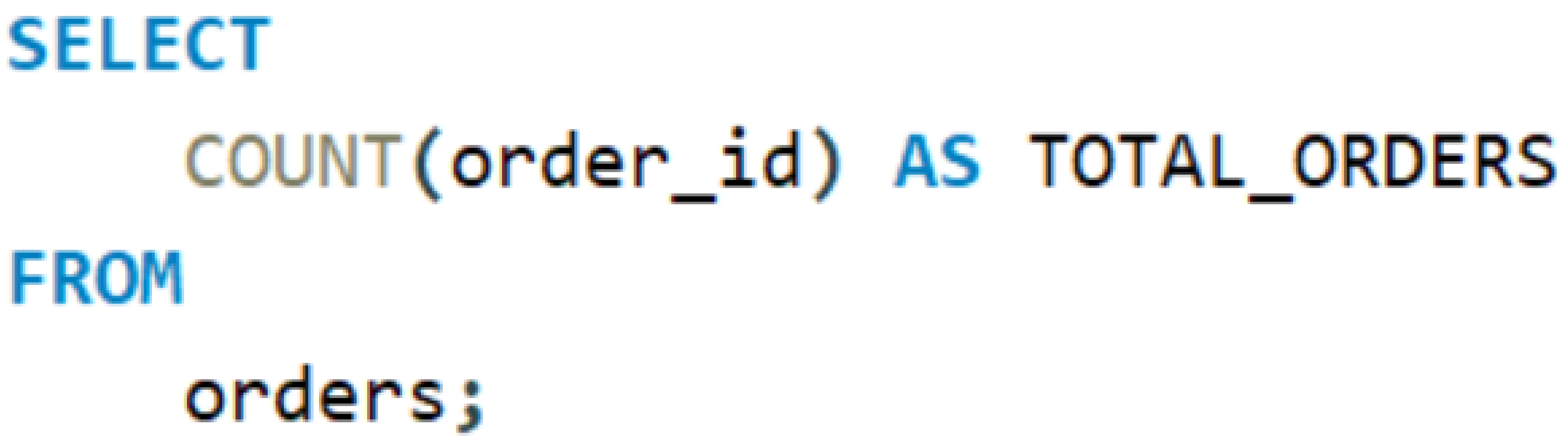
INTRO

Hello, I am Deepak, a student at DBE. In this project, I've leveraged SQL to uncover valuable insights into pizza sales.

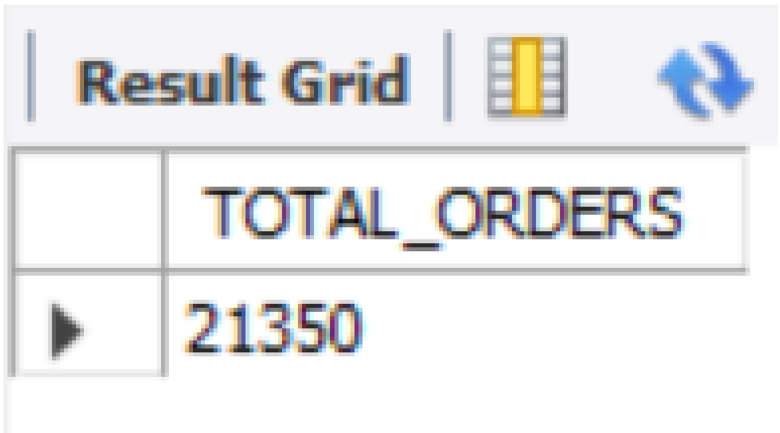
Project Steps:

- Database Creation: Developed a database named "pizzahut."
- Data Import: Imported .csv files into SQL, successfully adding the "pizza" and "pizza_types" tables with fewer than 100 entries each.
- Manual Table Creation: Due to import issues with the "orders" table, manually created a new table with columns for order_id, date, and time.
- Bulk Data Import: Successfully imported over 21k entries into the "orders" table using the import wizard, followed by the creation of the "order_details" table, which was populated with over 48k entries.
- Data Analysis: Performed SQL queries to extract insights from the pizza sales data.

Retrieve the total number of orders placed.

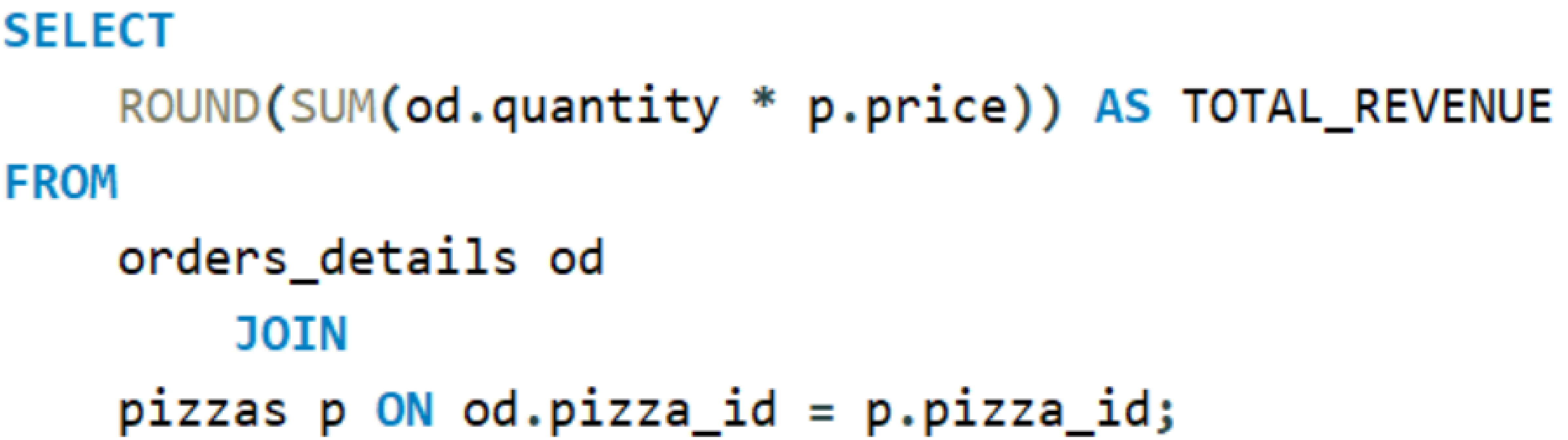


```
SELECT
    COUNT(order_id) AS TOTAL_ORDERS
FROM
    orders;
```

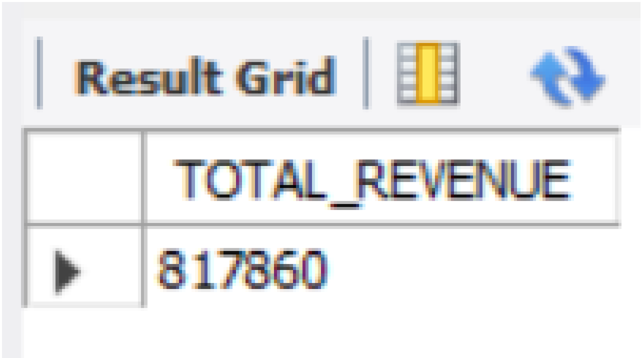


```
   TOTAL_ORDERS
▶  21350
```

Calculate the total revenue generated from pizza sales.



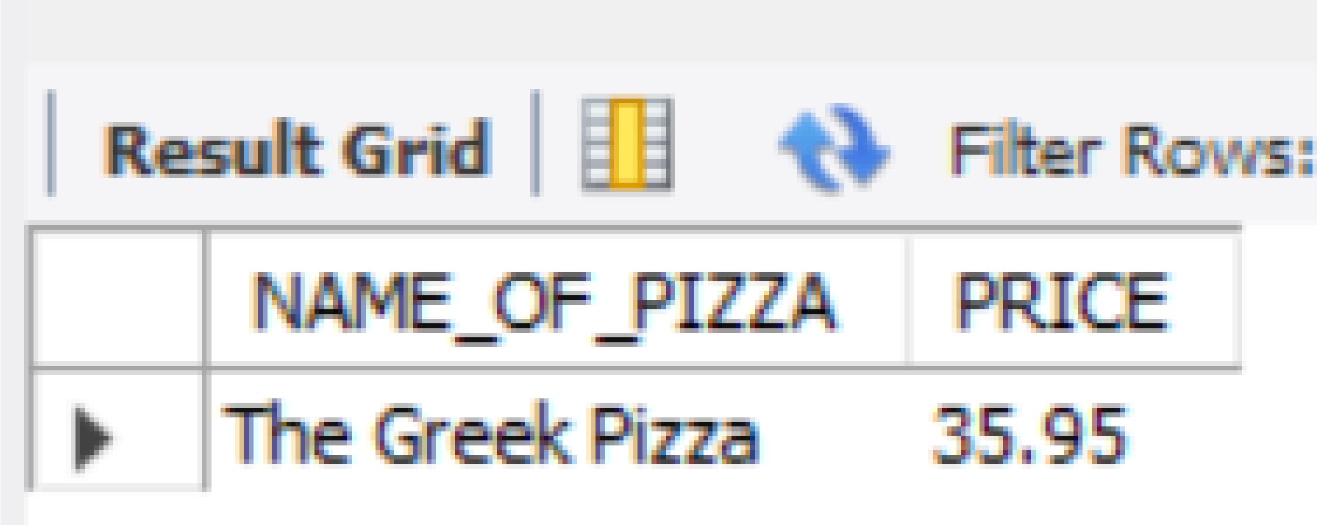
```
SELECT
    ROUND(SUM(od.quantity * p.price)) AS TOTAL_REVENUE
FROM
    orders_details od
        JOIN
    pizzas p ON od.pizza_id = p.pizza_id;
```



