Case Study #1 - Danny's Diner

Krishna Murthy

Problem Statement:

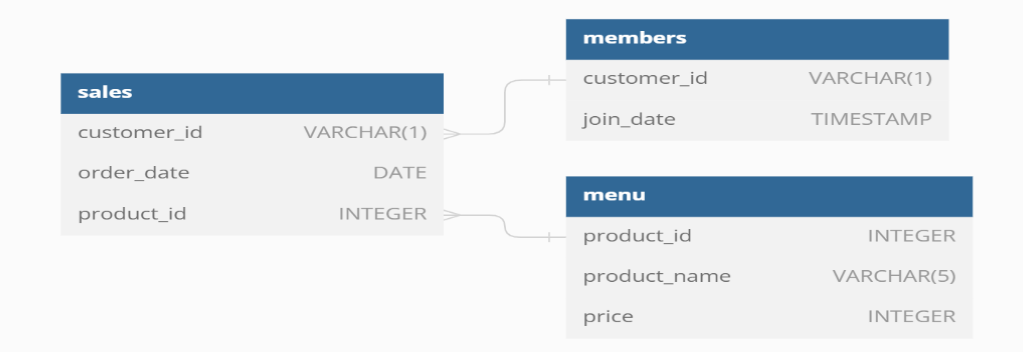
Danny wants to use the data to answer a few simple questions about his customers, especially about their visiting patterns, how much money they've spent and also which menu items are their favourite. Having this deeper connection with his customers will help him deliver a better and more personalised experience for his loyal customers. He plans on using these insights to help him decide whether he should expand the existing customer loyalty program.

Datasets used

Danny has shared with you 3 key datasets for this case study:

- sales: The sales table captures all customer_id level purchases with a corresponding order_date and product_id information for when and what menu items were ordered.
- menu: The menu table maps the product_id to the actual product_name and price of each menu item.
- members: The members table captures the join_date when a customer_id joined the beta version of the Danny's Diner loyalty program.

Entity Relationship Diagram:



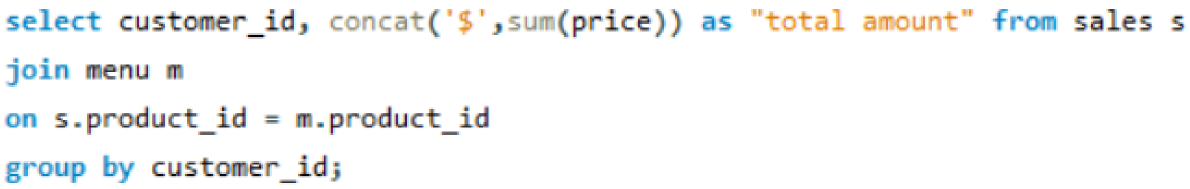
sales	
customer_id	VARCHAR(1)
order_date	DATE
product_id	INTEGER

members	
customer_id	VARCHAR(1)
join_date	TIMESTAMP

menu	
product_id	INTEGER
product_name	VARCHAR(5)
price	INTEGER

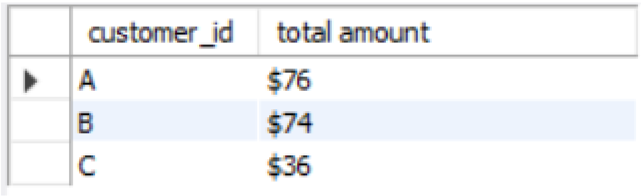
CASE STUDY QUESTIONS

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



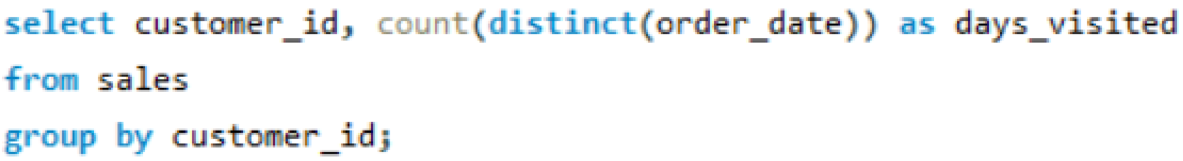
```
select customer_id, concat('$',sum(price)) as "total amount" from sales s
join menu m
on s.product_id = m.product_id
group by customer_id;
```

