

Pizza Sales Analysis

SQL Project

By Sanjana John





This presentation provides a comprehensive analysis of the data from the Pizza Ordering System which has been extracted, manipulated and analyzed using MySQL.

Our database “**pizzahut**” consists of four interconnected tables: pizzas, pizza_types, orders and order_details.

Queries had been designed and executed to :

- Aggregate data for meaningful insights
- Calculate key metrics such as revenue and order quantities
- Identify trends and patterns in pizza orders

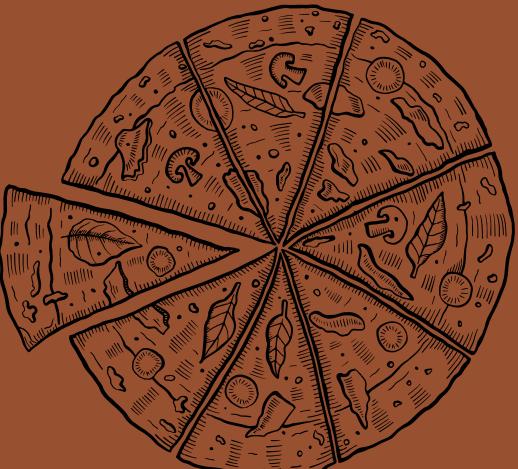


Table Relationships

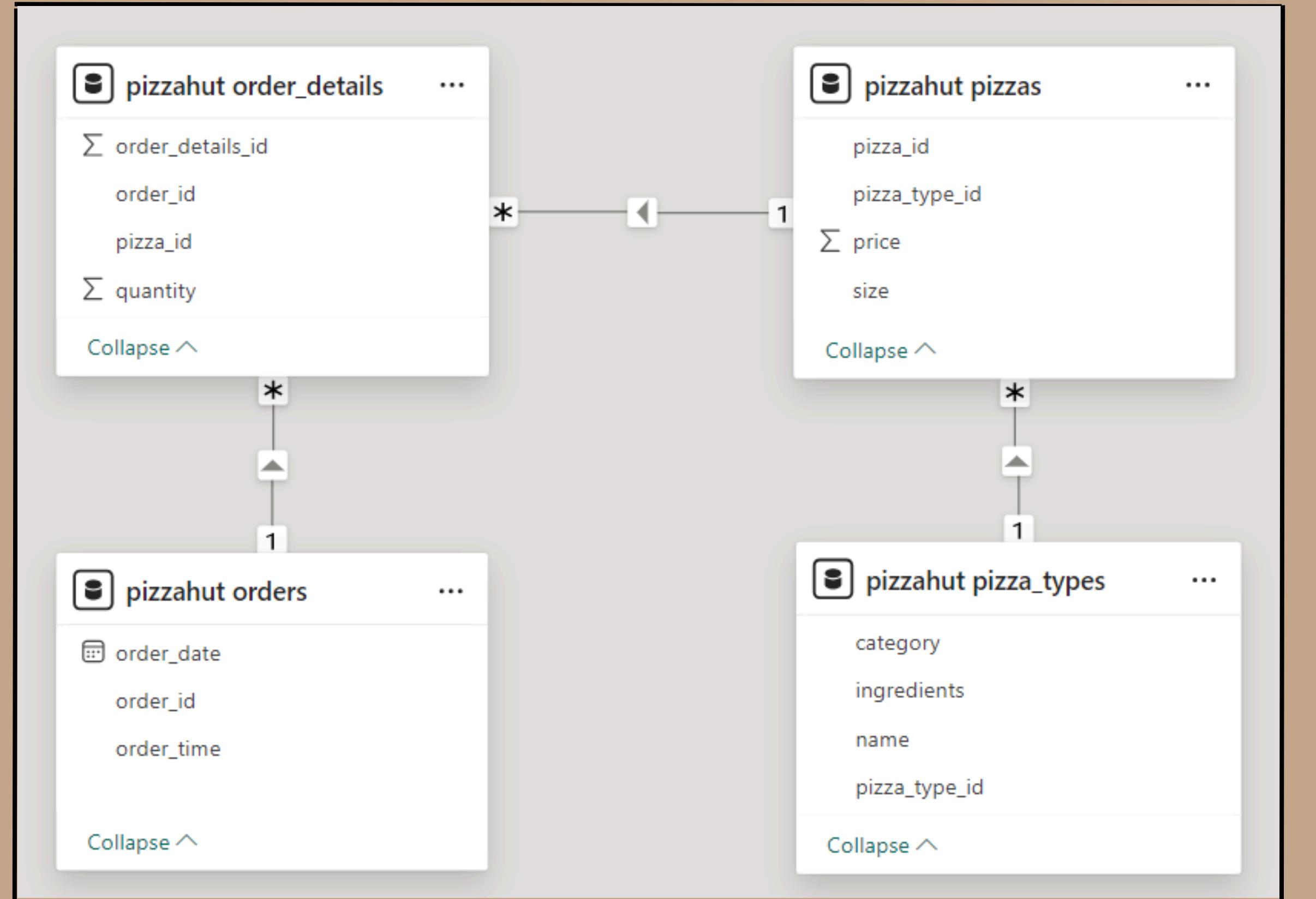
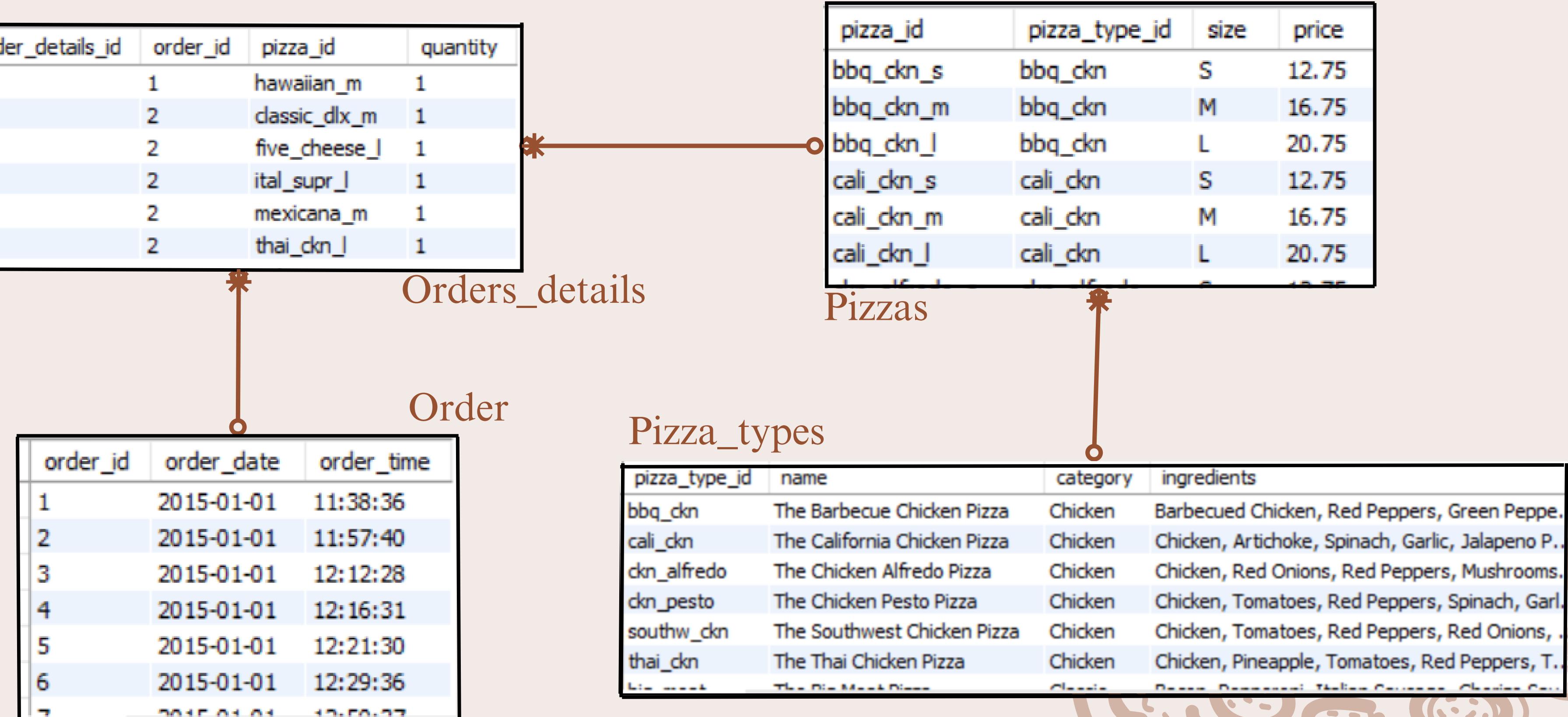


