Case Study #1 - Danny's Diner

Danny Ma · May 1, 2021

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 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!

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

- `sales`
- `menu`
- `members`

Case Study Questions

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

```
DANNYS_DINER.PUBLIC ▾   Settings ▾

61
62  SELECT CUSTOMER_ID, SUM(PRICE) AS TOTAL_PRICE
63  FROM SALES s
64  JOIN MENU m ON s.PRODUCT_ID = m.PRODUCT_ID
65  GROUP BY 1;
66
67
```

Results | Chart

	CUSTOMER_ID	TOTAL_PRICE
1	A	76
2	B	74
3	C	36

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

```
DANNYS_DINER.PUBLIC ▾   Settings ▾

68
69  SELECT CUSTOMER_ID, COUNT(DISTINCT ORDER_DATE) AS DAYS_VISITED
70  FROM SALES
71  GROUP BY 1;
72
73
```

Results | Chart

	CUSTOMER_ID	DAYS_VISITED
1	A	4
2	B	6
3	C	2

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

```
DANNYS_DINER.PUBLIC ▾   Settings ▾

74
75  SELECT CUSTOMER_ID, PRODUCT_NAME FROM
76  (SELECT CUSTOMER_ID, PRODUCT_NAME, DENSE_RANK() OVER(PARTITION BY CUSTOMER_ID ORDER BY ORDER_DATE) AS FIRST_ORDER
77  FROM SALES s
78  JOIN MENU m ON s.PRODUCT_ID = m.PRODUCT_ID)
79  WHERE FIRST_ORDER = 1
80  GROUP BY 1,2;
81
82
83
84
```

Results | Chart

	CUSTOMER_ID	PRODUCT_NAME
1	A	sushi
2	A	curry
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?

```
DANNYS_DINER.PUBLIC  Settings

SELECT m.product_name, COUNT(s.product_id) AS MOST_PURCHASED
FROM sales s
JOIN menu m ON s.product_id=m.product_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1;
```

Results | Chart

	PRODUCT_NAME	MOST_PURCHASED
1	ramen	8

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

```
DANNYS_DINER.PUBLIC  Settings

SELECT CUSTOMER_ID, PRODUCT_NAME FROM
(SELECT CUSTOMER_ID, PRODUCT_NAME, COUNT(*) AS PP, DENSE_RANK() OVER(PARTITION BY CUSTOMER_ID ORDER BY PP DESC) AS RNK
FROM sales s
JOIN menu m ON s.product_id=m.product_id
GROUP BY 1,2)
WHERE RNK=1;
```

Results | Chart

	CUSTOMER_ID	PRODUCT_NAME
1	A	ramen
2	B	curry
3	B	sushi
4	B	ramen
5	C	ramen

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

```
DANNYS_DINER.PUBLIC  Settings

WITH cte as
(SELECT s.CUSTOMER_ID, m.PRODUCT_NAME, RANK() OVER(PARTITION BY s.CUSTOMER_ID ORDER BY s.order_date) first_order
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)

SELECT CUSTOMER_ID,PRODUCT_NAME FROM
cte
WHERE first_order= 1
```

Results | Chart

	CUSTOMER_ID	PRODUCT_NAME
1	A	curry
2	B	sushi

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

```
DANNYS_DINER.PUBLIC ▾   Settings ▾

123
124  WITH cte1 as
125  (SELECT s.CUSTOMER_ID, m.PRODUCT_NAME, RANK() OVER(PARTITION BY s.CUSTOMER_ID ORDER BY s.order_date desc) first_order
126  FROM sales s
127  JOIN menu m ON s.product_id=m.product_id
128  JOIN MEMBERS mem ON s.customer_id = mem.customer_id
129  WHERE s.order_date < mem.join_date)
130
131  SELECT CUSTOMER_ID,PRODUCT_NAME FROM
132  cte1
133  WHERE first_order= 1
134
```

