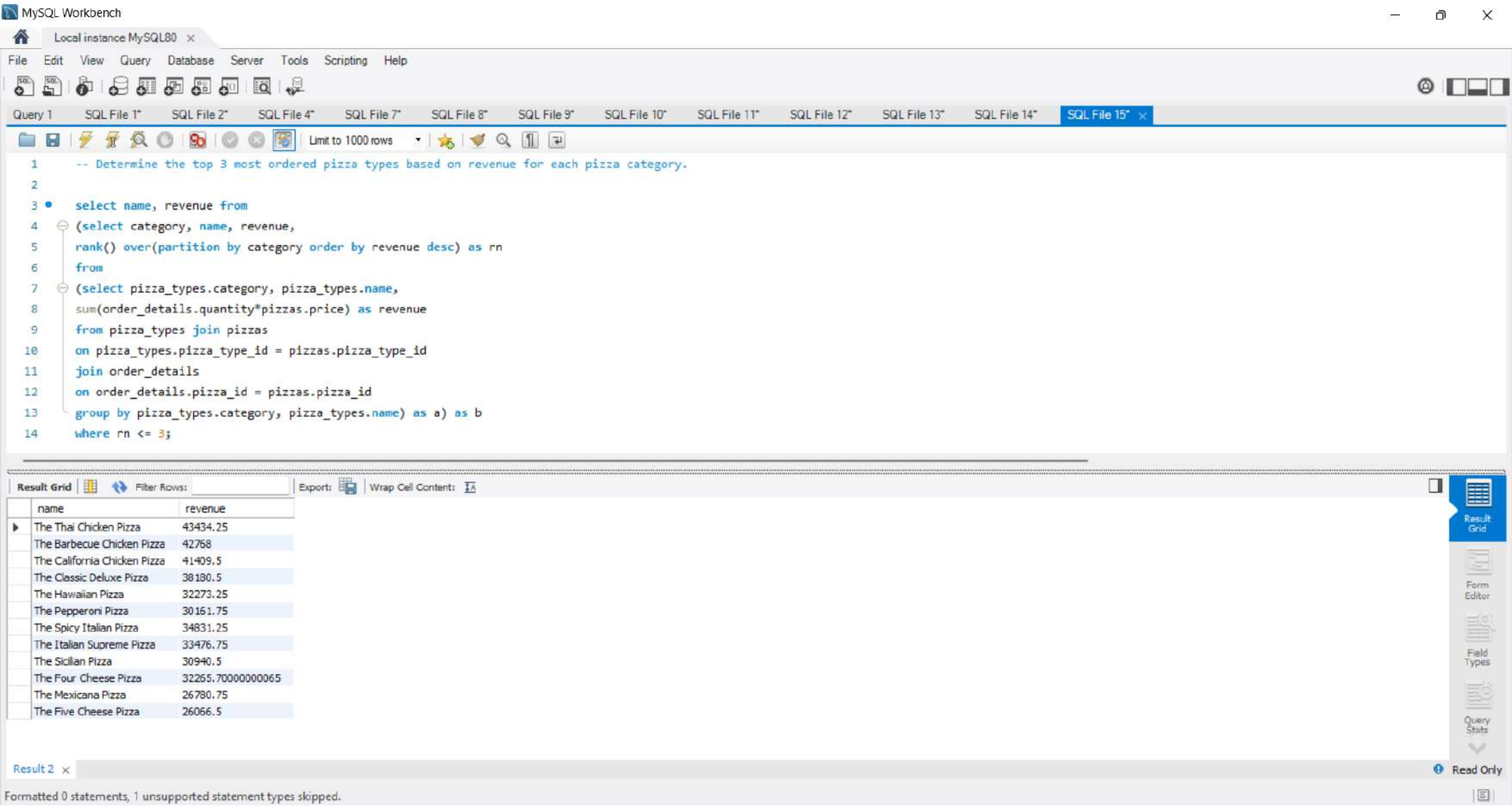
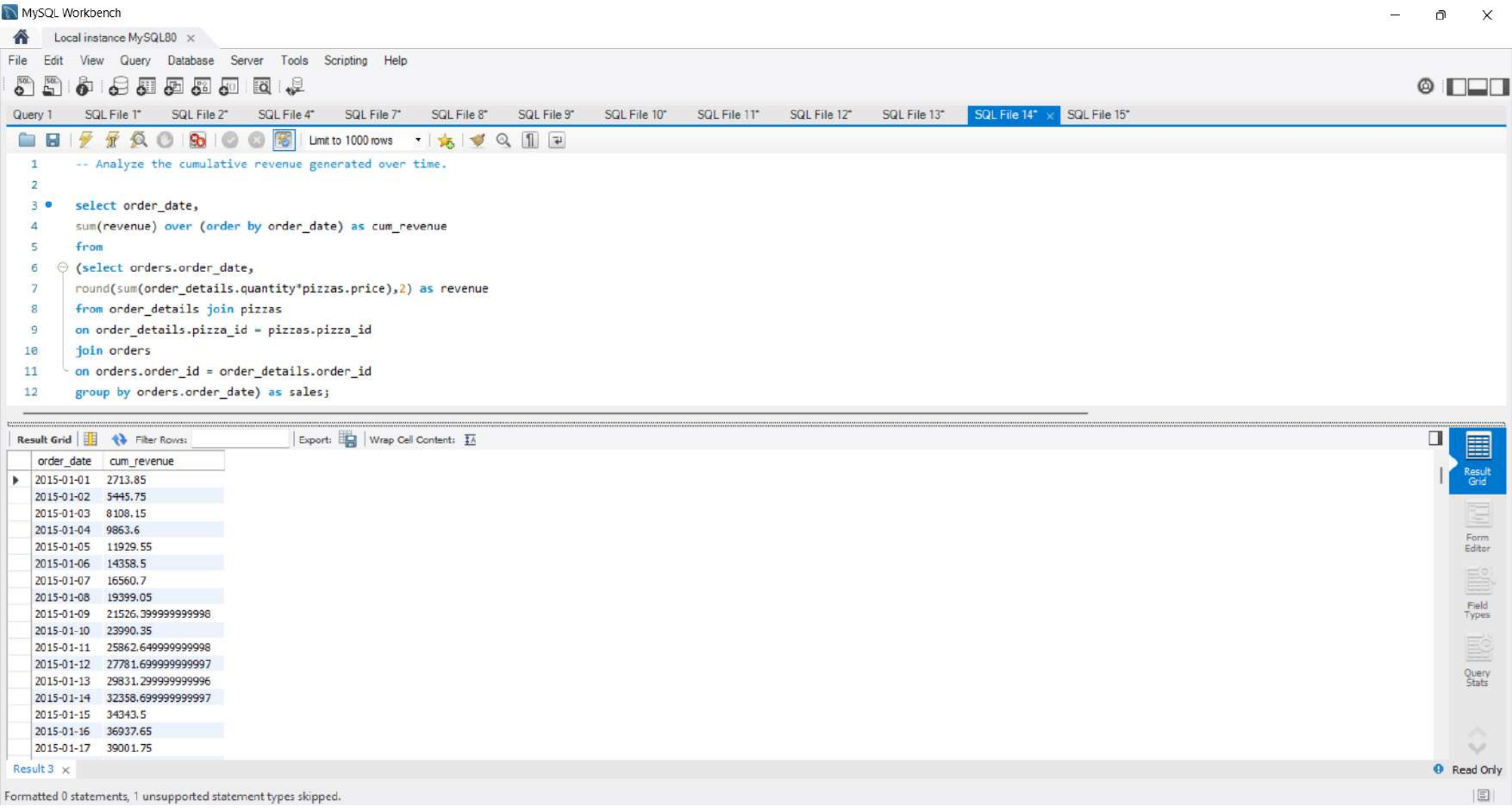
