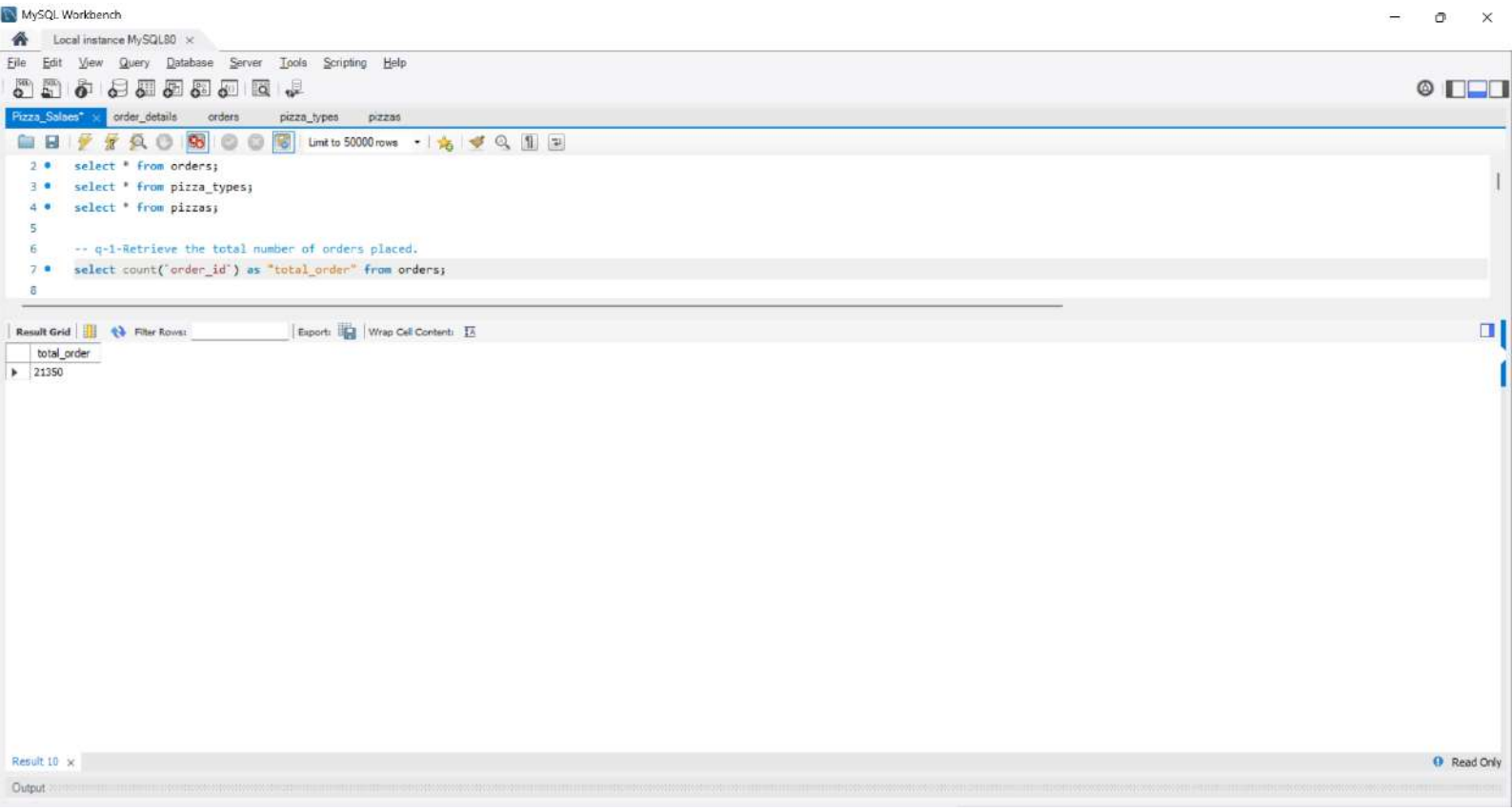
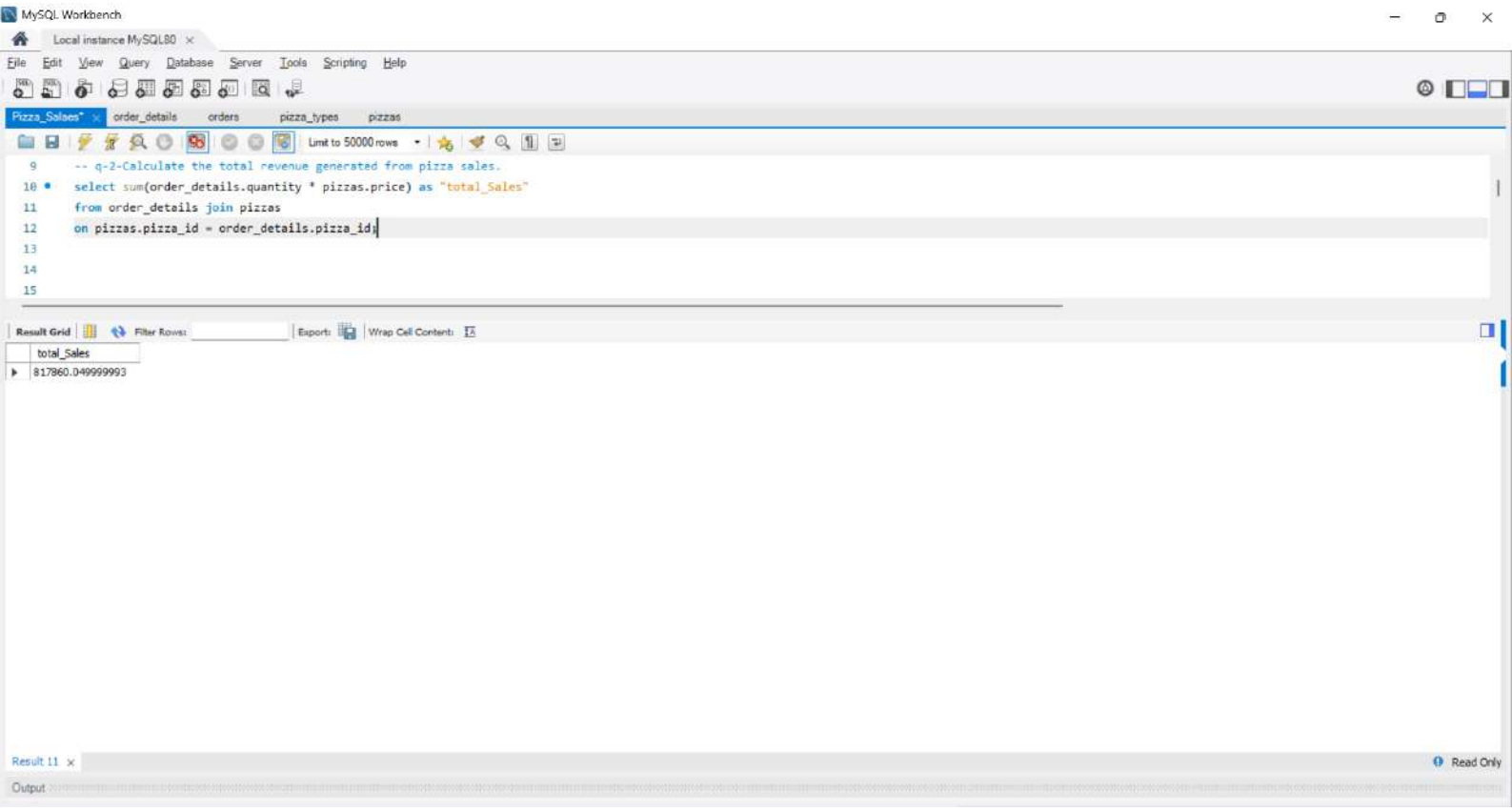
