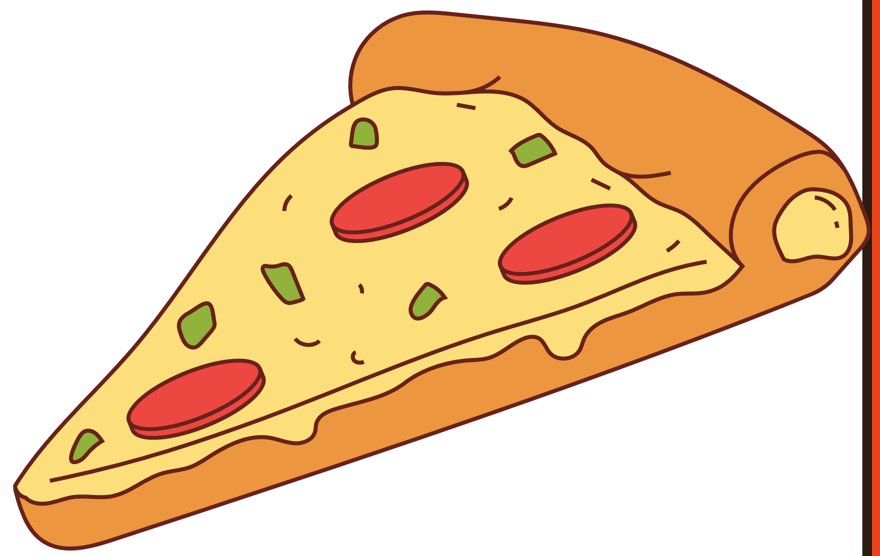
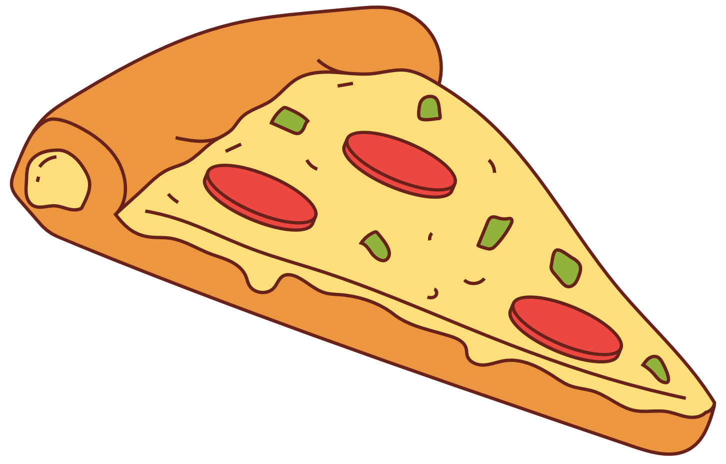
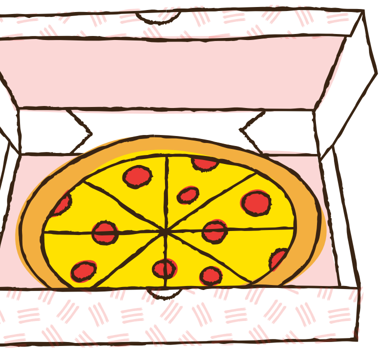
SQL PROJECT ON PIZZA SALES

HELLO!

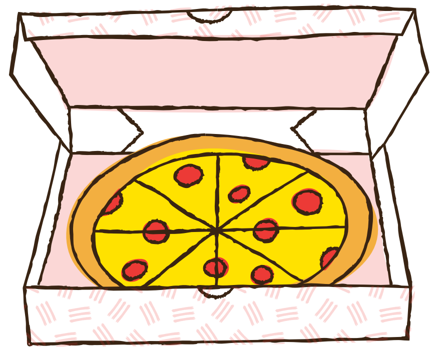


Hello! My name is Akshay Poojari, and I am excited to present my project on pizza sales analysis using SQL queries. The aim of this project is to provide insights into various aspects of pizza sales by leveraging data analysis techniques.



