



PIZZAZZ RESTAURANT

# PIZZAZZ PIES

@Shivkr1470



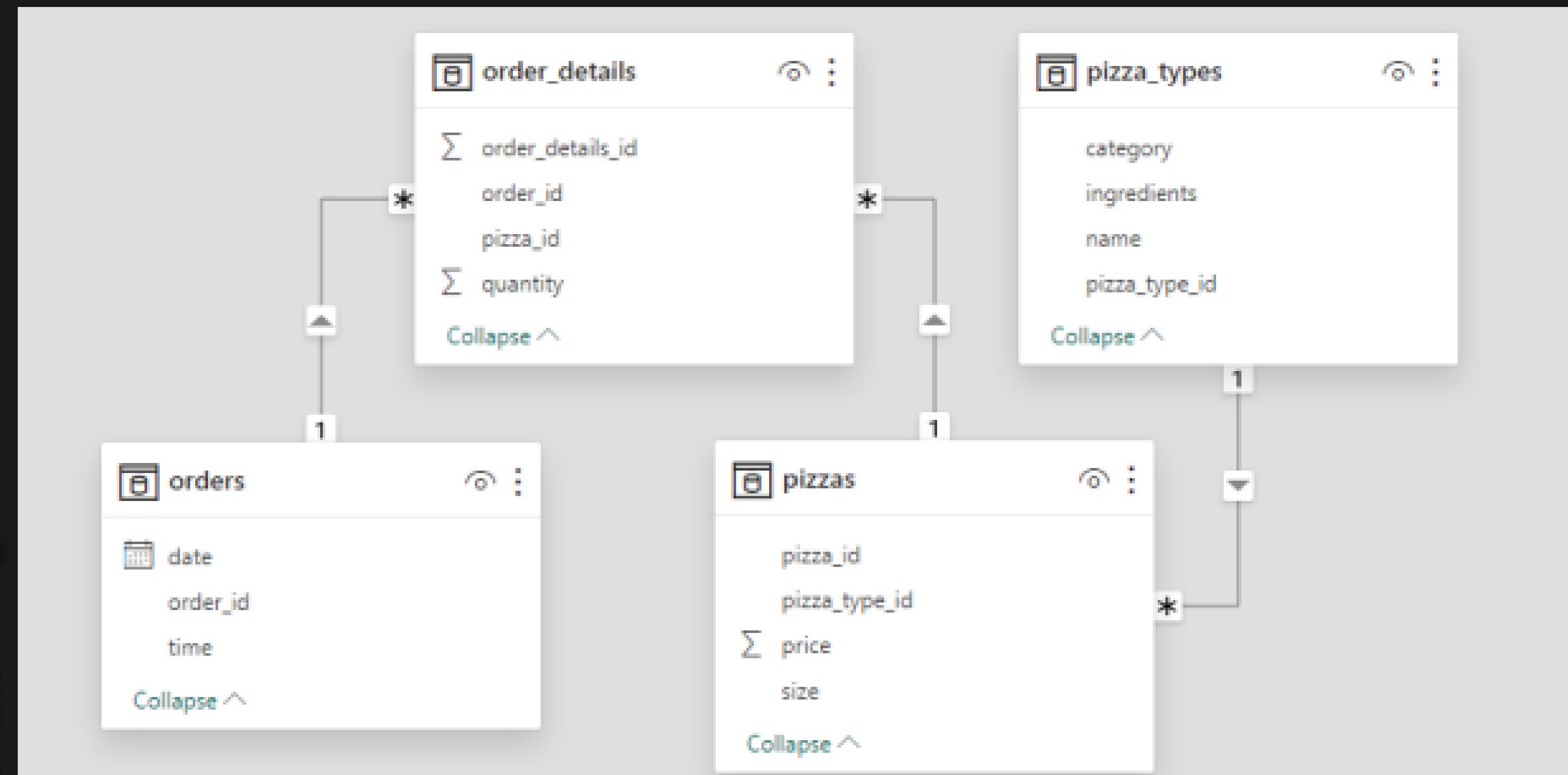
# PROJECT OVERVIEW

The Pizza Sales Analysis project leverages SQL to explore sales patterns, customer preferences, and business performance of a pizza restaurant. Basic analysis includes calculating total orders, revenue, identifying the highest-priced pizza, the most common pizza size, and the top 5 most ordered pizza types. Intermediate analysis involves joining tables to find total quantities of each pizza category, distribution of orders by hour, category-wise pizza distribution, daily average orders, and top 3 pizza types by revenue. Advanced analysis calculates each pizza type's revenue contribution, cumulative revenue over time, and top 3 pizza types by revenue within each category. These insights guide strategic decisions to optimize operations and boost profitability.



