## **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.

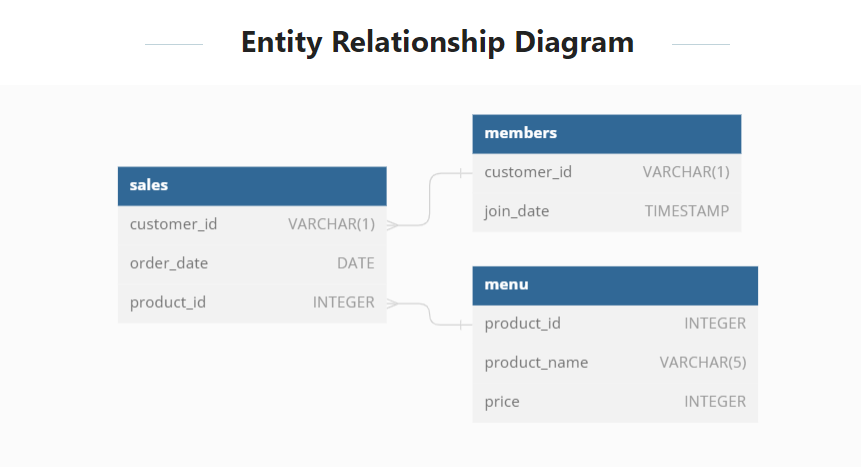
Danny has provided you with a sample of his overall customer data due to privacy issues - but he hopes that these examples are enough for you to write fully functioning SQL queries to help him answer his questions!

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.

**Case Study #1 - Danny's Diner**



**CREATE** **TABLE** sales (

"customer\_id" VARCHAR(1),

"order\_date" DATE,

"product\_id" INTEGER

);

**INSERT** **INTO** sales

("customer\_id", "order\_date", "product\_id")

**VALUES**

('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');

**CREATE** **TABLE** menu (

"product\_id" INTEGER,

"product\_name" VARCHAR(5),

"price" INTEGER

);

**INSERT** **INTO** menu

("product\_id", "product\_name", "price")

**VALUES**

('1', 'sushi', '10'),

('2', 'curry', '15'),

('3', 'ramen', '12');

**CREATE** **TABLE** members (

"customer\_id" VARCHAR(1),

"join\_date" DATE

);

**INSERT** **INTO** members

("customer\_id", "join\_date")

**VALUES**

('A', '2021-01-07'),

('B', '2021-01-09');

## **Case Study Questions**

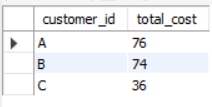
Each of the following case study questions can be answered using a single SQL statement:

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

select s.customer\_id,sum(m.price) as total\_cost from

sales s join menu m on s.product\_id=m.product\_id

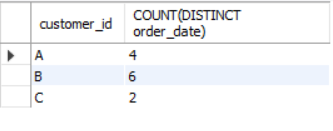
group by 1;



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

Select customer\_id,COUNT(DISTINCT order\_date)

FROM SALES group by 1 ;



**OUTPUT:-**

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

WITH CTE AS

(select s.customer\_id , s.order\_date,m.product\_name,

row\_number() over(partition by s.customer\_id ORDER BY s.order\_date) as ab

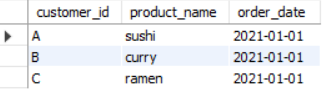
from SALES s join menu m on s.product\_id=m.product\_id)

SELECT customer\_id, product\_name,order\_date

FROM CTE

where ab = 1;

**OUTPUT:-**



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

SELECT

m.product\_name,

COUNT(s.product\_id) AS most\_purchased\_item

FROM sales s

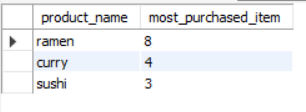
JOIN menu m

ON s.product\_id = m.product\_id

GROUP BY 1

ORDER BY 2 DESC;

**OUTPUT:-**



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

WITH CTE AS

(SELECT s.customer\_id,COUNT(m.product\_name) AS A,m.product\_name,DENSE\_RANK () OVER(PARTITION BY s.customer\_id ORDER BY COUNT(m.product\_name)DESC) AS B

FROM sales s

JOIN menu m ON s.product\_id=m.product\_id

GROUP BY 1,3)

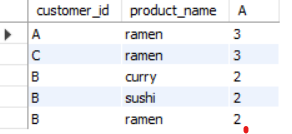
SELECT customer\_id,product\_name,A

FROM CTE

WHERE B=1

ORDER BY 3 DESC;

**OUTPUT:-**



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

WITH CTE AS

(SELECT a.customer\_id,b.join\_date,a.order\_date,c.product\_name,DENSE\_RANK() OVER(PARTITION BY a.customer\_id ORDER BY a.order\_date) AS AB

