Delicious Pizza for Everyone!

PIZZA

SALES DATA ANALYSIS

Hello!

Embark on a flavorful journey through my pizza sales data analysis project, where we dive deep into the crust, sauce, and cheese of business operations.

MEGHA BORA

about project

Welcome to our presentation on pizza sales data analysis using MySQL! In this session, we will delve into the world of pizza sales data and demonstrate how MySQL can serve as a powerful tool for extracting insights and making informed business decisions. we will showcase how MySQL's querying capabilities enable us to efficiently analyze large volumes of transactional data, uncovering valuable patterns and trends in customer behavior, product preferences, and sales performance.

SCHEMA

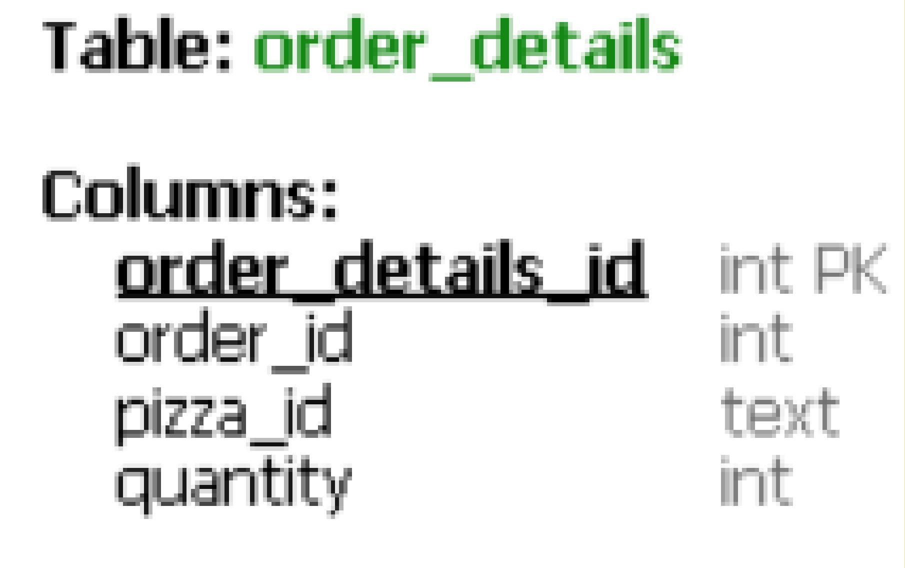


Table: order_details

Columns:

order_details_id	int PK
order_id	int
pizza_id	text
quantity	int

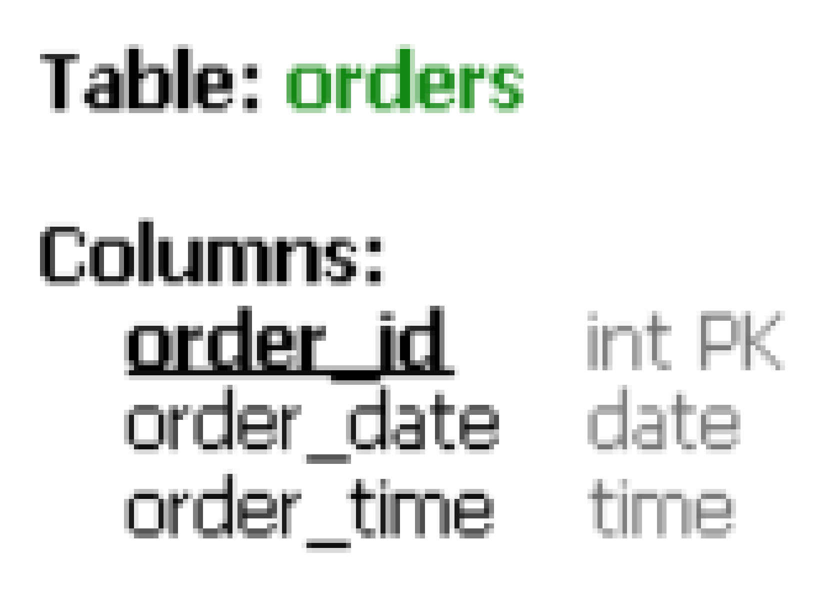


Table: orders

Columns:

order_id	int PK
order_date	date
order_time	time

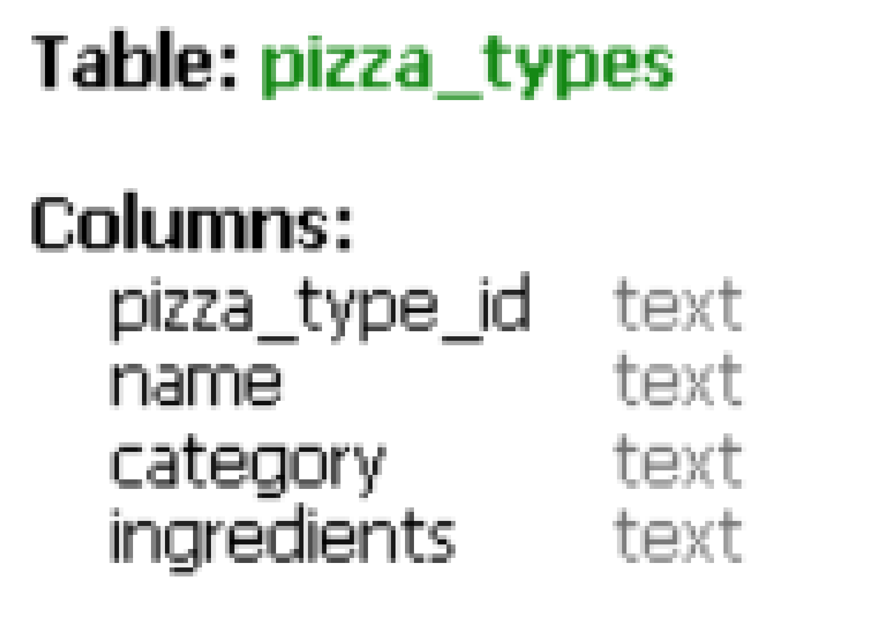


Table: pizza_types

Columns:

