BAKERS & CO.

Hello!

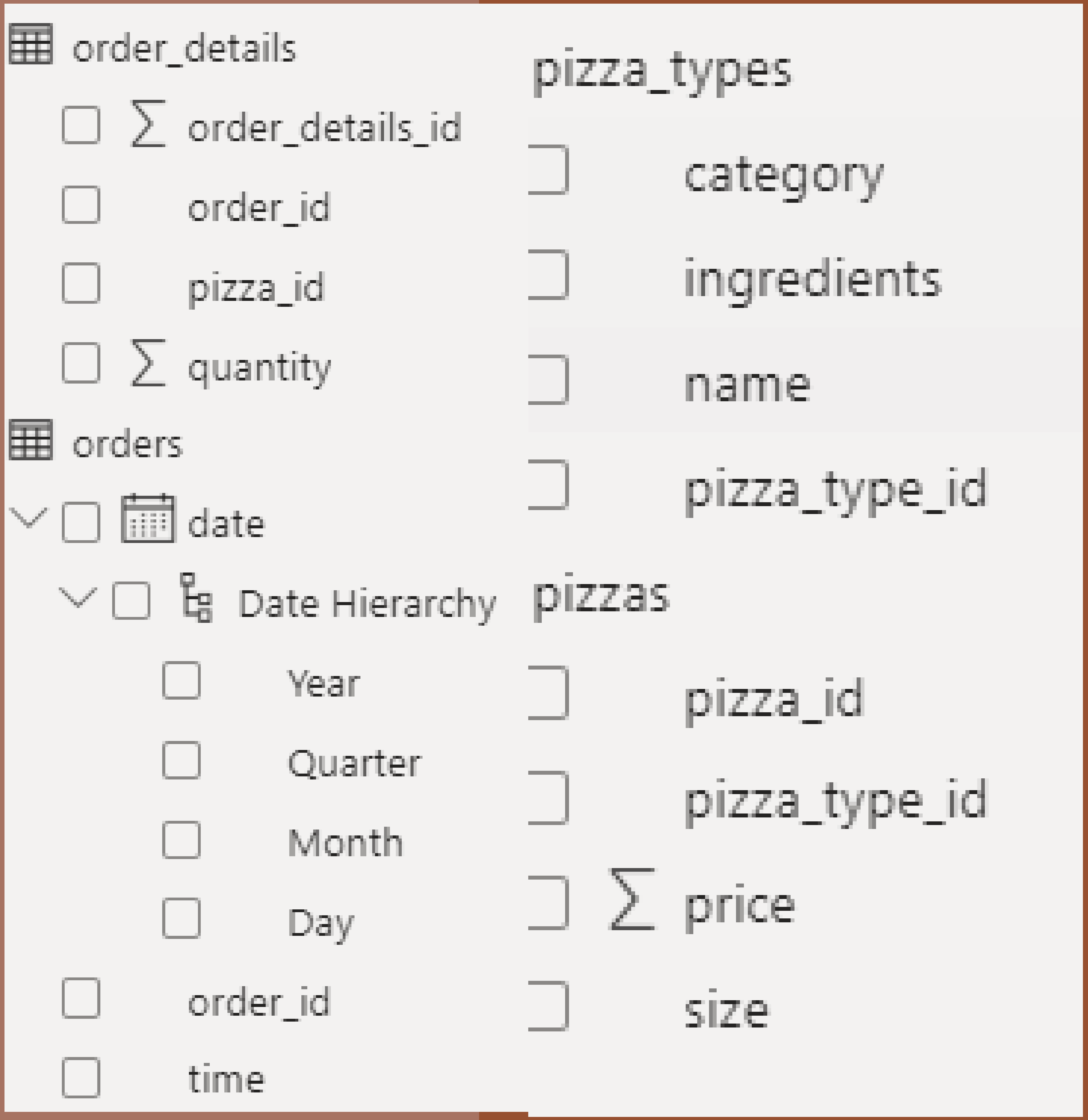
Bakers & Co. is a top bakery known for crafting delicious pizzas with high-quality ingredients. With a range of classic and specialty options, we aim to satisfy every pizza enthusiast's cravings.

