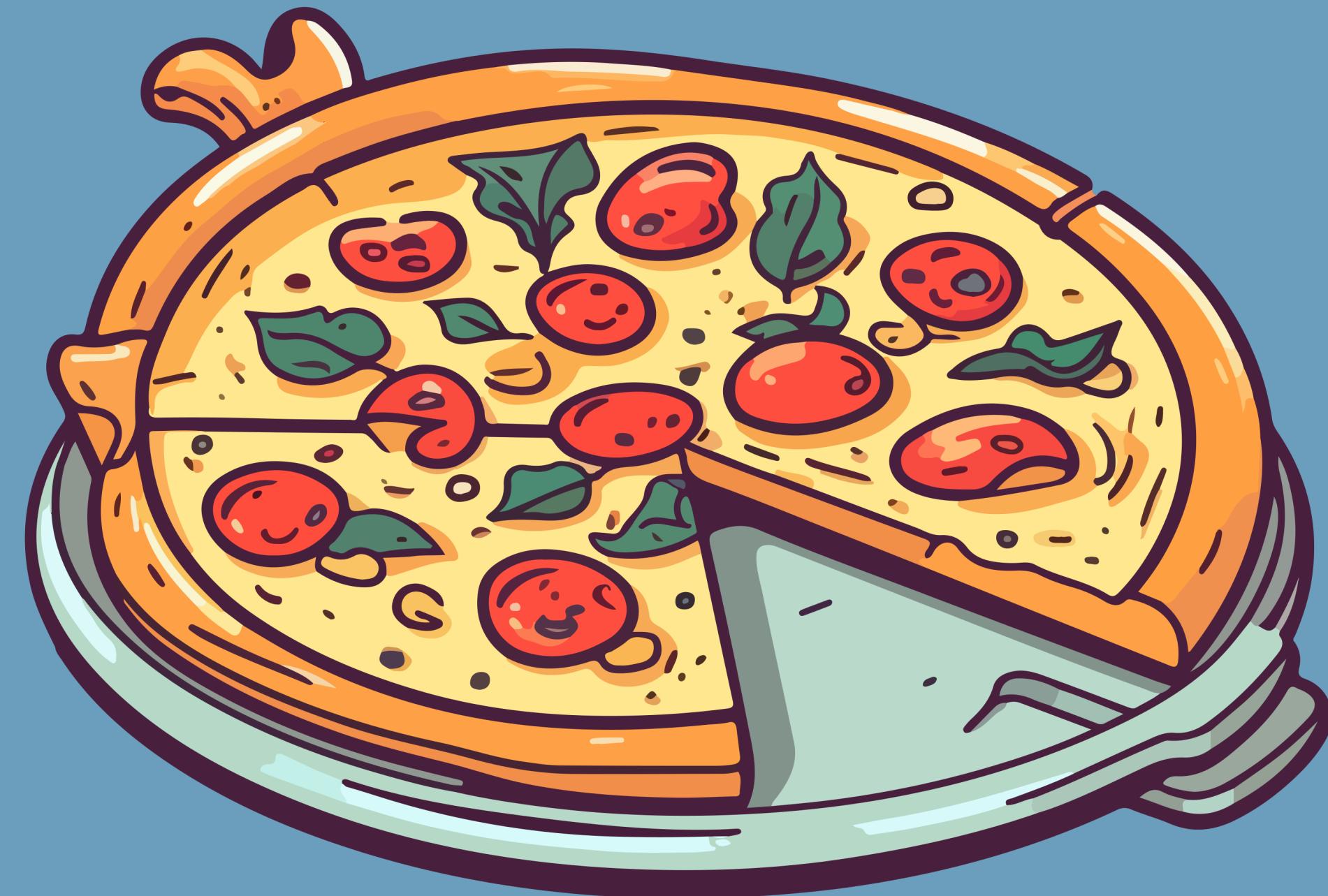
