TeeJay's Diner
SQL Data Analysis

Improving Customer Experience and Business Operations with Data Insights

- BHARATHIKANNAN | DATA ANALYST

INTRODUCTION

TeeJay 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**.

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

PROBLEM STATEMENT

Teejay 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.

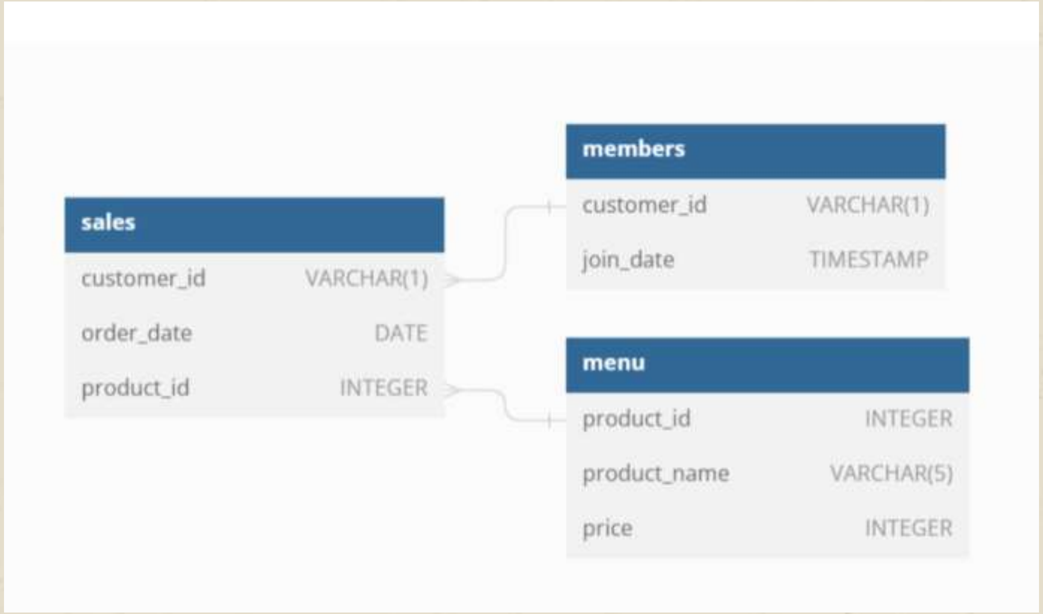
Teejay has shared with you 3 key datasets for this case study:

sales
Menu
members

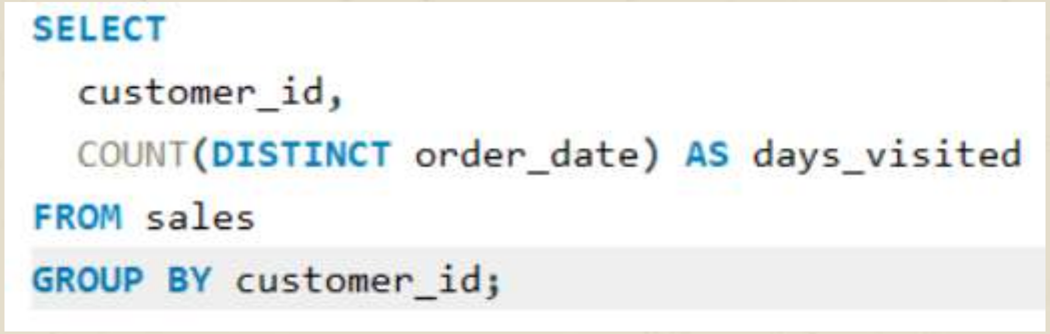
What Does TeeJay Want to Achieve?

- Understand customer visiting patterns
- Analyze customer spending habits
- Discover favorite menu items
- Expand the loyalty program based on insights

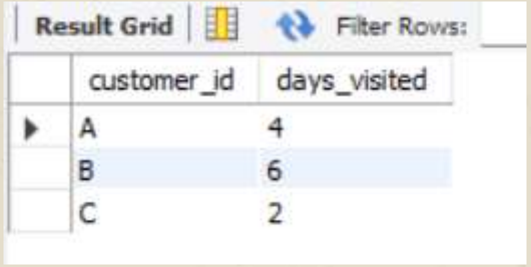
ENTITY RELATIONSHIP



How many days has each customer visited the restaurant?

