Danny's Diner

CASE STUDY 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.

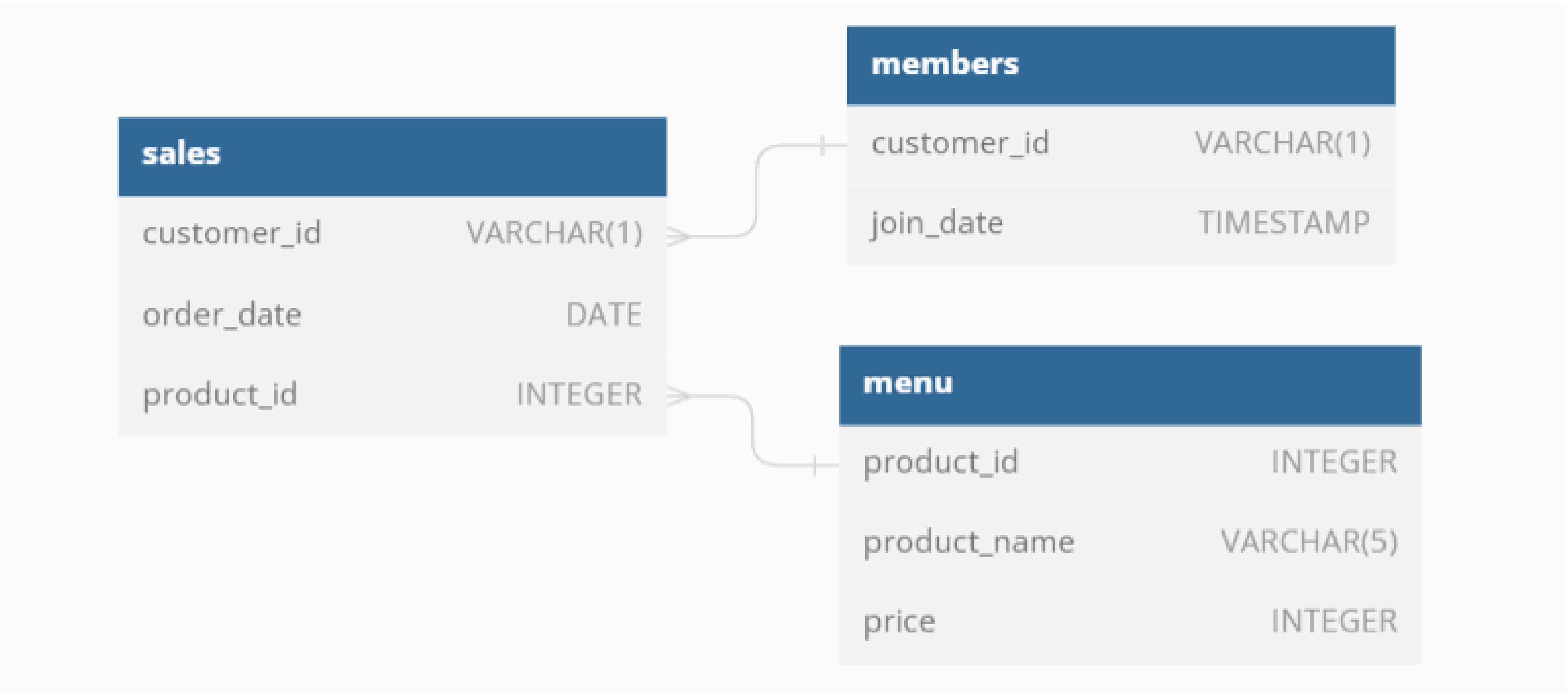
Problem Statement

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

Danny has shared 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

Datasets

01

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.

02

Menu

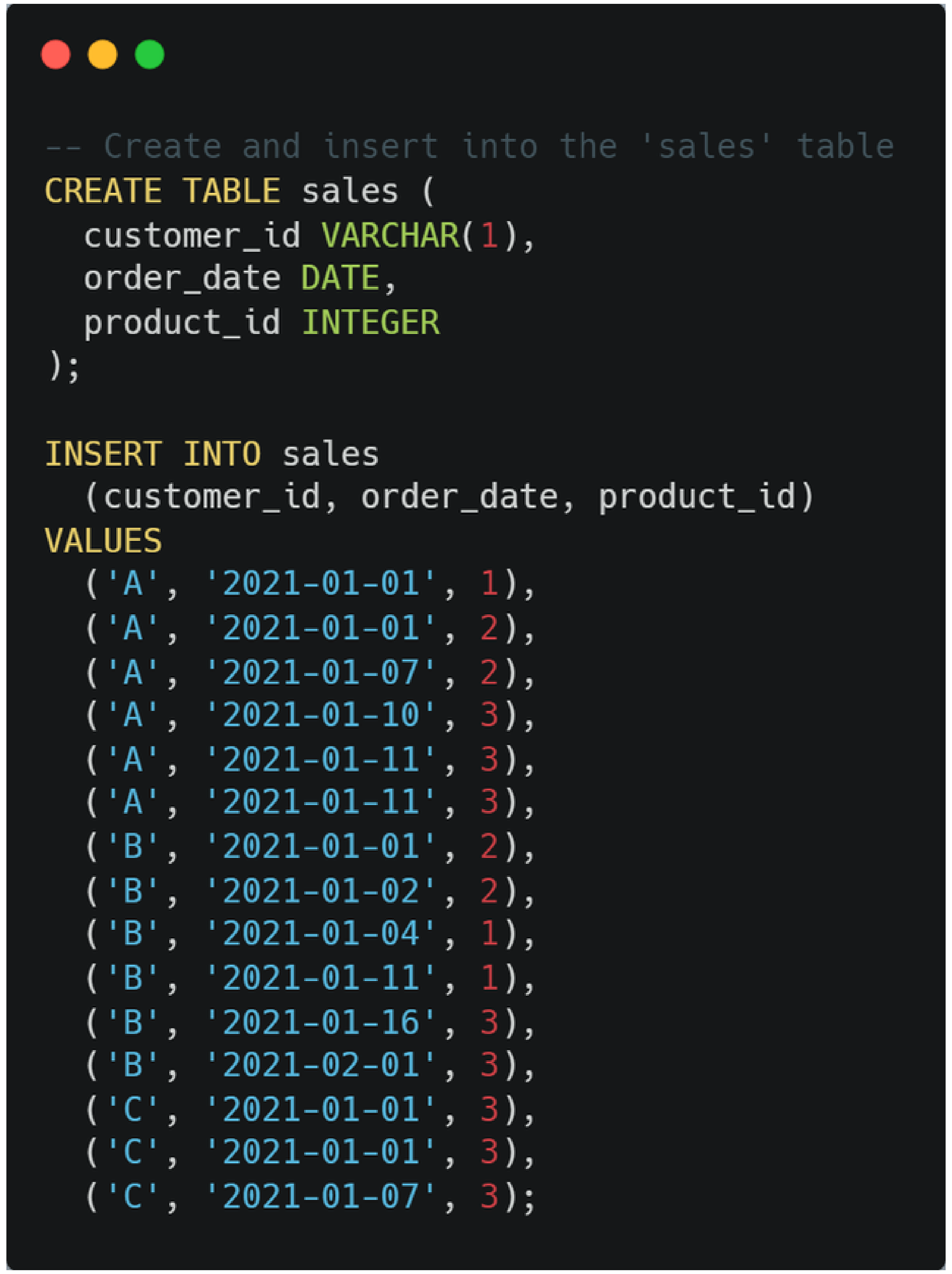
The menu table maps the product_id to the actual product_name and price of each menu item.

