🍽️

Danny’s Diner Case Study

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 favorite 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 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.

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

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_id	order_date	product_id
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

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

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	10
2	curry	15
3	ramen	12

Table 3: members

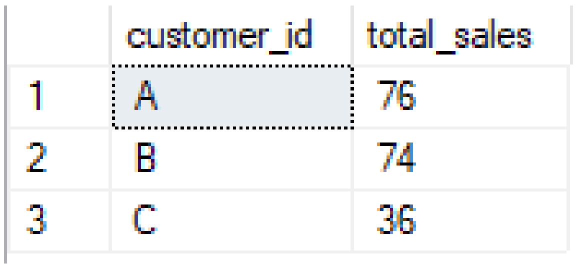
The final `members` table captures the `join_date` when a `customer_id` joined the beta version of the Danny's Diner loyalty program.

customer_id	join_date
A	2021-01-07
B	2021-01-09

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

```
SELECT
  sales.customer_id,
  SUM(menu.price) AS total_sales
FROM sales
INNER JOIN menu
  ON sales.product_id = menu.product_id
```

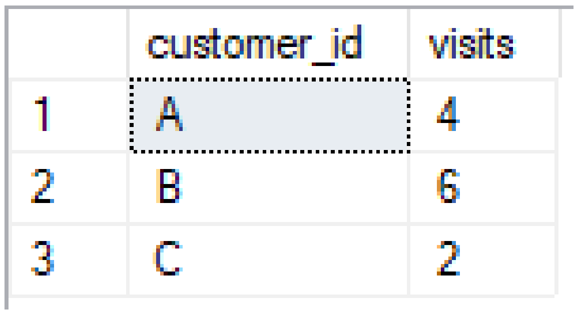
```
GROUP BY sales.customer_id
ORDER BY sales.customer_id ASC;
```



	customer_id	total_sales
1	A	76
2	B	74
3	C	36

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

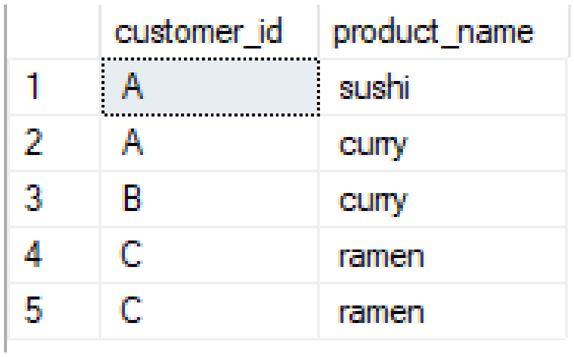
```
SELECT customer_id, COUNT(DISTINCT order_date) as visits
FROM sales
GROUP BY customer_id
```



	customer_id	visits
1	A	4
2	B	6
3	C	2

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

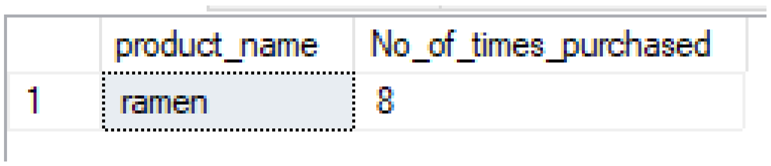
```
WITH customer_order_cte AS(
SELECT customer_id,
    order_date,
    product_name,
    RANK() OVER( PARTITION BY customer_id ORDER BY order_date) a

FROM sales as s INNER JOIN menu as m ON s.product_id = m.product
)
SELECT customer_id, product_name
FROM customer_order_cte
WHERE Rank = 1
```



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

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

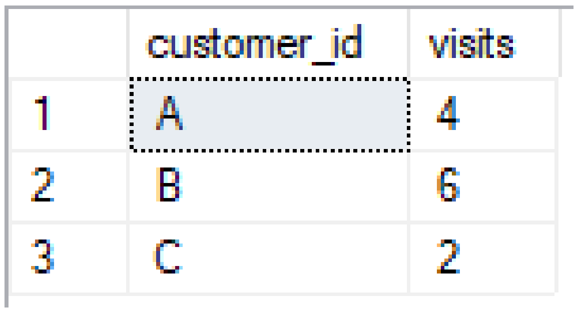
```
SELECT TOP(1) product_name, COUNT(s.product_id) as No_of_times_p
FROM sales as s INNER JOIN menu as m ON s.product_id = m.product
GROUP BY product_name
ORDER BY No_of_times_purchased DESC
```



	product_name	No_of_times_purchased
1	ramen	8

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

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

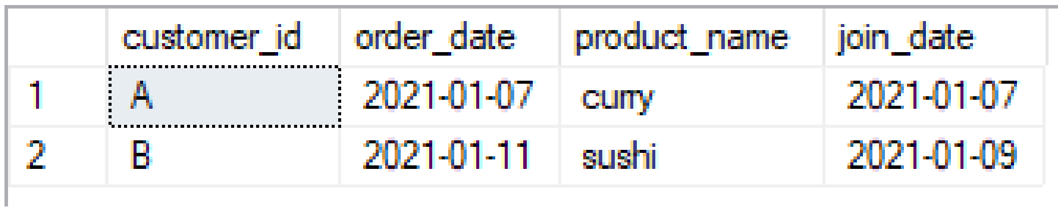


	customer_id	visits
1	A	4
2	B	6
3	C	2

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

```
WITH first_order AS(
SELECT s.customer_id,
        order_date,
        product_id,
        join_date,
        RANK() OVER(PARTITION BY s.customer_id ORDER BY order_da
FROM sales as s RIGHT JOIN members as mem ON s.customer_id = mem
WHERE order_date >= join_date
)
SELECT customer_id,
        order_date,
        product_name,
        join_date
```

