Name: Sumit Chauhan

Department: Data Engineering

SQL 8 Week Challenge

Case Study #1

```
-- -- Initial SETUP

CREATE SCHEMA dannys_diner;
SET search_path = dannys_diner;

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');

-- SELECT
SELECT * FROM sales;
```

	customer_id character varying (1)	order_date date	product_id integer
1	A	2021-01-01	1
2	A	2021-01-01	2
3	A	2021-01-07	2
4	A	2021-01-10	3
5	A	2021-01-11	3
6	A	2021-01-11	3
7	B	2021-01-01	2
8	B	2021-01-02	2
9	B	2021-01-04	1
10	B	2021-01-11	1
11	B	2021-01-16	3
12	B	2021-02-01	3
13	C	2021-01-01	3
14	C	2021-01-01	3
15	C	2021-01-07	3

```
SELECT * FROM menu;
```

	product_id integer	product_name character varying (5)	price integer
1	1	sushi	10
2	2	curry	15
3	3	ramen	12

```
SELECT * FROM members;
```

	customer_id character varying (1)	join_date date
1	A	2021-01-07
2	B	2021-01-09

```
------------------- Case Study Questions -------------------

-- 1. What is the total amount each customer spent at the restaurant?
SELECT customer_id, sum(m.price) total_amount
FROM sales s JOIN menu m ON s.product_id=m.product_id
```

```
GROUP BY customer_id;
```

	customer_id character varying (1)	total_amount bigint
1	B	74
2	C	36
3	A	76

```
-- 2. How many days has each customer visited the restaurant?
SELECT customer_id, COUNT(distinct order_date) visited_days
FROM sales
GROUP BY customer_id;
```

	customer_id character varying (1)	visited_days bigint
1	A	4
2	B	6
3	C	2

```
-- 3. What was the first item from the menu purchased by each customer?
-- since the on the same first day,
-- customer can purchase multiple prodcuts
-- we are using array_agg to return array of products
SELECT customer_id, ARRAY_AGG(DISTINCT m.product_name)
FROM sales s
JOIN menu m ON s.product_id=m.product_id
WHERE order_date = (
  SELECT MIN(order_date) FROM sales WHERE s.customer_id=customer_id
)
GROUP BY customer_id;
```

	customer_id character varying (1)	array_agg character varying[]
1	A	{curry,sushi}
2	B	{curry}
3	C	{ramen}

```
-- 4. What is the most purchased item on the menu and how many times was it
purchased by all customers?
WITH freq AS (
  SELECT product_id, COUNT(*) freq
  FROM sales
```

```
  GROUP BY product_id
)
SELECT f.product_id,m.product_name, freq
FROM freq f
JOIN menu m ON f.product_id=m.product_id
WHERE freq = (
  SELECT max(freq) FROM freq
);
-- remarks change cte freq name
```

	product_id integer	product_name character varying (5)	freq bigint
1	3	ramen	8

```
-- 5. Which item was the most popular for each customer?
-- assumption: if two prodcuts are equally popular,
-- product with higher product_id wins
WITH cust_item_cnt AS (
  SELECT customer_id, product_id, count(*) cnt
  FROM sales s
  GROUP BY customer_id, product_id
)
SELECT DISTINCT customer_id,
FIRST_VALUE(product_id) OVER (PARTITION BY customer_id ORDER BY cnt DESC,
product_id DESC) product_id
FROM cust_item_cnt;
-- JOIN for product_name
```

	customer_id character varying (1)	product_id integer
1	B	3
2	A	3
3	C	3

```
-- 6. Which item was purchased first by the customer after they became a member?
-- assumption: customer can order on the same day when he joined
-- that's why si.order_date >= mi.join_date
-- assumption: customers who are never joined (not in members table)
-- are not included in this list
SELECT customer_id, ARRAY_AGG(DISTINCT m.product_name)
FROM sales s
JOIN menu m ON s.product_id=m.product_id
WHERE order_date = (
  SELECT MIN(order_date) FROM sales si
```

```
  JOIN members mi ON mi.customer_id=si.customer_id
  WHERE s.customer_id=si.customer_id AND si.order_date >= mi.join_date
)
GROUP BY customer_id;
-- alias si means sales inner table
-- alias mi means member inner table
-- remarks: OPTIMIZATION PENDING
```

	customer_id character varying (1)	array_agg character varying[]
1	A	{curry}
2	B	{sushi}

```
-- 7. Which item was purchased just before the customer became a member?
-- assumption: customers who are never joined (not in members table)
-- are not included in this list
SELECT customer_id, ARRAY_AGG(DISTINCT m.product_name)
FROM sales s
JOIN menu m ON s.product_id=m.product_id
WHERE order_date = (
  SELECT MAX(order_date) FROM sales si
  JOIN members mi ON mi.customer_id=si.customer_id
  WHERE s.customer_id=si.customer_id AND si.order_date < mi.join_date
)
GROUP BY customer_id;
```

	customer_id character varying (1)	array_agg character varying[]
1	A	{curry,sushi}
2	B	{sushi}

```
-- 8. What is the total items and amount spent for each member before they became
a member?
-- assumption: the items purchased on different day will be counted seperately
SELECT s.customer_id, SUM(m.price) amount, COUNT(*) total_items
FROM sales s
JOIN menu m ON s.product_id=m.product_id
JOIN members mbr ON s.customer_id=mbr.customer_id
WHERE s.order_date < mbr.join_date
GROUP BY s.customer_id;
```

	customer_id character varying (1)	amount bigint	total_items bigint
1	B	40	3
2	A	25	2

```
-- 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(m.price *
  CASE
    WHEN s.product_id=1 THEN 20
    ELSE 10
  END
) points
FROM sales s
JOIN menu m ON s.product_id=m.product_id
GROUP BY s.customer_id;
```

	customer_id character varying (1)	points bigint
1	B	940
2	C	360
3	A	860

```
-- 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?

-- assumption: the question asking for junuary-2021
-- assumption: the question asking about all the orders till january-2021
-- and not only in the january month
SELECT s.customer_id, SUM(m.price *
  CASE
    WHEN s.order_date BETWEEN mbr.join_date AND mbr.join_date + INTERVAL '6 DAY'
      THEN 20
    WHEN s.product_id=1 THEN 20
    ELSE 10
  END
) points
FROM sales s
JOIN menu m ON s.product_id=m.product_id
JOIN members mbr ON s.customer_id=mbr.customer_id
```

```
WHERE order_date < DATE '2021-02-01'
GROUP BY s.customer_id;
```

	customer_id character varying (1)	points bigint
1	A	1370
2	B	820