Answer:



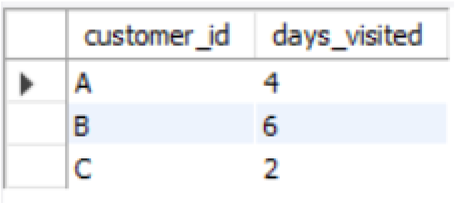
customer_id	total amount
A	$76
B	$74
C	$36

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



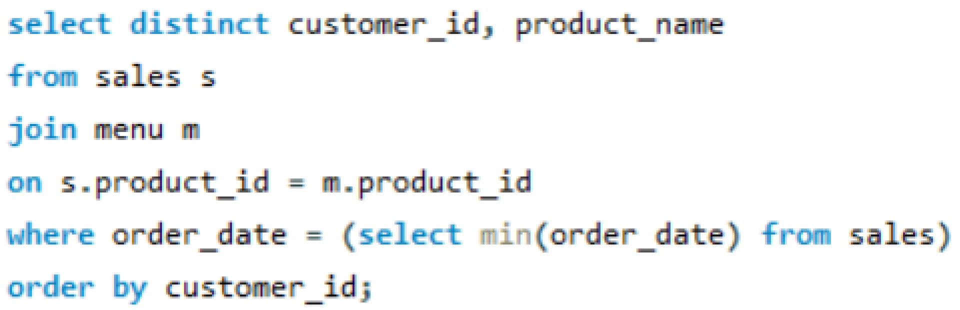
```
select customer_id, count(distinct(order_date)) as days_visited
from sales
group by customer_id;
```

Answer:



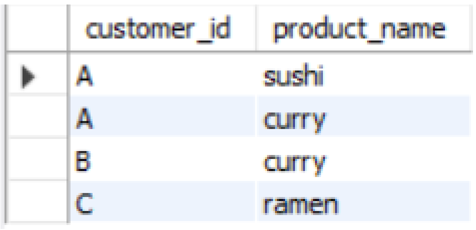
customer_id	days_visited
A	4
B	6
C	2

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



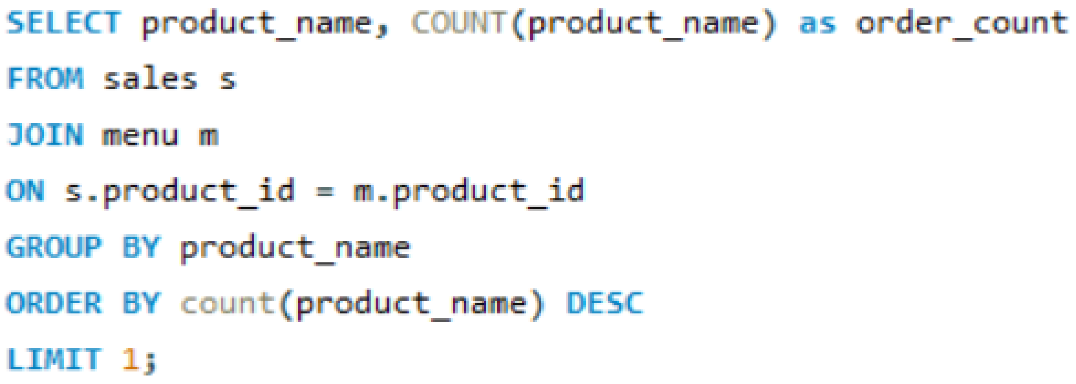
```
select distinct customer_id, product_name
from sales s
join menu m
on s.product_id = m.product_id
where order_date = (select min(order_date) from sales)
order by customer_id;
```

Answer



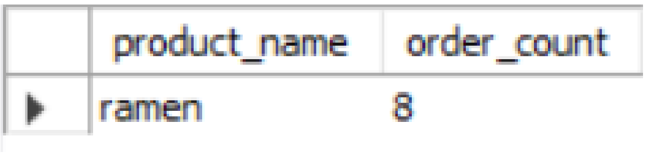
customer_id	product_name
A	sushi
A	curry
B	curry
C	ramen

