Akanksha Pandey

PizzaHut Sales Analysis Using SQL

MySQL

Project

Project Overview

Objective: To analyze PizzaHut's sales data using SQL to gain insights on performance, customer preferences, and revenue.

Tools: SQL(MySQL), Excel, CSV files

Focus Areas:-

Order volume and timing
Best-selling pizzas
Revenue patterns
Pizza category breakdowns

Akanksha Pandey

Dataset Summary

orders.csv – order_id, order_date, order_time
order_details.csv – order_details, order_id, pizza_id, quantity
pizzas.csv – pizza_id, pizza_type_id, size, price
pizza_types.csv – pizza_type_id, name, category, ingredients

PIZZAHUT SALES ANALYSIS

Basic SQL Analysis

Akanksha Pandey

- ✅Total Orders Placed: Query using COUNT(*) on orders
- ● 📸 Total Revenue: SUM(quantity * price) after joins
- ▯ Highest Priced Pizza: ORDER BY price DESC LIMIT 1
- ■ 📢 Most Common Size: GROUP BY size with max count
- ☆ Top 5 Most Ordered Pizzas:
- Joined order details with pizza types
- Ranked by quantity ordered

Intermediate SQL Analysis

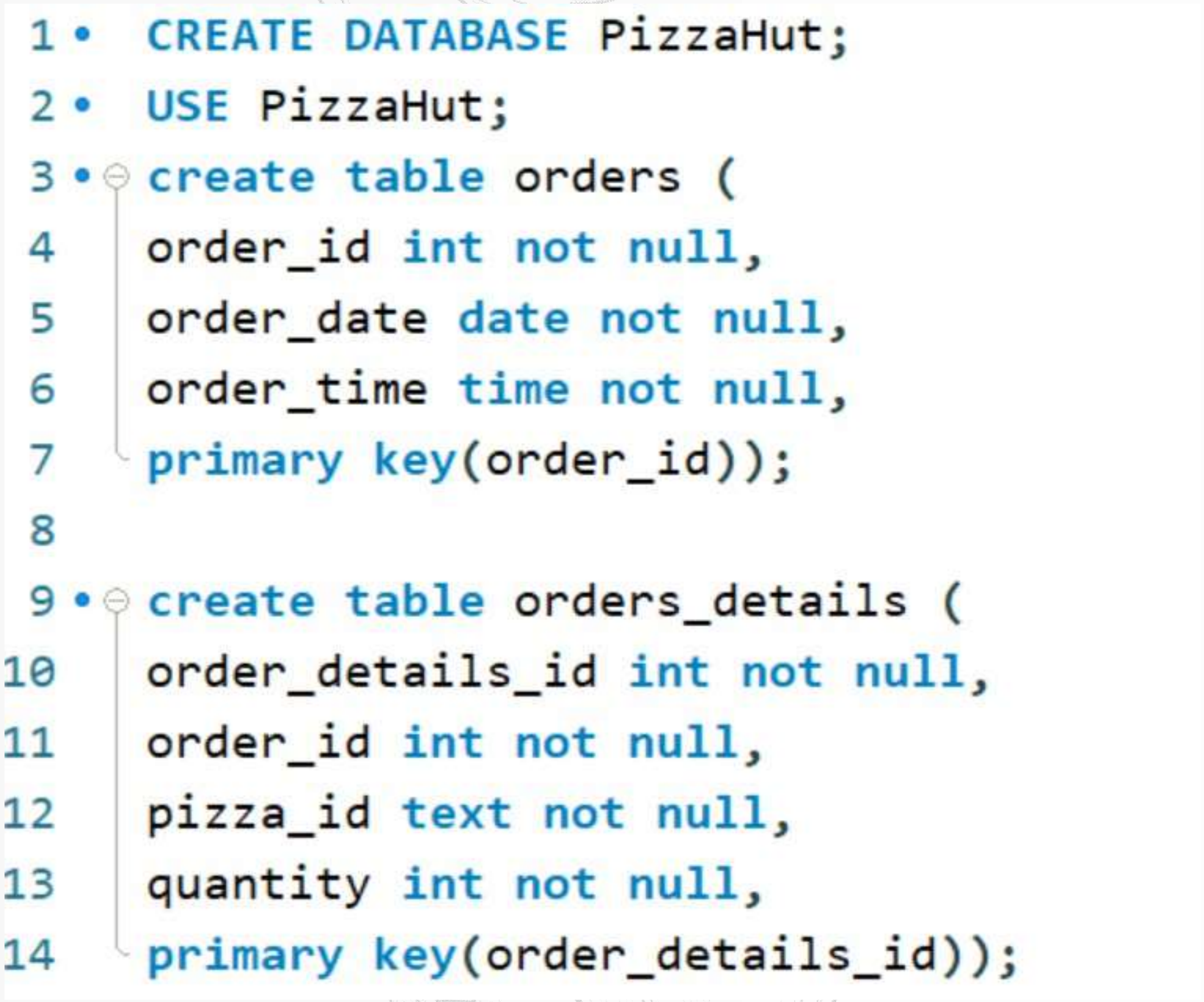
- ● 🕸 Total Quantity by Category: JOIN & GROUP BY category
- ▯Order Distribution by Hour: EXTRACT(HOUR FROM time)
- ● 🛎 Category-wise Pizza Distribution: JOIN pizza tables and group
- ● (Avg Pizzas Per Day: GROUP BY date → AVG(quantity)
- ● 🎹 Top 3 Revenue-Generating Pizzas:
- SUM(quantity * price) → ORDER BY revenue DESC LIMIT 3

Advanced SQL Analysis

- ● 📱 % Contribution to Revenue (Per Pizza):
- (Pizza Revenue / Total Revenue) * 100
- ● 📠 Cumulative Revenue Over Time:
- ORDER BY date + SUM(...) OVER(ORDER BY date)
- ■ 🕻 Top 3 Pizzas by Revenue in Each Category:
- GROUP BY category and pizza → Window function or subquery for TOP 3

Akanksha Pandey

MySQL Screenshots



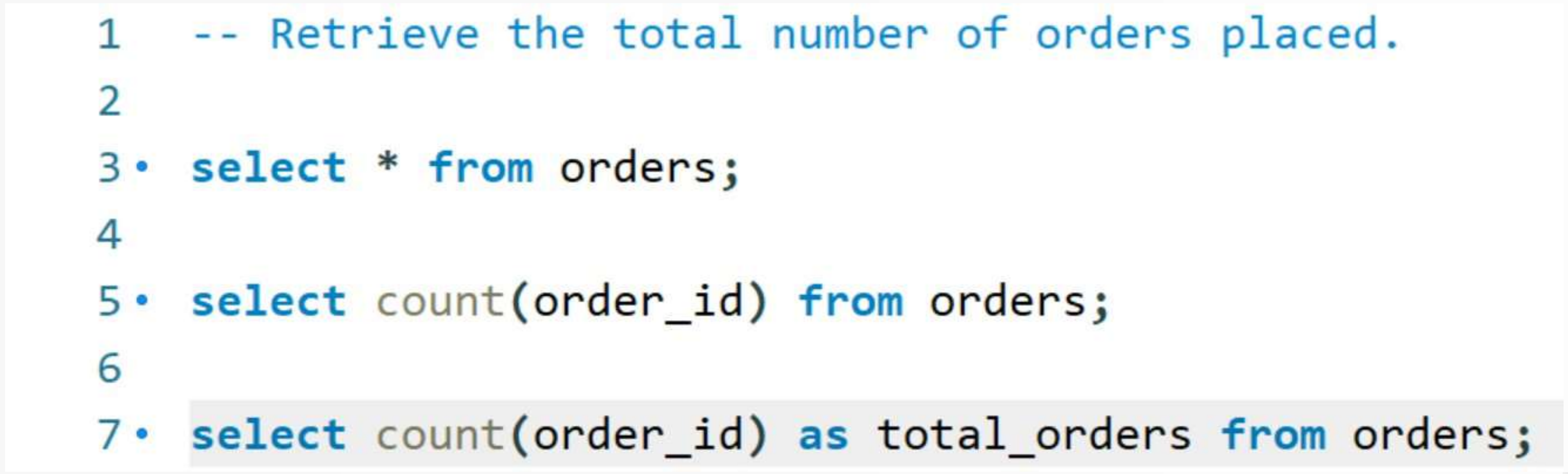
```sql
CREATE DATABASE PizzaHut;
USE PizzaHut;
create table orders (
order_id int not null,
order_date date not null,
order_time time not null,
primary key(order_id));

create table orders_details (
order_details_id int not null,
order_id int not null,
pizza_id text not null,
quantity int not null,
primary key(order_details_id));
```

Akanksha Pandey

PIZZAHUT SALES ANALYSIS

Retrieve the total number of orders placed.

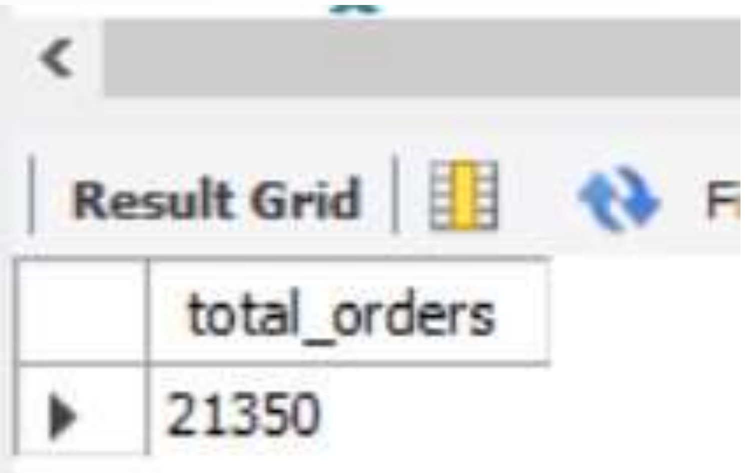


```sql
-- Retrieve the total number of orders placed.

select * from orders;

select count(order_id) from orders;

select count(order_id) as total_orders from orders;
```