pizza_type_id	text
name	text
category	text
ingredients	text

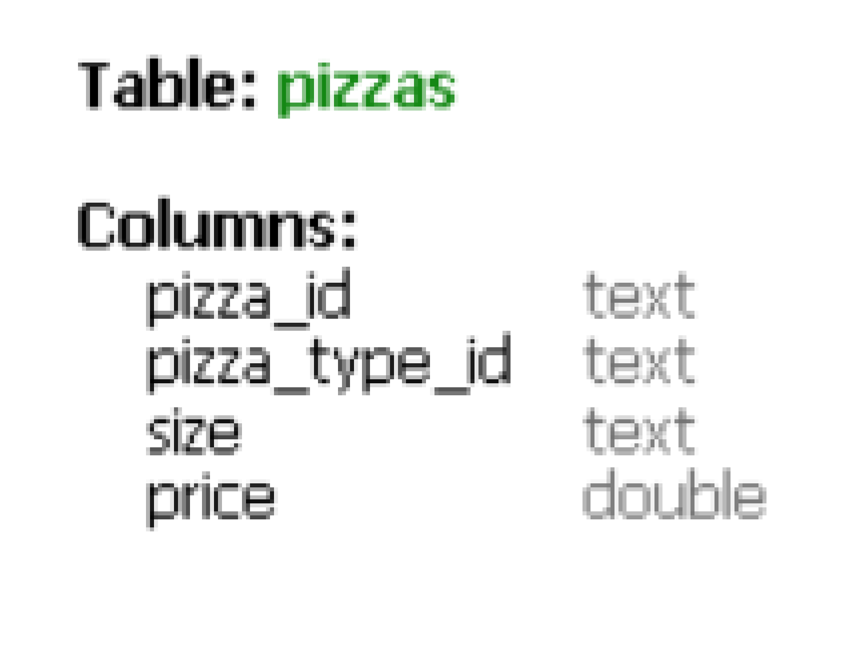
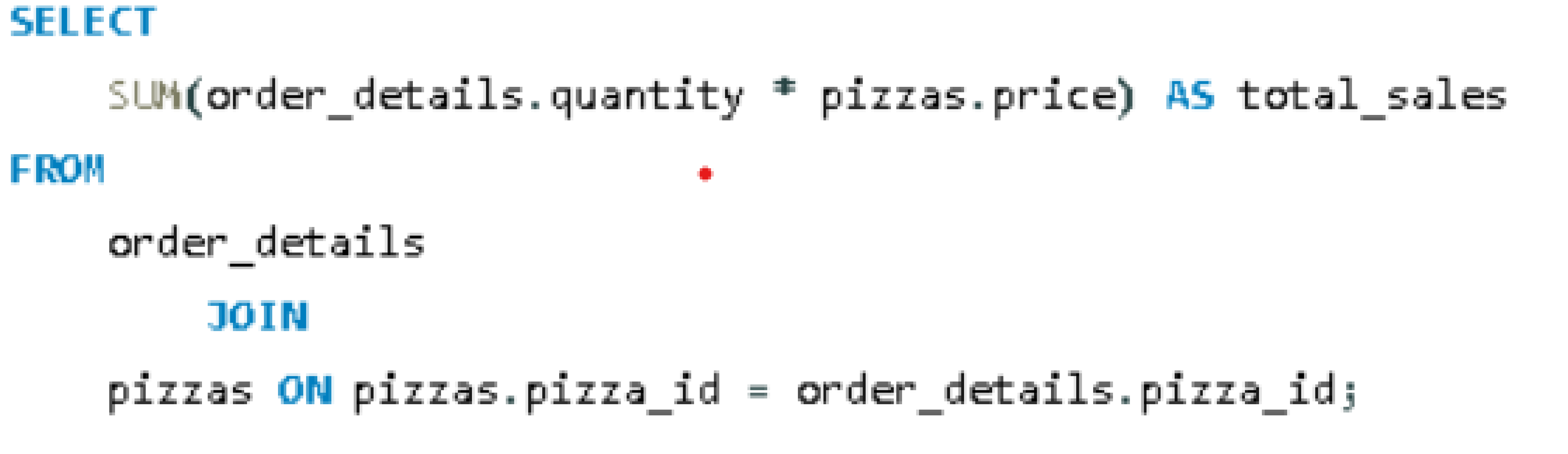


Table: pizzas

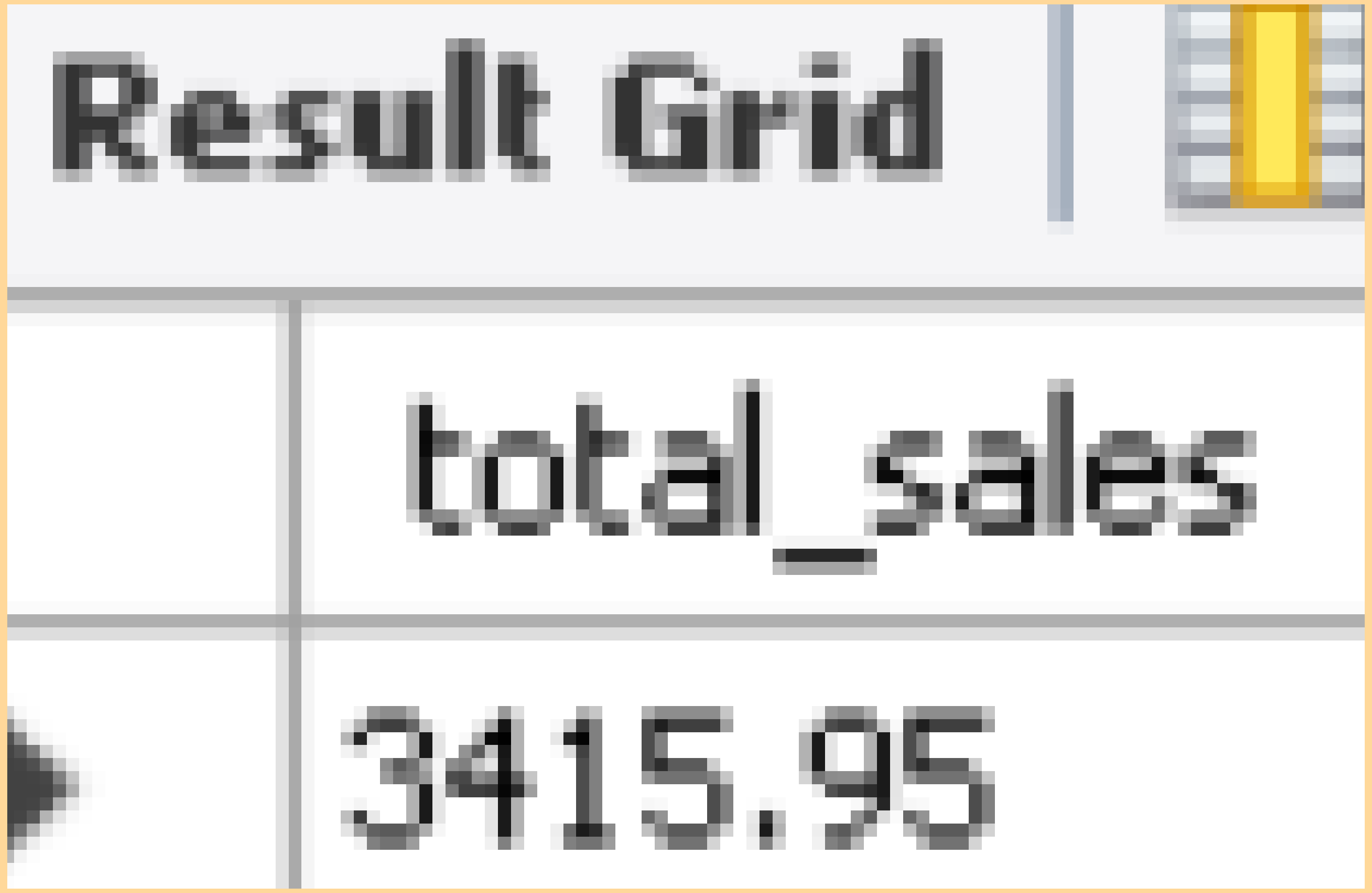
Columns:

