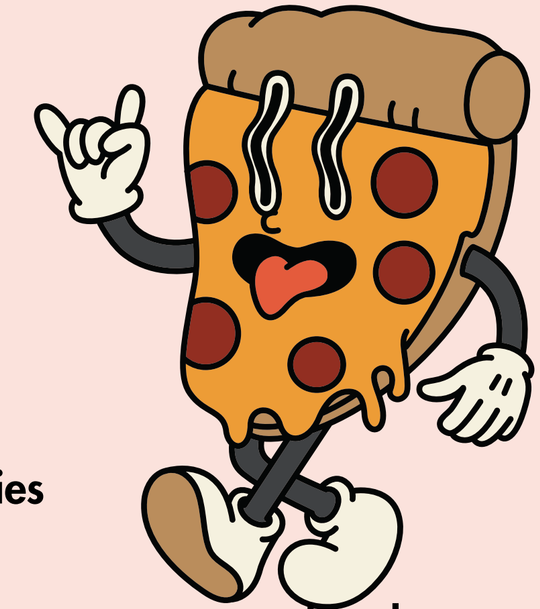
SQL PROJECT ON PIZZA SALES

HELLO!

Hello!!.. My name is Shahreyar Anwar.

In this project i have utilise SQL Query to solve question that where related to pizza sales

Questions



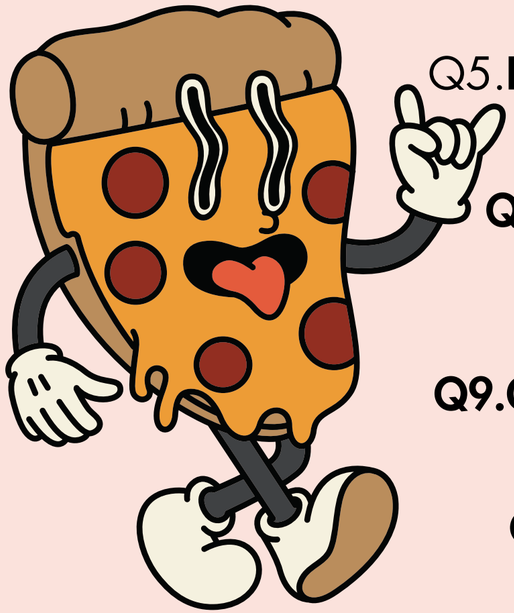
Q1.Retrieve the total number of orders placed.

Q2.Calculate the total revenue generated from pizza sales.

Q3.Identify the highest-priced pizza.

Q4.Identify the most common pizza size ordered.

Q5.List the top 5 most ordered pizza types along with their quantities



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

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

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

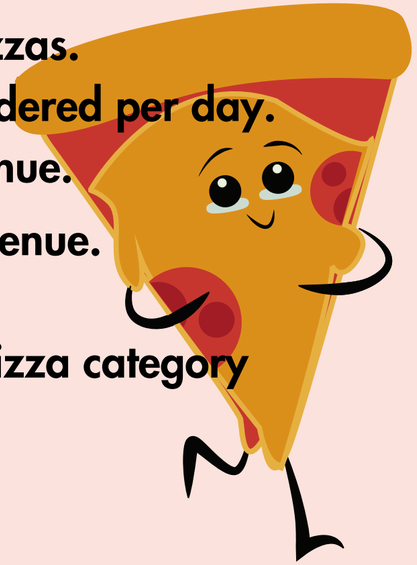
Q9.Group the orders by date and calculate the average number of pizzas ordered per day.

Q10.Determine the top 3 most ordered pizza types based on revenue.

