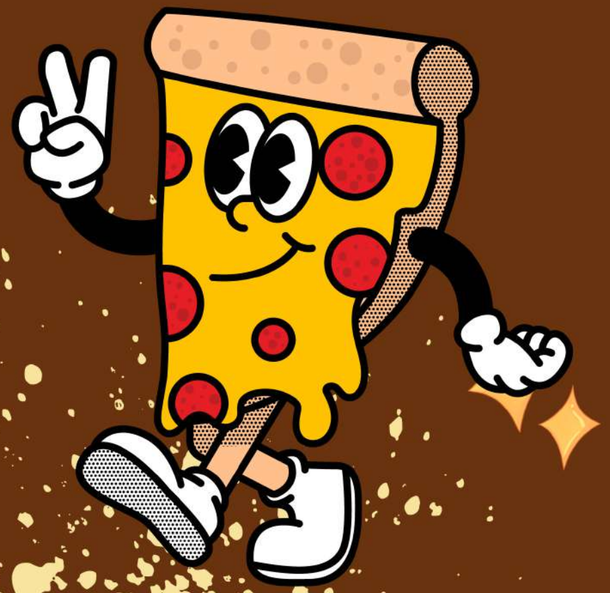
SQL PROJECT ON PIZZA SALES

INTRODUCTION

"My name is Sakshi Sharma, and I developed an SQL project focused on analyzing pizza sales using MySQL. This project explores sales trends, customer preferences, and performance metrics through structured queries and data analysis. By leveraging SQL, I extracted valuable insights that can help improve inventory management, optimize menu offerings, and enhance customer satisfaction in a pizza business



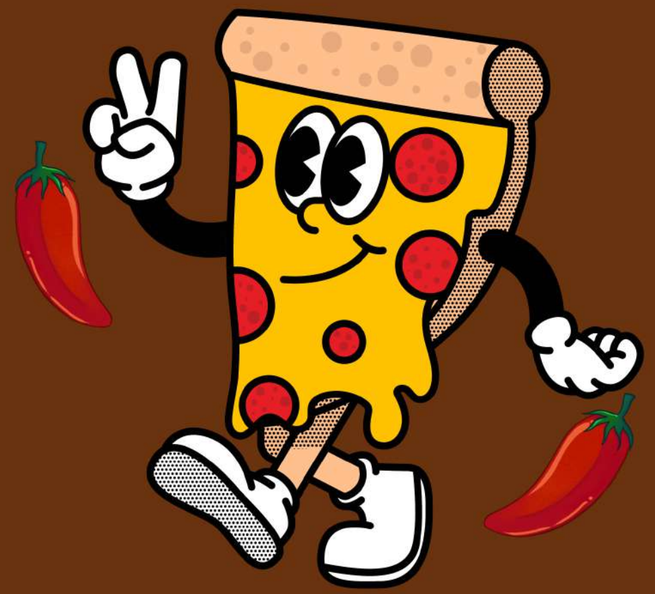
QUESTIONS

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

- Calculate the percentage contribution of each pizza type to total revenue.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Retrieve the total number of orders placed.