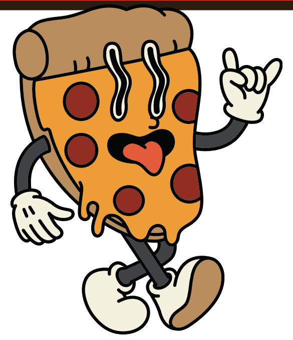


OBJECTIVES

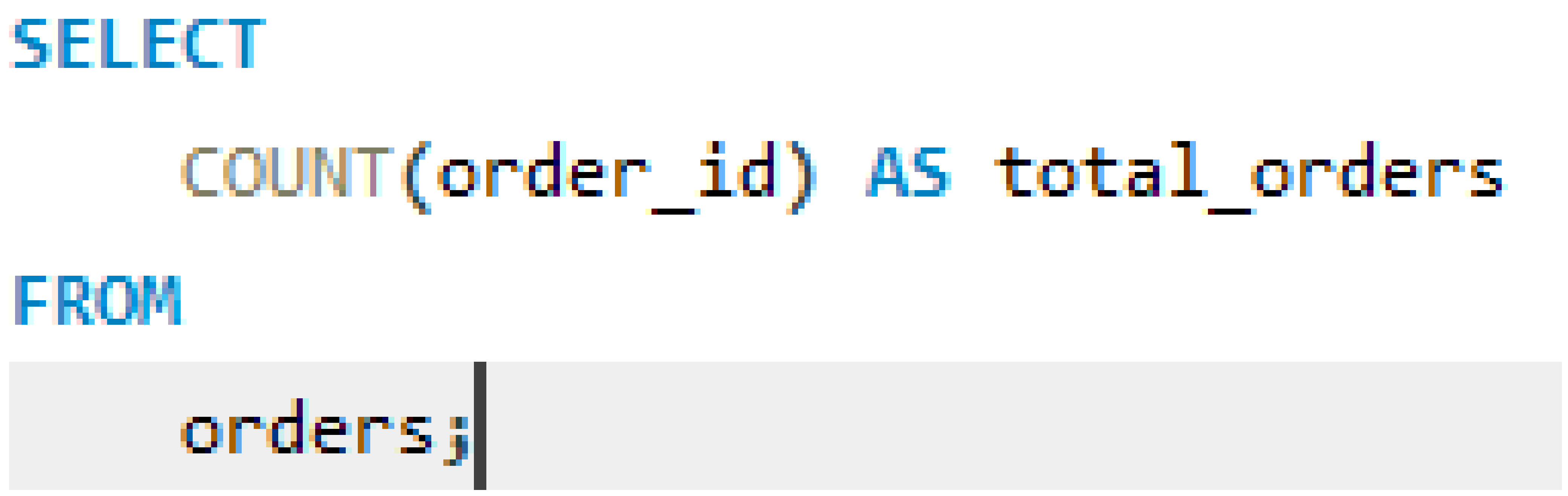


The primary objectives of this project are:

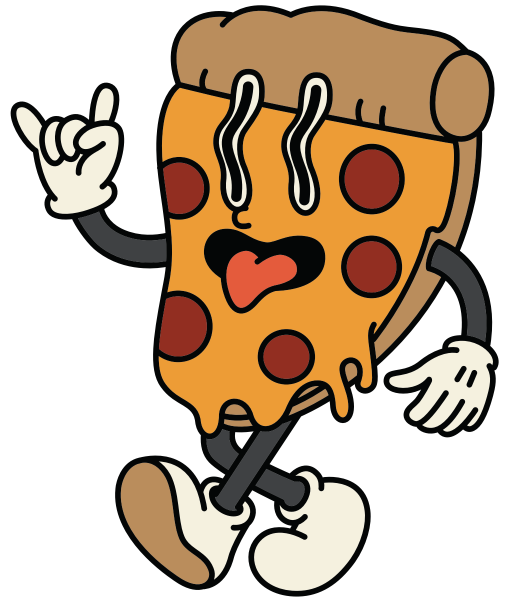
1. Retrieve the total number of orders placed: Understand the volume of orders to gauge customer activity.
2. Calculate the total revenue generated from pizza sales: Assess the financial performance.
3. Identify the highest-priced pizza: Highlight the premium products.
4. Identify the most common pizza size ordered: Determine customer preferences.
5. List the top 5 most ordered pizza types along with their quantities: Identify the most popular products.
6. Find the total quantity of each pizza category ordered: Understand the distribution of orders across different categories.
7. Determine the distribution of orders by hour of the day: Analyze peak ordering times.
8. Calculate the category-wise distribution of pizzas: Categorize sales for targeted marketing.
9. Calculate the average number of pizzas ordered per day: Determine daily sales trends.
10. Determine the top 3 most ordered pizza types based on revenue: Identify top revenue-generating products.
11. Calculate the percentage contribution of each pizza type to total revenue: Understand each product's financial impact.
12. Analyze the cumulative revenue generated over time: Track revenue growth trends.
13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

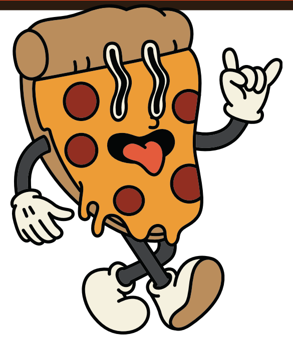


retrieve the total number of orders placed.



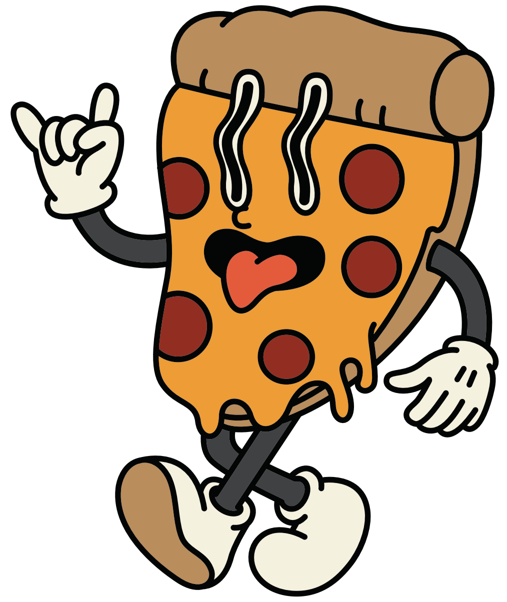
```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

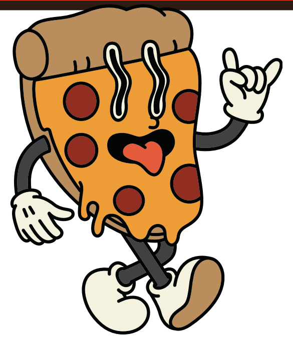




Calculate the total revenue generated from pizza sales.

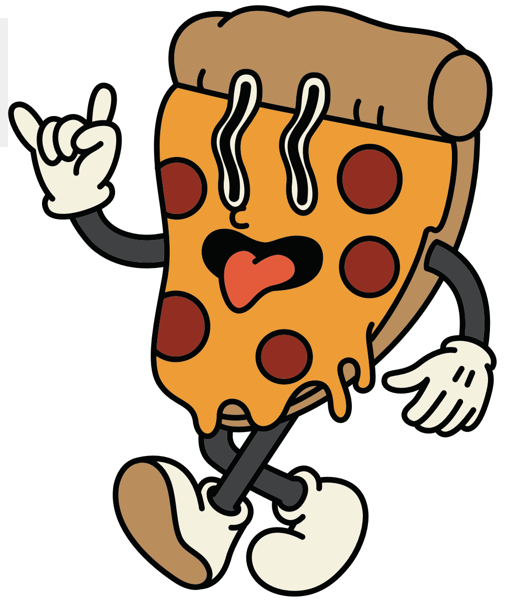
```
SELECT
    SUM(order_details.quantity * pizzas.price) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

