Pizza Data Analysis Report

Objective:

The primary goal of this project is to analyze pizza consumption trends and preferences using SQL queries applied to a dataset containing information about pizza orders.

Data Schema Overview

The dataset has 4 tables.

1. Pizzas

- Contains details about pizza size and price

2. pizza_types

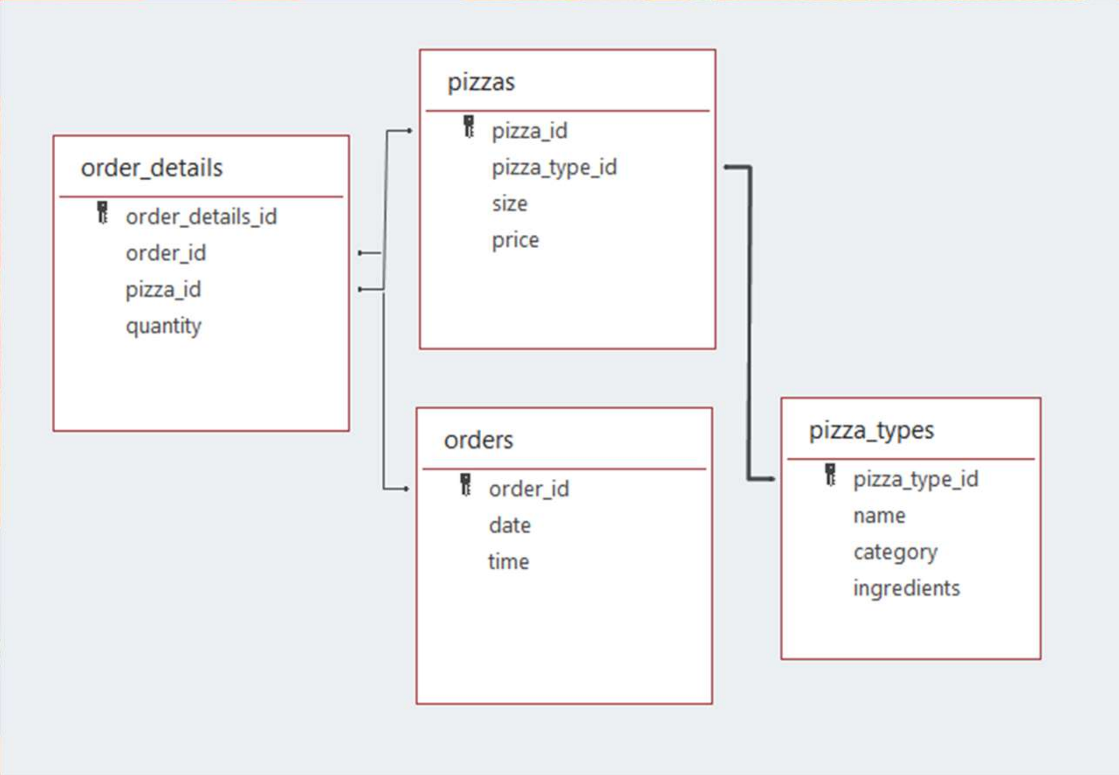
- Contains details about pizza type name,pizza category and pizza's ingredients

3.Orders

- Contains ordered date and time information from 01/01/2015 – 31/12/2015

4.order_details

- Contains information about ordered pizza quantity



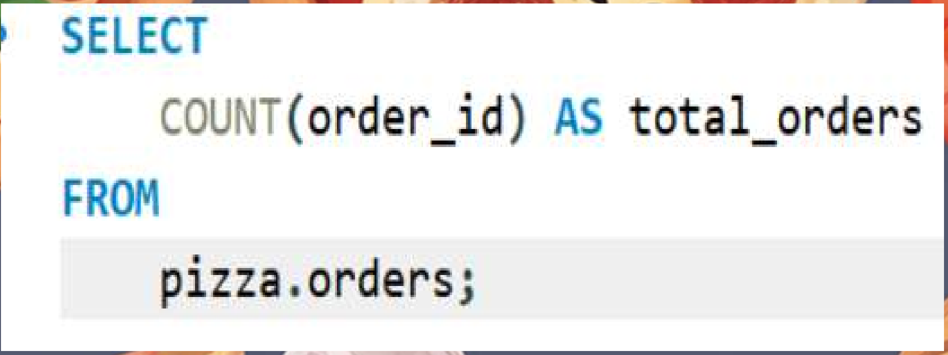
order_details
- order_details_id
- order_id
- pizza_id
- quantity

pizzas
- pizza_id
- pizza_type_id
- size
- price

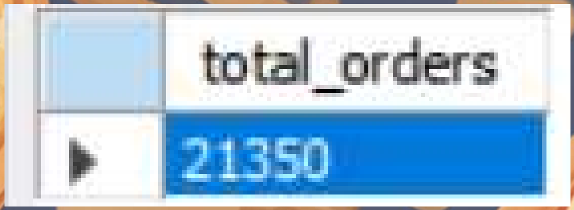
orders
- order_id
- date
- time

pizza_types
- pizza_type_id
- name
- category
- ingredients

Q-1 Retrieve the total number of orders placed.



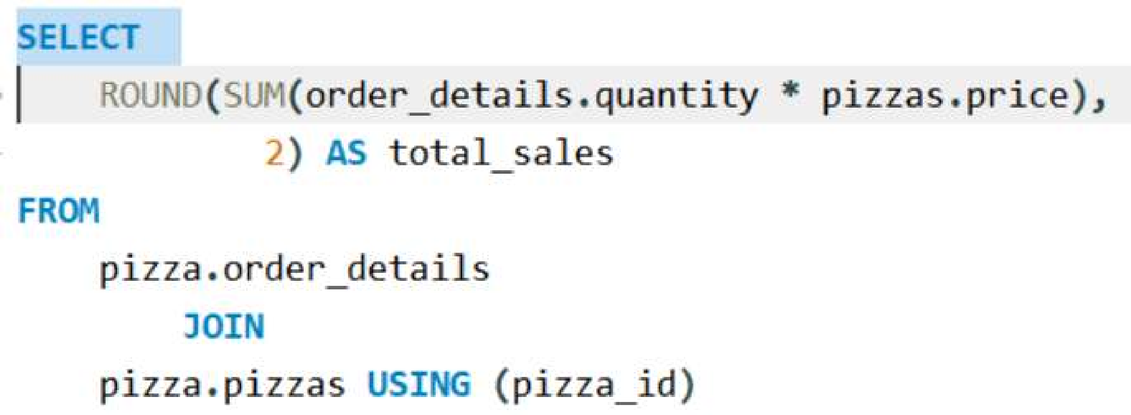
```
SELECT
    COUNT(order_id) AS total_orders
FROM
    pizza.orders;
```



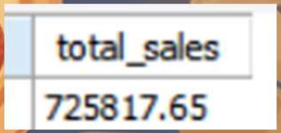
total_orders
21350

Total 21,350 Orders Were Placed.

