DANNY'S DINER

The taste of success

The SQL CASE 1

INTRODUCTION

Danny seriously loves Japanese food so in the beginning of 2021, he decides to embark upon a risky venture and opens up a cute little restaurant that sells his 3 favourite foods: sushi, curry and ramen.

Danny's Diner is in need of your assistance to help the restaurant stay afloat - the restaurant has captured some very basic data from their few months of operation but have no idea how to use their data to help them run the business.

PROBLEM STATEMENT

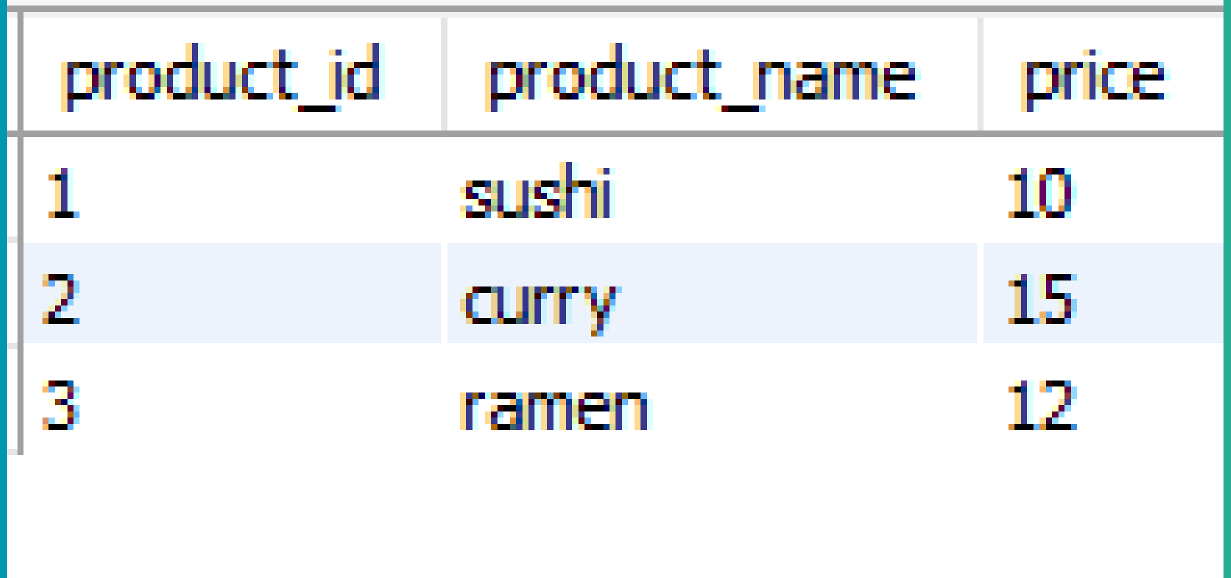
Danny wants to use the data to answer a few simple questions about his customers, especially about their visiting patterns, how much money they’ve spent and also which menu items are their favourite. Having this deeper connection with his customers will help him deliver a better and more personalised experience for his loyal customers.

He plans on using these insights to help him decide whether he should expand the existing customer loyalty program - additionally he needs help to generate some basic datasets so his team can easily inspect the data without needing to use SQL.

Danny has provided you with a sample of his overall customer data due to privacy issues - but he hopes that these examples are enough for you to write fully functioning SQL queries to help him answer his questions!