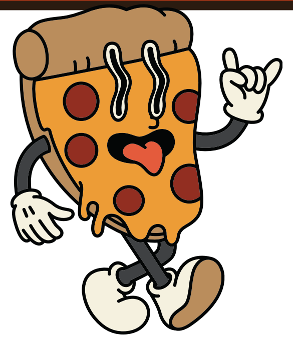




identify the highest-priced pizza

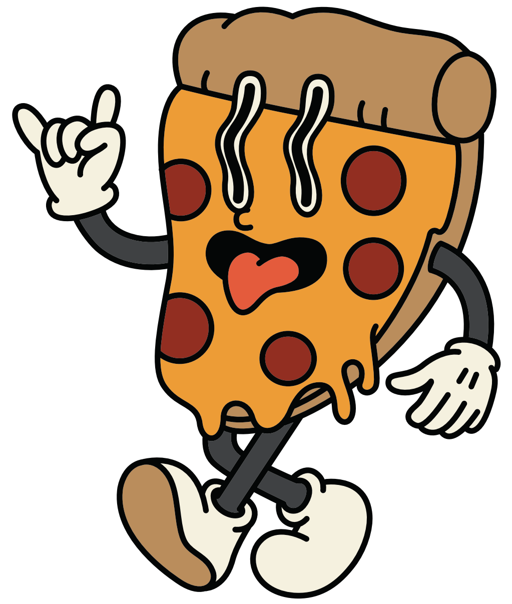
```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

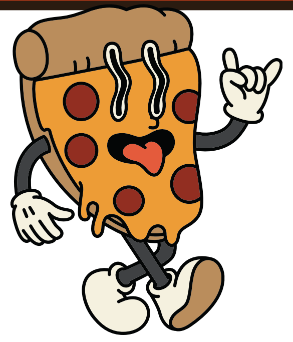




Identify the most common pizza size ordered.

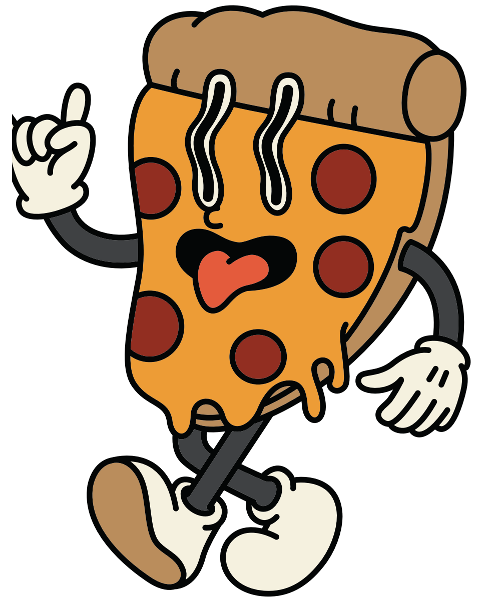
```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
```

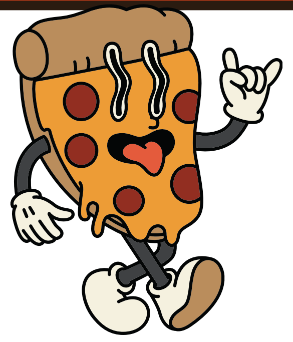




List the top 5 most ordered pizza types along with their quantities.

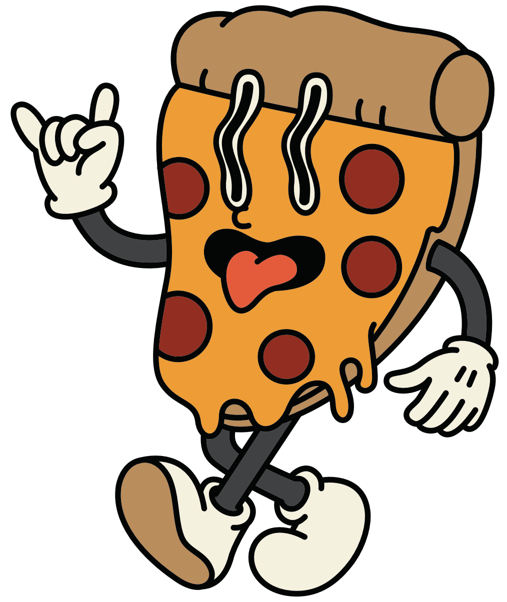
```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;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

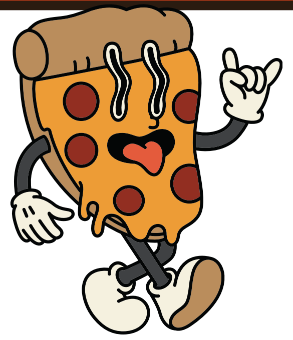




Join the necessary tables to find the total quantity of each pizza category ordered.

