



RDBMS



PIZZA SALE STUDY



QUERIES:

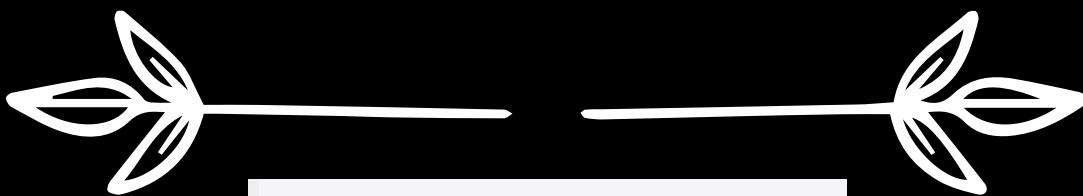
1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza type along with their quantities.
6. Join the necessary tables to find the total quantity of each pizza category.
7. Determine the distribution of orders by hour of the day.
8. Determine the top 3 most ordered pizza types based on revenue.
9. Calculate the percentage contribution of each pizza category of total revenue.
10. Analyse the cumulative revenue generated over time.





1. Retrieve the total number of orders placed.

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



Result Grid	
	Total_Orders
▶	21350





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** pizzas.pizza_id = order_details.pizza_id;



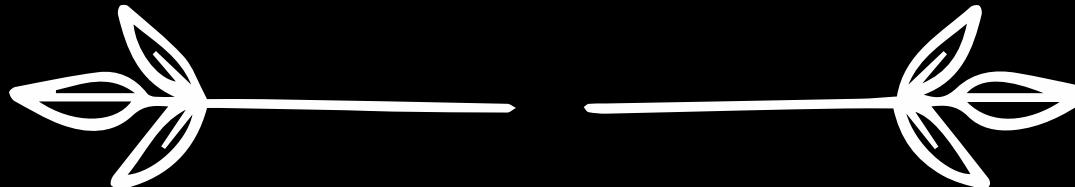
Result Grid

	Total_Sales
▶	817860.05



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

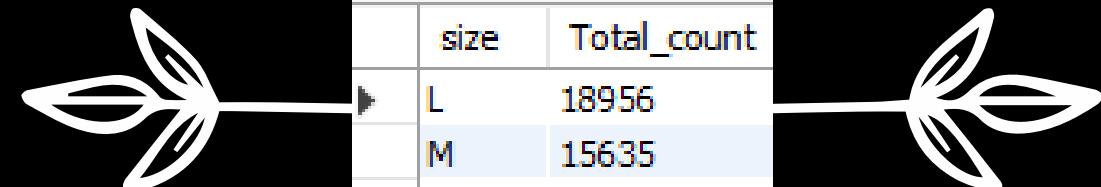




4. Identify the most common pizza size ordered.



```
SELECT pizzas.size, SUM(order_details.quantity) AS Total_count  
FROM pizzas  
JOIN order_details ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY pizzas.size  
ORDER BY Total_count DESC;
```