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



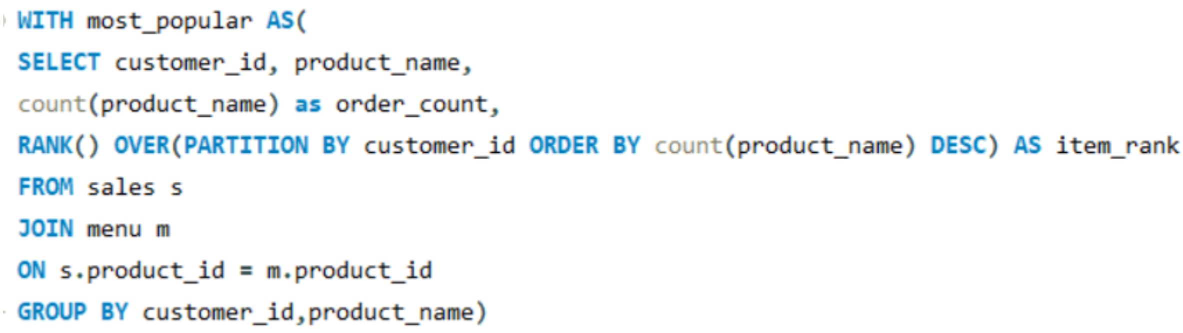
```
SELECT product_name, COUNT(product_name) as order_count
FROM sales s
JOIN menu m
ON s.product_id = m.product_id
GROUP BY product_name
ORDER BY count(product_name) DESC
LIMIT 1;
```

Answer:

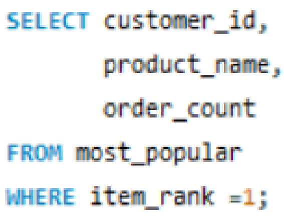


product_name	order_count
ramen	8

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

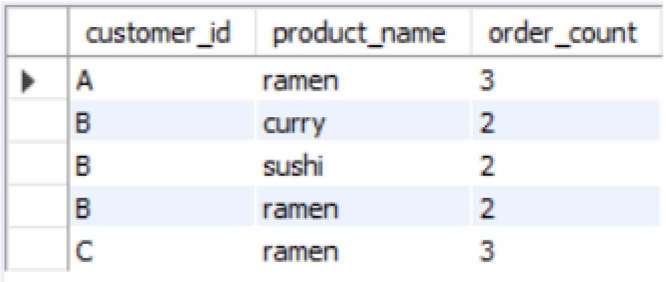


```
WITH most_popular AS(
SELECT customer_id, product_name,
count(product_name) as order_count,
RANK() OVER(PARTITION BY customer_id ORDER BY count(product_name) DESC) AS item_rank
FROM sales s
JOIN menu m
ON s.product_id = m.product_id
GROUP BY customer_id,product_name)
```



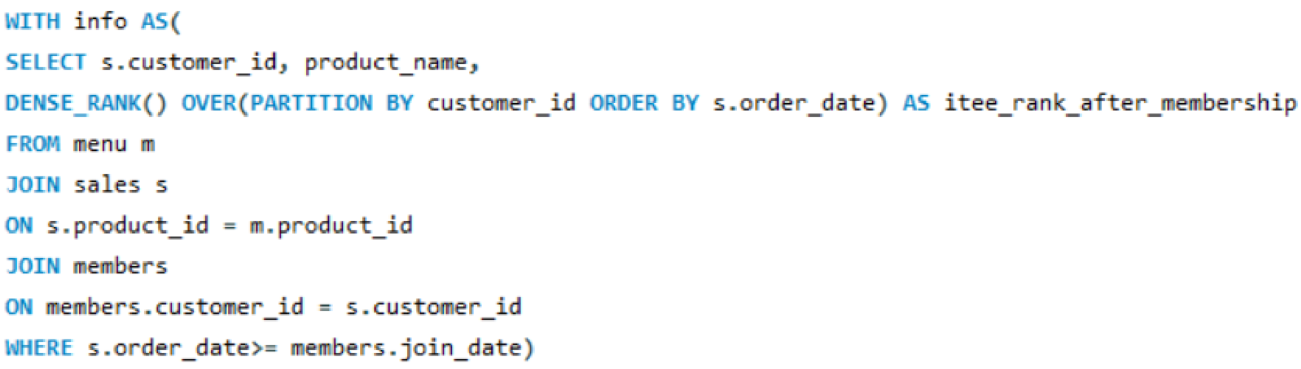
```
SELECT customer_id,
       product_name,
       order_count
FROM most_popular
WHERE item_rank =1;
```