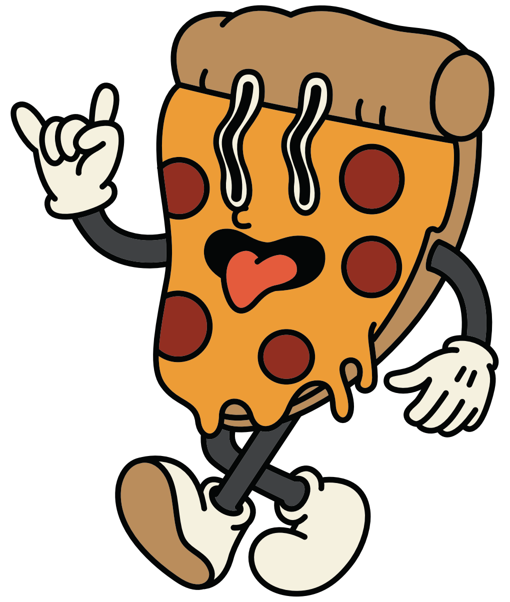
```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY total_quantity DESC;
```

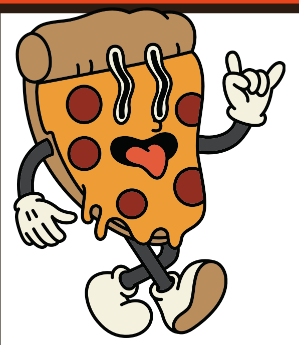




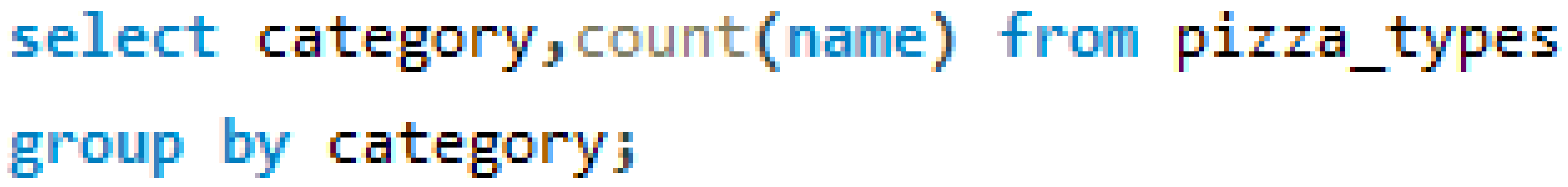
Determine the distribution of orders by hour of the day.

```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

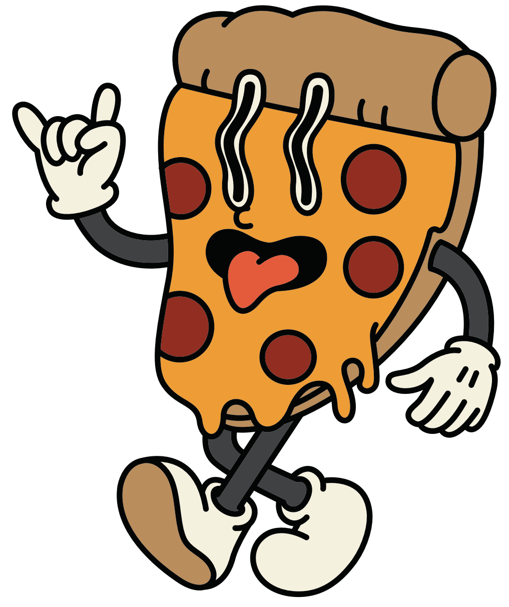


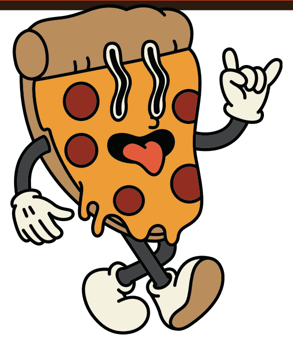


Join relevant tables to find the category-wise distribution of pizzas.



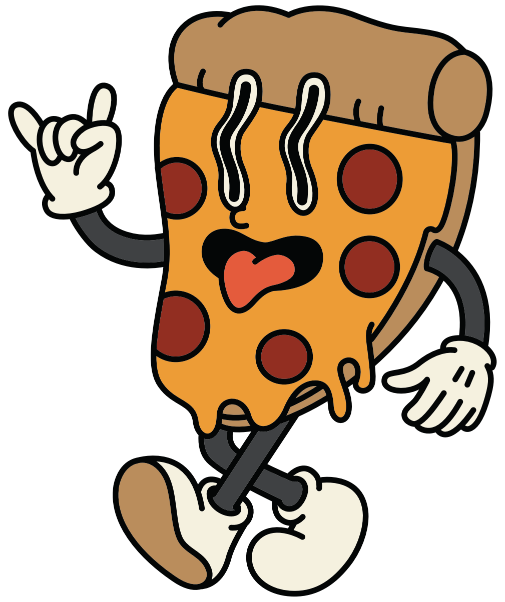
