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`

Example Datasets

All datasets exist within the `dannys_diner` database schema - be sure to include this reference within your SQL scripts as you start exploring the data and answering the case study questions.

Table 1: sales

The `sales` table captures all `customer_id` level purchases with an corresponding `order_date` and `product_id`information for when and what menu items were ordered.

customer_idorder_dateproduct_id

A2021–01–011A2021–01–012A2021–01–072A2021–01–103A2021–01–113A2021–01–113B2021–01–012B2021–01–022B2021–01–041B2021–01–111B2021–01–163B2021–02–013C2021–01–013C2021–01–013C2021–01–073

Table 2: menu

The menu table maps the `product_id` to the actual `product_name` and `price` of each menu item.

product_id product_name price

1 sushi

102 curry

153 ramen12

Table 3: members

The final `members` table captures the `join_date` when a `customer_id` joined the beta version of the Danny's Diner loyalty program.

customer_idjoin_dateA2021–01–07B2021–01–09

Interactive SQL Session

You can use the embedded DB Fiddle below to easily access these example datasets—this interactive session has everything you need to start solving these questions using SQL.

You can click on the `Edit on DB Fiddle` link on the top right hand corner of the embedded session below and it will take you to a fully functional SQL editor where you can write your own queries to analyse the data.

You can feel free to choose any SQL dialect you'd like to use, the existing Fiddle is using PostgreSQL 13 as default.

Serious SQL students have access to a dedicated SQL script in the 8 Week SQL Challenge section of the course which they can use to generate relevant temporary tables like we've done throughout the entire course!

Case Study Questions

Each of the following case study questions can be answered using a single SQL 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?

Bonus Questions

Join All The Things

The following questions are related creating basic data tables that Danny and his team can use to quickly derive insights without needing to join the underlying tables using SQL.

Recreate the following table output using the available data:

customer_idorder_dateproduct_namepricememberA2021–01–01curry15NA2021–01–01sushi10NA2021–01–07curry15YA2021–01–10ramen12YA2021–01–11ramen12YA2021–01–11ramen12YB2021–01–01curry15NB2021–01–02curry15NB2021–01–04sushi10NB2021–01–11sushi10YB2021–01–16ramen12YB2021–02–01ramen12YC2021–01–01ramen12NC2021–01–01ramen12NC2021–01–07ramen12N

Rank All The Things

Danny also requires further information about the `ranking` of customer products, but he purposely does not need the ranking for non-member purchases so he expects null `ranking` values for the records when customers are not yet part of the loyalty program.

customer_idorder_dateproduct_namepricememberrankingA2021–01–01curry15NnullA2021–01–01sushi10NnullA2021–01–07curry15Y1A2021–01–10ramen12Y2A2021–01–11ramen12Y3A2021–01–11ramen12Y3B2021–01–01curry15NnullB2021–01–02curry15NnullB2021–01–04sushi10NnullB2021–01–11sushi10Y1B2021–01–16ramen12Y2B2021–02–01ramen12Y3C2021–01–01ramen12NnullC2021–01–01ramen12NnullC2021–01–07ramen12Nnull

Next Steps

It's highly recommended to save all of your code in a separate IDE or text editor as you are trying to solve the problems in the provided SQL Fiddle instance above!

Ready for the next 8 Week SQL challenge case study?

Conclusion

I really hope you enjoyed this fun little case study—it definitely was fun for me to create!