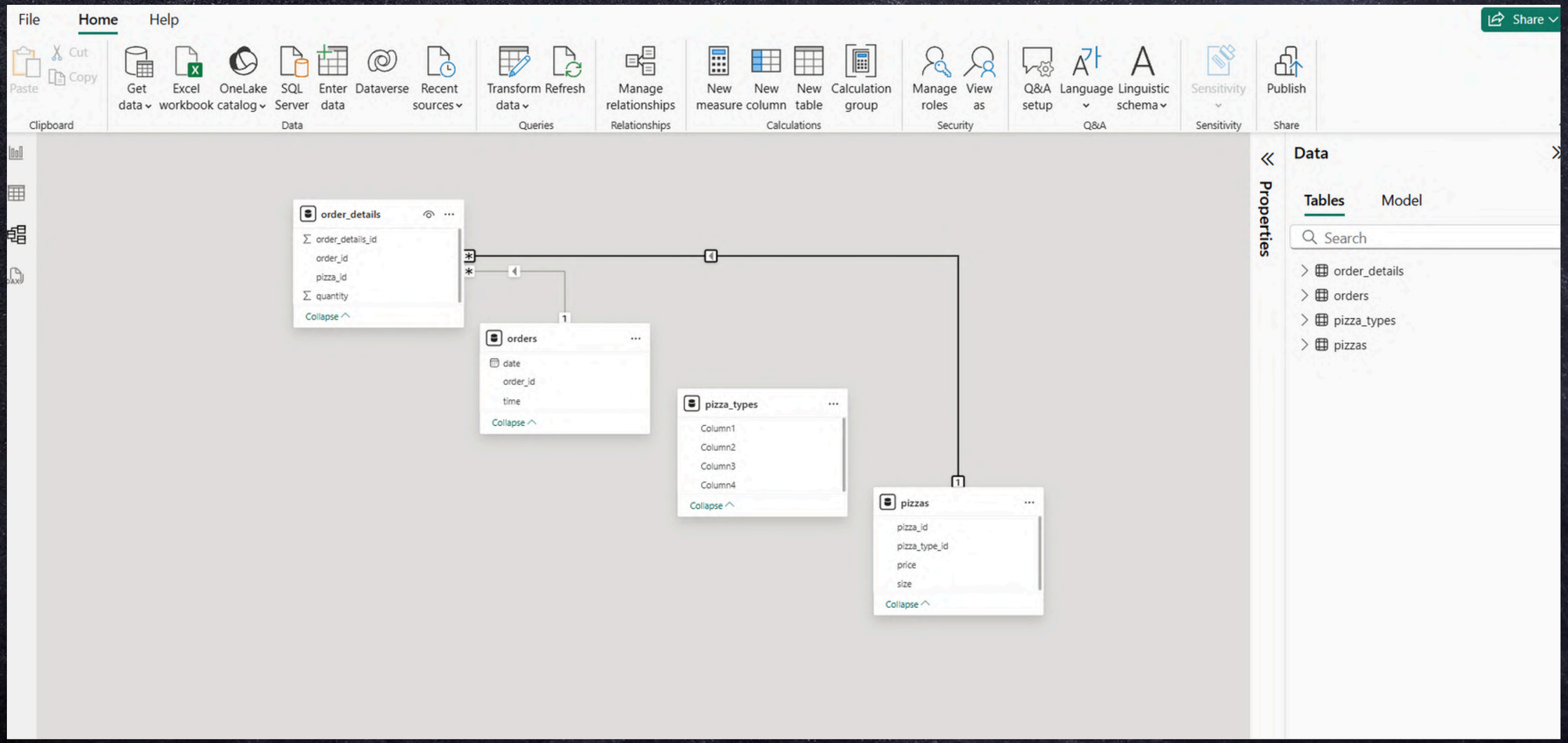
Hello, my name is Alok Gupta. In this project, I have analyzed pizza sales data using SQL. The goal was to solve various business-related queries by writing SQL statements to extract useful insights from the database. I explored key metrics such as best-selling pizzas, peak sales hours, order trends, and revenue breakdown. This analysis helps in understanding sales performance and customer preferences more effectively.