```
select category,count(name) from pizza_types
group by category;
```

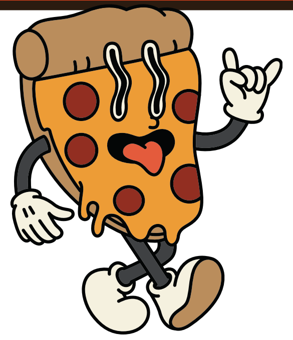




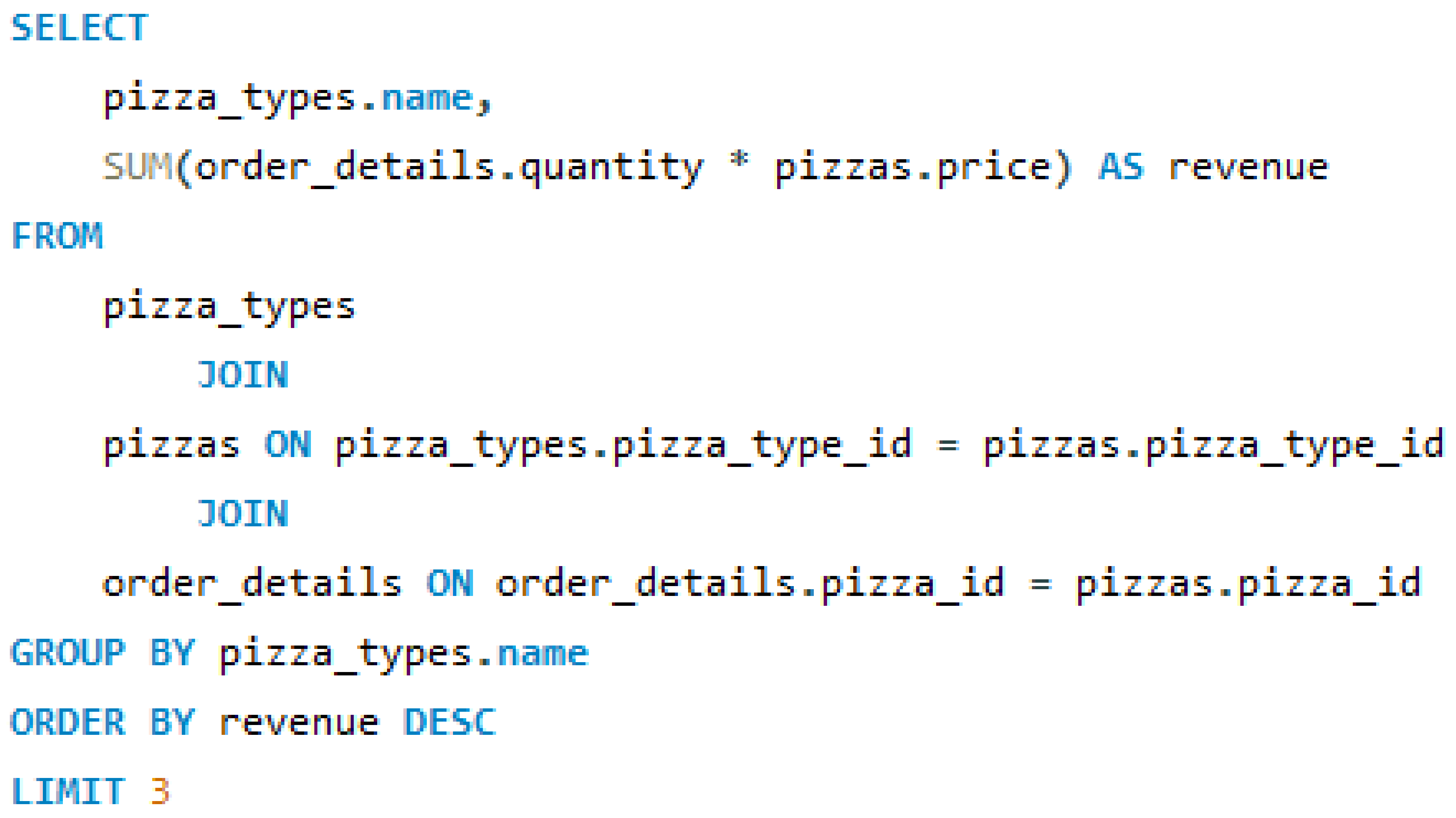
Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT
    ROUND(AVG(quantity), 0) as avreage_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

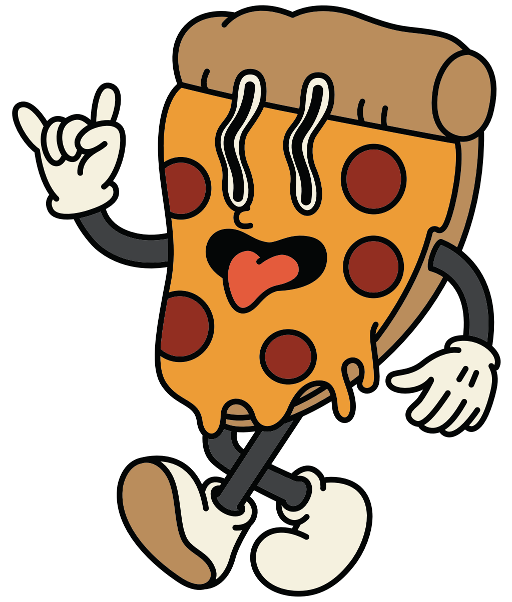


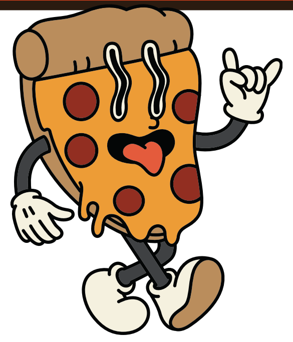


Determine the top 3 most ordered pizza types based on revenue.

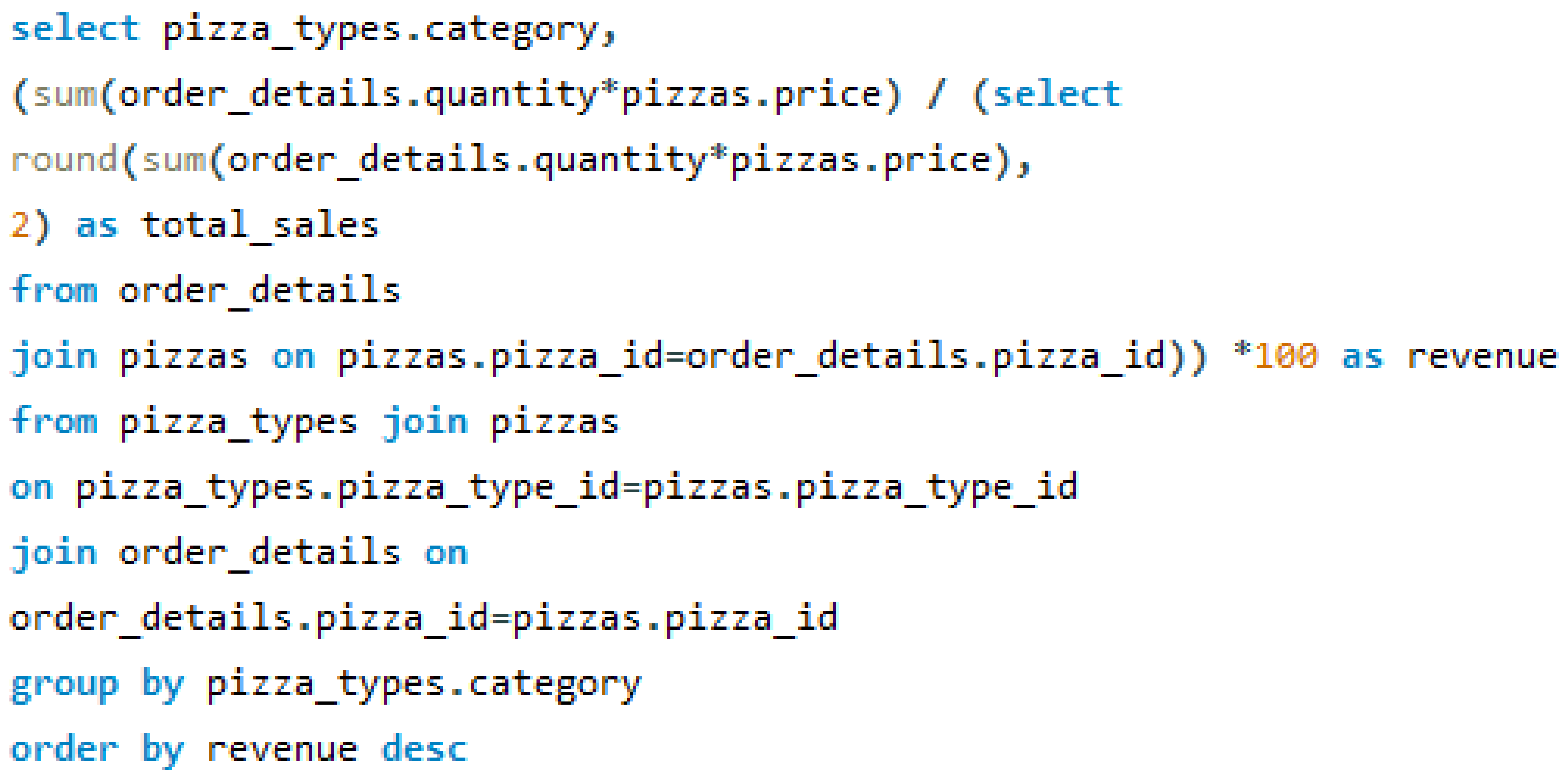


