

Where Every Slice is a Taste of Perfection

# WELCOME TO PIZZA-HUT

HI! I'm RACHANA KAKADE,  
In this Project I have utilized  
SQL Queries to solve  
questions were related to  
PIZZAS-SALE

|||||  
|||||  
**ORDER  
NOW**

Start Your Slide



|||||

# PIZZA SALES ANALYSIS PROJECT - INTRODUCTION

## Overview

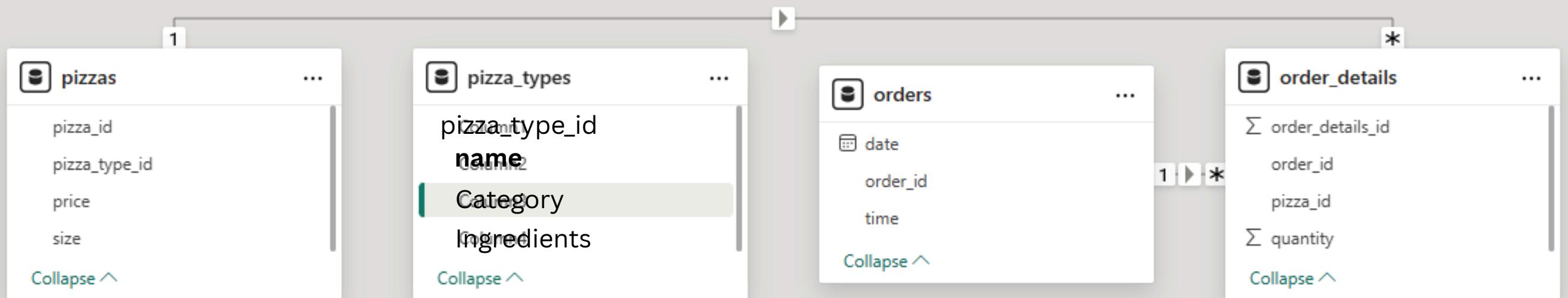
The Pizza Sales Analysis project is a database-driven system designed to track and analyze pizza sales efficiently. It provides insights into order trends, revenue generation, and customer preferences, helping businesses make data-driven decisions.

## Project Objectives

- Efficiently store and manage customer orders.
- Track different pizza types, sizes, and prices.
- Analyze sales performance, including total revenue and most popular pizzas.
- Generate insightful reports using SQL queries and views.



# Key Components



- **Customers Table**: Stores customer details.
- **Orders Table**: Tracks orders with timestamps.
- **Pizza Types & Pizzas Table**: Defines different pizza categories, sizes, and pricing.
- **Order Details Table**: Stores transaction-level data, including quantity and price.
- **Pizza Sales View**: Aggregates sales data to analyze total quantity sold and revenue.



# RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED:

```
SELECT count(DISTINCT order_id) AS total_orders  
FROM orders;
```

- Best Cheese
- Fresh Sausage

Result Grid	
	total_orders
▶	21350



# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES:



Margherita Pizza

```
SELECT count(DISTINCT order_id) AS total_orders  
FROM orders;
```

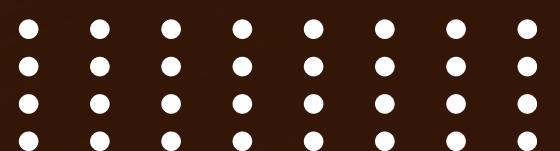
Result Grid	
	total_orders
▶	21350



Veggie Delight



Pepperoni Pizza



# Identify the highest-priced pizza:

```
• SELECT pt.name, max(price) AS highest_price  
  FROM pizzas AS p  
  JOIN pizza_types AS pt  
    ON p.pizza_type_id=pt.pizza_type_id  
  GROUP BY name  
  ORDER BY HIGHEST_PRICE DESC  
  LIMIT 1;
```