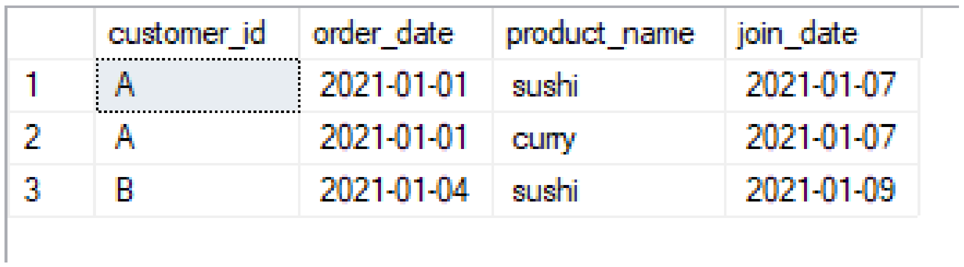
```
FROM first_order as fo INNER JOIN menu as m ON fo.product_id = r
WHERE Rank = 1
```



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

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

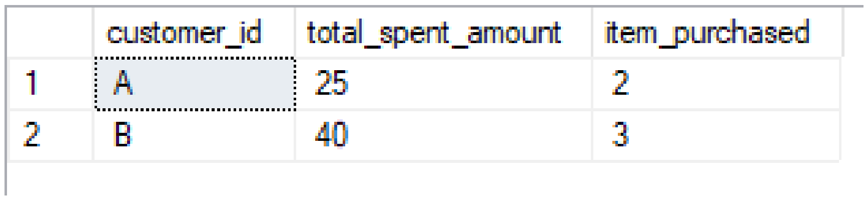
```
WITH Last_order AS(
SELECT s.customer_id,
        order_date,
        product_id,
        join_date,
        RANK() OVER(PARTITION BY s.customer_id ORDER BY order_da
FROM sales as s INNER JOIN members as mem ON s.customer_id = mer
WHERE order_date < join_date
)
SELECT customer_id,
        order_date,
        product_name,
        join_date
FROM Last_order as lo INNER JOIN menu as m ON lo.product_id = m
WHERE Rank = 1
```



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

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

```
SELECT s.customer_id, SUM(price) as total_spent_amount, COUNT(*
FROM sales as s INNER JOIN members as mem
    ON s.customer_id = mem.customer_id INNER JOIN menu as m ON s
WHERE order_date < join_date
GROUP BY s.customer_id
```

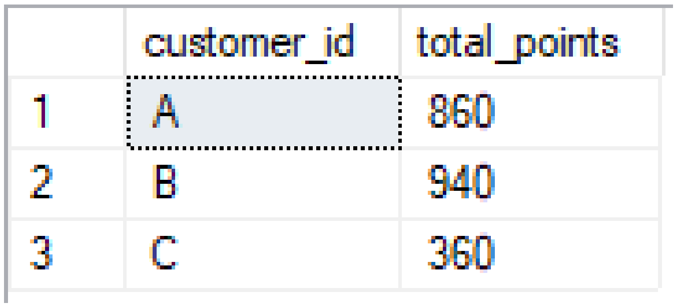


	customer_id	total_spent_amount	item_purchased
1	A	25	2
2	B	40	3

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

```
# APPROACH 1
SELECT customer_id,
    SUM(CASE WHEN m.product_id = 1 THEN price*20
    ELSE price*10 END) as points
FROM sales as s LEFT JOIN menu as m ON s.product_id = m.product_
GROUP BY customer_id
```

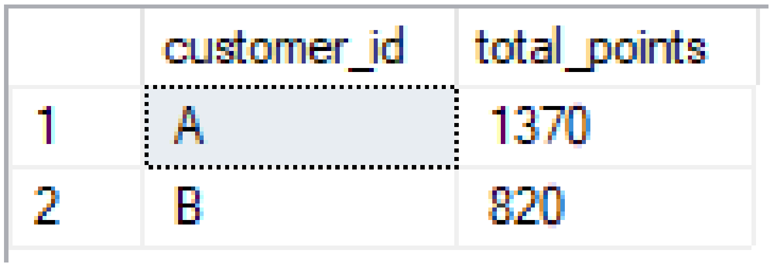
```
# APPROACH 2
WITH cte as(
SELECT *,
    (CASE WHEN product_name = 'sushi'  THEN price*20
    ELSE price*10 END) as points
FROM menu
)

SELECT customer_id, SUM(c.points) as total_points
FROM sales as s LEFT JOIN cte as c ON s.product_id = c.product_
GROUP BY customer_id
```



	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?

```
WITH dates_cte AS(
    SELECT *,
        DATEADD(DAY, 6, join_date) AS valid_date,
        EOMONTH('2021-01-1') AS last_date
    FROM members
)

SELECT s.customer_id,
        SUM( CASE
                WHEN product_name = 'sushi' THEN price*20
                WHEN s.order_date BETWEEN d.join_date AND d.vali
                ELSE price*10 END) AS total_points

FROM dates_cte as d JOIN sales as s ON d.customer_id = s.custome
WHERE order_date <= d.last_date
GROUP BY s.customer_id
```



	customer_id	total_points
1	A	1370
2	B	820