TOTAL_REVENUE
817860

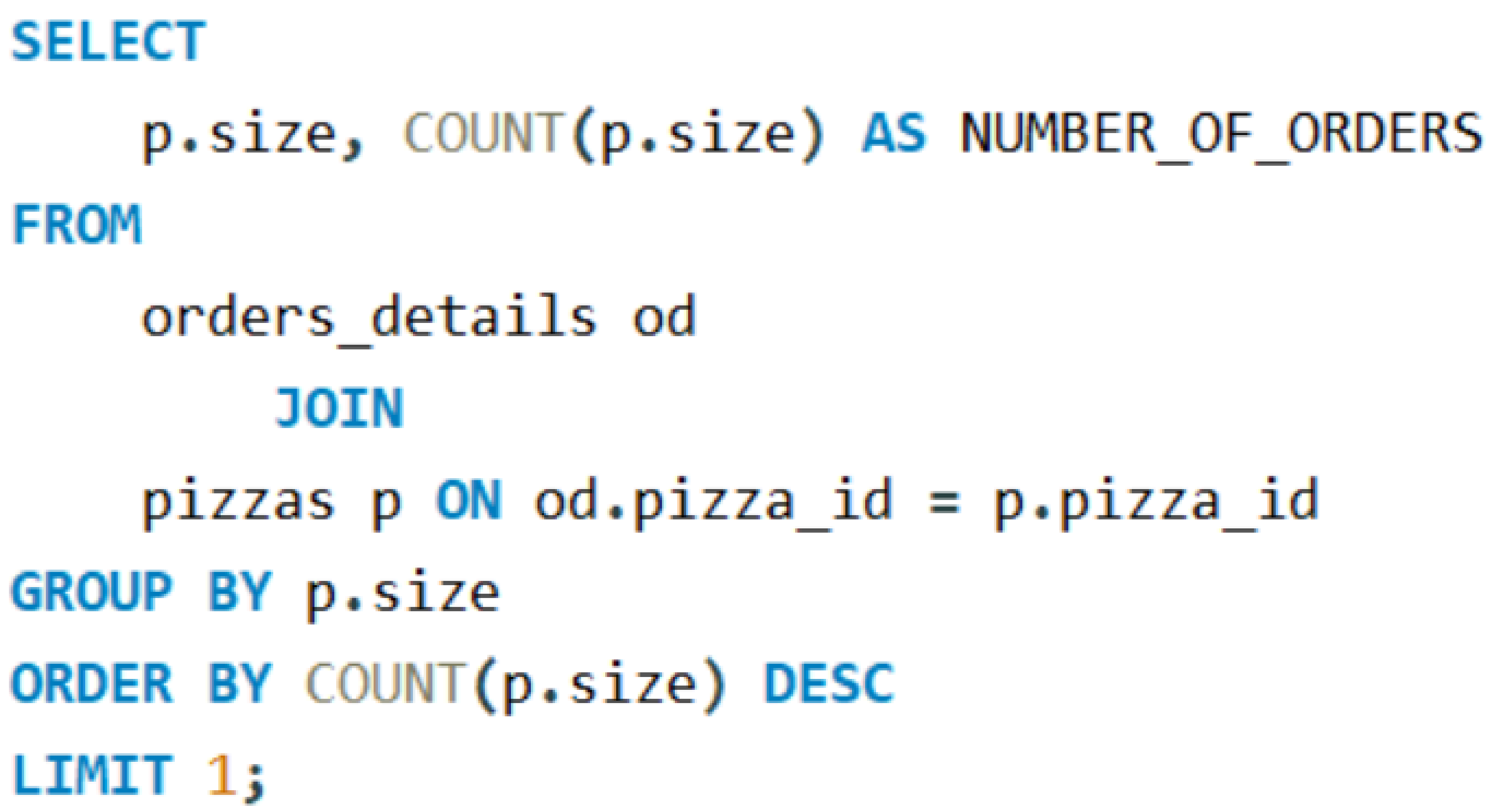
Identify the highest-priced pizza.

```
SELECT
    pt.name AS NAME_OF_PIZZA, p.price AS PRICE
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
ORDER BY p.price DESC
LIMIT 1;
```