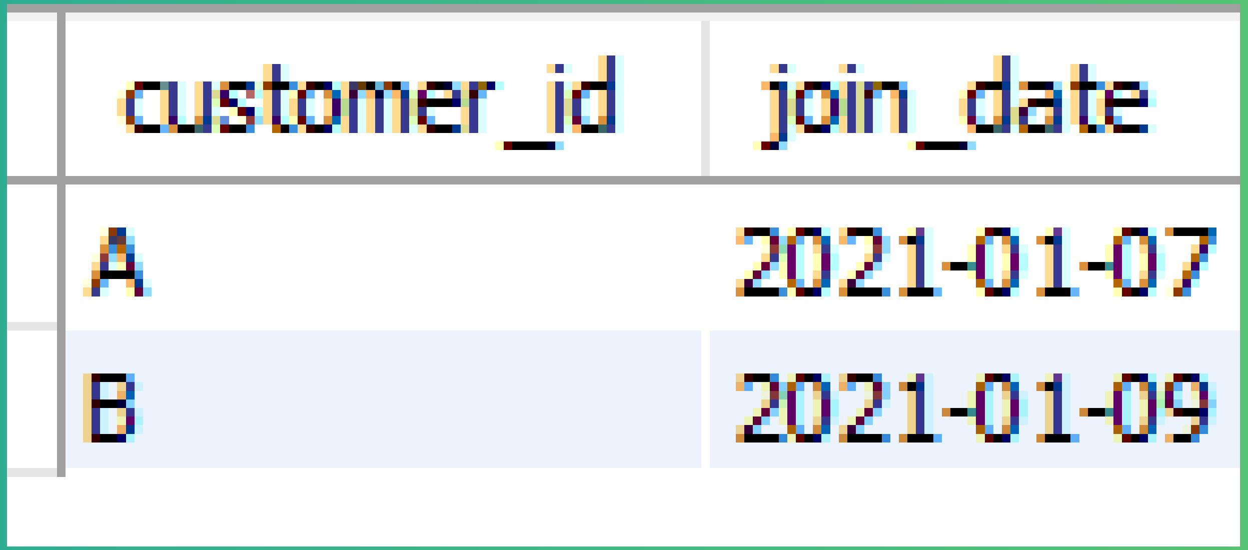
DATA SETS

MENU



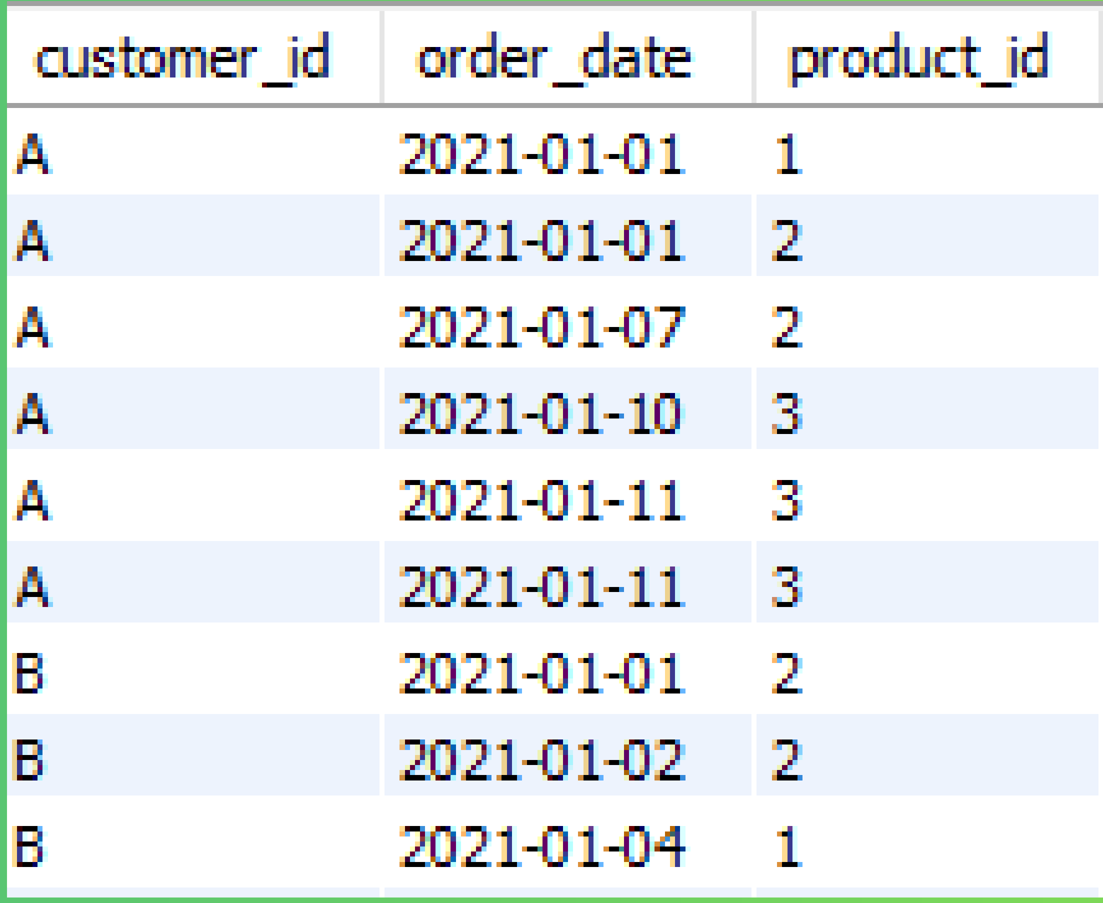
product_id	product_name	price
1	sushi	10
2	curry	15
3	ramen	12

MEMBERS



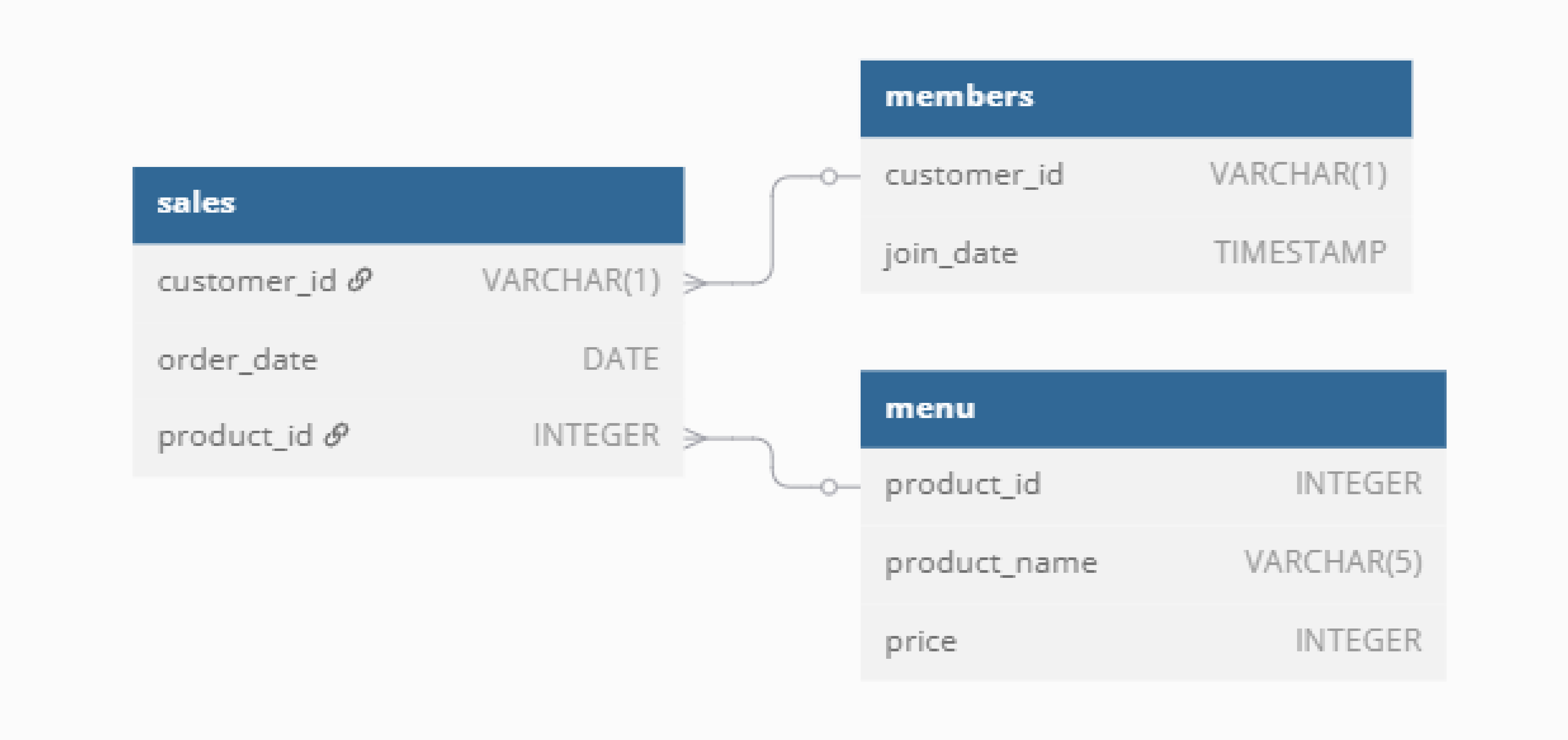
customer_id	join_date
A	2021-01-07
B	2021-01-09

SALES



customer_id	order_date	product_id
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

