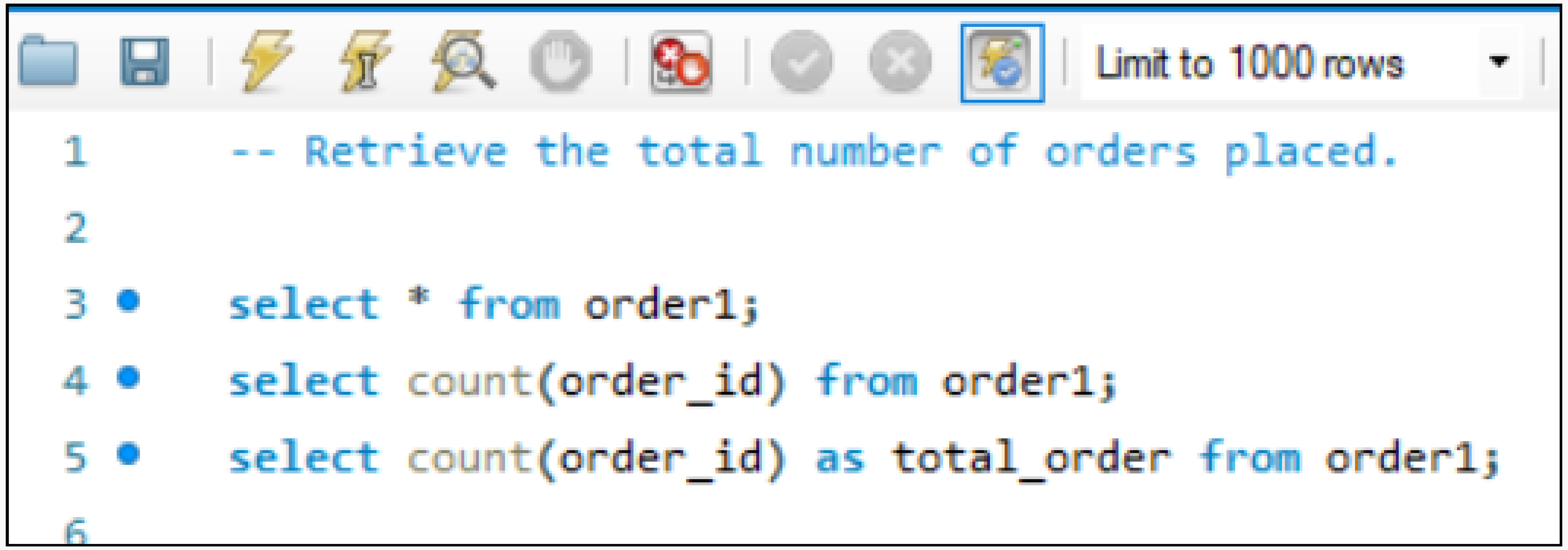
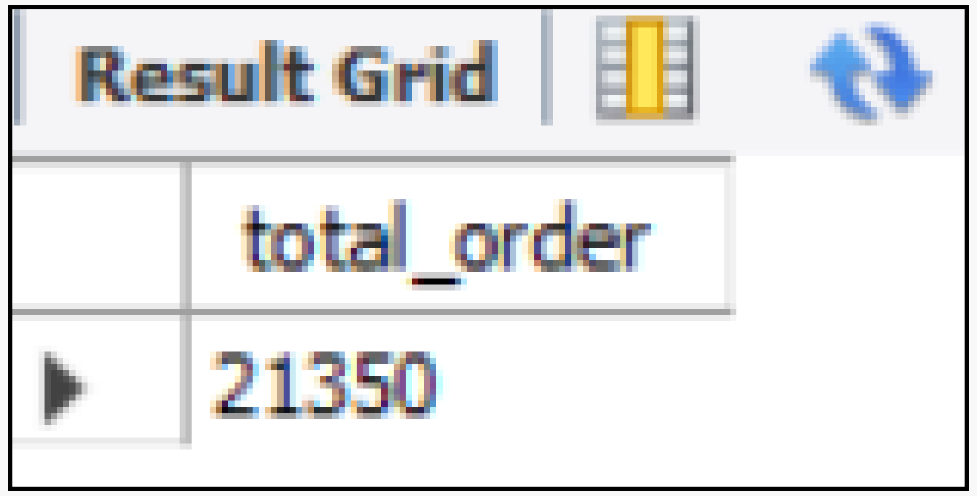
Pizza Hut Sales Project

Objective: To find useful insights from Pizza Hut's sales data that help make better business decisions, boost sales, and improve operations.

Retrieve the total number of orders placed.

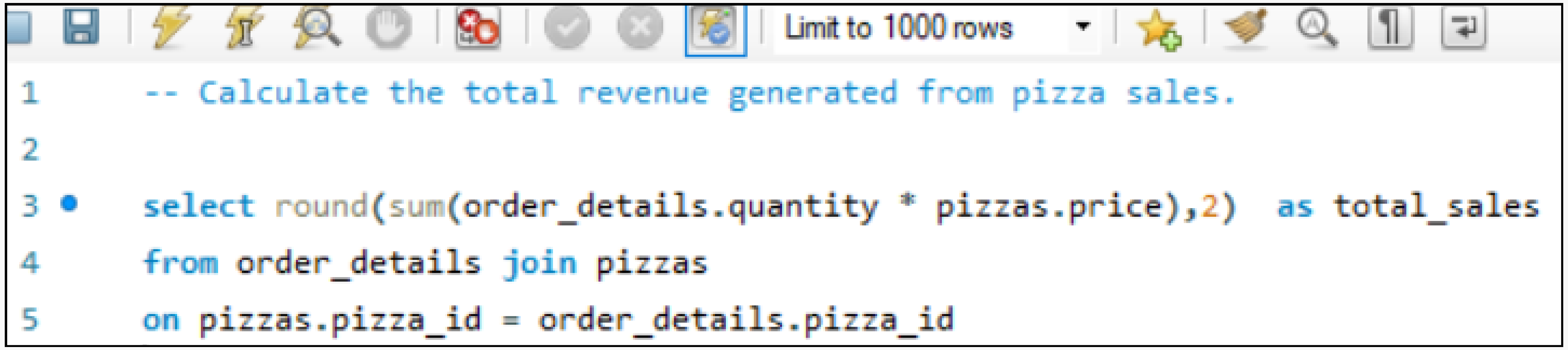


```sql
-- Retrieve the total number of orders placed.

select * from order1;
select count(order_id) from order1;
select count(order_id) as total_order from order1;
```

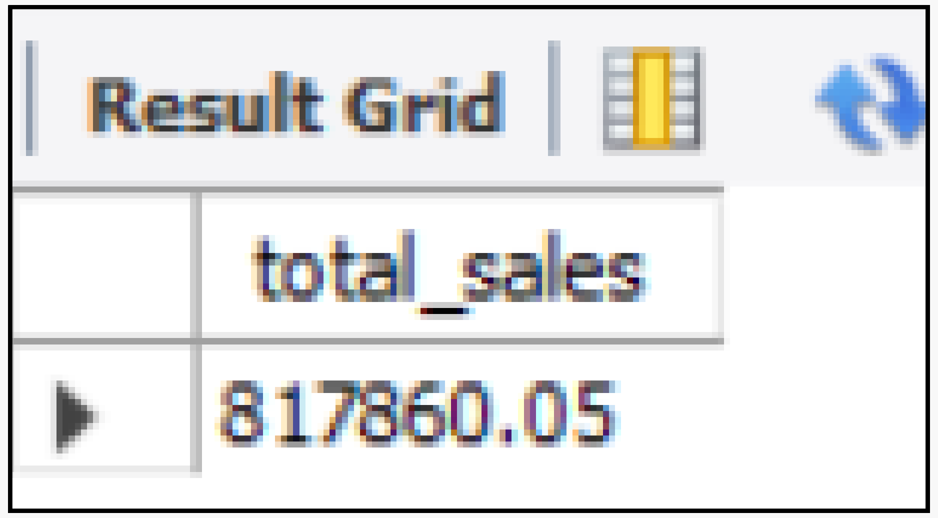


```
total_order
21350
```

Calculate the total revenue generated from pizza sales.

