



# PIZZA SALES PROJECT

22 MAY 2024

Guided By WsCube Tech and Ayush Jain. Created by Shrikant Kadam.





# INTRODUCTION

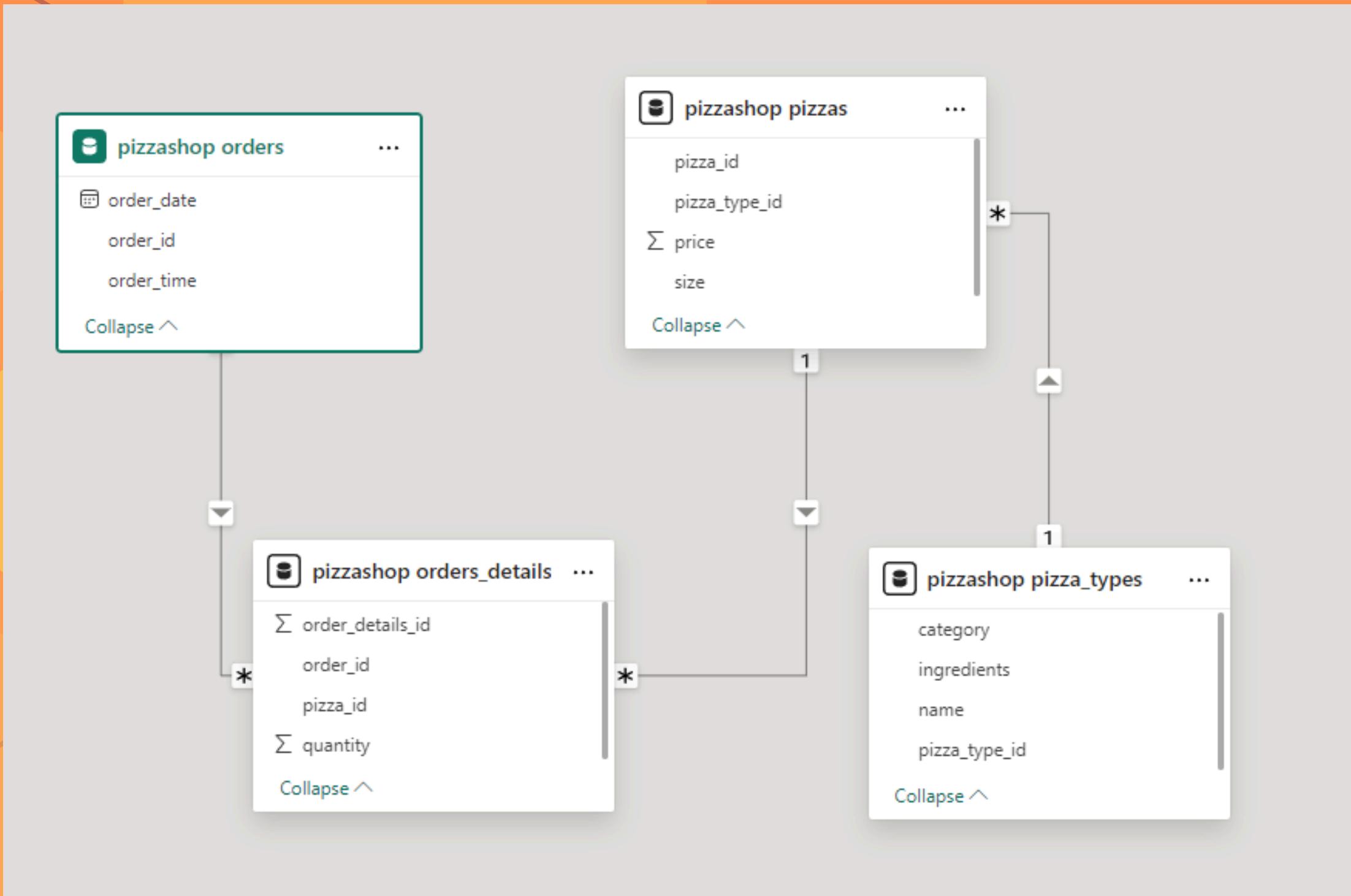


Hi, My name is Shrikant Kadam. I have created this project on Pizza Sales using MYSQL 6.0. In this project, I have used SQL queries which are of basic level to advance level. I hope you find this project helpful.



# PROJECT SCHEMA

Project Contains 4 tables named orders, order\_details, pizzas and pizza\_types. You can find its schema as follows:





# QUESTIONS SOLVED USING MYSQL 6.0

## **Basic:**

- 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.

## **Intermediate:**

- 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.

## **Advanced:**

- 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 THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT count(order_id) AS Total_Orders_Count FROM orders;
```

	Total_Orders_Count
▶	21350



# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT  
    ROUND(SUM(orders_details.quantity * pizzas.price),  
        2) AS Total_Revenue  
FROM  
    orders_details  
    JOIN  
    pizzas ON orders_details.pizza_id = pizzas.pizza_id;
```

	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 pizzas.price DESC
LIMIT 1;
```

	name	price
▶	The Greek Pizza	35.95



# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

**SELECT**

```
pizzas.size,  
COUNT(orders_details.order_details_id) AS order_count
```

**FROM**

```
pizzas
```

**JOIN**

```
orders_details ON pizzas.pizza_id = orders_details.pizza_id
```

**GROUP BY** pizzas.size

**ORDER BY** order\_count **DESC**;

	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(orders\_details.quantity) AS quantity**

**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 quantity DESC**

**LIMIT 5;**

	<b>name</b>	<b>quantity</b>
▶	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(orders_details.quantity) AS quantity  
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

**ORDER BY** quantity **DESC**;

	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 Hours, COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY Hours;
```

	Hours	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399



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

```
SELECT  
    category, COUNT(name)  
FROM  
    pizza_types  
GROUP BY category;
```

	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\_Pizzas\_ordered

**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** orders.order\_date) AS order\_quantity;

	AVG_Pizzas_ordered
▶	138



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

```
SELECT
    pizza_types.name AS pizza_name,
    ROUND(SUM(orders_details.quantity * pizzas.price),
          0) 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_name
ORDER BY revenue DESC
LIMIT 3;
```

	pizza_name	revenue
▶	The Thai Chicken Pizza	43434
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41410



# CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
SELECT
    pizza_types.category AS pizza_category,
    ROUND((SUM(orders_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(orders_details.quantity * pizzas.price),
        2) AS Total_Revenue
    )
    FROM
        orders_details
        JOIN
            pizzas ON orders_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
        orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_category
ORDER BY revenue DESC;
```

	pizza_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 cum_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 orders.order_date) AS 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
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05



# 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((orders_details.quantity) * pizzas.price) 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.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



**THANK YOU!!!**