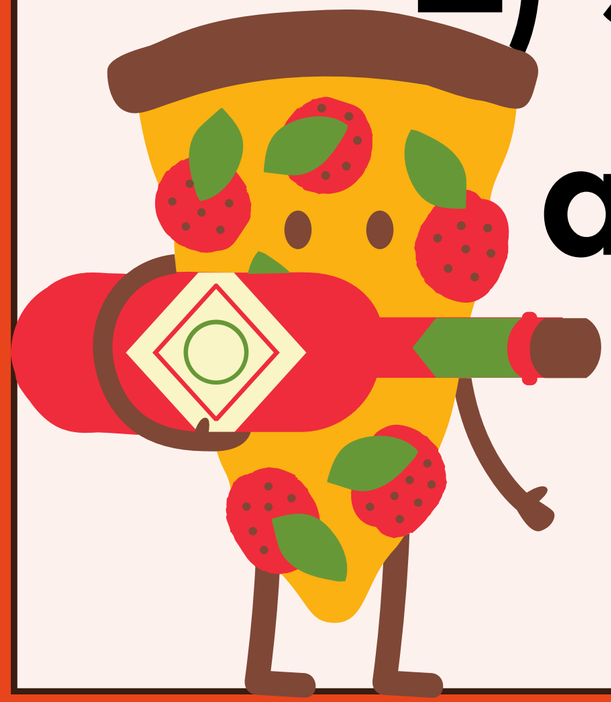
Q11.Calculate the percentage contribution of each pizza type to total revenue.

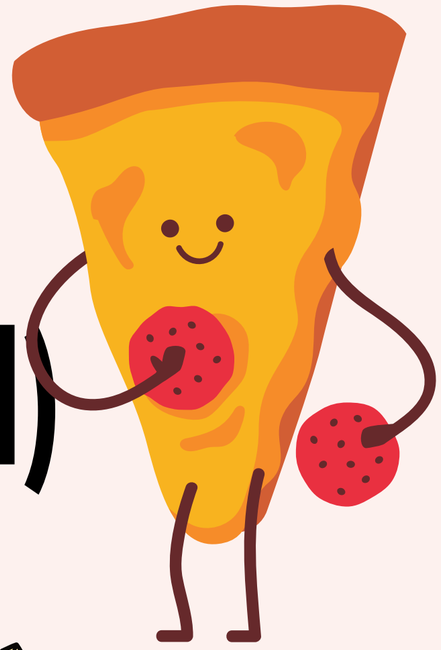
Q12.Analyze the cumulative revenue generated over time.

Q13.Determine the top 3 most ordered pizza types based on revenue for each pizza category



Q 1.Retrieve the total number of orders placed.





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

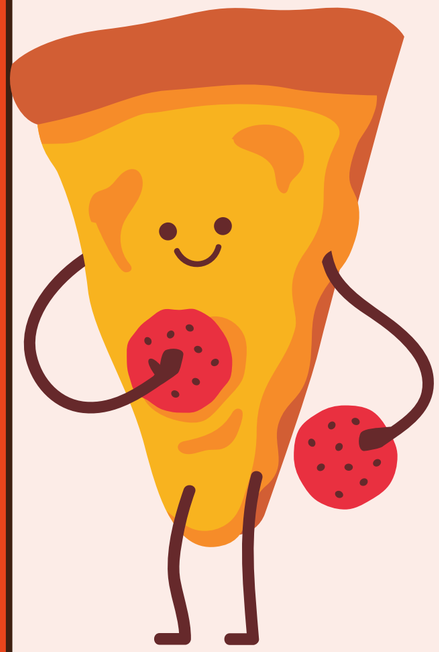
Q 2.Calculate the total revenue generated from pizza sales.

```
=) 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
```

