Case Study #1 - Danny's Diner

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 favorite 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 favorite. Having this deeper connection with his customers will help him deliver a better and more personalized experience for his loyal customers.

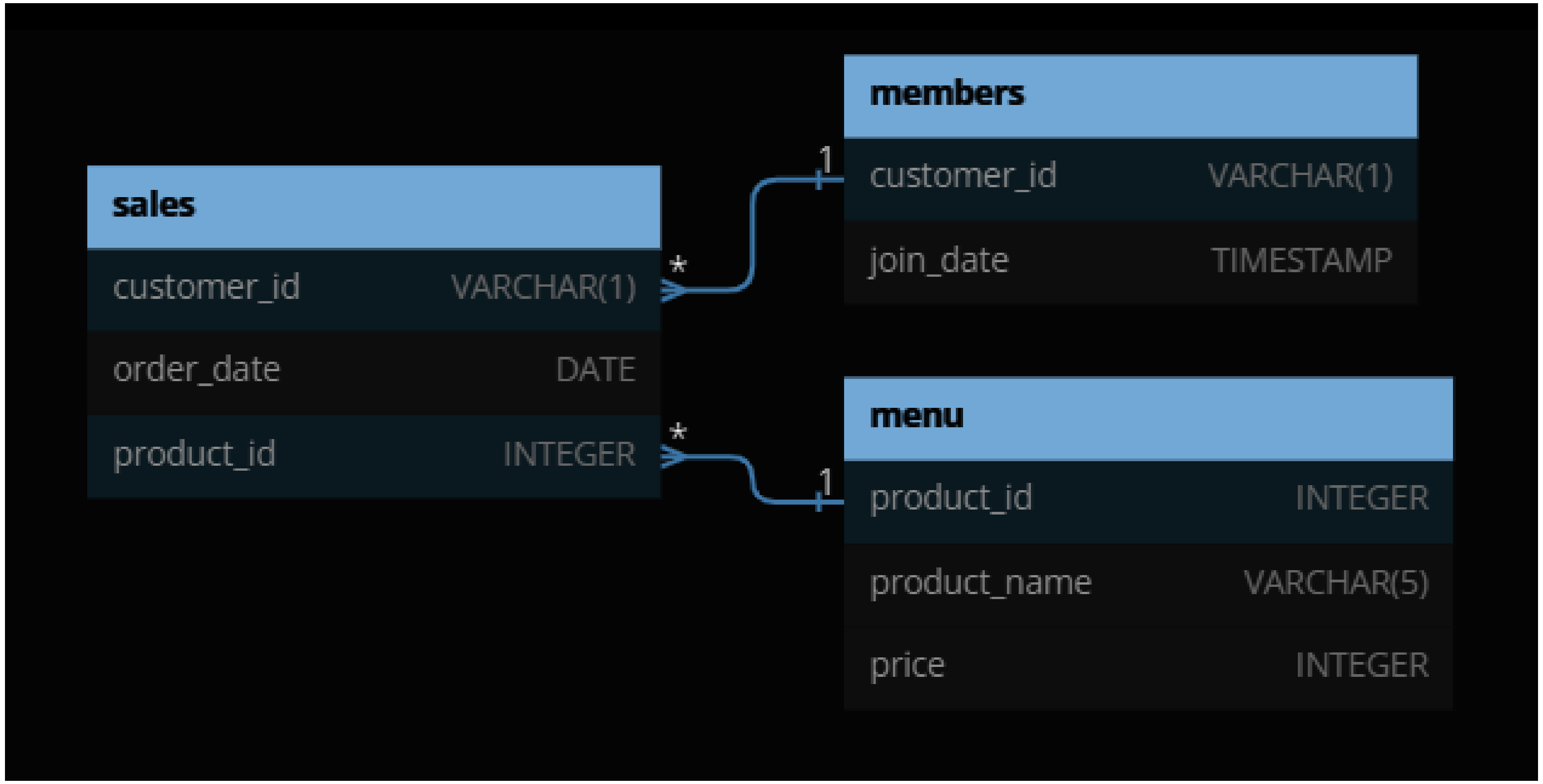
He plans on using these insights to help him decide whether he should expand the existing customer loyalty program - additionally he needs help to generate some basic datasets so his team can easily inspect the data without needing to use SQL.

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!

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

- sales
- menu
- members

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

Tables

Table 1: sales

The `sales` table captures all `customer_id` level purchases with an corresponding `order_date` and `product_id` information for when and what menu items were ordered.

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
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

Table 2: menu

The `menu` table maps the `product_id` to the actual `product_name` and `price` of each menu item.

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

Table 3: members

The final `members` table captures the `join_date` when a `customer_id` joined the beta version of the Danny's Diner loyalty program.

