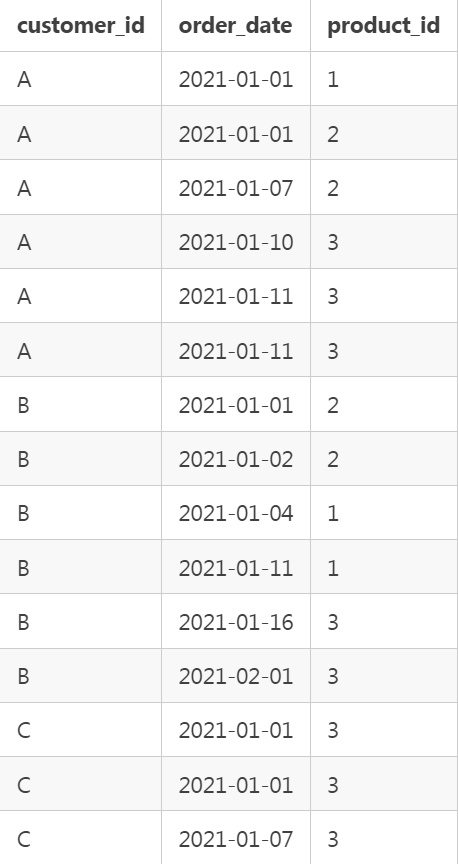


The schema query for MySQL.

CREATE TABLE sales   
(   
customer\_id VARCHAR(1),   
order\_date DATE,   
product\_id INTEGER); INSERT INTO sales   
(customer\_id, order\_date, product\_id)  
VALUES   
('A', '2021-01-01', '1'),   
('A', '2021-01-01', '2'),   
('A', '2021-01-07', '2'),   
('A', '2021-01-10', '3'),   
('A', '2021-01-11', '3'),   
('A', '2021-01-11', '3'),   
('B', '2021-01-01', '2'),   
('B', '2021-01-02', '2'),   
('B', '2021-01-04', '1'),   
('B', '2021-01-11', '1'),   
('B', '2021-01-16', '3'),   
('B', '2021-02-01', '3'),   
('C', '2021-01-01', '3'),   
('C', '2021-01-01', '3'),   
('C', '2021-01-07', '3'); CREATE TABLE menu   
(  
product\_id INTEGER,   
product\_name VARCHAR(5),   
price INTEGER); INSERT INTO menu   
(product\_id, product\_name, price)  
VALUES   
('1', 'sushi', '10'),   
('2', 'curry', '15'),   
('3', 'ramen', '12'); CREATE TABLE members   
(  
customer\_id VARCHAR(1),   
join\_date DATE); INSERT INTO members   
(customer\_id, join\_date)  
VALUES   
('A', '2021-01-07'),   
('B', '2021-01-09');

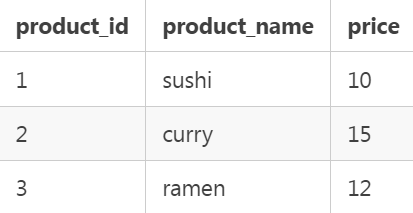
The tables

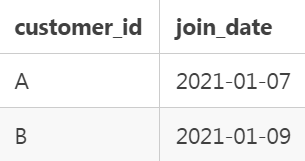


sales

menus

members

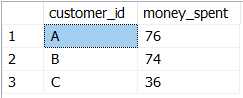




WHERE column\_name comparison\_operator ANY (subquery)

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

SELECT customer\_id,   
SUM(price) AS money\_spent   
FROM sales   
JOIN menu   
ON menu.product\_id = sales.product\_id  
GROUP BY customer\_id

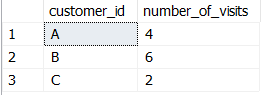


**-----------------------------------------------------------------------------------------------**

**2. How many days has each customer visited the restaurant?**

SELECT customer\_id,

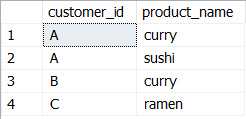
COUNT(DISTINCT(order\_date)) AS number\_of\_visits  
FROM sales  
GROUP BY customer\_id



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

SELECT DISTINCT(customer\_id),   
product\_name FROM sales s  
JOIN menu m   
ON m.product\_id = s.product\_id  
WHERE s.order\_date = ANY   
(  
SELECT MIN(order\_date)   
FROM sales   
GROUP BY customer\_id  
)

WHERE column\_name comparison\_operator ANY (subquery)



**---------------------------------------------------------------**

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

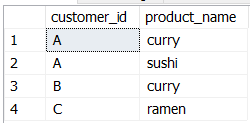
SELECT top (1)

COUNT(product\_name) AS count,   
product\_name FROM sales s   
JOIN menu m   
ON s.product\_id = m.product\_id  
GROUP BY product\_name   
ORDER BY count DESC



**5. Which item was the most popular for each customer?**

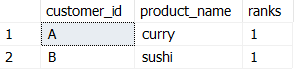
WITH r AS   
(  
SELECT s.customer\_id,

m.product\_name,  
COUNT(s.product\_id) as count,  
DENSE\_RANK() OVER (PARTITION BY s.customer\_id ORDER BY COUNT(s.product\_id) DESC) AS r  
FROM menu m   
JOIN sales s   
ON s.product\_id = m.product\_id  
GROUP BY s.customer\_id, s.product\_id, m.product\_name  
)   
SELECT customer\_id, product\_name, count  
FROM r  
WHERE r = 1