Result Grid | Filter Rows:

	name	highest_price
▶	The Greek Pizza	35.95



# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED:



- ```
SELECT p.size, count(*) AS order_count
FROM pizzas AS p
JOIN order_details AS od
ON p.pizza_id=od.pizza_id
GROUP BY p.size
ORDER BY order_count DESC
LIMIT 1;
```



Result Grid |

|   | size | order_count |
|---|------|-------------|
| ▶ | L    | 18526       |



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

- ```
SELECT pt.name AS pizza_type ,count(od.quantity) AS Total_quantity
FROM order_details AS od
JOIN pizzas AS p ON od.pizza_id=p.pizza_id
JOIN pizza_types AS pt ON p.pizza_type_id=pt.pizza_type_id
GROUP BY pt.name
ORDER BY Total_quantity DESC
LIMIT 5;
```

	pizza_type	Total_quantity
▶	The Classic Deluxe Pizza	2416
	The Barbecue Chicken Pizza	2372
	The Hawaiian Pizza	2370
	The Pepperoni Pizza	2369
	The Thai Chicken Pizza	2315



# Join the necessary tables to find the total quantity of each pizza category ordered:

```
SELECT pizza_types.category,sum(order_details.quantity) AS Total_quantity
FROM order_details
JOIN pizzas ON order_details.pizza_id=pizzas.pizza_id
JOIN pizza_types ON pizza_types.pizza_type_id=pizzas.pizza_type_id
GROUP BY category
ORDER BY Total_quantity DESC;
```

	category	Total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



# Determine the distribution of orders by hour of the day:

```
SELECT hour(order_time) AS hour, count(order_id) AS Total_orders FROM orders  
GROUP BY hour(order_time);
```

	hour	Total_orders
▶	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



Join relevant tables to find the category-wise distribution of pizzas:

```
SELECT pizza_types.category, count(DISTINCT pizzas.pizza_id) AS Pizza_count
FROM pizza_types
JOIN pizzas
ON pizza_types.pizza_type_id=pizzas.pizza_type_id
GROUP BY category
ORDER BY Pizza_count DESC;
```



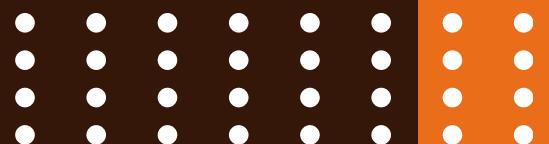
	category	pizza_count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



# Group the orders by date and calculate the average number of pizzas ordered per day:

```
SELECT date(orders.order_date) AS order_day, avg(order_details.quantity) AS num_of_pizza_per_day  
FROM orders  
JOIN order_details ON orders.order_id=order_details.order_id  
GROUP BY order_day
```

	order_day	num_of_pizza_per_day
▶	2015-01-01	1.0062
	2015-01-02	1.0313
	2015-01-03	1.0260
	2015-01-04	1.0000
	2015-01-05	1.0331
	2015-01-06	1.0208
	2015-01-07	1.0376
	2015-01-08	1.0117
	2015-01-09	1.0325
	2015-01-10	1.0069
	2015-01-11	1.0175
	2015-01-12	1.0085
	2015-01-13	1.0256
	2015-01-14	1.0417
	2015-01-15	1.0000
	2015-01-16	1.0194
	2015-01-17	1.0246
	2015-01-18	1.0252
	2015-01-19	1.0216
	2015-01-20	1.0288
	2015-01-21	1.0157



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

```
SELECT pizza_types.name, sum(order_details.quantity * pizzas.price) AS Total_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 total_revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

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



# 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_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  
ORDER BY revenue DESC
```



