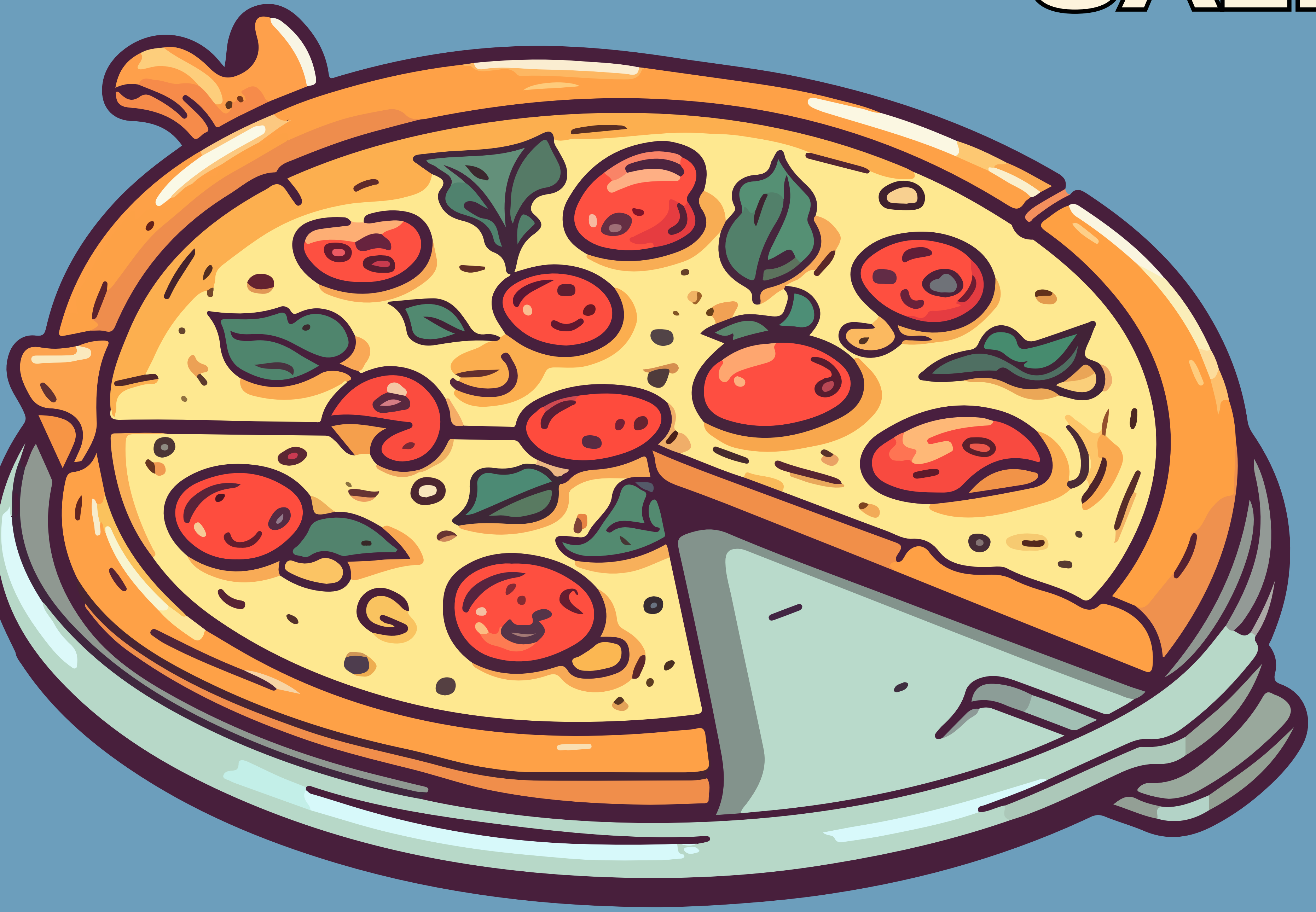
SQL DATA ANALYSIS PROJECT: PIZZA SALES INSIGHTS