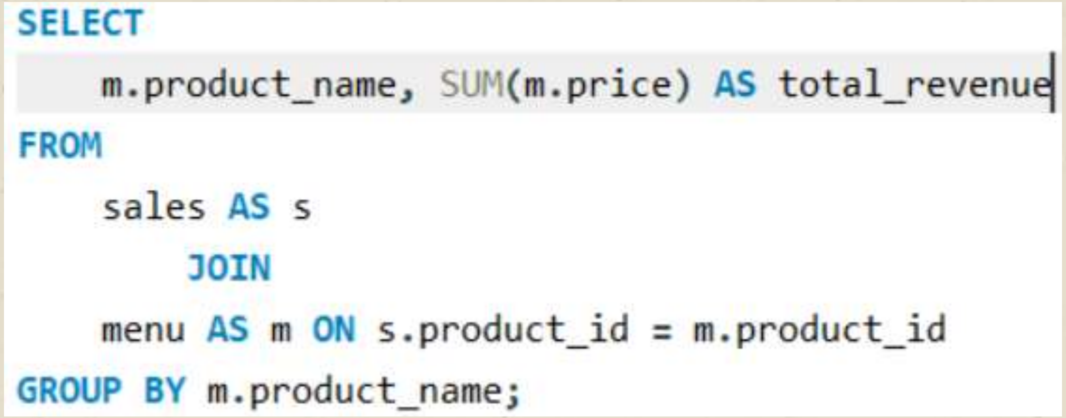


```
SELECT
  customer_id,
  COUNT(DISTINCT order_date) AS days_visited
FROM sales
GROUP BY customer_id;
```

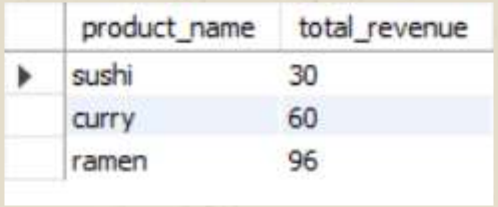


customer_id	days_visited
A	4
B	6
C	2

Get the total revenue generated for each product

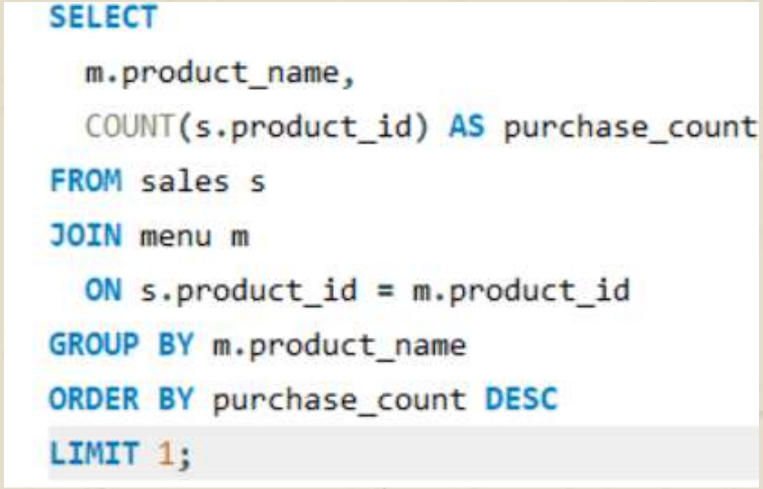


```
SELECT
    m.product_name, SUM(m.price) AS total_revenue
FROM
    sales AS s
        JOIN
    menu AS m ON s.product_id = m.product_id
GROUP BY m.product_name;
```