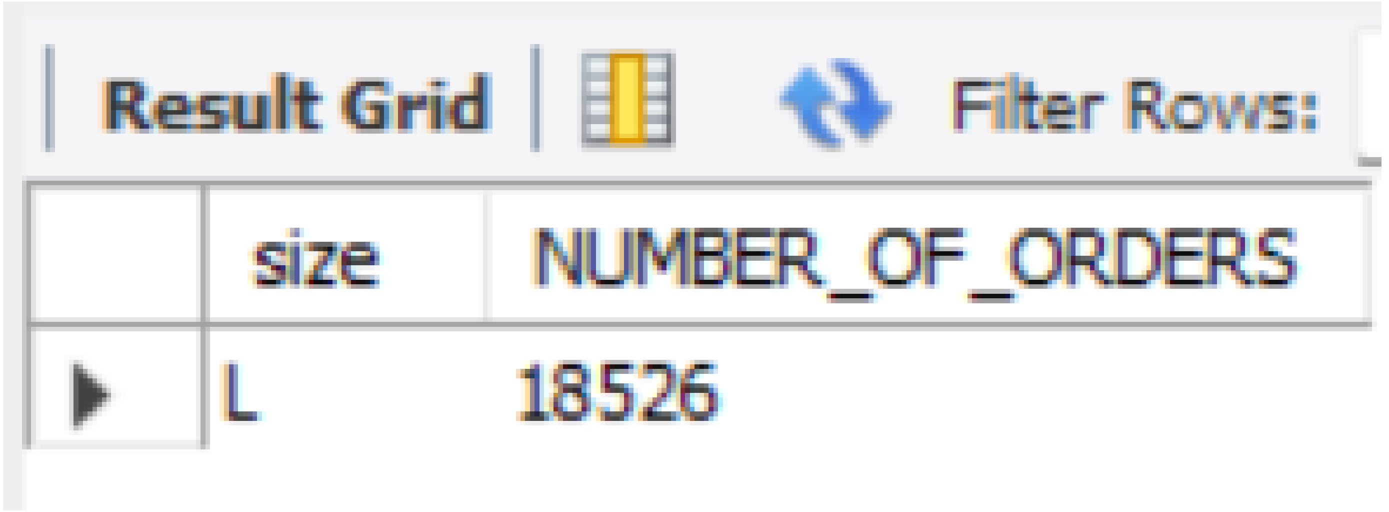


	NAME_OF_PIZZA	PRICE
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

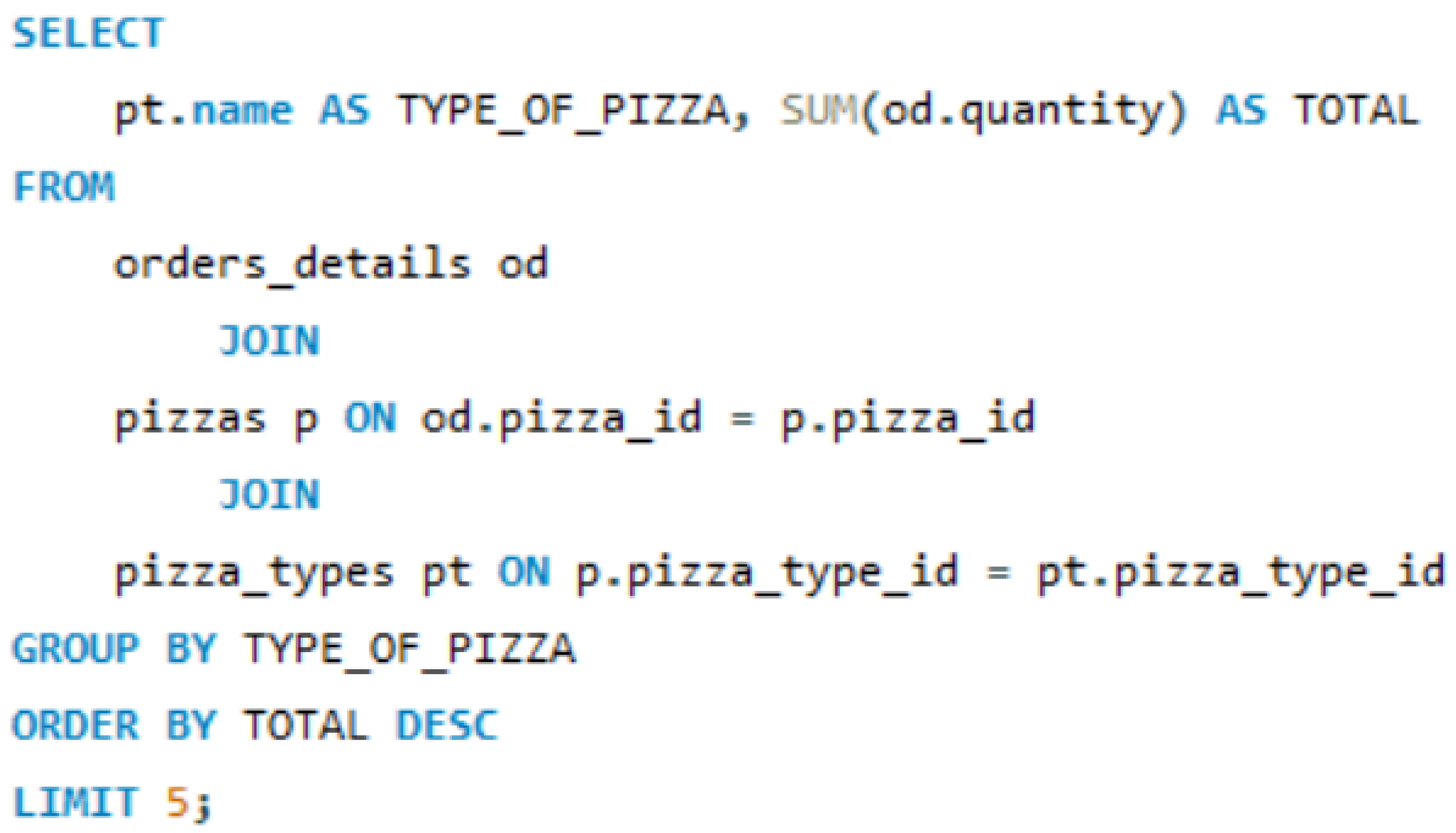


```
SELECT
    p.size, COUNT(p.size) AS NUMBER_OF_ORDERS
FROM
    orders_details od
        JOIN
    pizzas p ON od.pizza_id = p.pizza_id
GROUP BY p.size
ORDER BY COUNT(p.size) DESC
LIMIT 1;
```

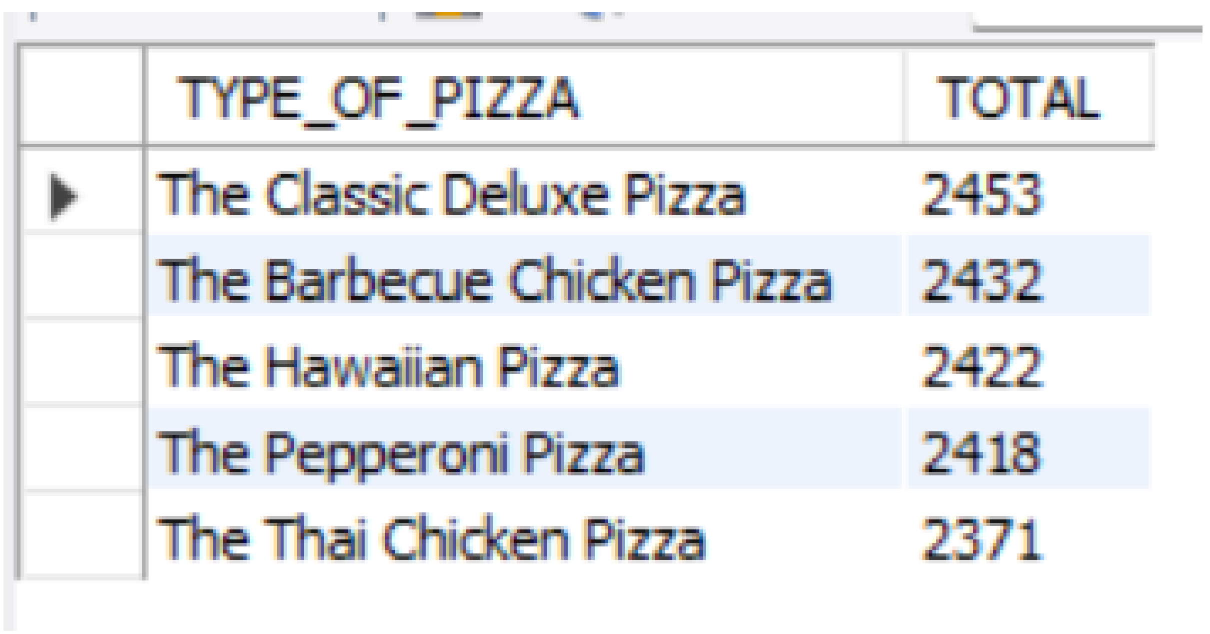


	size	NUMBER_OF_ORDERS
▶	L	18526

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

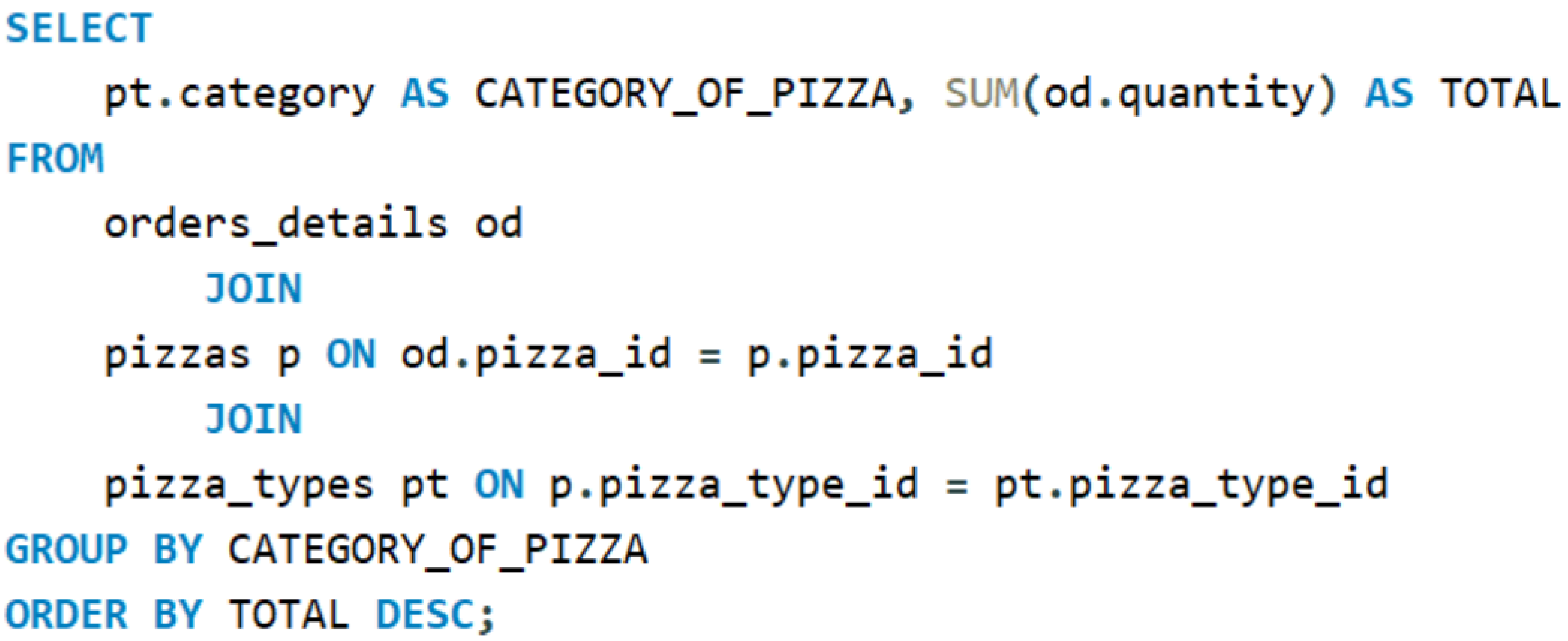


```
SELECT
    pt.name AS TYPE_OF_PIZZA, SUM(od.quantity) AS TOTAL
FROM
    orders_details od
        JOIN
    pizzas p ON od.pizza_id = p.pizza_id
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY TYPE_OF_PIZZA
ORDER BY TOTAL DESC
LIMIT 5;
```