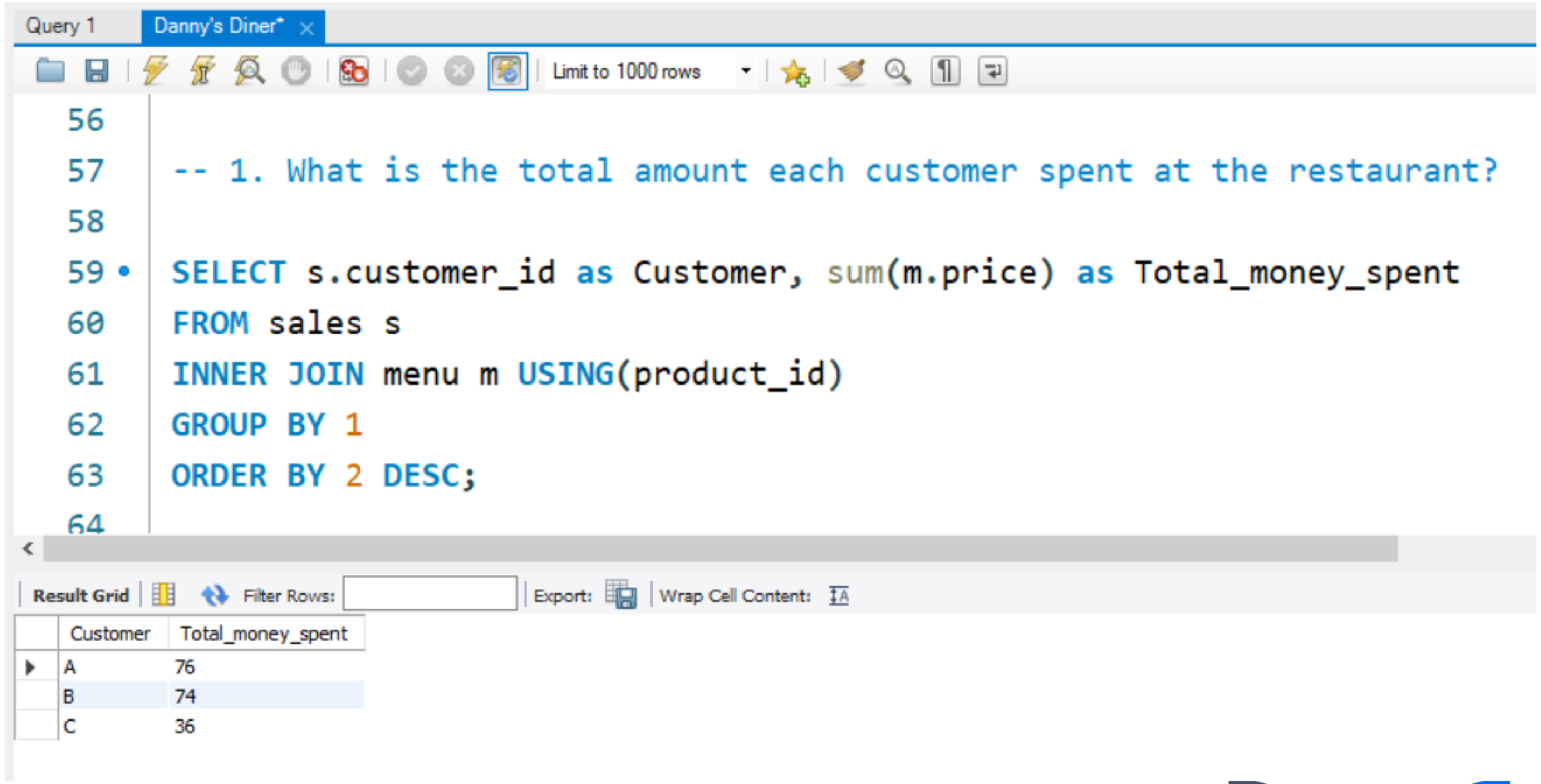
customer_id	join_date
A	2021-01-07
B	2021-01-09

Case Study Questions

Each of the following case study questions can be answered using a single SQL statement:

1. What is the total amount each customer spent at the restaurant?
2. How many days has each customer visited the restaurant?
3. What was the first item from the menu purchased by each customer?
4. What is the most purchased item on the menu and how many times was it purchased by all customers?
5. Which item was the most popular for each customer?
6. Which item was purchased first by the customer after they became a member?
7. Which item was purchased just before the customer became a member?
8. What is the total items and amount spent for each member before they became a member?
9. If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?
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?

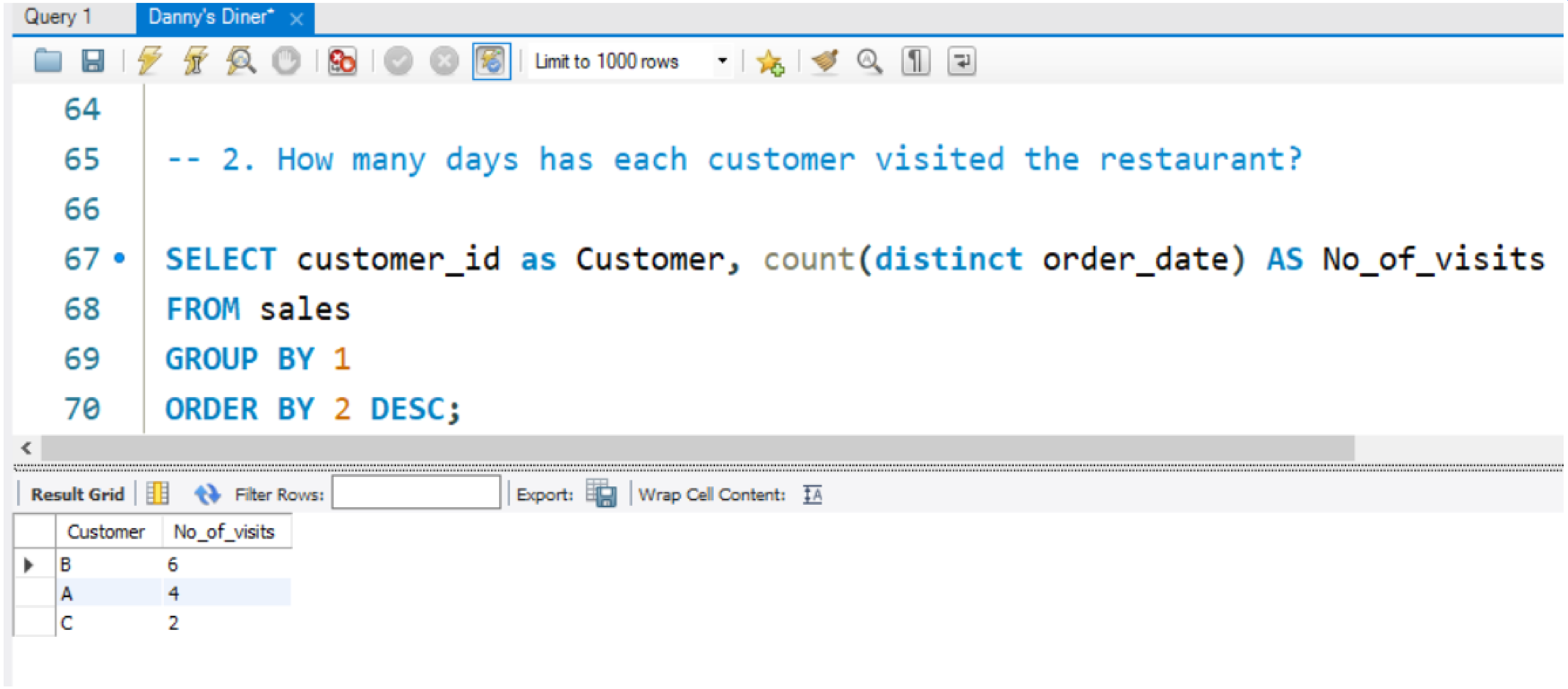
Q1. What is the total amount each customer spent at the restaurant?



```sql

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

SELECT s.customer_id as Customer, sum(m.price) as Total_money_spent
FROM sales s
INNER JOIN menu m USING(product_id)
GROUP BY 1
ORDER BY 2 DESC;

