Pizza Hut Outlest Sales Analysis

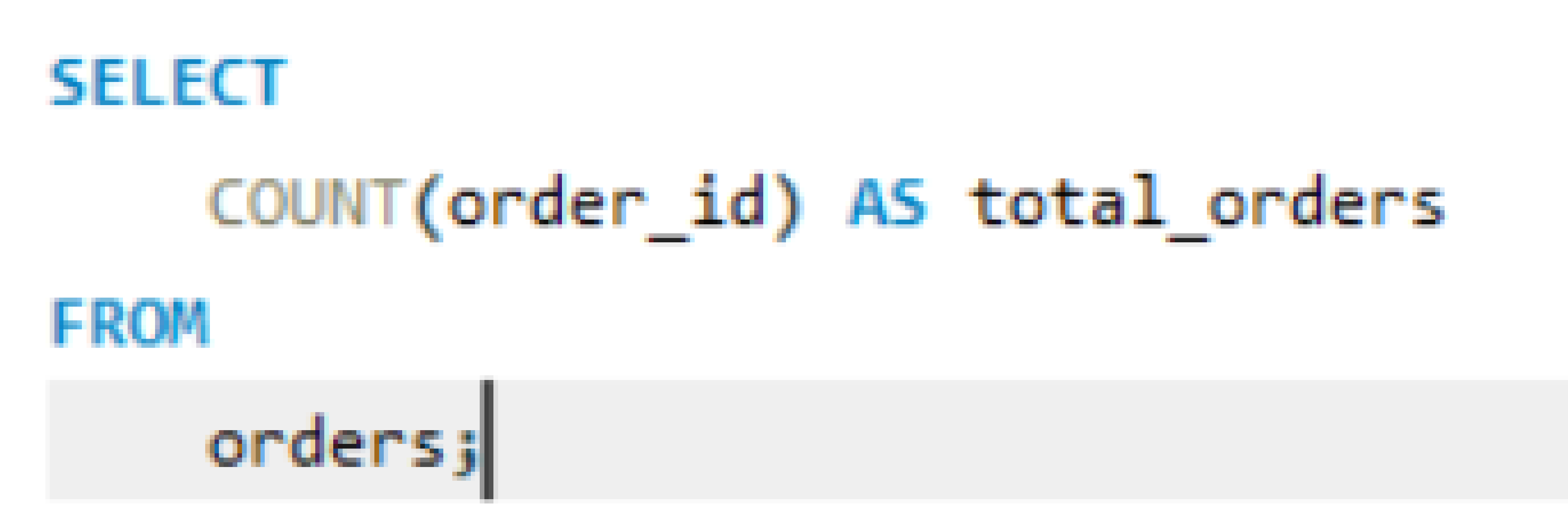
by using MY SQL

Amresh Kumar

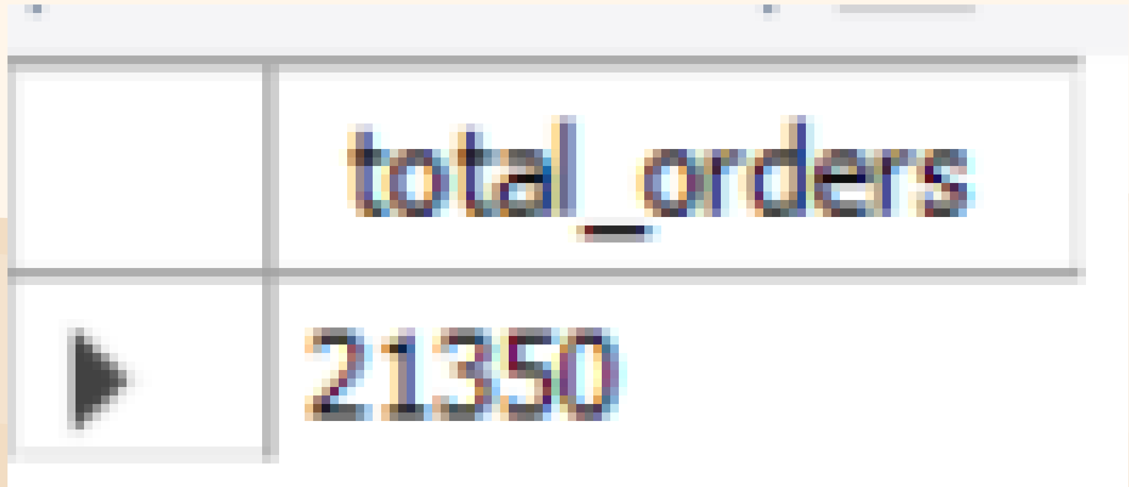
Brief Description about Data

- **Order Details: Lists individual pizza orders with their unique IDs, the specific pizza ordered, and the quantity.**
- **Order Log: Records the date and time for each order, providing a timeline of sales.**
- **Pizza Catalog: Describes different types of pizzas offered, categorized by ingredients like chicken, and details their specific toppings.**
- **Pricing Information: Shows the size variants (small, medium, large) for each pizza type and their respective prices.**

Retrieve the total number of orders placed.