Q 3. 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;
```

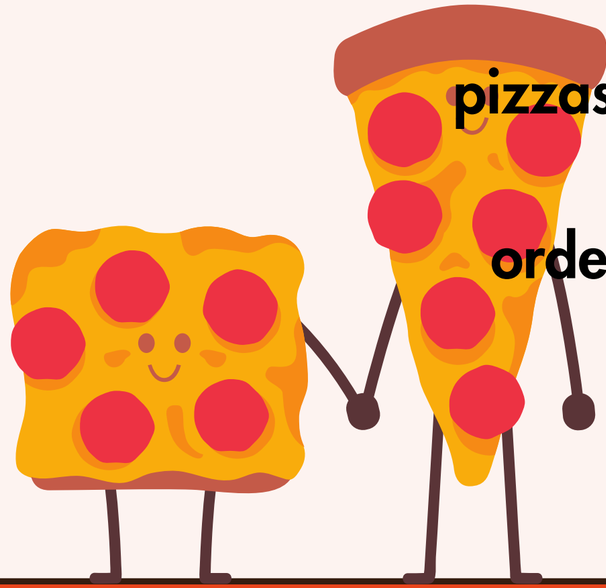
Q 4.Identify the most common pizza size ordered



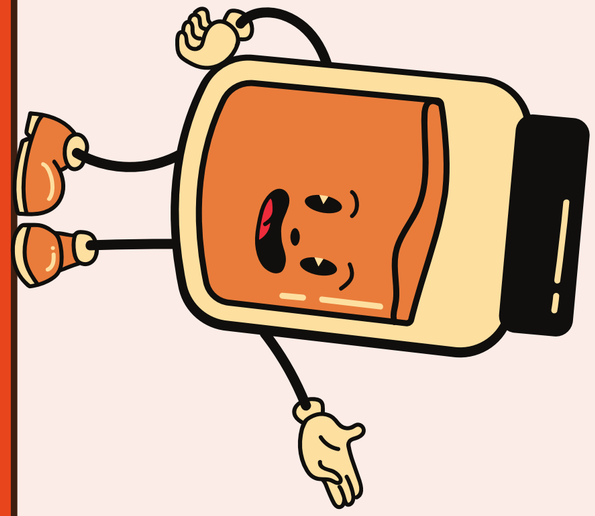
```
=) SELECT pizzas.size,
COUNT(order_details.order_details_id) AS order_count
FROM
pizzas
JOIN
order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

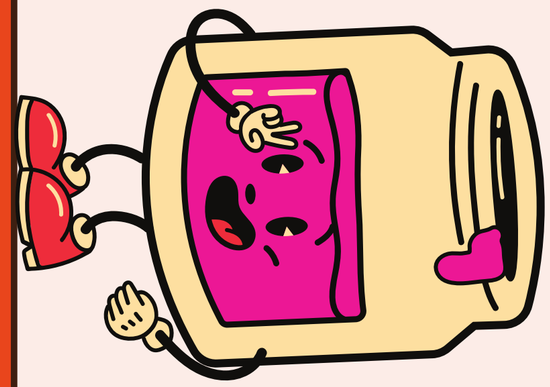
Q 5.List the top 5 most ordered pizza types along with their quantities

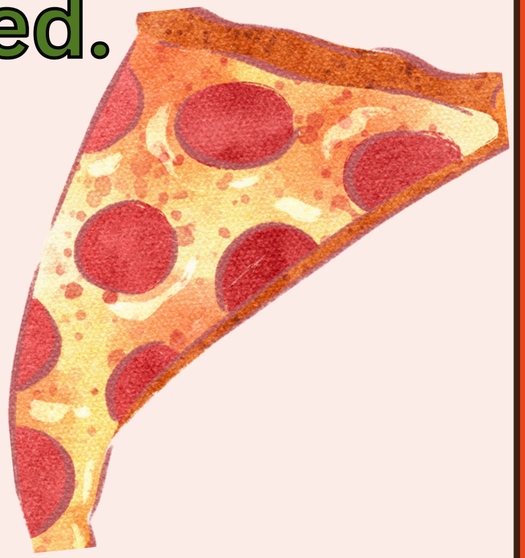
=) SELECT
pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
pizza_types
JOIN
pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN
order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;



