8 week SQL challenge

Case Study 1

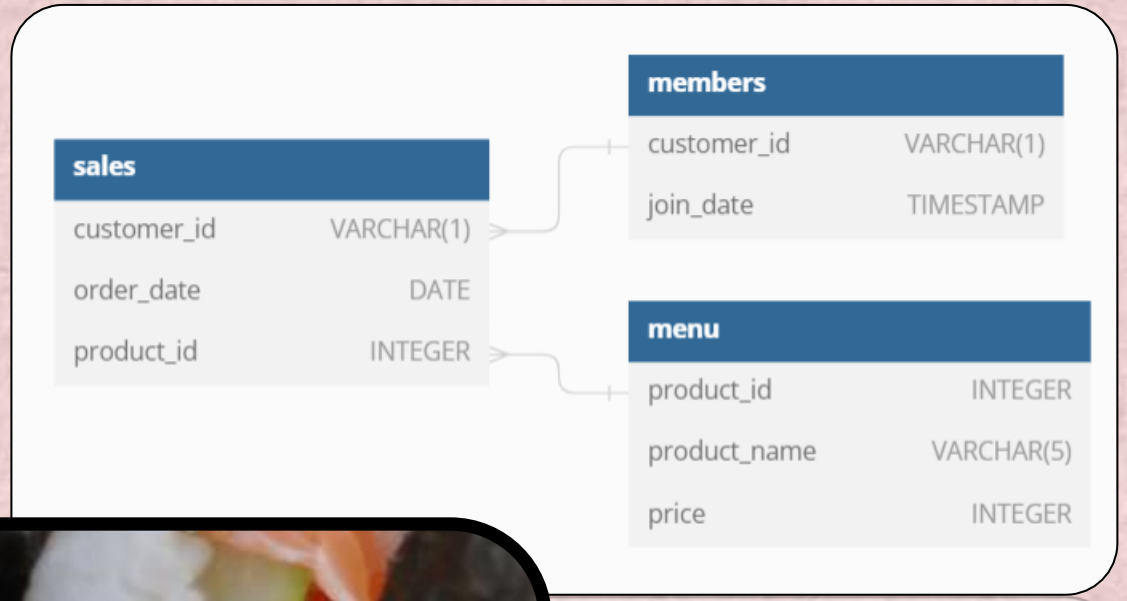
Danny’s Diner

- Harshavardhan Kumbhar

Problem Statement

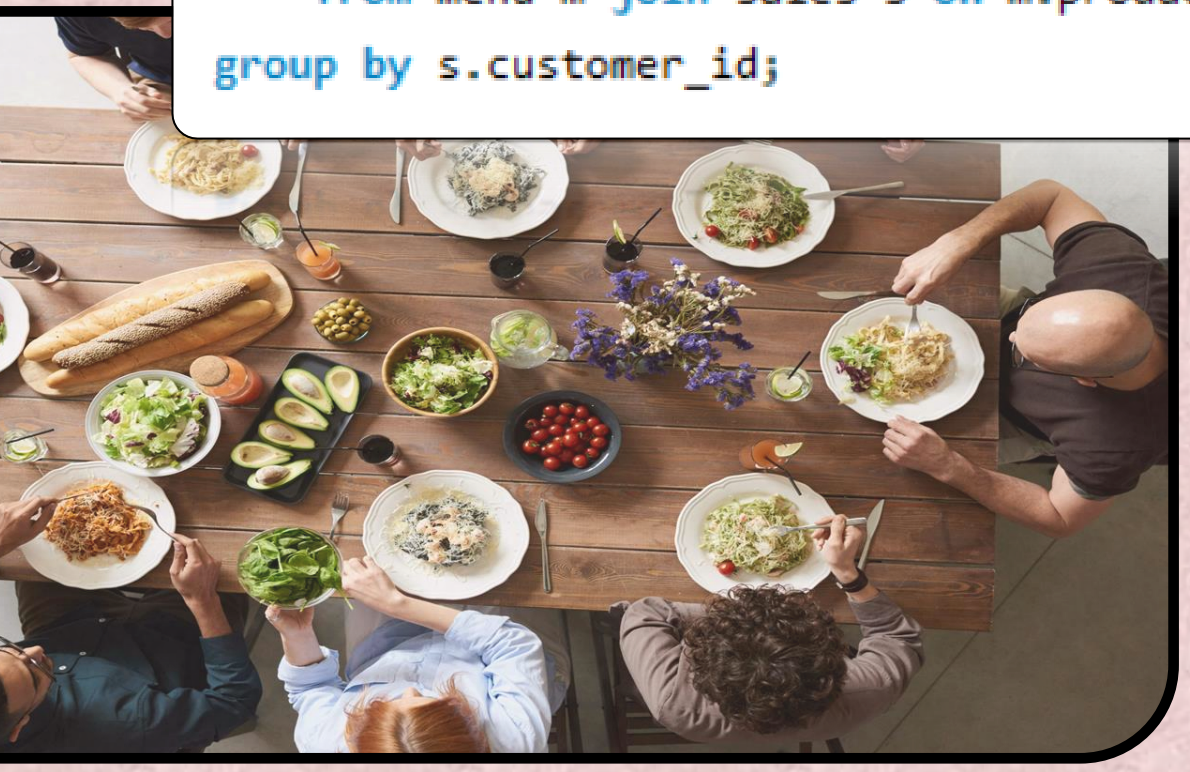
- Danny wants to use the data to answer a few simple questions about his customers, especially about their visiting patterns, how much money they've spent and also which menu items are their favorite.
- Having this deeper connection with his customers will help him deliver a better and more personalized experience for his loyal customers.

Entity Relationship Diagram



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

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



customer_id	total_sales
A	76
B	74
C	36

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

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

customer_id	No_of_days_visited
A	4
B	6
C	2

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

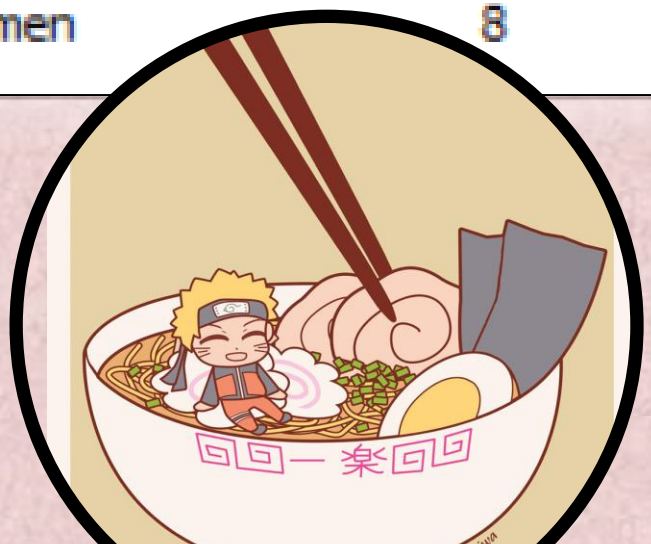
```
select s.customer_id , m.product_name
 from sales s join menu m on s.product_id = m.product_id
    where s.order_date = (select min(s.order_date) from sales s)
group by s.customer_id,m.product_name
order by s.customer_id;
```

customer_id	product_name
A	curry
A	sushi
B	curry
C	ramen

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

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

Most_purchased_item	Times_purchased
ramen	8



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

```sql
with popular_dish as (
    select s.customer_id,m.product_name,count(s.product_id) as order_count,
    dense_rank() over(partition by s.customer_id order by count(s.customer_id) desc) as rnk
    from menu m join sales s on m.product_id = s.product_id
    group by s.customer_id, m.product_name )
select customer_id, product_name,order_count from popular_dish
where rnk = 1;
```

customer_id	product_name	order_count
A	ramen	3
B	curry	2
B	sushi	2
B	ramen	2
C	ramen	3

