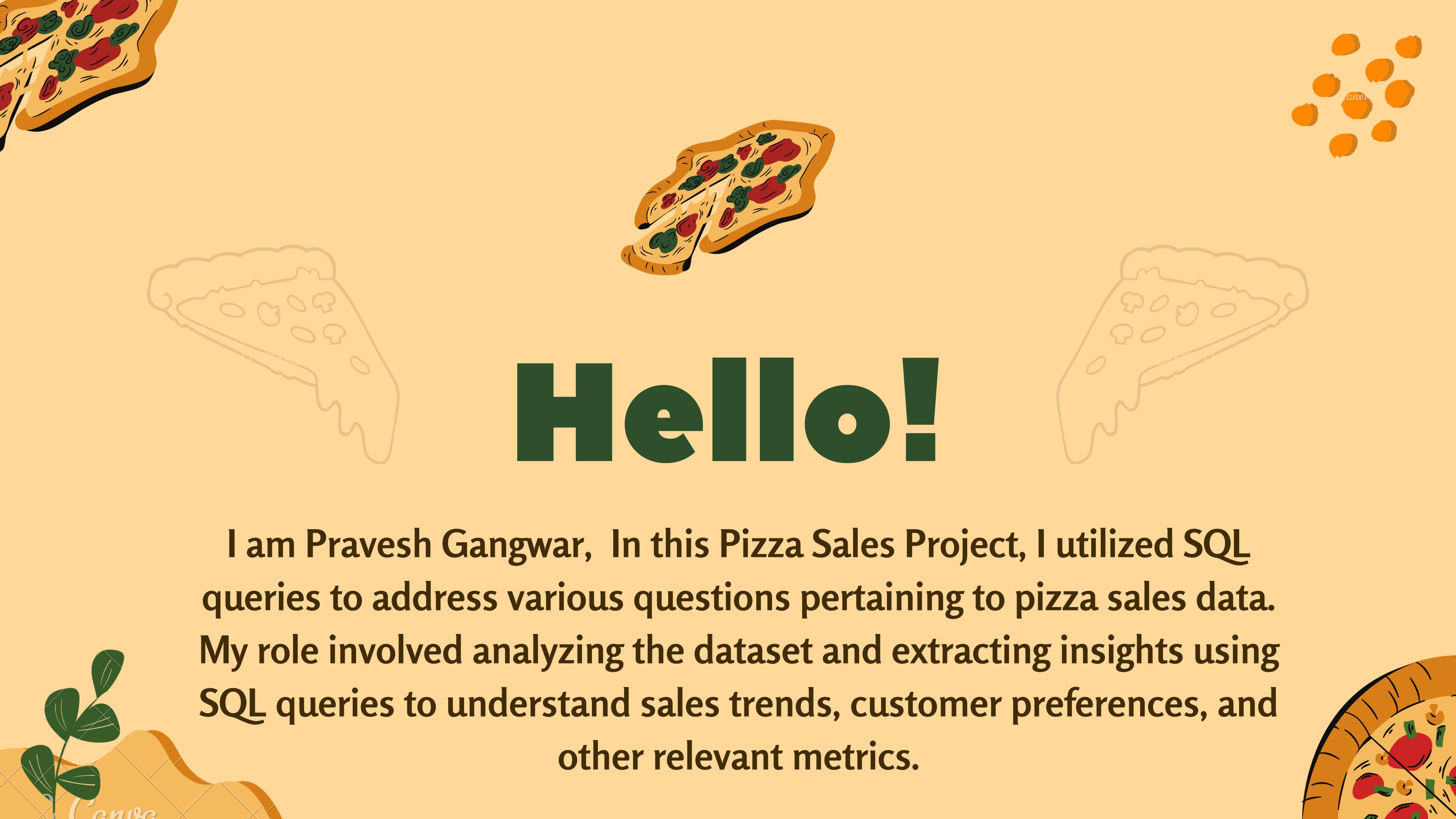


Delicious Pizza for Everyone!