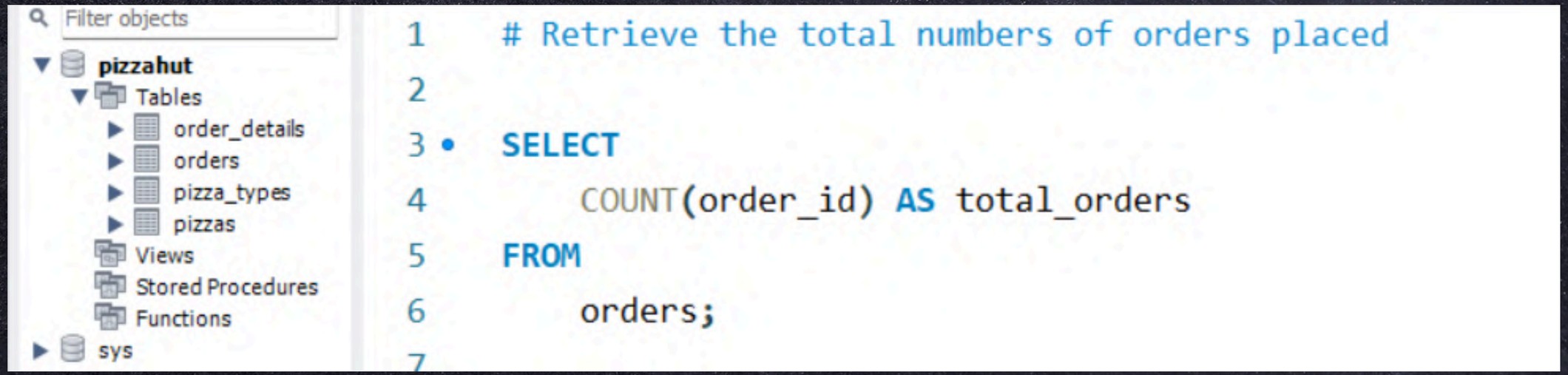
QUESTIONS

- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.
- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.
- IDENTIFY THE HIGHEST-PRICED PIZZA.
- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.
- LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.
- JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.
- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.
- JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.
- GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.
- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.
- CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

DATA MODEL

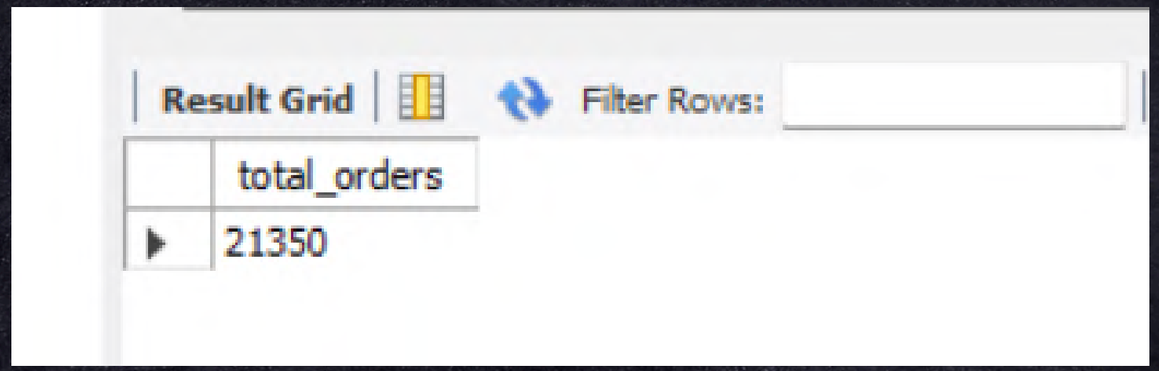

File
Home
Help
Share
Paste
Cut
Copy
Clipboard
Get data
Excel workbook
OneLake catalog
SQL Server
Enter data
Dataverse
Recent sources
Data
Transform data
Refresh
Queries
Manage relationships
Relationships
New measure
New column
New table
Calculation group
Calculations
Manage roles
View as
Security
Q&A setup
Language
Linguistic schema
Q&A
Sensitivity
Publish
Share
order_details
order_details_id
order_id
pizza_id
quantity
Collapse
orders
date
order_id
time
Collapse
pizza_types
Column1
Column2
Column3
Column4
Collapse
pizzas
pizza_id
pizza_type_id
price
size
Collapse
Properties
Data
Tables
Model
Search
order_details
orders
pizza_types
pizzas

Retrieve the total numbers of orders placed.



