



BOCELLE

R E S T A U R A N T

PIZZA SALES ANALYSIS USING SQL

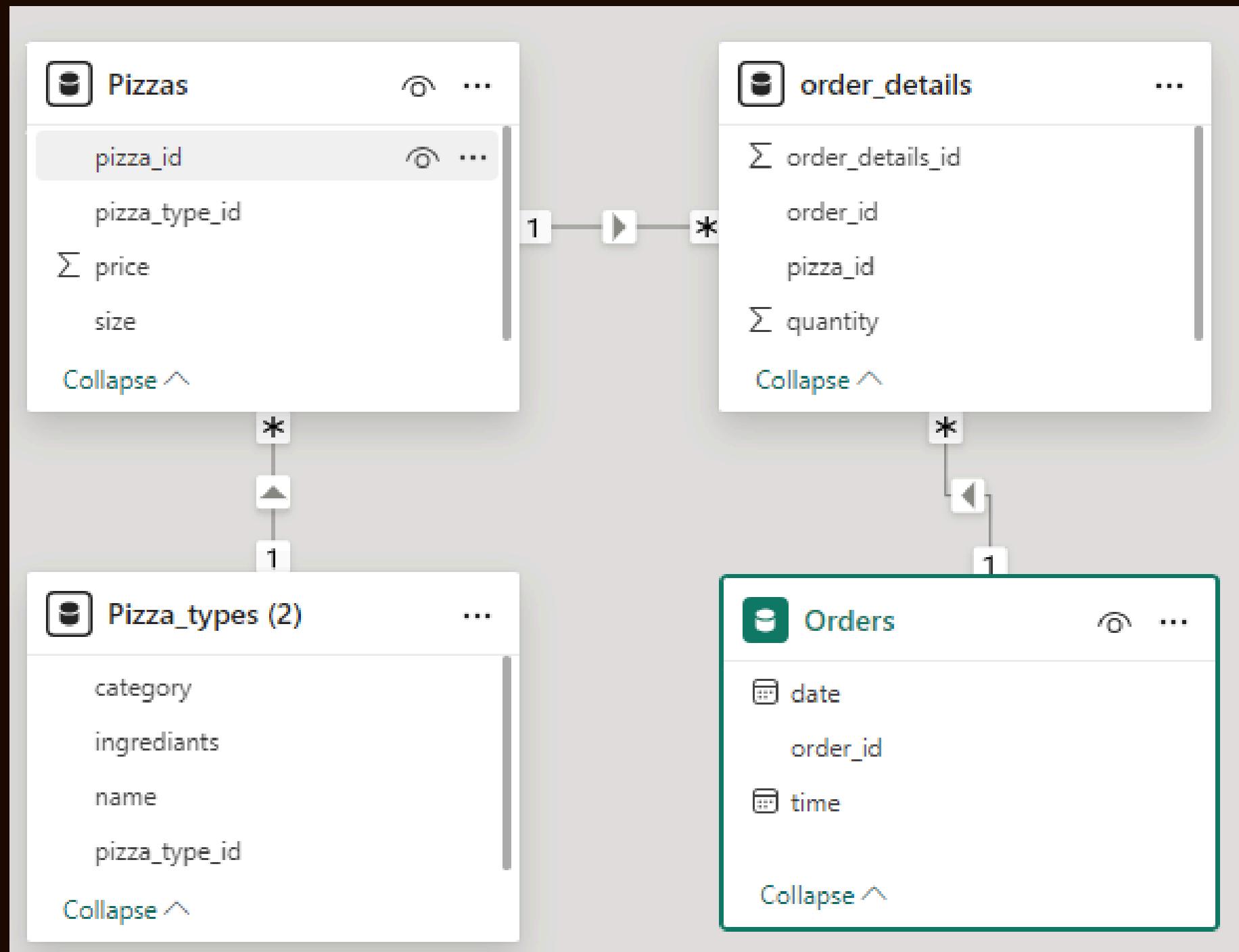


HELLO !