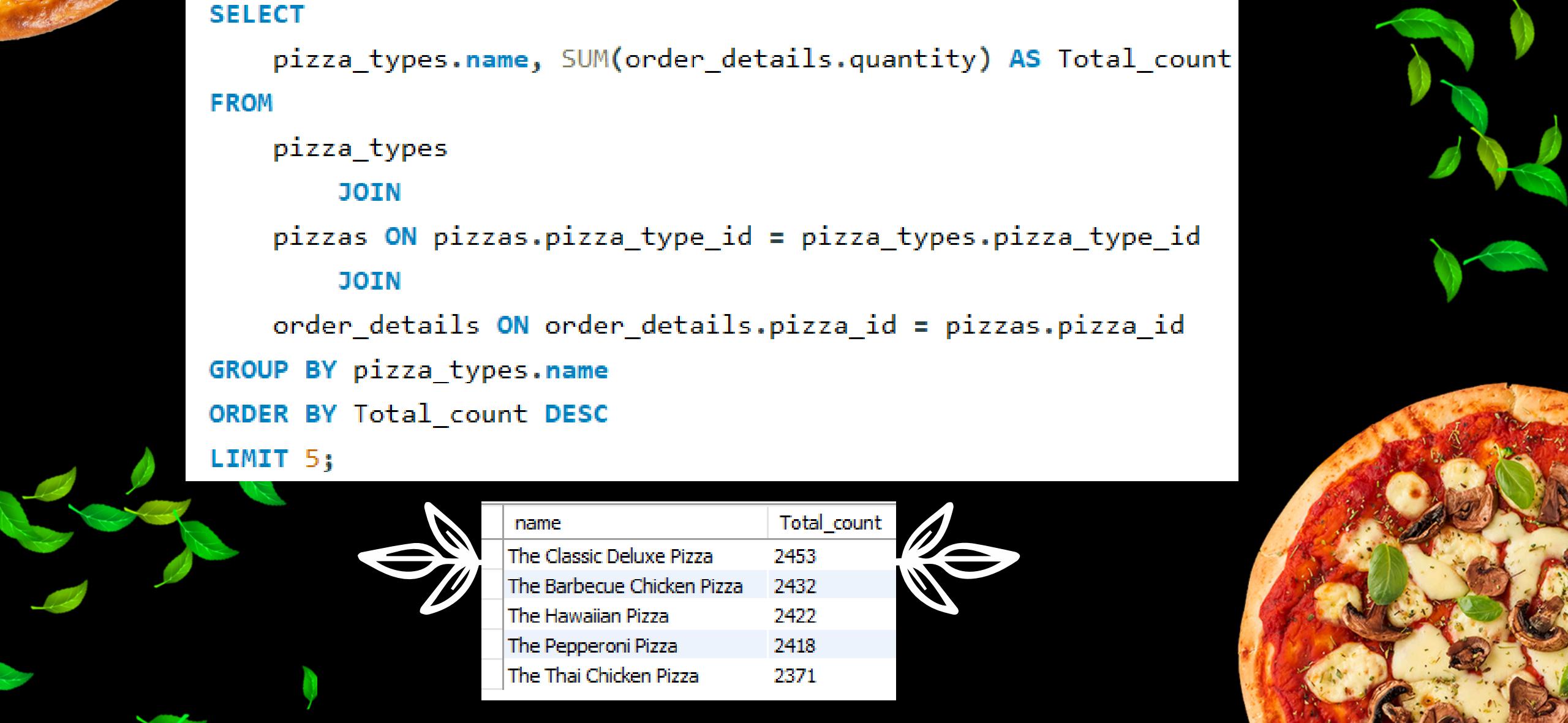
Result Grid		
	size	Total_count
	L	18956
	M	15635
	S	14403
	XL	552
	XXL	28





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

```
SELECT  
    pizza_types.name, SUM(order_details.quantity) AS Total_count  
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 Total_count DESC  
LIMIT 5;
```