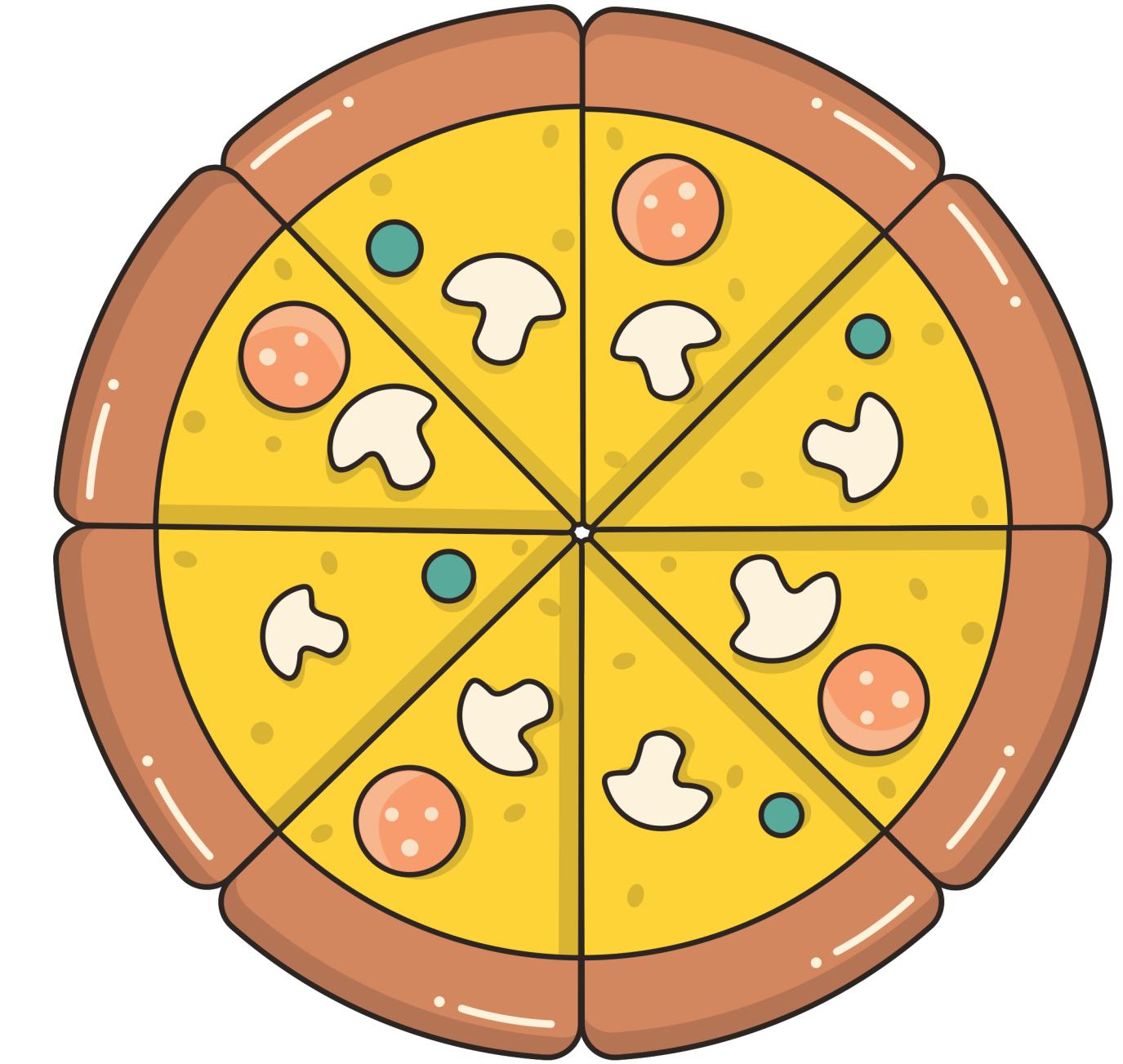