*MY name Tehmeed and in this project
I have utilize SQL queries to solve
questions related to Pizza Sales*



DATA MODEL :





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

```
SELECT
    category, SUM(quantity) AS Total_Ordered
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_detail ON order_detail.pizza_id = pizzas.pizza_id
GROUP BY category;
```

Result Grid | Filter Rows

	category	Total_Ordered
	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

#GROUP THE ORDERS BY DATE AND
CALCULATE THE
AVERAGE NUMBER OF PIZZAS
ORDERED PER DAY.

```
SELECT
    ROUND(AVG(quantity), 0) as Avg_Pizzas_per_day
FROM
    (SELECT
        order_date AS Order_date,
        SUM(order_detail.quantity) AS quantity
    FROM
        orders
    JOIN order_detail ON order_detail.order_id = orders.order_id
    GROUP BY Order_date) AS order_quantity;
```

Result Grid | F

Avg_Pizzas_per_day

138



#CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
SELECT
    category,
    ROUND(SUM(price * quantity) / (SELECT
        SUM(quantity * price)
    FROM
        order_detail
        JOIN
            pizzas ON order_detail.pizza_id = pizzas.pizza_id) * 100,
    2) AS Total_Revenue_Percentage
FROM
    pizza_types
    JOIN
        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
            order_detail ON order_detail.pizza_id = pizzas.pizza_id
GROUP BY category
```

	category	
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

-- ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
* select order_date,sum(revenue) over(order by order_date) as Cumulative_Revenue
  from
    (
      select orders.order_date,sum(price*quantity) as revenue
        from   orders
      join order_detail
        on orders.order_id=order_detail.order_id
      join pizzas
        on pizzas.pizza_id=order_detail.pizza_id
     group by order_date) as Cumulative
```

Result Grid | Filter Rows:

	order_date	Cumulative_Revenue
1	2015-01-01	2713.85
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.6
5	2015-01-05	11929.55
6	2015-01-06	14358.5
7	2015-01-07	16560.7
8	2015-01-08	19399.05
9	2015-01-09	21526.399999999998
10	2015-01-10	23990.35
11	2015-01-11	25862.649999999998
12	2015-01-12	27781.699999999997
13	2015-01-13	29831.299999999996
14	2015-01-14	32358.699999999997

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

hours	Orders
11	13336362
12	26929470
13	26615205
14	14867592
15	15634879
16	20551671
17	24312547
18	25808745
19	21634044
20	17668990
21	12868673
22	7269872
23	330700
10	73999
9	19176

```
SELECT  
    HOUR(order_time) AS hours, COUNT(order_id) AS Orders  
FROM  
    orders  
GROUP BY hours;  
  
select hour(order_time) as hours,sum(order_id) as Orders  
from orders  
group by hours;
```

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

```
SELECT
    name, SUM(quantity * price) AS revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
    order_detail ON order_detail.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



#CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

• `SELECT`

```
    ROUND(SUM(ord.quantity * pz.price), 2) AS Total_Revenue
```

`FROM`

```
order_detail AS ord
```

`JOIN`

```
pizzas AS pz ON ord.pizza_id = pz.pizza_id
```

Result Grid

Total_Revenue
817860.05



#DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
• select category,name,revenue,rank_
  from
  (select category,name,revenue, Rank()over(partition by category order by revenue desc) as rank_ from
  (
  select category,name,sum(price*quantity)as revenue
  from pizzas
  join order_detail
  on pizzas.pizza_id=order_detail.pizza_id
  join pizza_types
  on pizza_types.pizza_type_id=pizzas.pizza_type_id
  group by category,name) as tp
  ) as tq
  where rank_<=3;
```

	category	name	revenue	rank_
	Chicken	The Thai Chicken Pizza	43434.25	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41409.5	3
	Classic	The Classic Deluxe Pizza	38180.5	1
	Classic	The Hawaiian Pizza	32273.25	2



THANK YOU