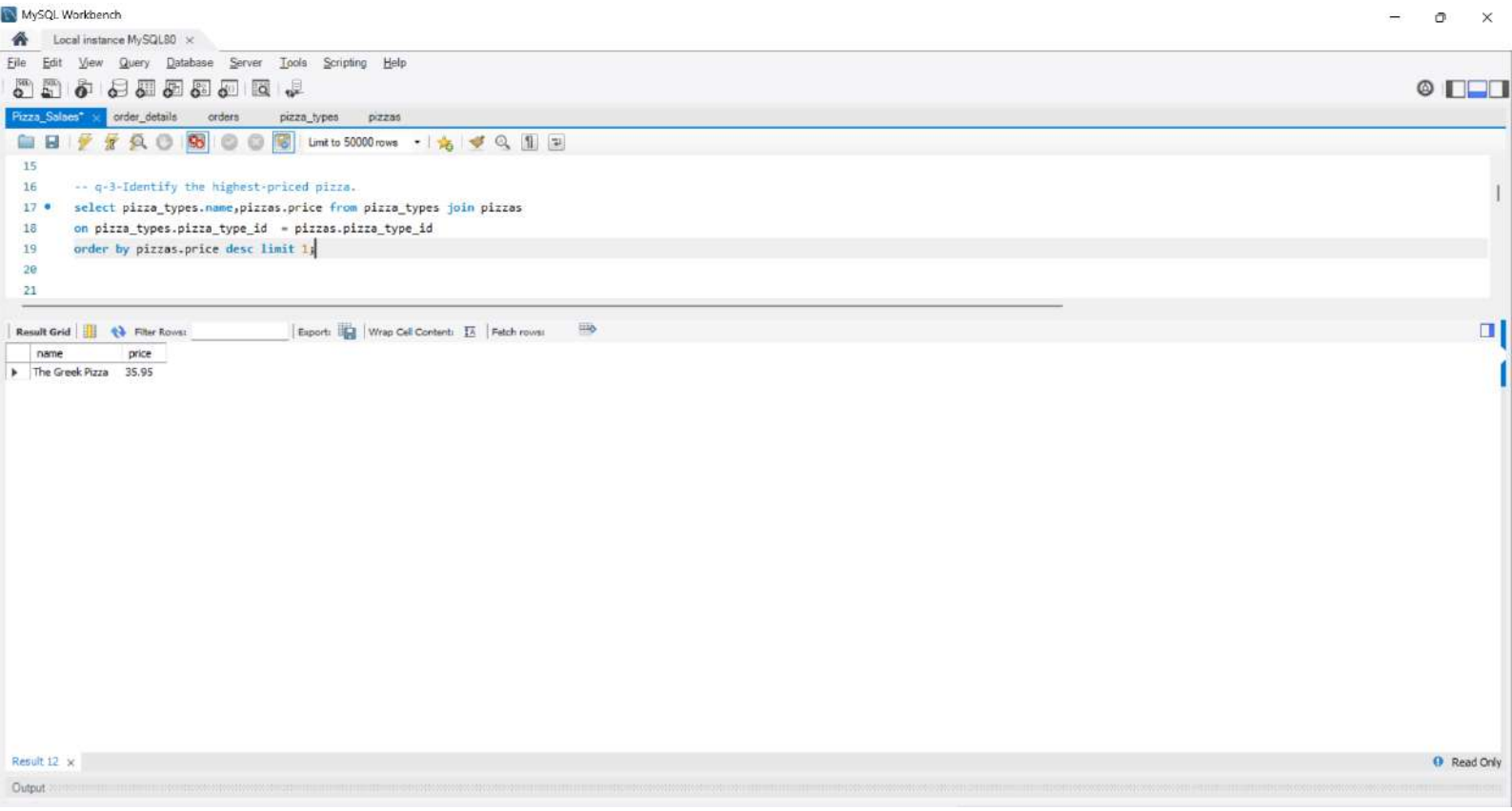
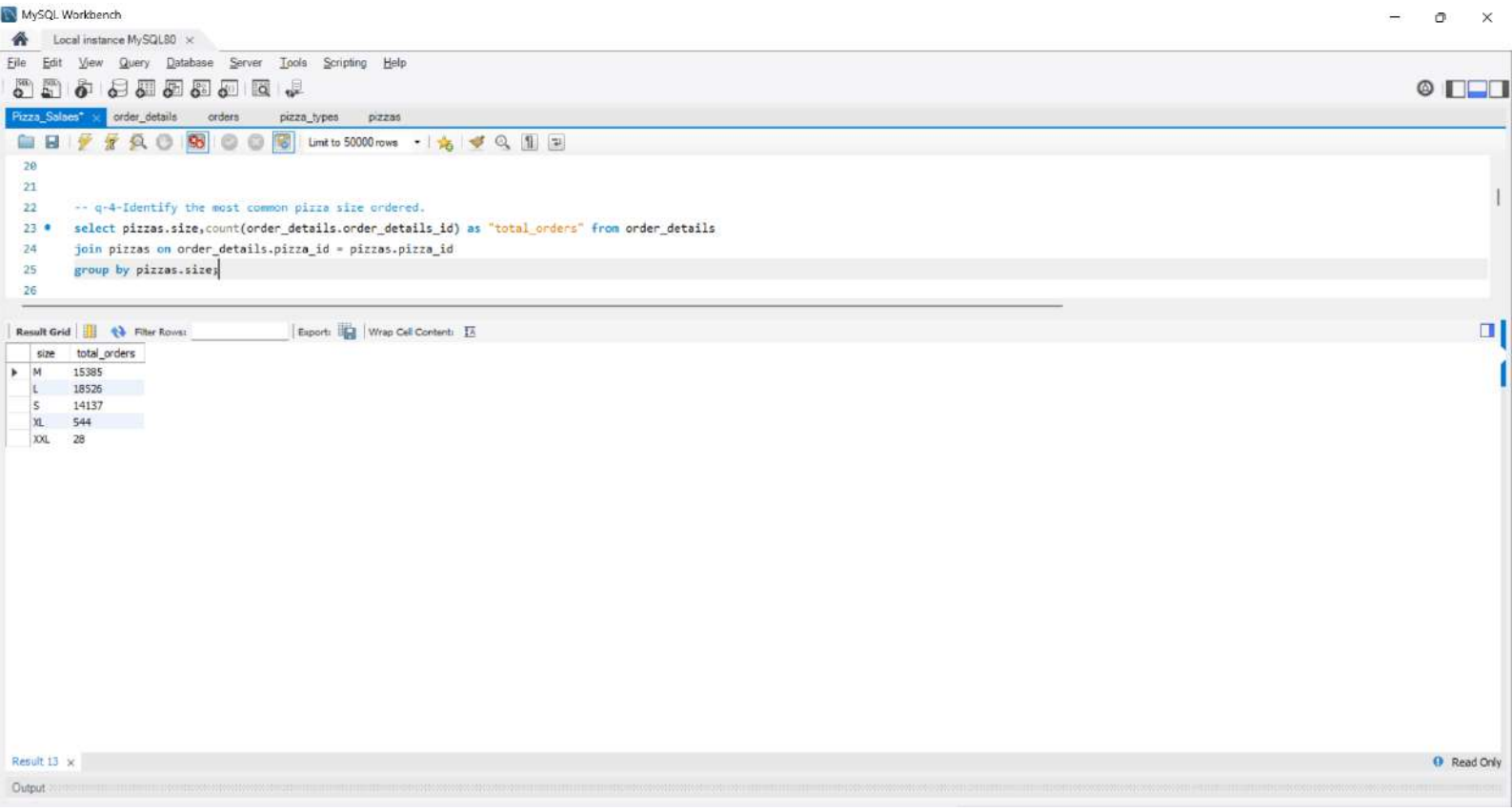