Entity Relationship Diagram

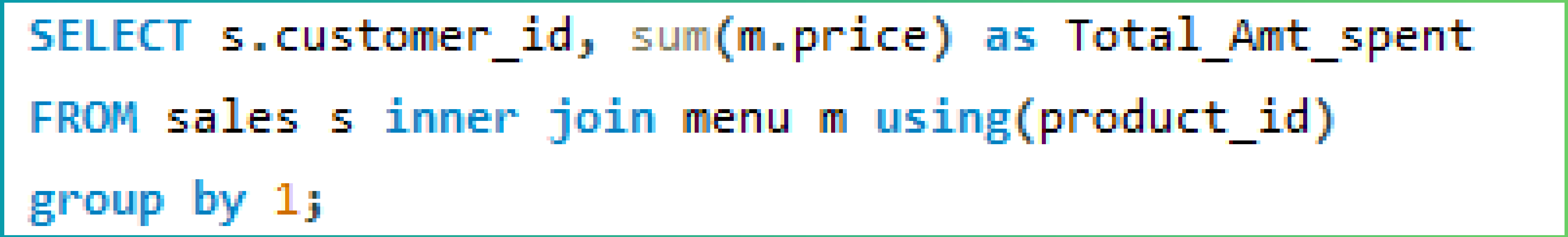


sales	
customer_id 🔗	VARCHAR(1)
order_date	DATE
product_id 🔗	INTEGER

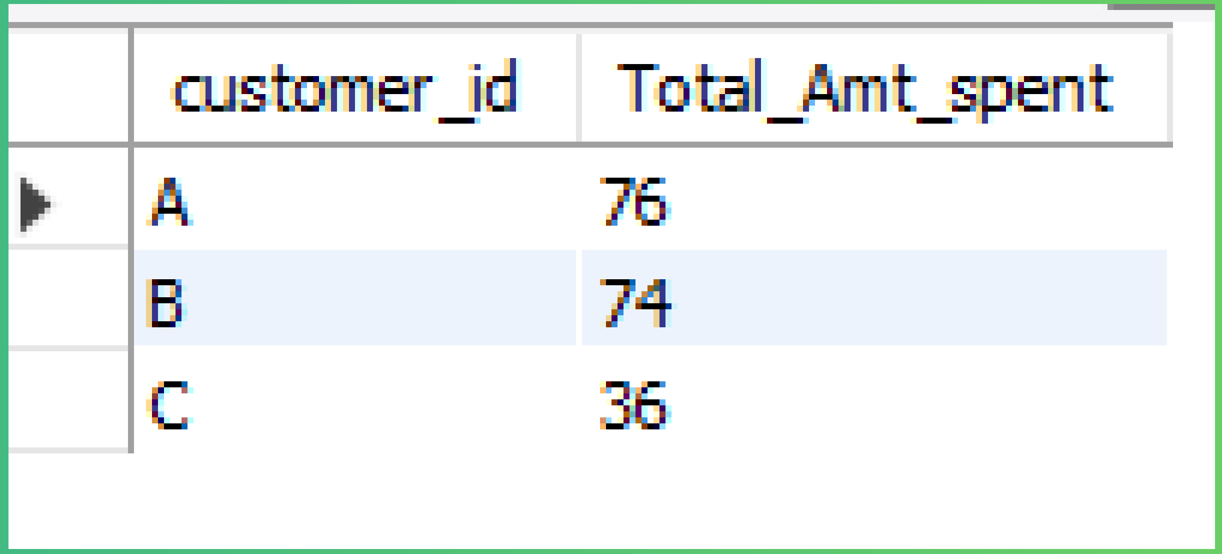
members	
customer_id	VARCHAR(1)
join_date	TIMESTAMP

menu	
product_id	INTEGER
product_name	VARCHAR(5)
price	INTEGER

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

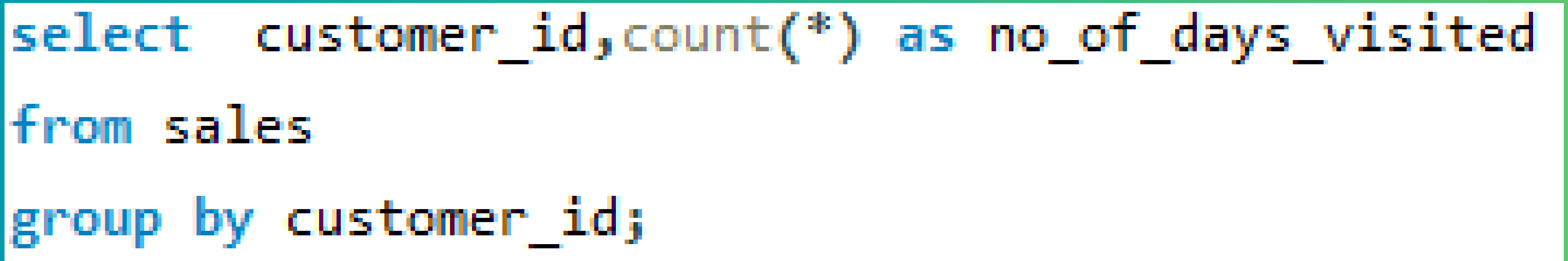


```sql
SELECT s.customer_id, sum(m.price) as Total_Amt_spent
FROM sales s inner join menu m using(product_id)
group by 1;
```

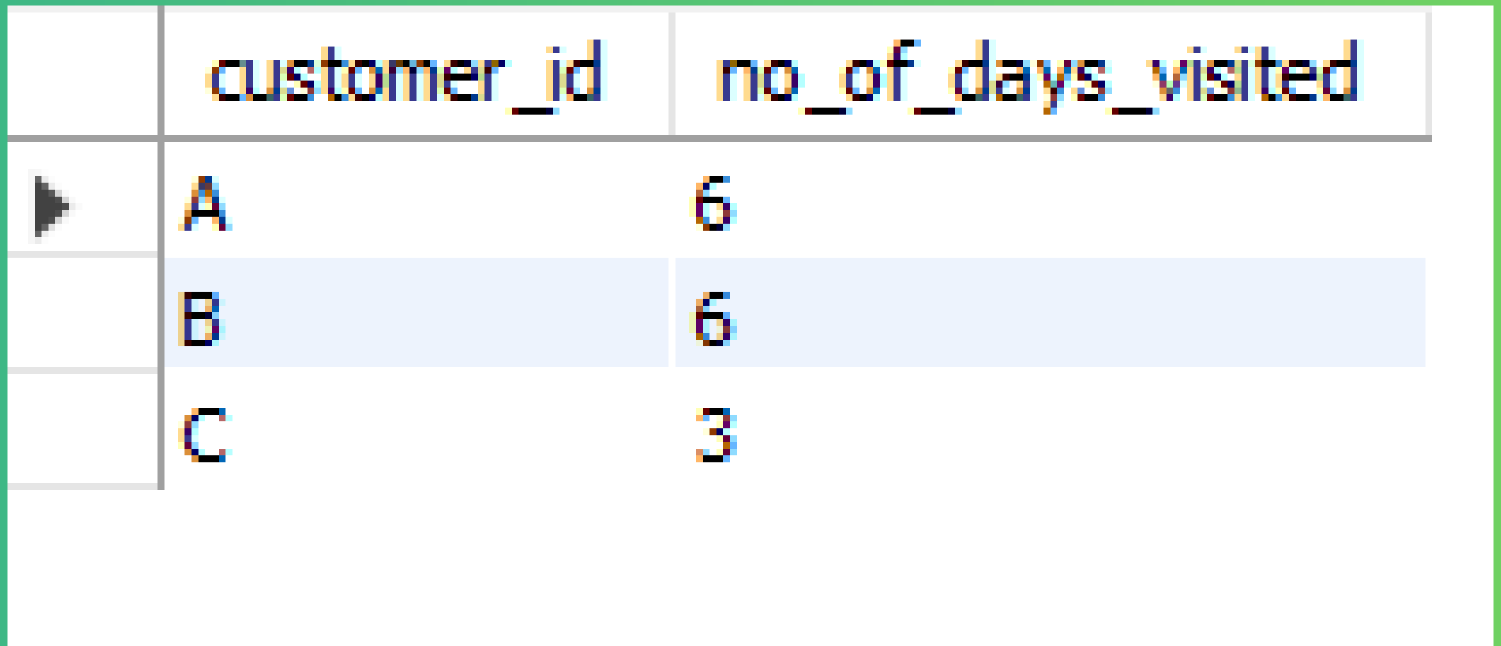


customer_id	Total_Amt_spent
A	76
B	74
C	36

How many days has each customer visited the restaurant?

