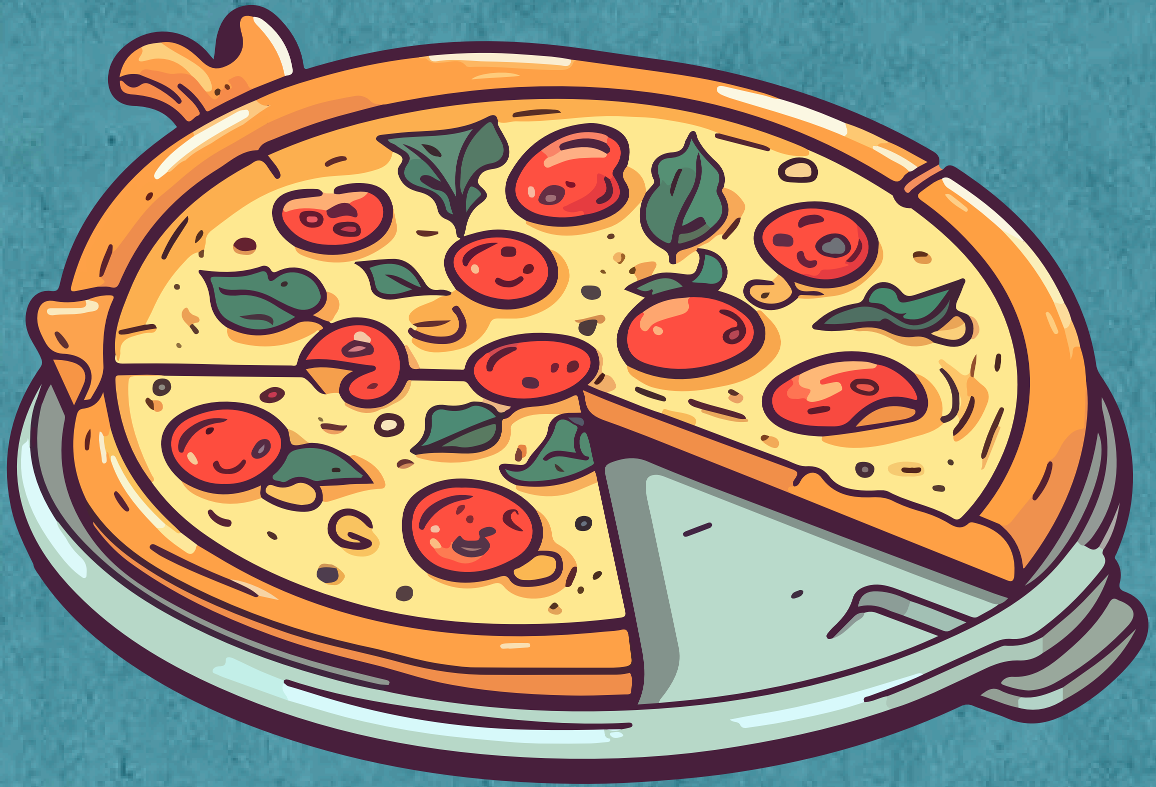
→

Analysis of sales of a pizza restaurant using **SQL**

Data Analysis using SQL

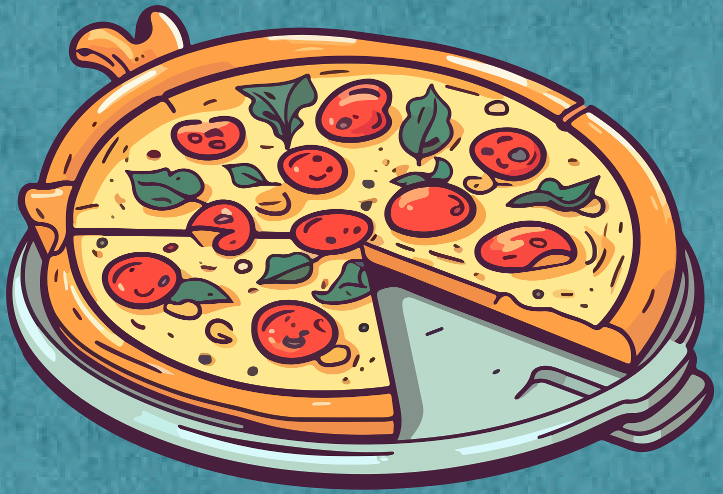


Presented by: Abdur Rehman

2024

Data Preparation and Analysis

Data Sheets Provided:



- pizza_types (pre-cleaned)
- pizzas (pre-cleaned)
- orders (cleaned using SQL)
- order_details (cleaned using SQL)

1. **Cleaning Process:** Conducted further cleaning on orders and order_details using SQL.
2. **Analysis:** Utilized SQL queries to address potential stakeholder questions related to pizza sales.
3. **Additional Resources:** **GitHub Repository** link with datasets and questions available for download.

Presented by: Abdur Rehman

2024

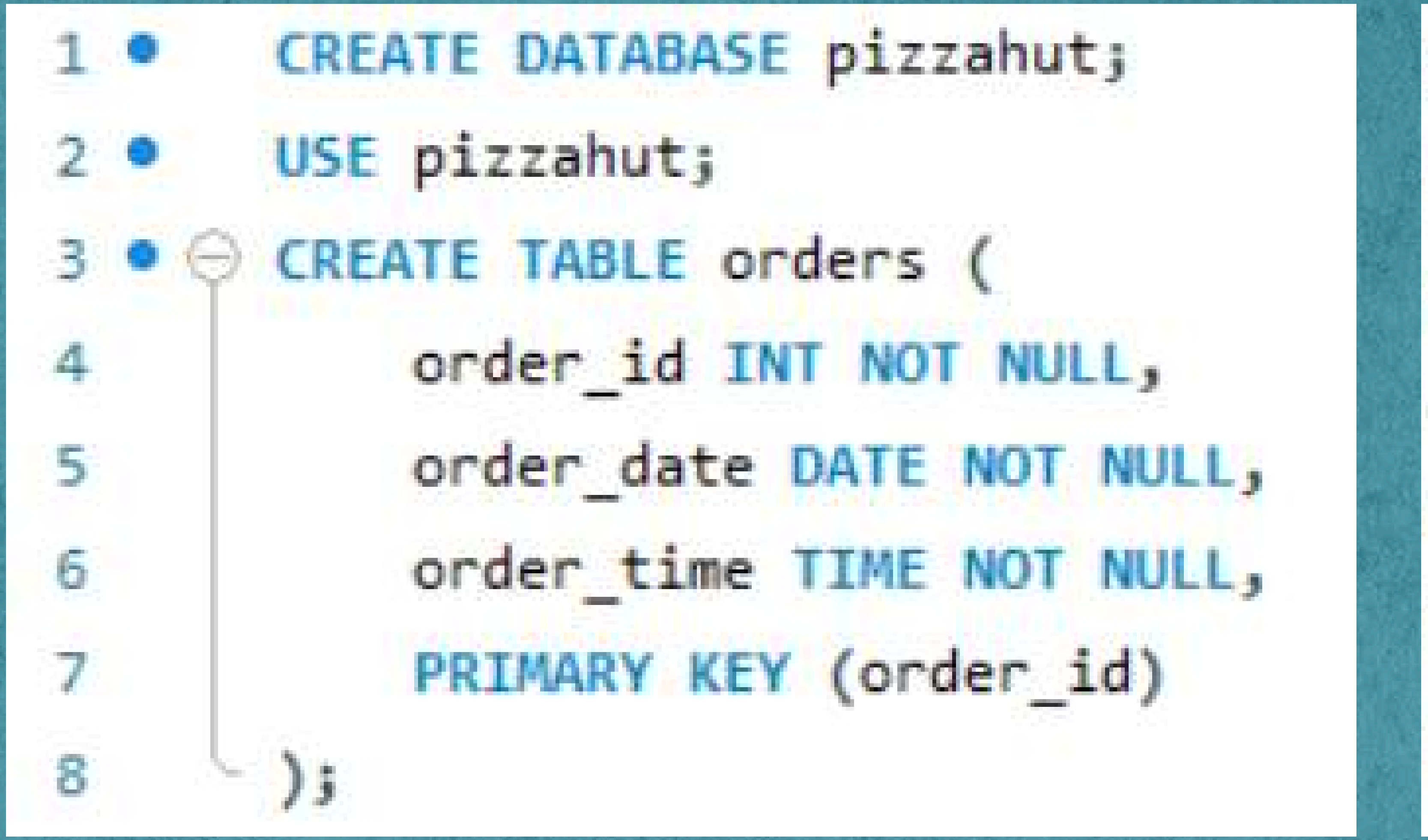
Potential Stakeholder Queries for a Pizza Restaurant

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.
6. Find the total quantity of each pizza category ordered.
7. Determine the distribution of orders by hour of the day.