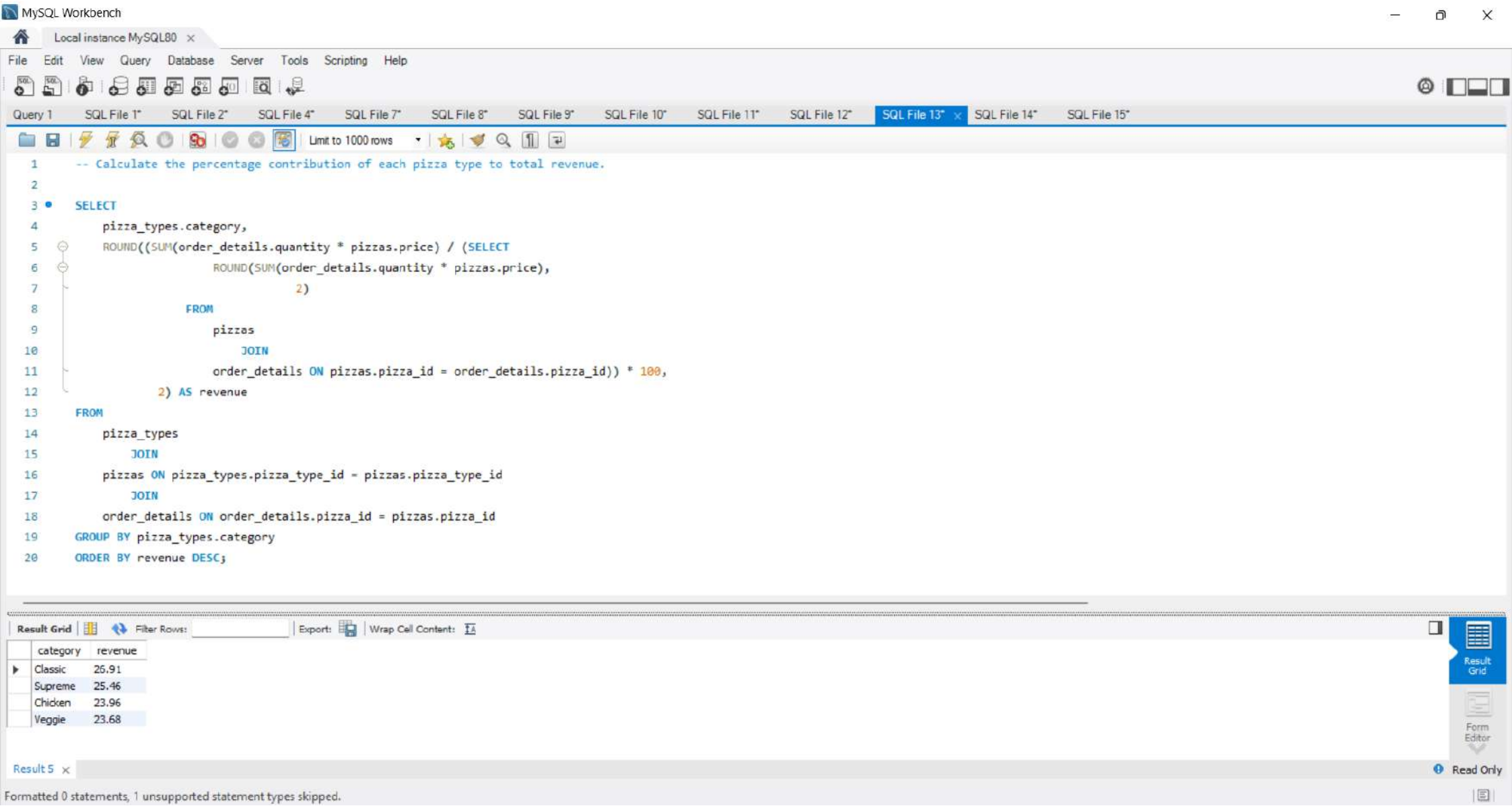
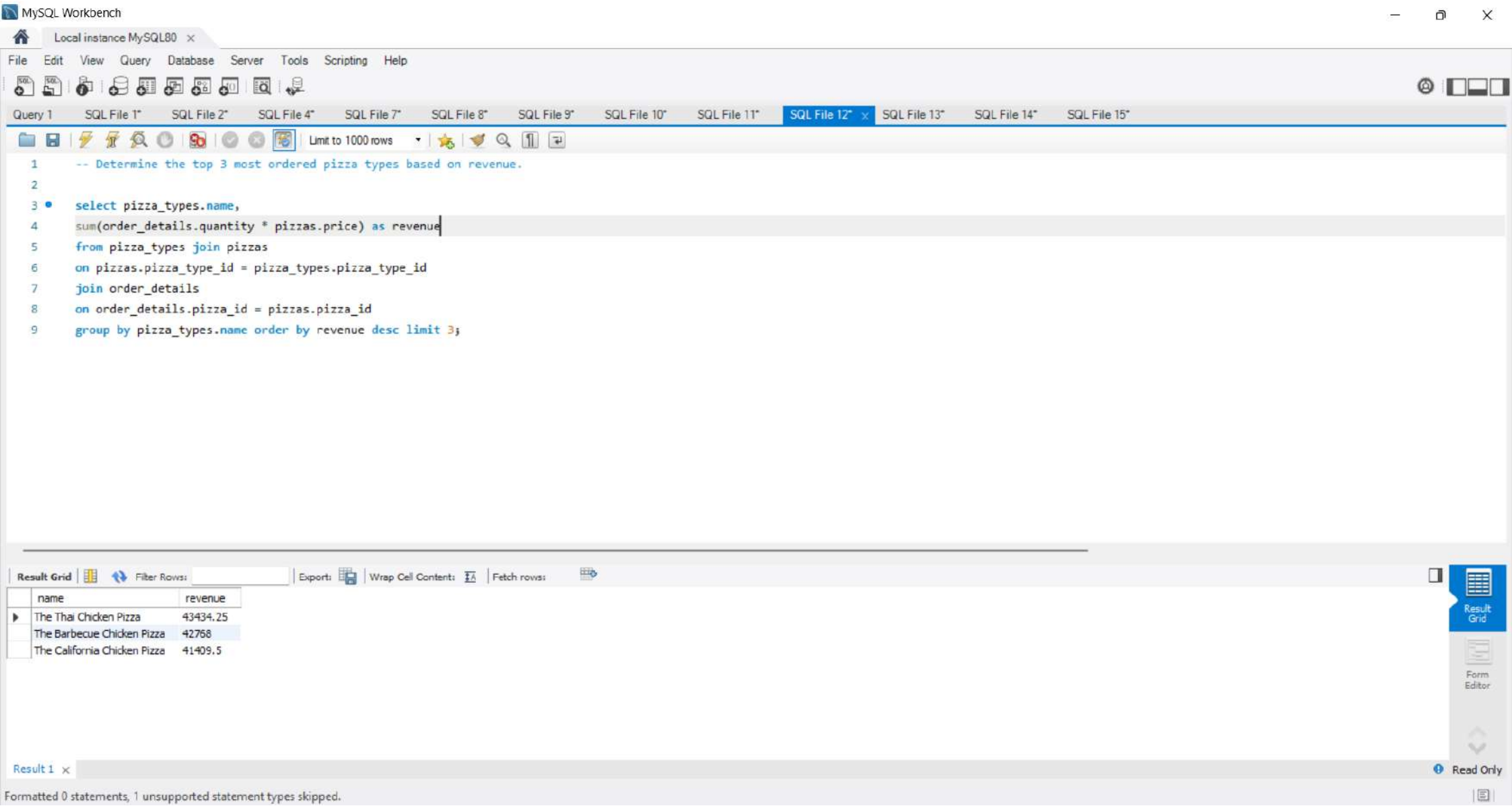