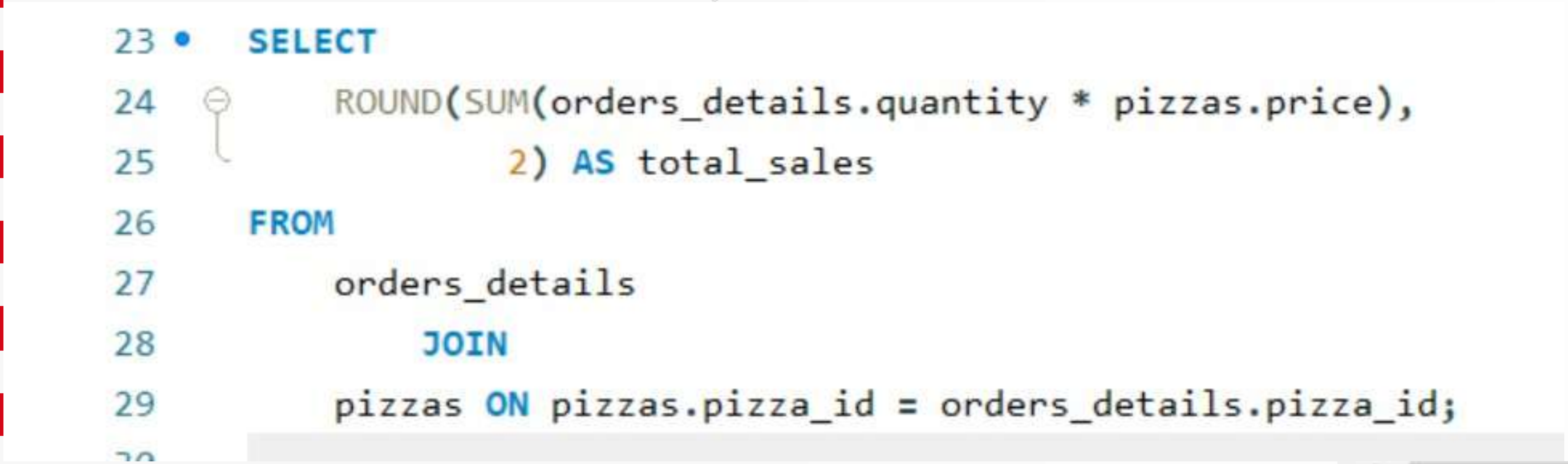


```
total_orders
21350
```

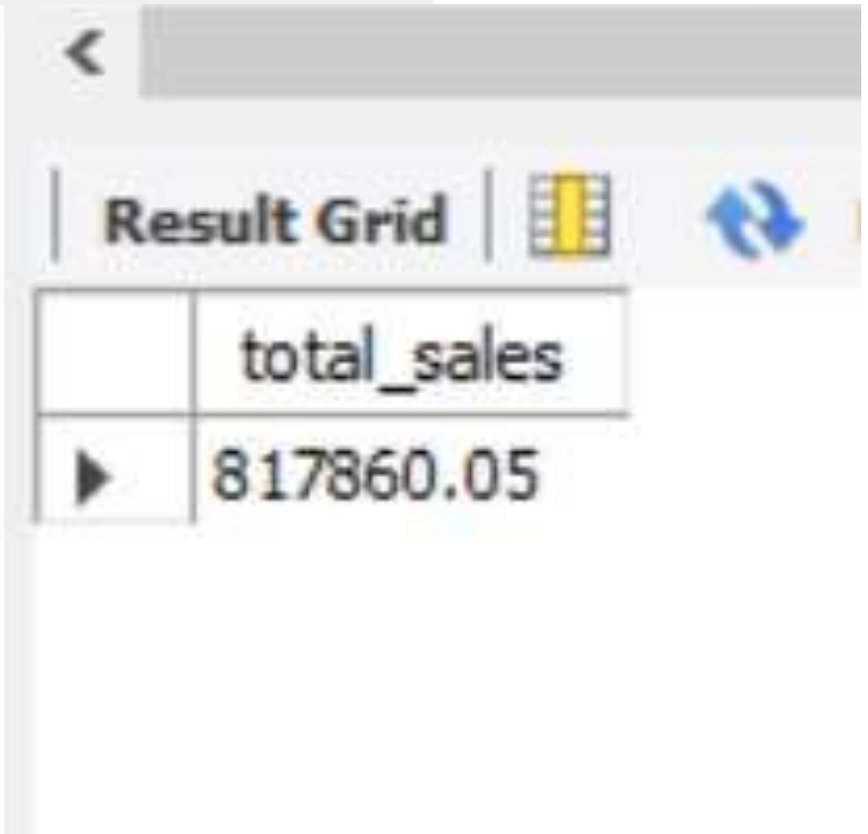
Akanksha Pandey

PIZZAHUT SALES ANALYSIS

Calculate the total revenue generated from pizza sales



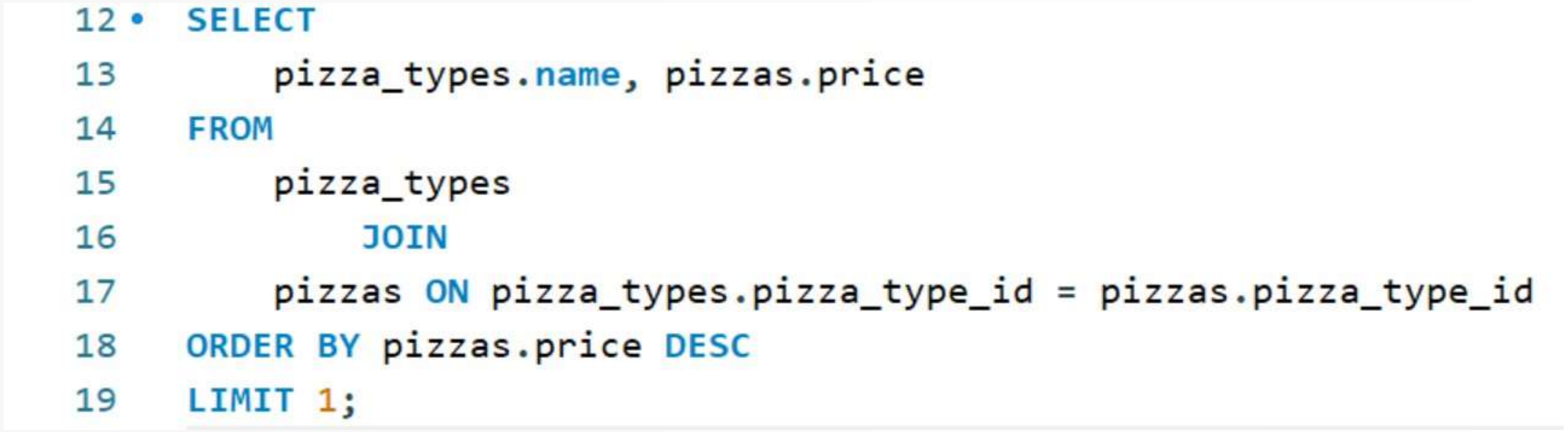
```
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```



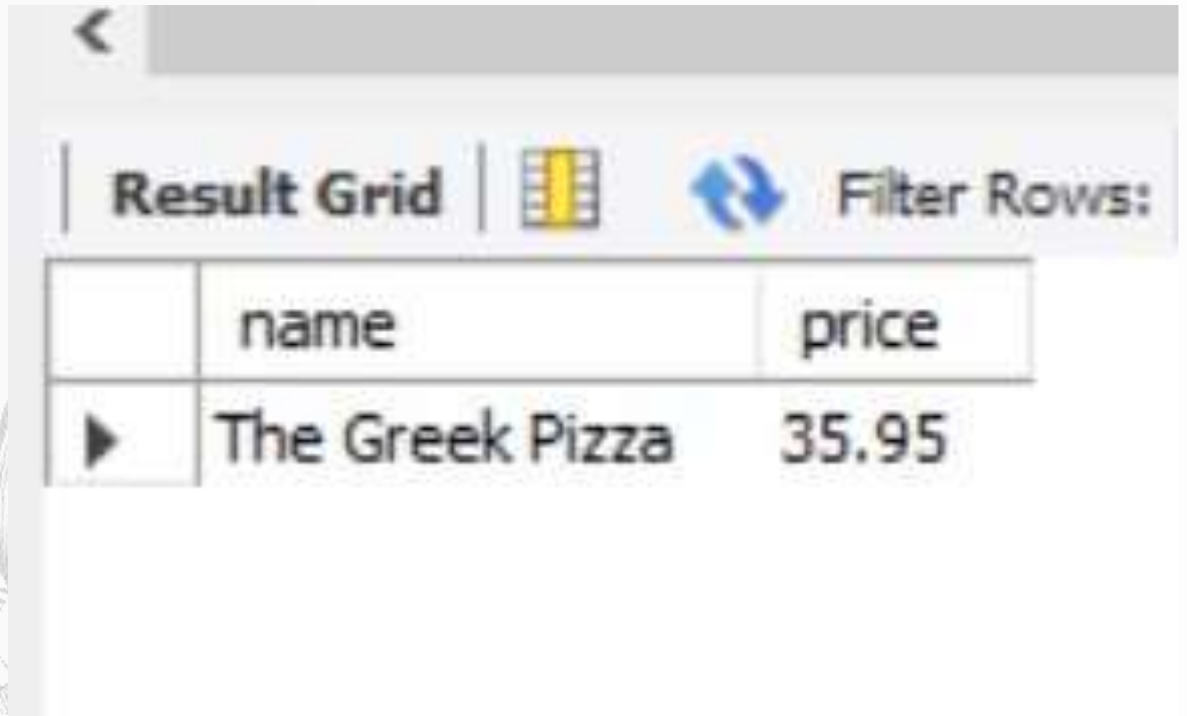
```
total_sales
817860.05
```

Akanksha Pandey

Identify the highest-priced pizza.



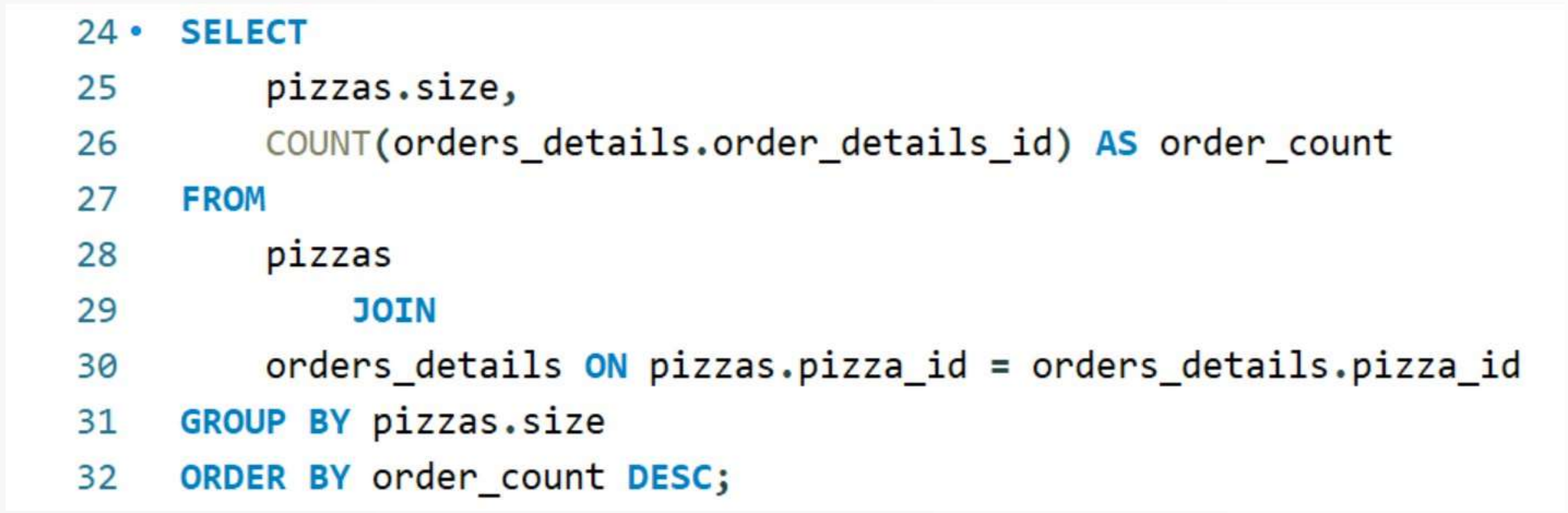
```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```



