welcome

Pizza Sales Analysis Project

by- Feroz Sayyed

Project Overview

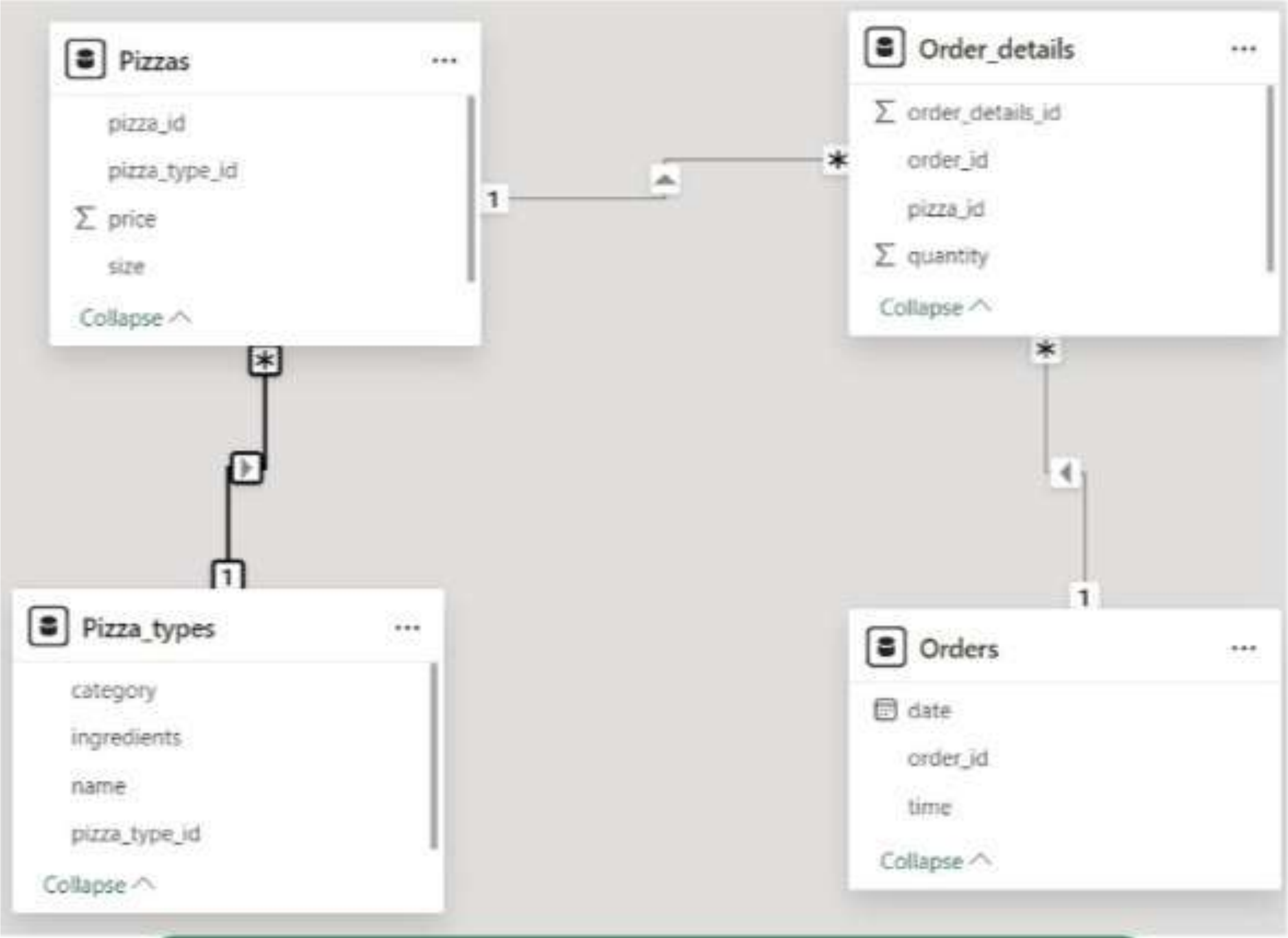
Objective:
To analyze pizza sales data to derive meaningful insights.

Data Source:
Extracted from CSV files into MySQL Workbench.

Tables Created:

- Orders
- Order Details
- Pizzas
- Pizza Types

Database Schema



Pizzas
- pizza_id
- pizza_type_id
- Σ price
- size

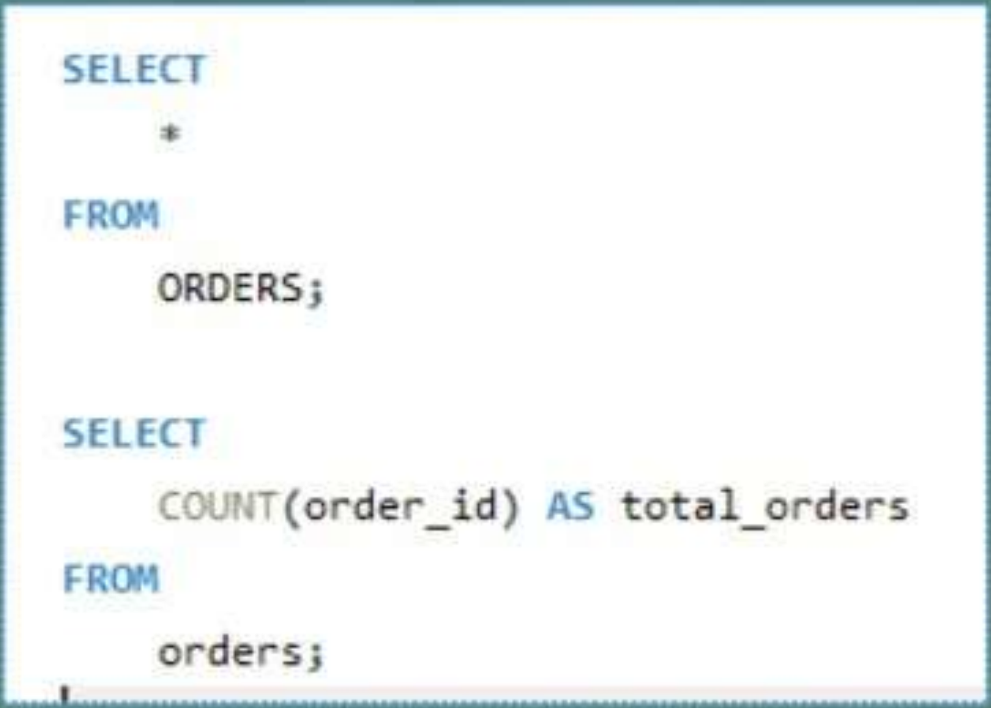
Order_details
- Σ order_details_id
- order_id
- pizza_id
- Σ quantity

Pizza_types
- category
- ingredients
- name
- pizza_type_id

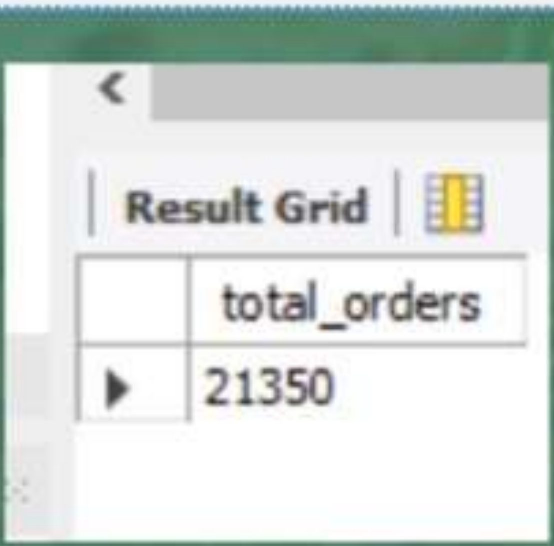
Orders
- date
- order_id
- time

Q1.

