

**8WEEKSQLCHALLENGE.COM**  
**CASE STUDY #1**



**THE TASTE OF SUCCESS**

**DATAWITHDANNY.COM**

## The schema query for MySQL.

```
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');
```

## The tables

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

sales

menus

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

members

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

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

```
SELECT customer_id,  
SUM(price) AS money_spent  
FROM sales  
JOIN menu  
ON menu.product_id = sales.product_id  
GROUP BY customer_id
```

	customer_id	money_spent
1	A	76
2	B	74
3	C	36

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

```
SELECT customer_id,  
COUNT(DISTINCT(order_date)) AS number_of_visits  
FROM sales  
GROUP BY customer_id
```

	customer_id	number_of_visits
1	A	4
2	B	6
3	C	2

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

```
SELECT DISTINCT(customer_id),  
product_name FROM sales s  
JOIN menu m  
ON m.product_id = s.product_id  
WHERE s.order_date = ANY  
(  
SELECT MIN(order_date)  
FROM sales  
GROUP BY customer_id  
)
```

WHERE column\_name comparison\_operator ANY (subquery)

	customer_id	product_name
1	A	curry
2	A	sushi
3	B	curry
4	C	ramen

---

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

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

	count	product_name
1	8	ramen

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

```
WITH r AS
(
SELECT s.customer_id,
m.product_name,
COUNT(s.product_id) as count,
DENSE_RANK() OVER (PARTITION BY s.customer_id ORDER BY
COUNT(s.product_id) DESC) AS r
FROM menu m
JOIN sales s
ON s.product_id = m.product_id
GROUP BY s.customer_id, s.product_id, m.product_name
)
SELECT customer_id, product_name, count
FROM r
WHERE r = 1
```

**DENSE\_RANK() OVER (PARTITION BY expr1[{,expr2...}]**

**ORDER BY expr1 [ASC|DESC], [{,expr2...}]**

	customer_id	product_name
1	A	curry
2	A	sushi
3	B	curry
4	C	ramen

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

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

**DENSE\_RANK() OVER (PARTITION BY expr1[{,expr2...}]**  
**ORDER BY expr1 [ASC|DESC], [{,expr2...}]**

	customer_id	product_name	ranks
1	A	curry	1
2	B	sushi	1

---

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

```
WITH ranks AS
(
  SELECT s.customer_id,
  s.order_date,
  m.product_name,
  DENSE_RANK() OVER (PARTITION BY s.customer_id ORDER BY s.order_date) AS
  ranks, mem.join_date
  FROM sales s
  JOIN menu m ON s.product_id = m.product_id
  JOIN members AS mem
  ON mem.customer_id = s.customer_id
  WHERE s.order_date < mem.join_date
)
SELECT * FROM ranks
WHERE ranks = 1
```

**DENSE\_RANK() OVER (PARTITION BY expr1[{,expr2...}]**  
**ORDER BY expr1 [ASC|DESC], [{,expr2...}]**

	customer_id	order_date	product_name	ranks	join_date
1	A	2021-01-01	sushi	1	2021-01-07
2	A	2021-01-01	curry	1	2021-01-07
3	B	2021-01-01	curry	1	2021-01-09

## 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(price) AS money_spent  
FROM sales AS s  
JOIN menu AS m  
ON m.product_id = s.product_id  
JOIN members AS mem  
ON s.customer_id = mem.customer_id  
WHERE s.order_date < mem.join_date  
GROUP BY s.customer_id
```

	customer_id	total_items	money_spent
1	A	2	25
2	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 points AS  
(  
  SELECT *,  
  CASE  
    WHEN m.product_name = 'sushi' THEN price * 20  
    WHEN m.product_name != 'sushi' THEN price * 10  
  END AS points  
FROM menu m  
)  
SELECT customer_id, SUM(points) AS points  
FROM sales s  
JOIN points p ON p.product_id = s.product_id  
GROUP BY s.customer_id
```

	customer_id	points
1	A	860
2	B	940
3	C	360

```
CASE expression  
  
WHEN when_expression_1 THEN  
    result_1  
  
WHEN when_expression_2 THEN  
    result_2  
  
WHEN when_expression_3 THEN  
    result_3  
  
...  
  
ELSE  
    else_result  
  
END
```



**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 customer_id,  
sum(points) points  
from (select s.customer_id,  
case  
  
when product_name = 'sushi' and  
s.order_date between dateadd(day,-1,ms.join_date) and dateadd(day, 6, ms.join_date)  
then m.price*40  
when product_name = 'sushi' or  
s.order_date between dateadd(day,-1,ms.join_date) and dateadd(day, 6, ms.join_date)  
then m.price*20  
else price*10 end points  
from members ms  
left join sales s on s.customer_id = ms.customer_id  
left join menu m on s.product_id = m.product_id  
where s.order_date <= '20210131') a  
group by customer_id;
```

	customer_id	points
1	A	1370
2	B	1020

## Bonus Questions

### 1. Recreate the table

```
SELECT s.customer_id,  
s.order_date,  
m.product_name,  
m.price,  
CASE  
WHEN s.order_date >= mem.join_date THEN 'Y'  
ELSE 'N'  
END AS member  
FROM sales s  
LEFT JOIN menu m ON m.product_id = s.product_id  
LEFT JOIN members mem ON mem.customer_id = s.customer_id  
ORDER BY customer_id, order_date, price DESC
```

	customer_id	order_date	product_name	price	member
1	A	2021-01-01	curry	15	N
2	A	2021-01-01	sushi	10	N
3	A	2021-01-07	curry	15	Y
4	A	2021-01-10	ramen	12	Y
5	A	2021-01-11	ramen	12	Y
6	A	2021-01-11	ramen	12	Y
7	B	2021-01-01	curry	15	N
8	B	2021-01-02	curry	15	N
9	B	2021-01-04	sushi	10	N
10	B	2021-01-11	sushi	10	Y
11	B	2021-01-16	ramen	12	Y
12	B	2021-02-01	ramen	12	Y
13	C	2021-01-01	ramen	12	N
14	C	2021-01-01	ramen	12	N
15	C	2021-01-07	ramen	12	N

## 2. Rank Members — fill non-members with null

```
select *,
       case when member = 'Y' then rank() over (partition by customer_id,
member order by order_date)
       else NULL end ranking
from (select s.customer_id,
            s.order_date,
            m.product_name,
            m.price,
            case when s.order_date >= ms.join_date then 'Y'
            else 'N' end member
      from sales s
      left join menu m on s.product_id = m.product_id
      left join members ms on s.customer_id = ms.customer_id) a
order by customer_id, order_date, product_name;
```

	customer_id	order_date	product_name	price	member	ranking
1	A	2021-01-01	curry	15	N	NULL
2	A	2021-01-01	sushi	10	N	NULL
3	A	2021-01-07	curry	15	Y	1
4	A	2021-01-10	ramen	12	Y	2
5	A	2021-01-11	ramen	12	Y	3
6	A	2021-01-11	ramen	12	Y	3
7	B	2021-01-01	curry	15	N	NULL
8	B	2021-01-02	curry	15	N	NULL
9	B	2021-01-04	sushi	10	N	NULL
10	B	2021-01-11	sushi	10	Y	1
11	B	2021-01-16	ramen	12	Y	2
12	B	2021-02-01	ramen	12	Y	3
13	C	2021-01-01	ramen	12	N	NULL
14	C	2021-01-01	ramen	12	N	NULL
15	C	2021-01-07	ramen	12	N	NULL