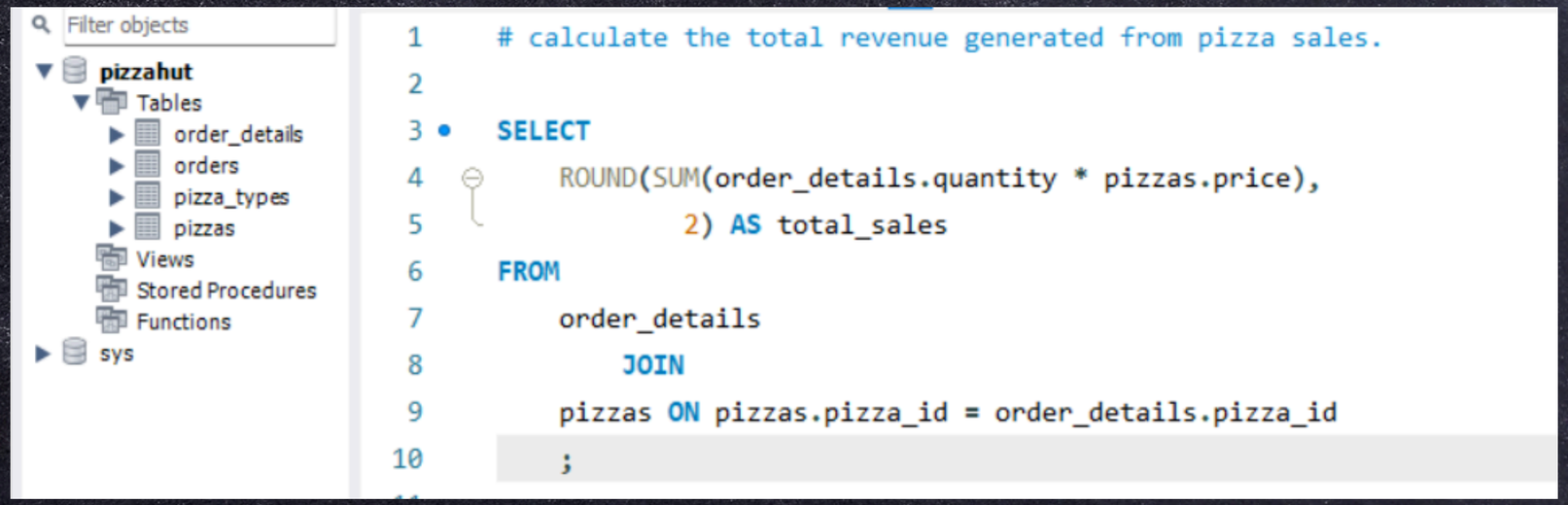

```
# Retrieve the total numbers of orders placed

SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```



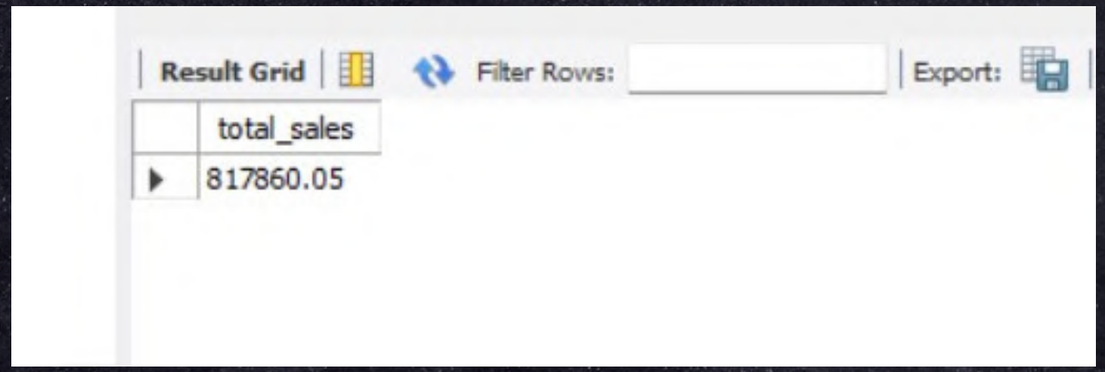
total_orders
21350

Calculate the total revenue generated from pizza sales.



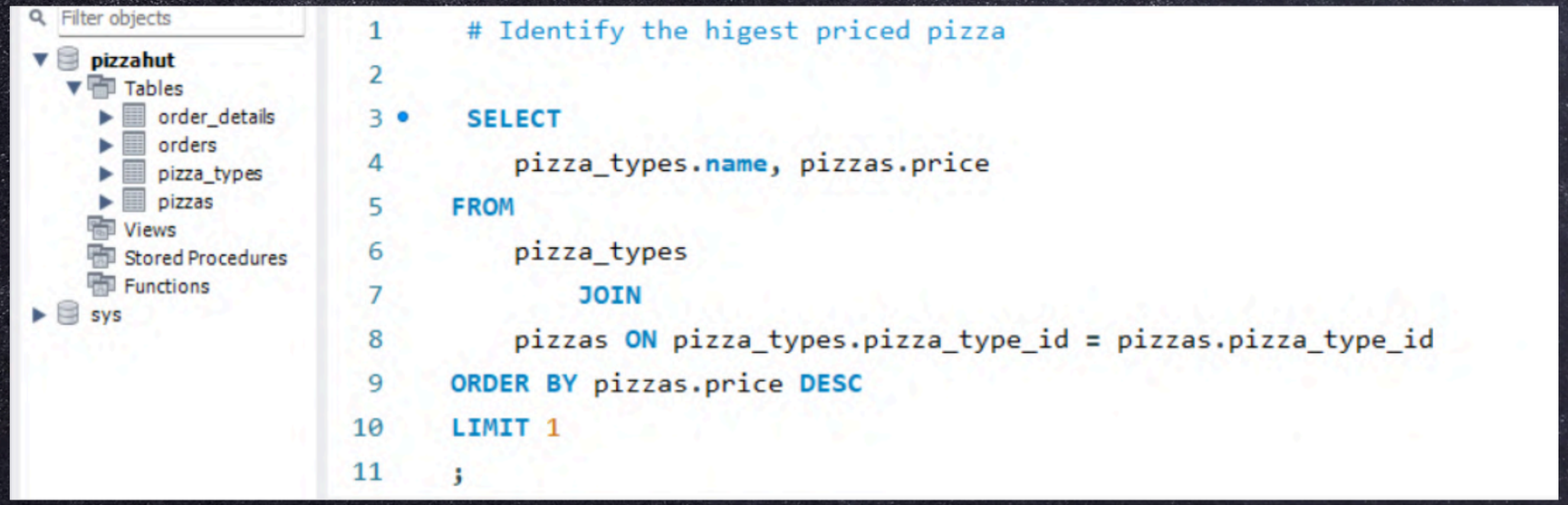