Retrieve the total number of orders placed.



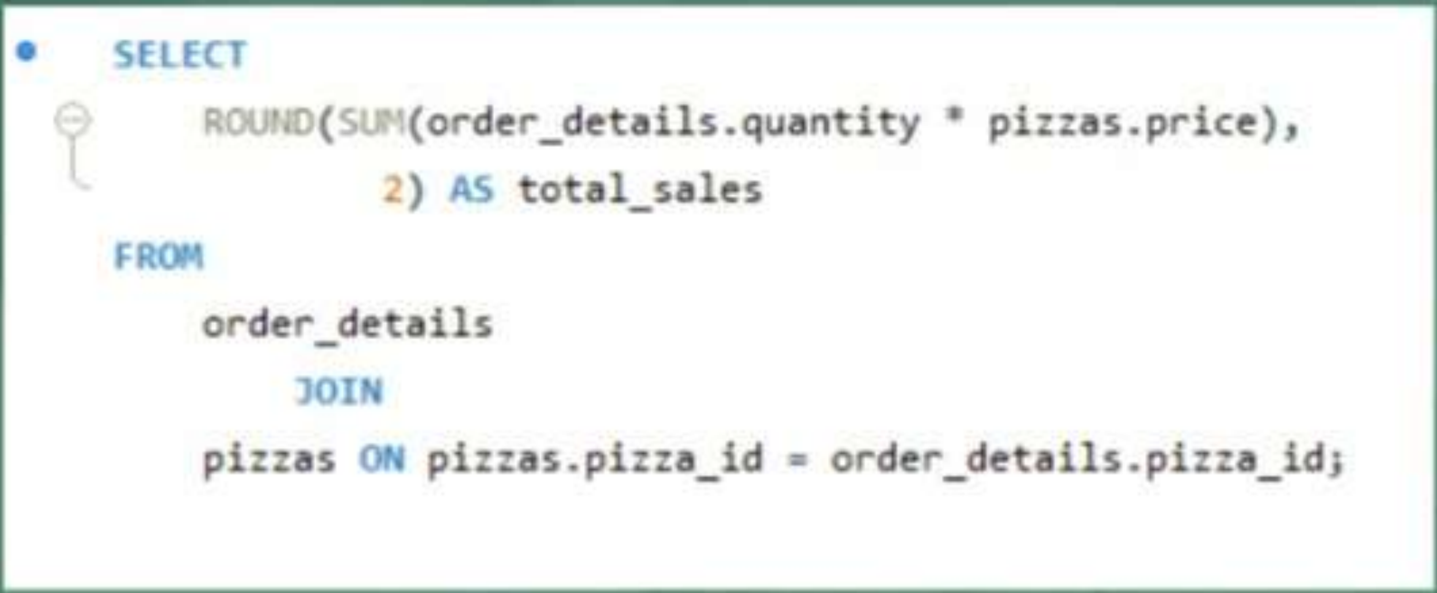
```
SELECT
    *
FROM
    ORDERS;


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



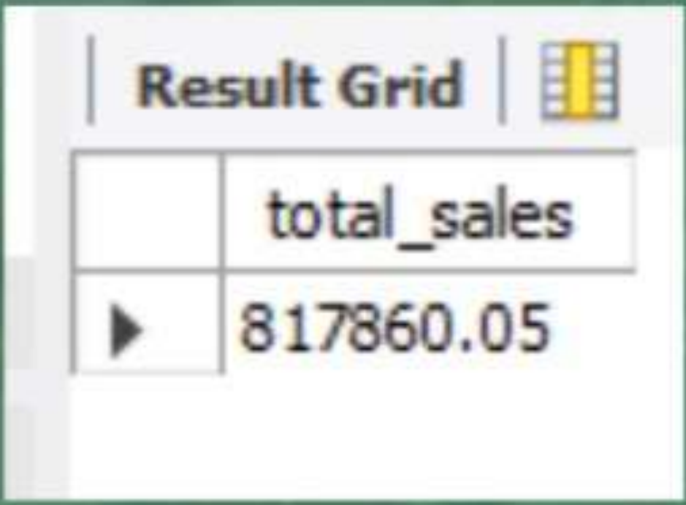
```
  total_orders
▶ 21350
```

Q2.

Calculate the total revenue generated from pizza sales.



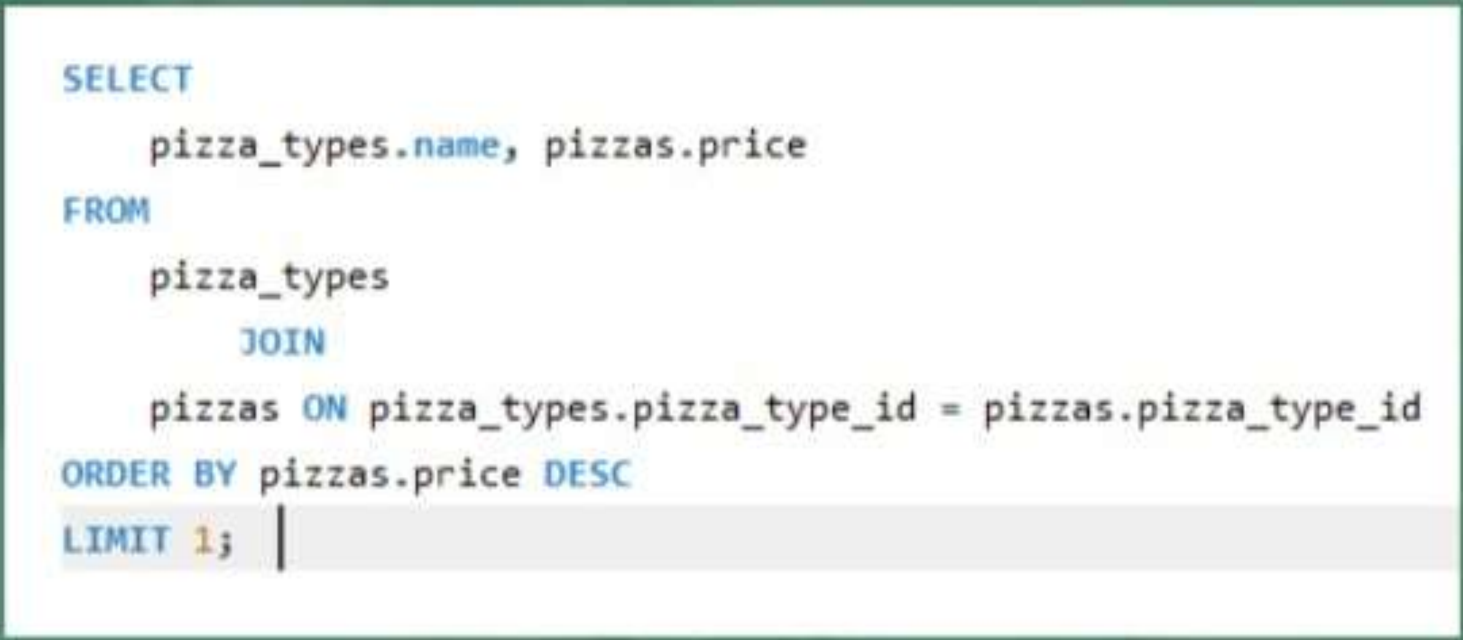
```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```



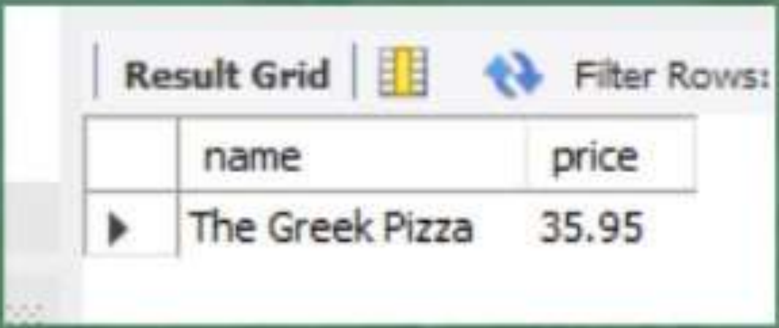
```
total_sales
817860.05
```

Q3.