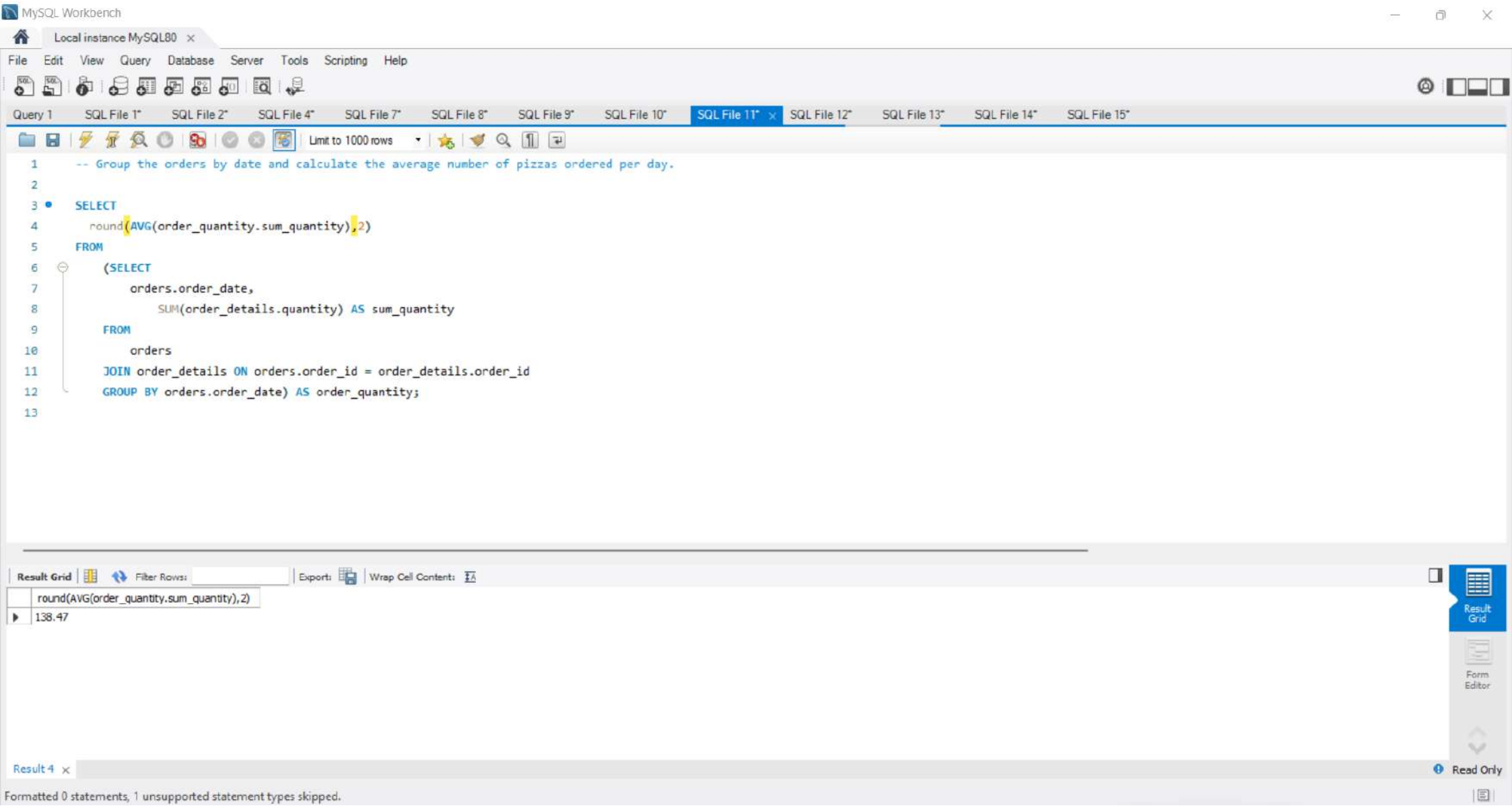
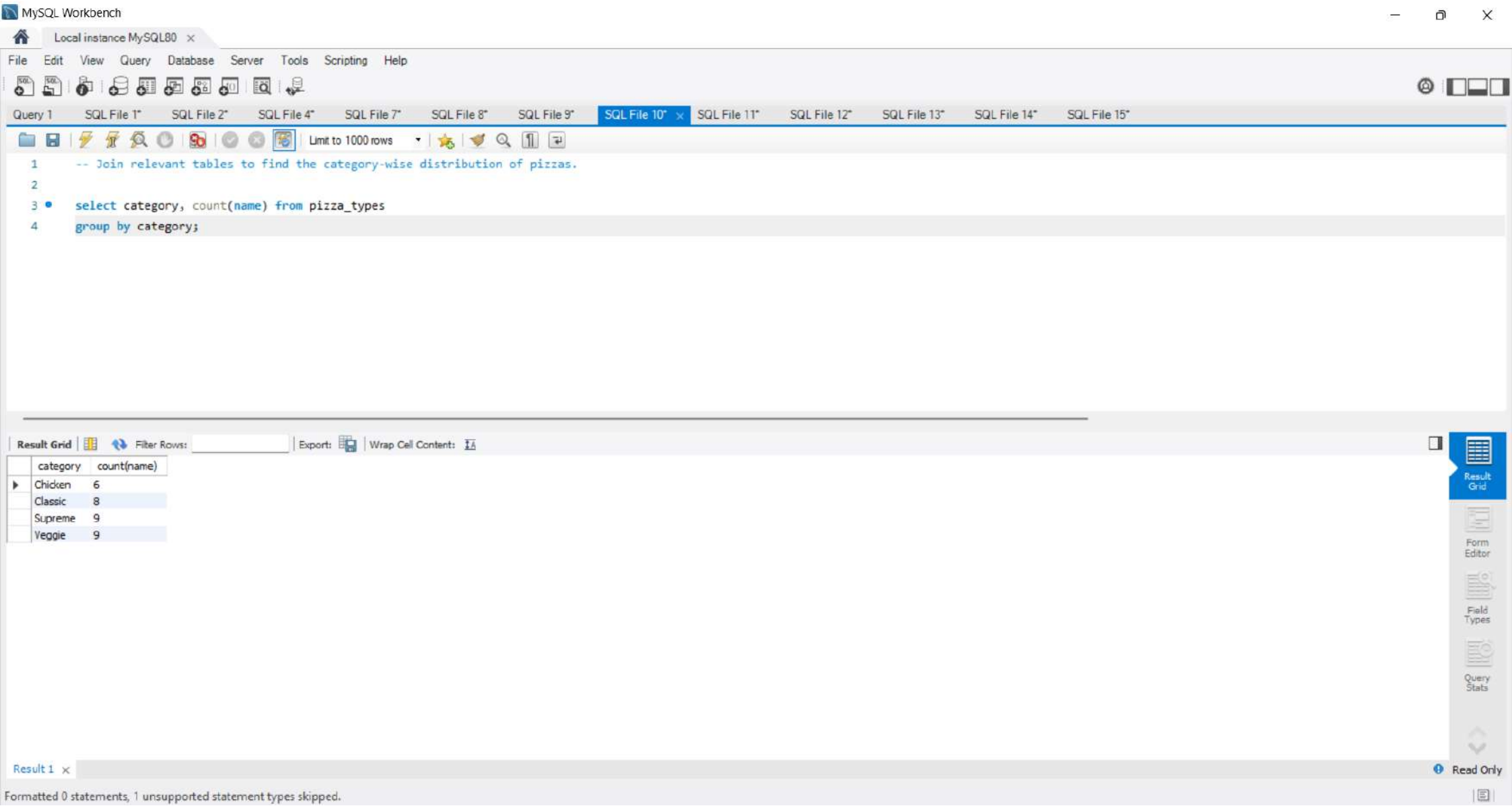