```
# calculate the total revenue generated from pizza sales.

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
    ;
```



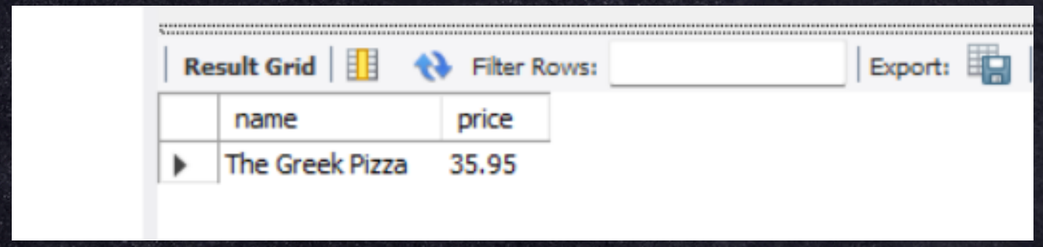
total_sales
817860.05

Identify the highest-priced pizza.




```sql
# Identify the higest 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
;
```



name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

Filter objects

- pizzahut
  - Tables
    - order_details
    - orders
    - pizza_types
    - pizzas
  - Views
  - Stored Procedures
  - Functions
- sys

```sql
# Identity the most common pizza size orderd.

SELECT
    quantity, COUNT(order_details_id)
FROM
    order_details
GROUP BY quantity;

