

# PIZZA SALES . ANALYSIS

USING SQL

By Adarsh kumar

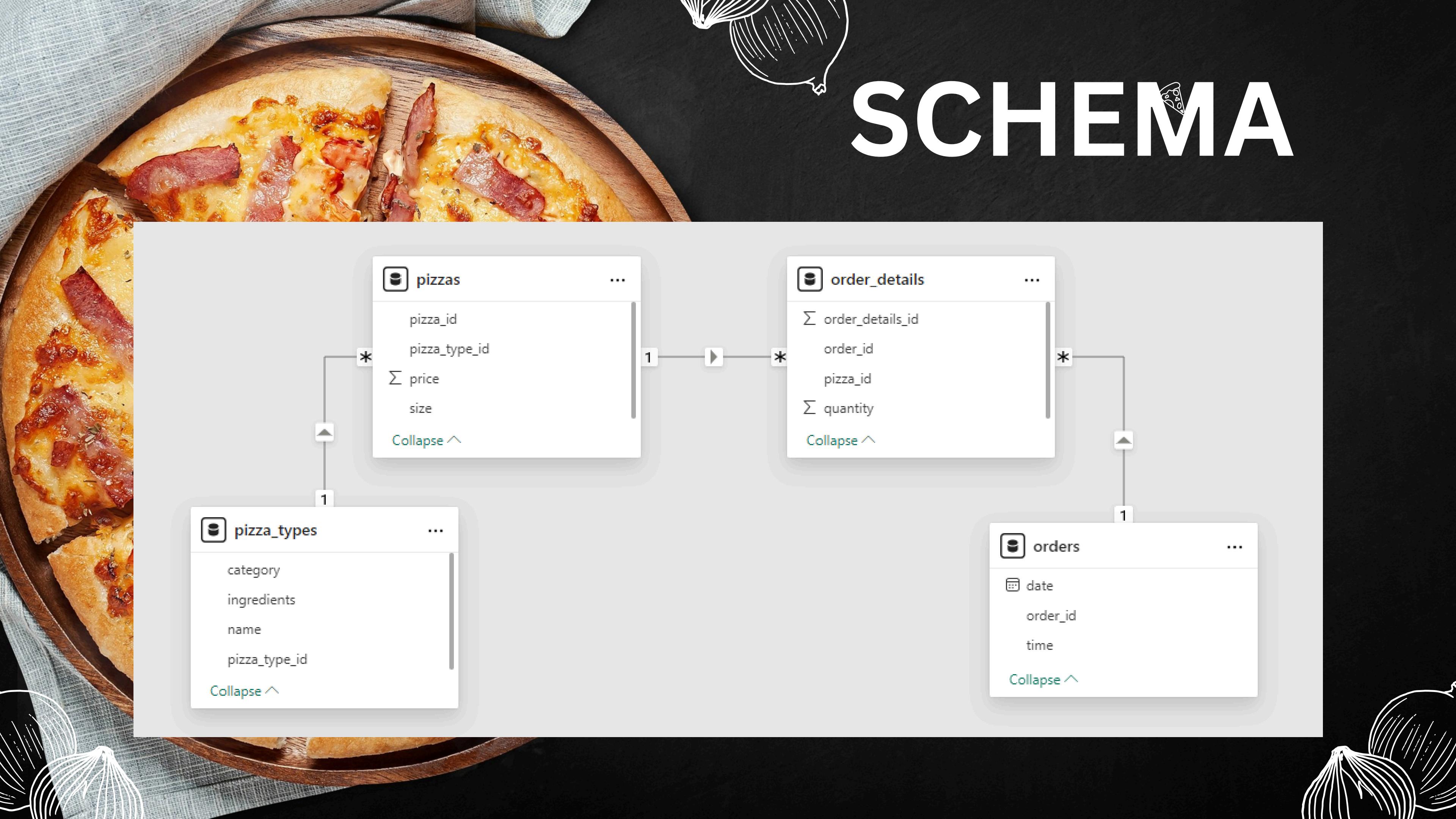


# HELLO EVERYONE

My name is Adarsh Kumar, and I completed a Pizza Sales Analysis project using SQL. In this project, I used SQL queries to explore the data and gain insights that could help improve sales and operations. This project showcases my SQL skills and ability to work with data to support business decisions."



# SCHEMA



# OBJECTIVES

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. Join the necessary tables to find the total quantity of each pizza category ordered.
7. Determine the distribution of orders by hour of the day.
8. Join relevant tables to 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. Calculate the percentage contribution of each pizza type to total revenue.
12. Analyze the cumulative revenue generated over time.
13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

# 1.RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

	total_orders
▶	21350

## 2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT  
    ROUND(SUM(quantity * price), 4) AS total_sales  
FROM  
    order_details  
    JOIN  
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

	total_sales
▶	817860.05

### 3. IDENTIFY THE HIGHEST-PRICED PIZZA.



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

	<b>name</b>	<b>price</b>
▶	The Greek Pizza	35.95

## 4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

	size	total_order
▶	L	18526

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

```
SELECT  
    name, SUM(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 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

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

```
SELECT  
    category, SUM(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 category  
ORDER BY quantity DESC;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

SELECT

```
HOUR(order_time) AS hour, COUNT(order_id)  
FROM  
    orders  
GROUP BY hour  
ORDER BY hour ASC;
```

	hour	COUNT(order_id)
▶	9	1
	10	8
	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009

## 8.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

# 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  
        order_date, SUM(quantity) AS quantity  
    FROM  
        orders  
    JOIN order_details ON orders.order_id = order_details.order_id  
    GROUP BY order_date) AS order_quantity;
```

	avg_pizza_ordered_per_day
▶	138

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

```
SELECT  
    name, SUM(quantity * 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 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. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

SELECT

```
category, round(SUM(quantity * price) / (SELECT  
ROUND(SUM(quantity * price), 4) AS total_sales  
  
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 category
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

## 12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

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

order_date	cumu_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
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7

# 13.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 rnk
    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 rnk <= 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



THANK  
YOU