```

Customer	Total_money_spent
A	76
B	74
C	36

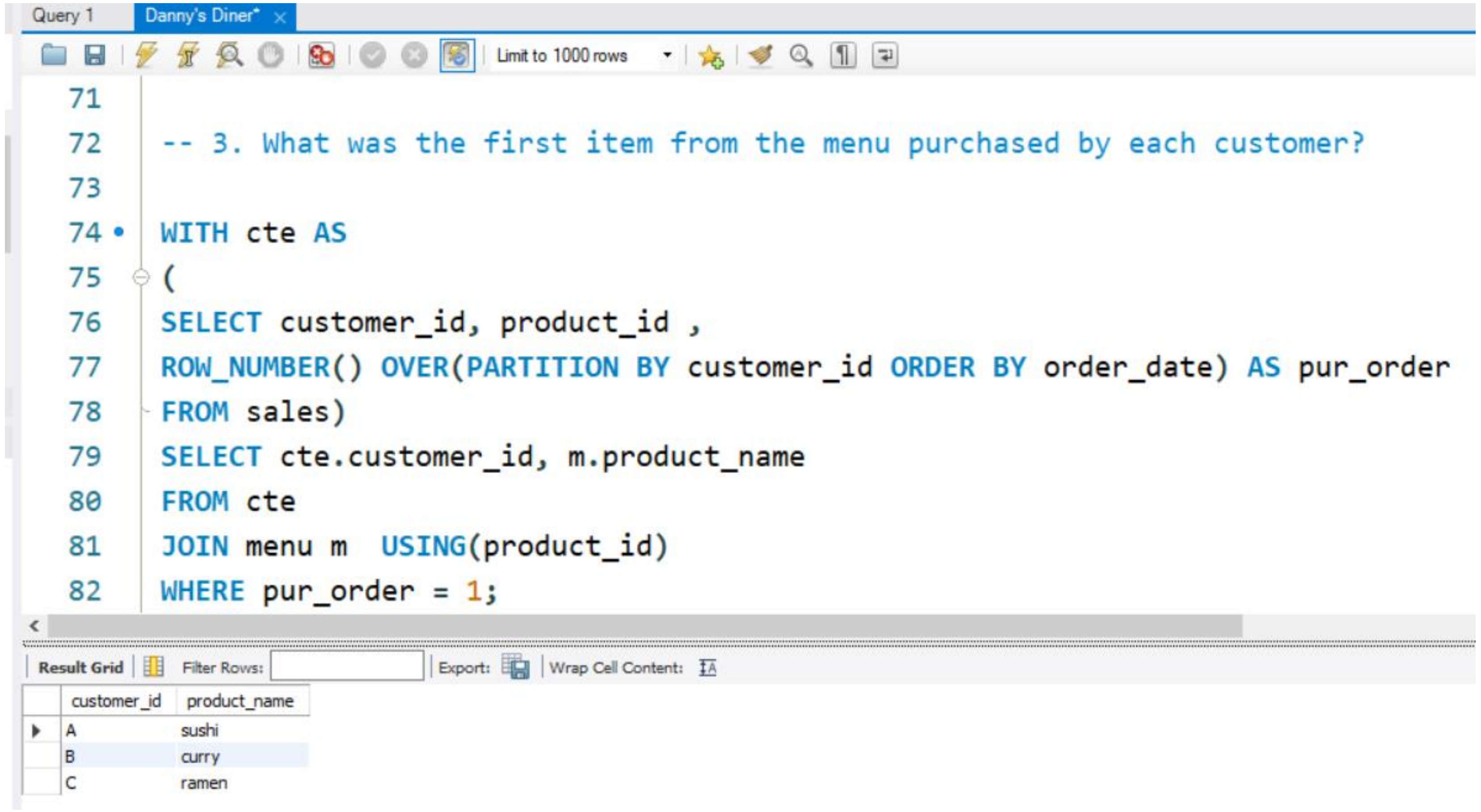
Q2. How many days has each customer visited the restaurant?



```sql

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

SELECT customer_id as Customer, count(distinct order_date) AS No_of_visits
FROM sales
GROUP BY 1
ORDER BY 2 DESC;
```