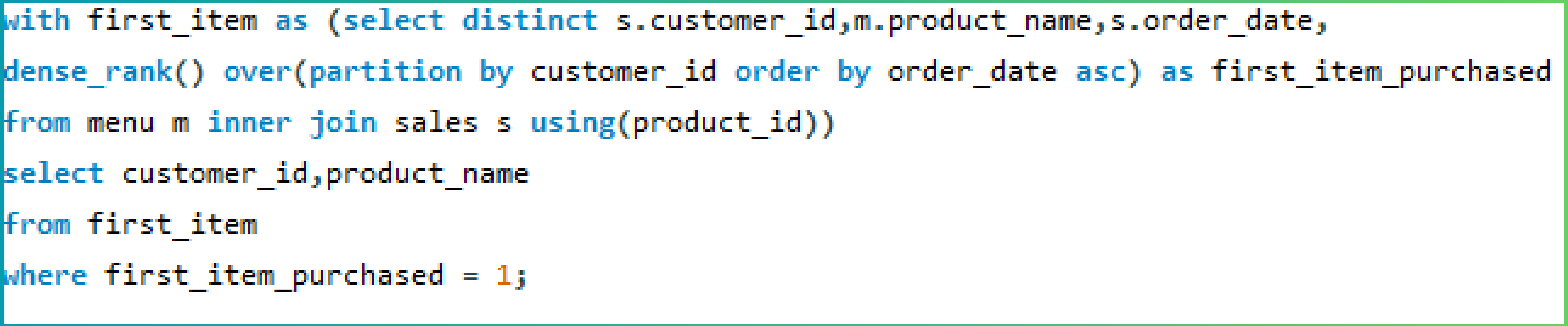


```
select  customer_id,count(*) as no_of_days_visited
from sales
group by customer_id;
```

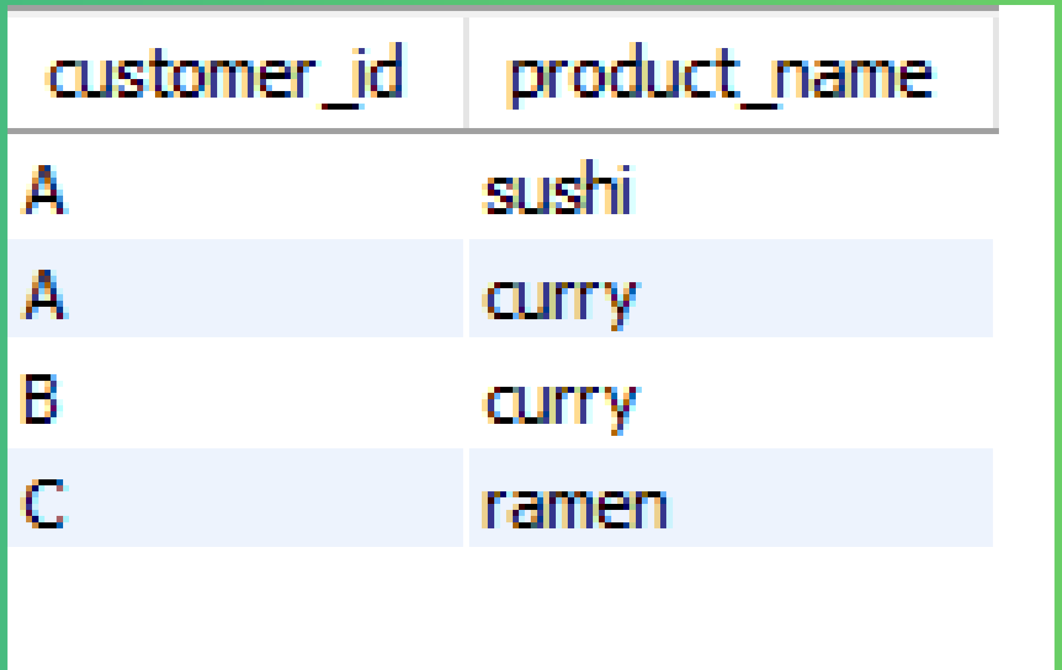


	customer_id	no_of_days_visited
▶	A	6
	B	6
	C	3

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

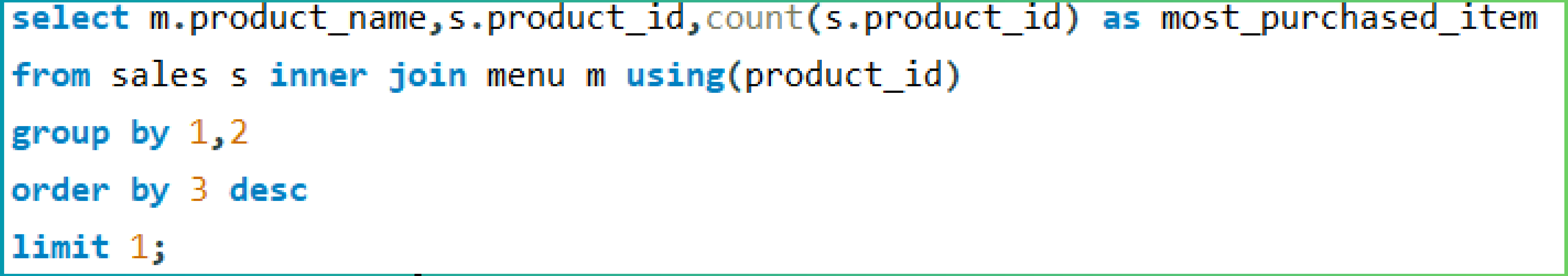


```
with first_item as (select distinct s.customer_id,m.product_name,s.order_date,
dense_rank() over(partition by customer_id order by order_date asc) as first_item_purchased
from menu m inner join sales s using(product_id))
select customer_id,product_name
from first_item
where first_item_purchased = 1;
```

