Case Study #1 - Danny's Diner

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

```
SELECT
        s.customer_id ,
        SUM( m.price ) AS total_amount
FROM sales s
JOIN
  menu m
ON
 s.product_id = m.product_id
GROUP BY 1 ;
```

customer_id	total_amount
A	76
B	74
C	36

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

```
SELECT
        customer_id ,
        COUNT( DISTINCT order_date ) AS visited_days
 FROM sales
GROUP BY 1 ;
```

customer_id	visited_days
A	4
B	6
C	2

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

```
 WITH result AS
 (
  SELECT
     s.customer_id ,
     s.order_date ,
     m.product_name ,
    RANK() OVER (PARTITION BY customer_id ORDER BY order_date ASC ) AS num_purchase
  FROM sales s
  LEFT JOIN
     menu m
  ON
    s.product_id = m.product_id
)
 SELECT
   DISTINCT customer_id ,
    order_date ,
    product_name
FROM result
WHERE num_purchase = 1 ;
```

customer_id	order_date	product_name
A	2021-01-01	sushi
A	2021-01-01	curry
B	2021-01-01	curry
C	2021-01-01	ramen

4- 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 num_purchased
FROM  menu m
JOIN
  sales s
ON
   m.product_id = s.product_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1 ;
```

	product_name	num_purchased
▶	ramen	8

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

```
 WITH semi_result AS
(
 SELECT s.customer_id , m.product_name , COUNT(s.order_date) AS num_purchase
FROM sales s
JOIN
menu m
ON
s.product_id = m.product_id
GROUP BY 1,2
) ,
 result AS
 (
 SELECT
       * ,
    RANK() OVER (PARTITION BY customer_id ORDER BY num_purchase desc) AS num
FROM semi_result
)
SELECT customer_id , product_name
 FROM result
 WHERE num = 1 ;
```

	customer_id	product_name
▶	A	ramen
	B	curry
	B	sushi
	B	ramen
	C	ramen

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

```
WITH result AS
(
SELECT
      s.customer_id ,
      s.order_date ,
      mem.join_date ,
      s.product_id,
      RANK() OVER (PARTITION BY customer_id ORDER BY order_date) AS ranks
FROM sales s
JOIN
members mem
ON
s.customer_id  = mem.customer_id
where s.order_date >= mem.join_date
)
SELECT
     r.customer_id ,
     m.product_name ,
     r.order_date
FROM result r
JOIN
menu m
ON
r.product_id = m.product_id
WHERE ranks = 1
ORDER BY 1 ;
```

	customer_id	product_name	order_date
▶	A	curry	2021-01-07
	B	sushi	2021-01-11

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

```
WITH result AS
(
SELECT
        s.customer_id ,
         s.order_date ,
         mem.join_date ,
         m.product_name ,
         RANK() OVER (PARTITION BY customer_id ORDER BY s.order_date DESC) AS num
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
FROM result
WHERE num = 1 ;
```

customer_id	product_name	order_date
A	sushi	2021-01-01
A	curry	2021-01-01
B	sushi	2021-01-04

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) total_amount
FROM sales s
JOIN
 menu m
ON
s.product_id = m.product_id
RIGHT JOIN
   members mem
ON
 s.customer_id = mem.customer_id
WHERE
       s.order_date < mem.join_date
GROUP BY 1
ORDER BY 1  ;
```

	customer_id	total_items	total_amount
▶	A	2	25
	B	3	40

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

```
WITH result AS
(
SELECT
        s.customer_id ,
        s.product_id ,
        m.product_name ,
        m.price ,
       CASE WHEN m.product_name = 'sushi'  THEN m.price*2*10 ELSE m.price*10 END  AS points
FROM sales s
JOIN
menu m
ON
s.product_id = m.product_id
)
SELECT
    customer_id ,
    SUM(points) as total_points
FROM result
GROUP BY 1 ;
```