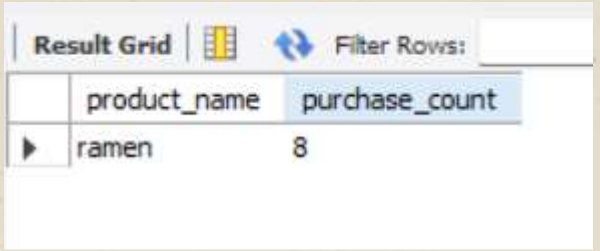


product_name	total_revenue
sushi	30
curry	60
ramen	96

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

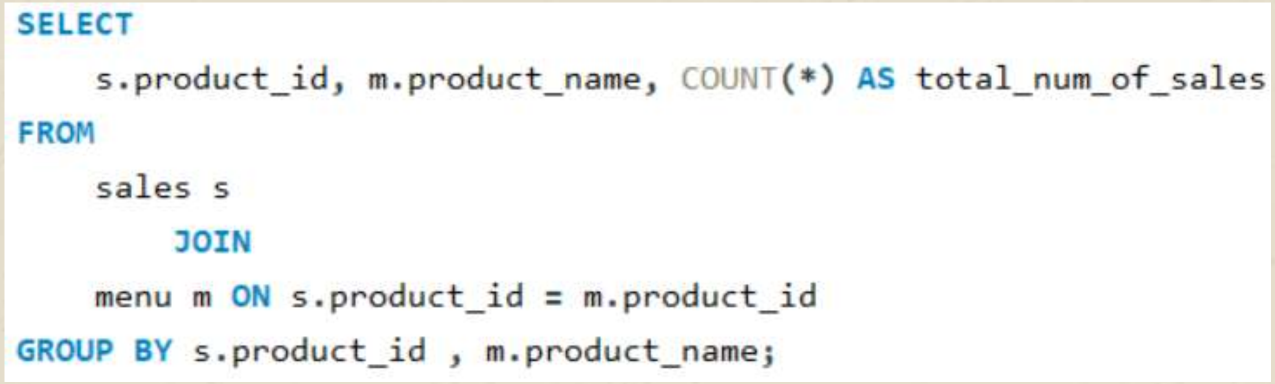


```
SELECT
  m.product_name,
  COUNT(s.product_id) AS purchase_count
FROM sales s
JOIN menu m
  ON s.product_id = m.product_id
GROUP BY m.product_name
ORDER BY purchase_count DESC
LIMIT 1;
```

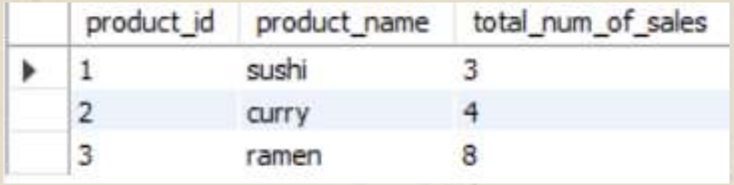


```
   product_name   purchase_count
▶  ramen          8
```

Find the total number of sales for each product

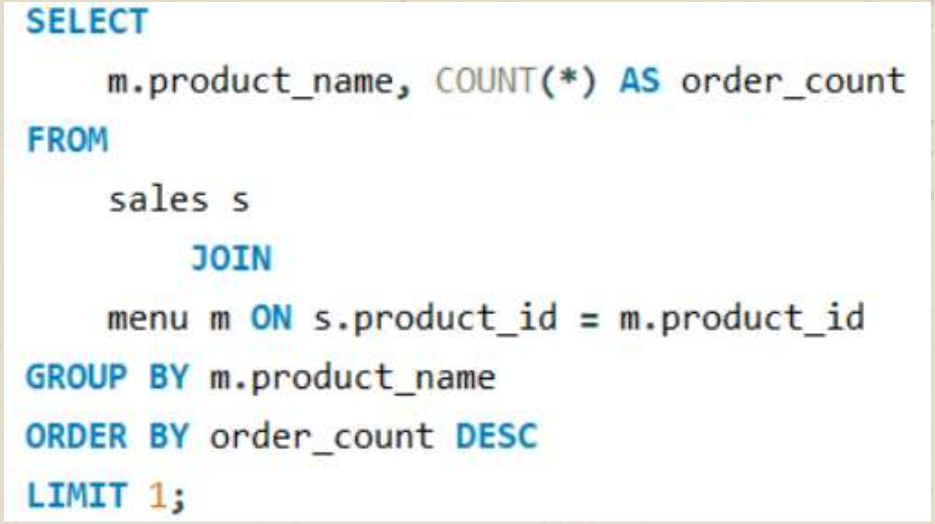


```
SELECT
    s.product_id, m.product_name, COUNT(*) AS total_num_of_sales
FROM
    sales s
        JOIN
    menu m ON s.product_id = m.product_id
GROUP BY s.product_id , m.product_name;
```

