

*Case Study*  
**Danny's Diner**  
*The Taste of Success*

By Saswati Sadhu





# *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-

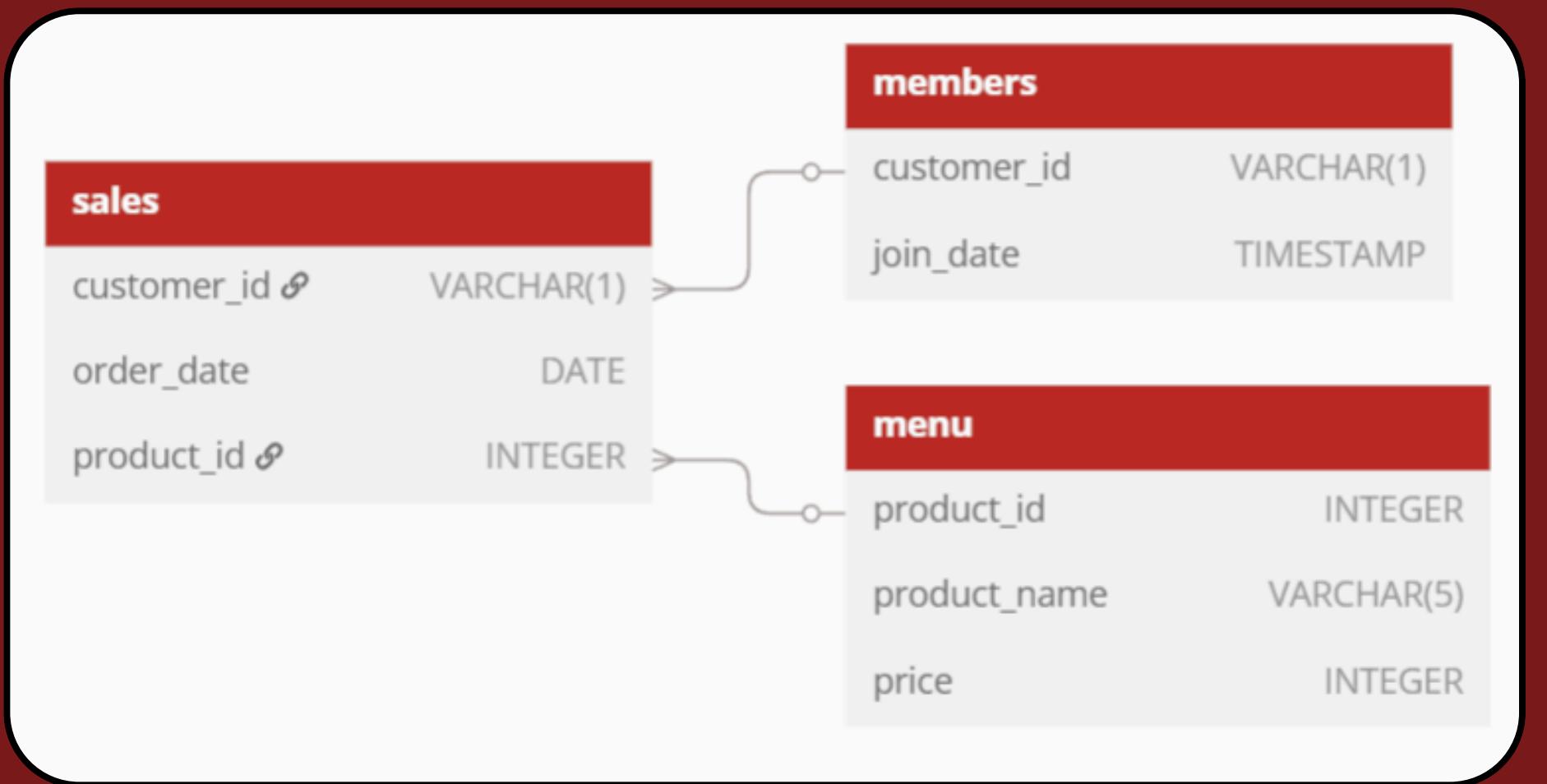
1. visiting patterns
2. how much money they've spent
3. which menu items are their favorite

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



# Dataset Overview

- sales
- menu
- members



- What is the total amount each customer spent at the restaurant?