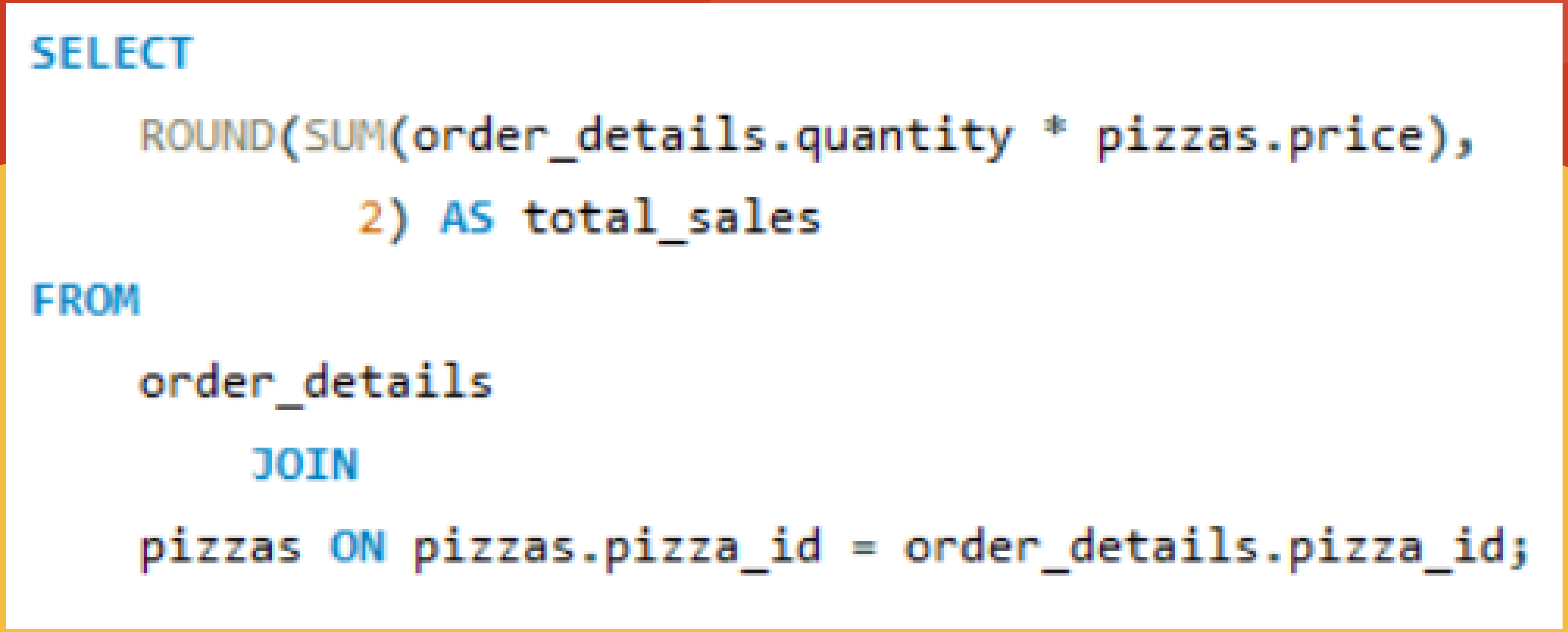


```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```



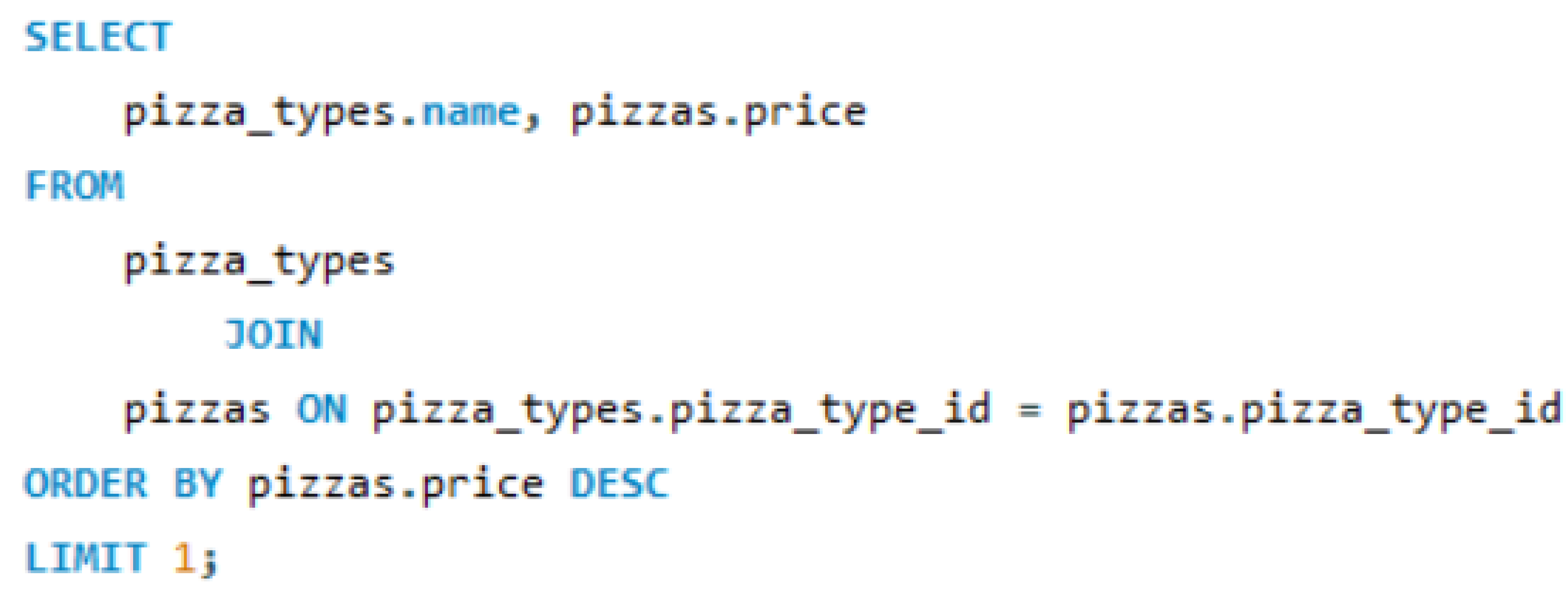
	total_orders
▶	21350

Calculate the total revenue generated from pizza sales.



