Case Study #1: Danny's Diner

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

```
SELECT
    customer_id,
    SUM(price)
FROM dannys_diner.sales
JOIN dannys_diner.menu
        ON dannys_diner.sales.product_id = dannys_diner.menu.pro
GROUP BY customer_id
ORDER BY customer_id ASC
```

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

```
SELECT
    customer_id,
    count(DISTINCT (order_date))
FROM dannys_diner.sales
GROUP BY customer_id
```

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

```
WITH first_order_date AS (
  SELECT
    customer_id,
    MIN(order_date) AS first_order_date
  FROM dannys_diner.sales
  GROUP BY customer_id
)


SELECT
    dannys_diner.sales.customer_id,
```

```
    product_name
FROM dannys_diner.sales
JOIN dannys_diner.menu on dannys_diner.sales.product_id = dannys
JOIN first_order_date on first_order_date.customer_id = dannys_d
    first_order_date.first_order_date = dannys_diner.sales.order
GROUP BY sales.customer_id, product_name
ORDER BY sales.customer_id;
```

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

```
SELECT
    product_name,
    COUNT(sales.product_id) AS purchased_time
FROM dannys_diner.sales AS sales
JOIN dannys_diner.menu AS menu ON menu.product_id = sales.produc
GROUP BY product_name
HAVING
    COUNT(sales.product_id) = (
        SELECT MAX(purchase_count)
        FROM (
            SELECT
                COUNT(product_id) AS purchase_count
            FROM
                dannys_diner.sales
            GROUP BY
                product_id
        ) AS max_count
    )
```

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

```
WITH cte_ranking AS (
    SELECT
```

```
        customer_id,
        product_name,
        count(m.product_id) as order_count,
        DENSE_RANK() OVER(PARTITION BY customer_id ORDER BY cou
    FROM sales AS s
    JOIN menu AS m ON s.product_id = m.product_id
    GROUP BY customer_id, product_name, m.product_id
)

SELECT
    customer_id,
    product_name,
    order_count
FROM cte_ranking
WHERE ranking = 1
```

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

```
WITH cte_ranking AS (
    SELECT
        s.customer_id,
        product_name, order_date,
        join_date,
        DENSE_RANK() over(PARTITION BY s.customer_id order by o
    FROM sales AS s
    JOIN menu AS m ON m.product_id = s.product_id
    JOIN members as mb ON mb.customer_id = s.customer_id
    WHERE s.order_date > mb.join_date
)

SELECT
    customer_id,
    product_name
```

```
FROM cte_ranking
WHERE ranking = 1
```

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

```
WITH cte_ranking AS (
    SELECT
        s.customer_id,
    m.product_id,
        product_name, order_date,
        join_date,
        ROW_NUMBER() over(PARTITION BY s.customer_id order by or
    FROM sales AS s
    JOIN menu AS m ON m.product_id = s.product_id
    JOIN members as mb ON mb.customer_id = s.customer_id
    WHERE s.order_date < mb.join_date
)

SELECT
    customer_id,
    product_name
FROM cte_ranking
WHERE ranking = 1
```

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

```
SELECT
        s.customer_id,
    count(m.product_name) as total_items,
        sum(price) as total_sales
    FROM sales AS s
    JOIN menu AS m ON m.product_id = s.product_id
    JOIN members as mb ON mb.customer_id = s.customer_id
```

```
    WHERE s.order_date < mb.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_id=1 THEN price*20
                ELSE price * 10 END) AS point
    FROM sales AS s
    JOIN menu AS m ON m.product_id = s.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
        s.customer_id,
        SUM(CASE
                WHEN s.order_date BETWEEN mb.join_date AND (mb.j
                WHEN m.product_name='sushi' THEN 2*10*price
                ELSE 10*price END) AS point
    FROM sales AS s
    JOIN menu AS m ON m.product_id = s.product_id
  JOIN members as mb ON mb.customer_id = s.customer_id
    WHERE s.order_date < '2021-01-31' and s.order_date >= mb.joi
    GROUP BY s.customer_id
```