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

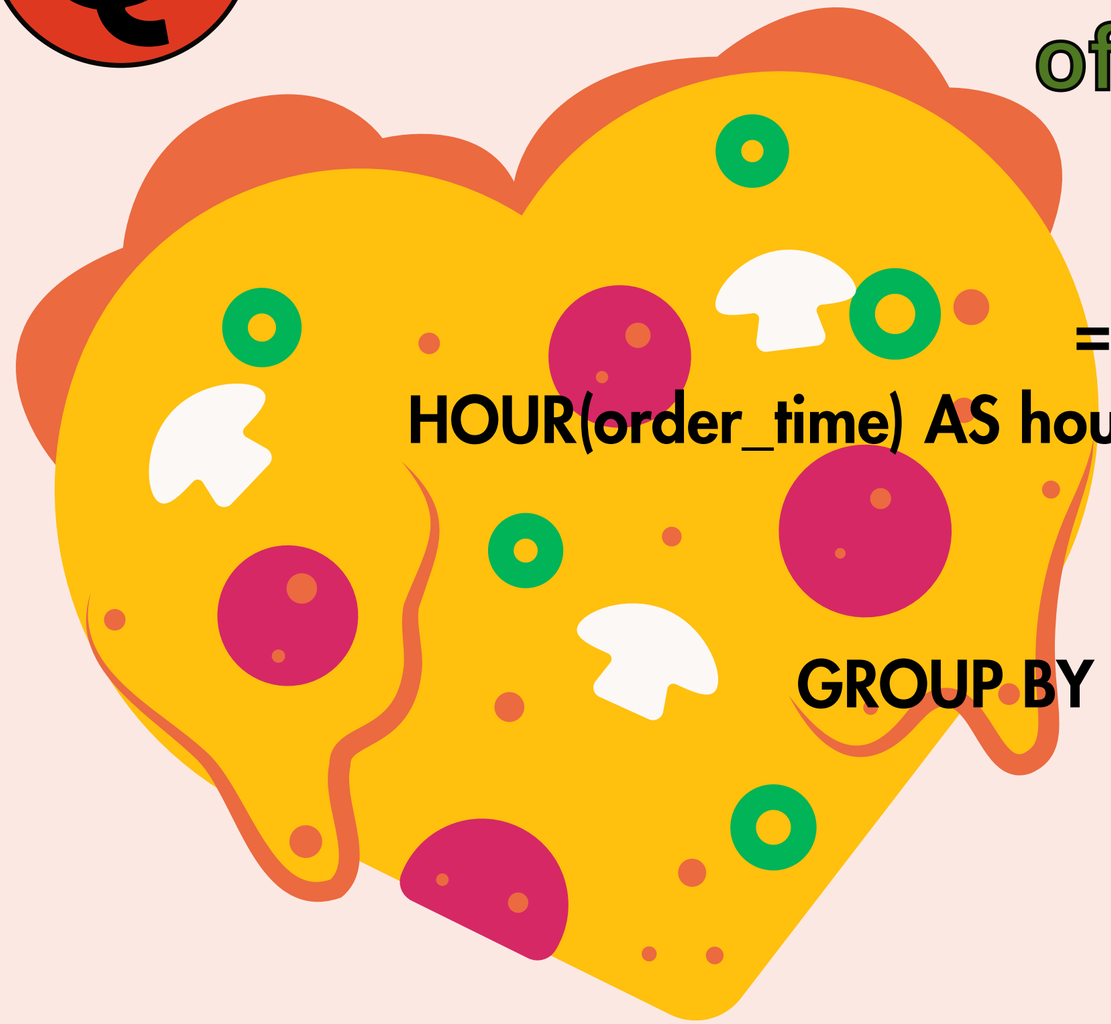




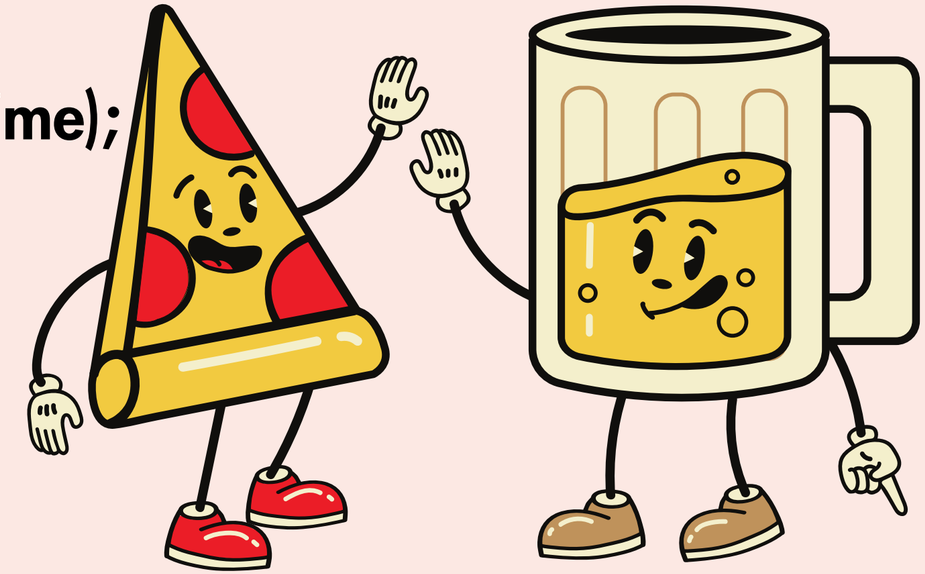


```
=) SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity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