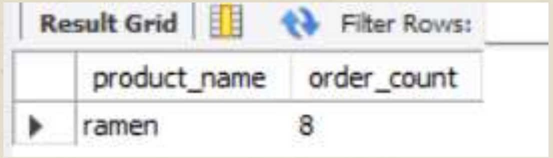


product_id	product_name	total_num_of_sales
1	sushi	3
2	curry	4
3	ramen	8

Find which product was ordered the most

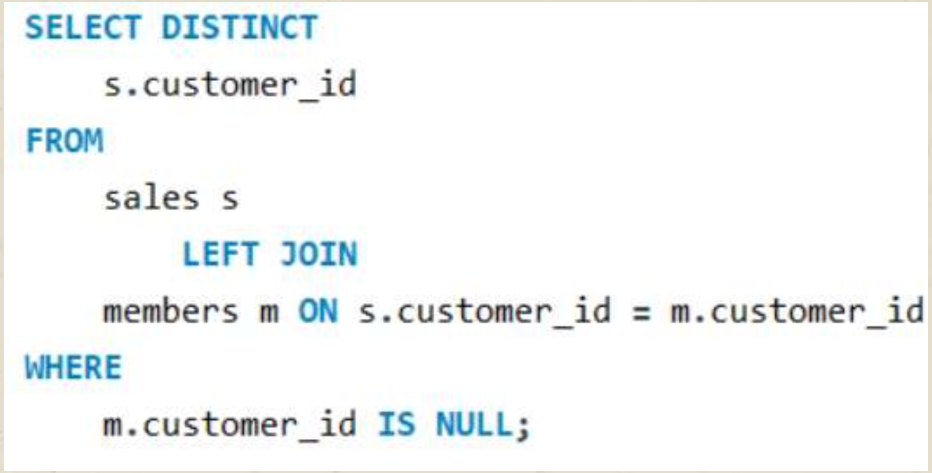


```
SELECT
    m.product_name, COUNT(*) AS order_count
FROM
    sales s
        JOIN
    menu m ON s.product_id = m.product_id
GROUP BY m.product_name
ORDER BY order_count DESC
LIMIT 1;
```

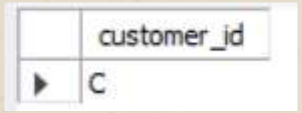


```
  product_name  order_count
▶ ramen         8
```

List all customers who have never joined the membership program

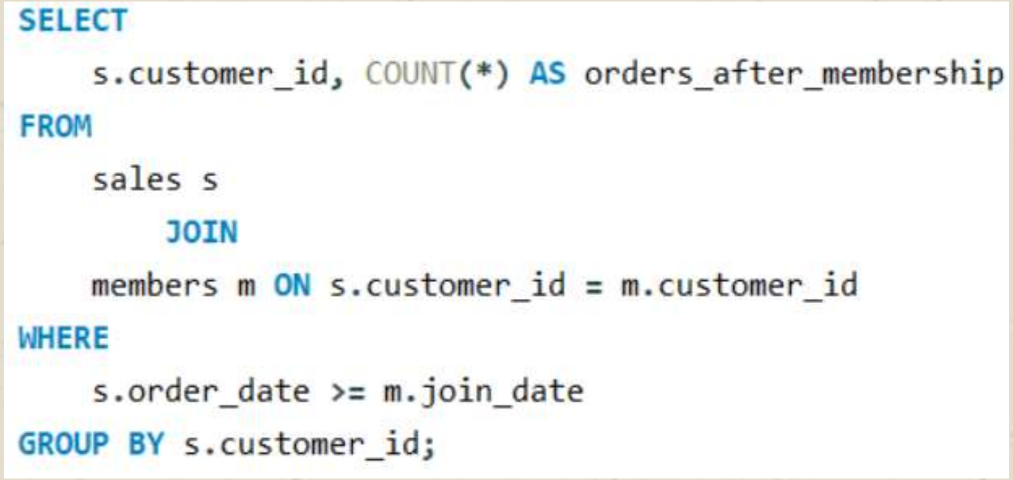


```
SELECT DISTINCT
    s.customer_id
FROM
    sales s
        LEFT JOIN
    members m ON s.customer_id = m.customer_id
WHERE
    m.customer_id IS NULL;
```