PIZZA STORE SALES ANALYSIS



INTRODUCTION

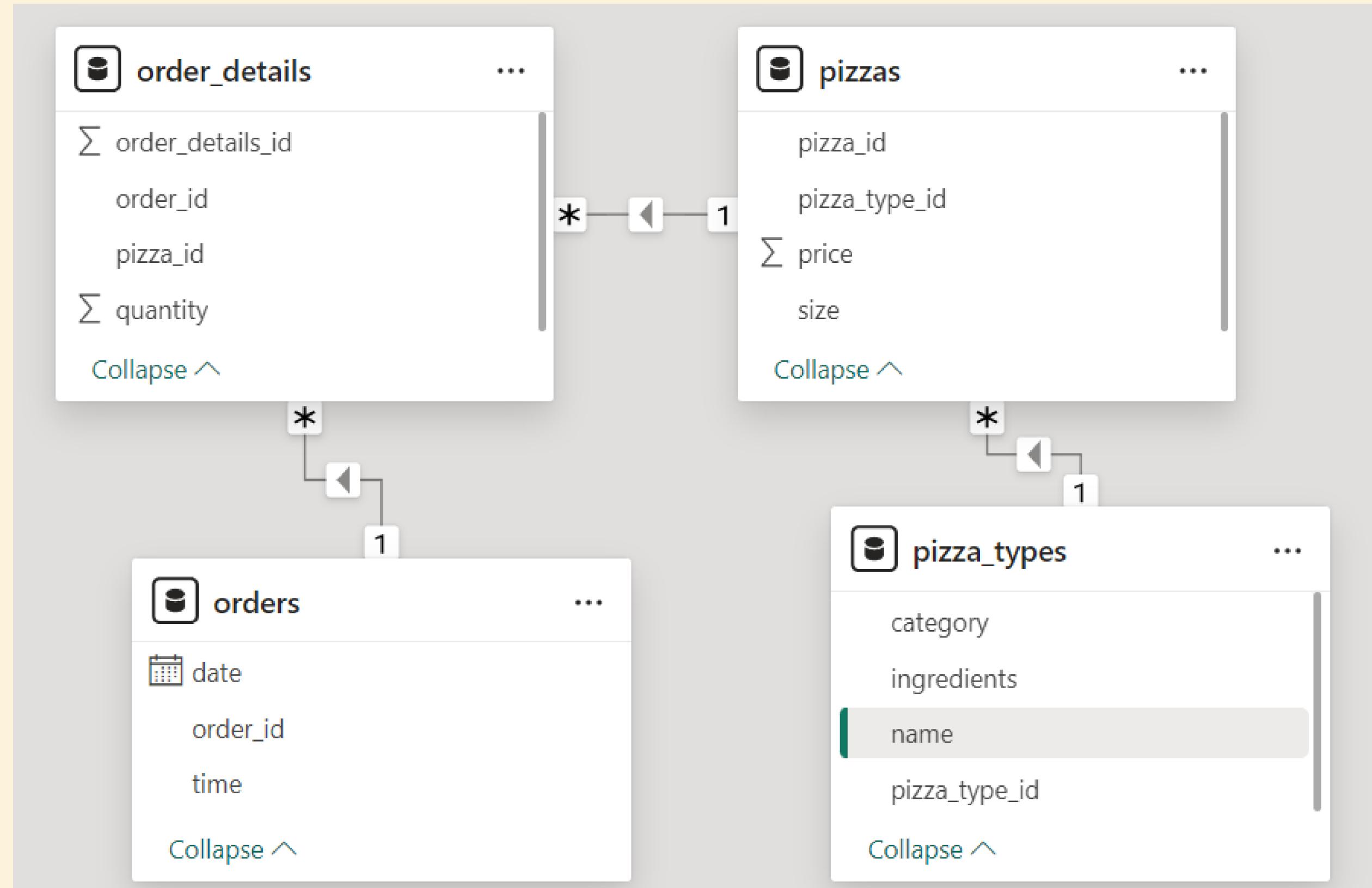
In this project, we dive into the world of pizza sales by analyzing data from a pizza store. With the power of SQL, we aim to answer various advanced business questions. Our dataset comprises four tables: pizzas, pizza types, orders, and order details. Through this analysis, we seek insights into customer preferences, popular pizza types, sales trends, and more, helping the pizza store optimize its operations and boost profitability.

