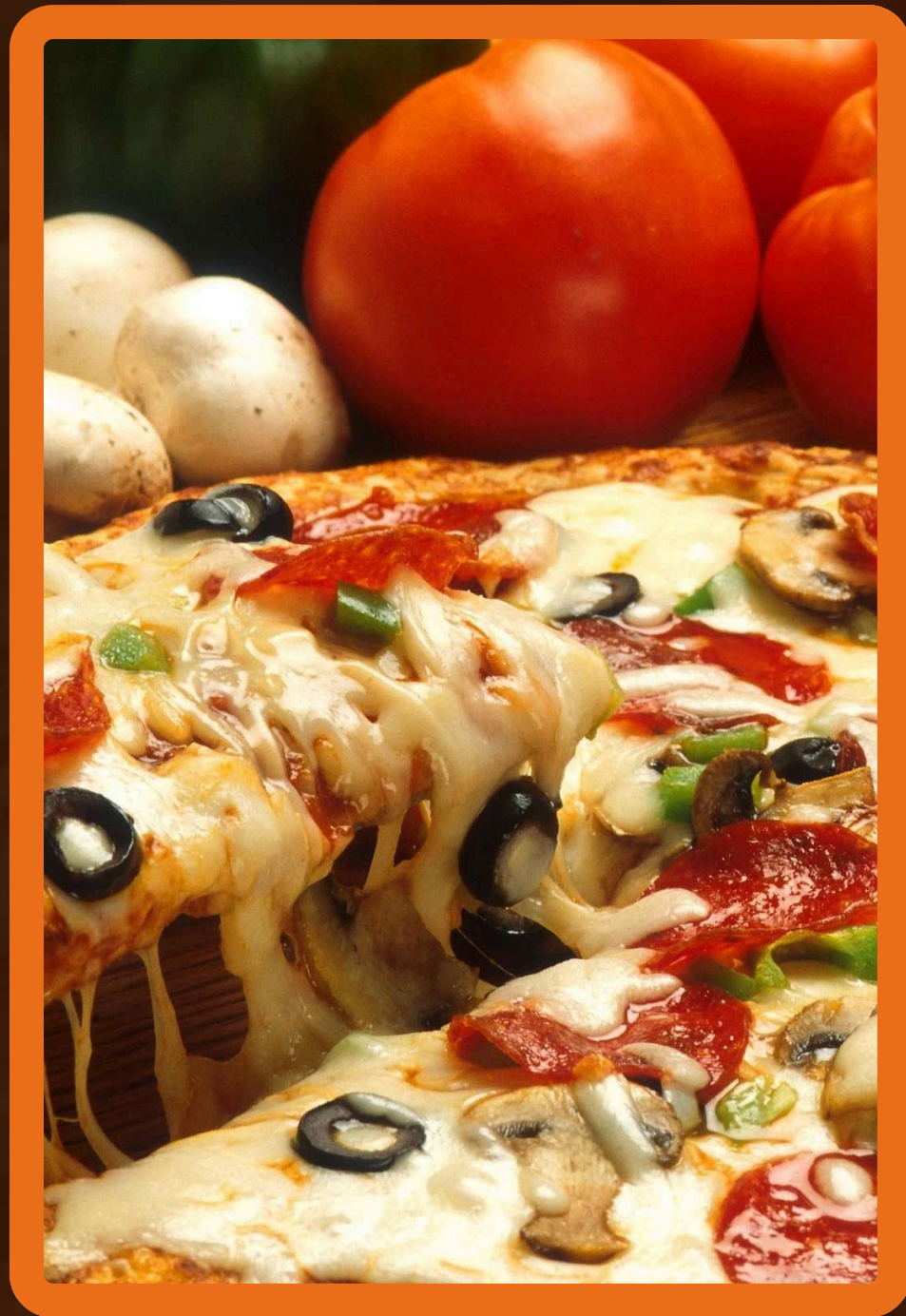


WELCOME TO SQL PROJECT ON PIZZA SALES

Presented By:
Jyoti Prakash





ABOUT THE PROJECT

In this project, I utilized SQL queries to analyze and address various questions related to pizza sales data.

QUESTIONS



- 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.
- 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.