Answer:

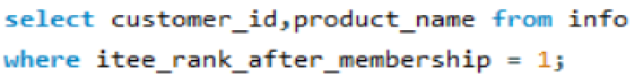


	customer_id	product_name	order_count
▶	A	ramen	3
	B	curry	2
	B	sushi	2
	B	ramen	2
	C	ramen	3

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

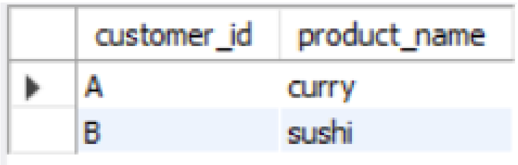


```
WITH info AS(
SELECT s.customer_id, product_name,
DENSE_RANK() OVER(PARTITION BY customer_id ORDER BY s.order_date) AS itee_rank_after_membership
FROM menu m
JOIN sales s
ON s.product_id = m.product_id
JOIN members
ON members.customer_id = s.customer_id
WHERE s.order_date>= members.join_date)
```



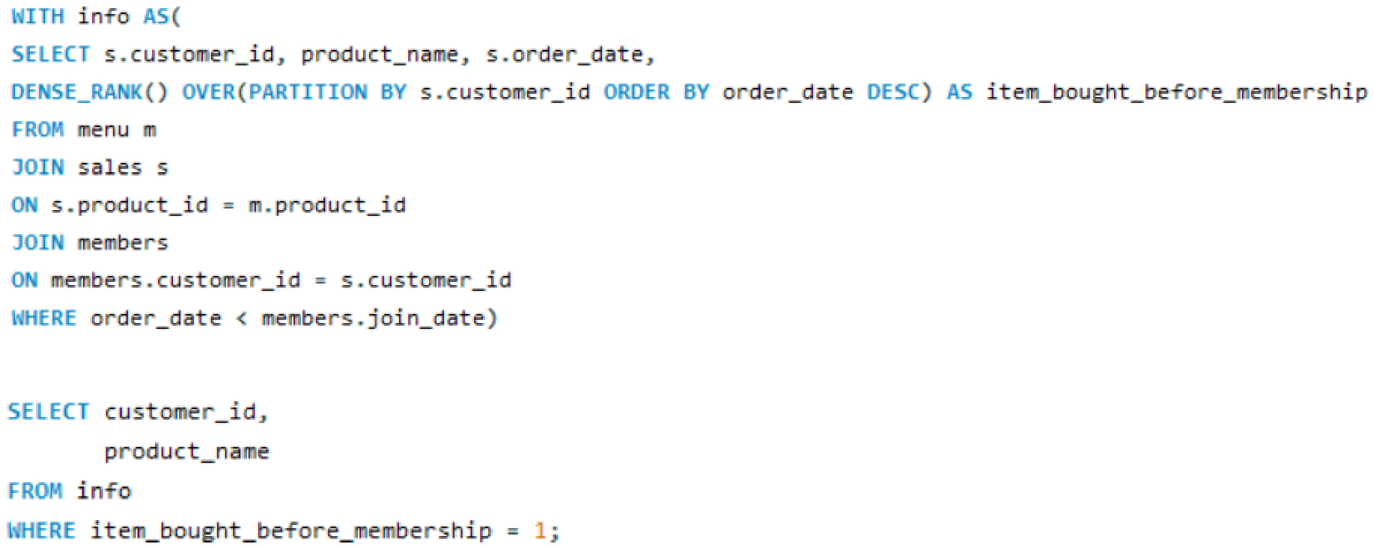
```
select customer_id,product_name from info
where itee_rank_after_membership = 1;
```

Answer:



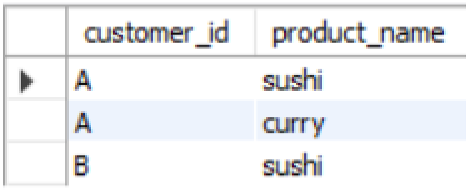
	customer_id	product_name
▶	A	curry
	B	sushi

