

SQL

DATA ANALYSIS

PROJECT

8WEEKSQLCHALLENGE.COM
CASE STUDY #1



THE TASTE OF SUCCESS

DATAWITHDANNY.COM

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.

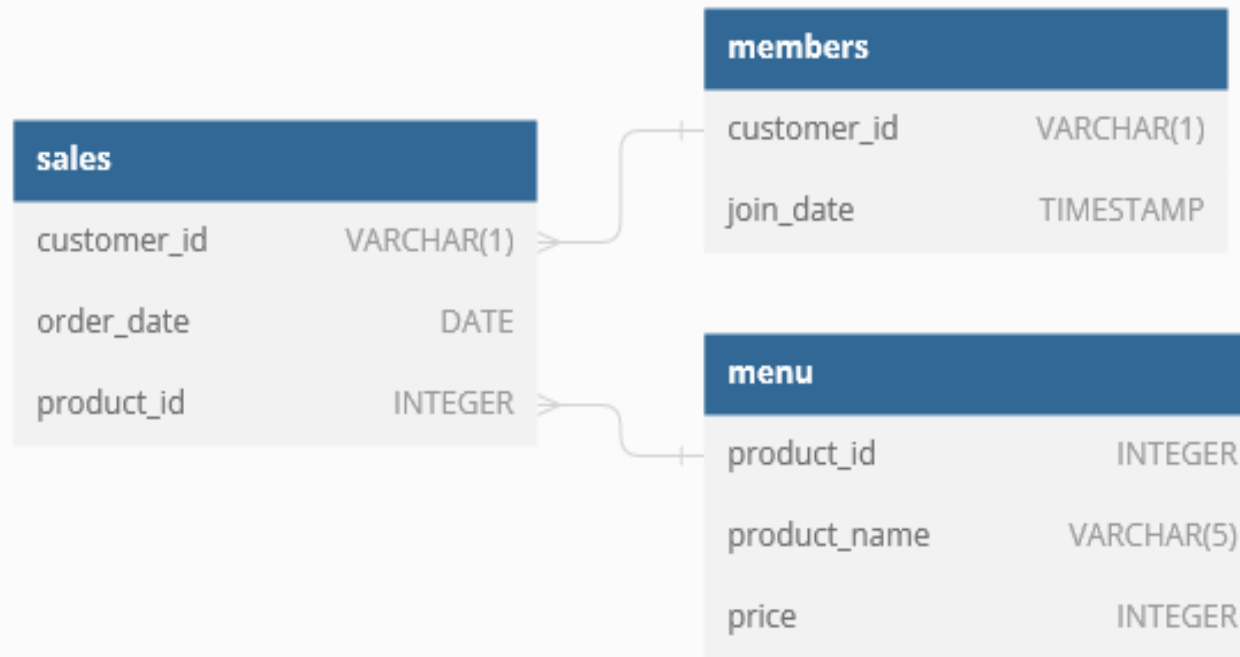
He plans on using these insights to help him decide whether he should expand the existing customer loyalty program - additionally he needs help to generate some basic datasets so his team can easily inspect the data without needing to use SQL.

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

ENTITY RELATIONSHIP DIAGRAM:



Case Study Questions:

- 1.What is the total amount each customer spent at the restaurant?
- 2.How many days has each customer visited the restaurant?
- 3.What was the first item from the menu purchased by each customer?
- 4.What is the most purchased item on the menu and how many times was it purchased by all customers?
- 5.Which item was the most popular for each customer?
- 6.Which item was purchased first by the customer after they became a member?
- 7.Which item was purchased just before the customer became a member?
- 8.What is the total items and amount spent for each member before they became a member?
- 9.If each \$1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?
- 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?

Q1.What is the total amount each customer spent at the restaurant?

```
1 • SELECT
2     customer_id, SUM(price) total_spent
3 FROM
4     sales a
5     INNER JOIN
6     menu b ON a.product_id = b.product_id
7 GROUP BY customer_id;
```

<

Result Grid



 Filter Rows:

Export: 

Wrap Cell Content: 

	customer_id	total_spent
▶	A	76
	B	74
	C	36

Q2. How many days has each customer visited the restaurant?

```
1 • SELECT
2     customer_id, COUNT(DISTINCT (order_date)) no_of_days
3 FROM
4     sales
5 GROUP BY customer_id;
```



Result Grid



Filter Rows:

Export:



Wrap Cell Content:

	customer_id	no_of_days
▶	A	4
	B	6
	C	2

Q3. What was the first item from the menu purchased by each customer?

```
1 • select customer_id,order_date,product_name from (  
2     select a.customer_id,a.order_date,b.product_name,  
3     rank() over(partition by customer_id order by order_date) first_order,  
4     row_number() over(partition by customer_id) as rn  
5     from sales a inner join menu b on a.product_id=b.product_id ) z  
6     where first_order = 1 and rn =1;
```