pizza_id	text
pizza_type_id	text
size	text
price	double

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

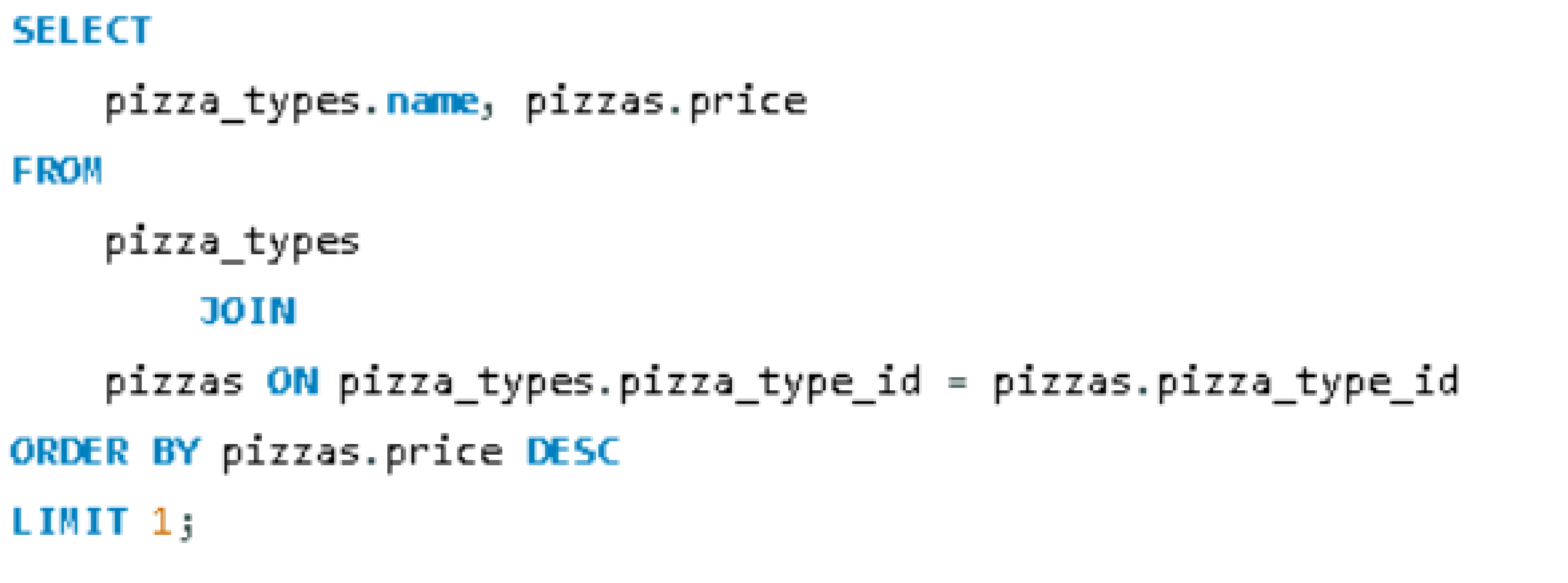


```
SELECT
    SUM(order_details.quantity * pizzas.price) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

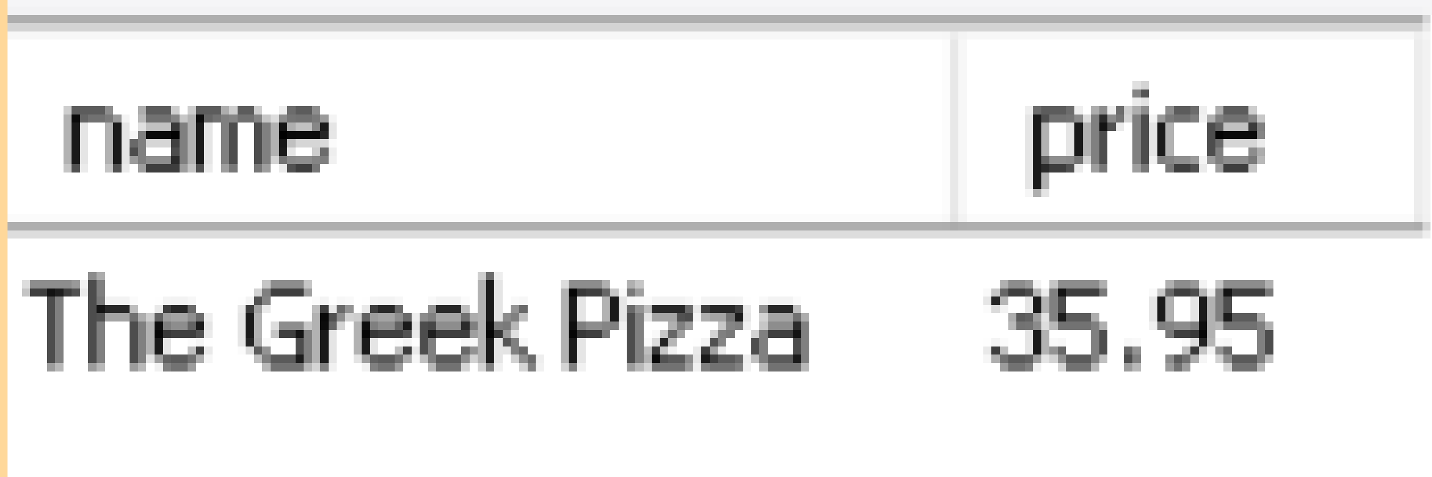


```
total_sales
3415.95
```

Identify the highest-priced pizza.

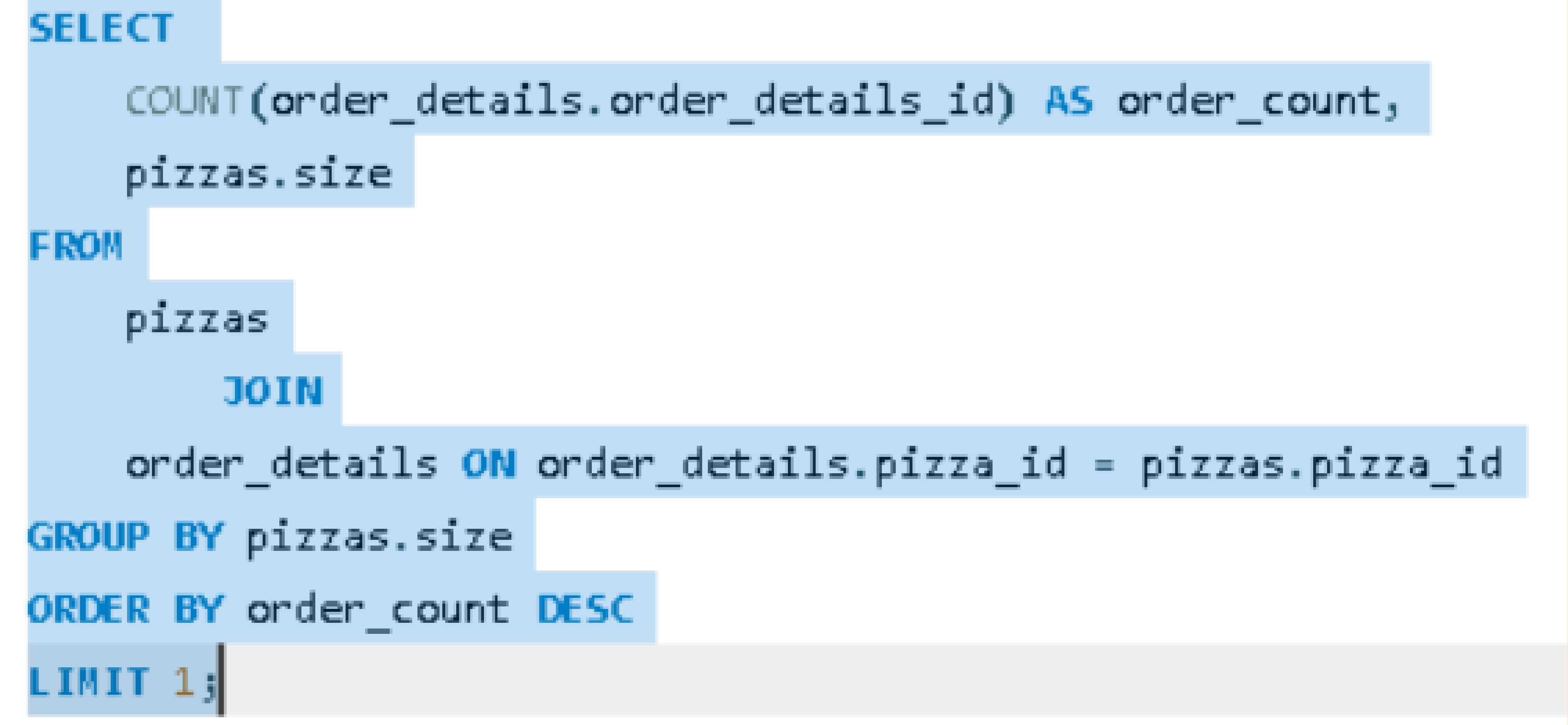


