PIZZA SALES REPORT

05 JULY 2024

BY TUSHAR PAL

INTRODUCTION

Hi my name is TUSHAR PAL and Welcome to our Sales Report Presentation. Today, we delve into a comprehensive overview of our sales performance, exploring the highs, challenges, and strategic insights that have shaped our journey. In this project we used sql queries to answer the question that are related to pizza sales.

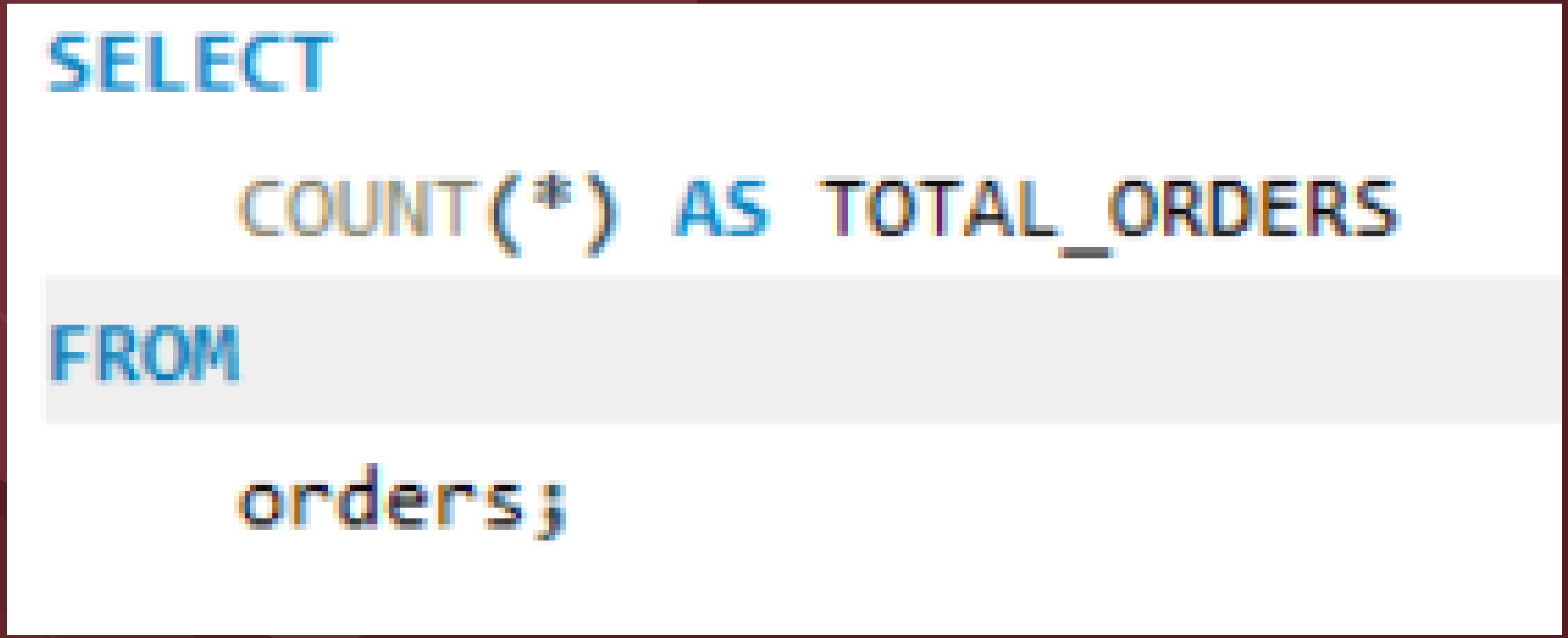
Data set link -
https://github.com/Tusharpal000/pizza_sales_sql

BASIC QUESTION

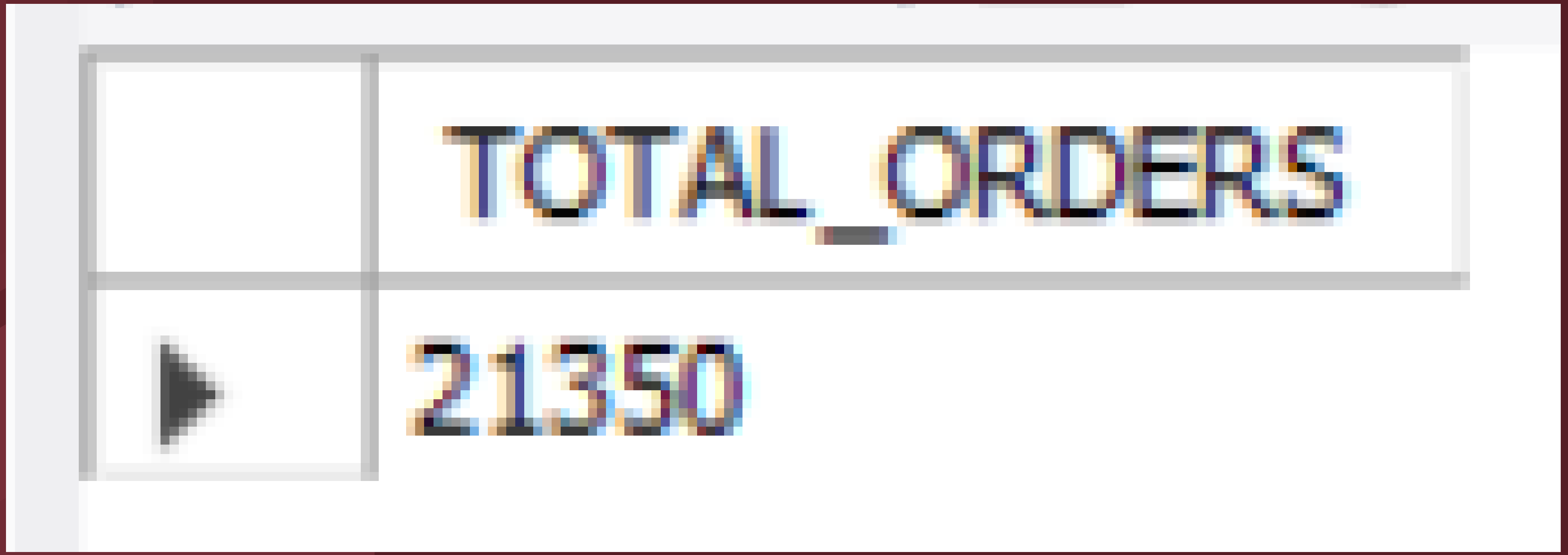
1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.

4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities

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

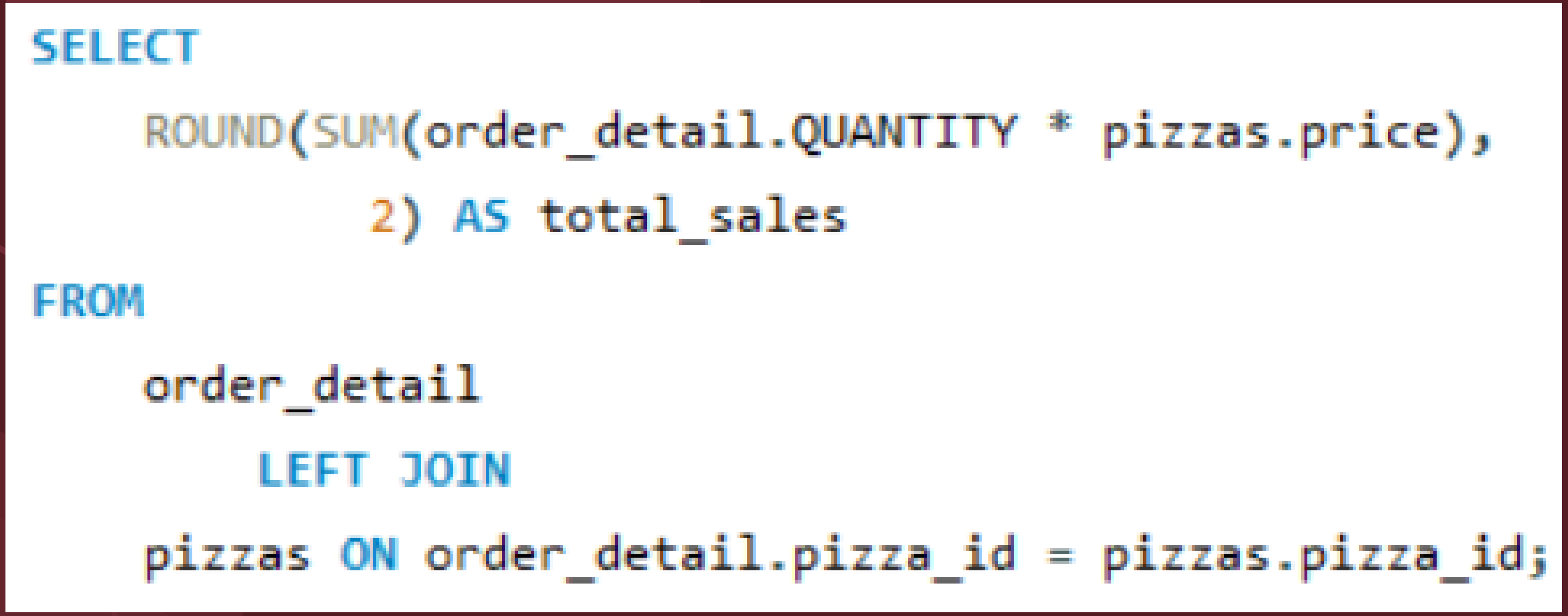


```
SELECT
    COUNT(*) AS TOTAL_ORDERS
FROM
    orders;
```

