-- 1Q:Retrieve the total number of orders placed?

SELECT COUNT (order\_id) AS total\_orders FROM orders;

--2 Q:Calculate the total revenue generated from pizza sales?

SELECT ROUND (SUM(p.price\*od.quantity),2) As total\_revenue

FROM order\_details AS od

INNER JOIN

pizzas AS p

ON p.pizza\_id = od.pizza\_id;

-- 3Q:Identify the highest-priced pizza?

SELECT pt.name As pizza\_name,p.price As pizza\_price FROM pizzas AS p

INNER JOIN

pizza\_types AS pt

ON p.pizza\_type\_id = pt.pizza\_type\_id

ORDER BY pizza\_price DESC LIMIT 1;

Or :

SELECT pt.name As pizza\_name,p.price As pizza\_price FROM pizzas AS p

INNER JOIN

pizza\_types AS pt

ON p.pizza\_type\_id = pt.pizza\_type\_id

WHERE price = (SELECT MAX(price) FROM pizzas);

-- 4Q:Identify the most common pizza size ordered?

SELECT p.size AS pizza\_size,COUNT(od.order\_id) AS order\_count FROM pizzas AS p

INNER JOIN

order\_details AS od

ON p.pizza\_id = od.pizza\_id

GROUP BY p.size

ORDER BY order\_count DESC;

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

SELECT pt.name AS pizza\_types,SUM(od.quantity) AS quantities

FROM pizzas AS p

INNER JOIN

order\_details AS od

ON p.pizza\_id = od.pizza\_id

INNER JOIN

pizza\_types AS pt

ON pt.pizza\_type\_id = p.pizza\_type\_id

GROUP BY pizza\_types

ORDER BY quantities DESC

LIMIT 5;

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

SELECT pt.category AS pizza\_category,SUM(od.quantity) AS total\_quantity

FROM pizzas AS p

INNER JOIN

pizza\_types AS pt

ON p.pizza\_type\_id = pt.pizza\_type\_id

INNER JOIN

order\_details AS od

ON p.pizza\_id = od.pizza\_id

GROUP BY pizza\_category

ORDER BY total\_quantity DESC;

-- 7Q:Determine the distribution of orders by hour of the day?

SELECT

EXTRACT(HOUR FROM order\_time) AS order\_hour,

COUNT (order\_id) AS order\_count

FROM orders

GROUP BY order\_hour;

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

SELECT category, COUNT(name) FROM pizza\_types

GROUP BY category;

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

**Solution 1: Using sub query**

SELECT

order\_date,

ROUND(AVG(total\_pizzas),2) AS avg\_pizzas\_per\_day

FROM (

SELECT

o.order\_date,

SUM(od.quantity) AS total\_pizzas

FROM

orders o

JOIN

order\_details od ON o.order\_id = od.order\_id

GROUP BY

o.order\_date

)

GROUP BY

order\_date

ORDER BY

order\_date;

**Solution 2: Using CTE & Window Function**

WITH daily\_pizza\_orders AS (

SELECT

o.order\_date,

SUM(od.quantity) AS total\_ord\_per\_day

FROM

orders o

JOIN

order\_details od ON o.order\_id = od.order\_id

GROUP BY

o.order\_date

)

SELECT

order\_date,

total\_ord\_per\_day,

AVG(total\_ord\_per\_day) OVER (PARTITION BY order\_date) AS avg\_pizzas\_per\_day

FROM

daily\_pizza\_orders

ORDER BY

order\_date;

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

SELECT

pt.name As pizza\_name,

SUM(p.price\*od.quantity)AS revenue

FROM pizza\_types AS pt

INNER JOIN

pizzas AS p

ON p.pizza\_type\_id = pt.pizza\_type\_id

INNER JOIN

order\_details AS od

ON p.pizza\_id = od.pizza\_id

GROUP BY pizza\_name

ORDER BY revenue DESC

LIMIT 3;

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

WITH each\_category\_revenue AS(

SELECT

pt.category aS pizza\_category ,

SUM(p.price\*od.quantity)AS revenue

FROM pizza\_types AS pt

INNER JOIN

pizzas AS p

ON p.pizza\_type\_id = pt.pizza\_type\_id

INNER JOIN

order\_details AS od

ON p.pizza\_id = od.pizza\_id

GROUP BY pizza\_category

),

total\_revenue AS(

SELECT SUM(revenue) As total\_revenue

FROM each\_category\_revenue

)

SELECT

ecr.pizza\_category,

ecr.revenue,

tr. total\_revenue,

ROUND((ecr.revenue/tr. total\_revenue)\*100,2) As percentage\_contribution

FROM

each\_category\_revenue AS ecr

CROSS JOIN

total\_revenue AS tr

ORDER BY percentage\_contribution DESC;

-- 12Q: Analyze the cumulative revenue generated over time?

SELECT

o.order\_date AS date,

SUM(p.price \* od.quantity) AS daily\_revenue,

SUM(SUM(p.price \* od.quantity)) OVER (ORDER BY o.order\_date) AS cumulative\_revenue

FROM pizza\_types AS pt

INNER JOIN pizzas AS p ON p.pizza\_type\_id = pt.pizza\_type\_id

INNER JOIN order\_details AS od ON p.pizza\_id = od.pizza\_id

INNER JOIN orders AS o ON o.order\_id = od.order\_id

GROUP BY o.order\_date

ORDER BY o.order\_date;

**Another way:**

SELECT

date,SUM(revenue) OVER (ORDER BY date) AS cumulative\_revenue

FROM

(SELECT

o.order\_date AS date,

SUM(p.price \* od.quantity) AS revenue

FROM pizza\_types AS pt

INNER JOIN pizzas AS p ON p.pizza\_type\_id = pt.pizza\_type\_id

INNER JOIN order\_details AS od ON p.pizza\_id = od.pizza\_id

INNER JOIN orders AS o ON o.order\_id = od.order\_id

GROUP BY o.order\_date

ORDER BY o.order\_date);

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

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

WITH revenue\_per\_pizza AS (

SELECT

pt.category AS pizza\_category,

pt.name AS pizza\_name,

SUM(p.price \* od.quantity) AS revenue

FROM pizza\_types AS pt

INNER JOIN pizzas AS p ON p.pizza\_type\_id = pt.pizza\_type\_id

INNER JOIN order\_details AS od ON p.pizza\_id = od.pizza\_id

GROUP BY pt.category, pt.name

),

ranked\_pizzas AS (

SELECT

pizza\_category,

pizza\_name,

revenue,

RANK() OVER (PARTITION BY pizza\_category ORDER BY revenue DESC) AS ranking

FROM revenue\_per\_pizza

)

SELECT

pizza\_category,

pizza\_name,

revenue,

ranking

FROM ranked\_pizzas

WHERE ranking <= 3

ORDER BY pizza\_category, ranking;