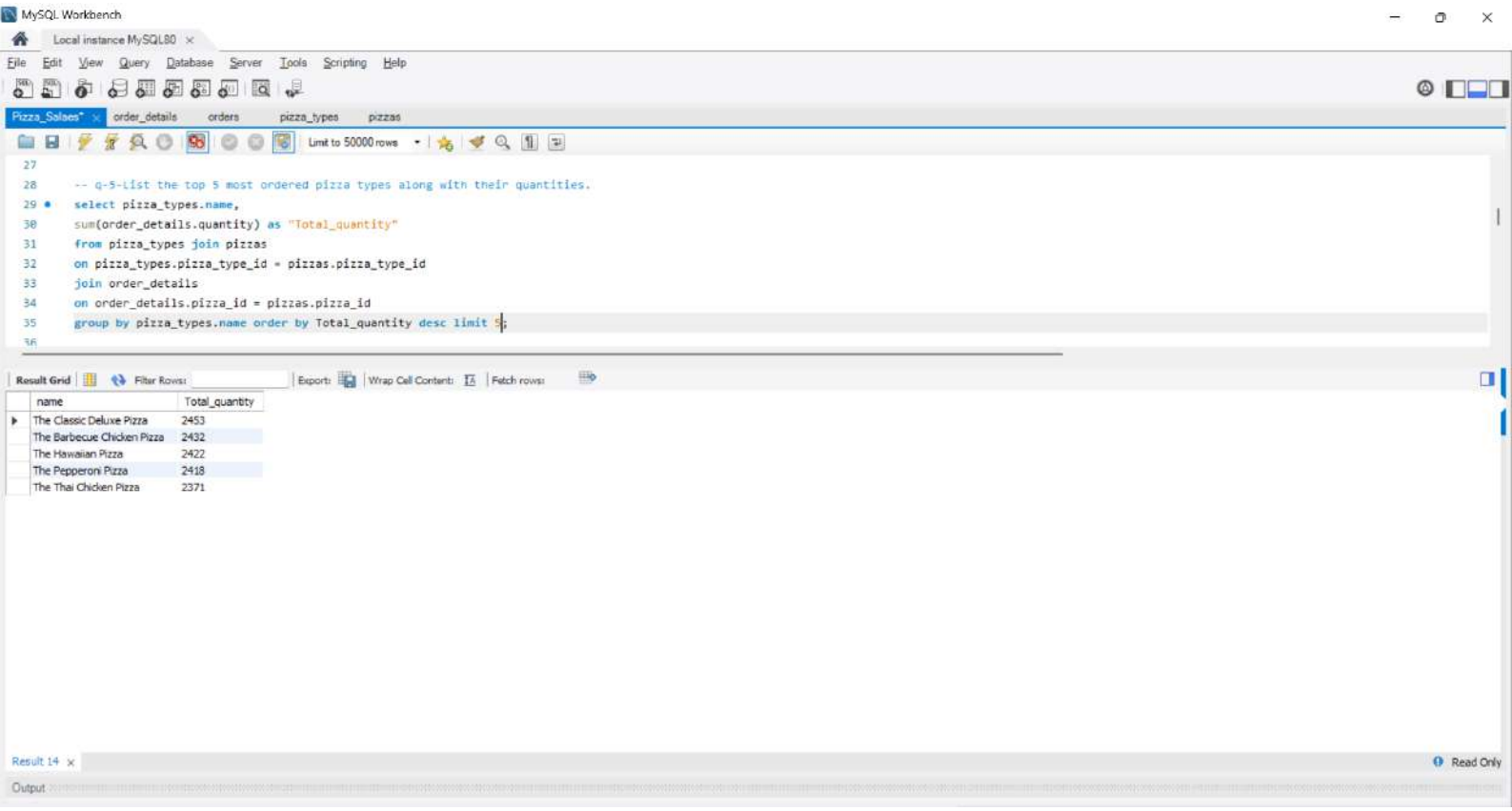
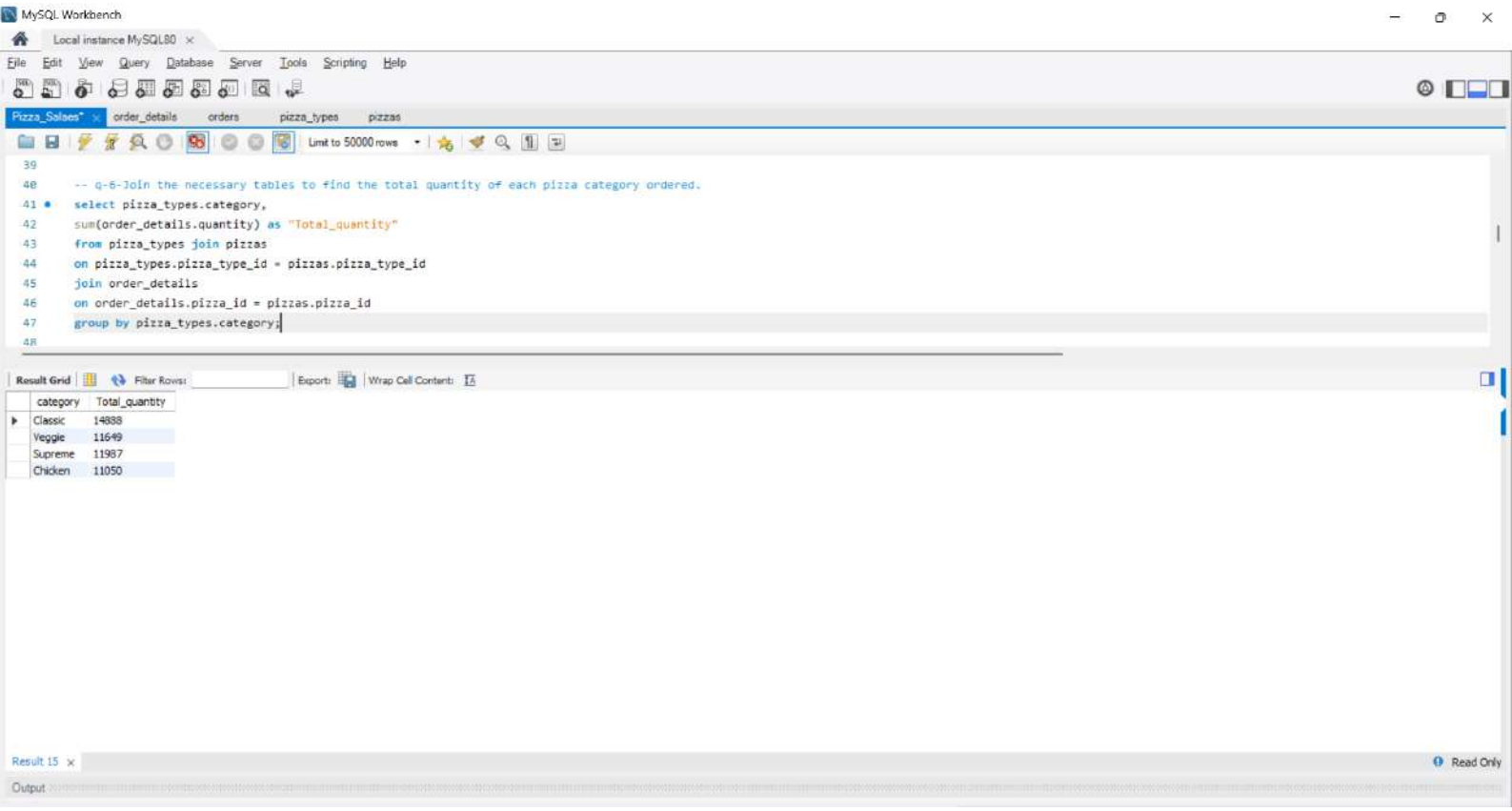