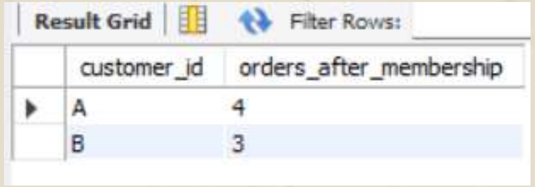


customer_id
C

Find the total number of orders made after a customer became a member

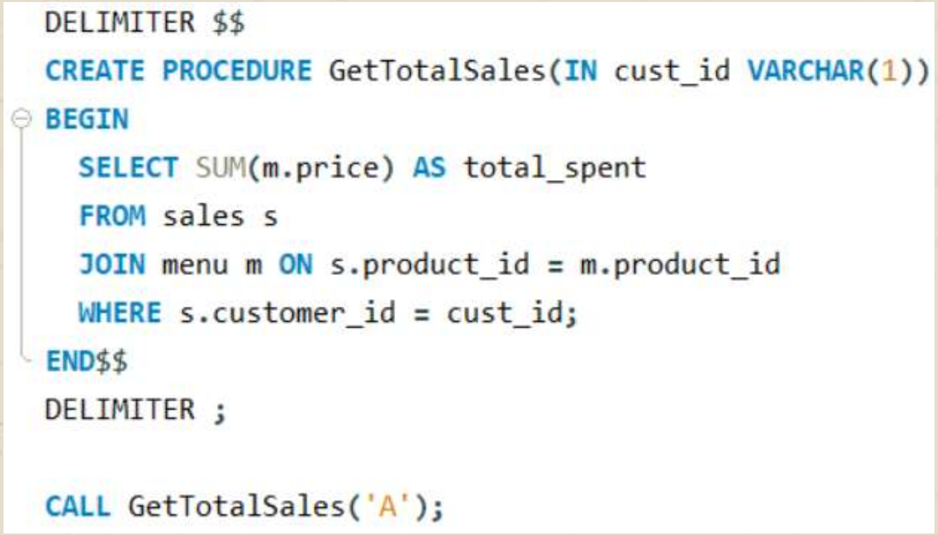


```
SELECT
    s.customer_id, COUNT(*) AS orders_after_membership
FROM
    sales s
        JOIN
    members m ON s.customer_id = m.customer_id
WHERE
    s.order_date >= m.join_date
GROUP BY s.customer_id;
```



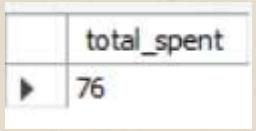
customer_id	orders_after_membership
A	4
B	3

Create a stored procedure that retrieves (total sales) for a given customer ID.



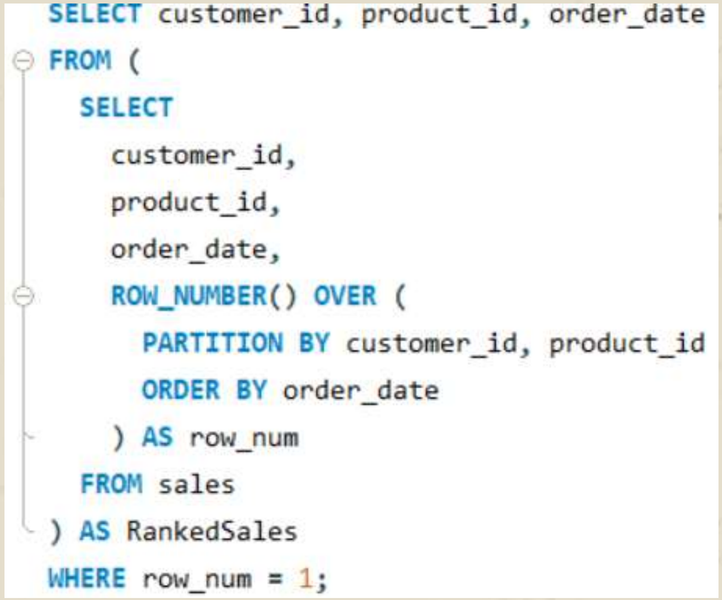
```
DELIMITER $$
CREATE PROCEDURE GetTotalSales(IN cust_id VARCHAR(1))
BEGIN
   SELECT SUM(m.price) AS total_spent
   FROM sales s
   JOIN menu m ON s.product_id = m.product_id
   WHERE s.customer_id = cust_id;
END$$
DELIMITER ;

CALL GetTotalSales('A');
```

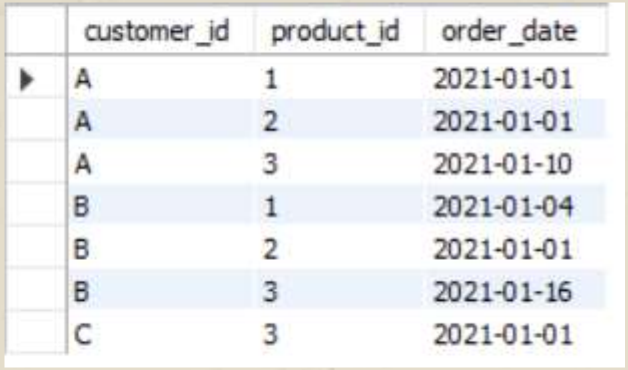


```
total_spent
76
```

Remove duplicate rows from the sales table based on customer_id, product_id, and order_date using a window function.

