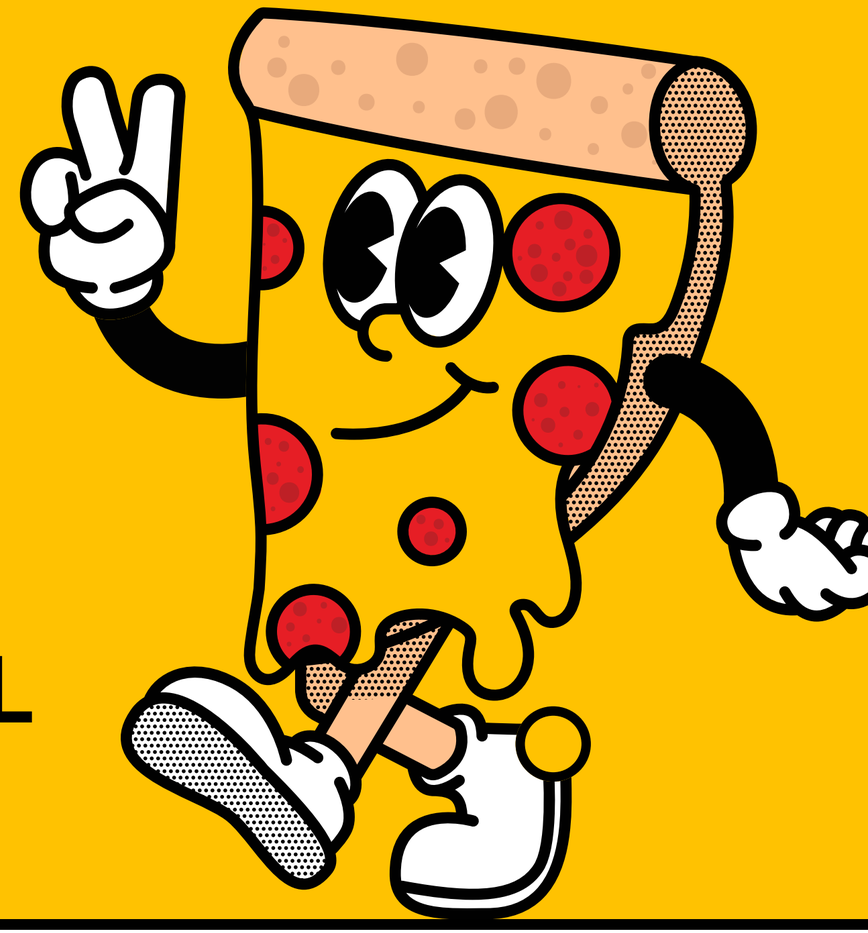
PIZZA SALES ANALYSIS PROJECT

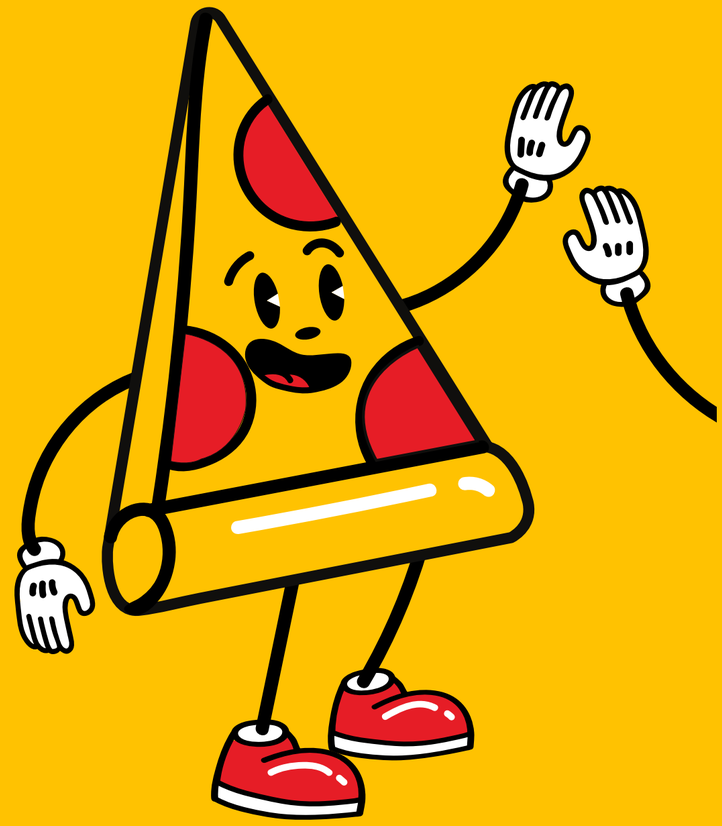
USING MYSQL



Solved 10 Question using SQL

OBJECTIVE

This project involves analyzing pizza sales data using SQL to study revenue.
product performance and customer order trends combining multiple dataset.