Q-2 Calculate the total revenue generated from pizza sales.



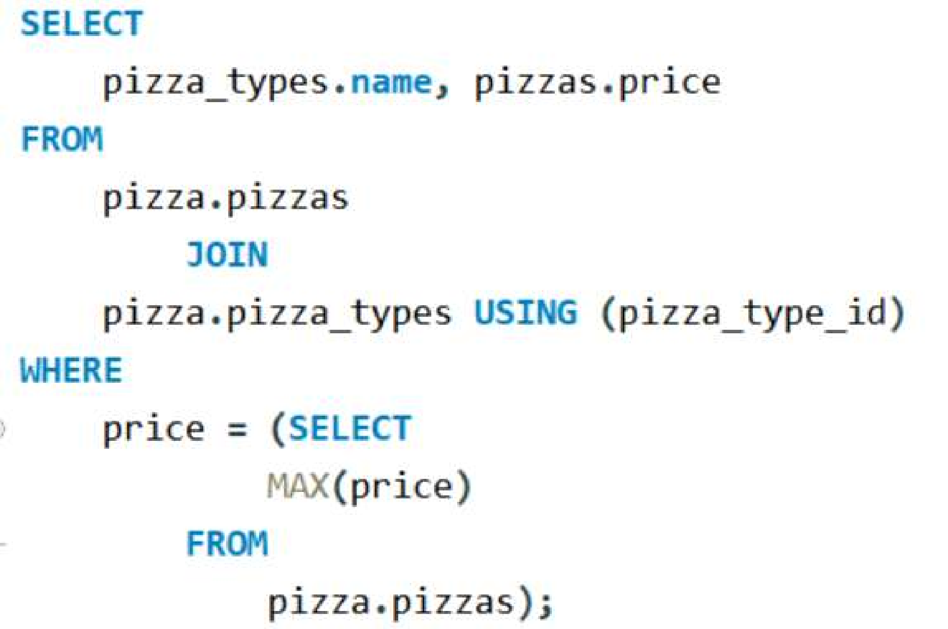
```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    pizza.order_details
        JOIN
    pizza.pizzas USING (pizza_id)
```



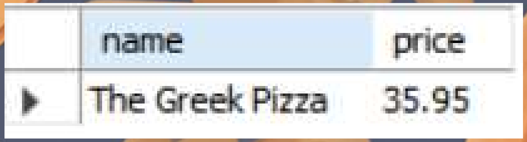
total_sales
725817.65

Total revenue generated from pizza sales is 725817.65.

Q-3 Identify the highest-priced pizza.



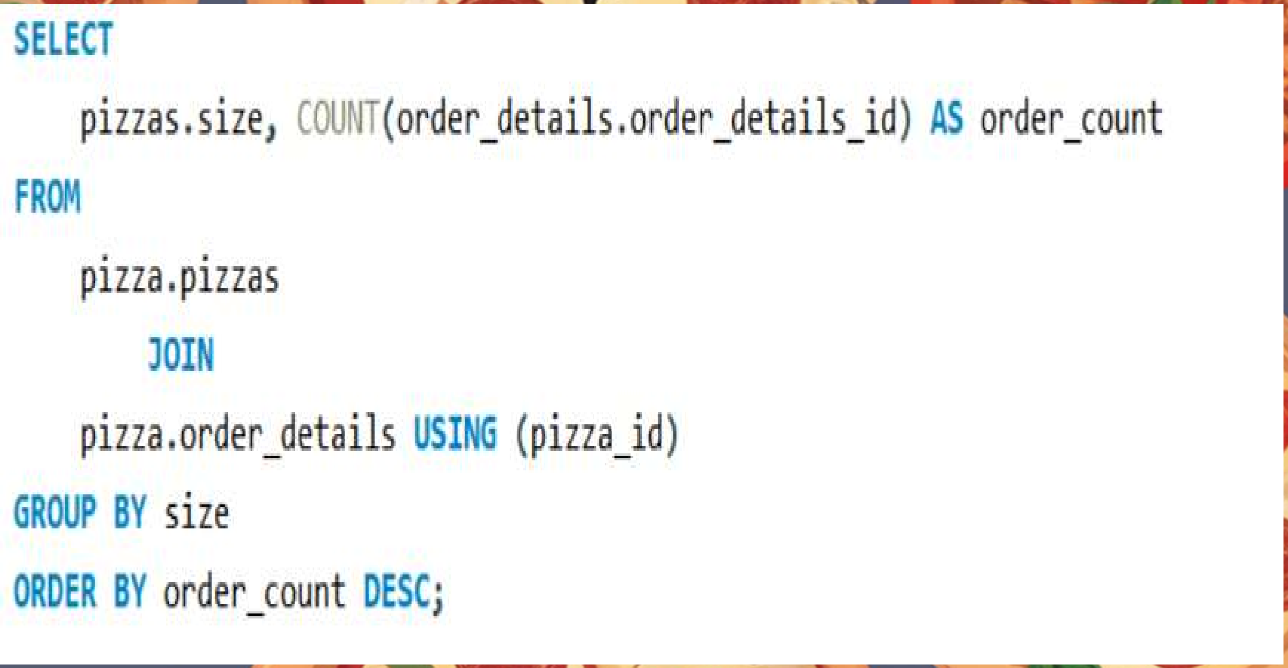
```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza.pizzas
        JOIN
    pizza.pizza_types USING (pizza_type_id)
WHERE
    price = (SELECT
            MAX(price)
        FROM
            pizza.pizzas);
```



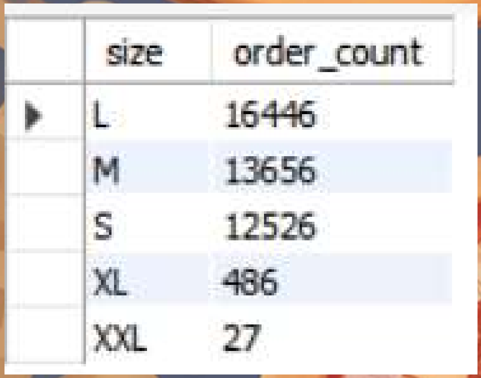
name	price
The Greek Pizza	35.95

The Greek Pizza has the highest price 35.95.