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

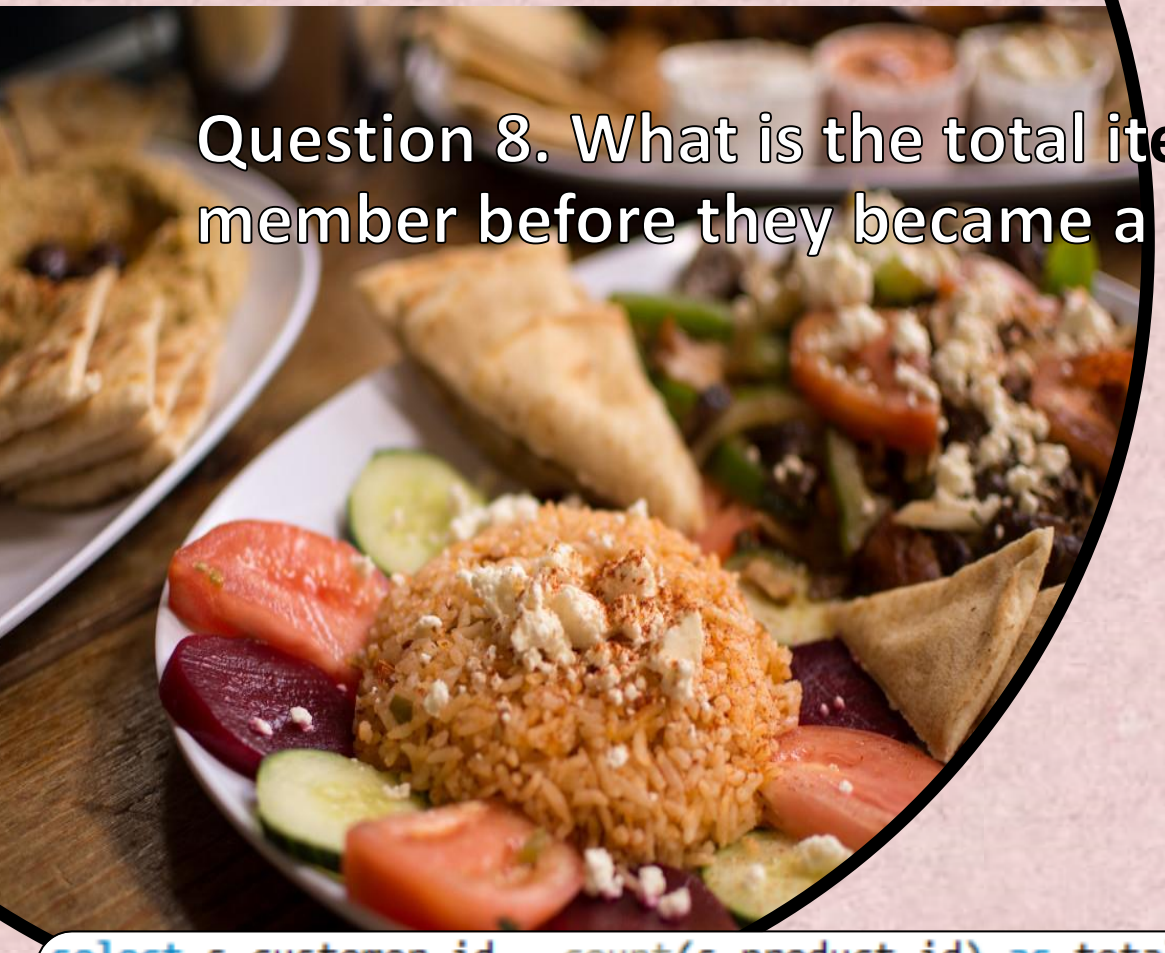
```sql
with joined_as_member as (
    select mm.customer_id , s.product_id ,
           row_number() over(partition by s.customer_id order by s.order_date) as rn
           from members mm join sales s on mm.customer_id = s.customer_id and s.order_date >= mm.join_date)
select customer_id, m.product_name
from joined_as_member jm join menu m on jm.product_id = m.product_id
where rn = 1
order by customer_id asc;
```

