



PIZZA SALES REPORT

20
24

presented by
Sagar Parkhe





INTRODUCTION

Welcome to the pizza sales report presentation. This project delves into the rich tapestry of pizza sales data, aiming to uncover insights that can drive strategic decision-making and drive business growth.

Using SQL, we analyzed a comprehensive dataset to answer key questions regarding order patterns, revenue generation, and customer preferences. From determining the total number of orders and calculating revenue to identifying the most popular pizza types and sizes, this project provides a clear view of the pizza landscape.

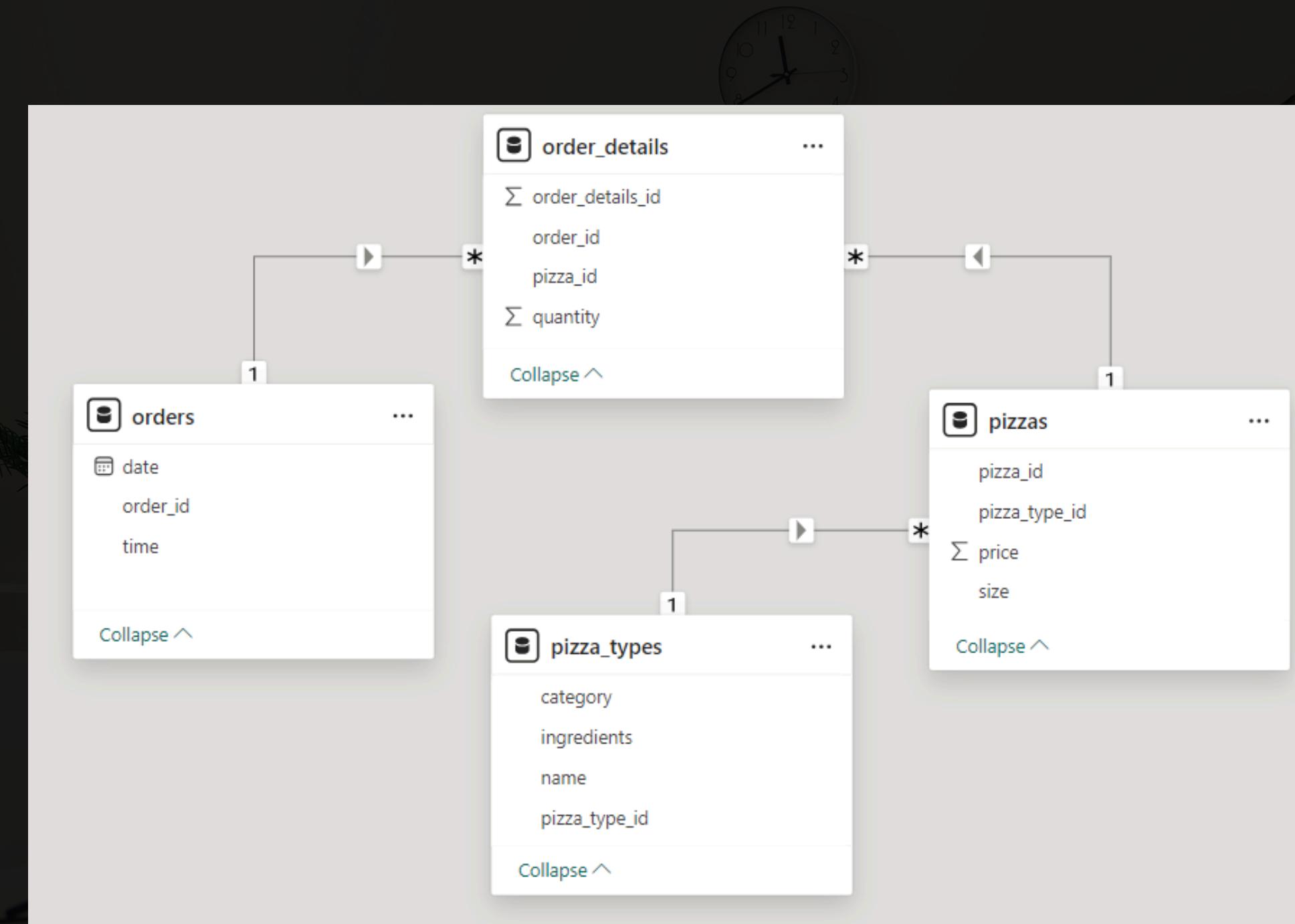


presented by
Sagar Parkhe

Data Model Overview

THIS DATA MODEL ILLUSTRATES THE RELATIONSHIPS BETWEEN THE KEY TABLES USED IN OUR PIZZA SALES ANALYSIS. THE STRUCTURE CONSISTS OF FOUR PRIMARY TABLES:

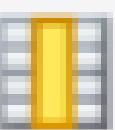
- 1. ORDERS:** CAPTURES ESSENTIAL ORDER DETAILS, INCLUDING ORDER IDS, DATES, AND TIMES.
- 2. ORDER DETAILS:** CONNECTS ORDERS TO SPECIFIC PIZZAS, DETAILING THE QUANTITIES OF EACH PIZZA ORDERED.
- 3. PIZZAS:** CONTAINS INFORMATION ABOUT INDIVIDUAL PIZZAS, INCLUDING THEIR IDs, PRICES, AND SIZES.
- 4. PIZZA TYPES:** CLASSIFIES PIZZAS INTO DIFFERENT CATEGORIES, LISTING THEIR NAMES AND INGREDIENTS.



