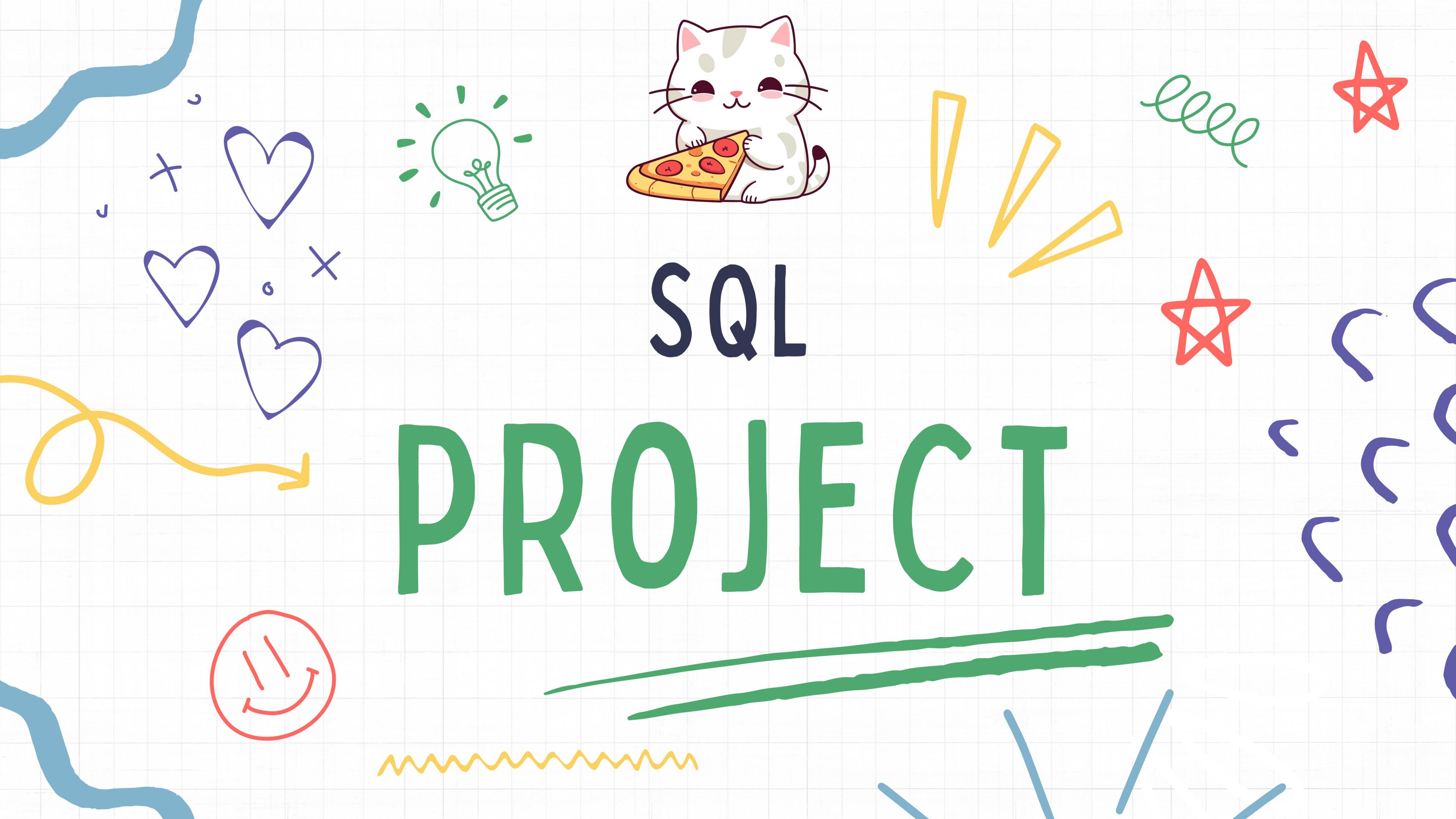


SQL PROJECT



THE CATEGORY



chicken

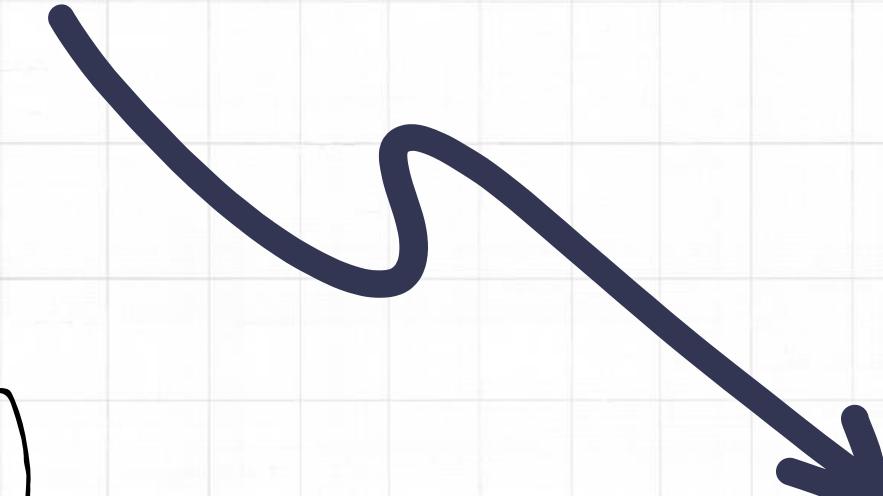
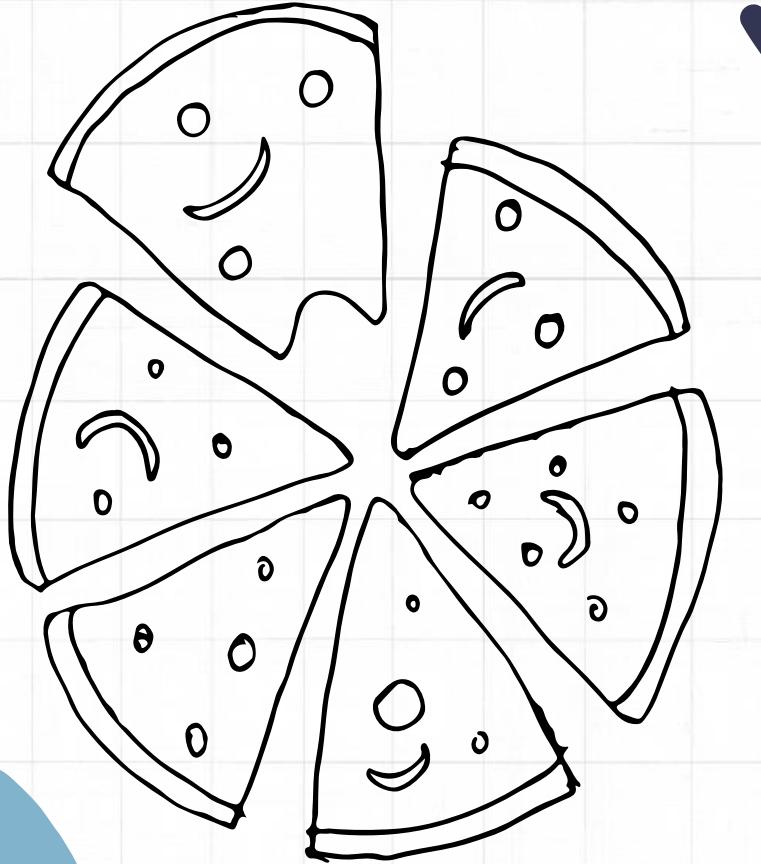