```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```



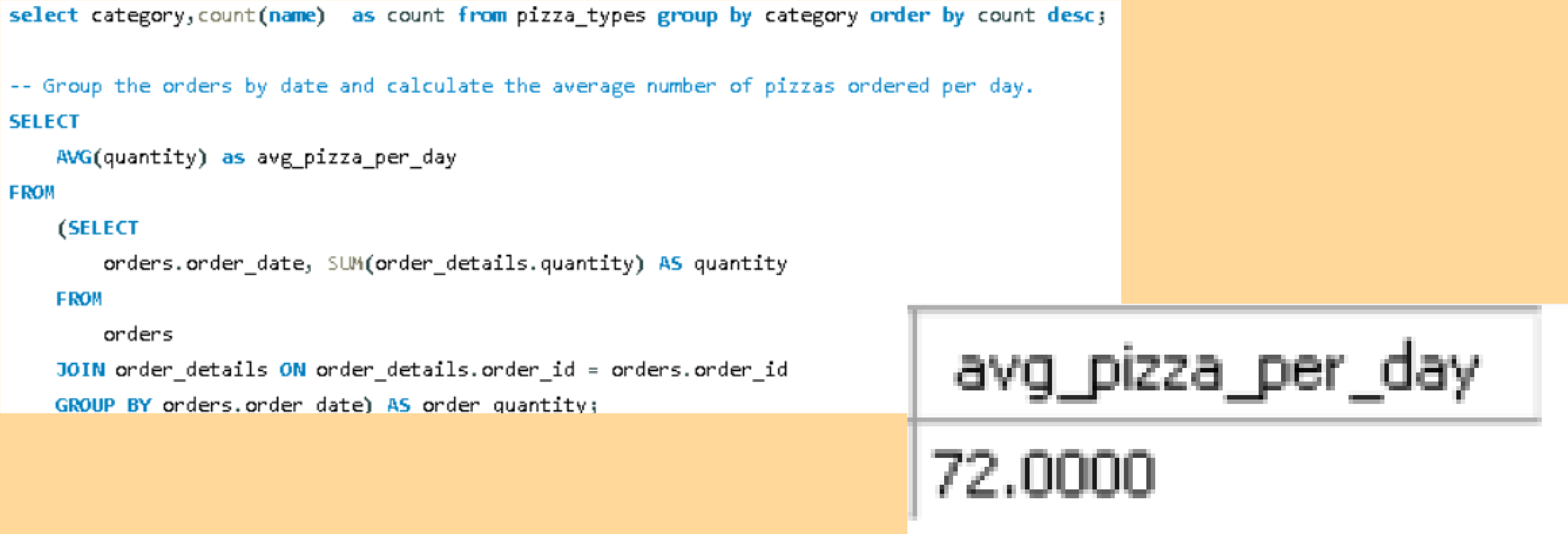
name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.



```sql
SELECT
    COUNT(order_details.order_details_id) AS order_count,
    pizzas.size
FROM
    pizzas
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

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

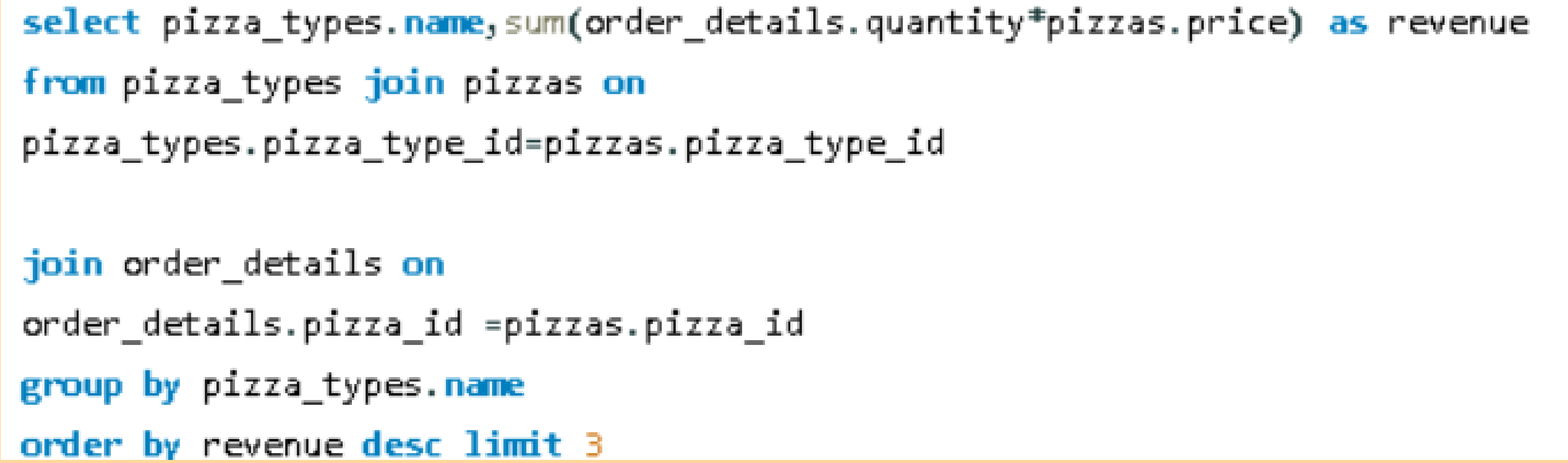


```
select category,count(name)  as count from pizza_types group by category order by count desc;


-- Group the orders by date and calculate the average number of pizzas ordered per day.
SELECT
    AVG(quantity) as avg_pizza_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON order_details.order_id = orders.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