03

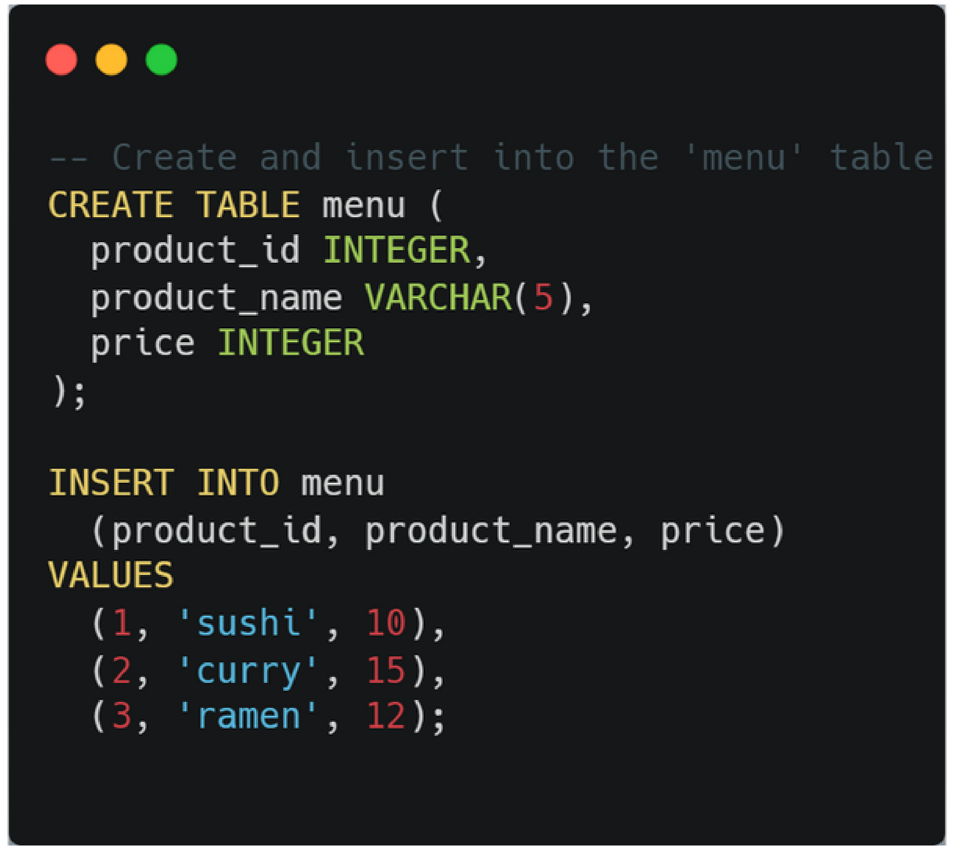
Members

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

sales

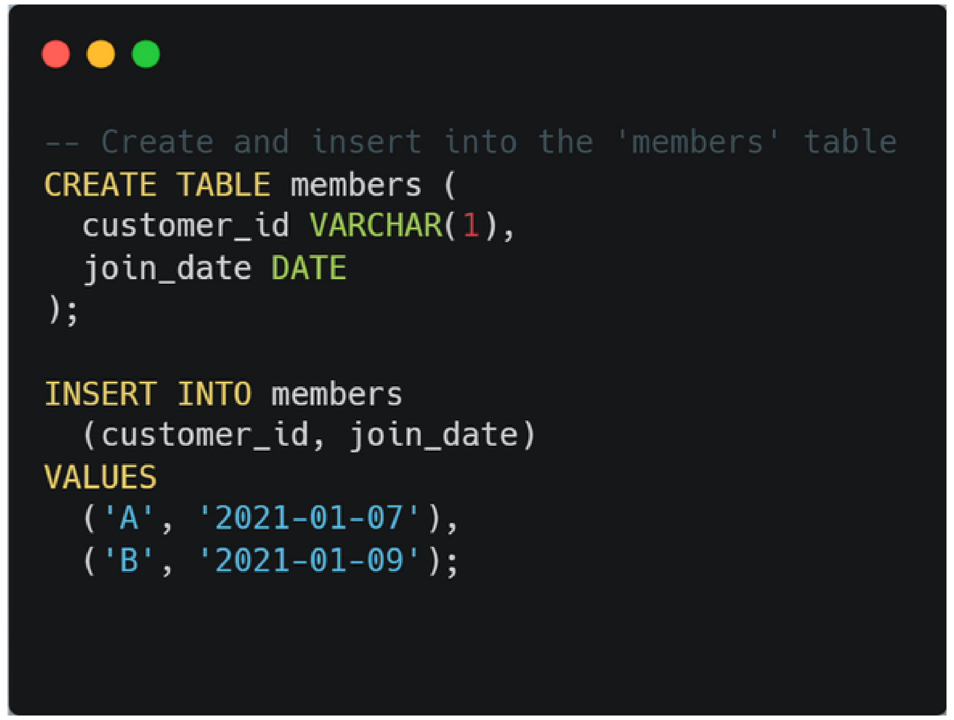


```sql
-- Create and insert into the 'sales' table
CREATE TABLE sales (
  customer_id VARCHAR(1),
  order_date DATE,
  product_id INTEGER
);

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);
```



```sql
-- Create and insert into the 'menu' table
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);
```