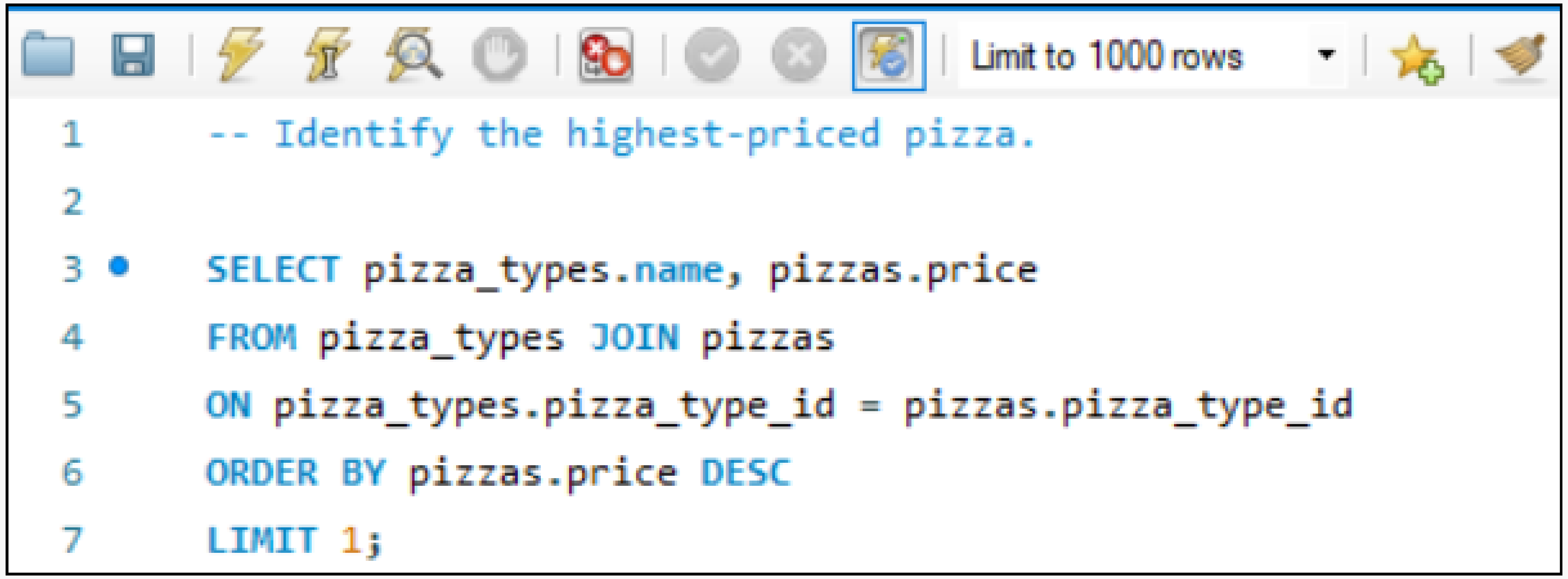


```sql
-- Calculate the total revenue generated from pizza sales.

select round(sum(order_details.quantity * pizzas.price),2)  as total_sales
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id
```



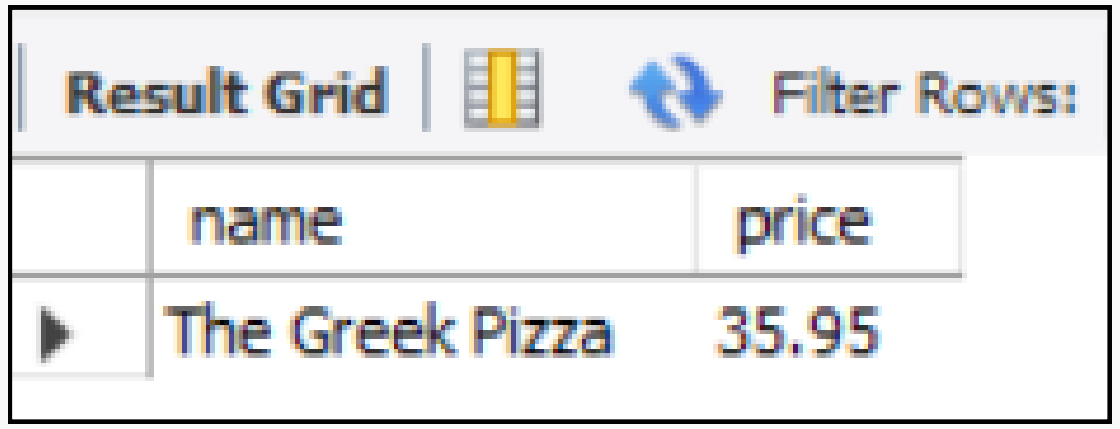
```
total_sales
817860.05
```

Identify the highest-priced pizza.



