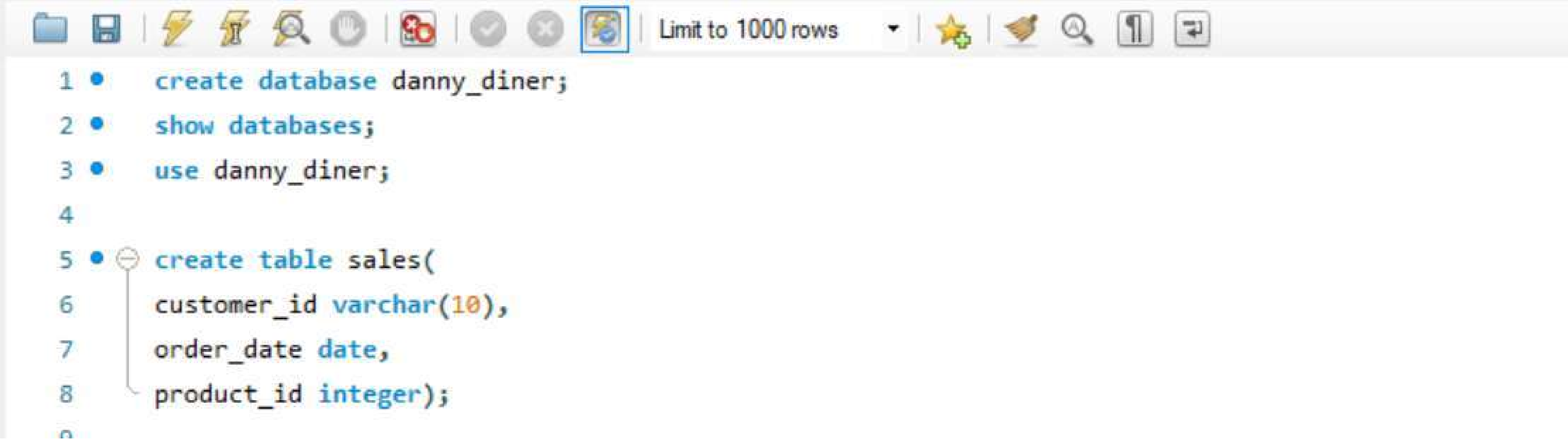
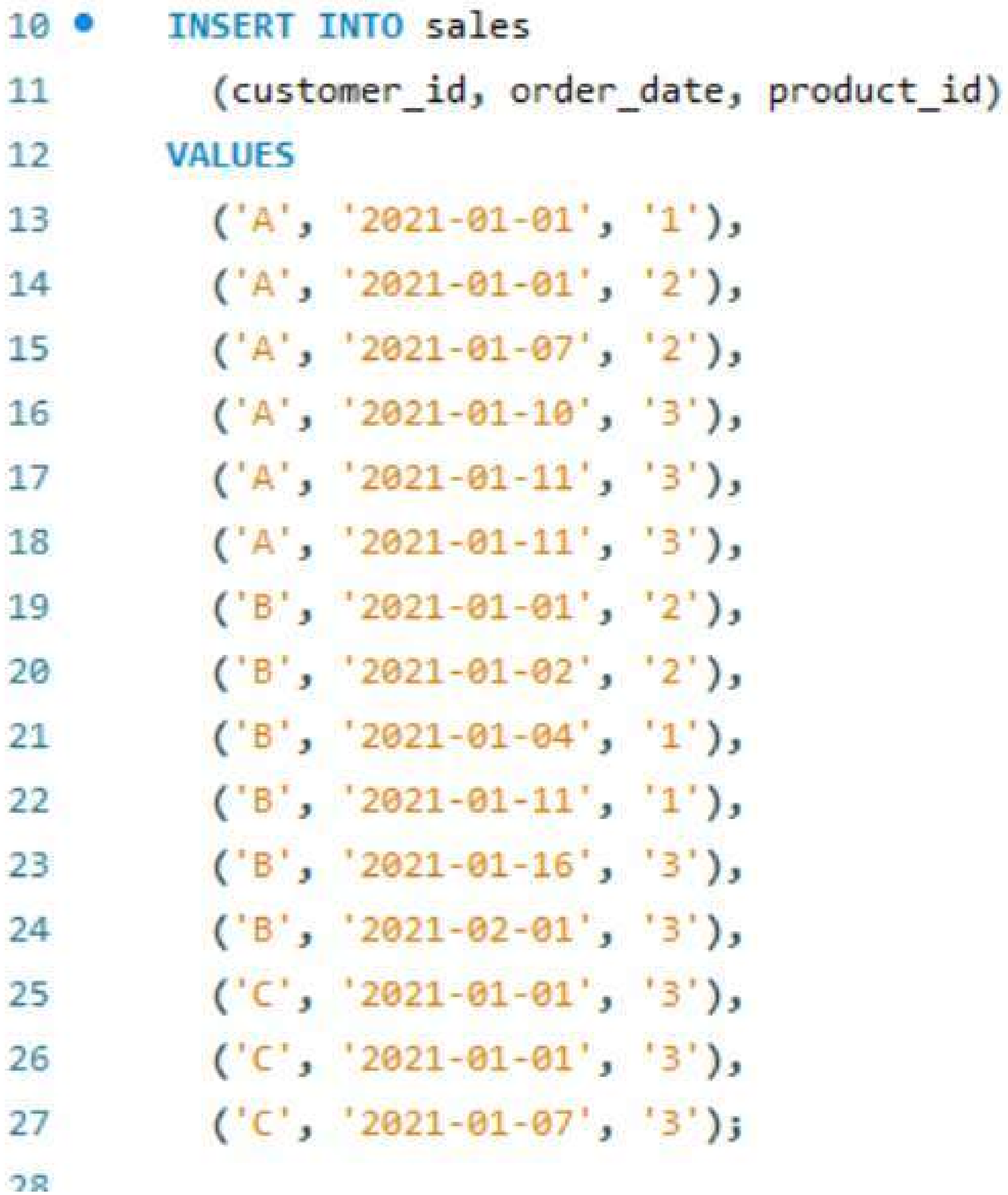
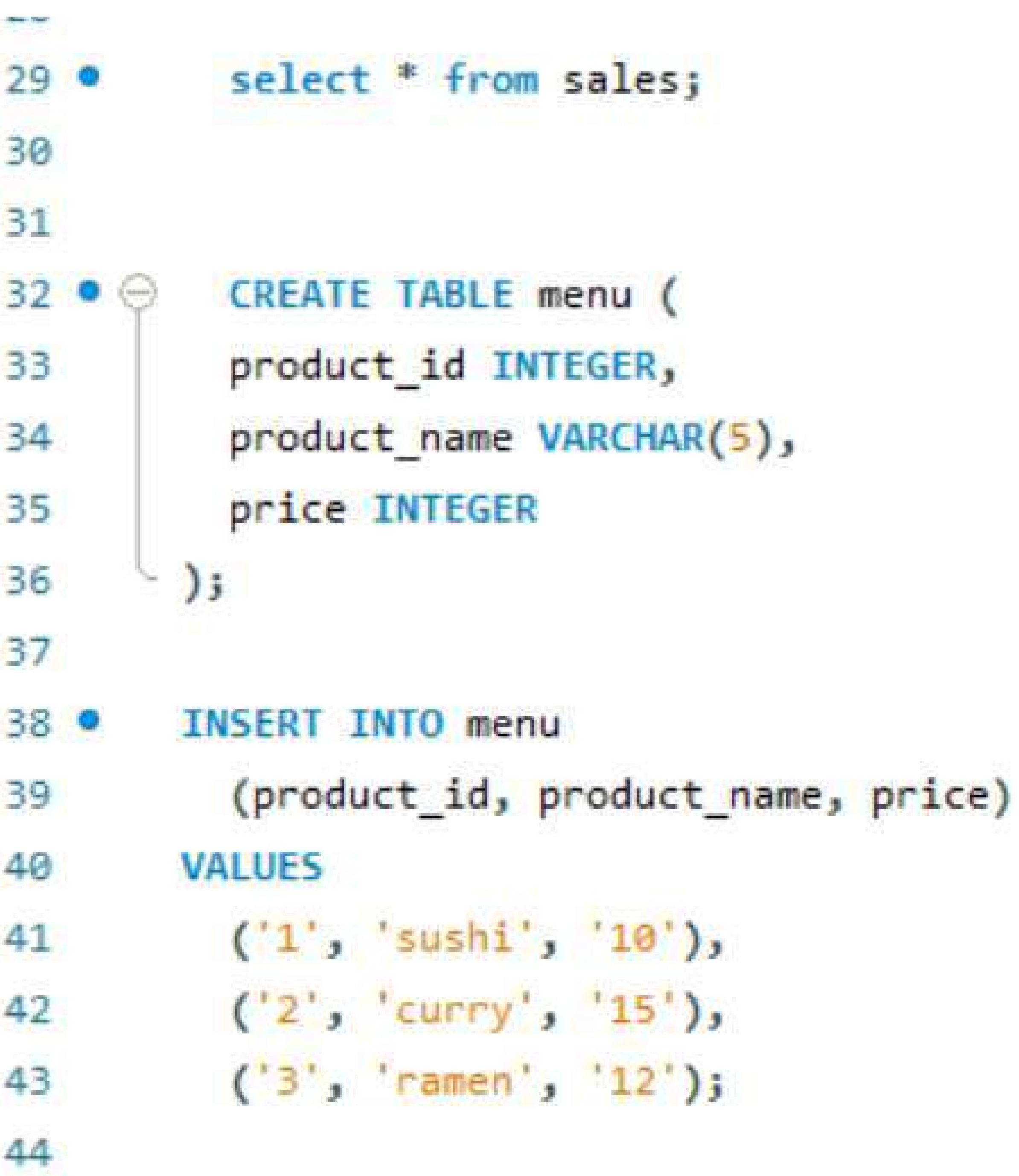