Table Details & Relationship



Queries

- 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.
- ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.
- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

Retrieve total numbers of orders placed

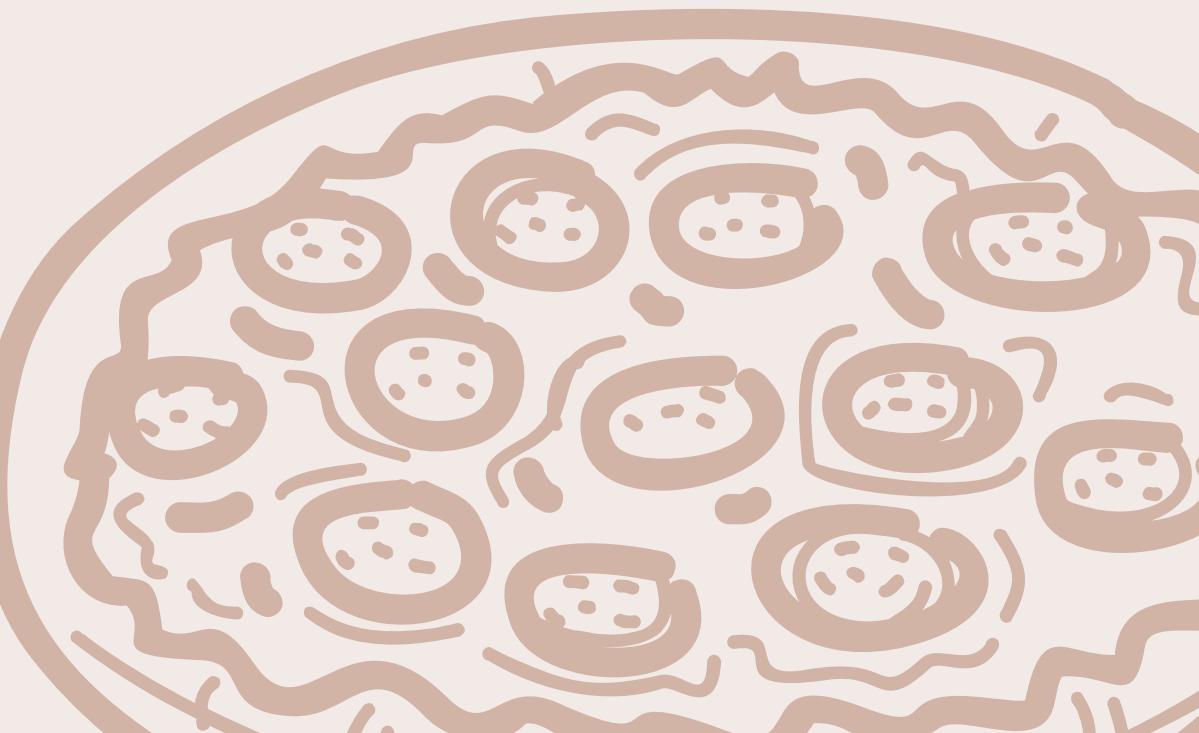
```
SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    orders;
```

	total_orders
	21350

Calculate total revenue generated from pizza sales.

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

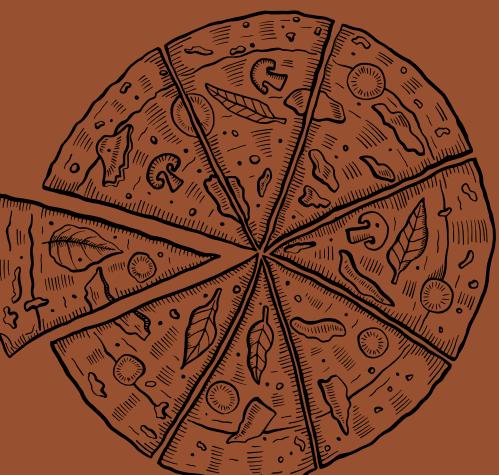
total_revenue
817860.05



Identify the highest priced pizza

```
SELECT pt.name as Pizza_Name, p.price as Pizza_Price  
FROM pizza_types pt  
JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id  
ORDER BY p.price DESC  
LIMIT 1;
```

Pizza_Name	Pizza_Price
The Greek Pizza	35.95



Identify the most common pizza size ordered.