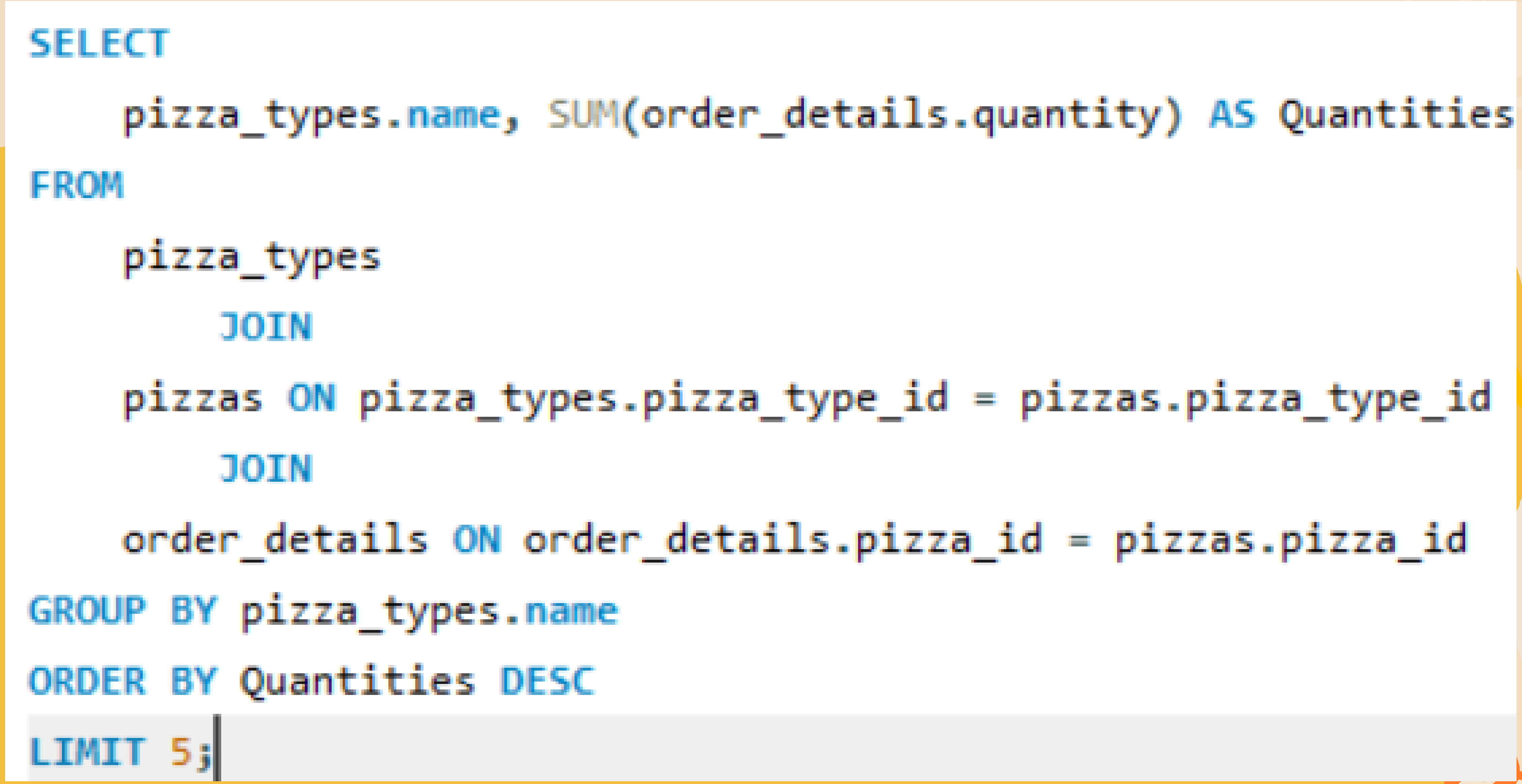
```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Identify the highest-priced pizza.



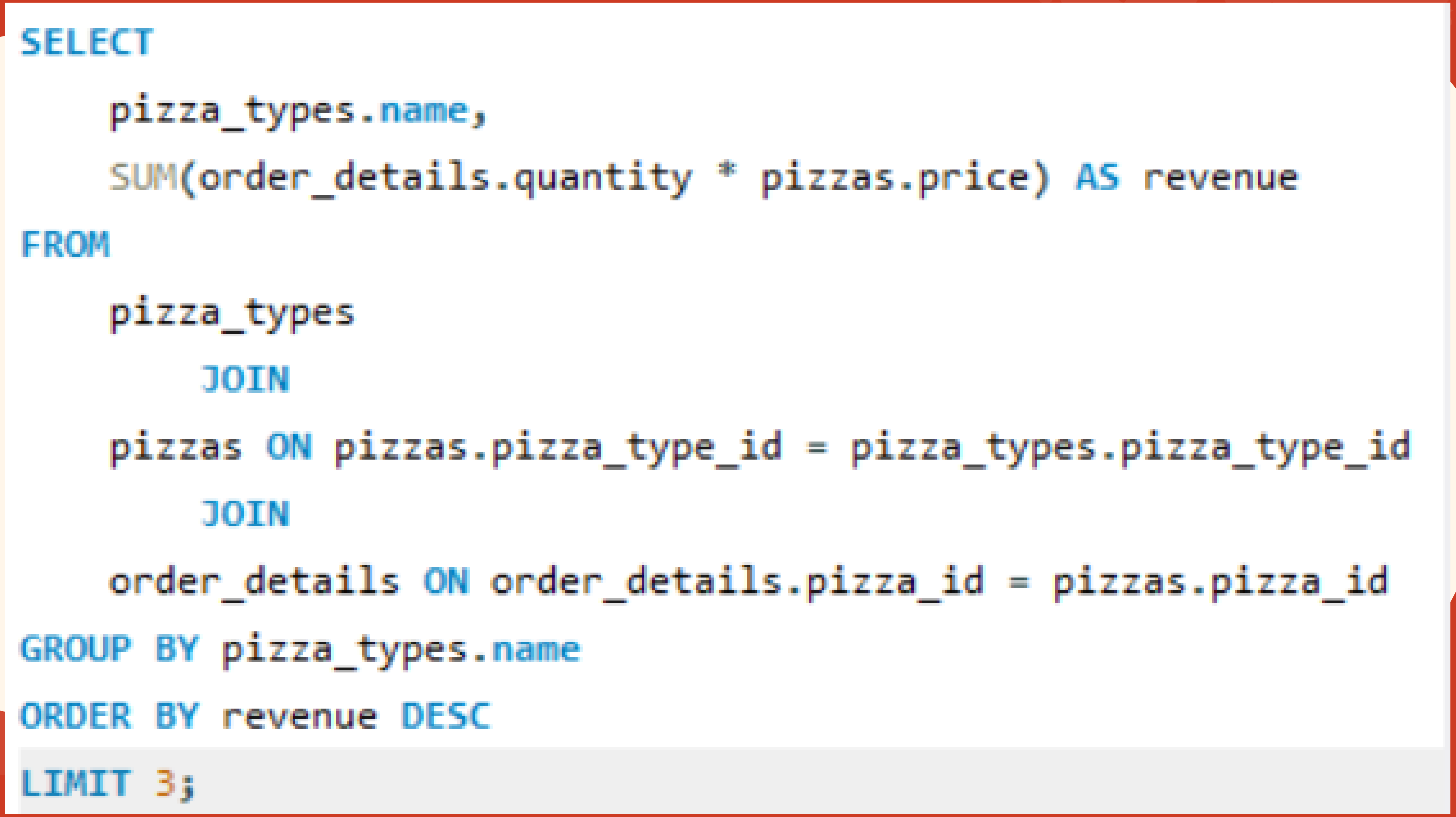
```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