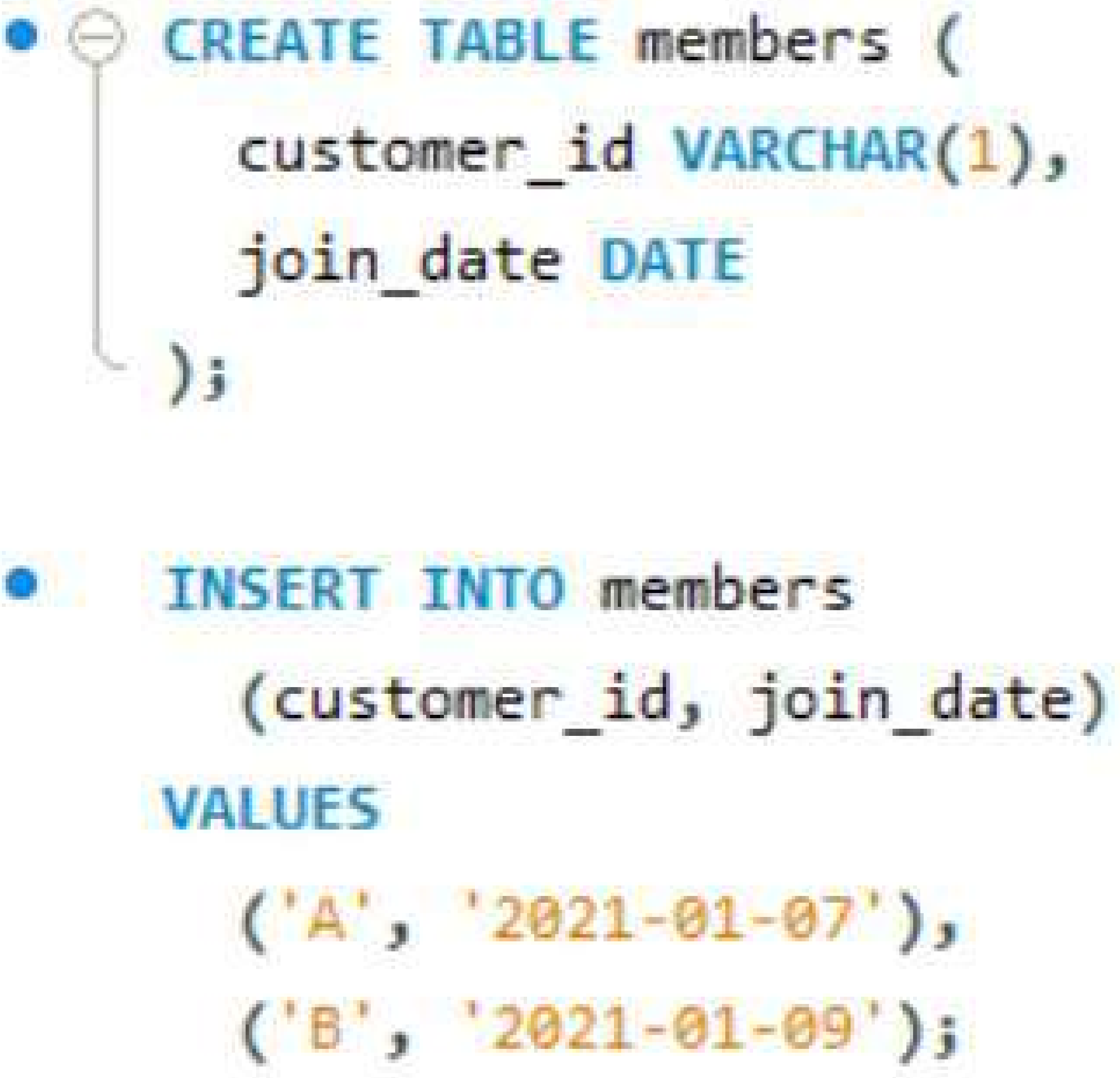
```
create database danny_diner;
show databases;
use danny_diner;

create table sales(
customer_id varchar(10),
order_date date,
product_id integer);
```



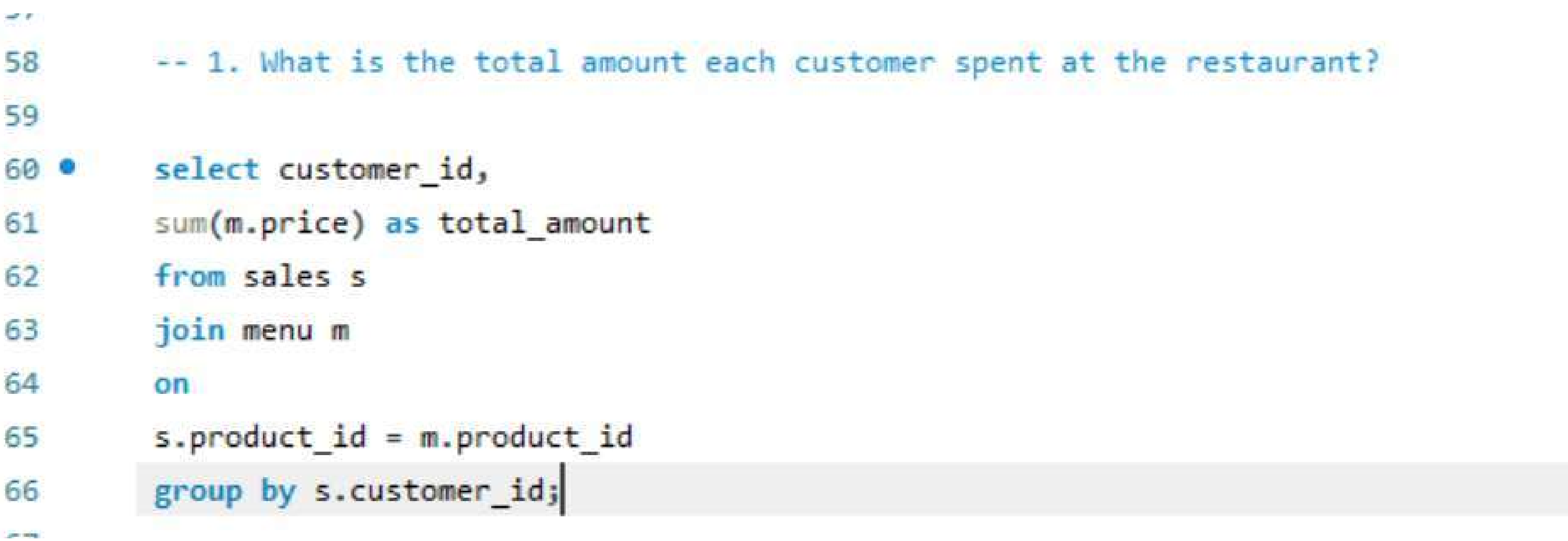
```sql
INSERT INTO sales
  (customer_id, order_date, product_id)
VALUES
  ('A', '2021-01-01', '1'),
  ('A', '2021-01-01', '2'),
  ('A', '2021-01-07', '2'),
  ('A', '2021-01-10', '3'),
  ('A', '2021-01-11', '3'),
  ('A', '2021-01-11', '3'),
  ('B', '2021-01-01', '2'),
  ('B', '2021-01-02', '2'),
  ('B', '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');
```



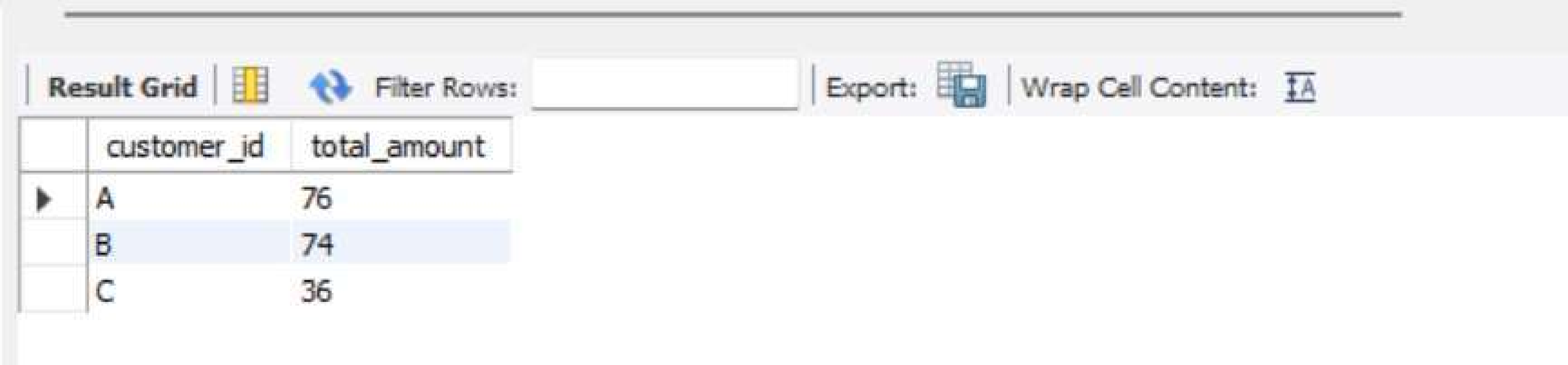
```
  select * from sales;


  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');