menu

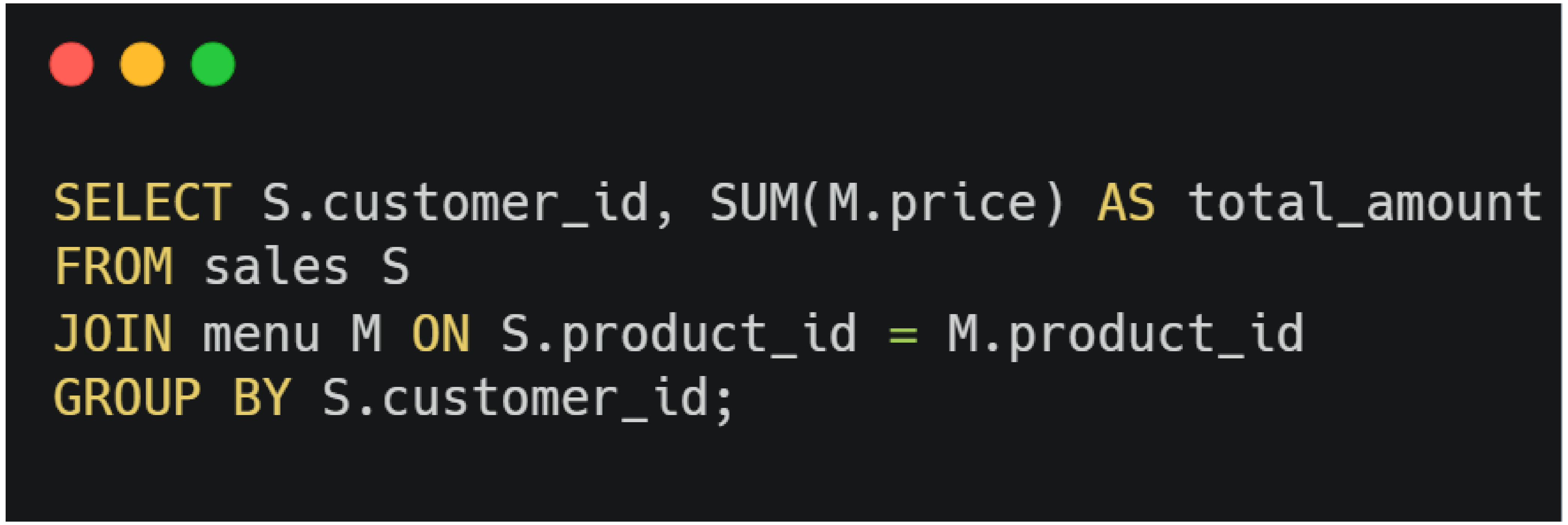


```sql
-- Create and insert into the 'members' table
CREATE TABLE members (
  customer_id VARCHAR(1),
  join_date DATE
);

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

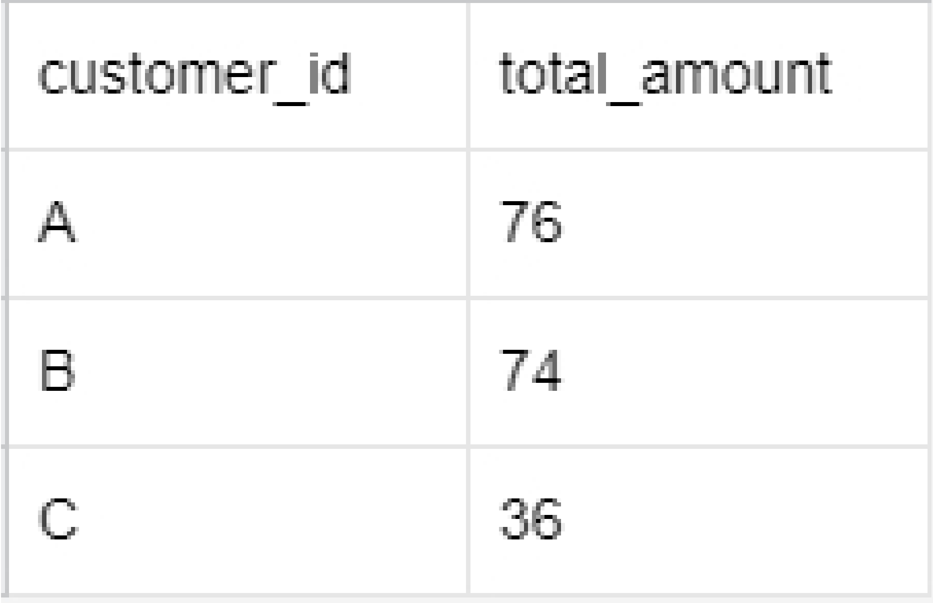
members

Case Study Questions

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



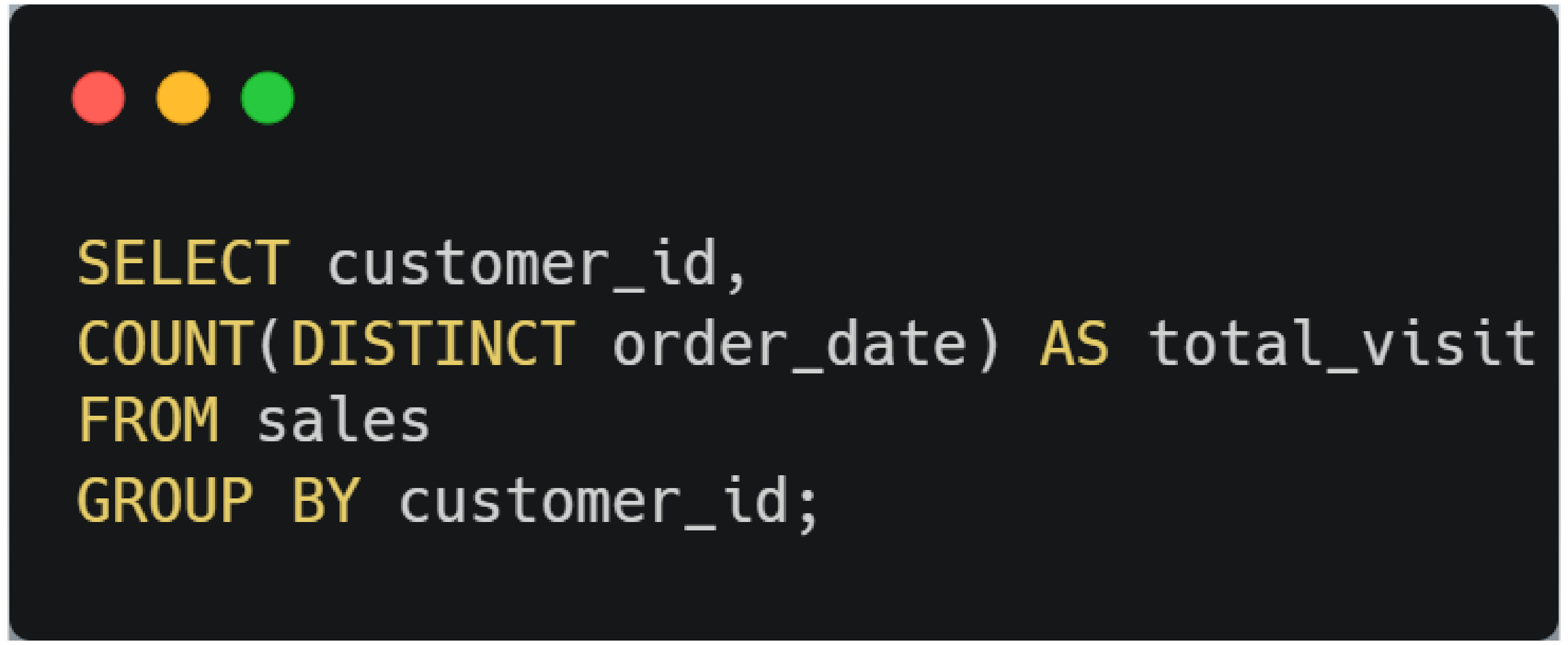
```sql
SELECT S.customer_id, SUM(M.price) AS total_amount
FROM sales S
JOIN menu M ON S.product_id = M.product_id
GROUP BY S.customer_id;
```



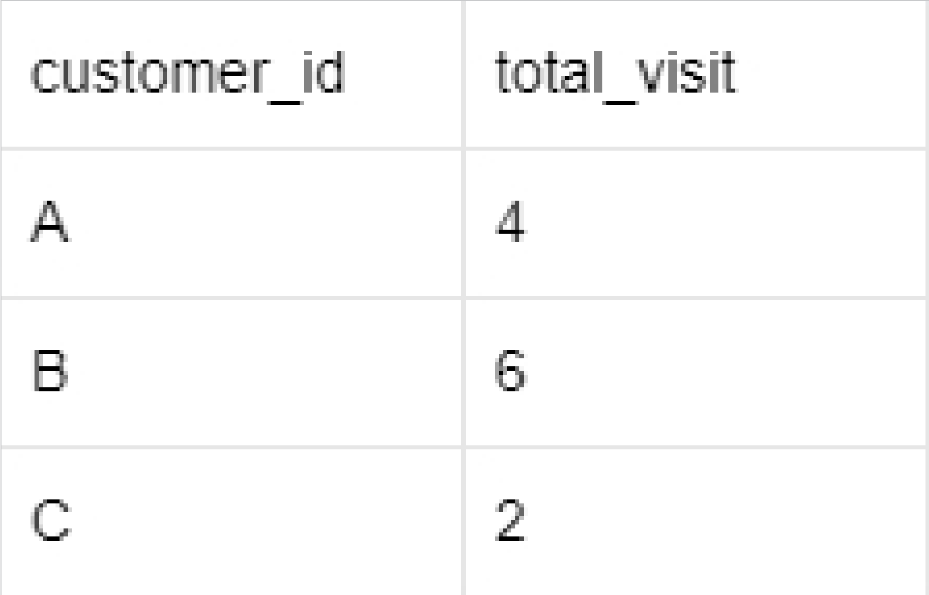
customer_id	total_amount
A	76
B	74
C	36

Customer A, B, and C have made expenditures of $76, $74, and $36 respectively.

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



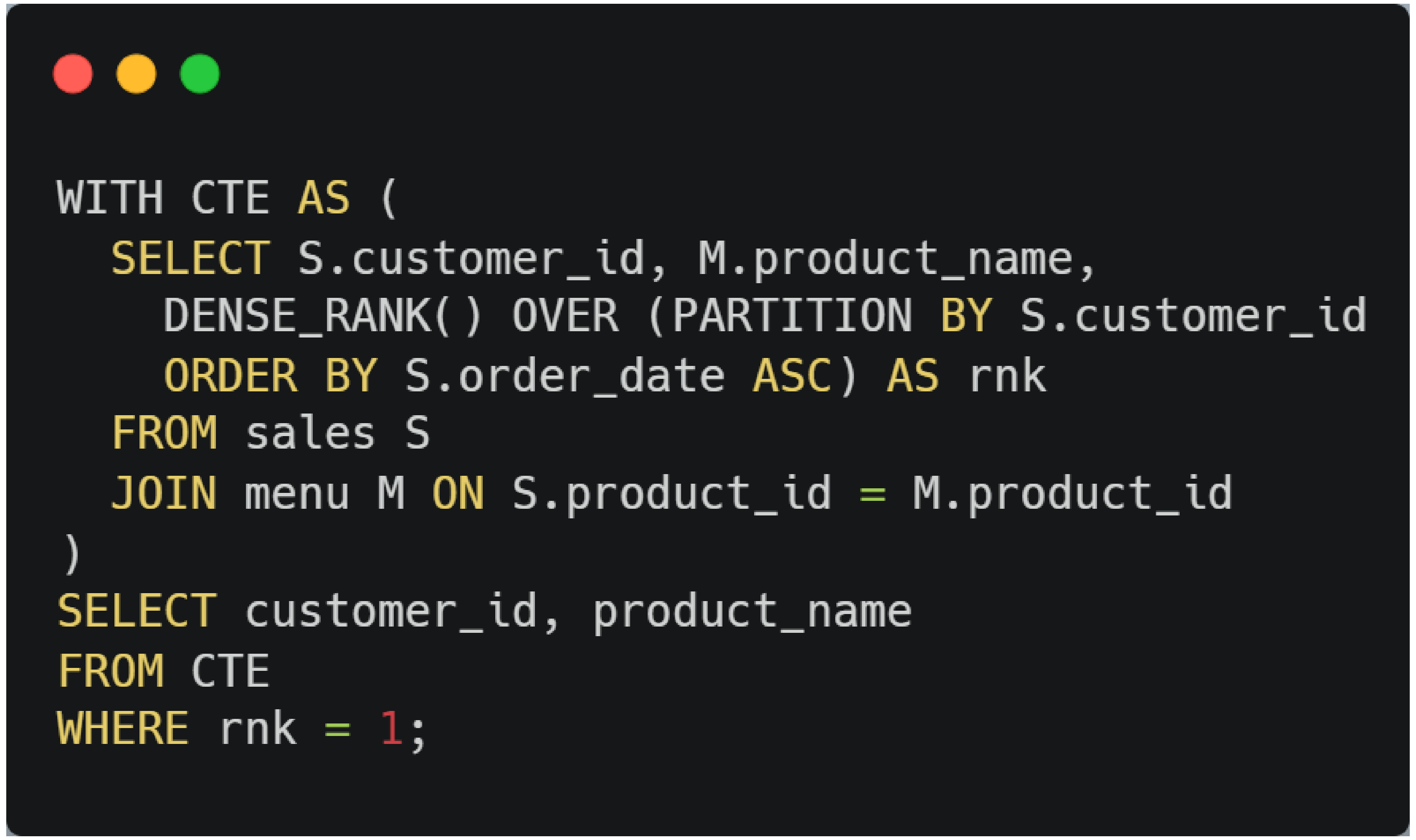
```
SELECT customer_id,
COUNT(DISTINCT order_date) AS total_visit
FROM sales
GROUP BY customer_id;
```



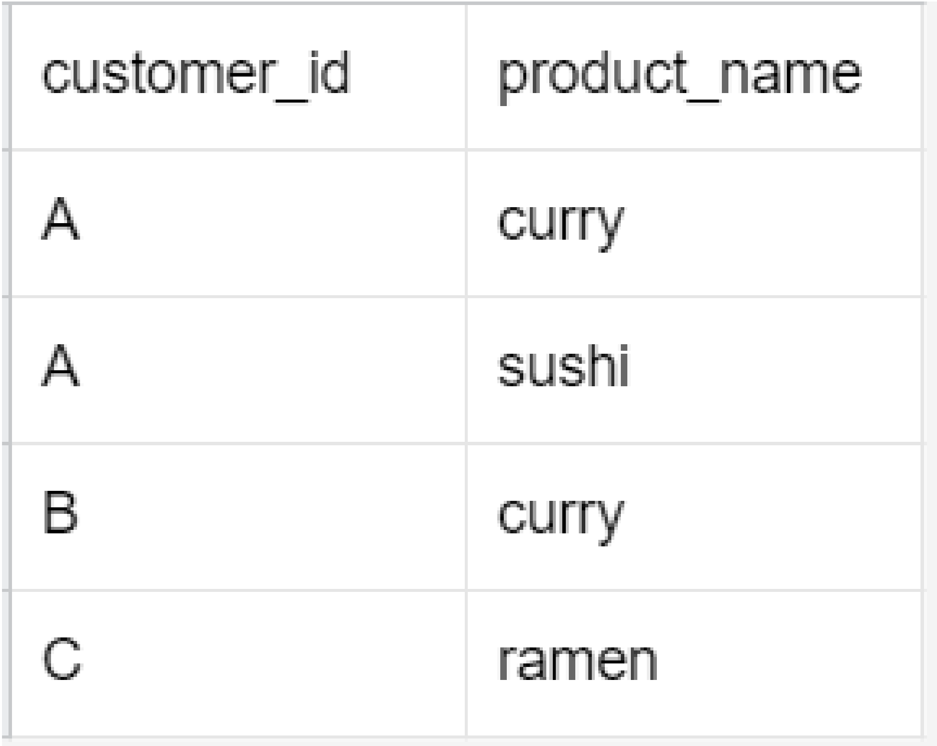
customer_id	total_visit