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Pizza_Sales* order_details orders pizza_types pizzas
Limit to 50000 rows
select * from orders;
select * from pizza_types;
select * from pizzas;
-- q-1-Retrieve the total number of orders placed.
select count(`order_id`) as "total_order" from orders;
Result Grid Filter Rows: Export: Wrap Cell Content:
total_order
21350
Result 10
Read Only
Output

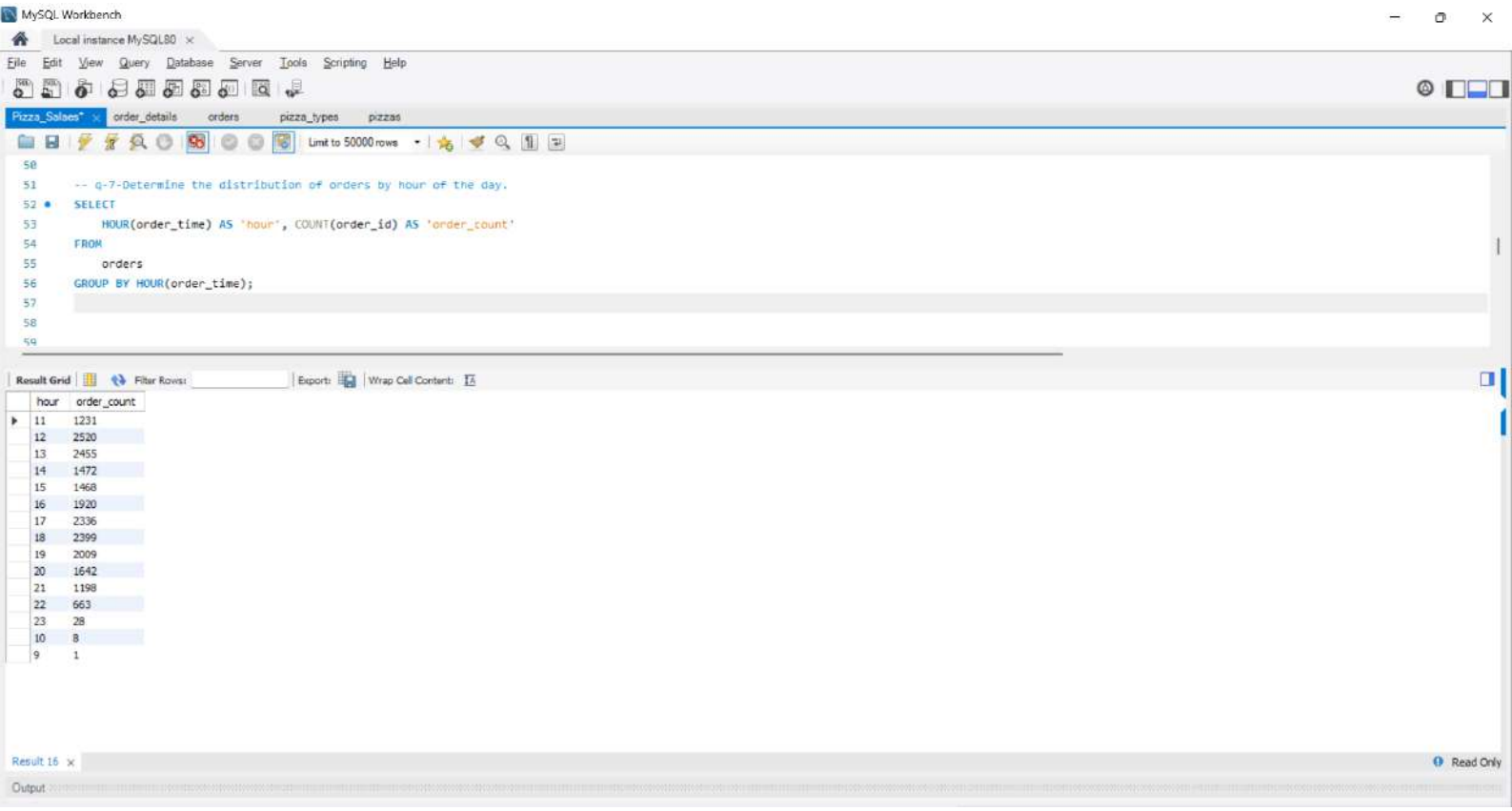

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Pizza_Sales* order_details orders pizza_types pizzas
Limit to 50000 rows
-- q-2-Calculate the total revenue generated from pizza sales.
select sum(order_details.quantity * pizzas.price) as "total_Sales"
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id;
Result Grid Filter Rows: Export: Wrap Cell Content:
total_Sales
817860.049999993
Result 11
Read Only
Output