Q-4 Identify the most common pizza size ordered.



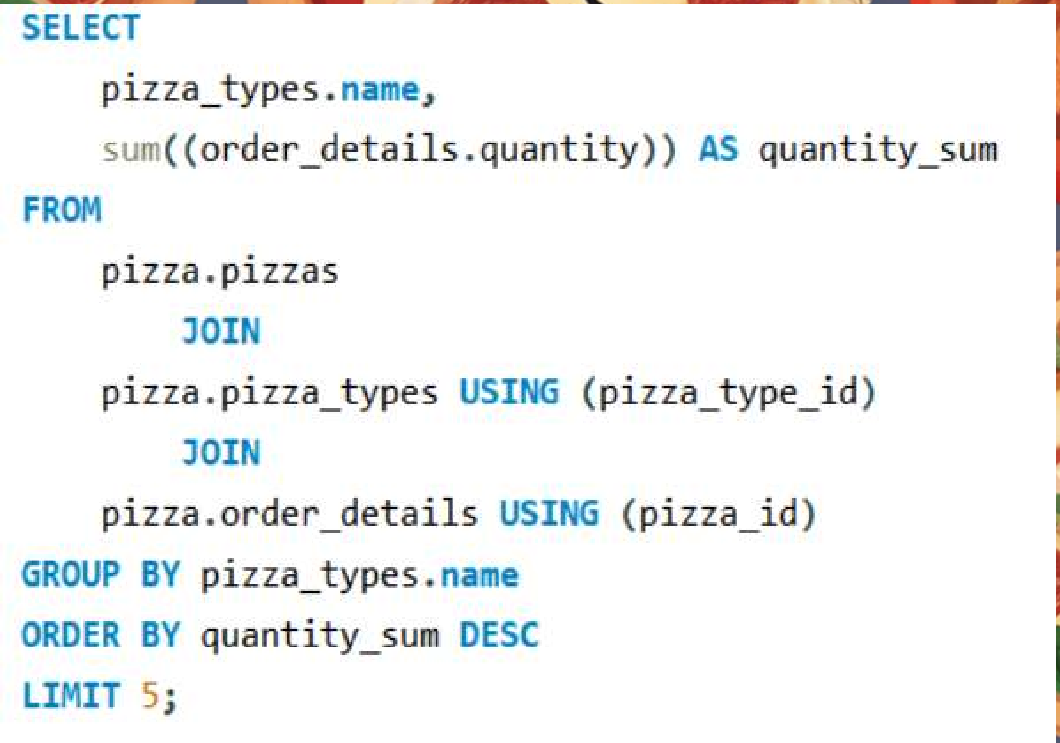
```sql
SELECT
    pizzas.size, COUNT(order_details.order_details_id) AS order_count
FROM
    pizza.pizzas
        JOIN
    pizza.order_details USING (pizza_id)
GROUP BY size
ORDER BY order_count DESC;
```



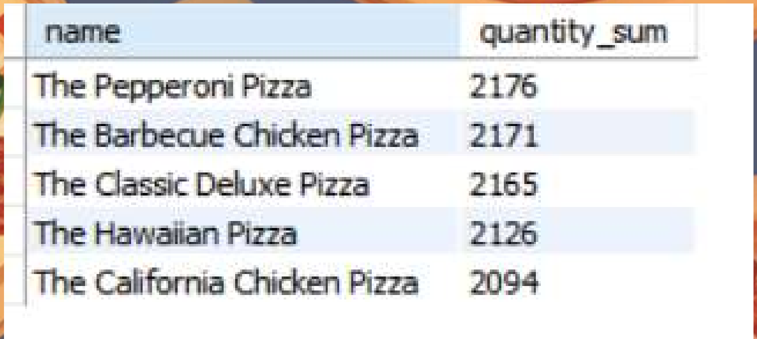
size	order_count
L	16446
M	13656
S	12526
XL	486
XXL	27

The most common pizza size is Large with Order count of 16446.

Q-5 List the top 5 most ordered pizza types along with their quantities.



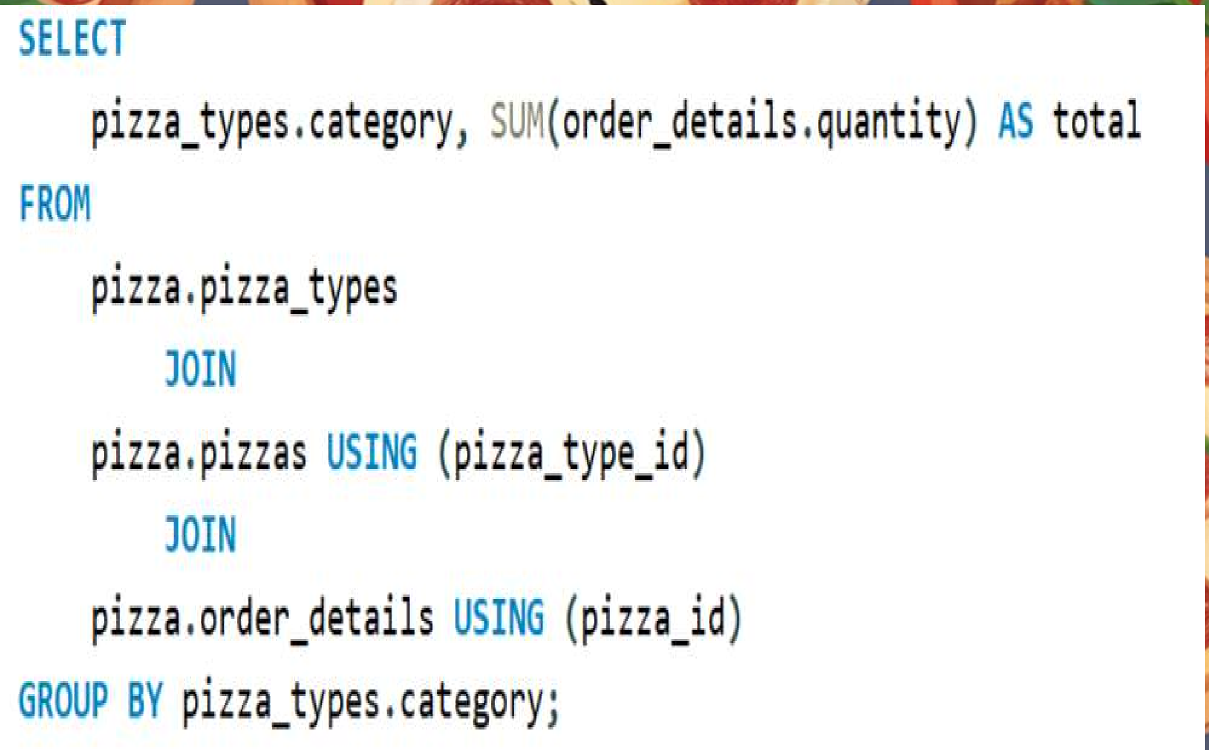
```
SELECT
    pizza_types.name,
    sum((order_details.quantity)) AS quantity_sum
FROM
    pizza.pizzas
        JOIN
    pizza.pizza_types USING (pizza_type_id)
        JOIN
    pizza.order_details USING (pizza_id)
GROUP BY pizza_types.name
ORDER BY quantity_sum DESC
LIMIT 5;
```