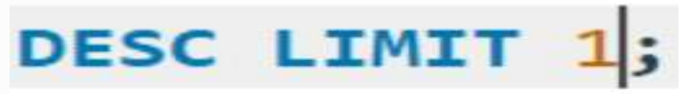
```
name             price
The Greek Pizza  35.95
```

Akanksha Pandey

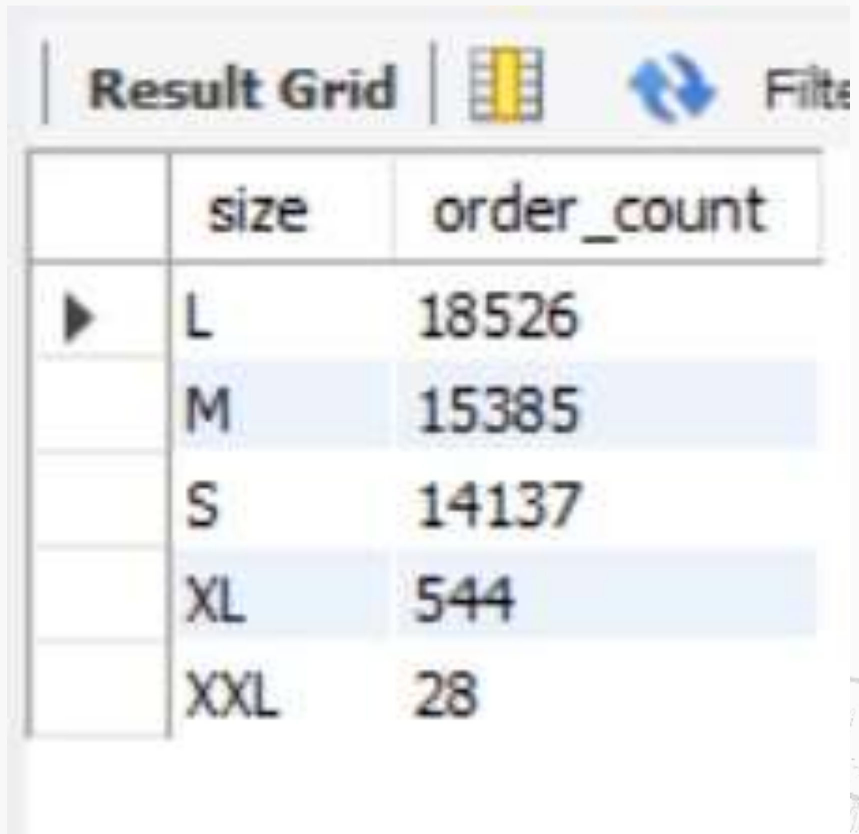
Identify the most common pizza size ordered.



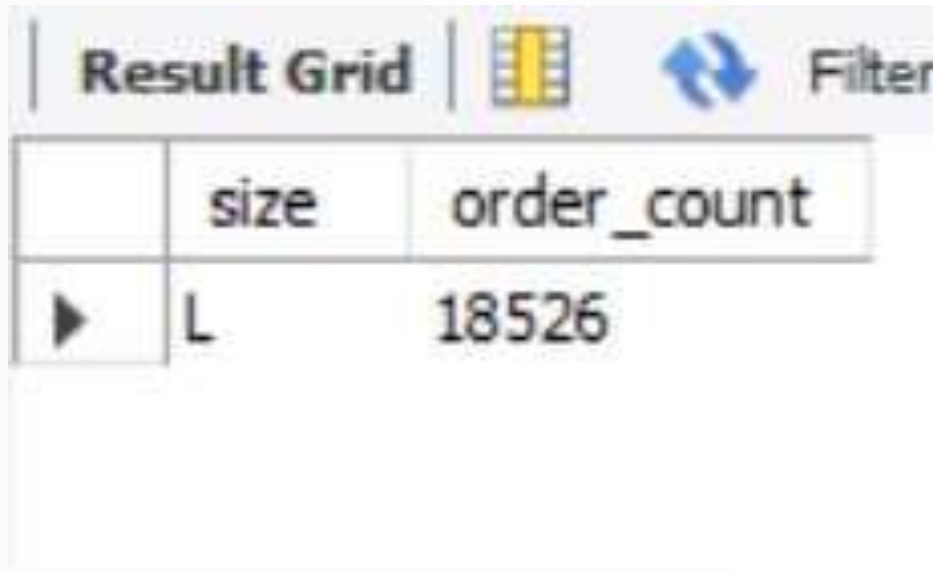
```
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```



```
DESC LIMIT 1;
```



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

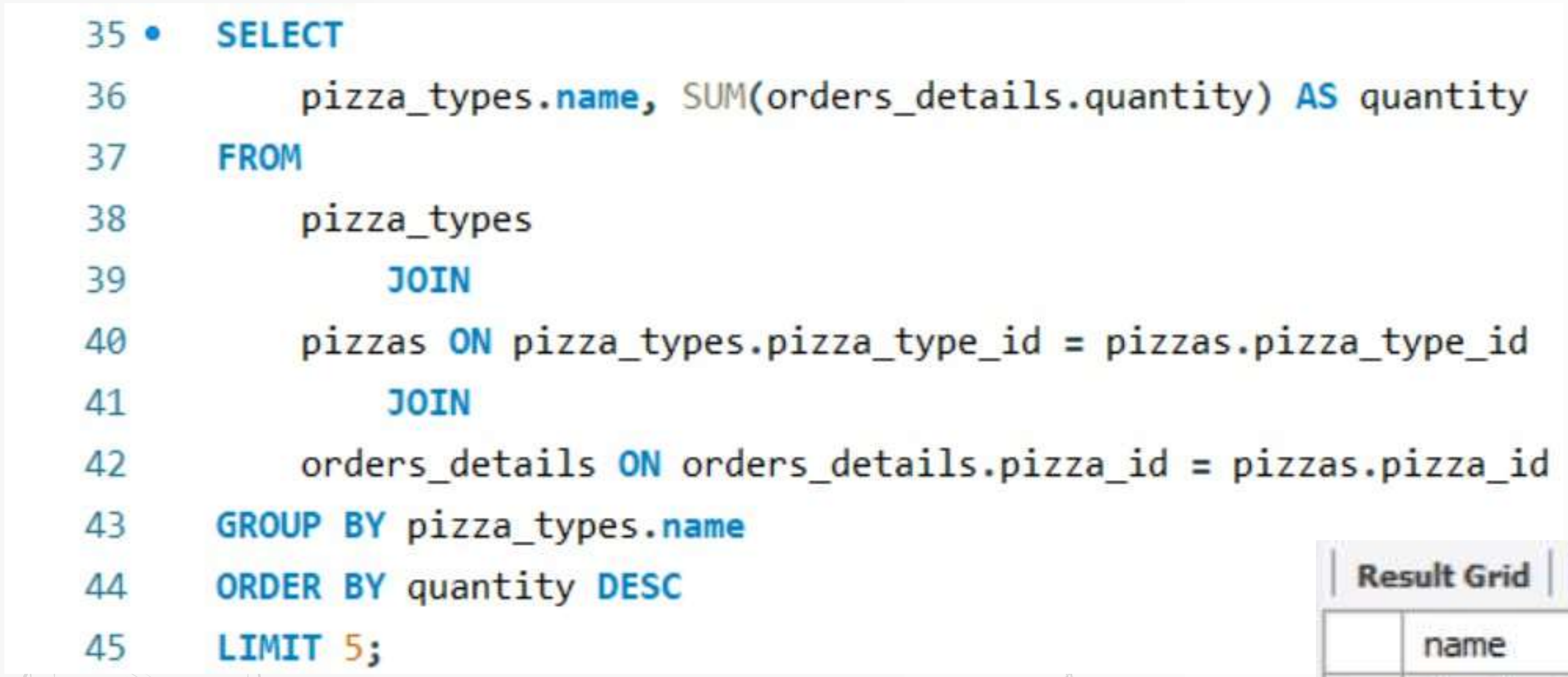


size	order_count
L	18526

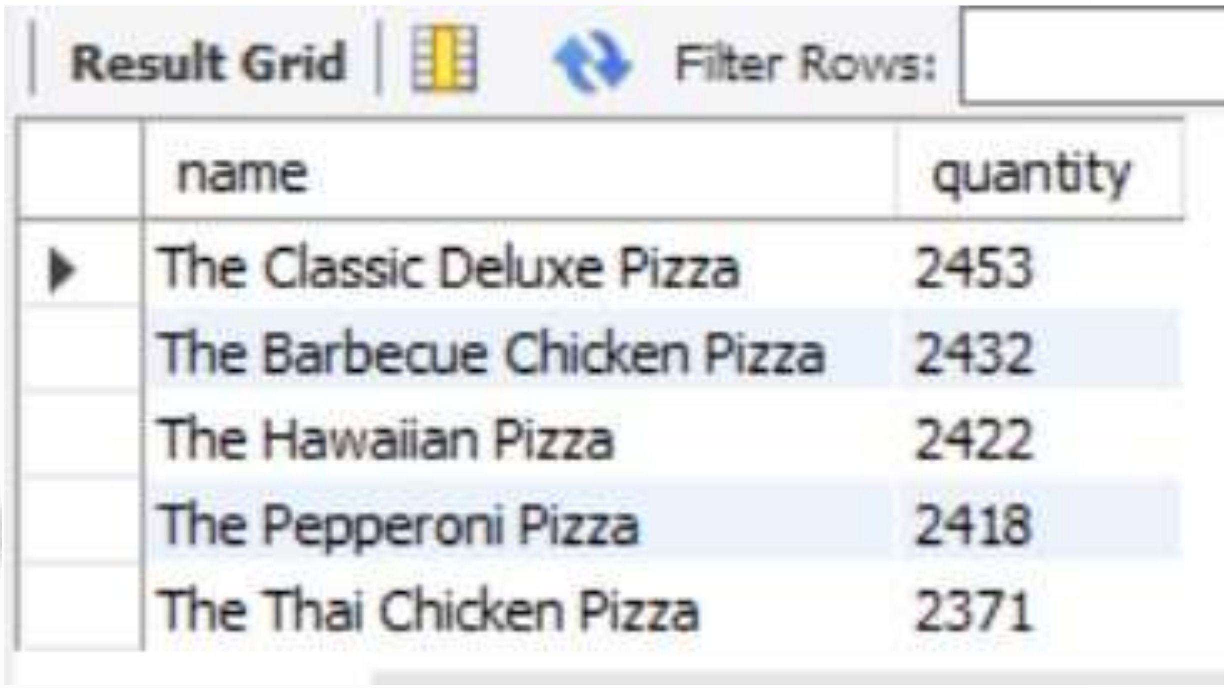
Akanksha Pandey

PIZZAHUT SALES ANALYSIS

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



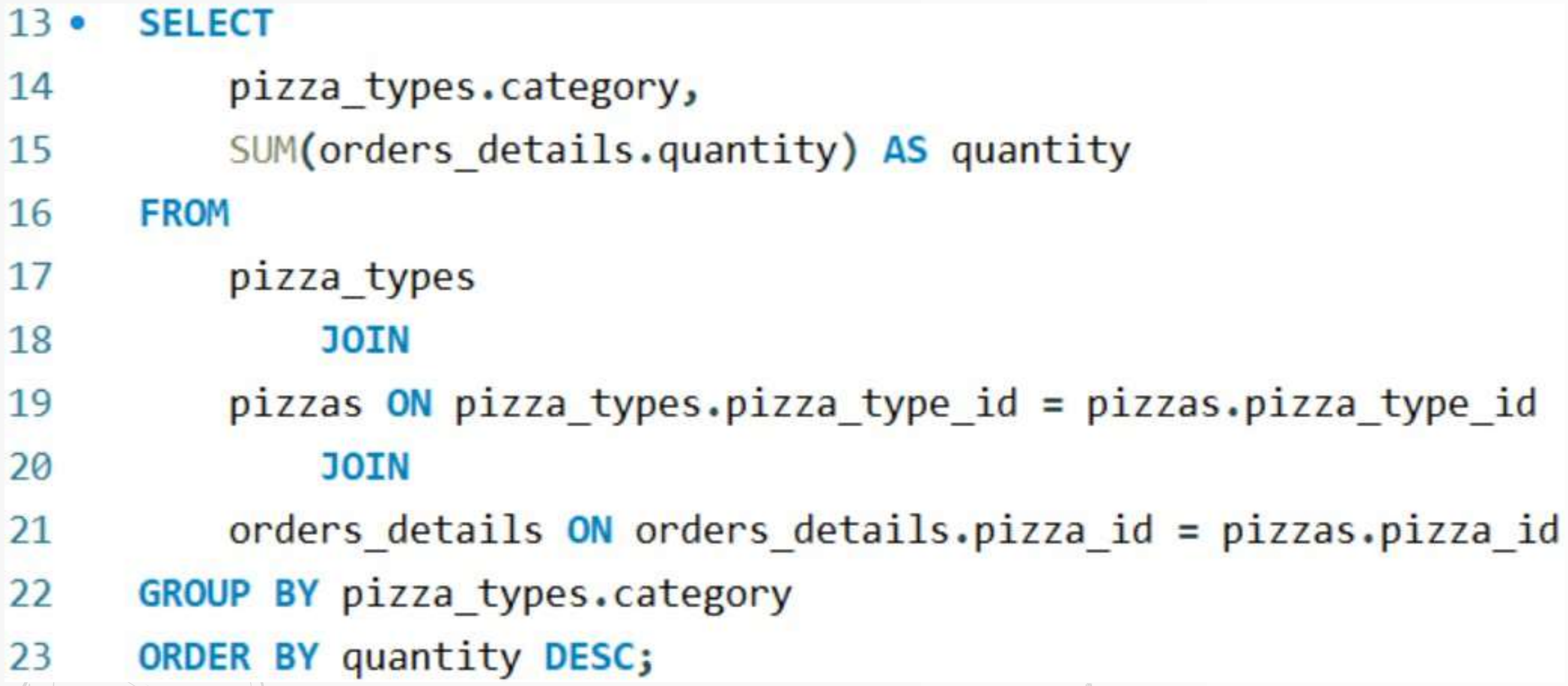
```
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT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