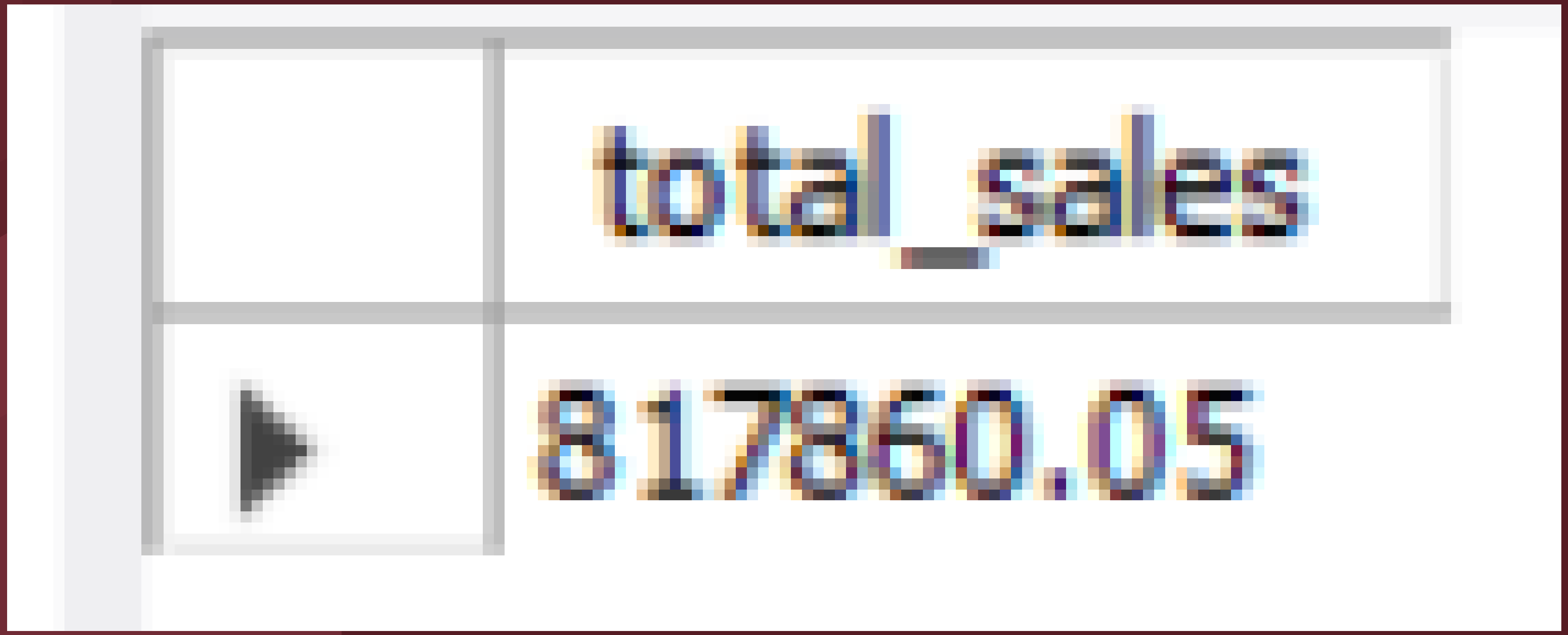


	TOTAL_ORDERS
▶	21350

-- Calculate the total revenue generated from pizza sales.--

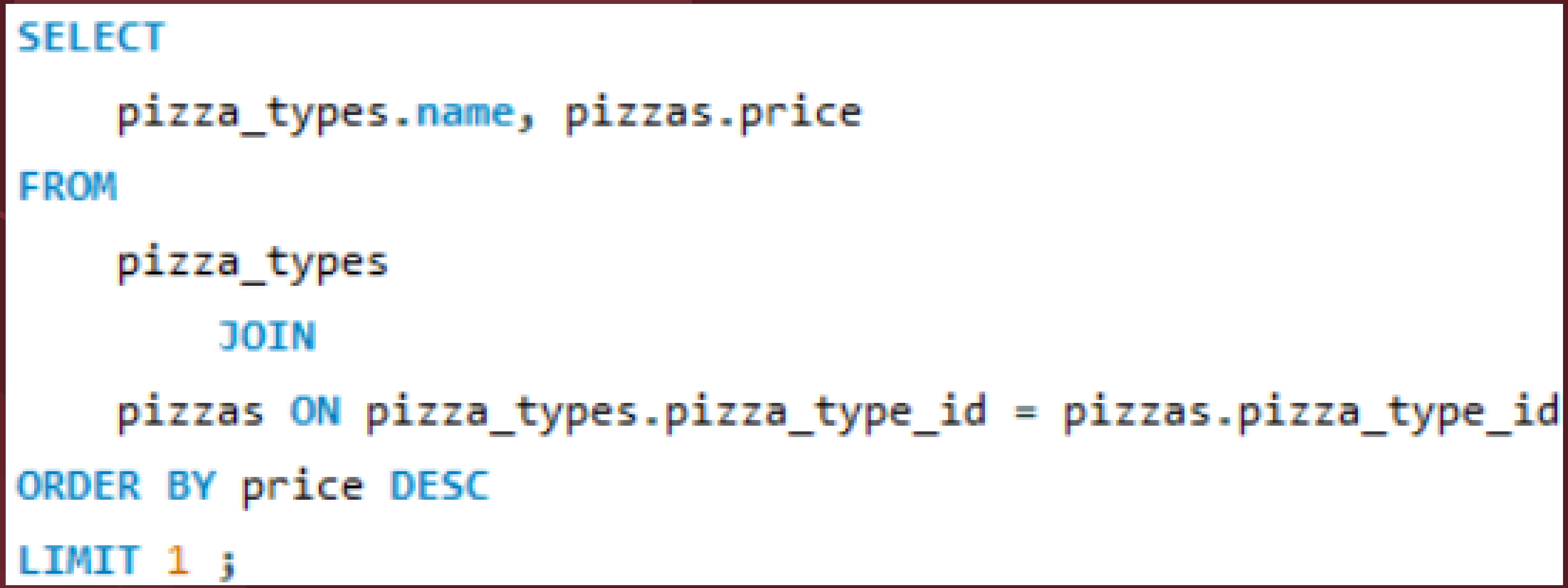


```
SELECT
    ROUND(SUM(order_detail.QUANTITY * pizzas.price),
            2) AS total_sales
FROM
    order_detail
        LEFT JOIN
    pizzas ON order_detail.pizza_id = pizzas.pizza_id;
```

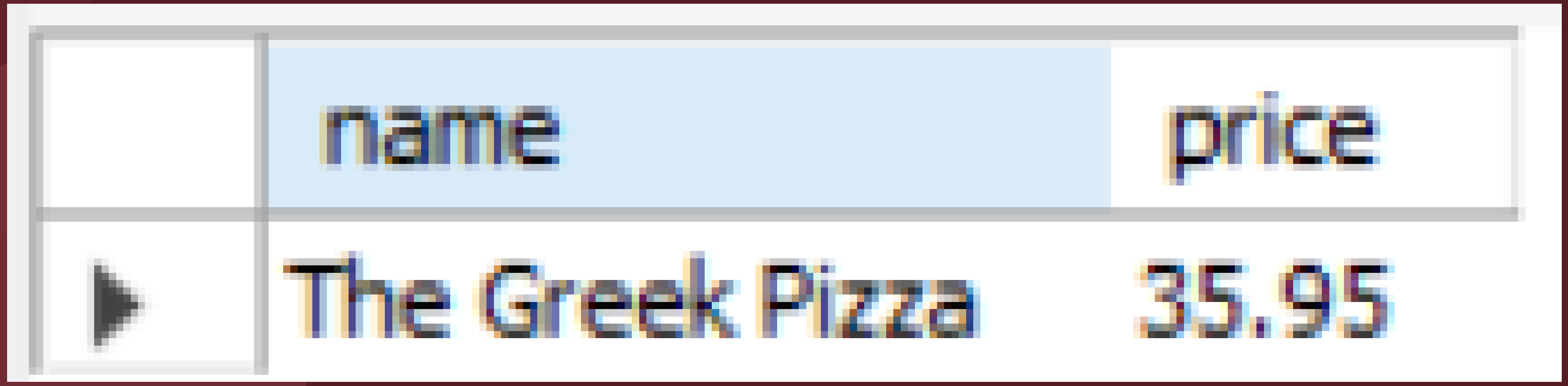


	total_sales
▶	817860.05

-- Identify the highest-priced pizza. --

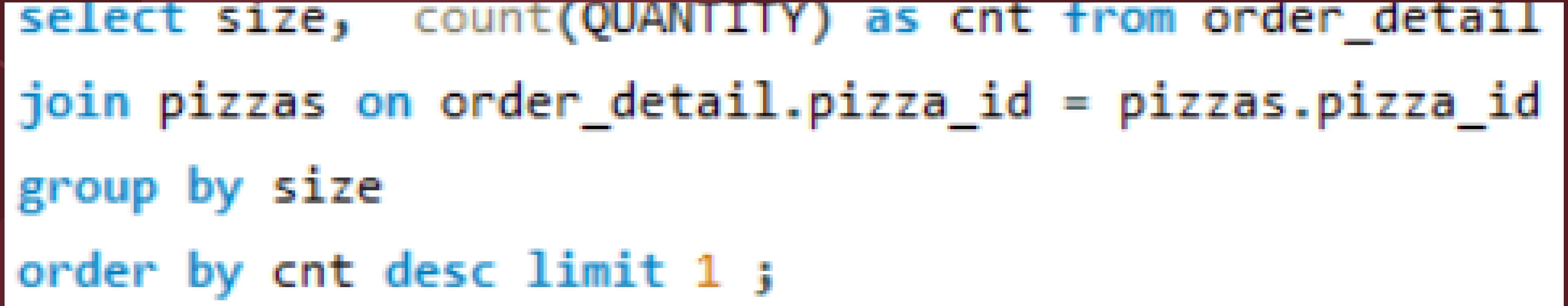


```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY price DESC
LIMIT 1 ;
```