veggie

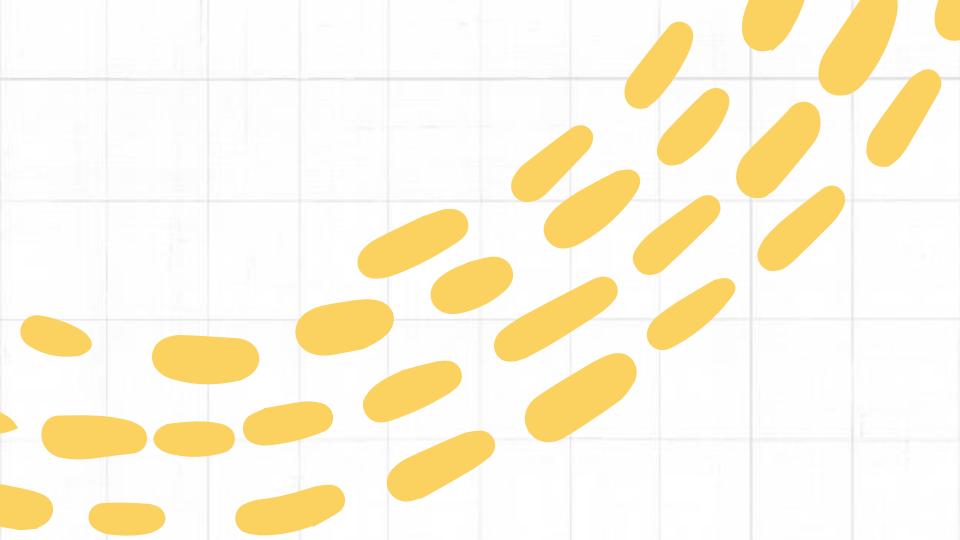


classic

THE PROJECT



Hello friends,
This is a pizza
sale project in
which there are
some questions of
basic,
intermediate and
advanced..



BASIC

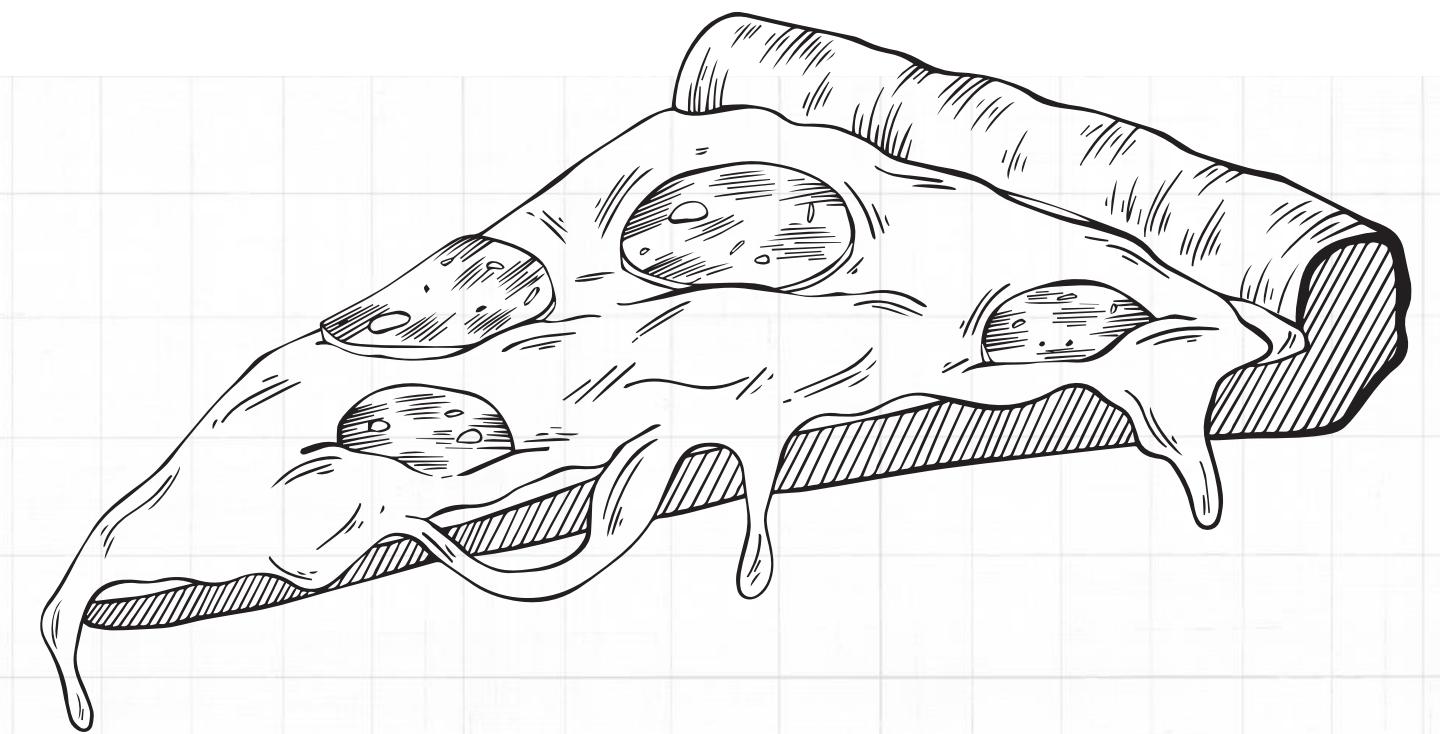
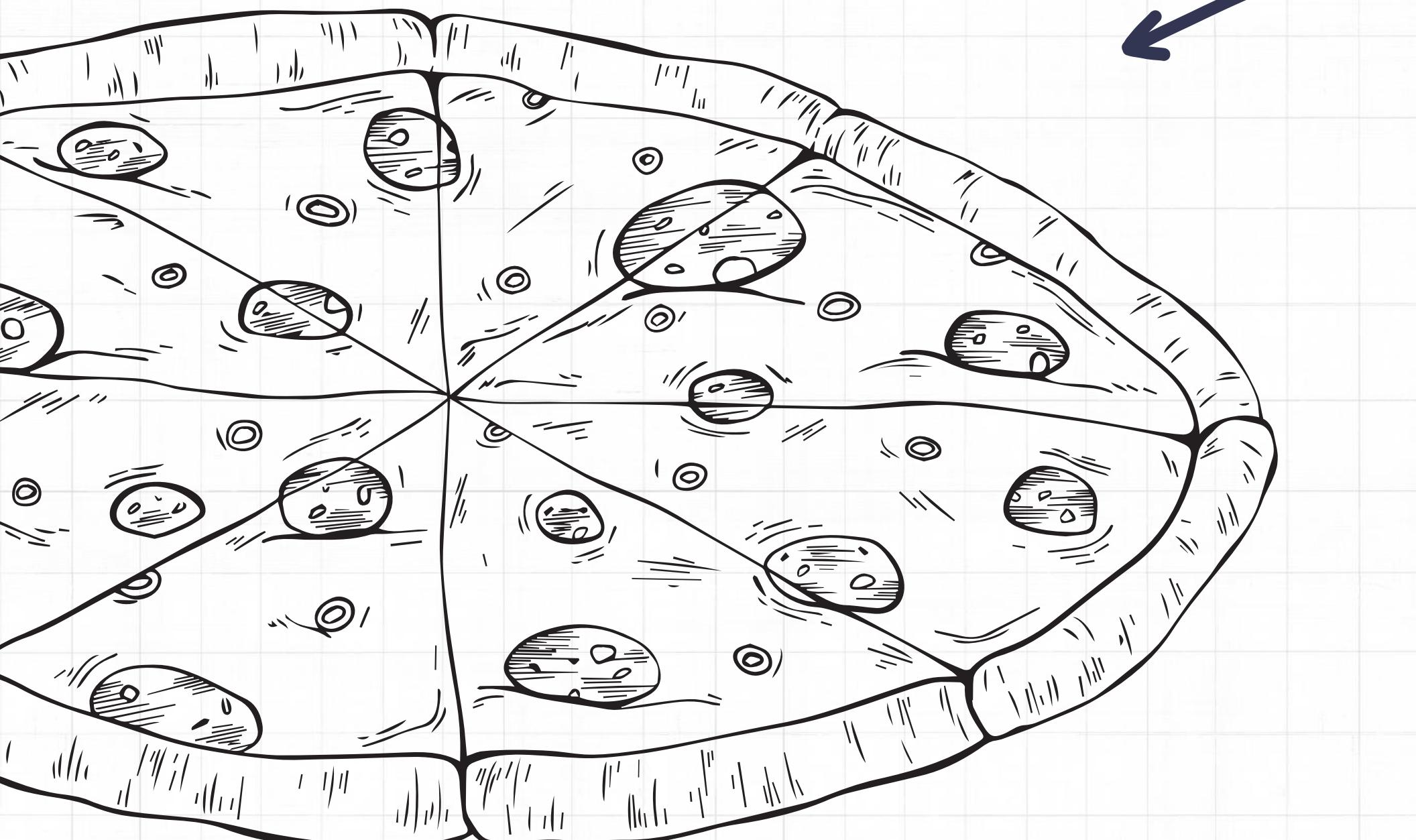
Q1. Retrieve the total number of orders placed.