```



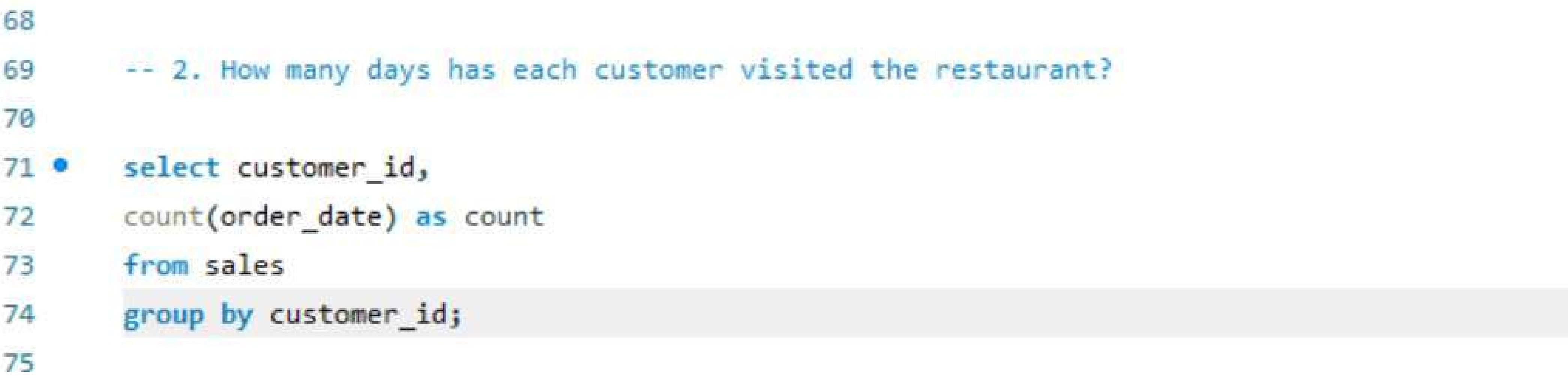
```sql
CREATE TABLE members (
  customer_id VARCHAR(1),
  join_date DATE
);

INSERT INTO members
  (customer_id, join_date)
VALUES
  ('A', '2021-01-07'),
  ('B', '2021-01-09');
```



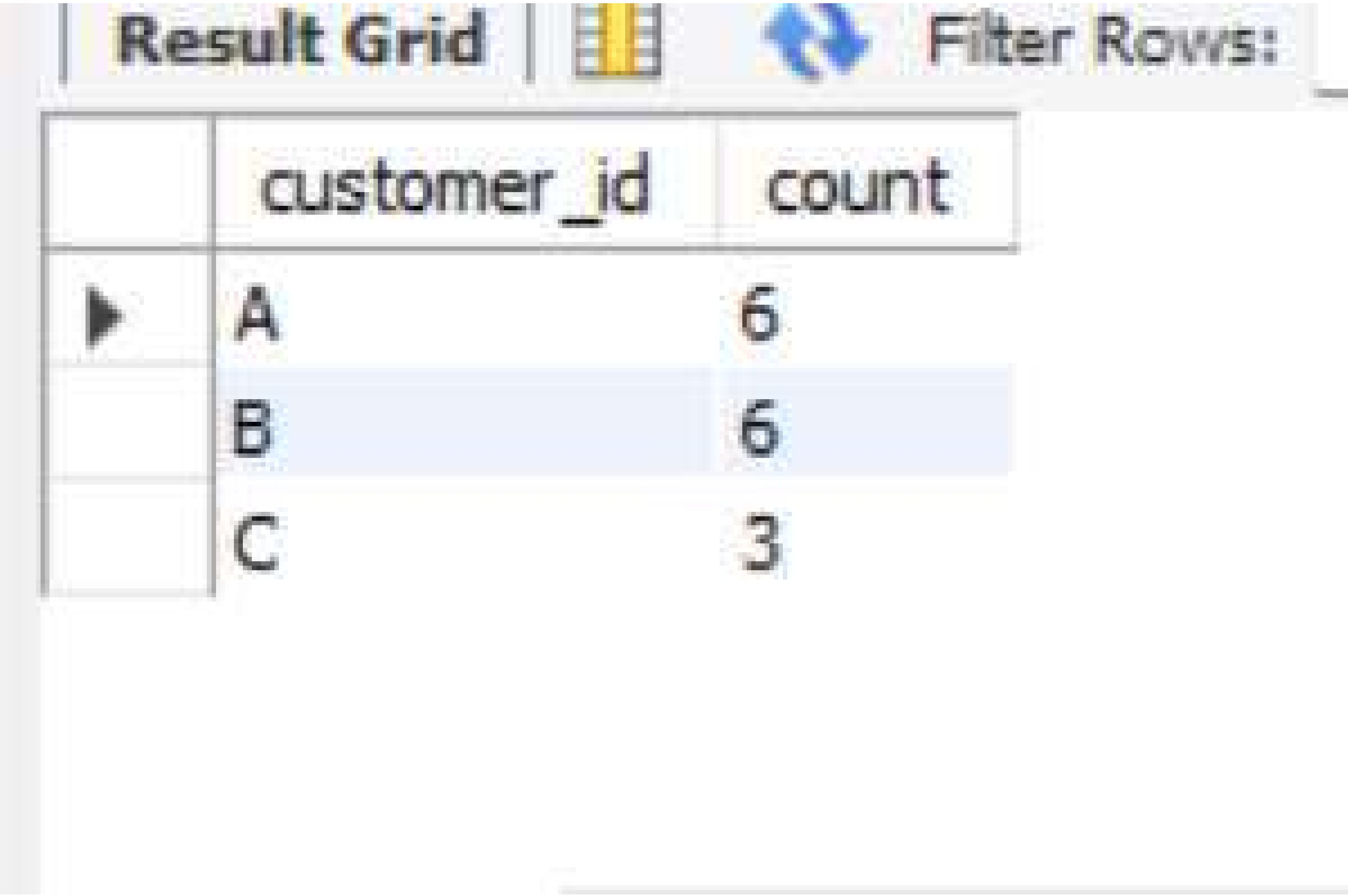
```sql
-- 1. What is the total amount each customer spent at the restaurant?

select customer_id,
sum(m.price) as total_amount
from sales s
join menu m
on
s.product_id = m.product_id
group by s.customer_id;
```