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



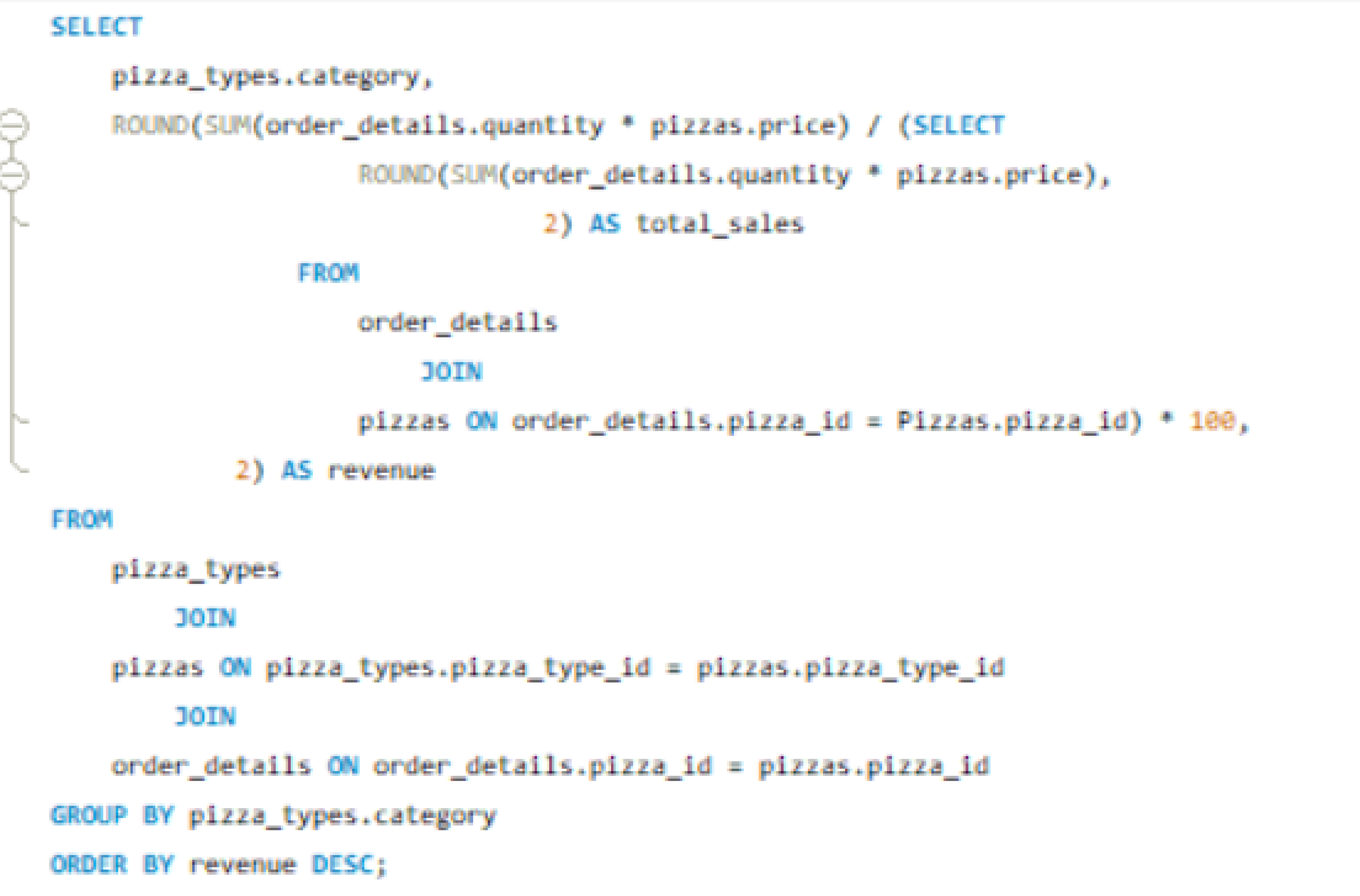
```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS Quantities