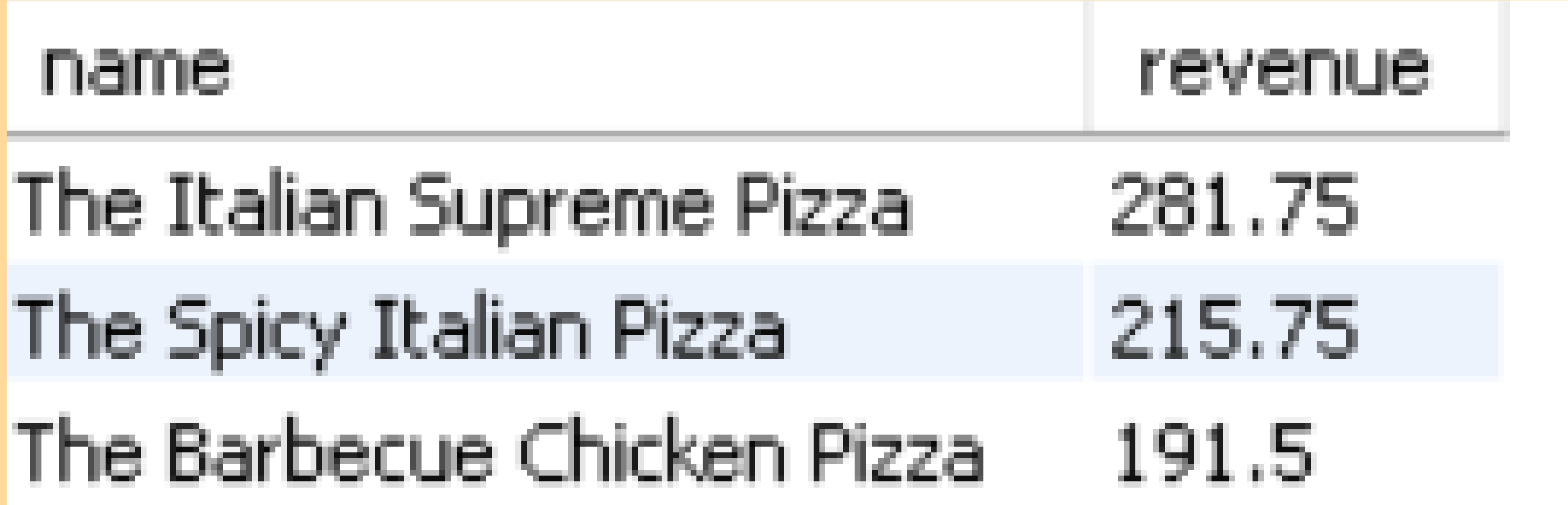
avg_pizza_per_day
72.0000

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



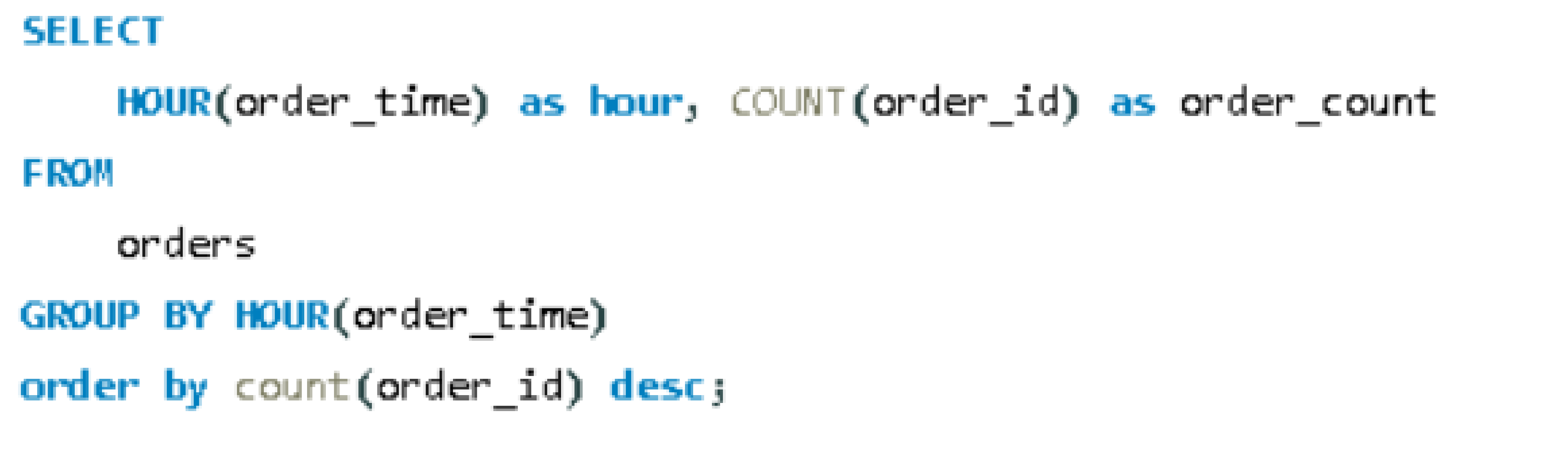
```
select pizza_types.name,sum(order_details.quantity*pizzas.price) as revenue
from pizza_types join pizzas on
pizza_types.pizza_type_id=pizzas.pizza_type_id

join order_details on
order_details.pizza_id =pizzas.pizza_id
group by pizza_types.name
order by revenue desc limit 3
```