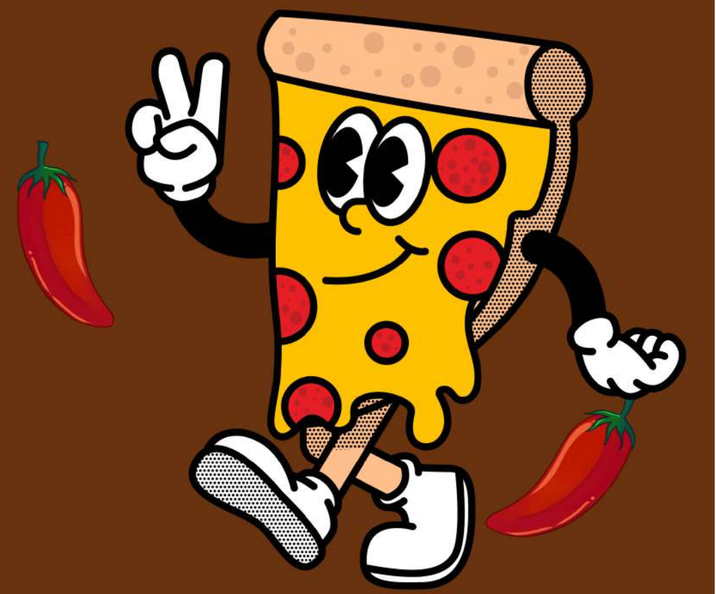


```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

Result Grid

total_orders
21350

Calculate the total revenue generated from pizza sales.

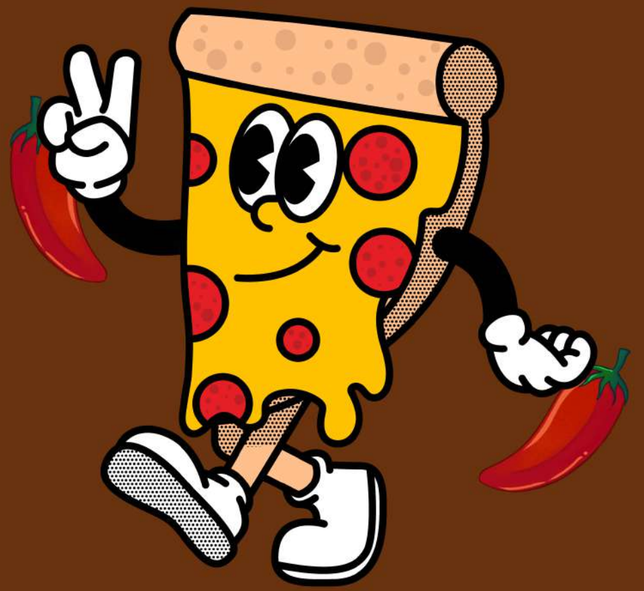


```sql
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid

	total_sales
▶	631697.95

Identify the highest-priced pizza.



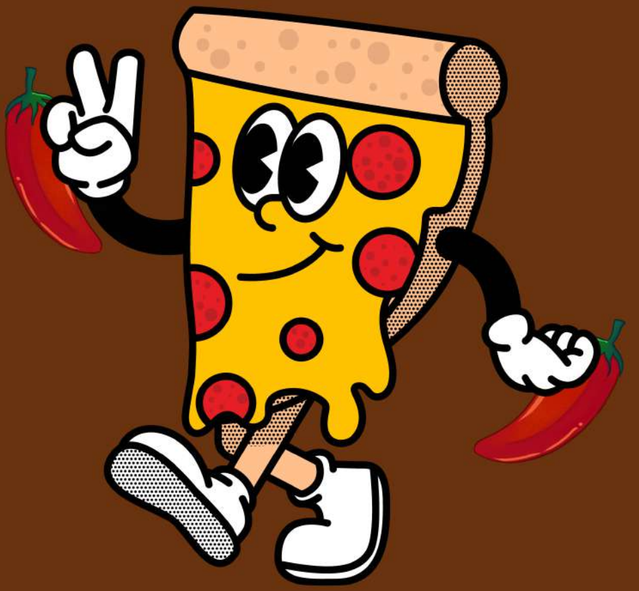
```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