A	4
B	6
C	2

Customer A visited 4 times, Customer B visited 6 times, and Customer C visited 2 times.

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



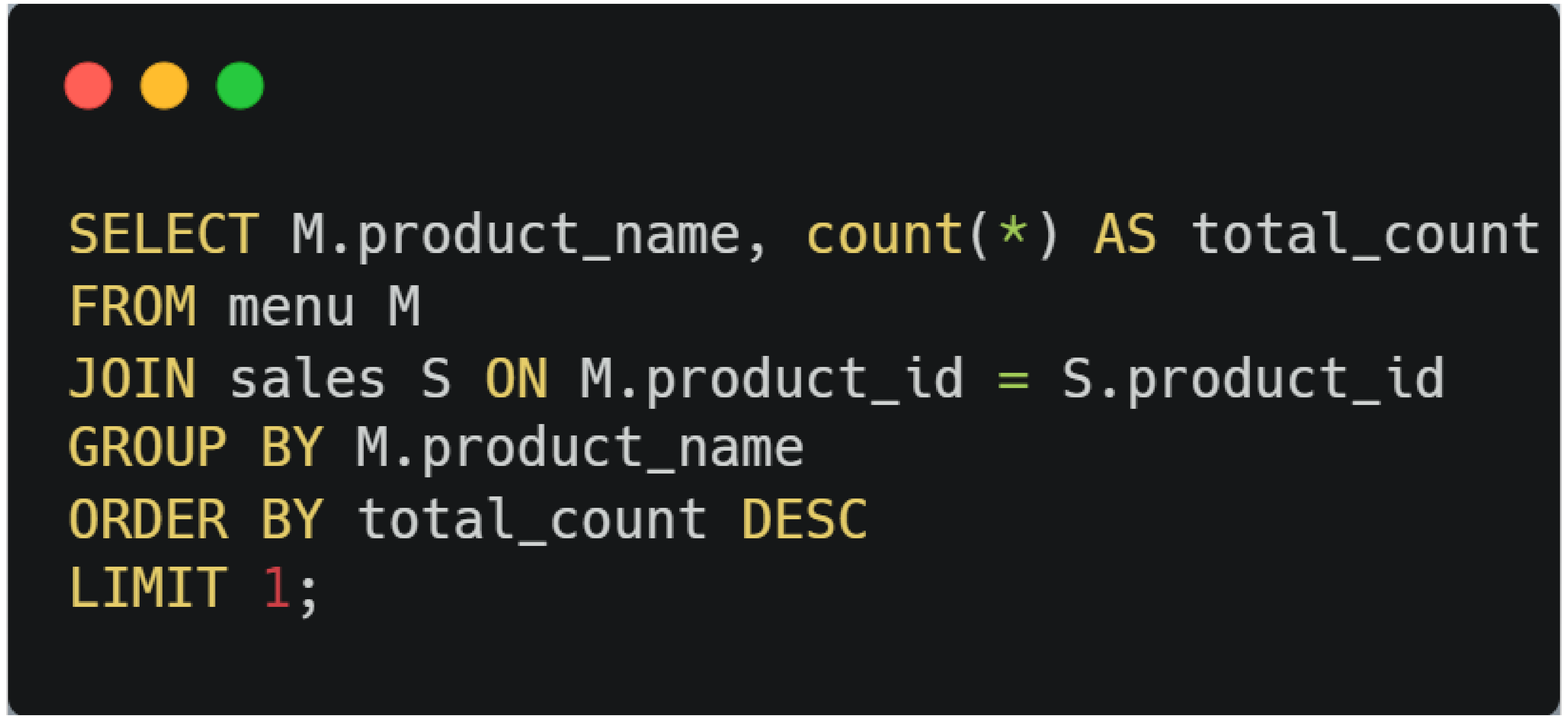
```
WITH CTE AS (
  SELECT S.customer_id, M.product_name,
    DENSE_RANK() OVER (PARTITION BY S.customer_id
    ORDER BY S.order_date ASC) AS rnk
  FROM sales S
  JOIN menu M ON S.product_id = M.product_id
)
SELECT customer_id, product_name
FROM CTE
WHERE rnk = 1;
```



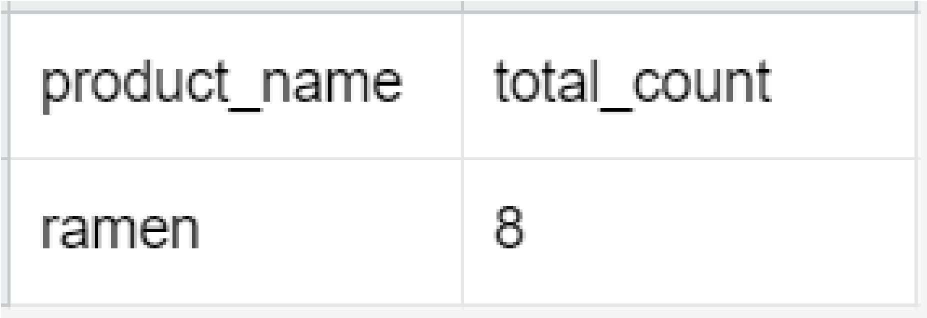
customer_id	product_name
A	curry
A	sushi
B	curry
C	ramen

- Customer A placed their first order for curry and sushi.
- Customer B placed their first order with curry.
- Customer C placed their first order with ramen.

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



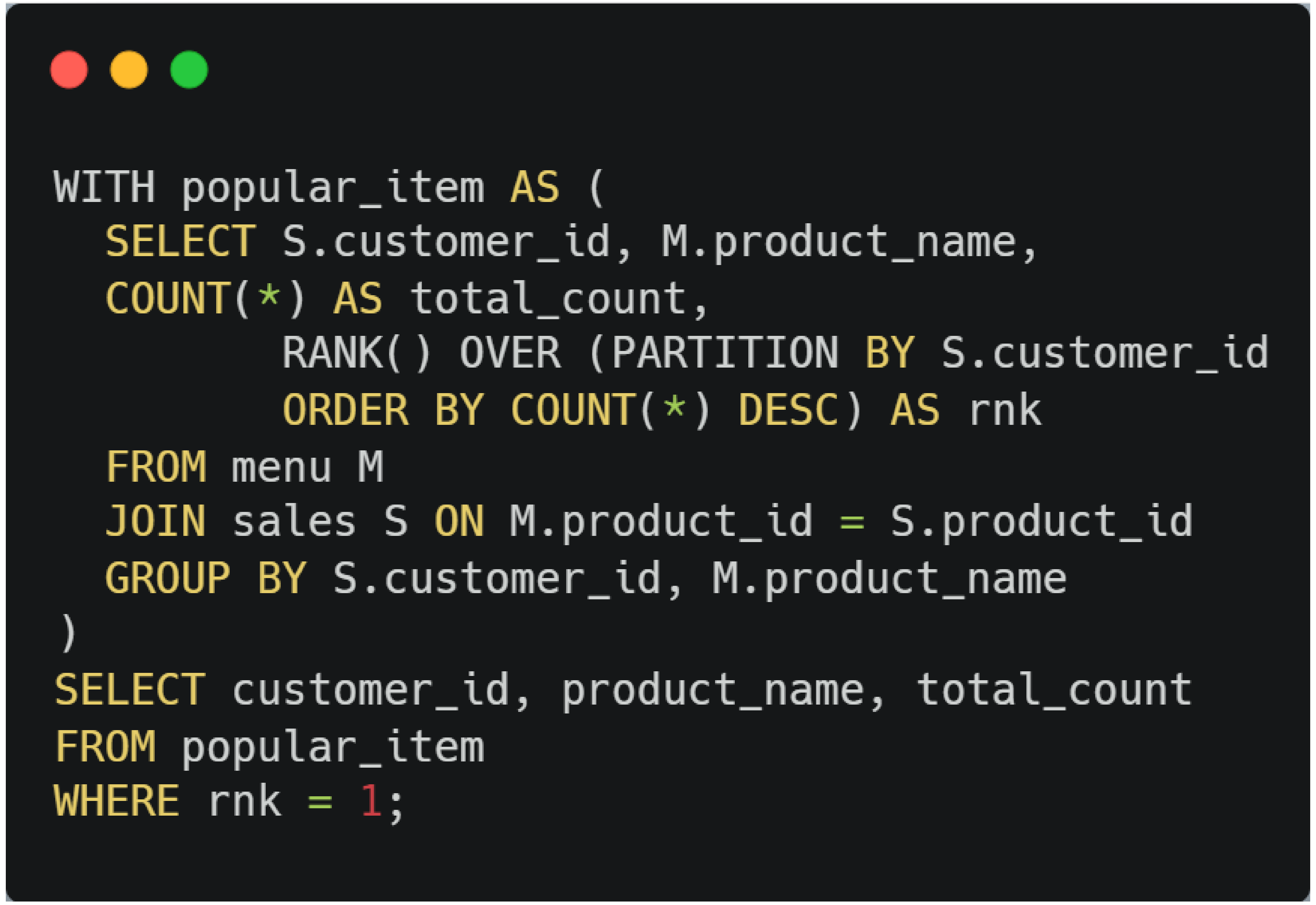
```
SELECT M.product_name, count(*) AS total_count
FROM menu M
JOIN sales S ON M.product_id = S.product_id
GROUP BY M.product_name
ORDER BY total_count DESC