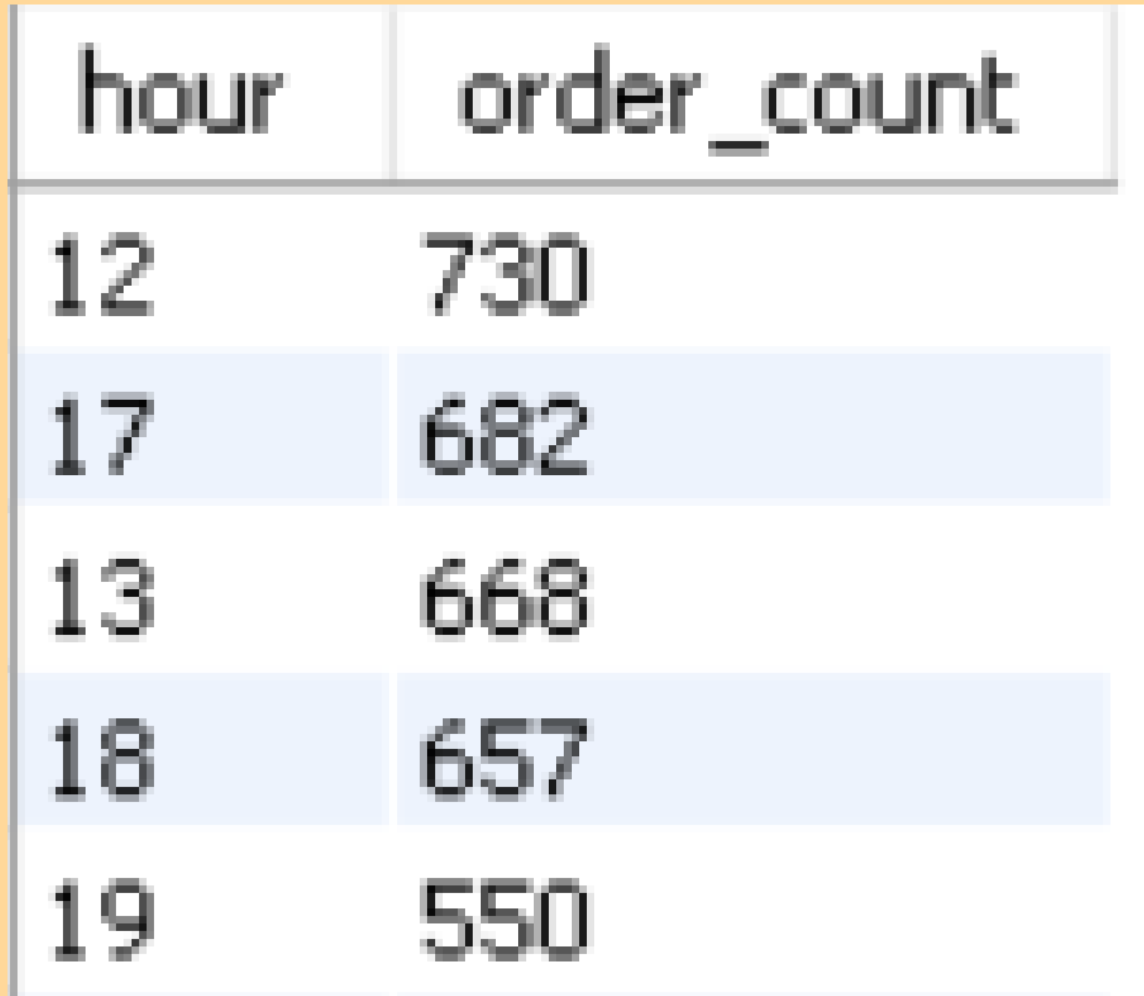


name	revenue
The Italian Supreme Pizza	281.75
The Spicy Italian Pizza	215.75
The Barbecue Chicken Pizza	191.5

Determine the distribution of orders by hour of the day

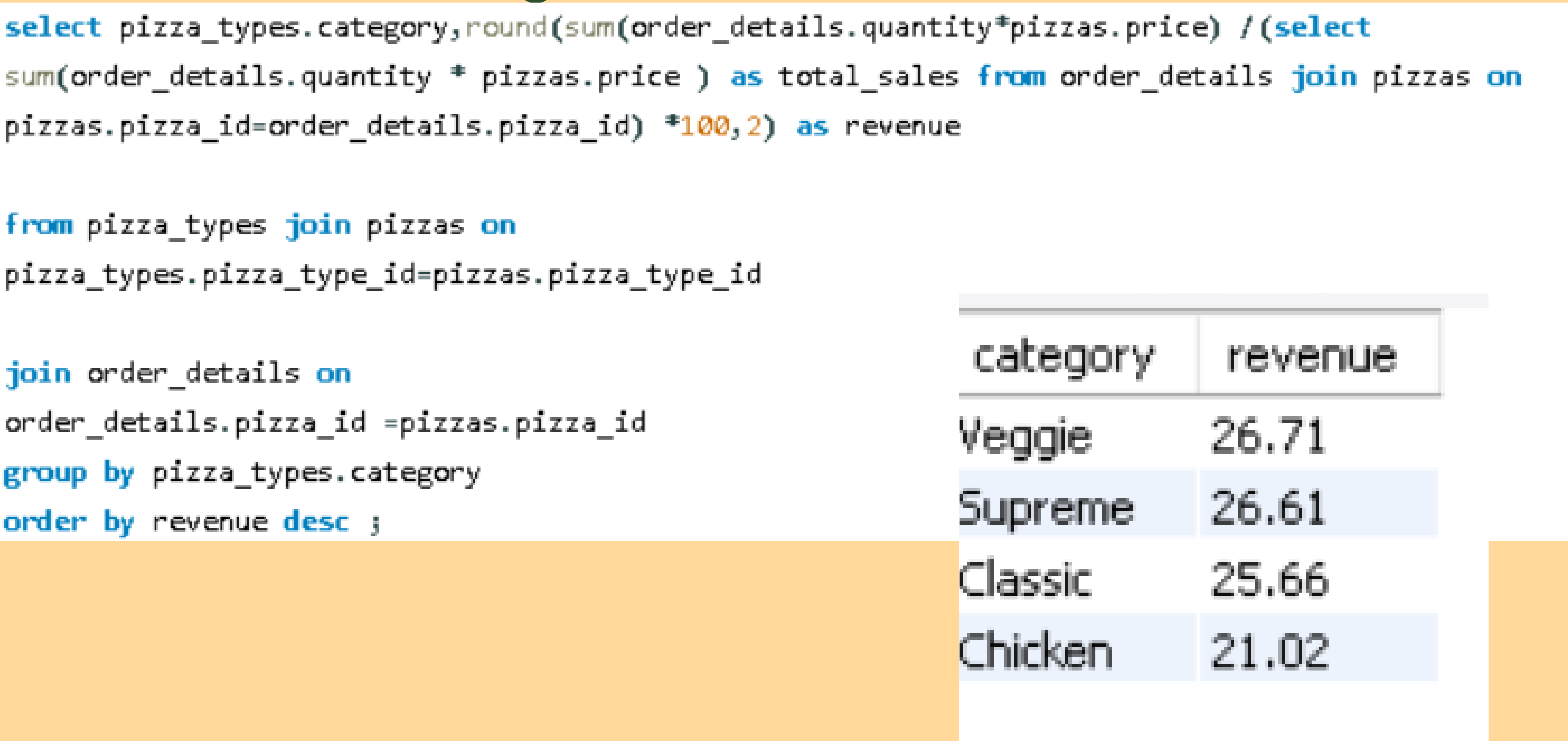


```
SELECT
    HOUR(order_time) as hour, COUNT(order_id) as order_count
FROM
    orders
GROUP BY HOUR(order_time)
order by count(order_id) desc;
```



