Danny's Dinner
The Taste of Success
Case Study #1

8 Week SQL Challenge

Nitesh Kumar Prajapati

Contents

- About Us
- Problem Statement
- Datasets
- Entity Relationship Diagram
- Insights and Questions
- Conclusion

About Us

Danny seriously loves Japanese food so in the beginning of 2021, he decides to embark upon a risky venture and opens up a cute little restaurant that sells his 3 favourite foods: sushi, curry, and ramen.

Danny’s Diner is in need of your assistance to help the restaurant stay afloat - the restaurant has captured some very basic data from their few months of operation but have no idea how to use their data to help them run the business.

• Problem Statement

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 provided you with a sample of his overall customer data due to privacy issues - but he hopes that these examples are enough for you to write fully functioning SQL queries to help him answer his questions!

• Datasets

Danny has shared with you 3 key datasets for this case study:

1. sales
2. menu
3. members

Menu Table

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

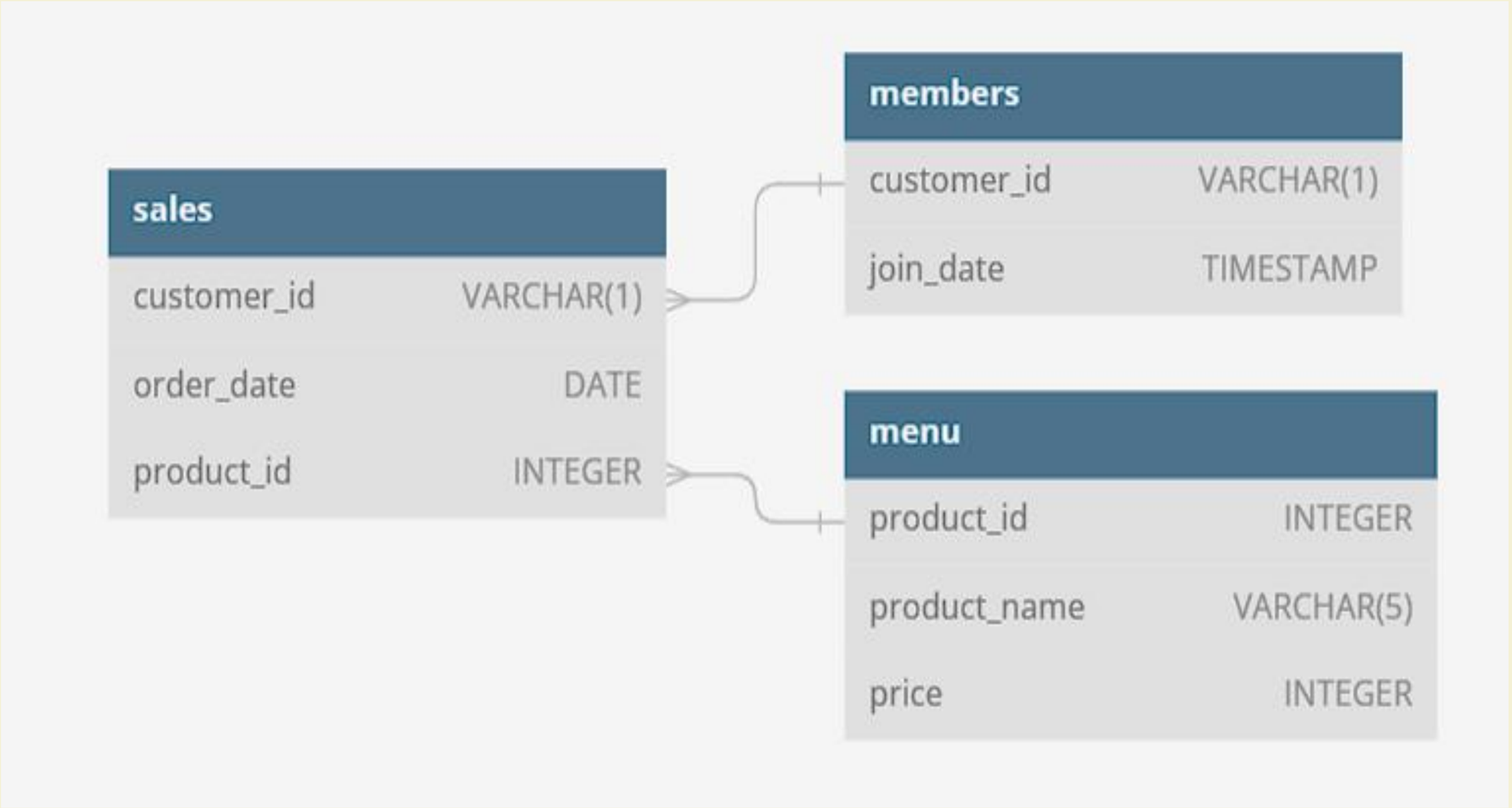
Members Table

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

Sales Table

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

• Entity Relationship Diagram



Insights and Questions

1.What is the total amount each customer spent at the restaurant?
2.How many days has each customer visited the restaurant?
3.What was the first item from the menu purchased by each customer?
4.What is the most purchased item on the menu and how many times was it purchased by all customers?
5.Which item was the most popular for each customer?
6.Which item was purchased first by the customer after they became a member?
7.Which item was purchased just before the customer became a member?
8.What is the total items and amount spent for each member before they became a member?
9.If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?
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 customers A and B have at the end of January?