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY Quantities DESC
LIMIT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

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



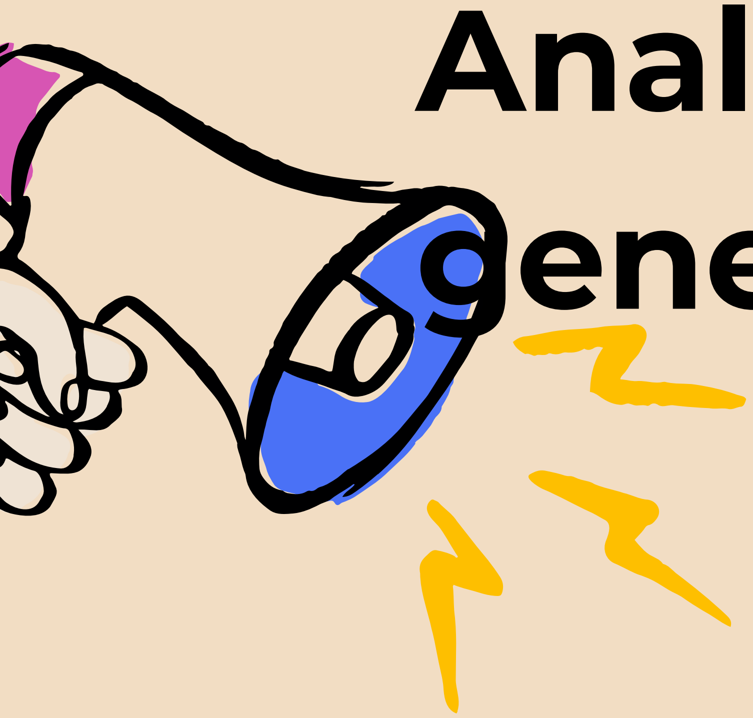
