# Danny's Dinner

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/case\_study1\_photo\_danny.png)

## Introduction

This Danny's Dinner is the first out of eight case studies by Danny Ma in his 8 weeks SQL challenge. Out of Danny's love Japanese food, he chooses to set up a restaurant at the start of 2021, that sold sushi, curry, and ramen which were 3 of his most special foods. After a few months of operation, some data and insight have been required based on the given data to see how the restaurant can still stay in business. Sales, Menu, and Members were the 3 tables provided alongside their values. SQL was the tool used for this project. All queries in this project were run on pgAdmin 4.

\*\*\_Disclaimer\_\*\*: \_The Danny Dinner's Case Study 1 (just before the introduction) was gotten from 8weeksqlchallenge.com and has only been used for illustrative purposes. Also, the datasets do not represent any specific Japanese restaurant, and have been utilized to demonstrate the various functions in SQL.\_

## Skills Demonstrated

\* Database Creation

\* Schema Creation

\* Table Creation

\* Inserting of Values

\* Joins

\* Group By

\* Order By

\* Conditional Statement (CASE WHEN)

\* Common Table Expressions (CTE)

\* Aggregate Functions

\* Window functions

## Problem Statement

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?

## Data Sourcing

The data used for this project was sourced from the [8 weeks sql challenge website](https://8weeksqlchallenge.com/case-study-1/), where links to other case studies can be found. All table values and problem statements are also displayed on the website.

## Tables and Database

Database and Schema were first created on pgAdmin 4

\* \*\*Database Creation\*\*

Due to data organization and management reasons, a database is created.

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/data\_base\_creation.png)

---

\* \*\*Schema Creation\*\*

This serves as a container or enclosure that holds database objects like tables and indexes. Whenever pgAdmin 4 is closed and restarted, the Schema query (SET search\_path = dannys\_dinner) must be run, before running previously saved queries to avoid errors.

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/schema\_creation.png)

---

\* \*\*Table Creation and Inserting of Values\*\*

There are 3 tables (Sales, Menu, and Members) were created for this project and their values were inserted accordingly:

\*\*1. Sales\*\*

This table has 3 columns namely; "customer\_id", "order\_date", and "product\_id", and 15 rows.

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/sales\_table\_creation.png)

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/sales\_table\_insert\_values.png)

---

\*\*2. Menu\*\*

Being the second table, Menu contains 3 columns ("product\_id, "product\_name", and "price") and 3 rows.

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/menu\_table\_creation.png)

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/menu\_table\_insert\_values.png)

---

\*\*3. Members\*\*

As seen in the screenshot below, the Members table contains 2 columns ("customer\_id", and "join\_date") and 2 rows.

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/members\_table\_creation.png)

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/members\_table\_insert\_values.png)

---

\* \*\*Entity Relationship\*\*

As a model, the Entity Relationship (ER) model gives a visual representation of the relationship between entities in the database. In this case, it shows the relationships between the 3 tables. It is evident that there is a link between the "customer\_id" columns of the Sales and Members tables, and there is a relationship between the "product\_id" of the Sales and Menu tables.

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/case\_study1\_entity\_relationship.png)

---

## Problem Statement Analysis

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

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question1cr\_code.png)

To get this data, since there are 3 tables ("Sales", "Menu", and "Members"), the Sales and Menu tables are joined (with "product\_id" as the common entity) to see the total amount that all customers spent. Recall that not all customers are members. "GROUP BY 1" entails that the result is grouped by the first column, which is the "customer\_id". "ORDER BY 1" like the group by, means that the results are arranged in order of increasing "customer\_id".

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question1\_answer.png)

---

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

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question2cr\_code.png)

Aggregate function (COUNT) was used to sum up the order date to know the number of days each customer visited the restaurant.

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question2\_answer.png)

---

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

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question3\_code.png)

Window function was used to rank the items on the menu for each customer, and the result was ordered by "s.order\_date" which is the order date from the Sales table.

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question3\_answer.png)

---

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

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question4cr\_code.png)

After joining the Sales and Menu tables, the distinct products were selected, and these distinct products were counted. The results were ordered in descending order so that the product that was purchased the most will come top. "LIMIT 1" was used to show only the topmost result.

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question4\_answer.png)

---

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

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question5\_code.png)

Common Table Expressions (CTE), join, and window functions were used to solve this problem. The CTE is a temporary table and it is not stored in the database.

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question5\_answer.png)

---

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

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question6\_code.png)

In addition to other functions that have been mentioned in the previous solutions, the conditional statement (CASE WHEN) was used to solve this problem.

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question6\_answer.png)

---

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

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question7\_code.png)

There are only 2 unique customer\_ids in this solution because C is not a member.

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question7\_answer.png)

---

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

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question8cr\_code.png)

All 3 tables were combined to get the total amount that customers A and B (both members) spent.

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question8\_answer.png)

---

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

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question9cr\_code.png)

Conditional statement was used to ascertain the maximum points that each customer had.

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question9\_answer.png)

---

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?

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question10\_code.png)

A combination of functions, including 2 CTEs, conditional statement, and joins were used to solve this problem.

![](https://github.com/iprincewill3/Danny-s-Dinner---Case-Study-1/blob/main/question10\_answer.png)

## Conclusion

\* In terms of the total amount spent by each customer, A spent $76, B spent $74, and C spent $36.

\* Customers A and B visited the restaurant for 6 days (each), while C visited the restaurant for 3 days.

\* The first item on the menu purchased by Customer A was Curry, B was curry, and C was ramen.

\* Ramen was the most purchased item on the menu.

\* Customers A and C had ramen as their most popular item on the menu, with 3 counts each for the number of times the item was purchased. Customer B had 2 purchases for sushi, and 2 for curry.

\* Ramen was the first item purchased by Customer A after becoming a member, while it was sushi for Customer B.

\* Before customers A and B became members, A had purchased 2 items for a combined sum of $25, and Customer B had purchased 3 items for a combined sum of $40.

## Recommendation

\* In the future, more data on the time of the day that each customer comes to the restaurants should be included, for example, afternoon, evening, or morning. This data can help to derive more insights to understand the best time of the day to give offers to customers that will attract them to the restaurant and drive more sales for the restaurant manager or owner.