Presented by: Arfa Pathan

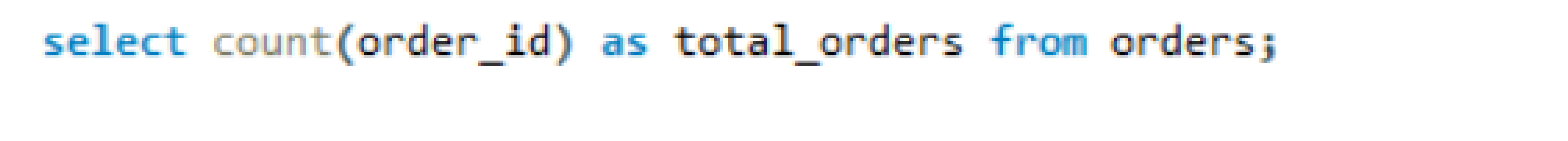
OVERVIEW OF THE PIZZA SALES DATASET

- **Orders Table: Contains data about individual orders placed, including order ID, date, and time.**
- **Pizzas Table: Provides information on the different pizzas offered, including pricing and categories.**
- **Order Details Table: Connects orders with specific pizzas ordered, along with quantities.**
- **Pizza Types: Details about pizza types, sizes, and other attributes.**

QUERY CATEGORIES IN THIS PROJECT

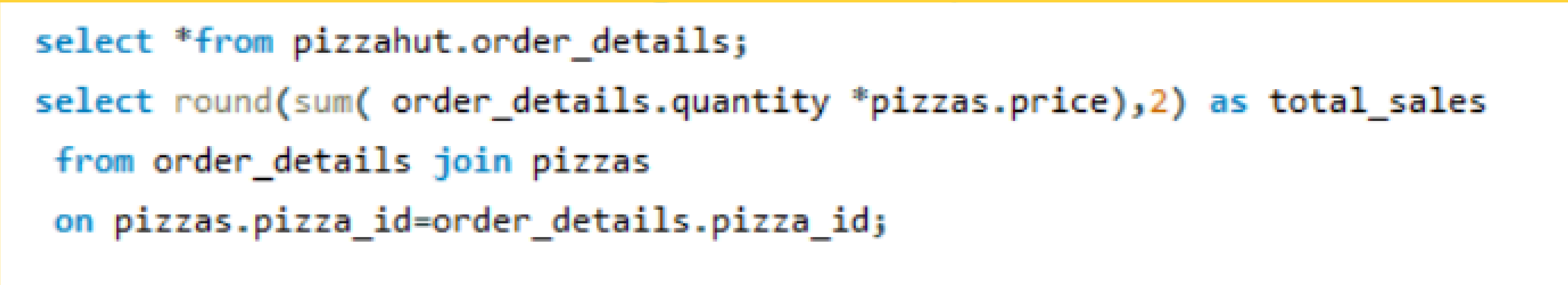
- Basic Queries: Retrieve total orders, calculate total revenue, and identify top pizzas.
- Intermediate Queries: Join tables to explore category-wise data, order distribution by time, and per-day averages.
- Advanced Queries: Perform in-depth analysis on revenue contributions, cumulative revenue over time, and category-specific top performers.

1 Retrieve the total number of orders placed.



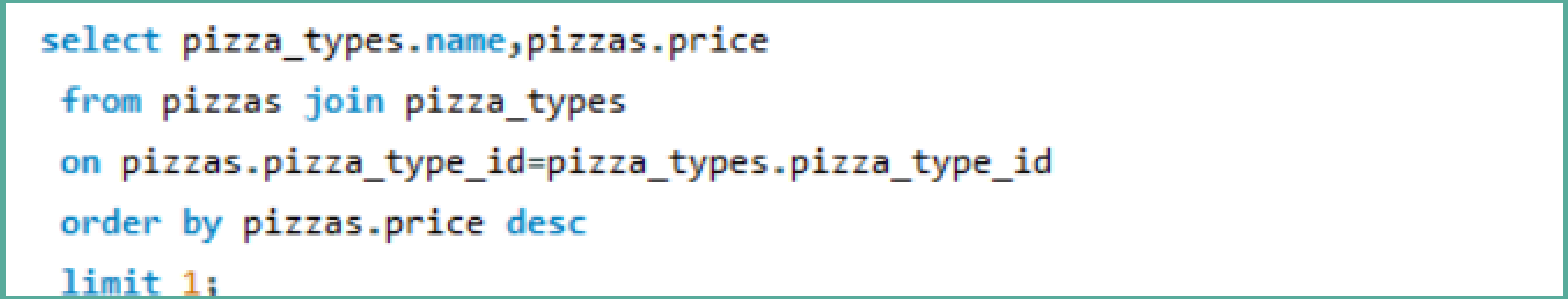
```sql
select count(order_id) as total_orders from orders;
```

