Case Study #1 - Danny's Diner

Presented By: Gurjeet Singh Sodhi

Introduction

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.

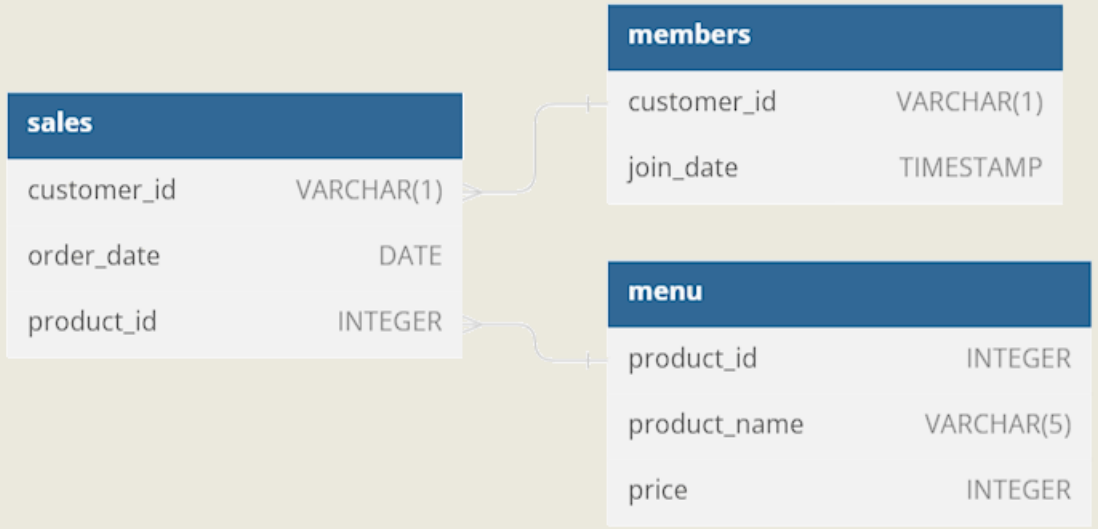
He plans on using these insights to help him decide whether he should expand the existing customer loyalty program - additionally he needs help to generate some basic datasets so his team can easily inspect the data without needing to use SQL.

Danny has provided us 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!

Dataset

Danny has shared 3 key datasets for this case study:

- sales
- menu
- members



Entity Relationship Diagram

Case Study 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 customer A and B have at the end of January?

Solution

I'm using Microsoft SQL Server and these are the functions used.

- Aggregate functions — SUM, MIN, MAX
- Numerical functions — TOP (WITH TIES)
- Joins — Inner join, left join
- Temp tables (CTE)
- Windows function
- Other Advanced functions- CAST, CASE STATEMENT

Let's look at the individual solution to the problem statement.

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

```
SELECT s.customer_id,
      '$'+ CAST(SUM(m.price) AS
VARCHAR) AS total_spent
FROM sales AS s
INNER JOIN menu AS m
ON s.product_id = m.product_id
GROUP BY s.customer_id;
```

Results | Messages

	customer_id	total_spent
1	A	$76
2	B	$74
3	C	$36

How many days has each customer visited the restaurant?

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

Results | Messages

	customer_id	visits
1	A	4
2	B	6
3	C	2

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

```
WITH first_item AS
(
SELECT s.customer_id,
      m.product_name, [rank] = RANK()
OVER(PARTITION BY customer_id ORDER BY
s.order_date ASC)
FROM sales AS s
INNER JOIN menu AS m
ON s.product_id = m.product_id
)
SELECT customer_id, product_name
FROM first_item
WHERE [rank] = 1
GROUP BY customer_id, product_name;
```

Results | Messages

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

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

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

Results | Messages

	product_id	product_name	total_sold
1	3	ramen	8

Which item was the most popular for each customer?

```
SELECT TOP 1 WITH TIES
                s.customer_id,
                COUNT(s.product_id) AS
fav_product,
                m.product_name
FROM sales AS s
INNER JOIN menu AS m
ON s.product_id = m.product_id
GROUP BY s.customer_id, m.product_name
ORDER BY DENSE_RANK() OVER(PARTITION
BY customer_id ORDER BY
COUNT(s.product_id) DESC);
```

Results | Messages

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

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

```
SELECT TOP 1 WITH TIES s.customer_id,
                       s.product_id,
                          s.order_date,
                          m.join_date,
                          me.product_name
FROM sales AS s
INNER JOIN members AS m
ON s.customer_id = m.customer_id
INNER JOIN menu AS me
ON me.product_id = s.product_id
WHERE s.order_date >= m.join_date
ORDER BY RANK() OVER(PARTITION BY
s.customer_id ORDER BY s.order_date ASC);
```