Identify the highest-priced pizza.



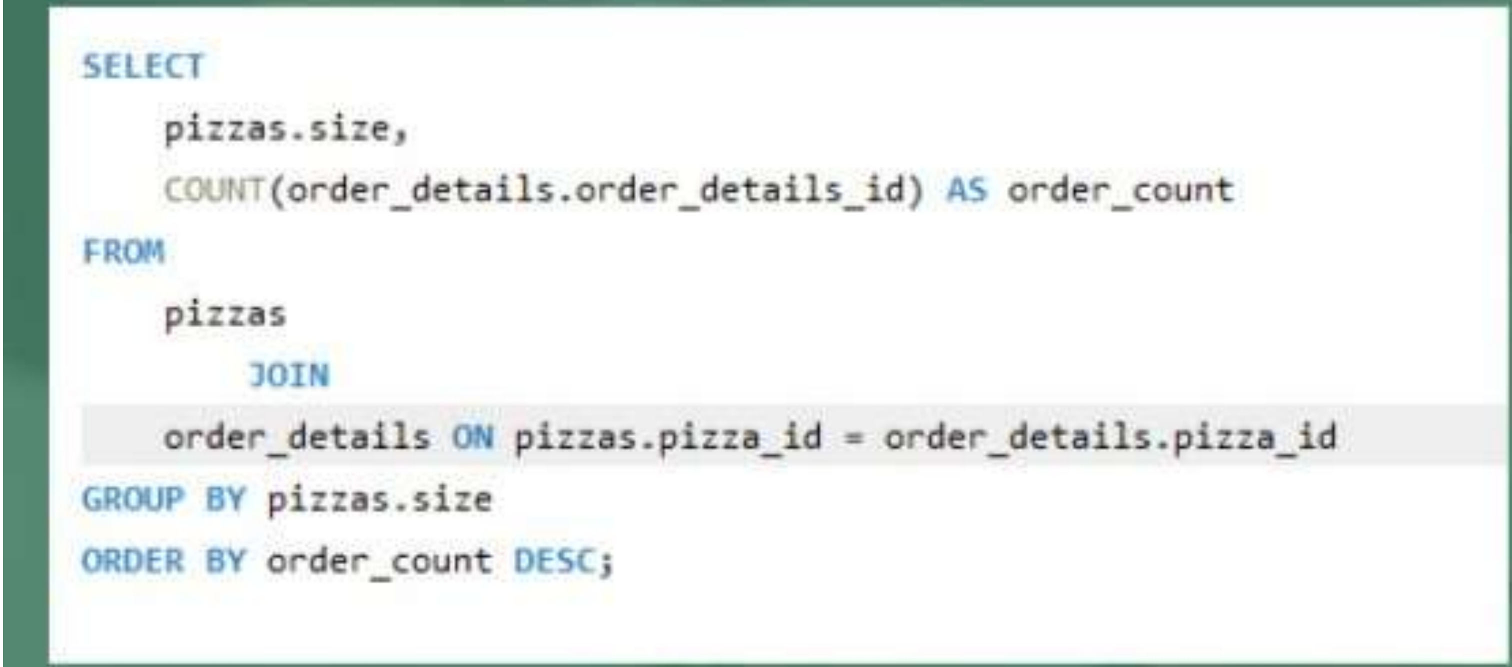
```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```



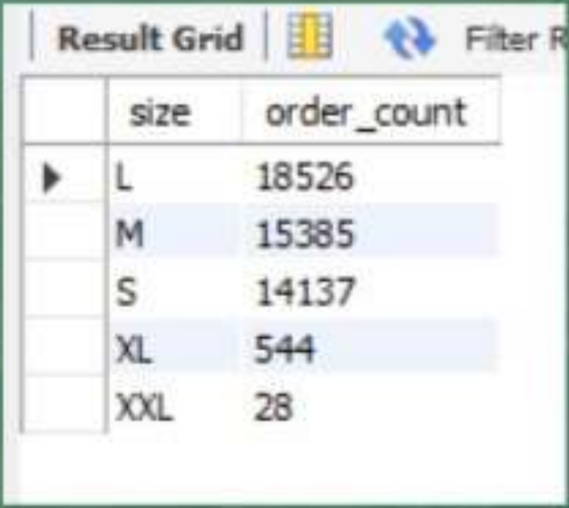
	name	price
▶	The Greek Pizza	35.95

Q4.

Identify the most common pizza size ordered.



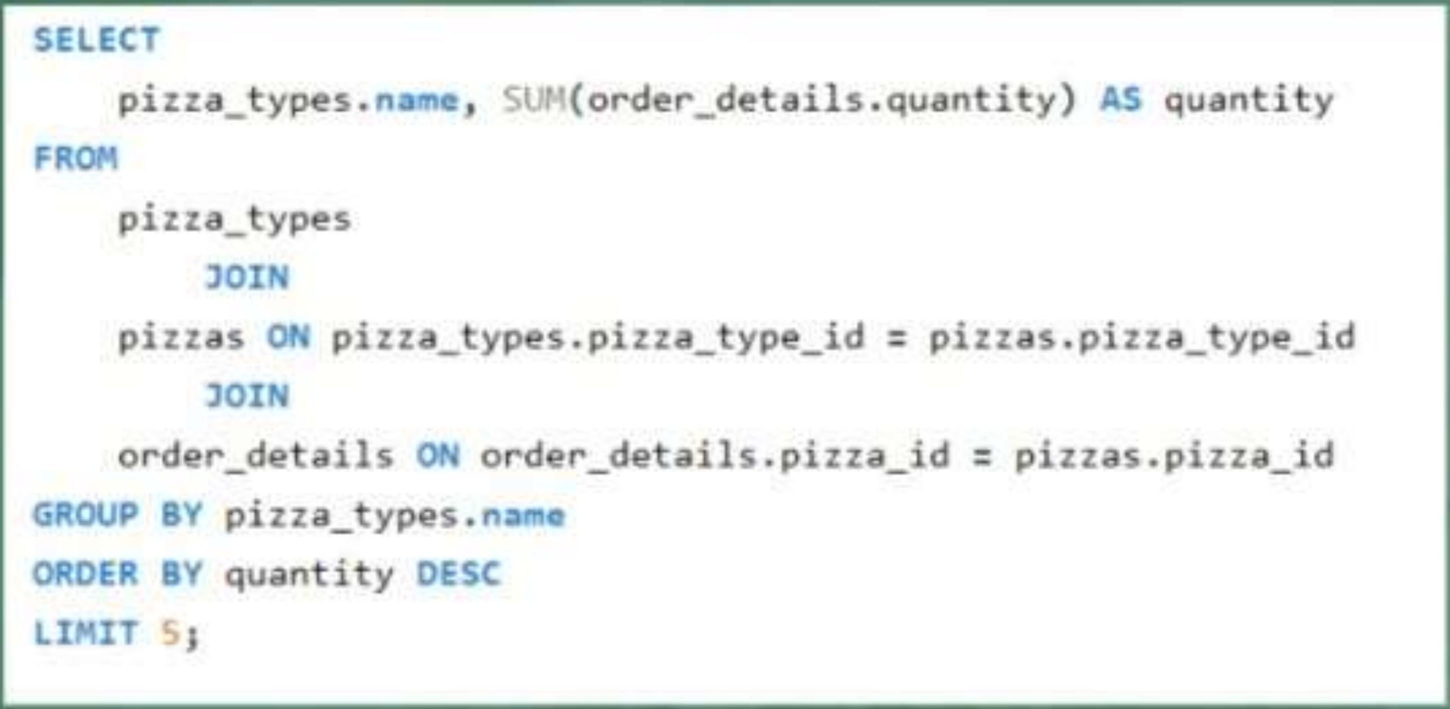
```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```