2 Calculate the total revenue generated from pizza sales.



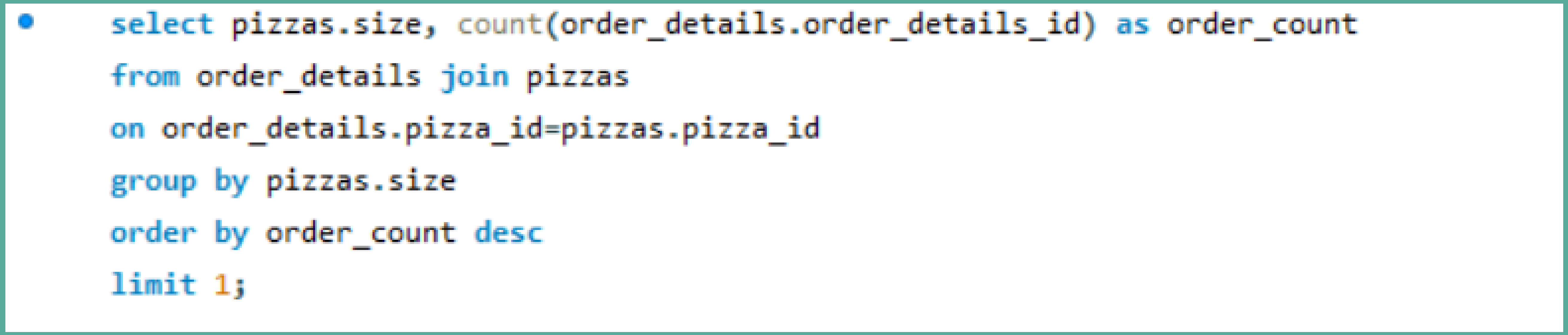
```sql
select *from pizzahut.order_details;
select round(sum( order_details.quantity *pizzas.price),2) as total_sales
 from order_details join pizzas
 on pizzas.pizza_id=order_details.pizza_id;
```

3 Identify the highest-priced pizza.



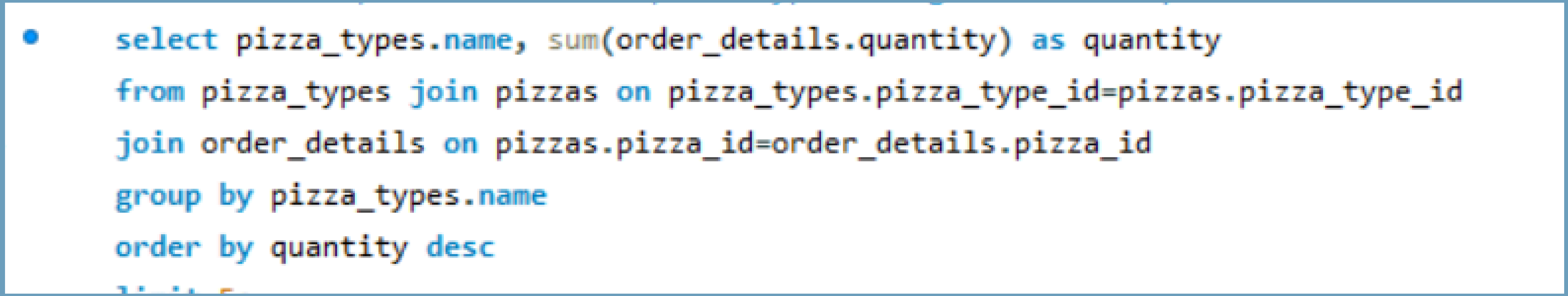
```
select pizza_types.name,pizzas.price
 from pizzas join pizza_types
 on pizzas.pizza_type_id=pizza_types.pizza_type_id
 order by pizzas.price desc
 limit 1;
```