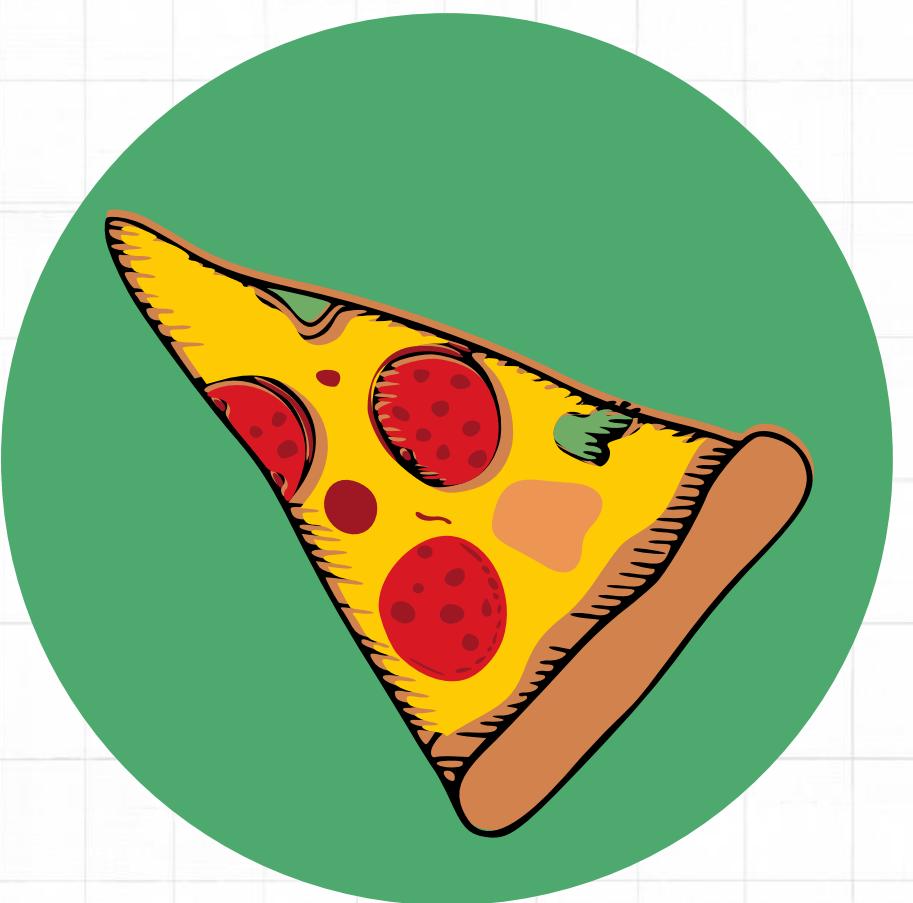
SELECT

COUNT(order_id) AS Total_Orders

FROM

orders





Q2. Calculate the total revenue generated from pizza sales.

