



PIZZA SALES ANALYSIS



Data Analysis using SQL





Data Analysis using SQL

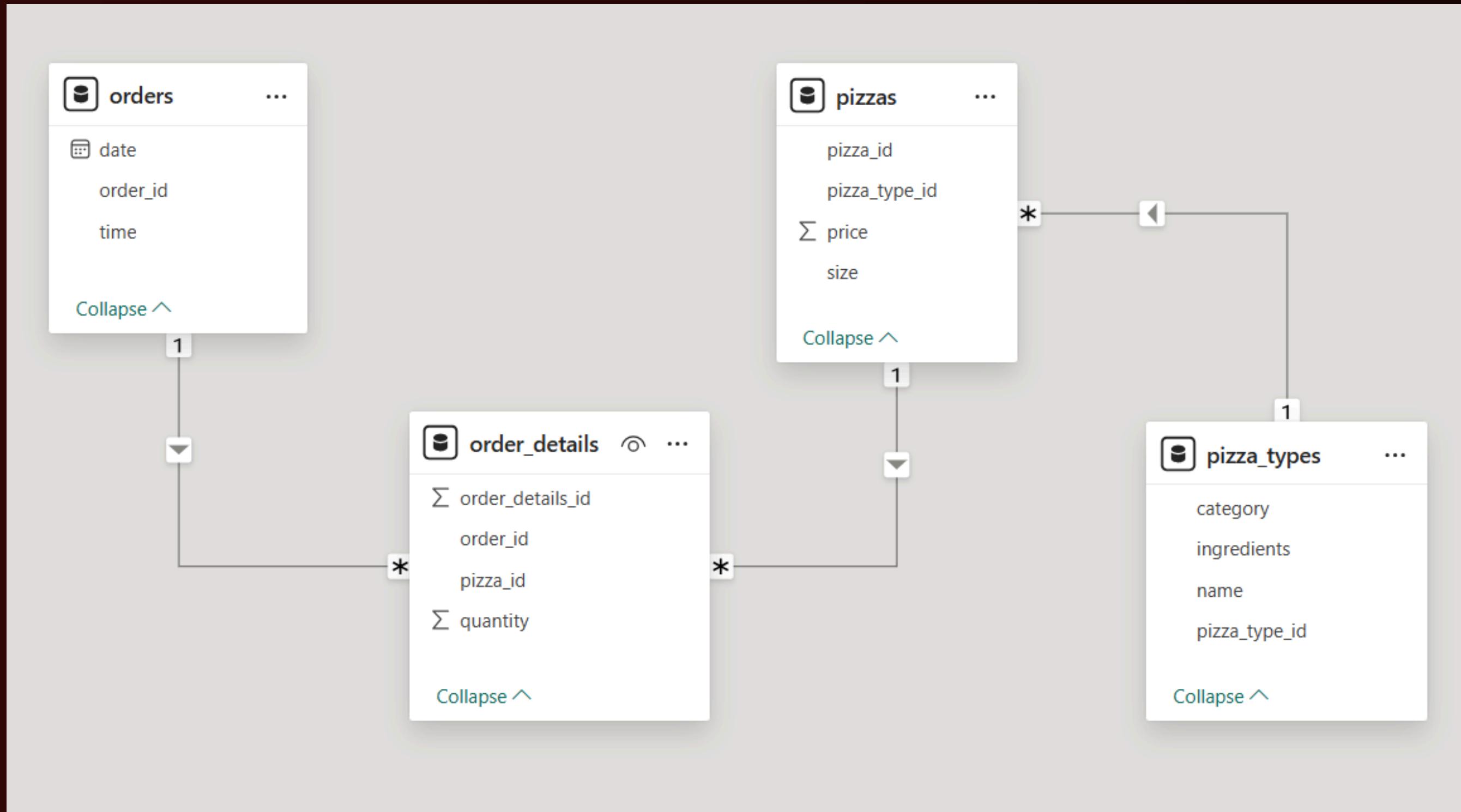


HELLO

MY NAME IS SHIVAM
AND I AM A DATA ANALYST

I HAVE MADE A PROJECT ON PIZZA SALES
DATA AND IN THIS PROJECT I HAVE UTILIZED
SQL QUERIES TO SOLVE QUESTIONS THAT
WERE RELATED TO PIZZA'S SALES.

Data Model



This data model follows star schema pattern where **order_details** serves as the **fact table**, holding transactional data such as pizza quantity and order references. The **dimension tables** such as **orders , pizzas and pizza_types** — provides a descriptive context like order date/time, pizza attributes like name, pizza types, size, price and pizza category details. This structure enables efficient analysis of product trends, sales performance, and time-based insights.



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.

```
SELECT  
    ROUND(SUM(quantity * price), 2) AS Total_sales  
FROM  
    order_details AS od  
    JOIN  
    pizzas AS p ON od.pizza_id = p.pizza_id;
```

Result Grid	
	Total_sales
▶	817860.05





Identify the highest-priced pizza.

