

# Pizza Sales report



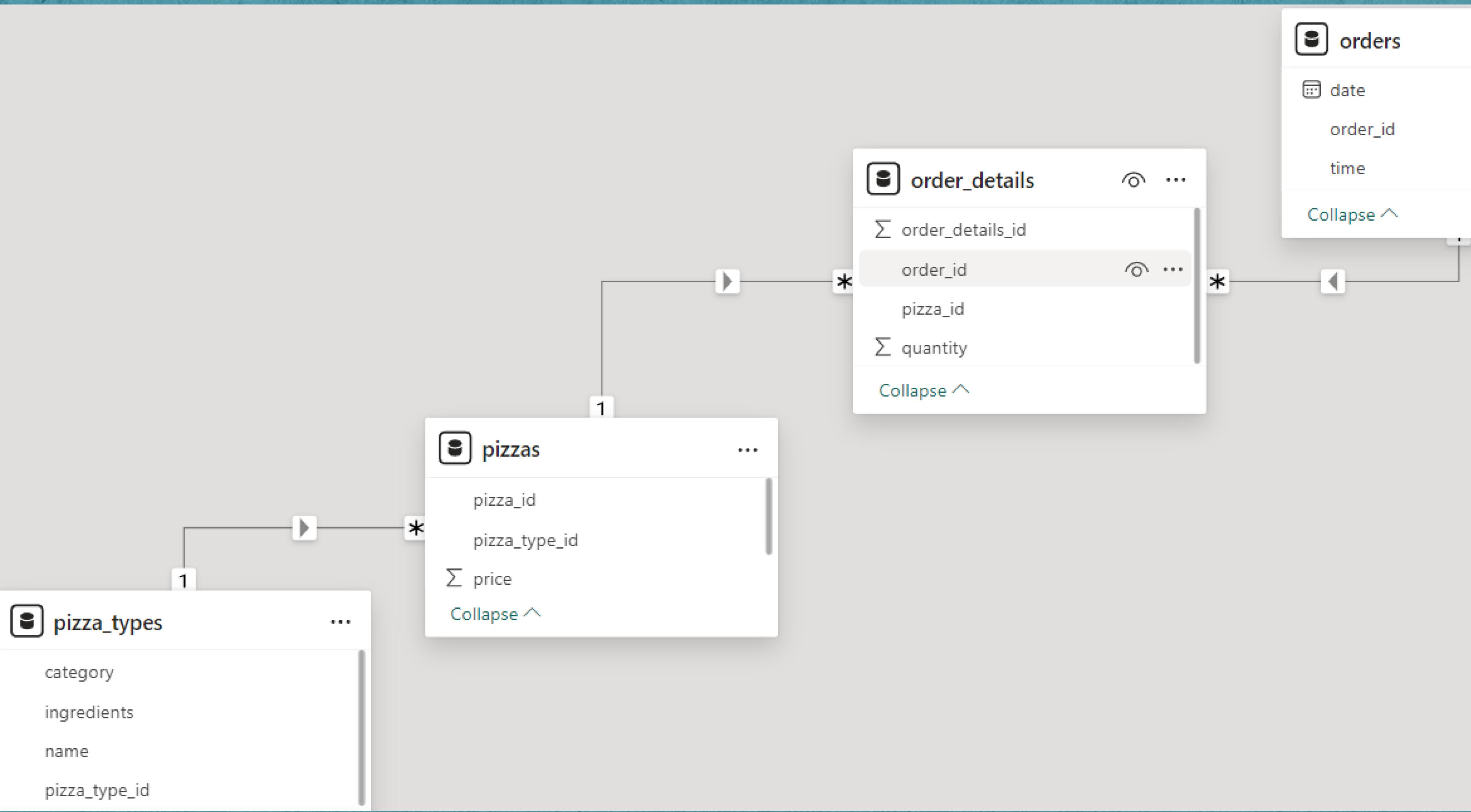
**Surendra Thapa**

# *HELLO PIZZA LOVERS*

Pizza have been delivering the pizza across the country for almost a decade and quite happy to see customer enjoying their pizzas.