Use MYSQL WORKBENCH

To Make a Work report / Annual report / Revenue report

To create a work report, annual report, or revenue report using MySQL Workbench, you would typically use SQL queries to extract the necessary data from your database tables. Here's a general outline of steps you can follow to generate the reports:

USED TABLES AND COLUMNS



order_details
- order_details_id
- order_id
- pizza_id
- quantity

orders
- date
  - Date Hierarchy
    - Year
    - Quarter
    - Month
    - Day
- order_id
- time

pizza_types
- category
- ingredients
- name
- pizza_type_id

pizzas
- pizza_id
- pizza_type_id
- price
- size

Easy Questions&Queries with solution

Q1

Retrieve the total number of orders placed.

A1

select count(order_id) as total_orders
from orders;

Ans - 21350

Q2

Calculate the total revenue generated from pizza sales.

A2

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

Ans - 817860.05

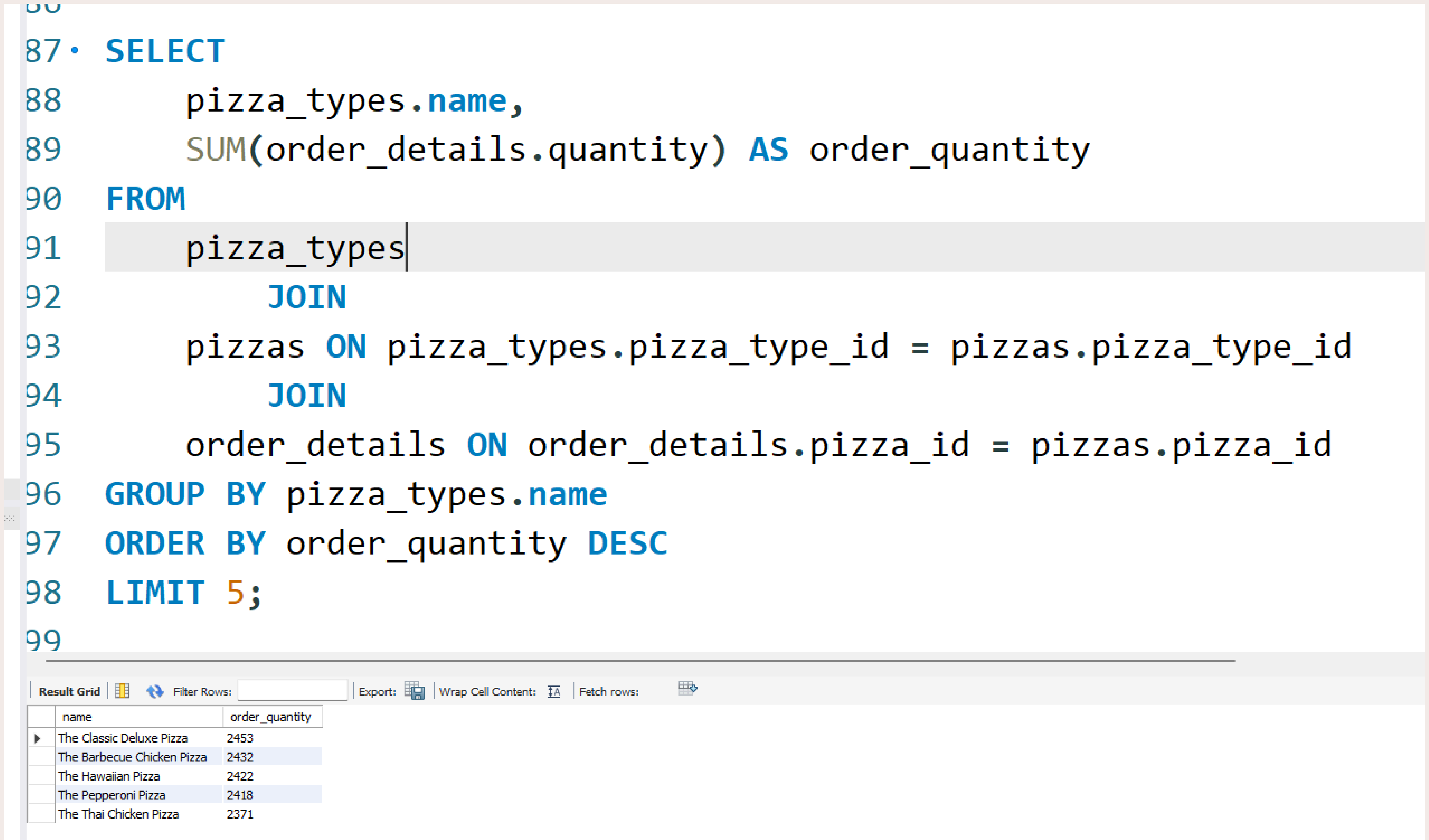
Q3

Identify the highest-priced pizza.