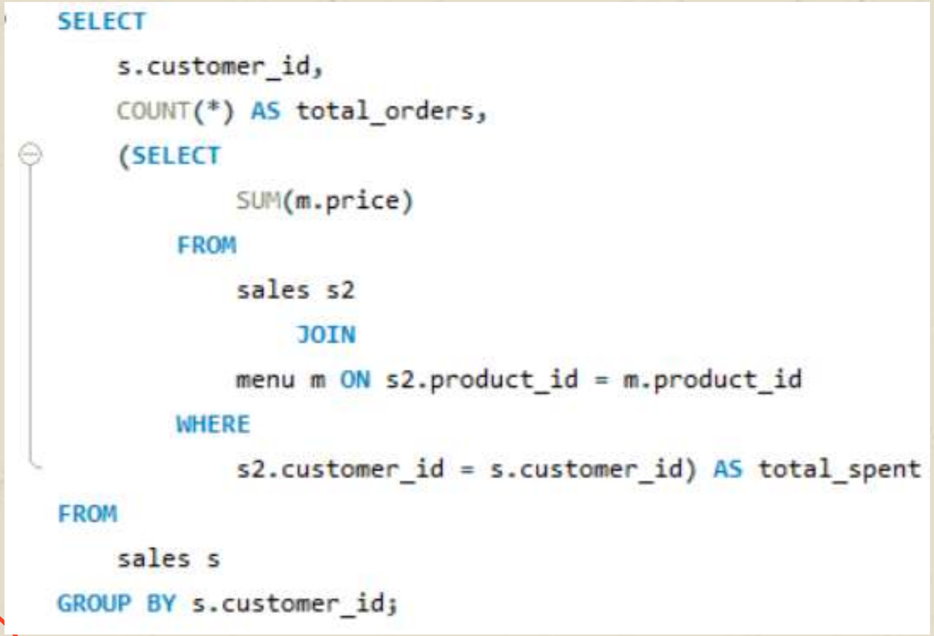


```
SELECT customer_id, product_id, order_date
FROM (
  SELECT
    customer_id,
    product_id,
    order_date,
    ROW_NUMBER() OVER (
      PARTITION BY customer_id, product_id
      ORDER BY order_date
    ) AS row_num
  FROM sales
) AS RankedSales
WHERE row_num = 1;
```

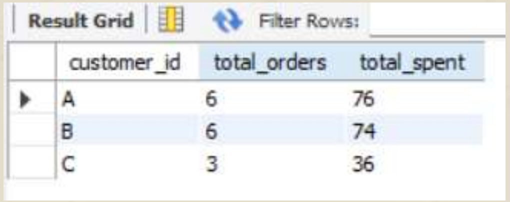


customer_id	product_id	order_date
A	1	2021-01-01
A	2	2021-01-01
A	3	2021-01-10
B	1	2021-01-04
B	2	2021-01-01
B	3	2021-01-16
C	3	2021-01-01

Find the total amount spent by each customer along with their total number of orders.