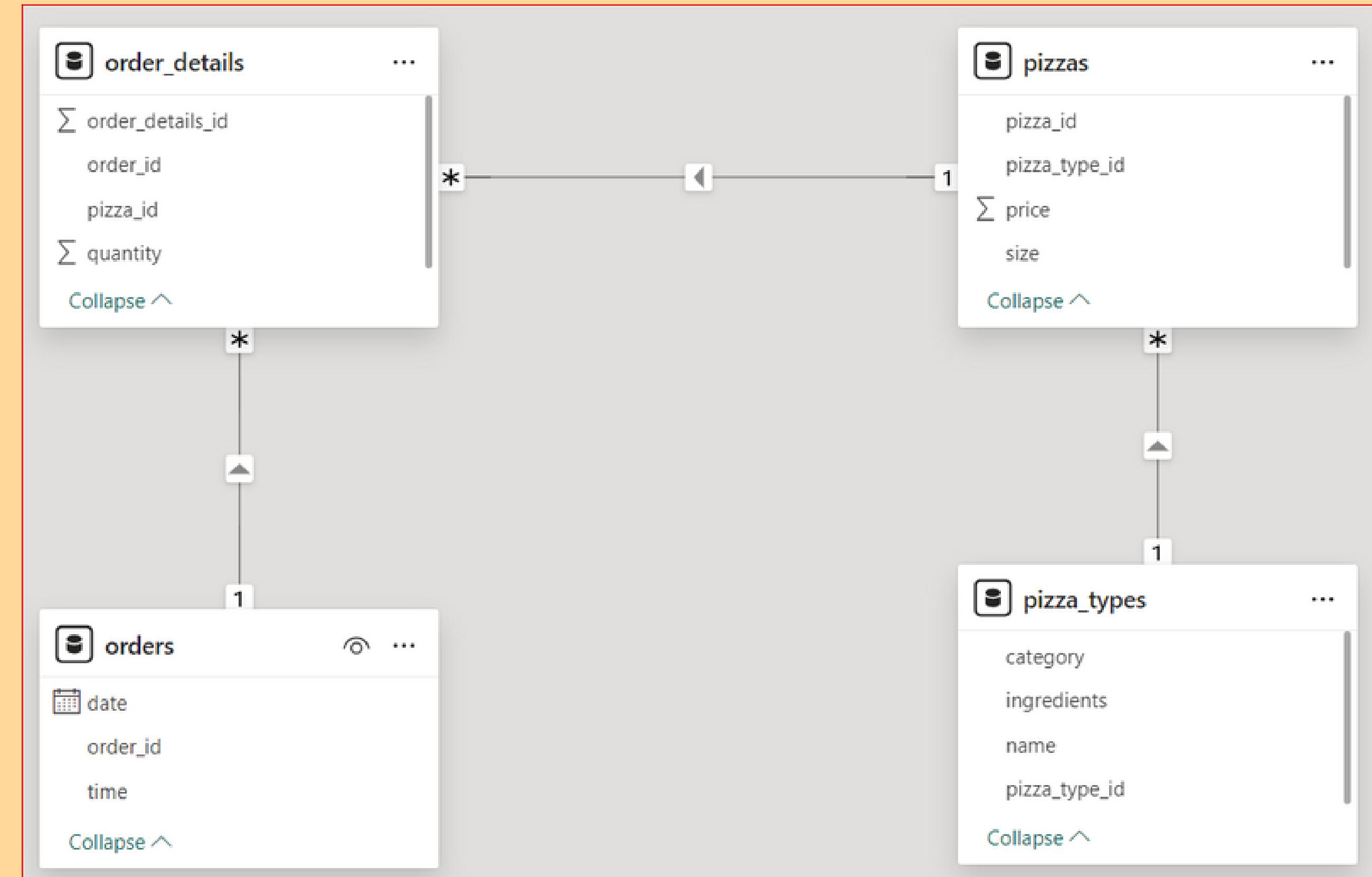
PIZZA SALES ANALYSIS



Hello!

I am Pravesh Gangwar, In this Pizza Sales Project, I utilized **SQL** queries to address various questions pertaining to pizza sales data. My role involved analyzing the dataset and extracting insights using **SQL** queries to understand sales trends, customer preferences, and other relevant metrics.

Data Model



Retrieve the total number of orders placed.

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

| Result Grid | |
|-------------|-------------|
| | total_order |
| ▶ | 21350 |

Calculate the total revenue generated from pizza sales.

```
SELECT  
    ROUND(SUM(quantity * price), 2) as Total_Revenue  
FROM  
    order_details  
    JOIN  
    pizzas USING (pizza_id);
```

| Result Grid | |
|-------------|---------------|
| | Total_Revenue |
| ▶ | 817860.05 |

Identify the highest-priced pizza.

```
SELECT  
    name, price  
FROM  
    pizzas  
    JOIN  
    pizza_types USING (pizza_type_id)  
ORDER BY price DESC  
LIMIT 1;
```

Result Grid | Filter Rows:

| | name | price |
|---|-----------------|-------|
| ▶ | The Greek Pizza | 35.95 |

Identify the most common pizza size ordered.

```
SELECT  
    size, COUNT(order_details_id) AS order_count  
FROM  
    pizzas  
        JOIN  
    order_details USING (pizza_id)  
GROUP BY size  
order by order_count DESC  
limit 1;
```

Result Grid | Filter

| | size | order_count |
|---|------|-------------|
| ▶ | L | 18526 |

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

SELECT

```
    pizza_types.name,  
    SUM(order_details.quantity) AS Total_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 Total_Quantity DESC  
  
LIMIT 5;
```

| | name | Total_Quantity |
|---|----------------------------|----------------|
| ▶ | The Classic Deluxe Pizza | 2453 |
| | The Barbecue Chicken Pizza | 2432 |
| | The Hawaiian Pizza | 2422 |
| | The Pepperoni Pizza | 2418 |
| | The Thai Chicken Pizza | 2371 |



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



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

| Result Grid | |
|-------------|-------|
| category | QTY |
| Classic | 14888 |
| Supreme | 11987 |
| Veggie | 11649 |
| Chicken | 11050 |



Determine the distribution of orders by hour of the day.

```
SELECT  
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(order_time)  
order by order_count DESC;
```

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



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



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

Result Grid   Filter Row

| | category | count(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 avg_pizza_ordered_per_day  
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_qty
```

| | avg_pizza_ordered_per_day |
|---|---------------------------|
| ▶ | 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 pizza_types.pizza_type_id = pizzas.pizza_type_id  
        JOIN  
    order_details ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY name  
ORDER BY revenue DESC  
LIMIT 3;
```

| Result Grid | | Filter Rows: |
|-------------|------------------------------|--------------|
| | 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.