Customer	No_of_visits
B	6
A	4
C	2

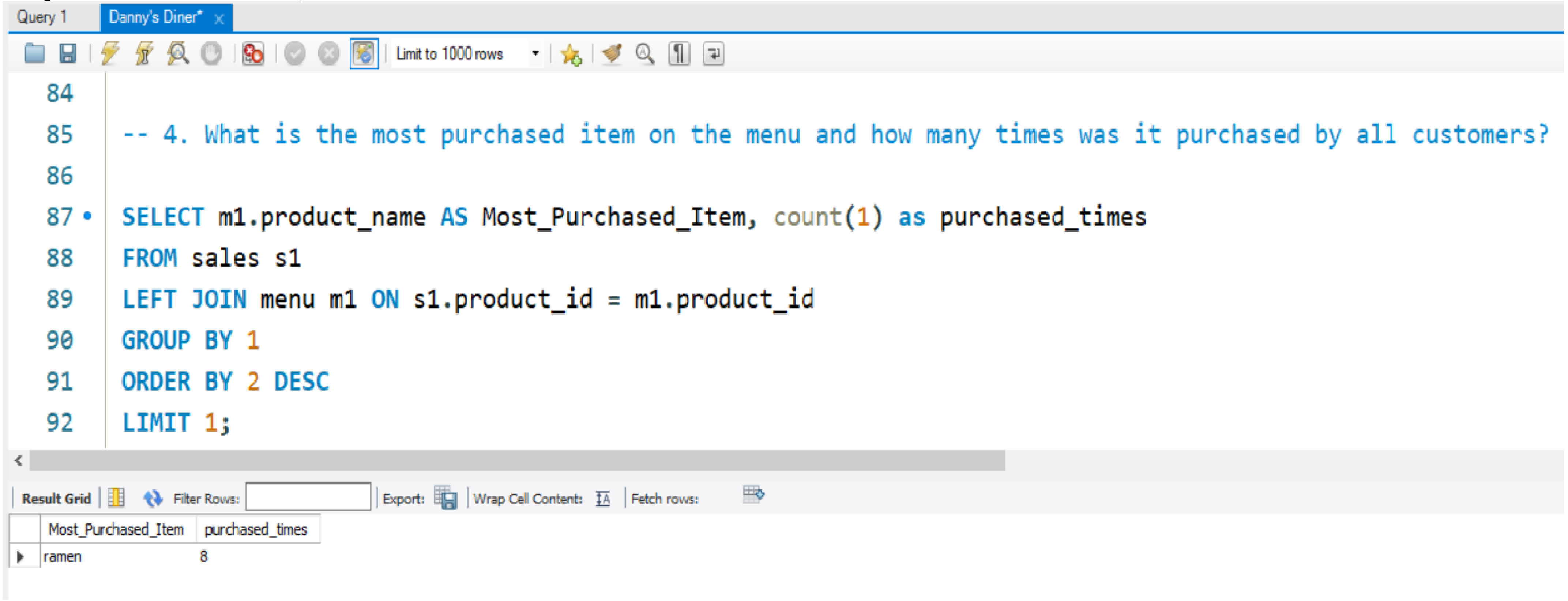
Q3. What was the first item from the menu purchased by each customer?



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

WITH cte AS
(
SELECT customer_id, product_id ,
ROW_NUMBER() OVER(PARTITION BY customer_id ORDER BY order_date) AS pur_order
FROM sales)
SELECT cte.customer_id, m.product_name
FROM cte
JOIN menu m  USING(product_id)
WHERE pur_order = 1;
```

customer_id	product_name
A	sushi
B	curry
C	ramen

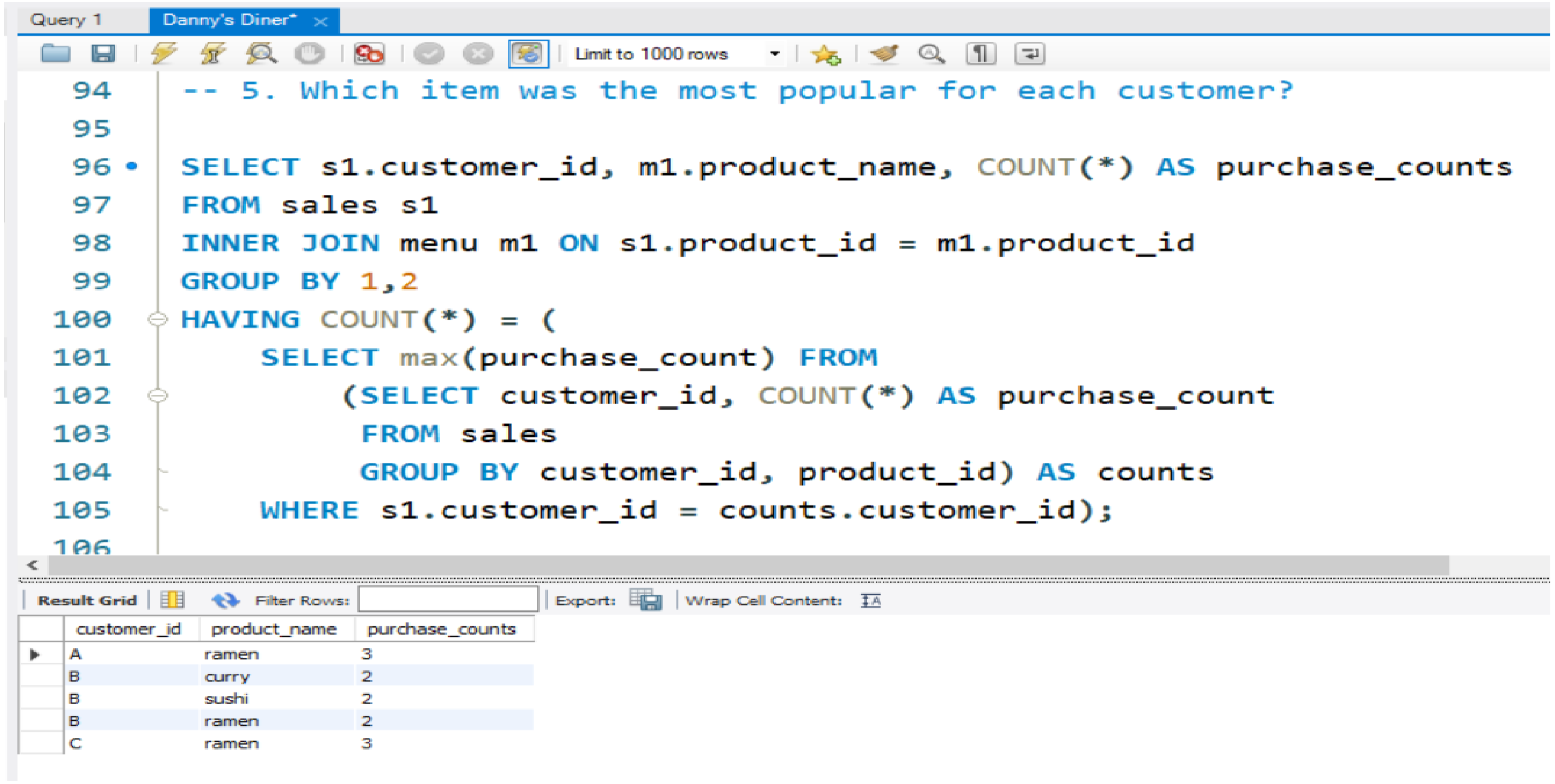
Q4. What is the most purchased item on the menu and how many times was it purchased by all customers?