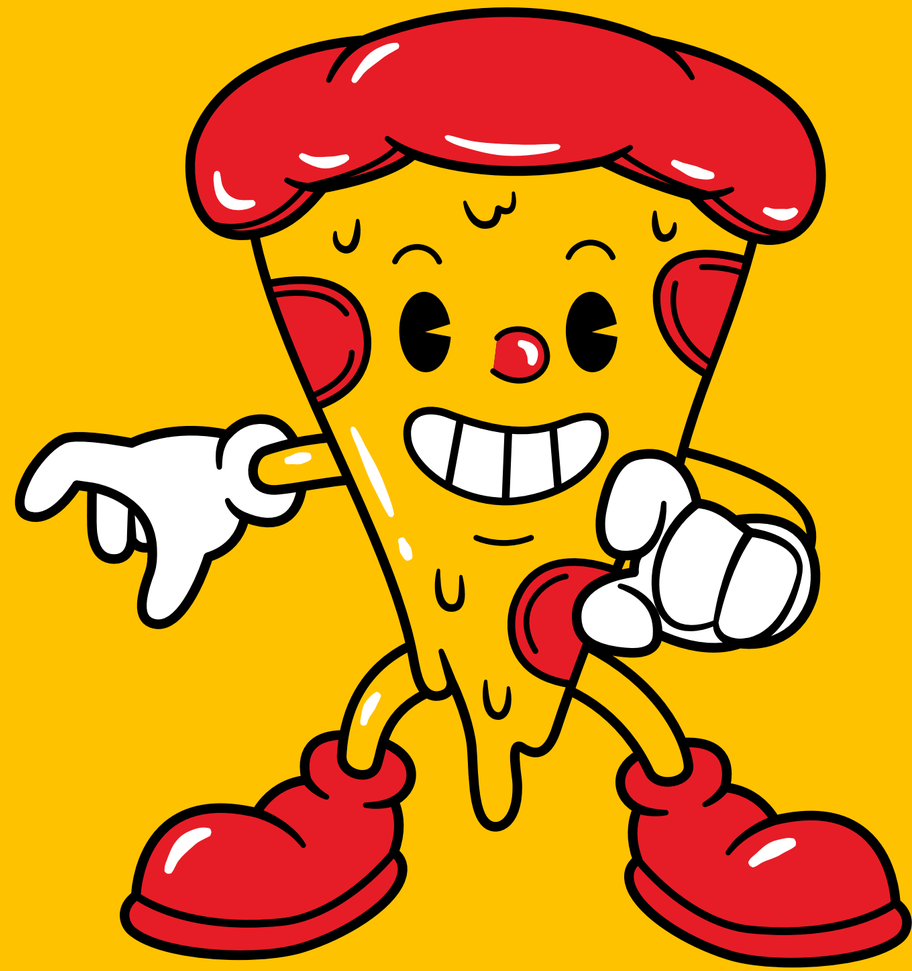
DATASET

- Orders Details :- Contains Order ID , pizza ID , quantity
- Pizza Types :- pizza types ID, Pizza name, category Ingredient
- Pizza :- include pizza size, and prize information
- Order :- stores order ID order date and order time