Limit to 1000 rows

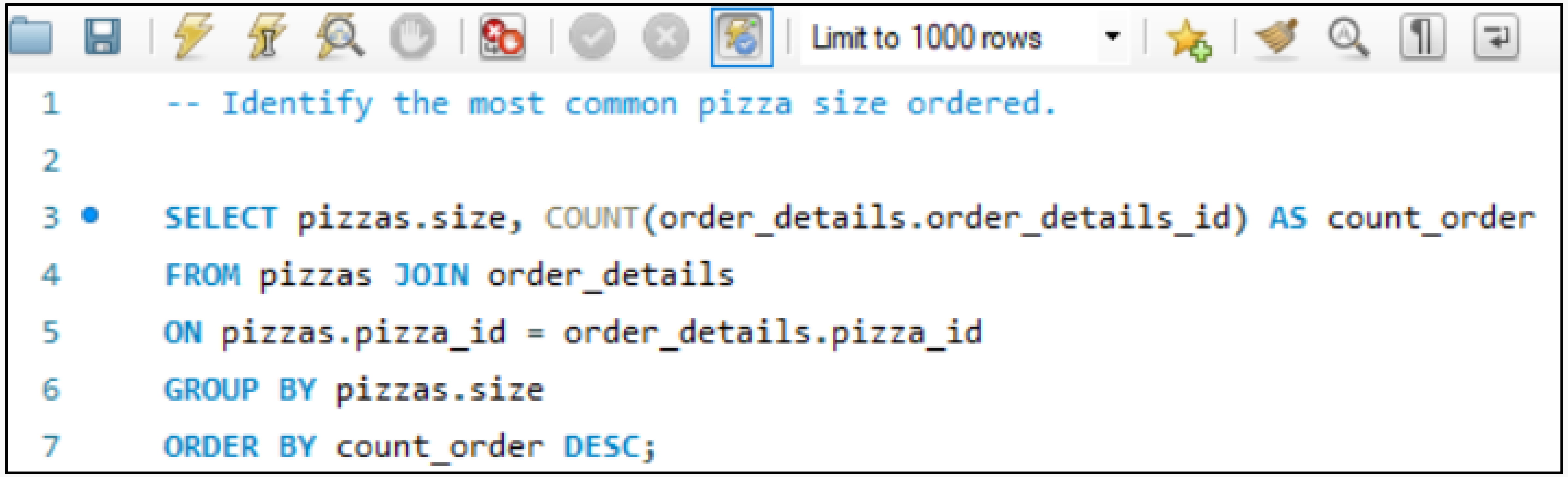
```
-- Identify the highest-priced pizza.

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



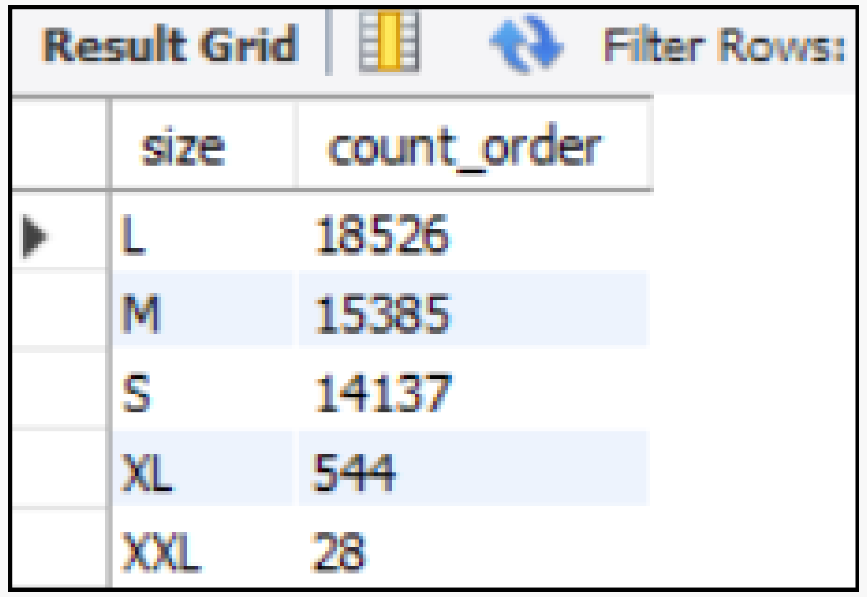
Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

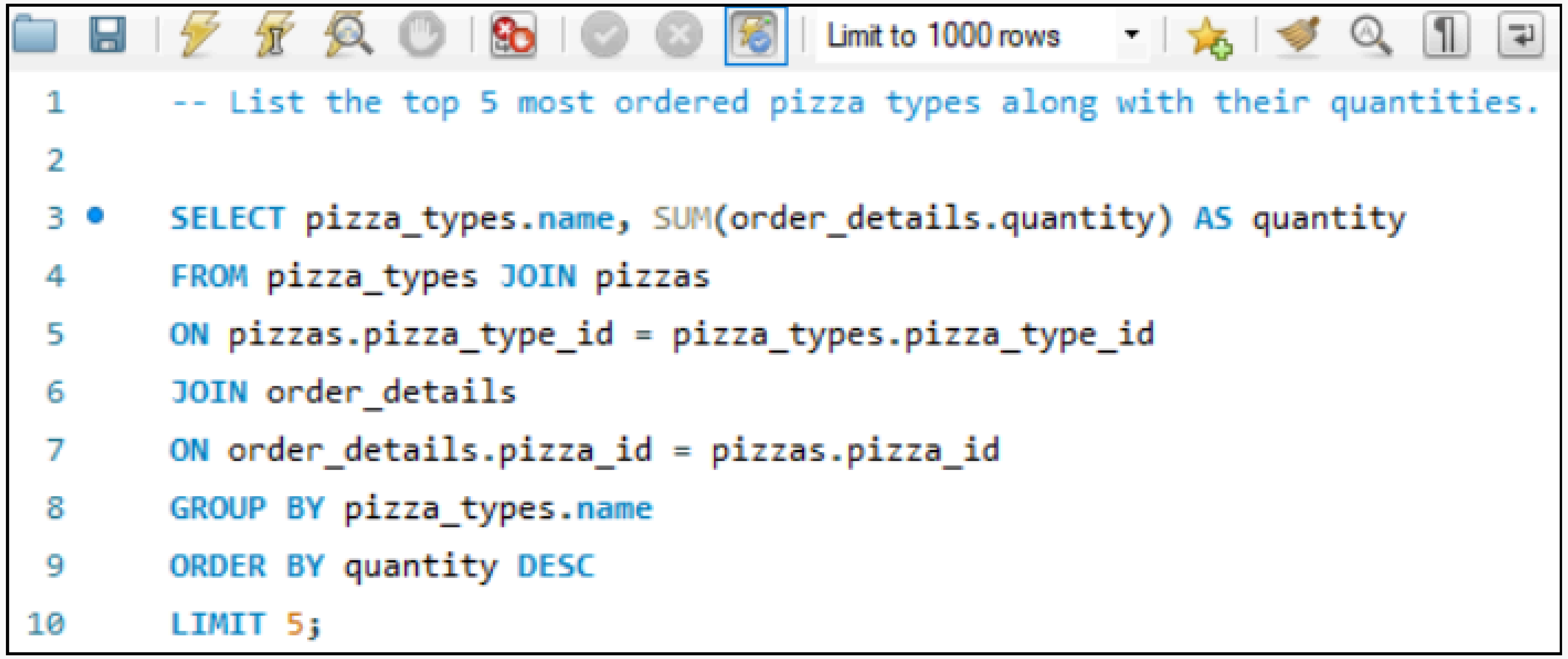


```
-- Identify the most common pizza size ordered.