# SCHEMA

## Entity Relationship Diagram



# ANALYSIS QUESTIONS

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.
6. Determine the distribution of orders by hour of the day.
7. Find the total quantity of each pizza category ordered.
8. Find the category-wise distribution of pizzas.
9. Group the orders by date and calculate the average number of pizzas ordered per day.
10. Determine the top 3 most ordered pizza types based on revenue.
11. Analyze the cumulative revenue generated over time.
12. Top 3 Pizza Sizes by Quantity Sold.



# 1. Retrieve the total number of orders placed.

```
55 •     SELECT COUNT(order_id) AS total_order  
56      FROM orders;  
57
```

	total_order
▶	21350

## 2. Calculate the total revenue generated from pizza sales.

```
61 •   SELECT
62     ROUND(SUM(order_details.quantity * pizzas.price),
63             2) AS total_sales
64   FROM
65     order_details
66     JOIN
67     pizzas ON order_details.pizza_id = pizzas.pizza_id
68
```

Result Grid | Filter Rows: \_\_\_\_\_ | Export: \_\_\_\_\_ | Wrap Cell Content: \_\_\_\_\_

total_sales
817860.05

### 3. Identify the highest-priced pizza.

```
72 •   SELECT
73       pizza_types.name, pizzas.price
74   FROM
75       pizza_types
76   JOIN
77       pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
78   ORDER BY pizzas.price DESC
79   LIMIT 1;
80
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

	name	price
▶	The Greek Pizza	35.95

## 4. Identify the most common pizza size ordered.

```
84 •   SELECT
85       pizzas.size, SUM(order_details.quantity) AS quantity
86   FROM
87       order_details
88       JOIN
89       pizzas ON pizzas.pizza_id = order_details.pizza_id
90   GROUP BY pizzas.size
91   ORDER BY quantity DESC
92   LIMIT 1;
```

Result Grid | Filter Rows:  Export:  Wrap Cell Content:

	size	quantity
▶	L	18956

## 5. List The top 5 most ordered pizza\_types along with their quantities.

```
97 •   SELECT
98     pizza_types.name, SUM(order_details.quantity) AS quantity
99   FROM
100     order_details
101       JOIN
102       pizzas ON pizzas.pizza_id = order_details.pizza_id
103       JOIN
104       pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
105   GROUP BY pizza_types.name
106   ORDER BY quantity DESC
107   LIMIT 5;
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

	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

## 6.Determine the distribution of orders by hour of the day.

```
126 •   SELECT  
127       HOUR(time) AS hour, COUNT(order_id) AS order_count  
128   FROM  
129     orders  
130 GROUP BY hour;
```

	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

## 7. Find the total quantity of each pizza category ordered.

```
112 •   SELECT
113       pz.category, SUM(od.quantity) AS total_quantity
114   FROM
115       pizza_types AS pz
116           JOIN
117       pizzas AS p ON pz.pizza_type_id = p.pizza_type_id
118           JOIN
119       order_details AS od ON od.pizza_id = p.pizza_id
120   GROUP BY pz.category
121   ORDER BY total_quantity DESC;
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

## 8.find the category-wise distribution of pizzas.

```
135 •      SELECT category, COUNT(name) AS count  
136        FROM pizza_types  
137        GROUP BY category;
```

Result Grid | Filter Rows:  Export:

	category	count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

```
142 •   SELECT ROUND(AVG(quantity),0) AS avg_order_per_day FROM
143   (SELECT o.date, SUM(od.quantity) AS quantity
144    FROM orders AS o
145    JOIN order_details AS od
146    ON o.order_id = od.order_id
147    GROUP BY o.date) AS order_quantity;
```

```
148
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	avg_order_per_day			
▶	138			

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

```
152 •   SELECT pt.name, ROUND(SUM(od.quantity * p.price),2) AS revenue  
153     FROM order_details AS od  
154     JOIN pizzas AS p  
155     ON od.pizza_id = p.pizza_id  
156     JOIN pizza_types AS pt  
157     ON pt.pizza_type_id = p.pizza_type_id  
158     GROUP BY pt.name  
159     ORDER BY revenue DESC  
160     LIMIT 3;
```

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

## 11. Analyze the cumulative revenue generated over time.

```
• SELECT
    date,
    SUM(revenue) OVER (ORDER BY date) AS cum_revenue
  FROM (
    SELECT
        o.date,
        SUM(od.quantity * p.price) AS revenue
      FROM orders AS o
      JOIN order_details AS od ON od.order_id = o.order_id
      JOIN pizzas AS p ON od.pizza_id = p.pizza_id
      GROUP BY o.date
  ) AS sales;
```

date	cum_revenue
2015-01-01	2713.850000000004
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	23990.35000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.30000000003
2015-01-14	32358.70000000004
2015-01-15	34343.50000000001

## 12. Top 3 Pizza Sizes by Quantity Sold

```
SELECT
    pizzas.size, SUM(order_details.quantity) AS total_sold
FROM
    pizzas
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY total_sold DESC
LIMIT 3;
```

size	total_sold
L	18956
M	15635
S	14403



THANK  
YOU