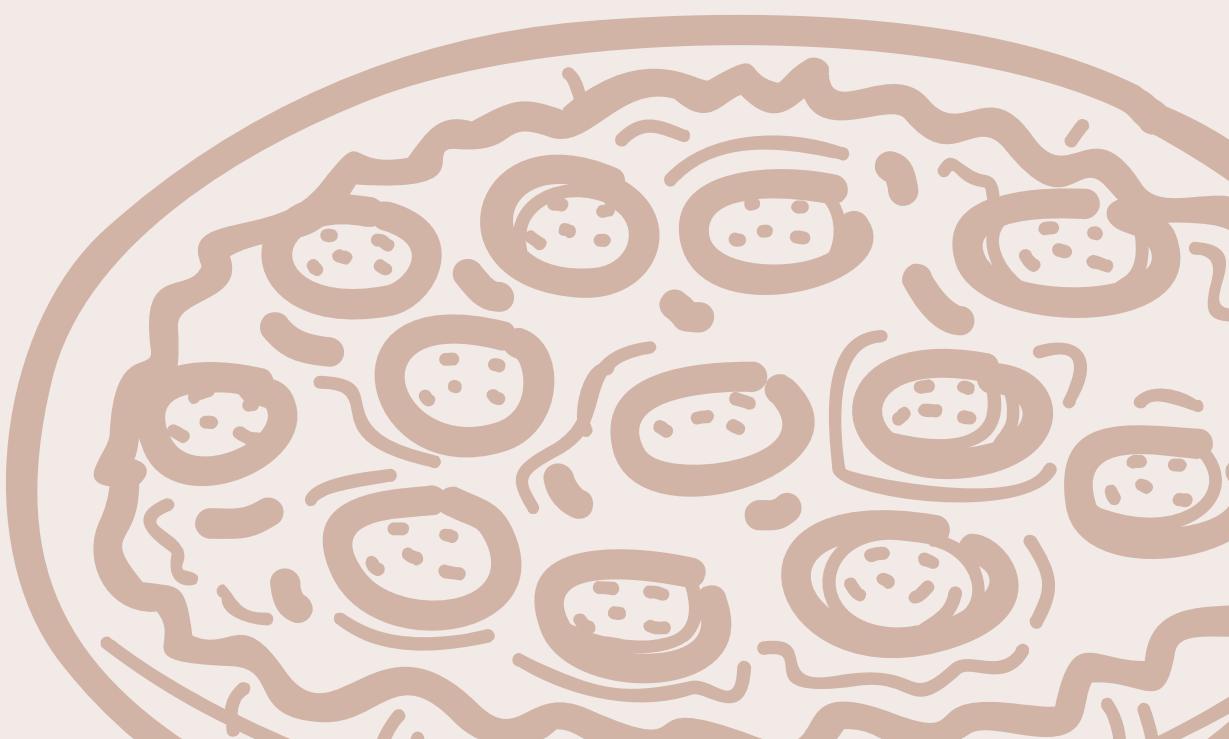
```
SELECT  
    size as Pizza_Size, COUNT(order_id) AS total_orders  
FROM  
    pizzas p  
        JOIN  
    order_details od ON p.pizza_id = od.pizza_id  
GROUP BY Pizza_Size  
ORDER BY total_orders DESC  
LIMIT 1;
```

Pizza_Size	total_orders
L	18526

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

```
SELECT  
    pizza_type_id AS Pizza_Type,  
    COUNT(order_id) AS Total_Order_Placed,  
    SUM(quantity) AS Total_Quantities_Ordered  
FROM  
    pizzas p  
        JOIN  
    order_details od ON p.pizza_id = od.pizza_id  
GROUP BY Pizza_Type  
ORDER BY Total_Order_Placed DESC  
LIMIT 5;
```

Pizza_Type	Total_Order_Placed	Total_Quantities_Ordered
classic_dlx	2416	2453
bbq_ckn	2372	2432
hawaiian	2370	2422
pepperoni	2369	2418
thai_ckn	2315	2371



Join the necessary tables to find the quantity of each pizza CATEGORY ordered

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

Pizza_Category	Total_qty_ordered
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



Determine the distribution of orders by hour of the day.