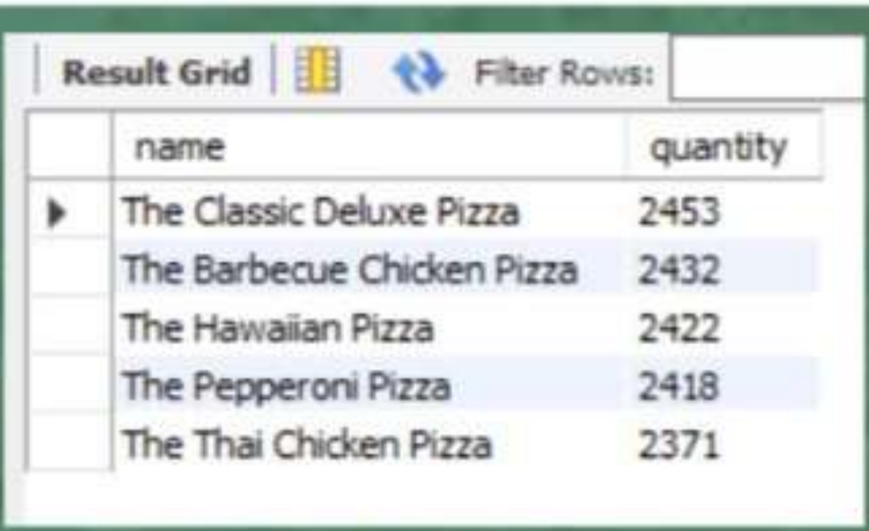
size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

Q5.

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



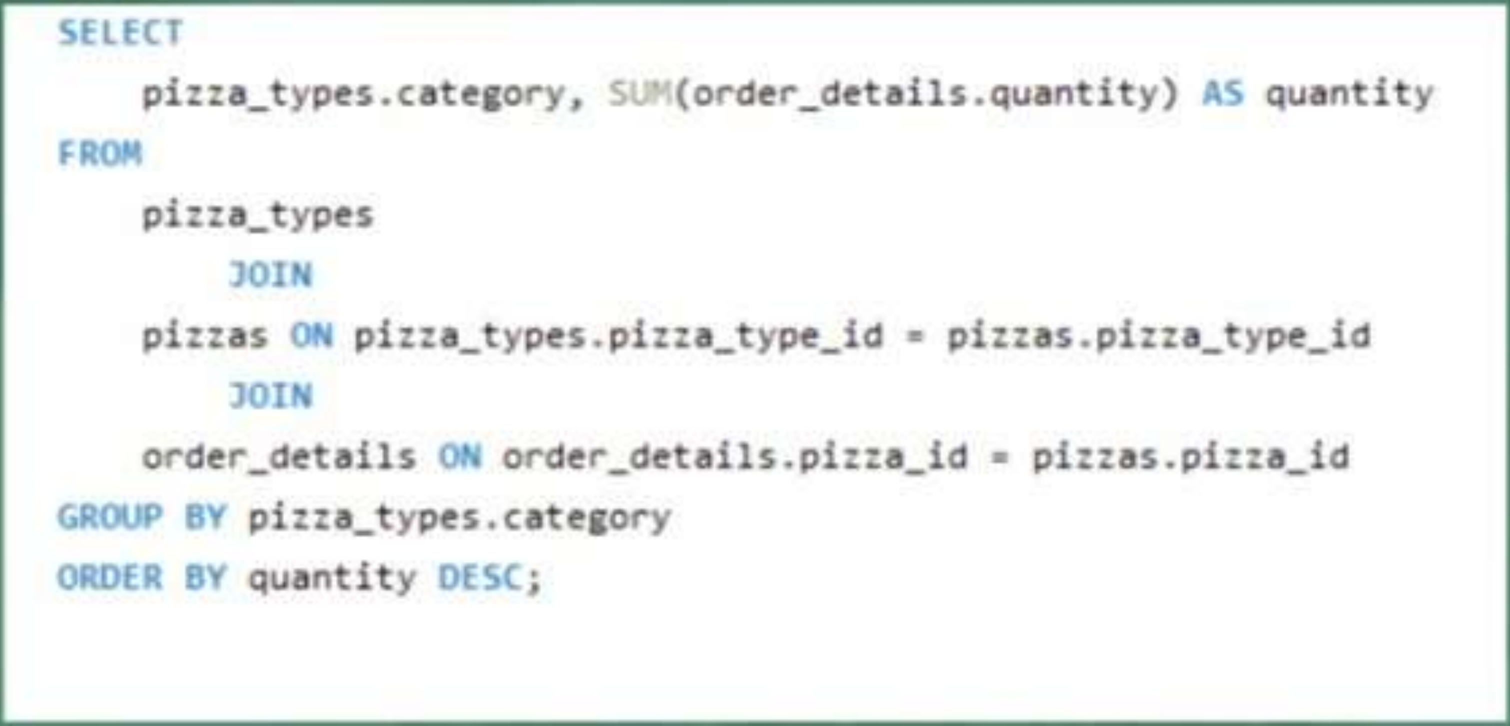
```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;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



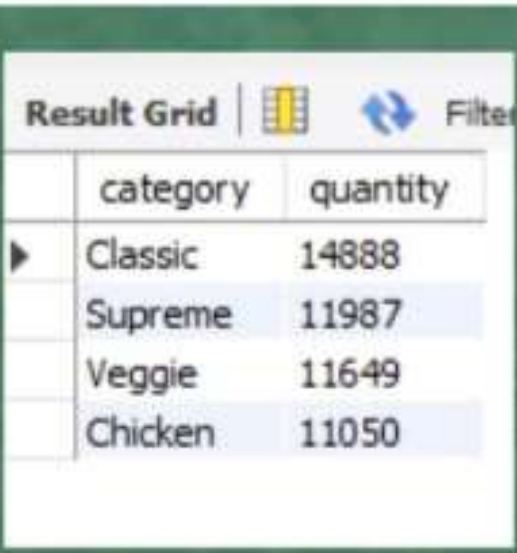
name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Q6.

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



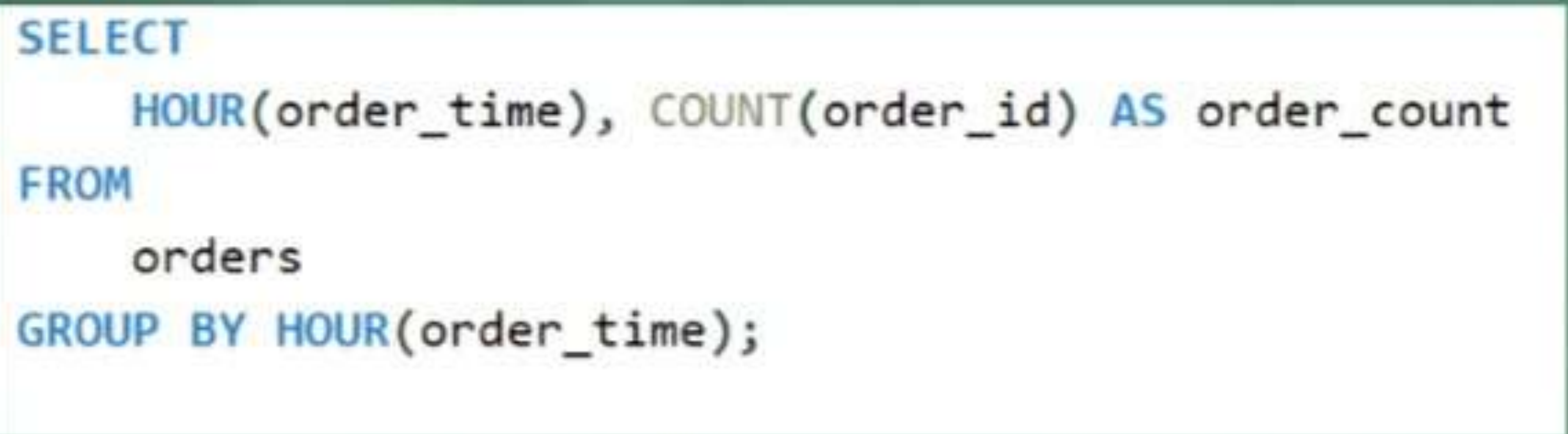
```
SELECT
    pizza_types.category, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;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