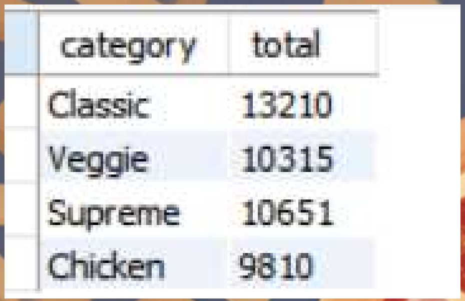
name	quantity_sum
The Pepperoni Pizza	2176
The Barbecue Chicken Pizza	2171
The Classic Deluxe Pizza	2165
The Hawaiian Pizza	2126
The California Chicken Pizza	2094

Top 5 most ordered pizza types are shown in above table with their ordered quantity.

Q-6 Join the necessary tables to find the total quantity of each pizza category ordered.



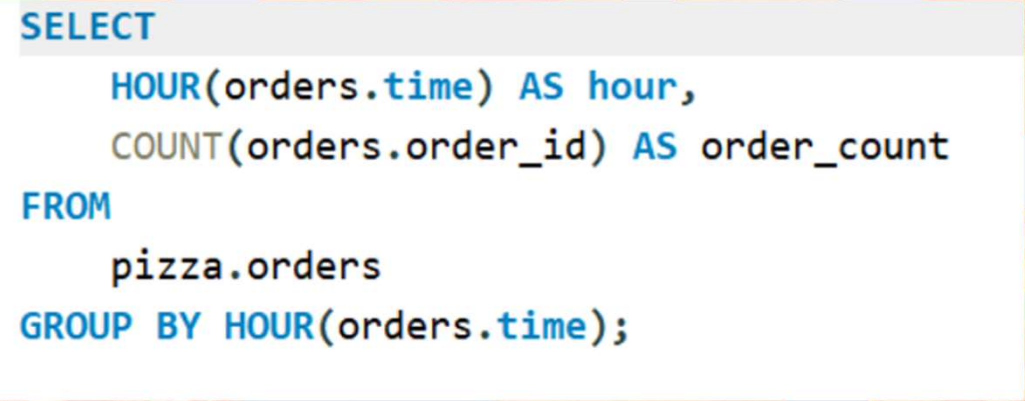
```
SELECT
    pizza_types.category, SUM(order_details.quantity) AS total
FROM
    pizza.pizza_types
        JOIN
    pizza.pizzas USING (pizza_type_id)
        JOIN
    pizza.order_details USING (pizza_id)
GROUP BY pizza_types.category;
```



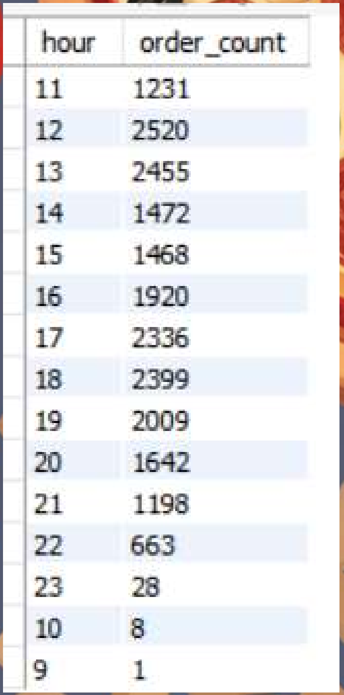
category	total
Classic	13210
Veggie	10315
Supreme	10651
Chicken	9810

Each pizza category are shown in above table with their ordered quantity.

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



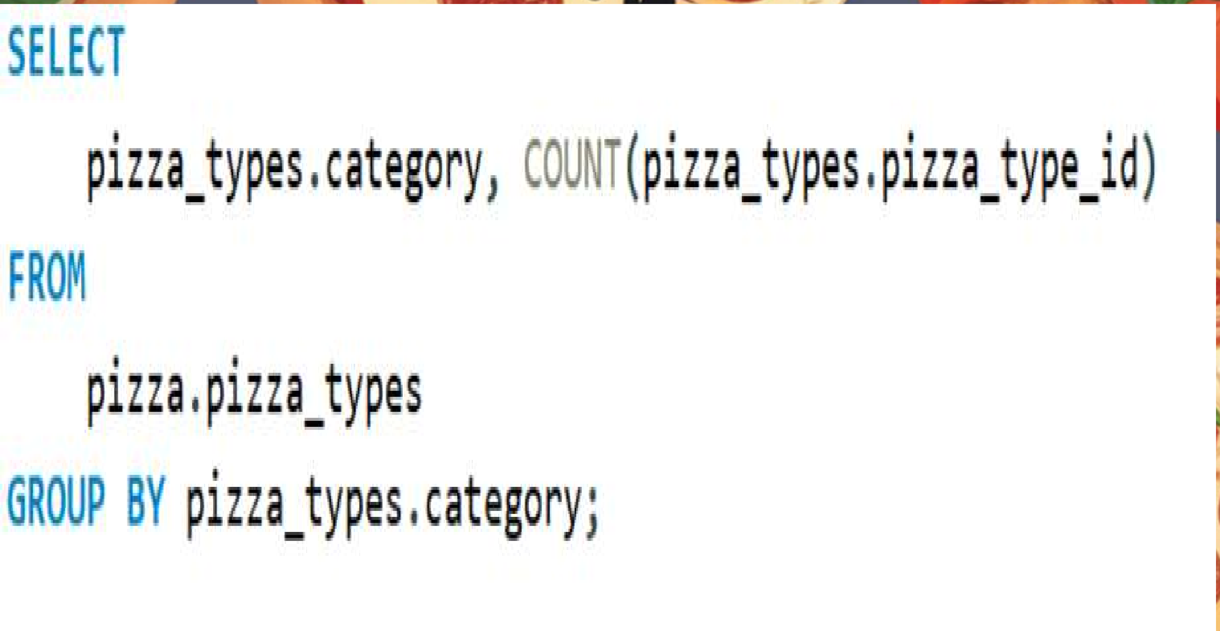
```
SELECT
    HOUR(orders.time) AS hour,
    COUNT(orders.order_id) AS order_count
FROM
    pizza.orders
GROUP BY HOUR(orders.time);
```