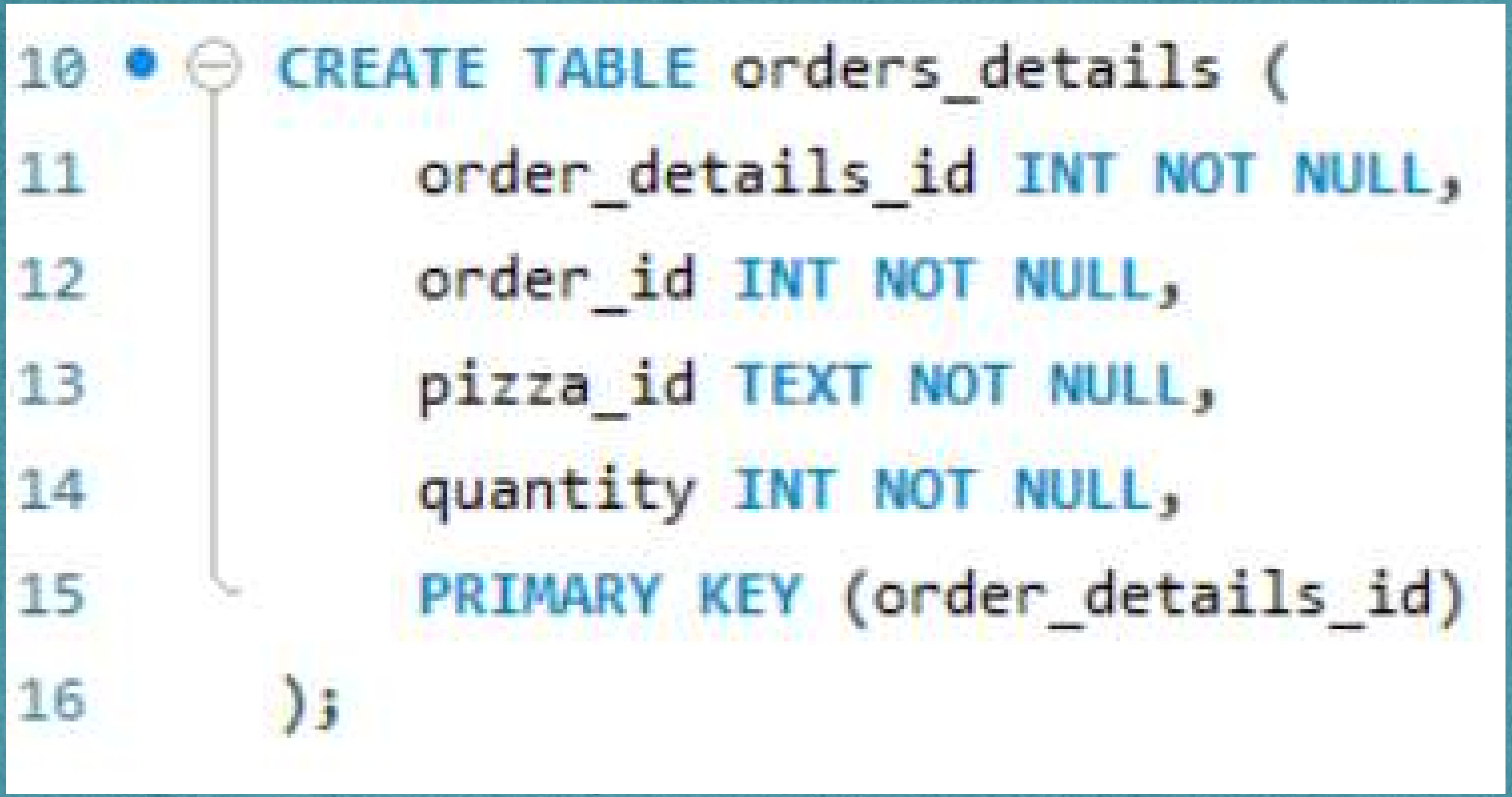
Potential Stakeholder Queries for a Pizza Restaurant

8. Find the category-wise distribution of pizzas.
9. Calculate the average number of pizzas ordered per day.
10. Determine the top 3 most ordered pizza types based on revenue.
11. Calculate the percentage contribution of each pizza type to total revenue.
12. Analyze the cumulative revenue generated over time.
13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

1. Queries to clean the sheets, ‘orders’ and ‘order_details’



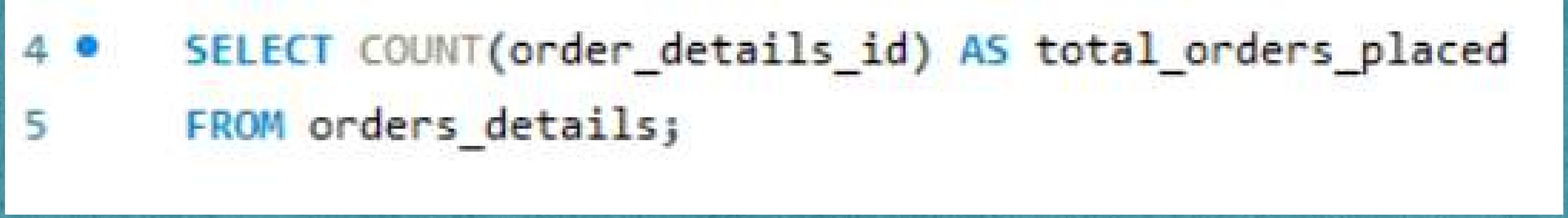
```sql
CREATE DATABASE pizzahut;
USE pizzahut;
CREATE TABLE orders (
    order_id INT NOT NULL,
    order_date DATE NOT NULL,
    order_time TIME NOT NULL,
    PRIMARY KEY (order_id)
);
```



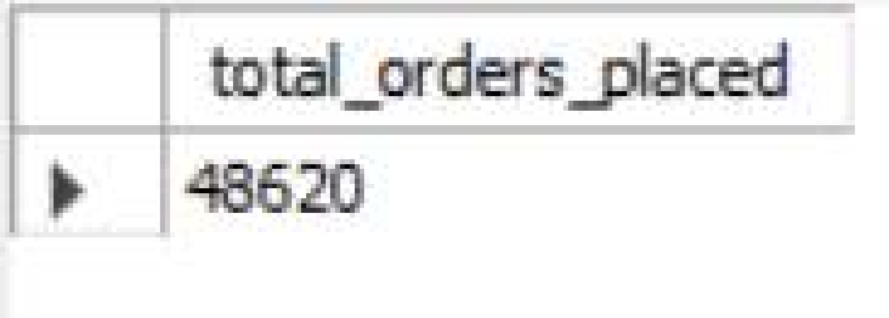
```sql
CREATE TABLE orders_details (
    order_details_id INT NOT NULL,
    order_id INT NOT NULL,
    pizza_id TEXT NOT NULL,
    quantity INT NOT NULL,
    PRIMARY KEY (order_details_id)
);
```

NEXT

2. Retrieve the total number of orders placed.

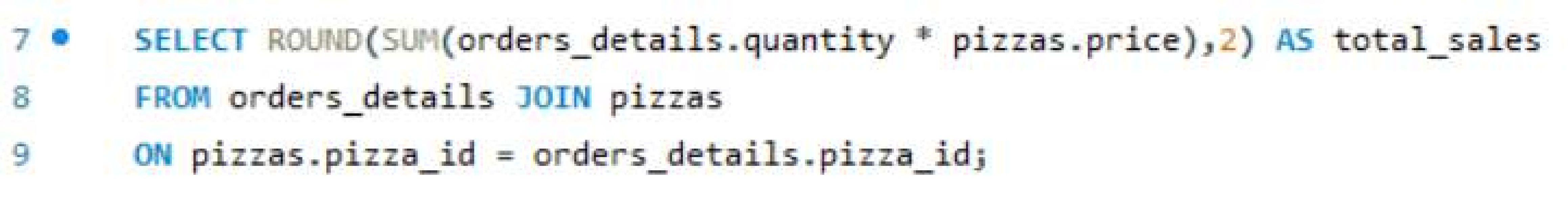


```
SELECT COUNT(order_details_id) AS total_orders_placed
FROM orders_details;
```