```

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

SELECT m1.product_name AS Most_Purchased_Item, count(1) as purchased_times
FROM sales s1
LEFT JOIN menu m1 ON s1.product_id = m1.product_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1;
```

Most_Purchased_Item	purchased_times
ramen	8

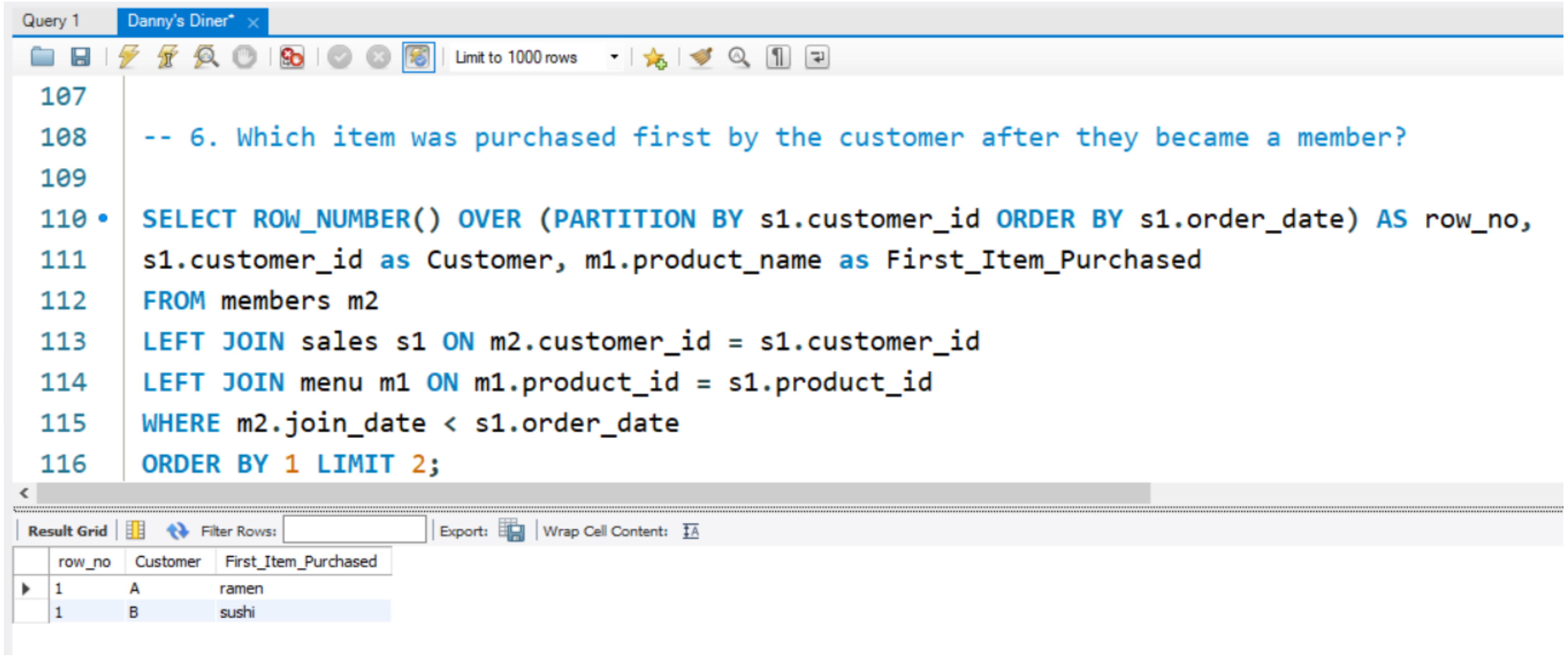
Q5. Which item was the most popular for each customer?



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

SELECT s1.customer_id, m1.product_name, COUNT(*) AS purchase_counts
FROM sales s1
INNER JOIN menu m1 ON s1.product_id = m1.product_id
GROUP BY 1,2
HAVING COUNT(*) = (
    SELECT max(purchase_count) FROM
        (SELECT customer_id, COUNT(*) AS purchase_count
         FROM sales
         GROUP BY customer_id, product_id) AS counts
    WHERE s1.customer_id = counts.customer_id);
```

customer_id	product_name	purchase_counts
A	ramen	3
B	curry	2
B	sushi	2
B	ramen	2
C	ramen	3