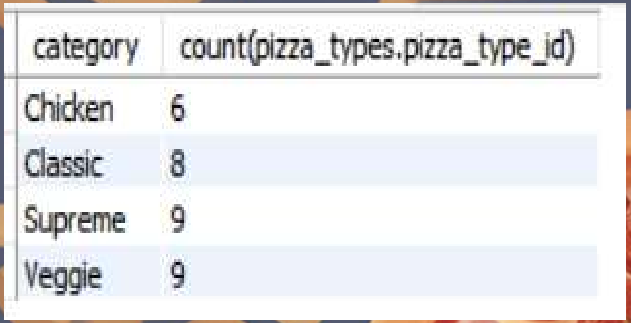
hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

We can see that hours 12,13 and 16-20 were the bussiest hours.

Q-8 Join relevant tables to find the category-wise distribution of pizzas.



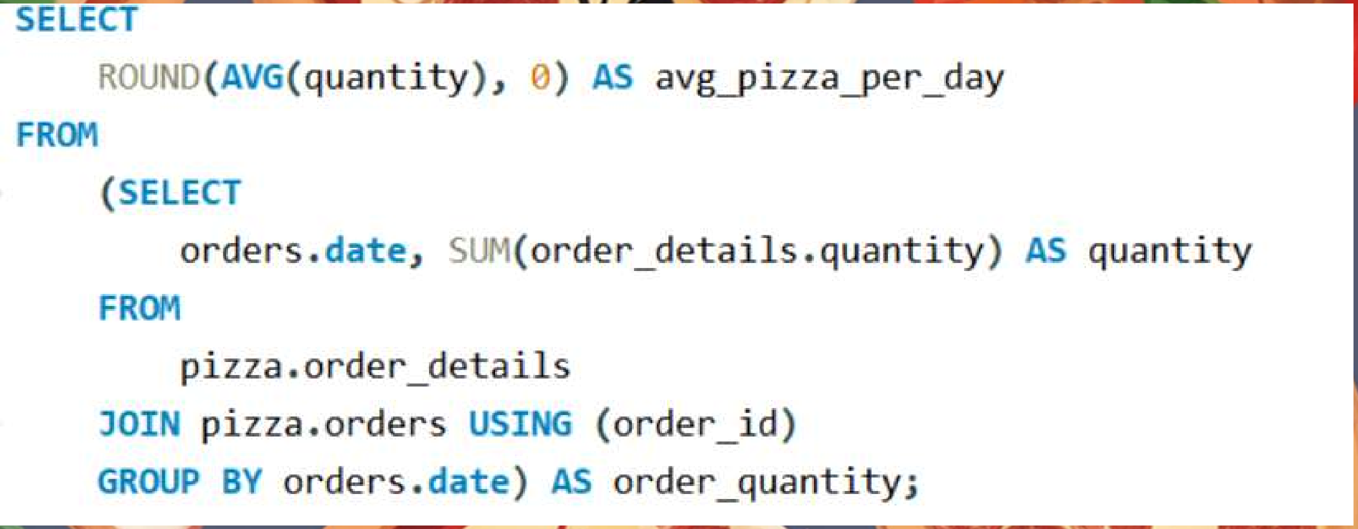
```
SELECT
    pizza_types.category, COUNT(pizza_types.pizza_type_id)
FROM
    pizza.pizza_types
GROUP BY pizza_types.category;
```



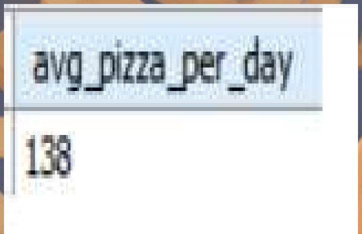
category	count(pizza_types.pizza_type_id)
Chicken	6
Classic	8
Supreme	9
Veggie	9

Supreme and Veggie category has most no of different pizzas.

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



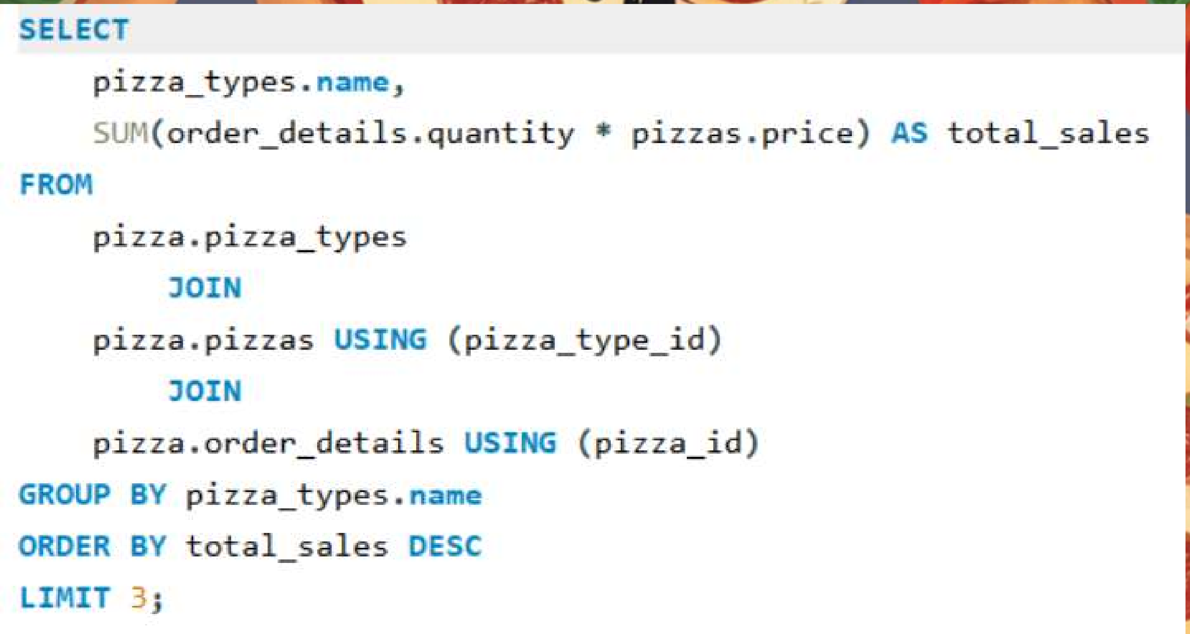
```
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizza_per_day
FROM
    (SELECT
        orders.date, SUM(order_details.quantity) AS quantity
    FROM
        pizza.order_details
    JOIN pizza.orders USING (order_id)
    GROUP BY orders.date) AS order_quantity;
```