A3

select pizzas.price , pizza_types.name
from pizzas join pizza_types
on pizzas.pizza_type_id =
pizza_types.pizza_type_id
order by pizzas.price desc limit 1;
Ans - # price, name
'35.95', 'The Greek Pizza'

Q4 -List the top 5 most ordered pizza types along with their quantities.

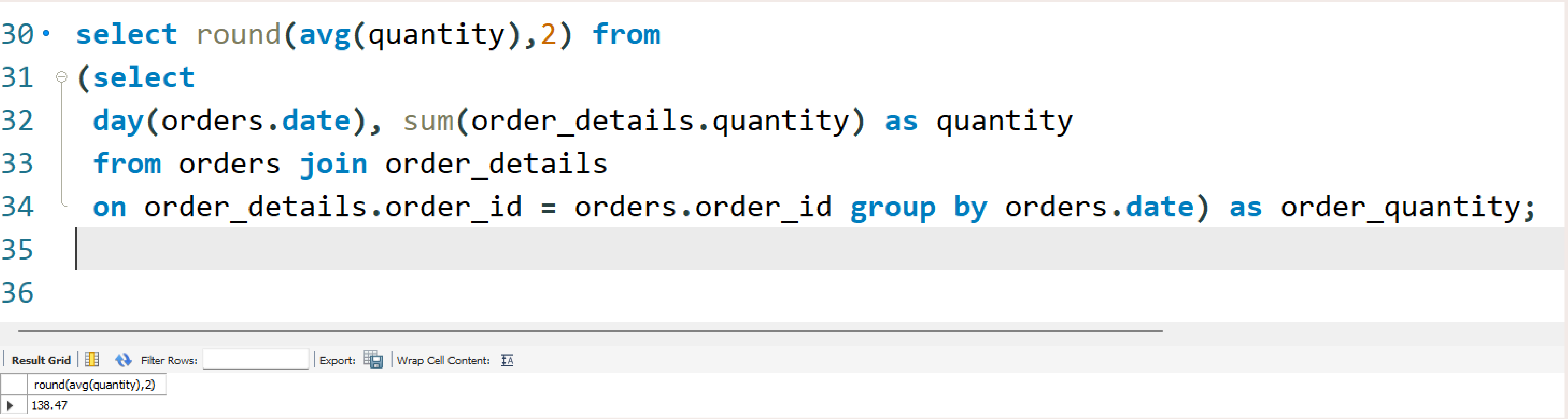


```
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS order_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY order_quantity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

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

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

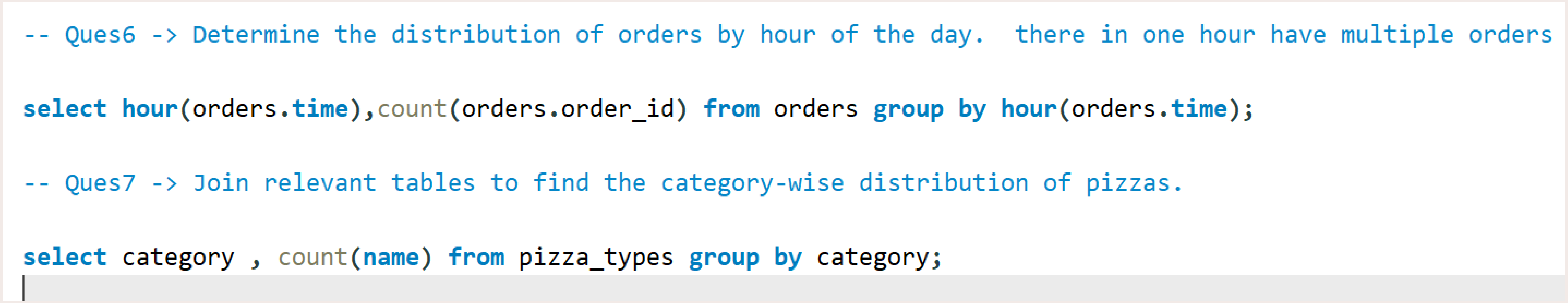
Q5- Group the orders by date and calculate the average number of pizzas ordered per day.