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

	name	Highest_price_pizza
▶	The Greek Pizza	35.95





Identify the most common pizza size ordered.

```
SELECT
    p.size, COUNT(od.quantity) AS Most_ordered_size
FROM
    order_details AS od
        JOIN
    pizzas AS p ON od.pizza_id = p.pizza_id
GROUP BY p.size
ORDER BY most_ordered_size DESC;
```

Result Grid		
	size	Most_ordered_size
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28



List the top 5 most ordered pizza types along with their quantities.

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

	name	Most_ordered_pizza_type
▶	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  
    pt.category, SUM(od.quantity) AS Pizza_category_total  
FROM  
    pizza_types AS pt  
        JOIN  
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id  
        JOIN  
    order_details AS od ON od.pizza_id = p.pizza_id  
GROUP BY pt.category  
ORDER BY Pizza_category_total DESC;
```

	category	Pizza_category_total
▶	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 Order_count_per_hour  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

	hour	Order_count_per_hour
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009

	hour	Order_count_per_hour
	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  
    category, COUNT(name) as Category_wise_distribution  
FROM  
    pizza_types  
GROUP BY category;
```

Result Grid | Filter Rows:

	category	Category_wise_distribution
▶	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_pizza_ordered_per_day  
FROM  
    (SELECT  
        o.order_date, SUM(od.quantity) AS quantity  
    FROM  
        orders AS o  
    JOIN order_details AS od ON o.order_id = od.order_id  
    GROUP BY o.order_date) AS order_quantity;
```

	Avg_pizza_ordered_per_day
▶	138



Determine the top 3 most ordered pizza types based on revenue.

```
SELECT
    pt.name, SUM(od.quantity * p.price) AS revenue
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details AS od ON p.pizza_id = od.pizza_id
GROUP BY pt.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
    pt.category,
    ROUND((SUM(od.quantity * p.price) / (SELECT SUM(quantity * price)
    FROM order_details AS od
    JOIN pizzas AS p ON od.pizza_id = p.pizza_id)) * 100,2) AS Revenue
FROM
    pizza_types AS pt
    JOIN pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
    JOIN order_details AS od ON p.pizza_id = od.pizza_id
GROUP BY pt.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 order_date,sum(revenue) over (order by order_date) as Cumulative_revenue  
from  
(select o.order_date , round(sum(od.quantity * p.price),2) as revenue  
from orders as o  
join order_details as od  
on o.order_id = od.order_id  
join pizzas as p  
on od.pizza_id = p.pizza_id  
group by o.order_date ) as sales;
```

	order_date	Cumulative_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	21526.399999999998

	order_date	Cumulative_revenue
▶	2015-01-10	23990.35
	2015-01-11	25862.649999999998
	2015-01-12	27781.699999999997
	2015-01-13	29831.299999999996
	2015-01-14	32358.699999999997
	2015-01-15	34343.5
	2015-01-16	36937.65
	2015-01-17	39001.75
	2015-01-18	40978.6

	order_date	Cumulative_revenue
▶	2015-01-19	43365.75
	2015-01-20	45763.65
	2015-01-21	47804.200000000004
	2015-01-22	50300.9
	2015-01-23	52724.6
	2015-01-24	55013.85
	2015-01-25	56631.4
	2015-01-26	58515.8
	2015-01-27	61043.850000000006





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

```
Select name,category ,Revenue from
(select category ,name , revenue ,
rank() over (partition by category order by revenue desc) as Rank_
from
(select pt.category,pt.name,sum(od.quantity * p.price) as Revenue
from pizza_types as pt
join pizzas as p
on pt.pizza_type_id = p.pizza_type_id
join order_details as od
on od.pizza_id = p.pizza_id
group by pt.category ,pt.name) as pizza_type ) as Category_wise_revenue
where Rank_<=3 ;
```

	name	category	Revenue
▶	The Thai Chicken Pizza	Chicken	43434.25
	The Barbecue Chicken Pizza	Chicken	42768
	The California Chicken Pizza	Chicken	41409.5
	The Classic Deluxe Pizza	Classic	38180.5
	The Hawaiian Pizza	Classic	32273.25
	The Pepperoni Pizza	Classic	30161.75
	The Spicy Italian Pizza	Supreme	34831.25
	The Italian Supreme Pizza	Supreme	33476.75
	The Sicilian Pizza	Supreme	30940.5

	name	category	Revenue
	The Classic Deluxe Pizza	Classic	38180.5
	The Hawaiian Pizza	Classic	32273.25
	The Pepperoni Pizza	Classic	30161.75
	The Spicy Italian Pizza	Supreme	34831.25
	The Italian Supreme Pizza	Supreme	33476.75
	The Sicilian Pizza	Supreme	30940.5
	The Four Cheese Pizza	Veggie	32265.70000000065
	The Mexicana Pizza	Veggie	26780.75
	The Five Cheese Pizza	Veggie	26066.5



Home

About

Contact

THANK YOU