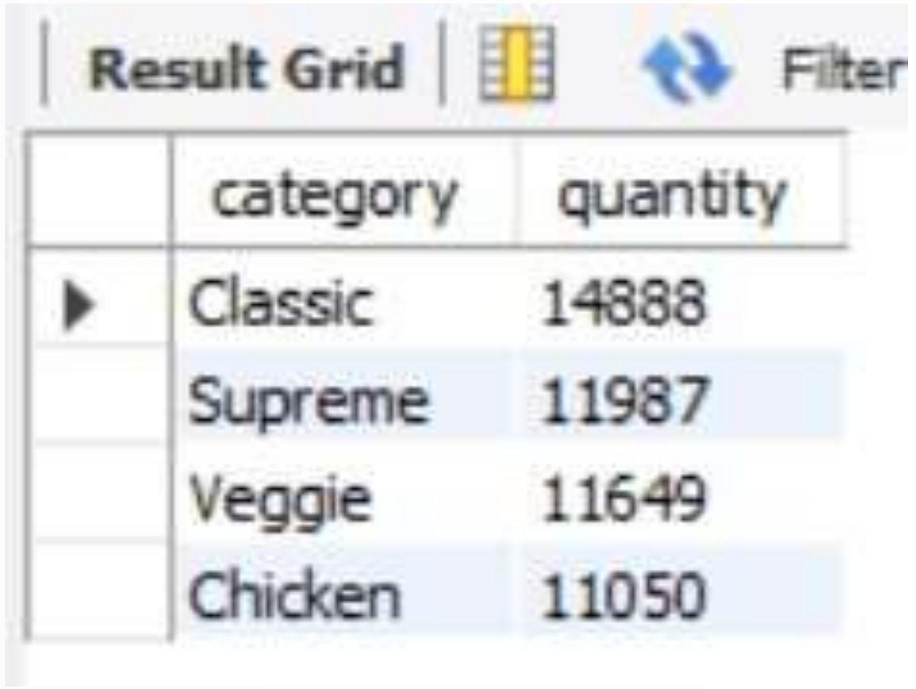
name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

PIZZAHUT SALES ANALYSIS

Join the necessary tables to find the total quantity of each pizza category ordered



```
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

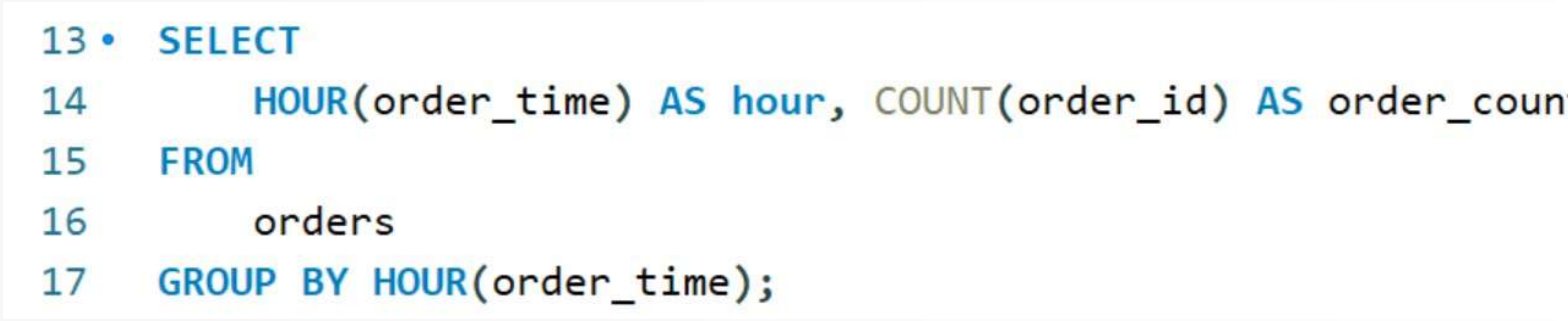


category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

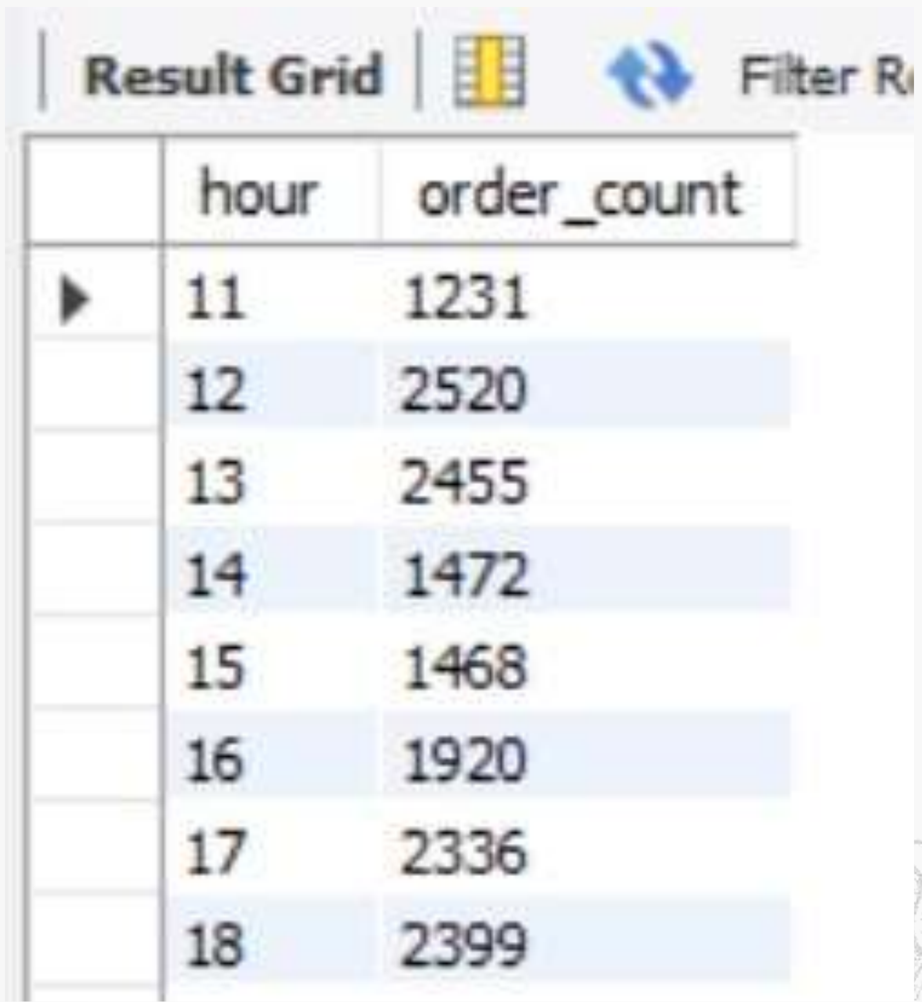
Akanksha Pandey

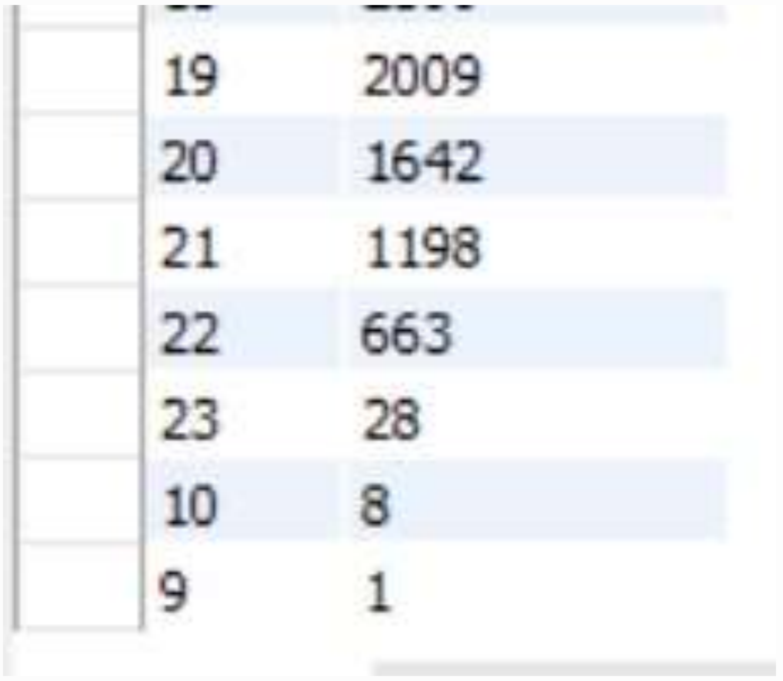
PIZZAHUT SALES ANALYSIS

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);
```