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Query 1 SQL File 1* SQL File 2* SQL File 4* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15*
Limit to 1000 rows
-- Determine the top 3 most 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 desc) 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;
Result Grid Filter Rows: Export: Wrap Cell Content:
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 32265.70000000065
The Mexicana Pizza 26780.75
The Five Cheese Pizza 26066.5
Result Grid
Form Editor
Field Types
Query Stats
Result 2
Read Only
Formatted 0 statements, 1 unsupported statement types skipped.

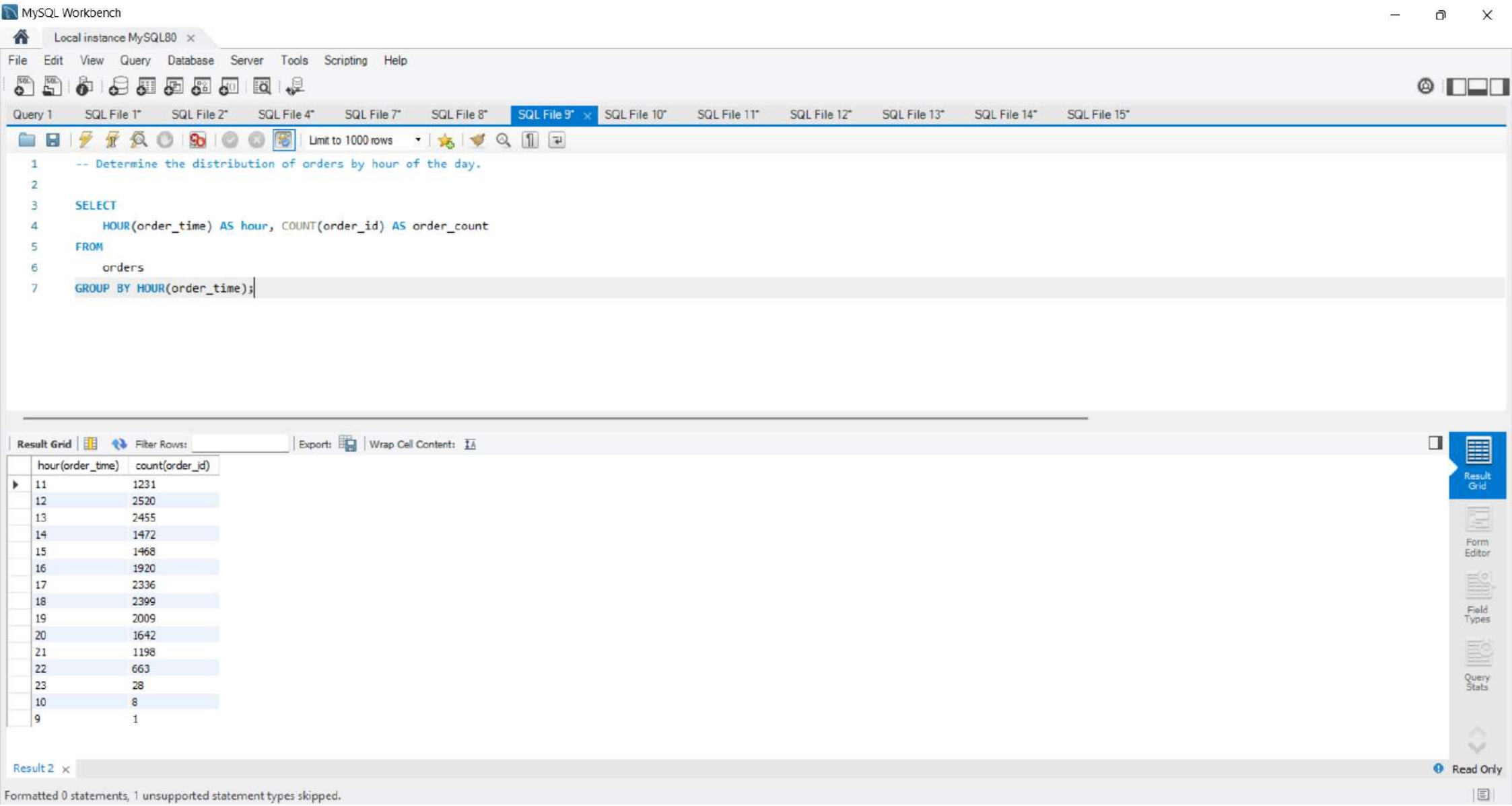

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Query 1 SQL File 1* SQL File 2* SQL File 4* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15*
Limit to 1000 rows
-- Analyze the cumulative revenue generated over time.
select order_date,
sum(revenue) over (order by order_date) as cum_revenue
from
(select orders.order_date,
round(sum(order_details.quantity*pizzas.price),2) 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;
Result Grid Filter Rows: Export: Wrap Cell Content:
order_date cum_revenue
2015-01-01 2713.85
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 21526.399999999998
2015-01-10 23990.35
2015-01-11 25862.649999999998
2015-01-12 27781.699999999997
2015-01-13 29831.299999999996
2015-01-14 32358.699999999997
2015-01-15 34343.5
2015-01-16 36937.65
2015-01-17 39001.75
Result 3
Result Grid
Form Editor
Field Types
Query Stats
Read Only
Formatted 0 statements, 1 unsupported statement types skipped.