```
select round(avg(quantity),2) from
(select
 day(orders.date), sum(order_details.quantity) as quantity
 from orders join order_details
 on order_details.order_id = orders.order_id group by orders.date) as order_quantity;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

```
round(avg(quantity),2)
138.47
```



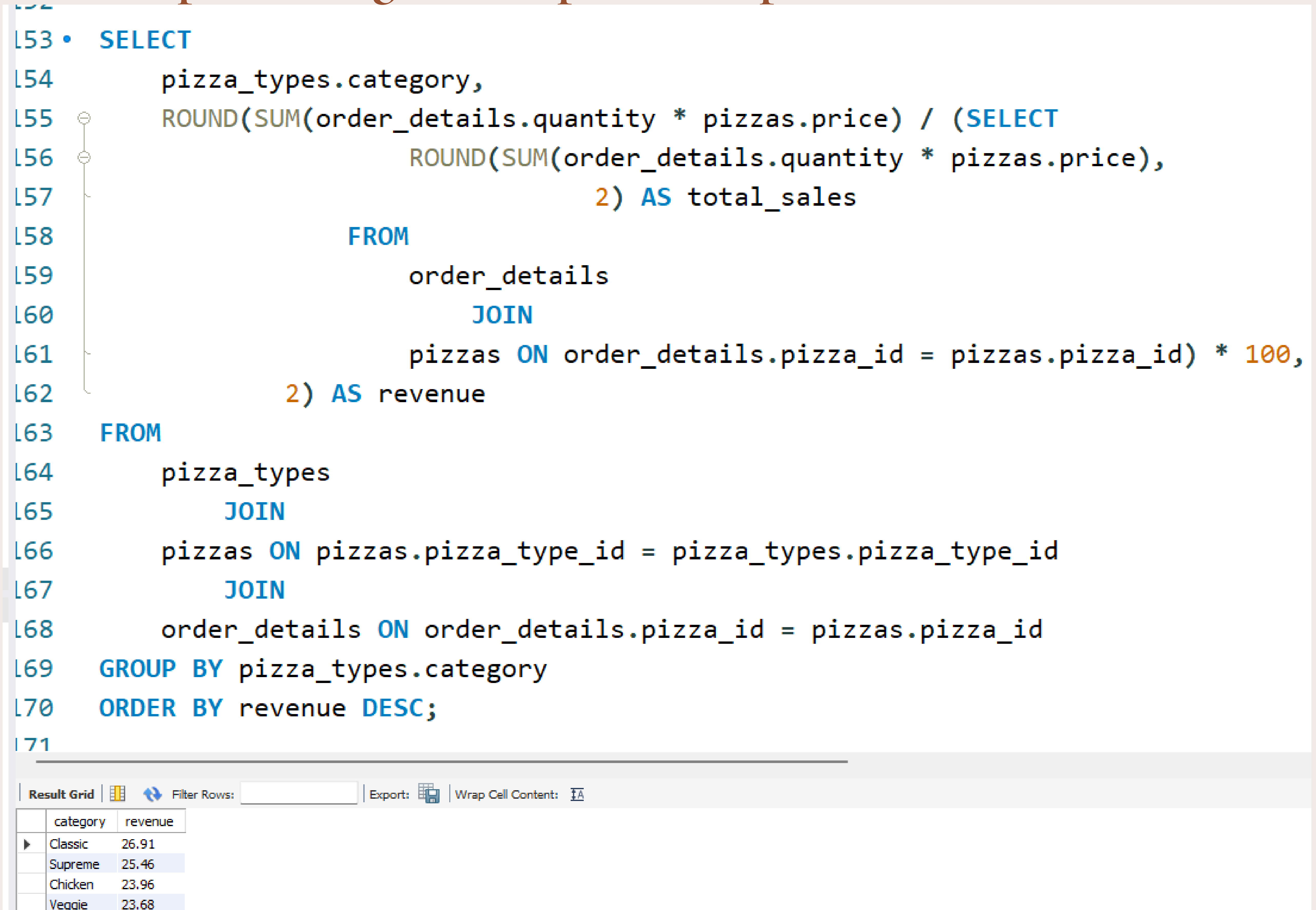
```
-- Ques6 -> Determine the distribution of orders by hour of the day.  there in one hour have multiple orders