4 Identify the most common pizza size ordered.



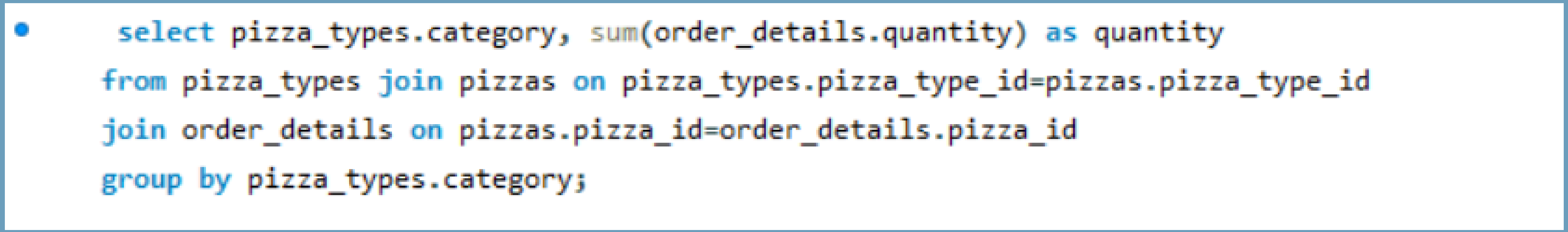
```
select pizzas.size, count(order_details.order_details_id) as order_count
from order_details join pizzas
on order_details.pizza_id=pizzas.pizza_id
group by pizzas.size
order by order_count desc
limit 1;
```

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



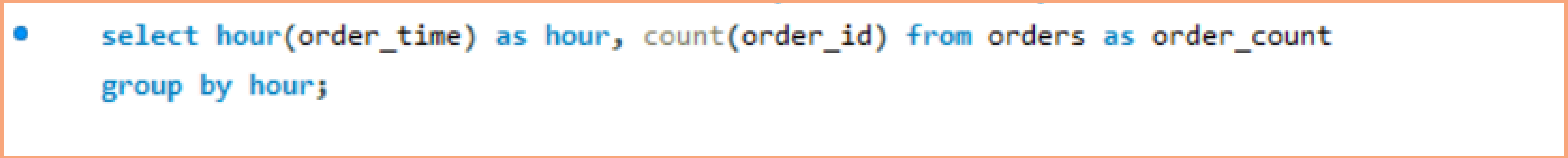
```
•     select pizza_types.name, sum(order_details.quantity) as quantity
      from pizza_types join pizzas on pizza_types.pizza_type_id=pizzas.pizza_type_id
      join order_details on pizzas.pizza_id=order_details.pizza_id
      group by pizza_types.name
      order by quantity desc
```

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



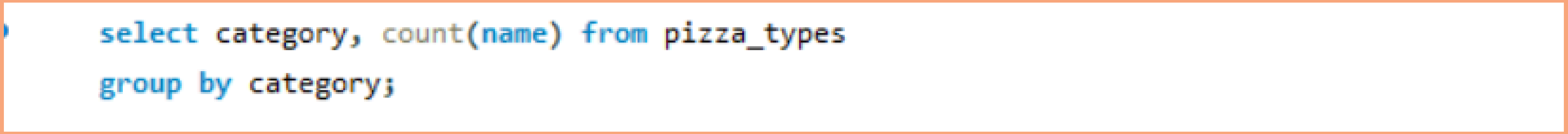
```
•      select pizza_types.category, sum(order_details.quantity) as quantity
      from pizza_types join pizzas on pizza_types.pizza_type_id=pizzas.pizza_type_id
      join order_details on pizzas.pizza_id=order_details.pizza_id
      group by pizza_types.category;
```

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



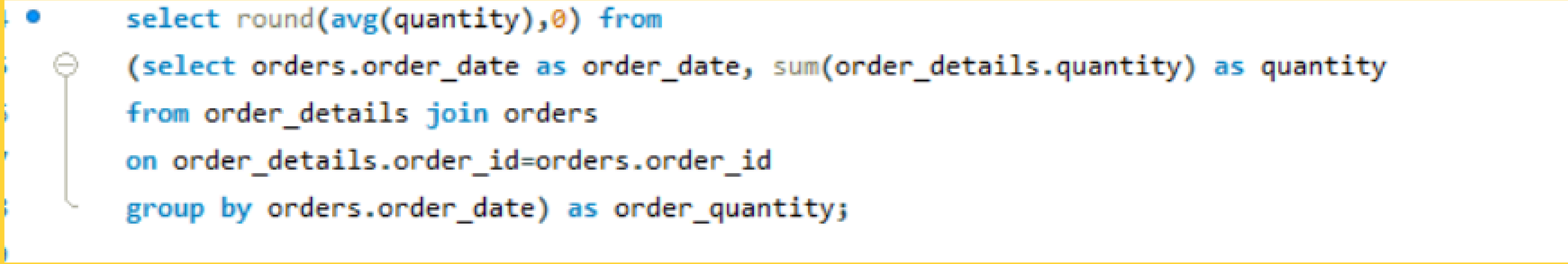
```
select hour(order_time) as hour, count(order_id) from orders as order_count
group by hour;
```