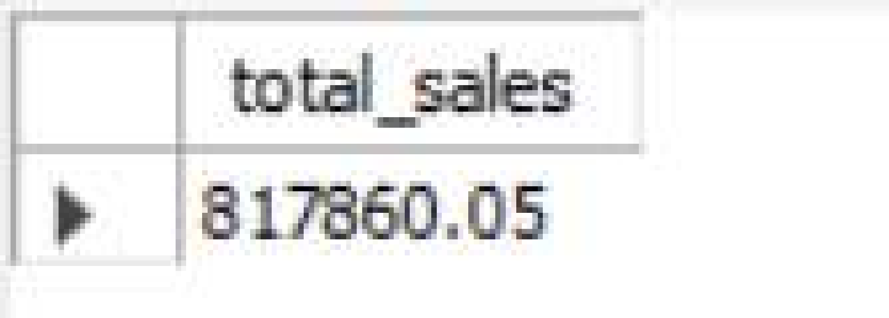


```
   total_orders_placed
▶  48620
```

3. Calculate the total revenue generated from pizza sales.



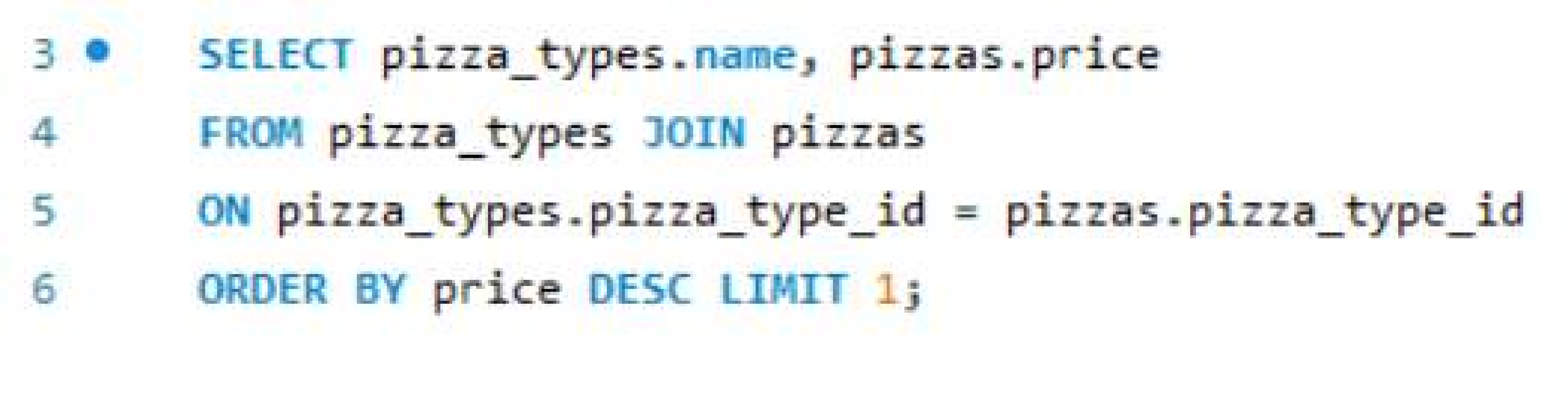
```
SELECT ROUND(SUM(orders_details.quantity * pizzas.price),2) AS total_sales
FROM orders_details JOIN pizzas
ON pizzas.pizza_id = orders_details.pizza_id;
```



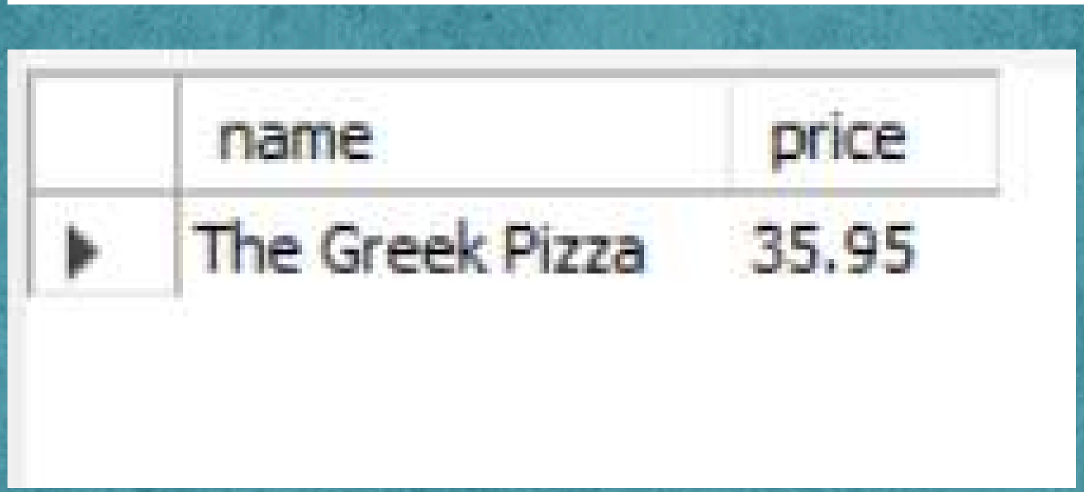
```
   total_sales
▶  817860.05
```

NEXT

4. Identify the highest-priced pizza.



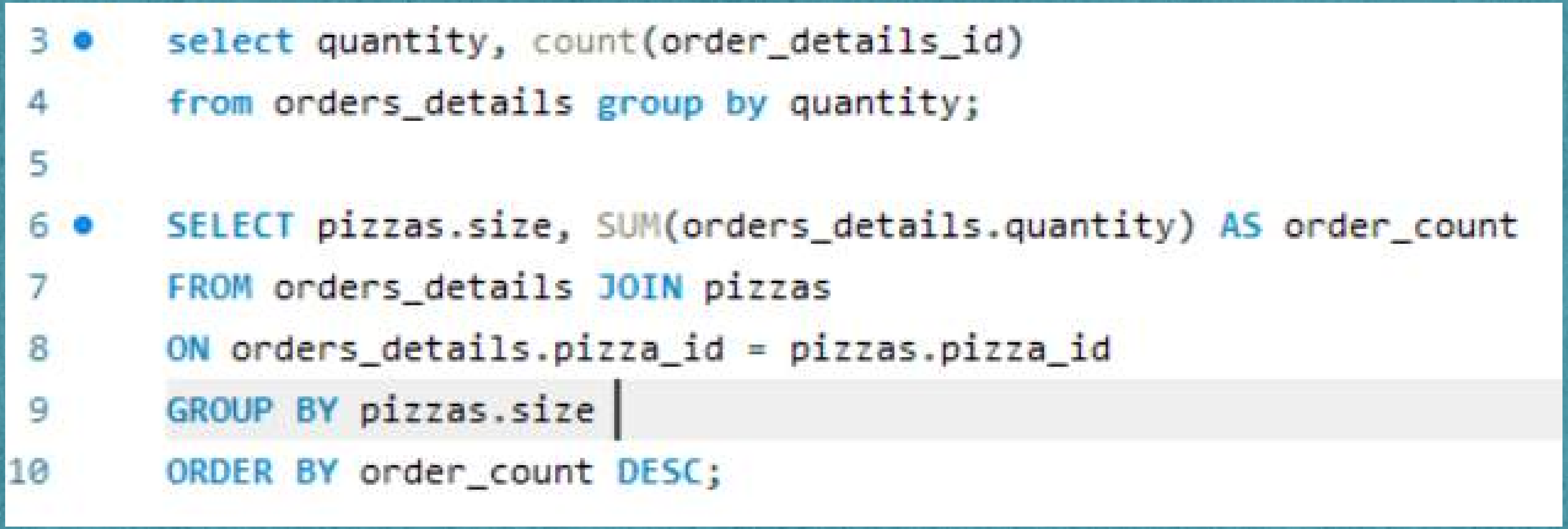
```
SELECT pizza_types.name, pizzas.price
FROM pizza_types JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY price DESC LIMIT 1;
```