select hour(orders.time),count(orders.order_id) from orders group by hour(orders.time);

-- Ques7 -> Join relevant tables to find the category-wise distribution of pizzas.

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

Q8-Calculate the percentage contribution of each pizza type to total revenue. hint -- Group by krne se each (product/category) ka sales price a jata h (for the percentage (one_product_price)/(total sales)*100

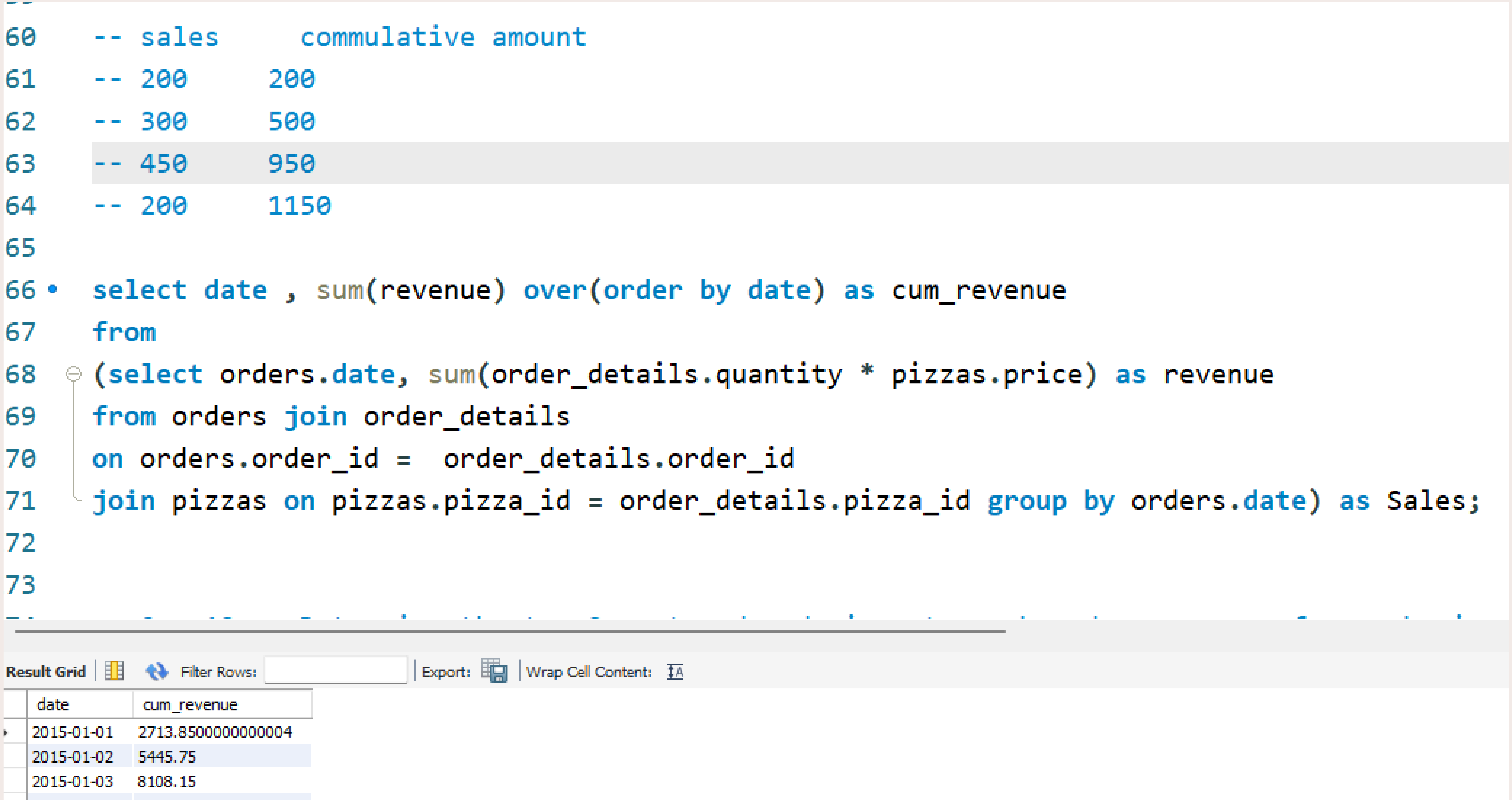