DENSE\_RANK() OVER (PARTITION BY expr1[{,expr2...}]

ORDER BY expr1 [ASC|DESC], [{,expr2...}])

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

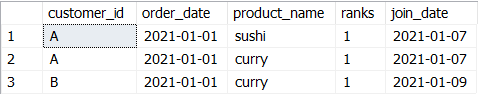
WITH ranks AS  
(  
SELECT s.customer\_id,  
m.product\_name,  
DENSE\_RANK() OVER (PARTITION BY s.customer\_id ORDER BY s.order\_date) AS ranks  
FROM sales s  
JOIN menu m ON s.product\_id = m.product\_id  
JOIN members AS mem  
ON mem.customer\_id = s.customer\_id  
WHERE s.order\_date >= mem.join\_date  
)  
SELECT \* FROM ranks  
WHERE ranks = 1

DENSE\_RANK() OVER (PARTITION BY expr1[{,expr2...}]

ORDER BY expr1 [ASC|DESC], [{,expr2...}])

-----------------------------------------------------------------------------------------------

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

WITH ranks AS  
(  
SELECT s.customer\_id,  
s.order\_date,  
m.product\_name,  
DENSE\_RANK() OVER (PARTITION BY s.customer\_id ORDER BY s.order\_date) AS ranks, mem.join\_date  
FROM sales s  
JOIN menu m ON s.product\_id = m.product\_id  
JOIN members AS mem  
ON mem.customer\_id = s.customer\_id  
WHERE s.order\_date < mem.join\_date  
)  
SELECT \* FROM ranks  
WHERE ranks = 1

DENSE\_RANK() OVER (PARTITION BY expr1[{,expr2...}]

ORDER BY expr1 [ASC|DESC], [{,expr2...}])

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

SELECT s.customer\_id,  
count(s.product\_id) AS total\_items,   
SUM(price) AS money\_spent  
FROM sales AS s  
JOIN menu AS m   
ON m.product\_id = s.product\_id  
JOIN members AS mem   
ON s.customer\_id = mem.customer\_id  
WHERE s.order\_date < mem.join\_date  
GROUP BY s.customer\_id



-----------------------------------------------------------------------------------------------

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

WITH points AS   
(  
SELECT \*,  
CASE   
WHEN m.product\_name = 'sushi' THEN price \* 20  
WHEN m.product\_name != 'sushi' THEN price \* 10  
END AS points  
FROM menu m  
)  
SELECT customer\_id, SUM(points) AS points  
FROM sales s  
JOIN points p ON p.product\_id = s.product\_id  
GROUP BY s.customer\_id

CASE expression

WHEN when\_expression\_1 THEN

result\_1

WHEN when\_expression\_2 THEN

result\_2

WHEN when\_expression\_3 THEN

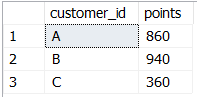
result\_3

...

ELSE

else\_result

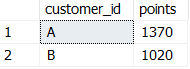
END



**10. 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?**

select customer\_id,  
sum(points) points  
from (select s.customer\_id,  
case

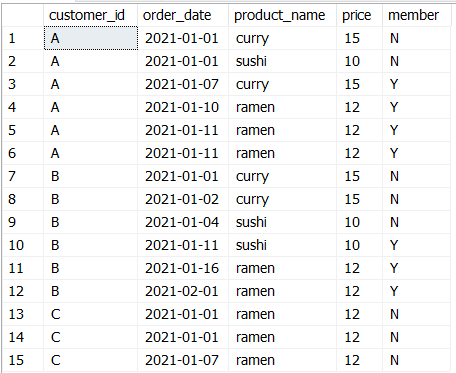
when product\_name = 'sushi' and  
s.order\_date between dateadd(day,-1,ms.join\_date) and dateadd(day, 6, ms.join\_date) then m.price\*40  
when product\_name = 'sushi' or  
s.order\_date between dateadd(day,-1,ms.join\_date) and dateadd(day, 6, ms.join\_date) then m.price\*20  
else price\*10 end points  
from members ms  
left join sales s on s.customer\_id = ms.customer\_id  
left join menu m on s.product\_id = m.product\_id  
where s.order\_date <= '20210131') a  
group by customer\_id;



**Bonus Questions**

1. **Recreate the table**

SELECT s.customer\_id,  
s.order\_date,  
m.product\_name,  
m.price,  
CASE   
WHEN s.order\_date >= mem.join\_date THEN 'Y'  
ELSE 'N'   
END AS member  
FROM sales s  
LEFT JOIN menu m ON m.product\_id = s.product\_id  
LEFT JOIN members mem ON mem.customer\_id = s.customer\_id  
ORDER BY customer\_id, order\_date, price DESC



**2. Rank Members — fill non-members with null**

select \*,

case when member = 'Y' then rank() over (partition by customer\_id, member order by order\_date)

else NULL end ranking

from (select s.customer\_id,

s.order\_date,

m.product\_name,

m.price,

case when s.order\_date >= ms.join\_date then 'Y'

else 'N' end member

from sales s

left join menu m on s.product\_id = m.product\_id

left join members ms on s.customer\_id = ms.customer\_id) a

order by customer\_id, order\_date, product\_name;

