

```
-- 8 Week SQL Challenge

-- Case Study #1 - Danny's Dinner

/*

Few things to know before starting the Week 1 Challenge

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 favourite. Having this
deeper connection with his customers will help him deliver a better and more personalised 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.

*/

-- Danny has shared with you 3 key datasets for this case study:
-- 1. Sales

SELECT * FROM sales;

/* Output

customer_id | order_date | product_id
-----
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

*/

-- 2. Menu

SELECT * FROM menu;

/* Output

product_id | product_name | price
-----
1 | sushi | 10
2 | curry | 15
3 | ramen | 12

*/

-- 3. Members

SELECT * FROM members;

/* Output

customer_id | join_date
-----
A | 2021-01-07
B | 2021-01-09

*/

-- Each of the following case study questions can be answered using a single SQL statement:

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

SELECT customer_id,
SUM(price) AS Amount_spent
FROM sales S
INNER JOIN menu M
ON S.product_id = M.Product_id
GROUP BY customer_id;

/* Output

customer_id | Amount_spent
-----
A | 76
B | 74
C | 36

*/

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

SELECT customer_id,
COUNT(DISTINCT DAY(order_date)) AS Days_visited
FROM sales
GROUP BY customer_id;

/* Output

customer_id | Days_visited
-----
A | 4
B | 5
C | 2

*/
```



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

SELECT customer_id,
       product_id,
       product_name
FROM (SELECT customer_id,
            order_date,
            S.product_id,
            product_name,
            row_number() OVER(partition by customer_id order by order_date) AS 'first_day'
      FROM sales S
     INNER JOIN menu M
              ON S.product_id = M.product_id) A
WHERE first_day = 1;

/* Output
customer_id | product_id | product_name
-----
A           | 1          | sushi
B           | 2          | curry
C           | 3          | ramen
*/

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

WITH cte_max_dish AS
  (SELECT product_id,
         COUNT(product_id) No_of_dishes
   FROM sales
  GROUP BY product_id)
SELECT M.product_id,
       product_name,
       no_of_dishes
FROM   cte_max_dish C
INNER JOIN menu M
      ON C.product_id = M.product_id
WHERE  No_of_dishes = (SELECT MAX(No_of_dishes) FROM cte_max_dish);

/* Output
product_id | product_name | no_of_dishes
-----
3          | ramen       | 8
*/

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

WITH cte_customerwise_max AS
  (SELECT customer_id,
         product_id,
         COUNT(product_id) no_of_dishes
   FROM sales
  GROUP BY customer_id, product_id)
SELECT customer_id,
       M.product_id,
       product_name
FROM   (SELECT customer_id,
            product_id,
            dense_rank() OVER(partition by customer_id ORDER BY no_of_dishes DESC) rank_of_dish
      FROM cte_customerwise_max) AS rank_table
INNER JOIN menu M
      ON rank_table.product_id = M.product_id
WHERE  rank_of_dish = 1;

/* Output
customer_id | product_id | product_name
-----
B           | 1          | sushi
B           | 2          | curry
A           | 3          | ramen
B           | 3          | ramen
C           | 3          | ramen
*/

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

SELECT customer_id,
       M.product_id,
       product_name
FROM   (SELECT S.customer_id,
            order_date,
            product_id,
            join_date,
            dense_rank() OVER(partition by S.customer_id ORDER BY order_date) order_date_rank
      FROM sales S
     INNER JOIN members M
              ON S.customer_id = M.customer_id
              WHERE order_date >= join_date) AS A
INNER JOIN menu M
      ON A.product_id = M.product_id
WHERE  order_date_rank = 1;

/* Output
customer_id | product_id | product_name
-----
B           | 1          | sushi
A           | 2          | curry
*/
```



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

SELECT customer_id,
       M.product_id,
       product_name
FROM (SELECT S.customer_id,
            order_date,
            product_id,
            join_date,dense_rank() OVER(partition by S.customer_id ORDER BY order_date DESC)
            AS order_date_rank
      FROM sales S
     INNER JOIN members M
            ON S.customer_id = M.customer_id
            WHERE order_date < join_date) AS C
INNER JOIN menu M
      ON C.product_id = M.product_id
WHERE order_date_rank = 1;

/* Output
customer_id | product_id | product_name
-----
A           |           1 | sushi
B           |           1 | sushi
A           |           2 | curry
*/

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

WITH cte_test3 AS
  (SELECT S.customer_id,
        order_date,
        product_id,
        join_date
    FROM sales S
   INNER JOIN members M
        ON S.customer_id = M.customer_id
        WHERE order_date < join_date)
SELECT customer_id,
       COUNT(C.product_id) AS Total_items,
       SUM(price) AS Amount_spent
FROM cte_test3 C
INNER JOIN menu M
      ON C.product_id = M.product_id
GROUP BY customer_id;

/* Output
customer_id | Total_items | Amount_spent
-----
B           |           3 |         40
A           |           2 |         25
*/

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

SELECT customer_id,
       SUM(CASE
           WHEN product_name = 'Sushi' THEN 20*price
           ELSE 10*price
           END) AS points
FROM sales S
INNER JOIN menu M
      ON S.product_id = M.product_id
GROUP BY customer_id;

/* Output
customer_id | points
-----
A           |    860
B           |    940
C           |   360
*/

-- 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 product_name = 'Sushi' OR order_date BETWEEN join_date AND
            date_add(join_date,interval 7 day) THEN 20*price
           ELSE 10*price
           END) AS points
FROM sales S
INNER JOIN menu M
      ON S.product_id = M.product_id
INNER JOIN members M1
      ON S.customer_id = M1.customer_id
WHERE MONTH(order_date) = 1
GROUP BY customer_id;

/* Output
customer_id | points
-----
B           |    940
A           |   1370
*/
```