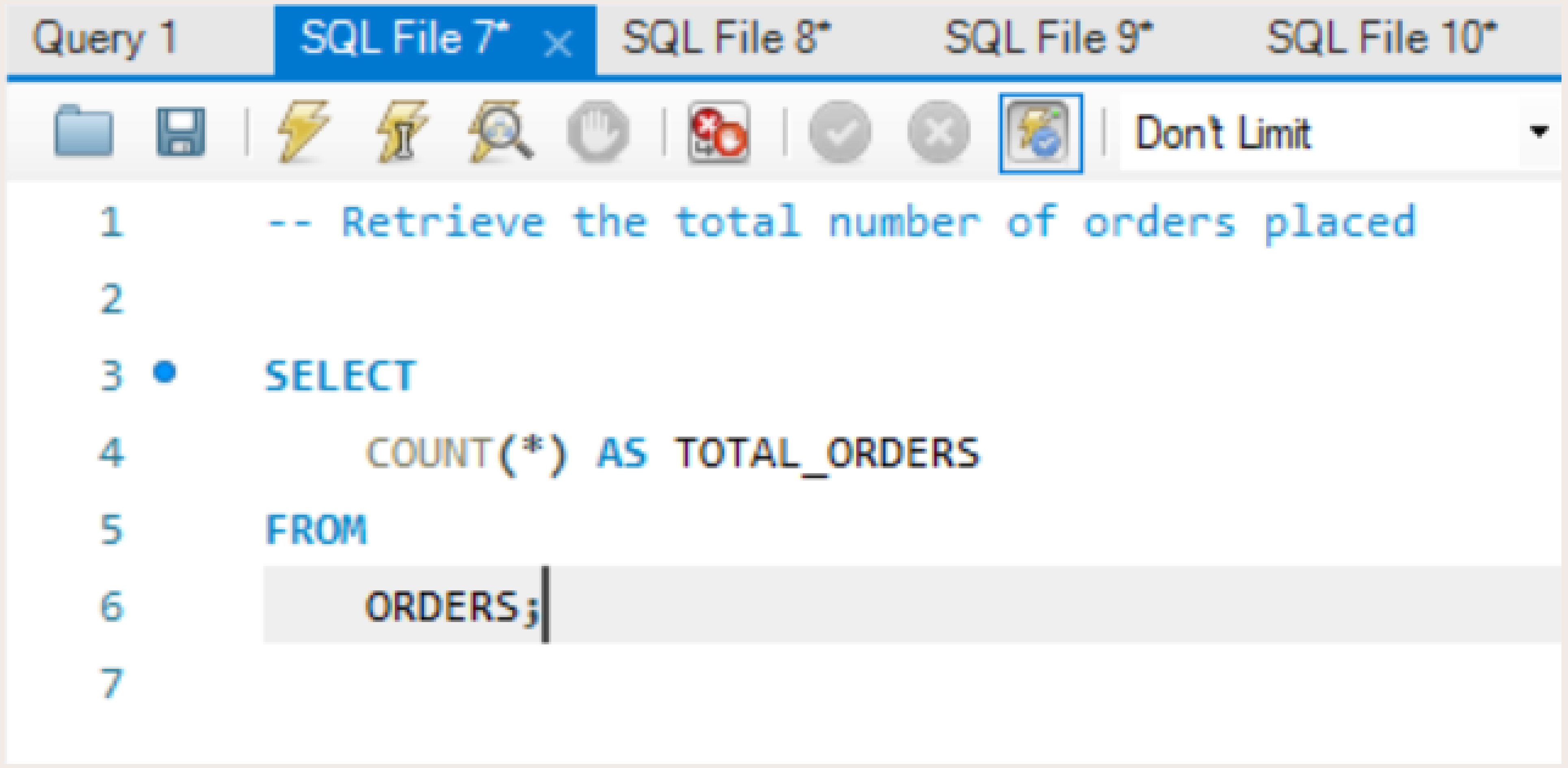
Pizza Blue

A sql Project

Pizza Blue Sales Analysis Project

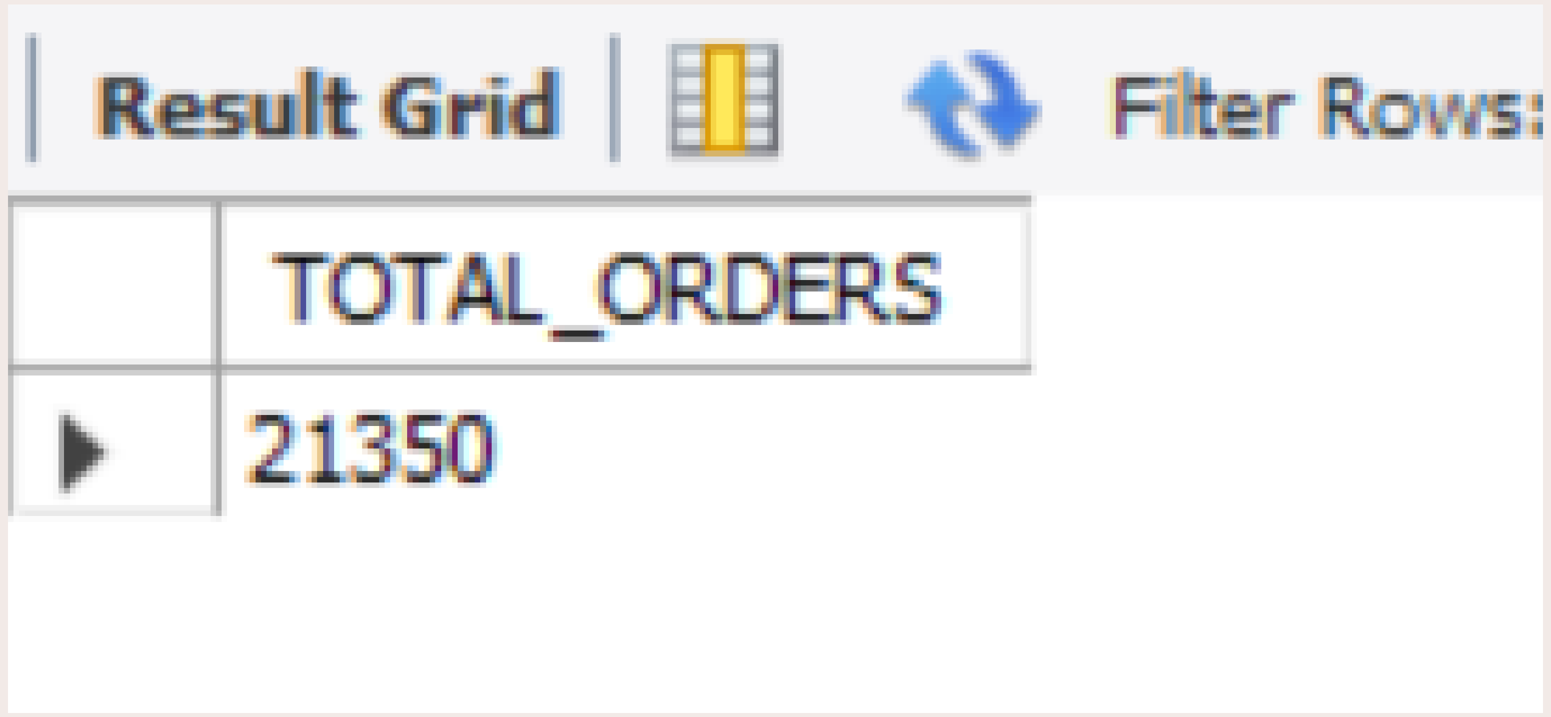
Welcome to the Pizza Blue Sales Analysis Project! Using SQL, we uncover key insights and trends in pizza sales. Our analysis covers sales performance, revenue, and customer preferences, aiding strategic decision-making and optimization. Join us to explore the data and reveal the story behind Pizza Blue's success.

Retrieve the total number of orders placed



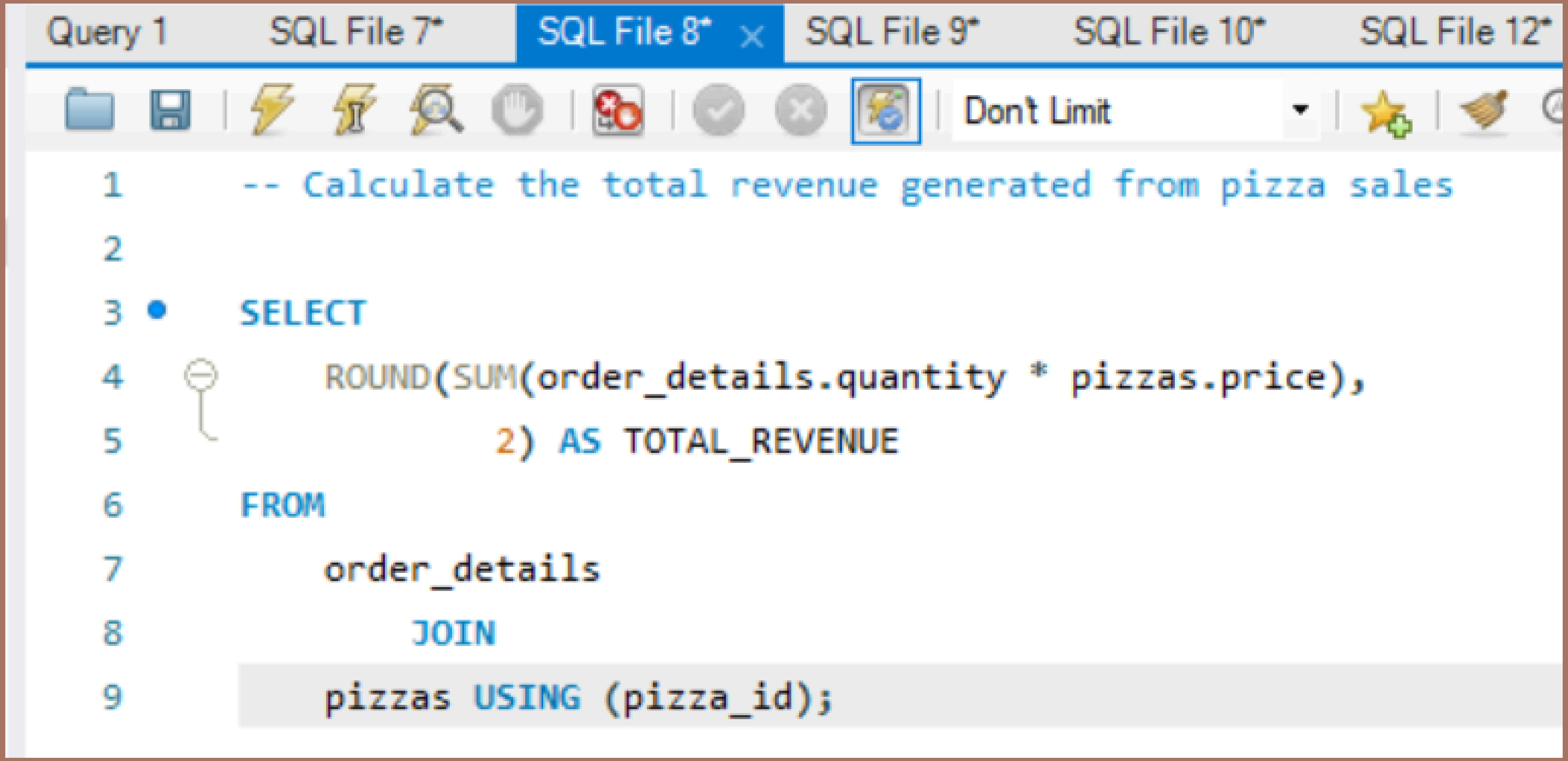
```
-- Retrieve the total number of orders placed

SELECT
    COUNT(*) AS TOTAL_ORDERS
FROM
    ORDERS;
```