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

Query:

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

Output:

	customer_id	Total_Amount_Spent
1	A	76
2	B	74
3	C	36

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

Query:

```
SELECT customer_id,
        COUNT(DISTINCT order_date) AS Total_days_visited
FROM Danny_Diner.dbo.sales
GROUP BY customer_id
ORDER BY customer_id;
```

Output:

	customer_id	Total_days_visited
1	A	4
2	B	6
3	C	2

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

Query:

```
WITH CTE as
   (SELECT "customer_id", "order_date",
    a."product_id",product_name,min("order_date") OVER(partition by
    customer_id) as Min_Date
    FROM Danny_Diner.dbo.sales  as a inner join Danny_Diner.dbo.menu as b
    ON a.product_id = b.product_id )
SELECT "customer_id",product_name
FROM CTE
WHERE "order_date"=Min_Date
GROUP BY   "customer_id",product_name
```

Output:

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

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

Query:

```
SELECT m.product_id,product_name, price,
COUNT(s.product_id) AS total_purchases
FROM Danny_Diner.dbo.menu AS m
INNER JOIN Danny_Diner.dbo.sales AS s
ON m.product_id= s.product_id
GROUP BY m.product_id, product_name, price
ORDER BY total_purchases DESC;
```

Output:

	product_id	product_name	price	total_purchases
1	3	ramen	12	8
2	2	curry	15	4
3	1	sushi	10	3

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

Query:

```
WITH popular_items AS (
  SELECT customer_id, product_id,
  COUNT(*) AS order_count,
      ROW_NUMBER() OVER (PARTITION BY customer_id ORDER BY
COUNT(*) DESC) AS rn
  FROM Danny_Diner.dbo.sales
  GROUP BY customer_id, product_id
)
SELECT p.customer_id, m.product_name AS most_popular_item
FROM popular_items p
JOIN Danny_Diner.dbo.menu m
ON p.product_id = m.product_id
WHERE p.rn = 1
ORDER BY p.customer_id;
```

Output:

	customer_id	most_popular_item
1	A	ramen
2	B	sushi
3	C	ramen

Q6. Which item was purchased first by the customer after they became a member?