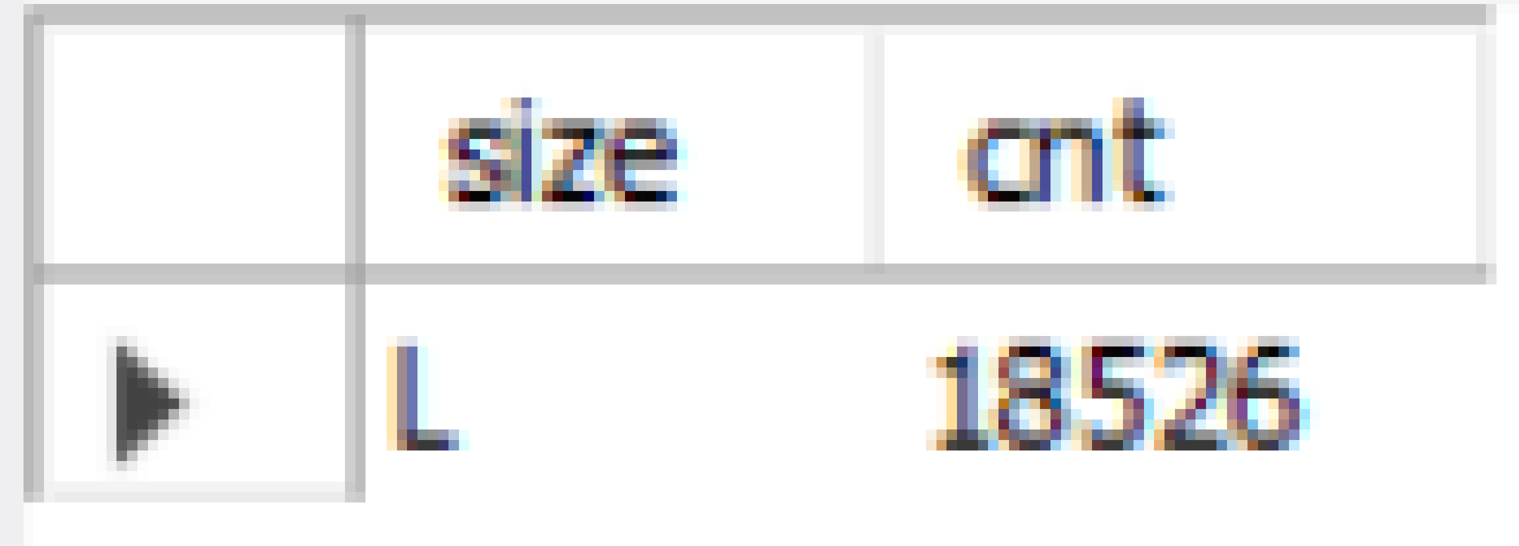


	name	price
▶	The Greek Pizza	35.95

-- Identify the most common pizza size ordered. --

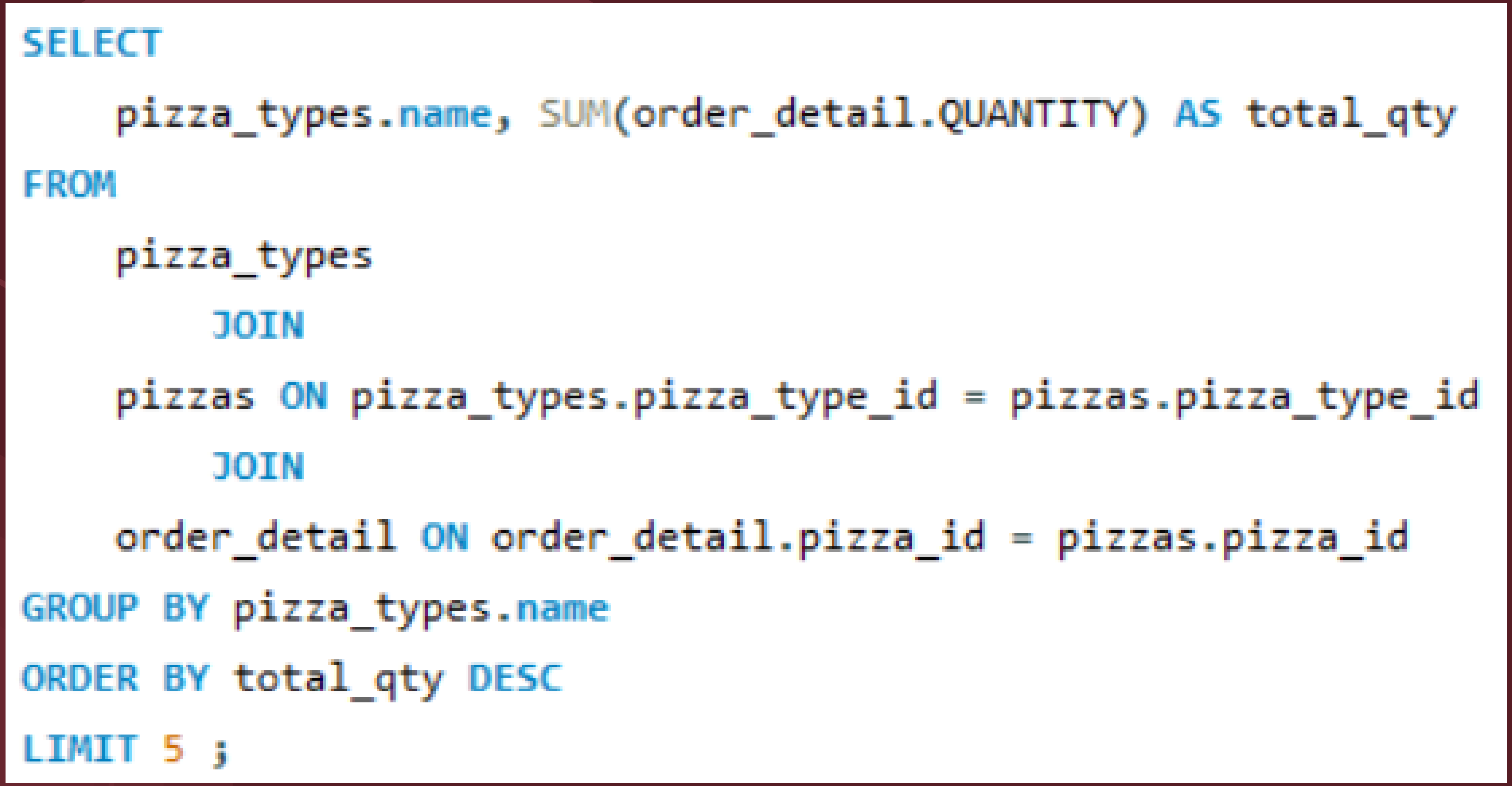


```
select size,  count(QUANTITY) as cnt from order_detail
join pizzas on order_detail.pizza_id = pizzas.pizza_id
group by size
order by cnt desc limit 1 ;
```