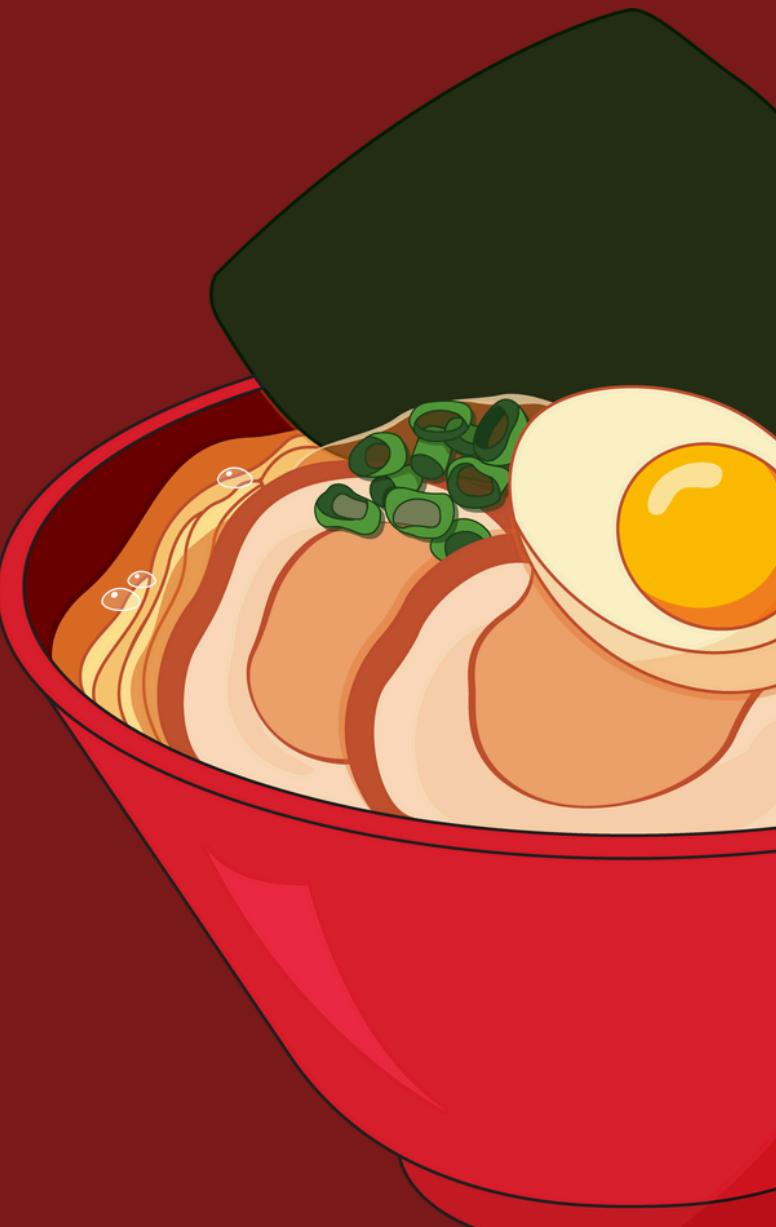
```
select s.customer_id,  
       sum(m.price) as total_amount_spent  
  from sales s inner join menu m  
  on s.product_id = m.product_id  
 group by s.customer_id;
```

	customer_id	total_amount_spent
▶	A	76
	B	74
	C	36

- *How many days has each customer visited the restaurant?*

```
select customer_id, count(order_date) as days_visited  
from sales  
group by customer_id;
```

	customer_id	days_visited
▶	A	6
	B	6
	C	3



- *What was the first item from the menu purchased by each customer?*

```
with ranked_sales as (
  select s.customer_id,s.order_date,m.product_name,
         row_number()over(partition by s.customer_id order by order_date asc) as rn
  from sales s inner join menu m
  on s.product_id = m.product_id
)
select customer_id,product_name,order_date
from ranked_sales
where rn=1;
```

	customer_id	product_name	order_date
▶	A	sushi	2021-01-01
	B	curry	2021-01-01
	C	ramen	2021-01-01

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

```
select m.product_name, count(s.product_id) as total_purchases  
from sales s inner join menu m on s.product_id = m.product_id  
group by m.product_name  
order by total_purchases desc  
limit 1;
```

	product_name	total_purchases
→	ramen	8

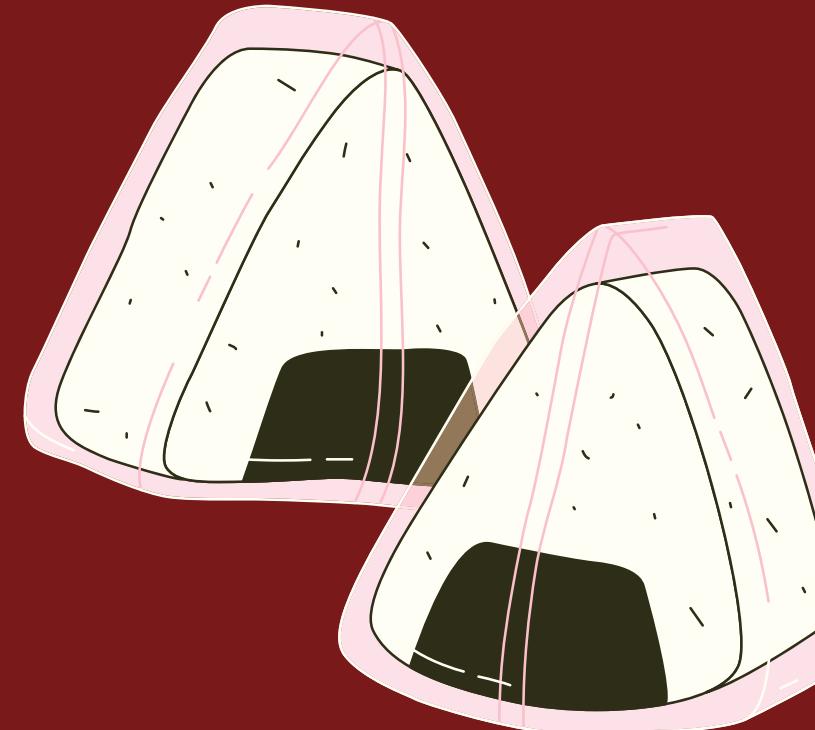
- Which item was the most popular for each customer?

```
with cte as(
    select s.customer_id,
        m.product_name,
        count(s.product_id) as purchase_times,
        row_number()over(partition by s.customer_id order by count(s.product_id) DESC) as rn
    from sales s inner join menu m
    on s.product_id = m.product_id
    group by s.customer_id, m.product_name
)
select customer_id, product_name
from cte
where rn =1;
```