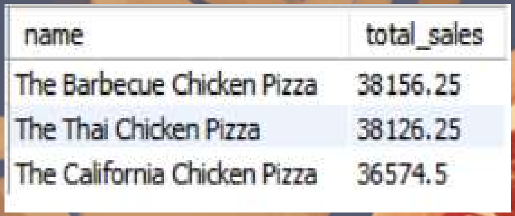
avg_pizza_per_day
138

Average 138 no of pizzas ordered per day.

Q-10 Determine the top 3 most ordered pizza types based on revenue.



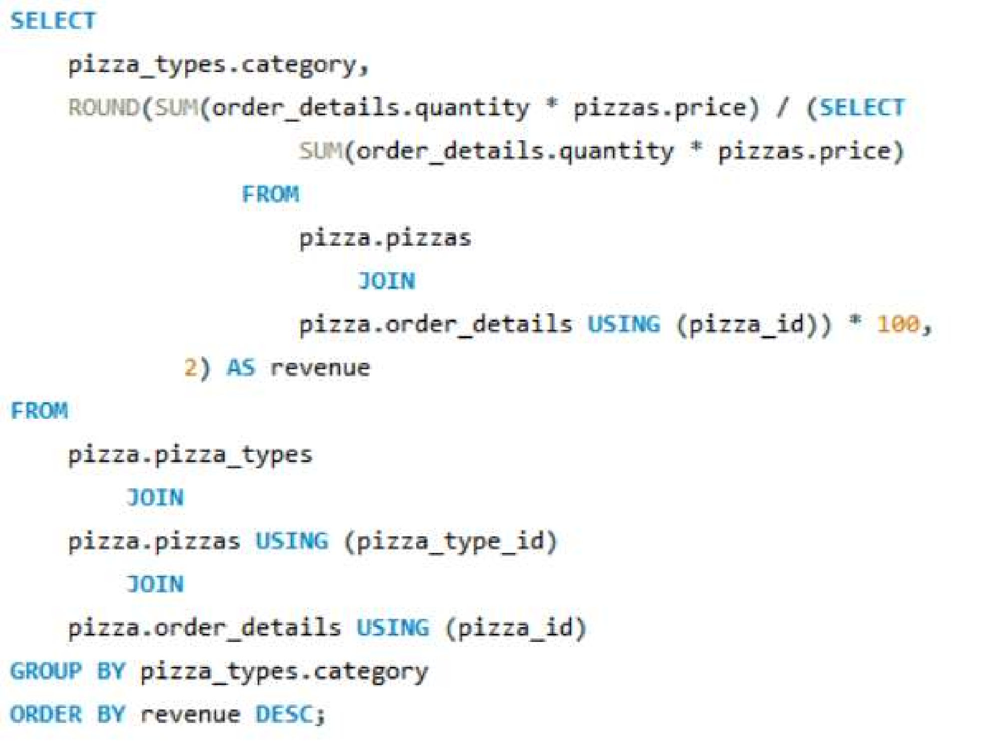
```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS total_sales
FROM
    pizza.pizza_types
        JOIN
    pizza.pizzas USING (pizza_type_id)
        JOIN
    pizza.order_details USING (pizza_id)
GROUP BY pizza_types.name
ORDER BY total_sales DESC
LIMIT 3;
```



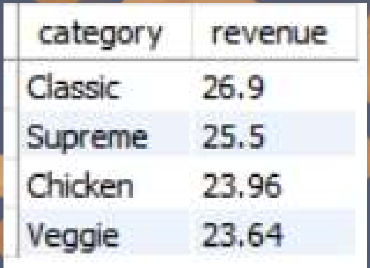
name	total_sales
The Barbecue Chicken Pizza	38156.25
The Thai Chicken Pizza	38126.25
The California Chicken Pizza	36574.5

Top 3 most ordered pizza types with most revenue are shown in above table.

Q-11 Calculate the percentage contribution of each pizza type to total revenue.



