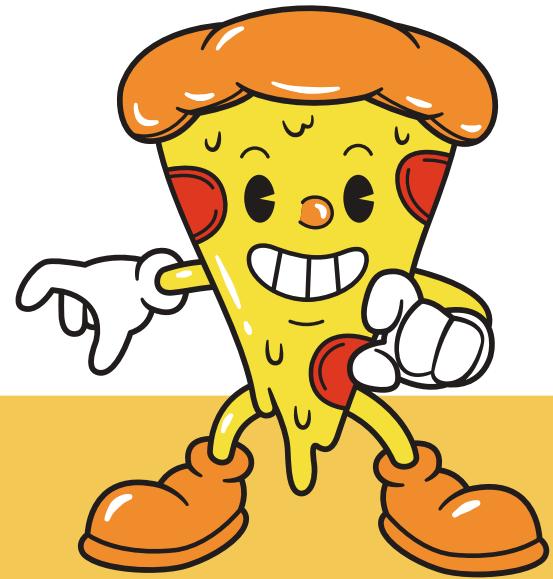


PIZZA SALES ANALYSIS

Using SQL

-BY ROHIT TOSKAR

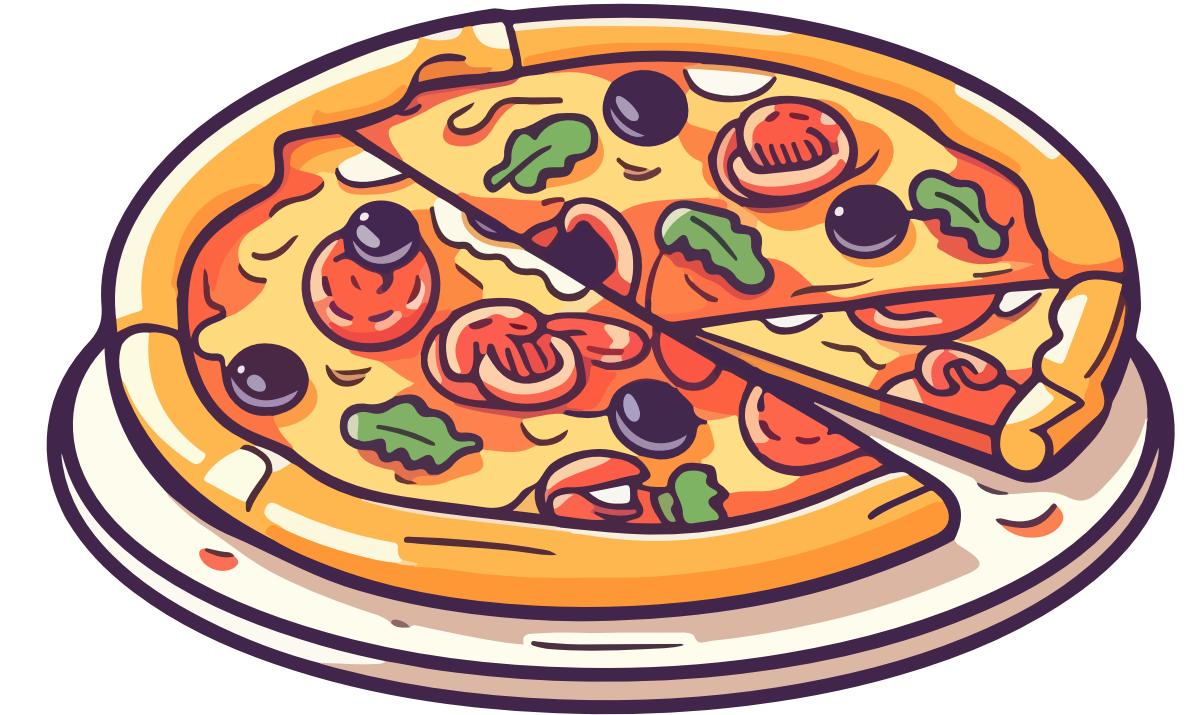


OBJECTIVES

The objective of the Pizza Sales Analysis Project is to use SQL to analyze pizza sales data. This includes evaluating sales performance, identifying popular and profitable pizzas, understanding customer ordering patterns, and improving operational efficiency. The goal is to provide insights and recommendations to help make better business decisions and boost overall performance.

Tools used :

- Database management system : SQL
- Querry tool : MySQL workbench



Methodology :



01

Database setup :
created database named “pizahut” and imported tables (order_details, orders, pizza_types, Pizzas) into the database.



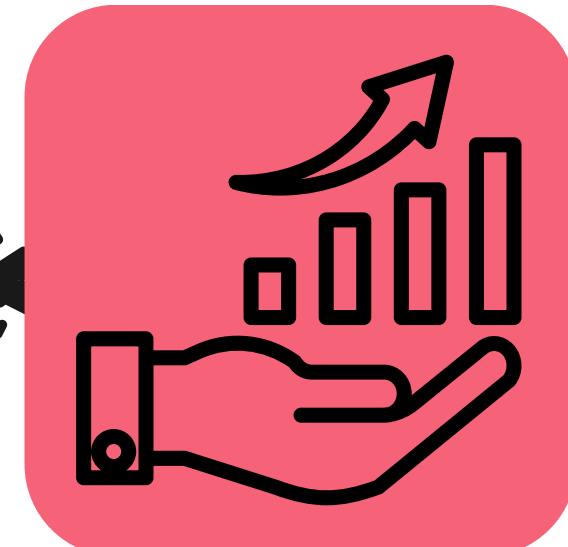
02

Data exploration and cleaning :
Explored structure and contents of each table to understand the relationships and data points. Ensured data integrity by checking for any inconsistencies