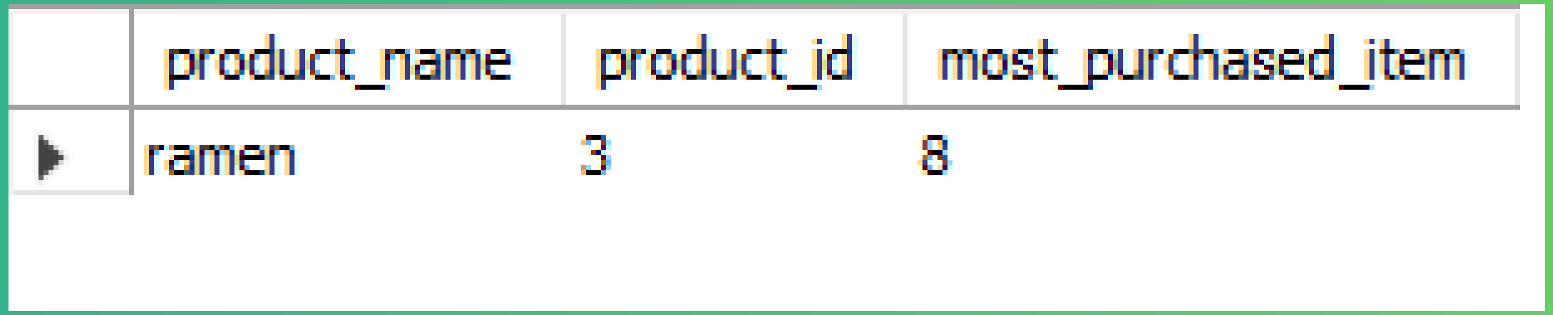


customer_id	product_name
A	sushi
A	curry
B	curry
C	ramen

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

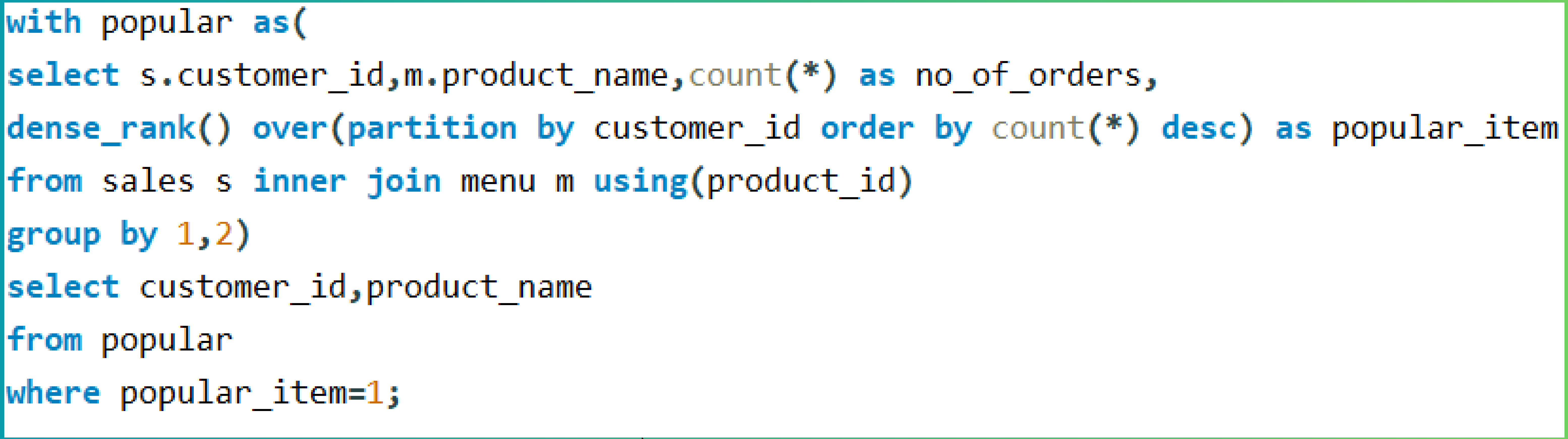


```
select m.product_name,s.product_id,count(s.product_id) as most_purchased_item
from sales s inner join menu m using(product_id)
group by 1,2
order by 3 desc
limit 1;
```

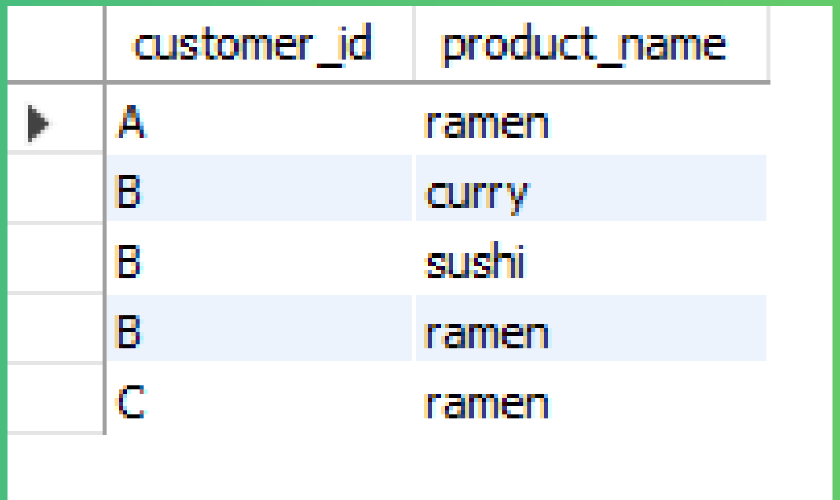


product_name	product_id	most_purchased_item
ramen	3	8

Which item was the most popular for each customer?

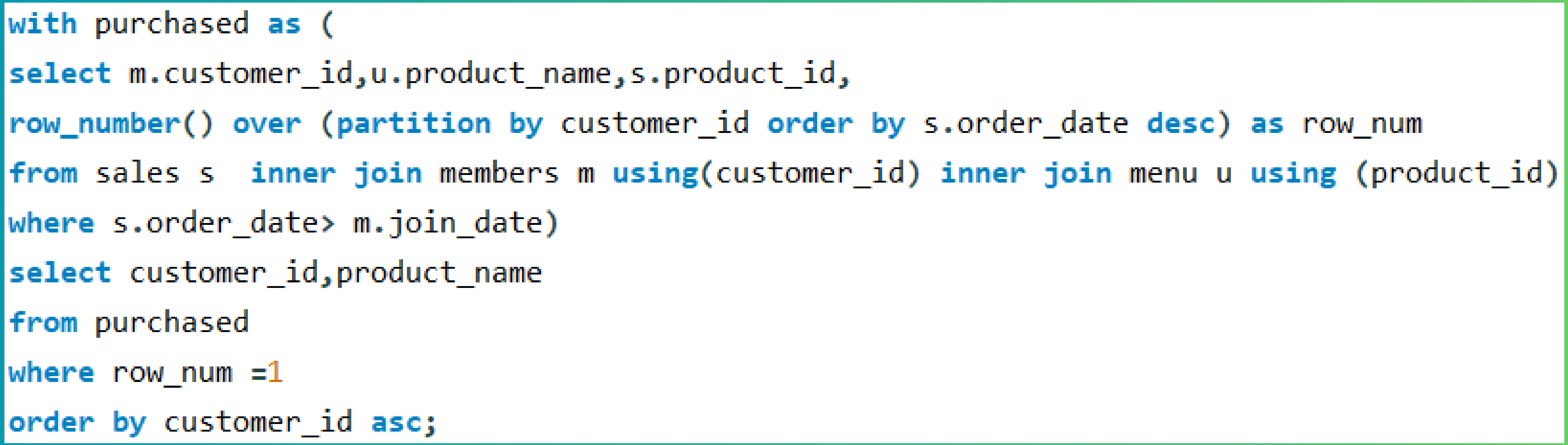


```
with popular as(
select s.customer_id,m.product_name,count(*) as no_of_orders,
dense_rank() over(partition by customer_id order by count(*) desc) as popular_item
from sales s inner join menu m using(product_id)
group by 1,2)
select customer_id,product_name
from popular
where popular_item=1;
```