<

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



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

Q4. What is the most purchased item on the menu and how many times was it purchased by all customers?

```
1 • SELECT
2     product_name, COUNT(product_name) max_orders
3 FROM
4     sales a
5     INNER JOIN
6     menu b ON a.product_id = b.product_id
7 GROUP BY product_name
8 ORDER BY max_orders DESC
9 LIMIT 1;
```



Result Grid



Filter Rows:

Export:



Wrap Cell Content:





Fetch rows:



	product_name	max_orders
▶	ramen	8

Q5. Which item was the most popular for each customer?

```
1 • select customer_id,product_name from (  
2   select * from (  
3     select * ,  
4     rank() over(partition by customer_id order by popular_item desc) rnk  
5   from(  
6     select customer_id,product_name,count(product_name) popular_item from (  
7       select a.customer_id,b.product_name from sales a inner join menu b on a.product_id=b.product_id) z  
8     group by customer_id,product_name )x ) c  
9   where rnk =1 ) k ;
```

<		
Result Grid		
Filter Rows: <input type="text"/>		
Export:  Wrap Cell Content: 		
	customer_id	product_name
▶	A	ramen
	B	curry
	B	sushi
	B	ramen
	C	ramen

Q6. Which item was purchased first by the customer after they became a member?

```
1 • select customer_id,product_name,order_date,join_date from (  
2   select *,  
3   rank() over (partition by customer_id order by order_date) rnk  
4   from(  
5   select * from (  
6   select b.customer_id,a.join_date,b.order_date,b.product_id,c.product_name from members a left join sales b on a.customer_id=b.customer_id  
7   inner join menu c on b.product_id = c.product_id) z  
8   where join_date <=order_date ) x) c  
9   where rnk =1;
```

Result Grid				
Filter Rows: <input type="text"/>				
Export:  Wrap Cell Content: 				
	customer_id	product_name	order_date	join_date
▶	A	curry	2021-01-07	2021-01-07
	B	sushi	2021-01-11	2021-01-09

Q7. Which item was purchased just before the customer became a member?




```
1 • select customer_id,product_name from (  
2   select *,  
3   rank() over(partition by customer_id order by order_date desc) first_order_before_joining  
4   from (  
5   select * from (  
6   select a.customer_id,a.join_date,b.order_date,b.product_id,c.product_name from members a left join sales b on a.customer_id=b.customer_id  
7   inner join menu c on b.product_id=c.product_id ) z  
8   where order_date < join_date ) x) c  
9   where first_order_before_joining =1;
```

<		
Result Grid		
Filter Rows: <input type="text"/>		
Export:  Wrap Cell Content: 		
	customer_id	product_name
▶	A	sushi
	A	curry
	B	sushi

Q8. What is the total items and amount spent for each member before they became a member?

```
1 • SELECT
2     a.customer_id, COUNT(product_name) total_items, SUM(price) amount_spent
3 FROM
4     members a
5     LEFT JOIN
6     sales b ON a.customer_id = b.customer_id
7     INNER JOIN
8     menu c ON b.product_id = c.product_id
9 WHERE
10    order_date < join_date
```

<

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

	customer_id	total_items	amount_spent
▶	B	3	40
	A	2	25

Q9. If each \$1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?

```
1 • SELECT
2     customer_id, SUM(points) points
3 FROM
4     (SELECT
5         a.customer_id,
6         b.product_id,
7         b.product_name,
8         b.price,
9         CASE
10            WHEN product_name = 'sushi' THEN price * 20
11            ELSE price * 10
12        END AS points
13     FROM
14         sales a
15     INNER JOIN menu b ON a.product_id = b.product_id) z
16 GROUP BY customer_id;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	customer_id	points		
▶	A	860		
	B	940		
	C	360		

Q10. 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?

```
1 • SELECT customer_id, SUM(new_price) * 10 as points FROM
2   (SELECT a.customer_id, order_date, product_name, price,
3     CASE
4       WHEN product_name = 'sushi' THEN 2 * price
5       WHEN order_date BETWEEN join_date AND (join_date + INTERVAL 6 DAY) THEN 2 * price
6       ELSE price
7     END AS new_price
8   FROM
9     sales a
10    JOIN menu b ON a.product_id = b.product_id
11    JOIN members c ON a.customer_id = c.customer_id
12   WHERE
13     order_date <= '2021-01-31') z
14  GROUP BY customer_id;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	customer_id	points			
▶	B	820			
	A	1370			

THANK YOU