```sql
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;
```

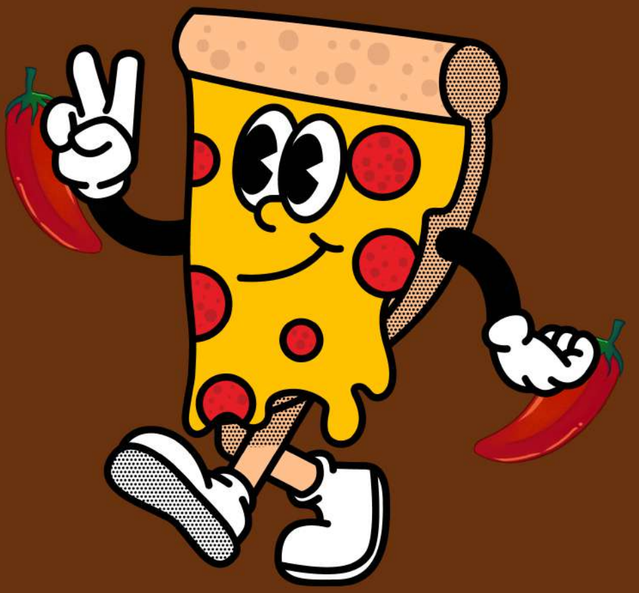


Result Grid | Filter Rows:

size	order_count
L	14329
M	11868
S	10897
XL	431
XXL	22

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

```sql
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;
```

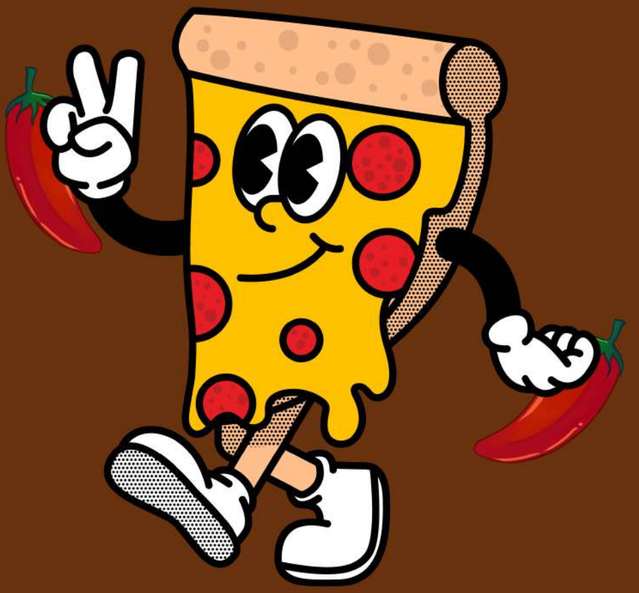


Result Grid | Filter Rows:

name	quantity
The Barbecue Chicken Pizza	1905
The Pepperoni Pizza	1886
The Classic Deluxe Pizza	1879
The Hawaiian Pizza	1843
The California Chicken Pizza	1818

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