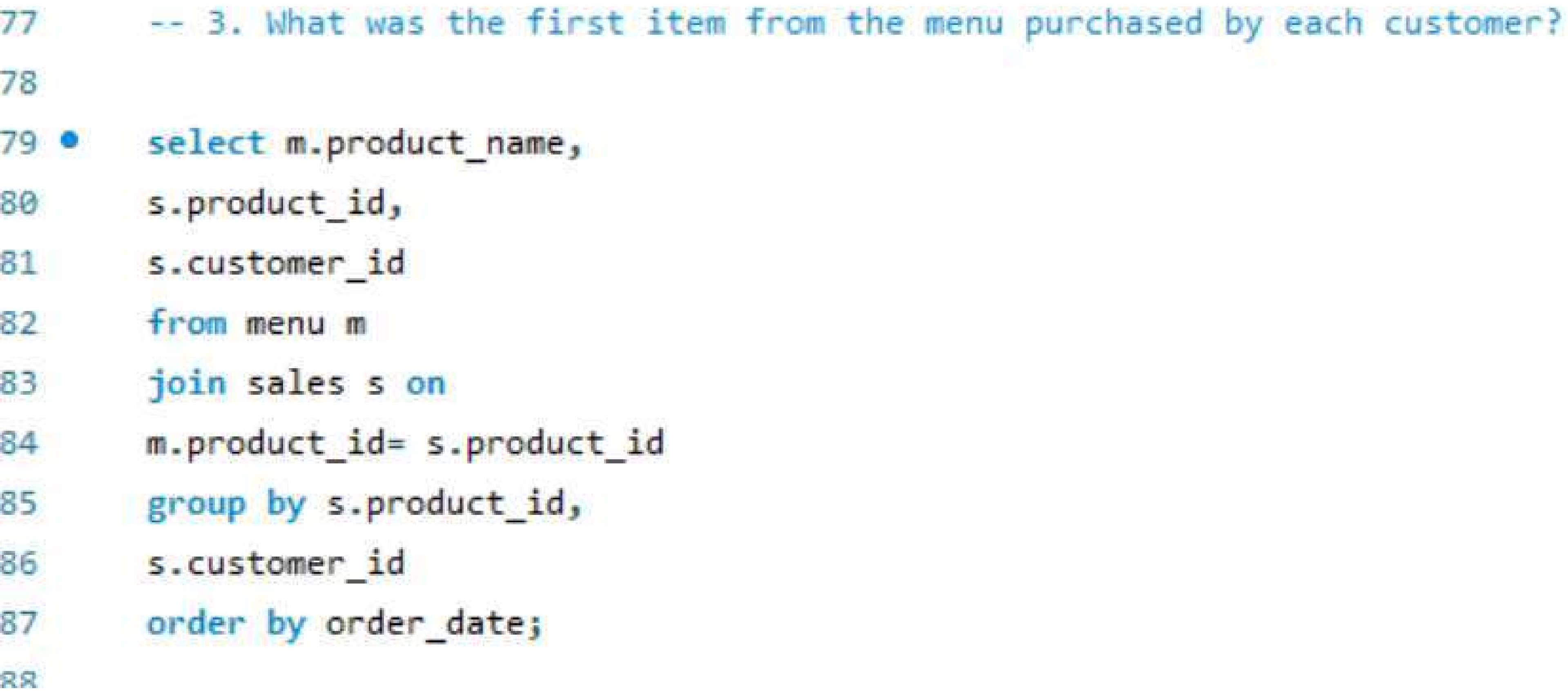
customer_id	total_amount
A	76
B	74
C	36



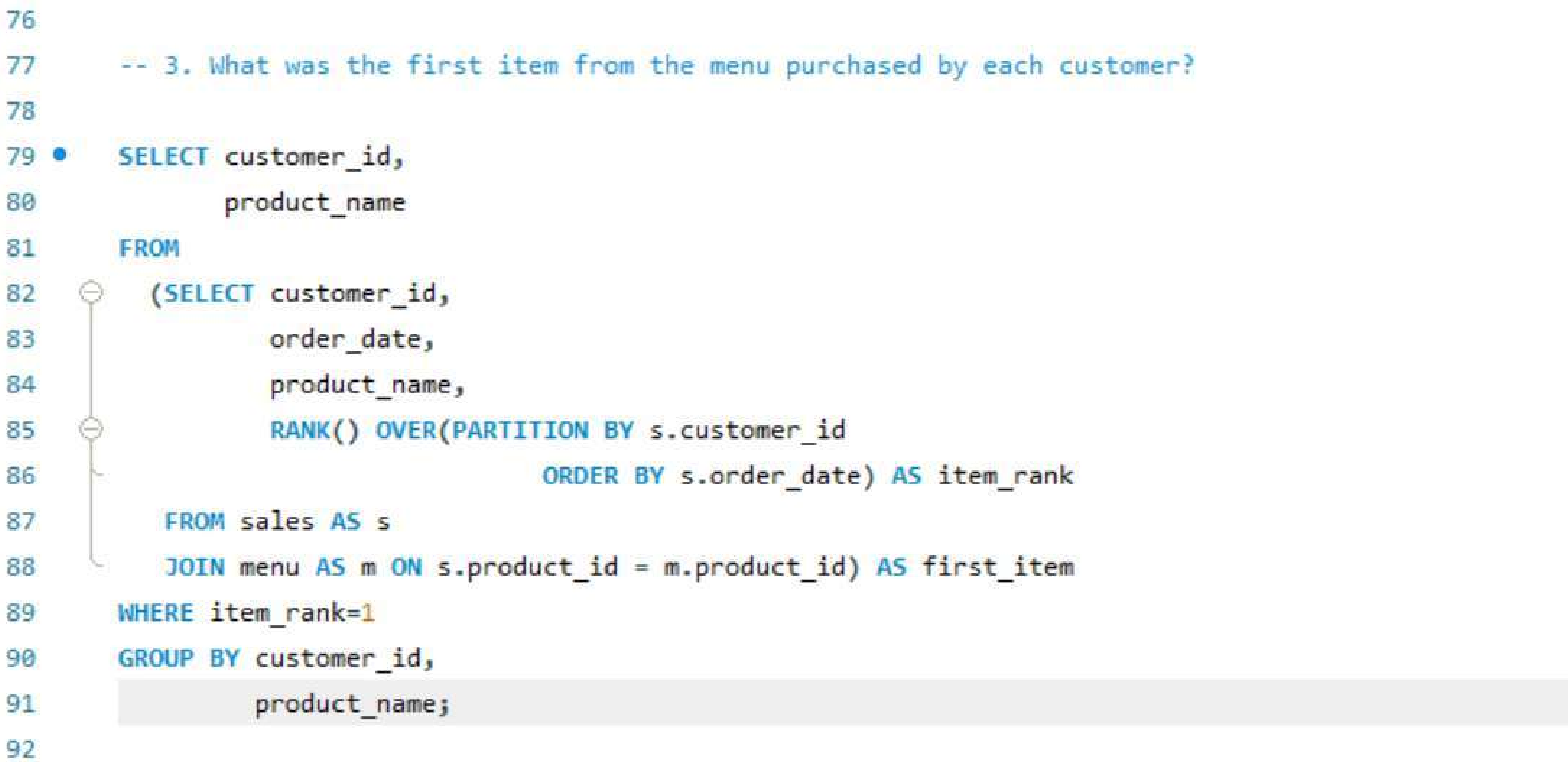
```
-- 2. How many days has each customer visited the restaurant?

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



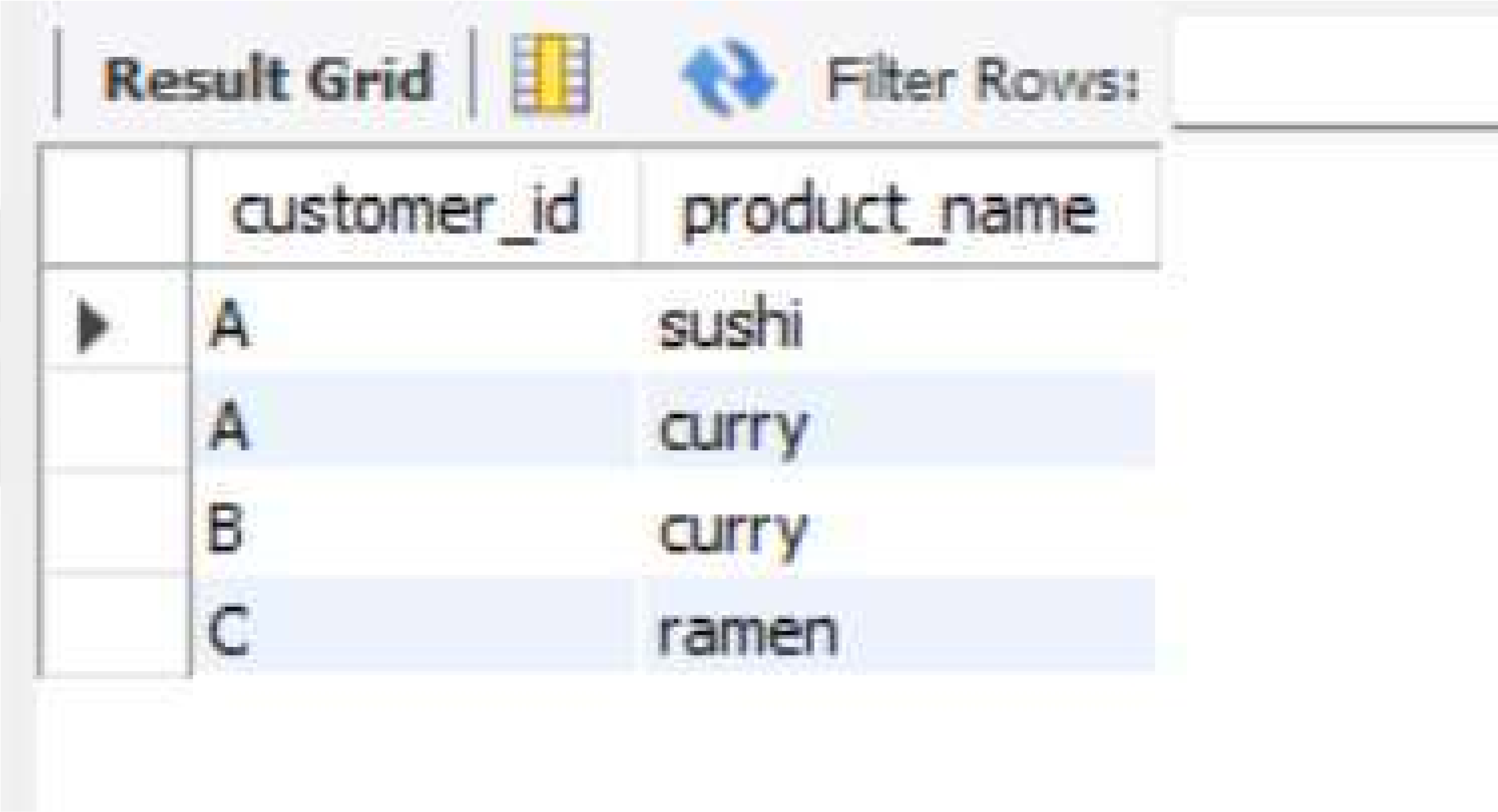
customer_id	count
A	6
B	6
C	3



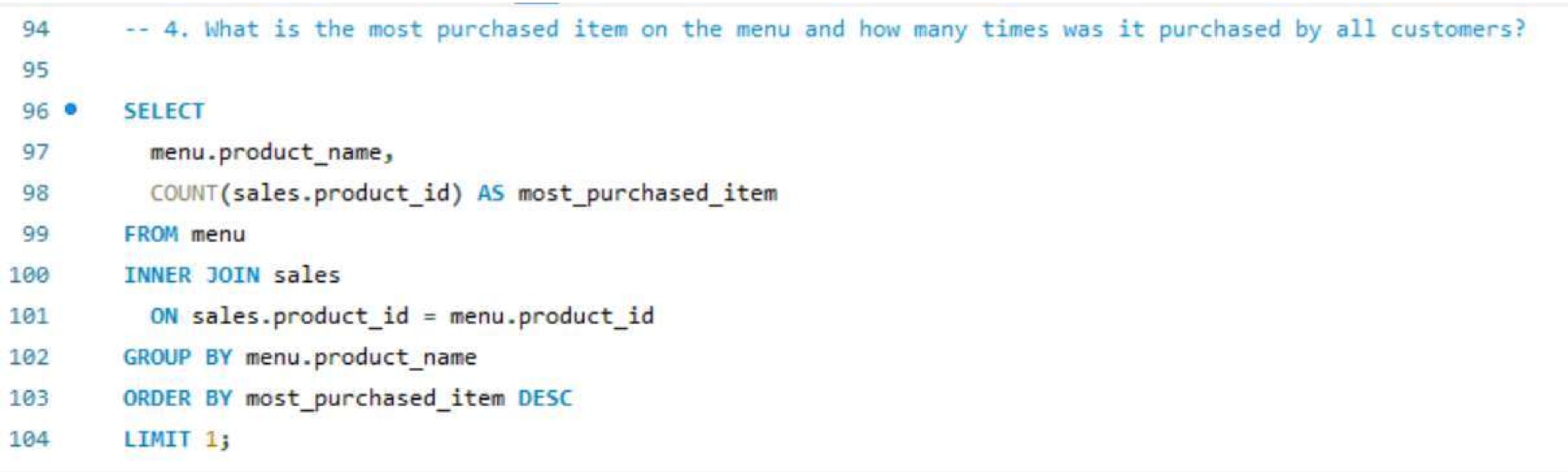
```
-- 3. What was the first item from the menu purchased by each customer?

