8WEEKSQLCHALLENGE.COM CASE STUDY #1



DATAWITHDANNY.COM

Made by: Mai Huy Khang

Note: all results use for practicing SQL (MySQL) only

INTRODUCTION

Context

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 favorite foods: sushi, curry and ramen.

Danny's Diner needs 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 favorite. Having this deeper connection with his customers will help him deliver a better and more personalized 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.

CHAPTER 1: CREATE DATASET AS REQUIRED

Coding part

```
CREATE SCHEMA dannys diner;
SET search path = dannys diner;
CREATE TABLE sales (
   customer id VARCHAR(1),
  order date DATE,
  product id INTEGER
);
INSERT INTO sales
   (customer_id, order_date, product_id)
VALUES
   ('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'),
          '2021-01-01', '2'),
  ('B', '2021-01-16', '3'),

('B', '2021-02-01', '3'),

('C', '2021-01-01', '3'),

('C', '2021-01-01', '3'),

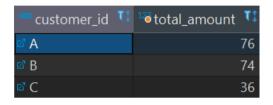
('C', '2021-01-07', '3');
          '2021-01-16', '3'),
CREATE TABLE menu (
   product id INTEGER,
  product_name VARCHAR(5),
  price INTEGER
);
INSERT INTO menu
   (product id, product name, price)
VALUES
   ('1', 'sushi', '10'),
   ('2', 'curry', '15'), ('3', 'ramen', '12');
```

```
CREATE TABLE members (
  customer_id VARCHAR(1),
  join_date DATE
);

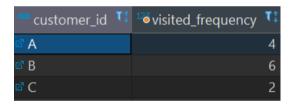
INSERT INTO members
  (customer_id, join_date)
VALUES
  ('A', '2021-01-07'),
  ('B', '2021-01-09');
```

CHAPTER 2: SOLVE CASE STUDY QUESTION

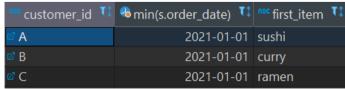
1) What is the total amount each customer spent at the restaurant?



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?

```
count(s.product_id) as times
FROM sales as s
LEFT JOIN menu as m
USING(product_id)
GROUP BY m.product_name
ORDER BY times DESC limit 1
***Count(s.product_id) as times

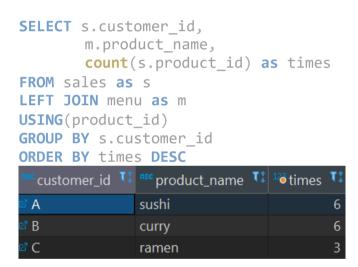
### USING(product_id)

### ORDER BY times Times

**Totimes Times

**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes Times
**Totimes
**Totimes Times
**Totimes
```

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

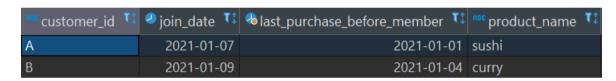


6) Which item was purchased first by the customer after they became a member?

```
A 2021-01-07 2021-01-07 curry

B 2021-01-09 2021-01-11 sushi
```

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?

```
With points as (
                 SELECT product id ,product name, price,
                        CASE when product name = 'sushi' then 20
                        else 10
                        END as point per dollar
                 FROM menu
SELECT s.customer id,
        sum(price*point per dollar) as total points
FROM sales as s
LEFT JOIN points as p
USING(product id)
GROUP BY s.customer id
 customer_id 🌃 🌃 total_points 👯
'A
₫ B
                         940
<sup>a</sup> 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?

```
With points as (
                 SELECT product id,
                         product name,
                         price,
                         CASE when product name = 'sushi' then 20
                         else 10
                         END as point per dollar
                 FROM menu
                      )
 , points daily as (
                 SELECT s.customer id,
                         sum(price*point per dollar) as
total points
                 FROM sales as s
                 LEFT JOIN points as p
                 USING(product id)
                 LEFT JOIN members as m
```

```
USING(customer id)
                  WHERE (date(order date) NOT BETWEEN
date(join date) AND date_add(join date, INTERVAL 7 DAY)) AND
month(order date) = 1
                  GROUP BY s.customer id
 , point firstweek as (
                        SELECT s.customer id,
                                sum(price*20) as total points
                        FROM sales as s
                              LEFT JOIN menu as m
                              USING(product id)
                              LEFT JOIN members as m2
                              USING(customer id)
                        WHERE (date(order date) BETWEEN
date(join date) AND date add(join date, INTERVAL 7 DAY)) AND
month(order date) = 1
                        GROUP BY s.customer id
                        )
 , final table as (
                        SELECT customer_id,
                                total_points
                        FROM point firstweek
                        UNION ALL
                        SELECT customer_id,
                                total points
                        FROM points daily
SELECT customer_id,
        sum(total points) as total points Jan
FROM final table
GROUP BY customer id
  customer_id Table 123 total_points_Jan Table 123 total_points_Jan
                           1,370
```

940

#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:

ď A	2021-01-01	sushi	10	N
ď A	2021-01-01	curry	15	N
ď A	2021-01-07	curry	15	Υ
ď A	2021-01-10	ramen	12	Υ
ď A	2021-01-11	ramen	12	Υ
ď A	2021-01-11	ramen	12	Υ
⊠ B	2021-01-01	curry	15	N
⊠ B	2021-01-02	curry	15	N
⊠ B	2021-01-04	sushi	10	N
⊠ B	2021-01-11	sushi	10	Υ
⊠ B	2021-01-16	ramen	12	Υ
⊠ B	2021-02-01	ramen	12	Υ
ď C	2021-01-01	ramen	12	N
ď C	2021-01-01	ramen	12	N
ď C	2021-01-07	ramen	12	N

#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_id 👣	order_date 😲	product_name 👣	123 price T:	member_regis 👣	[№] ranking T
ď A	2021-01-01	sushi	10	N	null
ď A	2021-01-01	curry	15	N	null
ď A	2021-01-07	curry	15	Υ	1
ď A	2021-01-10	ramen	12	Υ	2
ď A	2021-01-11	ramen	12	Υ	3
ď A	2021-01-11	ramen	12	Υ	3
ď B	2021-01-01	curry	15	N	null
ď B	2021-01-02	curry	15	N	null
ď B	2021-01-04	sushi	10	N	null
ď B	2021-01-11	sushi	10	Υ	1
ď B	2021-01-16	ramen	12	Υ	2