	customer_id	product_name
→	A	ramen
	B	curry
	C	ramen

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

```
with cte as(
    select s.customer_id, m.product_name,s.order_date,
           row_number()over(partition by s.customer_id order by s.order_date) as rn
    from sales s inner join menu m
    on s.product_id = m.product_id
    inner join members mem
    on s.customer_id = mem.customer_id
    where s.order_date >= mem.join_date
)
select customer_id, product_name,order_date
from cte
where rn=1;
```



	customer_id	product_name	order_date
▶	A	curry	2021-01-07
	B	sushi	2021-01-11

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

```
with cte as(
    select s.customer_id, m.product_name,s.order_date,
           row_number()over(partition by s.customer_id order by s.order_date desc) as rn
    from sales s inner join menu m
    on s.product_id = m.product_id
    inner join members mem
    on s.customer_id = mem.customer_id
    where s.order_date < mem.join_date
)
select customer_id, product_name,order_date
from cte
where rn = 1;
```

	customer_id	product_name	order_date
▶	A	sushi	2021-01-01
	B	sushi	2021-01-04

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

```
select s.customer_id, count(s.product_id) as total_items , sum(m.price) as amount_spent  
from sales s inner join menu m  
on s.product_id = m.product_id  
inner join members mem  
on s.customer_id = mem.customer_id  
where s.order_date < mem.join_date  
group by s.customer_id;
```

	customer_id	total_items	amount_spent
▶	B	3	40
	A	2	25



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

```
select s.customer_id,  
       sum(case  
             when m.product_name = 'sushi' then m.price * 20  
             else m.price * 10  
           end) as points  
  from sales s  
 join menu m on s.product_id = m.product_id  
 group by s.customer_id;
```

	customer_id	points
▶	A	860
	B	940
	C	360

- 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?

```
select s.customer_id,  
       sum(case  
           when s.order_date between mem.join_date and date_add(mem.join_date, interval 6 day)  
           then m.price * 20  
           when s.order_date > date_add(mem.join_date, interval 6 day) and m.product_name = 'sushi'  
           then m.price * 20  
           else m.price * 10 end) as points  
  from sales s inner join members mem on s.customer_id = mem.customer_id  
         join menu m on s.product_id = m.product_id  
 where s.order_date >= mem.join_date  
       and s.order_date <= '2021-01-31'  
       and s.customer_id in ('A', 'B')  
 group by s.customer_id;
```

	customer_id	points
▶	B	320
▶	A	1020

# *Key Insights*

## **1. Top Spenders:**

- Customer A spent the most (\$76), followed by B (\$74); C spent the least (\$36).

## **2. Visit Frequency:**

- A and B visited 6 times; C visited 3 times.

## **3. First Purchases:**

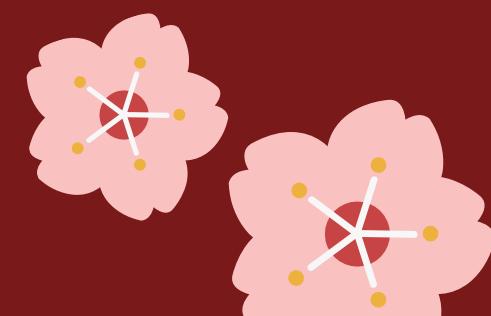
- A chose Sushi, B chose Curry, C chose Ramen.

## **4. Most Ordered Item:**

- Ramen was the overall favorite dish.

## **5. Customer Preferences:**

- A & C preferred Ramen; B consistently chose Curry.



# *Key Insights*

## **5. Membership Impact:**

- After joining, A shifted to Curry (spent \$25), B stuck with Sushi (spent \$40).

## **6. Points Earned:**

- B earned 940 points, A got 860, and C had 360.

## **7. First Week Boost:**

- A reached 1,020 points, B earned 320 by end of January with double point offer.





## *Recommendations*

- **Promote Ramen more** – it's the most popular dish among all customers.
- **Offer targeted deals** to re-engage low-spending customers like Customer C.
- Highlight first-week double points in the **membership program to boost sign-ups**.
- **Encourage personalized offers** based on individual preferences (e.g., curry for B, ramen for A & C).
- **Use bonus points or loyalty rewards** to increase repeat visits and overall spend.

*Thank You*

