

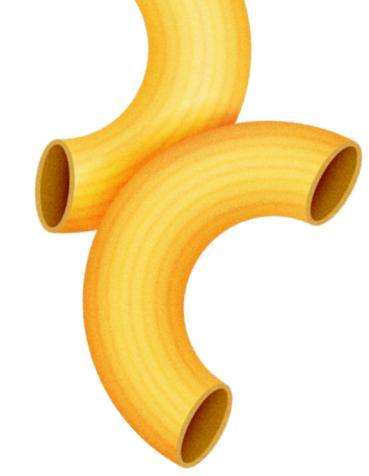
Danny's Dinner

The Taste of Success

SQL CASE STUDY #1



Table of Content

- About Us
 - Problem
 - Datasets
 - Entity Relationship Diagram
 - Finding Insights
 - Conclusion
- 

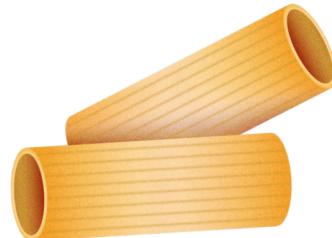
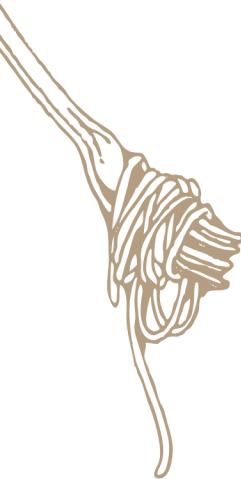




About Us

Danny seriously loves Japanese food so at 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 fundamental data from its few months of operation. Still, it has no idea how to use their data to help them run the business.



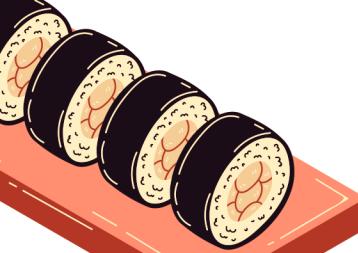
Problem

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!





Datasets

Danny has shared with you 3 key datasets for this case study:

1. **Sales**
2. **Menu**
3. **Members**

Members

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



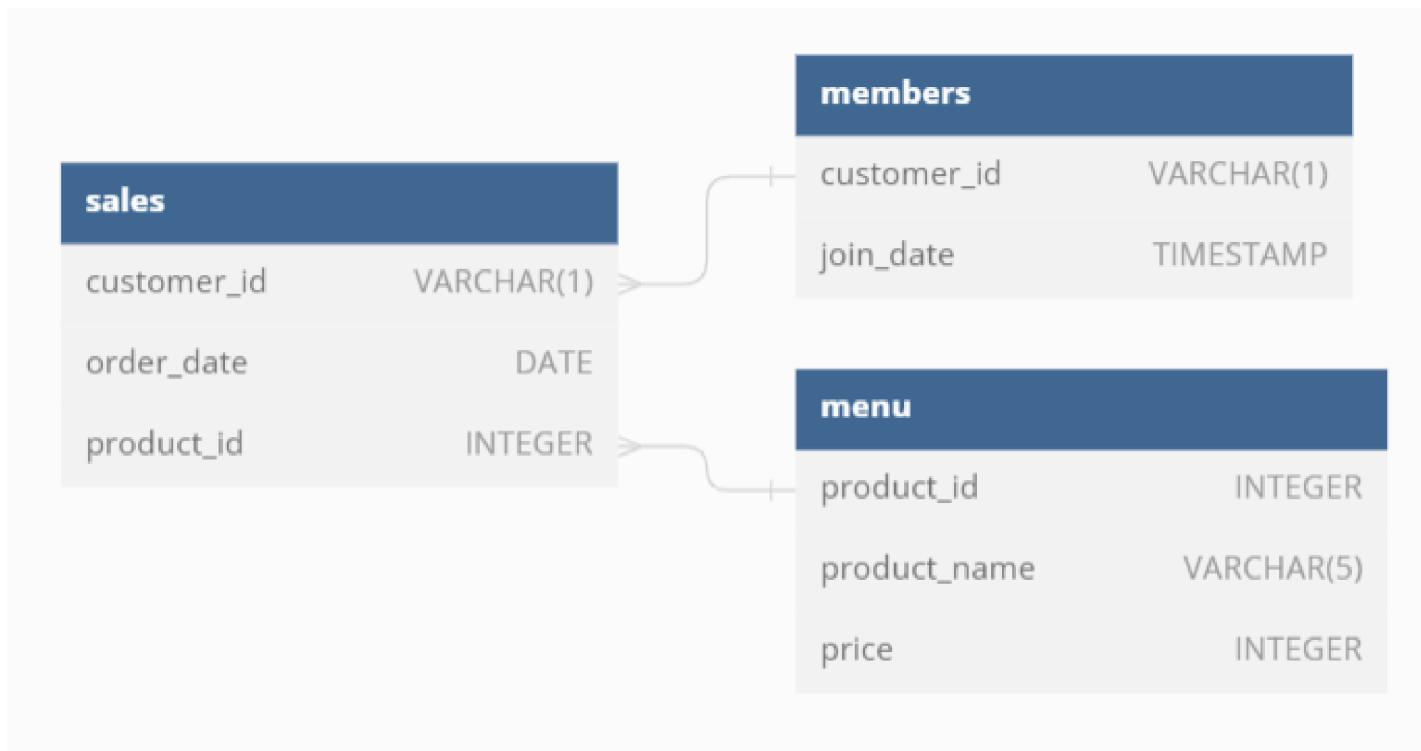
Menu

	product_id	product_name	price
▶	1	sushi	10
	2	curry	15
	3	ramen	12

Sales

	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

Entity Relationship Diagram





Finding Insights

by using **MySQL** Database Server





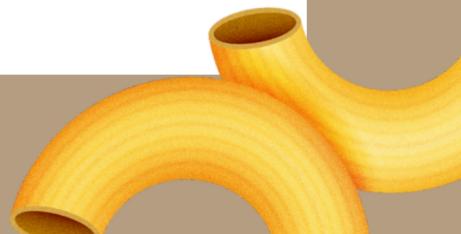
What is the total amount each customer spent at the restaurant?

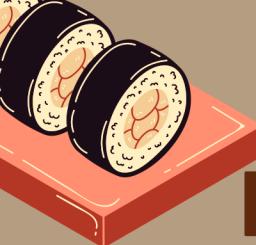
danny_dinner* x

1 • `select s.customer_id, sum(m.price) as total_amount_spent`
2 `from sales s`
3 `join menu m using(product_id)`
4 `group by 1;`
5
6
7

Result Grid | Filter Rows: Export: Wrap Cell Content:

	customer_id	total_amount_spent
▶	A	76
	B	74
	C	36





How many days has each customer visited the restaurant?

danny_dinner* ×

1 • `select s.customer_id, count(*) as no_of_days`
2 `from sales s`
3 `group by 1;`

4
5
6
7

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	customer_id	no_of_days
▶	A	6
	B	6
	C	3

Illustration of a yellow pasta noodle on the right side of the slide.



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



```
danny_dinner* x
with first_item as (
    select distinct s.customer_id, s.order_date, m.product_name,
    dense_rank() over(partition by customer_id order by order_date) as rnk
    from sales s
    join menu m using(product_id)
)
select customer_id, product_name as first_order
from first_item
where rnk = 1;
```

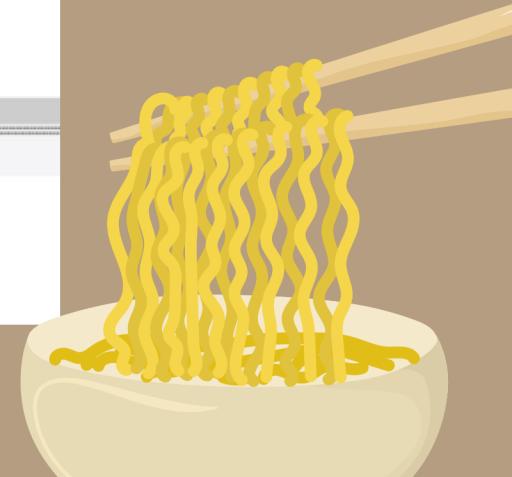
customer_id	first_order
A	sushi
A	curry
B	curry
C	ramen

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

```
danny_dinner* x
  ↳
111 • Ⓛ with productname as (
112     select customer_id, product_id, count(*) as times_purchased
113     from sales
114     where product_id = (
115         with item_name as (
116             select s.product_id, count(*) as purchased_items
117             from sales s
118             join menu m using(product_id)
119             group by 1 order by 2 desc limit 1)
120             select product_id
121             from item_name)
122         group by 1,2)
123     select p.customer_id, m.product_name, p.times_purchased
124     from productname p
125     left join menu m using(product_id);
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	customer_id	product_name	times_purchased
▶	A	ramen	3
	B	ramen	2
	C	ramen	3



Which item was the most popular for each customer?

danny_dinner* ×

File Edit View Insert Tools Window Help

1 • with popular as (

2 select s.customer_id, m.product_name, count(*) as items_ordered

3 from sales s

4 join menu m using(product_id)

5 group by 1, 2

6)

7 select customer_id, product_name

8 from popular

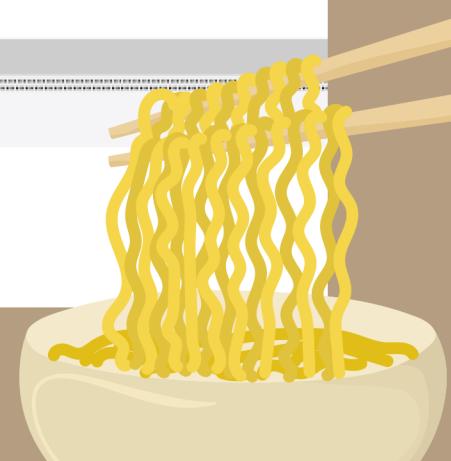
9 where items_ordered = (select max(items_ordered) from popular);

10

11

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	customer_id	product_name
▶	A	ramen
	C	ramen

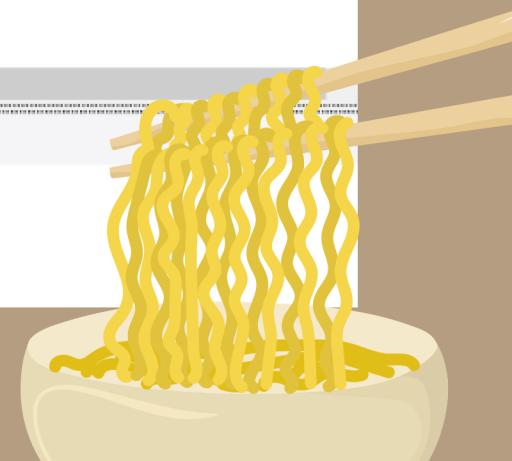


Which item was purchased first by the customer after they became a member?