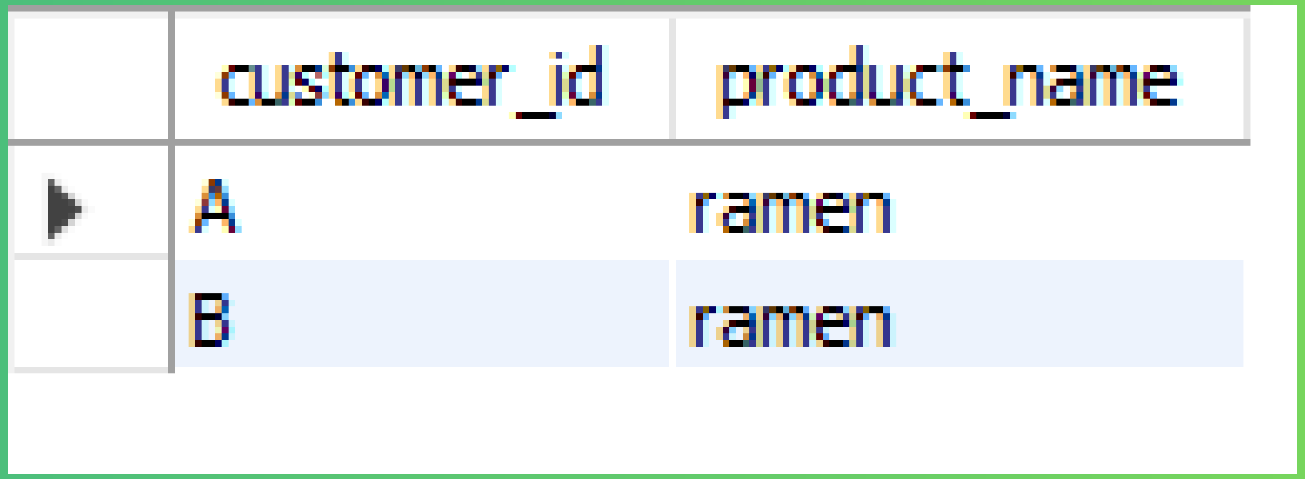


customer_id	product_name
A	ramen
B	curry
B	sushi
B	ramen
C	ramen

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

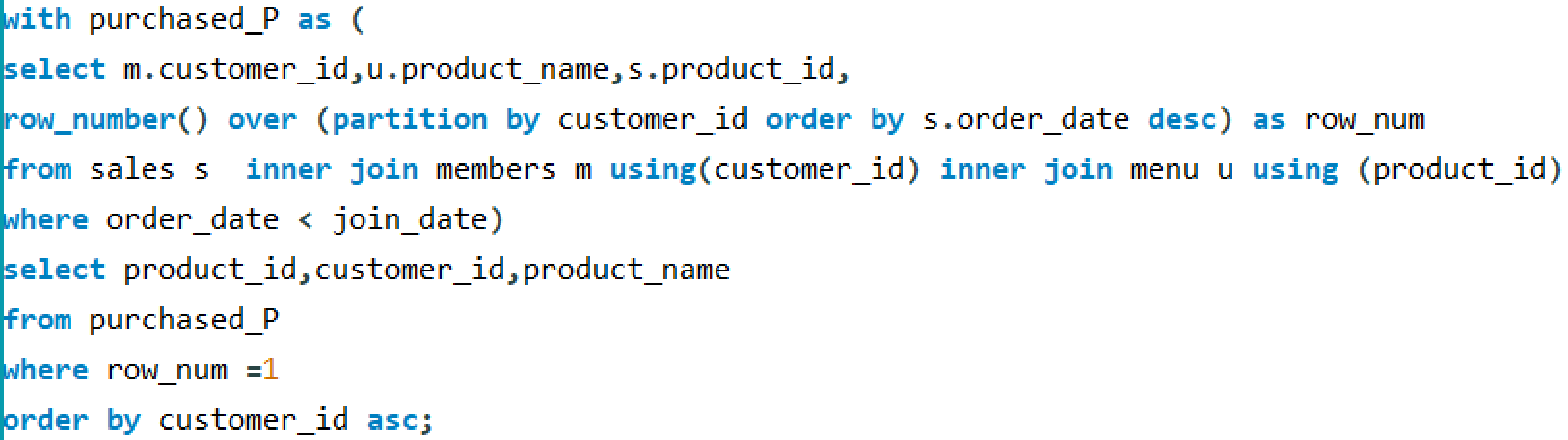


```sql
with purchased as (
select m.customer_id,u.product_name,s.product_id,
row_number() over (partition by customer_id order by s.order_date desc) as row_num
from sales s  inner join members m using(customer_id) inner join menu u using (product_id)
where s.order_date> m.join_date)
select customer_id,product_name
from purchased
where row_num =1
order by customer_id asc;
```