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



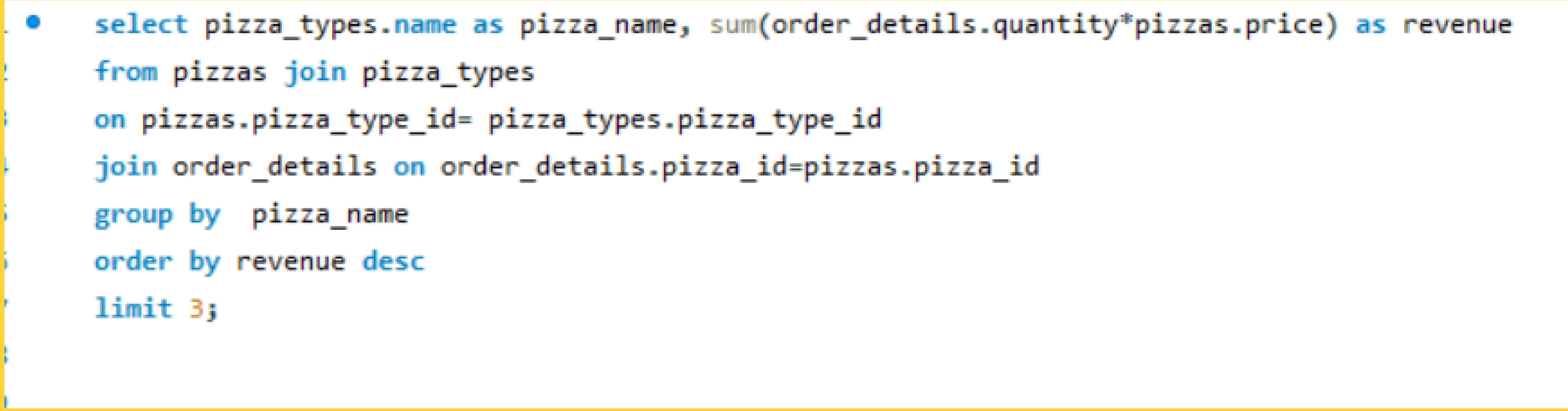
```
select category, count(name) from pizza_types
group by category;
```

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



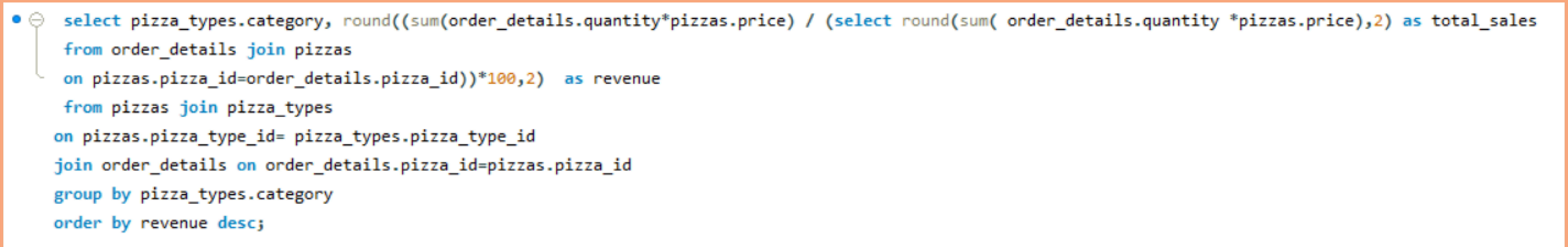
```
select round(avg(quantity),0) from
(select orders.order_date as order_date, sum(order_details.quantity) as quantity
from order_details join orders
on order_details.order_id=orders.order_id
group by orders.order_date) as order_quantity;
```