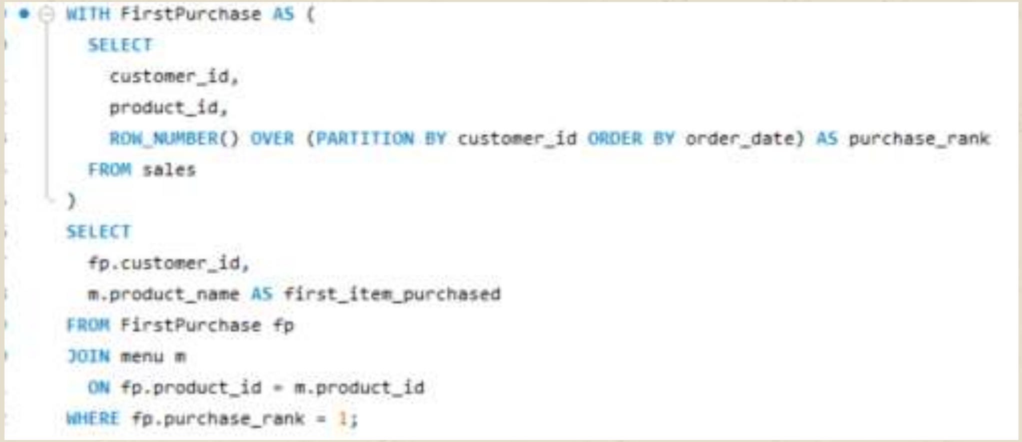


```
SELECT
    s.customer_id,
    COUNT(*) AS total_orders,
    (SELECT
            SUM(m.price)
        FROM
            sales s2
                JOIN
            menu m ON s2.product_id = m.product_id
        WHERE
            s2.customer_id = s.customer_id) AS total_spent
FROM
    sales s
GROUP BY s.customer_id;
```

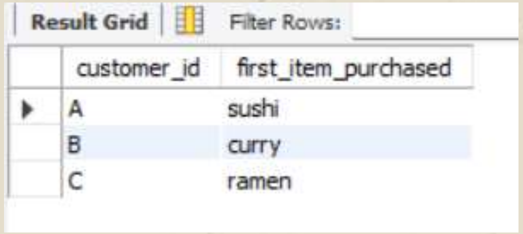


customer_id	total_orders	total_spent
A	6	76
B	6	74
C	3	36

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



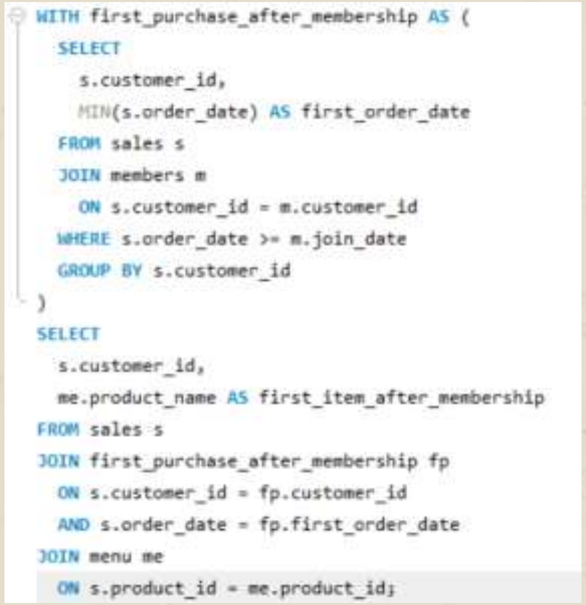
```
WITH FirstPurchase AS (
  SELECT
    customer_id,
    product_id,
    ROW_NUMBER() OVER (PARTITION BY customer_id ORDER BY order_date) AS purchase_rank
  FROM sales
)
SELECT
  fp.customer_id,
  m.product_name AS first_item_purchased
FROM FirstPurchase fp
JOIN menu m
  ON fp.product_id = m.product_id
WHERE fp.purchase_rank = 1;
```



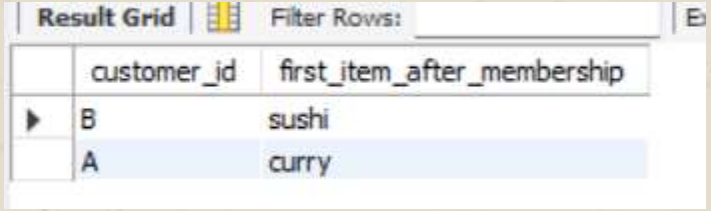
Result Grid | Filter Rows:

customer_id	first_item_purchased
A	sushi
B	curry
C	ramen

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



```
WITH first_purchase_after_membership AS (
  SELECT
    s.customer_id,
    MIN(s.order_date) AS first_order_date
  FROM sales s
  JOIN members m
    ON s.customer_id = m.customer_id
  WHERE s.order_date >= m.join_date
  GROUP BY s.customer_id
)
SELECT
  s.customer_id,
  me.product_name AS first_item_after_membership
FROM sales s
JOIN first_purchase_after_membership fp
  ON s.customer_id = fp.customer_id
  AND s.order_date = fp.first_order_date
JOIN menu me
  ON s.product_id = me.product_id;
```



customer_id	first_item_after_membership
B	sushi
A	curry

Thank You