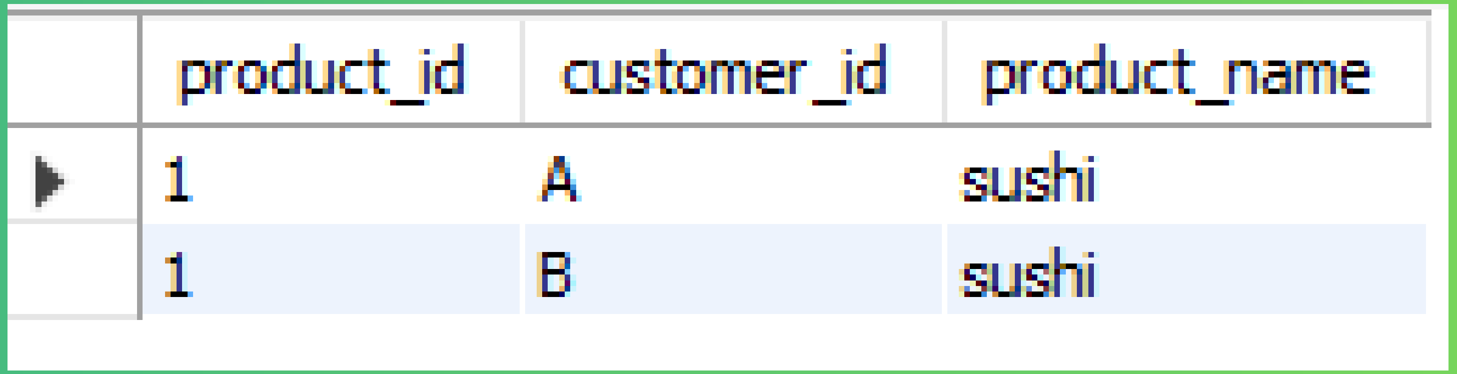


customer_id	product_name
A	ramen
B	ramen

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

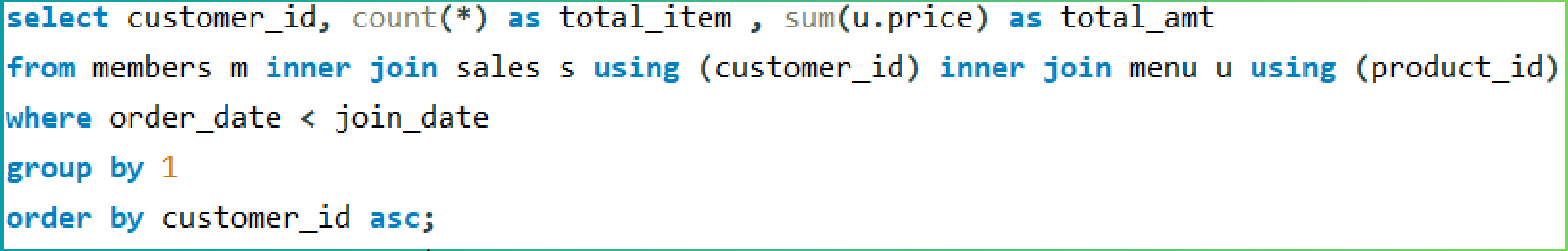


```
with purchased_P as (
select m.customer_id,u.product_name,s.product_id,
row_number() over (partition by customer_id order by s.order_date desc) as row_num
from sales s  inner join members m using(customer_id) inner join menu u using (product_id)
where order_date < join_date)
select product_id,customer_id,product_name
from purchased_P
where row_num =1
order by customer_id asc;
```

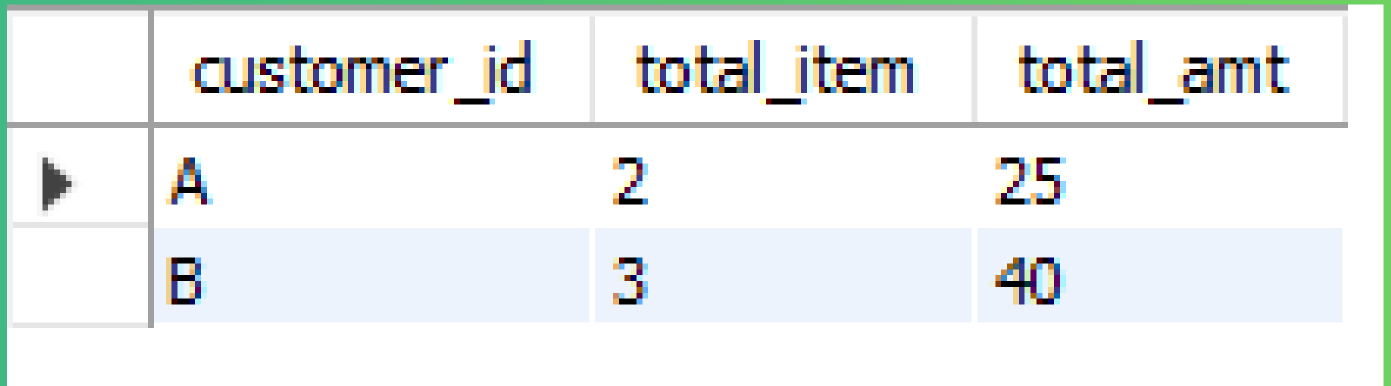


	product_id	customer_id	product_name
▶	1	A	sushi
	1	B	sushi

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

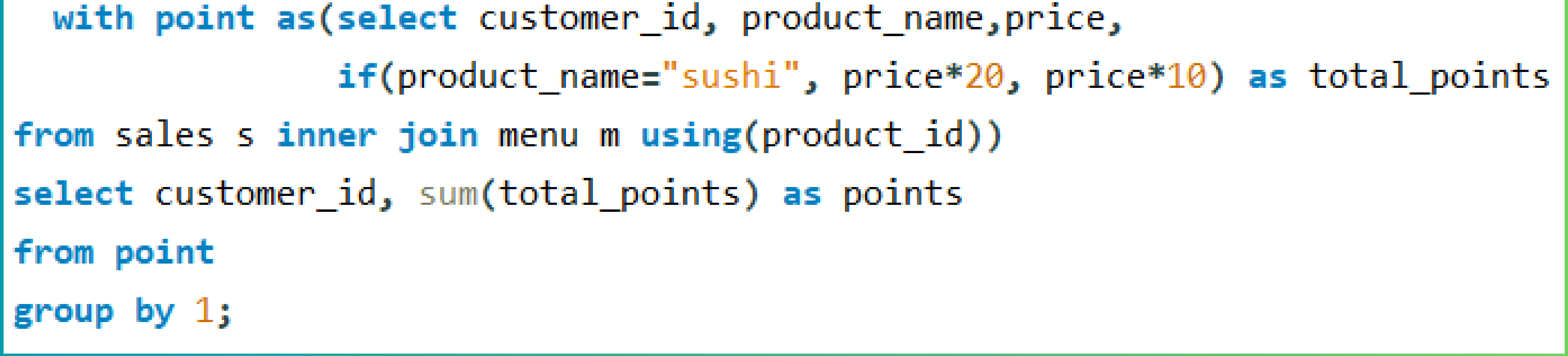


```
select customer_id, count(*) as total_item , sum(u.price) as total_amt
from members m inner join sales s using (customer_id) inner join menu u using (product_id)
where order_date < join_date
group by 1
order by customer_id asc;
```