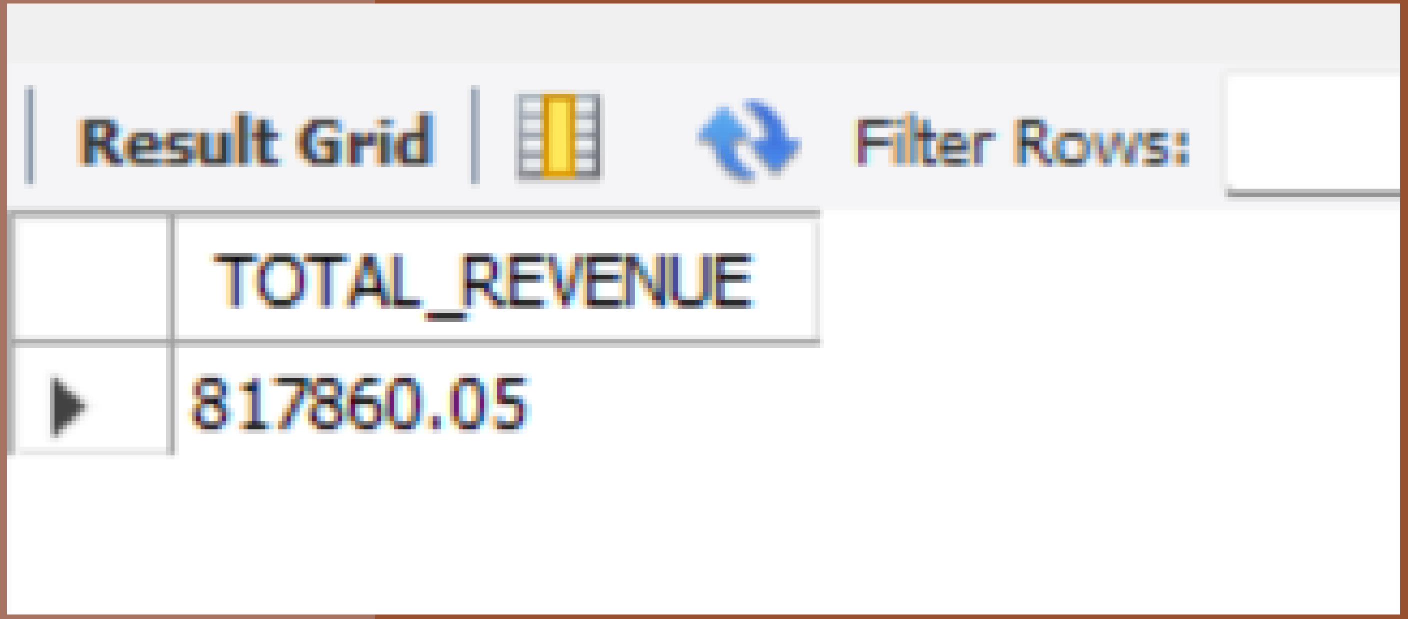
```
TOTAL_ORDERS
21350
```

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES



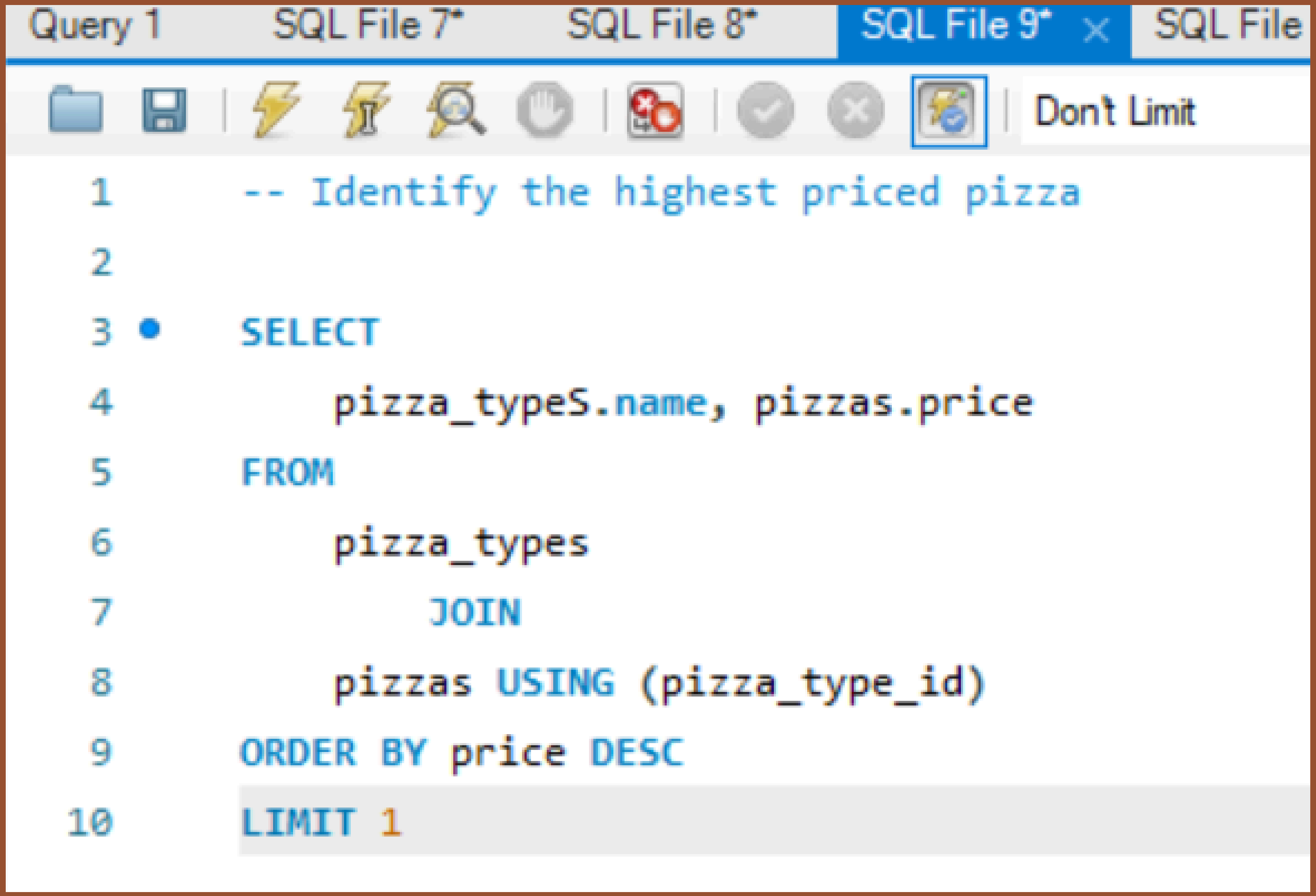
```sql
-- Calculate the total revenue generated from pizza sales

SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS TOTAL_REVENUE
FROM
    order_details
        JOIN
    pizzas USING (pizza_id);
```