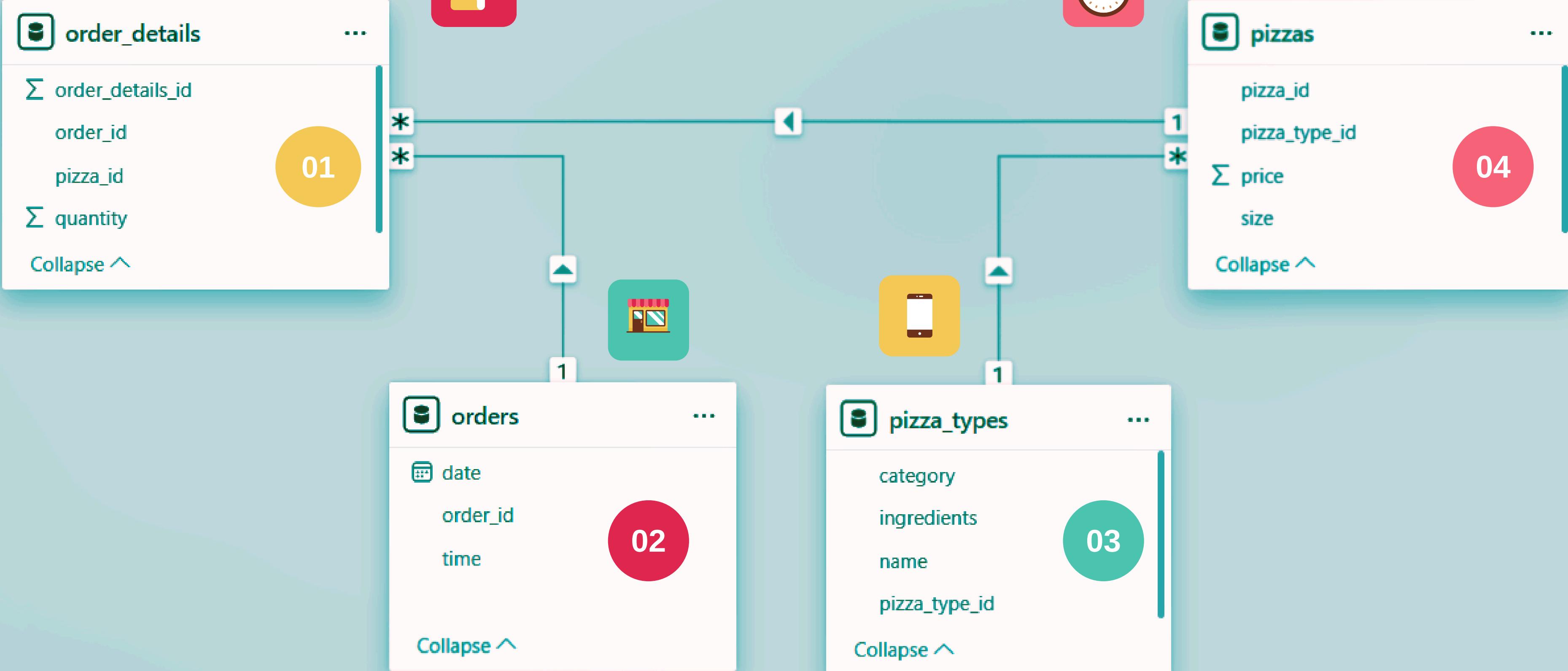
03

Querying and analysis :
SQL queries were written to extract meaningful insights from the data. These queries addressed various aspects of pizza sales including total orders, revenue generation, popular pizza type and order distribution etc.





PROJECT MODEL VIEW



KEY OBJECTIVES

1



BASIC ANALYSIS

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

2



INTERMEDIATE ANALYSIS

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

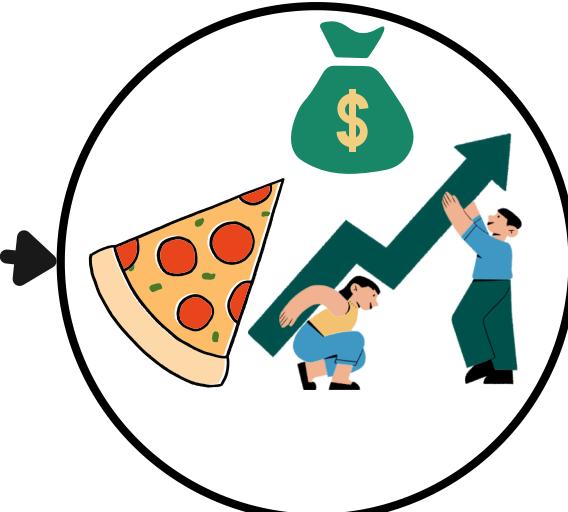
3



ADVANCED ANALYSIS

1. Calculate the percentage contribution of each pizza type to total revenue.
2. Analyze the cumulative revenue generated over time.
3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

4





Q1. Retrieve the total number of orders placed.

Query :

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

Output :

	total_orders
▶	21350



Q2. Calculate the total revenue generated from pizza sales.



