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 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?

Question no. 1: What is the total amount each customer spent at the restaurant?

Query used:

```
SELECT customer_id AS customer
       , SUM(price) AS total_spent
FROM sales s
JOIN menu m
    ON s.product_id = m.product_id
GROUP BY customer_id
ORDER BY customer_id
```

Output:

	customer character varying (1)	total_spent bigint
1	A	76
2	B	74
3	C	36

Question no. 2: How many days has each customer visited the restaurant?

Query used:

```
SELECT customer_id AS customer
       , COUNT(DISTINCT(order_date)) AS total_visits
FROM sales
GROUP BY customer_id
ORDER BY customer_id
```

Output:

	customer character varying (1)	total_visits bigint
1	A	4
2	B	6
3	C	2

Question no. 3: What was the first item from the menu purchased by each customer?

Query used:

```
WITH rank_cte AS(
                SELECT s.customer_id
                        , m.product_name
                        , s.order_date
                        , DENSE_RANK() OVER(PARTITION BY s.customer_id
                                            ORDER BY s.order_date) AS purchase_rank
                FROM sales s
                JOIN menu m
                    ON s.product_id = m.product_id
                )

SELECT customer_id
        , product_name
FROM rank_cte
WHERE purchase_rank = 1
```

Output:

	customer_id character varying (1)	product_name character varying (5)
1	A	curry
2	A	sushi
3	B	curry
4	C	ramen
5	C	ramen

Question no. 4: What is the most purchased item on the menu and how many times was it purchased by all customers?

Query used:

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

Output:

	product_name character varying (5)	number_of_purchases bigint
1	ramen	8

Question no. 5: Which item was the most popular for each customer?

Query used:

```
WITH rank_cte AS (
                  SELECT s.customer_id AS customer_id
                         , m.product_name AS product_name
                         , COUNT(s.product_id) AS product_count
                         , DENSE_RANK () OVER (PARTITION BY s.customer_id
                                               ORDER BY COUNT(s.product_id) DESC) AS product_rank
                  FROM sales s
                  JOIN menu m
                      ON s.product_id = m.product_id
                  GROUP BY s.customer_id
                         , m.product_name
                         , s.product_id
                  )


SELECT customer_id
        , product_name
        , product_count
FROM rank_cte
WHERE product_rank = 1
```

Output:

	customer_id character varying (1)	product_name character varying (5)	product_count bigint
1	A	ramen	3
2	B	sushi	2
3	B	ramen	2
4	B	curry	2
5	C	ramen	3

Question no. 6: Which item was purchased first by the customer after they became a member?

Query used:

```
WITH rank_cte AS (
                  SELECT s.customer_id AS customer_id
                        , m.product_name AS product_name
                        , s.order_date AS order_date
                        , mem.join_date AS join_date
                        , DENSE_RANK () 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 mem
                      ON s.customer_id = mem.customer_id
                      WHERE s.order_date >= mem.join_date
                 )


SELECT customer_id
        , product_name
        , order_date
        , join_date
FROM rank_cte
WHERE rnk = 1
```

Output:

	customer_id character varying (1)	product_name character varying (5)	order_date date	join_date date
1	A	curry	2021-01-07	2021-01-07
2	B	sushi	2021-01-11	2021-01-09

Question no. 7: Which item was purchased just before the customer became a member?

Query used:

```
WITH rank_cte AS (
                  SELECT s.customer_id AS customer_id
                        , m.product_name AS product_name
                        , s.order_date AS order_date
                        , mem.join_date AS join_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 mem
                      ON s.customer_id = mem.customer_id
                      WHERE s.order_date < mem.join_date
                 )


SELECT customer_id
        , product_name
        , order_date
        , join_date
FROM rank_cte
WHERE rnk = 1
```

Output:

	customer_id character varying (1)	product_name character varying (5)	order_date date	join_date date
1	A	sushi	2021-01-01	2021-01-07
2	A	curry	2021-01-01	2021-01-07
3	B	sushi	2021-01-04	2021-01-09

Question no. 8: What is the total items and amount spent for each member before they became a member?

Query used:

```
SELECT s.customer_id
        , COUNT(s.product_id) AS items_purchased
        , SUM(m.price) AS total_spent
FROM sales s
JOIN menu m
    ON s.product_id = m.product_id
JOIN members mem
    ON s.customer_id = mem.customer_id
    WHERE s.order_date < mem.join_date
GROUP BY s.customer_id
ORDER BY s.customer_id
```

Output:

	customer_id character varying (1)	items_purchased bigint	total_spent bigint
1	A	2	25
2	B	3	40

Question no. 9: If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?

Query used:

```
SELECT s.customer_id
        , SUM(CASE
                WHEN m.product_id = 1 THEN price * 20
                ELSE price * 10
            END) AS total_points
FROM sales s
JOIN menu m
    ON s.product_id = m.product_id
GROUP BY s.customer_id
ORDER BY s.customer_id
```

Output:

	customer_id character varying (1)	total_points bigint
1	A	860
2	B	940
3	C	360

Question no. 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?

Query used:

```
WITH date_cte AS (
                  SELECT *
                          , join_date + 6 AS date_interval
                          , TO_DATE('2021-01-31', 'YYYY-MM-DD') AS last_date
                  FROM members
                  )

SELECT s.customer_id
        , SUM(CASE
                  WHEN m.product_id = 1 THEN m.price * 20
                  WHEN s.order_date BETWEEN cte.join_date AND date_interval THEN m.price * 20
                  ELSE m.price * 10
              END) AS total_points
FROM sales s
JOIN menu m
    ON s.product_id = m.product_id
JOIN date_cte cte
    ON s.customer_id = cte.customer_id
    WHERE s.order_date < last_date
GROUP BY s.customer_id
ORDER BY s.customer_id
```

Output:

	customer_id character varying (1)	total_points bigint
1	A	1370
2	B	820