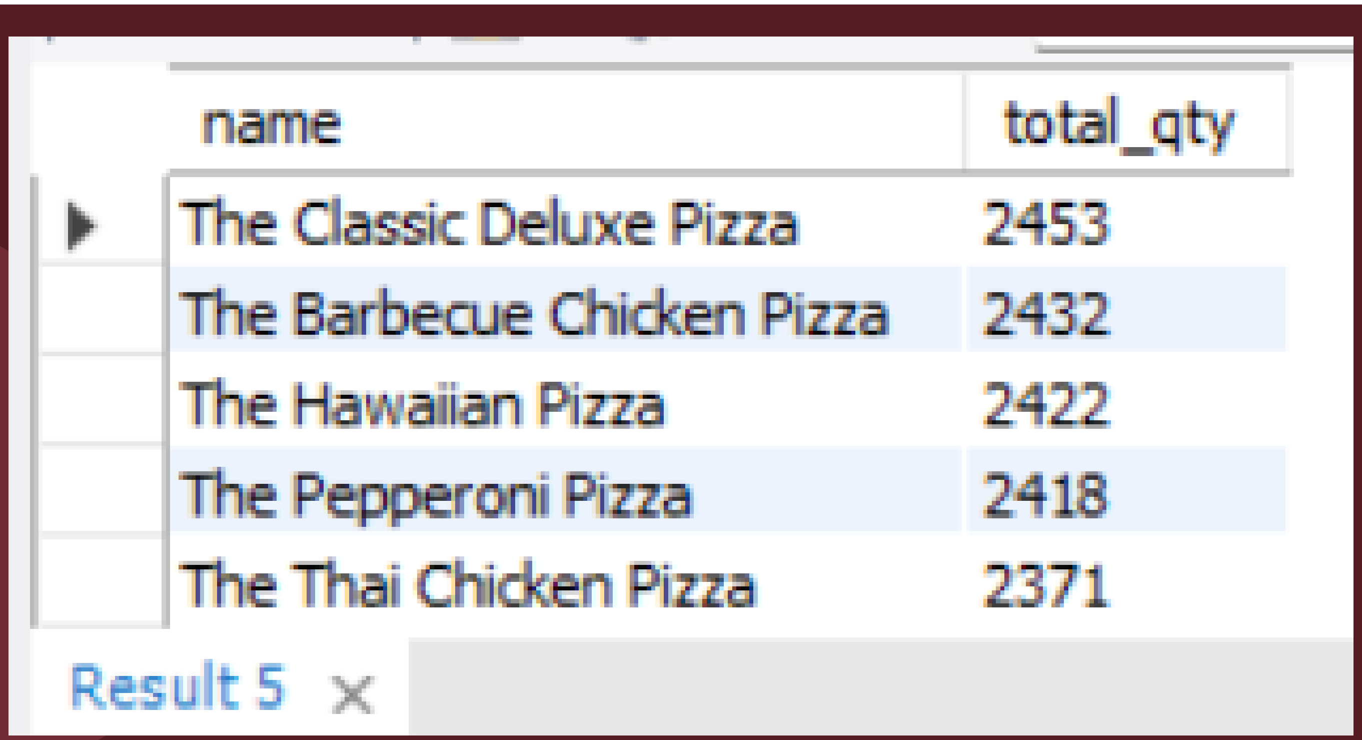


	size	cnt
▶	L	18526

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



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



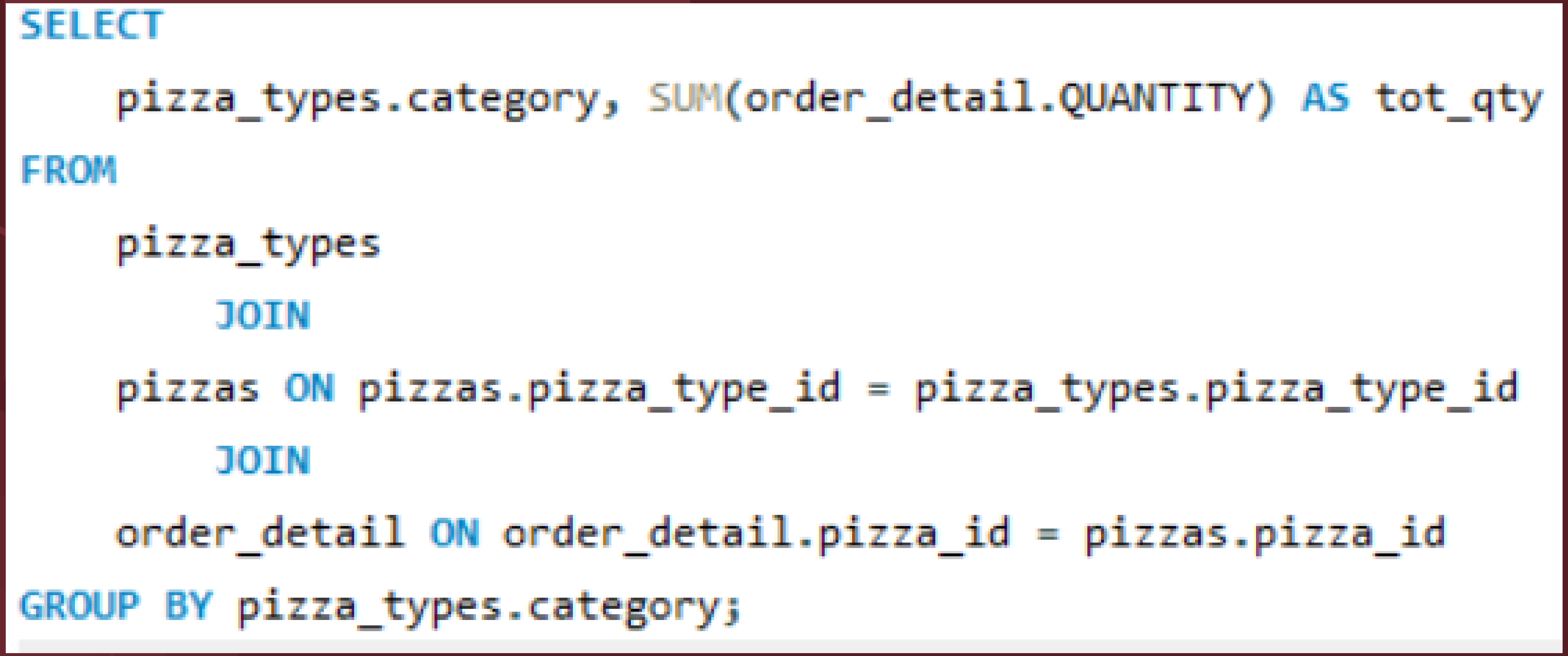
name	total_qty
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Ingoude Company

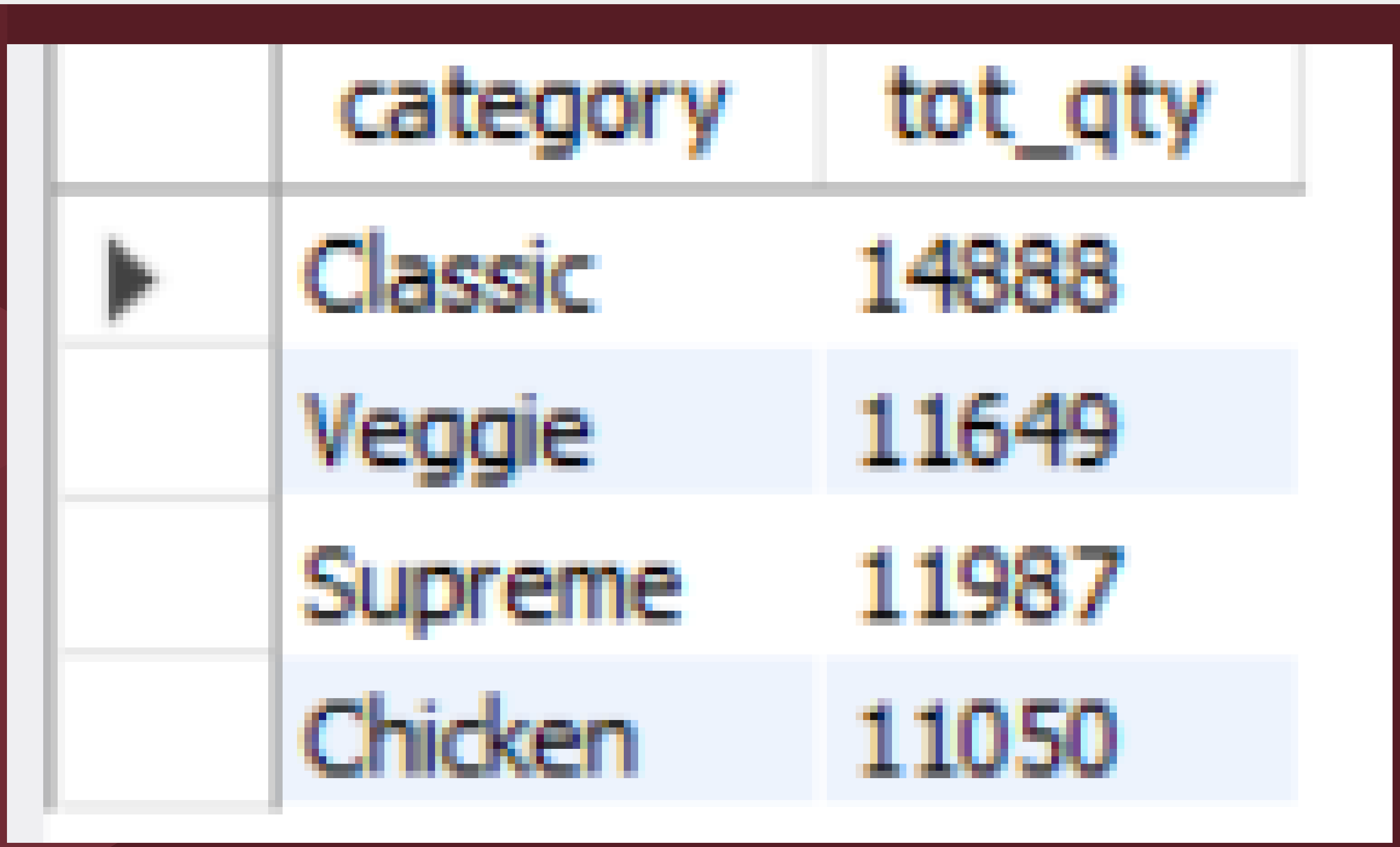
INTERMEDIATE QUESTION

1. Join the necessary tables to find the total quantity of each pizza category ordered.
2. Determine the distribution of orders by hour of the day.
3. Join relevant tables to find the category-wise distribution of pizzas.
4. Group the orders by date and calculate the average number of pizzas ordered per day.
5. Determine the top 3 most ordered pizza types based on revenue.

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

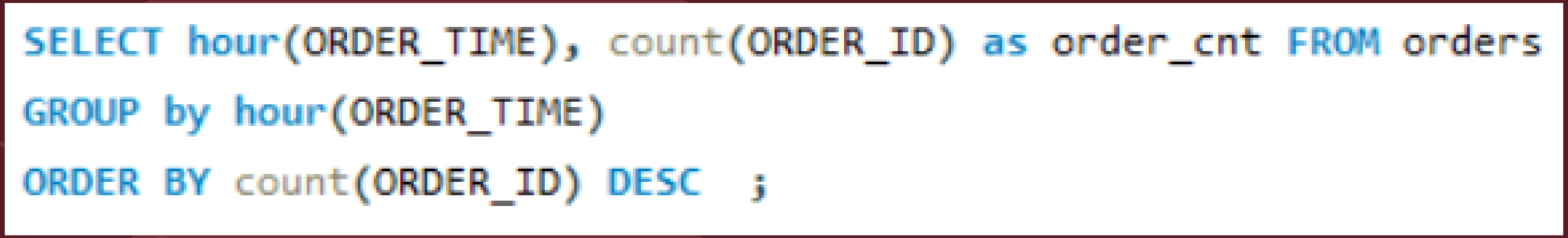


```
SELECT
    pizza_types.category, SUM(order_detail.QUANTITY) AS tot_qty
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_detail ON order_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category;
```