LIMIT 1;
```



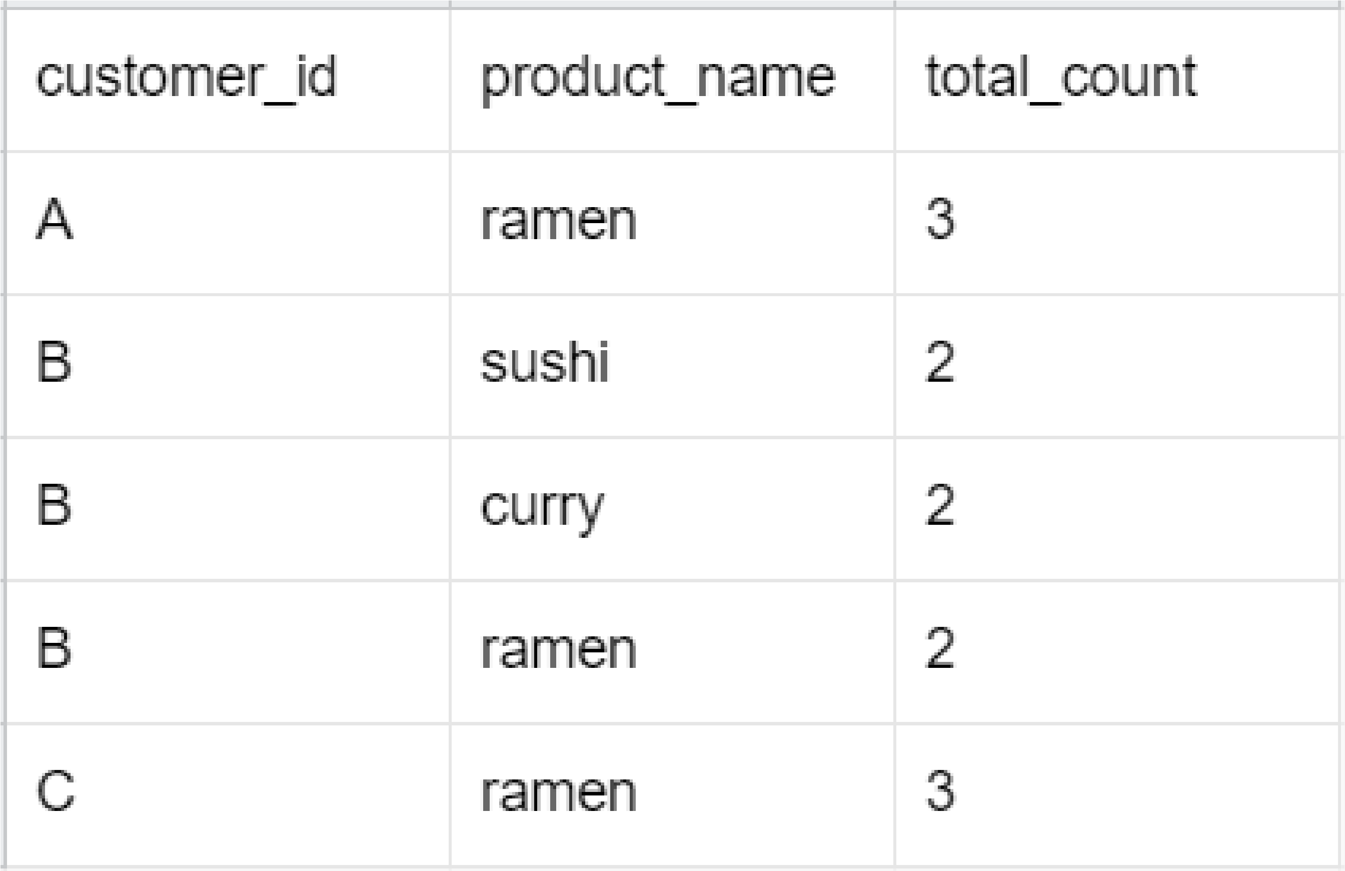
product_name	total_count
ramen	8

The menu item that has been most frequently purchased is 'ramen,' with a total of 8 orders.

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



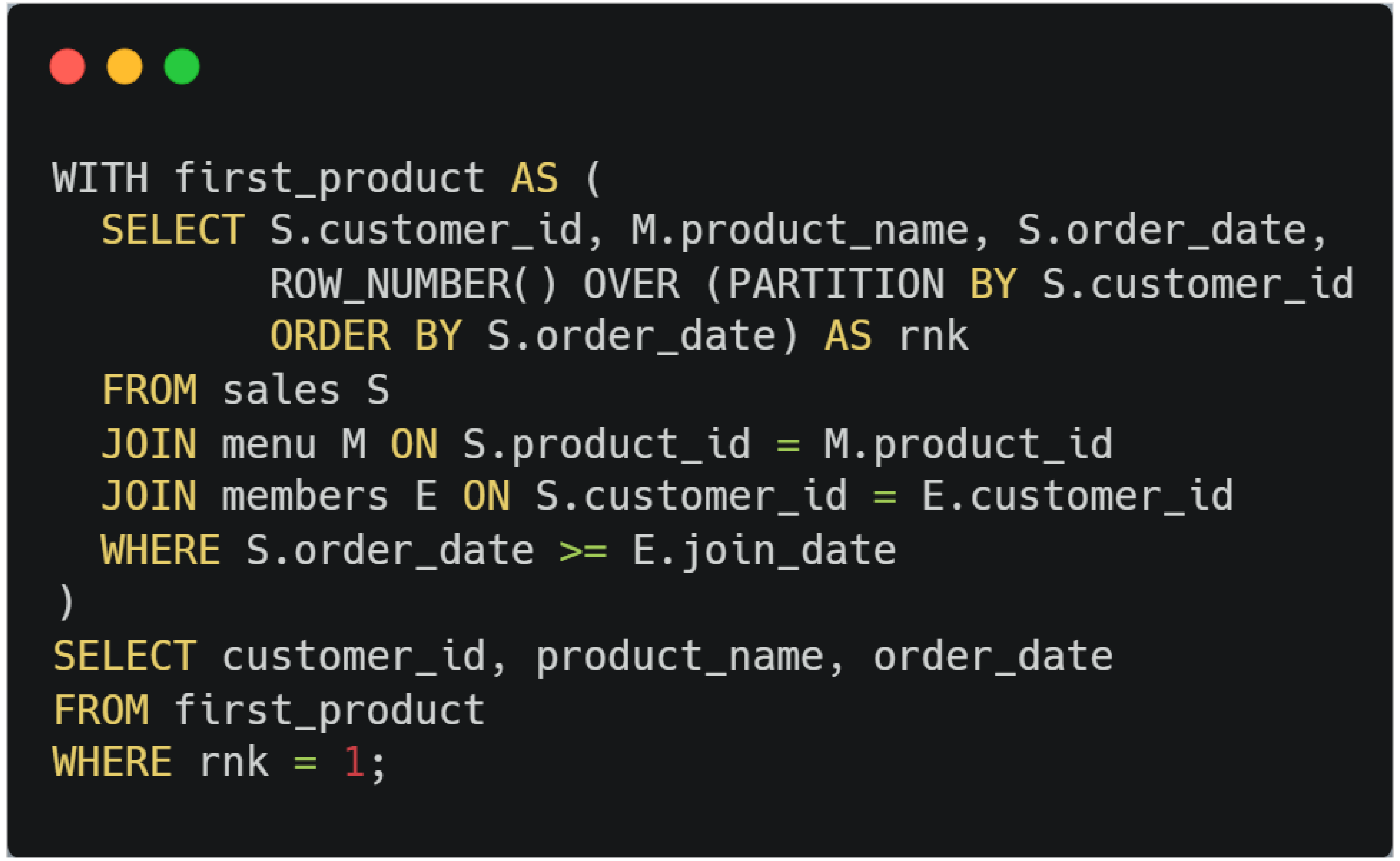
```
WITH popular_item AS (
  SELECT S.customer_id, M.product_name,
  COUNT(*) AS total_count,
         RANK() OVER (PARTITION BY S.customer_id
         ORDER BY COUNT(*) DESC) AS rnk
  FROM menu M
  JOIN sales S ON M.product_id = S.product_id
  GROUP BY S.customer_id, M.product_name
)
SELECT customer_id, product_name, total_count
FROM popular_item
WHERE rnk = 1;
```



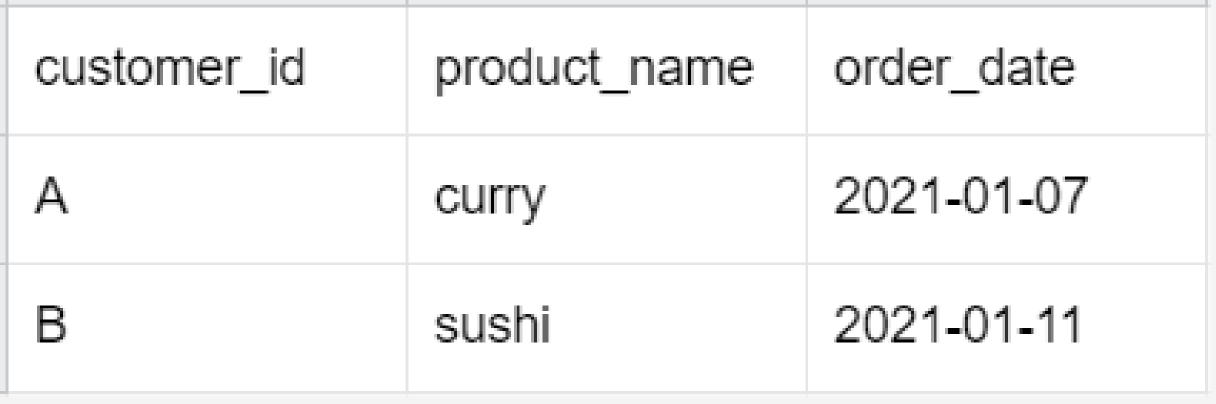
customer_id	product_name	total_count
A	ramen	3
B	sushi	2
B	curry	2
B	ramen	2
C	ramen	3

Ramen is the preferred choice for both Customer A and Customer C, while Customer B enjoys a variety of items from the menu.

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