Q1 - 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 5;
```

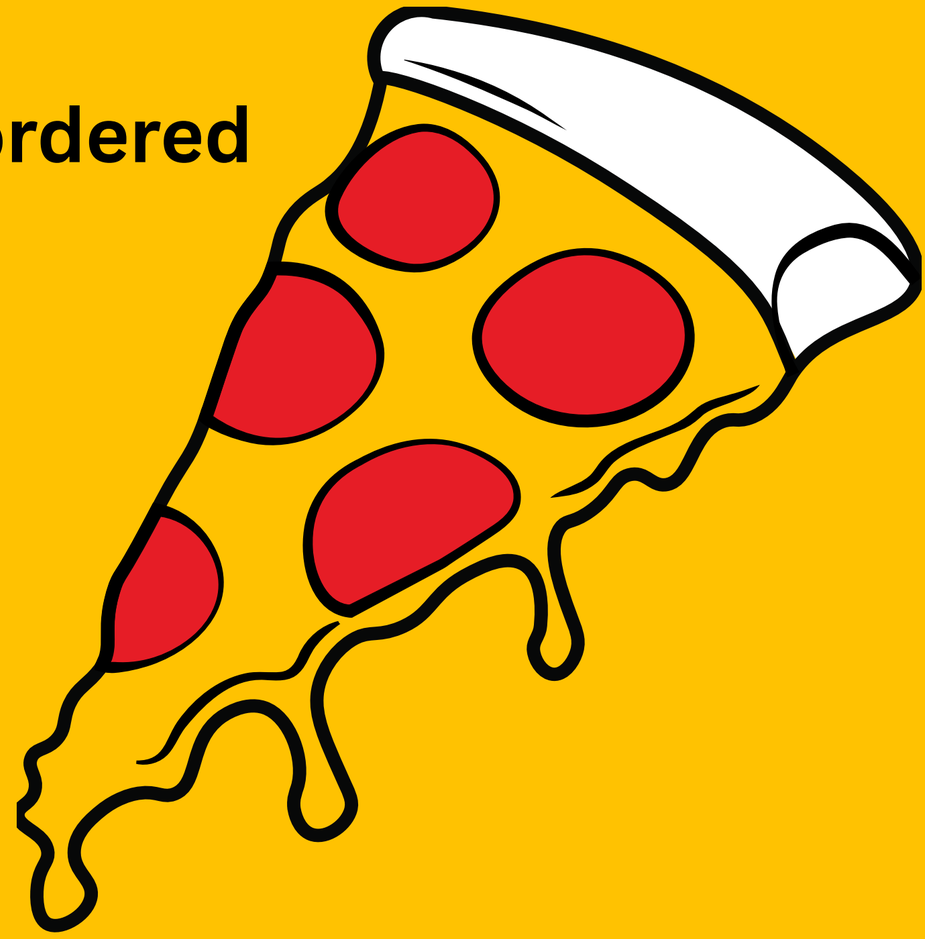
name	price
The Greek Pizza	35.95
The Greek Pizza	25.5
The Brie Carre Pizza	23.65
The Italian Vegetables Pizza	21
The Barbecue Chicken Pizza	20.75



Q2- Identify the most common pizza size ordered

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

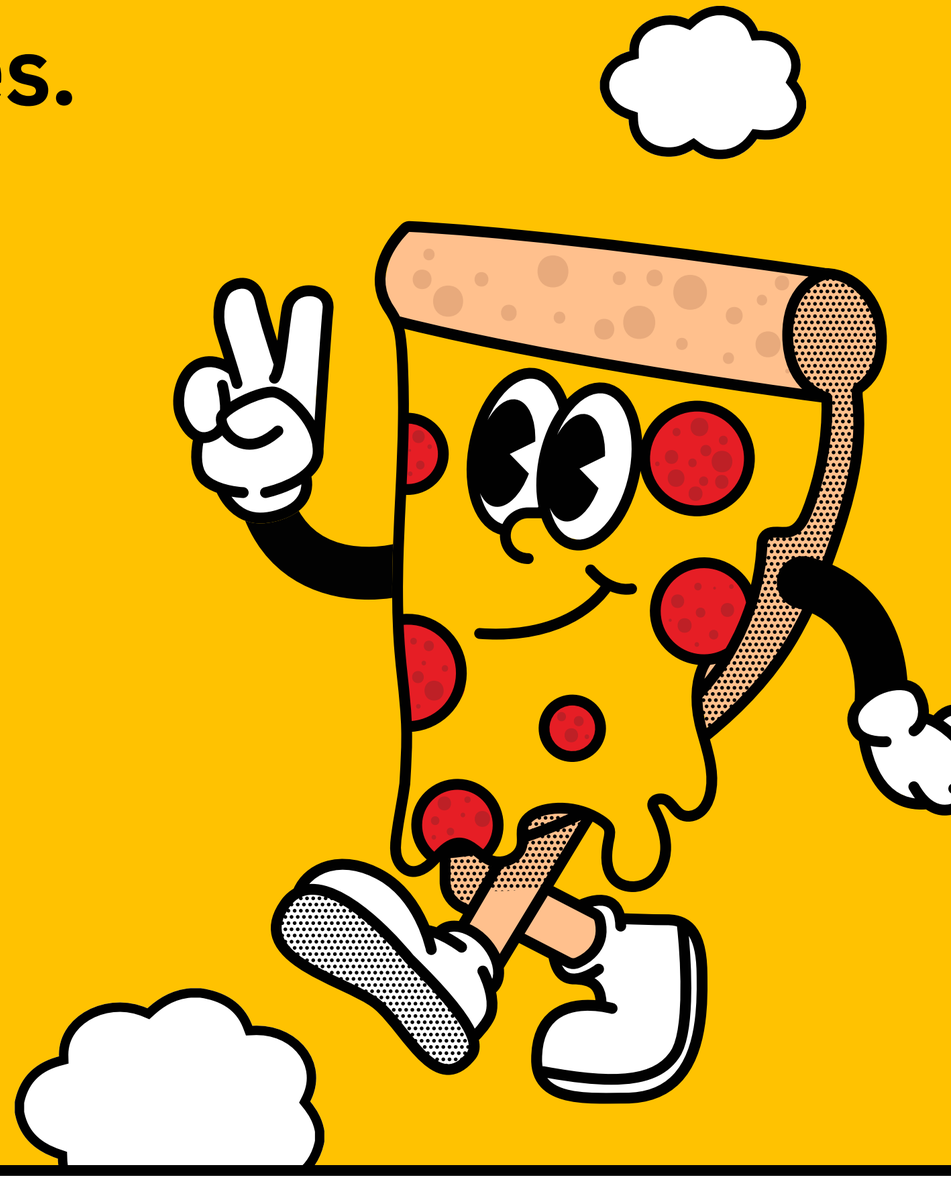
size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28



Q3 - List the top 5 most ordered pizza types along with their quantities.

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

