Case Study #1 - Danny's Diner

Introduction

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 very basic data from its few months of operation but has 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 favorite. Having this deeper connection with his customers will help him deliver a better and more personalized experience for his loyal customers.

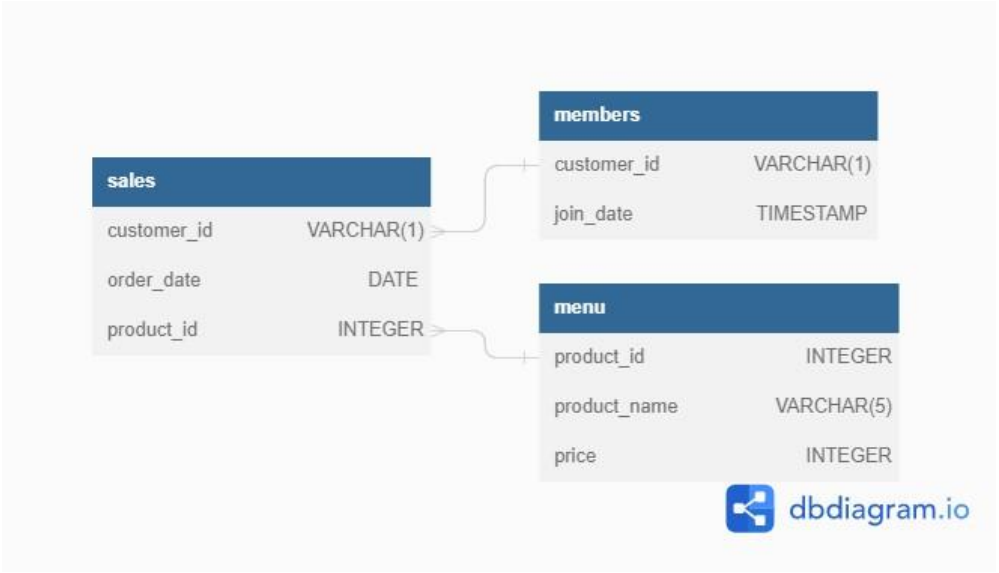
He plans on using these insights to help him decide whether he should expand the existing customer loyalty program - additionally, he needs help to generate some basic datasets so his team can easily inspect the data without needing to use SQL.

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!

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.



Case Study Questions

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

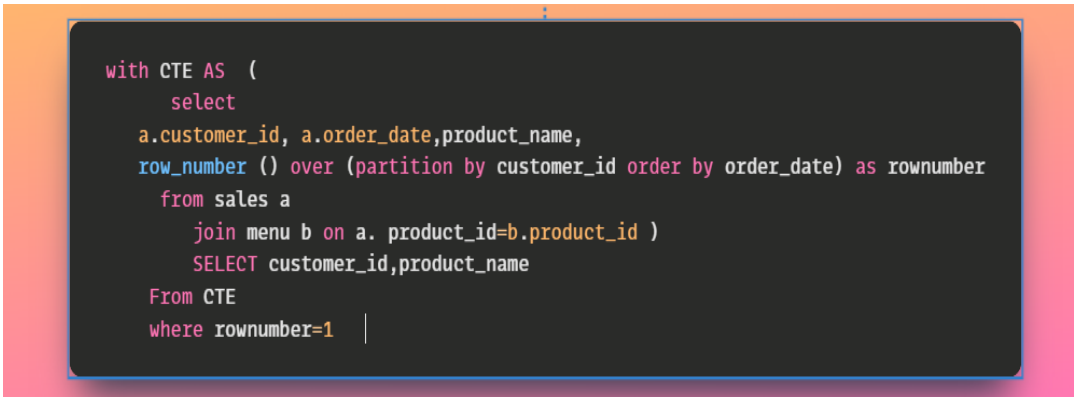
1. What is the total amount each customer spent at the restaurant?

```
select
      a.customer_id ,sum(b.price) total_amount_spent
      from sales a
      join  menu b on a.product_id=b.product_id
    group by a.customer_id
    order by total_amount_spent desc
```

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

```
select
      customer_id ,count(distinct order_date) no_of_days
      from sales
      group by customer_id
```

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



```
with CTE AS  (
      select
   a.customer_id, a.order_date,product_name,
   row_number () over (partition by customer_id order by order_date) as rownumber
     from sales a
        join menu b on a. product_id=b.product_id )
        SELECT customer_id,product_name
    From CTE
    where rownumber=1
```

4. What is the most purchased item on the menu and how many times did all customers purchase it?

```
--- most purchased item on the menu
 select
     top 1 b.product_name,count( a.product_id ) most_purchased_item
   from sales a
     join menu b on a.product_id=b.product_id
     group by b.product_name
     order by most_purchased_item desc

 ---many times was it purchased by all customers
  select
     a.customer_id,count( a.product_id )purchased_by_all_customer
   from sales a
     join menu b on a.product_id=b.product_id
   where a.product_id=3
   group by a.customer_id;
```

