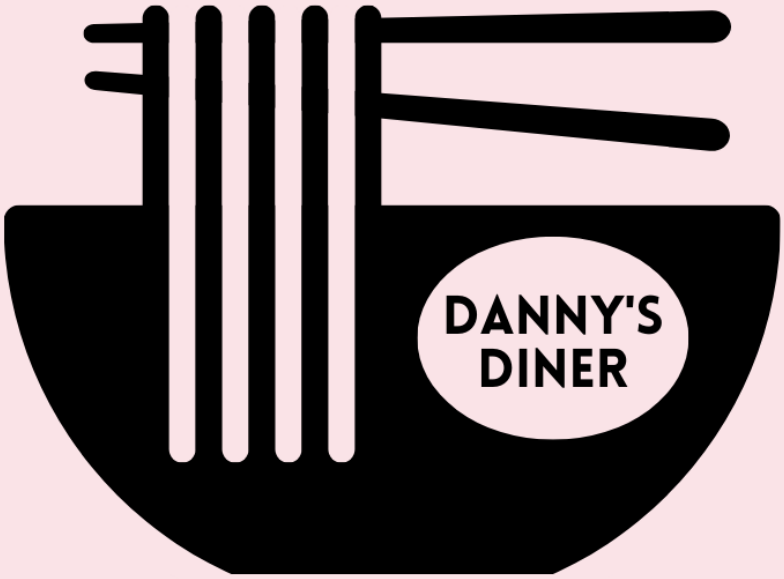
Case study 1: Danny's Diner

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.

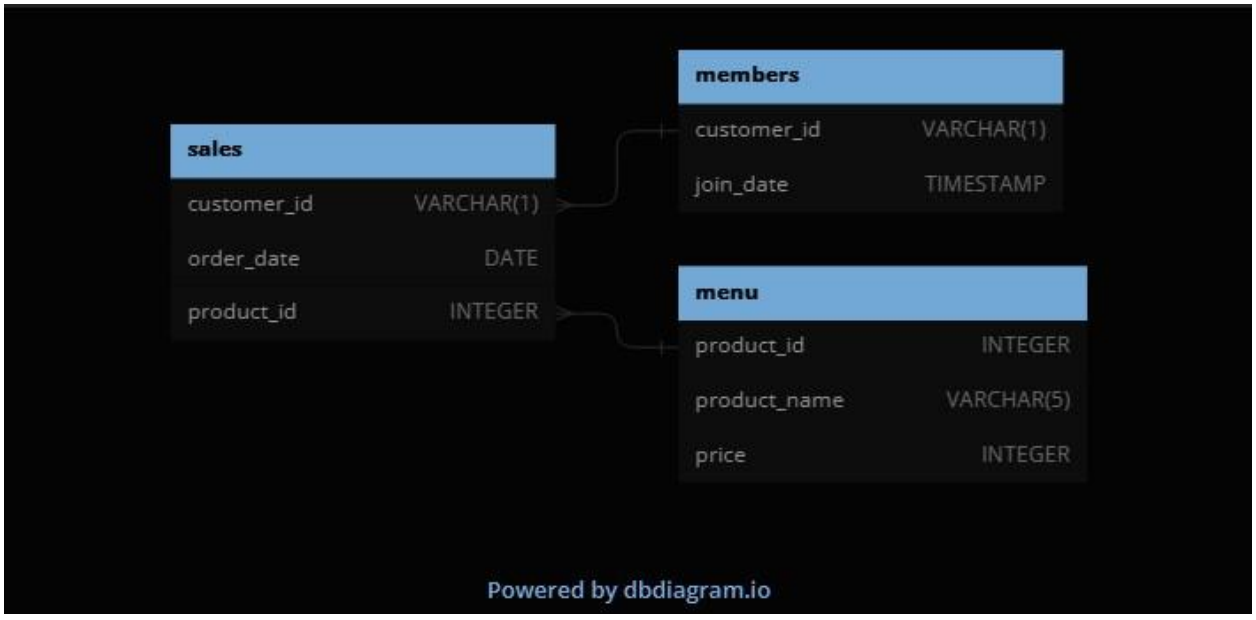
Danny has provided you with a sample of his overall customer data due to privacy issues - but he hopes that these examples are enough for you to write fully functioning SQL queries to help him answer his questions!

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

- sales
- menu
- members

You can inspect the entity relationship diagram and example data below.

Entity Relationship Diagram



Example Datasets

All datasets exist within the dannys_diner database schema - be sure to include this reference within your SQL scripts as you start exploring the data and answering the case study questions.

Table 1: 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

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

Table 2: menu

The menu table maps the product_id to the actual product_name and price of each

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

Table 3: members

The final members table captures the join_date when a customer_id joined the beta version of the Danny's Diner loyalty program.