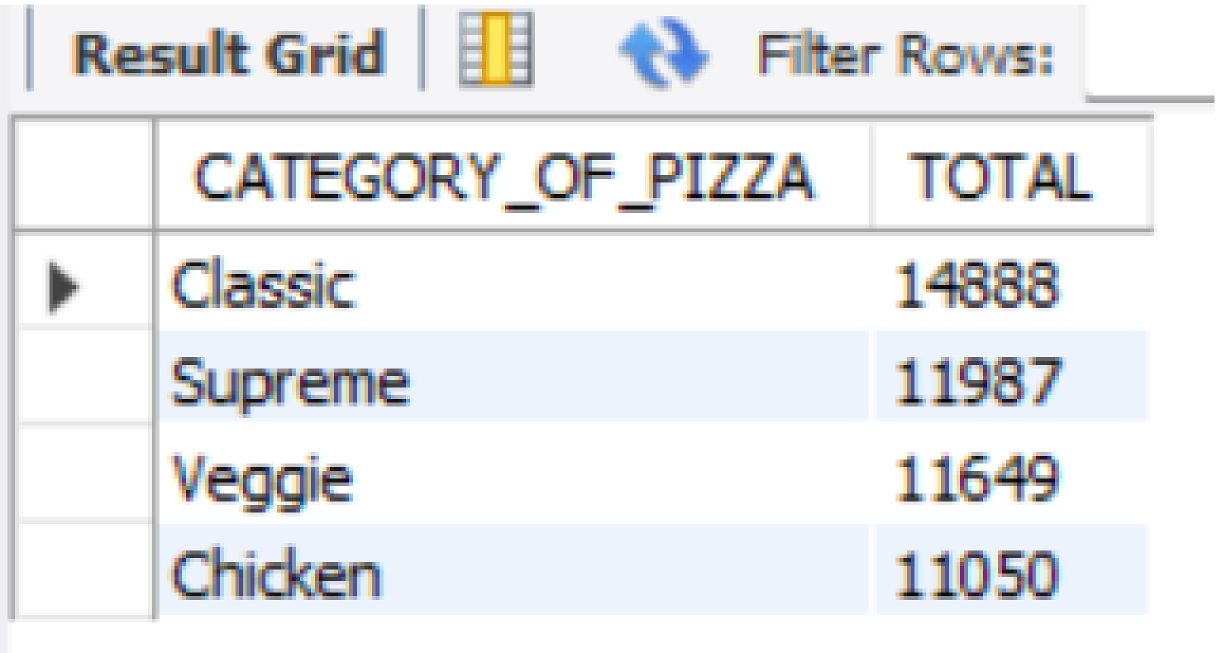


TYPE_OF_PIZZA	TOTAL
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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



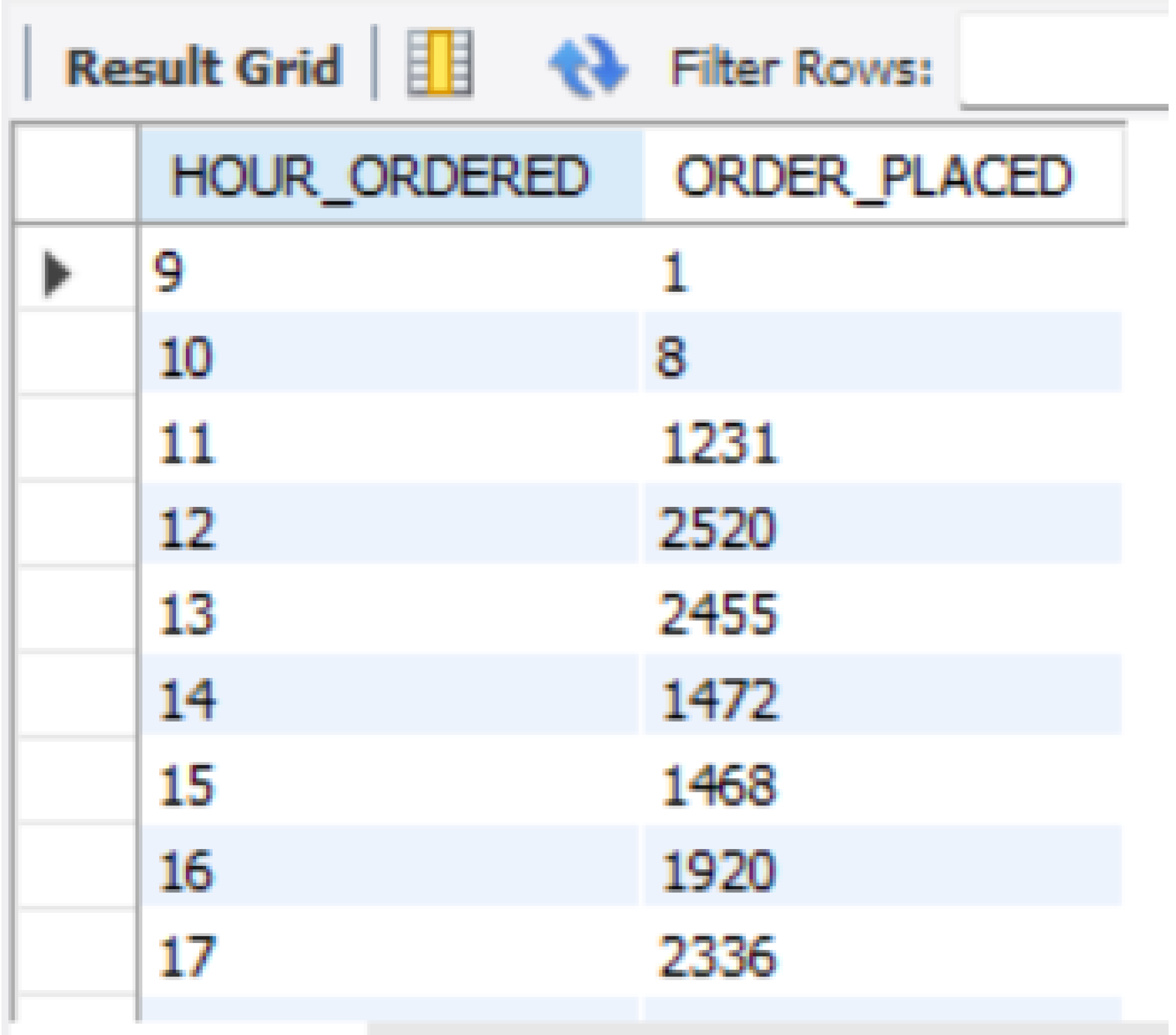
```
SELECT
    pt.category AS CATEGORY_OF_PIZZA, SUM(od.quantity) AS TOTAL
FROM
    orders_details od
        JOIN
    pizzas p ON od.pizza_id = p.pizza_id
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY CATEGORY_OF_PIZZA
ORDER BY TOTAL DESC;
```



CATEGORY_OF_PIZZA	TOTAL
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

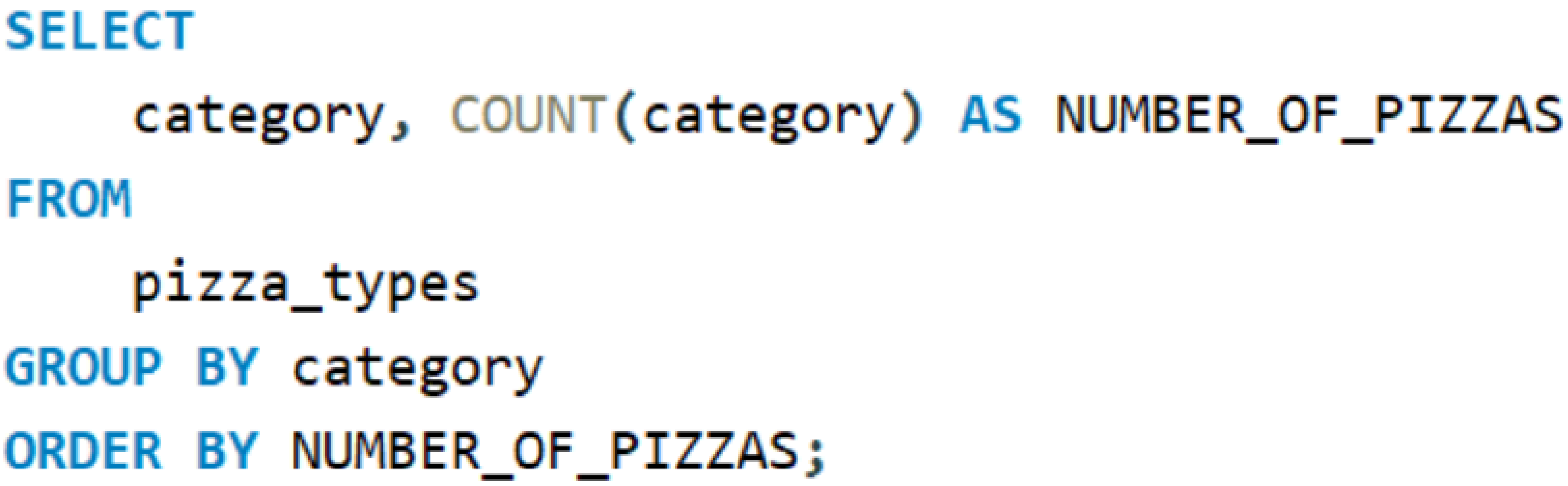
Determine the distribution of orders by hour of the day.

```
SELECT
    HOUR(o.order_time) AS HOUR_ORDERED,
    COUNT(o.order_id) AS ORDER_PLACED
FROM
    orders o
GROUP BY HOUR_ORDERED
ORDER BY HOUR_ORDERED;
```