```
WITH first_product AS (
  SELECT S.customer_id, M.product_name, S.order_date,
         ROW_NUMBER() OVER (PARTITION BY S.customer_id
         ORDER BY S.order_date) AS rnk
  FROM sales S
  JOIN menu M ON S.product_id = M.product_id
  JOIN members E ON S.customer_id = E.customer_id
  WHERE S.order_date >= E.join_date
)
SELECT customer_id, product_name, order_date
FROM first_product
WHERE rnk = 1;
```

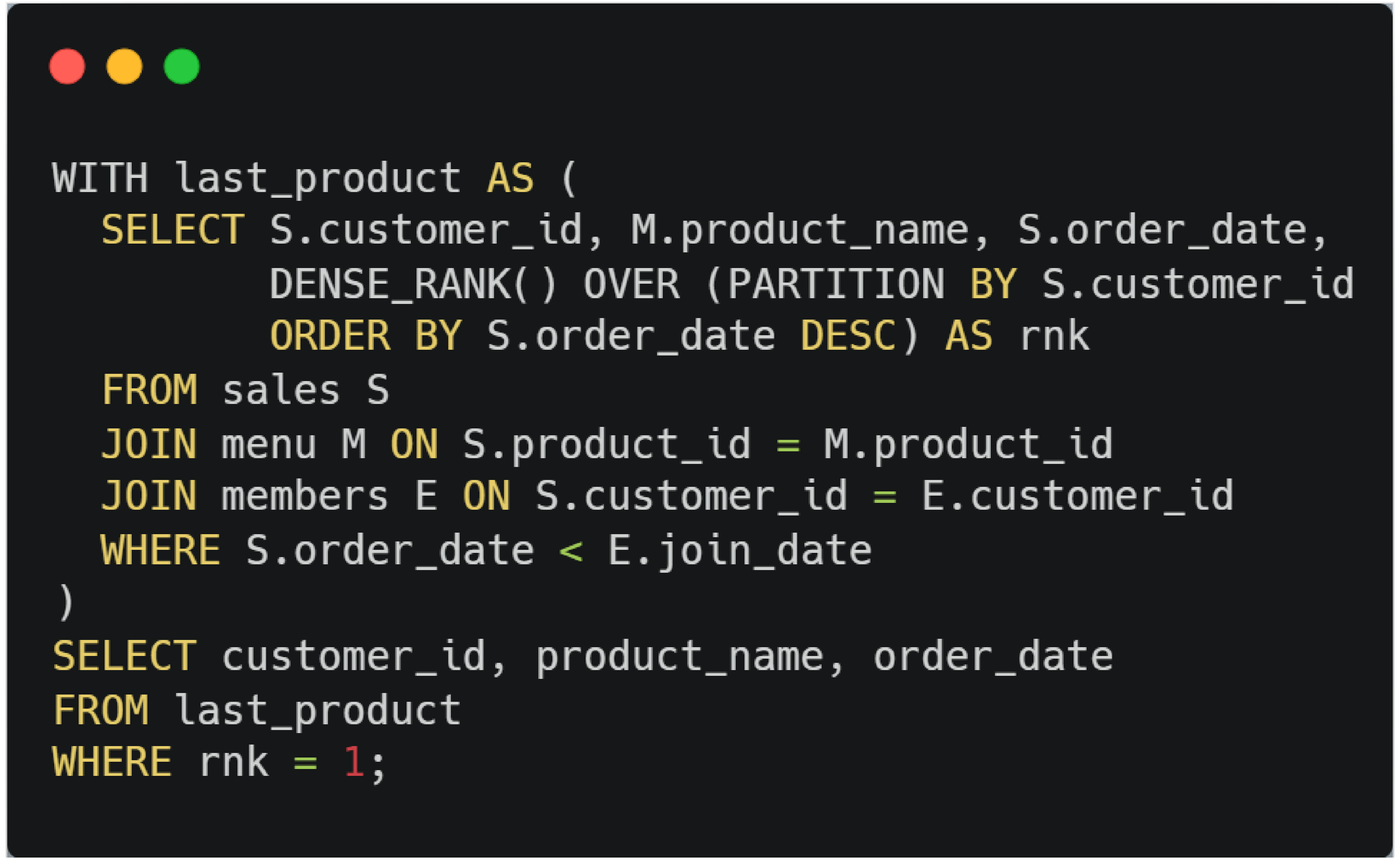


customer_id	product_name	order_date
A	curry	2021-01-07
B	sushi	2021-01-11

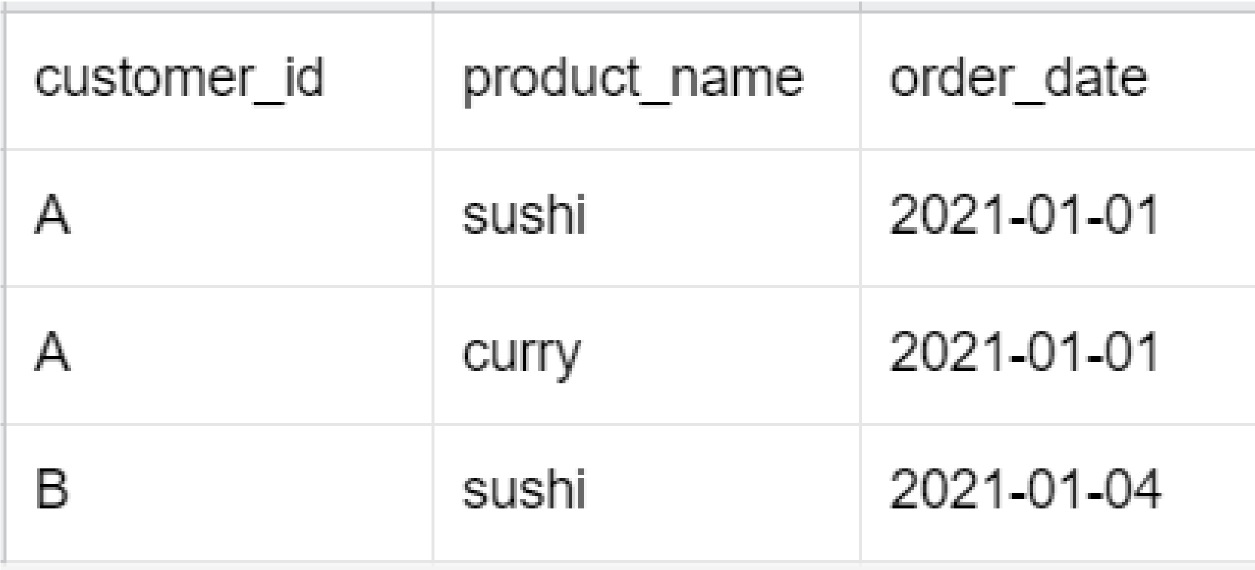
After becoming a member,

- Customer A's first order was curry.
- Customer B's first order was sushi.

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