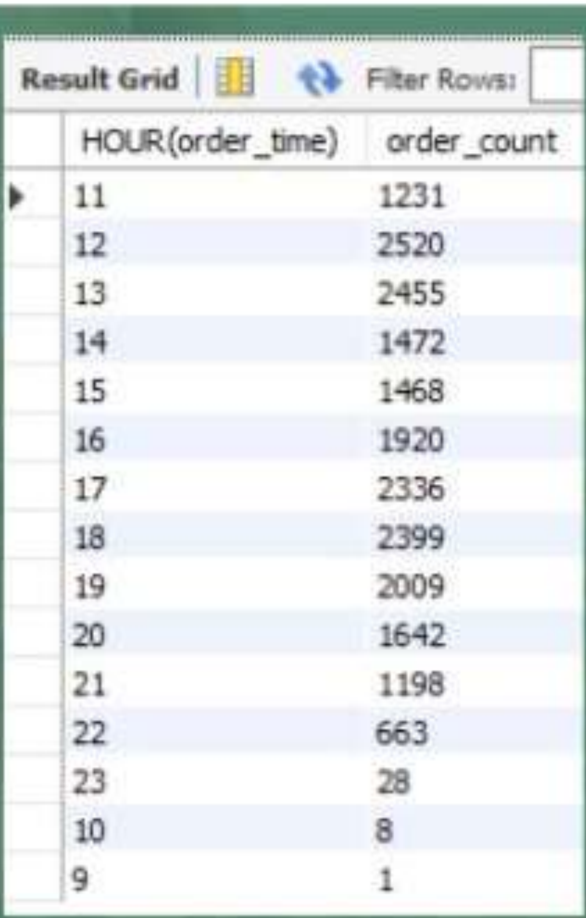
category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q7.

Determine the distribution of orders by hour of the day.



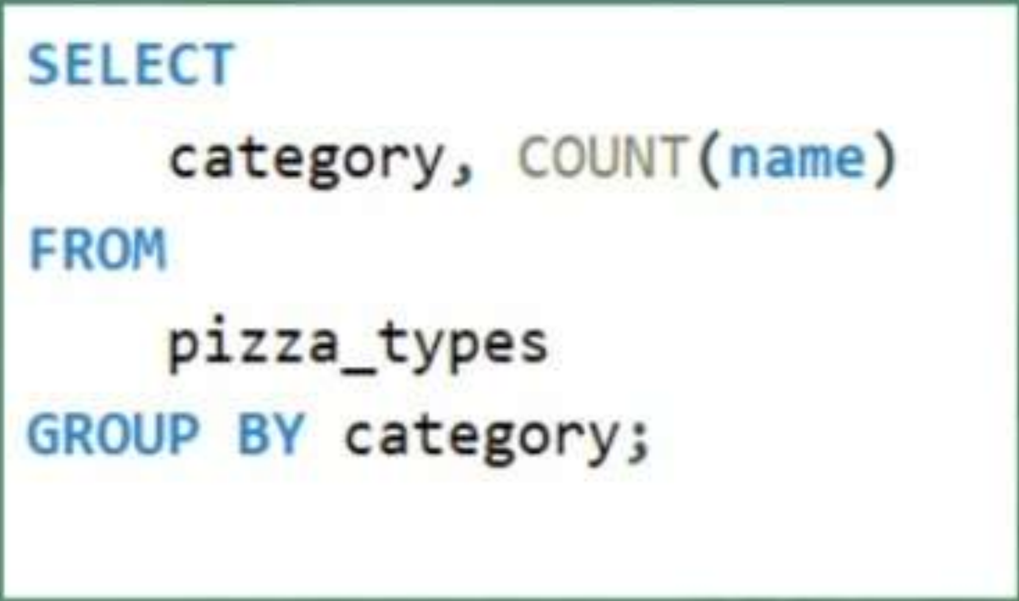
```sql
SELECT
    HOUR(order_time), COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```



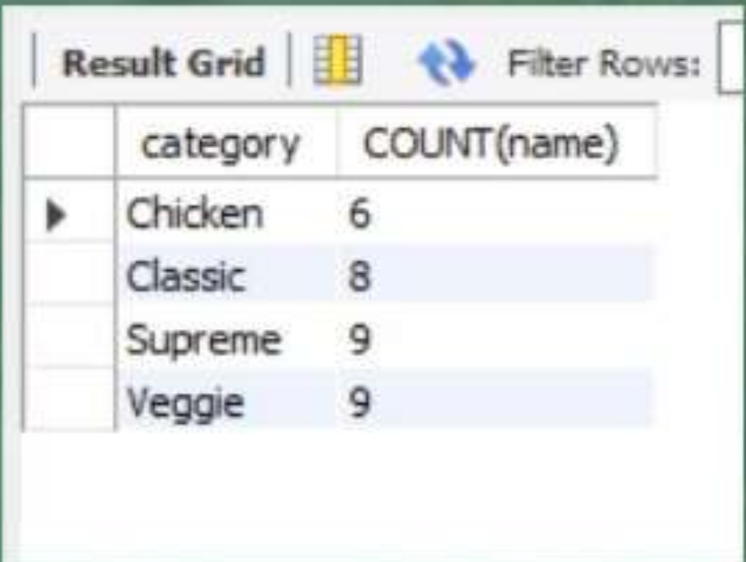
HOUR(order_time)	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

Q8.

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



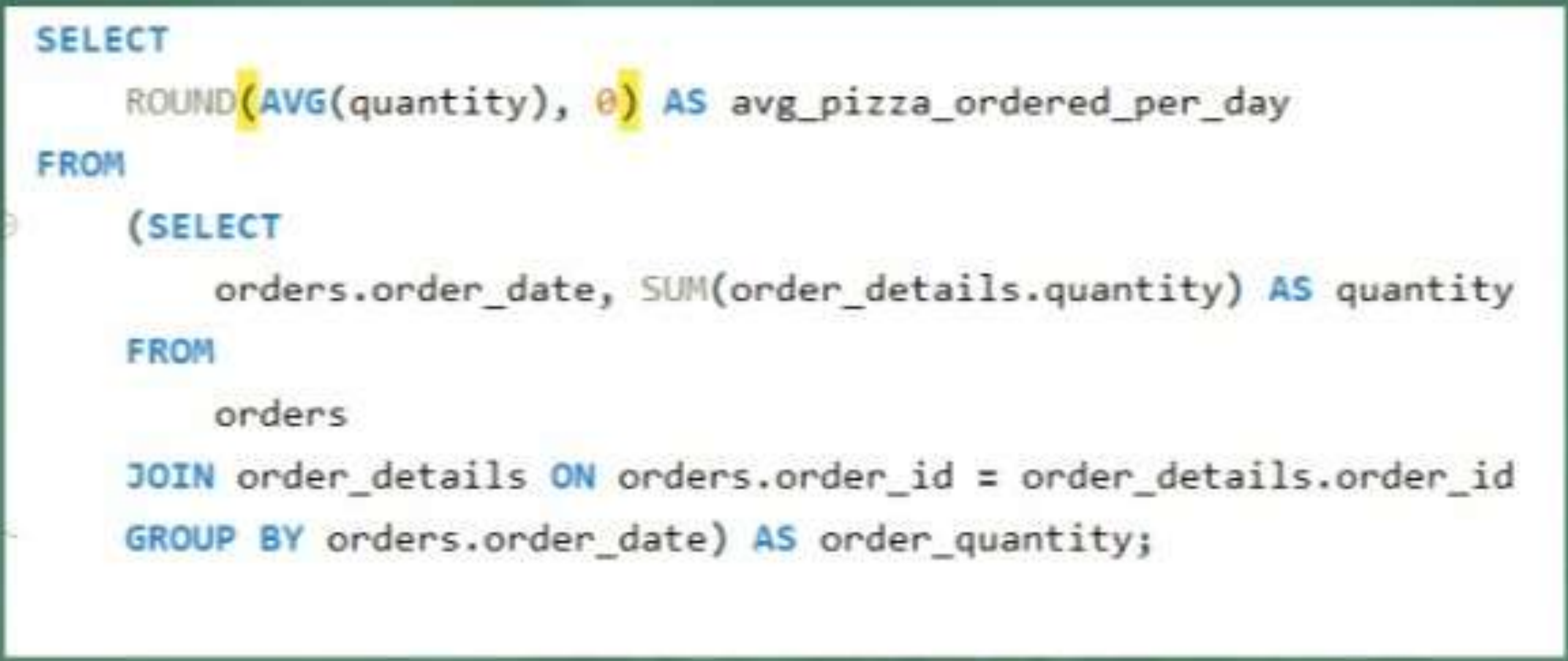
```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```