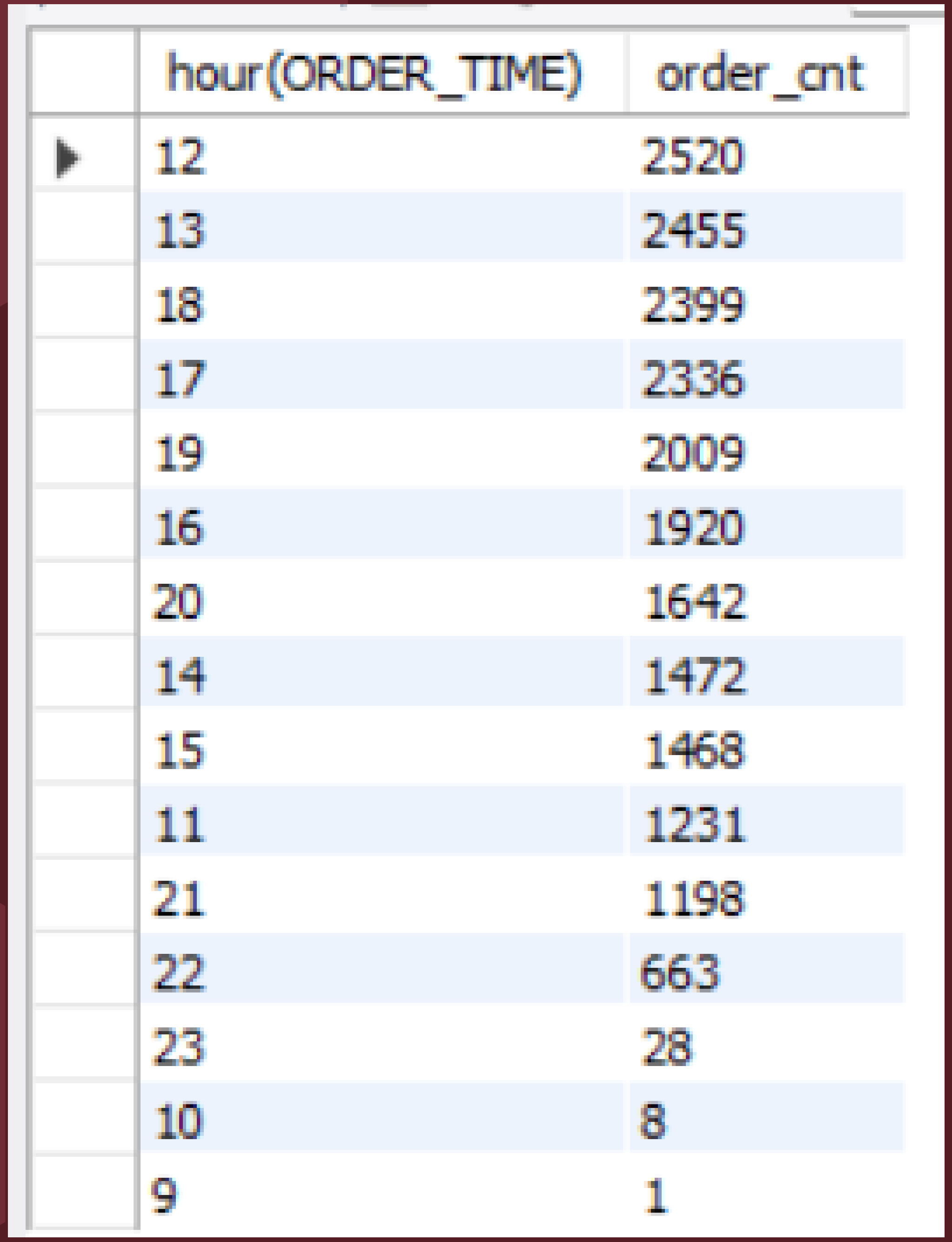


	category	tot_qty
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

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

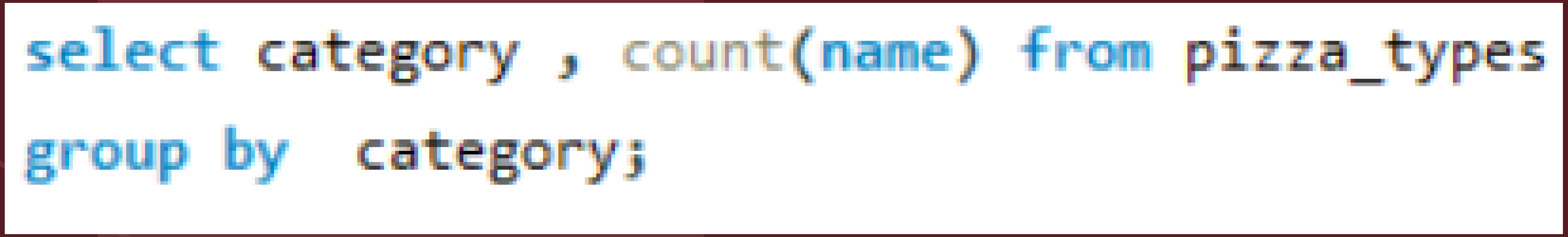


```
SELECT hour(ORDER_TIME), count(ORDER_ID) as order_cnt FROM orders
GROUP by hour(ORDER_TIME)
ORDER BY count(ORDER_ID) DESC  ;
```

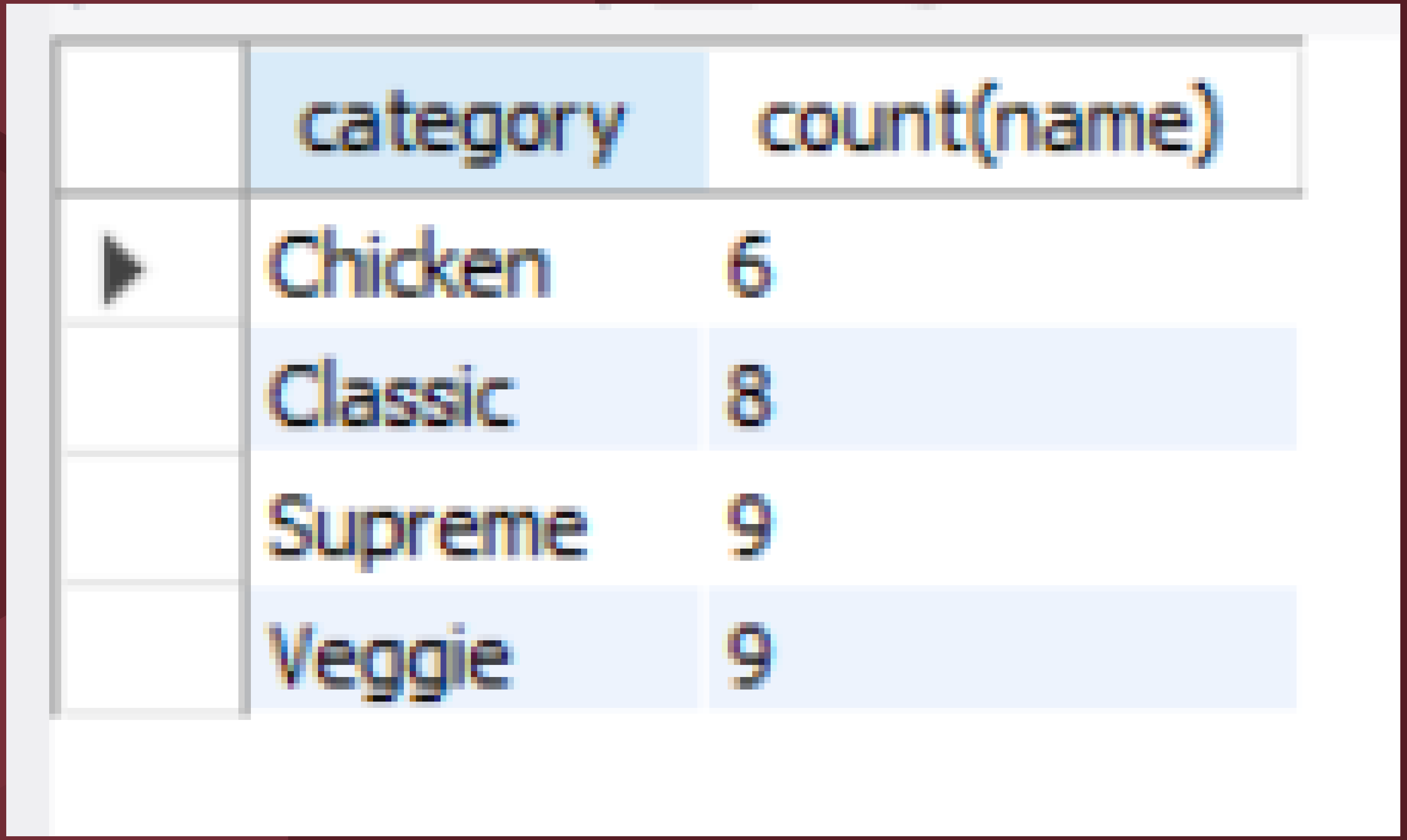


hour(ORDER_TIME)	order_cnt
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231
21	1198
22	663
23	28
10	8
9	1

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

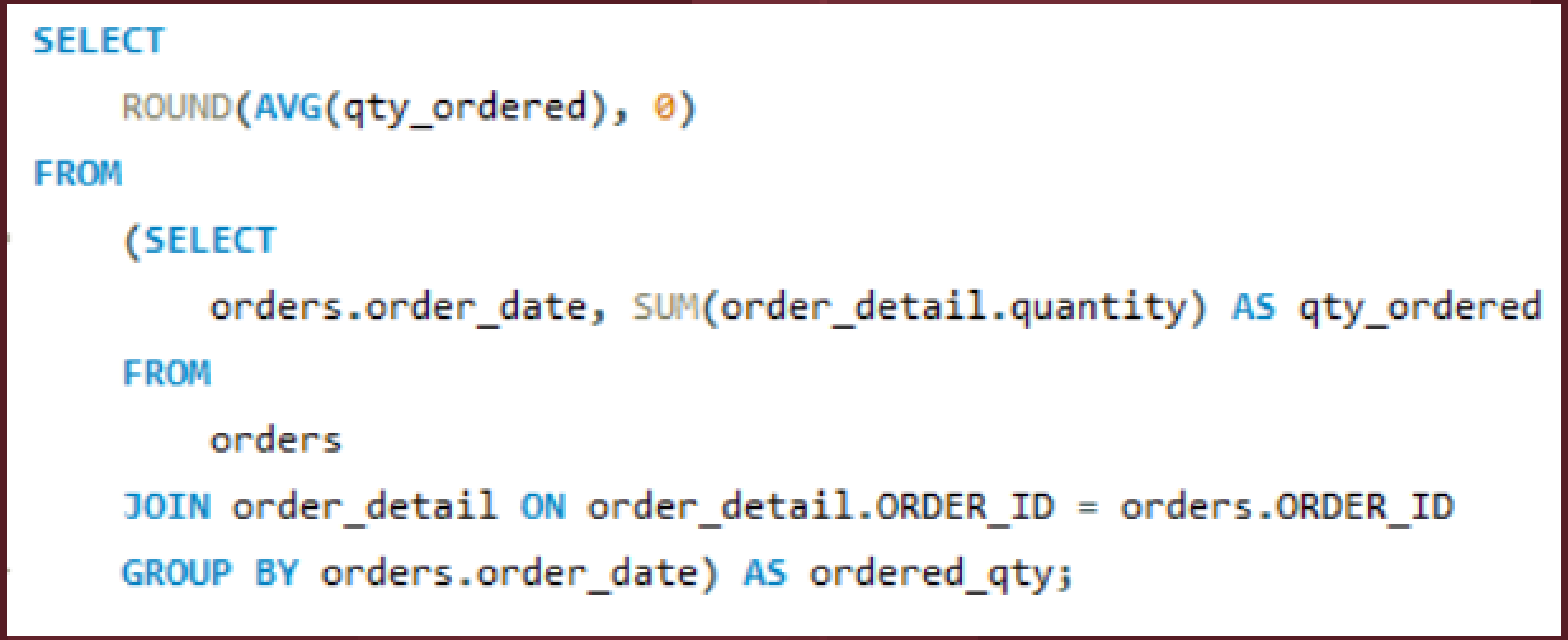


```
select category , count(name) from pizza_types
group by  category;
```