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

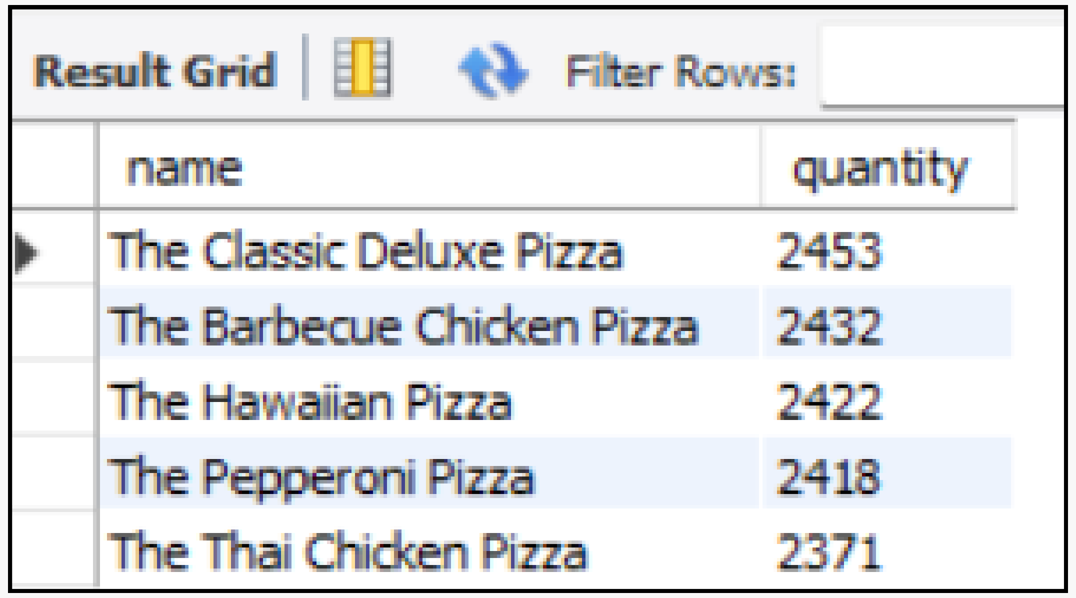


size	count_order
L	18526
M	15385
S	14137
XL	544
XXL	28

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

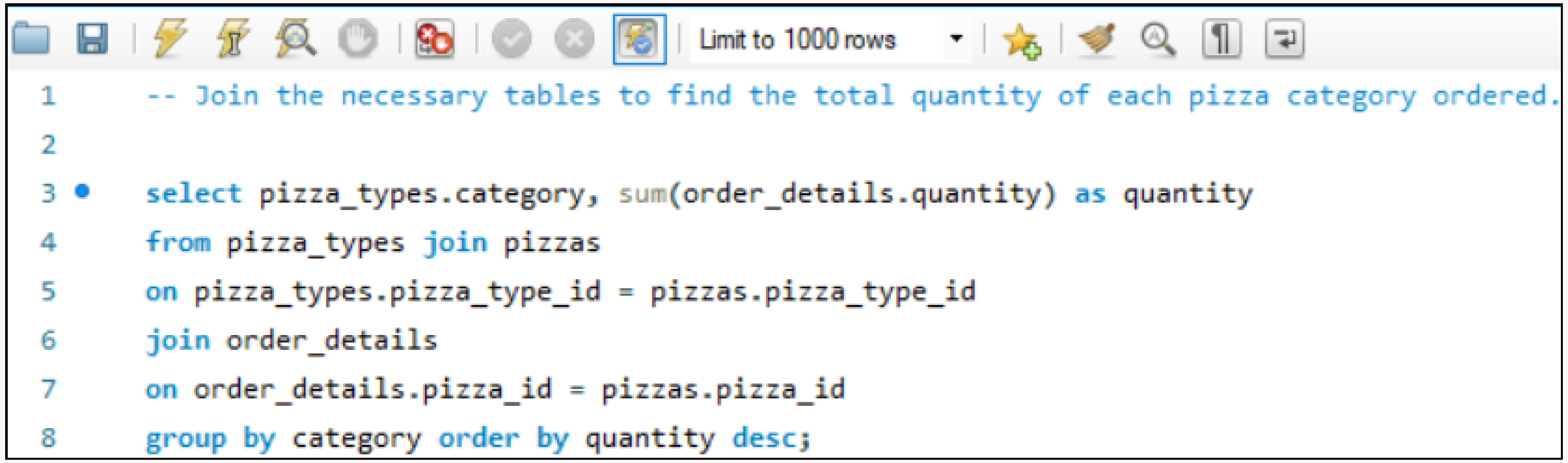


```sql
-- List the top 5 most ordered pizza types along with their quantities.

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

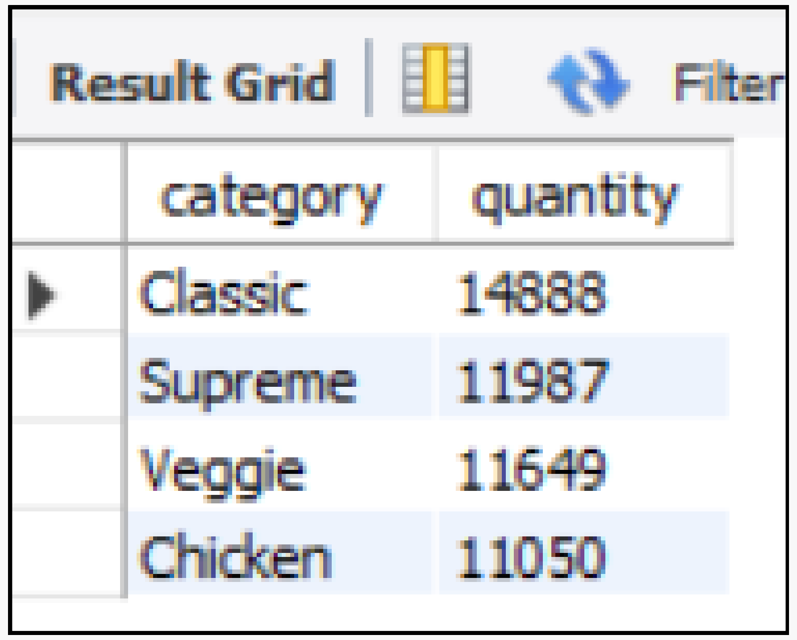

Result Grid
Filter Rows:

name	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.

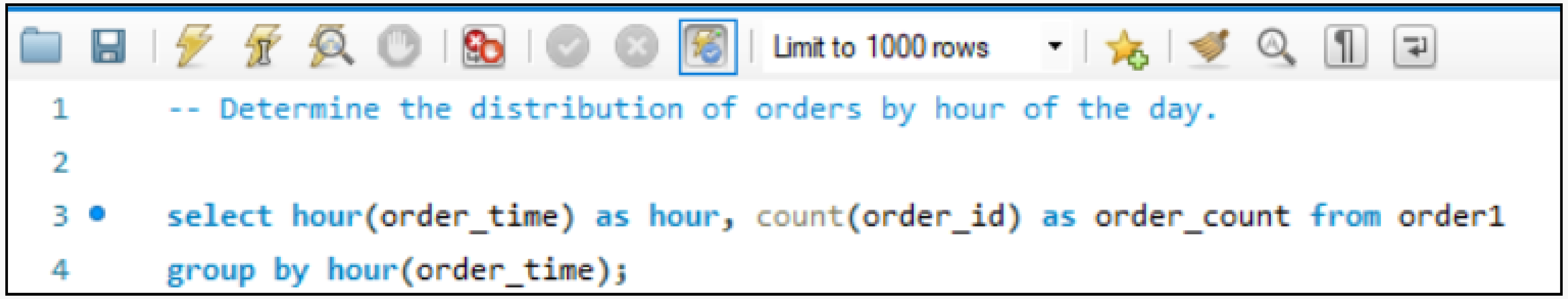


```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered.

select pizza_types.category, 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 order_details.pizza_id = pizzas.pizza_id
group by category order by quantity desc;
```

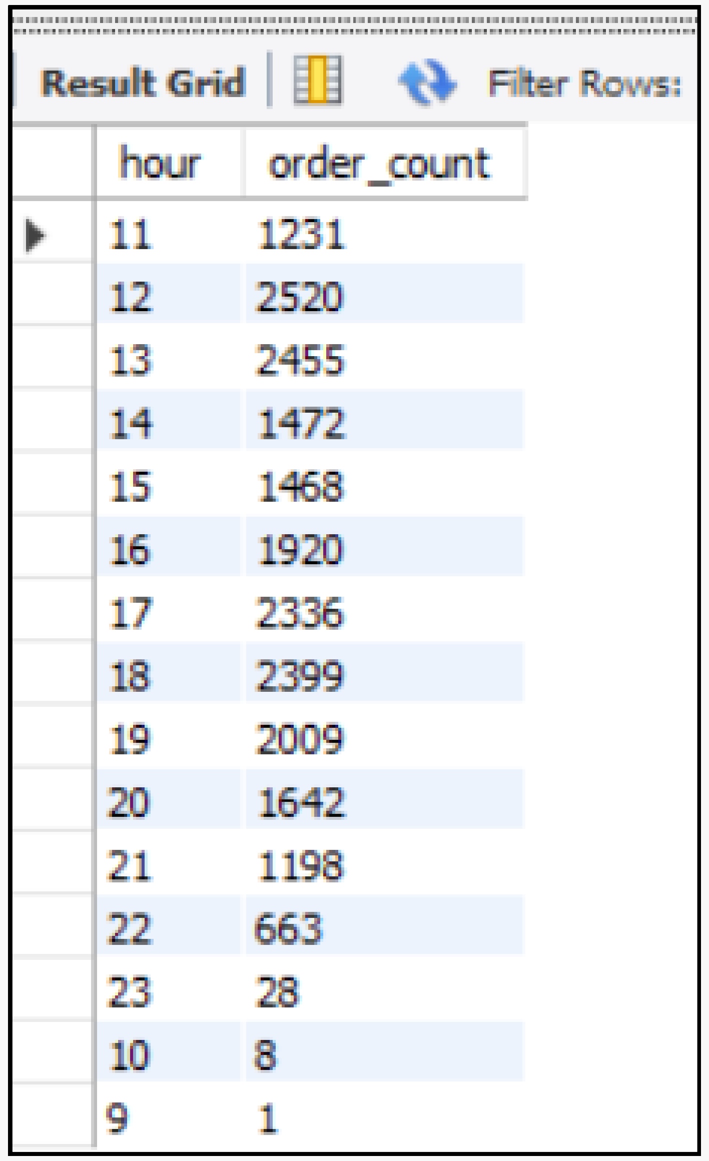


category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

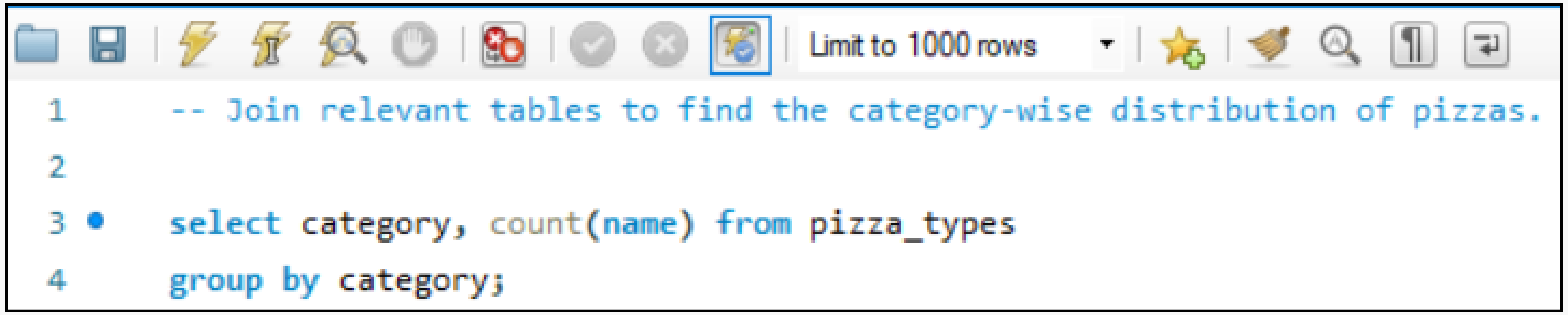