SELECT

ROUND(SUM(order_details.quantity * pizzas.price),
2) **AS Total_sale**

FROM

order_details

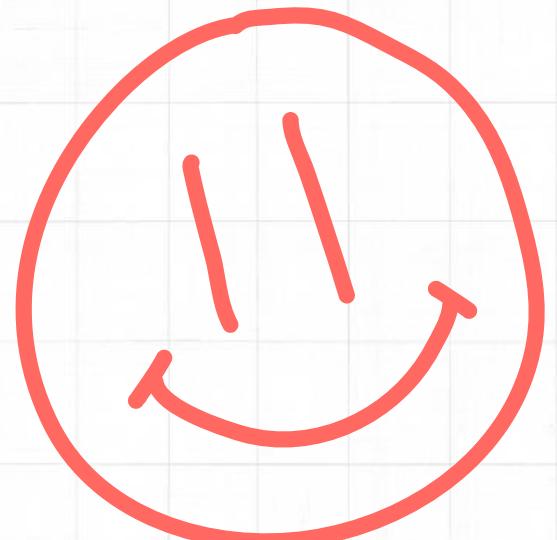
JOIN

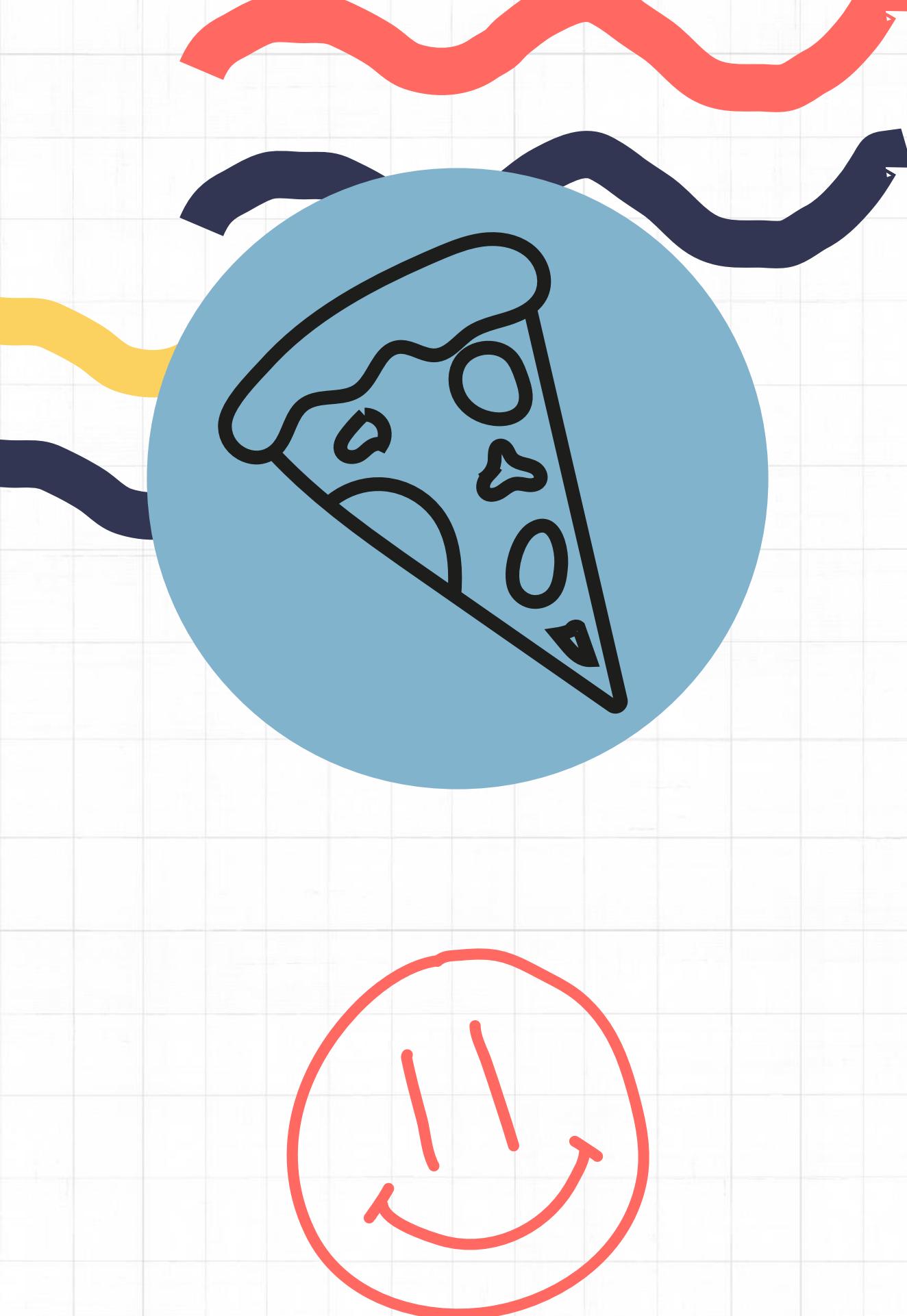
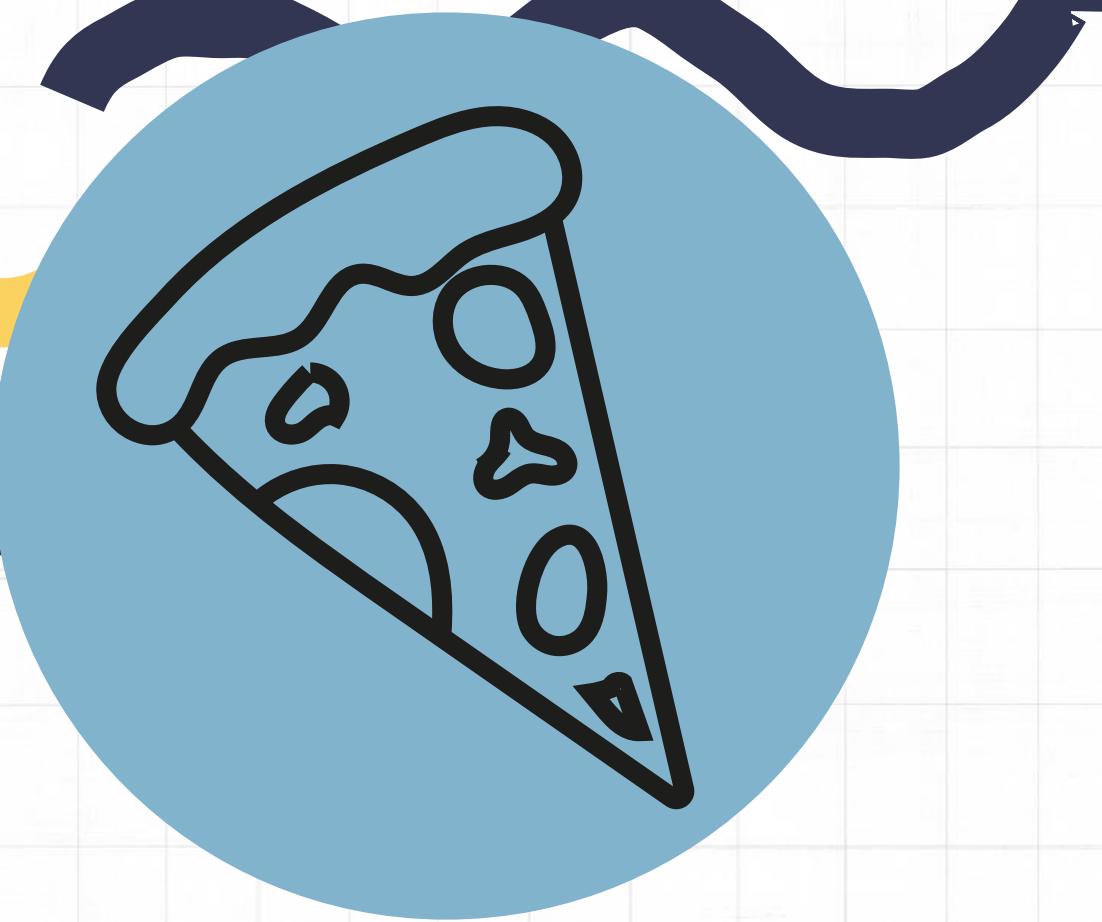
pizzas **ON** pizzas.pizza_id = order_details.pizza_id