```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue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

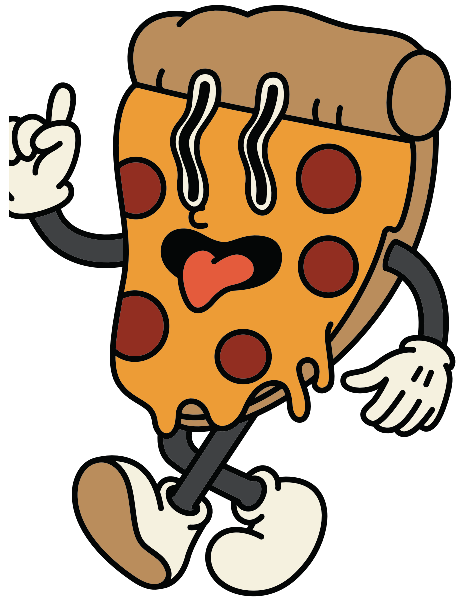


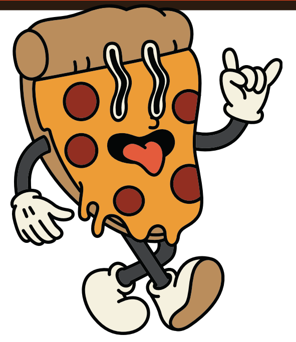


Calculate the percentage contribution of each pizza type to total revenue.