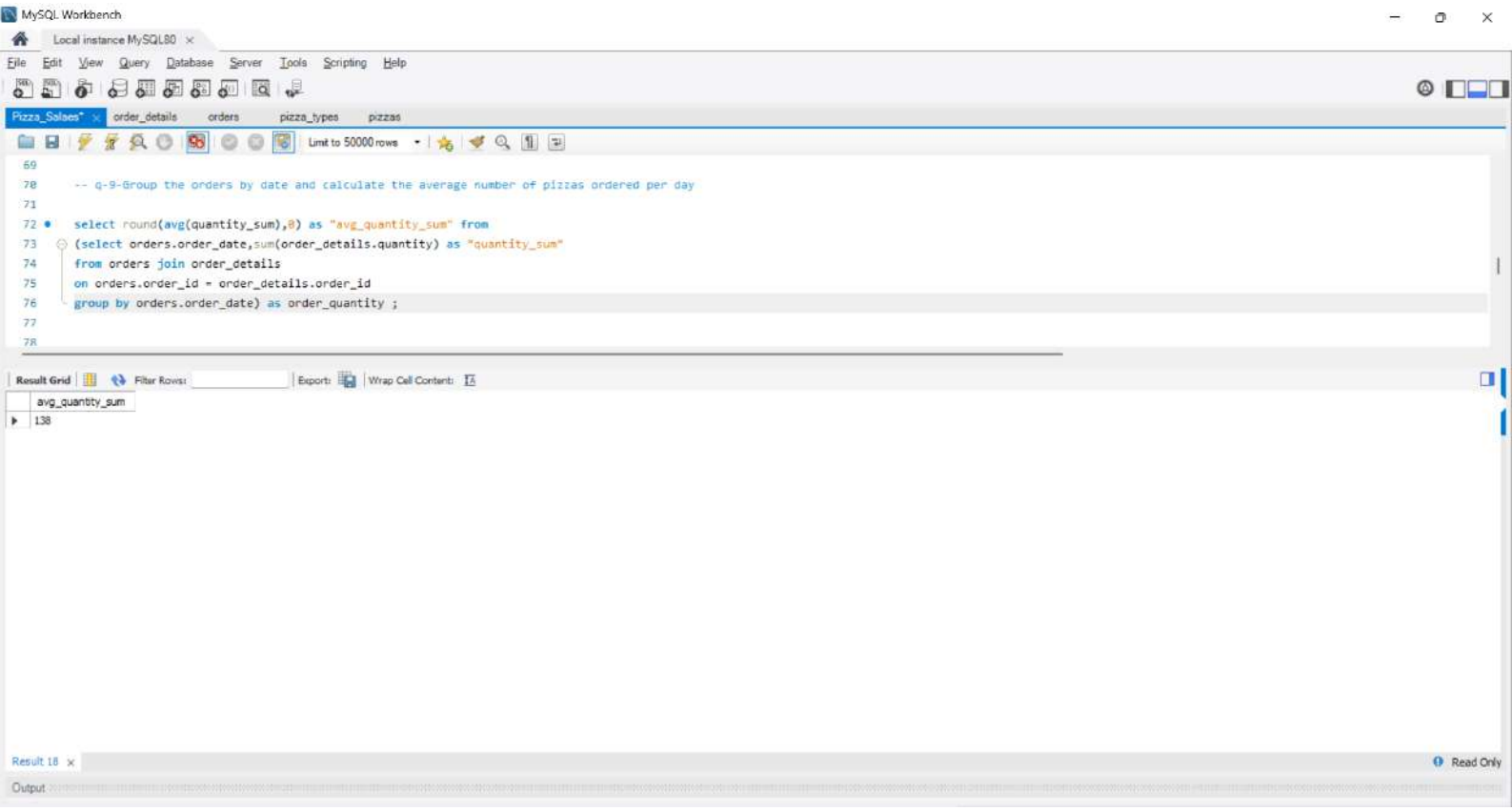

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Pizza_Sales* order_details orders pizza_types pizzas
Limit to 50000 rows
-- q-3-Identify the highest-priced pizza.
select pizza_types.name,pizzas.price from pizza_types join pizzas
on pizza_types.pizza_type_id = 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 12
Read Only
Output