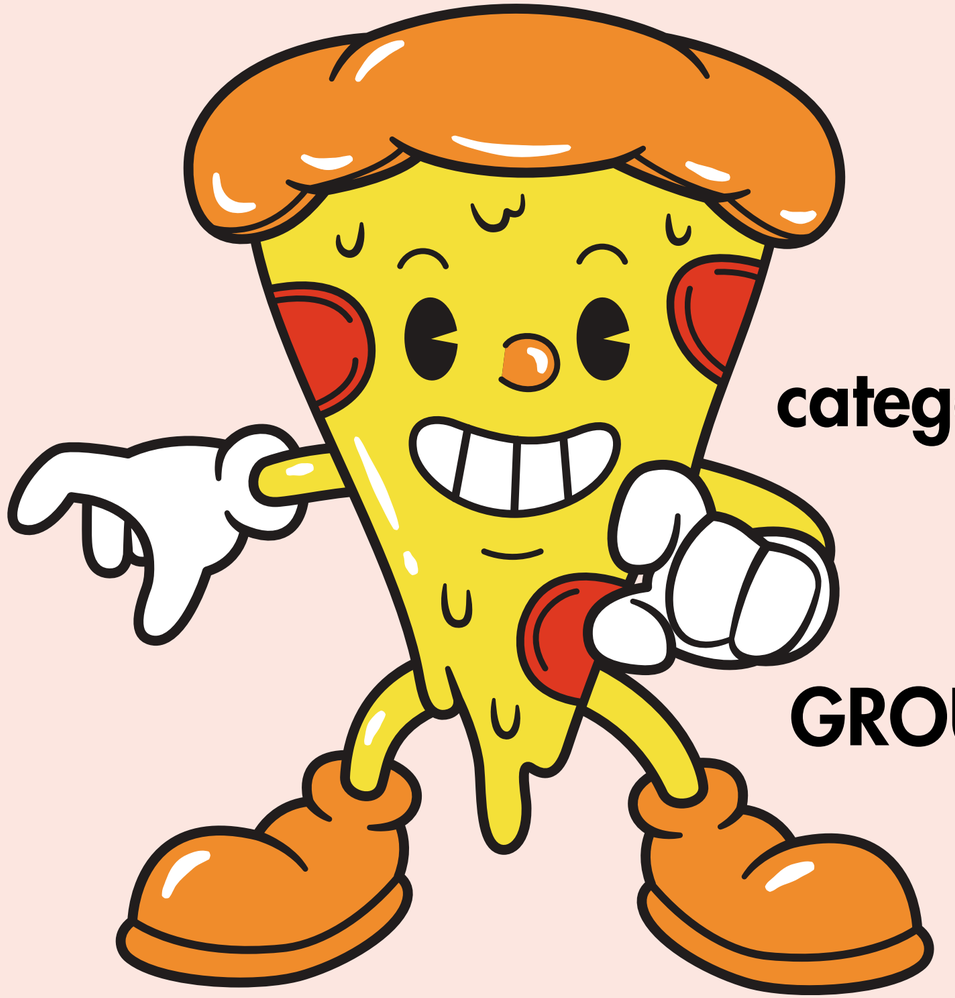
Q 7.Determine the distribution of orders by hour of the day.



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



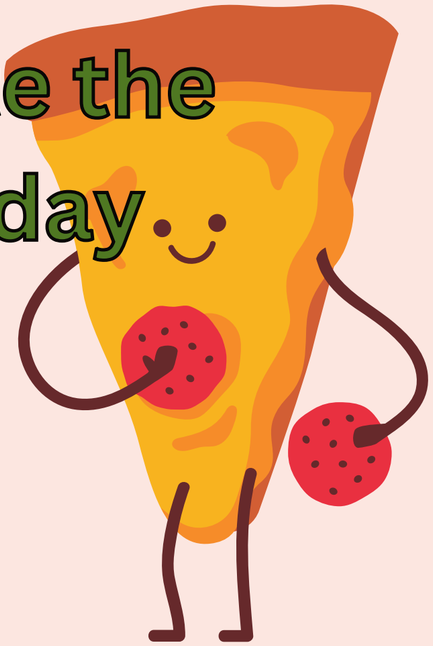
Q 8.Join relevant tables to find the category-wise distribution of pizzas



=) SELECT
category, COUNT(name)
FROM
pizza_types
GROUP BY category;

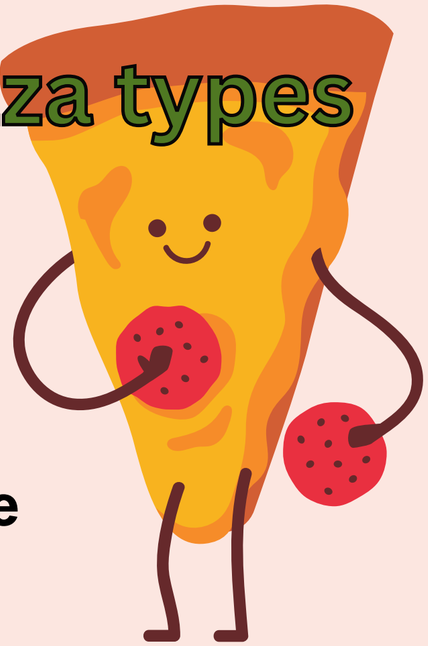
Q

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, 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;
```