customer_id	total_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?

```
 WITH result AS
(
 SELECT
       s.customer_id ,
       s.order_date ,
       mem.join_date ,
       m.product_name ,
        m.price  ,
       CASE
          WHEN   m.product_name = 'sushi'  THEN  m.price*2*10
           WHEN   s.order_date >= mem.join_date
           AND
               s.order_date < date_add(mem.join_date , interval 1 WEEK) THEN m.price*2*10
           ELSE m.price*10 END AS point
FROM sales s
JOIN
menu m
ON
s.product_id = m.product_id
JOIN
members mem
ON
s.customer_id = mem.customer_id
s.order_date <= '2021-01-31'
)
SELECT
     customer_id ,
     SUM(point) as total_points
FROM result
GROUP BY 1
ORDER BY 1;
```

	customer_id	total_points
▶	A	1370
	B	820

BONUS QUESTIONS

Join All The Table - Recreate the table : customer_id,order_date ,product_name, price,member (Y / N)

```
SELECT
        s.customer_id ,
        s.order_date ,
        m.product_name ,
        m.price ,
        CASE
            WHEN mem.join_date is null THEN 'N'
            WHEN mem.join_date is not null AND s.order_date < mem.join_date THEN 'N'
            ELSE 'Y' END AS member
FROM sales s
 LEFT JOIN
       menu m
 ON
  s.product_id = m.product_id
 LEFT JOIN
   members mem
 ON
   s.customer_id = mem.customer_id
 ORDER BY 1 ;
```

customer_id	order_date	product_name	price	member
A	2021-01-01	sushi	10	N
A	2021-01-01	curry	15	N
A	2021-01-07	curry	15	Y
A	2021-01-10	ramen	12	Y
A	2021-01-11	ramen	12	Y
A	2021-01-11	ramen	12	Y
B	2021-01-01	curry	15	N
B	2021-01-02	curry	15	N
B	2021-01-04	sushi	10	N
B	2021-01-11	sushi	10	Y
B	2021-01-16	ramen	12	Y
B	2021-02-01	ramen	12	Y
C	2021-01-01	ramen	12	N
C	2021-01-01	ramen	12	N
C	2021-01-07	ramen	12	N

Rank All The Things

Danny also requires further information about the ranking of customer products, but he purposely does not need the ranking for non-member purchases so he expects null ranking values for the records when customers are not yet part of the loyalty program.

```
WITH result AS
(
    SELECT
        s.customer_id ,
        s.order_date ,
        m.product_name ,
        m.price ,
```

```
        CASE
            WHEN mem.join_date is null THEN 'N'
            WHEN mem.join_date is not null AND s.order_date < mem.join_date THEN 'N'
            ELSE 'Y' END AS member
 FROM sales s
  LEFT JOIN
        menu m
  ON
    s.product_id = m.product_id
 LEFT JOIN
    members mem
 ON
    s.customer_id = mem.customer_id
 ORDER BY 1 ;
)
SELECT
        * ,
    CASE
    WHEN member = 'Y' THEN RANK() OVER (PARTITION BY customer_id ,member ORDER BY order_date)
    ELSE NULL END AS ranking
FROM result ;
```

customer_id	order_date	product_name	price	member	ranking
A	2021-01-01	sushi	10	N	NULL
A	2021-01-01	curry	15	N	NULL
A	2021-01-07	curry	15	Y	1
A	2021-01-10	ramen	12	Y	2
A	2021-01-11	ramen	12	Y	3
A	2021-01-11	ramen	12	Y	3
B	2021-01-01	curry	15	N	NULL
B	2021-01-02	curry	15	N	NULL
B	2021-01-04	sushi	10	N	NULL
B	2021-01-11	sushi	10	Y	1
B	2021-01-16	ramen	12	Y	2
B	2021-02-01	ramen	12	Y	3
C	2021-01-01	ramen	12	N	NULL
C	2021-01-01	ramen	12	N	NULL
C	2021-01-07	ramen	12	N	NULL

Name - Sudhanshu Tomar

Linkedin - https://www.linkedin.com/in/sudhanshu-tomar-493633255/

Source - https://8weeksqlchallenge.com/case-study-1/