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
;
```

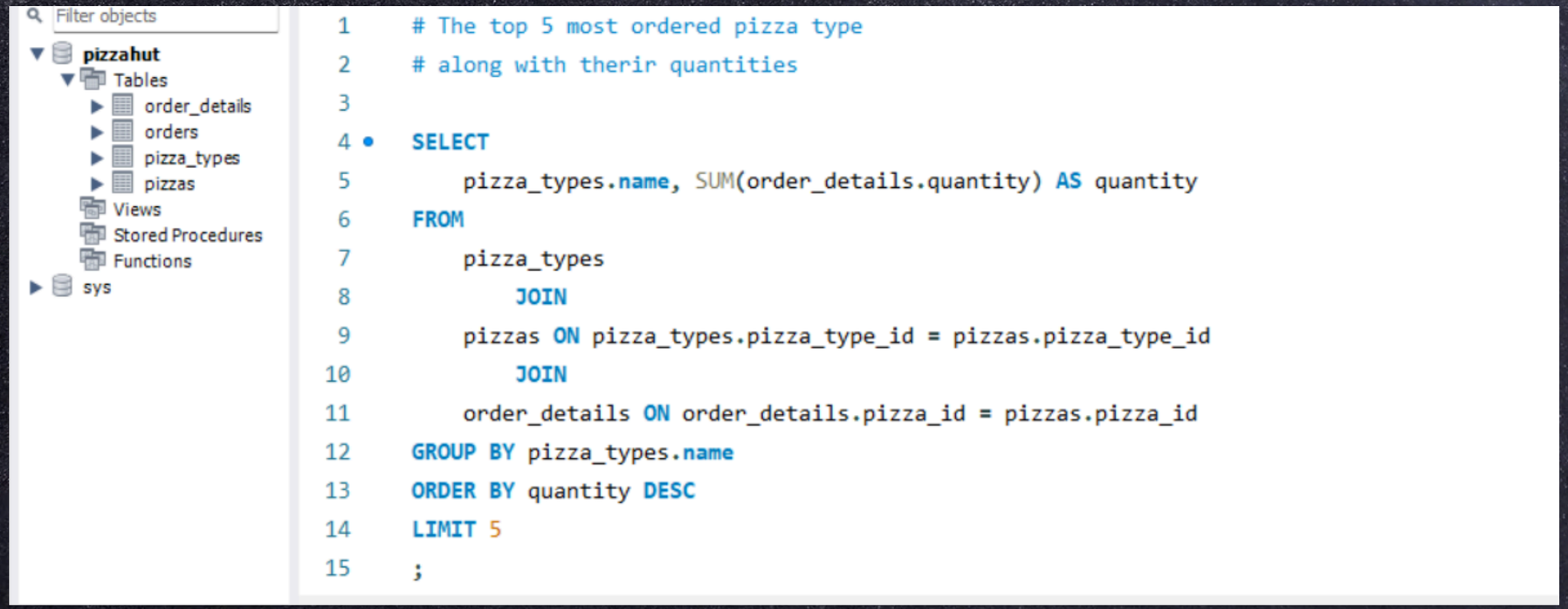
Result Grid | Filter Rows:

quantity	count(order_details_id)
1	47693
2	903
3	21
4	3

Result Grid | Filter Rows: | Export:

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

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

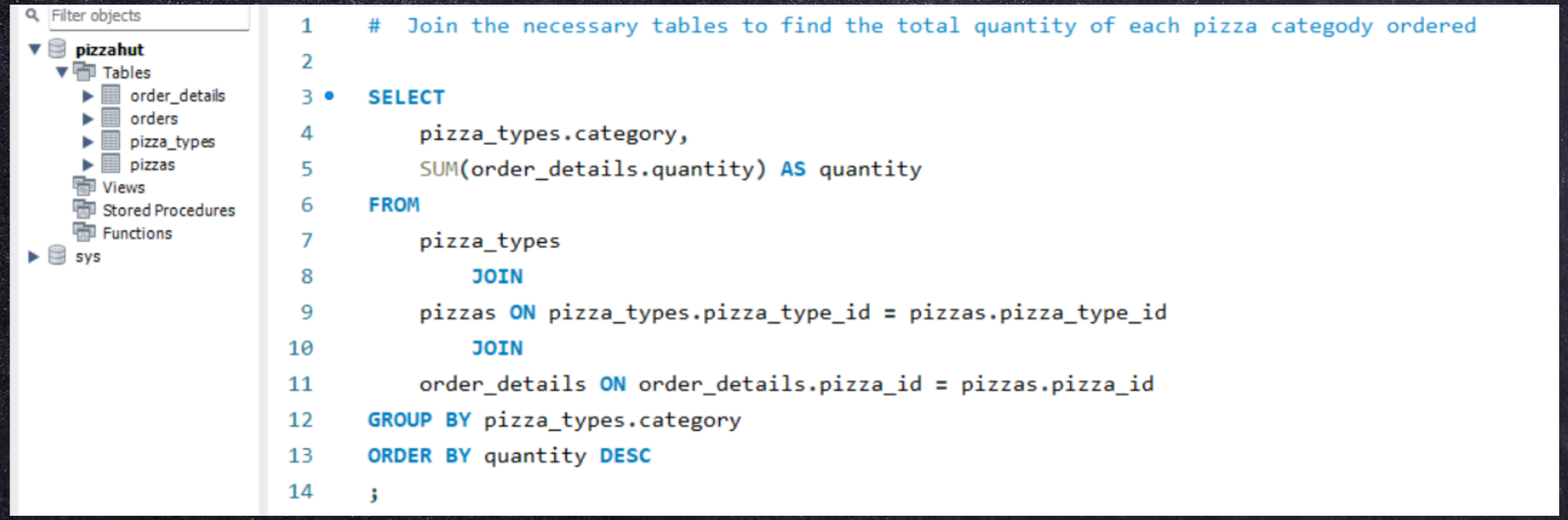