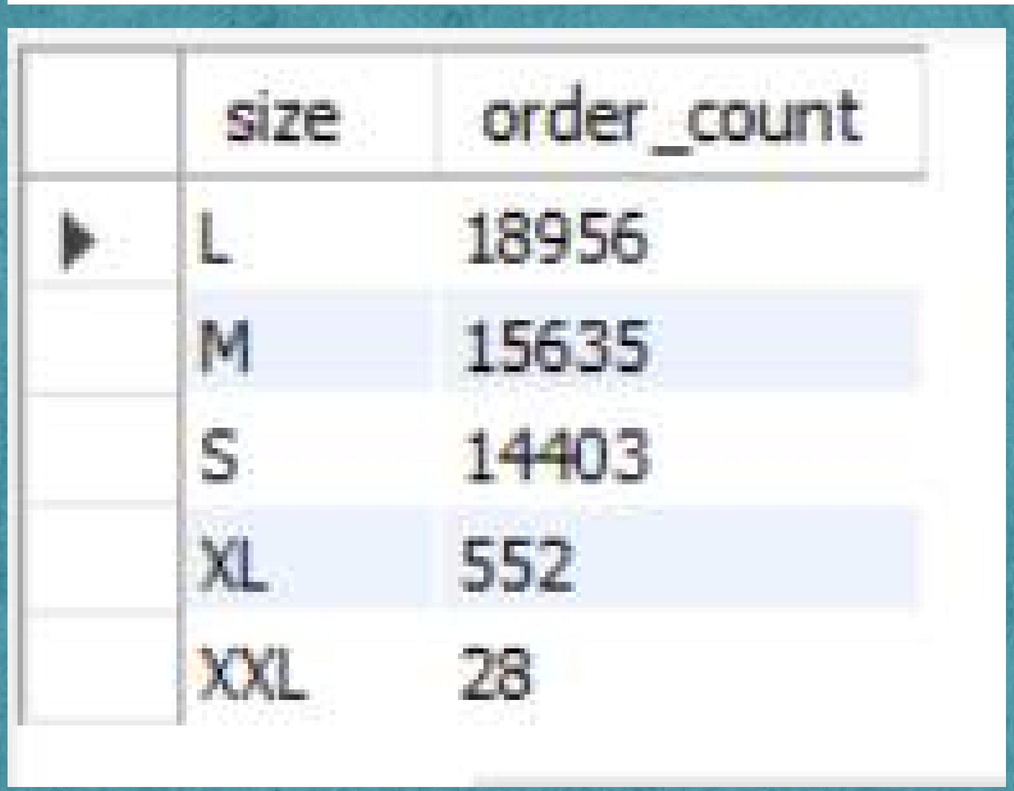
	name	price
▶	The Greek Pizza	35.95

NEXT

5. Identify the most common pizza size ordered.



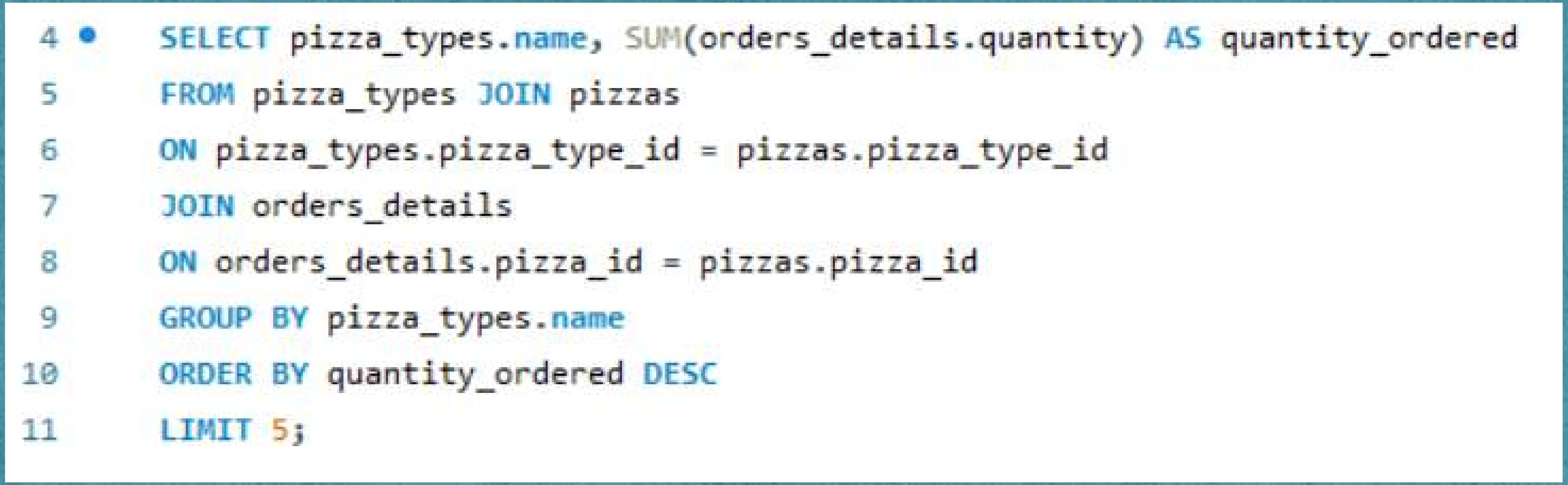
```sql
select quantity, count(order_details_id)
from orders_details group by quantity;

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



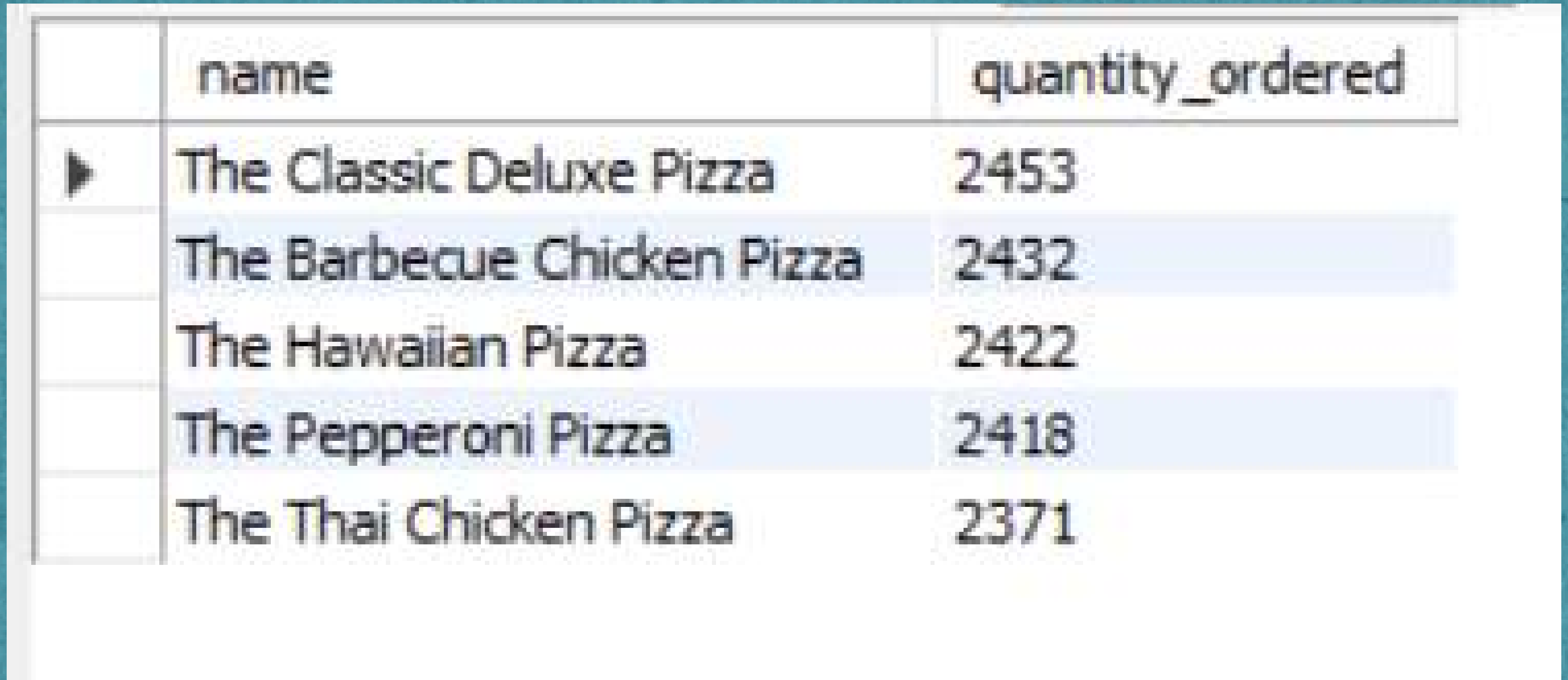
size	order_count
L	18956
M	15635
S	14403
XL	552
XXL	28

NEXT

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



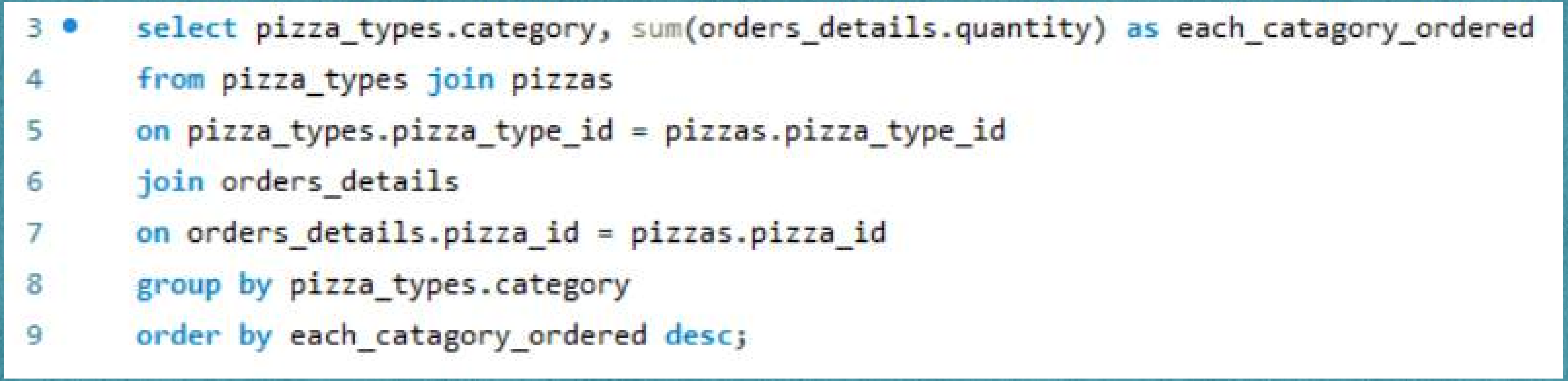
```sql
SELECT pizza_types.name, SUM(orders_details.quantity) AS quantity_ordered
FROM pizza_types JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN orders_details
ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity_ordered DESC
LIMIT 5;
```



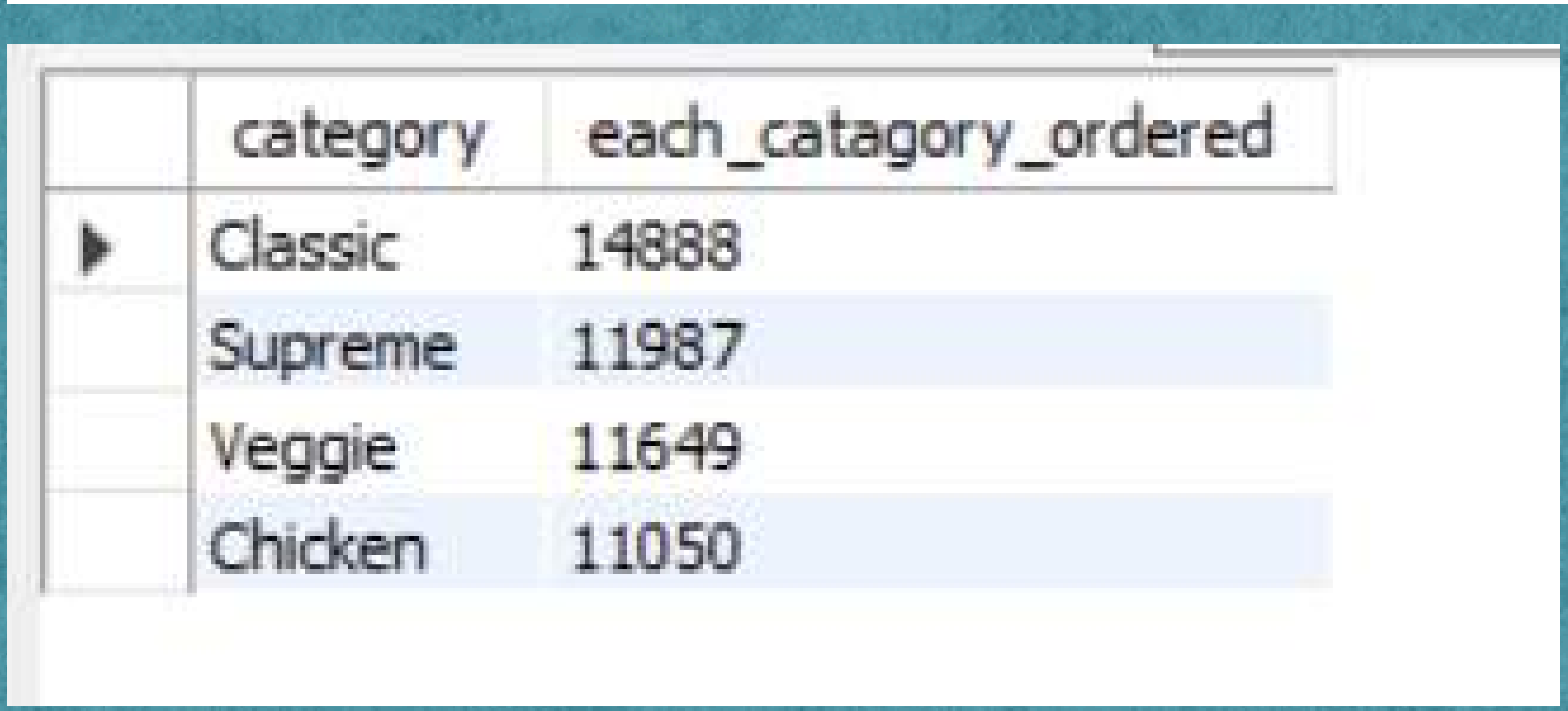
name	quantity_ordered
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

NEXT

7. Find the total quantity of each pizza category ordered.



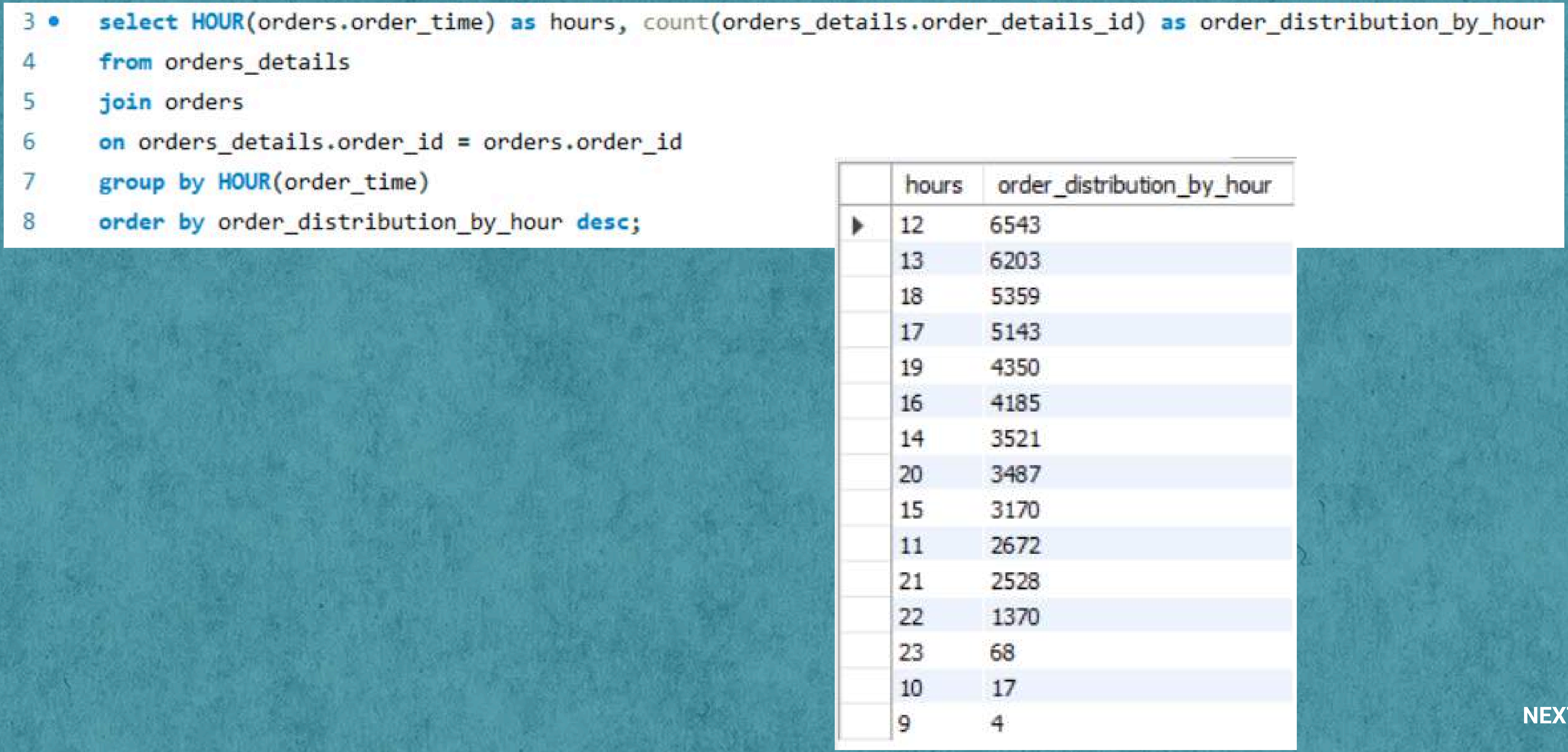
```sql
select pizza_types.category, sum(orders_details.quantity) as each_catagory_ordered
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.category
order by each_catagory_ordered desc;
```



category	each_catagory_ordered
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

NEXT

8. Determine the distribution of orders by hour of the day.

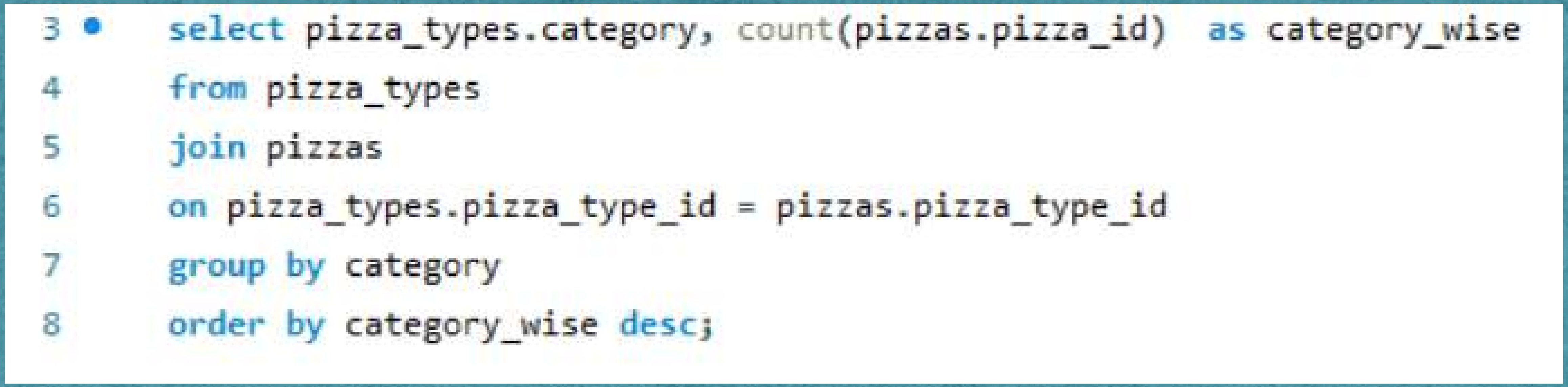