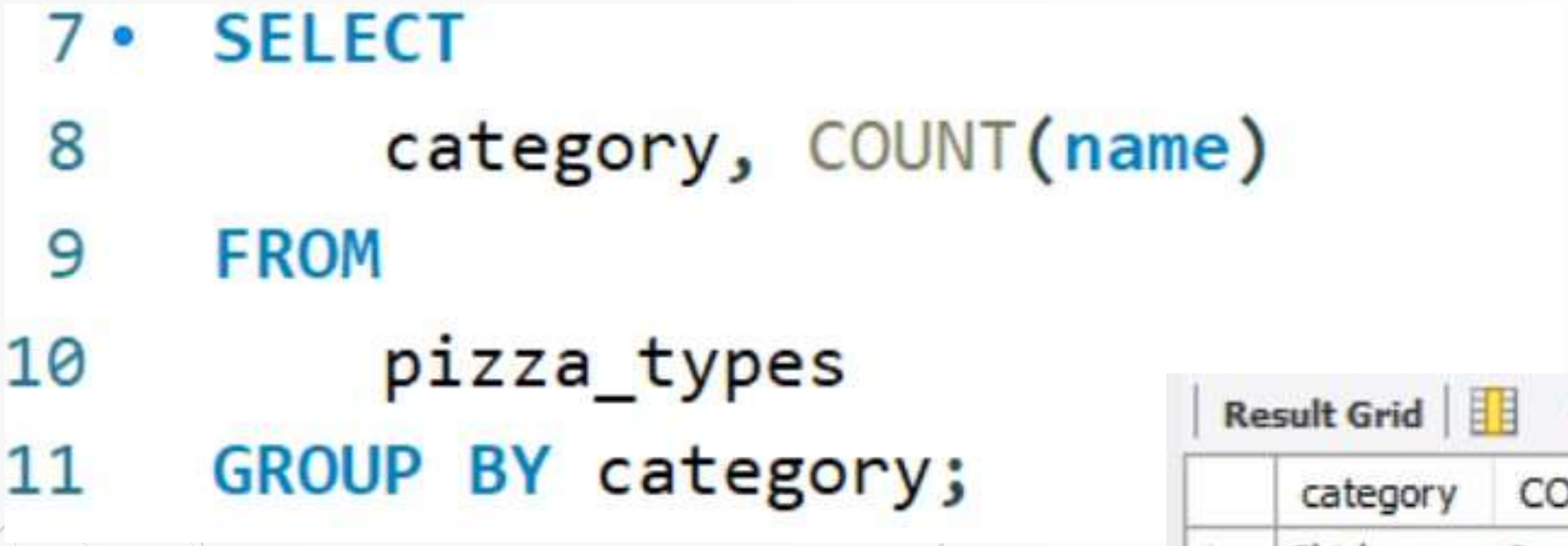


hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

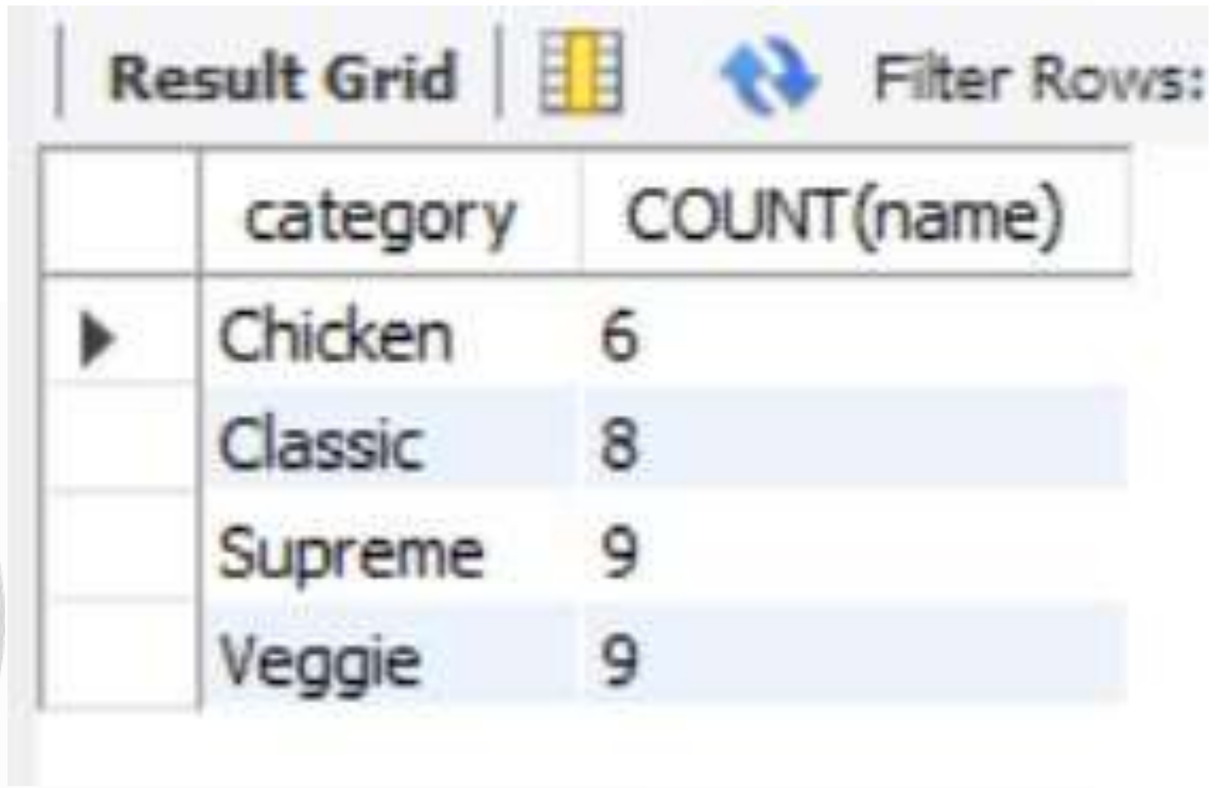
Akanksha Pandey

PIZZAHUT SALES ANALYSIS

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



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



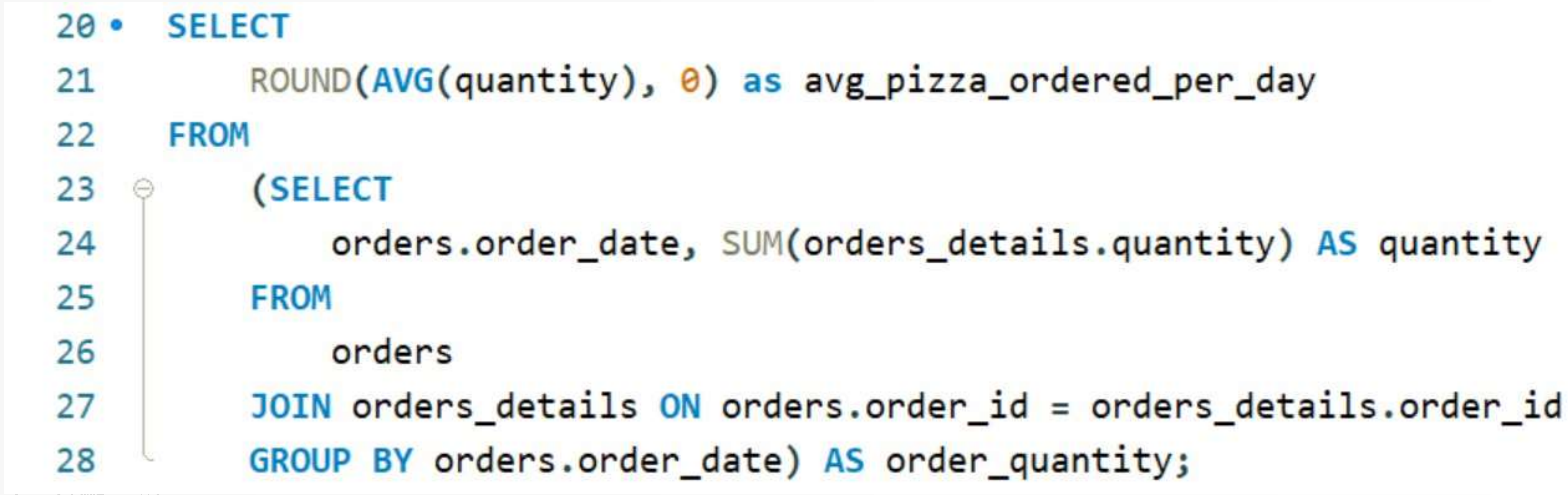
Result Grid | Filter Rows:

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

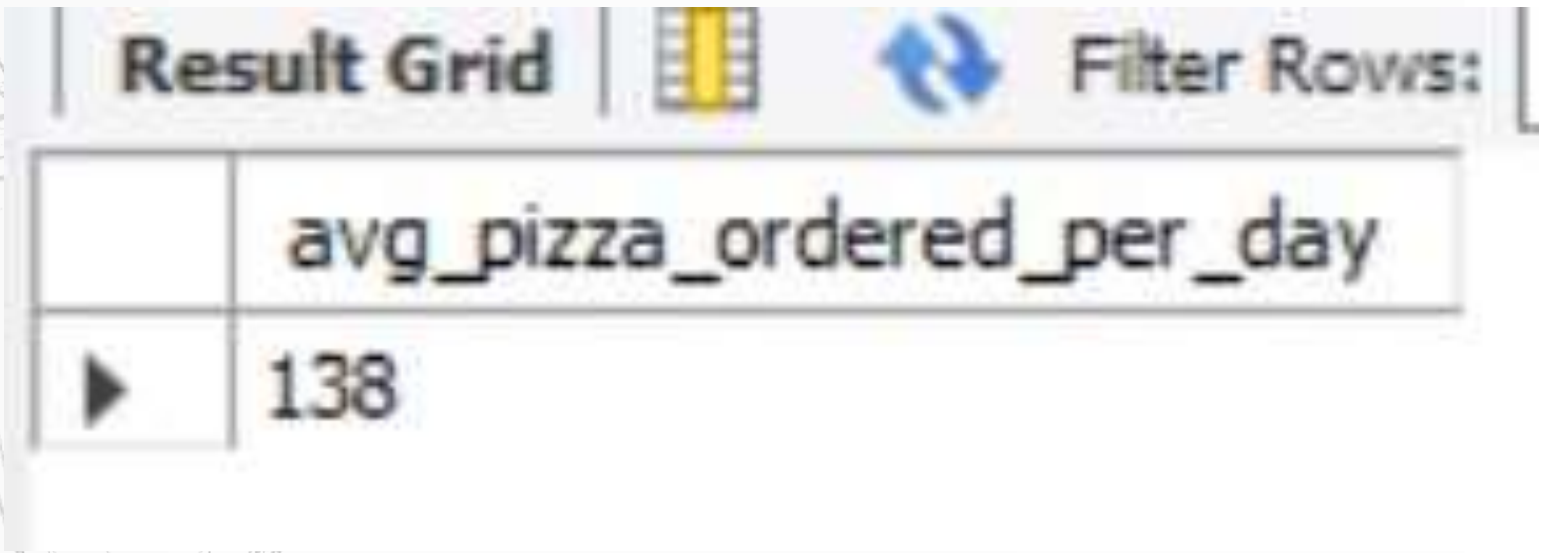
Akanksha Pandey

PIZZAHUT SALES ANALYSIS

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



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

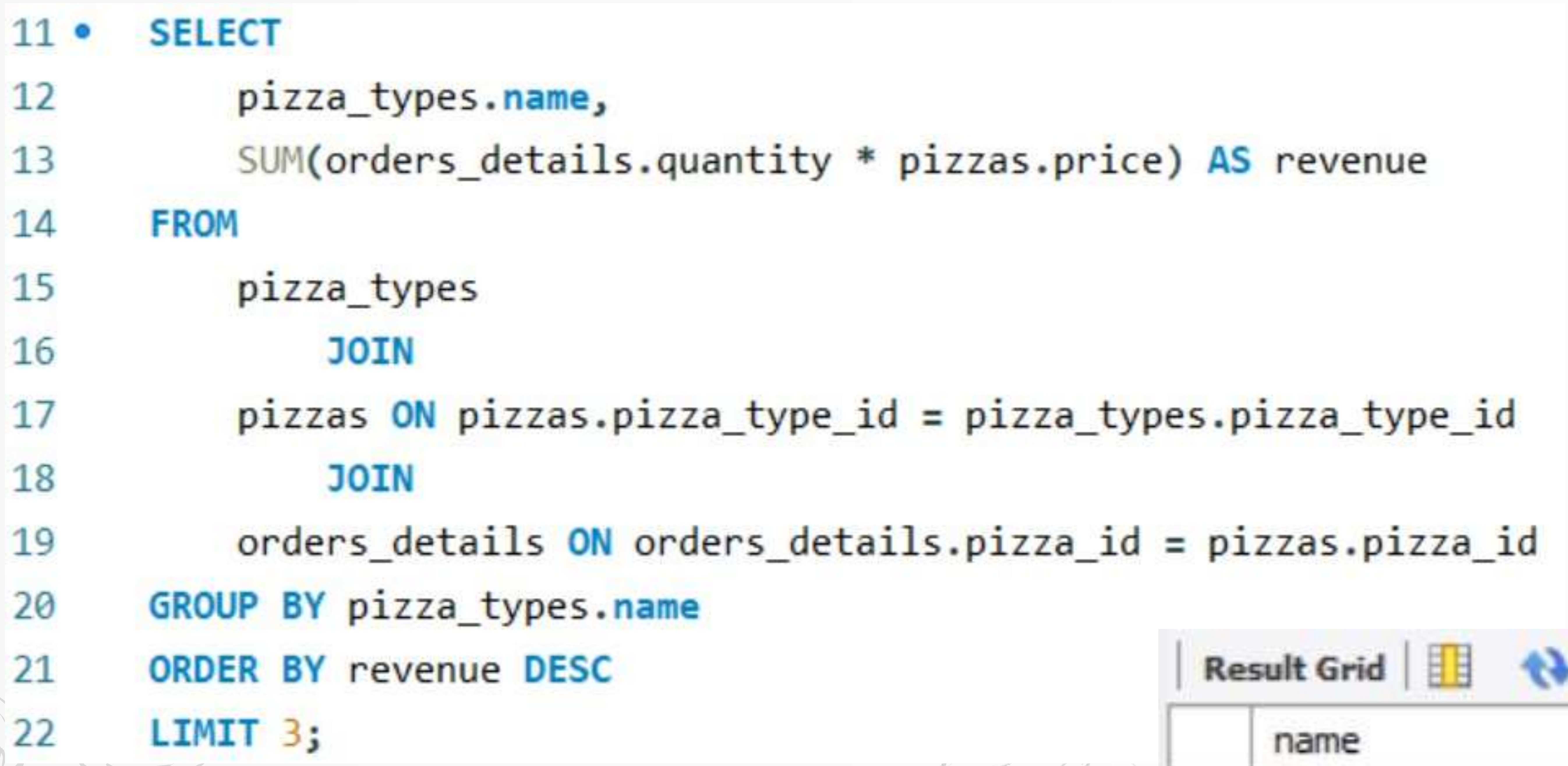


```
avg_pizza_ordered_per_day
138
```

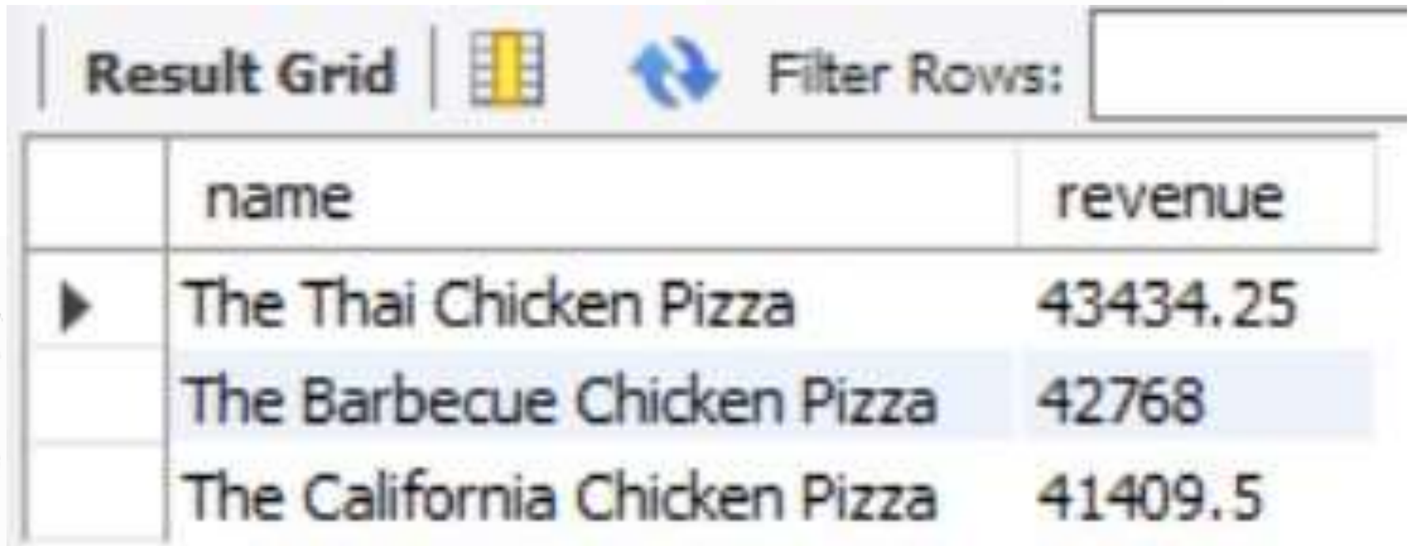
Akanksha Pandey

PIZZAHUT SALES ANALYSIS

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



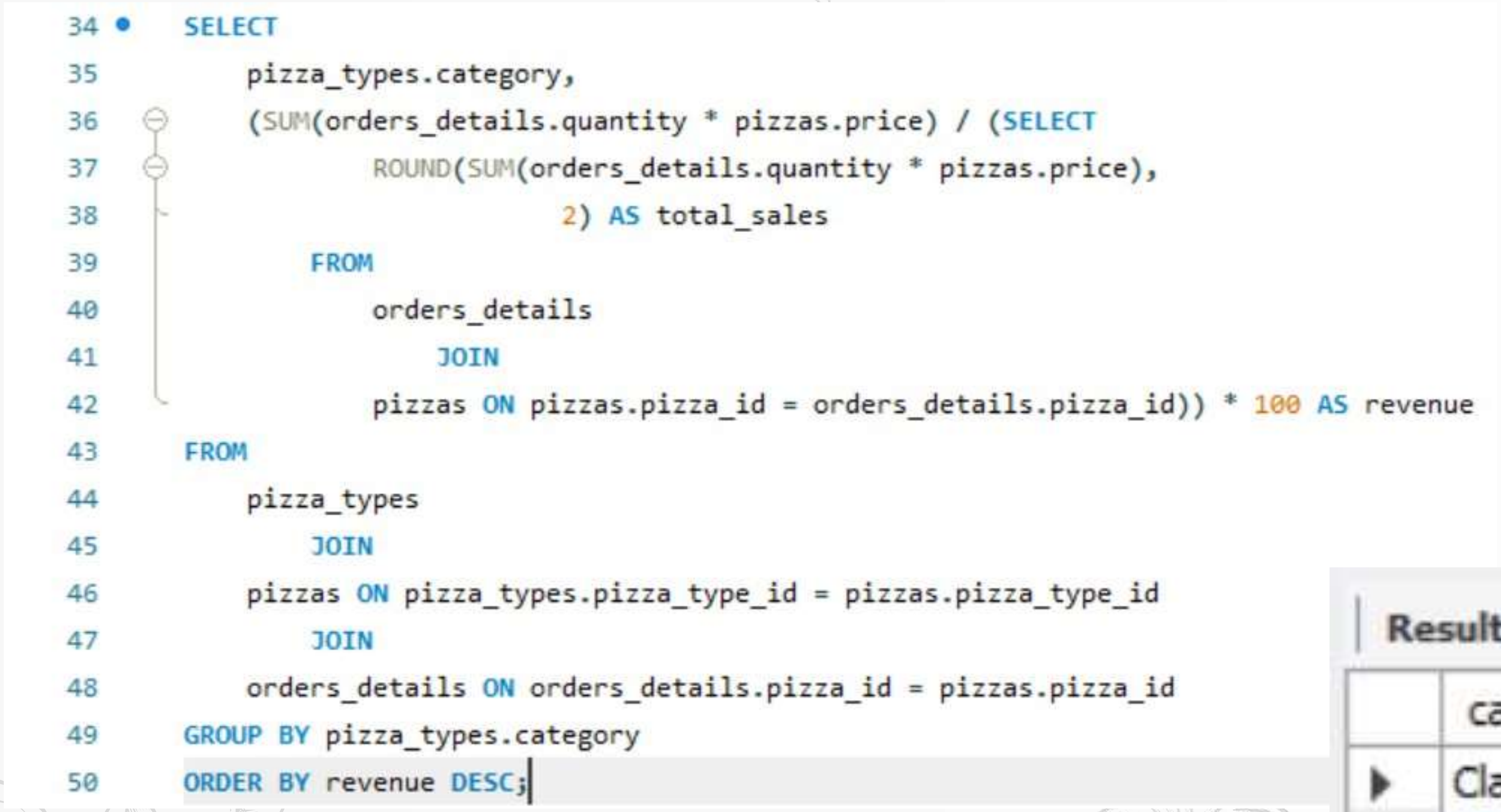
```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```



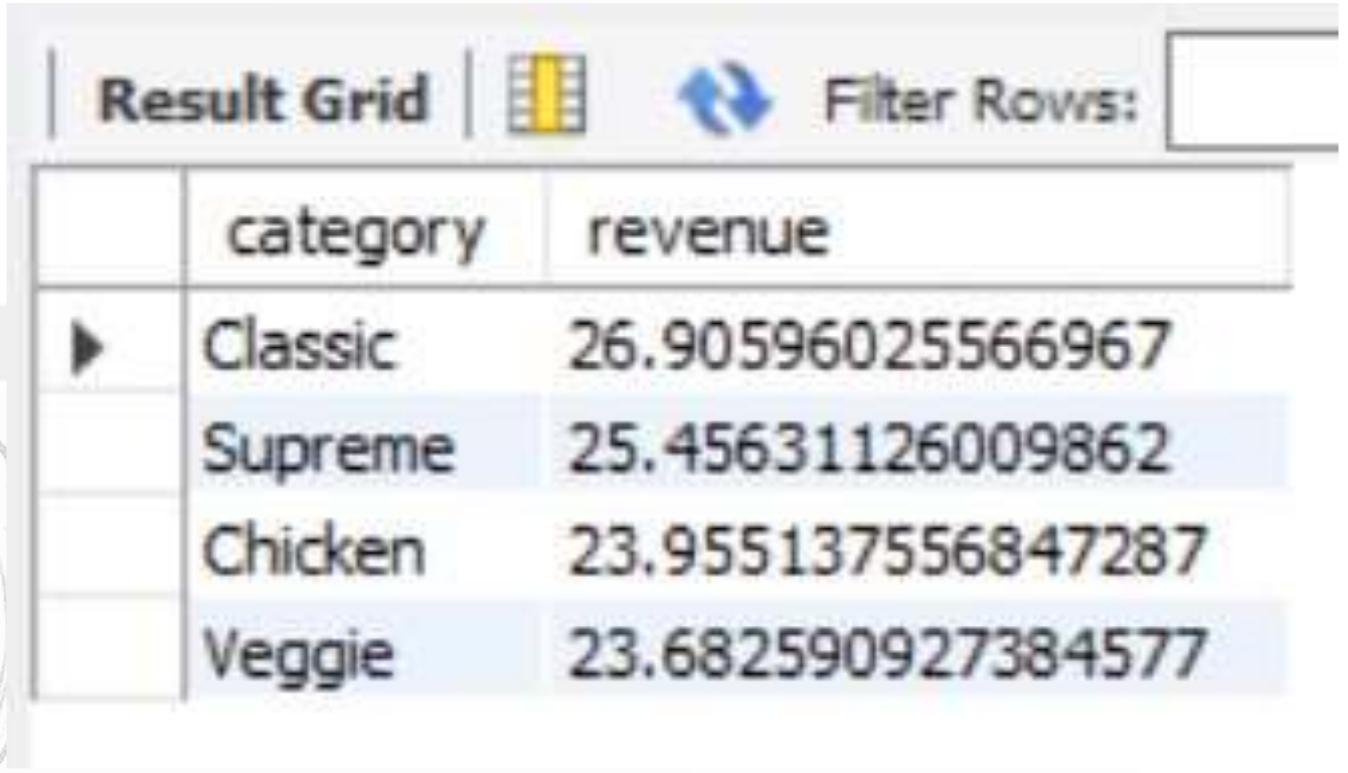
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Akanksha Pandey

