



*The tastiest pizza  
on the planet*



**PIZZA SALES INSIGHTS**



# ABOUT THE PROJECT

Hello , I am Neetish Mishra currently working as a Data Analyst in Coforge LTD.

In this project I used SQL queries to convert the Pizza sales data into actionable insights .

We solve few question that are related to sales Insights using SQL.



# SCHEMA

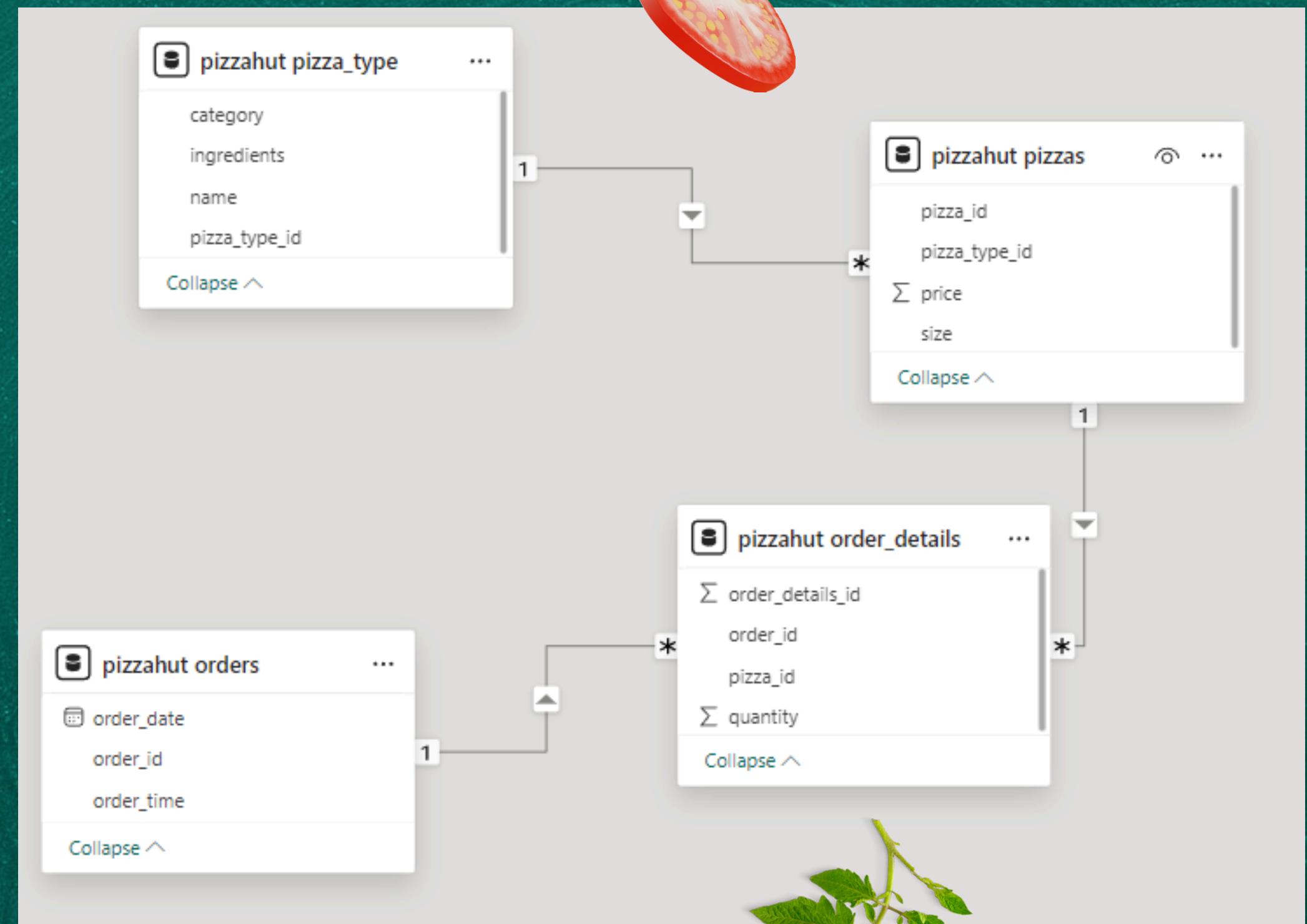
Tables Used

PIZZA\_TYPE

PIZZAS

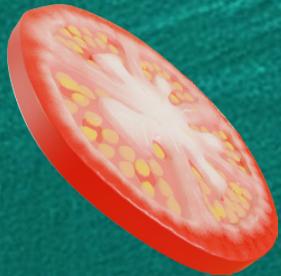
ORDER\_DETAILS

ORDERS





# Q1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



```
SELECT  
    COUNT(order_id) AS total_order  
FROM  
    orders;
```

total_order
21350



## Q2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

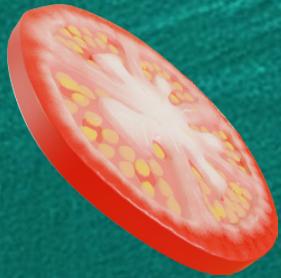
**SELECT**  
ROUND(SUM(order\_details.quantity \* pizzas.price),  
02) AS total\_sales  
**FROM**  
order\_details  
**JOIN**  
pizzas  
**WHERE**  
order\_details.pizza\_id = pizzas.pizza\_id;



	<b>total_sales</b>
▶	817860.05



# Q3. IDENTIFY THE HIGHEST-PRICED PIZZA



```
SELECT
    pizza_type.name, pizzas.price
FROM
    pizzas
        JOIN
    pizza_type
WHERE
    pizza_type.pizza_type_id = pizzas.pizza_type_id
ORDER BY price DESC
LIMIT 5;
```



	name	price
▶	The Greek Pizza	35.95
	The Greek Pizza	25.5
	The Brie Carre Pizza	23.65
	The Italian Vegetables Pizza	21
	The Barbecue Chicken Pizza	20.75



## Q4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT pizzas.size, COUNT(order_details_id)
FROM order_details
JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size;
```

size	count(order_details_id)
M	15385
L	18526
S	14137
XL	544
XXL	28



# Q5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.



```