Query :

```
SELECT  
    ROUND(SUM(od.quantity * p.price), 2) AS Total_revenue_generated  
FROM  
    order_details od  
    JOIN  
    pizzas p ON od.pizza_id = p.pizza_id
```

Output :

Total_revenue_generated
817860.05



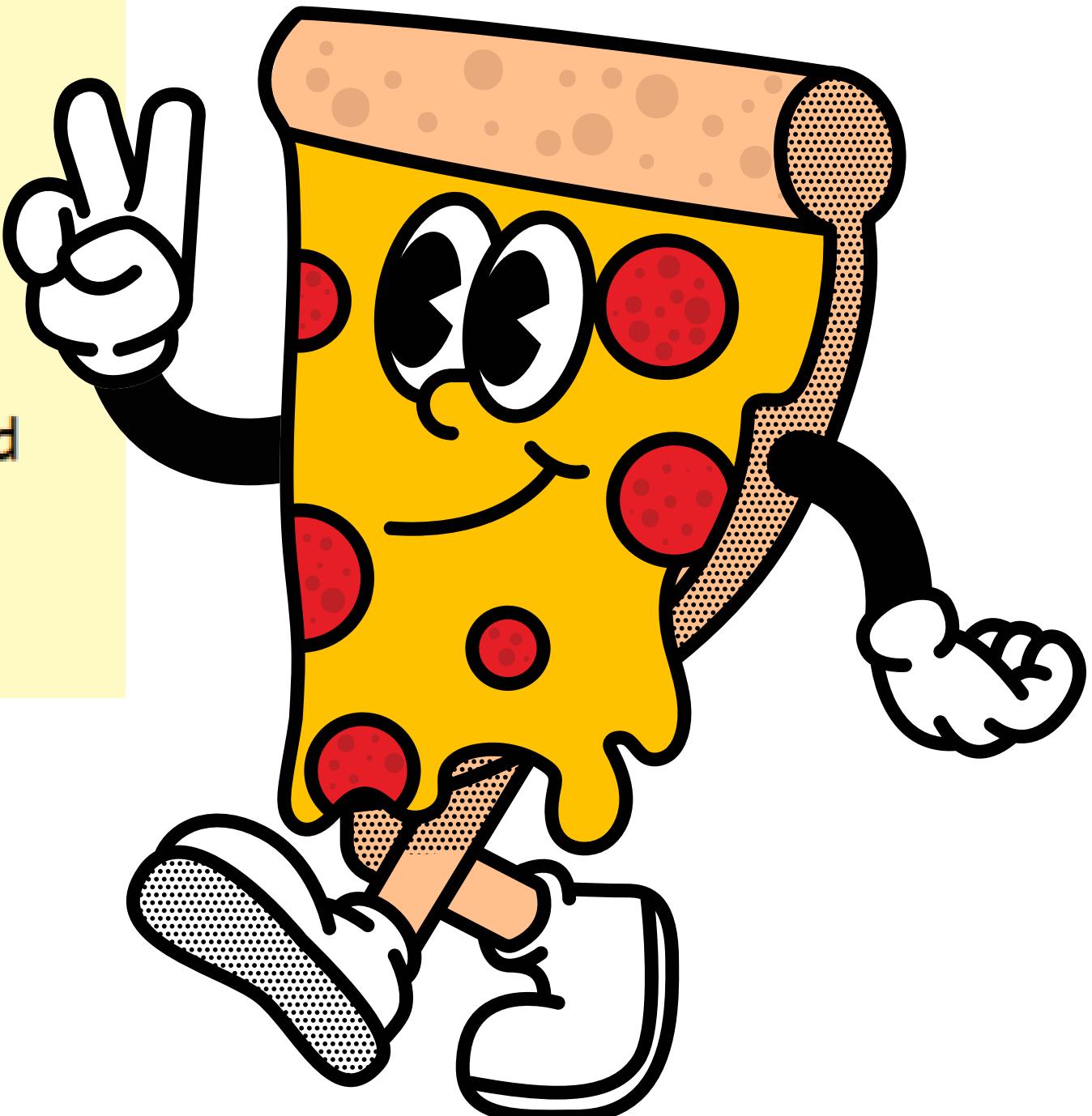
Q3. Identify the highest-priced pizza.