FROM sales a

JOIN members b ON a.customer\_id=b.customer\_id

JOIN menu c ON a.product\_id=c.product\_id

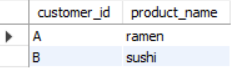
WHERE a.order\_date>b.join\_date)

SELECT customer\_id,product\_name

FROM CTE

WHERE AB=1;

**OUTPUT:-**



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

WITH CTE AS

(SELECT a.customer\_id,b.join\_date,a.order\_date,c.product\_name,DENSE\_RANK() OVER(PARTITION BY a.customer\_id ORDER BY a.order\_date desc) AS AB

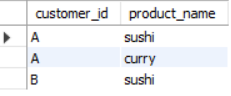
FROM sales a

JOIN members b ON a.customer\_id=b.customer\_id

JOIN menu c ON a.product\_id=c.product\_id

WHERE a.order\_date<b.join\_date)

SELECT customer\_id,product\_name **OUTPUT:-**

FROM CTE

WHERE AB=1;

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

SELECT a.customer\_id,

count(c.product\_id) as total\_items,

sum(c.price) as total\_sales

FROM sales a

JOIN members b ON a.customer\_id=b.customer\_id

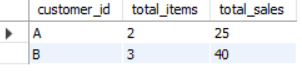
JOIN menu c ON a.product\_id=c.product\_id

WHERE a.order\_date<b.join\_date

GROUP BY 1

ORDER BY 1;

**OUTPUT:-**



1. **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 product\_id,

CASE

WHEN product\_id=1 THEN price \* 20

ELSE price \* 10

END AS AB

FROM menu)

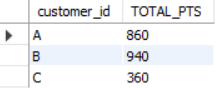
SELECT a.customer\_id,SUM(AB) AS TOTAL\_PTS

FROM CTE

JOIN sales a ON CTE.product\_id=a.product\_id

GROUP BY 1;

**OUTPUT:-**



1. **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?**

WITH CTE AS

(SELECT a.customer\_id,a.order\_date,b.product\_id,

CASE

WHEN a.order\_date - c.join\_date >=0 AND a.order\_date - c.join\_date <=6 THEN price \* 20

WHEN b.product\_id = 1 THEN price \* 20

ELSE price \* 10

END AS AB

FROM sales a

JOIN menu b ON a.product\_id=b.product\_id

JOIN members c ON a.customer\_id=c.customer\_id

WHERE a.order\_date<'2021-02-01')

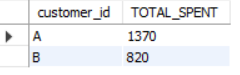
SELECT customer\_id,SUM(AB) AS TOTAL\_SPENT

FROM CTE

GROUP BY 1

ORDER BY 1;

**OUTPUT:-**



**Bonus Questions**

**Join All The Things**

**SELECT a.customer\_id, a.order\_date,b.product\_name, b.price,**

**CASE**

**WHEN c.join\_date > a.order\_date THEN 'N'**

**WHEN c.join\_date <= a.order\_date THEN 'Y'**

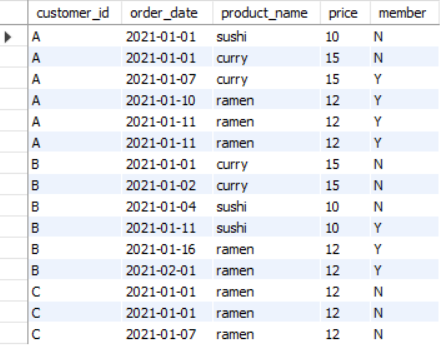
**ELSE 'N' END AS member**

**FROM sales a LEFT JOIN members c ON a.customer\_id = c.customer\_id**

**JOIN menu b ON a.product\_id = b.product\_id**

**ORDER BY 1,2;**

**OUTPUT:-**



## **Rank All The Things**

WITH CTE AS (

SELECT a.customer\_id, a.order\_date, b.product\_name, b.price,

CASE

WHEN c.join\_date > a.order\_date THEN 'N'

WHEN c.join\_date <= a.order\_date THEN 'Y'

ELSE 'N' END AS memberr

FROM sales a

LEFT JOIN members c

ON a.customer\_id = c.customer\_id

JOIN menu b

ON a.product\_id = b.product\_id

ORDER BY 1,2 )

SELECT customer\_id,order\_date,product\_name,price,memberr,

CASE

WHEN memberr = 'N' then NULL

ELSE RANK () OVER(

PARTITION BY customer\_id, memberr

ORDER BY order\_date) END AS rankk

FROM CTE;

**OUTPUT :-**