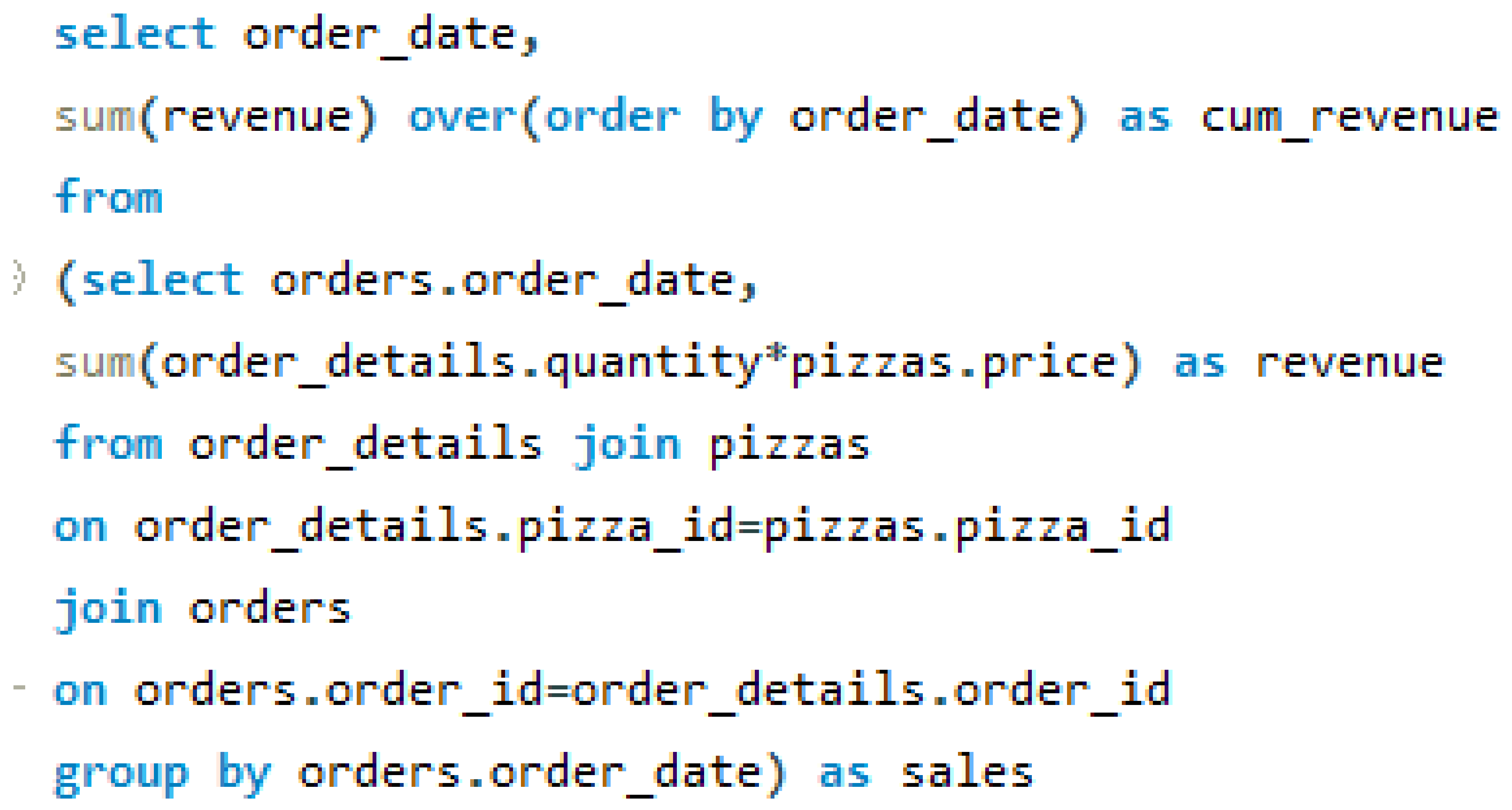


```
select pizza_types.category,
(sum(order_details.quantity*pizzas.price) / (select
round(sum(order_details.quantity*pizzas.price),
2) as total_sales
from order_details
join pizzas on pizzas.pizza_id=order_details.pizza_id)) *100 as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id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

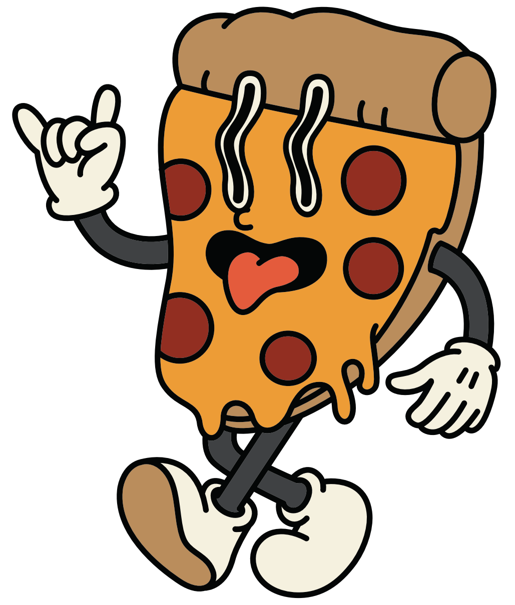


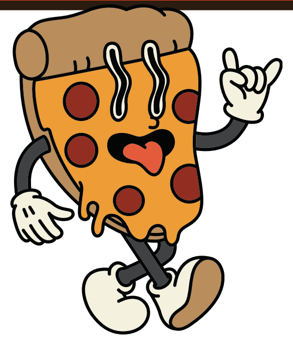


Analyze the cumulative revenue generated over time.

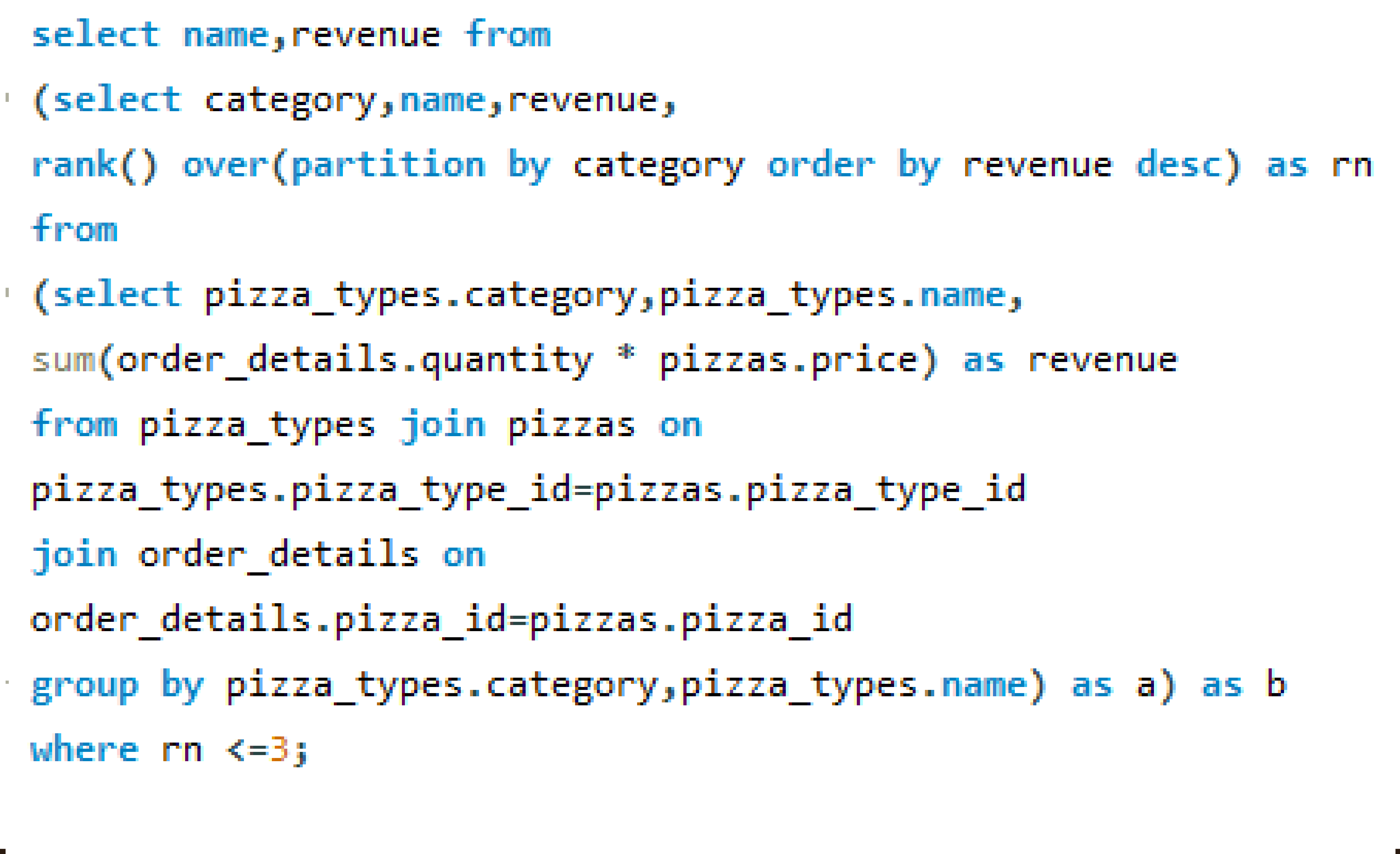