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



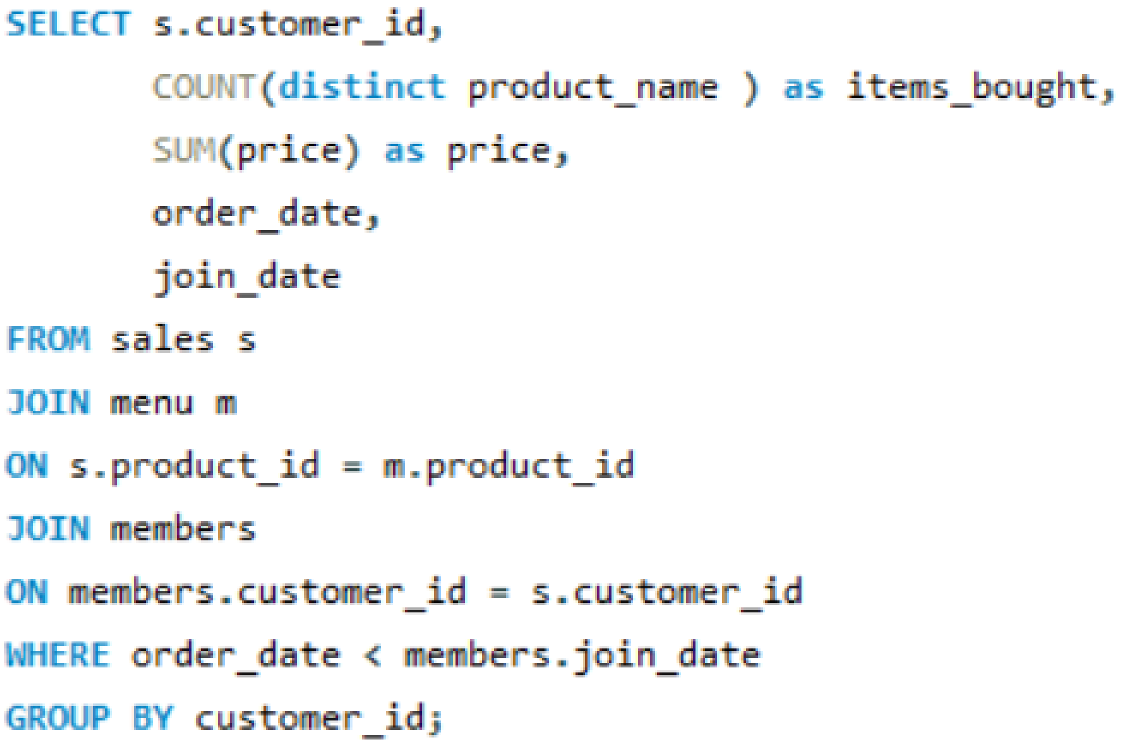
```
WITH info AS(
SELECT s.customer_id, product_name, s.order_date,
DENSE_RANK() OVER(PARTITION BY s.customer_id ORDER BY order_date DESC) AS item_bought_before_membership
FROM menu m
JOIN sales s
ON s.product_id = m.product_id
JOIN members
ON members.customer_id = s.customer_id
WHERE order_date < members.join_date)


SELECT customer_id,
       product_name
FROM info
WHERE item_bought_before_membership = 1;
```

Answer:



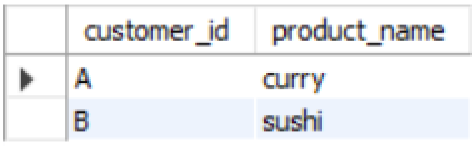
	customer_id	product_name
▶	A	sushi
	A	curry
	B	sushi