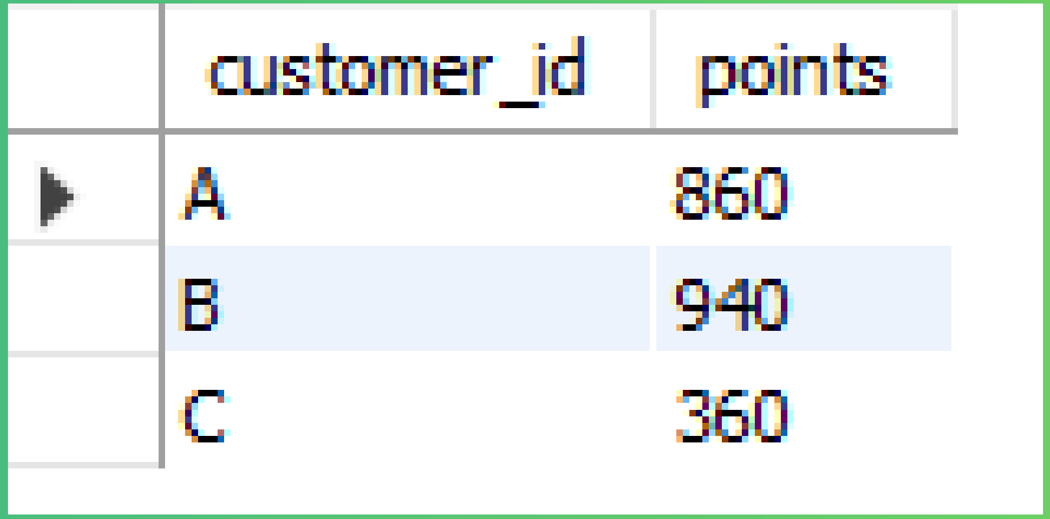


customer_id	total_item	total_amt
A	2	25
B	3	40

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

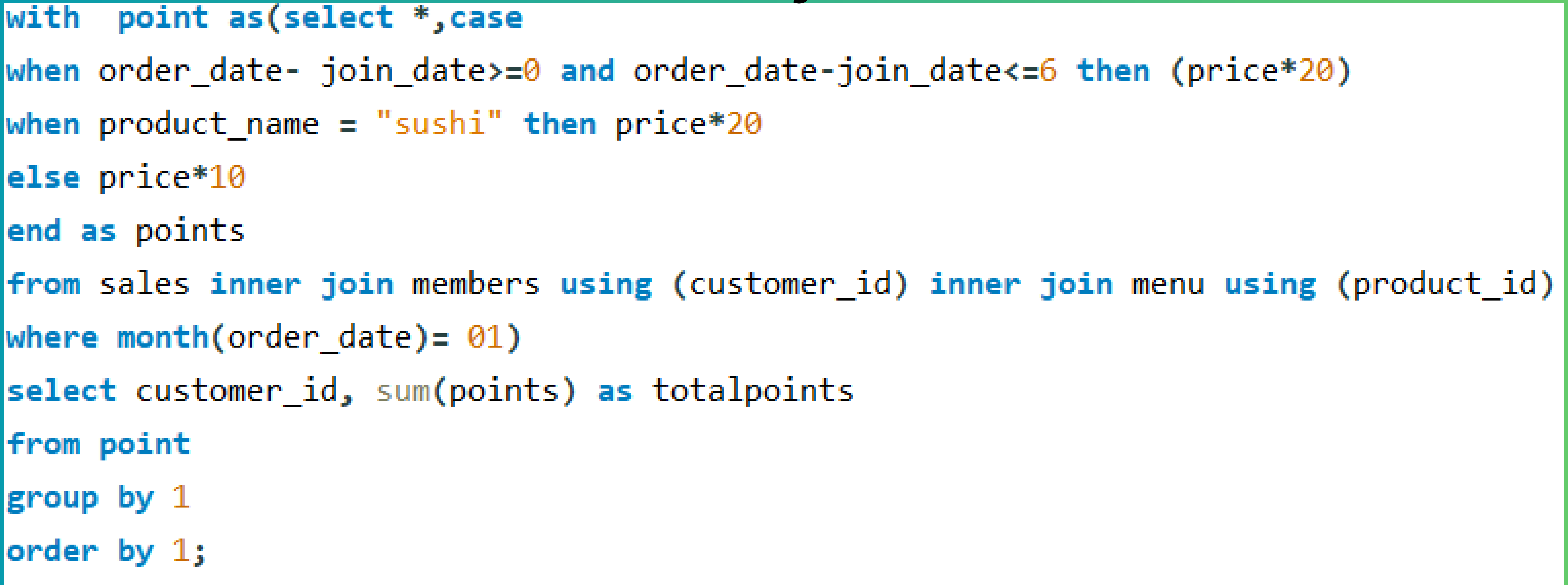


```
  with point as(select customer_id, product_name,price,
                if(product_name="sushi", price*20, price*10) as total_points
from sales s inner join menu m using(product_id))
select customer_id, sum(total_points) as points
from point
group by 1;
```

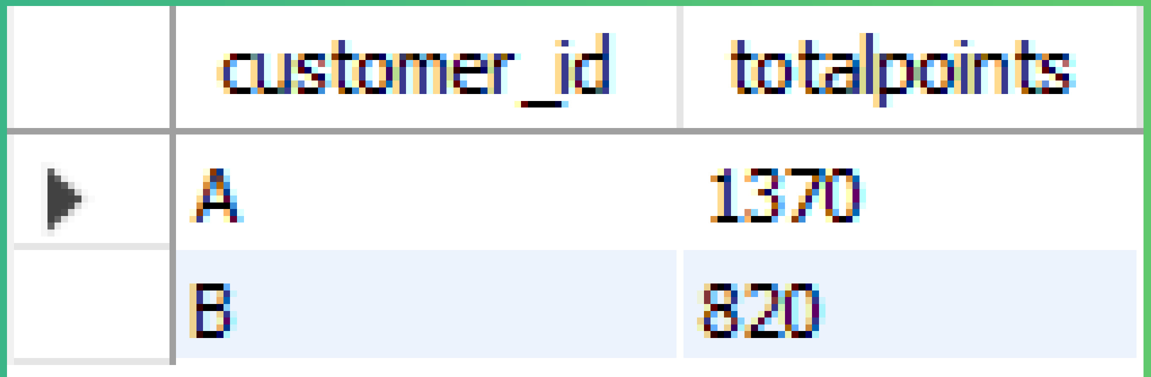


	customer_id	points
▶	A	860
	B	940
	C	360

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  point as(select *,case
when order_date- join_date>=0 and order_date-join_date<=6 then (price*20)
when product_name = "sushi" then price*20
else price*10
end as points
from sales inner join members using (customer_id) inner join menu using (product_id)
where month(order_date)= 01)
select customer_id, sum(points) as totalpoints
from point
group by 1
order by 1;
```



	customer_id	totalpoints
▶	A	1370
	B	820

CONCLUSION

- **Customer A** spent the most ($76), closely followed by **Customer B** ($74).
- **Customer A & B** visited the restaurant equally (6 times each).
- The customers purchased **Sushi** just before becoming members. After joining the membership program, they purchased **Ramen**.
- **Ramen** is the most popular item, both in individual and overall purchases.
- **Customer A** spent $25 for 2 items and **Customer B** spent $40 on 3 items.
- **Both A & B** bought Sushi but had different preferences for Ramen.
- **Customer A** had the highest total points at 1370, while Customer B had 820.

Thank You