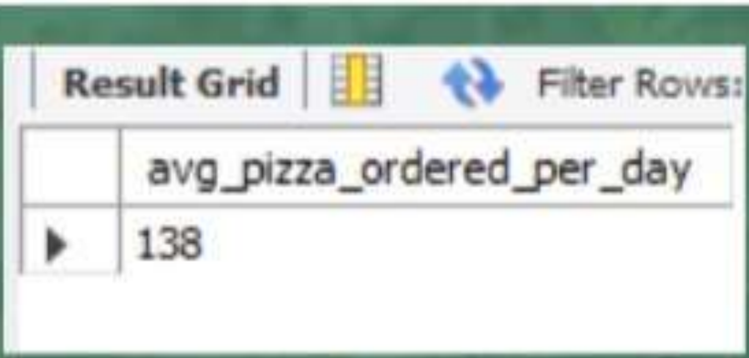
category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

Q9.

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



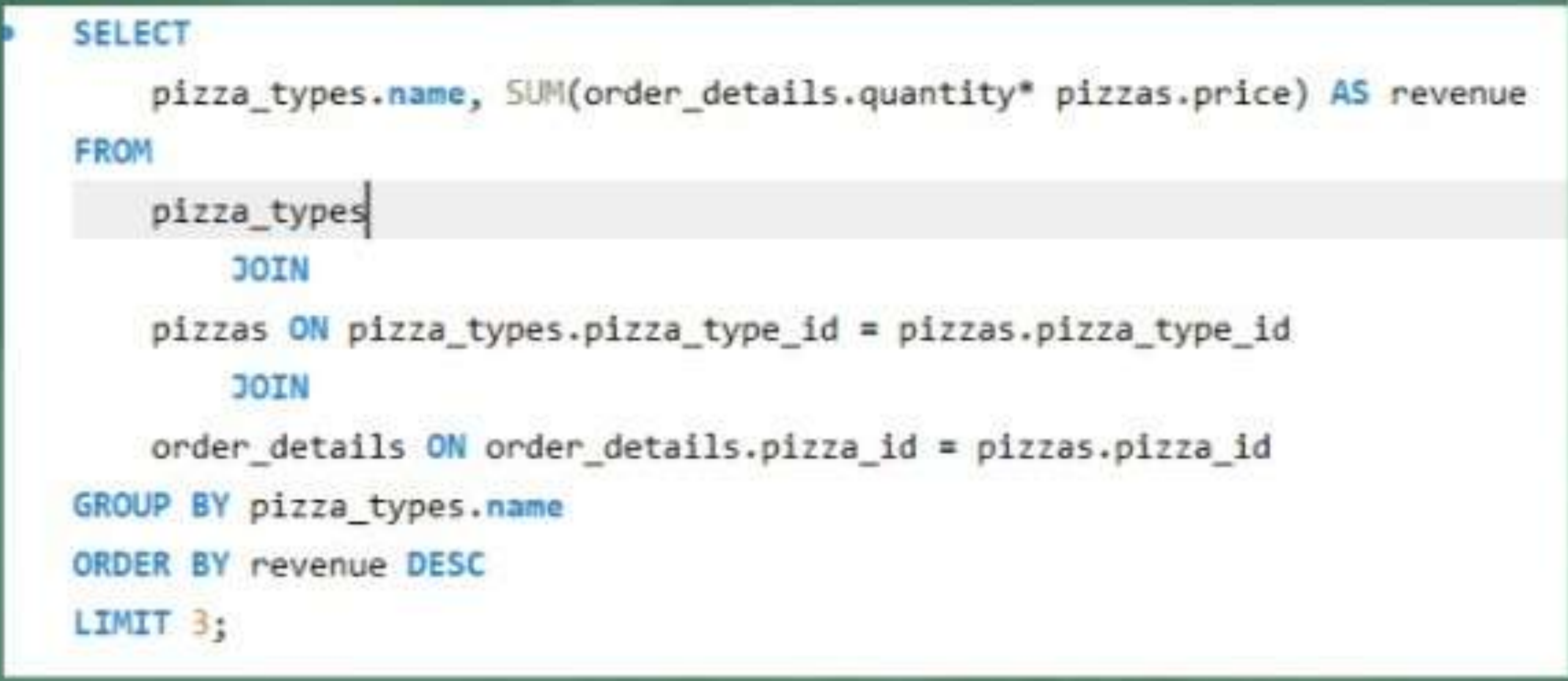
```
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```



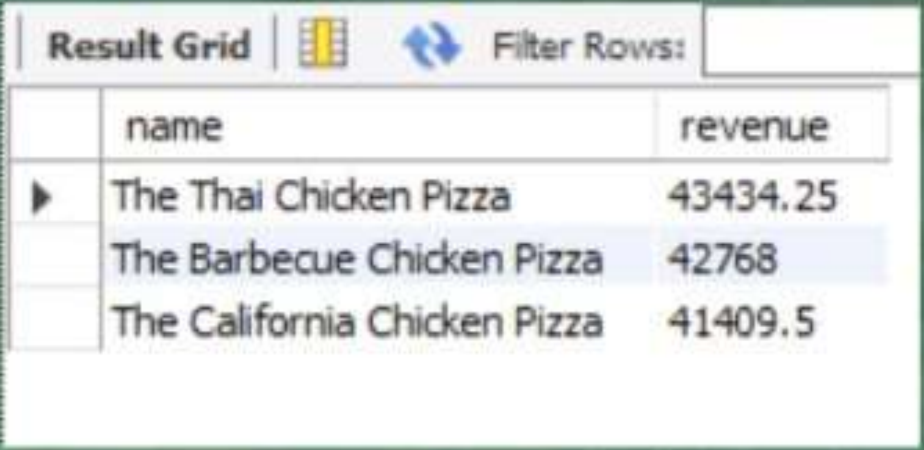
```
avg_pizza_ordered_per_day
138
```

Q10.

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



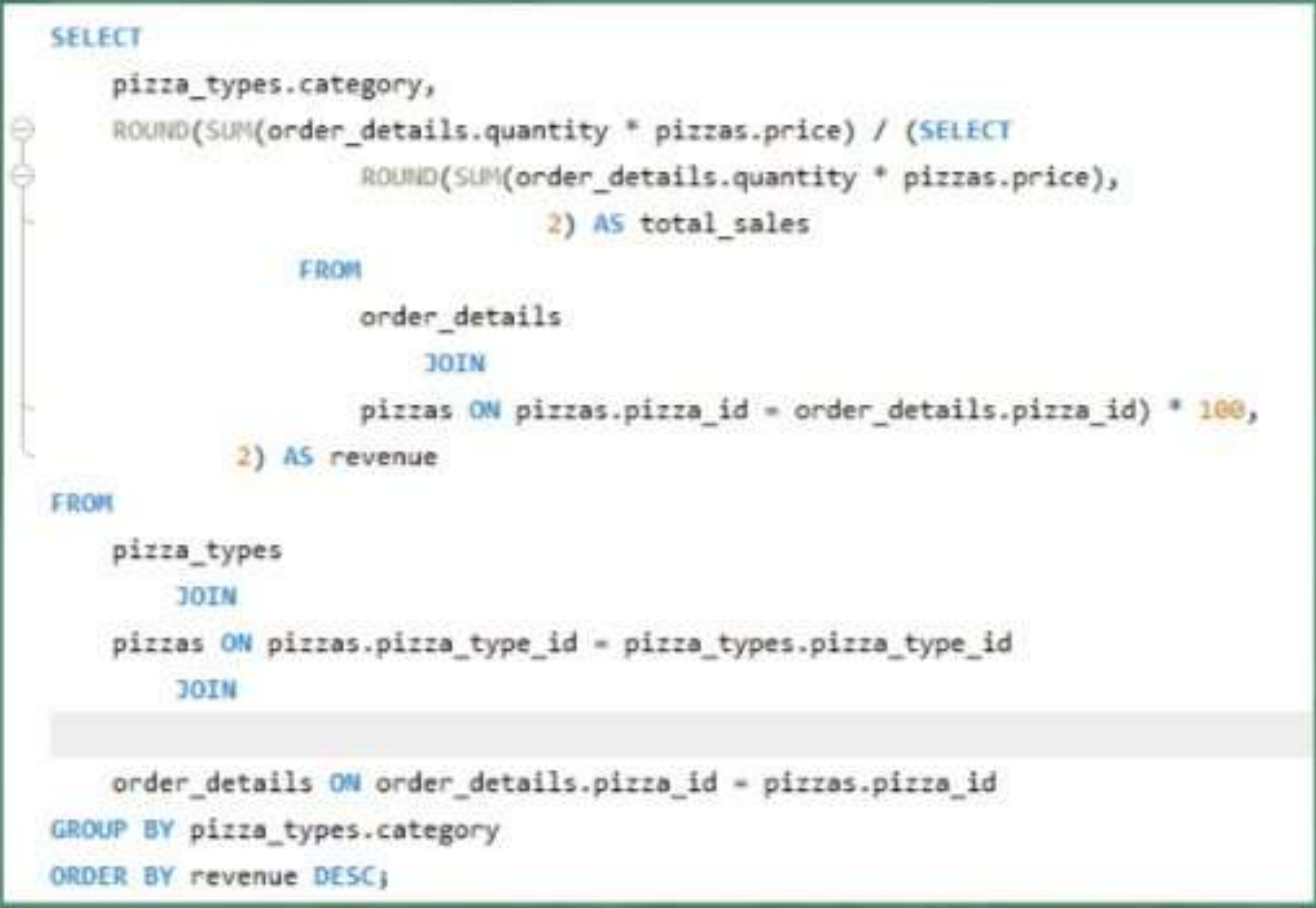
```sql
SELECT
    pizza_types.name, SUM(order_details.quantity* pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue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