Q3. 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;
```





Q4. IDENTIFY THE
MOST COMMON
PIZZA SIZE ORDERED.

```
SELECT  
    quantity, COUNT(order_details_id) AS Most_order_items  
FROM  
    order_details  
GROUP BY quantity
```



Q5. 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
```

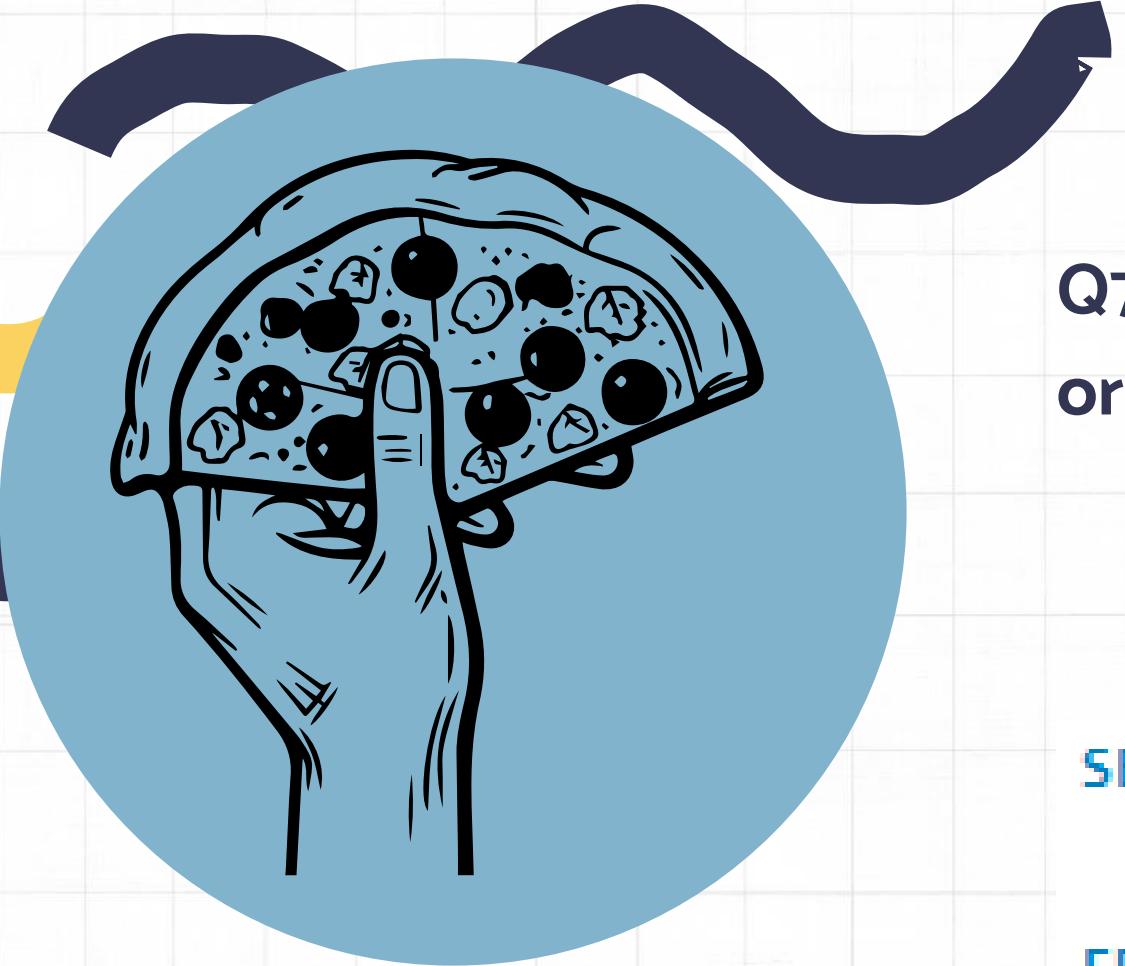


INTERMEDIATE

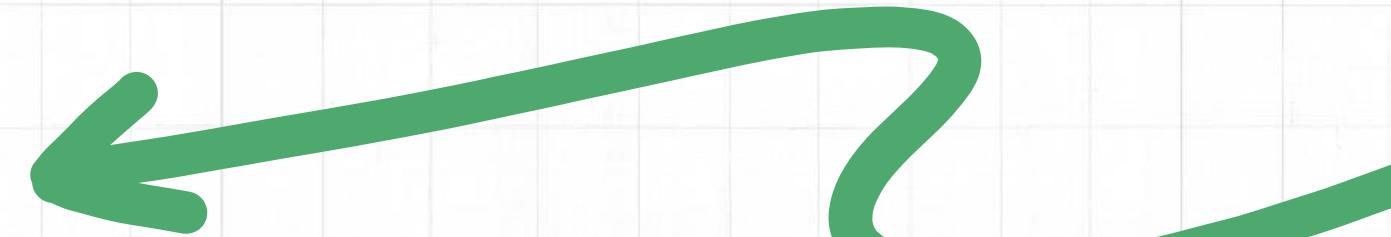


Q6. Join the necessary tables to find the total quantity of each pizza category ordered.