Query :

```
SELECT
    pt.name AS 'Name', p.price 'Price'
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```

Output :

Name	Price
The Greek Pizza	35.95



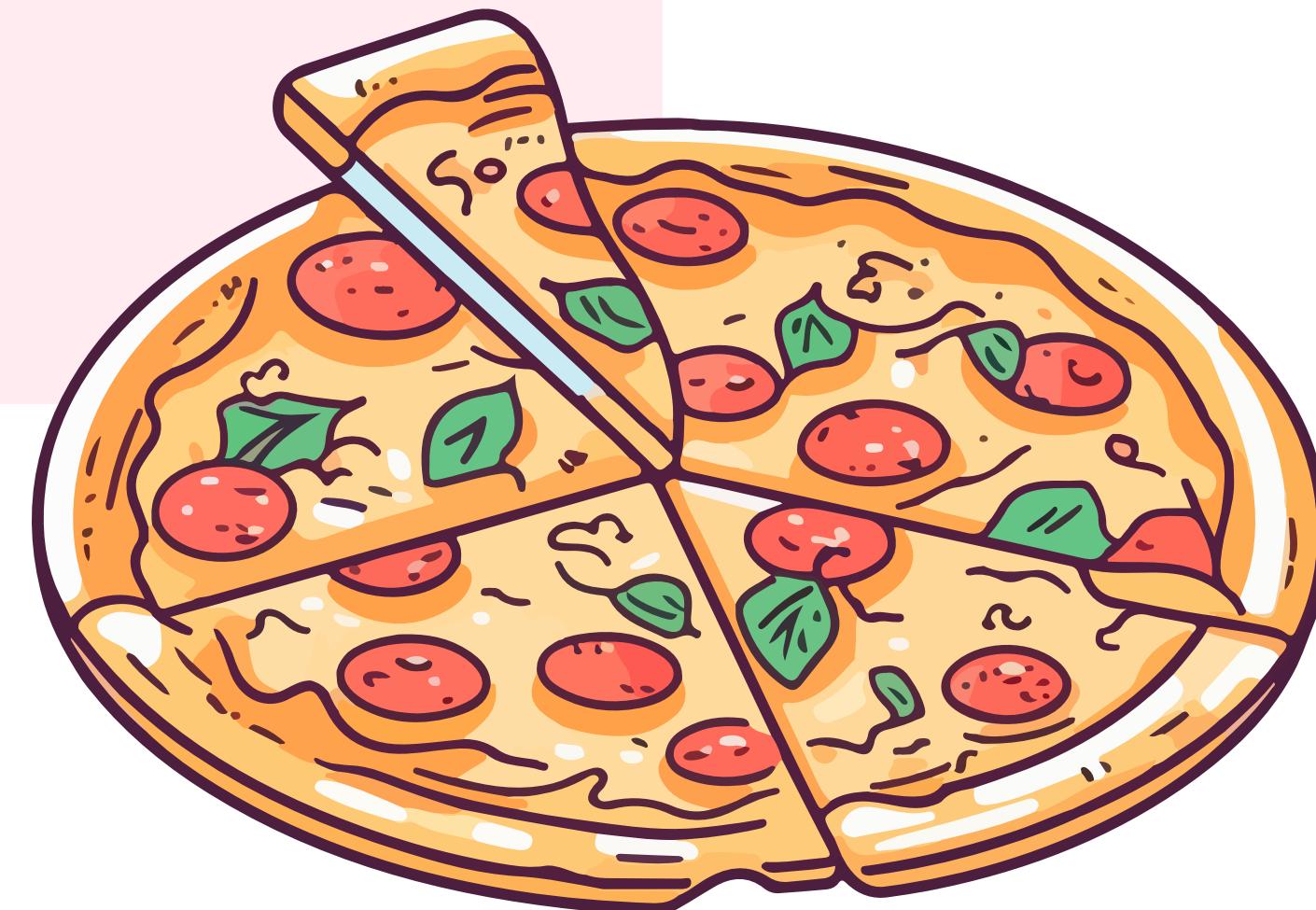
Q4. Identify the most common pizza size ordered.

Query :

```
SELECT
    p.size AS 'Size', COUNT(od.order_id) AS 'Quantity_ordered'
FROM
    order_details od
    JOIN
    pizzas p ON p.pizza_id = od.pizza_id
GROUP BY Size
ORDER BY Quantity_ordered DESC ;
```

Output :

Size	Quantity_ordered
L	18526
M	15385
S	14137
XL	544
XXL	28



S
M
L
XL
XXL

