```sql
CREATE DATABASE dannys_diner;

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');


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');


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');


USE dannys_diner;

SELECT * FROM sales;
SELECT * FROM menu;
SELECT * FROM members;


 /* --------------------
   Case Study Questions
   --------------------*/

-- 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 custome rs?
-- 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?

-------------------------------------Let's Start:-----------------------------------------

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

-- Select the customer ID and the total amount each customer spent
SELECT s.customer_id, SUM(m.price) as total_amount_each_customer_spent

-- Specify the tables we are retrieving data from and assign aliases
FROM sales as s
JOIN menu as m
ON s.product_id = m.product_id

-- Group the results by customer ID
GROUP BY s.customer_id;


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

-- Select the customer ID and count of order dates for each customer
SELECT customer_id, COUNT(order_date) as total_amount_each_customer_spent
FROM sales

-- Group the results by customer ID
GROUP BY customer_id;

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

```sql
SELECT customer_id, order_date, first_item
FROM (
  -- This is the inner subquery where we perform the calculations
  SELECT
    s.customer_id,
    ROW_NUMBER() OVER (PARTITION BY s.customer_id ORDER BY s.order_date ASC) as row_num, -- Assigns a row number to
       each row within each customer group based on the order date
    order_date,
    m.product_name as first_item
  FROM sales as s
  JOIN menu as m
  ON s.product_id = m.product_id
) subquery -- Alias for the subquery
WHERE row_num = 1; -- Selects only the rows with a row number of 1 (i.e., the first row within each  customer group)

-- 4. What is the most purchased item on the menu and how many times was it purchased by all custome rs?
SELECT TOP 1 m.product_name, COUNT(*) as purchase_count, s.product_id
FROM sales as s
JOIN menu as m ON s.product_id = m.product_id
GROUP BY s.product_id,m.product_name
ORDER BY purchase_count DESC;

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

SELECT customer_id, product_name, product_id, total_orders
FROM (
    -- Subquery to calculate the total orders for each customer and product combination and assign r ow numbers
    SELECT s.customer_id, m.product_name, s.product_id, COUNT(*) AS total_orders,
           ROW_NUMBER() OVER (PARTITION BY s.customer_id ORDER BY COUNT(*) DESC) AS rn
    FROM sales AS s
    JOIN menu AS m ON s.product_id = m.product_id
    GROUP BY s.customer_id, m.product_name, s.product_id
) AS t
WHERE rn = 1 -- Filter the rows where the row number is 1 (most popular item for each customer)
```

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

SELECT t.customer_id, t.first_purchase_date, m.product_name
FROM( select s.customer_id, MIN(s.order_date) as first_purchase_date
      from sales as s
      JOIN menu as m
      ON s.product_id = m.product_id
      JOIN members as mm
      ON mm.customer_id= s.customer_id
      Where s.order_date > mm.join_date
      GROUP BY s.customer_id
) as t
JOIN sales AS s ON t.customer_id = s.customer_id AND t.first_purchase_date = s.order_date
JOIN menu AS m ON s.product_id = m.product_id;

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

SELECT s.customer_id, MAX(s.order_date) AS last_purchase_date, s.product_id, m.product_name
FROM sales AS s
JOIN members AS mm ON s.customer_id = mm.customer_id
JOIN menu as m ON s.product_id= m.product_id
WHERE mm.join_date > s.order_date
GROUP BY s.customer_id, s.product_id,m.product_name;

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

SELECT s.customer_id, COUNT(s.product_id) AS total_items, SUM(m.price) AS total_amount_spent
FROM sales AS s
JOIN members AS mm ON s.customer_id = mm.customer_id
JOIN menu AS m ON s.product_id = m.product_id
WHERE s.order_date < mm.join_date
GROUP BY s.customer_id;

-- 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 m.product_name = 'sushi' THEN 2 * m.price ELSE m.price END) * 10 AS total_points
FROM sales AS s
JOIN menu AS m ON s.product_id = m.product_id
GROUP BY s.customer_id;


/*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
  sales.customer_id,
  SUM(CASE
    WHEN order_date <= DATEADD(DAY, 6, join_date) THEN menu.price * 2
    ELSE menu.price
  END) AS total_points
FROM sales
JOIN members ON sales.customer_id = members.customer_id
JOIN menu ON sales.product_id = menu.product_id
WHERE YEAR(order_date) = 2021 AND MONTH(order_date) = 1
GROUP BY sales.customer_id
```