SELECT pizza_type.name, sum(order_details.quantity) as quantity  
FROM pizza_type  
JOIN pizzas ON pizza_type.pizza_type_id=pizzas.pizza_type_id  
JOIN order_details ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_type.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



# Q6. JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.



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

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



# Q7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.



```
SELECT  
    HOUR(order_time) AS hour, COUNT(orders.order_id)  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

hour	count(orders.order_id)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009



# Q8. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.



```
select category, count(name) from pizza_type group by category;
```

category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9



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



```
SELECT  
    AVG(quantity)  
FROM  
    (SELECT  
        SUM(order_details.quantity) as quantity, orders.order_date  
    FROM  
        orders  
    JOIN order_details ON orders.order_id = order_details.order_id  
    GROUP BY order_date)as order_quantity;
```

Avg(quantity)

138.4749



# Q10 . DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

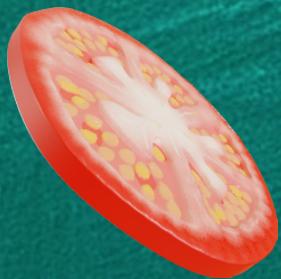


```
SELECT
    pizza_type.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_type
        JOIN
    pizzas ON pizza_type.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_type.name
ORDER BY revenue DESC
LIMIT 3;
```

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5



# Q 11. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.