```sql
# The top 5 most ordered pizza type
# along with therir quantities

SELECT
    pizza_types.name, 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 pizza_types.name
ORDER BY quantity DESC
LIMIT 5
;
```

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.



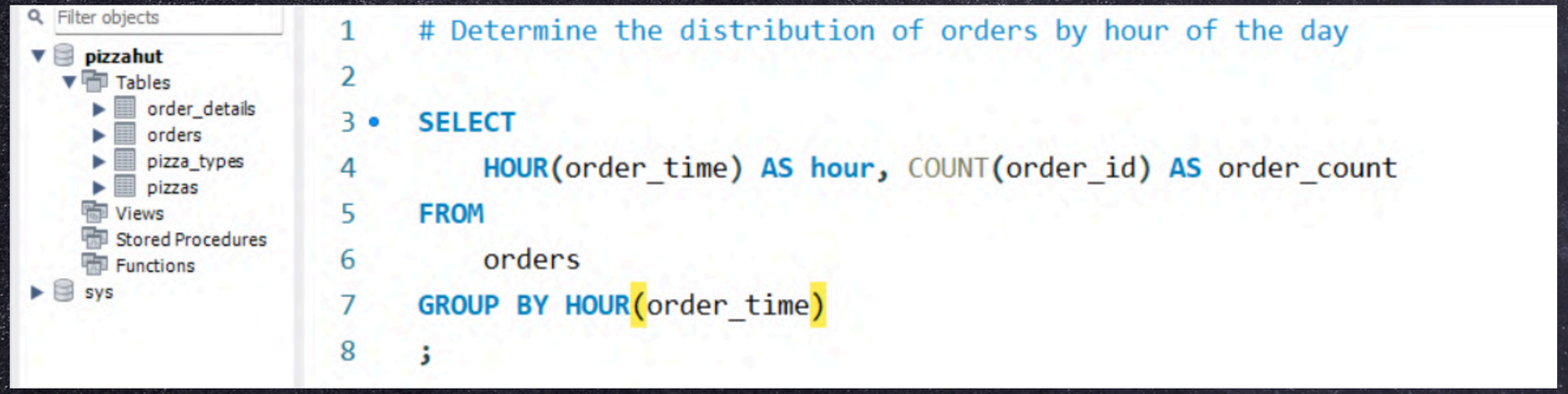

```sql
#  Join the necessary tables to find the total quantity of each pizza categody ordered

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 pizza_types.category
ORDER BY quantity DESC
;
```

Result Grid | Filter Rows:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

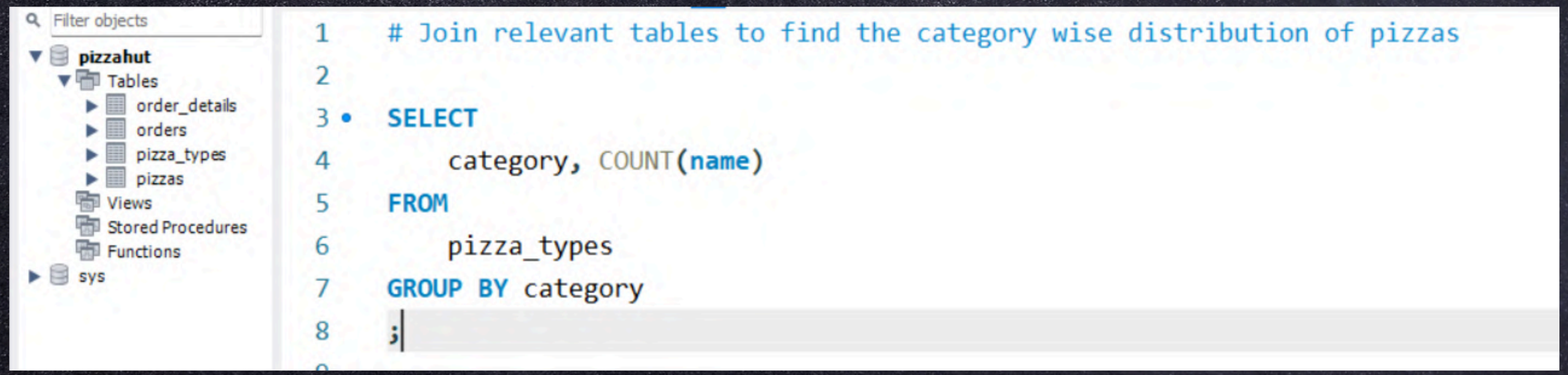



```sql
# Determine the distribution of orders by hour of the day

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

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.

