

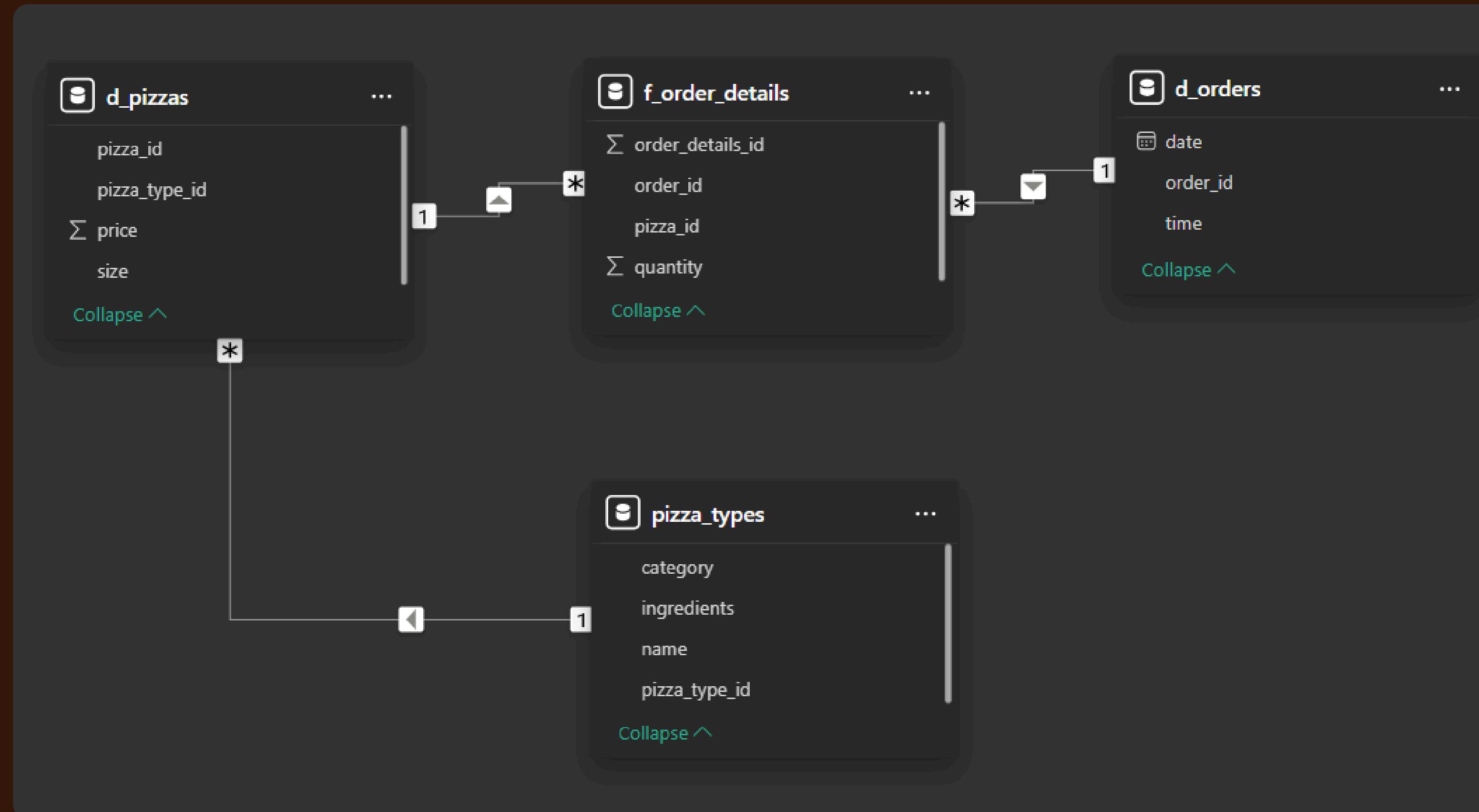
# SQL PROJECT ON PIZZA SALES



# HELLO!

My name is Gurkamal Singh.  
In this project, I have utilize SQL queries  
to solve a questions that were related to  
pizza sales

# DATA MODEL VIEW



# 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. List the top 5 most ordered pizza types along with their quantities.
6. Join the necessary tables to find the total quantity of each pizza category ordered.
7. Determine the distribution of orders by hour of the day.
8. Join relevant tables to find the category-wise distribution of pizzas.
9. Group the orders by date and calculate the average number of pizzas ordered per day.
10. Determine the top 3 most ordered pizza types based on revenue.
11. Calculate the percentage contribution of each pizza type to total revenue.
12. Analyze the cumulative revenue generated over time.
13. Determine the 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;
```

total_orders
21350

# Calculate the total revenue generated from pizza sales

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

toatal_sales
79169678.00

# Identify the highest-priced pizza.

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

name	price
The Greek Pizza	36

# Identify the most common pizza size ordered

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

size	order_count
S	1555840

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

SELECT

```
pizza_types.name, SUM(order_details.quantity) AS quantity
```

FROM

```
pizza_types
```

JOIN

```
pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
```

JOIN

```
order_details ON order_details.pizza_id = pizzas.pizza_id
```

GROUP BY pizza\_types.name

ORDER BY quantity DESC

LIMIT 5;

name	quantity
The Greek Pizza	247870
The Barbecue Chicken Pizza	148722
The California Chicken Pizza	148722
The Chicken Alfredo Pizza	148722
The Chicken Pesto Pizza	148722

# Join the necessary tables to find the total quantity of each pizza category ordered

SELECT

```
    pizza_types.category,  
    SUM(order_details.quantity) AS quantity  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
        JOIN  
    order_details ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.category  
ORDER BY quantity DESC;
```

category	quantity
Veggie	1338498
Classic	1288924
Supreme	1239350
Chicken	892332

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

SELECT

```
HOUR(order_time) AS hours, COUNT(order_id) AS order_count
```

FROM

```
orders
```

```
GROUP BY HOUR(order_time)
```

```
ORDER BY order_count ASC;
```

hours	order_count
9	1
10	8
23	28
22	663
21	1198
11	1231
15	1468
14	1472

# Join relevant tables to find the category-wise distribution of pizzas

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

Category	COUNT(pizza_types.name)
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 pizza_order_perday  
FROM  
    (SELECT  
        orders.order_date, SUM(order_details.quantity) AS quantity  
    FROM  
        orders  
    JOIN order_details ON orders.order_id = order_details.order_id  
    GROUP BY orders.order_date) AS order_count;
```

pizza_order_perday
138

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

SELECT

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

name	revenue
The Brie Carre Pizza	1189776
The Pepperoni Pizza	1883812
The Hawaiian Pizza	2032534

# Calculate the percentage contribution of each pizza type to total revenue

```
select pizza_types.category,  
round(SUM(order_details.quantity * pizzas.price)/(select  
round(SUM(order_details.quantity * pizzas.price),2) AS sales  
from order_details join pizzas  
on pizzas.pizza_id = order_details.pizza_id)*100,2) as revenue  
  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join order_details  
on order_details.pizza_id = pizzas.pizza_id  
group by pizza_types.category order by revenue desc;
```

category	revenue
Veggie	27.24
Classic	26.93
Supreme	26.68
Chicken	19.16

# 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	258714
2015-01-02	522219
2015-01-03	774545
2015-01-04	943827
2015-01-05	1143452
2015-01-06	1378211
2015-01-07	1598597
2015-01-08	1874878



# THANK YOU