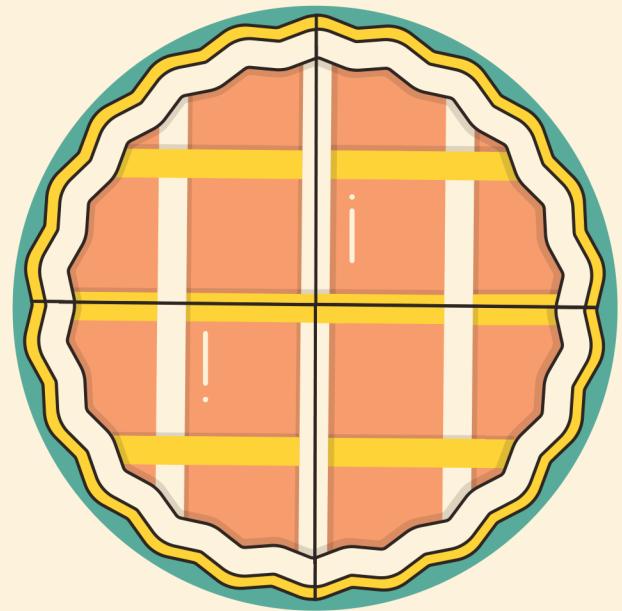


DATASET

- **Orders Table:** Contains unique order identifiers, along with the date and time of each order.
- **Order Details Table:** Includes details of each order, such as the specific pizza ordered (referenced by pizza_id) and the quantity.
- **Pizza Table:** Provides information on each pizza, including its unique identifier, type, size, and price.
- **Pizza Types Table:** Stores details about pizza types, including their unique identifiers, names, categories, and ingredients.

Entity Relationship Diagram





QUESTION 1

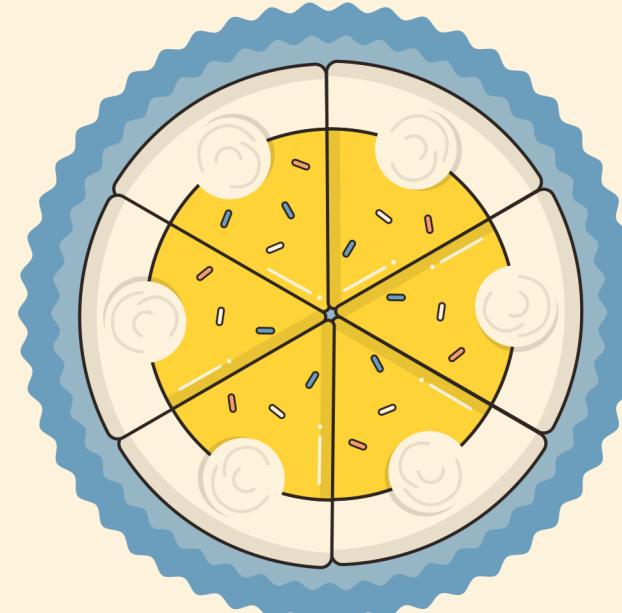
Retrieve the total no. of orders placed.

```
SELECT  
    COUNT(order_id) AS Total_Orders  
FROM  
    orders;
```