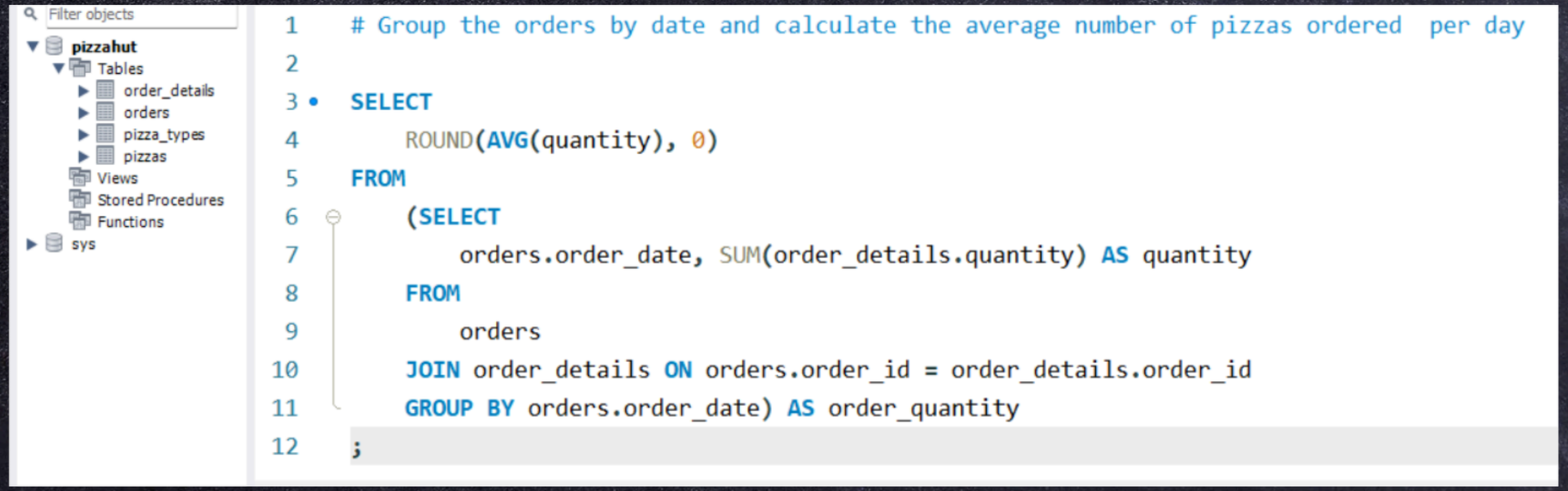



```sql
# Join relevant tables to find the category wise distribution of pizzas

SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category
;
```

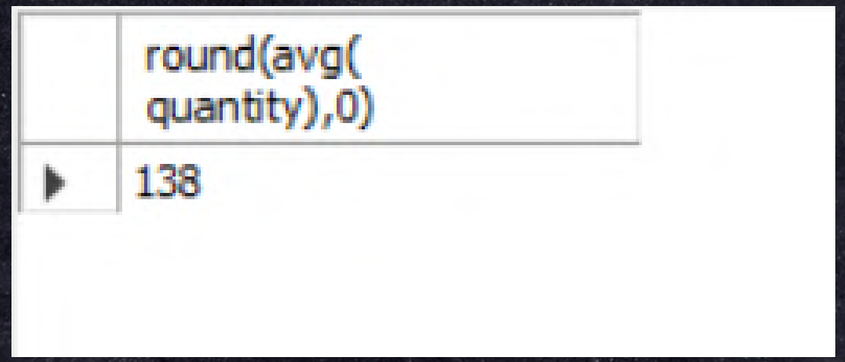
category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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