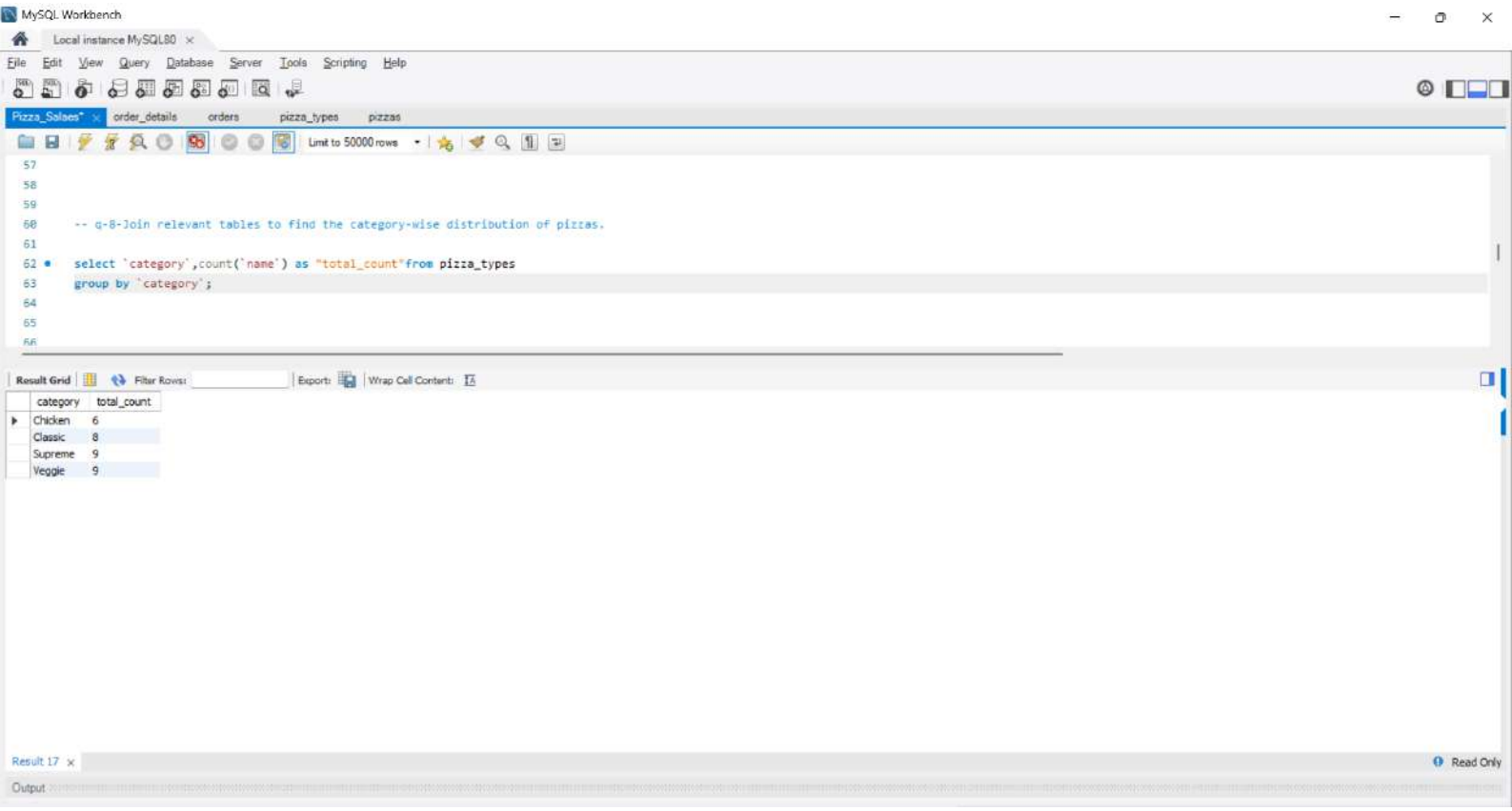

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Pizza_Sales* order_details orders pizza_types pizzas
Limit to 50000 rows
-- q-4-Identify the most common pizza size ordered.
select pizzas.size,count(order_details.order_details_id) as "total_orders" from order_details
join pizzas on order_details.pizza_id = pizzas.pizza_id
group by pizzas.size;
Result Grid Filter Rows: Export: Wrap Cell Content:
size total_orders
M 15385
L 18526
S 14137
XL 544
XXL 28
Result 13
Read Only
Output

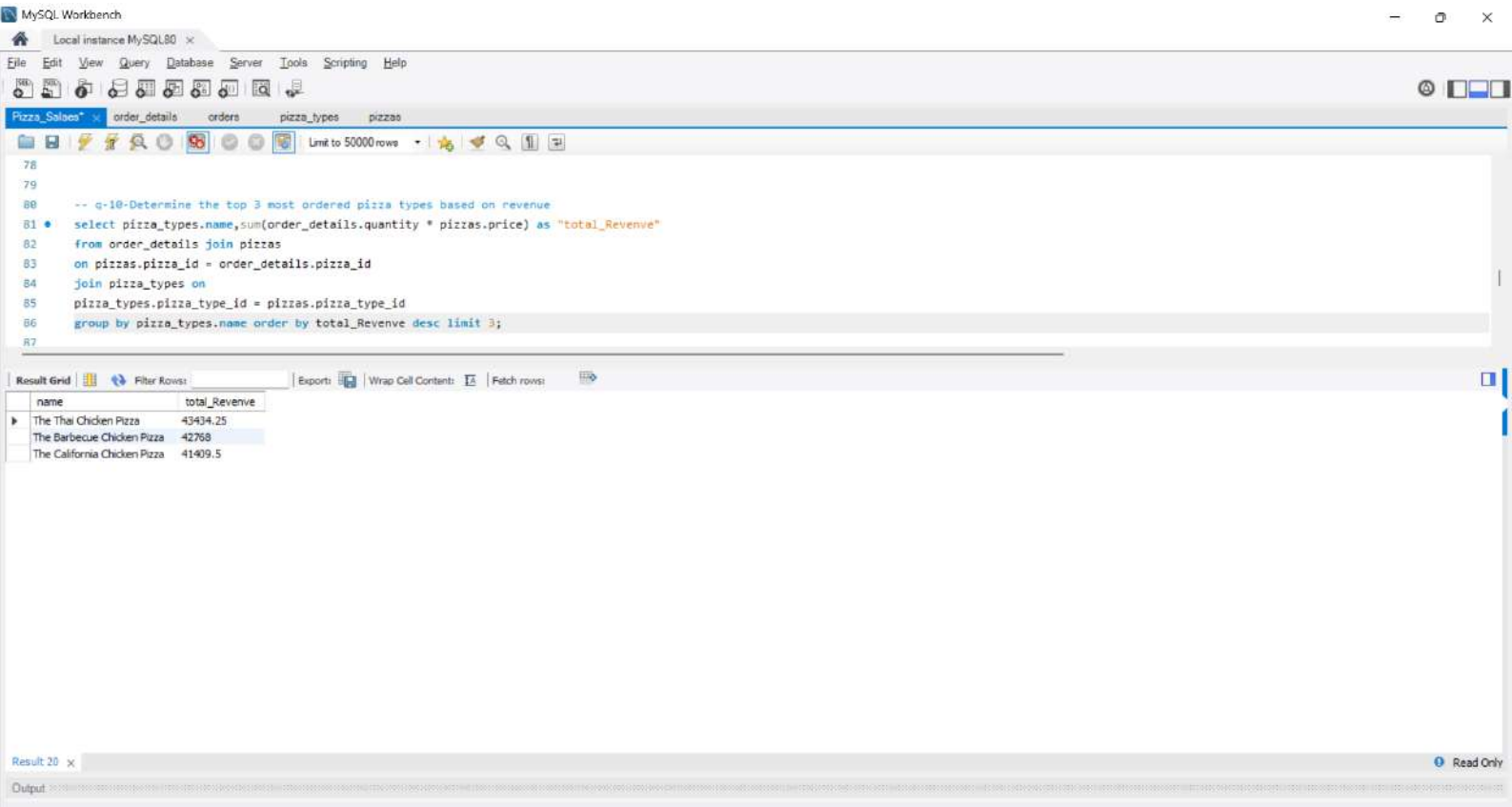

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Pizza_Sales* order_details orders pizza_types pizzas
Limit to 50000 rows
-- q-5-List the top 5 most ordered pizza types along with their quantities.
select pizza_types.name,
sum(order_details.quantity) as "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;
Result Grid Filter Rows: Export: Wrap Cell Content: Fetch rows:
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
Result 14
Read Only
Output