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

```
with CTE AS(
        select
        a.customer_id ,b.product_name ,count(a.product_id) popular_item ,
        rank() over(partition by a.customer_id order by count(a.product_id) desc)as rnk
        from sales a
       join menu b on a.product_id=b.product_id
       group by a.customer_id ,b.product_name
        )
        select customer_id,product_name,popular_item
         from CTE
         where rnk =1
```

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

```
with CTE AS (
  select
    a.customer_id,a.order_date,a.product_id ,b.join_date ,c.product_name,
    rank() over (partition by a.customer_id order by order_date ) as rnk
    from sales a
    join members b on a.customer_id=b.customer_id
    join menu c on a.product_id=c.product_id
    where order_date>join_date)
  select customer_id ,product_name
   from CTE
   where rnk=1
```

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

```
WITH CTE AS (
   select
     a.customer_id,a.order_date,a.product_id ,b.join_date ,c.product_name,
     row_number() over (partition by a.customer_id order by order_date desc ) as rnk
    from sales a
    join members b on a.customer_id=b.customer_id
    join menu c on a.product_id=c.product_id
    where order_date<join_date)
select customer_id, product_name
   from CTE
   where rnk=1
```

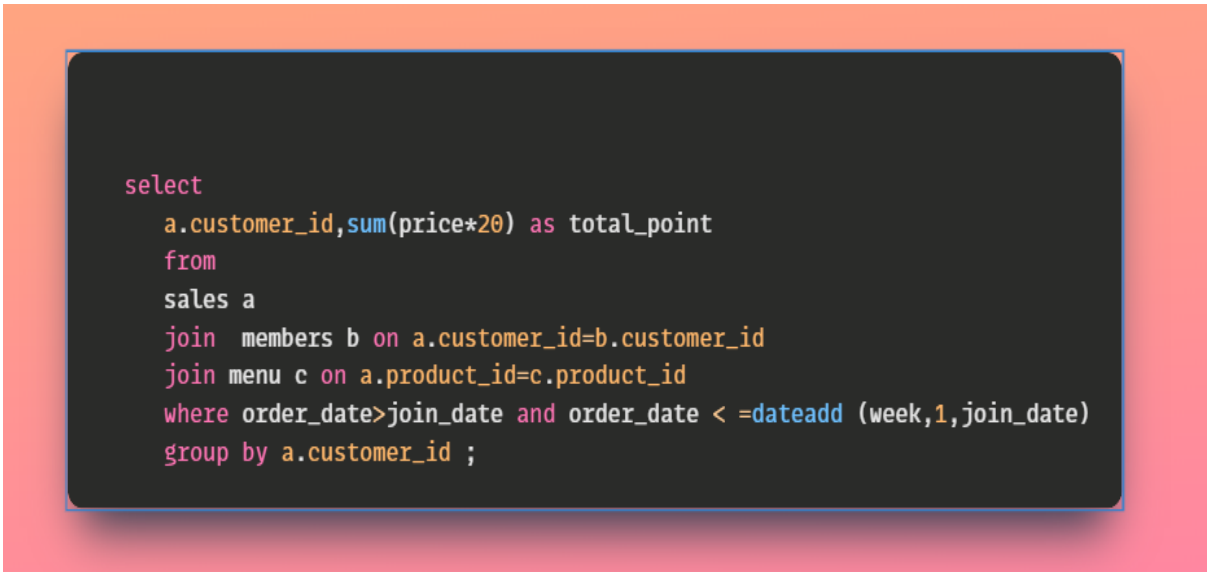
8. What are the total items and amount spent for each member before they became a member?

```
with CTE AS (
    select
     a.customer_id,a.order_date,a.product_id ,b.join_date ,c.product_name,c.price
    from sales a
     join members b on a.customer_id=b.customer_id
     join menu c on a.product_id=c.product_id
     where order_date<join_date)
   select customer_id,count( product_id) total_item ,sum(price) total_sales
   from CTE
   group by customer_id;
```

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

```
with CTE AS (
        select
         a.*,b.price ,case when a.product_id=1 then price*20 when a.product_id=2 then price*10
         when a.product_id=3 then price*10  else 0 end as points from
         sales a join menu b on a.product_id=b.product_id)
 select customer_id, sum( points) as total_points
    from CTE
    group by customer_id order by total_points desc
```

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 customers A and B have at the end of January?



```
    select
       a.customer_id,sum(price*20) as total_point
       from
       sales a
       join  members b on a.customer_id=b.customer_id
       join menu c on a.product_id=c.product_id
       where order_date>join_date and order_date < =dateadd (week,1,join_date)
       group by a.customer_id ;
```

11.Recreate the table with: customer_id, order_date, product_name, price, member (Y/N) ?

```
with CTE AS (
     select
       a.customer_id ,a.order_date,b.product_name,b.price,c.join_date,
       case when a.order_date>= c.join_date then 'Y' when a.order_date< c.join_date then 'N'
        else 'N' end as member
      from sales a
      join menu b on a.product_id=b.product_id
      left join  members c on a.customer_id=c.customer_id)
 select customer_id,order_date,product_name,member
     from CTE
```