```sql
WITH last_product AS (
  SELECT S.customer_id, M.product_name, S.order_date,
         DENSE_RANK() OVER (PARTITION BY S.customer_id
         ORDER BY S.order_date DESC) AS rnk
  FROM sales S
  JOIN menu M ON S.product_id = M.product_id
  JOIN members E ON S.customer_id = E.customer_id
  WHERE S.order_date < E.join_date
)
SELECT customer_id, product_name, order_date
FROM last_product
WHERE rnk = 1;
```

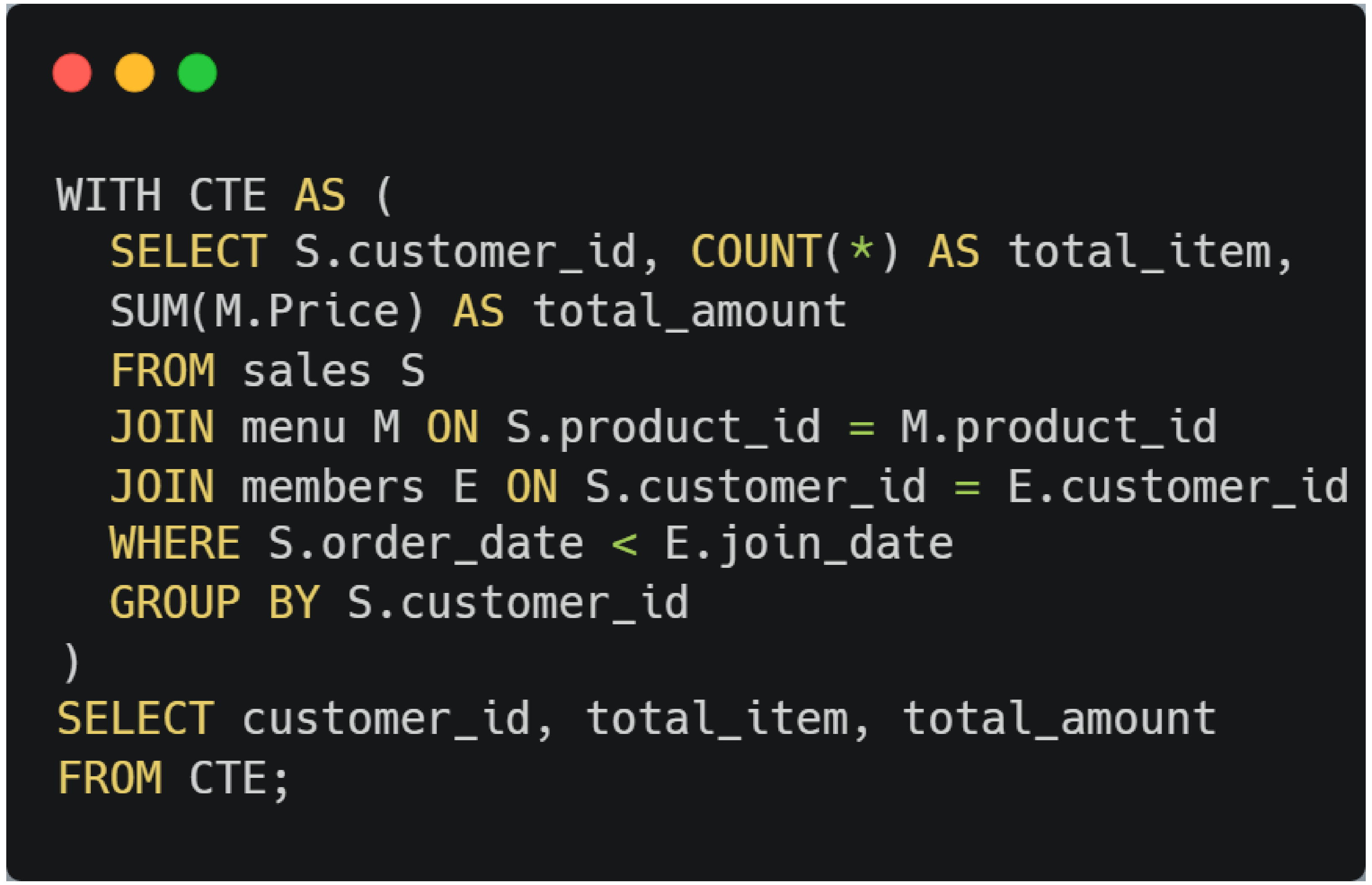


customer_id	product_name	order_date
A	sushi	2021-01-01
A	curry	2021-01-01
B	sushi	2021-01-04

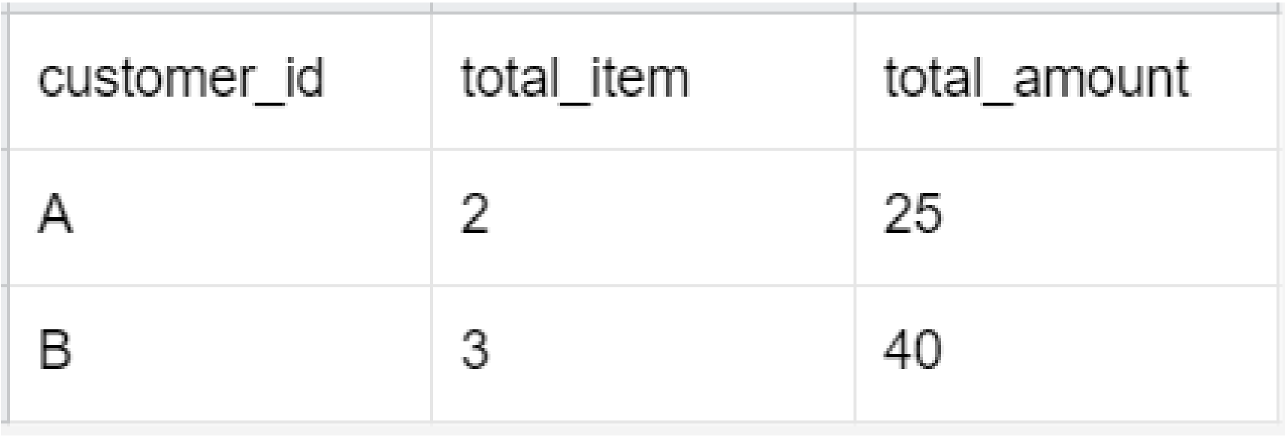
Before becoming a member,

- Customer A’s last order was sushi and curry.
- Customer B’s last order sushi.

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