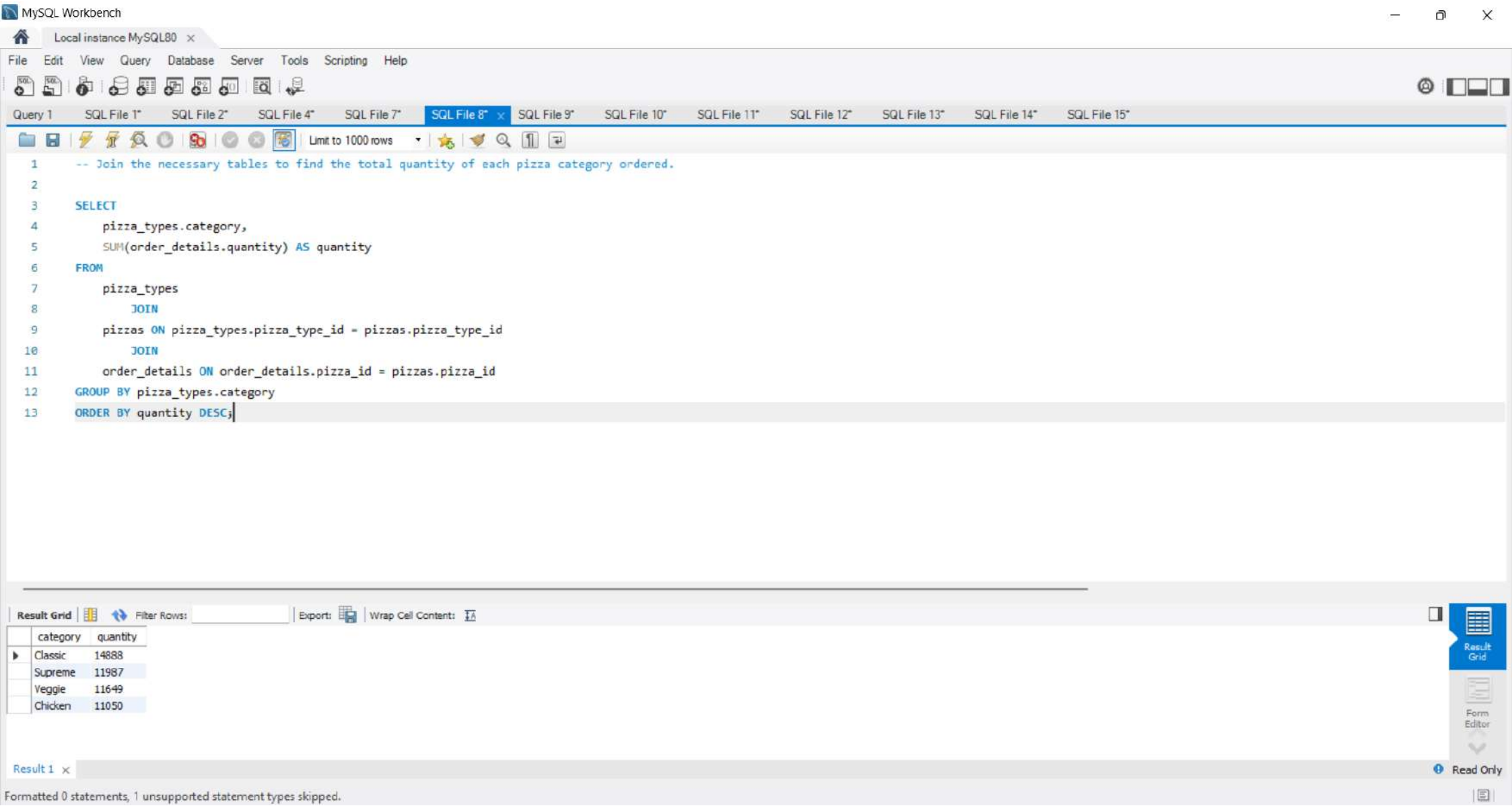

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Query 1 SQL File 1* SQL File 2* SQL File 4* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15*
Limit to 1000 rows
-- Calculate the percentage 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
pizzas
JOIN
order_details ON pizzas.pizza_id = order_details.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;
Result Grid Filter Rows: Export: Wrap Cell Content:
category revenue
Classic 26.91
Supreme 25.46
Chicken 23.96
Veggie 23.68
Result 5
Read Only
Formatted 0 statements, 1 unsupported statement types skipped.