```
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,
sum(order_details.quantity*pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id=pizzas.pizza_id
join orders
on orders.order_id=order_details.order_id
group by orders.order_date)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

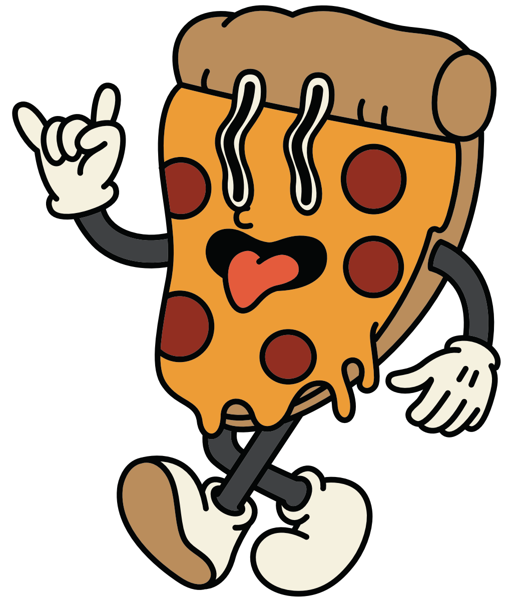




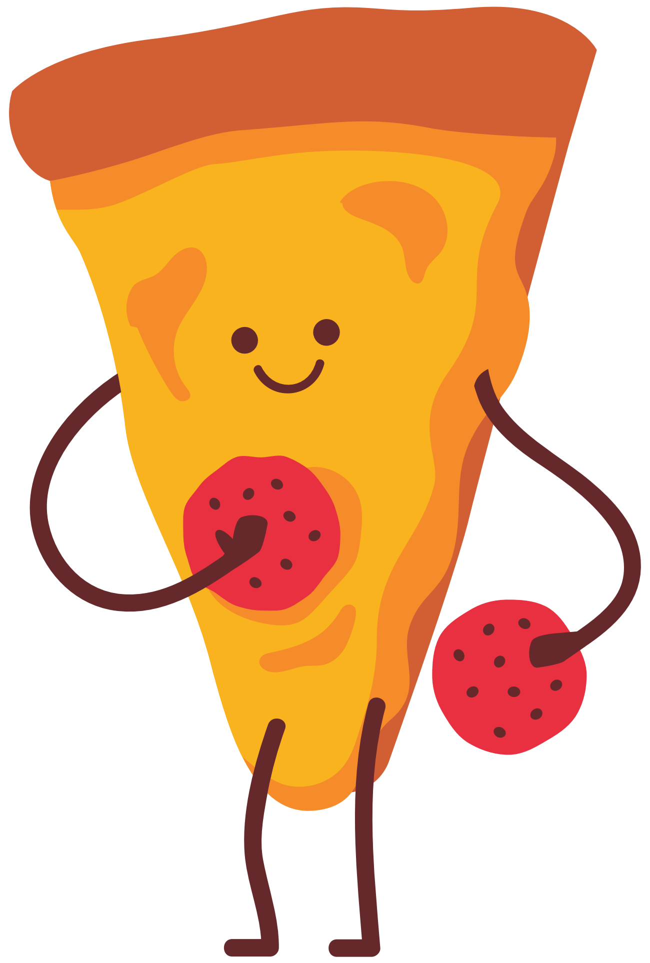
Determine the top 3 most ordered pizza types based on revenue for each pizza category.



```
select name,revenue from
(select category,name,revenue,
rank() over(partition by category order by revenue desc) as 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
order_details.pizza_id=pizzas.pizza_id
group by pizza_types.category,pizza_types.name) as a) as b
where rn <=3;
```



CONCLUSION



This project aims to demonstrate the power of SQL in analyzing real-world business data. The insights gained from this analysis can help in optimizing sales strategies, enhancing customer satisfaction, and ultimately driving business growth. Thank you for your interest, and let's dive into the data to uncover some exciting findings about pizza sales!