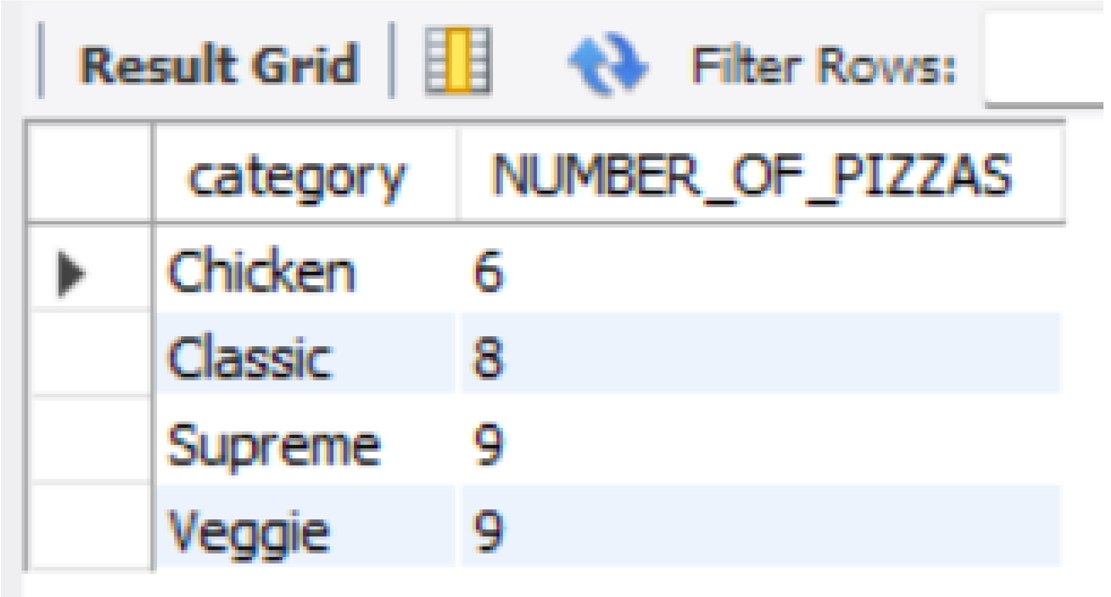


HOUR_ORDERED	ORDER_PLACED
9	1
10	8
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336

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

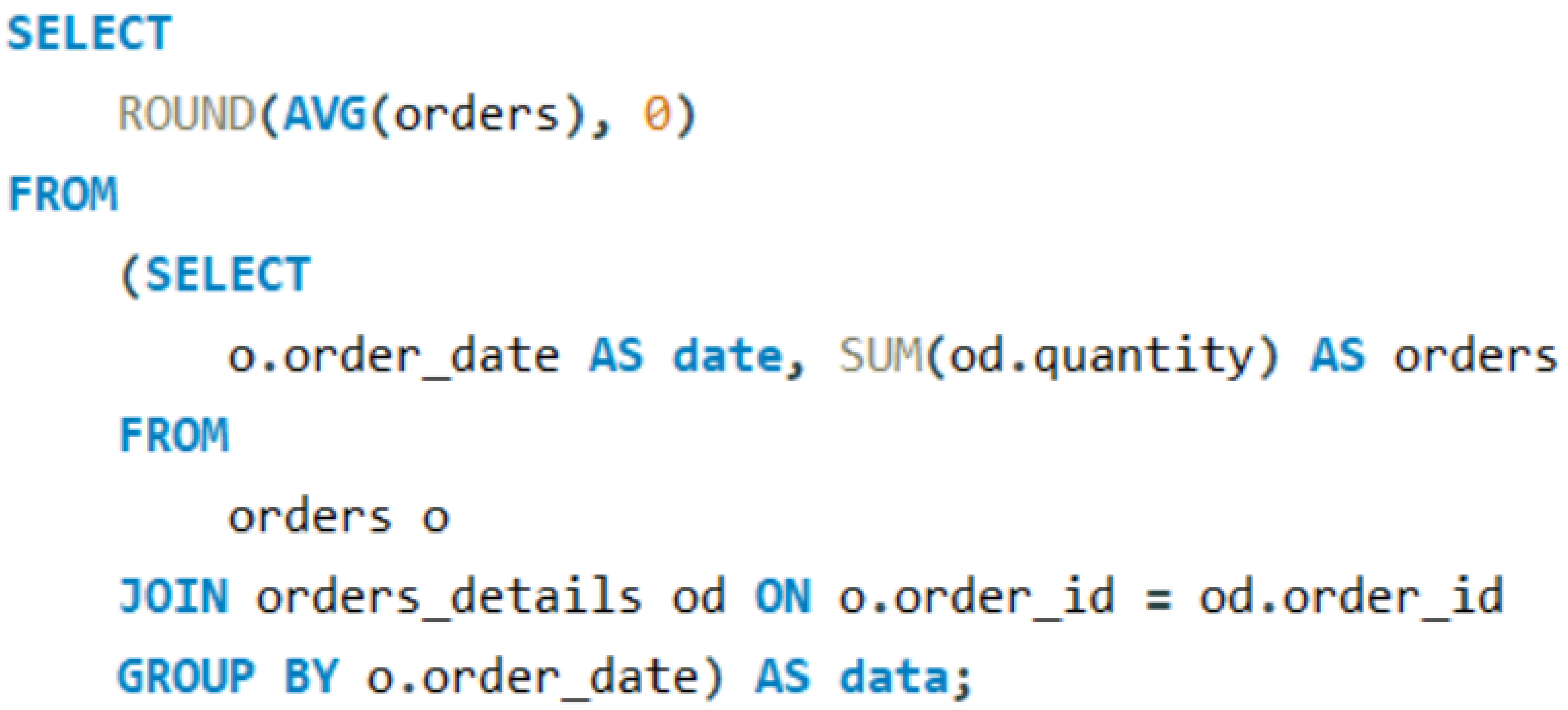


```
SELECT
    category, COUNT(category) AS NUMBER_OF_PIZZAS
FROM
    pizza_types
GROUP BY category
ORDER BY NUMBER_OF_PIZZAS;
```