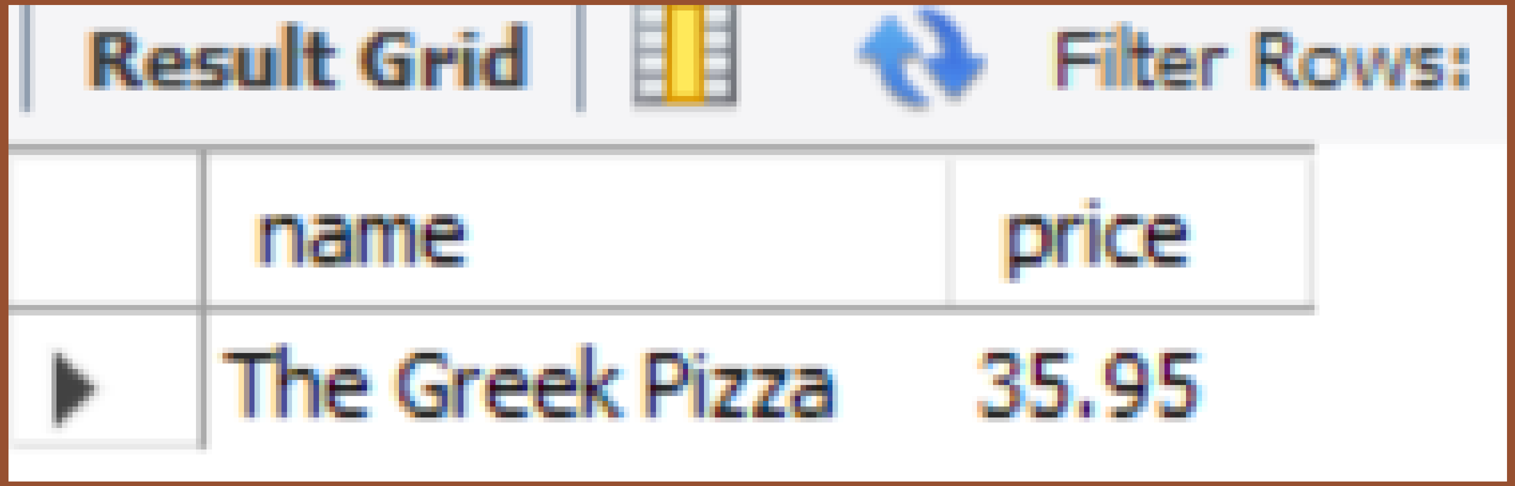
```
TOTAL_REVENUE
817860.05
```

IDENTIFY THE HIGHEST PRICED PIZZA



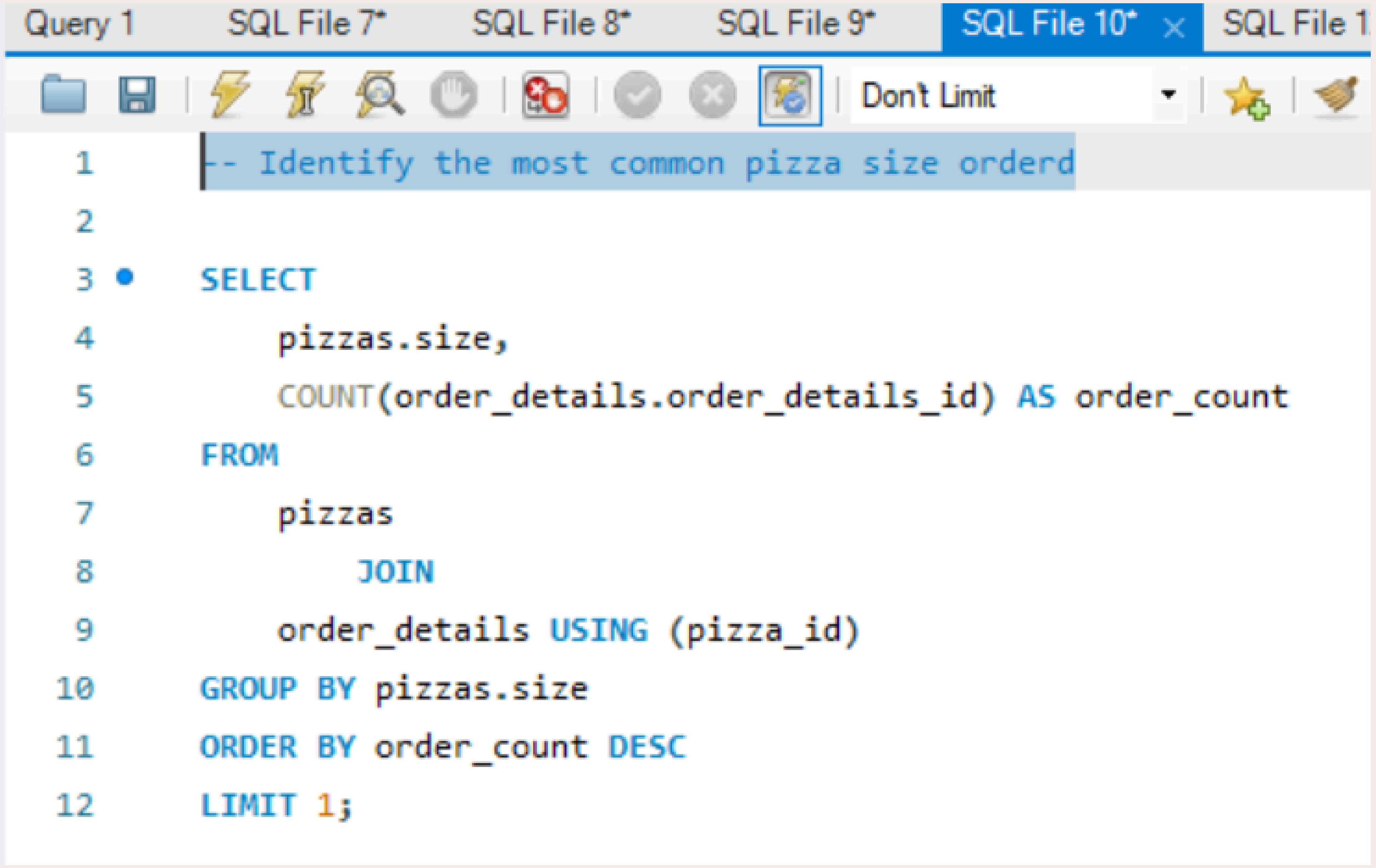
```
-- Identify the highest priced pizza

SELECT
    pizza_typeS.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas USING (pizza_type_id)
ORDER BY price DESC
LIMIT 1
```