hour	order_count
12	730
17	682
13	668
18	657
19	550

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



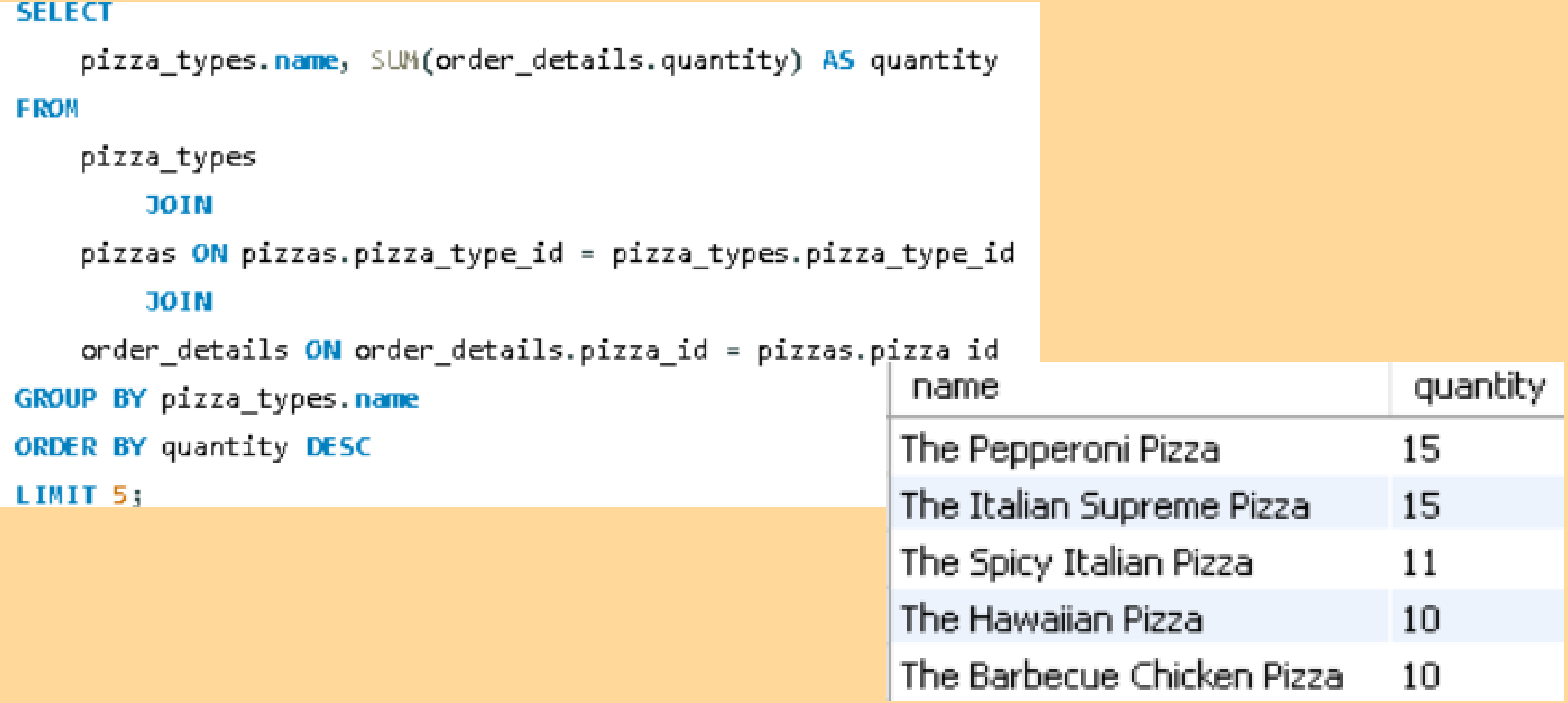
```
select pizza_types.category,round(sum(order_details.quantity*pizzas.price) /(select
sum(order_details.quantity * pizzas.price ) as total_sales from order_details join pizzas on
pizzas.pizza_id=order_details.pizza_id) *100,2) as revenue

from pizza_types join pizzas on
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 revenue desc ;
```

category	revenue
Veggie	26.71
Supreme	26.61
Classic	25.66
Chicken	21.02

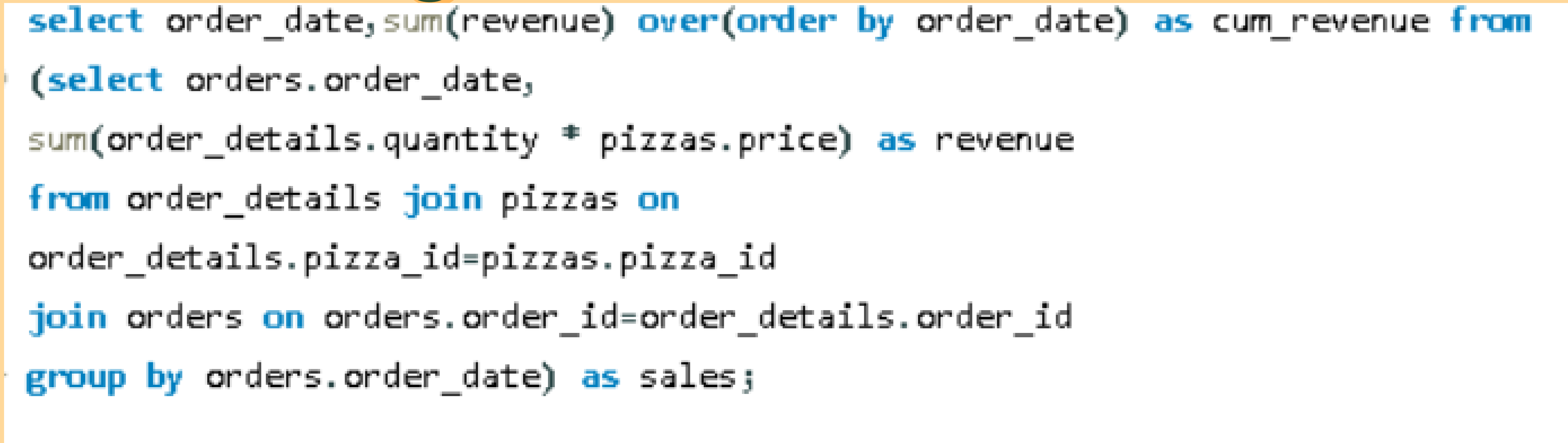
List the top 5 most ordered pizza types along with their quantities.



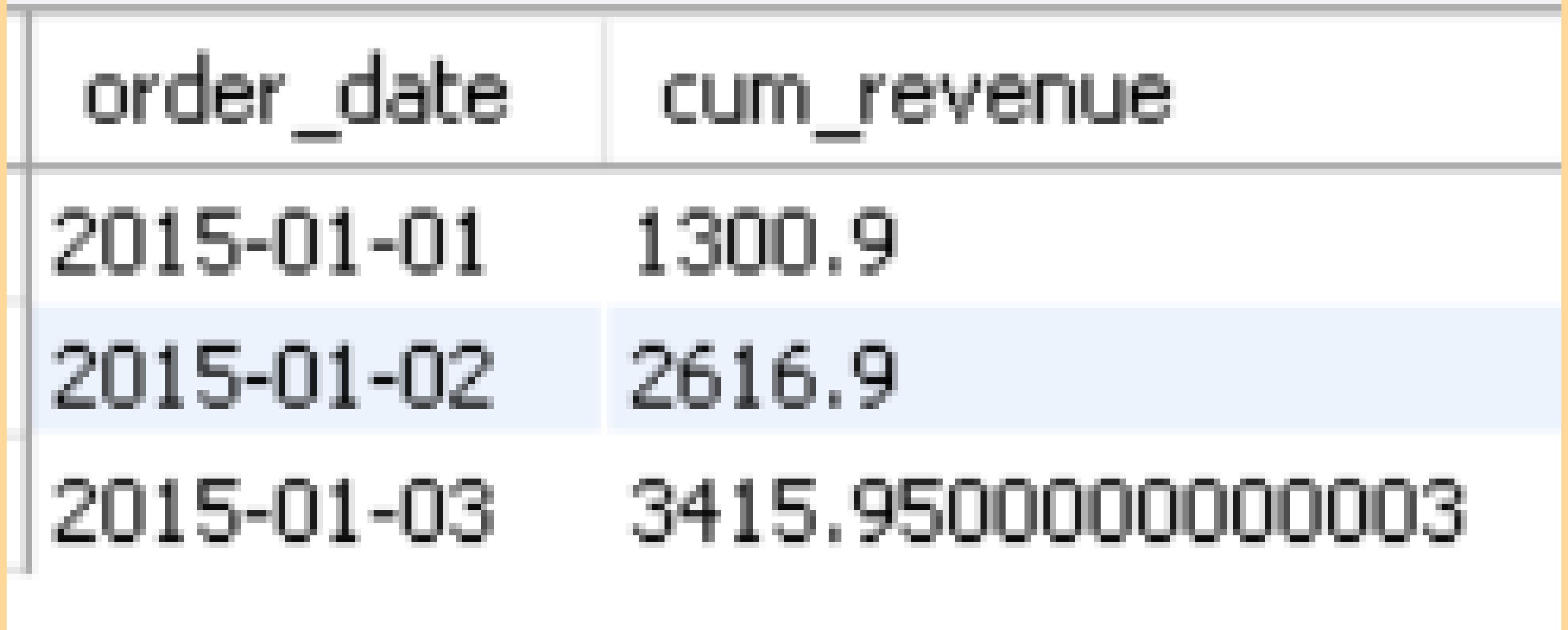
```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;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

name	quantity
The Pepperoni Pizza	15
The Italian Supreme Pizza	15
The Spicy Italian Pizza	11
The Hawaiian Pizza	10
The Barbecue Chicken Pizza	10

Analyze the cumulative revenue generated over time.

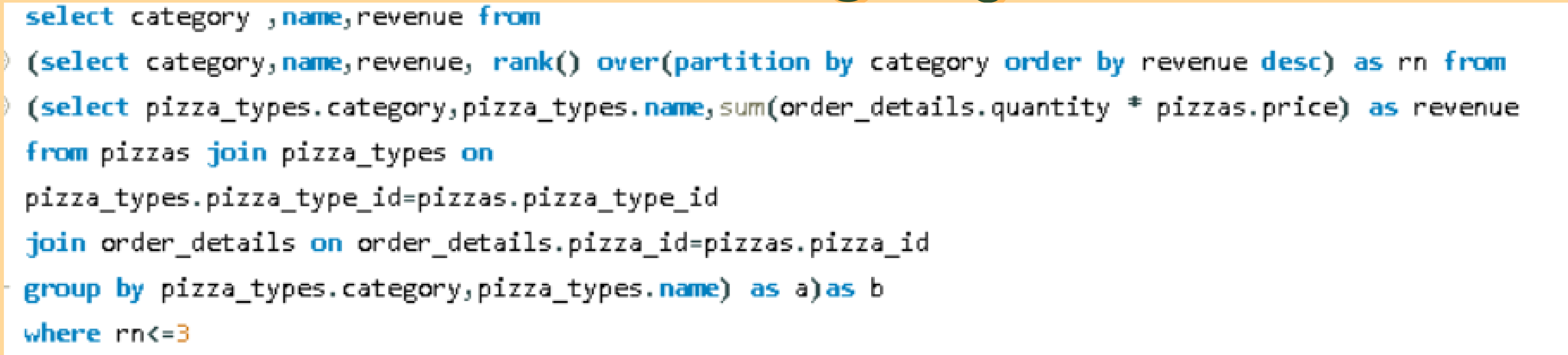