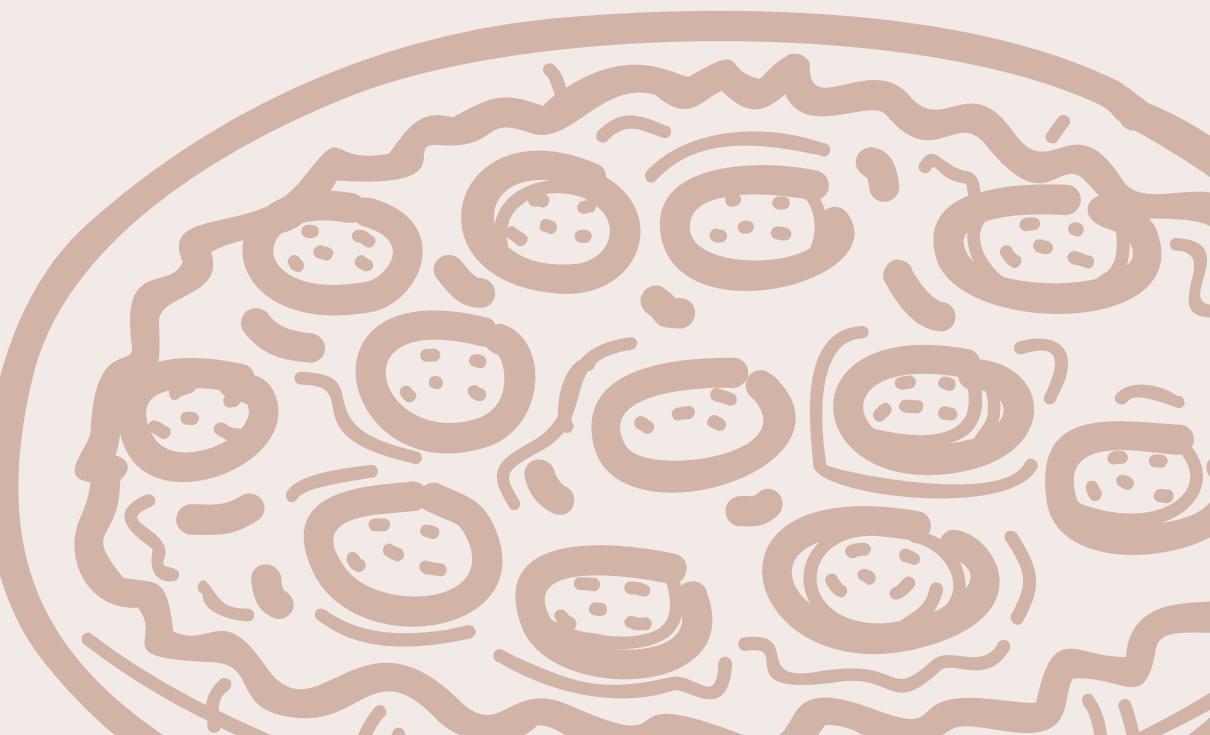
```
SELECT  
    HOUR(order_time) AS Hour_of_day,  
    COUNT(order_id) AS Order_Count  
FROM  
    orders  
GROUP BY Hour_of_day  
ORDER BY Order_Count DESC;
```

Hour_of_day	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

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

```
SELECT  
    category AS Pizza_Category,  
    COUNT(pizza_type_id) AS Pizza_type_count  
FROM  
    pizza_types  
GROUP BY Pizza_Category;
```