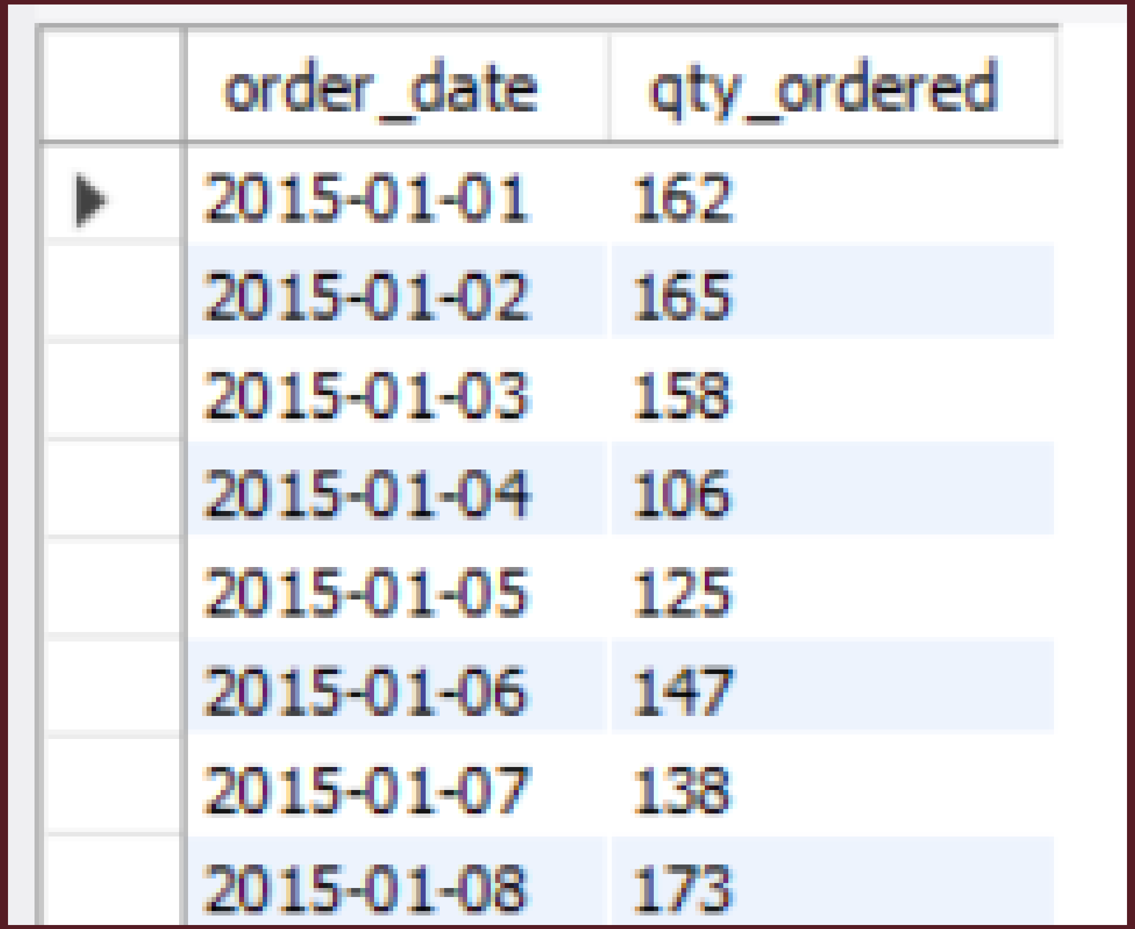


category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

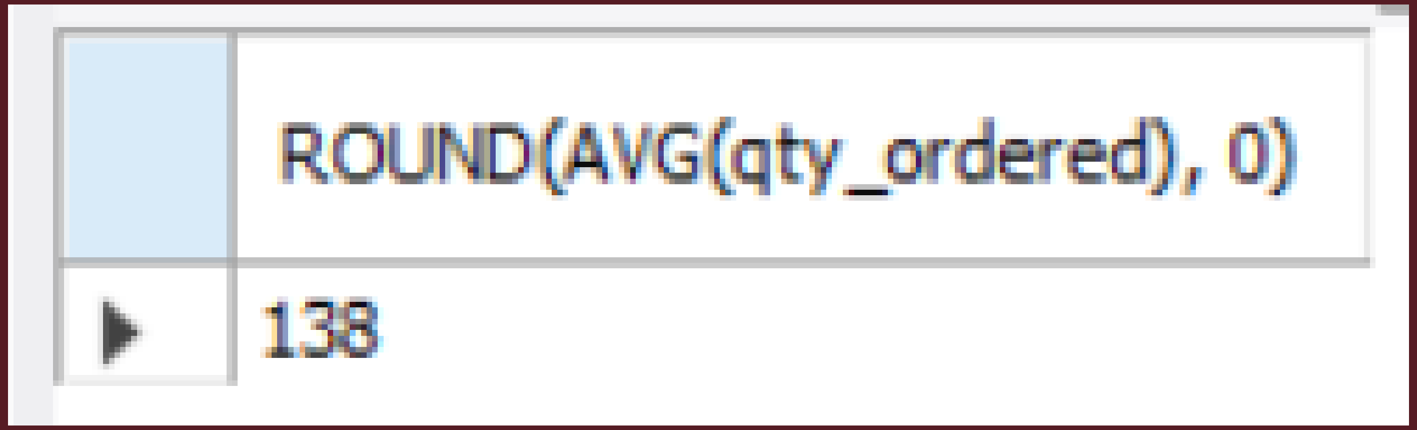
-- Group the orders by date and calculate the average number of pizzas ordered per day. --



```
SELECT
    ROUND(AVG(qty_ordered), 0)
FROM
    (SELECT
        orders.order_date, SUM(order_detail.quantity) AS qty_ordered
    FROM
        orders
    JOIN order_detail ON order_detail.ORDER_ID = orders.ORDER_ID
    GROUP BY orders.order_date) AS ordered_qty;
```

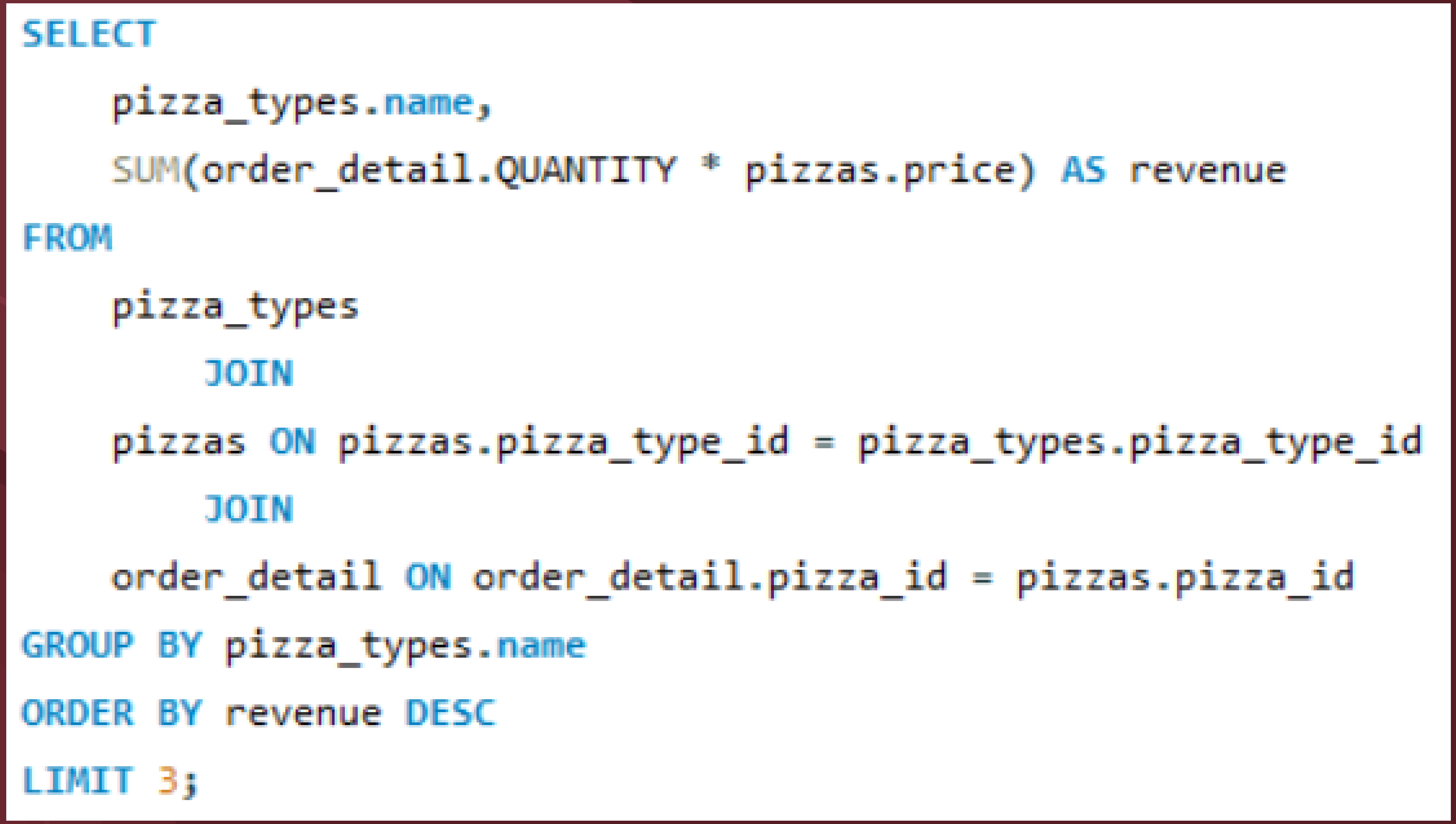


	order_date	qty_ordered
▶	2015-01-01	162
	2015-01-02	165
	2015-01-03	158
	2015-01-04	106
	2015-01-05	125
	2015-01-06	147
	2015-01-07	138
	2015-01-08	173