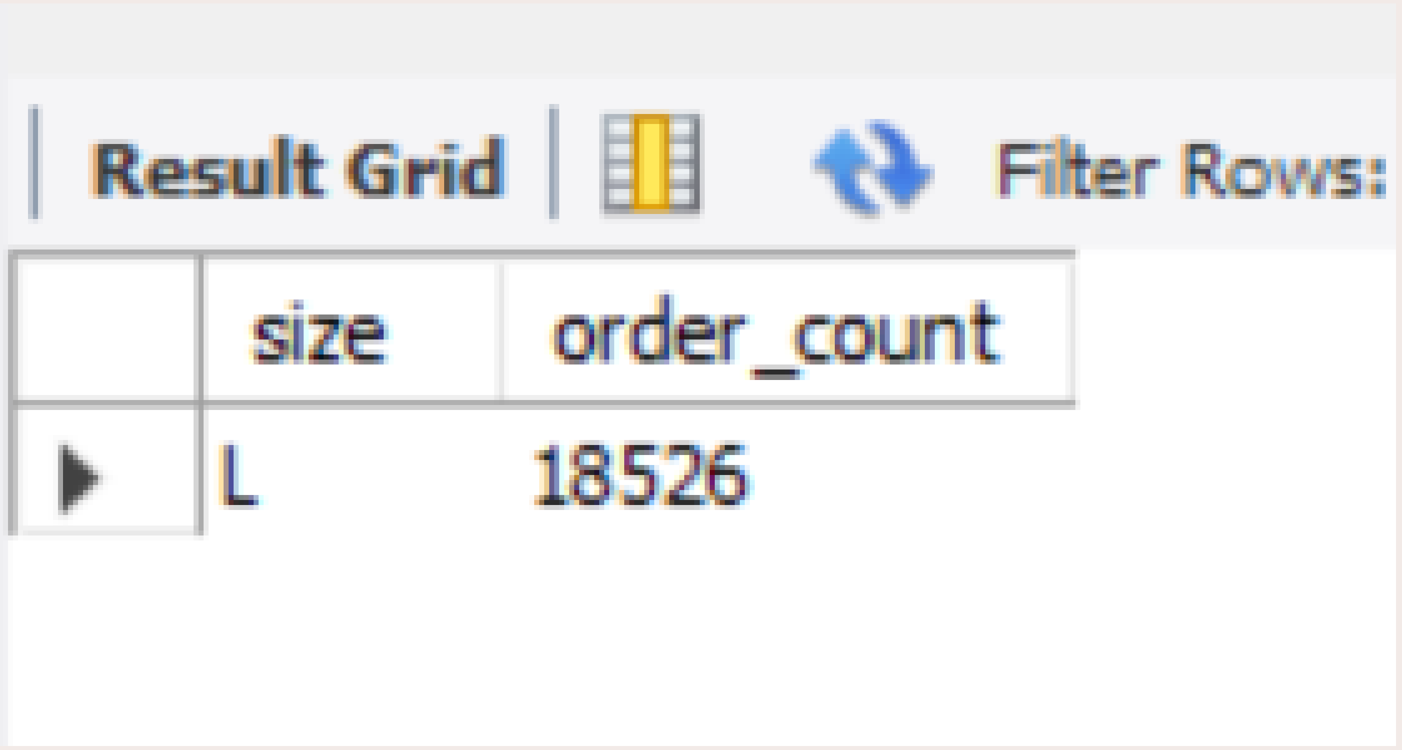
name	price
The Greek Pizza	35.95

Identify the most common pizza size orderd



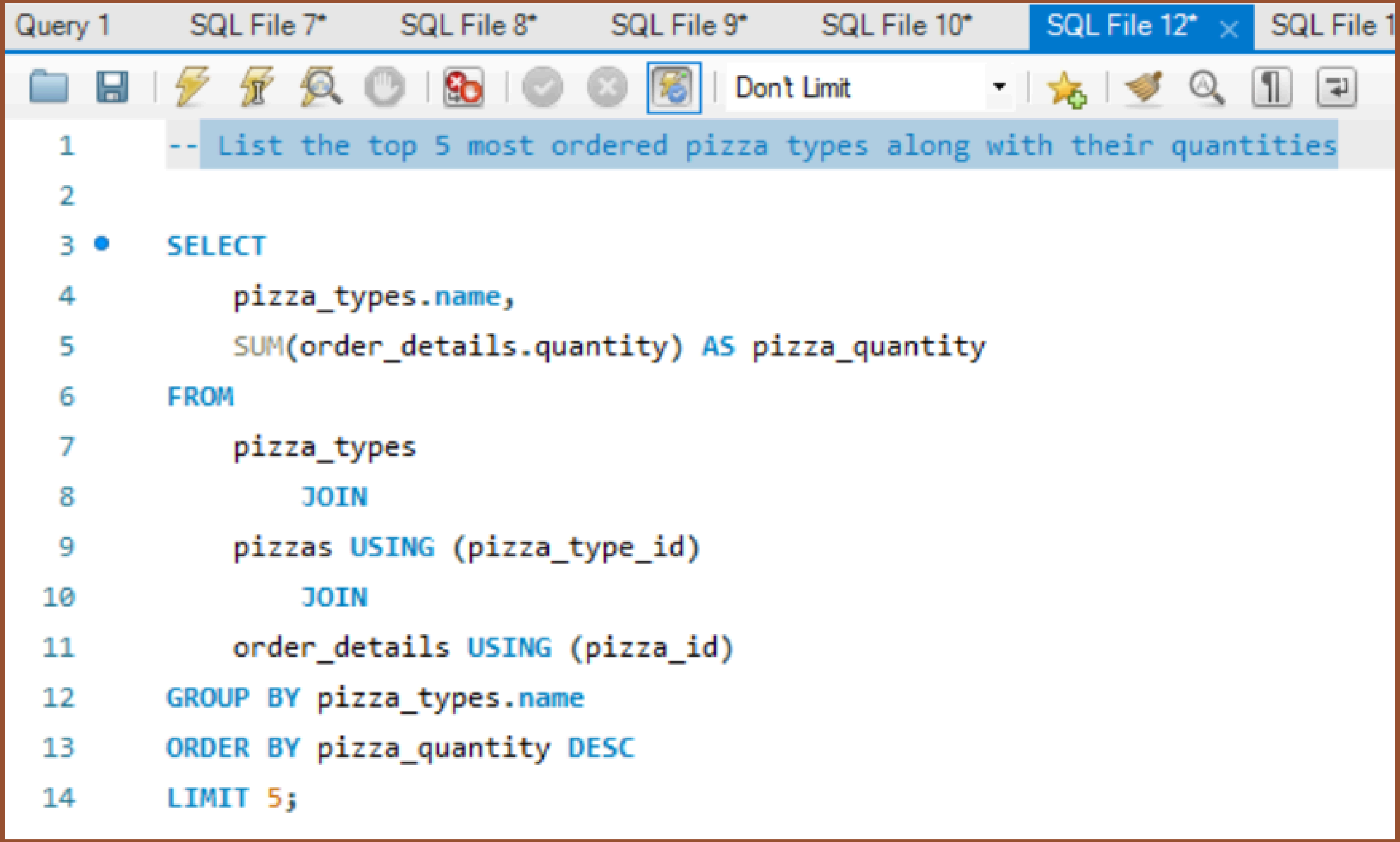
```sql
-- Identify the most common pizza size orderd

SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details USING (pizza_id)
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```



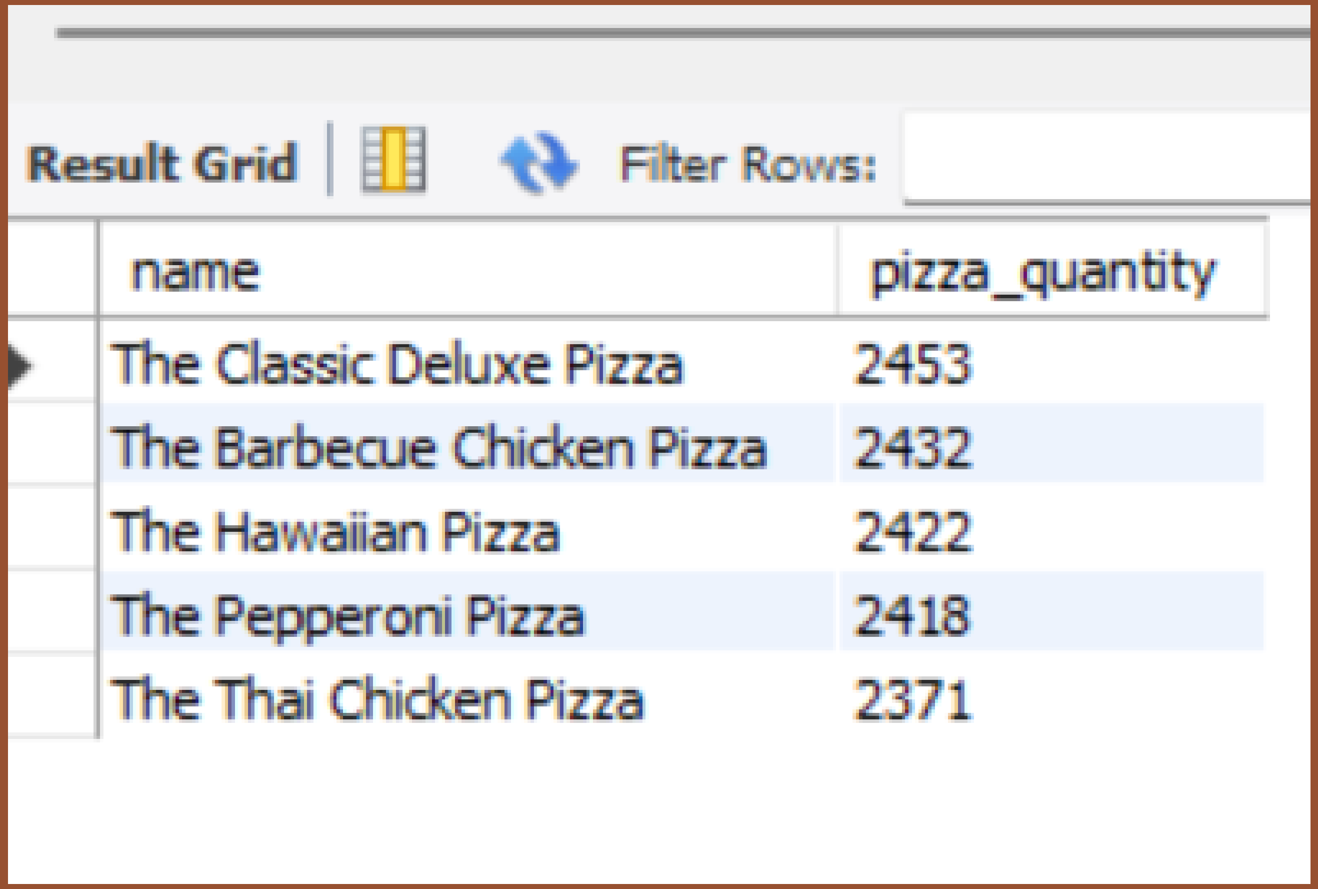
size	order_count
L	18526

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



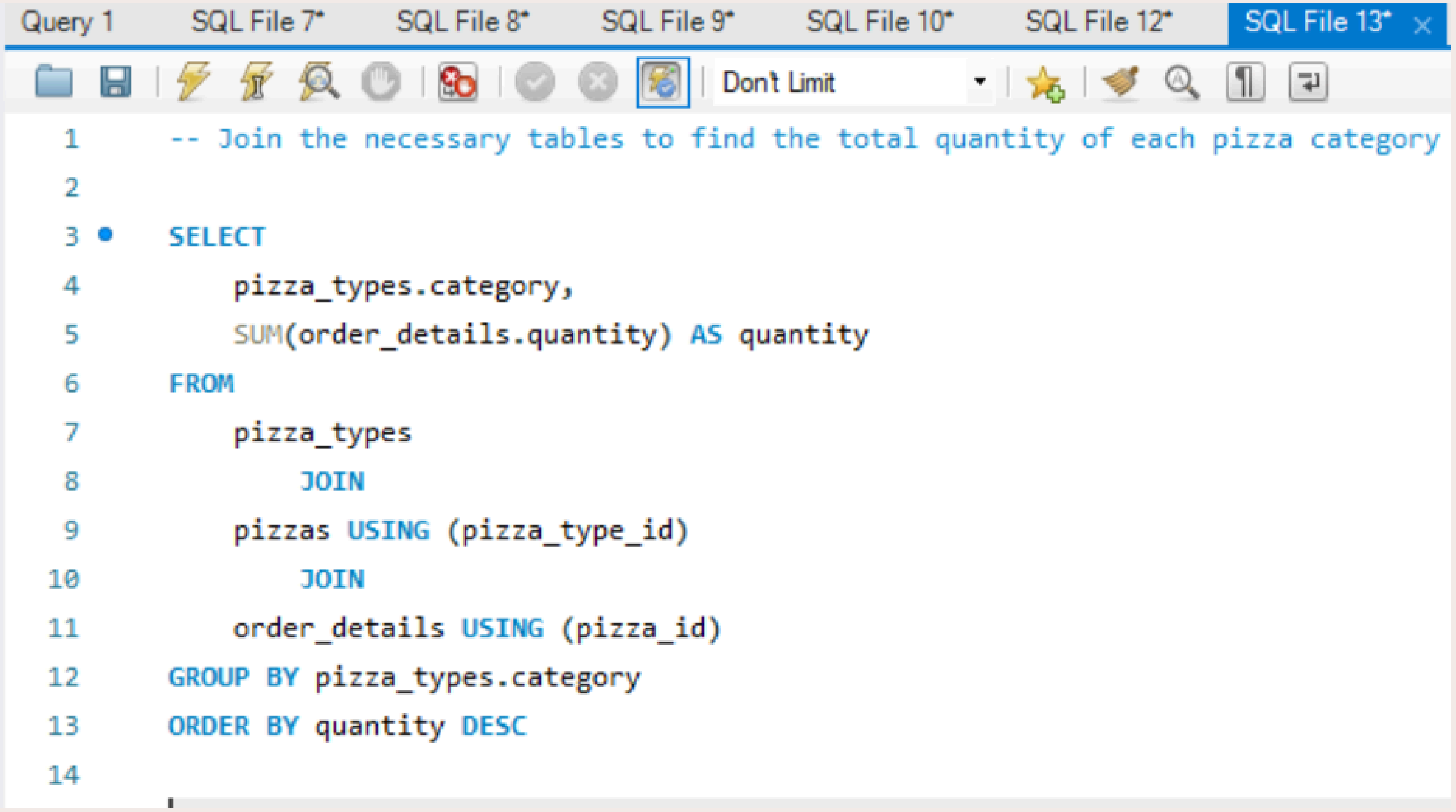
```sql
-- List the top 5 most ordered pizza types along with their quantities

SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS pizza_quantity
FROM
    pizza_types
        JOIN
    pizzas USING (pizza_type_id)
        JOIN
    order_details USING (pizza_id)
GROUP BY pizza_types.name
ORDER BY pizza_quantity DESC
LIMIT 5;
```



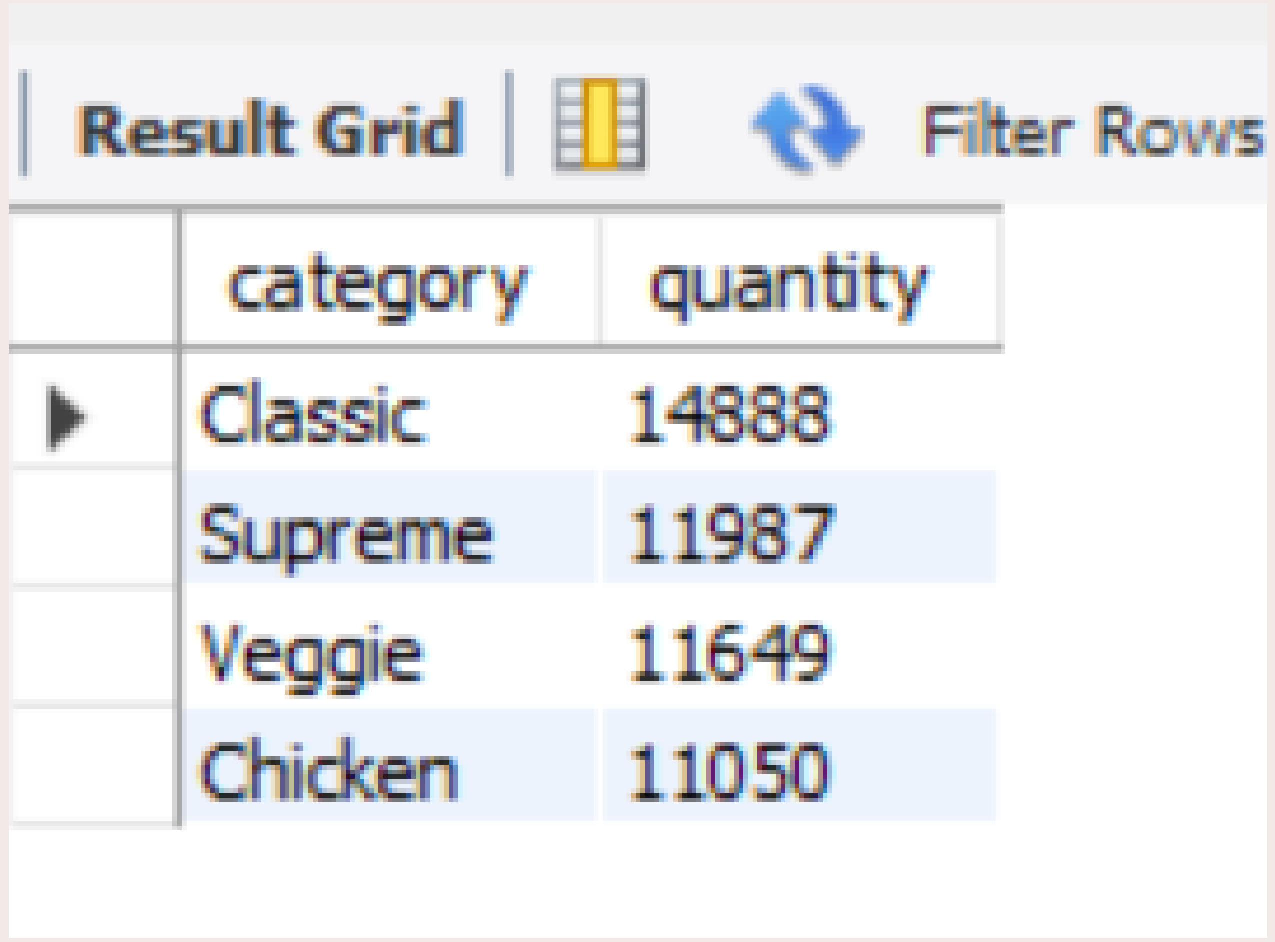
name	pizza_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY



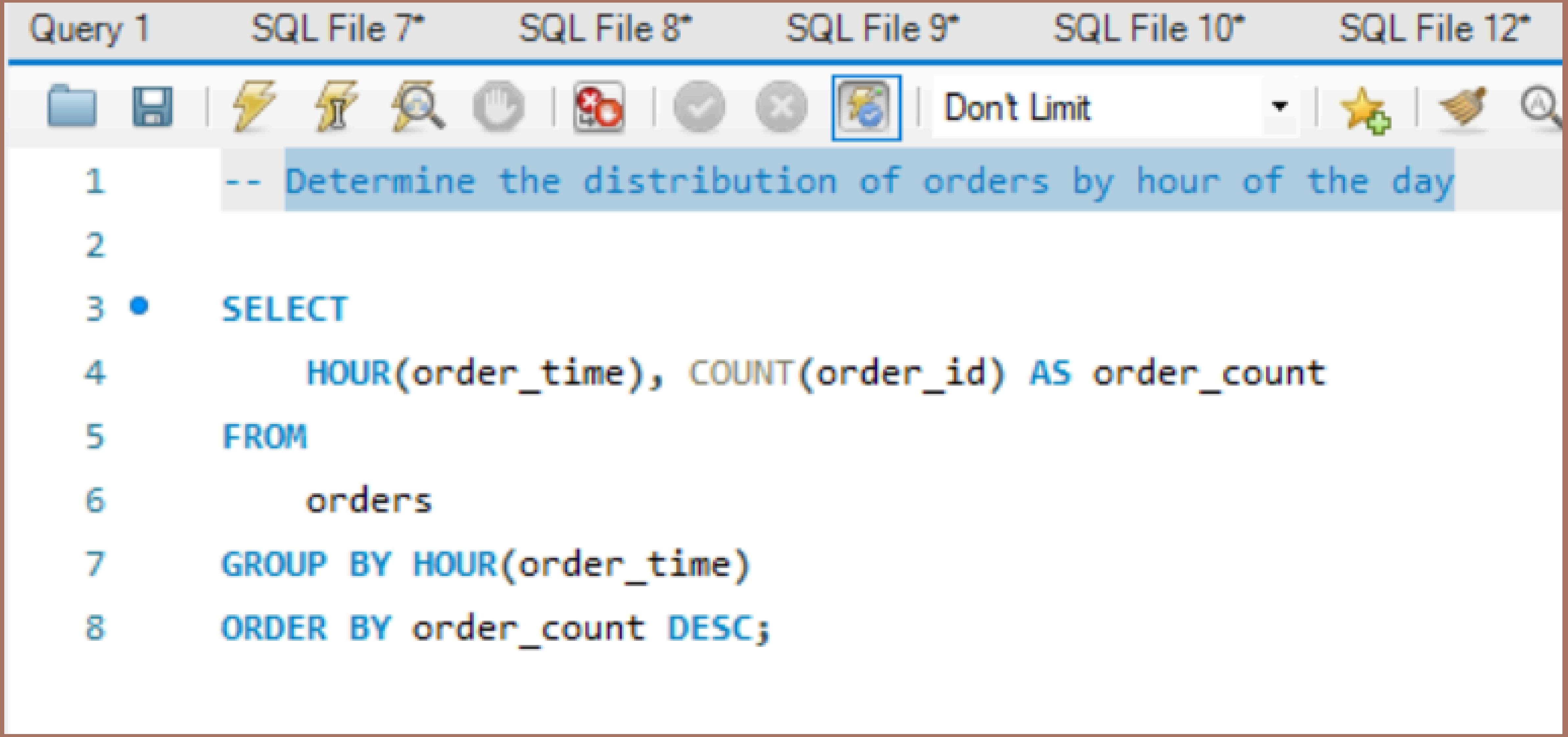
```sql
-- Join the necessary tables to find the total quantity of each pizza category