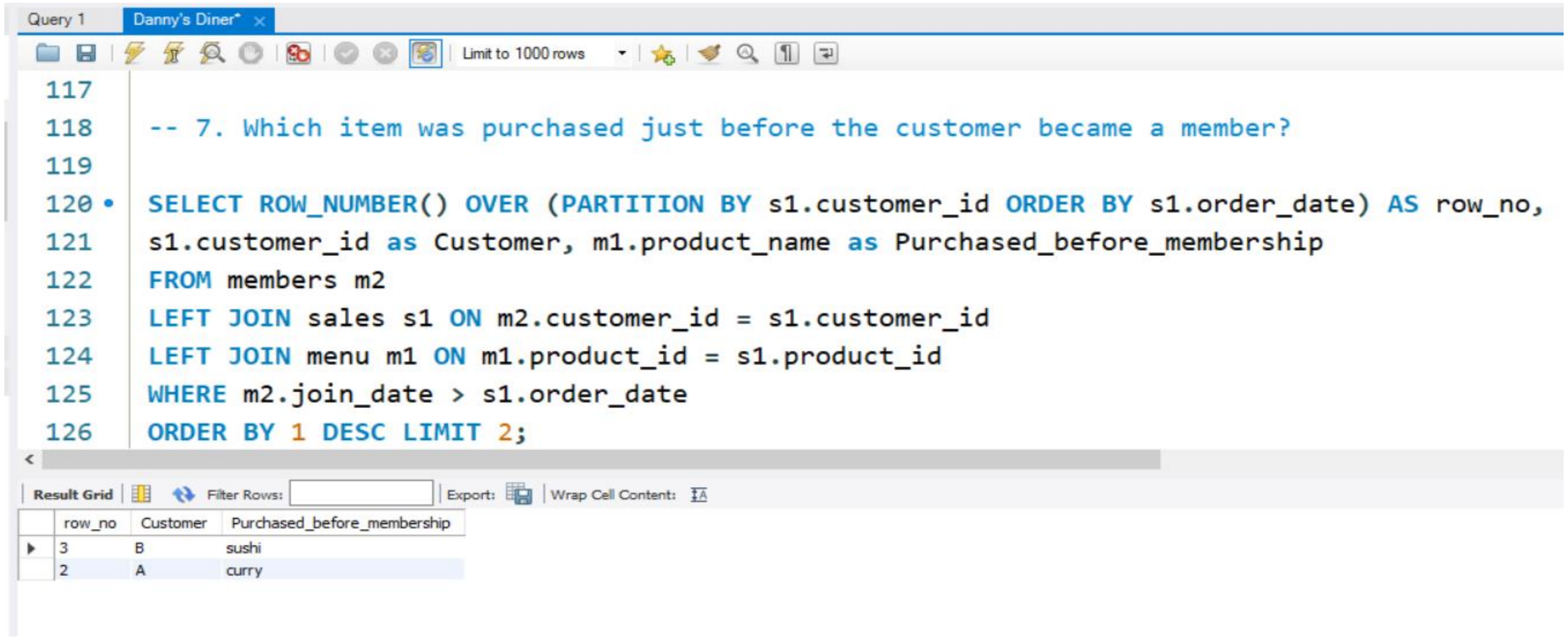
Q6. Which item was purchased first by the customer after they became a member?