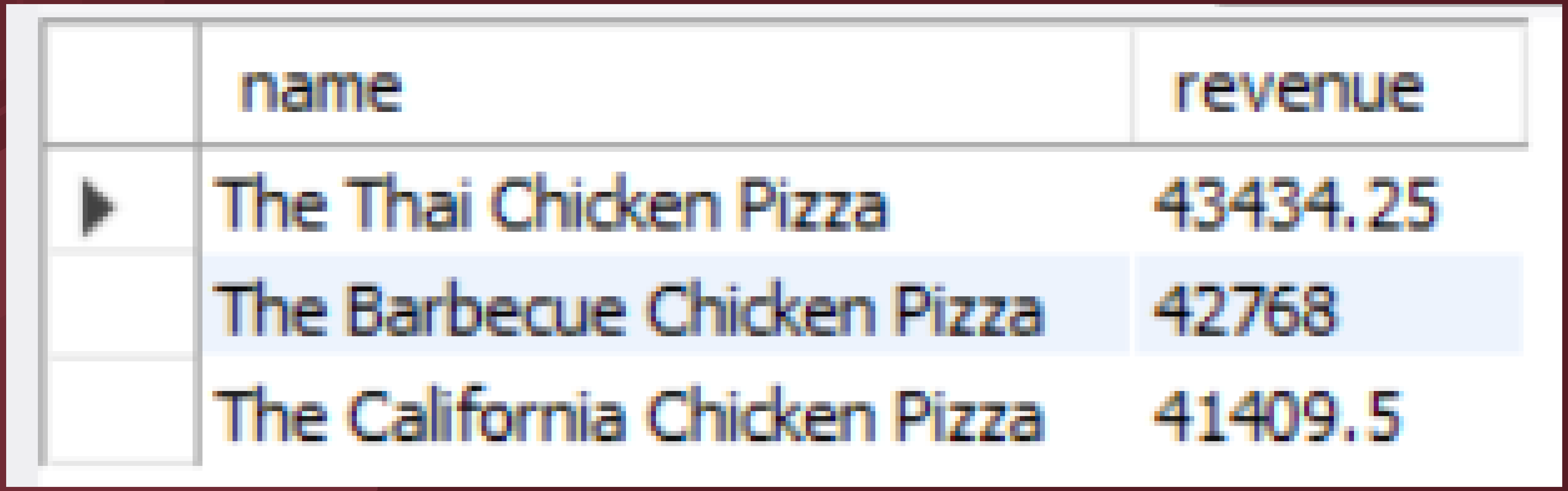


	ROUND(AVG(qty_ordered), 0)
▶	138

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



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

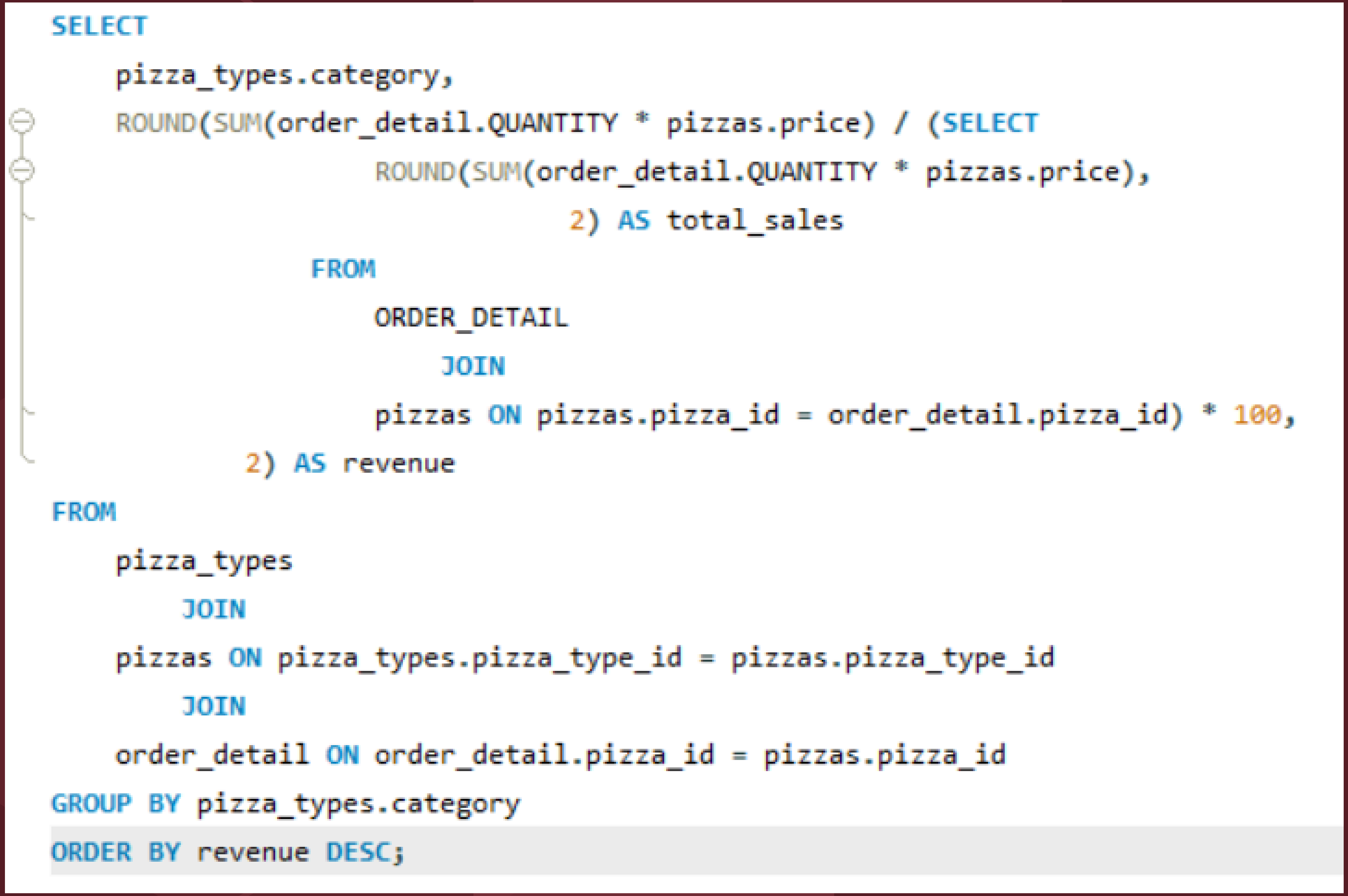


	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

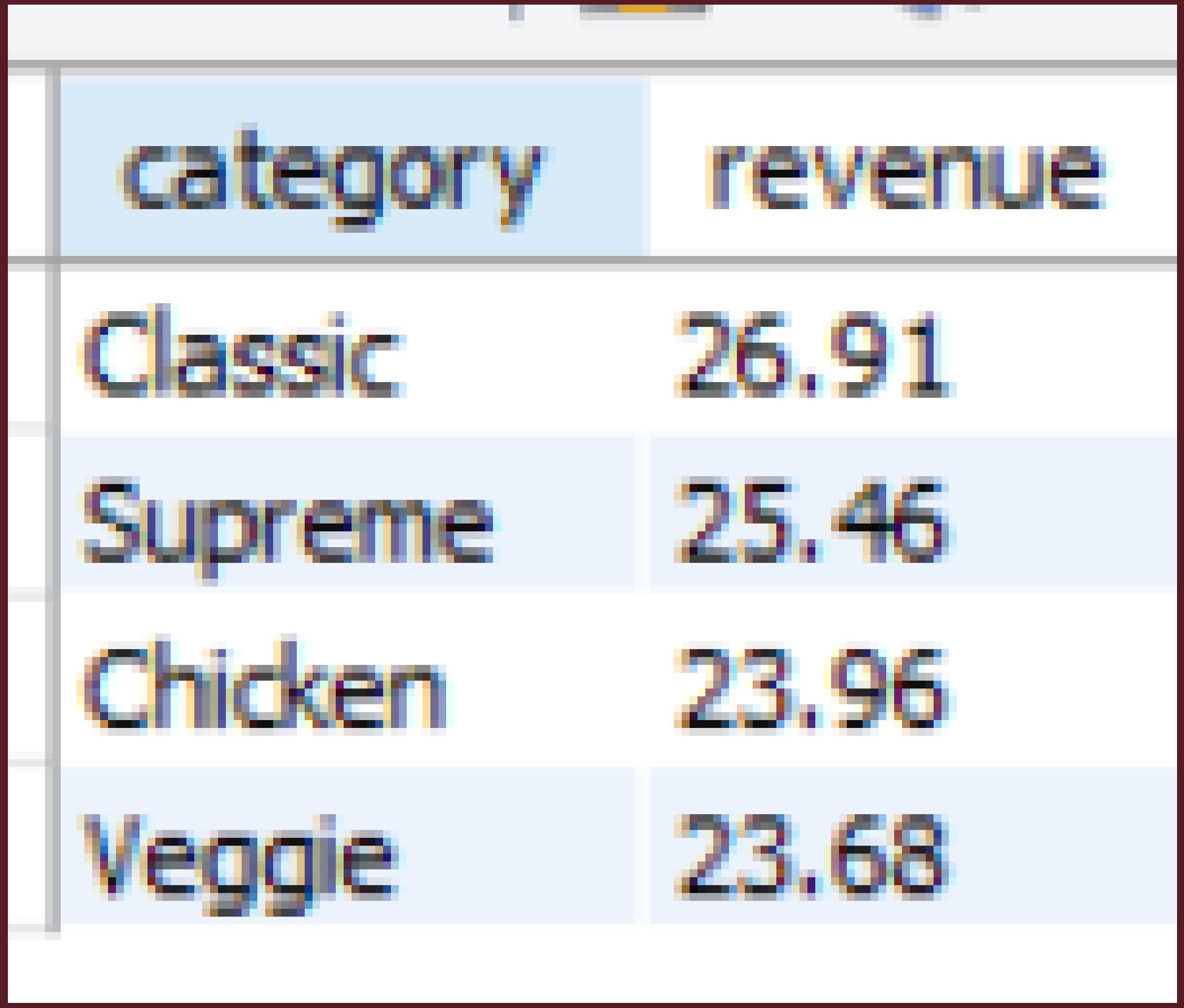
ADVANCE QUESTION

1. Calculate the percentage contribution of each pizza type to total revenue.
2. Analyze the cumulative revenue generated over time.
3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