name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371



Q4- Join the necessary tables to find the total quantity of each pizza category ordered.

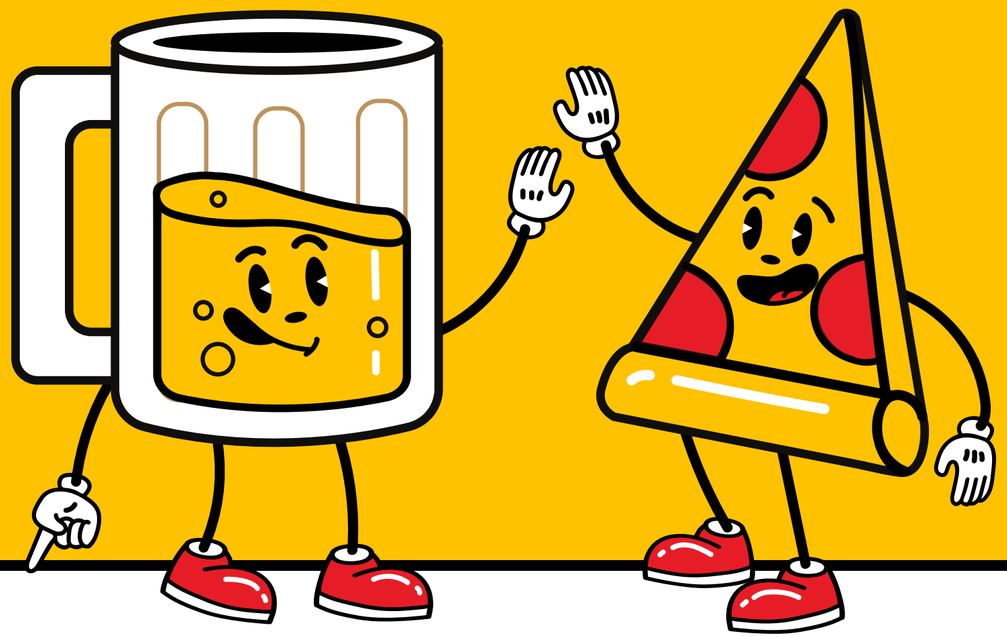
```sql
SELECT
pizza_types.category,
SUM(orders_details.quantity) AS quantity
FROM pizza_types JOIN
pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN
orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q5- Determine the distribution of orders by hour of the day.

```sql
SELECT
HOUR(order_time), COUNT(order_id)
FROM orders
GROUP BY HOUR(order_time);
```