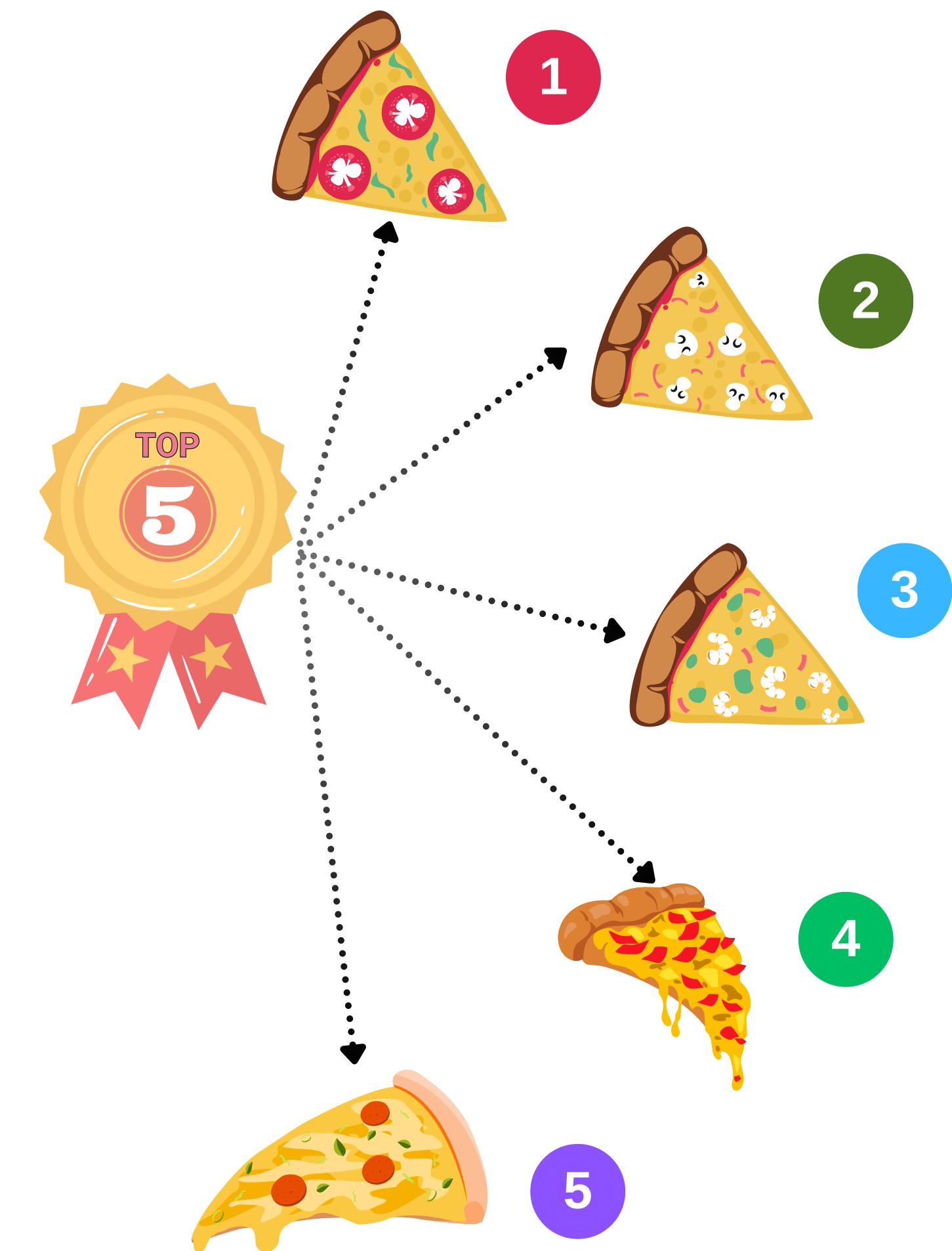
Q5. List the top 5 most ordered pizza types along with their quantities.

Query :

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

Output :

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



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

Query :

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



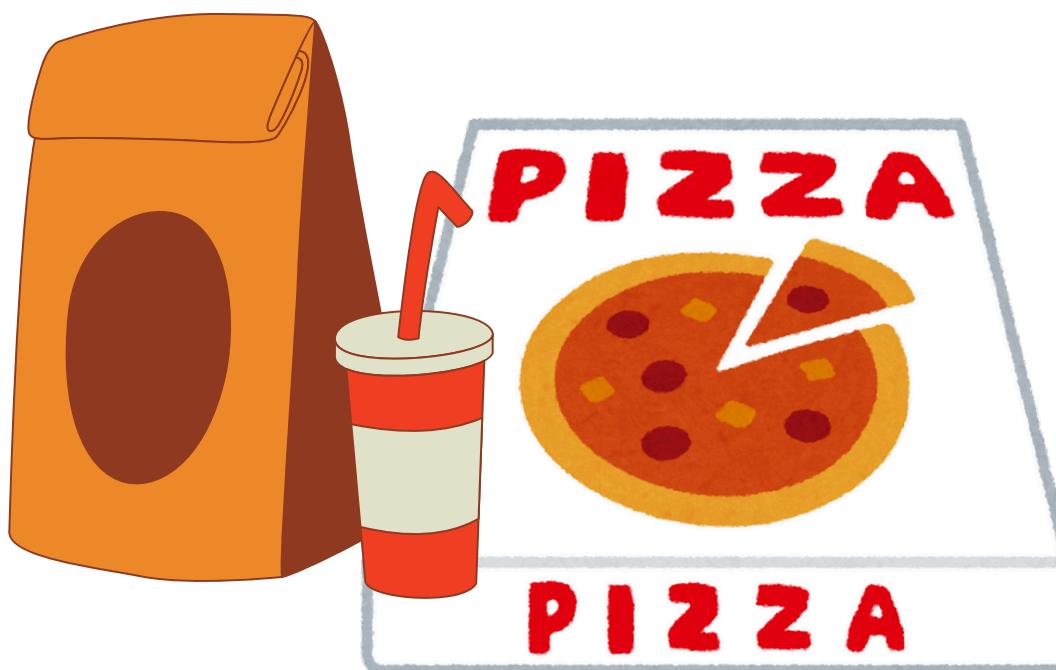
Output :

category	Total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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

Query :