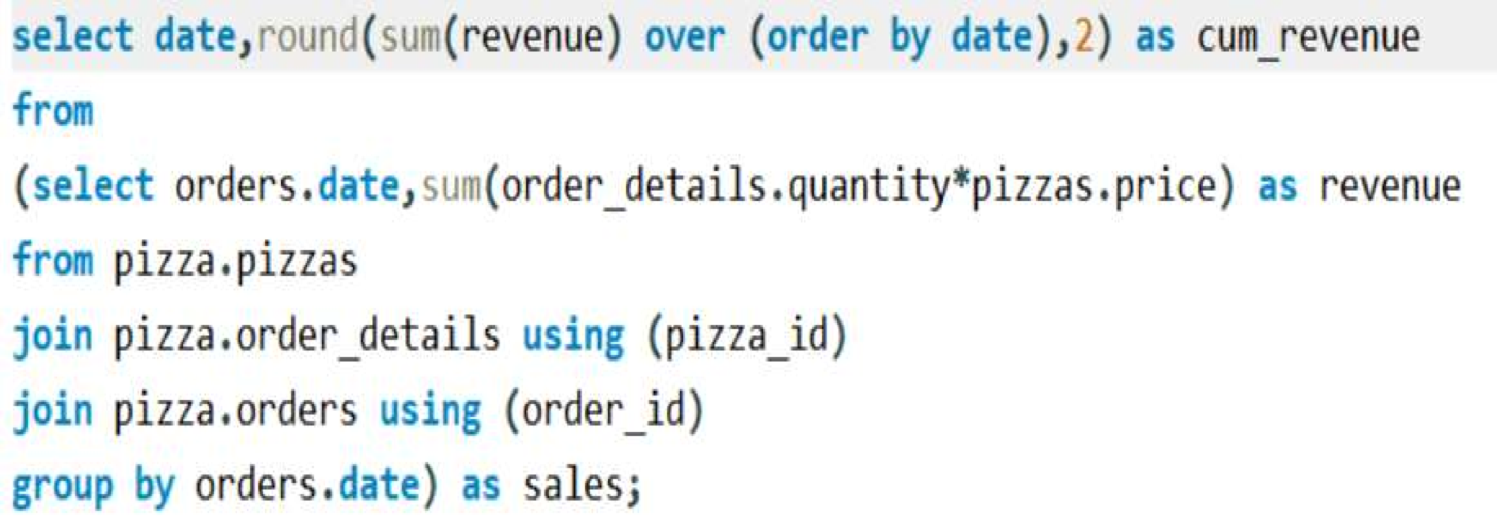
```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    SUM(order_details.quantity * pizzas.price)
                FROM
                    pizza.pizzas
                        JOIN
                    pizza.order_details USING (pizza_id)) * 100,
            2) AS revenue
FROM
    pizza.pizza_types
        JOIN
    pizza.pizzas USING (pizza_type_id)
        JOIN
    pizza.order_details USING (pizza_id)
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```



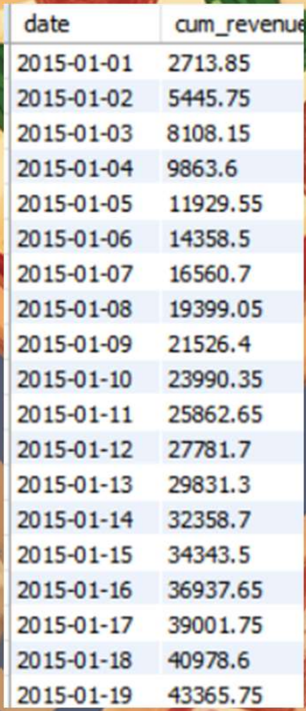
category	revenue
Classic	26.9
Supreme	25.5
Chicken	23.96
Veggie	23.64

The percentage wise contribution of each pizza type to total revenue are shown in above table.

Q-12 Analyze the cumulative revenue generated over time.



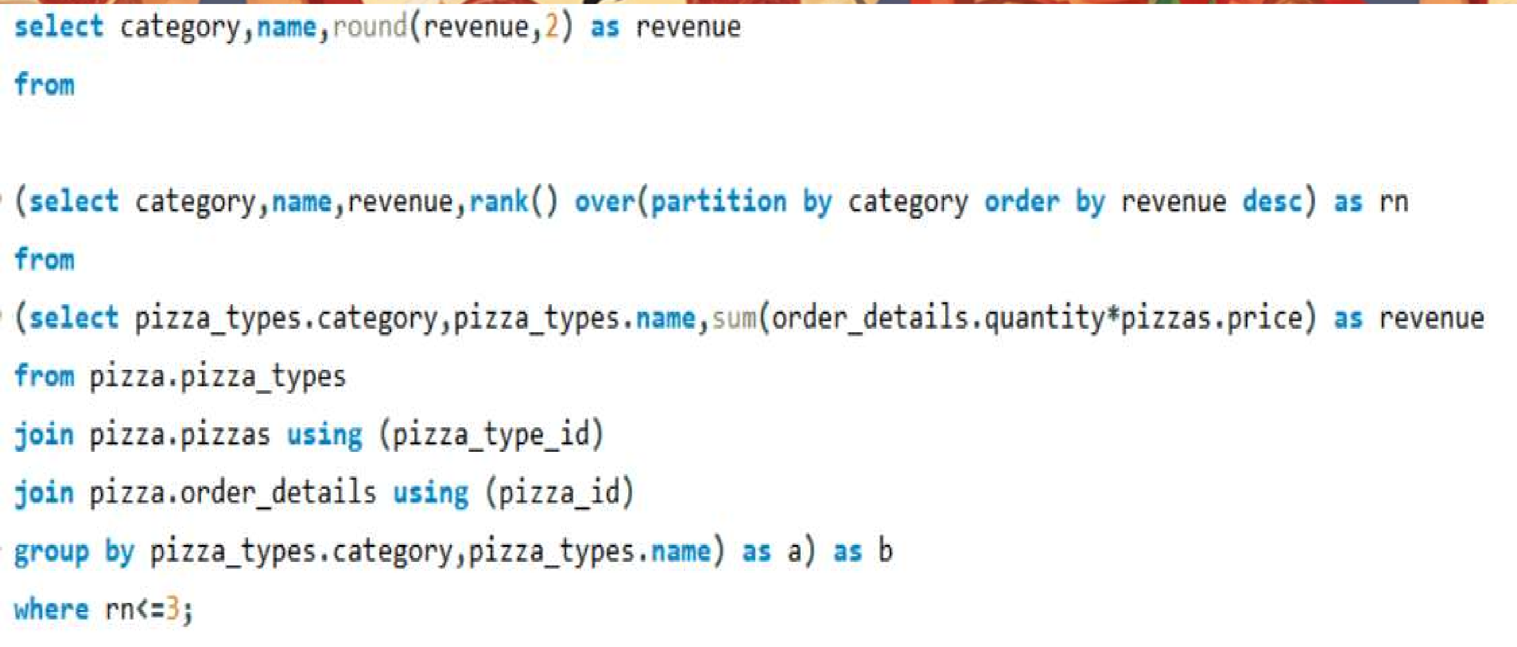
```
select date,round(sum(revenue) over (order by date),2) as cum_revenue
from
(select orders.date,sum(order_details.quantity*pizzas.price) as revenue
from pizza.pizzas
join pizza.order_details using (pizza_id)
join pizza.orders using (order_id)
group by orders.date) as sales;
```



date	cum_revenue
2015-01-01	2713.85
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
2015-01-09	21526.4
2015-01-10	23990.35
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3
2015-01-14	32358.7
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6
2015-01-19	43365.75

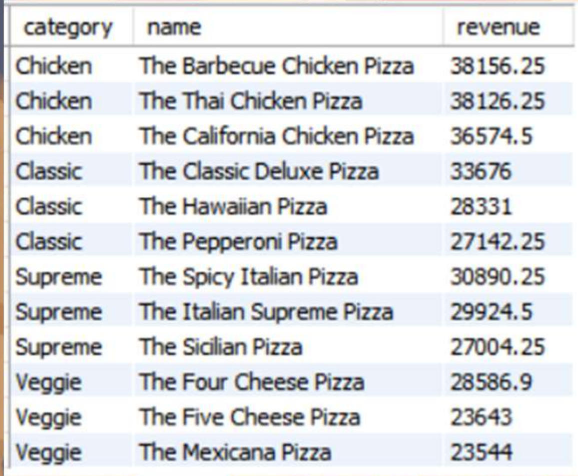
The cumulative revenue generated over time are shown in above tabel.

Q-13 Determine the top 3 most ordered pizza types based on revenue for each pizza category.