QUESTIONS

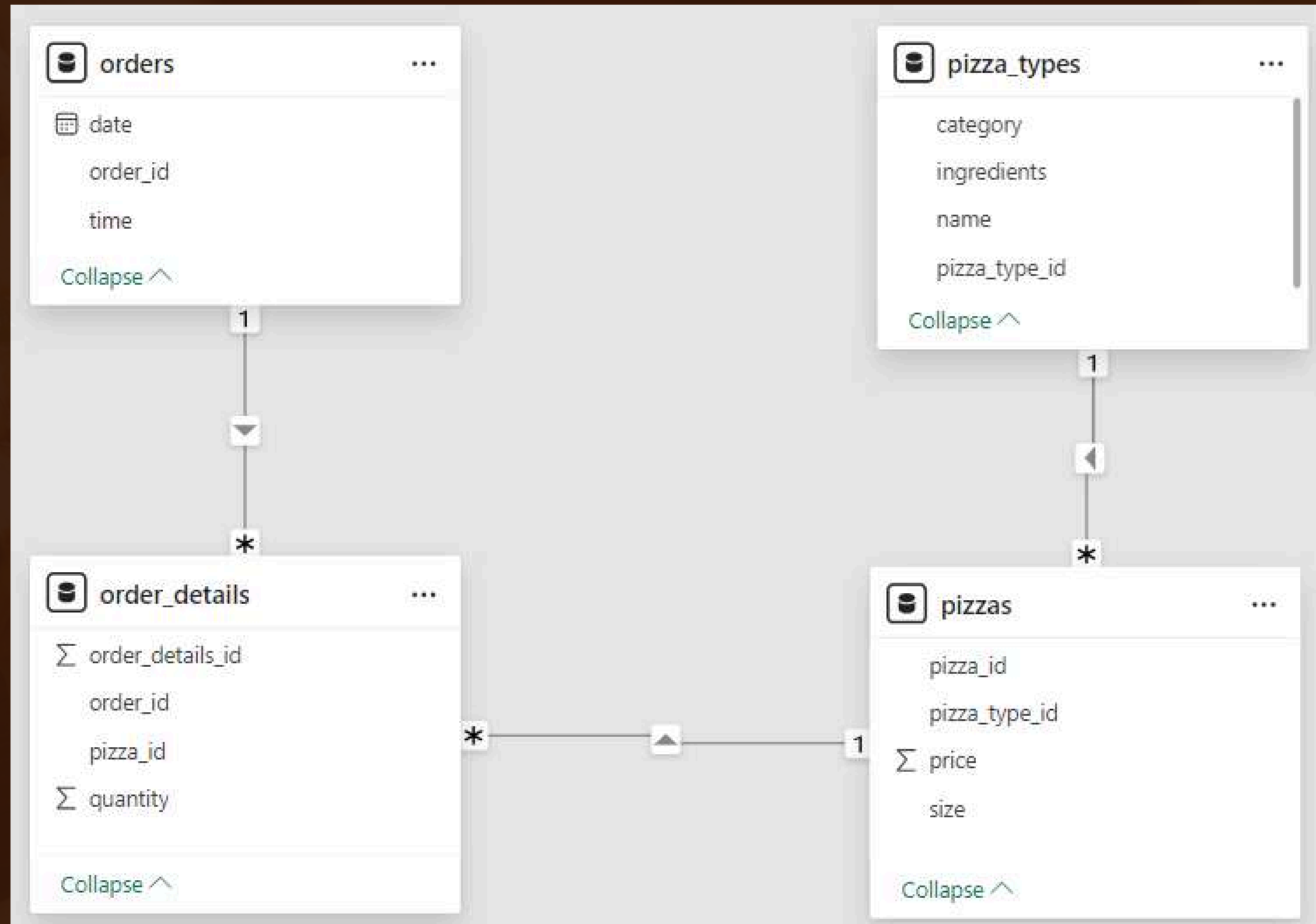


- Determine the top 3 most ordered pizza types based on revenue.
- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

*Tool
Used*



DATA MODEL



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED



Select
`count(order_id)` as Total_Orders
From
orders

Result Grid	
	Total_Orders
▶	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES



Result Grid	
	Total_Sales
▶	817860.05

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


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;
```

Result Grid			Filter Rows
	name	price	
▶	The Greek Pizza	35.95	



IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS total_count
FROM
    pizzas
    JOIN
        orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY total_count DESC
```



Result Grid			Filter Rows:
	name	price	
▶	The Greek Pizza	35.95	

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;



Result Grid			Filter Rows:
	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	

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 pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

Result Grid			Filter R
	category	quantity	
▶	Classic	14888	
	Supreme	11987	
	Veggie	11649	
	Chicken	11050	

DETERMINE THE DISTRIBUTION OF ORDERS : : : : : : BY HOUR OF THE DAY

```
SELECT
    HOUR(time), COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(time)
ORDER BY HOUR(time) DESC;
```



Result Grid			Filter Rows:
	HOUR(time)	order_count	
▶	23	28	
	22	663	
	21	1198	
	20	1642	
	19	2009	
	18	2399	
	17	2336	
	16	1920	
	15	1468	
	14	1472	
	13	2455	
	12	2520	
	11	1231	
	10	8	
	9	1	

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



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

Result Grid			Filter Rows
	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_order_per_day
FROM
    (SELECT
        orders.date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.date) AS order_quatity;
```



	Avg_order_per_day
▶	138

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE :::::

```
SELECT
    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.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

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
    order_details
JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
JOIN pizza_types ON pizzas.pizza_type_id =
    pizza_types.pizza_type_id
GROUP BY pizza_types.category
ORDER BY Revenue DESC;
```

	category	Revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME



```
Select date,  
ROUND(sum(revenue) over(order by date),2) as Cumilitive_Revenue  
from  
(Select orders.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 orders.order_id = order_details.order_id  
Group By orders.date) as Sales;
```

	date	Cumilitive_Revenue
▶	2015-01-01	2713.85
	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	23990.35
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.3

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,  
        ROUND(SUM(order_details.quantity * pizzas.price),  
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 pizza_types.category, pizza_types.name  
    ) AS revenue_summary) AS ranked_revenue  
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
	The Spicy Italian Pizza	34831.25
	The Italian Supreme Pizza	33476.75
	The Sicilian Pizza	30940.5
	The Four Cheese Pizza	32265.7
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5

Pizza Resto Presentation

THANK YOU
FOR ATTENTION

See You Next