```sql
select HOUR(orders.order_time) as hours, count(orders_details.order_details_id) as order_distribution_by_hour
from orders_details
join orders
on orders_details.order_id = orders.order_id
group by HOUR(order_time)
order by order_distribution_by_hour desc;
```

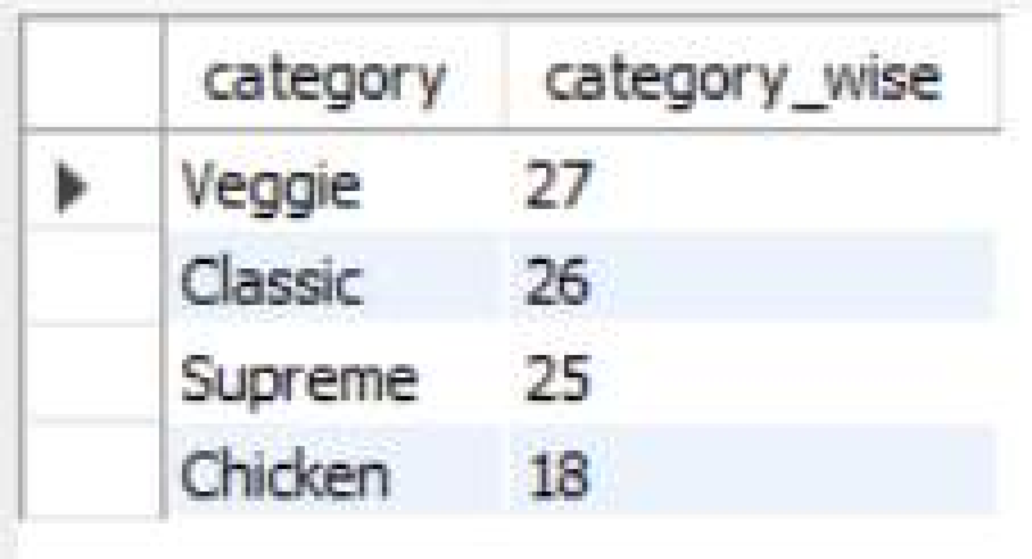
hours	order_distribution_by_hour
12	6543
13	6203
18	5359
17	5143
19	4350
16	4185
14	3521
20	3487
15	3170
11	2672
21	2528
22	1370
23	68
10	17
9	4

NEXT

9. Find the category-wise distribution of pizzas.



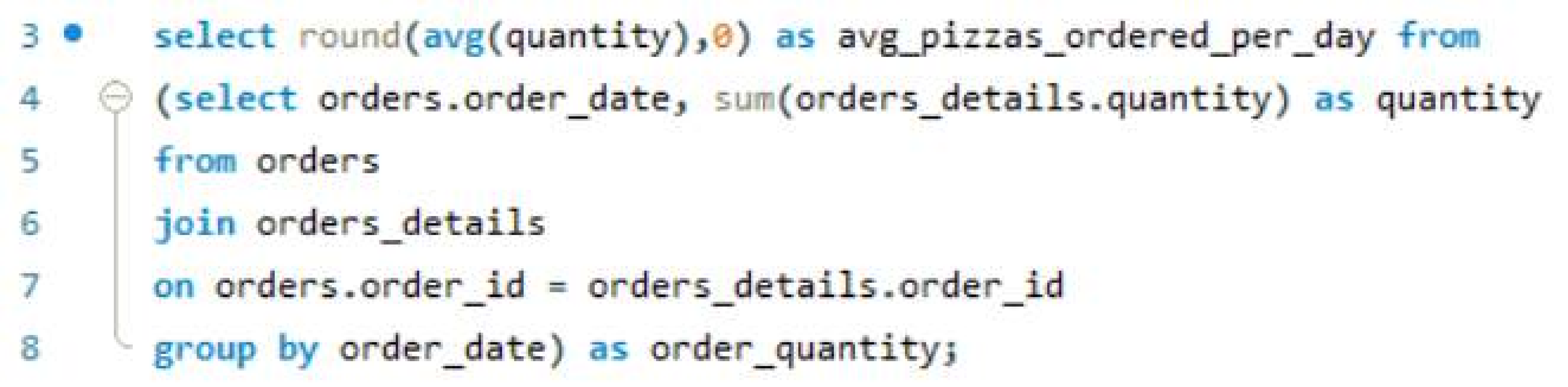
```sql
select pizza_types.category, count(pizzas.pizza_id)  as category_wise
from pizza_types
join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
group by category
order by category_wise desc;
```



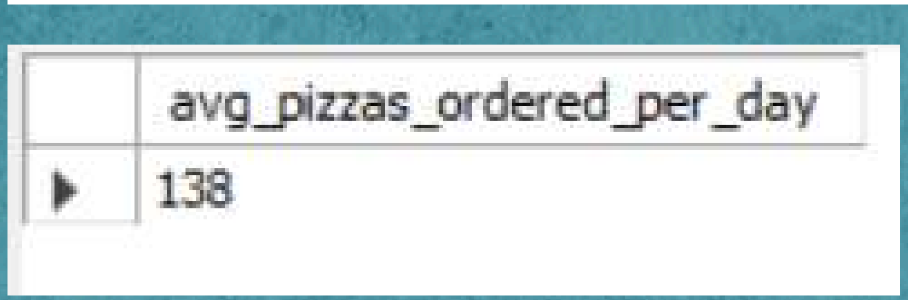
category	category_wise
Veggie	27
Classic	26
Supreme	25
Chicken	18

NEXT

10. Calculate the average number of pizzas ordered per day.