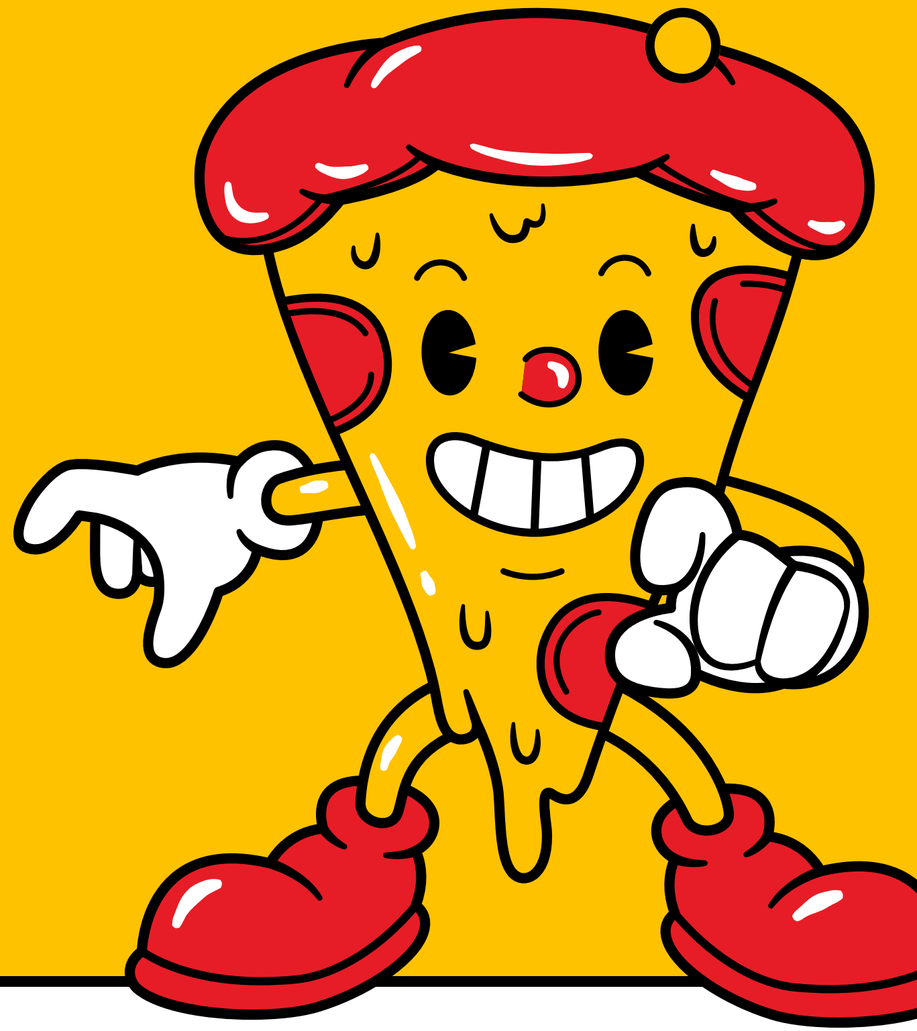
HOUR(order_time)	COUNT(order_id)
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1