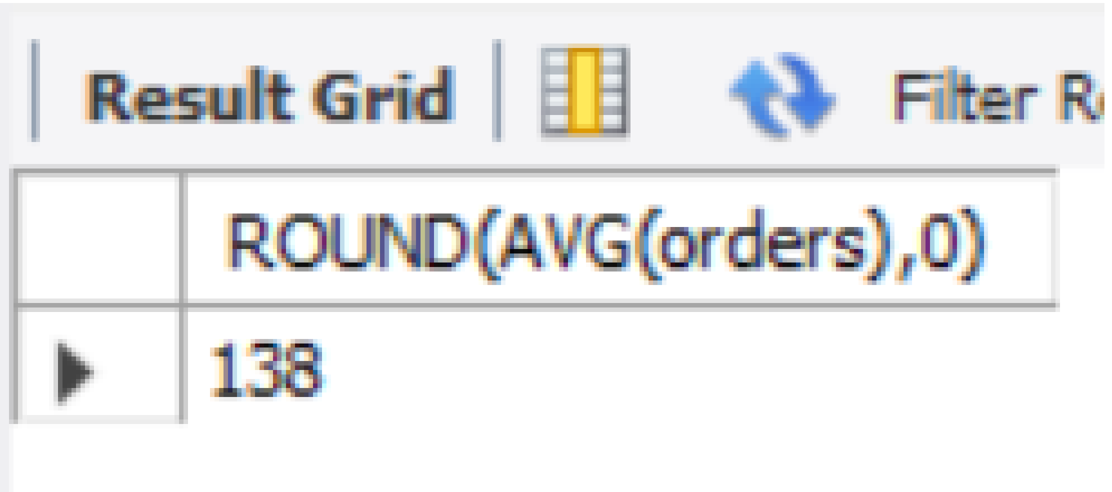


category	NUMBER_OF_PIZZAS
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

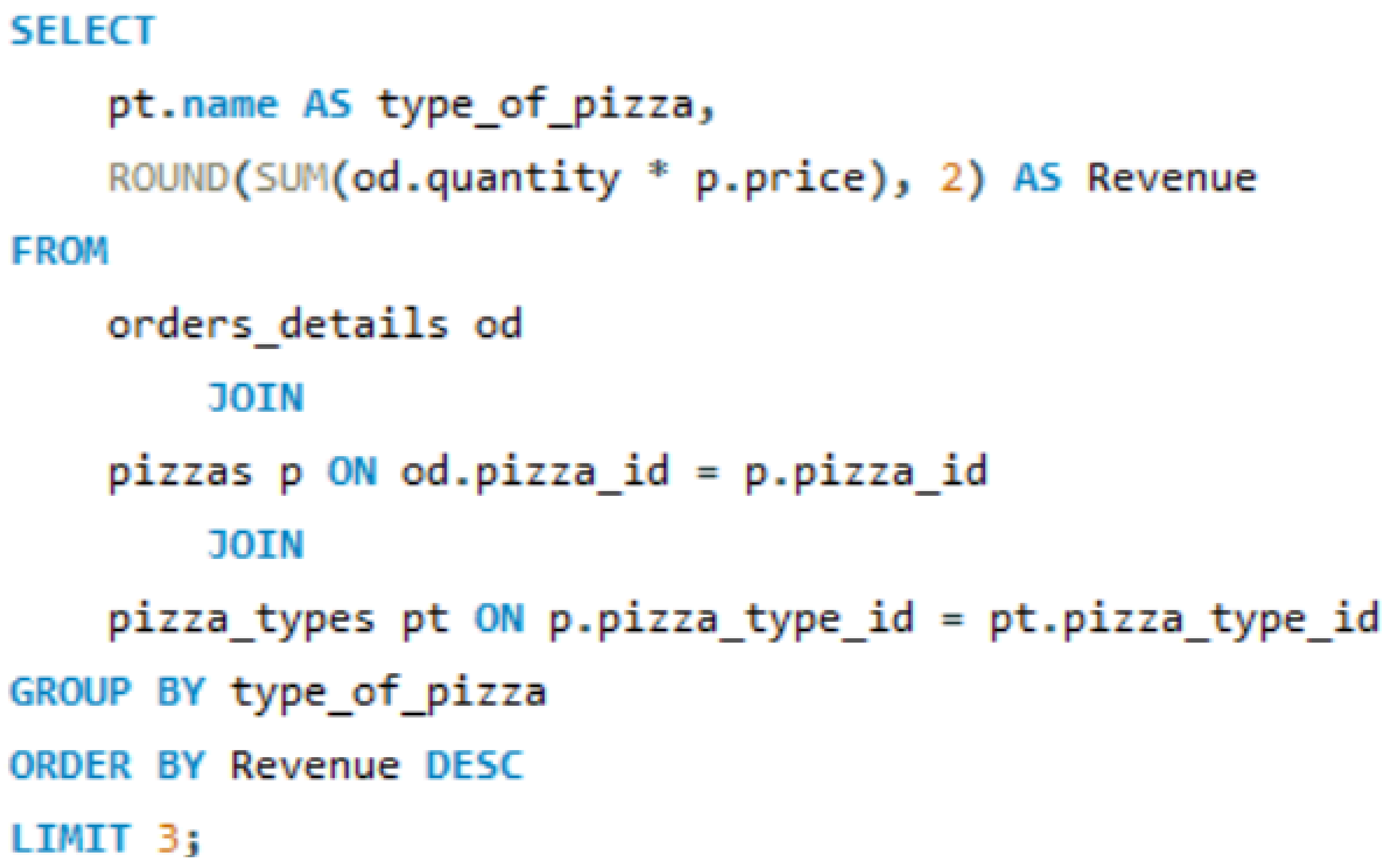


```sql
SELECT
    ROUND(AVG(orders), 0)
FROM
    (SELECT
        o.order_date AS date, SUM(od.quantity) AS orders
    FROM
        orders o
    JOIN orders_details od ON o.order_id = od.order_id
    GROUP BY o.order_date) AS data;
```

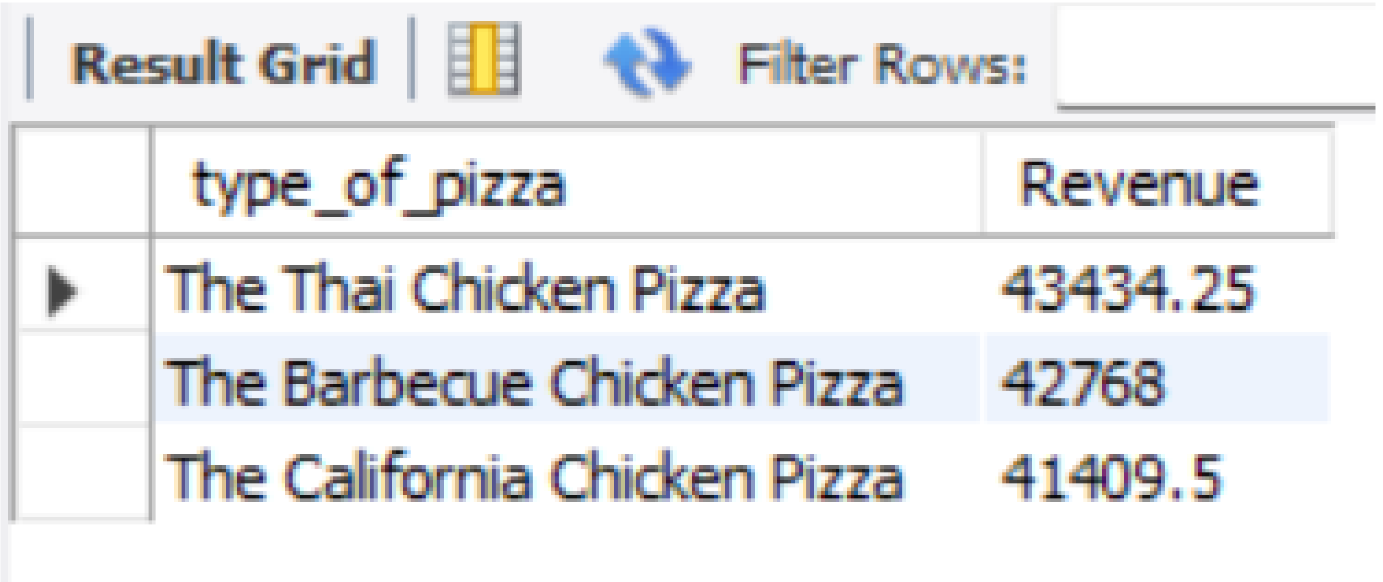


	ROUND(AVG(orders),0)
▶	138

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

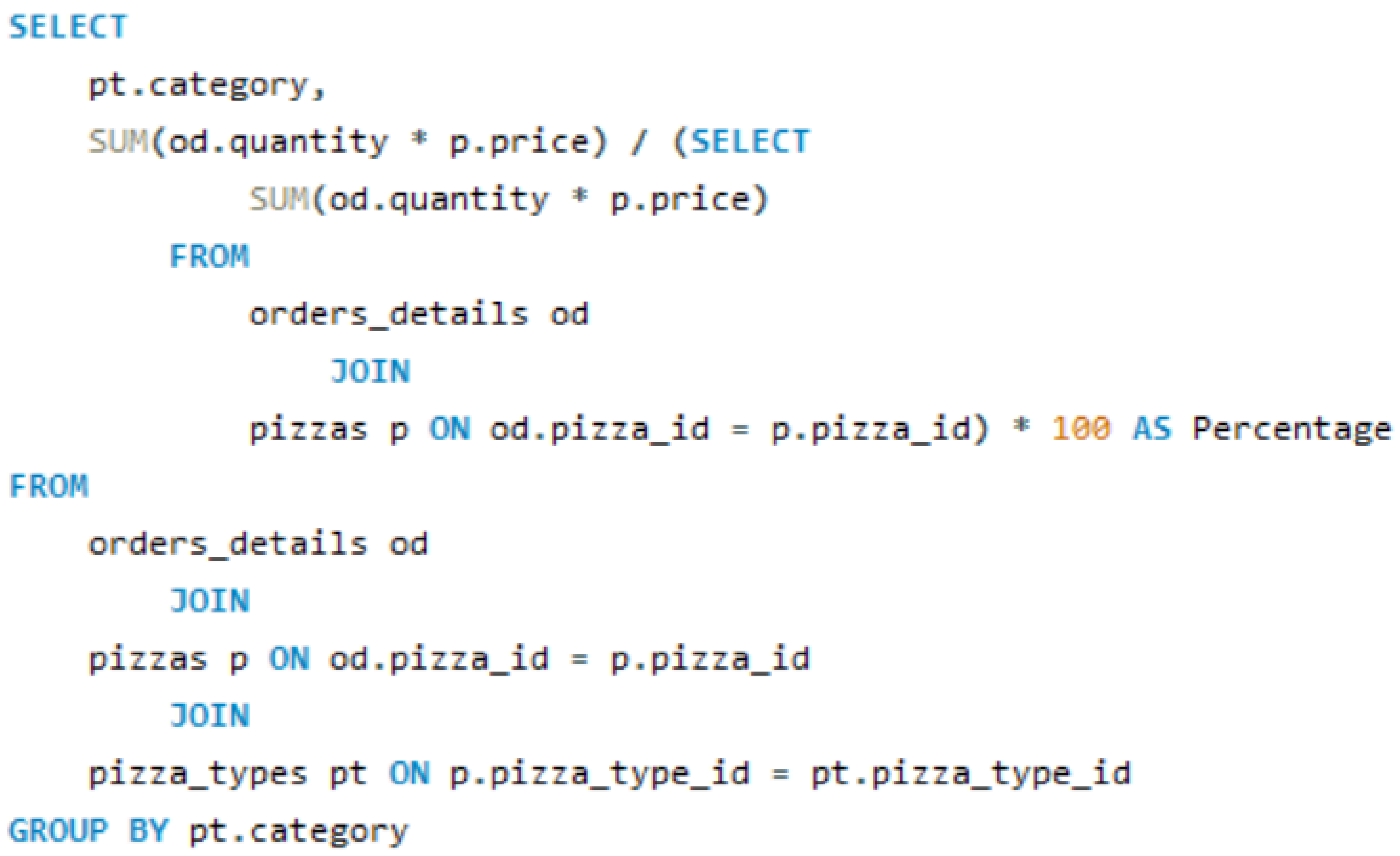


```
SELECT
    pt.name AS type_of_pizza,
    ROUND(SUM(od.quantity * p.price), 2) AS Revenue
FROM
    orders_details od
        JOIN
    pizzas p ON od.pizza_id = p.pizza_id
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY type_of_pizza
ORDER BY Revenue DESC
LIMIT 3;
```