This relational model allows for comprehensive queries to analyze order patterns, revenue generation, and customer preferences effectively. The connections between tables enable seamless data retrieval, providing valuable insights into our pizza sales operations.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

| Result Grid |  

	total_orders
▶	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

SELECT

```
    ROUND(SUM((order_details.quantity * pizzas.price)),  
          2) AS total_revenue
```

FROM

```
order_details
```

JOIN

```
pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid	
	total_revenue
▶	817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

SELECT

 pizza_types.name, pizzas.price

FROM

 pizza_types

 JOIN

 pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id

ORDER BY price DESC

LIMIT 1;

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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	18526
	M	15385
	S	14137
	XL	544
	XXL	28

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
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;
```

	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

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

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 pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizza_types.category  
ORDER BY quantity DESC;
```

Result Grid | Filter Rows:

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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

	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

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

SELECT

 pizza_types.category, COUNT(name)

FROM

 pizza_types

GROUP BY pizza_types.category;

Result Grid | Filter Rows:

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.

SELECT

```
ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
```

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_quantity;
```

Result Grid		  Filter Rows:
	avg_pizza_ordered_per_day	
▶	138	

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

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 pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizza_types.name  
ORDER BY revenue DESC  
LIMIT 3;
```

Result Grid		Filter Rows:
	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 TYPE TO TOTAL REVENUE.

SELECT

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

Result Grid		Filter
	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

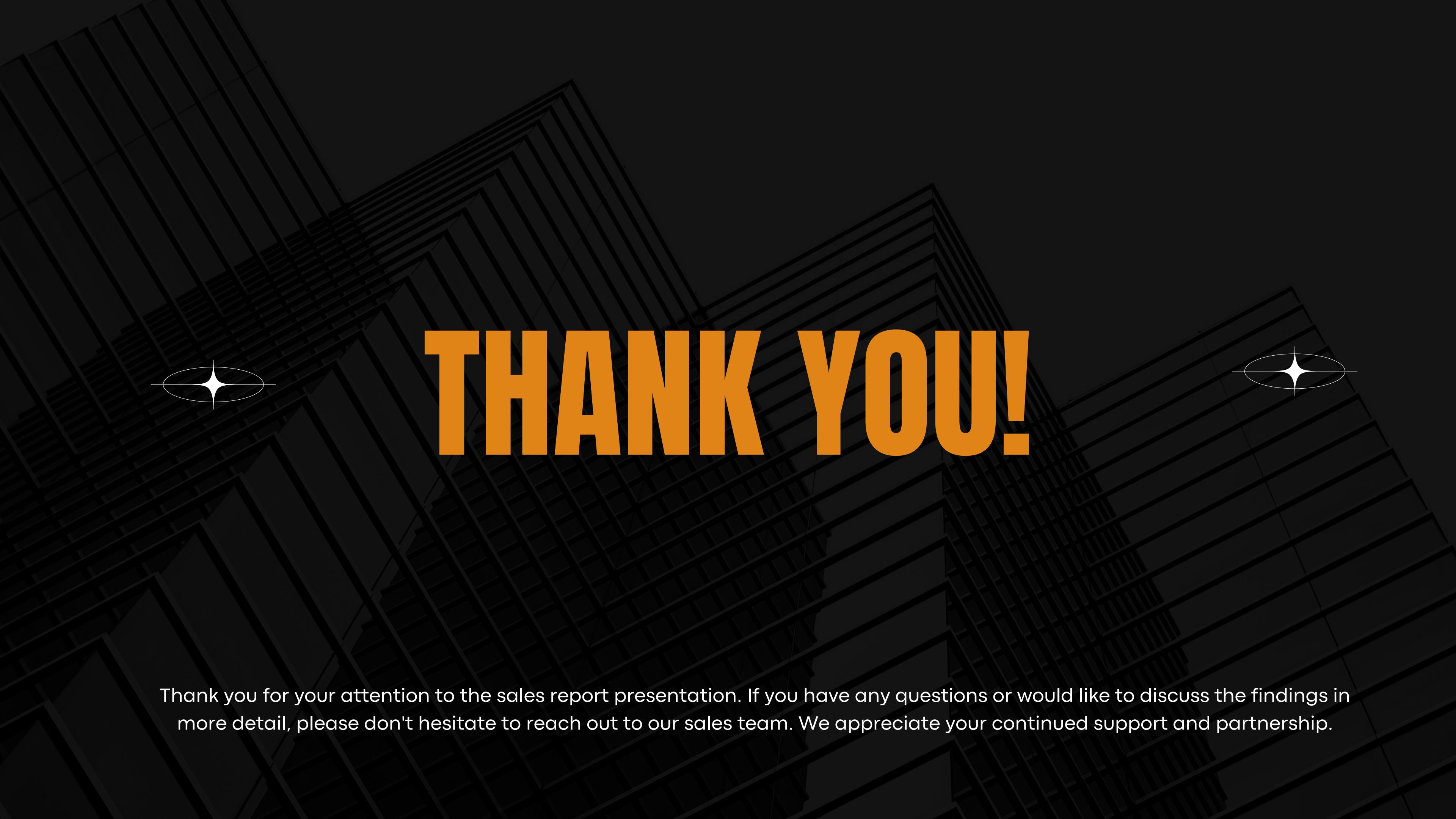
ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
Select order_date, sum(revenue) over (order by order_date) as cumulative_revenue
from
(select orders.order_date, sum(order_details.quantity*pizzas.price) as revenue
from orders
join order_details
on orders.order_id = order_details.order_id
JOIN pizzas
on order_details.pizza_id = pizzas.pizza_id
group by orders.order_date) as sales;
```

	order_date	cumulative_revenue
▶	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11627.2

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
select category, 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 pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn <= 3;
```



THANK YOU!

Thank you for your attention to the sales report presentation. If you have any questions or would like to discuss the findings in more detail, please don't hesitate to reach out to our sales team. We appreciate your continued support and partnership.