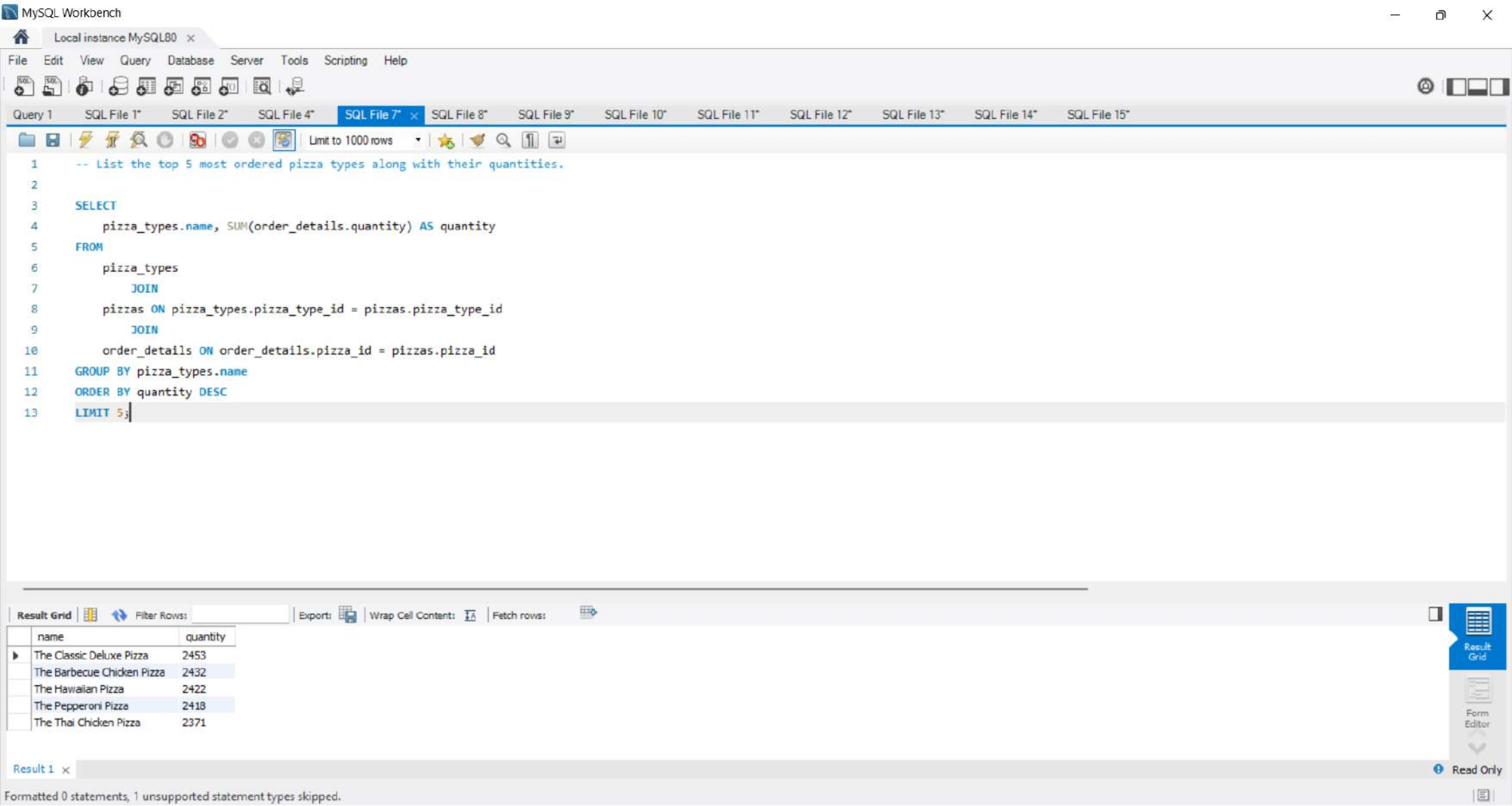

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Query 1 SQL File 1* SQL File 2* SQL File 4* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15*
Limit to 1000 rows
-- Determine the top 3 most ordered pizza types based on revenue.
select pizza_types.name,
sum(order_details.quantity * pizzas.price) as revenue
from pizza_types join pizzas
on 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 revenue desc limit 3;
Result Grid Filter Rows: Export: Wrap Cell Content: Fetch rows:
name revenue
The Thai Chicken Pizza 43434.25
The Barbecue Chicken Pizza 42768
The California Chicken Pizza 41409.5
Result Grid
Form Editor
Result 1
Read Only
Formatted 0 statements, 1 unsupported statement types skipped.

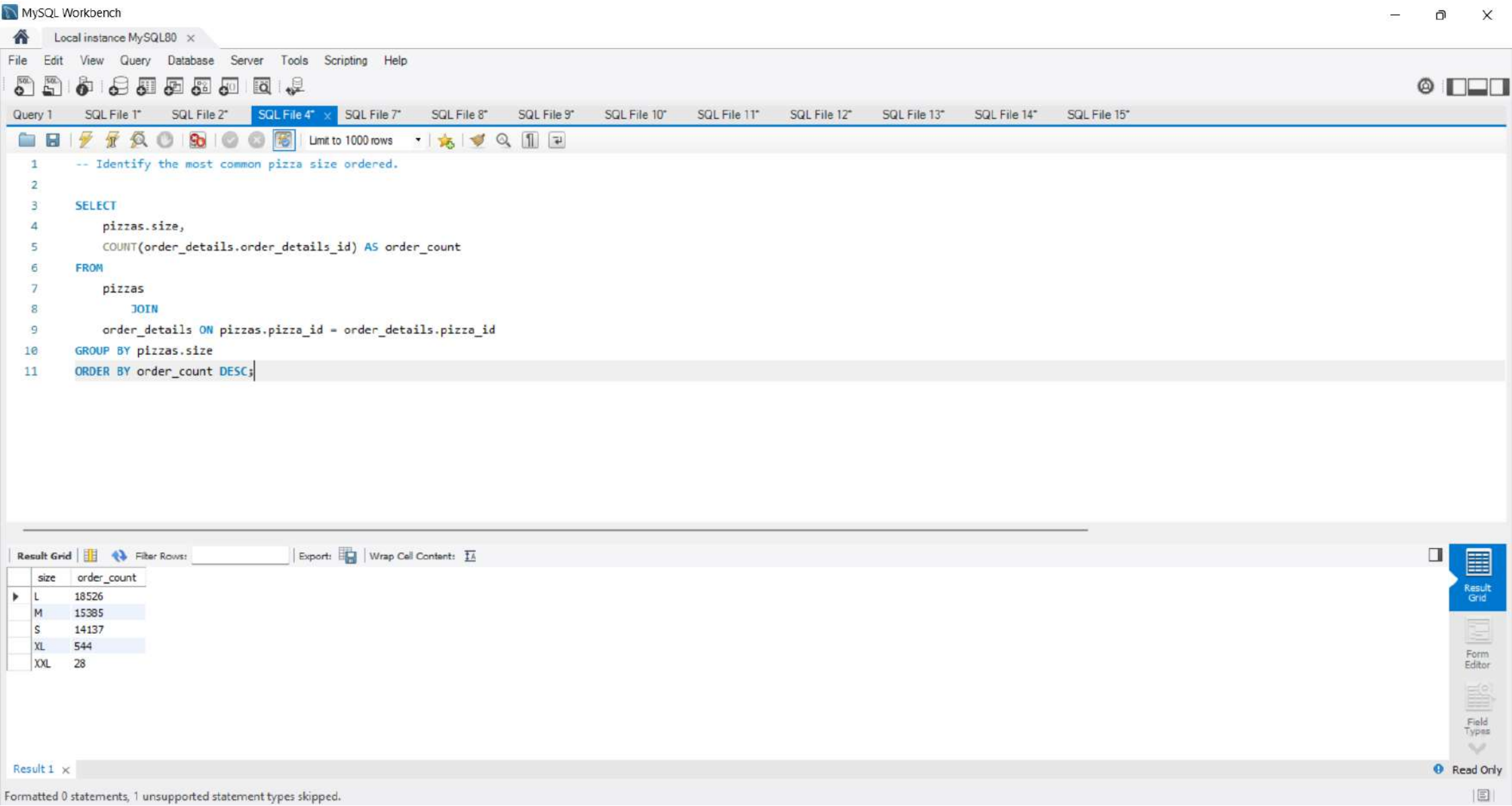

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Query 1 SQL File 1* SQL File 2* SQL File 4* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15*
Limit to 1000 rows
-- Group the orders by date and calculate the average number of pizzas ordered per day.
SELECT
round(AVG(order_quantity.sum_quantity),2)
FROM
(SELECT
orders.order_date,
SUM(order_details.quantity) AS sum_quantity
FROM
orders
JOIN order_details ON orders.order_id = order_details.order_id
GROUP BY orders.order_date) AS order_quantity;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
round(AVG(order_quantity.sum_quantity),2)
138.47
Result Grid
Form Editor
Result 4
Read Only
Formatted 0 statements, 1 unsupported statement types skipped.