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



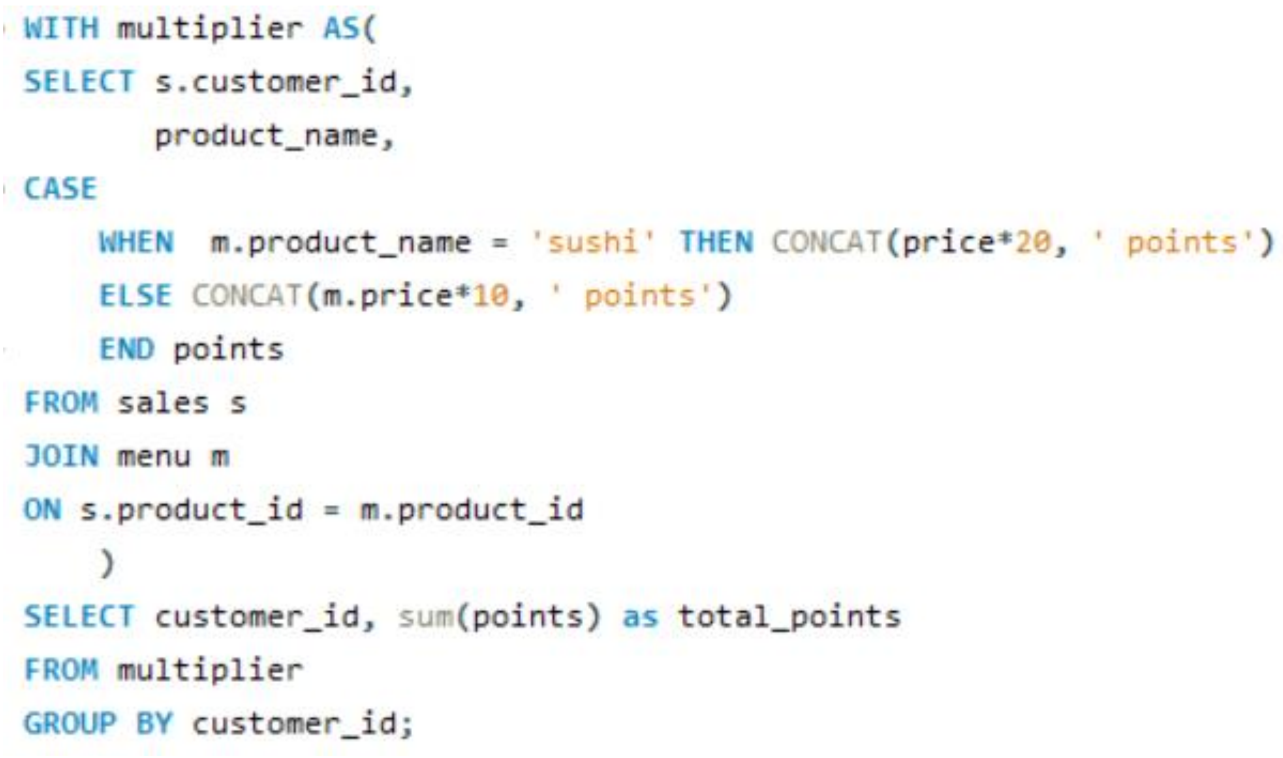
```
SELECT s.customer_id,
       COUNT(distinct product_name ) as items_bought,
       SUM(price) as price,
       order_date,
       join_date
FROM sales s
JOIN menu m
ON s.product_id = m.product_id
JOIN members
ON members.customer_id = s.customer_id
WHERE order_date < members.join_date
GROUP BY customer_id;
```

Answer:



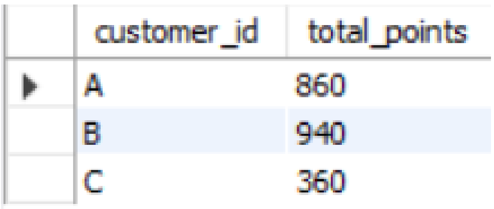
	customer_id	product_name
▶	A	curry
	B	sushi