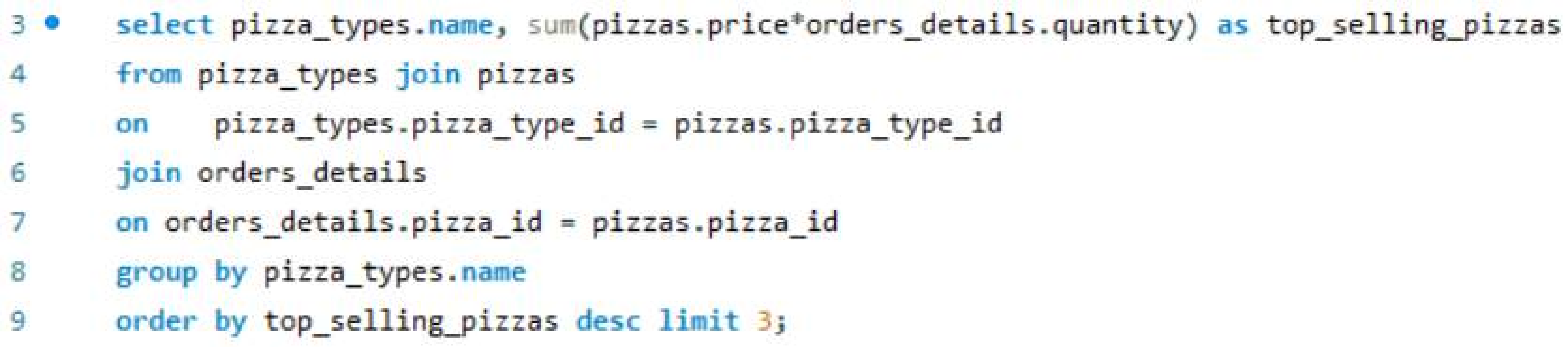
```sql
select round(avg(quantity),0) as avg_pizzas_ordered_per_day from
(select orders.order_date, sum(orders_details.quantity) as quantity
from orders
join orders_details
on orders.order_id = orders_details.order_id
group by order_date) as order_quantity;
```



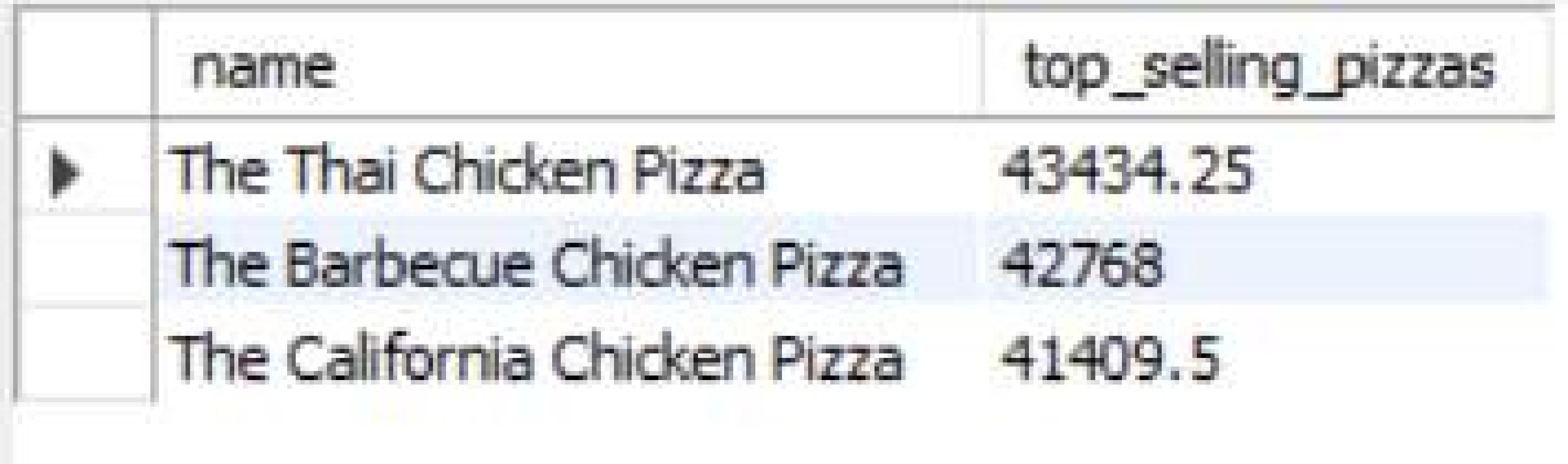
```
avg_pizzas_ordered_per_day
138
```

NEXT

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



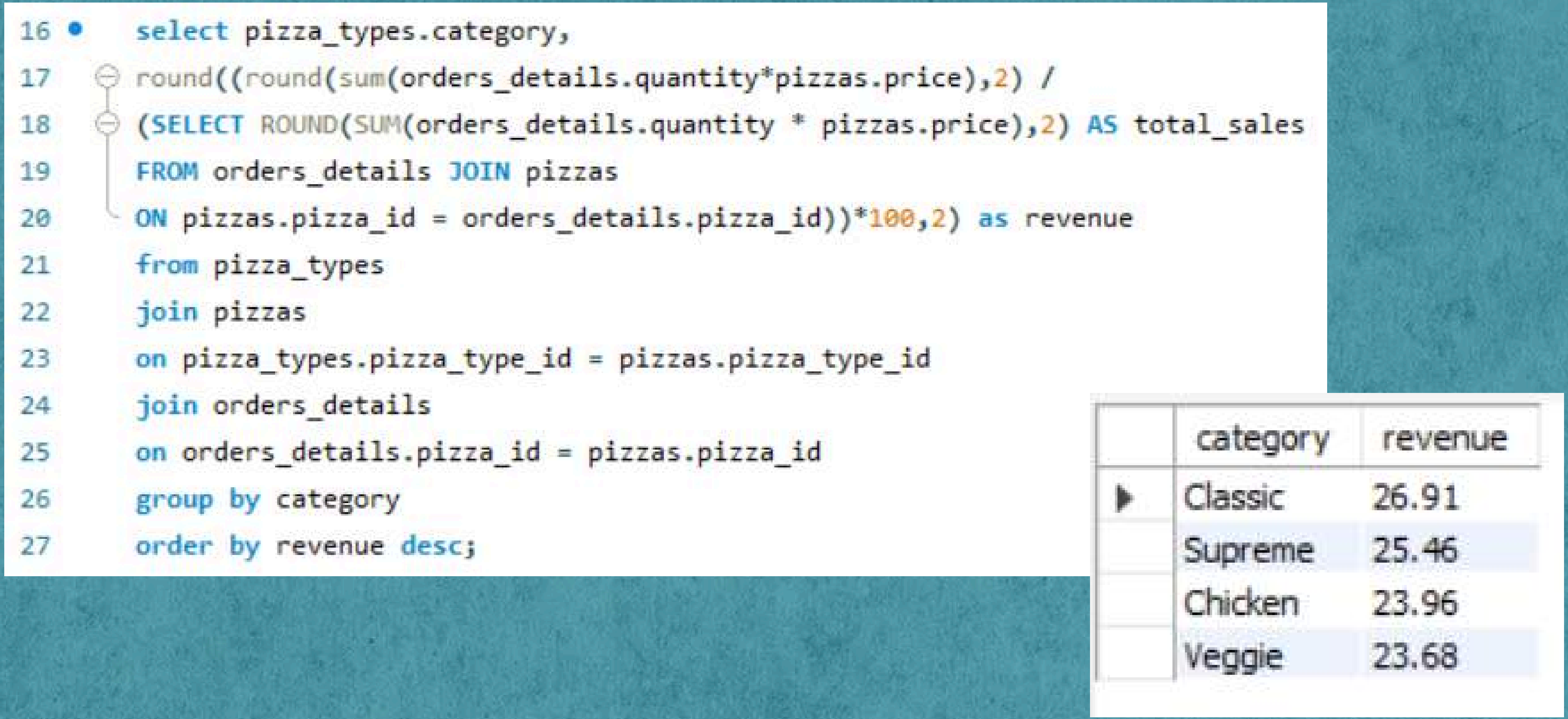
```sql
select pizza_types.name, sum(pizzas.price*orders_details.quantity) as top_selling_pizzas
from pizza_types join pizzas
on     pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by top_selling_pizzas desc limit 3;
```



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

NEXT

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

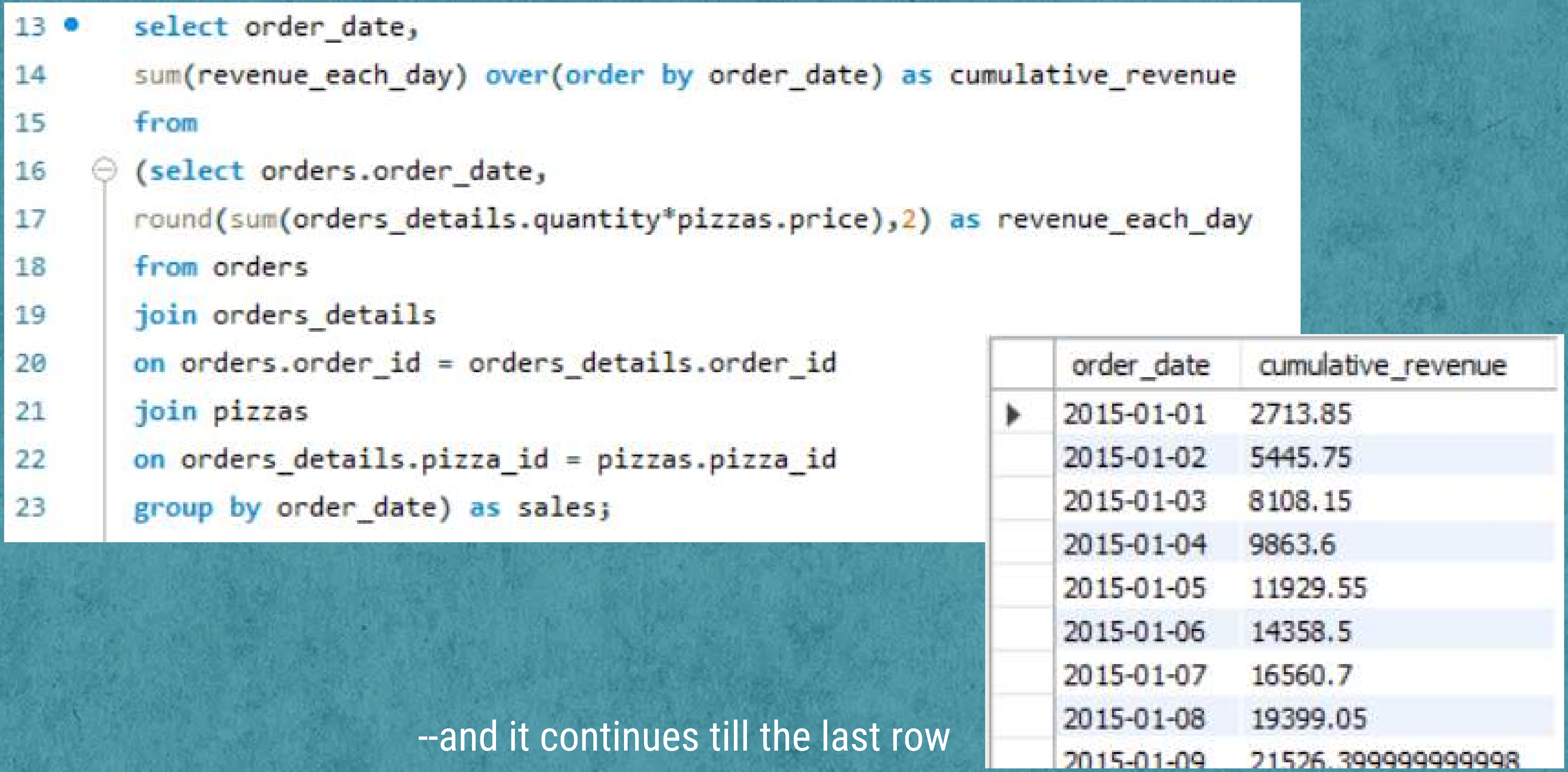