```
SELECT  
    HOUR(o.order_time) AS 'Hour', COUNT(order_id) AS Order_count  
FROM  
    orders o  
GROUP BY HOUR(o.order_time);
```



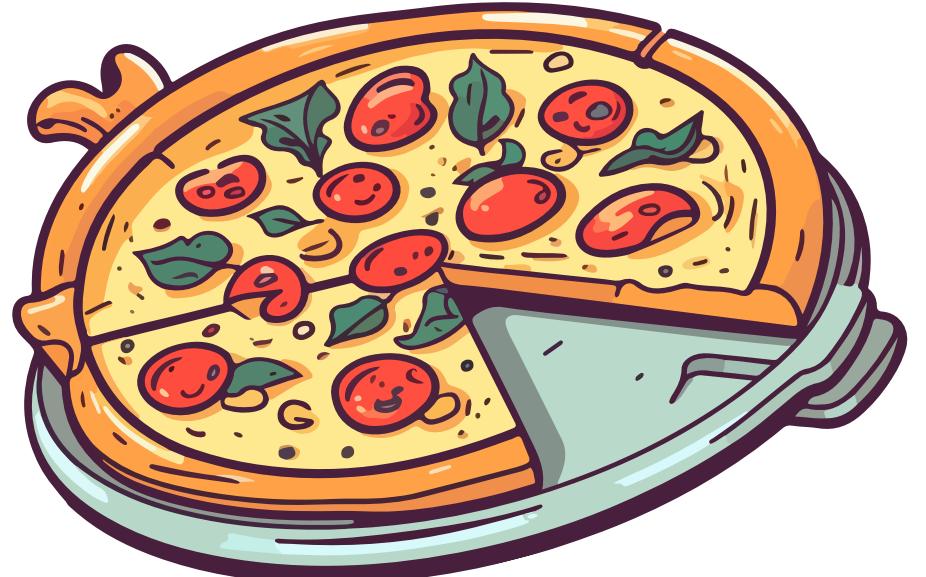
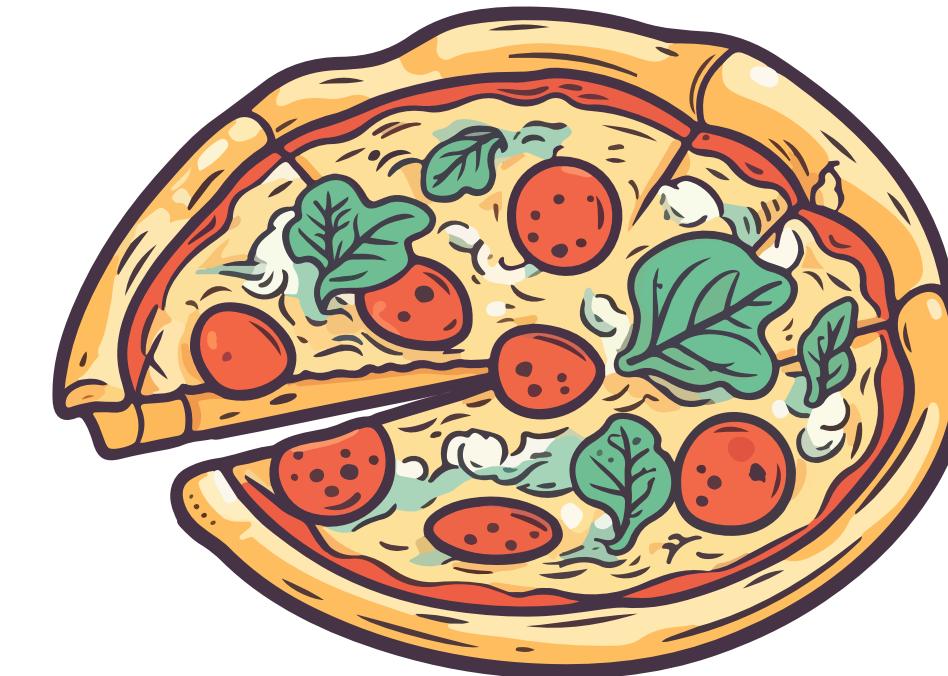
Output :

Hour	Order_count
11	1231
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

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

Query :

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



Output :

category	Pizza_types
Chicken	6
Classic	8
Supreme	9
Veggie	9



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

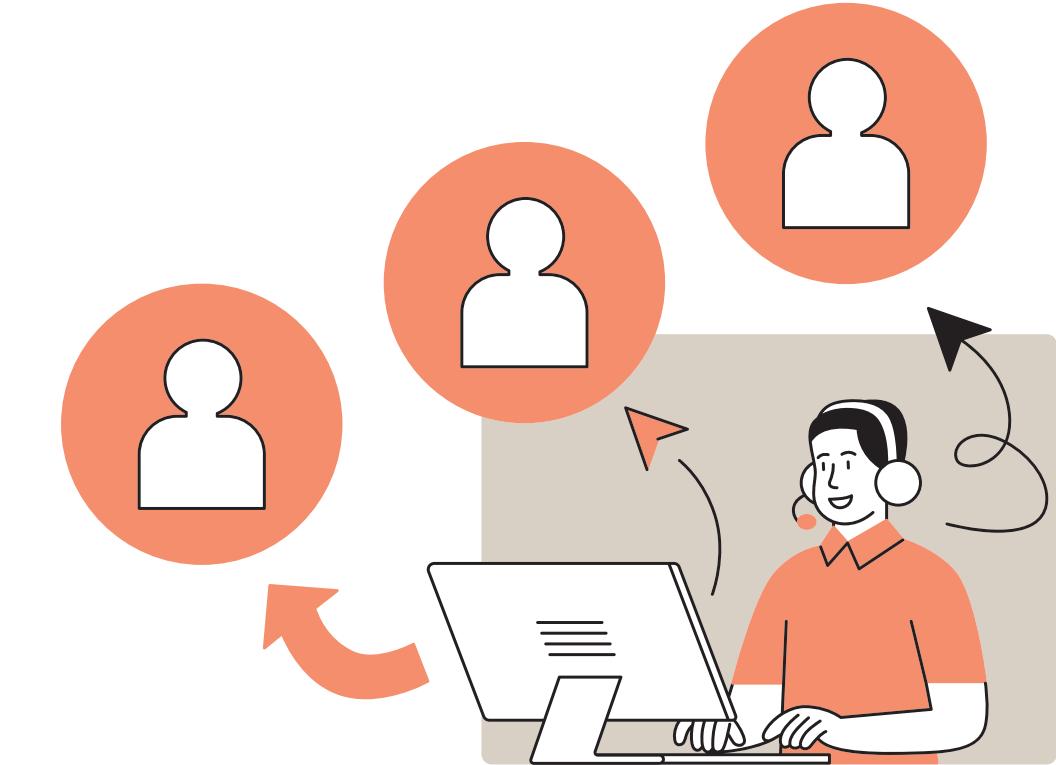
Query :