Pizza_Category	Pizza_type_count
Chicken	6
Classic	8
Supreme	9
Veggie	9



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

```
SELECT  
    ROUND(AVG(total_pizzas_ordered), 0) AS avg_orders_per_day  
FROM  
    (SELECT  
        order_date, SUM(quantity) AS total_pizzas_ordered  
    FROM  
        orders o  
    JOIN order_details od ON o.order_id = od.order_id  
    GROUP BY order_date) AS orders_per_day;
```

avg_orders_per_day
138



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

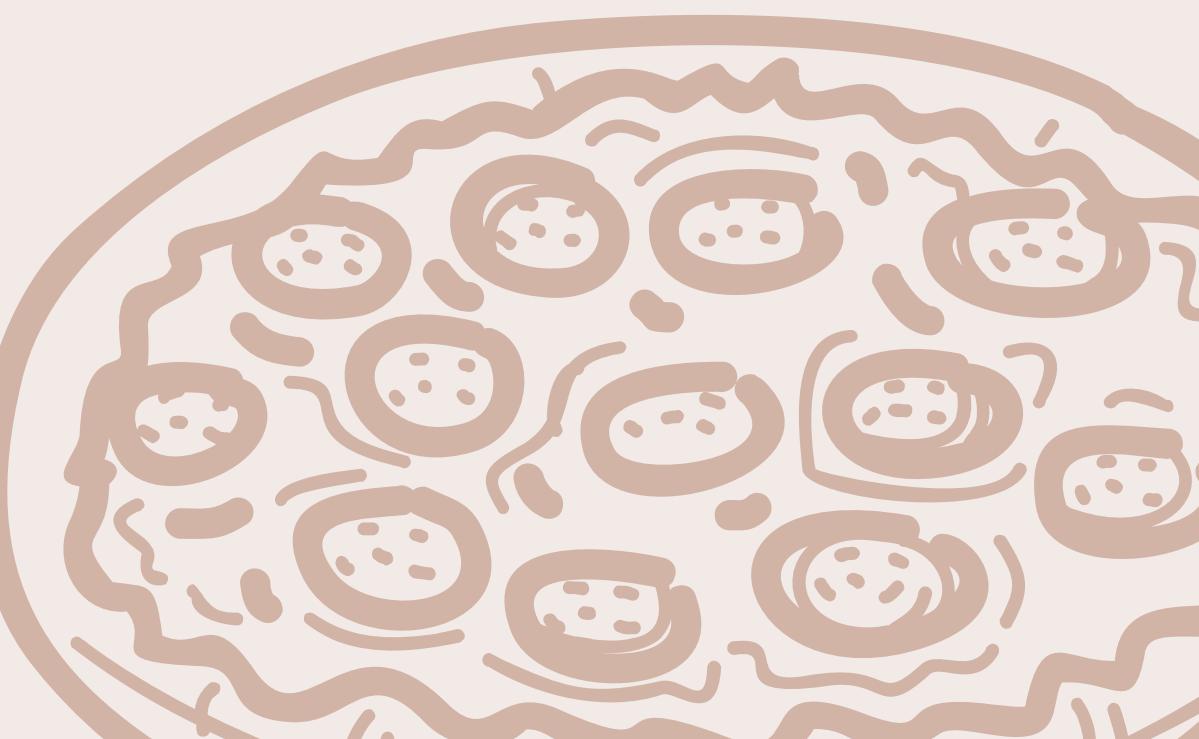
```
SELECT pt.name AS Pizza_Name, SUM(od.quantity * p.price) AS Revenue
FROM pizza_types pt
    JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
    JOIN order_details od ON p.pizza_id = od.pizza_id
GROUP BY Pizza_Name
ORDER BY revenue DESC
LIMIT 3;
```

Pizza_Name	Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Calculate the percentage contribution of each pizza category 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_revenue
                                         FROM pizzas p
                                         JOIN order_details od ON p.pizza_id = od.pizza_id)) * 100,
          2) AS Percentage_Contribution
FROM pizza_types pt
    JOIN pizzas p ON p.pizza_type_id = pt.pizza_type_id
    JOIN order_details od ON p.pizza_id = od.pizza_id
GROUP BY Pizza_Category
ORDER BY Percentage_Contribution DESC;
```

Pizza_Category	Percentage_Contribution
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



Analyze the cumulative revenue generated over time.

