



**SQL
PROJECT
ON
PIZZA
SALES**

Bharati Khaire



I have utilized sql queries to analyze pizza sales

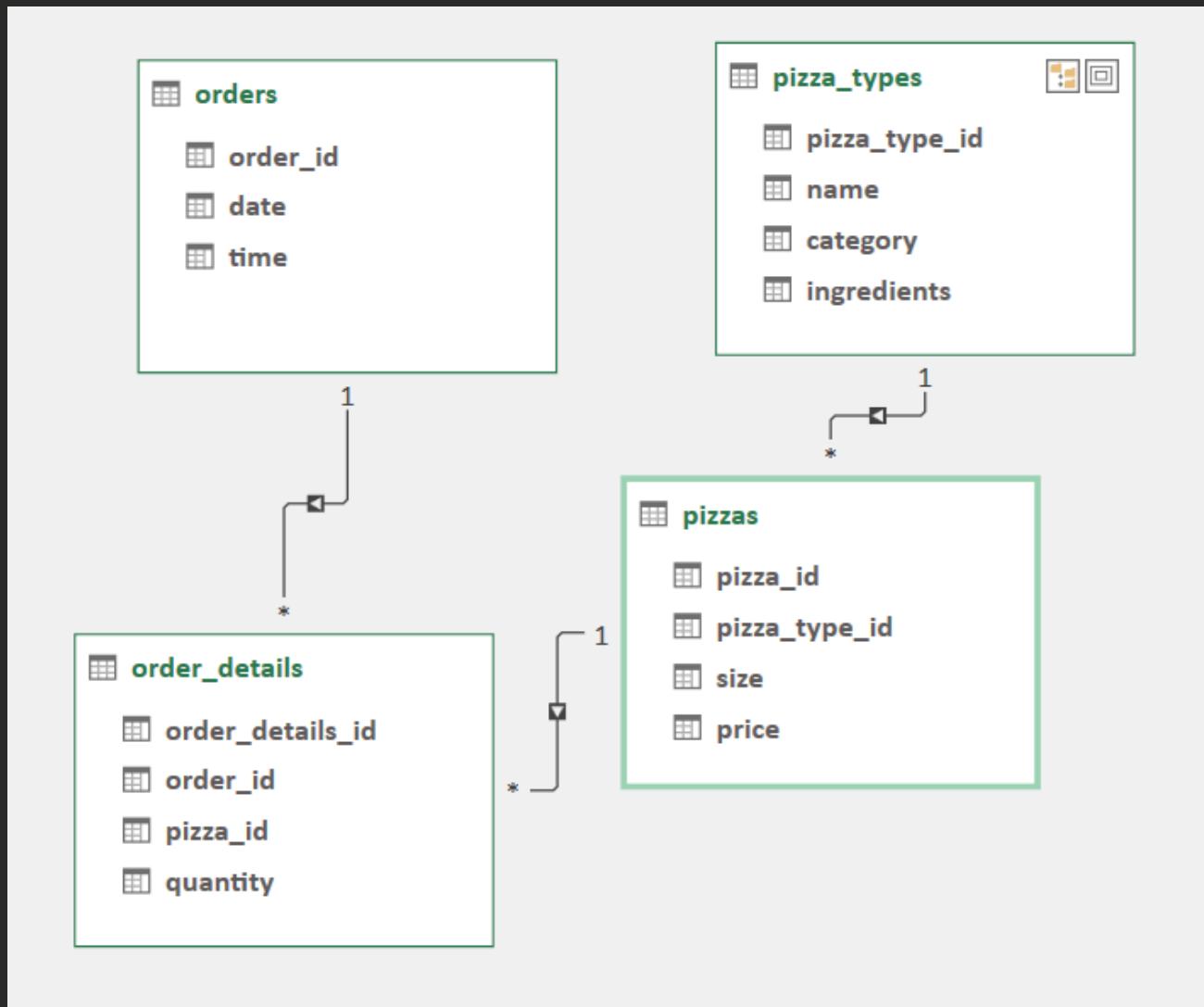
There are four tables -

1. Pizzas - pizza_id, Pizza_type, size, price
2. orders - order_id, order_date, order_time
3. pizza_types - pizza_type_id, name, category, ingredients
4. order_details - order_details, order_id, pizza_id, quantity

Topics Used-

Aggregate function, where, Group by,
Order by,
LIMIT, Join, Subqueries, CTE, Rank()







Questions

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. Top 5 most ordered pizza types along with their quantities
6. Join the necessary tables to find the total quantities of each pizza category ordered
7. Determine the distri. of orders by hour of the day
8. Find the category wise distribution of pizzas
9. Group the orders by date and calculate average number of pizzas ordered per day
10. Top 3 most ordered pizza types based on revenue
11. calculate the % contri. of each pizza type to total revenue
12. Analyze the cumulative revenue generated over time
13. 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  
FROM  
    orders;
```

Result Grid	
	Total_Orders
▶	21350



2. 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;
```

Result Grid	
	Total_Revenue
▶	817860.05



3. Identify the highest price pizza

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

Result Grid | Filter Row

	name	price
▶	The Greek Pizza	35.95



4. Identify the most common pizza size ordered

```
SELECT  
    Pizzas.size,  
    COUNT(orders_details.quantity) AS order_quantity  
FROM  
    Pizzas  
        JOIN  
    orders_details ON orders_details.pizza_id = pizzas.pizza_id  
GROUP BY size  
ORDER BY order_quantity DESC;
```

	size	order_quantity
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28



5. 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 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

```
SELECT  
    SUM(orders_details.quantity) AS Total_Quantity,  
    pizza_types.category  
FROM  
    orders_details  
        JOIN  
    pizzas ON orders_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;
```

Total_Quantity	category
14888	Classic
11987	Supreme
11649	Veggie
11050	Chicken



7. Determine the distribution of the orders by hour of the day

SELECT

HOUR(order_time) AS Hour, COUNT(order_id) AS order_count

FROM

orders

GROUP BY hour;

Hour	order_count
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



8.join the relevant tables to find the category wise distribution of pizzas

SELECT

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

category	pizza_types
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**

orders.order_date, SUM(orders_details.quantity) AS quantity

FROM

orders

JOIN orders_details **ON** orders_details.order_id = orders.order_id

GROUP BY order_date) AS order_quantity;

avg_pizza_ordered_per_day

138



10. Determine the top 3 most ordered pizza type based on revenue

```
SELECT  
    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 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
    pizza_types.category,
    round(SUM(orders_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(orders_details.quantity * pizzas.price),
        2) AS Total_sales
    )
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 category
ORDER BY revenue DESC;
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



12. Analyse the cumulative revenue generated over time

```
select order_date,round(sum(revenue) over(order by order_date),2) as cumulative_sales  
from  
  (SELECT  
    orders.order_date,  
    sum(orders_details.quantity * pizzas.price) AS revenue  
  FROM  
    orders  
  JOIN  
    orders_details ON orders.order_id = orders_details.order_id  
  JOIN  
    pizzas ON orders_details.pizza_id = pizzas.pizza_id  
  group by order_date) as sales;
```

order_date	cumulative_sales
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



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

```
select category, name, revenue from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn from
(select pizza_types.name, pizza_types.category,
sum(orders_details.quantity * pizzas.price) as Revenue 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 category, name) as a) as b
where rn <=3;
```

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5



Thank you