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



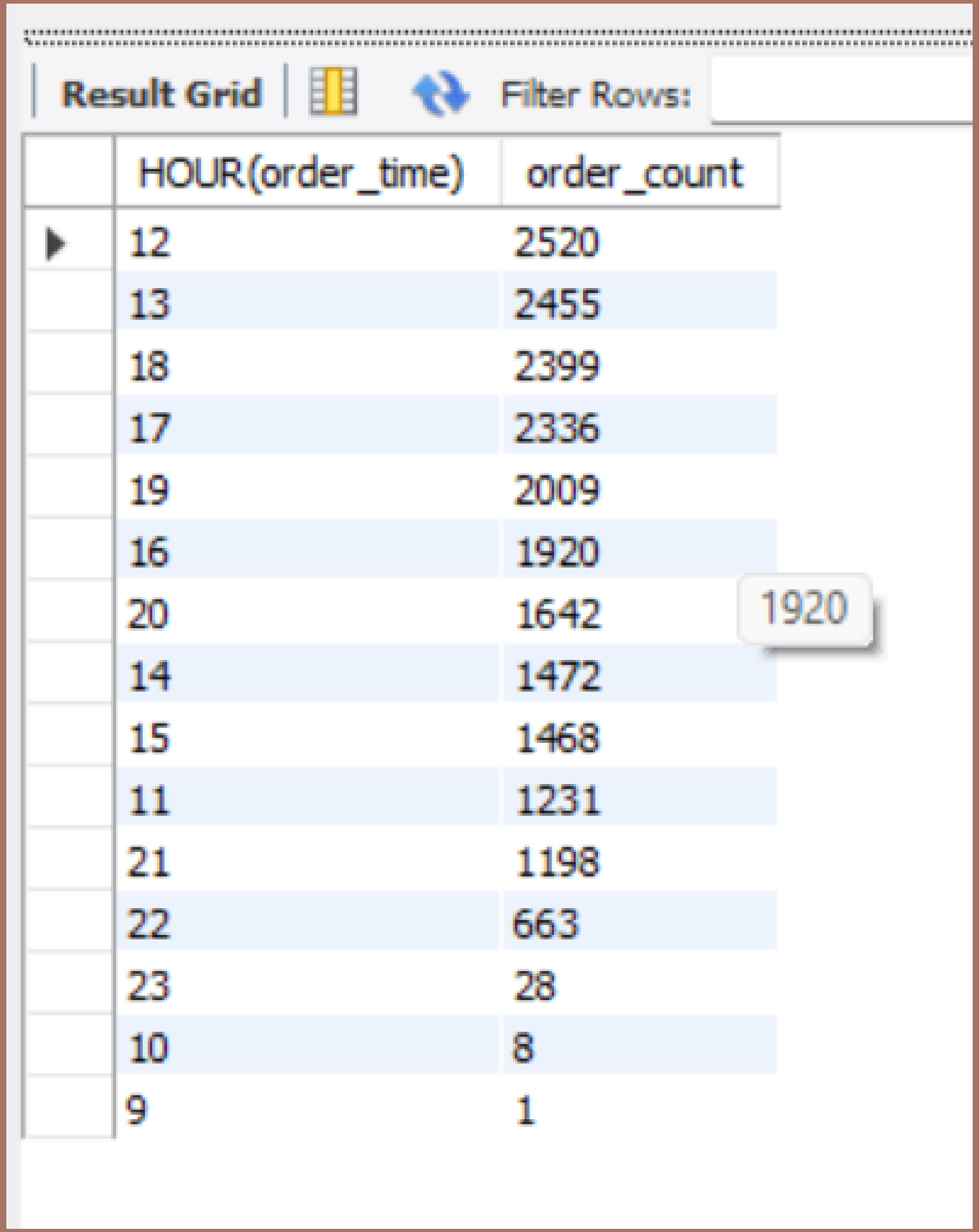
category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY



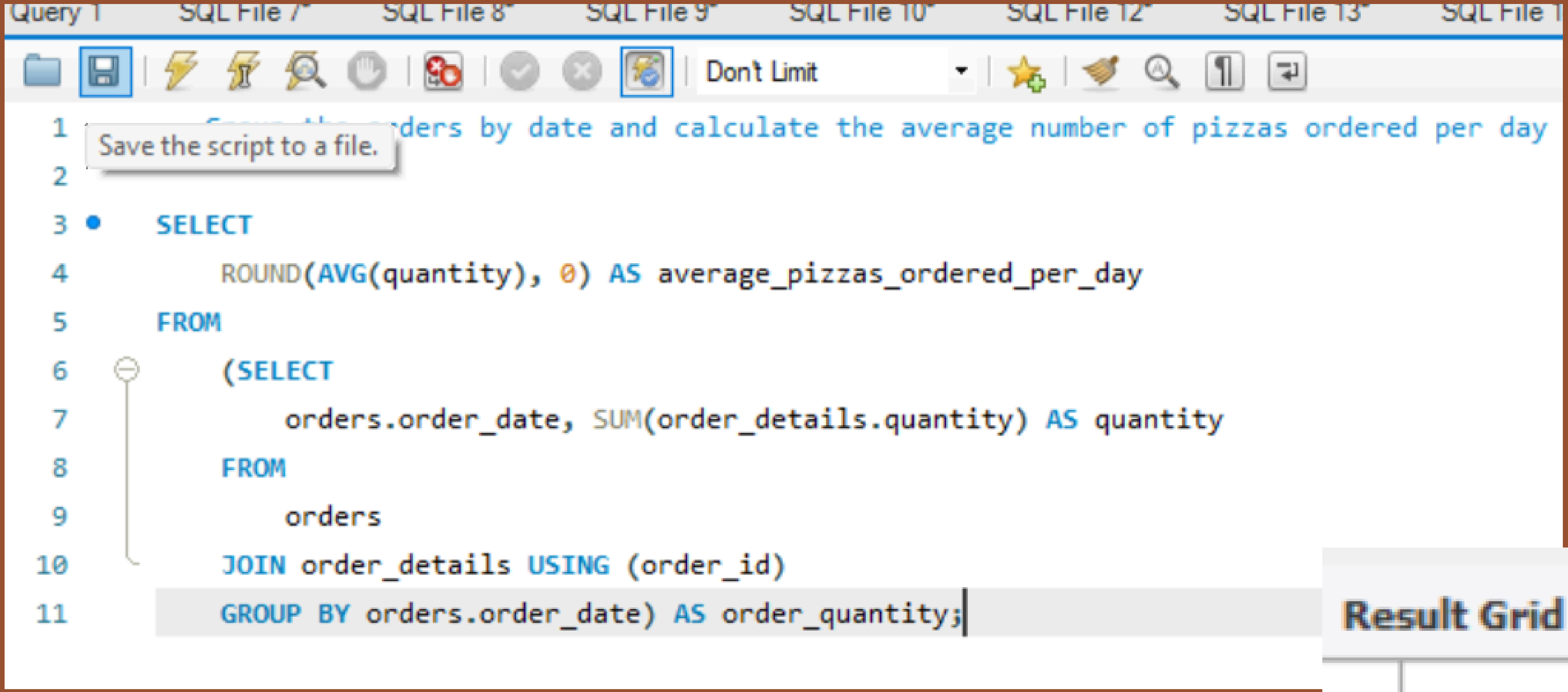
```
-- Determine the distribution of orders by hour of the day

SELECT
    HOUR(order_time), COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time)
ORDER BY order_count DESC;
```