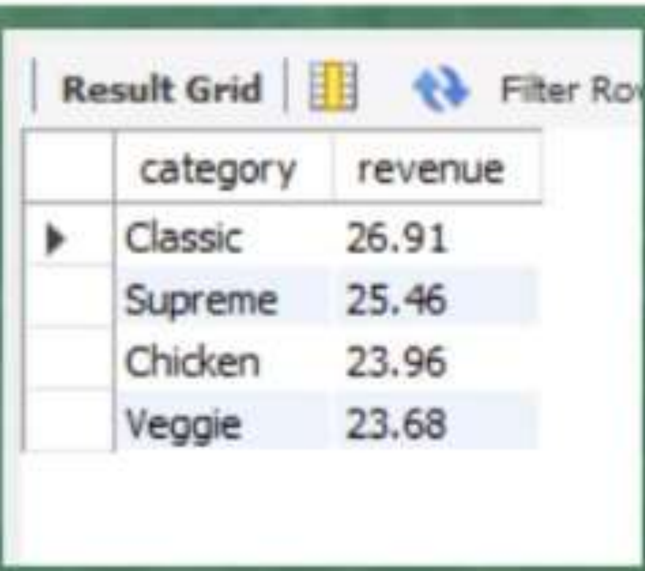
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Q11.