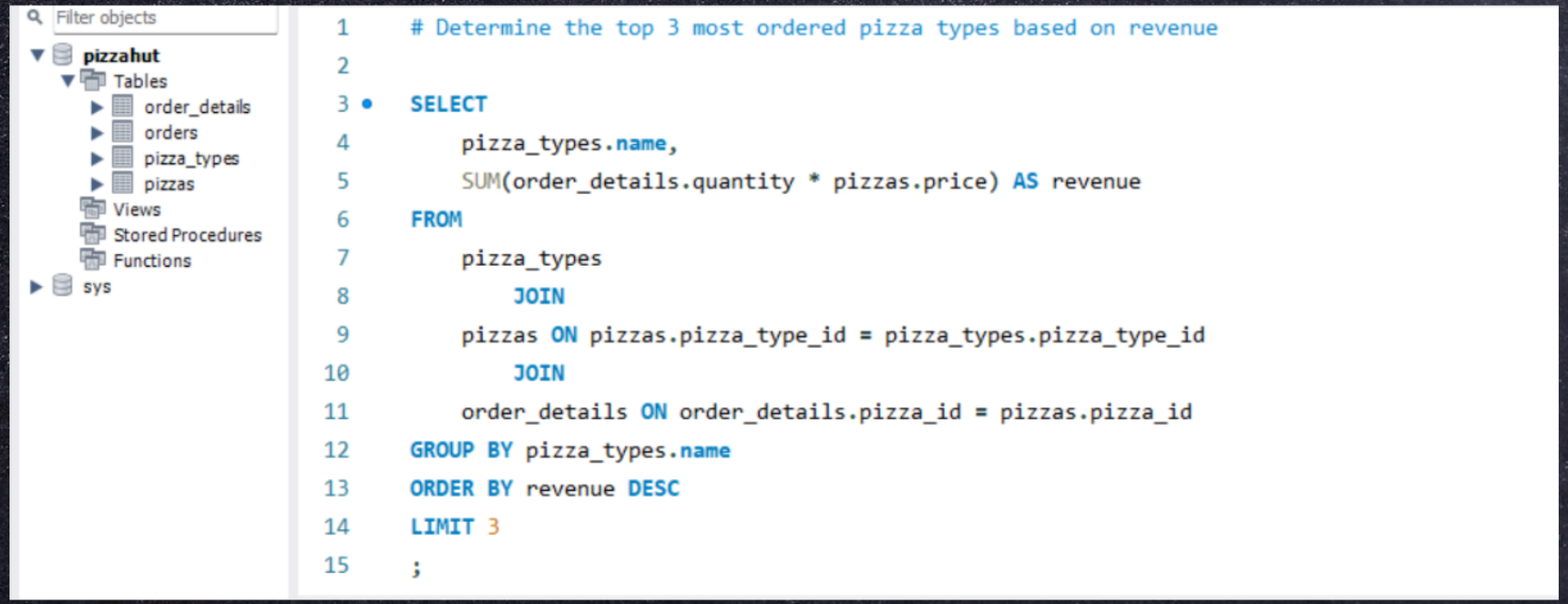

```sql
# Group the orders by date and calculate the average number of pizzas ordered  per day

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



round(avg(quantity),0)
138

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

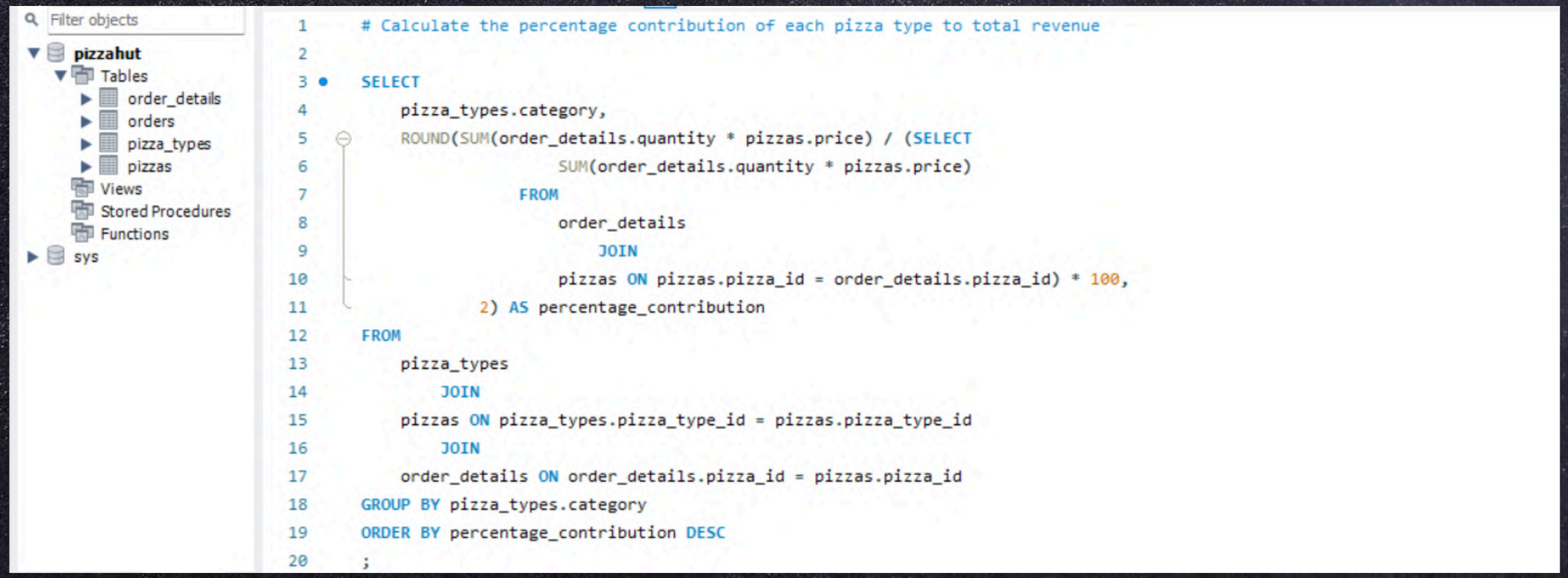



```sql
# Determine the top 3 most ordered pizza types based on revenue

SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON 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
;
```

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.




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

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

category	percentage_contribution
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Result 1