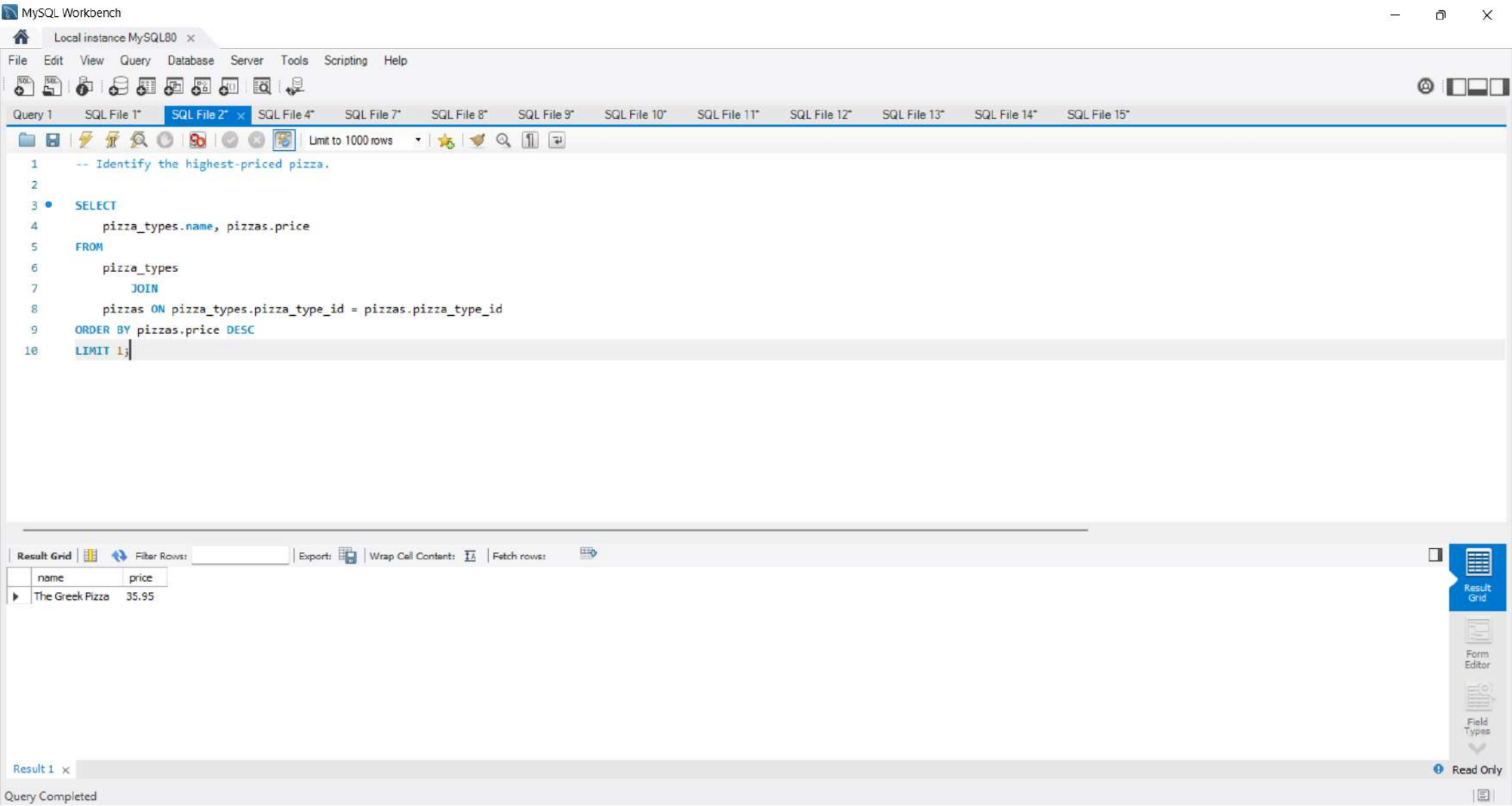

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Query 1 SQL File 1* SQL File 2* SQL File 4* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15*
Limit to 1000 rows
1 -- Join relevant tables to find the category-wise distribution of pizzas.
2
3 select category, count(name) from pizza_types
4 group by category;
Result Grid Filter Rows: Export: Wrap Cell Content:
category count(name)
Chicken 6
Classic 8
Supreme 9
Veggie 9
Result Grid
Form Editor
Field Types
Query Stats
Result 1
Read Only
Formatted 0 statements, 1 unsupported statement types skipped.

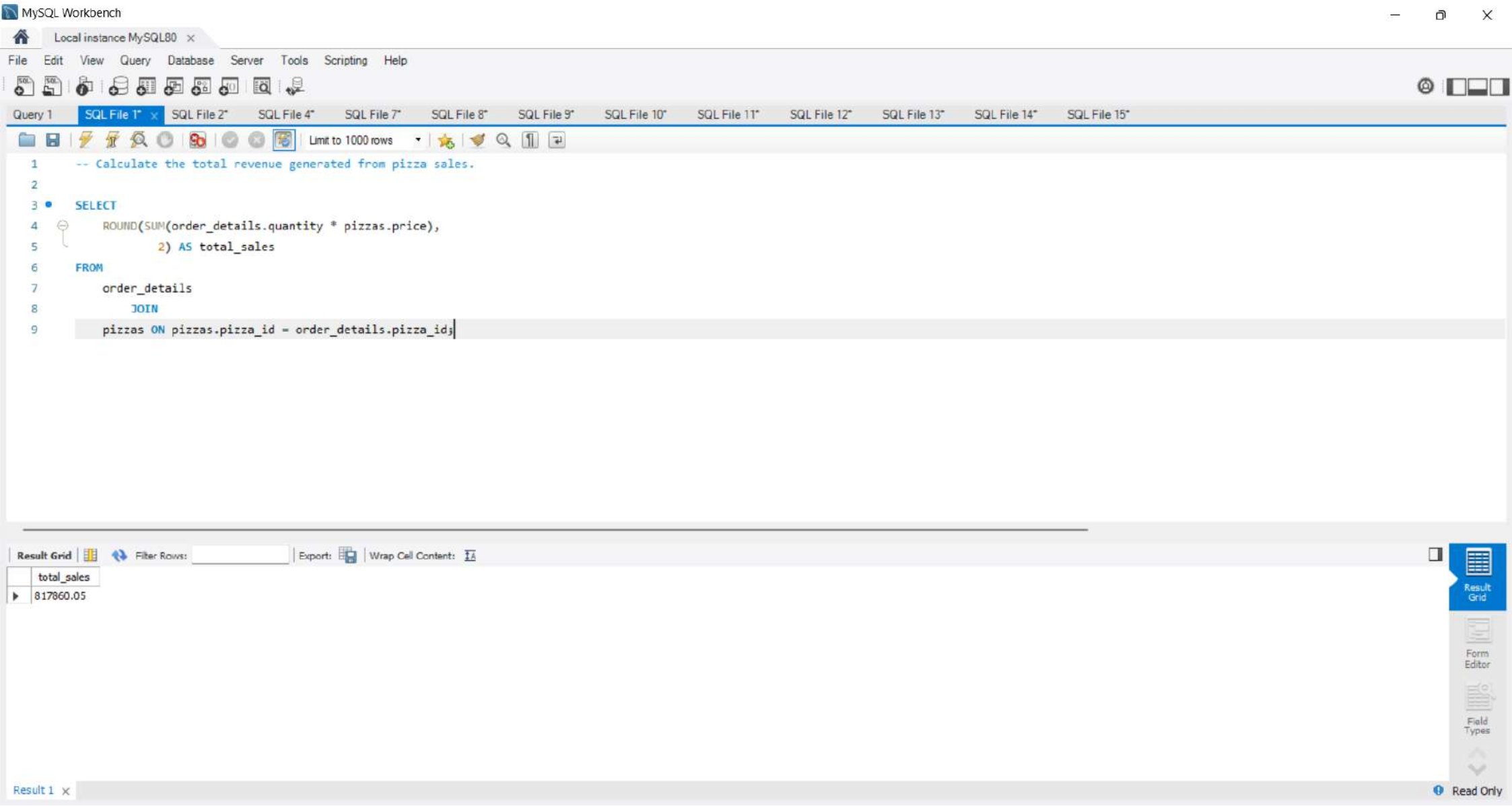

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Query 1 SQL File 1* SQL File 2* SQL File 4* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15*
Limit to 1000 rows
1 -- Determine the distribution of orders by hour of the day.
2
3 SELECT
4 HOUR(order_time) AS hour, COUNT(order_id) AS order_count
5 FROM
6 orders
7 GROUP BY HOUR(order_time);
Result Grid Filter Rows: Export: Wrap Cell Content:
hour(order_time) count(order_id)
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
Result 2
Result Grid
Form Editor
Field Types
Query Stats
Read Only
Formatted 0 statements, 1 unsupported statement types skipped.