	name	Total_count
	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

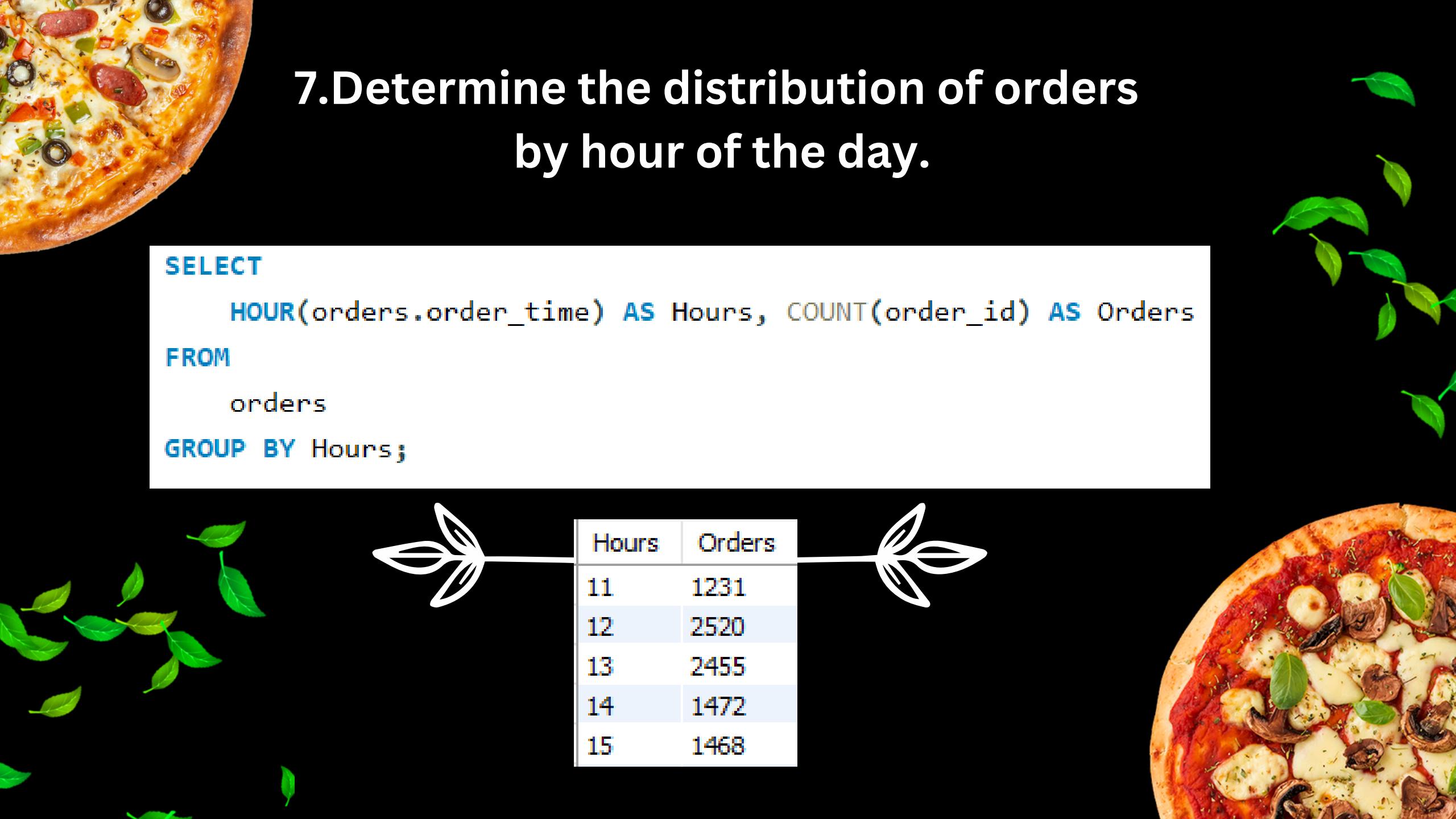


6.Join the necessary tables to find the total quantity of each pizza category.

```
SELECT  
    pizza_types.category,  
    COUNT(order_details.quantity) AS Total_count  
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.category  
ORDER BY Total_count DESC;
```

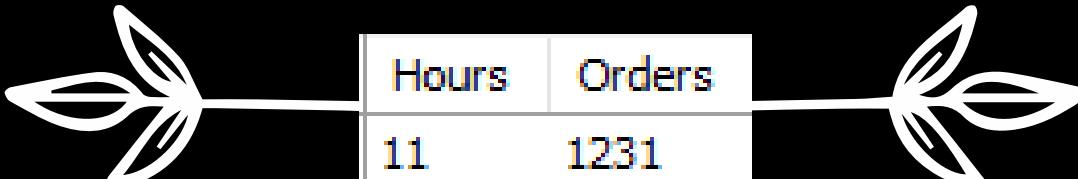
category	Total_count
Classic	14579
Supreme	11777
Veggie	11449
Chicken	10815





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

```
SELECT  
    HOUR(orders.order_time) AS Hours, COUNT(order_id) AS Orders  
FROM  
    orders  
GROUP BY Hours;
```



Hours	Orders
11	1231
12	2520
13	2455
14	1472
15	1468



8.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 DESC  
LIMIT 3;
```

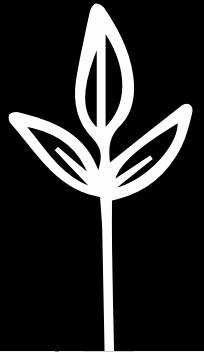
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5



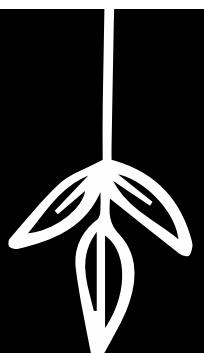


9. Calculate the percentage contribution of each pizza category of total revenue.

```
SELECT  
    pizza_types.category,  
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT  
        ROUND(SUM(order_details.quantity * pizzas.price),  
        2) AS Total_sale  
    ) * 100,  
    2) AS revenue  
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
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



10. Analyse the cumulative revenue generated over time.

```
SELECT order_date, sum(revenue) OVER (ORDER BY order_date) AS cum_revenue
FROM
(SELECT
    orders.order_date,
    round(SUM(order_details.quantity * pizzas.price),2) AS revenue
FROM
    orders
    JOIN
    order_details ON order_details.order_id = orders.order_id
    JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
GROUP BY orders.order_date) as Sales;
```

order_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





THANK YOU



By: ANMOL KAPOOR