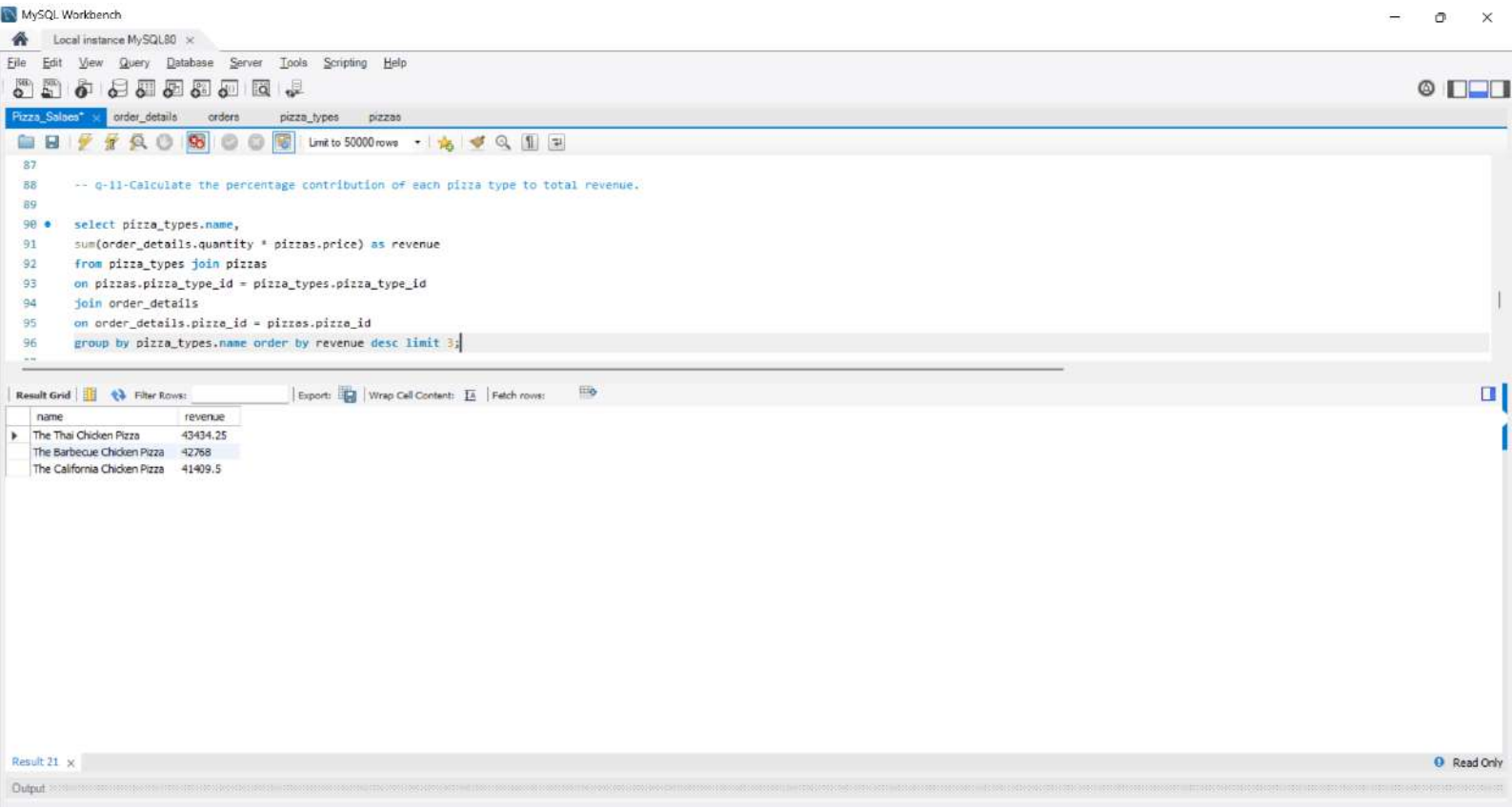

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Pizza_Sales* order_details orders pizza_types pizzas
Limit to 50000 rows
-- q-6-Join the necessary tables to find the total quantity of each pizza category ordered.
select pizza_types.category,
sum(order_details.quantity) 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;
Result Grid Filter Rows: Export: Wrap Cell Content:
category Total_quantity
Classic 14888
Veggie 11649
Supreme 11987
Chicken 11050
Result 15
Read Only
Output

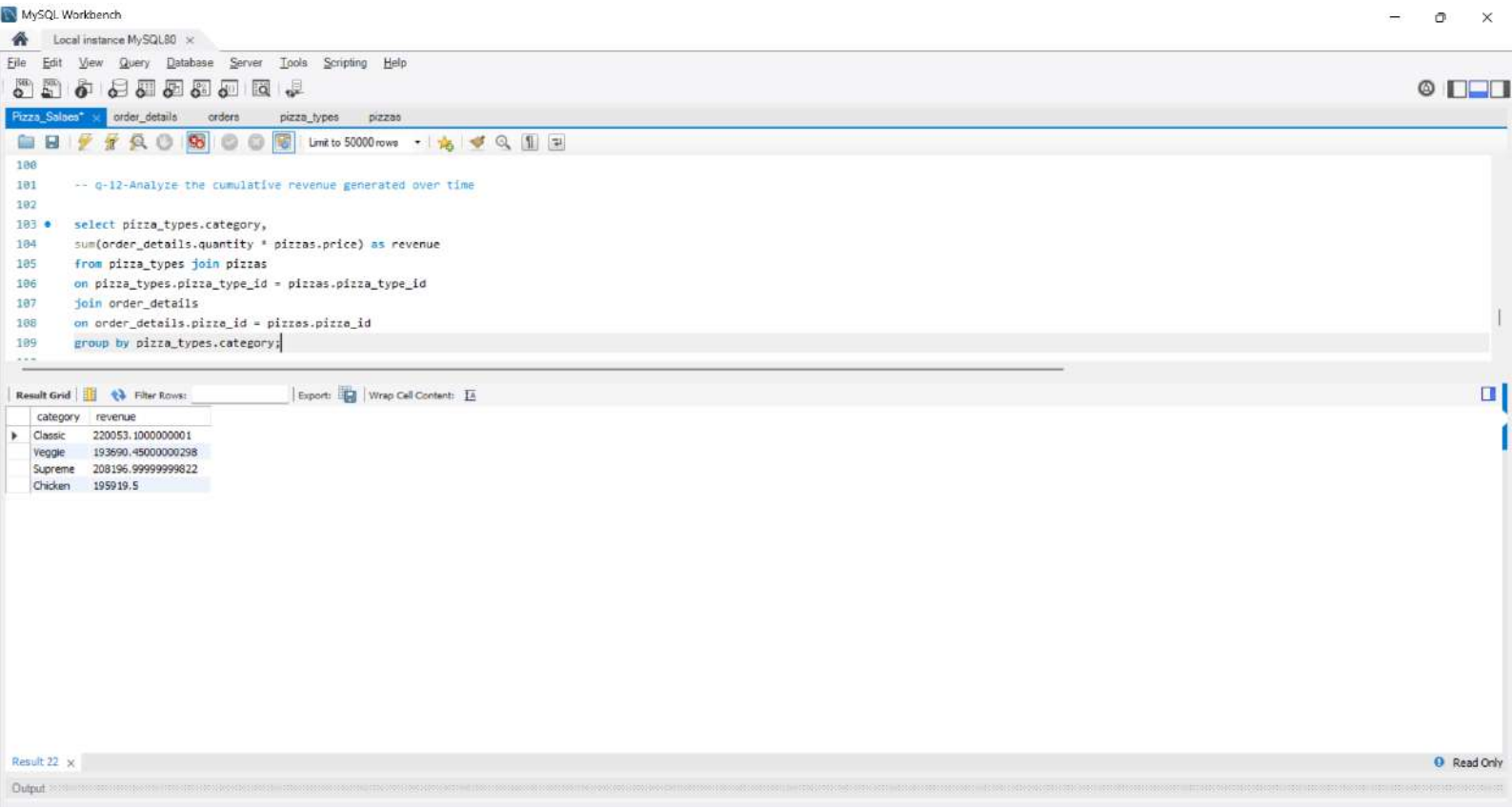

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Pizza_Sales* order_details orders pizza_types pizzas
Limit to 50000 rows
50
51 -- q-7-Determine the distribution of orders by hour of the day.
52 SELECT
53 HOUR(order_time) AS 'hour', COUNT(order_id) AS 'order_count'
54 FROM
55 orders
56 GROUP BY HOUR(order_time);
57
58
59
Result Grid Filter Rows: Export: Wrap Cell Content:
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
Result 16
Read Only
Output