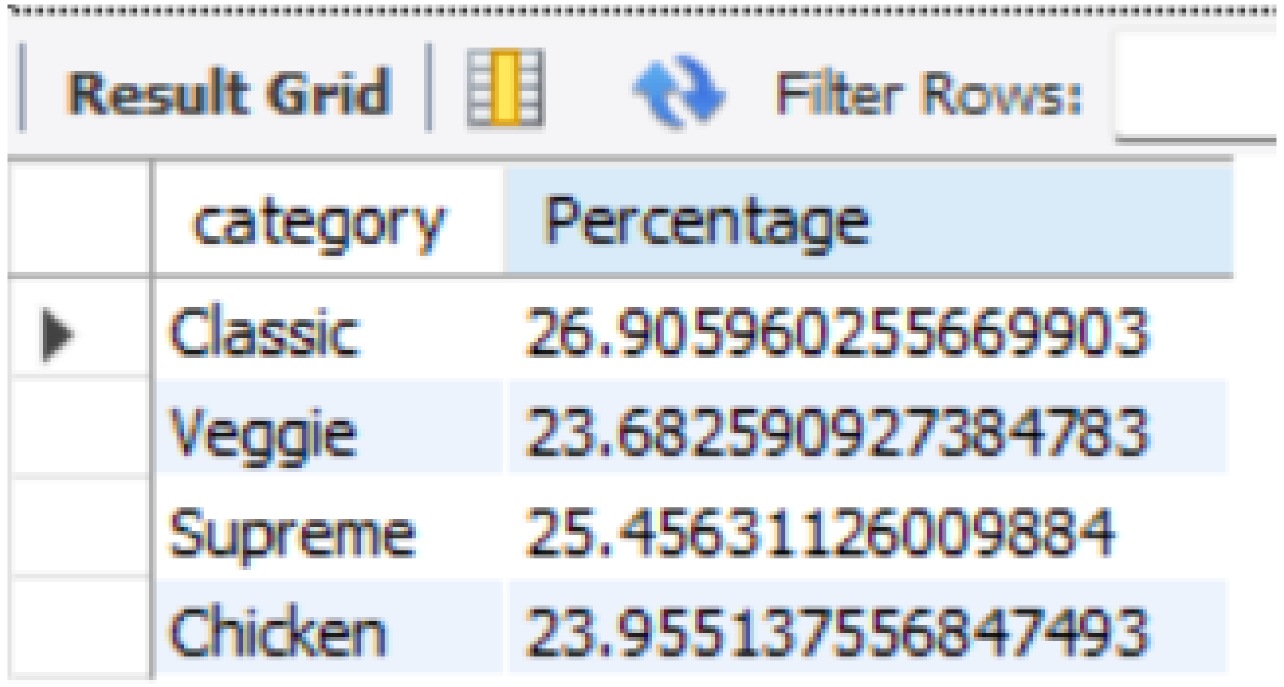


	type_of_pizza	Revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

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



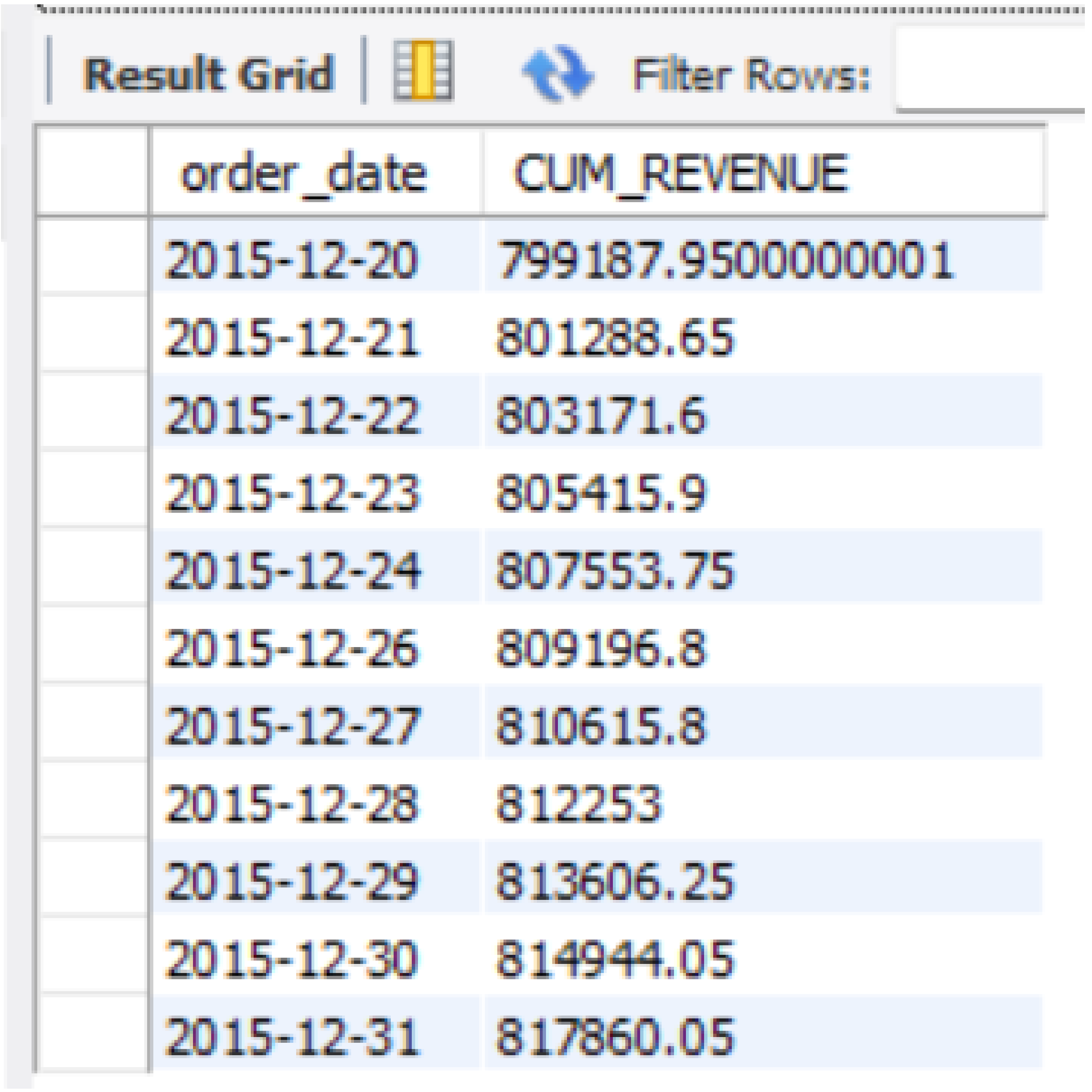
```
SELECT
    pt.category,
    SUM(od.quantity * p.price) / (SELECT
            SUM(od.quantity * p.price)
        FROM
            orders_details od
                JOIN
            pizzas p ON od.pizza_id = p.pizza_id) * 100 AS Percentage
FROM
    orders_details od
        JOIN
    pizzas p ON od.pizza_id = p.pizza_id
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category
```



category	Percentage
Classic	26.905960255669903
Veggie	23.682590927384783
Supreme	25.45631126009884
Chicken	23.955137556847493