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



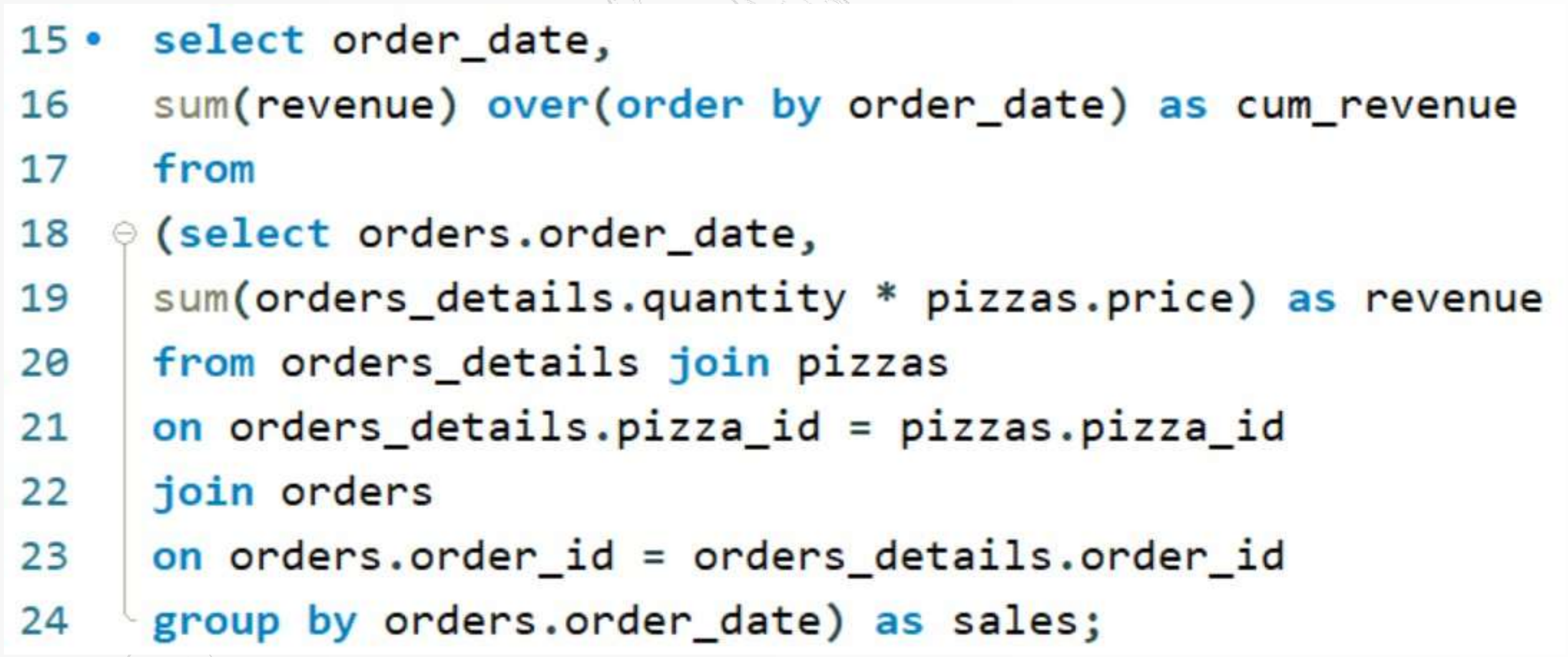
```sql
SELECT
    pizza_types.category,
    (SUM(orders_details.quantity * pizzas.price) / (SELECT
            ROUND(SUM(orders_details.quantity * pizzas.price),
                        2) AS total_sales
        FROM
            orders_details
                JOIN
            pizzas ON pizzas.pizza_id = orders_details.pizza_id)) * 100 AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```



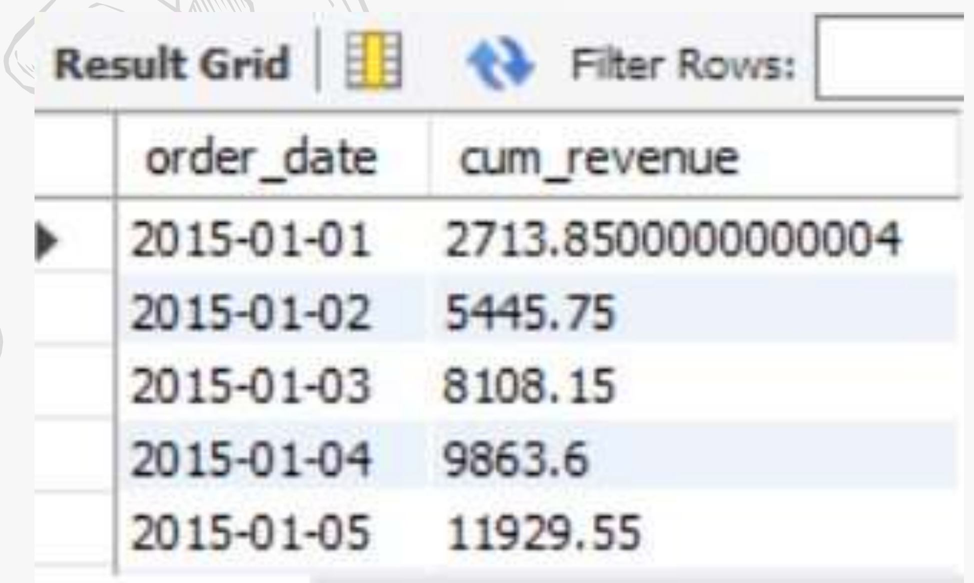
```
category   revenue
Classic    26.9059602556696 7
Supreme    25.4563112600986 2
Chicken    23.9551375568472 87
Veggie     23.6825909273845 77
```