```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue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

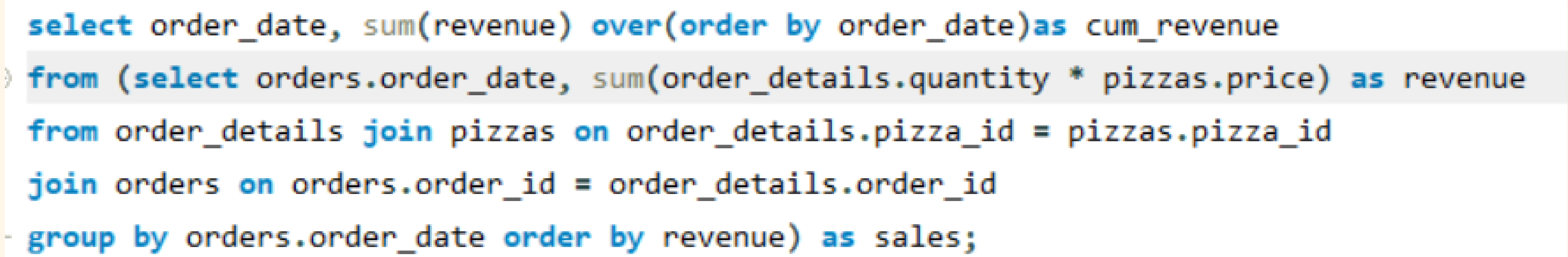
Calculate the percentage contribution of each pizza type to total revenue.