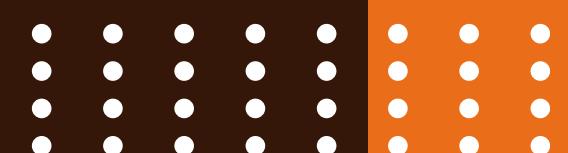
Result Grid | F

	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(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 sales
```

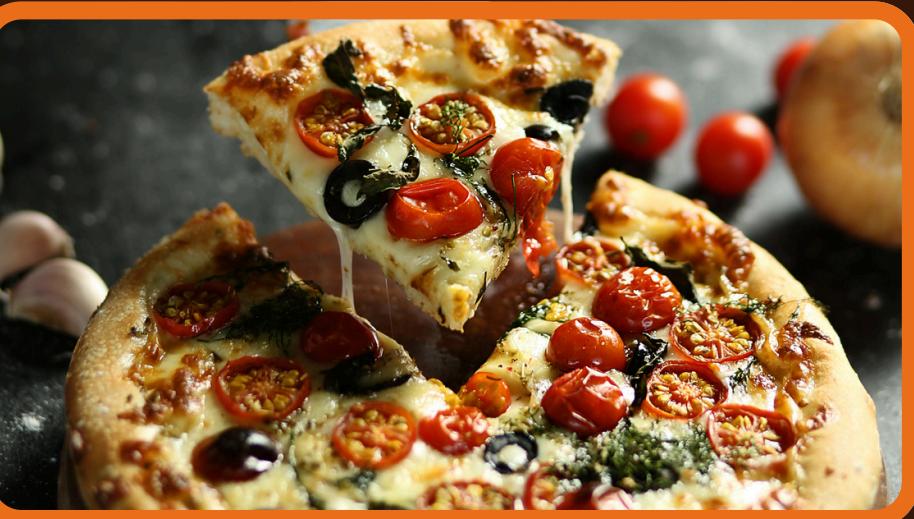
|   | order_date | cum_revenue        |
|---|------------|--------------------|
| ▶ | 2015-01-01 | 2713.8500000000004 |
|   | 2015-01-02 | 5445.75            |
|   | 2015-01-03 | 8108.15            |
|   | 2015-01-04 | 9863.6             |
|   | 2015-01-05 | 11929.55           |
|   | 2015-01-06 | 14358.5            |
|   | 2015-01-07 | 16560.7            |
|   | 2015-01-08 | 19399.05           |
|   | 2015-01-09 | 21526.4            |
|   | 2015-01-10 | 23990.350000000002 |
|   | 2015-01-11 | 25862.65           |
|   | 2015-01-12 | 27781.7            |
|   | 2015-01-13 | 29831.300000000003 |
|   | 2015-01-14 | 32358.700000000004 |
|   | 2015-01-15 | 34343.500000000001 |
|   | 2015-01-16 | 36937.650000000001 |



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

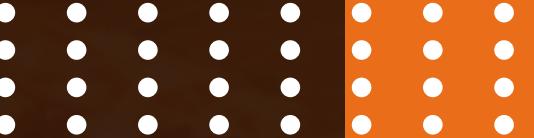
```
• SELECT name,revenue
  FROM
    (SELECT category,name,revenue,
    rank() over(PARTITION BY category ORDER BY revenue DESC) AS RN
  FROM
    (SELECT pizza_types.category,pizza_types.name,SUM(order_details.quantity * pizzas.price) AS revenue
  FROM pizza_types
  JOIN pizzas ON pizza_types.pizza_type_id=pizzas.pizza_type_id
  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;
```

| Result Grid |                              | Filter Rows:      | ⋮ |
|-------------|------------------------------|-------------------|---|
|             | 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           | ⋮ |

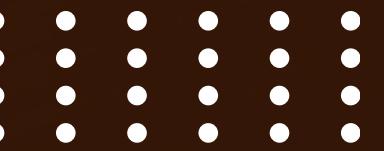


## Use Cases

- Identifying best-selling pizzas and peak sales hours.
- Understanding revenue contribution by pizza category.
- Analyzing order trends for better inventory management.



# THANK YOU FOR ATTENTION



See You Next

[www.linkedin.com/in/rachana-kakade](https://www.linkedin.com/in/rachana-kakade)