

## 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

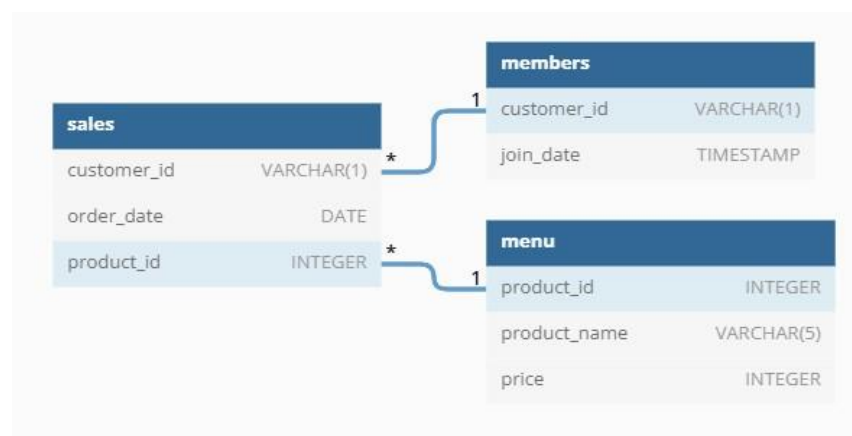
Danny wants to use the data to answer a few simple questions about his customers, especially about their visiting patterns, how much money they've spent and also which menu items are their favourite. Having this deeper connection with his customers will help him deliver a better and more personalised experience for his loyal customers. He plans on using these insights to help him decide whether he should expand the existing customer loyalty program.

## Datasets used

Three key datasets for this case study

- **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.
- **Menu:** The menu table maps the product\_id to the actual product\_name and price of each menu item.
- **Members:** The members table captures the join\_date when a customer\_id joined the beta version of the Danny's Diner loyalty program.

## Entity Relationship Diagram



## CASE STUDY ON THE DATASET

### 1. what is the total amount of each customer spend at the restuarant?

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

**output:**

customer_id	total_purchased_price
A	76
B	74
C	36

### 2. How many days each customer visited the restuarant?

```
select customer_id,  
  
count(distinct(order_date)) as days_visited  
  
from project.sales  
  
group by customer_id order by days_visited desc;
```

**output:**

customer_id	days_visited
B	6
A	4
C	2

### 3. what was the first item from the maximum purchased by each customer?

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

**output:**

customer_id	product_name
A	sushi
B	curry
C	ramen

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

```
select * from(
select m.product_name ,
row_number() over (partition by product_name ) as no_of_times_items_purchased
from project.sales as s join project.menu as m on s.product_id=m.product_id)as X
order by X.no_of_times_items_purchased desc limit 1;
```

**output:**

product_name	no_of_times_items_purchased
ramen	8

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

with most\_popular as

```
(select s.customer_id,m.product_name,
rank() over (partition by customer_id order by count(m.product_id) desc) as product_rank
from project.sales as s
join project.menu as m on s.product_id = m.product_id
group by customer_id,product_name)
```

```
select * from most_popular where product_rank= 1;
```

**output:**

customer_id	product_name	product_rank
A	ramen	1
B	curry	1
B	sushi	1
B	ramen	1
C	ramen	1

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

with first\_purchased as(

```
select w.product_name,w.join_date,w.order_date,w.customer_id,
rank() over(partition by join_date order by order_date) as rank_col
from
```

```

(select b.join_date,x.*
from
(select m.product_name,s.customer_id,s.order_date,s.product_id
from project.sales as s join project.menu as m
on s.product_id=m.product_id)as x
join project.members as b on b.customer_id=x.customer_id
order by order_date) as w
where w.order_date>=w.join_date)
select customer_id,product_name,order_date,join_date from first_purchased where rank_col=1;

```

**output:**

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

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

```

with first_purchased as(
select w.product_name,w.join_date,w.order_date,w.customer_id,
rank() over(partition by customer_id order by order_date desc ) as rank_col
from
(select b.join_date,x.*
from
(select m.product_name,s.customer_id,s.order_date,s.product_id
from project.sales as s join project.menu as m
on s.product_id=m.product_id)as x
join project.members as b on b.customer_id=x.customer_id
order by order_date) as w
where w.order_date<w.join_date)
select customer_id,product_name,order_date,join_date,rank_col from first_purchased where rank_col=1 ;

```

**output:**

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

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

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

**output:**

customer_id	total_items	amount_spent
A	2	25
B	3	40

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

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

**output:**

customer_id	member_points
A	860
B	940

**#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 jan_member_points as
(select m.customer_id as customer,
       sum(case when s.order_date < m.join_date then
```

```

        case
        when m2.product_name = 'sushi' then (m2.price * 20)
        else (m2.price * 10)
        end
    when s.order_date > (m.join_date + 6) then
        case
        when m2.product_name = 'sushi' then (m2.price * 20)
        else (m2.price * 10)
        end
    else (m2.price * 20)
end) as member_points

from project.members as m

join project.sales as s on s.customer_id = m.customer_id

join project.menu as m2 on s.product_id = m2.product_id

where s.order_date <= '2021-01-31'

group by customer)

select *

from jan_member_points

order by customer;

```

**output:**

customer	member_points
A	1370
B	820