```
select category,name,round(revenue,2) as revenue
from

(select category,name,revenue,rank() over(partition by category order by revenue desc) as rn
from
(select pizza_types.category,pizza_types.name,sum(order_details.quantity*pizzas.price) as revenue
from pizza.pizza_types
join pizza.pizzas using (pizza_type_id)
join pizza.order_details using (pizza_id)
group by pizza_types.category,pizza_types.name) as a) as b
where rn<=3;
```

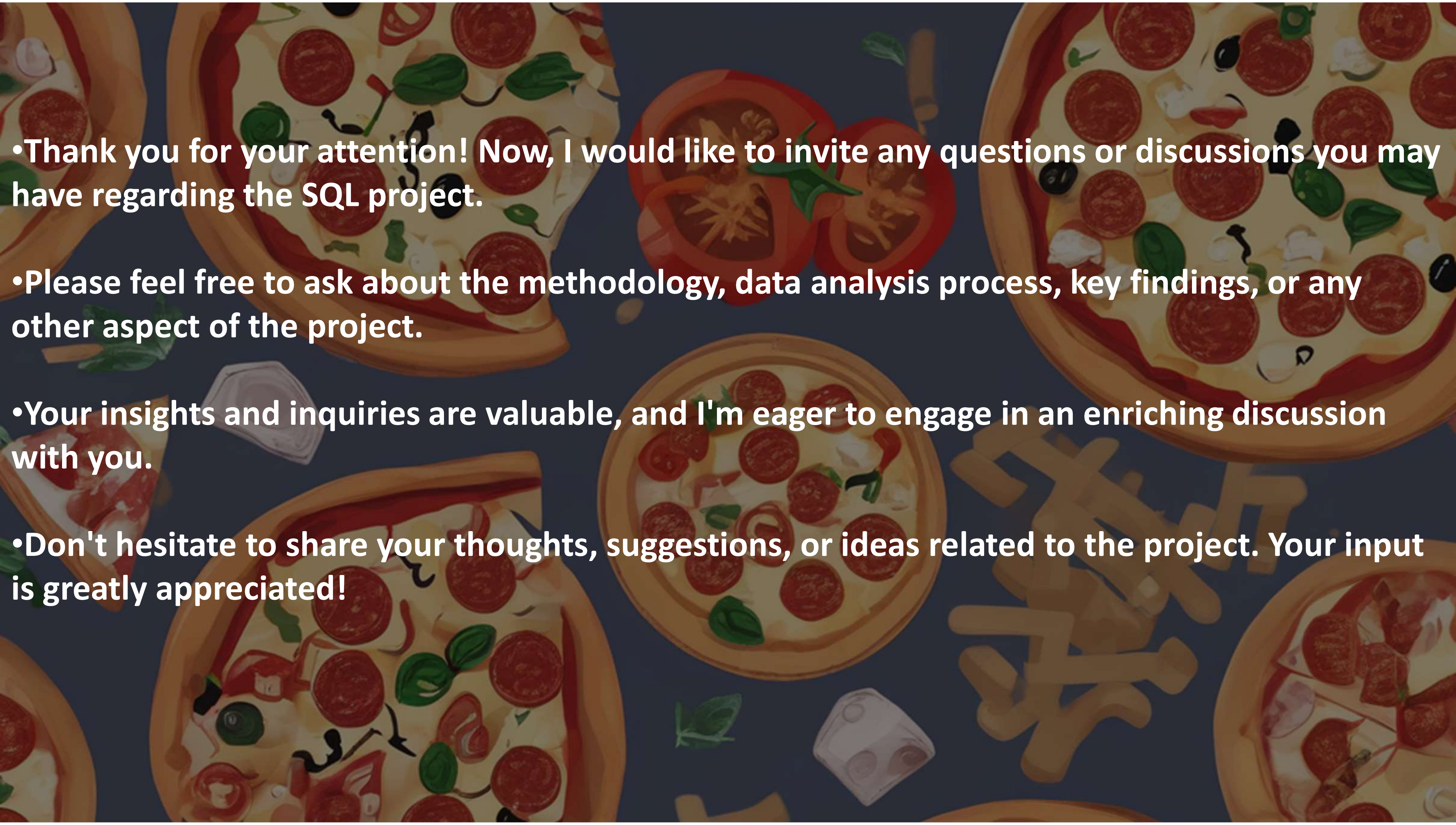


category	name	revenue
Chicken	The Barbecue Chicken Pizza	38156.25
Chicken	The Thai Chicken Pizza	38126.25
Chicken	The California Chicken Pizza	36574.5
Classic	The Classic Deluxe Pizza	33676
Classic	The Hawaiian Pizza	28331
Classic	The Pepperoni Pizza	27142.25
Supreme	The Spicy Italian Pizza	30890.25
Supreme	The Italian Supreme Pizza	29924.5
Supreme	The Sicilian Pizza	27004.25
Veggie	The Four Cheese Pizza	28586.9
Veggie	The Five Cheese Pizza	23643
Veggie	The Mexicana Pizza	23544

Top 3 most ordered pizza types based on revenue for each pizza category are shown in above table.

Conclusion:

- Through SQL analysis of the pizza order dataset, significant insights were gained into pizza consumption trends and preferences. These insights can inform decision-making processes for pizza businesses, including menu planning, marketing strategies, and operational optimizations.



- Thank you for your attention! Now, I would like to invite any questions or discussions you may have regarding the SQL project.

- Please feel free to ask about the methodology, data analysis process, key findings, or any other aspect of the project.

- Your insights and inquiries are valuable, and I'm eager to engage in an enriching discussion with you.

- Don't hesitate to share your thoughts, suggestions, or ideas related to the project. Your input is greatly appreciated!