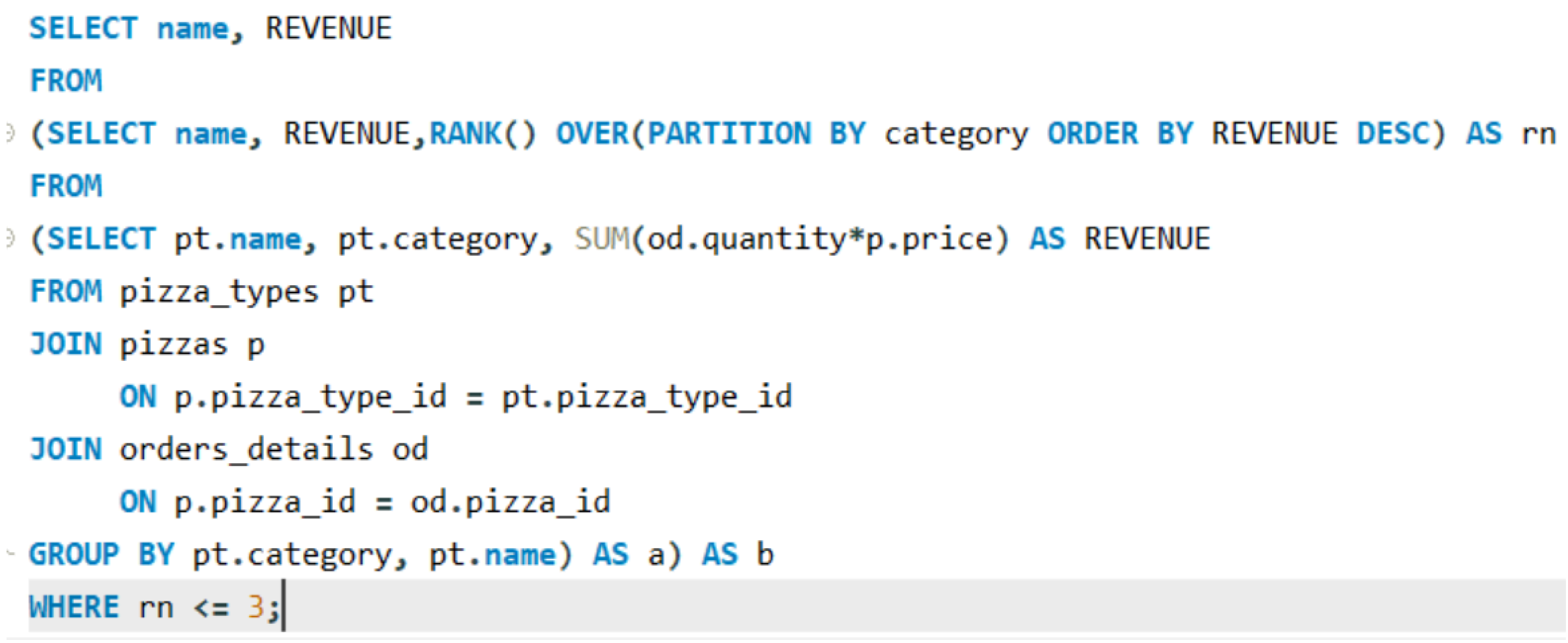
Analyze the cumulative revenue generated over time.

```
SELECT order_date, SUM(REVENUE) OVER(ORDER BY order_date) AS CUM_REVENUE
FROM
(SELECT o.order_date, SUM(od.quantity*p.price) AS REVENUE
FROM orders o
JOIN orders_details od
     ON o.order_id = od.order_id
JOIN pizzas p
     ON od.pizza_id = p.pizza_id
GROUP BY o.order_date) AS SALES
```

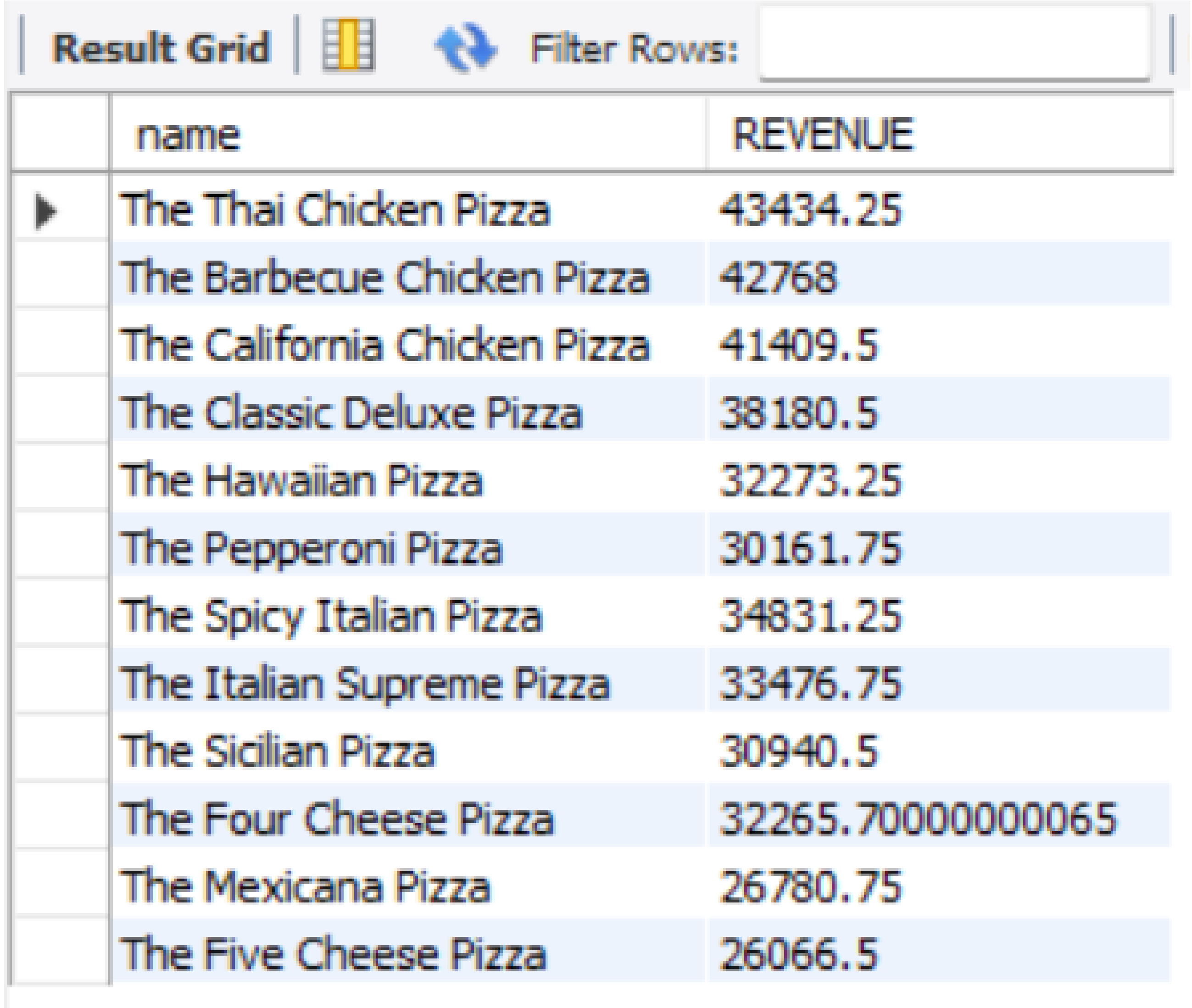


order_date	CUM_REVENUE
2015-12-20	799187.9500000001
2015-12-21	801288.65
2015-12-22	803171.6
2015-12-23	805415.9
2015-12-24	807553.75
2015-12-26	809196.8
2015-12-27	810615.8
2015-12-28	812253
2015-12-29	813606.25
2015-12-30	814944.05
2015-12-31	817860.05

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



```
SELECT name, REVENUE
FROM
(SELECT name, REVENUE,RANK() OVER(PARTITION BY category ORDER BY REVENUE DESC) AS rn
FROM
(SELECT pt.name, pt.category, SUM(od.quantity*p.price) AS REVENUE
FROM pizza_types pt
JOIN pizzas p
    ON p.pizza_type_id = pt.pizza_type_id
JOIN orders_details od
    ON p.pizza_id = od.pizza_id
GROUP BY pt.category, pt.name) AS a) AS b
WHERE rn <= 3;
```



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	32265.7000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

PIZZA PARTY!