select m.product_name,
s.product_id,
s.customer_id
from menu m
join sales s on
m.product_id= s.product_id
group by s.product_id,
s.customer_id
order by order_date;
```



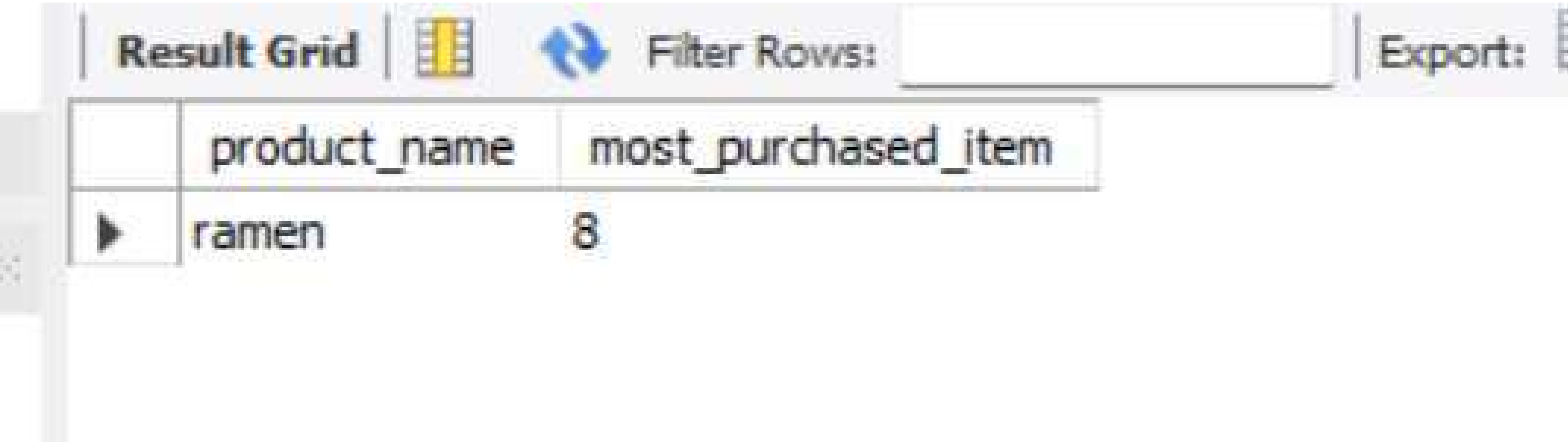
```sql

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

SELECT customer_id,
       product_name
FROM
  (SELECT customer_id,
          order_date,
          product_name,
          RANK() OVER(PARTITION BY s.customer_id
                            ORDER BY s.order_date) AS item_rank
   FROM sales AS s
   JOIN menu AS m ON s.product_id = m.product_id) AS first_item
WHERE item_rank=1
GROUP BY customer_id,
         product_name;

```



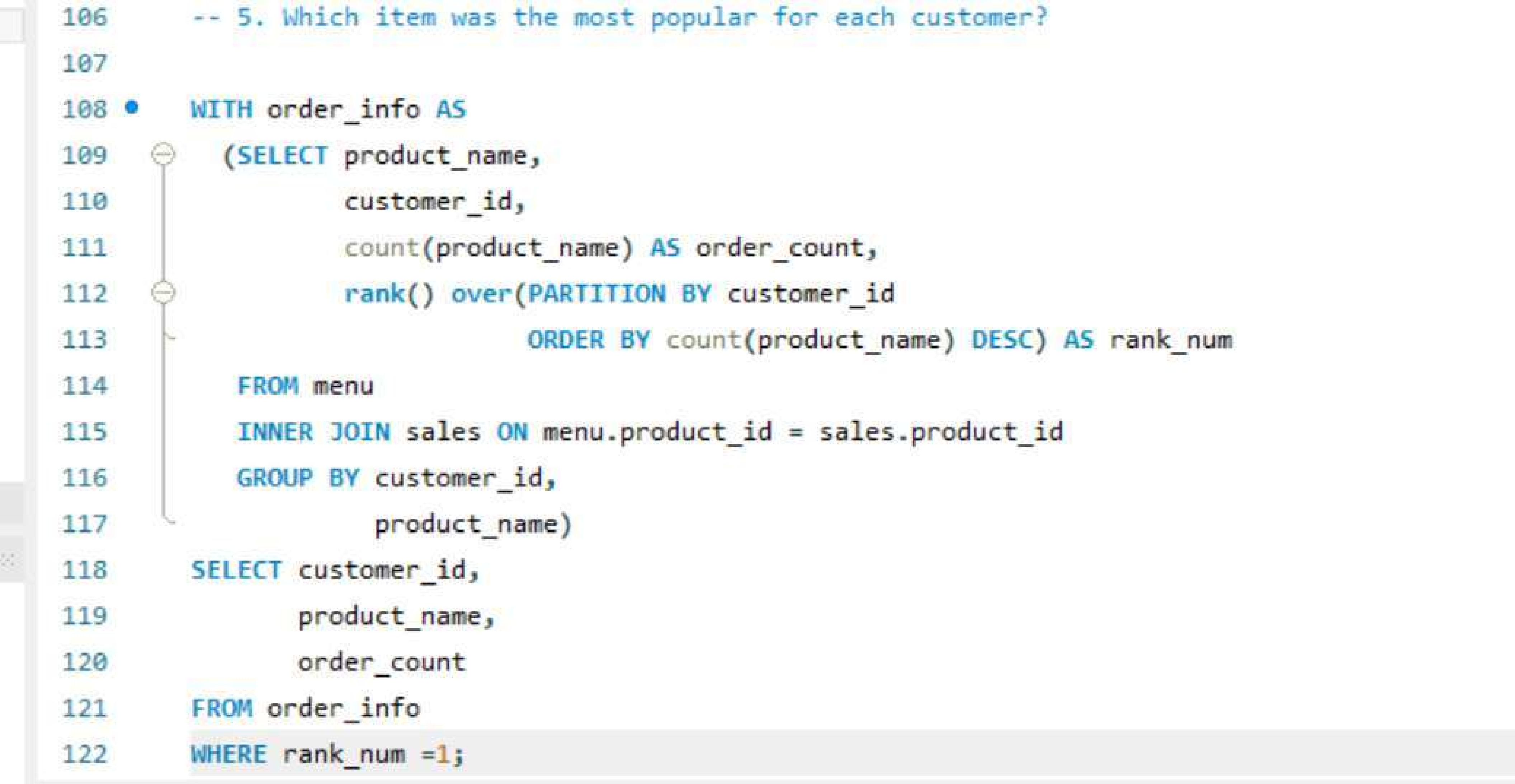
customer_id	product_name
A	sushi
A	curry
B	curry
C	ramen



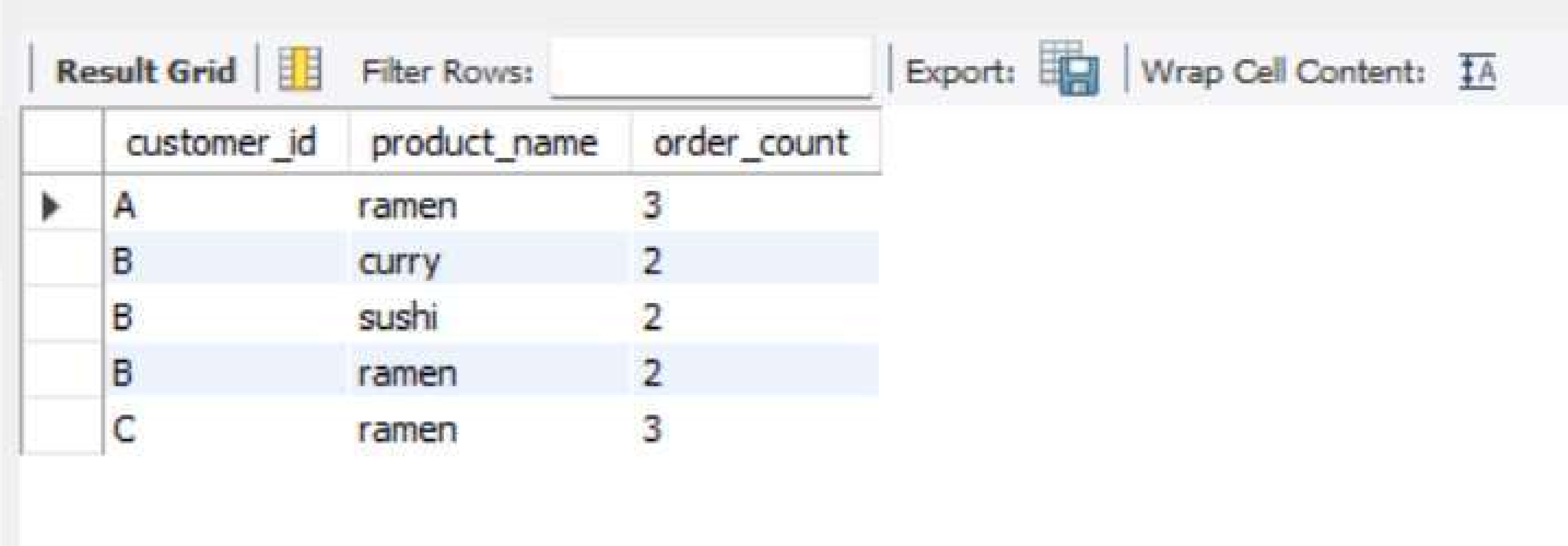
```sql
-- 4. What is the most purchased item on the menu and how many times was it purchased by all customers?