```
select order_date,sum(revenue) over(order by order_date) as cum_revenue from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from order_details join pizzas on
order_details.pizza_id=pizzas.pizza_id
join orders on orders.order_id=order_details.order_id
group by orders.order_date)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

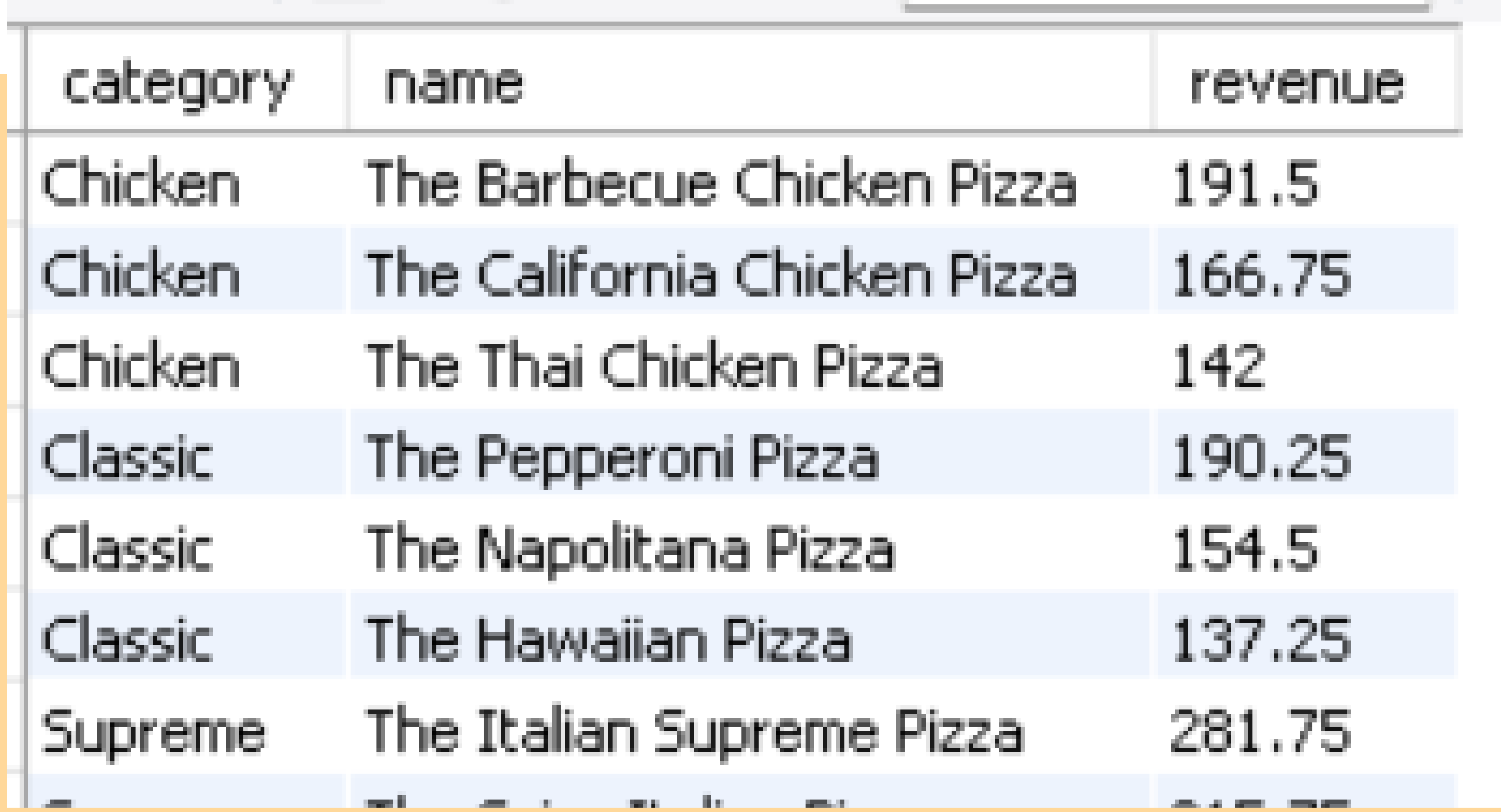


order_date	cum_revenue
2015-01-01	1300.9
2015-01-02	2616.9
2015-01-03	3415.9500000000003

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



```
select category ,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(order_details.quantity * pizzas.price) as revenue
from pizzas join pizza_types 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,pizza_types.name) as a)as b
where rn<=3
;
```



category	name	revenue
Chicken	The Barbecue Chicken Pizza	191.5
Chicken	The California Chicken Pizza	166.75
Chicken	The Thai Chicken Pizza	142
Classic	The Pepperoni Pizza	190.25
Classic	The Napolitana Pizza	154.5
Classic	The Hawaiian Pizza	137.25
Supreme	The Italian Supreme Pizza	281.75

MEGHA BORA PRESENTS

THANK YOU