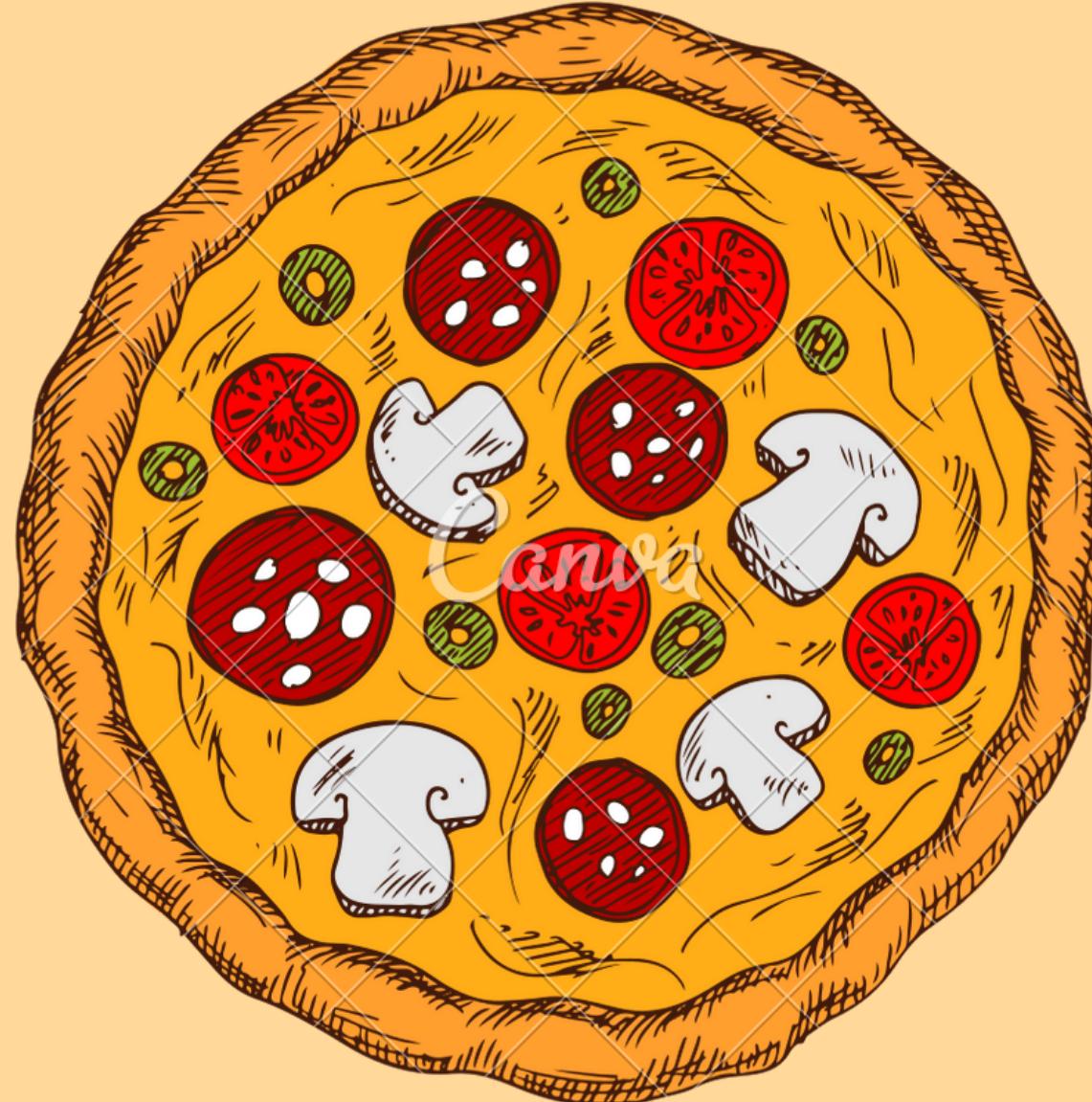
```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
        2) AS Total_Revenue
    )
    FROM
        order_details
        JOIN
            pizzas ON order_details.pizza_id = pizzas.pizza_id) * 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 |
|---|----------|---------|
| ▶ | Classic | 26.91 |
| | Supreme | 25.46 |
| | Chicken | 23.96 |
| | Veggie | 23.68 |

Analyze the cumulative revenue generated over time.

```
select order_date,sum(total_revenue) over(order by order_date) as cum_revenue  
from  
(select orders.order_date,sum(order_details.quantity*pizzas.price) as total_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 |
| | 2015-01-06 | 14358.5 |



**THANK
YOU**

praveshgangwar10@gmail.com