SELECT
  menu.product_name,
  COUNT(sales.product_id) AS most_purchased_item
FROM menu
INNER JOIN sales
  ON sales.product_id = menu.product_id
GROUP BY menu.product_name
ORDER BY most_purchased_item DESC
LIMIT 1;
```



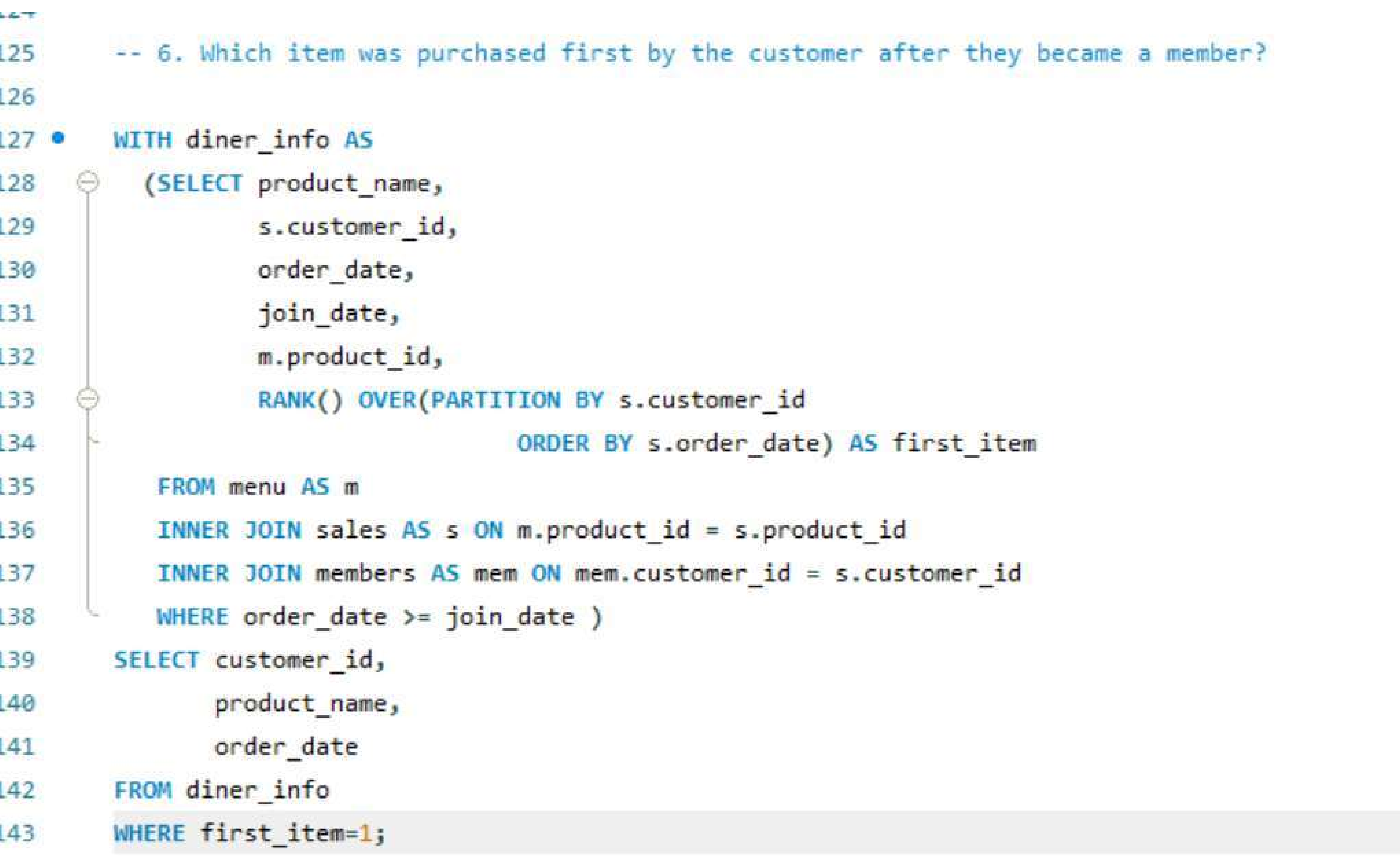
product_name	most_purchased_item
ramen	8



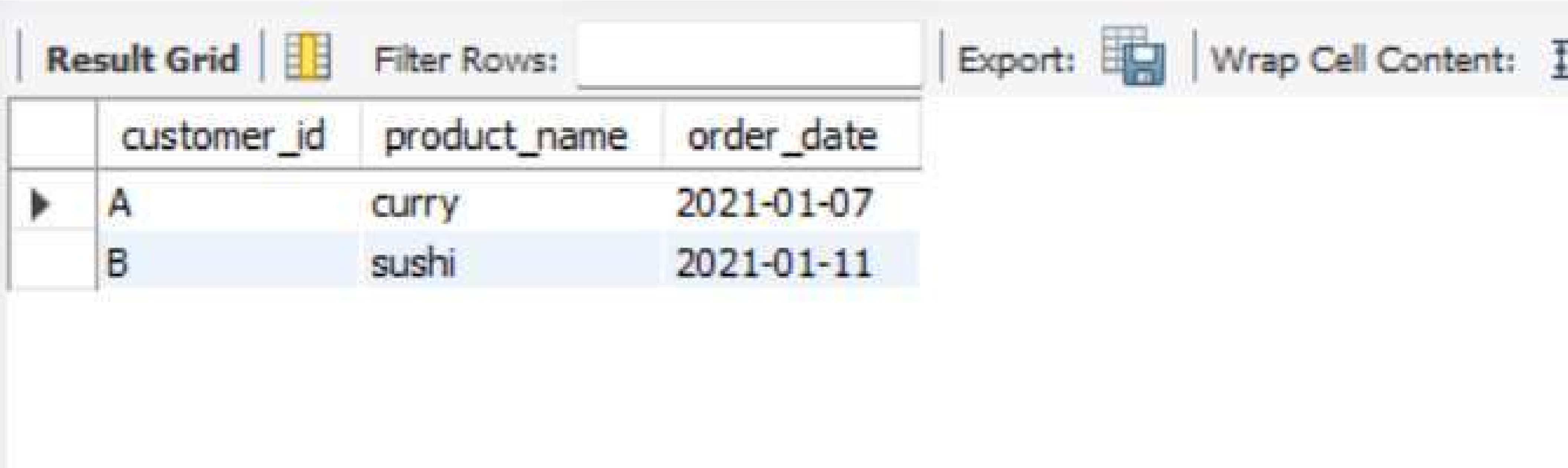
```
-- 5. Which item was the most popular for each customer?

WITH order_info AS
  (SELECT product_name,
          customer_id,
          count(product_name) AS order_count,
          rank() over(PARTITION BY customer_id
                      ORDER BY count(product_name) DESC) AS rank_num
   FROM menu
   INNER JOIN sales ON menu.product_id = sales.product_id
   GROUP BY customer_id,
            product_name)
SELECT customer_id,
       product_name,
       order_count
FROM order_info
WHERE rank_num =1;
```



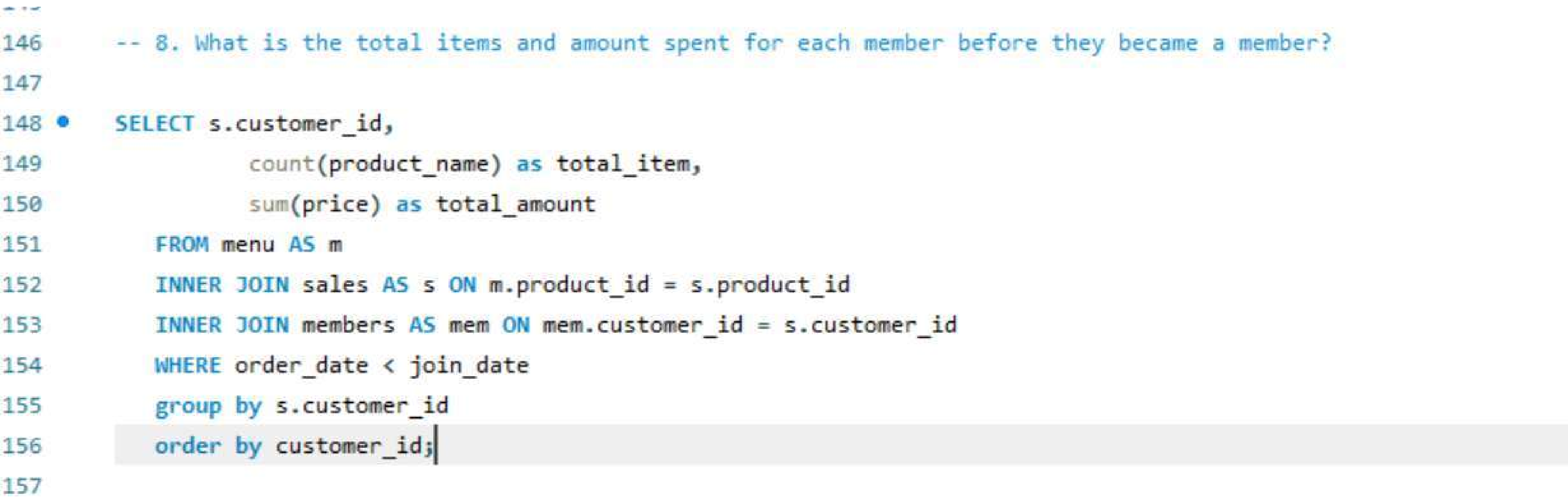
customer_id	product_name	order_count
A	ramen	3
B	curry	2
B	sushi	2
B	ramen	2
C	ramen	3



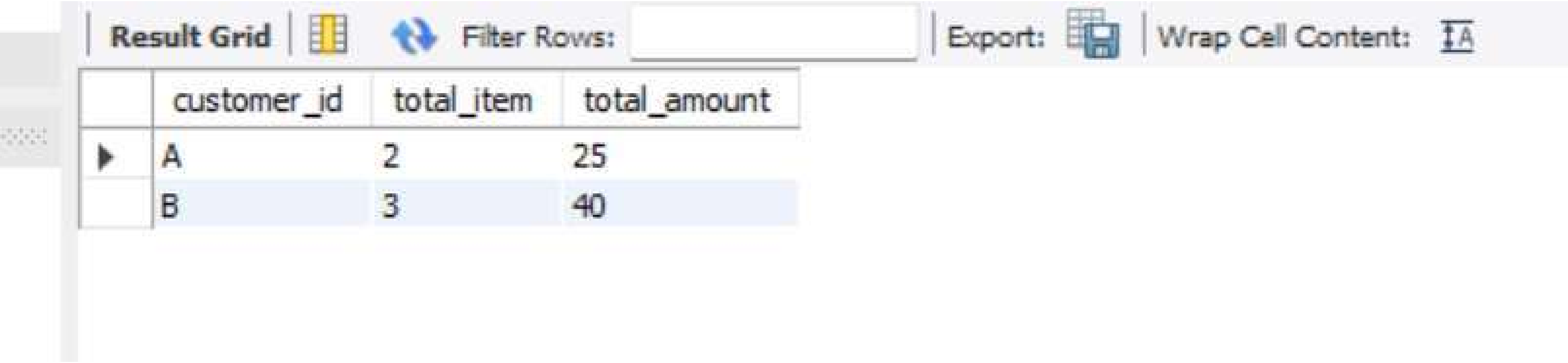
```sql
-- 6. Which item was purchased first by the customer after they became a member?