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



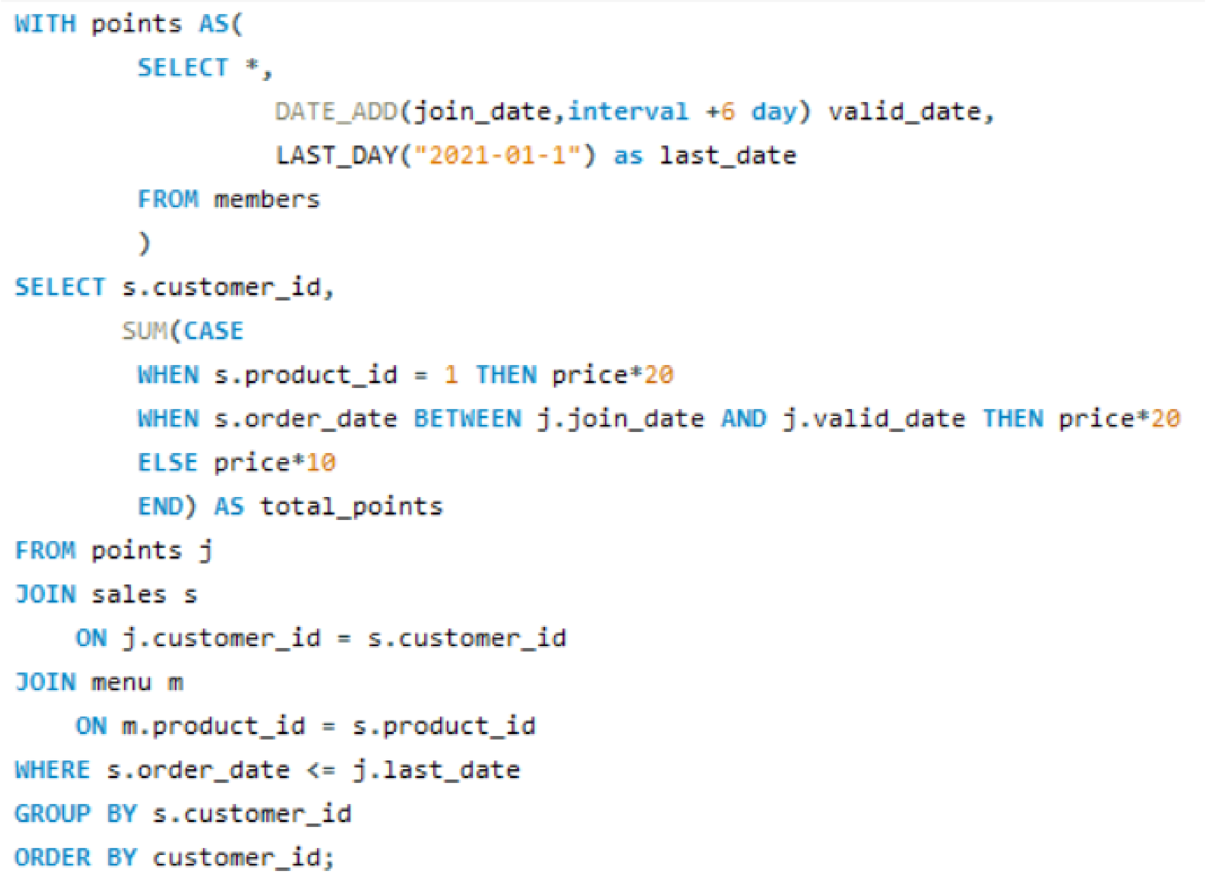
```
WITH multiplier AS(
SELECT s.customer_id,
       product_name,
CASE
    WHEN  m.product_name = 'sushi' THEN CONCAT(price*20, ' points')
    ELSE CONCAT(m.price*10, ' points')
    END points
FROM sales s
JOIN menu m
ON s.product_id = m.product_id
    )
SELECT customer_id, sum(points) as total_points
FROM multiplier
GROUP BY customer_id;
```

Answer:



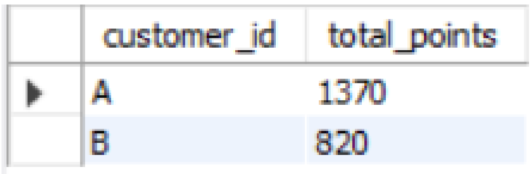
	customer_id	total_points
▶	A	860
	B	940
	C	360

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?



```
WITH points AS(
        SELECT *,
                 DATE_ADD(join_date,interval +6 day) valid_date,
                 LAST_DAY("2021-01-1") as last_date
        FROM members
        )
SELECT s.customer_id,
       SUM(CASE
        WHEN s.product_id = 1 THEN price*20
        WHEN s.order_date BETWEEN j.join_date AND j.valid_date THEN price*20
        ELSE price*10
        END) AS total_points
FROM points j
JOIN sales s
    ON j.customer_id = s.customer_id
JOIN menu m
    ON m.product_id = s.product_id
WHERE s.order_date <= j.last_date
GROUP BY s.customer_id
ORDER BY customer_id;
```

Answer:



	customer_id	total_points
▶	A	1370
	B	820