```sql
select pizza_types.category,
round((round(sum(orders_details.quantity*pizzas.price),2) /
(SELECT ROUND(SUM(orders_details.quantity * pizzas.price),2) AS total_sales
FROM orders_details JOIN pizzas
ON pizzas.pizza_id = orders_details.pizza_id))*100,2) as revenue
from pizza_types
join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group by category
order by revenue desc;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

NEXT

13. Analyze the cumulative revenue generated over time.



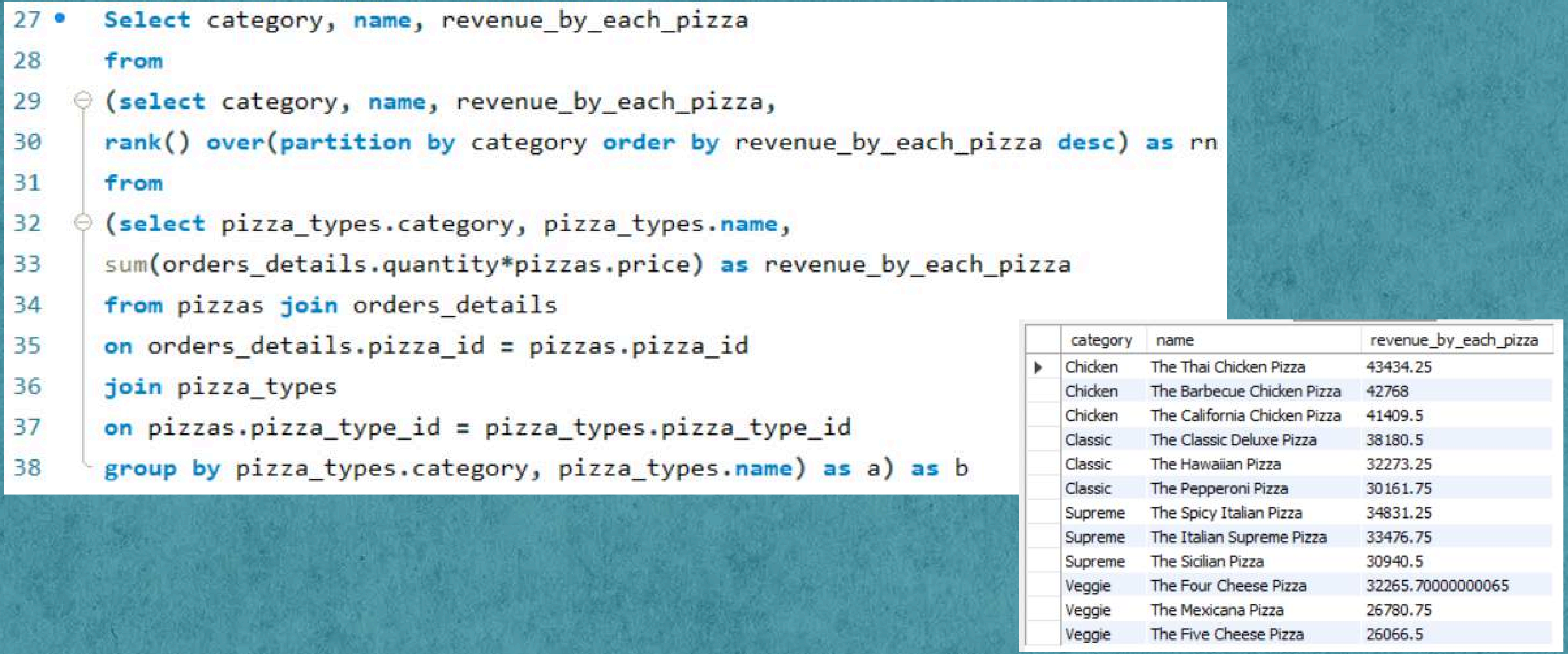
```sql
select order_date,
sum(revenue_each_day) over(order by order_date) as cumulative_revenue
from
(select orders.order_date,
round(sum(orders_details.quantity*pizzas.price),2) as revenue_each_day
from orders
join orders_details
on orders.order_id = orders_details.order_id
join pizzas
on orders_details.pizza_id = pizzas.pizza_id
group by order_date) as sales;
```

order_date	cumulative_revenue
2015-01-01	2713.85
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	21526.399999999998

--and it continues till the last row

NEXT

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



```
Select category, name, revenue_by_each_pizza
from
(select category, name, revenue_by_each_pizza,
rank() over(partition by category order by revenue_by_each_pizza desc) as rn
from
(select pizza_types.category, pizza_types.name,
sum(orders_details.quantity*pizzas.price) as revenue_by_each_pizza
from pizzas join orders_details
on orders_details.pizza_id = pizzas.pizza_id
join pizza_types
on pizzas.pizza_type_id = pizza_types.pizza_type_id
group by pizza_types.category, pizza_types.name) as a) as b
```

category	name	revenue_by_each_pizza
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

NEXT

→ For more content like this, visit my GitHub.

→ To discuss collaboration opportunities, contact me at: *abdurrehman.riaz11@gmail.com.*

Thank you!

-by: Abdur Rehman

2024