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



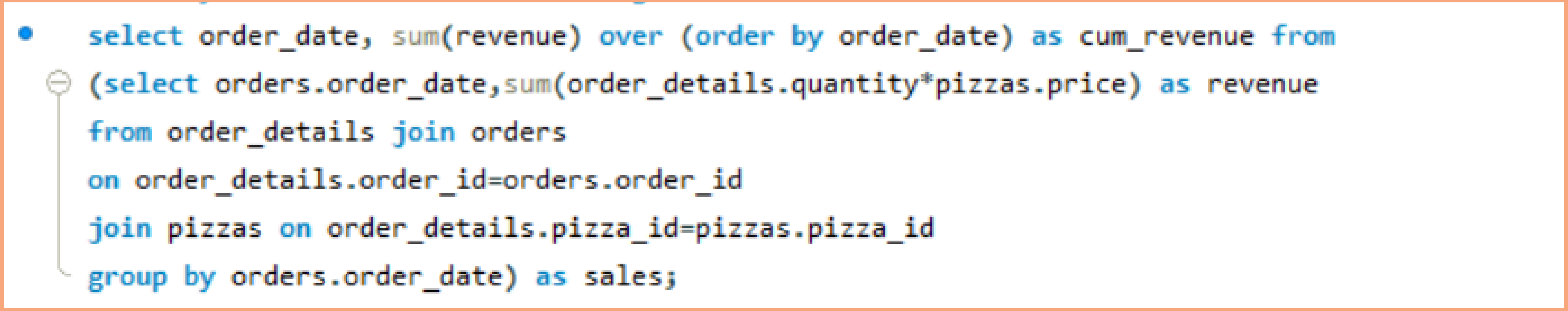
```
select pizza_types.name as pizza_name, sum(order_details.quantity*pizzas.price) as revenue
from pizzas join pizza_types
on pizzas.pizza_type_id= pizza_types.pizza_type_id
join order_details on order_details.pizza_id=pizzas.pizza_id
group by  pizza_name
order by revenue desc
limit 3;
```

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



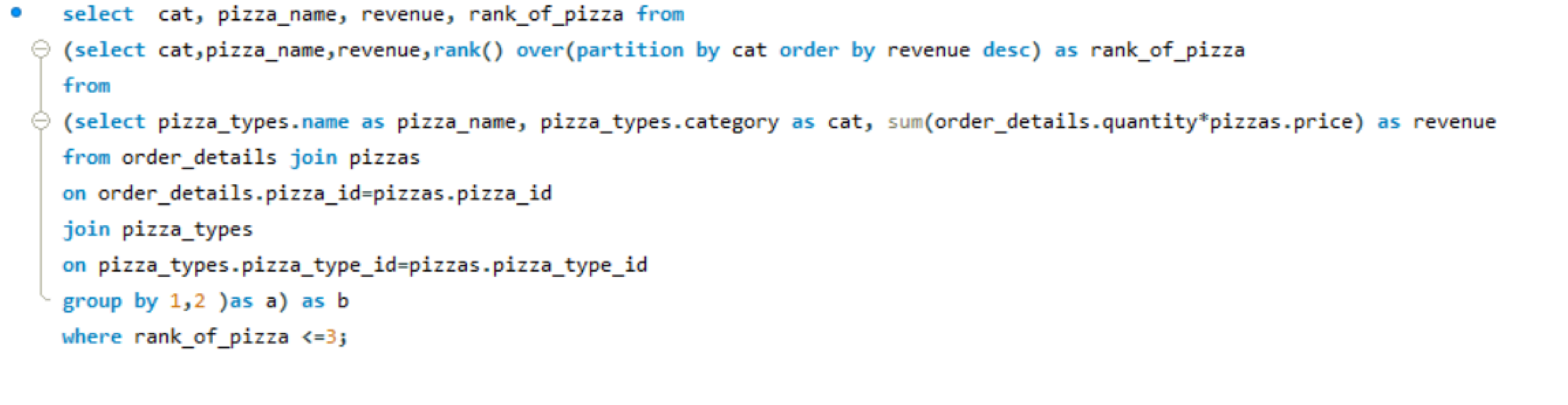
```
 select pizza_types.category, round((sum(order_details.quantity*pizzas.price) / (select round(sum( order_details.quantity *pizzas.price),2) as total_sales
 from order_details join pizzas
 on pizzas.pizza_id=order_details.pizza_id))*100,2)  as revenue
 from pizzas join pizza_types
on pizzas.pizza_type_id=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

12 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 orders
on order_details.order_id=orders.order_id
join pizzas on order_details.pizza_id=pizzas.pizza_id
group by orders.order_date) as sales;
```

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



```
select  cat, pizza_name, revenue, rank_of_pizza from
(select cat,pizza_name,revenue,rank() over(partition by cat order by revenue desc) as rank_of_pizza
from
(select pizza_types.name as pizza_name, pizza_types.category as cat, sum(order_details.quantity*pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id=pizzas.pizza_id
join pizza_types
on pizza_types.pizza_type_id=pizzas.pizza_type_id
group by 1,2 )as a) as b
where rank_of_pizza <=3;
```