```
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 order_details.pizza_id = Pizzas.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;
```

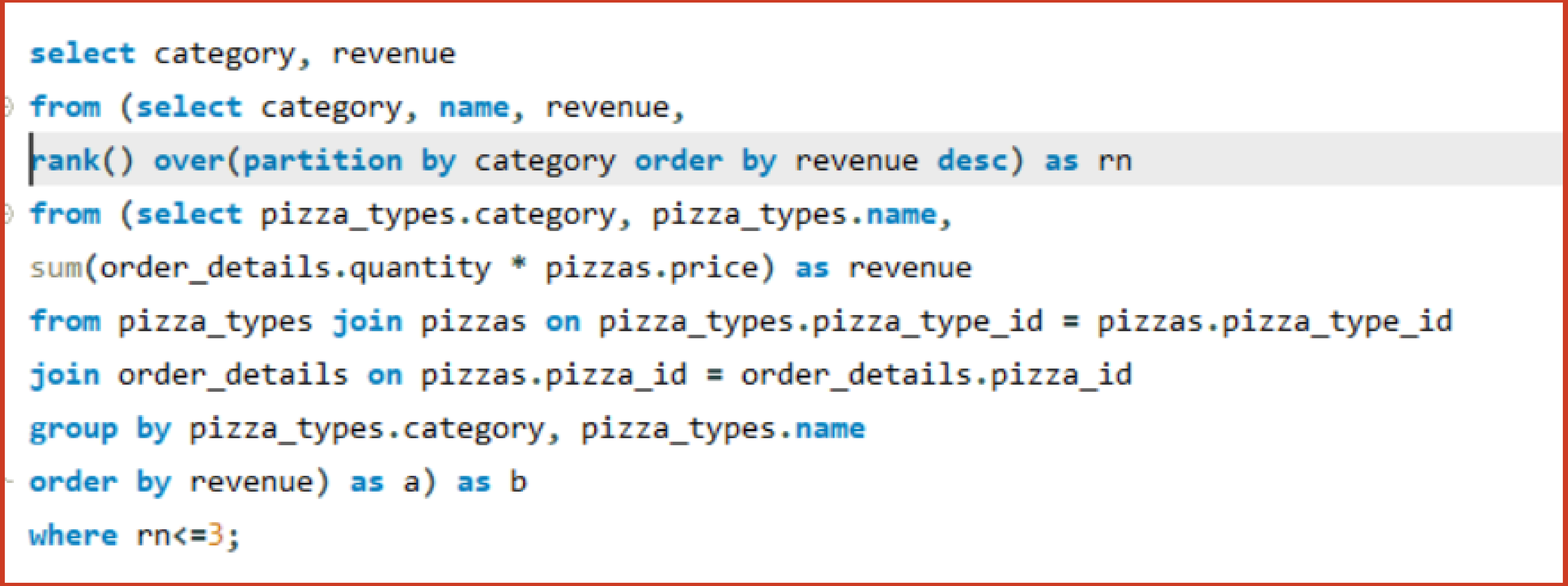
Analyze the cumulative revenue generated over time.





```
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 order by revenue) as sales;
```

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



```
select category, 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 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
order by revenue) as a) as b
where rn<=3;
```

Thank You!

✉ **Email**
usb.18mba1186@gmail.com

linkedin
www.linkedin.com/in/amresh-kumar1995