

PROJECT

PIZZA SALES ANALYSIS

UNCOVERING INSIGHTS
FROM SALES DATA

BY - Prashant Gowda

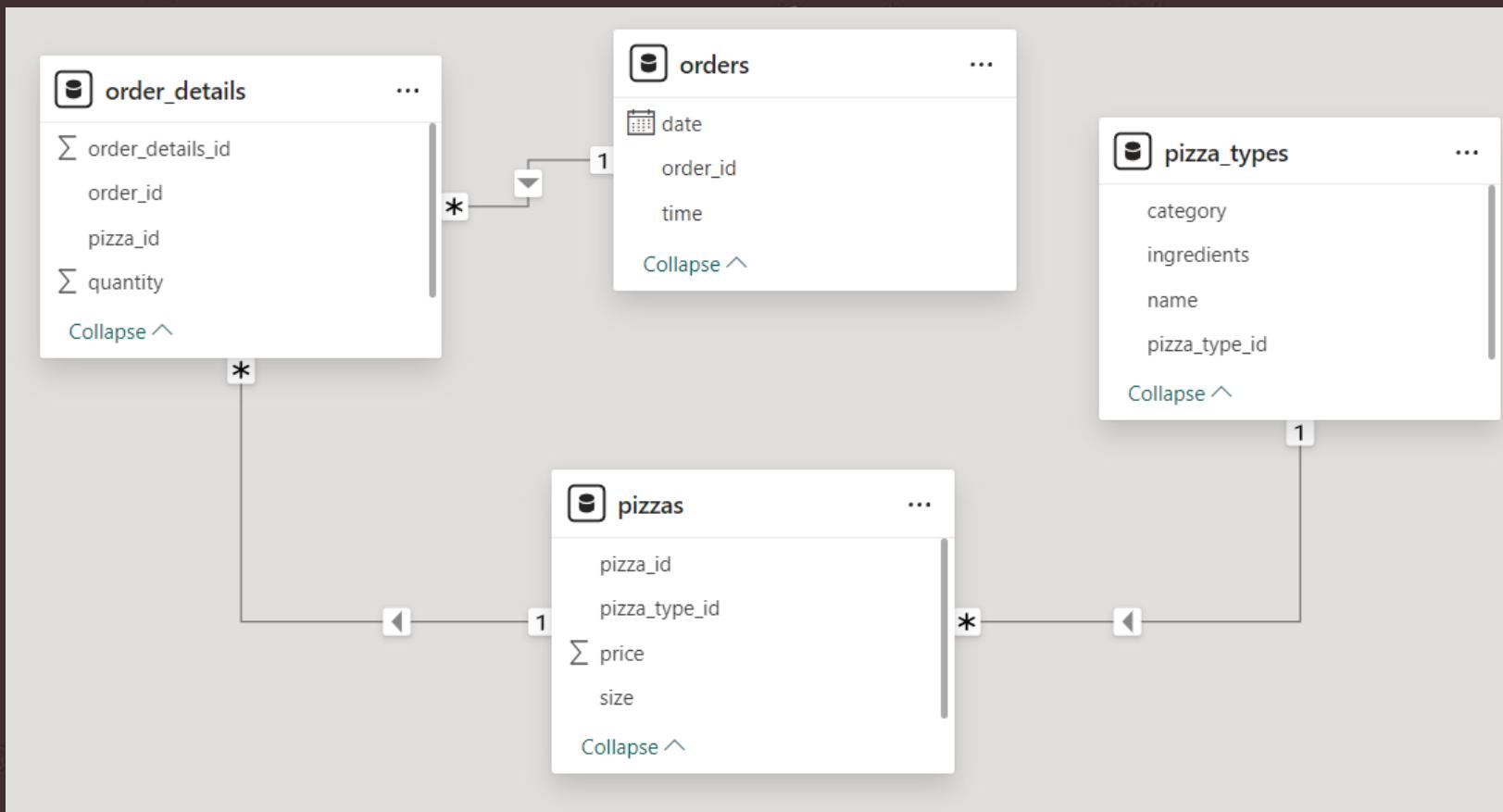


PURPOSE OF ANALYSIS

The primary goal of this analysis is to leverage pizza sales data to uncover key business insights that can help improve sales strategies, enhance customer satisfaction, and optimize operations.



SCHEMA & DATA



QUERIES: BASIC

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.



BASIC QUERIES

1. Retrieve The Total Number Of Orders Placed.

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

	total_orders
▶	21350



BASIC QUERIES

2 .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 order_details.pizza_id = pizzas.pizza_id;
```

	total_revenue
▶	817860.05



BASIC QUERIES

3. Identify the highest-priced pizza.

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

	name	price
▶	The Greek Pizza	35.95



BASIC QUERIES

4. Identify the most common pizza size ordered.

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

	size	total_orders
▶	L	18526



BASIC QUERIES

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

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

	name	total_quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371



QUERIES: INTERMEDIATE

1. Total Quantity by Pizza Category
2. Hourly Order Distribution
3. Category-Wise Pizza Distribution
4. Daily Average Orders
5. Top 3 Pizzas by Revenue



INTERMEDIATE QUERIES

1. Total Quantity by Pizza Category

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_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 category
ORDER BY total_quantity DESC;
```

	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



INTERMEDIATE QUERIES

2. Hourly Order Distribution

```
SELECT  
    HOUR(order_time) AS hours, COUNT(order_id)  
FROM  
    orders  
GROUP BY hours;
```

	hours	COUNT(order_id)
▶	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



INTERMEDIATE QUERIES

3. Category-Wise Pizza Distribution

```
select category, count(name) as types from pizza_types  
group by category;
```

	category	types
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



INTERMEDIATE QUERIES

4. Daily Average Orders

```
SELECT
    ROUND(AVG(quantity), 0) as avg_quantity
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;
```

	avg_quantity
▶	138



INTERMEDIATE QUERIES

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

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5



QUERIES: ADVANCED

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.



ADVANCED QUERIES

1. 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 order_details.pizza_id = pizzas.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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



ADVANCED QUERIES

2. Analyze the cumulative revenue generated over time.

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

	order_date	cum_revenue
▶	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55



ADVANCED QUERIES

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

```
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;
```

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5



INSIGHTS

1. The Veg category has the highest total quantity ordered, indicating a strong preference for vegetarian options among customers.
2. Peak order times are around lunch (12 PM - 2 PM) and dinner (6 PM - 8 PM).
3. Non-Veg pizzas also have a significant share, particularly popular in larger sizes.
4. The top 3 pizzas contributing the most to revenue are The Thai Chicken Pizza, The Barbecue Chicken Pizza, and The California Chicken Pizza.
5. Classic And Supreme Pizzas are the most ordered Pizzas

SUGGESTION TO INCREASE SALES

1. Consider expanding the Veg menu to include more variety or special offers to boost sales further.
2. Optimize staffing and kitchen operations during peak hours to ensure timely service and maximize order fulfillment.
3. Introduce combo deals or meal packages that include popular Non-Veg pizzas to increase average order value.
4. Enhance loyalty programs to reward frequent customers with discounts, free items, or exclusive deals, encouraging repeat business.
5. Focus marketing efforts on these top sellers with special promotions, feature them prominently on the menu, and ensure consistent availability.