```
select order_date, per_day_revenue,  
round(sum(per_day_revenue) over(order by order_date),2)  
as Cumulative_Revenue  
from  
(select o.order_date, round(sum(p.price*od.quantity),2)  
as per_day_revenue  
from orders o join order_details od  
on o.order_id = od.order_id  
join pizzas p on od.pizza_id = p.pizza_id  
group by order_date) as day_rev;
```

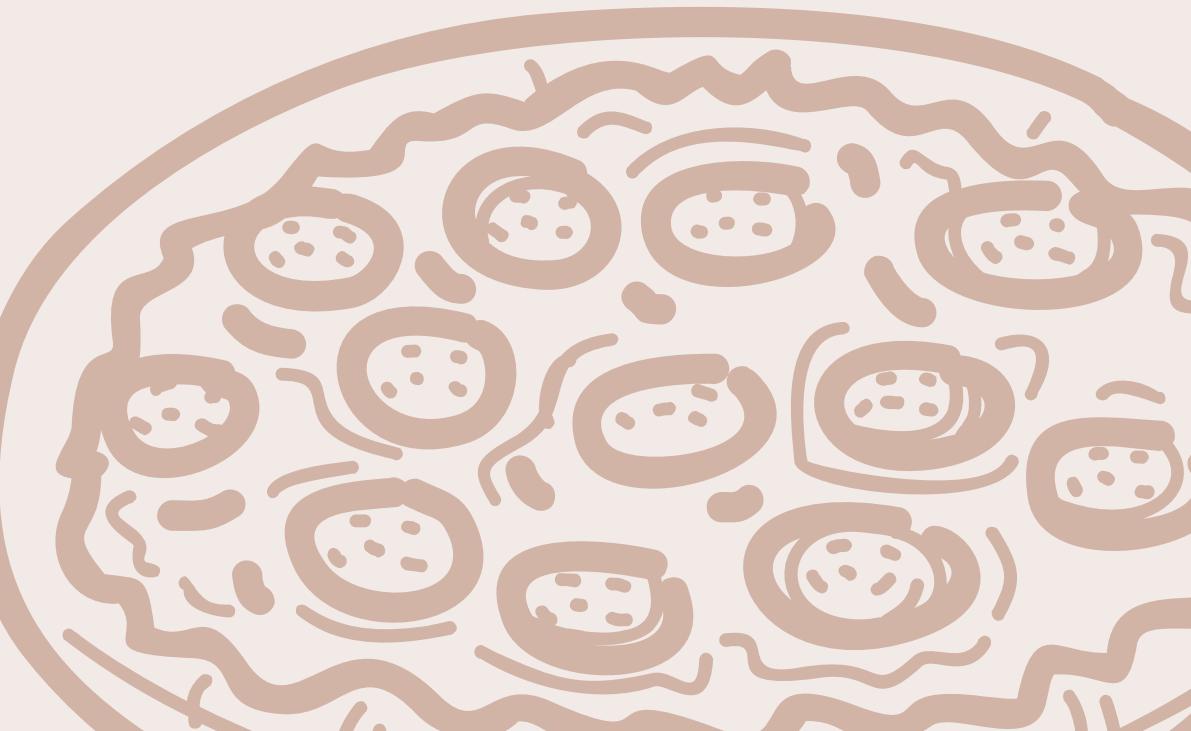
order_date	per_day_revenue	Cumulative_Revenue
2015-01-01	2713.85	2713.85
2015-01-02	2731.9	5445.75
2015-01-03	2662.4	8108.15
2015-01-04	1755.45	9863.6
2015-01-05	2065.95	11929.55
2015-01-06	2428.95	14358.5
2015-01-07	2202.2	16560.7
2015-01-08	2838.35	19399.05
2015-01-09	2127.35	21526.4
2015-01-10	2463.95	23990.35
2015-01-11	1872.3	25862.65
2015-01-12	1919.05	27781.7
2015-01-13	2049.6	29831.3
2015-01-14	2527.4	32358.7
2015-01-15	1984.8	34343.5
2015-01-16	2594.15	36937.65
2015-01-17	2064.1	39001.75
2015-01-18	1976.85	40978.6

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

```
select category, name, revenue
from
(select category, name, total_ordered, revenue,
rank() over (partition by category order by revenue desc)
as rn
from
(select pt.category, pt.name, sum(quantity) as total_ordered,
round(sum(p.price*od.quantity),2) as revenue
from pizza_types pt join pizzas p
on pt.pizza_type_id = p.pizza_type_id
join order_details od on p.pizza_id = od.pizza_id
group by category, pt.name) as a) as b
where rn <=3;
```

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	32265.7
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

For more details of this project, [Click here](#) to navigate to my GitHub portfolio



Thank you !