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

SELECT ROW_NUMBER() OVER (PARTITION BY s1.customer_id ORDER BY s1.order_date) AS row_no,
s1.customer_id as Customer, m1.product_name as First_Item_Purchased
FROM members m2
LEFT JOIN sales s1 ON m2.customer_id = s1.customer_id
LEFT JOIN menu m1 ON m1.product_id = s1.product_id
WHERE m2.join_date < s1.order_date
ORDER BY 1 LIMIT 2;
```

row_no	Customer	First_Item_Purchased
1	A	ramen
1	B	sushi

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



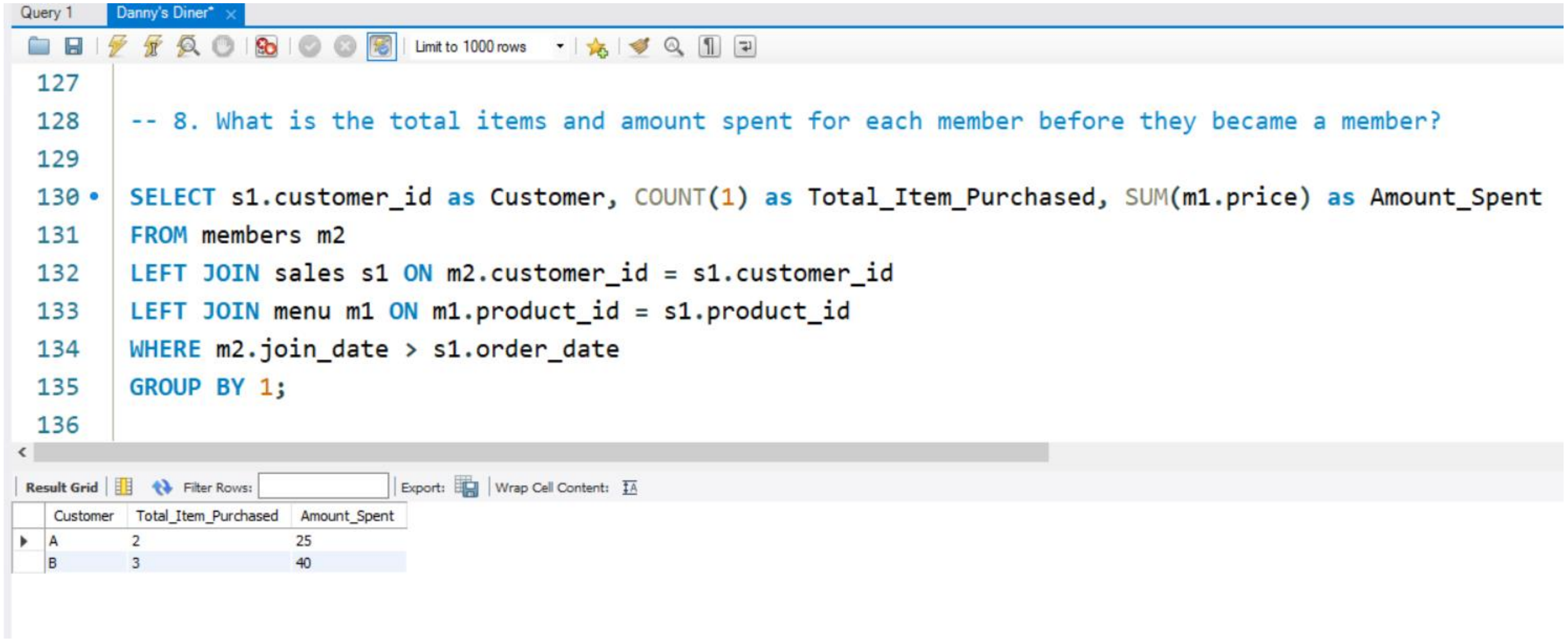
Query 1 | Danny's Diner*

Limit to 1000 rows

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

SELECT ROW_NUMBER() OVER (PARTITION BY s1.customer_id ORDER BY s1.order_date) AS row_no,
s1.customer_id as Customer, m1.product_name as Purchased_before_membership
FROM members m2
LEFT JOIN sales s1 ON m2.customer_id = s1.customer_id
LEFT JOIN menu m1 ON m1.product_id = s1.product_id
WHERE m2.join_date > s1.order_date
ORDER BY 1 DESC LIMIT 2;
```

row_no	Customer	Purchased_before_membership
3	B	sushi
2	A	curry

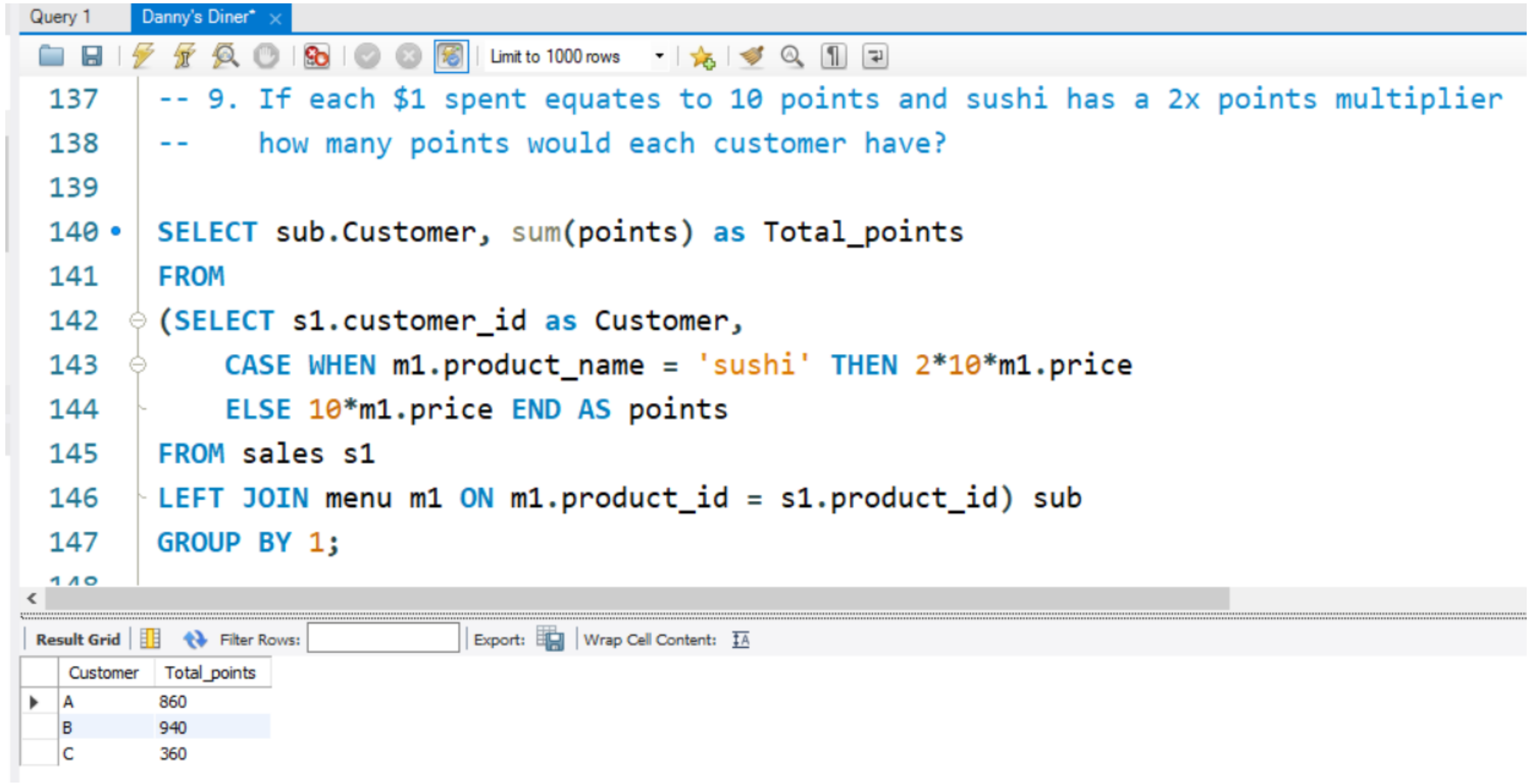
Q8. What is the total items and amount spent for each member before they became a member?



```sql

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

SELECT s1.customer_id as Customer, COUNT(1) as Total_Item_Purchased, SUM(m1.price) as Amount_Spent
FROM members m2
LEFT JOIN sales s1 ON m2.customer_id = s1.customer_id
LEFT JOIN menu m1 ON m1.product_id = s1.product_id
WHERE m2.join_date > s1.order_date
GROUP BY 1;