```
danny_dinner* x
File Edit View Insert Cell Help
Limit to 1000 rows | Star | Print | Copy | Paste | Find | Replace | Refresh | Help
1 • with first_purchase as (
2     select s.customer_id, m.join_date, s.order_date, m1.product_name,
3             rank() over(partition by customer_id order by order_date asc) as rnk
4     from members m
5     left join sales s using(customer_id)
6     join menu m1 using(product_id)
7     where m.join_date < s.order_date
8 )
9     select customer_id, product_name, join_date, order_date
10    from first_purchase
11    where rnk = 1;
12
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	customer_id	product_name	join_date	order_date
▶	A	ramen	2021-01-07	2021-01-10
	B	sushi	2021-01-09	2021-01-11



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

danny_dinner* x

```
1 • with before_member as (
2     select s.customer_id, m.join_date, s.order_date, m1.product_name,
3         rank() over(partition by customer_id order by order_date desc) as rnk
4     from members m
5     left join sales s using(customer_id)
6     join menu m1 using(product_id)
7     where m.join_date > s.order_date
8 )
9 select customer_id, product_name, join_date, order_date
10 from before_member
11 where rnk = 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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



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

danny_dinner* x

```
1 •  select s.customer_id, count(*) as total_items, sum(price) as amount_spent
2   from members m
3   left join sales s using(customer_id)
4   join menu m1 using(product_id)
5   where m.join_date > s.order_date
6   group by 1 order by 1;
7
```

<

Result Grid | Filter Rows: Export: Wrap Cell Content:

	customer_id	total_items	amount_spent
▶	A	2	25
	B	3	40



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

danny_dinner* x

```
1 • with customer_points as (
2     select *, if(m.product_name = "sushi", m.price * 20, m.price * 10) as points
3     from sales s
4     join menu m using(product_id)
5 )
6     select customer_id, sum(points) as total_points
7     from customer_points
8     group by 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	total_points
A	860
B	940
C	360

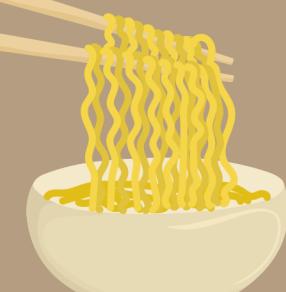


In the first week after a customer joins the program (including their join date), they earn 2x points on all items, not just sushi - how many points do customers A and B have at the end of January?

```
danny_dinner* x
File | New | Open | Save | Print | Help | Limit to 1000 rows | Star | Find | Replace | Refresh | 
1 • with cust_details as (
2     select distinct *, datediff(order_date, join_date) as days,
3             (m.price * 20) as points
4     from members m1
5     left join sales s using(customer_id)
6     join menu m using(product_id)
7     where m1.join_date <= s.order_date and s.order_date < '2021-01-31'
8     order by 2
9 )
10    select customer_id, sum(points) as total_points
11    from cust_details
12    where days <= 7
13    group by 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	total_points
A	780
B	440





Conclusion

- **Customer A** has made the *highest* total purchase amount at the restaurant.
- Both **customer A** and **customer B** have visited the restaurant for a total of *6 days* each.
- **Customers A, B, and C** ordered the *first item* as sushi, curry, and ramen, respectively.
- **Ramen** is the most frequently purchased item on the menu.
- **Customer A** and **Customer B**, after becoming members, purchased ramen and sushi as their *first items*, respectively.
- Before becoming a member, **customer A** spent \$25 on 2 items, while **customer B** spent \$40 on 3 items.
- **Customer B** has accumulated the *highest* number of points among all the customers.

Thank You