```sql
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;
```



Result Grid | Filter

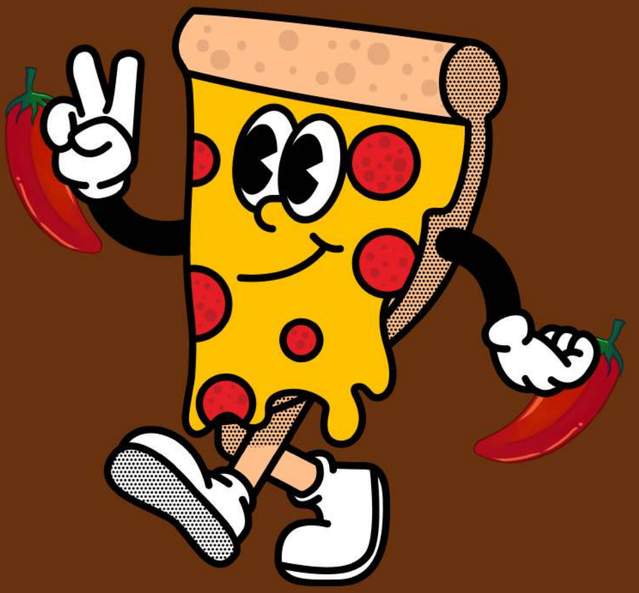
category	quantity
Classic	11514
Supreme	9201
Veggie	9061
Chicken	8504

Determine the distribution of orders by hour of the day.

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

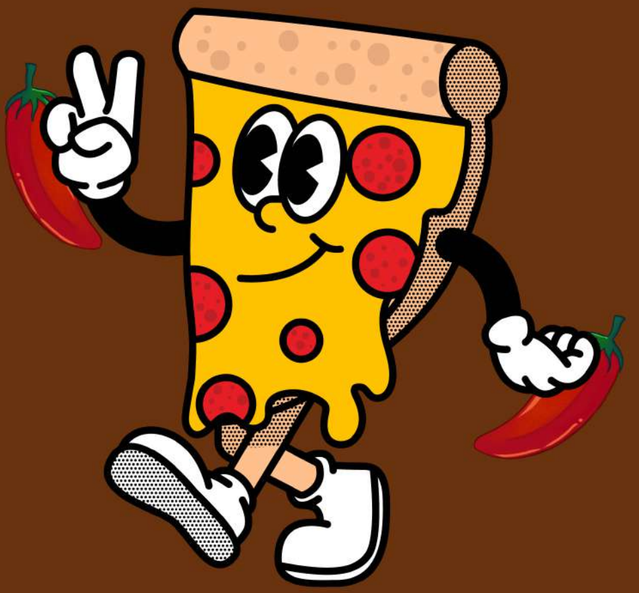
Result Grid | Filte

hour	order_count
12	2520
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



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

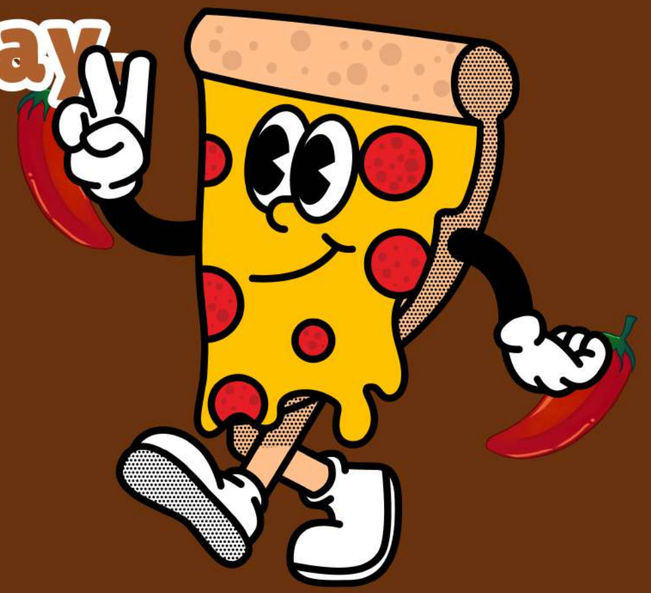
```sql
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```



Result Grid | Filter Ro

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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



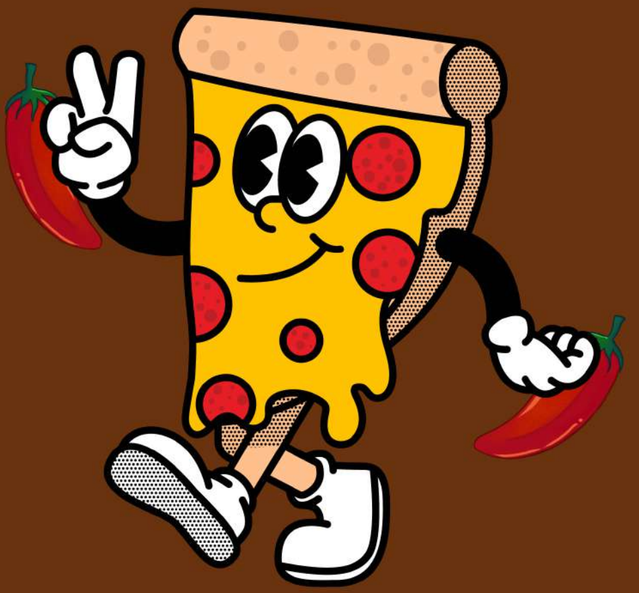
```sql
SELECT
    round(AVG(quantity),0)
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quanity;
```