```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                ROUND(SUM(order_details.quantity * pizzas.price),
                            2) AS total_sales
            FROM
                order_details
                    JOIN
                pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
            2) 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.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

Q9 -Analyze the cumulative revenue generated over time.
hint - it means by one by one day how's the incresement in their revenue



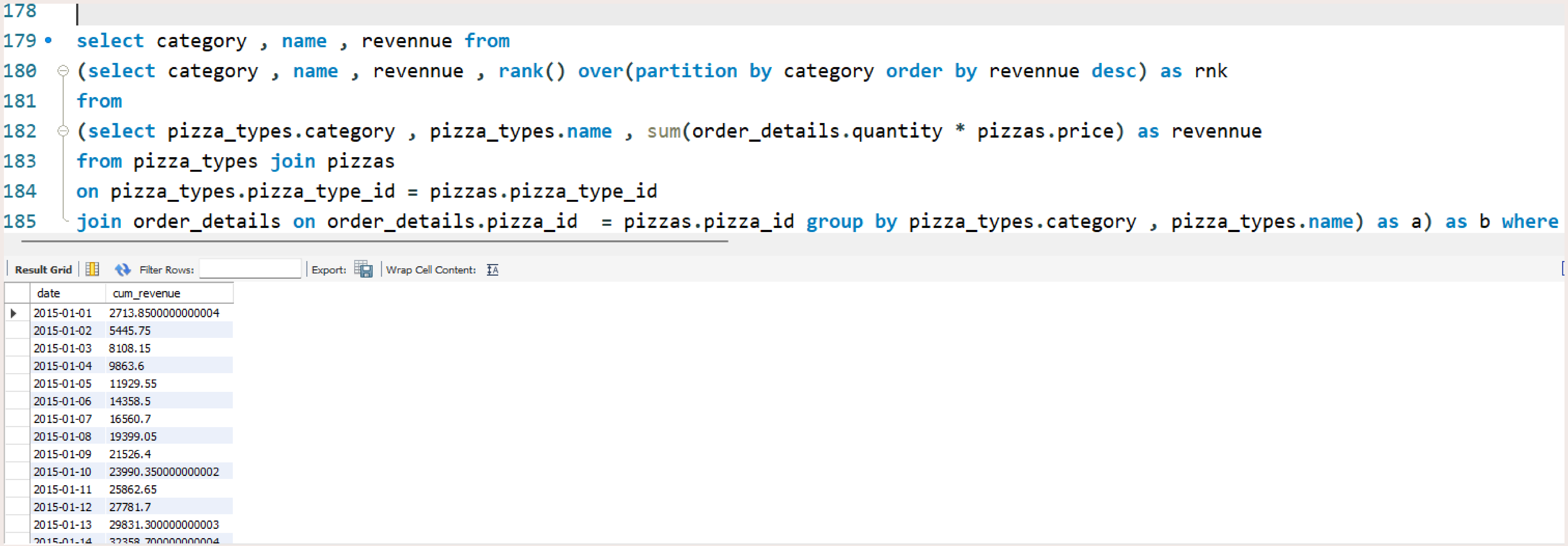
```
-- sales      commulative amount
-- 200      200
-- 300      500
-- 450      950
-- 200      1150

select date , sum(revenue) over(order by date) as cum_revenue
from
(select orders.date, sum(order_details.quantity * pizzas.price) as revenue
from orders join order_details
on orders.order_id =  order_details.order_id
join pizzas on pizzas.pizza_id = order_details.pizza_id group by orders.date) as Sales;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

```
date        cum_revenue
2015-01-01  2713.8500000000004
2015-01-02  5445.75
2015-01-03  8108.15
```

Q10 -Determine the top 3 most ordered pizza types based on revenue for each pizza category.
hint - means rank according to the each pizza category



```sql

select category , name , revennue from
(select category , name , revennue , rank() over(partition by category order by revennue desc) as rnk
from
(select pizza_types.category , pizza_types.name , sum(order_details.quantity * pizzas.price) as revennue
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
```

```
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
```

This is all necessary Questions or Queries which helps to find the better result

BAKERS & CO.