Query:

```
SELECT m.customer_id, m.join_date,
    MIN(s.order_date) AS first_purchase_date,
     u.product_name AS first_purchase_item
FROM Danny_Diner.dbo.members m
JOIN Danny_Diner.dbo.sales s ON m.customer_id = s.customer_id
JOIN Danny_Diner.dbo.menu u ON s.product_id = u.product_id
WHERE s.order_date > m.join_date
GROUP BY m.customer_id, m.join_date, u.product_name
ORDER BY m.customer_id;
```

Output:

	customer_id	join_date	first_purchase_date	first_purchase_item
1	A	2021-01-07	2021-01-10	ramen
2	B	2021-01-09	2021-01-16	ramen
3	B	2021-01-09	2021-01-11	sushi

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

Query:

```
SELECT m.customer_id, m.join_date,
       MAX(s.order_date) AS last_purchase_date,
       u.product_name AS last_purchase_item
FROM Danny_Diner.dbo.members m
JOIN Danny_Diner.dbo.sales s
ON m.customer_id = s.customer_id
JOIN Danny_Diner.dbo.menu u
ON s.product_id = u.product_id
WHERE s.order_date < m.join_date
GROUP BY m.customer_id, m.join_date, u.product_name
ORDER BY m.customer_id;
```

Output:

	customer_id	join_date	last_purchase_date	last_purchase_item
1	A	2021-01-07	2021-01-01	curry
2	A	2021-01-07	2021-01-01	sushi
3	B	2021-01-09	2021-01-02	curry
4	B	2021-01-09	2021-01-04	sushi

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

Query:

```
SELECT m.customer_id, m.join_date,
    COUNT(s.product_id) AS total_items,
    SUM(u.price) AS total_amount_spent
FROM Danny_Diner.dbo.members m
JOIN Danny_Diner.dbo.sales s
ON m.customer_id = s.customer_id
JOIN Danny_Diner.dbo.menu u
ON s.product_id = u.product_id
WHERE s.order_date < m.join_date
GROUP BY m.customer_id, m.join_date
ORDER BY m.customer_id;
```

Output:

	customer_id	join_date	total_items	total_amount_spent
1	A	2021-01-07	2	25
2	B	2021-01-09	3	40

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

Query:

```
SELECT s.customer_id,
SUM(
    CASE
    WHEN m.product_name = 'sushi' THEN 20 * m.price
    ELSE 10 * m.price
    END
) AS total_points
FROM Danny_Diner.dbo.sales s
JOIN Danny_Diner.dbo.menu m
ON s.product_id = m.product_id
GROUP BY s.customer_id
ORDER BY s.customer_id;
```

Output:

	customer_id	total_points
1	A	860
2	B	940
3	C	360

Q 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?

Query:

```
Select s.customer_id,
       Sum(CASE
       When (DATEDIFF(DAY, me.join_date, s.order_date) between 0
             and 7) or (m.product_ID = 1)
       Then m.price * 20
       Else m.price * 10
       END) As Points
From Danny_Diner.dbo.members as me
Inner Join Danny_Diner.dbo.sales as s on s.customer_id = me.customer_id
Inner Join Danny_Diner.dbo.menu as m on m.product_id = s.product_id
where s.order_date >= me.join_date and s.order_date <= CAST('2021-01-31'
AS DATE)
Group by s.customer_id
```

Output:

	customer_id	Points
1	A	1020
2	B	440

Conclusion

1.Customer A spent more (76) followed by B(74) and C (36). This makes Customer A the most valuable customer at the moment.
2.B had the most visits (6) while C had the least visits (2).
3.Customer A first orders are curry and sushi, Customer B's first order is curry, and Customer C's first order is ramen.
4.With 8 purchases, ramen is the most purchased item.
5.Customer's favourite item is ramen.
6.After becoming members, Customer A ordered curry and Customer B ordered sushi.
7.Just before becoming a member, Customer A bought sushi and curry while Customer B bought sushi.
8.Before becoming member, Customer A had spent 25 on 2 items and Customer B, 40 on 3 items.
9.A got 860 points, B got 940 points and C got 360 points.
10.A got 1370 points and B got 820 points.

Thanks

Nitesh Kumar Prajapati