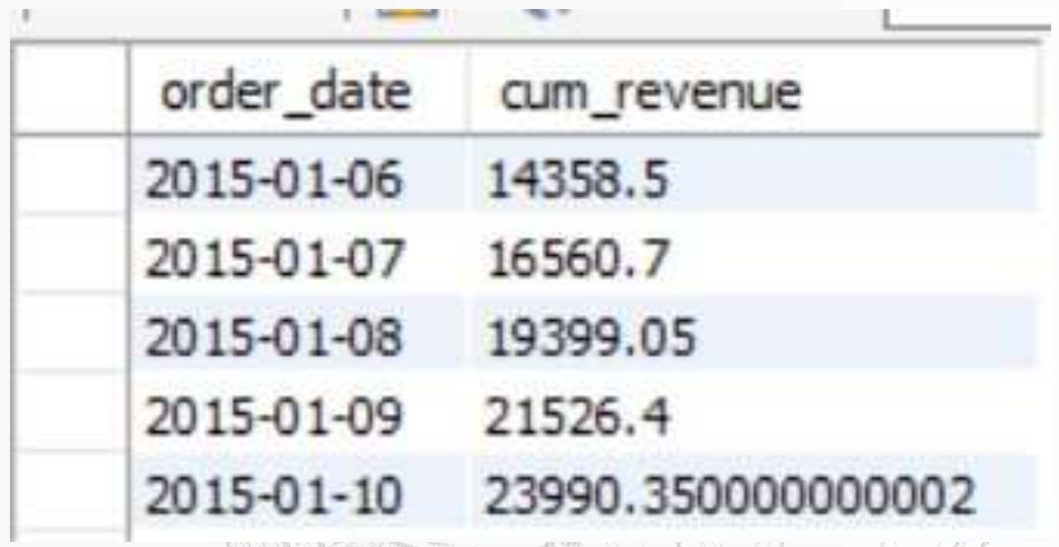
Akanksha Pandey

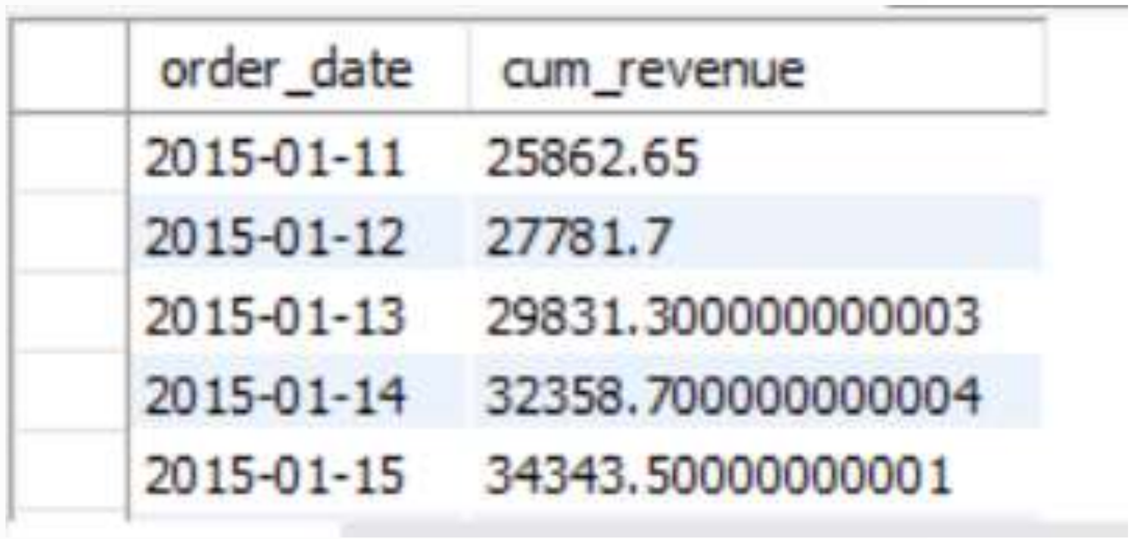
Analyze the cumulative revenue generated over time.



```sql
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,
sum(orders_details.quantity * pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = orders_details.order_id
group by orders.order_date) as sales;
```



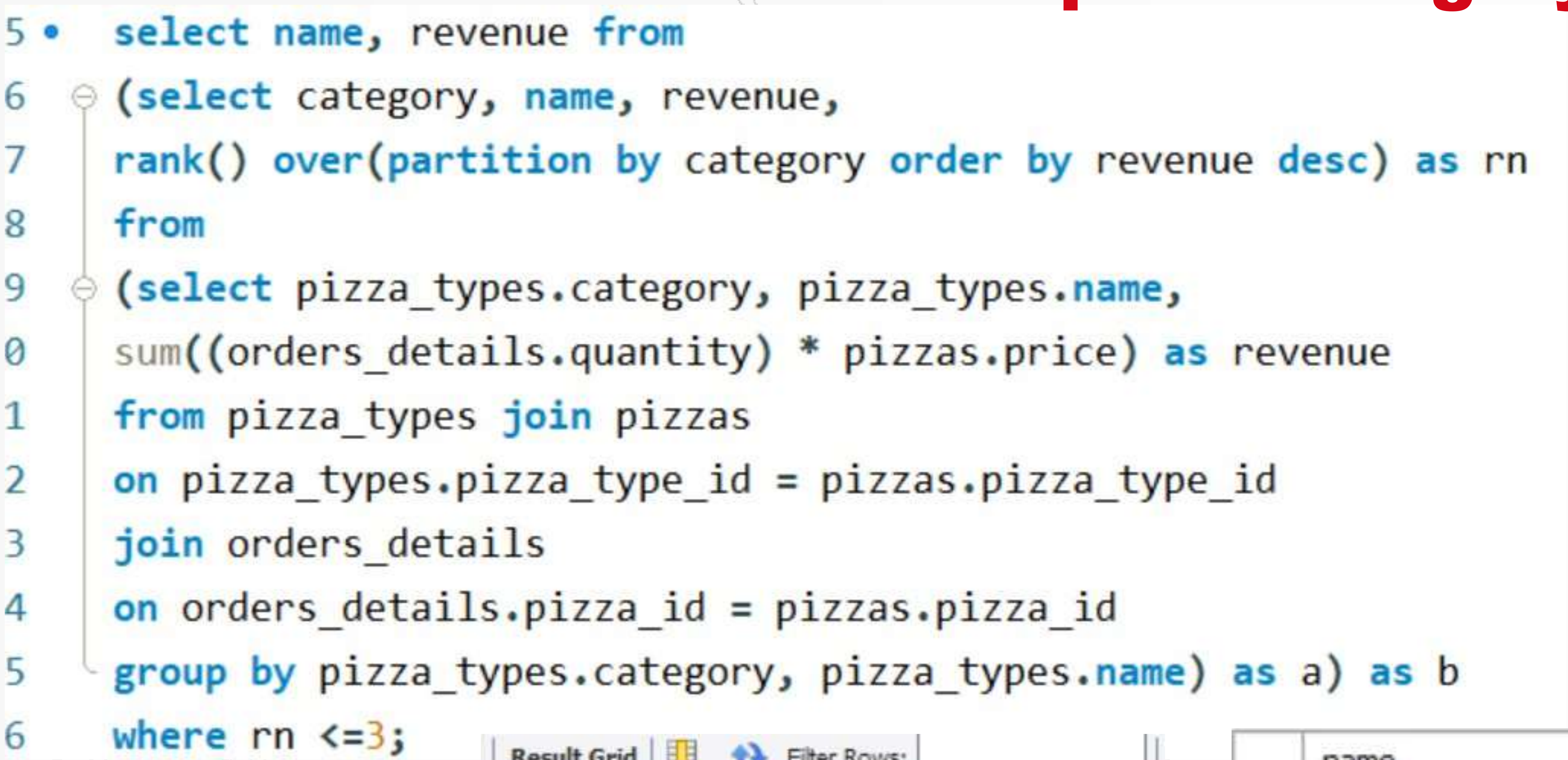




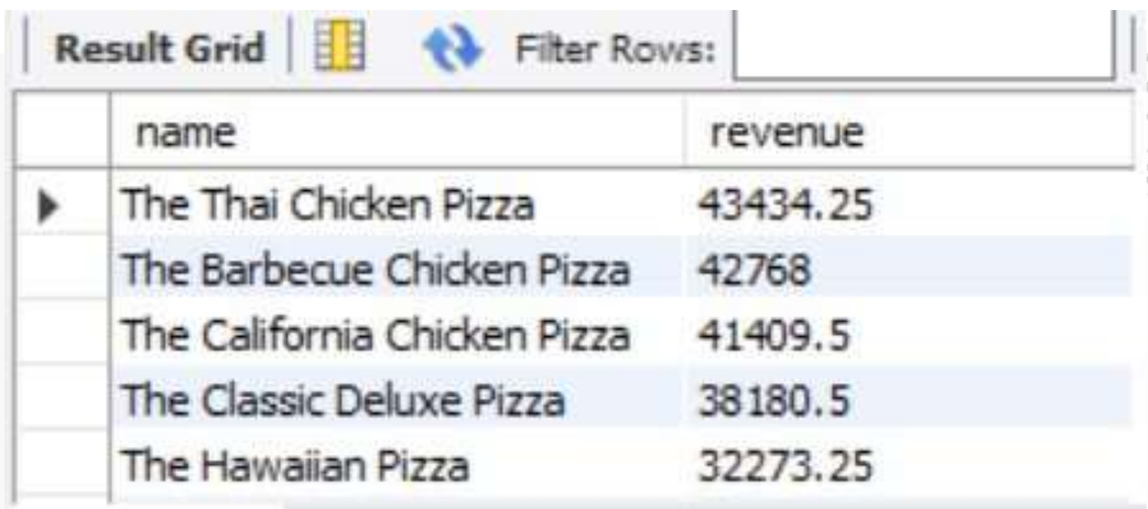
order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001

Akanksha Pandey

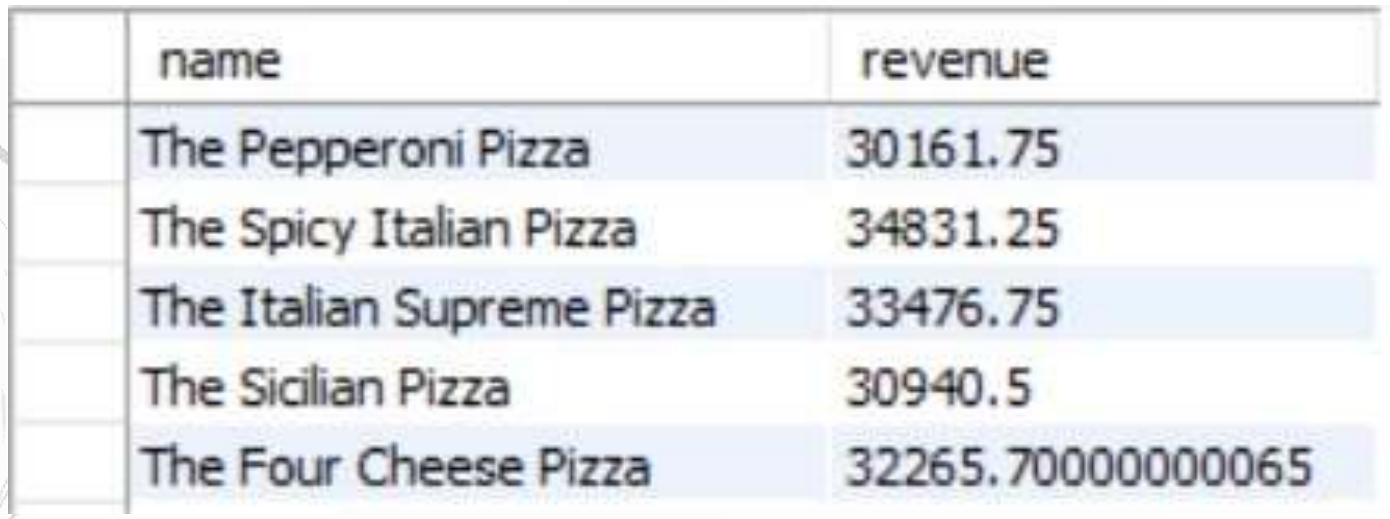
Determine the top 3 most ordered pizza types based on revenue for each pizza category.



```sql
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((orders_details.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_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:

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



name	revenue
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

Presented by: Akanksha Pandey I Data Analyst

Email: pandeyakanksha0002@gmail.com

“SQL projects demonstrate a data analyst’s ability to extract, manipulate, and analyze data from real-world databases. They showcase practical skills in deriving actionable insights through structured queries and data-driven thinking.”

Thank You!