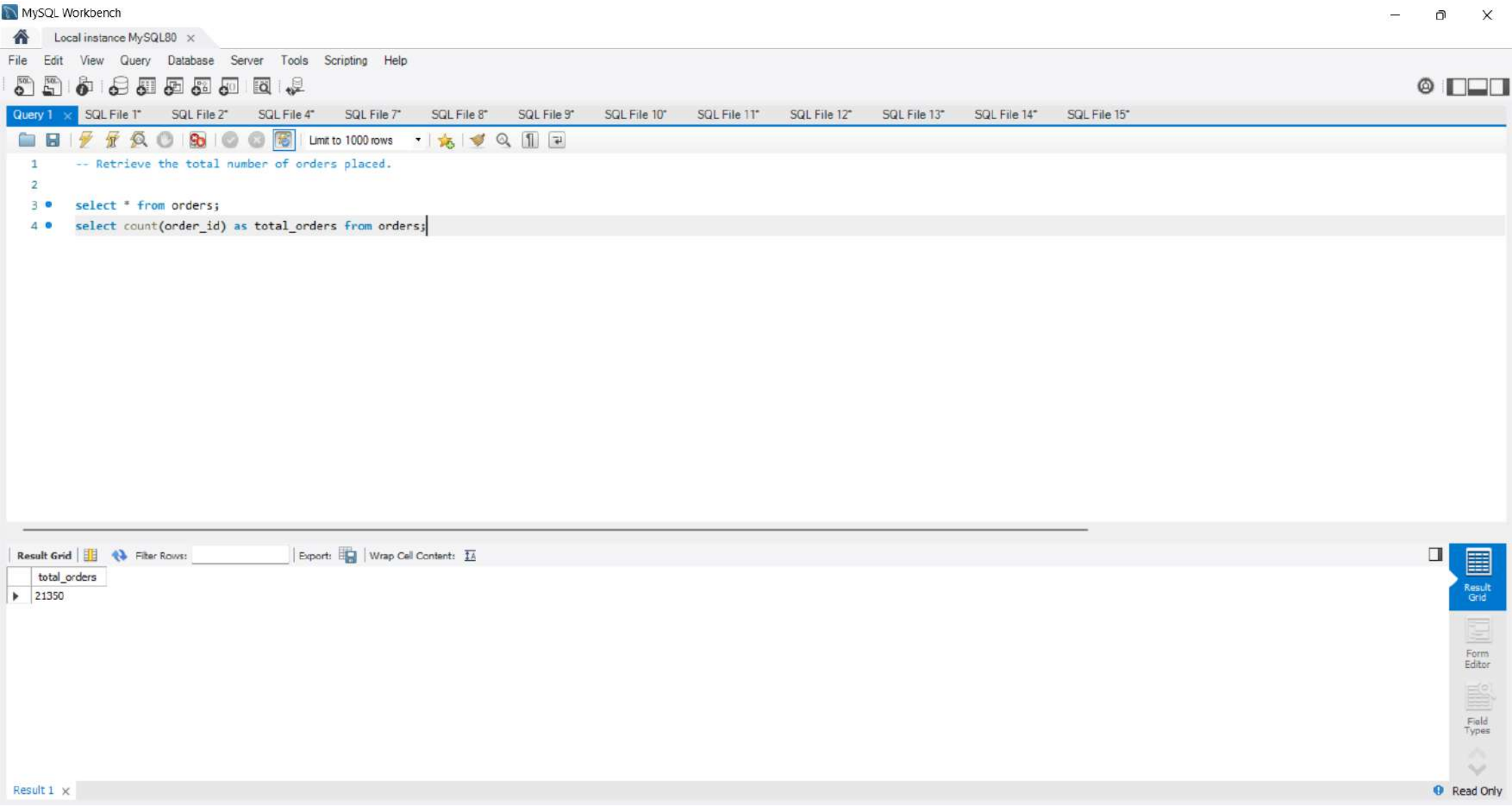

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Query 1 SQL File 1* SQL File 2* SQL File 4* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15*
Limit to 1000 rows
1 -- Join the necessary tables to find the total quantity of each pizza category ordered.
2
3 SELECT
4 pizza_types.category,
5 SUM(order_details.quantity) AS quantity
6 FROM
7 pizza_types
8 JOIN
9 pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10 JOIN
11 order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.category
13 ORDER BY quantity DESC;
Result Grid Filter Rows: Export: Wrap Cell Content:
category quantity
Classic 14888
Supreme 11987
Veggie 11649
Chicken 11050
Result Grid
Form Editor
Result 1
Read Only
Formatted 0 statements, 1 unsupported statement types skipped.


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Query 1 SQL File 1* SQL File 2* SQL File 4* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15*
Limit to 1000 rows
1 -- List the top 5 most ordered pizza types along with their quantities.
2
3 SELECT
4 pizza_types.name, SUM(order_details.quantity) AS quantity
5 FROM
6 pizza_types
7 JOIN
8 pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9 JOIN
10 order_details ON order_details.pizza_id = pizzas.pizza_id
11 GROUP BY pizza_types.name
12 ORDER BY quantity DESC
13 LIMIT 5;
Result Grid Filter Rows: Export: Wrap Cell Content: Fetch 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
Result 1
Result Grid
Form Editor
Read Only
Formatted 0 statements, 1 unsupported statement types skipped.


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Query 1 SQL File 1* SQL File 2* SQL File 4* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15*
Limit to 1000 rows
-- 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_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
Result Grid Filter Rows: Export: Wrap Cell Content:
size order_count
L 18526
M 15385
S 14137
XL 544
XXL 28
Result Grid
Form Editor
Field Types
Result 1
Read Only
Formatted 0 statements, 1 unsupported statement types skipped.


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Query 1 SQL File 1* SQL File 2* SQL File 4* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15*
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: Export: Wrap Cell Content: Fetch rows:
name price
The Greek Pizza 35.95
Result Grid
Form Editor
Field Types
Result 1
Read Only
Query Completed


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Query 1 SQL File 1* SQL File 2* SQL File 4* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15*
Limit to 1000 rows
1 -- Calculate the total revenue generated from pizza sales.
2
3 SELECT
4 ROUND(SUM(order_details.quantity * pizzas.price),
5 2) AS total_sales
6 FROM
7 order_details
8 JOIN
9 pizzas ON pizzas.pizza_id = order_details.pizza_id;
Result Grid Filter Rows: Export: Wrap Cell Content:
total_sales
817860.05
Result Grid
Form Editor
Field Types
Result 1
Read Only


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Query 1 SQL File 1* SQL File 2* SQL File 4* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15*
Limit to 1000 rows
1 -- Retrieve the total number of orders placed.
2
3 select * from orders;
4 select count(order_id) as total_orders from orders;
Result Grid Filter Rows: Export: Wrap Cell Content:
total_orders
21350
Result Grid
Form Editor
Field Types
Result 1
Read Only