Results | Chart

	CUSTOMER_ID	PRODUCT_NAME
1	A	sushi
2	A	curry
3	B	sushi

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

```
DANNYS_DINER.PUBLIC ▾   Settings ▾

136
137  SELECT s.CUSTOMER_ID, COUNT(s.PRODUCT_ID) AS TOTAL_ITEMS , SUM(m.PRICE) AS AMOUNT_SPENT
138  FROM sales s
139  JOIN menu m ON s.product_id=m.product_id
140  JOIN MEMBERS mem ON s.customer_id = mem.customer_id
141  WHERE s.order_date < mem.join_date
142  GROUP BY 1;
143
```

Results | Chart

	CUSTOMER_ID	TOTAL_ITEMS	AMOUNT_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?

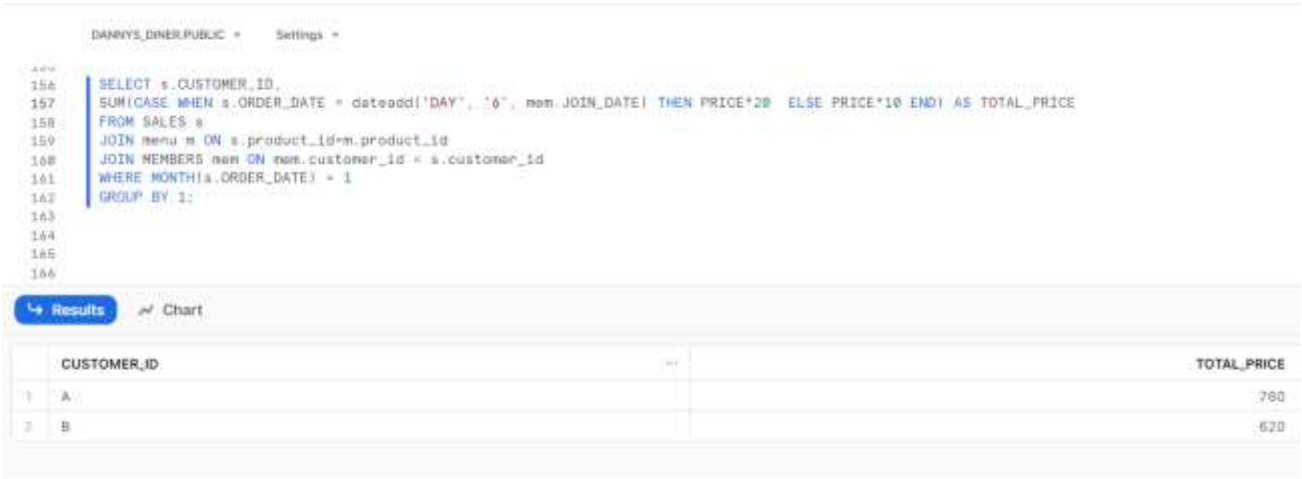
```
DANNYS_DINER.PUBLIC ▾   Settings ▾

146
147  SELECT CUSTOMER_ID,
148  SUM(CASE WHEN PRODUCT_NAME = 'sushi' THEN PRICE*20 ELSE PRICE*10 END) AS TOTAL_POINTS
149  FROM SALES s
150  JOIN menu m ON s.product_id=m.product_id
151  GROUP BY 1;
152
```

Results | Chart

	CUSTOMER_ID	TOTAL_POINTS
1	A	860
2	B	940
3	C	360

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?



DANNYS_DINER.PUBLIC ▾ Settings ▾

```
SELECT s.CUSTOMER_ID,
SUM(CASE WHEN s.ORDER_DATE = dateadd('DAY', '6', mem.JOIN_DATE) THEN PRICE*20  ELSE PRICE*10 END) AS TOTAL_PRICE
FROM SALES s
JOIN menu m ON s.product_id=m.product_id
JOIN MEMBERS mem ON mem.customer_id = s.customer_id
WHERE MONTH(s.ORDER_DATE) = 1
GROUP BY 1;
```

Results | Chart

	CUSTOMER_ID	TOTAL_PRICE
1	A	760
2	B	620