Results | Messages

	customer_id	product_id	order_date	join_date	product_name
1	A	2	2021-01-07	2021-01-07	curry
2	B	1	2021-01-11	2021-01-09	sushi

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

```
SELECT TOP 1 WITH TIES s.customer_id,
                       s.product_id,
                       s.order_date,
                       m.join_date,
                       me.product_name
FROM sales AS s
INNER JOIN members AS m
ON s.customer_id = m.customer_id
INNER JOIN menu AS me
ON me.product_id = s.product_id
WHERE s.order_date >= m.join_date
ORDER BY RANK() OVER(PARTITION BY
s.customer_id ORDER BY s.order_date ASC);
```

Results | Messages

	customer_id	product_id	order_date	join_date	product_name
1	A	1	2021-01-01	2021-01-07	sushi
2	A	2	2021-01-01	2021-01-07	curry
3	B	1	2021-01-04	2021-01-09	sushi

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

```
SELECT s.customer_id,
        COUNT(me.product_name) AS
total_items,
        '$'+CAST(SUM (me.price) AS
VARCHAR) AS amount_spend
FROM sales AS s
INNER JOIN members AS m
ON s.customer_id = m.customer_id
INNER JOIN menu AS me
ON me.product_id = s.product_id
WHERE s.order_date < m.join_date
GROUP BY s.customer_id;
```

Results | Messages

	customer_id	total_items	amount_spend
1	A	2	$25
2	B	3	$40

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(CASE
 WHEN m.product_name = 'sushi'
THEN m.price * 20
ELSE m.price * 10
        END) AS total_points

FROM sales AS s
INNER JOIN menu AS m
ON s.product_id = m.product_id
GROUP BY s.customer_id;
```

Results | Messages

	customer_id	total_points
1	A	860
2	B	940
3	C	360

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 dates AS (
                SELECT *, DATEADD(DAY,6, join_date) AS
first_week
                FROM members
            )
SELECT s.customer_id,
            SUM(CASE WHEN m.product_name = 'sushi' THEN
m.price * 20
                    WHEN s.order_date BETWEEN join_date
AND first_week THEN m.price * 20
                    ELSE m.price * 10 END) AS points
FROM dates AS dt
LEFT JOIN sales AS s
ON dt.customer_id = s.customer_id
INNER JOIN menu AS m
ON s.product_id = m.product_id
WHERE s.order_date < CAST( '2021-02-01' AS DATE)
GROUP BY s.customer_id;
```

Results | Messages

	customer_id	points
1	A	1370
2	B	820

Bonus Questions

Join All The Things

```
SELECT s.customer_id,
        s.order_date,
       me.product_name,
       me.price,
      (CASE
         WHEN m.join_date > s.order_date THEN 'N'
         WHEN m.join_date IS NULL THEN 'N'
         ELSE 'Y'
         END) AS member

FROM sales AS s
LEFT JOIN members AS m
ON s.customer_id = m.customer_id
INNER JOIN menu AS me
ON me.product_id = s.product_id;
```

Results | Messages

	customer_id	order_date	product_name	price	member
1	A	2021-01-01	sushi	10	N
2	A	2021-01-01	curry	15	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

Bonus Questions

Rank All The Things

```
WITH ranking_cte AS (
SELECT s.customer_id,
      s.order_date,
           me.product_name,
           me.price,
           (CASE
                      WHEN m.join_date > s.order_date THEN 'N'
                      WHEN m.join_date IS NULL THEN 'N'
                      ELSE 'Y' END) AS member
FROM sales AS s
LEFT JOIN members AS m
ON s.customer_id = m.customer_id
INNER JOIN menu AS me
ON me.product_id = s.product_id
)
SELECT customer_id,
      order_date,
         product_name,
         price,
         member,
        (CASE
           WHEN member = 'N' THEN NULL
                      ELSE RANK() OVER (PARTITION BY
customer_id,member ORDER BY order_date ASC)
            END)AS ranking
                      FROM ranking_cte;
```

Results | Messages

	customer_id	order_date	product_name	price	member
1	A	2021-01-01	sushi	10	N
2	A	2021-01-01	curry	15	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

Key Insights

- Total amount spent by A is $76, for B it is $74 and C spent $36
- B is the most frequent visitor among all the Customers
- Ramen is the most ordered food item
- Before they became members, both Customers A and B spent $25 and $40 respectively
- Total points for Customer A, B and C are 860, 940 and 360 respectively

Thank You !