QUERY

	Total_Orders
▶	21350

RESULT



QUESTION 2

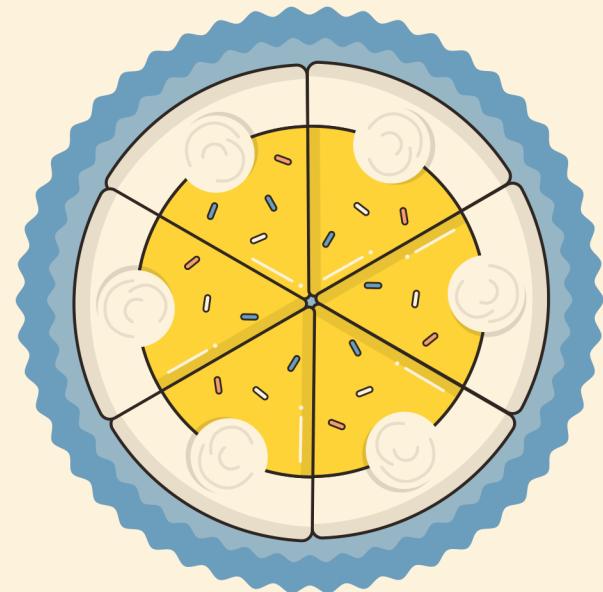
Calculate total revenue generated from pizza sales.

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

	Total_Sales
▶	817860.05

QUERY

RESULT



QUESTION 3

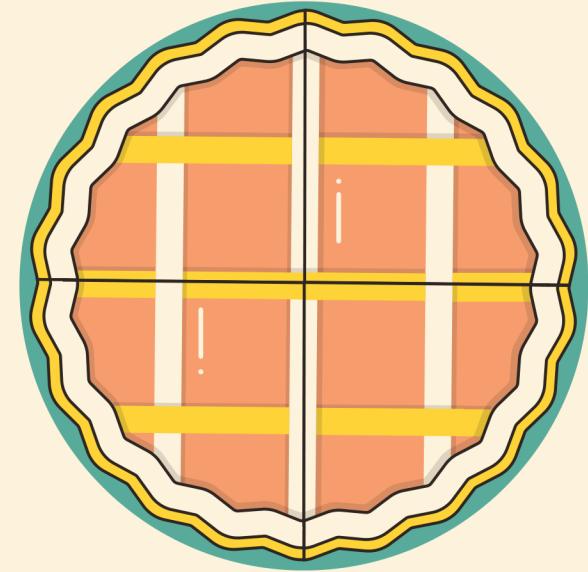
Identify the highest-priced pizza.

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

	Name	price
▶	The Greek Pizza	35.95

RESULT

QUERY



QUESTION 4

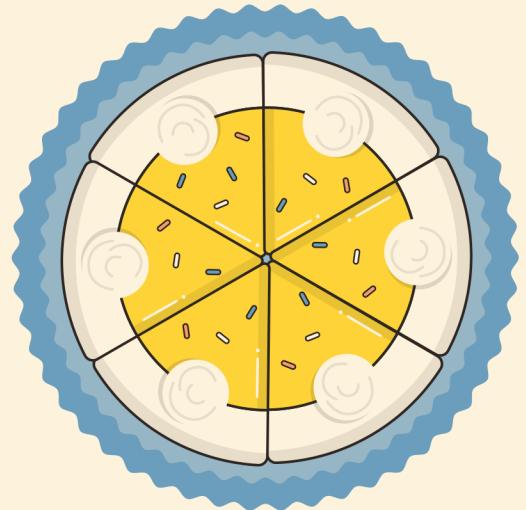
Identify the most common pizza size ordered