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

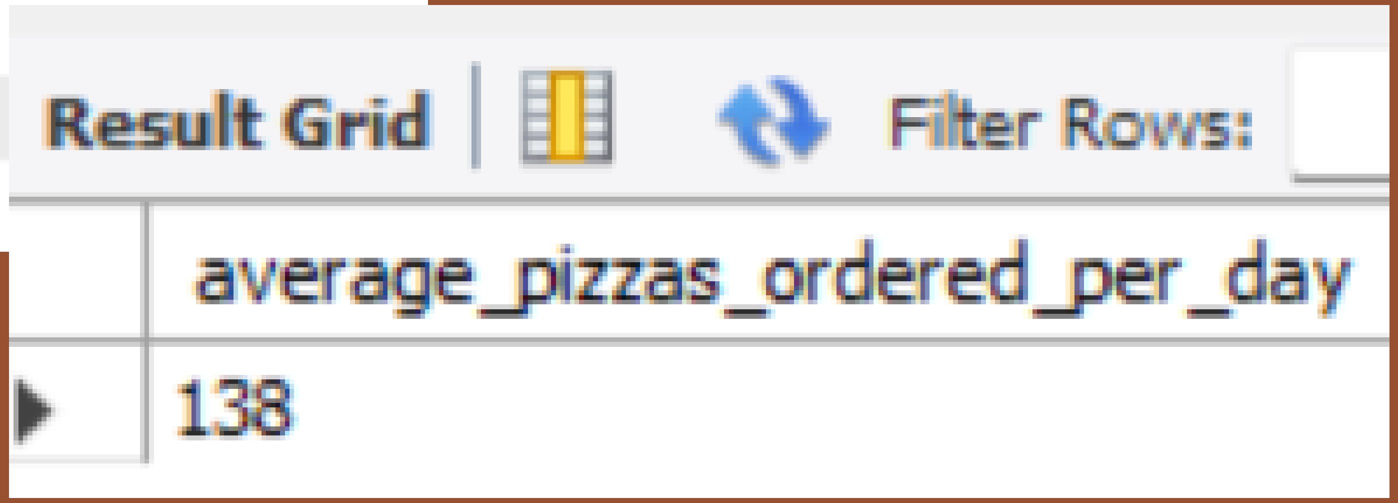
GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY



Don't Limit

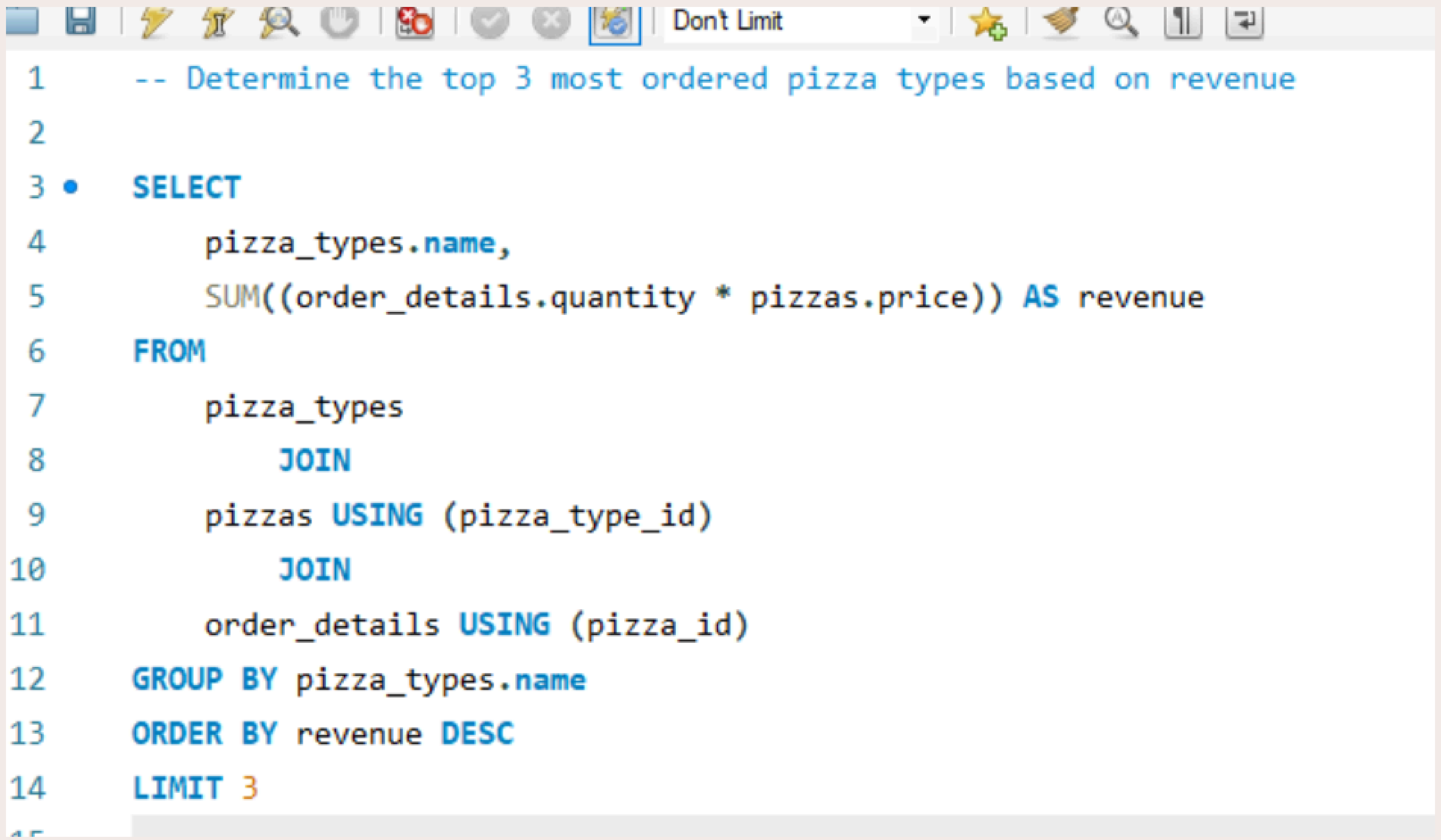
```
-- Group the orders by date and calculate the average number of pizzas ordered per day

SELECT
    ROUND(AVG(quantity), 0) AS average_pizzas_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details USING (order_id)
    GROUP BY orders.order_date) AS order_quantity;
```



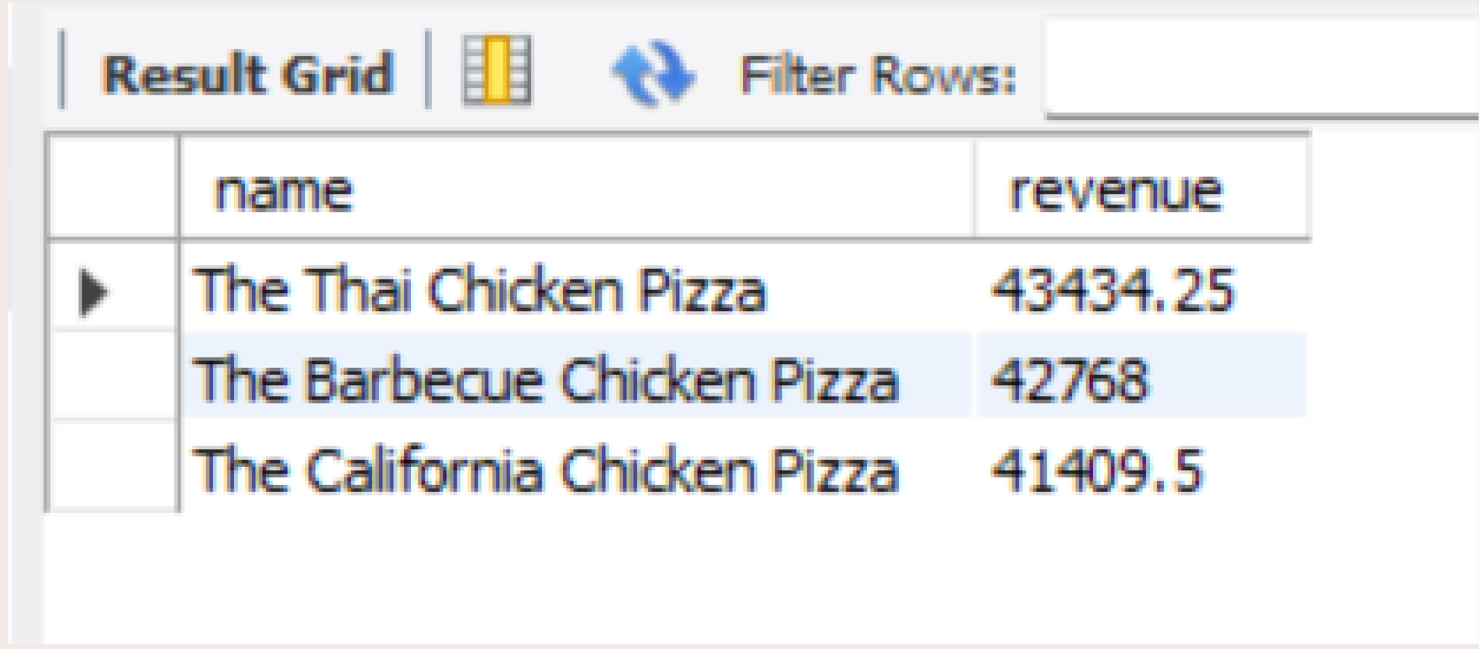
```
average_pizzas_ordered_per_day
138
```