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



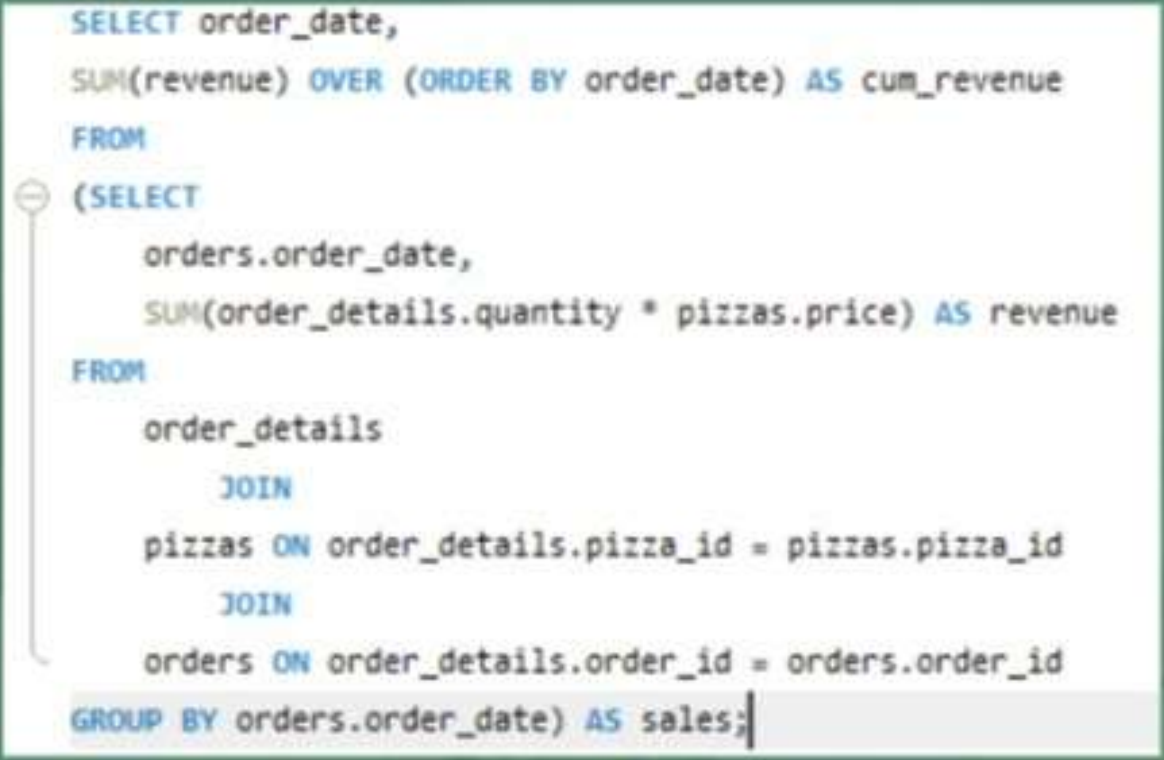
```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2) AS total_sales
                FROM
                    order_details
                        JOIN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



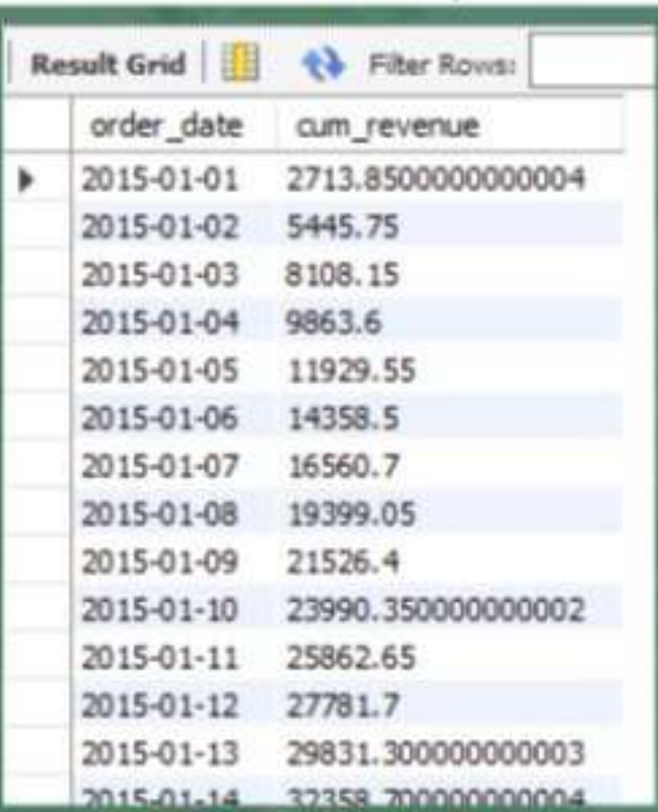
	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Q12.

Analyze the cumulative revenue generated over time.



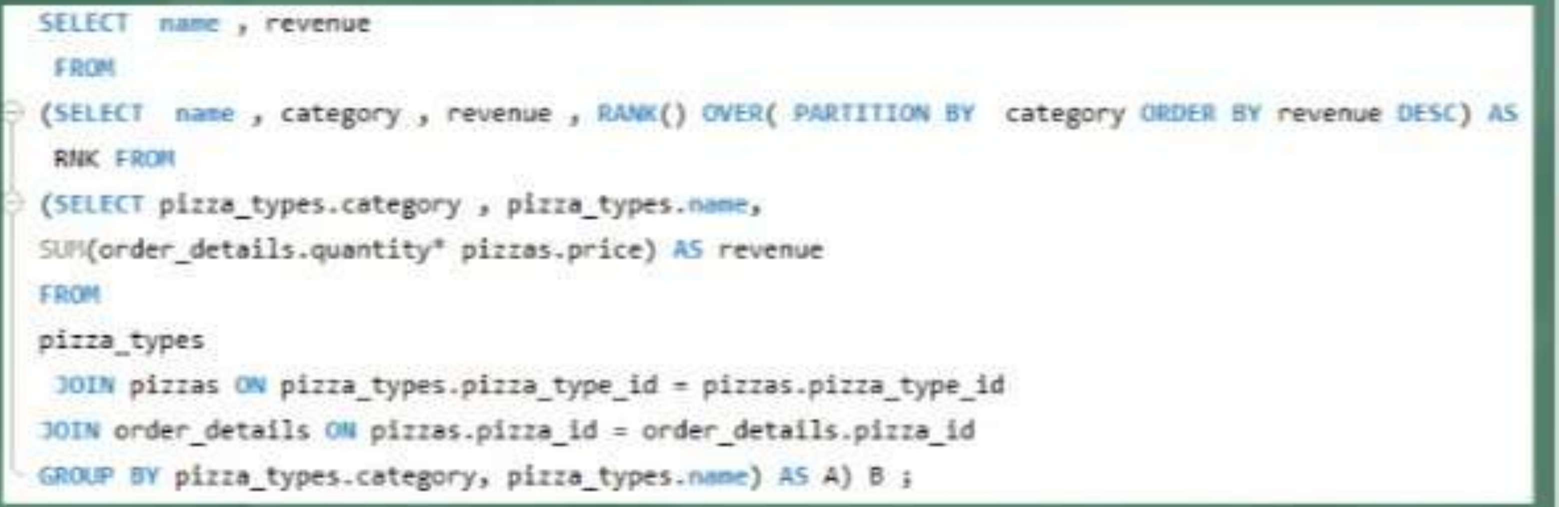
```sql
SELECT order_date,
SUM(revenue) OVER (ORDER BY order_date) AS cum_revenue
FROM
(SELECT
    orders.order_date,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    orders ON order_details.order_id = orders.order_id
GROUP BY orders.order_date) AS sales;
```



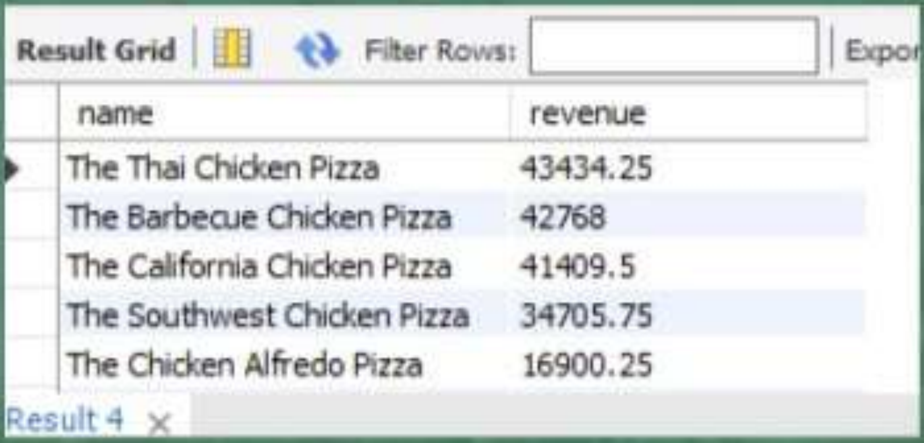
```
  order_date   cum_revenue
▶ 2015-01-01   2713.8500000000004
  2015-01-02   5445.75
  2015-01-03   8108.15
  2015-01-04   9863.6
  2015-01-05   11929.55
  2015-01-06   14358.5
  2015-01-07   16560.7
  2015-01-08   19399.05
  2015-01-09   21526.4
  2015-01-10   23990.350000000002
  2015-01-11   25862.65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

Q13.

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



```sql
SELECT  name , revenue
 FROM
(SELECT  name , category , revenue , RANK() OVER( PARTITION BY  category ORDER BY revenue DESC) AS
 RNK 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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category, pizza_types.name) AS A) B ;
```



name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Southwest Chicken Pizza	34705.75
The Chicken Alfredo Pizza	16900.25

Key Findings:

Order and Revenue Insights: Identified total number of orders and total revenue generated.
Top Pizzas and Sizes: Found the highest-priced pizza, most common pizza size, and top 5 pizza types ordered.
Order Patterns: Analyzed order distribution by hour and revenue contributions of top pizza types.

Challenges and Learnings:

Data Extraction: Faced challenges in cleaning and importing data, highlighting the need for meticulous data preparation.
Complex Queries: Improved skills in writing and optimizing complex SQL queries.
Data Interpretation: Gained experience in translating SQL results into actionable business insights.

Conclusion:

Objective Met: Achieved the goal of analyzing pizza sales data effectively.
SQL Proficiency: Demonstrated practical application of SQL for detailed data analysis.
Future Work: Opportunities for enhancing analysis with data visualization and deeper customer insights.

THANK YOU

Thank you for taking the time to review my Pizza Sales Analysis project. I appreciate your interest and hope you found the insights valuable. Special thanks to everyone who supported and guided me throughout this project.

If you have any questions or feedback, please feel free to reach out. I'm eager to continue learning and growing in the field of data analysis.