



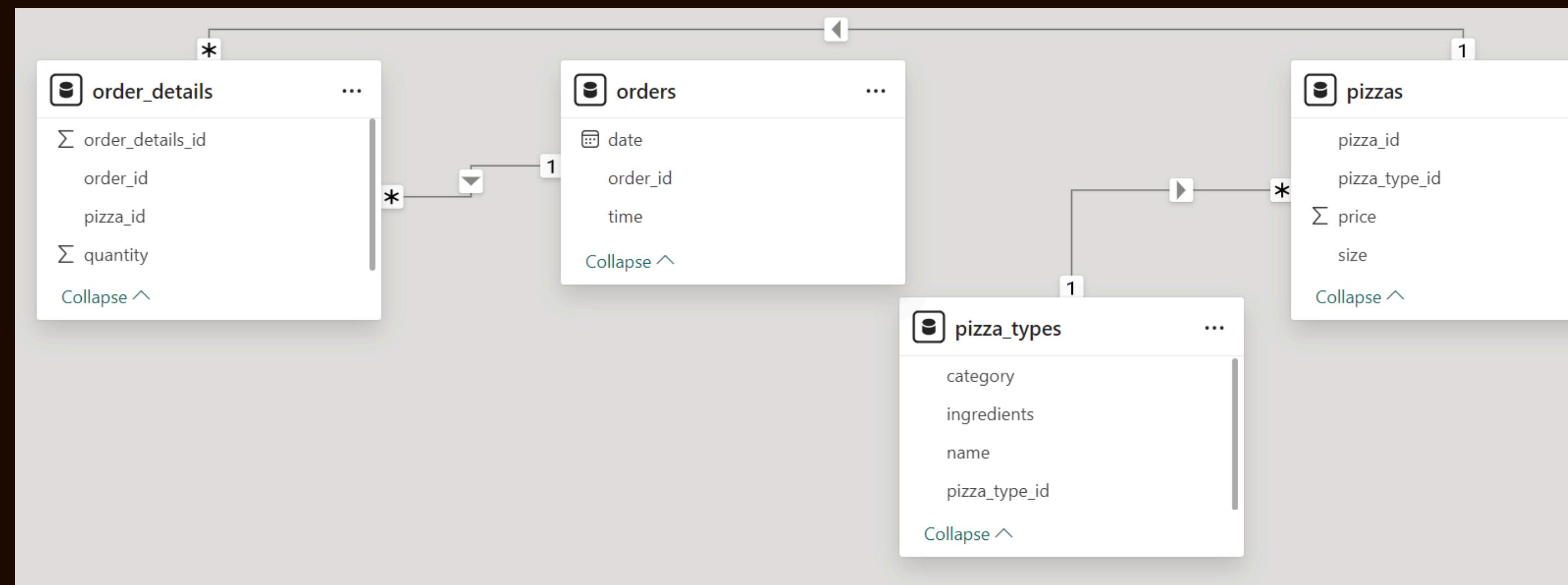
A N A L Y S I S O F
PIZZA SALES





INTRODUCTION

This analysis is about an year's pizza sales of a pizza store and is done to know how the sales has done in the current year which gives insights about how to proceed in the next year and the analysis is done in SQL language. Data is in the form of four files and are connected as shown.



QUESTIONS TO ANSWER

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.



QUESTIONS TO ANSWER

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



ANALYSIS

1. Retrieve the total number of orders placed.

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

total_orders
21350

ANALYSIS

2. Calculate the total revenue generated from pizza sales.

```
SELECT
```

```
    ROUND(SUM(order_details.quantity * pizzas.price),  
          2) AS total_sales
```

```
FROM
```

```
order_details
```

```
JOIN
```

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

	total_sales
▶	817860.05

ANALYSIS

3. Identify the highest-priced pizza.

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

name	price
The Greek Pizza	35.95



ANALYSIS

4. Identify the most common pizza size ordered.

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

most_common_pizza_size	count
L	18526
M	15385
S	14137
XL	544
XXL	28



ANALYSIS

5. 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 Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371



ANALYSIS

- 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
Classic	14388
Supreme	11987
Veggie	11649
Chicken	11050