WITH diner_info AS
  (SELECT product_name,
          s.customer_id,
          order_date,
          join_date,
          m.product_id,
          RANK() OVER(PARTITION BY s.customer_id
                            ORDER BY s.order_date) AS first_item
   FROM menu AS m
   INNER JOIN sales AS s ON m.product_id = s.product_id
   INNER JOIN members AS mem ON mem.customer_id = s.customer_id
   WHERE order_date >= join_date )
SELECT customer_id,
       product_name,
       order_date
FROM diner_info
WHERE first_item=1;
```



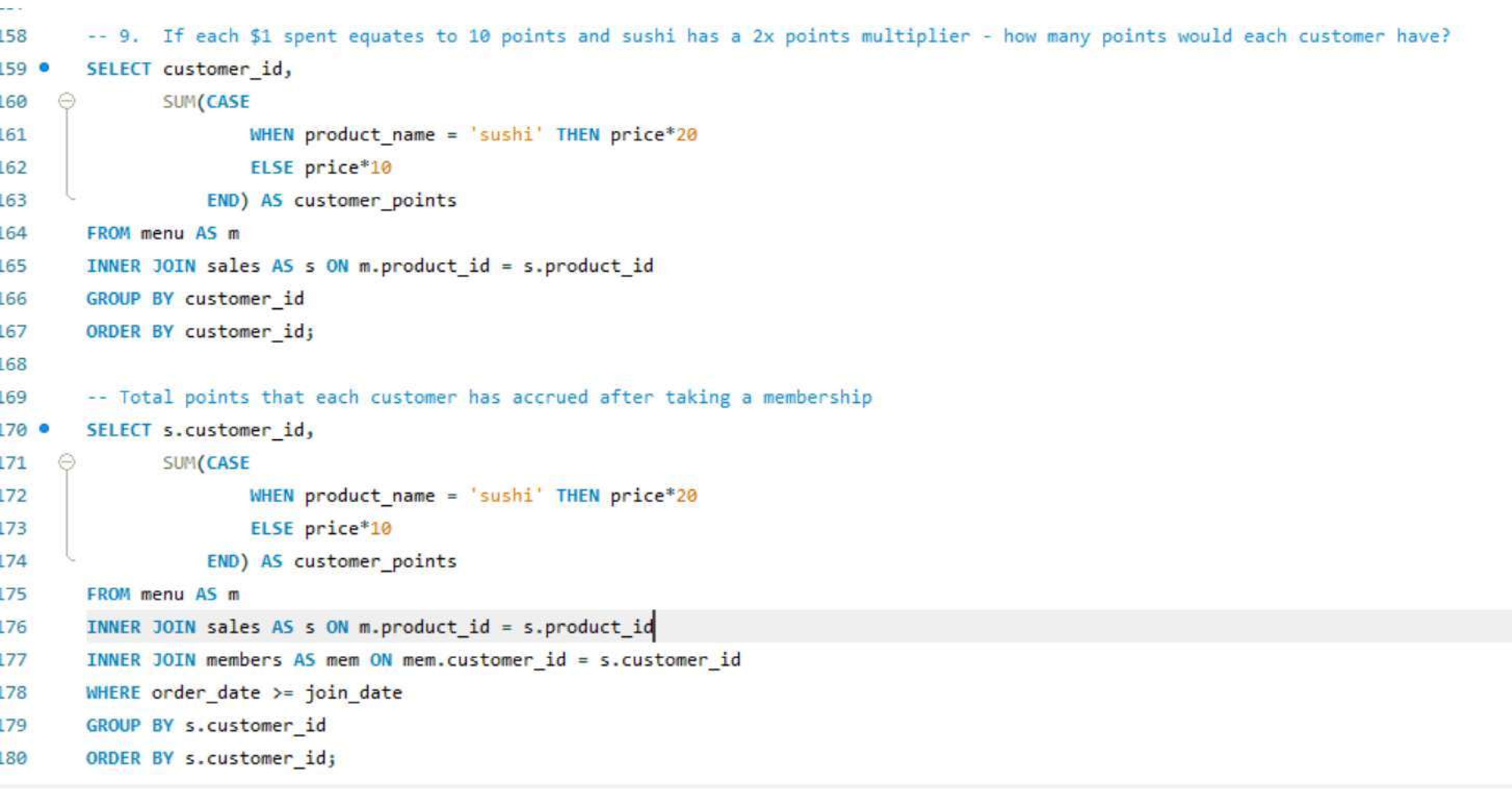
customer_id	product_name	order_date
A	curry	2021-01-07
B	sushi	2021-01-11



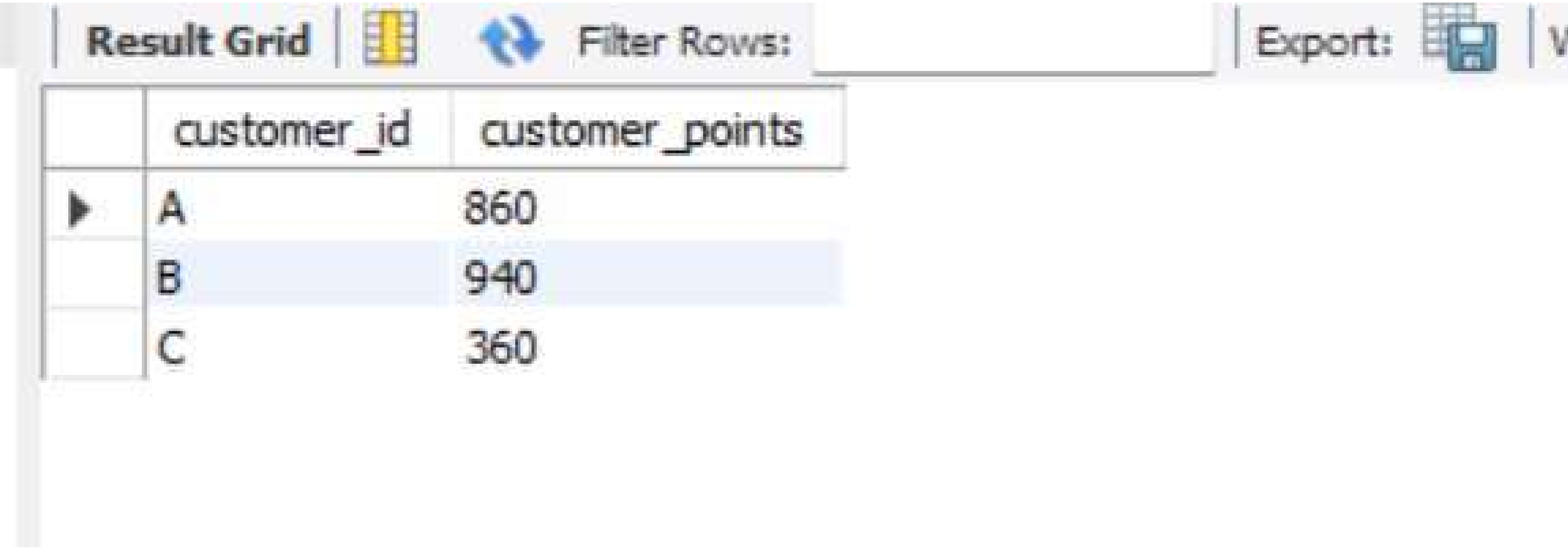
```sql
-- 8. What is the total items and amount spent for each member before they became a member?