customer_id	product_name
A	curry
B	sushi

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

```sql
with joined_as_member as (
    select mm.customer_id , s.product_id ,
           row_number() over(partition by mm.customer_id order by s.order_date desc) as rn
           from members mm join sales s on mm.customer_id = s.customer_id and s.order_date < mm.join_date)
select jm.customer_id, m.product_name
from joined_as_member jm join menu m on jm.product_id = m.product_id
where rn = 1
order by jm.customer_id asc;
```

customer_id	product_name
A	sushi
B	sushi



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

	customer_id	total_item	amount
▶	A	2	25
	B	3	40

```sql
select s.customer_id , count(s.product_id) as total_item , sum(m.price) as amount
from sales s right join menu m on s.product_id = m.product_id
right join members mm on s.customer_id = mm.customer_id
where s.order_date < mm.join_date
group by s.customer_id
order by s.customer_id asc;
```

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

```sql
with points_table as (
    select * , case when product_id = 1 then price*20
                    else price*10
                        end as points
     from menu)
select  s.customer_id, sum(pt.points) as Total_points
from sales s join points_table pt on s.product_id = pt.product_id
group by s.customer_id;
```

customer_id	Total_points
A	860
B	940
C	360

Question 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?

```sql
with date_cte as (
    select * , date_add(join_date,interval 6 day) as valid_date,
               dayofmonth('2021-01-31') as last_date_jan
    from members
)
select s.customer_id ,
       sum( case when m.product_id = 1 then m.price * 20
                 when s.order_date between d.join_date and d.valid_date then m.price*20
                 else m.price*10
                 end) as points
from date_cte d join sales s
on d.customer_id = s.customer_id
join menu m
on m.product_id = s.product_id
where s.order_date <= '2021-01-31' and s.order_date >= d.join_date
group by s.customer_id
order by s.customer_id;
```

customer_id	points
A	1020
B	320

Bonus Query 1.

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.

```sql
select s.customer_id, s.order_date , m.product_name, m.price,
        if (s.order_date > mm.join_date, 'Y','N') as member
from sales s right join menu m
on s.product_id = m.product_id
join members mm
on mm.customer_id = s.customer_id
order by s.customer_id;
```

customer_id	order_date	product_name	price	member
A	2021-01-01	sushi	10	N
A	2021-01-07	curry	15	N
A	2021-01-01	curry	15	N
A	2021-01-11	ramen	12	Y
A	2021-01-11	ramen	12	Y
A	2021-01-10	ramen	12	Y
B	2021-01-11	sushi	10	Y
B	2021-01-04	sushi	10	N
B	2021-01-02	curry	15	N
B	2021-01-01	curry	15	N
B	2021-02-01	ramen	12	Y
B	2021-01-16	ramen	12	Y

Bonus Query 2.

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.

```sql
with DD_cte as (
    select s.customer_id, s.order_date , m.product_name, m.price,
        if (s.order_date > mm.join_date, 'Y','N') as member
from sales s right join menu m
on s.product_id = m.product_id
join members mm
on mm.customer_id = s.customer_id
order by s.customer_id)
select * ,
        if (member = 'N', null ,
            dense_rank() over(partition by s.customer_id,member order by s.order_date)) as ranking
from DD_cte;
```

customer_id	order_date	product_name	price	member	ranking
A	2021-01-01	sushi	10	N	NULL
A	2021-01-01	curry	15	N	NULL
A	2021-01-07	curry	15	N	NULL
A	2021-01-10	ramen	12	Y	1
A	2021-01-11	ramen	12	Y	2
A	2021-01-11	ramen	12	Y	2
B	2021-01-01	curry	15	N	NULL
B	2021-01-02	curry	15	N	NULL
B	2021-01-04	sushi	10	N	NULL
B	2021-01-11	sushi	10	Y	1
B	2021-01-16	ramen	12	Y	2
B	2021-02-01	ramen	12	Y	3

Thank You !

THE TASTE OF SUCCESS