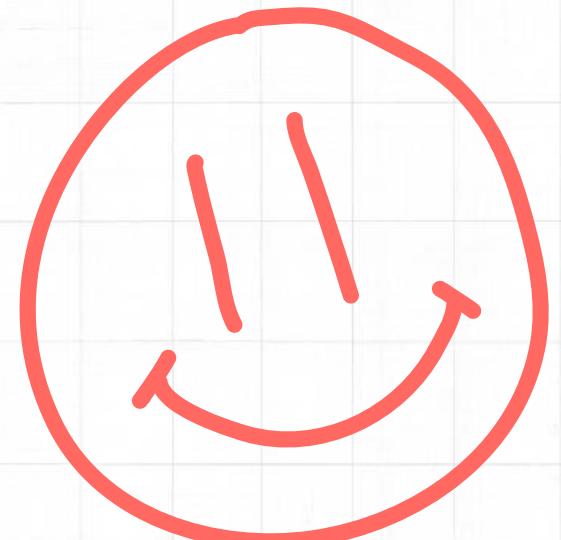
```
SELECT  
    pizza_types.category,  
    SUM(order_details.quantity) AS quantity  
FROM  
    pizza_types  
    JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
    JOIN  
    order_details ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.category  
ORDER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

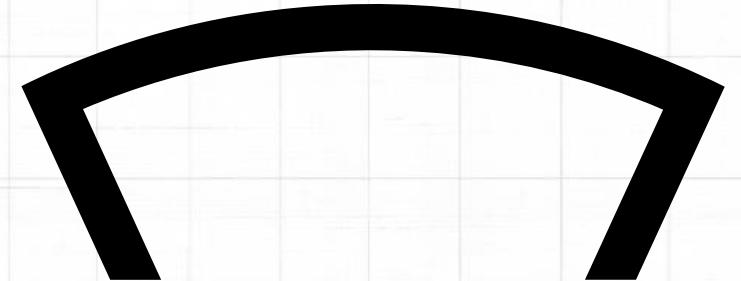


Q7. Determine the distribution of orders by hour of the day.

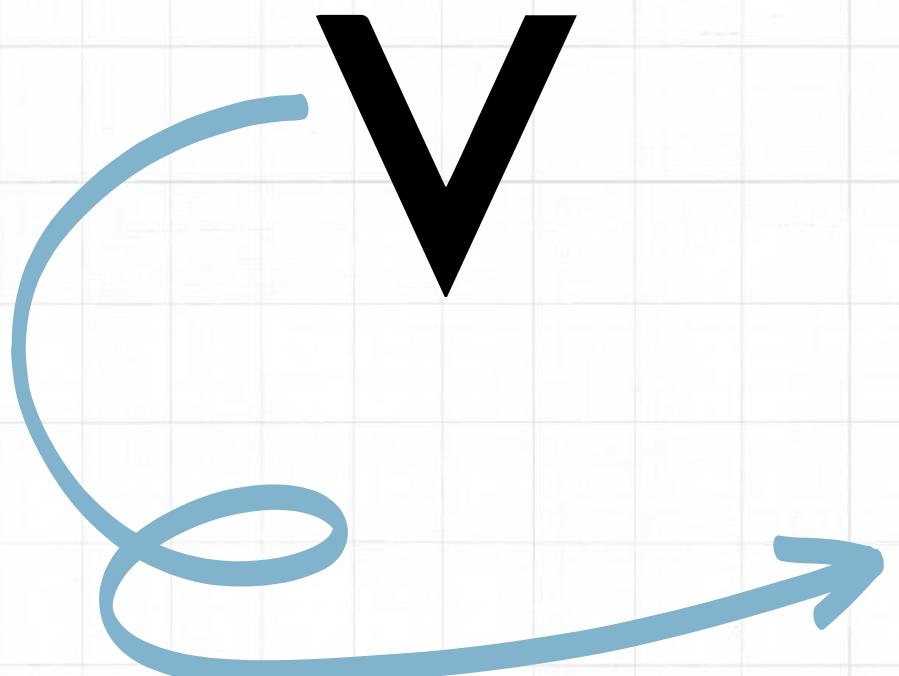


