Q1. What is the total amount each customer spent at the restaurant?

QUERY:

SELECT sales.customer\_id,

SUM(menu.price) AS total\_spend

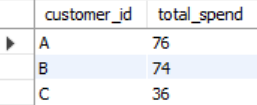
FROM sales

JOIN menu

ON sales.product\_id=menu.product\_id

GROUP BY sales.customer\_id

ANSWER:



Q2. How many days has each customer visited the restaurant?

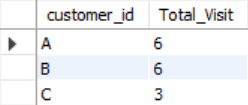
QUERY:

SELECT customer\_id, count(order\_date) AS Total\_Visit

FROM sales

GROUP BY customer\_id

Answer



Q3. What was the first item from the menu purchased by each customer?

QUERY:

SELECT customer\_id,

product\_name,

order\_date

FROM (

SELECT s.customer\_id,

m.product\_name,

s.order\_date,

ROW\_NUMBER() OVER (PARTITION BY s.customer\_id ORDER BY s.order\_date) AS rn

FROM sales s

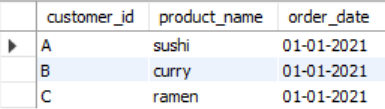
JOIN menu m

ON s.product\_id = m.product\_id

) t

WHERE rn = 1;

ANSWER



Q4. What is the most purchased item on the menu and how many times was it purchased by all customers?

QUERY

SELECT product\_id, COUNT(Product\_id) AS Total\_purchese

FROM Sales

GROUP BY product\_id

ORDER BY Total\_purchese DESC

LIMIT 1

ANSWER



Q5. Which item was the most popular for each customer?

QUERY

SELECT customer\_id, COUNT(Product\_id)

FROM sales

WHERE product\_id = 3

Group BY customer\_id

SELECT customer\_id, product\_id, purchase\_count

FROM (

    SELECT customer\_id,

           product\_id,

           COUNT(product\_id) AS purchase\_count,

           ROW\_NUMBER() OVER (

               PARTITION BY customer\_id

               ORDER BY COUNT(product\_id) DESC

           ) AS rn

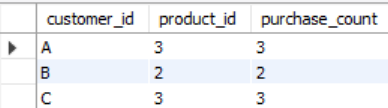
    FROM sales

    GROUP BY customer\_id, product\_id

) t

WHERE rn = 1;

ANSWER



Q6. Which item was purchased first by the customer after they became a member?

QUERY

SELECT customer\_id,

product\_name,

order\_date

FROM (

SELECT s.customer\_id,

m.product\_name,

s.order\_date,

ROW\_NUMBER() OVER (

PARTITION BY s.customer\_id

ORDER BY s.order\_date

) AS rn

FROM sales s

JOIN menu m

ON s.product\_id = m.product\_id

JOIN members mem

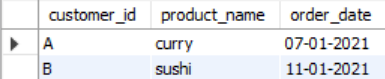
ON s.customer\_id = mem.customer\_id

WHERE s.order\_date >= mem.join\_date -- ✅ only sales after membership

) t

WHERE rn = 1;

Answer



Q7. Which item was purchased just before the customer became a member?

QUERY

SELECT customer\_id, product\_name, order\_date

FROM (

SELECT

s.customer\_id,

m.product\_name,

s.order\_date,

ROW\_NUMBER() OVER (

PARTITION BY s.customer\_id

ORDER BY s.order\_date

) AS rn

FROM sales s

JOIN menu m

ON s.product\_id = m.product\_id

JOIN members mem

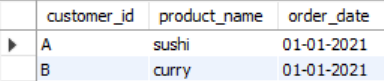
ON s.customer\_id = mem.customer\_id

WHERE s.order\_date < mem.join\_date -- before becoming member

) t

WHERE rn = 1;

ANSWER



Q8. What is the total items and amount spent for each member before they became a member?

QUERY

SELECT

sales.customer\_id,

SUM(menu.price) AS total\_spend

FROM sales

JOIN menu

ON sales.product\_id = menu.product\_id

JOIN members

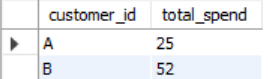
on sales.customer\_id = members.customer\_id

WHERE sales.order\_date < members.join\_date

GROUP BY sales.customer\_id

ORDER BY customer\_id

ANSWER



Q9. If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?

SELECT

sales.customer\_id,

SUM(

CASE

WHEN sales.product\_id < 3 THEN menu.price \* 10

ELSE menu.price \* 20

END

) AS total\_spend

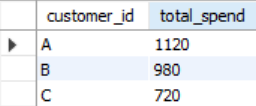
FROM sales

JOIN menu

ON sales.product\_id = menu.product\_id

GROUP BY sales.customer\_id;

ANSWER



Q10. In the first week after a customer joins the program (including their join date) they earn 2x points on all items, not just sushi - how many points do customer A and B have at the end of January?

QUERY

SELECT

sales.customer\_id,

SUM(

CASE

WHEN sales.order\_date <= DATE\_ADD(members.join\_date, INTERVAL 7 DAY)

THEN menu.price \* 20

ELSE menu.price \* 10

END

) AS Amount\_Spend

FROM sales

JOIN menu

ON sales.product\_id = menu.product\_id

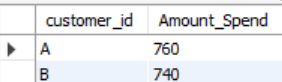
JOIN members

ON sales.customer\_id = members.customer\_id

GROUP BY sales.customer\_id

ORDER BY sales.customer\_id;

ANSWER



Q11. **Create a detailed dataset that combines customer orders with product details and membership status, and also assigns a ranking to each product purchased after the customer became a member.**

**QUERY**

SELECT sales.customer\_id,

sales.product\_id,

menu.product\_name,

CASE

WHEN members.join\_date IS NULL THEN 'N' -- never became a member

WHEN sales.order\_date < members.join\_date THEN 'N'

ELSE 'Y'

END AS membership,

CASE

WHEN members.join\_date IS NULL THEN 'N' -- never became a member

WHEN sales.order\_date < members.join\_date THEN 'N'

ELSE 'Y'

END AS ranking

FROM sales

JOIN menu

ON sales.product\_id = menu.product\_id

LEFT JOIN members

ON sales.customer\_id = members.customer\_id;

ANSWER