```sql
-- Determine the distribution of orders by hour of the day.

select hour(order_time) as hour, count(order_id) as order_count from order1
group by hour(order_time);
```

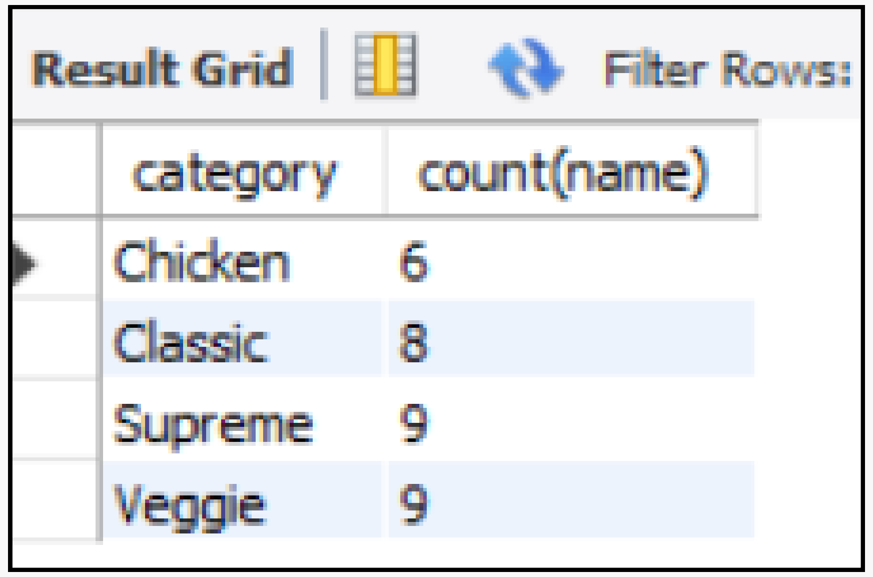

Result Grid
Filter Rows:

hour	order_count
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.

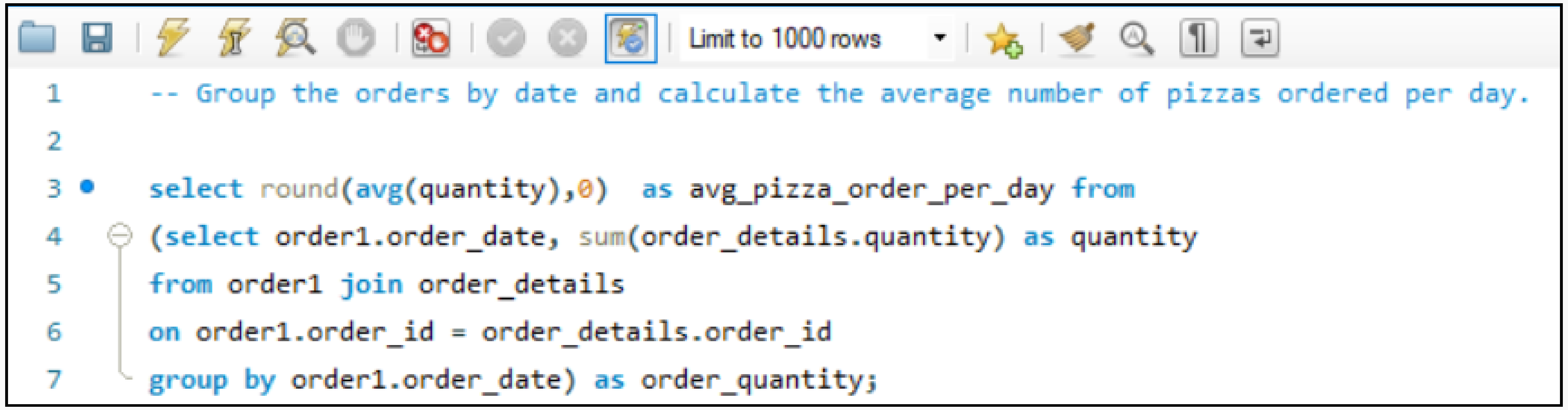


```sql
-- Join relevant tables to find the category-wise distribution of pizzas.

