Q1. Retrieve the total number of orders placed.

SELECT count(order\_id) AS total\_orders

FROM orders;



Q2.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;

SELECT

ROUND(SUM(order\_details.quantity \* pizzas.price),

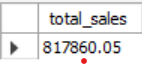
2) AS total\_sales

FROM

order\_details

JOIN

pizzas ON pizzas.pizza\_id = order\_details.pizza\_id;



Q3.Identify the highest-priced pizza.

SELECT

pizza\_types.name, pizzas.price

FROM

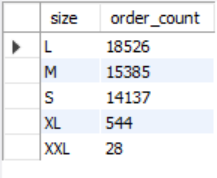
pizza\_types

JOIN

pizzas ON pizzas.pizza\_type\_id = pizza\_types.pizza\_type\_id

ORDER BY pizzas.price DESC

LIMIT 1;



Q4. 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

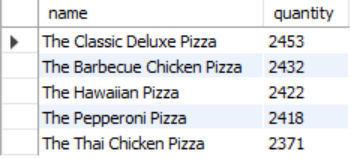
JOIN

order\_details ON order\_details.pizza\_id = pizzas.pizza\_id

GROUP BY pizza\_types.name

ORDER BY quantity DESC

LIMIT 5;



Q5. 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

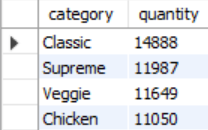
pizzas ON pizza\_types.pizza\_type\_id = pizzas.pizza\_type\_id

JOIN

order\_details ON order\_details.pizza\_id = pizzas.pizza\_id

GROUP BY pizza\_types.category

ORDER BY quantity DESC;



Q6. Determine the distribution of orders by hour of the day.

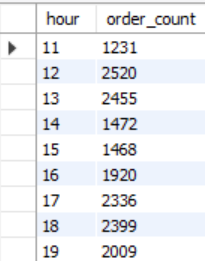
SELECT

HOUR(order\_time) AS hour, COUNT(order\_id) AS order\_count

FROM

orders

GROUP BY HOUR(order\_time);





Q7. 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\_quantity;



Q8. 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

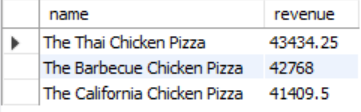
JOIN

order\_details ON order\_details.pizza\_id = pizzas.pizza\_id

GROUP BY pizza\_types.name

ORDER BY revenue DESC

LIMIT 3;



Q9. 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 pizzas.pizza\_id=order\_details.pizza\_id)\*100,2) AS revenue

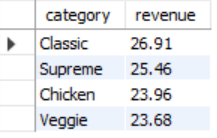
FROM pizza\_types

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;



Q10. Analyze the cumulative revenue generated over time.

SELECT order\_date,

SUM(revenue) OVER (ORDER BY order\_date) AS cum\_revenue

FROM

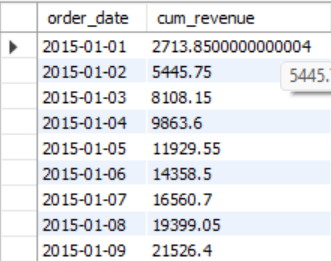
(SELECT orders.order\_date,SUM(order\_details.quantity \* pizzas.price) AS revenue

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;



Q11.Determine the top 3 most ordered pizza types based on revenue for each pizza category.

SELECT name,revenue

FROM

(SELECT category,name,revenue,

RANK()OVER(PARTITION BY category ORDER BY name DESC) AS rn

FROM

(SELECT pizza\_types.category,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.category,pizza\_types.name) AS a) AS b

WHERE rn <= 3;