ANALYSIS

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

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

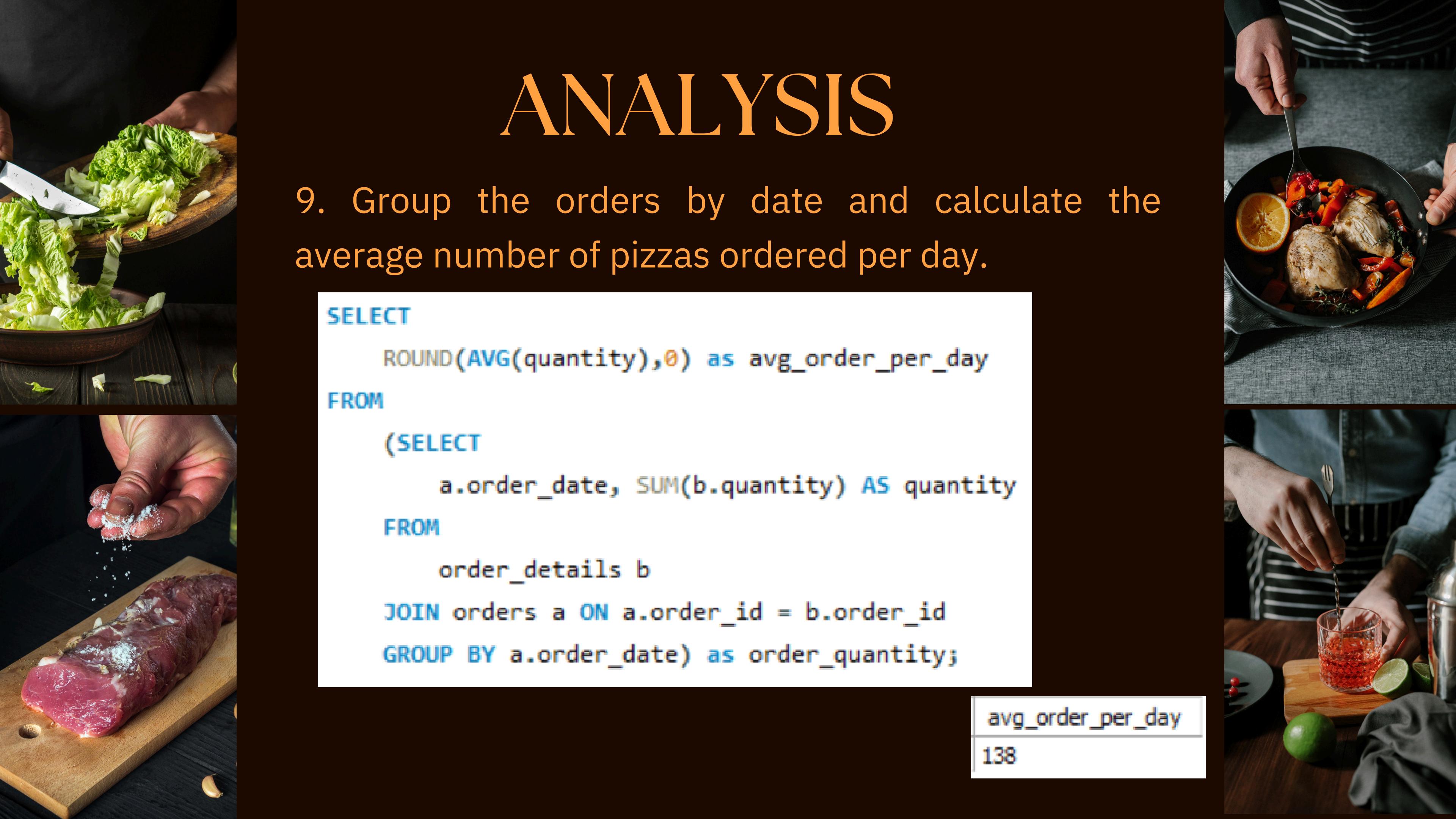
hour	COUNT(order_id)
9	1
10	8
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663

ANALYSIS

8. Find the category-wise distribution of pizzas.

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

category	COUNT(category)
Chicken	6
Classic	8
Supreme	9
Veggie	9



ANALYSIS

9. Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT  
    ROUND(AVG(quantity),0) as avg_order_per_day  
FROM  
(SELECT  
    a.order_date, SUM(b.quantity) AS quantity  
FROM  
    order_details b  
JOIN orders a ON a.order_id = b.order_id  
GROUP BY a.order_date) as order_quantity;
```

avg_order_per_day
138



ANALYSIS

10. 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 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 revenue DESC  
LIMIT 3;
```

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



ANALYSIS

11. 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 total_sales
    )
    FROM
        order_details
        JOIN
            pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
    2) AS revenue_percentage
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_percentage DESC;
```

category	revenue_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYSIS

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

```
SELECT
  *
FROM
  (
    SELECT
      category, name, revenue, RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS rank_of_revenue
    FROM
      (
        SELECT
          pizza_types.category, pizza_types.name, ROUND(SUM(order_details.quantity * pizzas.price)
            ,0) 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, pizza_types.name) AS category_name_revenue) AS top_3_pizza_category_revenue
WHERE
  rank_of_revenue <= 3;
```

category	name	revenue	rank_of_revenue
Chicken	The Thai Chicken Pizza	43434	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41410	3
Classic	The Classic Deluxe Pizza	38180	1
Classic	The Hawaiian Pizza	32273	2
Classic	The Pepperoni Pizza	30162	3
Supreme	The Spicy Italian Pizza	34831	1
Supreme	The Italian Supreme Pizza	33477	2
Supreme	The Sicilian Pizza	30940	3
Veggie	The Four Cheese Pizza	32266	1
Veggie	The Mexicana Pizza	26781	2
Veggie	The Five Cheese Pizza	26066	3



CONCLUSION

After having rigorous discussions with the stakeholders, questions were framed and the answers to it were extracted from data, which helped the clients to take further decisions on planning the next year for the pizza store

THANK YOU