select category, count(name) from pizza_types
group by category;
```

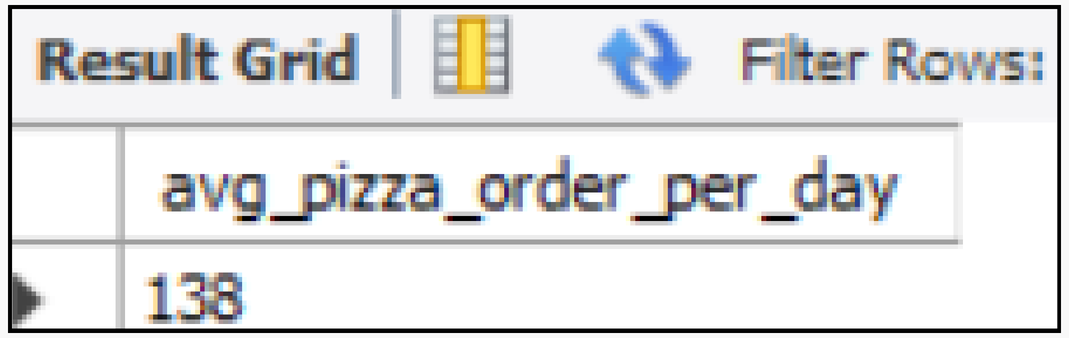


category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

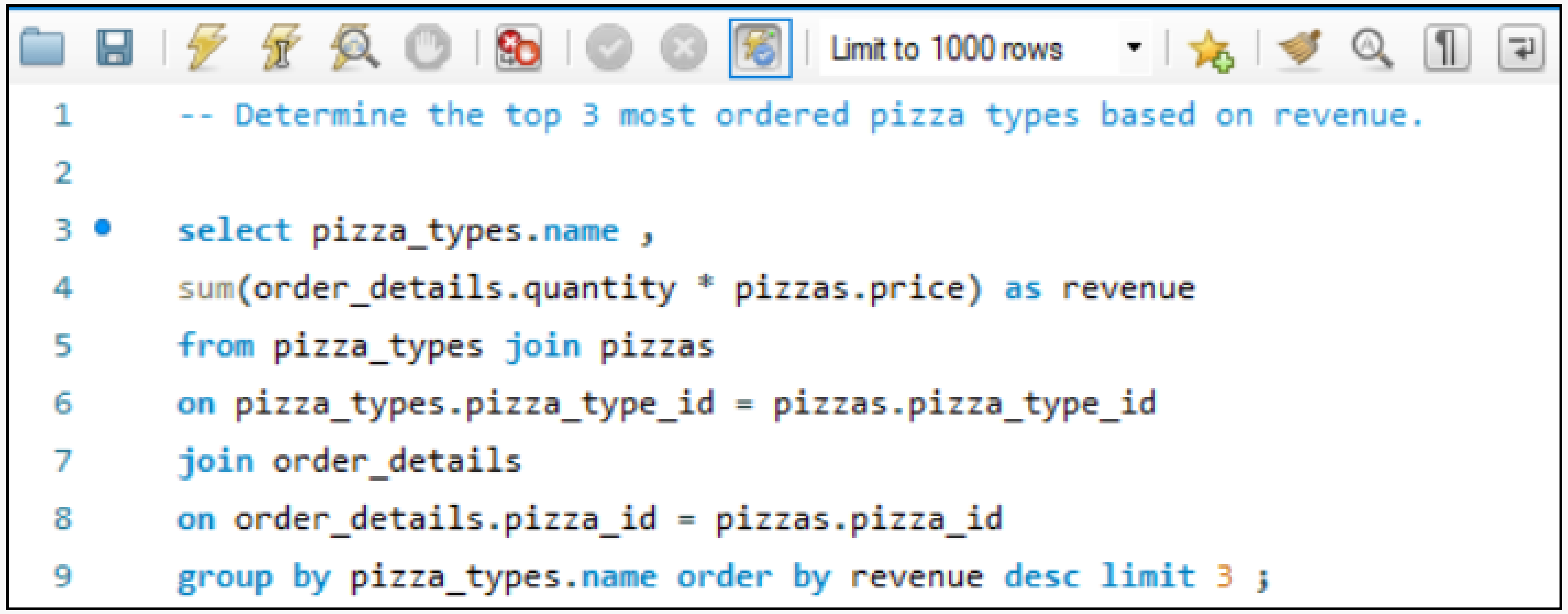


```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.

select round(avg(quantity),0)  as avg_pizza_order_per_day from
(select order1.order_date, sum(order_details.quantity) as quantity
from order1 join order_details
on order1.order_id = order_details.order_id
group by order1.order_date) as order_quantity;
```

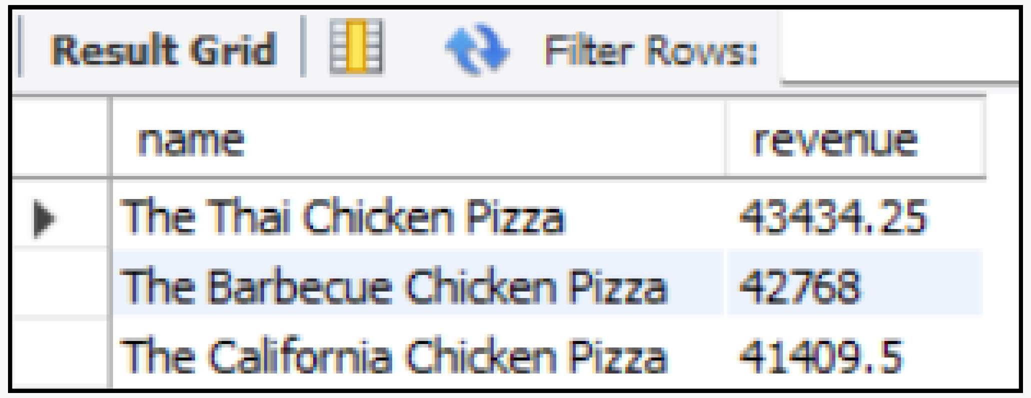


```
avg_pizza_order_per_day
138
```

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

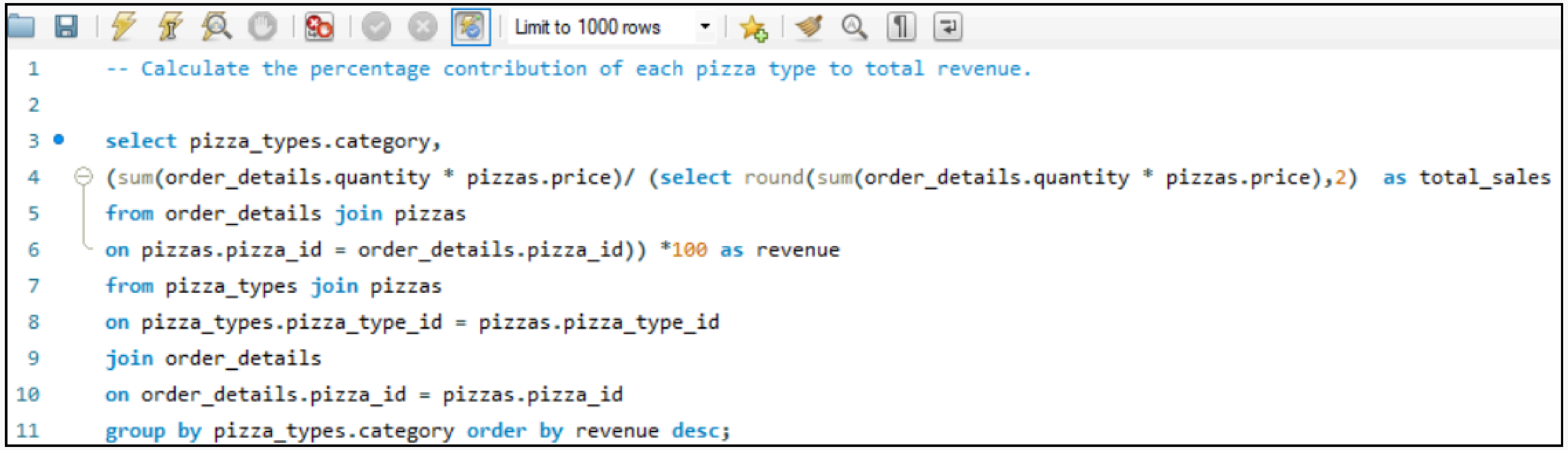


```sql
-- Determine the top 3 most ordered pizza types based on revenue.