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

QUERY

	Size	Total_Ordered
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

RESULT



QUESTION 5

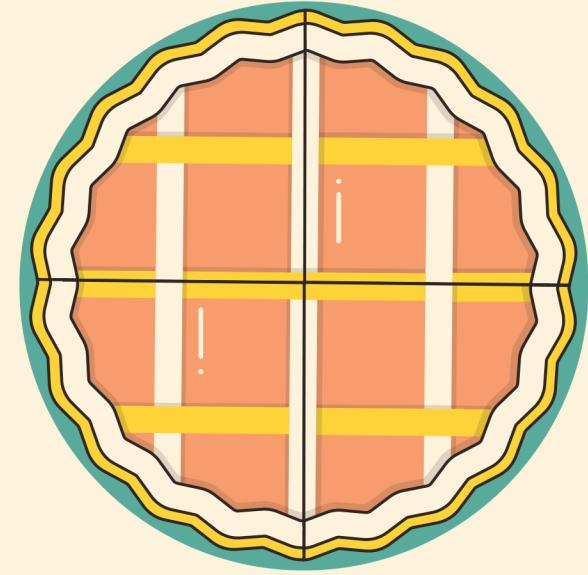
List the top 5 most ordered pizza types along with their quantities.

```
SELECT
    pt.name AS Name, SUM(quantity) AS Quantity
FROM
    pizza_types pt
        INNER JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        INNER JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY 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

RESULT

QUERY



QUESTION 6

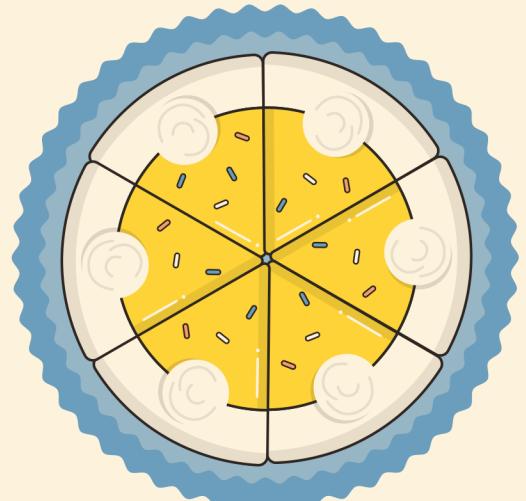
Join the necessary tables to find the total quantity of each pizza category ordered

```
SELECT
    pt.Category, SUM(Quantity) AS Total_Quantity
FROM
    pizza_types pt
        INNER JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        INNER JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.Category
ORDER BY Total_Quantity DESC;
```

	Category	Total_Quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

RESULT

QUERY



QUESTION 7

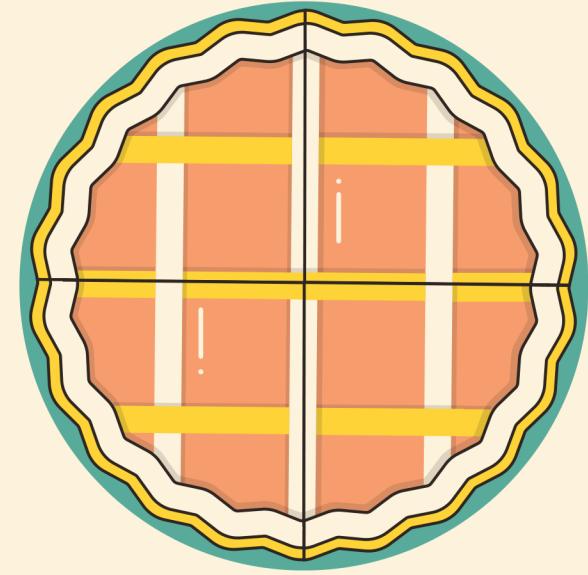
Determine the distribution of orders by hour of the day.

```
SELECT  
    HOUR(order_time) AS Hour, COUNT(order_id) AS Order_Count  
FROM  
    orders  
GROUP BY Hour  
ORDER BY Order_Count DESC;
```

QUERY

	Hour	Order_Count
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642
	14	1472
	15	1468
	11	1231
	21	1198
	22	663
	23	28
	10	8
	9	1

RESULT



QUESTION 8

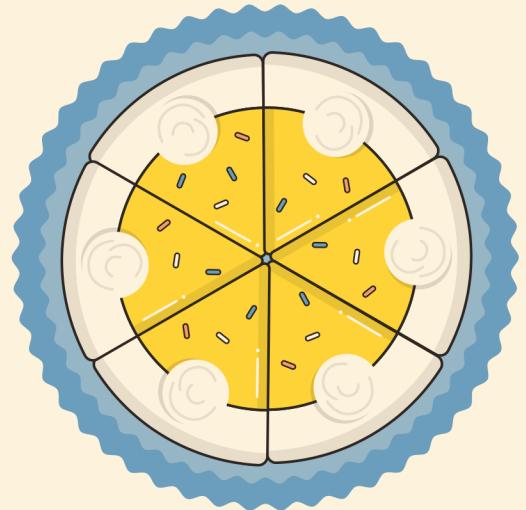
Join relevant tables to find the category-wise distribution of pizzas.