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE



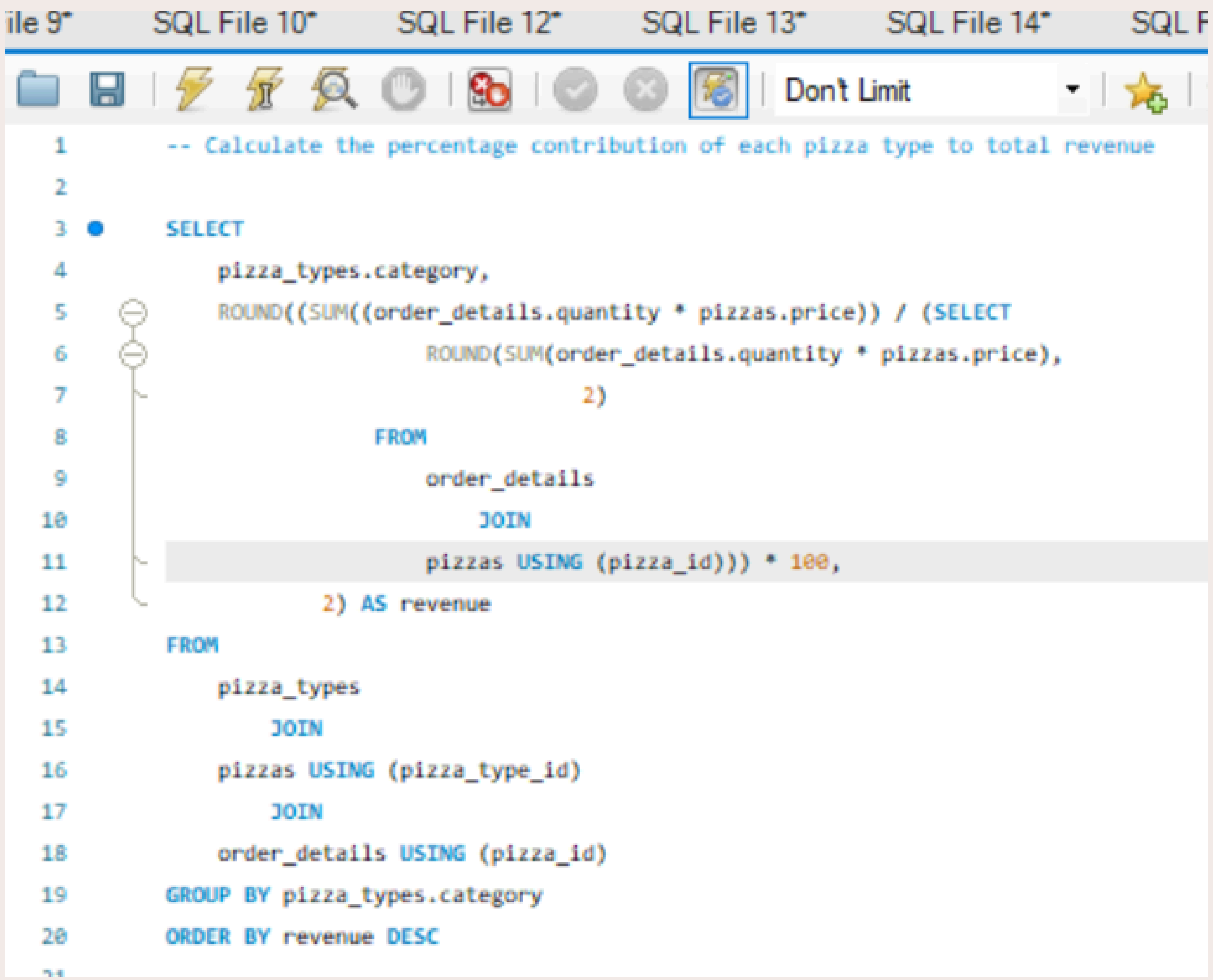
```sql
-- Determine the top 3 most ordered pizza types based on revenue

SELECT
    pizza_types.name,
    SUM((order_details.quantity * pizzas.price)) AS revenue
FROM
    pizza_types
        JOIN
    pizzas USING (pizza_type_id)
        JOIN
    order_details USING (pizza_id)
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3
```



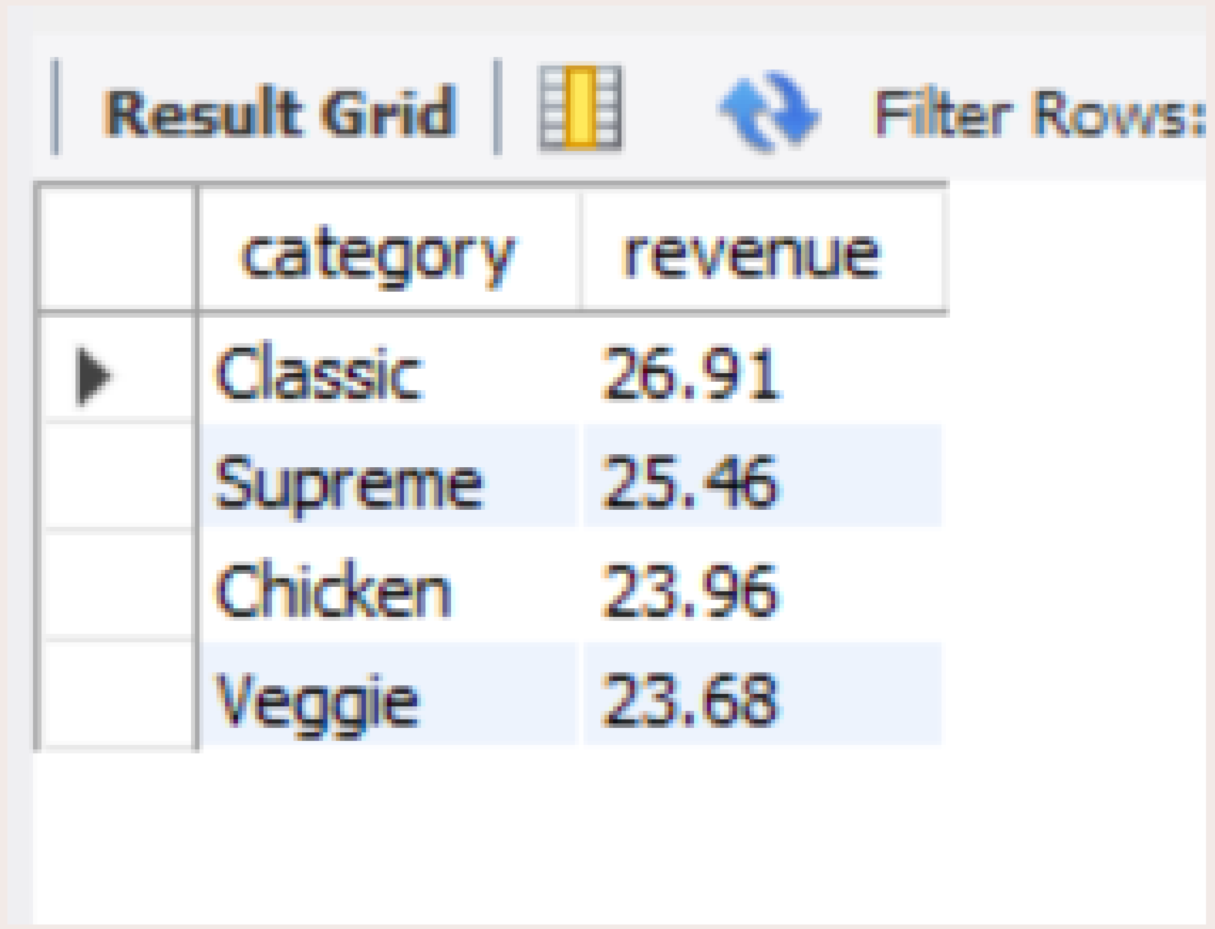
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Calculate the percentage contribution of each pizza type to total revenue



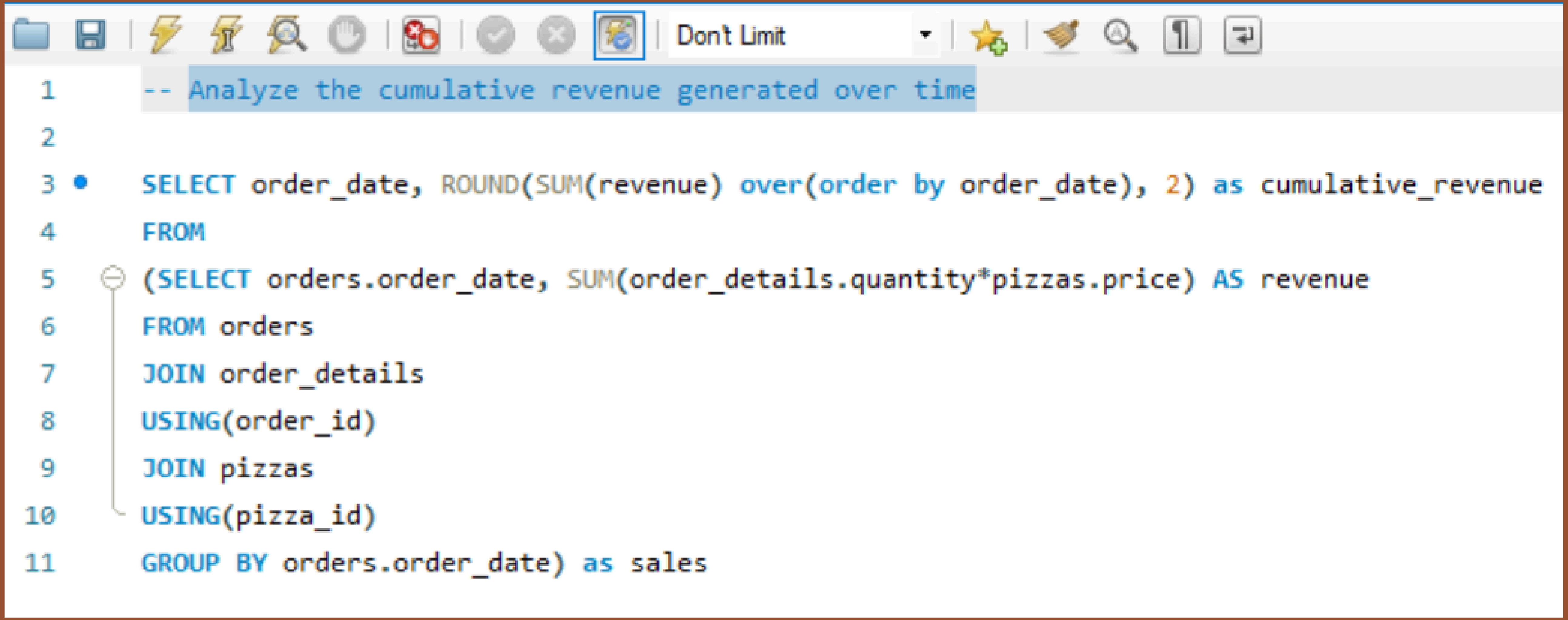
```
-- Calculate the percentage contribution of each pizza type to total revenue

SELECT
    pizza_types.category,
    ROUND((SUM((order_details.quantity * pizzas.price)) / (SELECT
                ROUND(SUM(order_details.quantity * pizzas.price),
                        2)
            FROM
                order_details
                    JOIN
                pizzas USING (pizza_id))) * 100,
            2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas USING (pizza_type_id)
        JOIN
    order_details USING (pizza_id)
GROUP BY pizza_types.category
ORDER BY revenue DESC
```