```
SELECT pizza_type.category,
       ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                                                               ROUND(SUM(order_details.quantity * pizzas.price),
                                                               02) AS total_sales
                                                       FROM
                                                       order_details
                                                       JOIN
                                                       pizzas
                                                       WHERE
                                                       order_details.pizza_id = pizzas.pizza_id)*100,
       2) AS revenue
  FROM
    pizza_type
   JOIN
    pizzas ON pizza_type.pizza_type_id = pizzas.pizza_type_id
   JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
 GROUP BY pizza_type.category
 ORDER BY revenue DESC;
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



# Q12. 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 orders.order_id = order_details.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



# Q 13. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

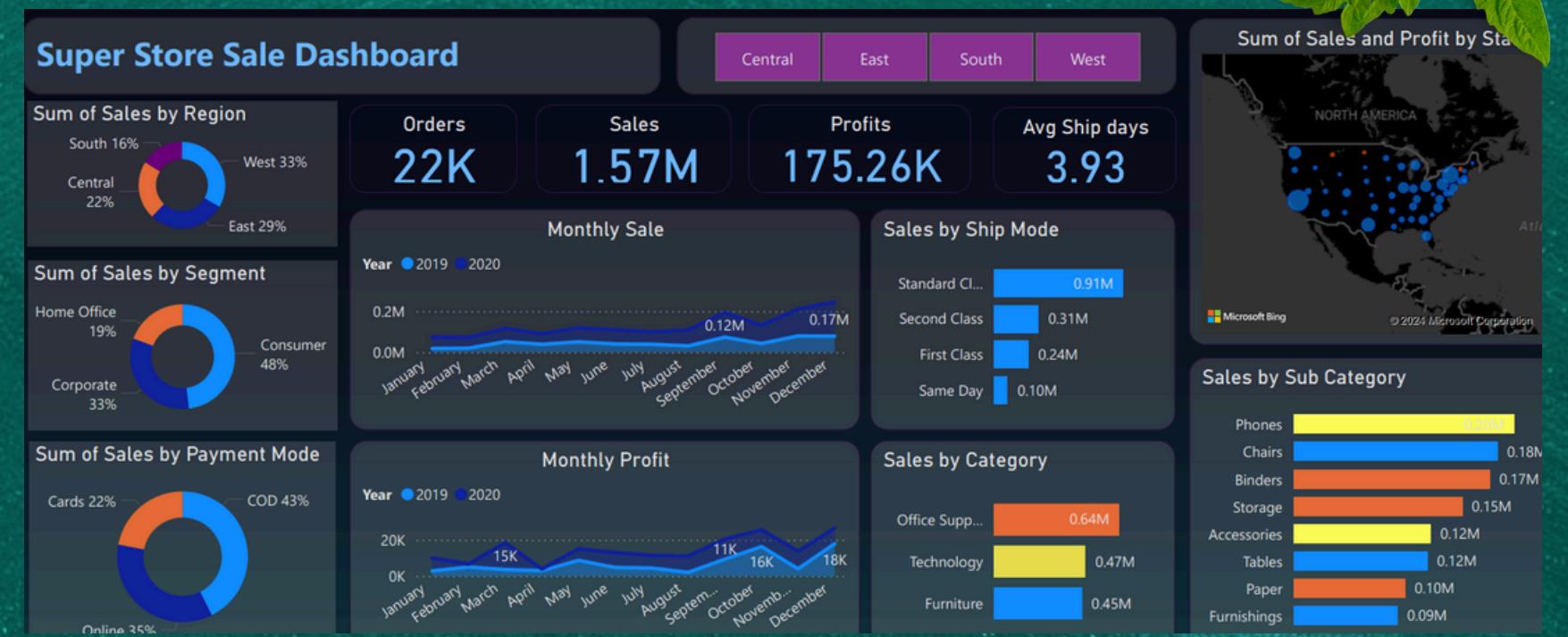


```
① select category , name, revenue from (select category,name,revenue,
② rank() over (partition by category order by revenue desc) as rn from(SELECT
    pizza_type.category,
    pizza_type.name,
    SUM(order_details.quantity * pizzas.price) as revenue
  FROM
    order_details
    JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
    JOIN
    pizza_type ON pizza_type.pizza_type_id = pizzas.pizza_type_id
  - GROUP BY pizza_type.category , pizza_type.name) as basic) as final
  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



# OTHER POWERBI PROJECT





## CONTACT :



<https://www.linkedin.com/in/neetish-mishra/>

neetishmishra.career@gmail.com

<https://github.com/nkongit>