```
SELECT  
    HOUR(order_time) AS hours, COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(order_time)
```





PIZZA



Q8. Join relevant tables to find
the category-wise distribution of
pizzas.

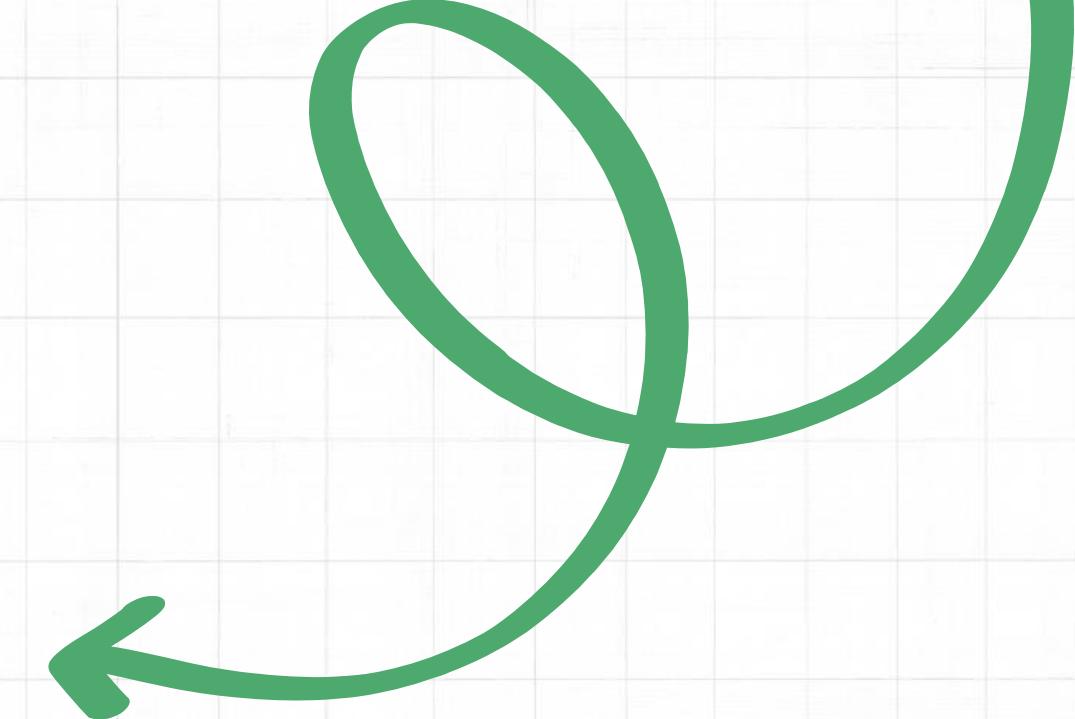
SELECT

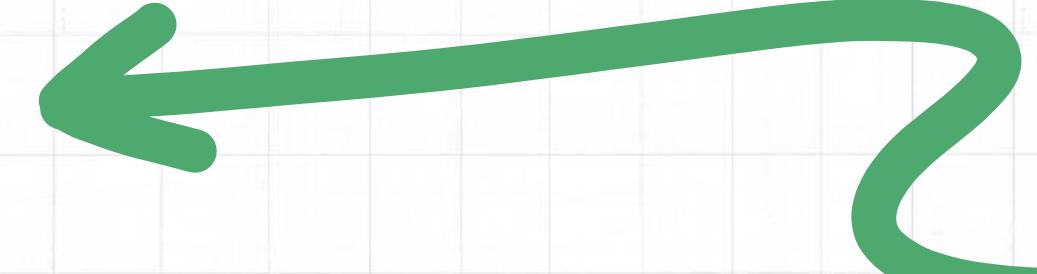
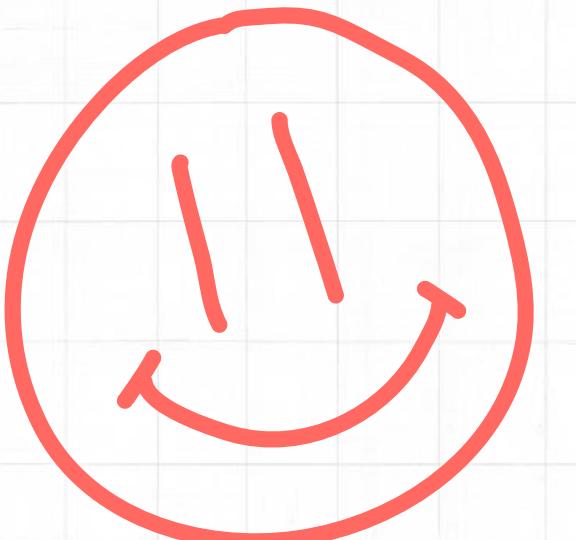
category, COUNT(name) AS Num_of_Category

FROM

pizza_types

GROUP BY category



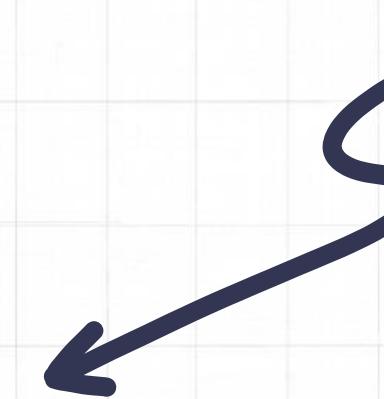
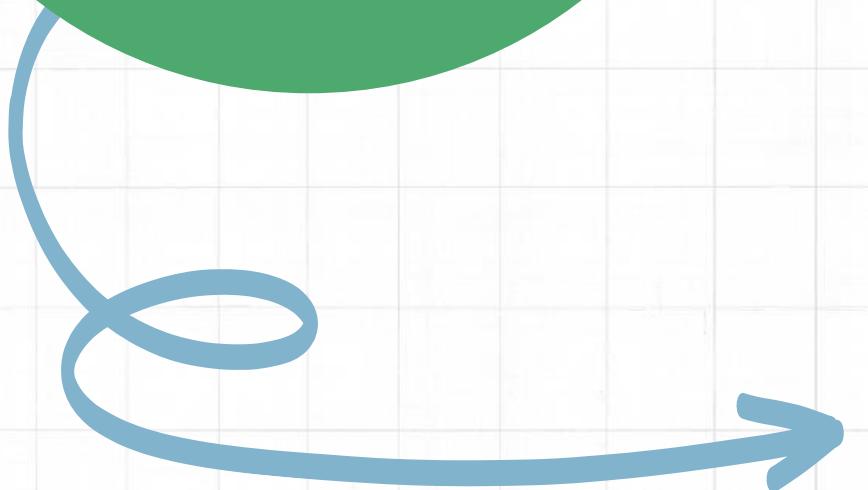
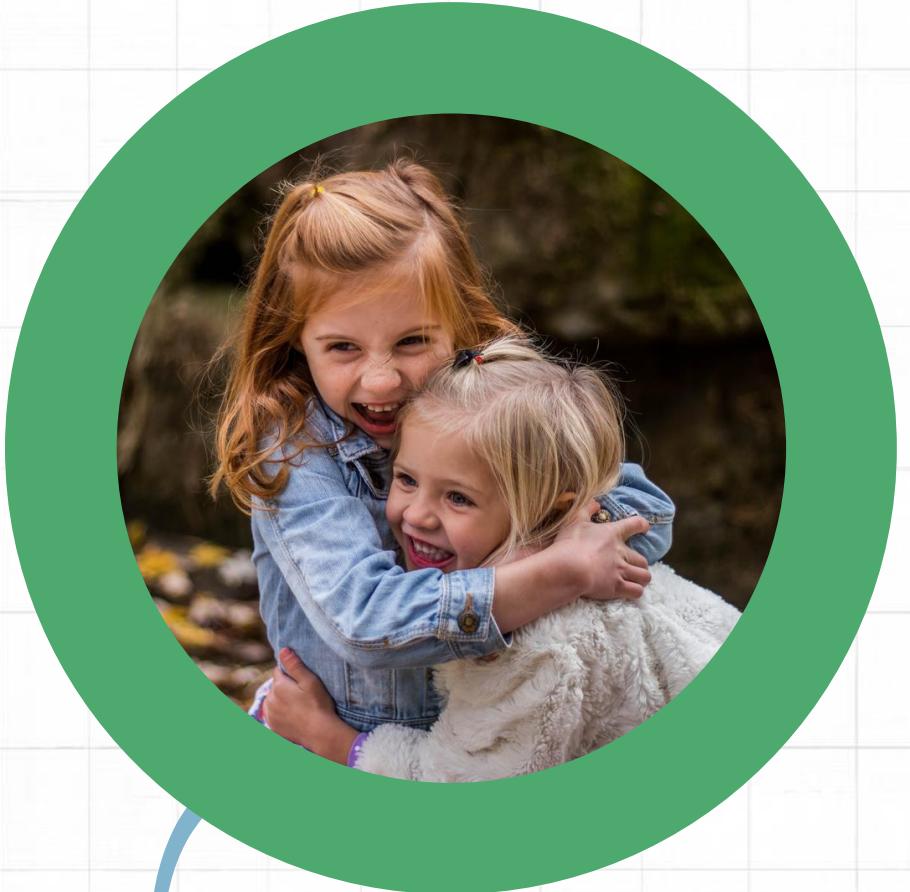


Q9. Determine the top 3 most ordered pizza types based on revenue.

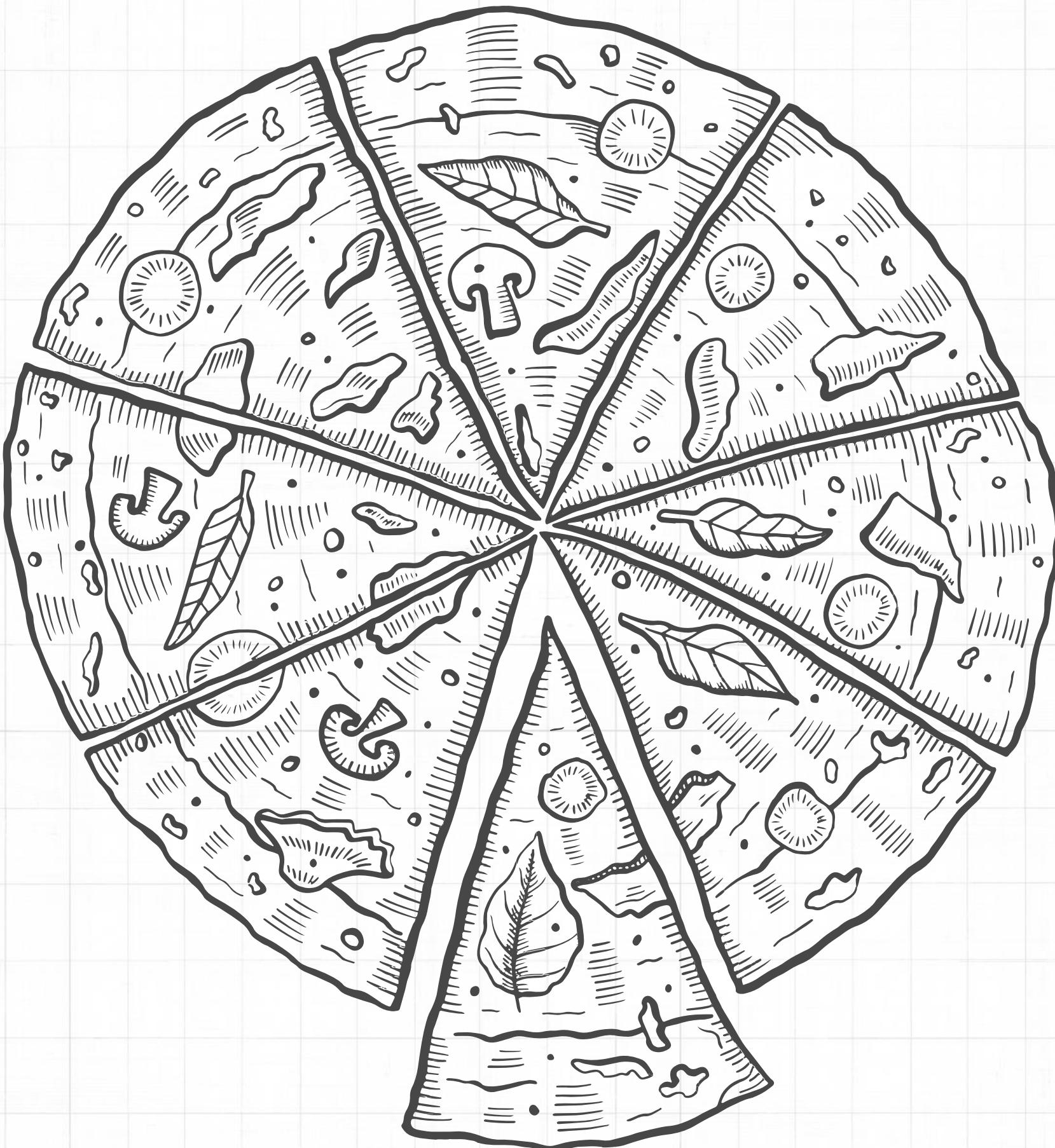
```
SELECT
    pizza_types.name AS Top_Three_most_order,
    SUM(order_details.quantity * pizzas.price) AS Revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name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