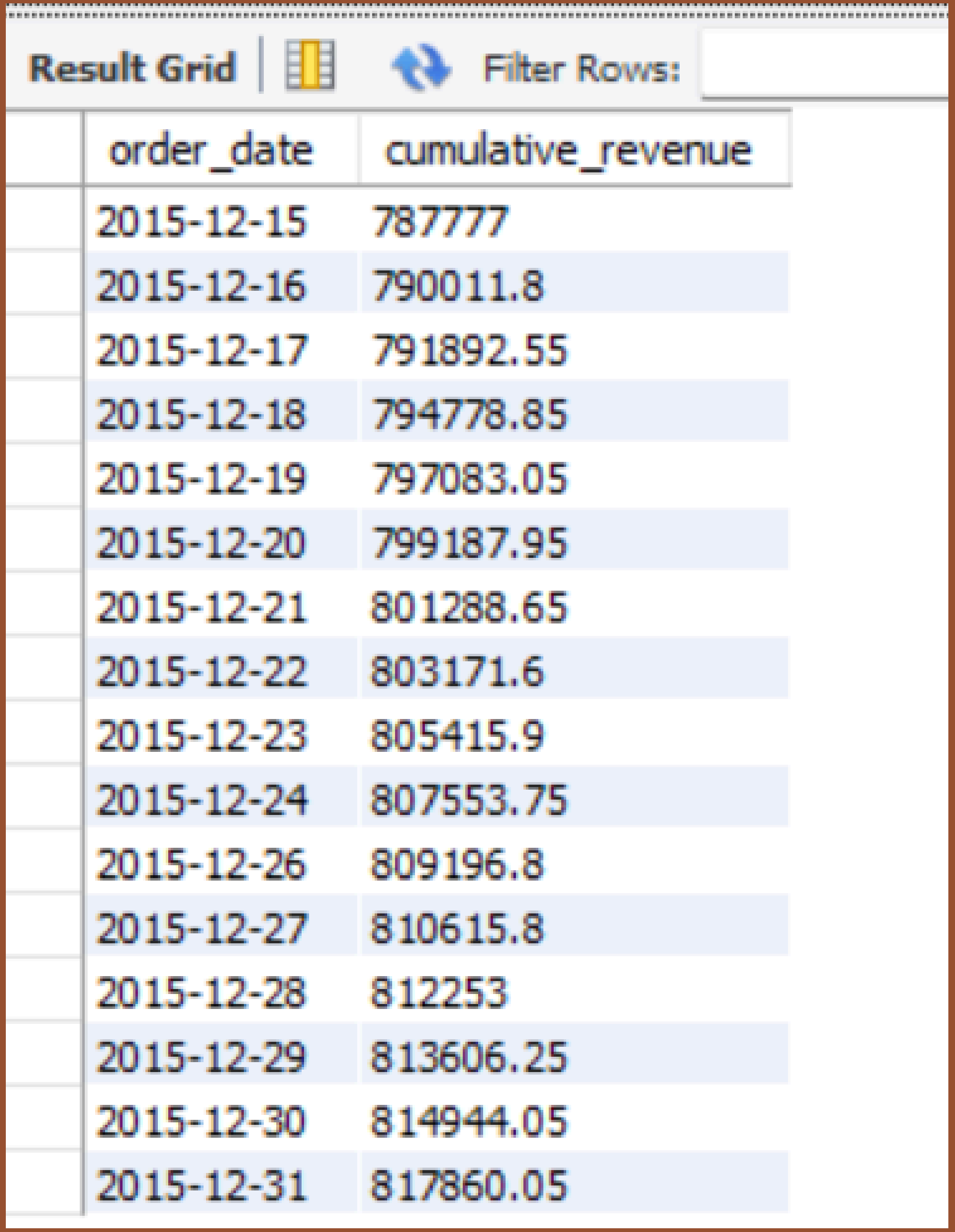
category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME



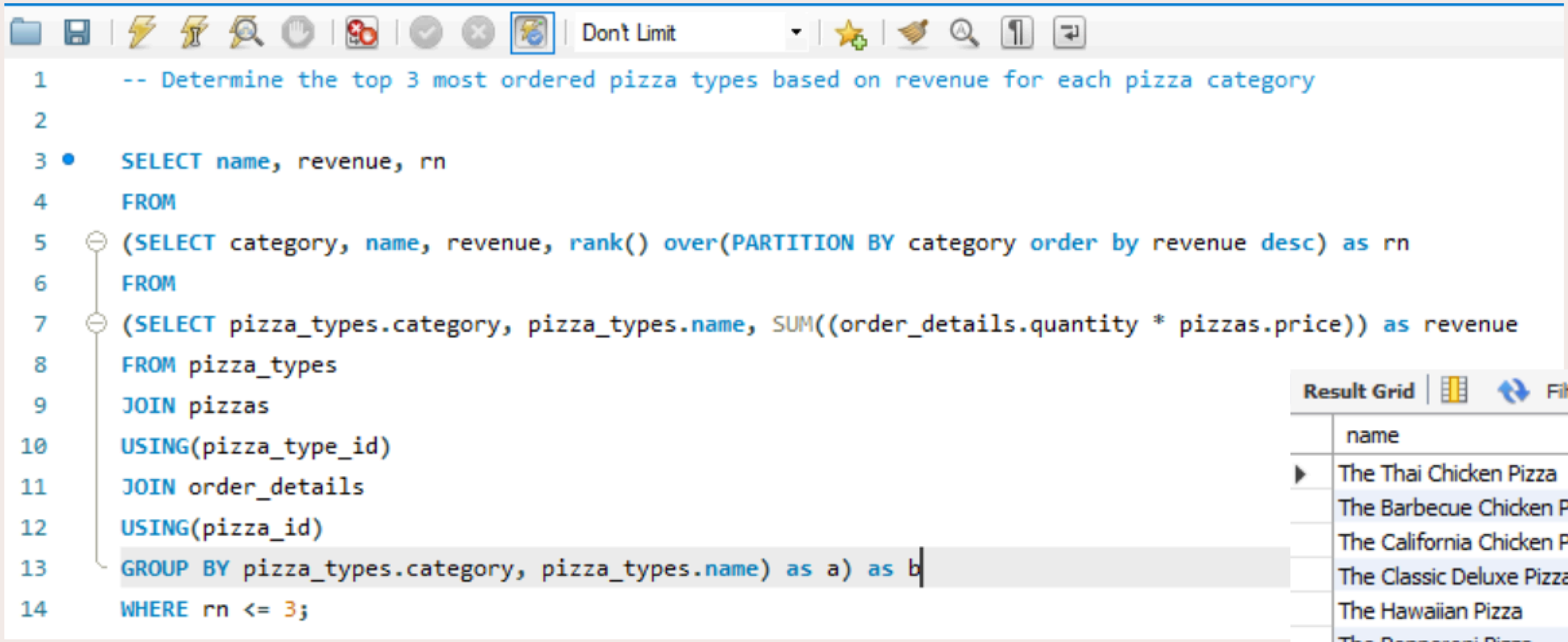
```sql
-- Analyze the cumulative revenue generated over time

SELECT order_date, ROUND(SUM(revenue) over(order by order_date), 2) as cumulative_revenue
FROM
(SELECT orders.order_date, SUM(order_details.quantity*pizzas.price) AS revenue
FROM orders
JOIN order_details
USING(order_id)
JOIN pizzas
USING(pizza_id)
GROUP BY orders.order_date) as sales
```



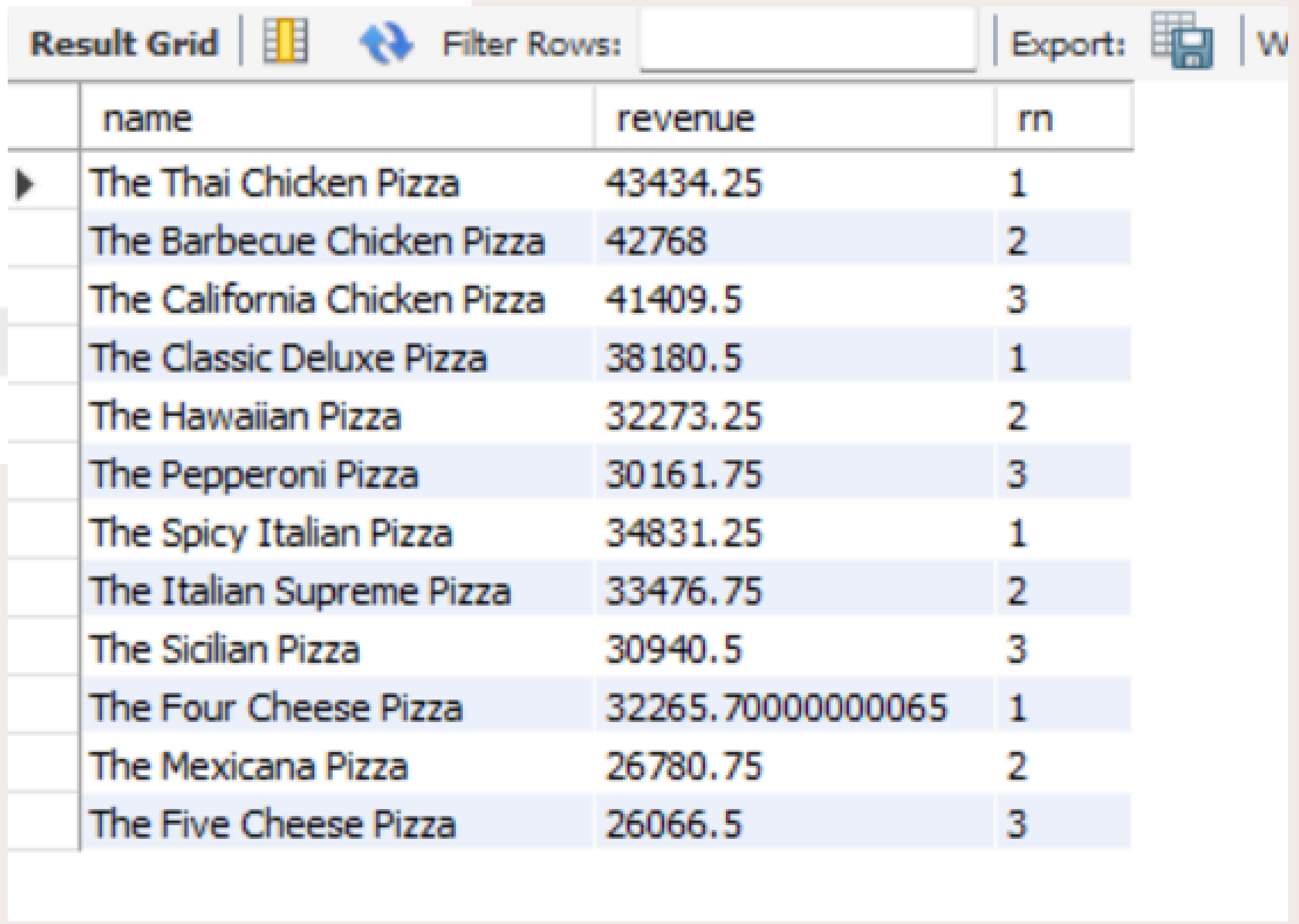
order_date	cumulative_revenue
2015-12-15	787777
2015-12-16	790011.8
2015-12-17	791892.55
2015-12-18	794778.85
2015-12-19	797083.05
2015-12-20	799187.95
2015-12-21	801288.65
2015-12-22	803171.6
2015-12-23	805415.9
2015-12-24	807553.75
2015-12-26	809196.8
2015-12-27	810615.8
2015-12-28	812253
2015-12-29	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



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

SELECT name, revenue, rn
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
USING(pizza_type_id)
JOIN order_details
USING(pizza_id)
GROUP BY pizza_types.category, pizza_types.name) as a) as b
WHERE rn <= 3;
```



name	revenue	rn
The Thai Chicken Pizza	43434.25	1
The Barbecue Chicken Pizza	42768	2
The California Chicken Pizza	41409.5	3
The Classic Deluxe Pizza	38180.5	1
The Hawaiian Pizza	32273.25	2
The Pepperoni Pizza	30161.75	3
The Spicy Italian Pizza	34831.25	1
The Italian Supreme Pizza	33476.75	2
The Sicilian Pizza	30940.5	3
The Four Cheese Pizza	32265.70000000065	1
The Mexicana Pizza	26780.75	2
The Five Cheese Pizza	26066.5	3

THANK YOU !