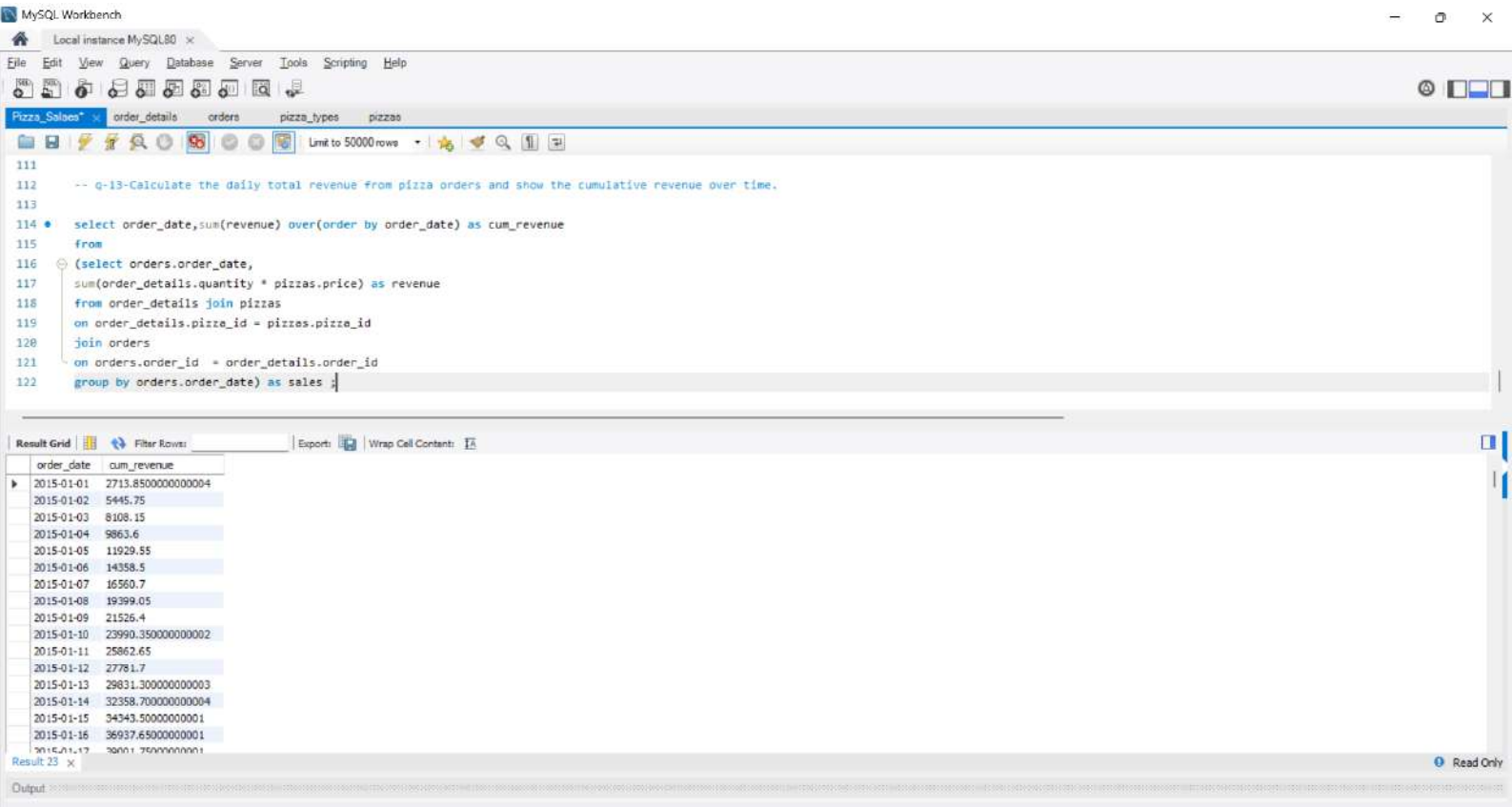

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Pizza_Sales*
order_details
orders
pizza_types
pizzas
Limit to 50000 rows
-- q-9-Group the orders by date and calculate the average number of pizzas ordered per day
select round(avg(quantity_sum),0) as "avg_quantity_sum" from
(select orders.order_date,sum(order_details.quantity) as "quantity_sum"
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:
avg_quantity_sum
138
Result 18
Read Only
Output


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Pizza_Sales* order_details orders pizza_types pizzas
Limit to 50000 rows
-- q-8-Join relevant tables to find the category-wise distribution of pizzas.
select `category`,count(`name`) as "total_count"from pizza_types
group by `category`;
Result Grid Filter Rows: Export: Wrap Cell Content:
category total_count
Chicken 6
Classic 8
Supreme 9
Veggie 9
Result 17
Read Only
Output


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Pizza_Sales* order_details orders pizza_types pizzas
Limit to 50000 rows
78
79
80 -- q-10-Determine the top 3 most ordered pizza types based on revenue
81 select pizza_types.name,sum(order_details.quantity * pizzas.price) as "total_Revenve"
82 from order_details join pizzas
83 on pizzas.pizza_id = order_details.pizza_id
84 join pizza_types on
85 pizza_types.pizza_type_id = pizzas.pizza_type_id
86 group by pizza_types.name order by total_Revenve desc limit 3;
87
Result Grid Filter Rows: Export: Wrap Cell Contents: Fetch rows:
name total_Revenve
The Thai Chicken Pizza 43434.25
The Barbecue Chicken Pizza 42768
The California Chicken Pizza 41409.5
Result 20
Read Only
Output


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Pizza_Sales* order_details orders pizza_types pizzas
Limit to 50000 rows
87
88 -- q-11-Calculate the percentage contribution of each pizza type to total revenue.
89
90 select pizza_types.name,
91 sum(order_details.quantity * pizzas.price) as revenue
92 from pizza_types join pizzas
93 on pizzas.pizza_type_id = pizza_types.pizza_type_id
94 join order_details
95 on order_details.pizza_id = pizzas.pizza_id
96 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 21
Read Only
Output


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Pizza_Sales* order_details orders pizza_types pizzas
Limit to 50000 rows
-- q-12-Analyze the cumulative revenue generated over time
select pizza_types.category,
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;
Result Grid Filter Rows: Export: Wrap Cell Content:
category revenue
Classic 220053.1000000001
Veggie 193690.45000000298
Supreme 208196.99999999822
Chicken 195919.5
Result 22
Read Only
Output


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Pizza_Sales* order_details orders pizza_types pizzas
Limit to 50000 rows
-- q-13-Calculate the daily total revenue from pizza orders and show the cumulative revenue over time.
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 ;
Result Grid Filter Rows: Export: Wrap Cell Content:
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
Result 23
Read Only
Output