select pizza_types.name ,
sum(order_details.quantity * pizzas.price) as revenue
from pizza_types join pizzas
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 revenue desc limit 3 ;
```



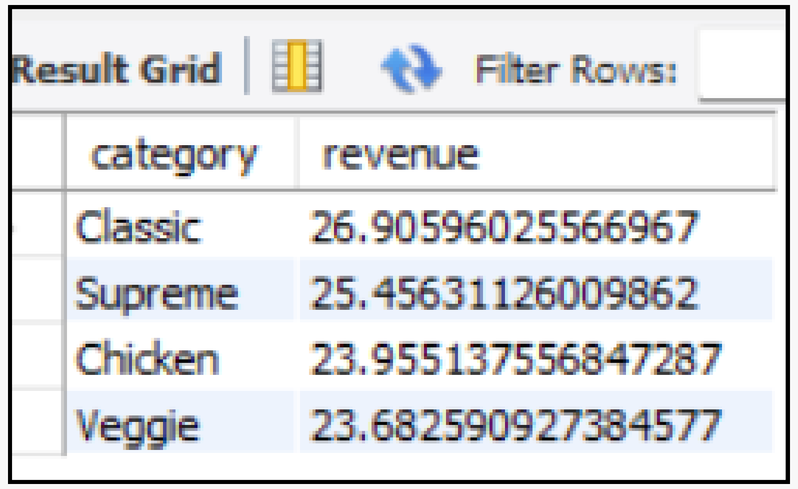
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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



Limit to 1000 rows

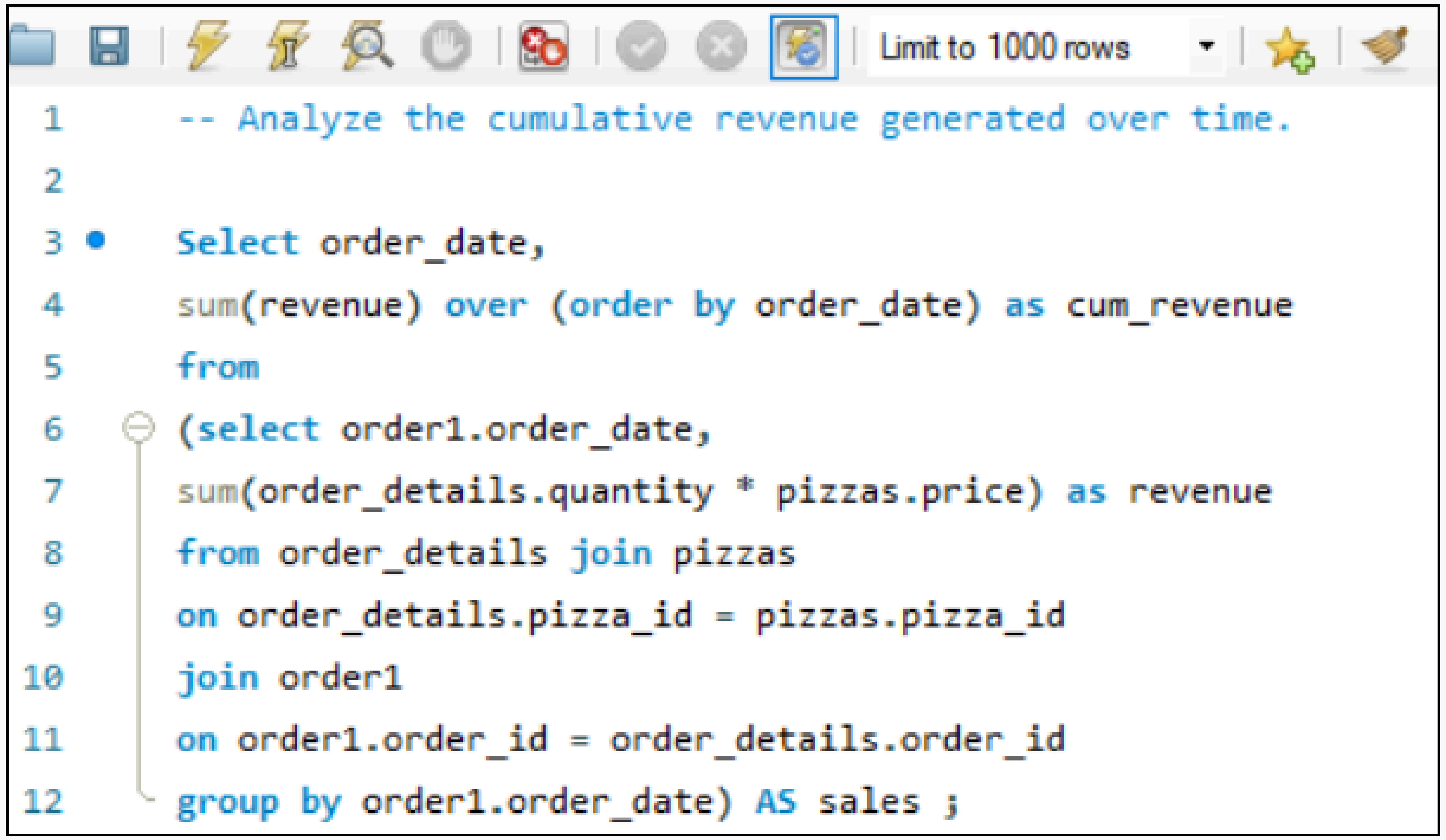
```
-- Calculate the percentage contribution of each pizza type to total revenue.