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

Case Study Questions

Each of the following case study questions can be answered using a single SQL statement:

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

```
61    -- 1. What is the total amount each customer spent at the restaurant?
62    select s.customer_id as Customer,sum(m.price) as Total_amount_spent
63    from sales s JOIN menu m
64    on m.product_id = s.product_id
65    group by 1;
66
```

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

Customer	Total_amount_spent
A	76
B	74
C	36

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

```
57    -- 2. How many days has each customer visited the restaurant?
58    select customer_id as Customer, count(distinct order_date) as day_visited from sales group by 1 ;
59
70    -- 3. What was the first item from the menu purchased by each customer?
```

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

Customer	day_visited
A	4
B	6
C	2

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

```
70      -- 3. What was the first item from the menu purchased by each customer?
71 •    select s.customer_id as Customer, m.product_name as first_item_purchsed
72      from sales s join menu m
73      on s.product_id = m.product_id group by 1;
74
```

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

Customer	first_item_purchsed
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?**

```
75      -- 4. What is the most purchased item on the menu and how many times was it purchased by all customers?
76 •    select m.product_name, count(*) as most_purchased_item from sales s
77      join menu m on s.product_id = m.product_id
78      group by 1
79      order by 2 desc limit 1;
80
81      # for each item how many time purchased
```

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

product_name	most_purchased_item
ramen	8

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

```
86      -- 5. Which item was the most popular for each customer?
87 •    select s.customer_id as customer, m.product_name as most_popular_item,count(*) as total_item
88      from sales s join menu m
89      on s.product_id = m.product_id group by 1;
90
91      -- 6. Which item was purchased first by the customer after they became a member?
```

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

customer	most_popular_item	total_item
A	sushi	6
B	curry	6
C	ramen	3

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

```
91    -- 6. Which item was purchased first by the customer after they became a member?
92    select s.customer_id as customer,m.product_name as item_purchased_first,m1.join_date,s.order_date from sales s
93    join menu m on s.product_id = m.product_id
94    join members m1 on s.customer_id = m1.customer_id
95    where s.order_date = m1.join_date or s.order_date > m1.join_date
96    group by 1
97    order by 2 ;
98
```

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

customer	item_purchased_first	join_date	order_date
A	curry	2021-01-07	2021-01-07
B	sushi	2021-01-09	2021-01-11

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

```
99    -- 7. Which item was purchased just before the customer became a member?
100   select s.customer_id as customer,m.product_name as item_purchased_before_member,m1.join_date,
101   s.order_date from sales s
102   join menu m on s.product_id = m.product_id
103   join members m1 on s.customer_id = m1.customer_id
104   where s.order_date = m1.join_date or s.order_date < m1.join_date
105   group by 1
106   order by 1 ;
107
```

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

customer	item_purchased_before_member	join_date	order_date
A	sushi	2021-01-07	2021-01-01
B	sushi	2021-01-09	2021-01-04

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

```
108   -- 8. What is the total items and amount spent for each member before they became a member?
109   select s.customer_id as customer,count(*) as total_item,
110   sum(m.price) as Total_amount_spent,m1.join_date,s.order_date from sales s
111   join menu m on s.product_id = m.product_id
112   join members m1 on s.customer_id = m1.customer_id
113   where s.order_date = m1.join_date or s.order_date > m1.join_date
114   group by 1
115   order by 1 ;
116
```

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

customer	total_item	Total_amount_spent	join_date	order_date
A	4	51	2021-01-07	2021-01-07
B	3	34	2021-01-09	2021-01-11

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

```
/* 9. If each $1 spent equates to 10 points and sushi has a 2x points multiplier
 how many points would each customer have ?*/
select sub.customer,sum(points) as Total_points
from
(select s.customer_id as customer,
case
when m.product_name ='sushi' then 2*10*m.price
else 10*m.price
end as points
from sales s left join menu m on s.product_id = m.product_id) sub
group by 1;
```

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

customer	Total_points
A	860
B	940
C	360

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?**

```
/* 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? */
 select s.customer_id ,
 sum(case
    when s.order_date between m1.join_date and date_add(m1.join_date,interval 6 day)
    then m.price*2*10 when m.product_name = 'sushi' then m.price*2*10
    else m.price*10
    end) as total_points from sales s
 join menu m on m.product_id = s.product_id
 join members m1 on m1.customer_id =s.customer_id
 where date_format(s.order_date,'%Y-%m-01')='2021-01-01'
 group by 1 order by 1;
```

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

customer_id	total_points
A	1370
B	820

Thank You !

Swati Khedekar