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

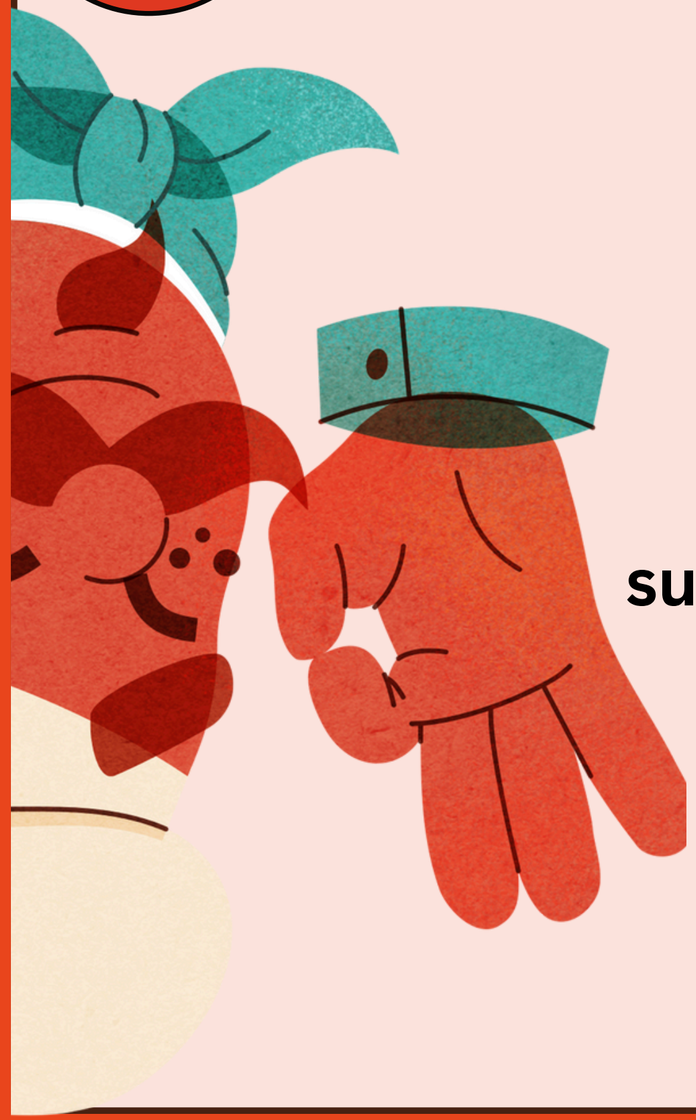


```
=)SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue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

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

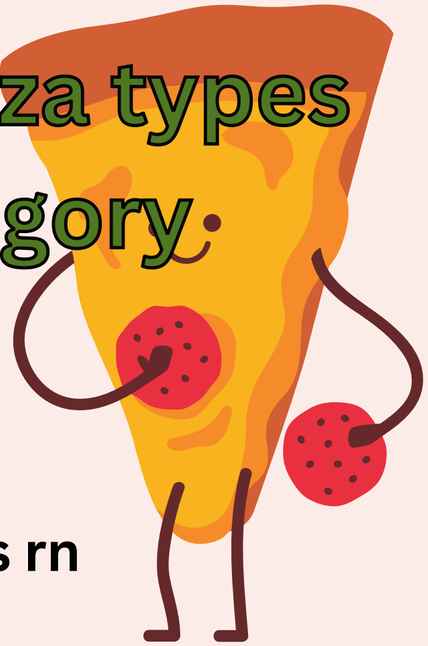
```
=)SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
            ROUND(SUM(order_details.quantity * pizzas.price),
                    2) AS total_sales
                FROM
                    order_details
                        JOIN
                    pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
            2) AS revenue
        FROM
    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;
```

Q 12.Analyze the cumulative revenue generated over time

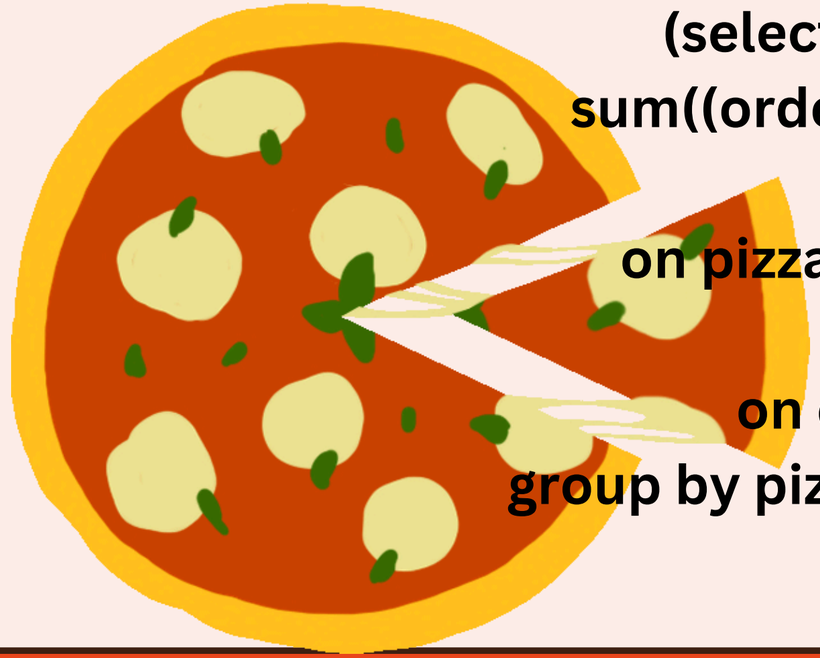


```
select order_date,
sum(revenue) over(order by order_date) as
cum_revenue
from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by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

Q 13.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;



PIZZA PARTY!