Result Grid | Filter R

	round(AVG(quantity),0)
▶	138

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

```sql
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;
```

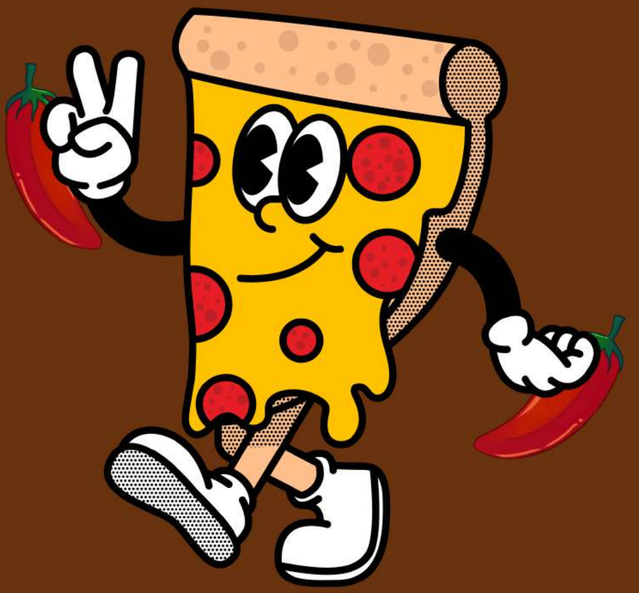


Result Grid | Filter Rows:

name	revenue
The Barbecue Chicken Pizza	33488.75
The Thai Chicken Pizza	32843
The California Chicken Pizza	31763.5

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

```sql
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;
```

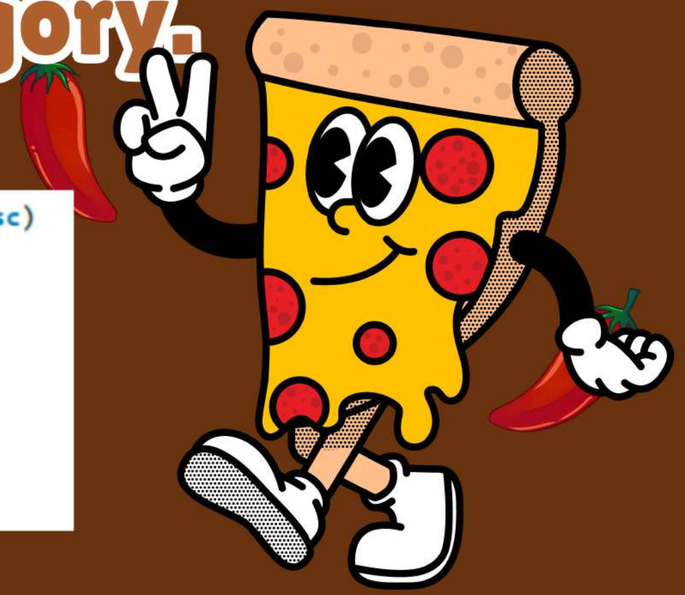


Result Grid | Filter Rows:

category	revenue
Classic	26.970469035082356
Supreme	25.330333904043673
Chicken	23.850322768975268
Veggie	23.848874291898472

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((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;
```



Result Grid | Filter Rows:

name	revenue
The Barbecue Chicken Pizza	33488.75
The Thai Chicken Pizza	32843
The California Chicken Pizza	31763.5
The Classic Deluxe Pizza	29211.5
The Hawaiian Pizza	24633.5
The Pepperoni Pizza	23542
The Spicy Italian Pizza	26938.75
The Italian Supreme Pizza	25921
The Sicilian Pizza	23444

Thank you for your time and attention! I hope this project provided valuable insights into how SQL can be used to analyze and optimize pizza sales. Your feedback and suggestions are greatly appreciated, and I look forward to any discussions or questions.