

# PIZZA SALES ANALYSIS



# WELCOME TO MY PROJECT

My Name Is Shuham Kumar Singh and In This project dives into pizza sales data using SQL to uncover top-sellers, peak hours, and revenue trends—turning raw data into smart decisions for better growth using sql queries .

Tools used: MySQL, Power BI, Excel.

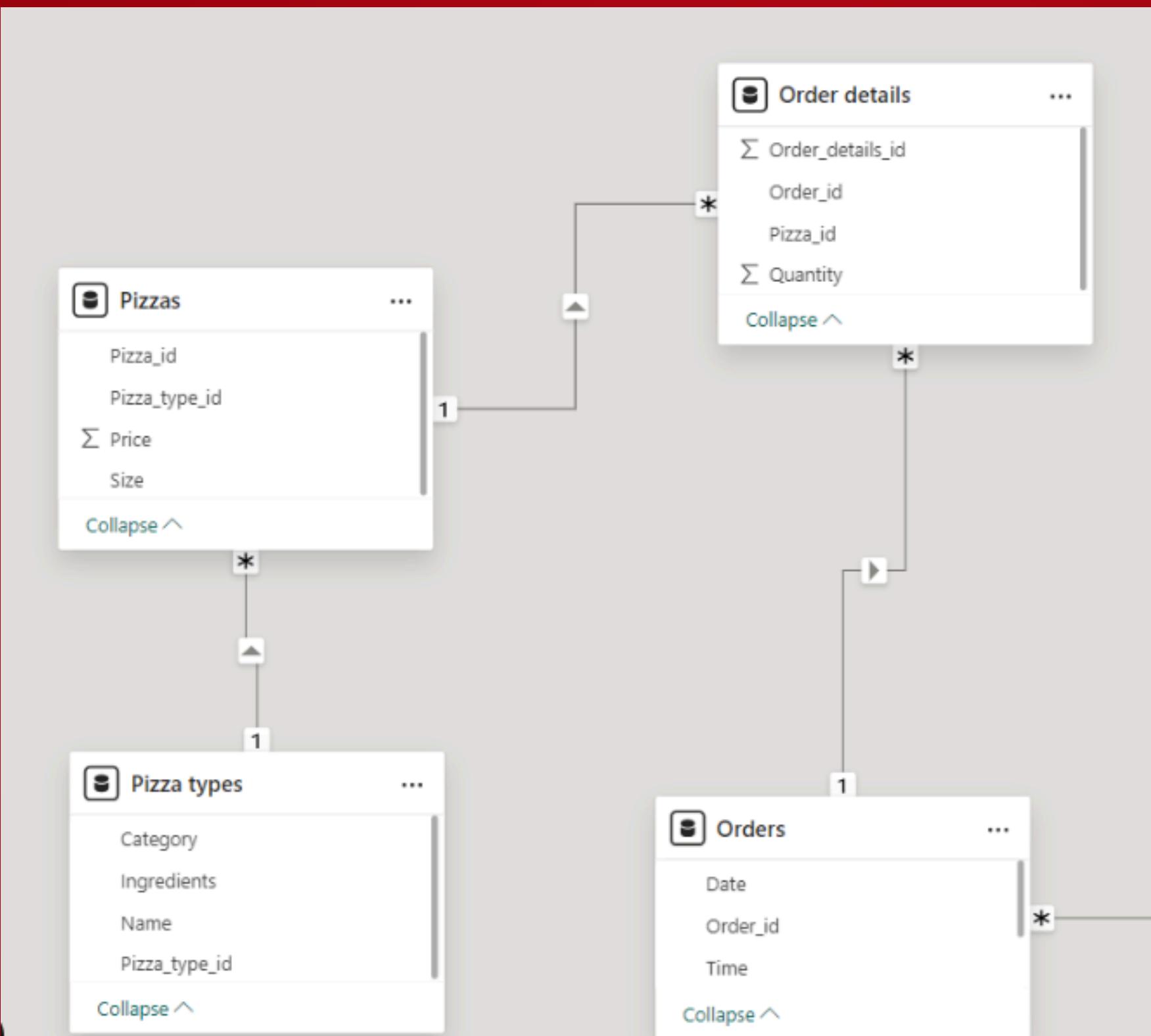




## DATASET USED-

- orders – order\_id , order\_date , and order\_time.
- orders\_details – order\_detail\_id order\_id , pizza\_id and quantity.
- pizzas – pizza\_id, size, and price.
- pizza\_types – pizza\_type\_id , name , category and ingredients

# DATA MODELLING



# 1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

select

```
count(order_id) as total_order
```

from

```
pizzahut.orders;
```

total_order
-------------

21350
-------



2.Calculate the total revenue generated from pizza sales.

```
select round(sum(orders_details.quantity * pizzas.price),2) as tol_sales  
from orders_details  
join pizzas  
on pizzas.pizza_id = orders_details.pizza_id;
```

tol_sales
-----------

817860.05
-----------



### 3.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;
```



	name	price
▶	The Greek Pizza	35.95

4. Identify the most common pizza size ordered.



```
select pizzas.size,  
       count(orders_details.order_detail_id) as c_o  
  from pizzas  
join orders_details  
on pizzas.pizza_id = orders_details.pizza_id  
group by pizzas.size  
order by c_o desc  
limit 1;
```

	size	c_o
▶	L	18526





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

```
select pizza_types.name, sum(orders_details.quantity) as c_o  
from pizza_types  
join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join orders_details  
on orders_details.pizza_id = pizzas.pizza_id  
group by pizza_types.name  
order by c_o desc  
limit 5 ;
```

name	c_o
The Classic Deluxe Pizza	2500
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

```
select pizza_types.category, sum(orders_details.quantity) as count_order  
from pizza_types  
join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join orders_details  
on orders_details.pizza_id = pizzas.pizza_id  
group by pizza_types.category;
```

category	count_order
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

## 7.DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
select left(order_time, 2) as hrs ,count(order_id)  
from orders  
group by hrs;
```

hrs	count(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920

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

```
select pizza_types.category, count(orders_details.quantity) as count_order  
from pizza_types  
join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join orders_details  
on orders_details.pizza_id = pizzas.pizza_id  
group by pizza_types.category;
```

category	count_order
Classic	14579
Veggie	11449
Supreme	11777
Chicken	10815

## 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(orders_details.quantity) AS quantity  
FROM  
    orders  
    JOIN  
    orders_details ON orders.order_id = orders_details.order_id  
GROUP BY 1) AS order_quantity;
```

avg\_pizza\_ordered\_per\_day

138

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

```
select pizza_types.name, round(sum(orders_details.quantity*pizzas.price),2) as revenue  
from pizza_types  
join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join orders_details  
ON orders_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

## 11. 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(orders_details.quantity * pizzas.price) AS revenue
FROM orders_details
JOIN pizzas
ON orders_details.pizza_id = pizzas.pizza_id
JOIN orders ON orders.order_id = orders_details.order_id
GROUP BY 1) AS sales;
```

order_date	cumulative_revenue
2015-01-01	2713.850000000004
2015-01-02	5445.75
2015-01-03	8108.544575
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5

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

```
select pizza_types.name, pizza_types.category, round(sum(orders_details.quantity*pizzas.price),2) as rev  
from pizza_types  
join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join orders_details  
ON orders_details.pizza_id = pizzas.pizza_id  
group by pizza_types.name, pizza_types.category  
order by revenue desc  
limit 3;
```

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

THANK YOU !