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

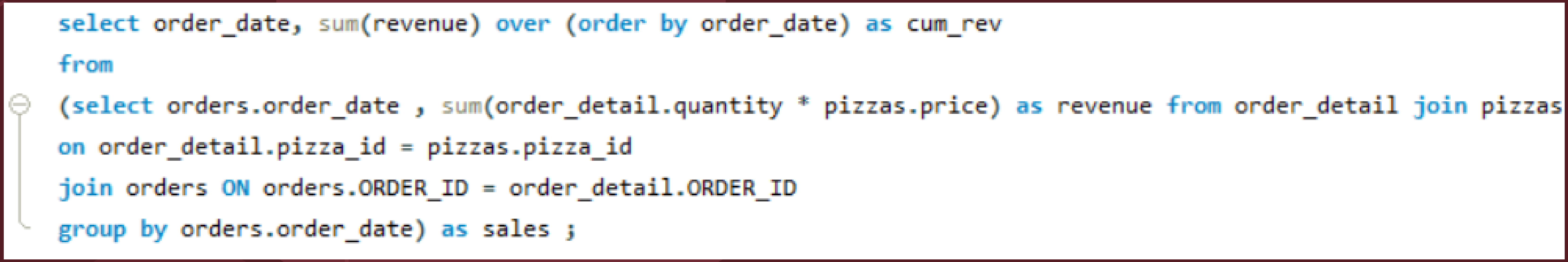


```
SELECT
    pizza_types.category,
    ROUND(SUM(order_detail.QUANTITY * pizzas.price) / (SELECT
                    ROUND(SUM(order_detail.QUANTITY * pizzas.price),
                                2) AS total_sales
                FROM
                    ORDER_DETAIL
                        JOIN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

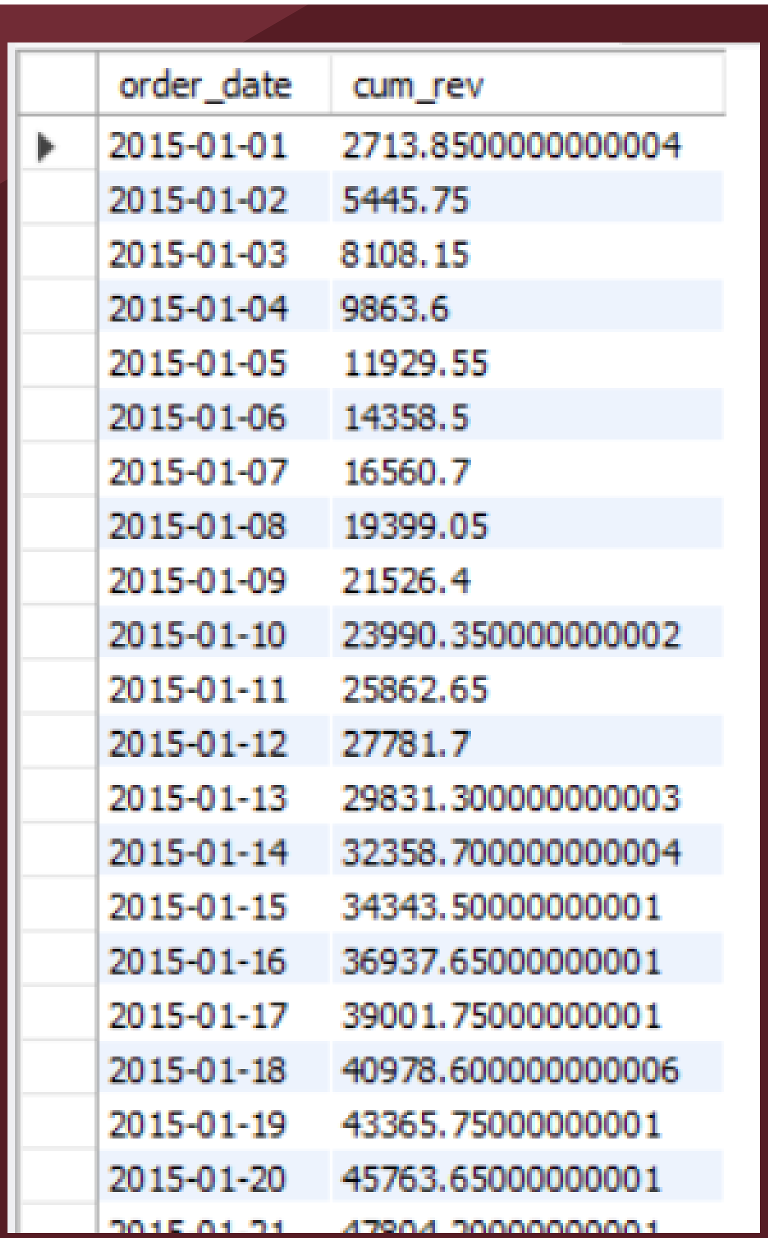


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

-- Analyze the cumulative revenue generated over time. --

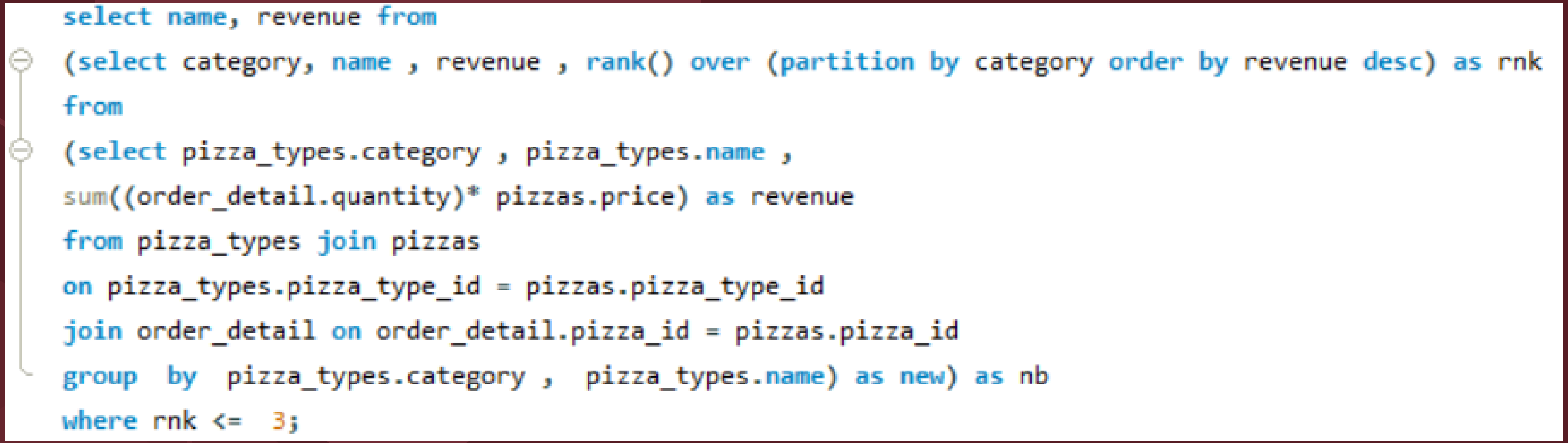


```
select order_date, sum(revenue) over (order by order_date) as cum_rev
from
(select orders.order_date , sum(order_detail.quantity * pizzas.price) as revenue from order_detail join pizzas
on order_detail.pizza_id = pizzas.pizza_id
join orders ON orders.ORDER_ID = order_detail.ORDER_ID
group by orders.order_date) as sales ;
```

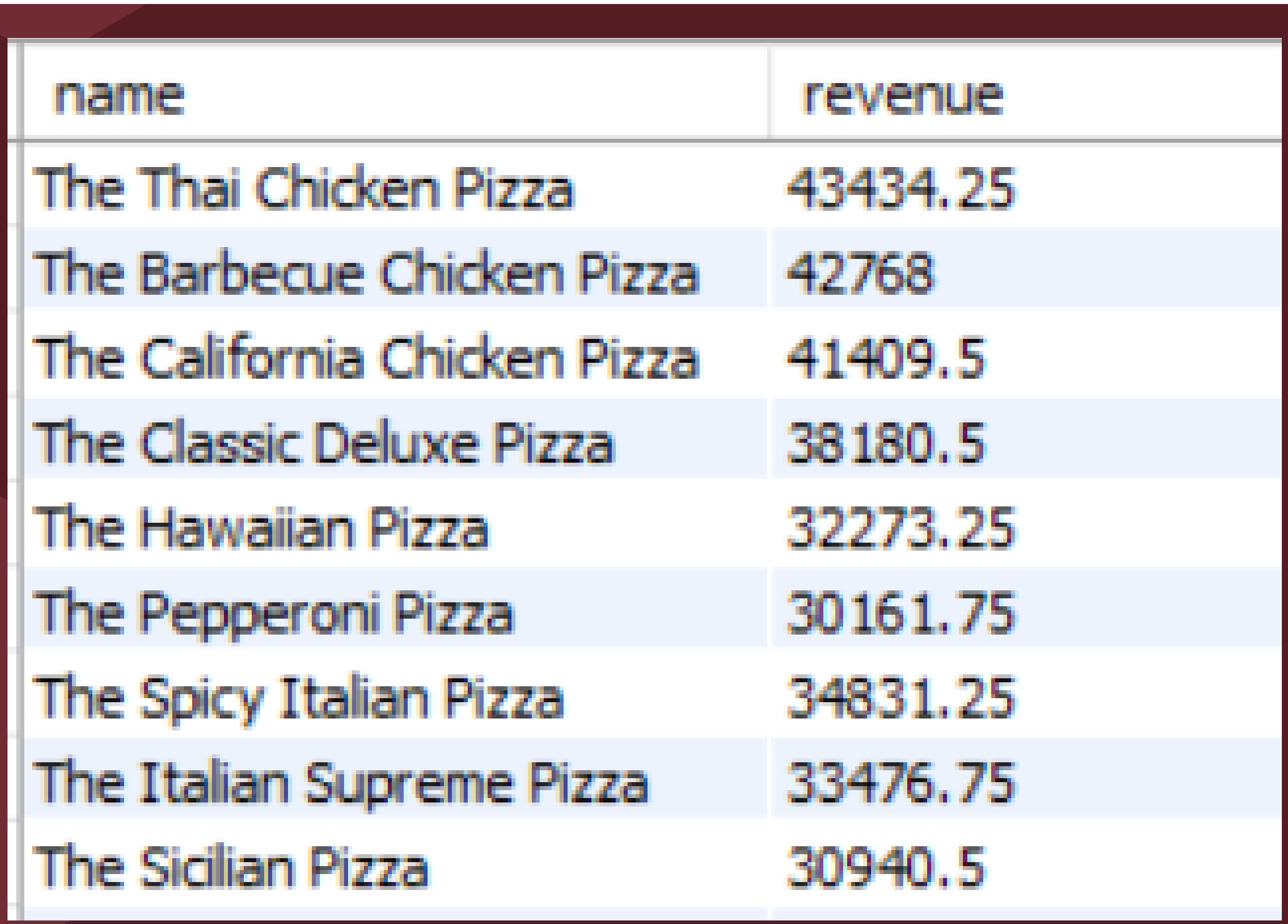


order_date	cum_rev
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001

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



```
select name, revenue from
(select category, name , revenue , rank() over (partition by category order by revenue desc) as rnk
from
(select pizza_types.category , pizza_types.name ,
sum((order_detail.quantity)* pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_detail on order_detail.pizza_id = pizzas.pizza_id
group  by  pizza_types.category ,  pizza_types.name) as new) as nb
where rnk <=  3;
```



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
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5

THANK YOU

05 JULY, 2024