Q6- Join relevant tables to find the category-wise distribution of pizzas.

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

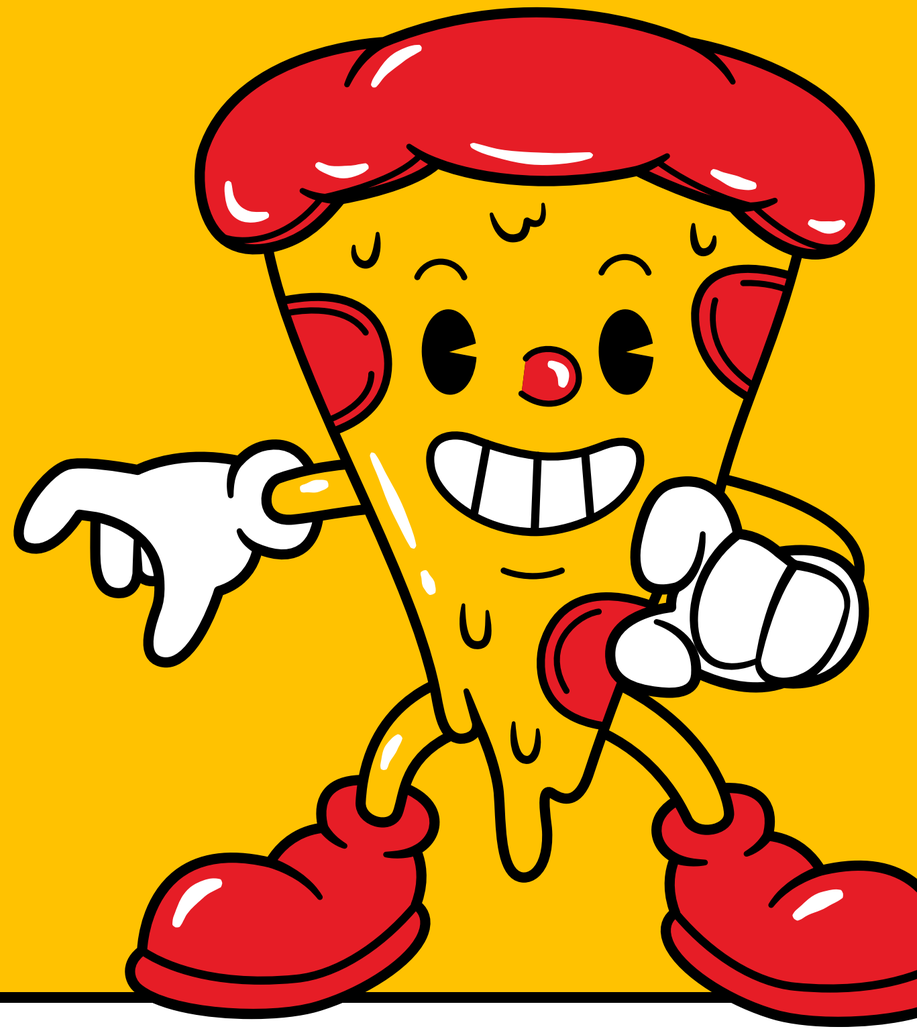
category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9



Q7- Group the orders by date and calculate the average number of pizzas ordered per day.

```sql
SELECT ROUND(avg(quantity), 0) as avg_pizza_ordered_per_day
FROM
(SELECT orders.order_date, SUM(orders_details.quantity) AS quantity
FROM orders
JOIN orders_details ON orders.order_id = orders_details.order_id
GROUP BY orders.order_date) AS order_quantity;
```