ADVANCED

Q10. 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_sale
    )
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
```

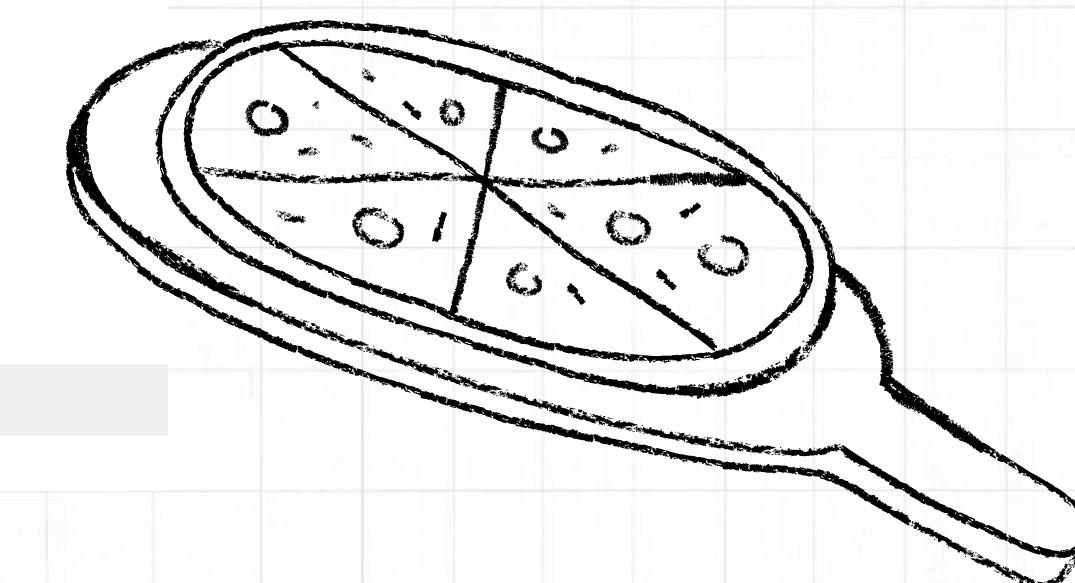


Q11. Analyze the cumulative revenue generated over time.

```
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
```

Q12. 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_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 rnk <= 3
```



THANK YOU