```
select Category, count(name) as Pizza_Count  
from pizza_types  
group by Category  
order by Pizza_Count desc;
```

QUERY

	Category	Pizza_Count
▶	Supreme	9
	Veggie	9
	Classic	8
	Chicken	6

RESULT



QUESTION 9

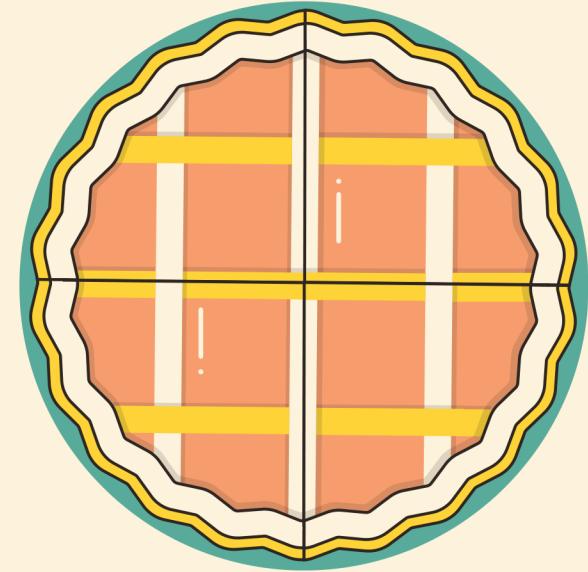
Group the orders by date and calculate the average number of pizzas ordered per day

```
SELECT  
    ROUND(AVG(quantity), 2) AS Avg_pizza_Order_per_day  
FROM  
    (SELECT  
        o.order_date, SUM(od.quantity) AS quantity  
    FROM  
        orders o  
    INNER JOIN order_details od ON o.order_id = od.order_id  
    GROUP BY o.order_date) AS order_quantity;
```

	Avg_pizza_Order_per_day
▶	138.47

RESULT

QUERY



QUESTION 10

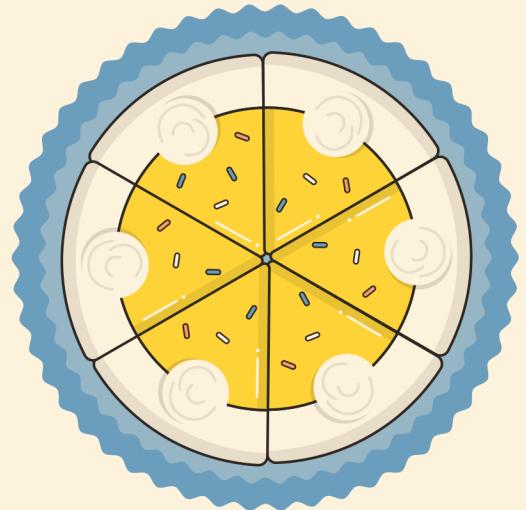
Determine the top 3 most ordered pizza types based on revenue

```
SELECT
    pt.name AS Name, SUM(od.quantity * p.price) 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 Name
ORDER BY Revenue DESC
LIMIT 3;
```

QUERY

	Name	Revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

RESULT



QUESTION 11

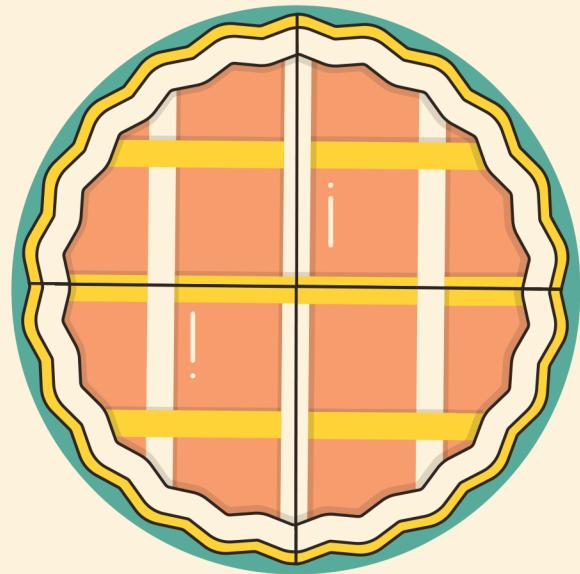
Calculate the percentage contribution of each pizza type to total revenue