```

Customer	Total_Item_Purchased	Amount_Spent
A	2	25
B	3	40

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


Query 1 | Danny's Diner*

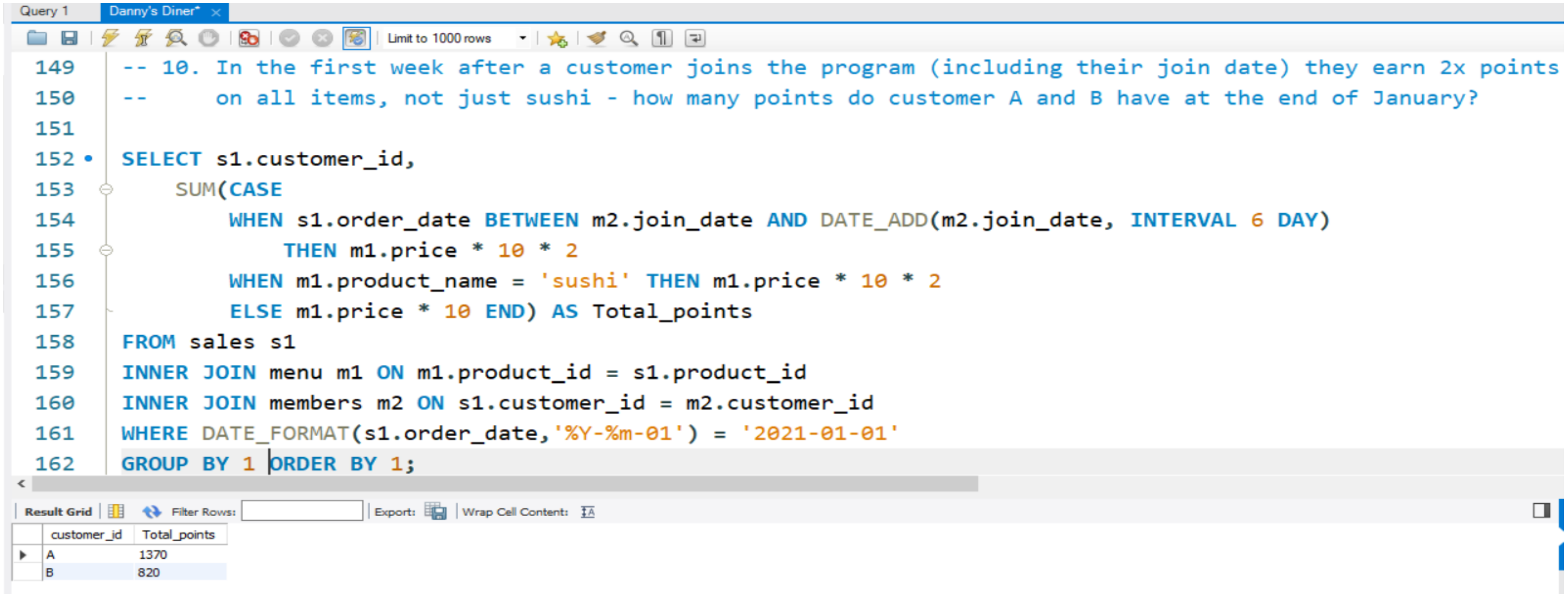
Limit to 1000 rows

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

SELECT sub.Customer, sum(points) as Total_points
FROM
(SELECT s1.customer_id as Customer,
    CASE WHEN m1.product_name = 'sushi' THEN 2*10*m1.price
    ELSE 10*m1.price END AS points
FROM sales s1
LEFT JOIN menu m1 ON m1.product_id = s1.product_id) sub
GROUP BY 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Customer	Total_points
A	860
B	940
C	360

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?



```
-- 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 s1.customer_id,
    SUM(CASE
        WHEN s1.order_date BETWEEN m2.join_date AND DATE_ADD(m2.join_date, INTERVAL 6 DAY)
            THEN m1.price * 10 * 2
        WHEN m1.product_name = 'sushi' THEN m1.price * 10 * 2
        ELSE m1.price * 10 END) AS Total_points
FROM sales s1
INNER JOIN menu m1 ON m1.product_id = s1.product_id
INNER JOIN members m2 ON s1.customer_id = m2.customer_id
WHERE DATE_FORMAT(s1.order_date,'%Y-%m-01') = '2021-01-01'
GROUP BY 1 ORDER BY 1;
```

customer_id	Total_points
A	1370
B	820

Key Insights

- ❖ Customer A spent maximum amount i.e. $76.
- ❖ The most purchased item is Ramen.
- ❖ Customer B is most frequent customer.
- ❖ Not all customer who came to the restaurant becomes member of the restaurant so the conversion rate is 66.66%.

Thank you

Shivam Panwar