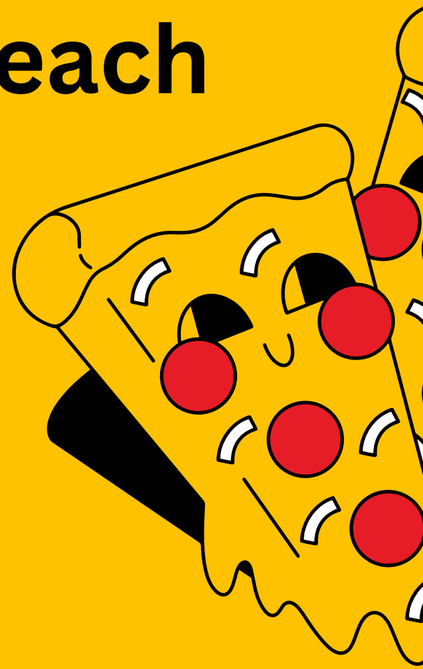
avg_pizza_ordered_per_day
138

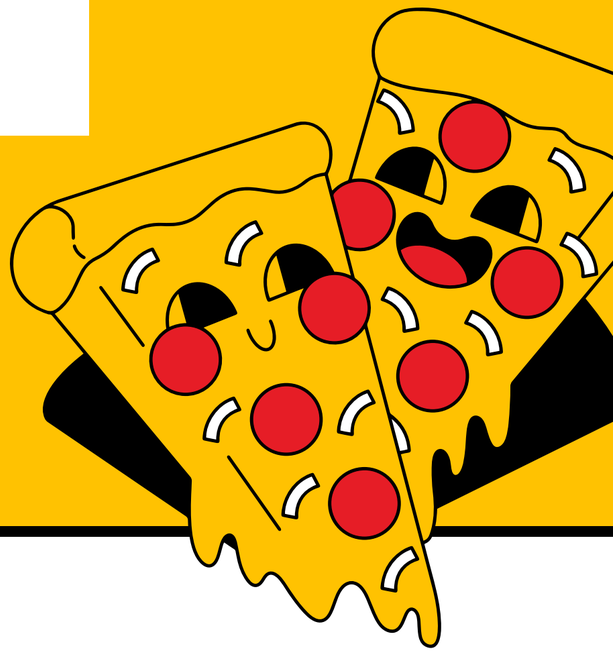


Q8- Calculate the percentage contribution of each pizza type to total revenue.

```
SELECT pizza_types.category,
ROUND(SUM(orders_details.quantity * pizzas.price) / (SELECT
ROUND(SUM(orders_details.quantity * pizzas.price),
2) AS total_sales
FROM orders_details JOIN
pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,2) AS revenue
FROM pizza_types JOIN
pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN
orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

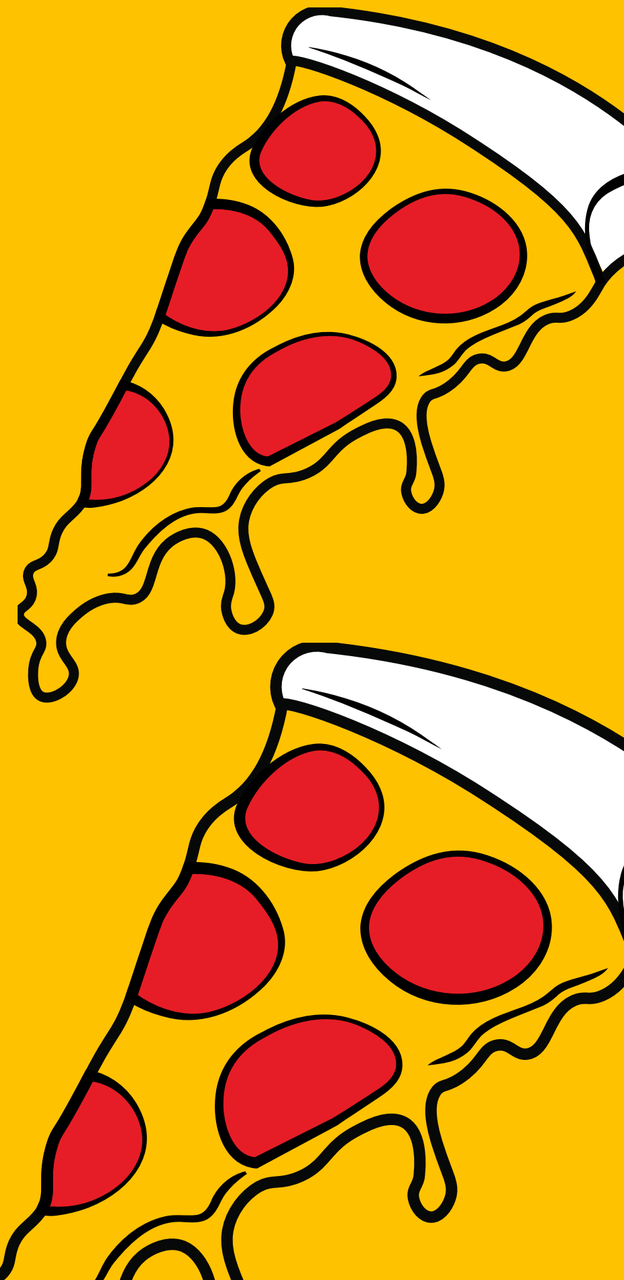




Q9- Analyze the cumulative revenue generated over time.