```
SELECT  
    ROUND(AVG(quantity), 0) as "Avg of pizzas ordered per day"  
FROM  
    (SELECT  
        o.order_date, SUM(od.quantity) AS quantity  
    FROM  
        orders o  
    JOIN order_details od ON od.order_id = o.order_id  
    GROUP BY o.order_date) AS order_qty;
```



Output :

Avg of pizzas ordered per day
138



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

Query :

```
SELECT
    pt.name, ROUND(SUM(od.quantity * p.price), 0) AS revenue
FROM
    pizzas p
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY revenue DESC
LIMIT 3;
```



Output :

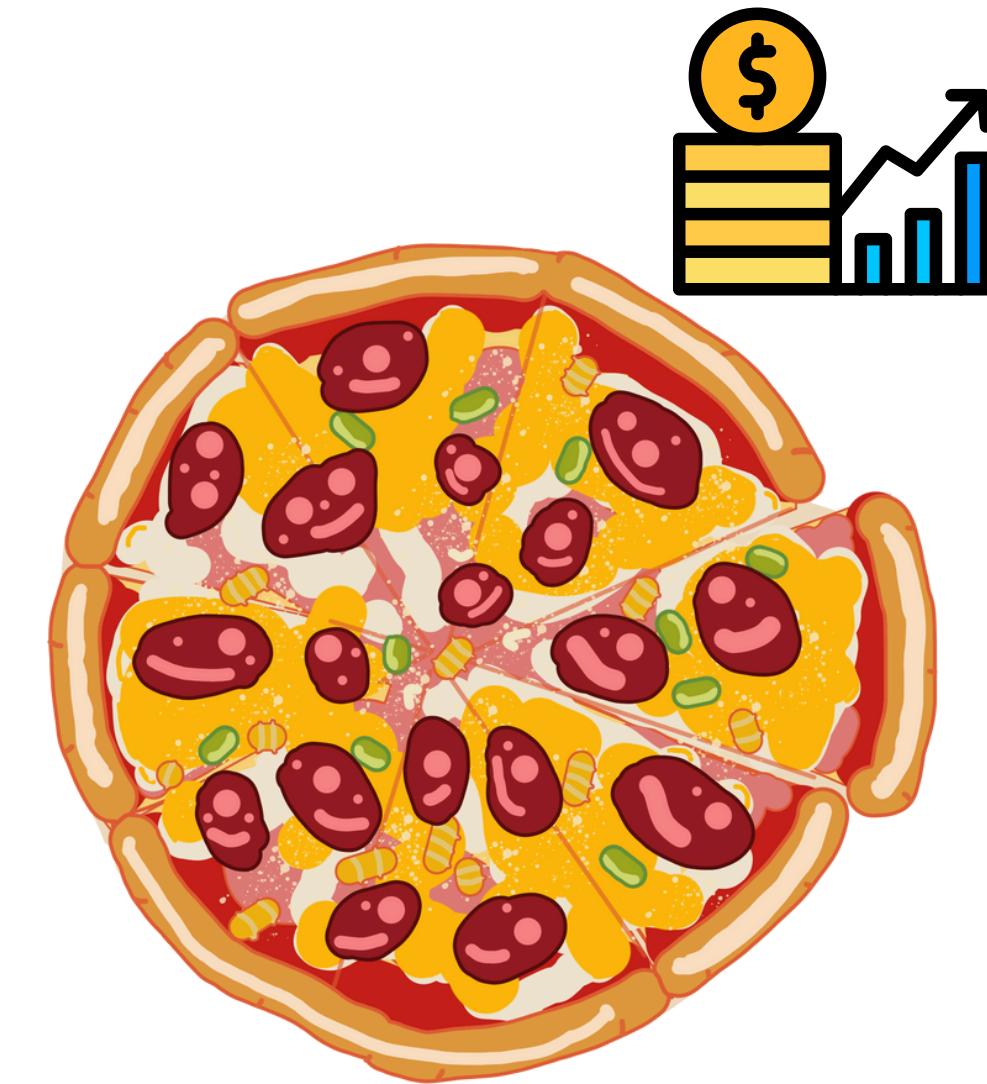
name	revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410



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

Query :

```
SELECT
    pt.category,
    CONCAT(ROUND(SUM(od.quantity * p.price) / (SELECT
                                                SUM(od.quantity * p.price) AS Total_revenue
                                            FROM
                                                pizzas p
                                            JOIN
                                                order_details od ON p.pizza_id = od.pizza_id) * 10
                                            2),
    '%') AS Total_revenue
FROM
    pizzas p
    JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
    JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY Total_revenue DESC;
```



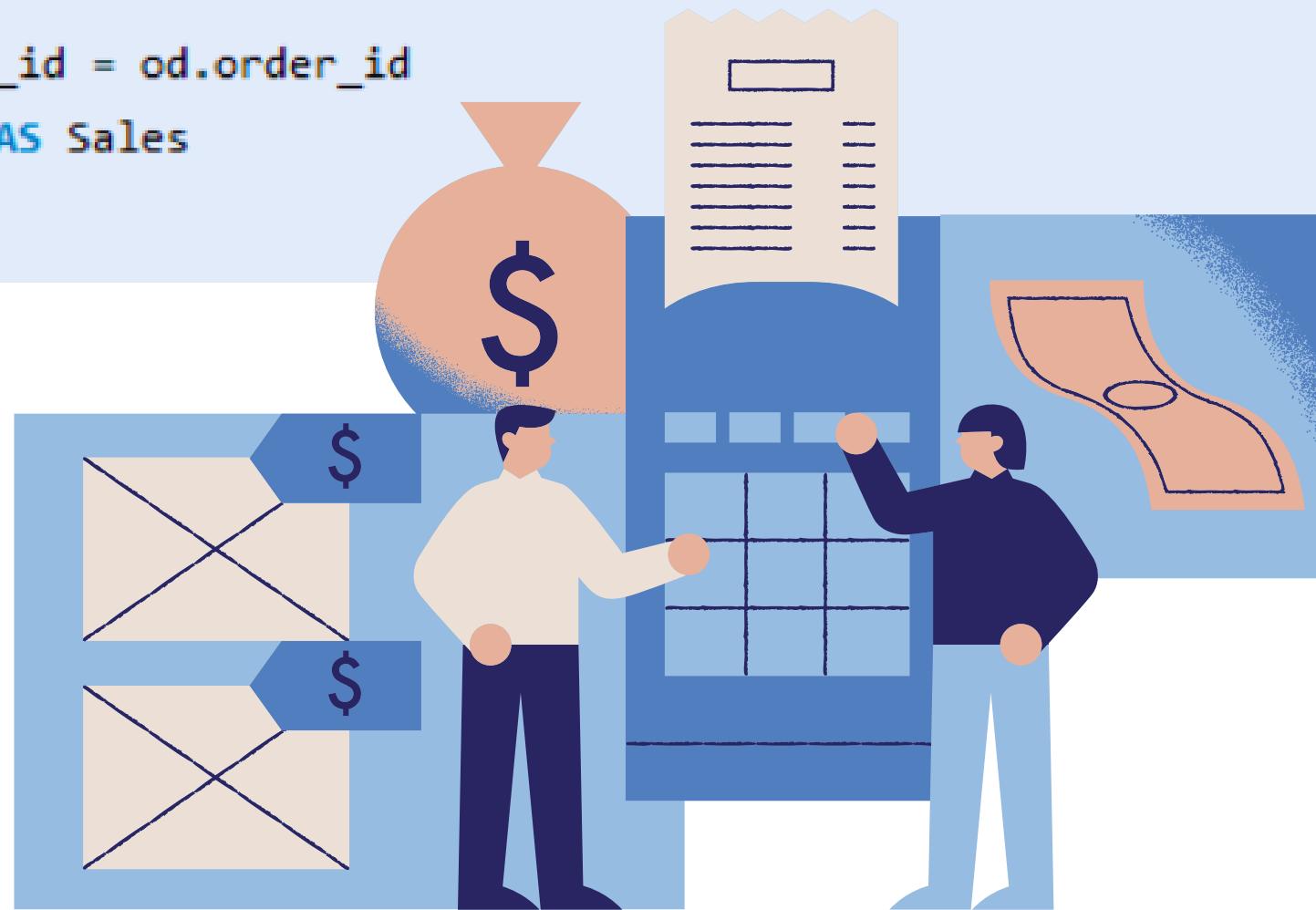
Output :

category	Total_revenue
Classic	26.91%
Supreme	25.46%
Chicken	23.96%
Veggie	23.68%

Q12. Analyze the cumulative revenue generated over time.

Query :