# Project

what i really have done is collect the pizzahut data and foundout the valuable insights using the queries in mysql



# RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

## Query

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

## Result

	total_orders
▶	21350

# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

## Query

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

## Result

	total_revenue
▶	817860.05

# IDENTIFY THE HIGHEST-PRICED PIZZA.

## Query

```
SELECT
```

```
    p.name, ps.price
```

```
FROM
```

```
pizza_types p
```

```
    INNER JOIN
```

```
pizzas ps ON p.pizza_type_id = ps.pizza_type_id
```

```
ORDER BY ps.price DESC
```

```
LIMIT 1;
```

## Result

name	price
The Greek Pizza	35.95

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

## Query

```
SELECT  
    p.size, count(*) as number_of_ordersize  
FROM  
    orders_details o  
        INNER JOIN  
    pizzas p ON o.pizza_id = p.pizza_id  
group by p.size  
ORDER BY count(*) DESC;
```

## Result

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

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

## Query

```
SELECT pi.category,  
       SUM(o.quantity) AS total_quantity  
  FROM pizza_types pi  
 INNER JOIN pizzas p  
    ON pi.pizza_type_id = p.pizza_type_id  
 INNER JOIN orders_details o  
    ON o.pizza_id = p.pizza_id  
 GROUP BY pi.category  
 ORDER BY total_quantity DESC;
```

## Result

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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

## Query

```
SELECT *
FROM orders;
SELECT HOUR(order_time),
COUNT(*) AS orderperhour
FROM
orders
GROUP BY HOUR(order_time)
order by 1 ;
```

## Result

HOUR(order_time)	orderperhour
9	1
10	8
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920

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

## Query

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

## Result

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

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

## Query

```
SELECT
    AVG(x) AS average_per_day
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS x
    FROM
        orders
    JOIN orders_details
    ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS total_quantity
```

## Result

average_per_day
138.4749

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

## Query

```
SELECT pi.name,  
SUM(o.quantity * p.price) AS revenue  
FROM pizza_types pi  
INNER JOIN pizzas p  
ON pi.pizza_type_id = p.pizza_type_id  
INNER JOIN  
orders_details o ON o.pizza_id = p.pizza_id  
GROUP BY pi.name  
ORDER BY revenue DESC  
LIMIT 3;
```

## Result

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.

## Query

```
select pi.category,  
sum(o.quantity*p.price)*100/(select sum(quantity*price)  
from orders_details  
inner join pizzas on  
orders_details.pizza_id = pizzas.pizza_id) as contribution  
from pizza_types pi  
inner join pizzas p  
on pi.pizza_type_id = p.pizza_type_id  
inner join orders_details o  
on o.pizza_id = p.pizza_id  
group by pi.category;
```

## Result

category	contribution
Classic	26.9059602556699
Veggie	23.682590927384783
Supreme	25.45631126009884
Chicken	23.955137556847493

# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

## Query

```
select order_date,  
sum(revenue) over(order by order_date) as cumulative_revenue  
from  
(SELECT  
    o.order_date, SUM(od.quantity * p.price) AS revenue  
FROM  
    orders_details od  
        INNER JOIN  
    pizzas p ON od.pizza_id = p.pizza_id  
        INNER JOIN  
    orders o ON o.order_id = od.order_id  
GROUP BY o.order_date  
ORDER BY o.order_date) as total_rev
```

## Result

order_date	cumulative_revenue
2015-01-01	2713.850000000004
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

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

## Query

```
with pizza_category as(select pi.category,pi.name,  
sum(o.quantity*p.price) as revenue  
from pizza_types pi  
inner join pizzas p  
on pi.pizza_type_id = p.pizza_type_id  
inner join orders_details o  
on o.pizza_id = p.pizza_id  
group by pi.category,pi.name  
,  
pizza_rank as(  
select * ,  
dense_rank() over (partition by category order by revenue desc)  
as rank_pizza  
from pizza_category  
)  
select category,name,revenue  
from pizza_rank  
where rank_pizza <= 3
```

## Result

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

# Thank you!



*Surendra Thapa*