```
SELECT order_date,
SUM(revenue) OVER(ORDER BY order_date) as cum_revenue
from
(SELECT orders.order_date,
SUM(orders_details.quantity * pizzas.price) as revenue
FROM orders_details JOIN pizzas
on orders_details.pizza_id = pizzas.pizza_id
JOIN orders
on orders.order_id  = orders_details.order_id
GROUP BY orders.order_date) as sales;
```

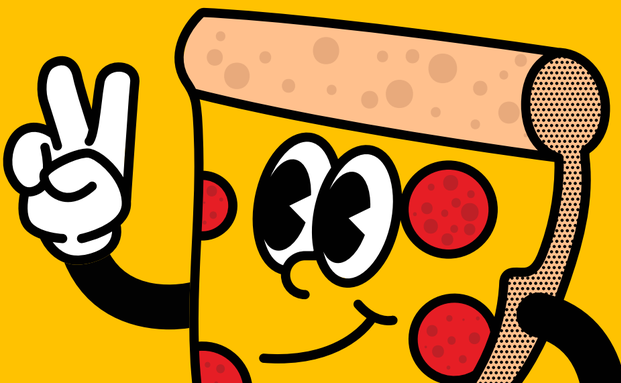
order_date	cum_revenue
2015-01-28	63059.85000000001
2015-01-29	65105.150000000016
2015-01-30	67375.45000000001
2015-01-31	69793.30000000002
2015-02-01	72982.50000000001
2015-02-02	75311.10000000002
2015-02-03	77925.90000000002
2015-02-04	80159.80000000002
2015-02-05	82375.60000000002
2015-02-06	84885.55000000002
2015-02-07	87123.20000000001
2015-02-08	89158.20000000001
2015-02-09	91353.55000000002
2015-02-10	93410.05000000002

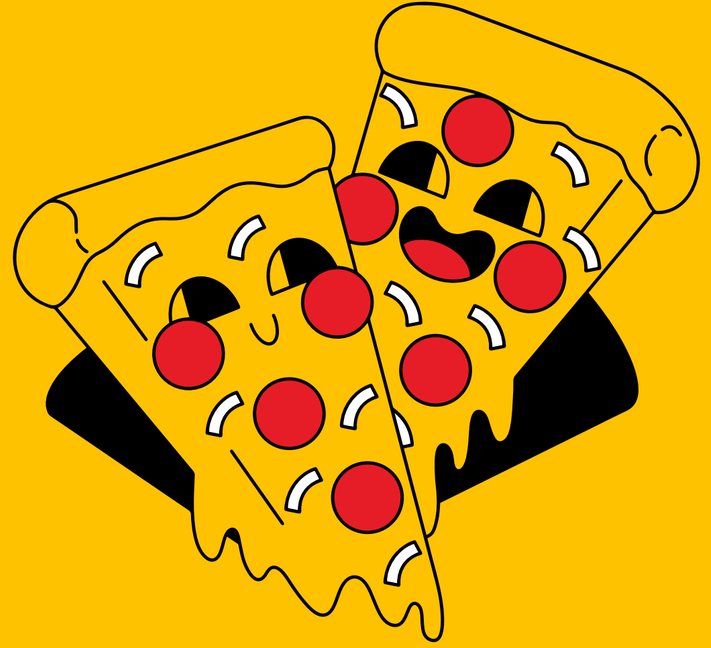


Q10- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
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((orders_details.quantity) * pizzas.price) as revenue
FROM pizza_types JOIN pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN orders_details
on orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category , pizza_types.name) as a)  as b
 where rn <= 3;
```

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5





THANK YOU