```
select order_date, round(sum(revenue) over (order by order_date),2) as Cum_revenue from
(SELECT
    o.order_date, SUM(p.price * od.quantity) AS Revenue
    FROM
        pizzas p
        JOIN
        order_details od ON p.pizza_id = od.pizza_id
        JOIN
        orders o ON o.order_id = od.order_id
    GROUP BY o.order_date) AS Sales
```



Output :

order_date	Cum_revenue
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.35
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3
2015-01-14	32358.7
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6
2015-01-19	43365.75
2015-01-20	45763.65
2015-01-21	47804.2
2015-01-22	50300.9
2015-01-23	52724.6

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

Query :

```
select category, name, Revenue from
(select category, name, Revenue,
rank() over(partition by category order by Revenue Desc) as Rank_No from
(SELECT
    pt.category, pt.name, SUM(p.price * od.quantity) AS Revenue
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category , pt.name) as A) as B
where Rank_no <= 3;
```

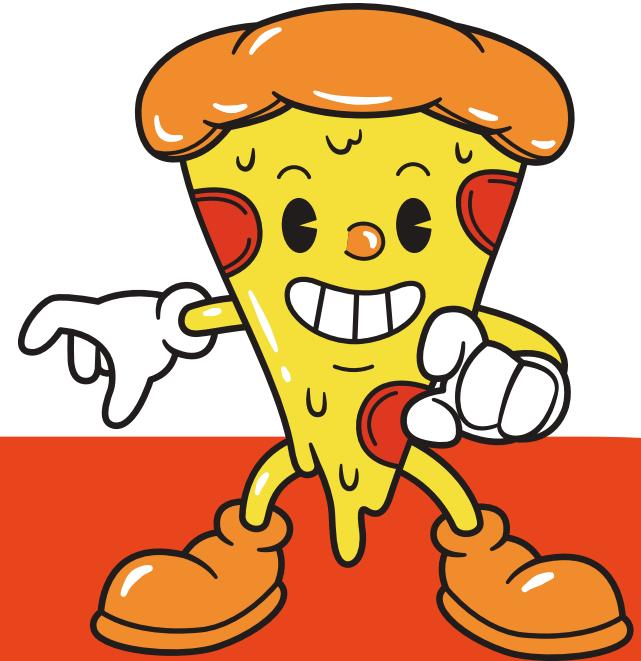


Output :

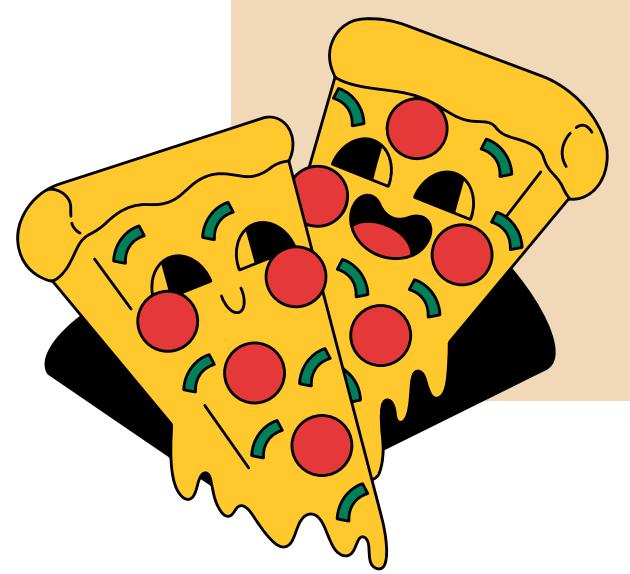
category	name	Revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	25066.5



CONCLUSION



The objective of the Pizza Sales Analysis Project is to use SQL to analyze pizza sales data. This includes evaluating sales performance, identifying popular and profitable pizzas, understanding customer ordering patterns, and improving operational efficiency. The goal is to provide insights and recommendations to help make better business decisions and boost overall performance.



THANK YOU!