```
WITH CTE AS (
  SELECT S.customer_id, COUNT(*) AS total_item,
  SUM(M.Price) AS total_amount
  FROM sales S
  JOIN menu M ON S.product_id = M.product_id
  JOIN members E ON S.customer_id = E.customer_id
  WHERE S.order_date < E.join_date
  GROUP BY S.customer_id
)
SELECT customer_id, total_item, total_amount
FROM CTE;
```

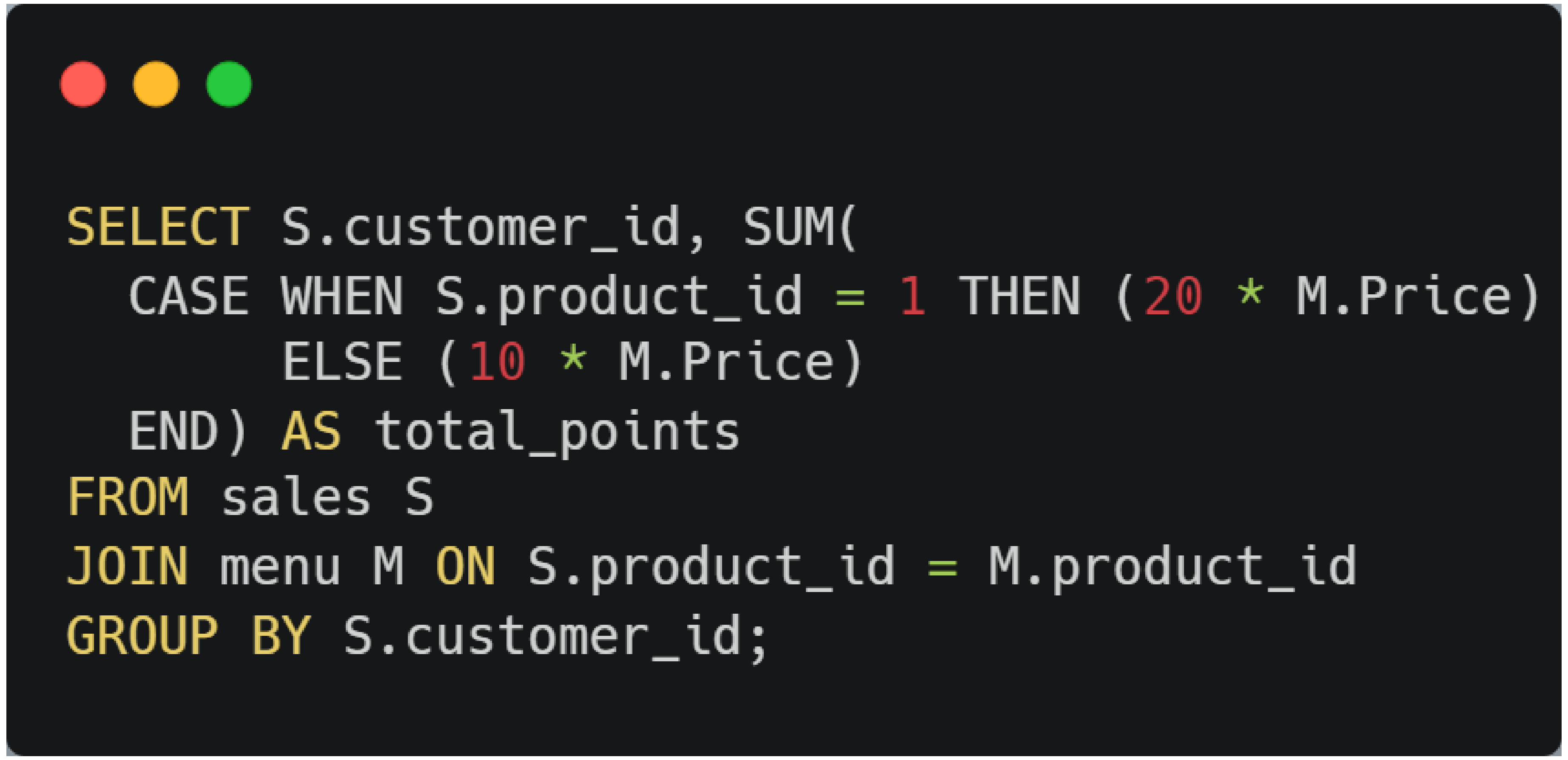


customer_id	total_item	total_amount
A	2	25
B	3	40

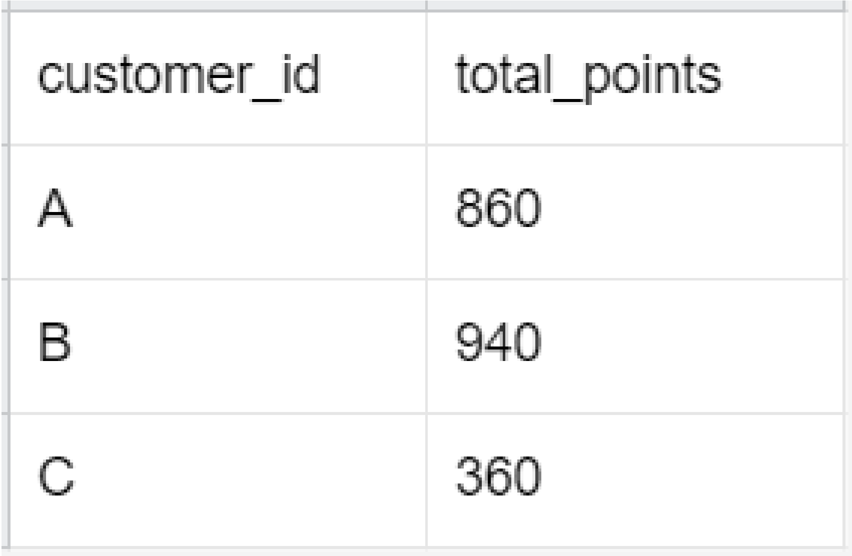
Before becoming a member,

- Customer A spent $25 on 2 items.
- Customer B spent $40 on 3 items.

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



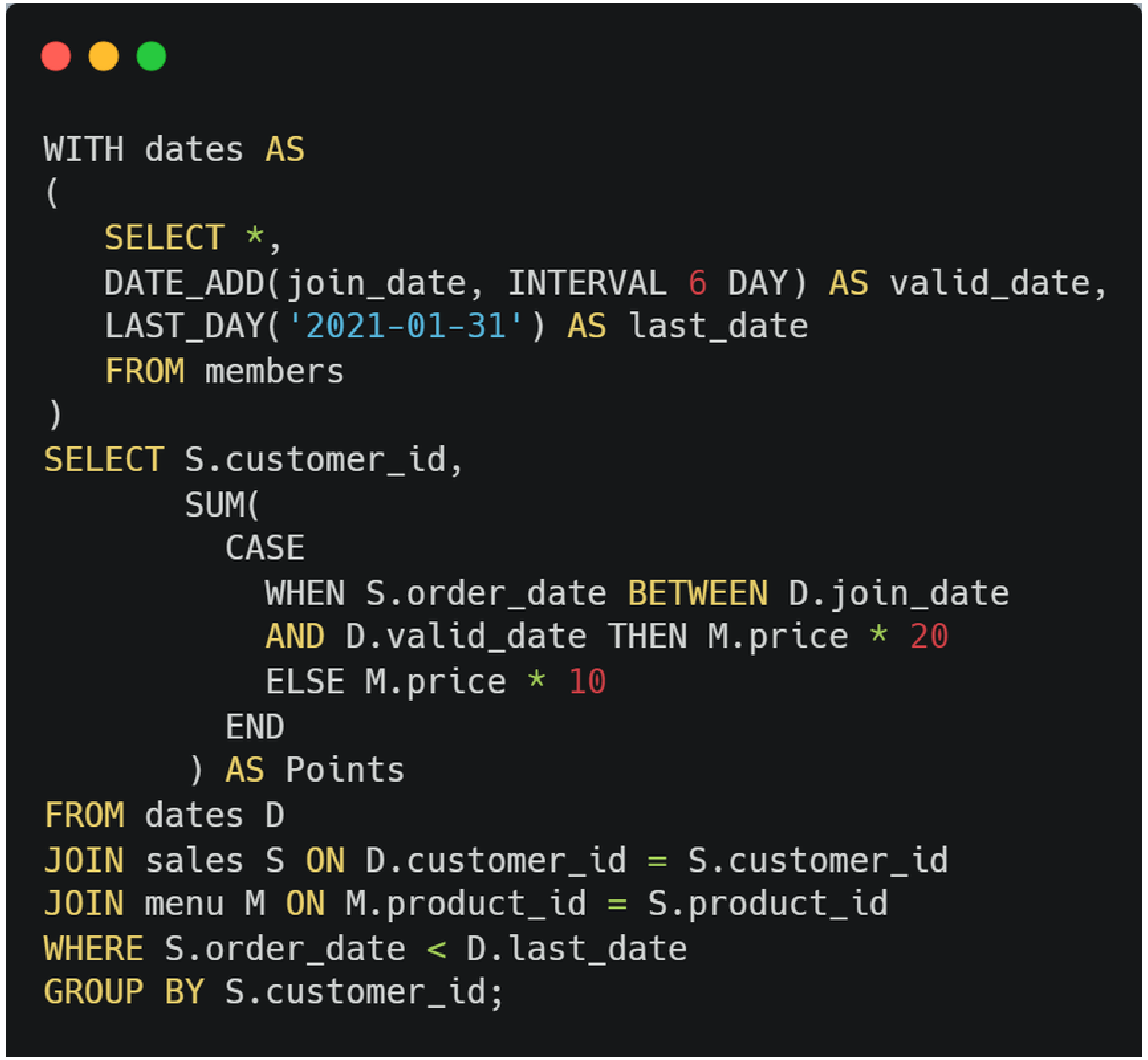
```
SELECT S.customer_id, SUM(
  CASE WHEN S.product_id = 1 THEN (20 * M.Price)
       ELSE (10 * M.Price)
  END) AS total_points
FROM sales S
JOIN menu M ON S.product_id = M.product_id
GROUP BY S.customer_id;
```



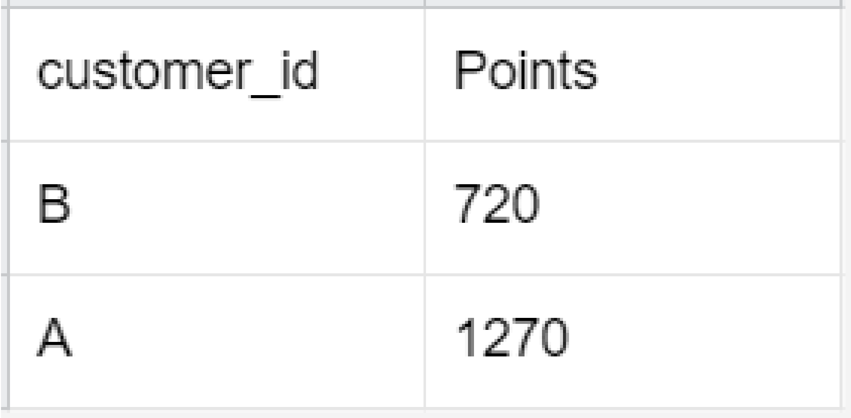
customer_id	total_points
A	860
B	940
C	360

Customer A has accumulated a total of 860 points, while Customer B has earned 940 points, and Customer C has a total of 360 points.

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 dates AS
(
   SELECT *,
   DATE_ADD(join_date, INTERVAL 6 DAY) AS valid_date,
   LAST_DAY('2021-01-31') AS last_date
   FROM members
)
SELECT S.customer_id,
       SUM(
         CASE
           WHEN S.order_date BETWEEN D.join_date
           AND D.valid_date THEN M.price * 20
           ELSE M.price * 10
         END
       ) AS Points
FROM dates D
JOIN sales S ON D.customer_id = S.customer_id
JOIN menu M ON M.product_id = S.product_id
WHERE S.order_date < D.last_date
GROUP BY S.customer_id;
```



customer_id	Points
B	720
A	1270

Total points for Customer A and B are 1,270 and 720 respectively.