select pizza_types.category,
(sum(order_details.quantity * pizzas.price)/ (select round(sum(order_details.quantity * pizzas.price),2)  as total_sales
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id)) *100 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;
```



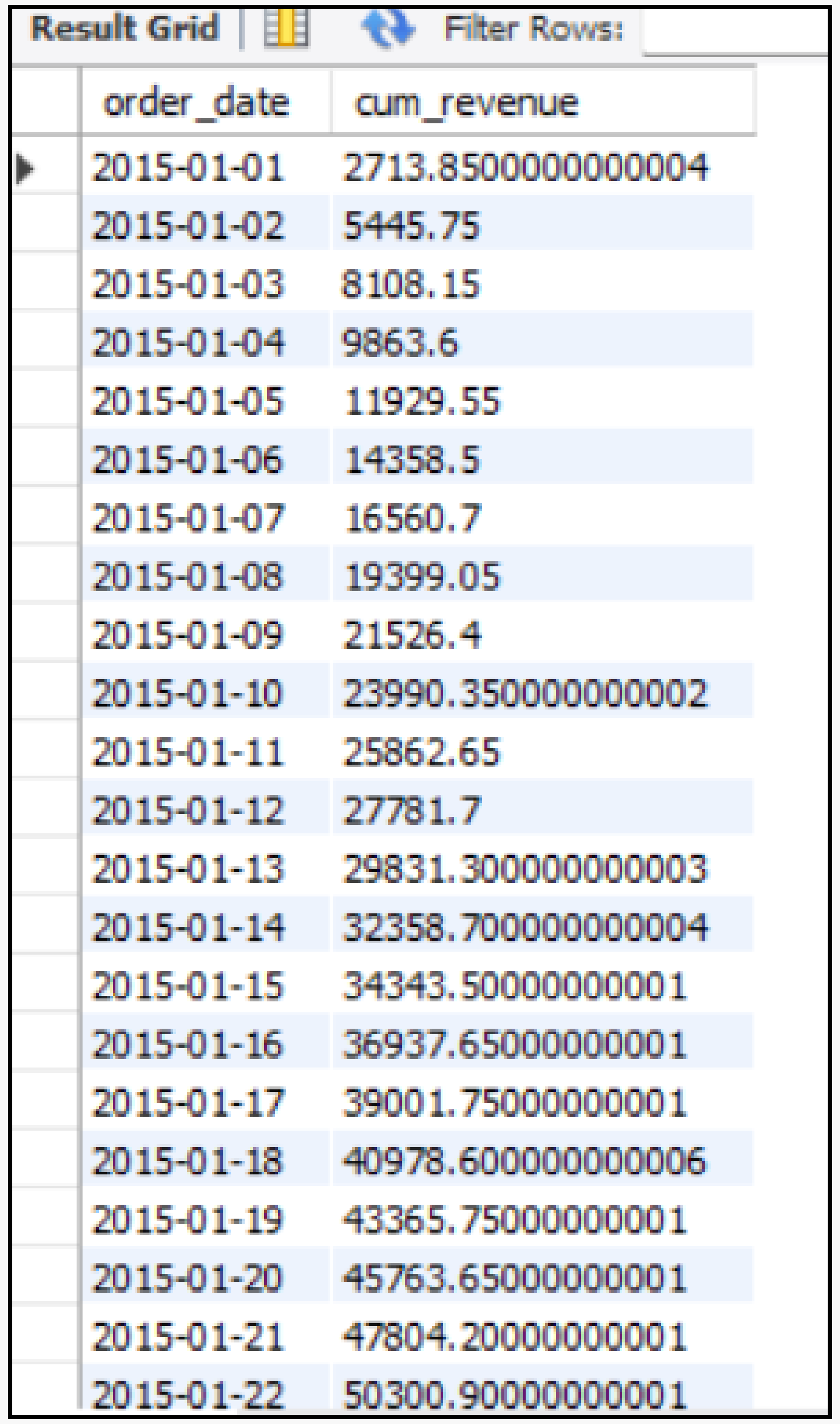
Result Grid | Filter Rows:

category	revenue
Classic	26.90596025566967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

Analyze the cumulative revenue generated over time.

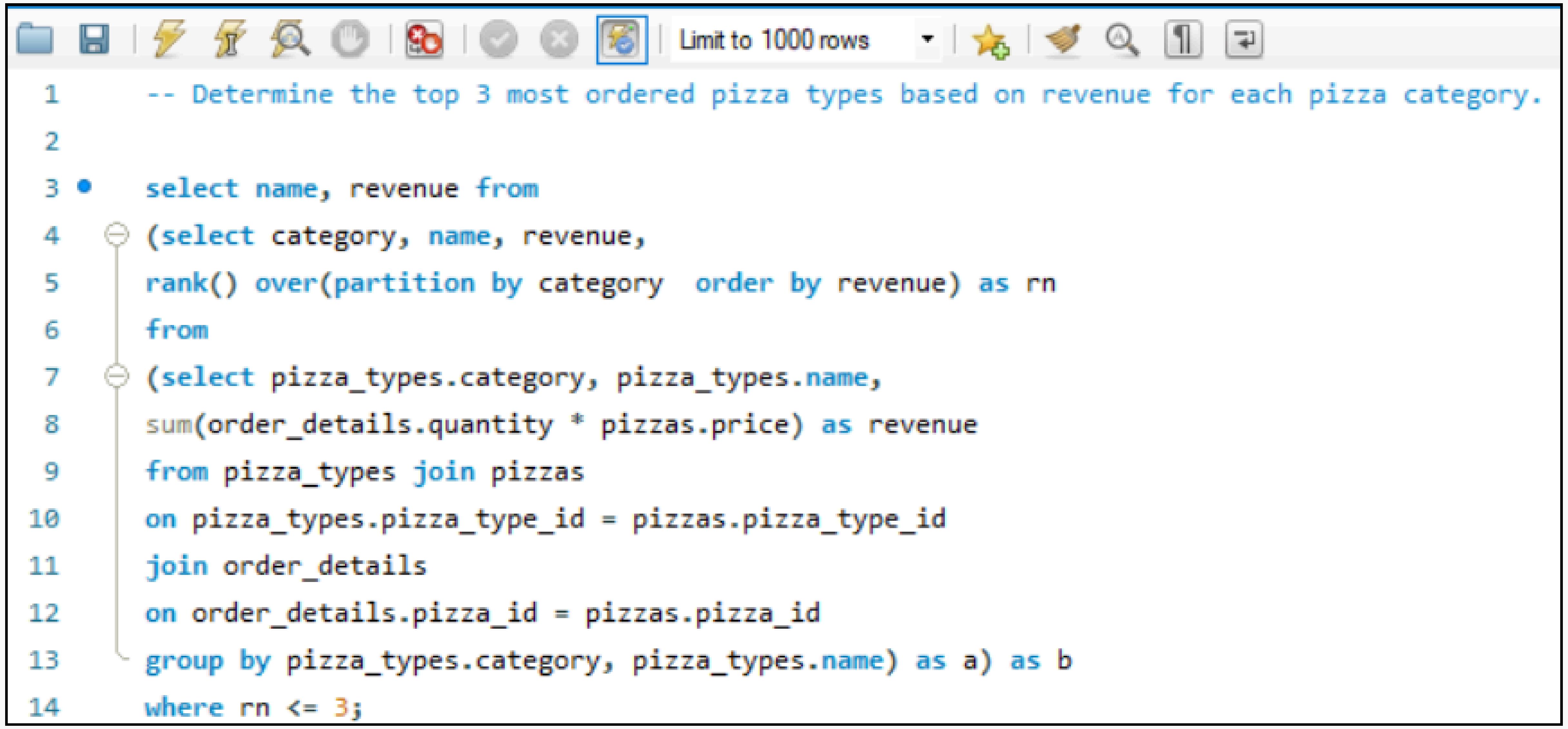


```sql
-- Analyze the cumulative revenue generated over time.

Select order_date,
sum(revenue) over (order by order_date) as cum_revenue
from
(select order1.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join order1
on order1.order_id = order_details.order_id
group by order1.order_date) AS sales ;
```

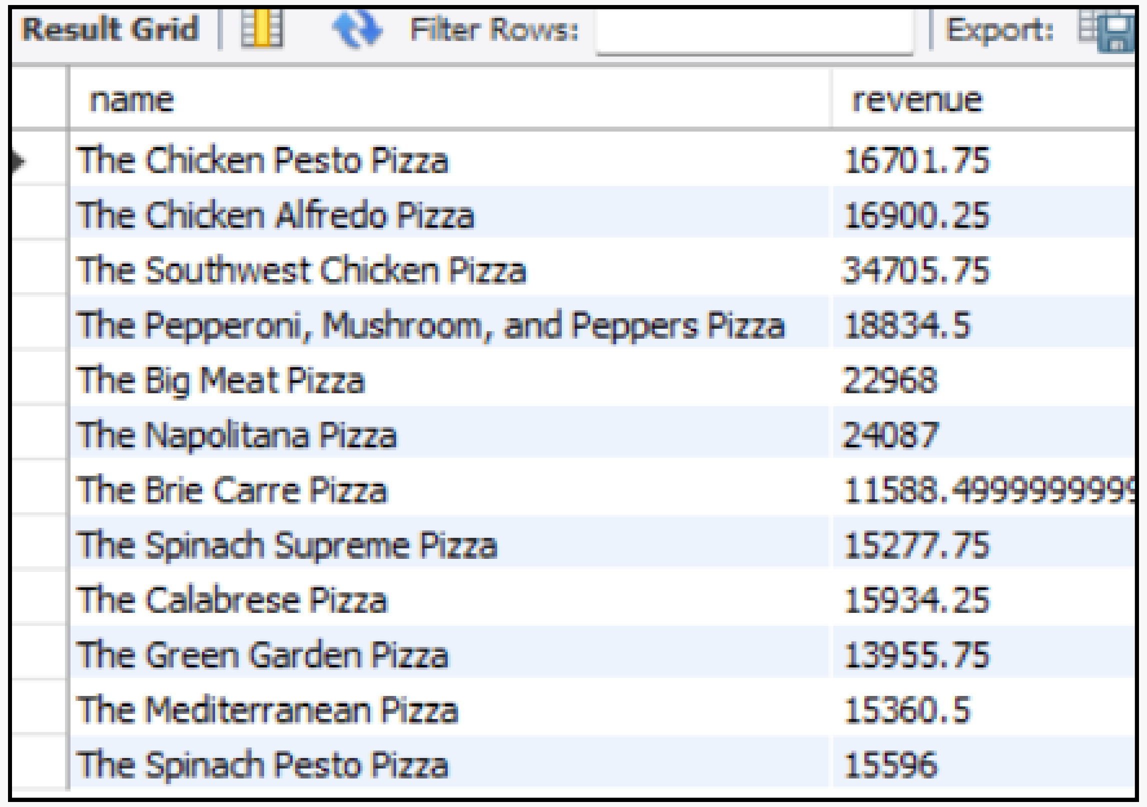


order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
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

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



```sql
-- 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) 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;
```



name	revenue
The Chicken Pesto Pizza	16701.75
The Chicken Alfredo Pizza	16900.25
The Southwest Chicken Pizza	34705.75
The Pepperoni, Mushroom, and Peppers Pizza	18834.5
The Big Meat Pizza	22968
The Napolitana Pizza	24087
The Brie Carre Pizza	11588.4999999999
The Spinach Supreme Pizza	15277.75
The Calabrese Pizza	15934.25
The Green Garden Pizza	13955.75
The Mediterranean Pizza	15360.5
The Spinach Pesto Pizza	15596

Thank you