# SQL Case Study – Day 5 Assignment

## Amaan Shaikh

### Database Creation and Insertion

```
CREATE DATABASE Rams_Diner;

USE Rams_Diner;

CREATE TABLE sales (
   customer_id CHAR(1),
   order_date DATE,
   product_id INT
);

CREATE TABLE menu (
   product_id INT PRIMARY KEY,
   product_name VARCHAR(50),
   price DECIMAL(5, 2)
);

CREATE TABLE members (
   customer_id CHAR(1),
   join_date DATE
);

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);

INSERT INTO menu (product_id, product_name, price) VALUES

(1, 'sushi', 10),

(2, 'curry', 15),

(3, 'ramen', 12);

INSERT INTO members (customer_id, join_date) VALUES

('A', '2021-01-07'),

('B', '2021-01-09');

Qs 1-10:

```sql
--Q1 Amount spent by each customer

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

110 %

Results | Messages

| | customer_id | total_spend |
|---|---|---|
| 1 | A | 76.00 |
| 2 | B | 74.00 |
| 3 | C | 36.00 |

```sql
--2 Number of days each customer visited

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

110 %

Results | Messages

| | customer_id | visit_days |
|---|---|---|
| 1 | A | 4 |
| 2 | B | 6 |
| 3 | C | 2 |

```sql
--Q3 First Item Purchased by each Customer

SELECT s.customer_id, m.product_name
FROM sales s
JOIN menu m ON s.product_id = m.product_id
WHERE s.order_date = (
        SELECT MIN(order_date)
        FROM sales s2
        WHERE s2.customer_id = s.customer_id
                    );
```

110 %

Results | Messages

| | customer_id | product_name |
|---|---|---|
| 1 | A | sushi |
| 2 | A | curry |
| 3 | B | curry |
| 4 | C | ramen |
| 5 | C | ramen |

```sql
-- Q4 Most Purchased Item and Quantity

SELECT TOP 1 m.product_name, COUNT(*) AS purchase_count
FROM sales s
JOIN menu m ON s.product_id = m.product_id
GROUP BY m.product_name
ORDER BY purchase_count DESC;


--5 Most popular item by Customer
```

Results | Messages

| | product_name | purchase_count |
|---|---|---|
| | ramen | 8 |

```sql
--Q5 Most popular item by Customer

SELECT s.customer_id, m.product_name, COUNT(*) AS purchase_count
FROM sales s
JOIN menu m ON s.product_id = m.product_id
GROUP BY s.customer_id, m.product_name
HAVING COUNT(*) = (
        SELECT MAX(count)
        FROM (SELECT COUNT(*) AS count
              FROM sales s2
              WHERE s2.customer_id = s.customer_id
              GROUP BY s2.product_id) AS counts
                  );

--6 First Item Purchased After becoming a member

SELECT TOP 3 s.customer_id, m.product_name
```

110 %

Results | Messages

| | customer_id | product_name | purchase_count |
|---|---|---|---|
| 1 | B | curry | 2 |
| 2 | A | ramen | 3 |
| 3 | B | ramen | 2 |
| 4 | C | ramen | 3 |
| 5 | B | sushi | 2 |

SQL CASE STUDY....71Q2\Admin (77))*

```sql
--Q6 First Item Purchased After becoming a member

SELECT TOP 3 s.customer_id, m.product_name
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
ORDER BY s.order_date;

--7 Item purchased just before becoming a member
```

110 %

Results | Messages

| | customer_id | product_name |
|---|---|---|
| 1 | A | curry |
| 2 | A | ramen |
| 3 | A | ramen |

```sql
--Q7 Item purchased just before becoming a member

SELECT TOP 1 s.customer_id, m.product_name
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
ORDER BY s.order_date DESC;

--8 Total Items and Spend before becoming member
SELECT s.customer_id, COUNT(*) AS total_items, SUM(m
```

.0 %

Results | Messages

| | customer_id | product_name |
|---|---|---|
| 1 | B | sushi |

```sql
--Q8 Total Items and Spend before becoming member
SELECT s.customer_id, COUNT(*) AS total_items, 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;


--9 Points earned by each Customer
SELECT s.customer_id,
```

Results | Messages

| customer_id | total_items | total_spent |
|---|---|---|
| A | 2 | 25.00 |
| B | 3 | 40.00 |

```sql
--Q9 Points earned by each Customer
SELECT s.customer_id,
    SUM(
        CASE
            WHEN m.product_name = 'sushi' 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
GROUP BY s.customer_id;
```

Results | Messages

| | customer_id | total_points |
|---|---|---|
| 1 | A | 860.00 |
| 2 | B | 940.00 |
| 3 | C | 360.00 |

```sql
--Q10 Points earned by A & B in first week of membership, before the end of January
SELECT s.customer_id,
    SUM(
        CASE
            WHEN s.order_date BETWEEN mem.join_date AND DATEADD(day, 6, mem.join_date) THEN m.price * 20
            WHEN m.product_name = 'sushi' 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 members mem ON s.customer_id = mem.customer_id
WHERE s.order_date <= '2021-01-31'
GROUP BY s.customer_id
HAVING s.customer_id IN ('A', 'B');
```

110 %

Results | Messages

| | customer_id | total_points |
|---|---|---|
| 1 | A | 1370.00 |
| 2 | B | 820.00 |