```
SELECT
    pt.category AS Pizza_Category,
    ROUND(SUM(od.quantity * p.price) / (SELECT
                                            ROUND(SUM(od.quantity * p.price), 2) AS Total_Sales
                                         FROM
                                             order_details od
                                             INNER JOIN
                                             pizzas p ON od.pizza_id = p.pizza_id) * 100,
          2) 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 Pizza_Category
ORDER BY Revenue DESC;
```

	Pizza_Category	Revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

RESULT

QUERY



QUESTION 12

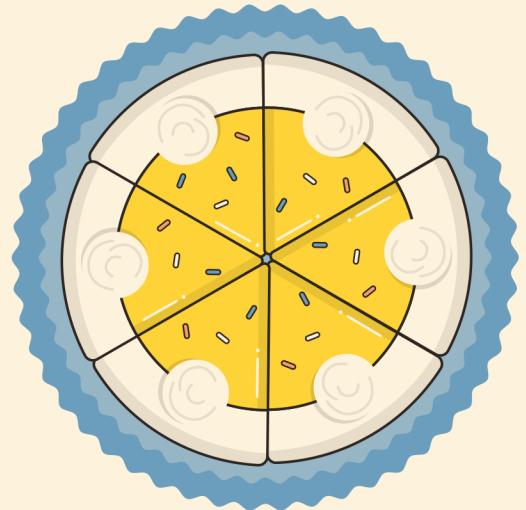
Analyze the cumulative revenue generated over time.

```
SELECT Order_Date,  
       round(sum(revenue) OVER (ORDER BY order_date),2) AS Cum_Revenue  
  FROM  
    (SELECT  
        o.order_date, SUM(od.quantity * p.price) AS revenue  
     FROM  
        order_details AS od  
        JOIN  
        orders o ON od.order_id = o.order_id  
        JOIN  
        pizzas p ON od.pizza_id = p.pizza_id  
    GROUP BY o.order_date) AS Sales;
```

QUERY

RESULT

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



QUESTION 13

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

```
select rn as C_Rank, Category, Name, Revenue
from
(select Category, Name, Revenue,
rank() over ( partition by Category order by revenue desc) as rn
from
(
select pt.category as Category, pt.name as Name, SUM(od.quantity * p.price) AS revenue
FROM
order_details AS od
JOIN
orders o ON od.order_id = o.order_id
JOIN
pizzas p ON od.pizza_id = p.pizza_id
join
pizza_types pt on p.pizza_type_id = pt.pizza_type_id
group by Category, Name
) as a
) as b
where rn <= 3;
```

	C_Rank	Category	Name	Revenue
	1	Chicken	The Thai Chicken Pizza	43434.25
	2	Chicken	The Barbecue Chicken Pizza	42768
	3	Chicken	The California Chicken Pizza	41409.5
	1	Classic	The Classic Deluxe Pizza	38180.5
	2	Classic	The Hawaiian Pizza	32273.25
	3	Classic	The Pepperoni Pizza	30161.75
	1	Supreme	The Spicy Italian Pizza	34831.25
	2	Supreme	The Italian Supreme Pizza	33476.75
	3	Supreme	The Sicilian Pizza	30940.5
	1	Veggie	The Four Cheese Pizza	32265.70000000065
	2	Veggie	The Mexicana Pizza	26780.75
	3	Veggie	The Five Cheese Pizza	26066.5

RESULT

QUERY

THE END