SELECT s.customer_id,
          count(product_name) as total_item,
          sum(price) as total_amount
   FROM menu AS m
   INNER JOIN sales AS s ON m.product_id = s.product_id
   INNER JOIN members AS mem ON mem.customer_id = s.customer_id
   WHERE order_date < join_date
   group by s.customer_id
   order by customer_id;
```



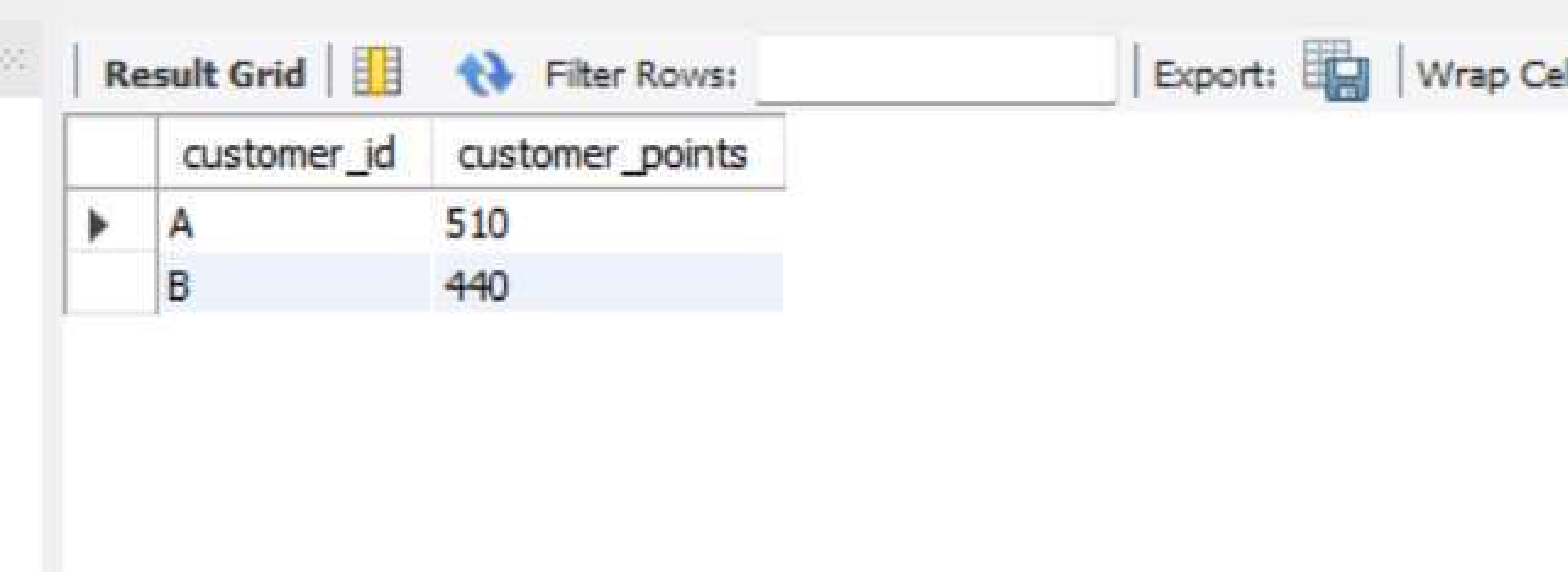
customer_id	total_item	total_amount
A	2	25
B	3	40



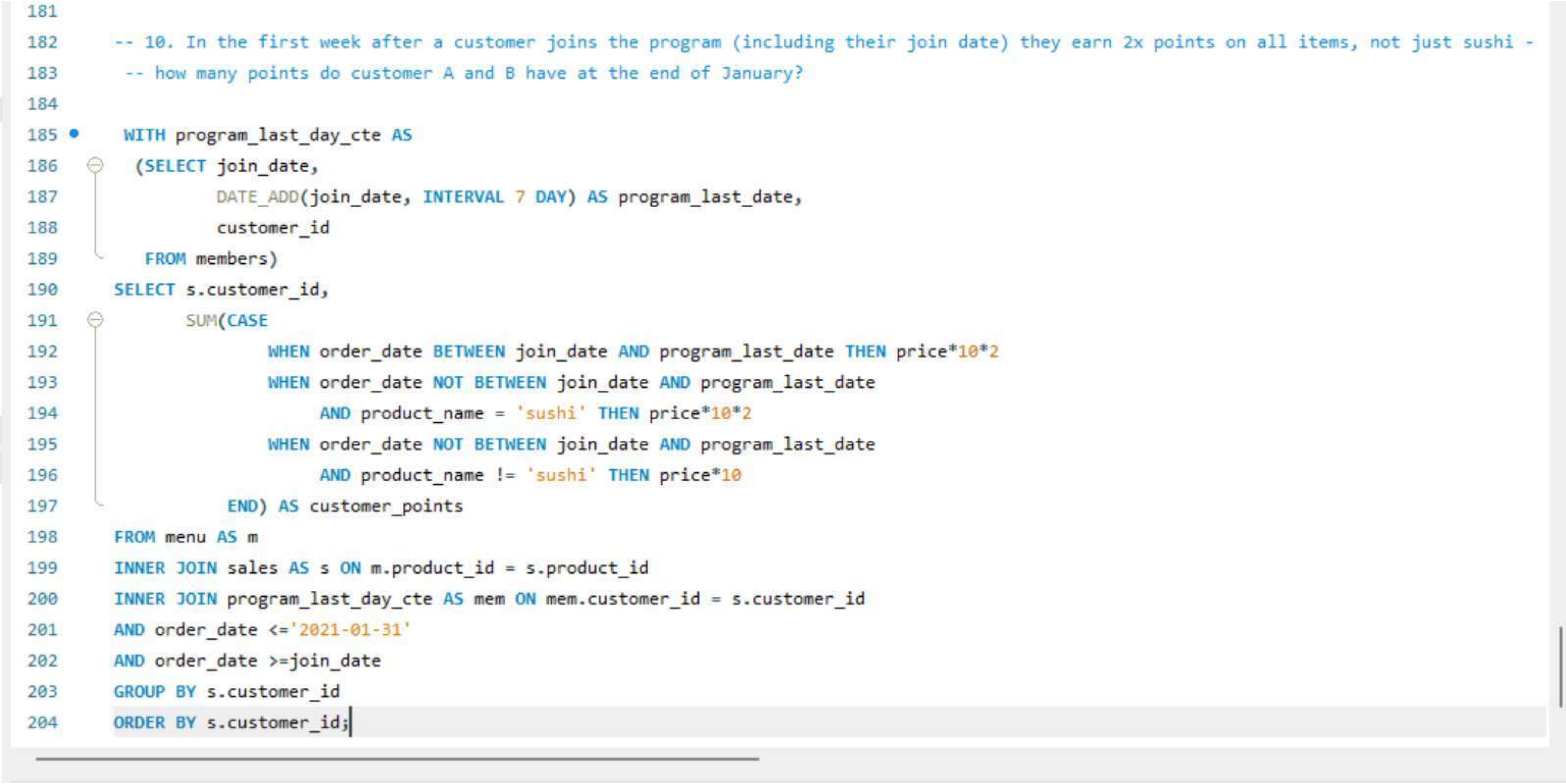
```sql
-- 9.  If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?
SELECT customer_id,
       SUM(CASE
               WHEN product_name = 'sushi' THEN price*20
               ELSE price*10
           END) AS customer_points
FROM menu AS m
INNER JOIN sales AS s ON m.product_id = s.product_id
GROUP BY customer_id
ORDER BY customer_id;

-- Total points that each customer has accrued after taking a membership
SELECT s.customer_id,
       SUM(CASE
               WHEN product_name = 'sushi' THEN price*20
               ELSE price*10
           END) AS customer_points
FROM menu AS m
INNER JOIN sales AS s ON m.product_id = s.product_id
INNER JOIN members AS mem ON mem.customer_id = s.customer_id
WHERE order_date >= join_date
GROUP BY s.customer_id
ORDER BY s.customer_id;
```



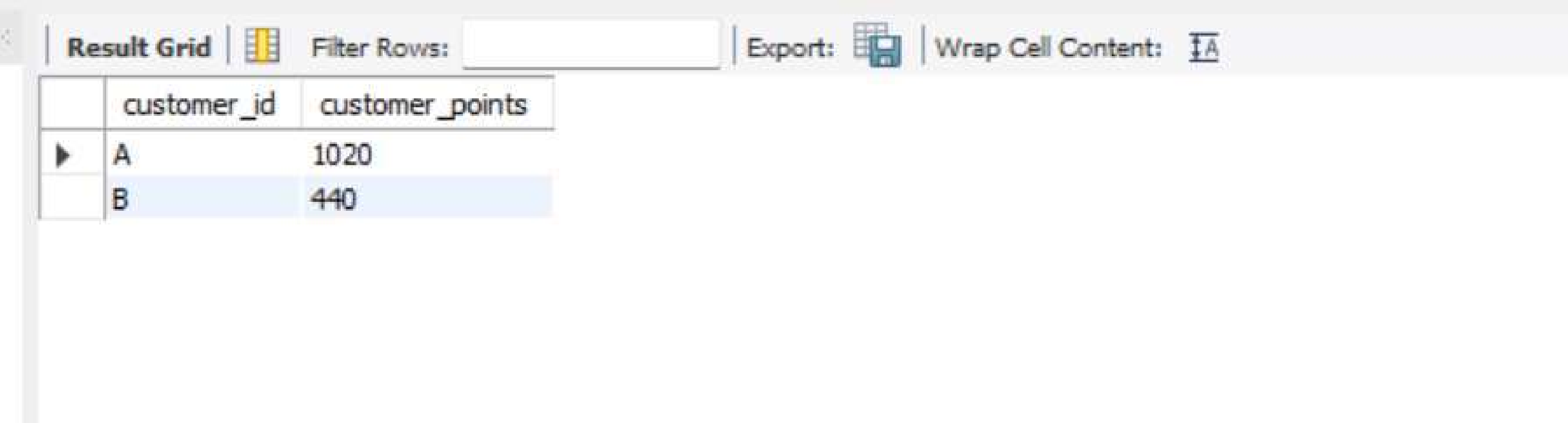
customer_id	customer_points
A	860
B	940
C	360



customer_id	customer_points
A	510
B	440



```sql
-- 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 program_last_day_cte AS
  (SELECT join_date,
          DATE_ADD(join_date, INTERVAL 7 DAY) AS program_last_date,
          customer_id
   FROM members)
SELECT s.customer_id,
       SUM(CASE
               WHEN order_date BETWEEN join_date AND program_last_date THEN price*10*2
               WHEN order_date NOT BETWEEN join_date AND program_last_date
                    AND product_name = 'sushi' THEN price*10*2
               WHEN order_date NOT BETWEEN join_date AND program_last_date
                    AND product_name != 'sushi' THEN price*10
           END) AS customer_points
FROM menu AS m
INNER JOIN sales AS s ON m.product_id = s.product_id
INNER JOIN program_last_day_cte AS mem ON mem.customer_id = s.customer_id
AND order_date <='2021-01-31'
AND order_date >=join_date
GROUP BY s.customer_id
ORDER BY s.customer_id;
```



customer_id	customer_points
A	1020
B	440