

DANNY'S DINNER

Sathwika Reddy

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

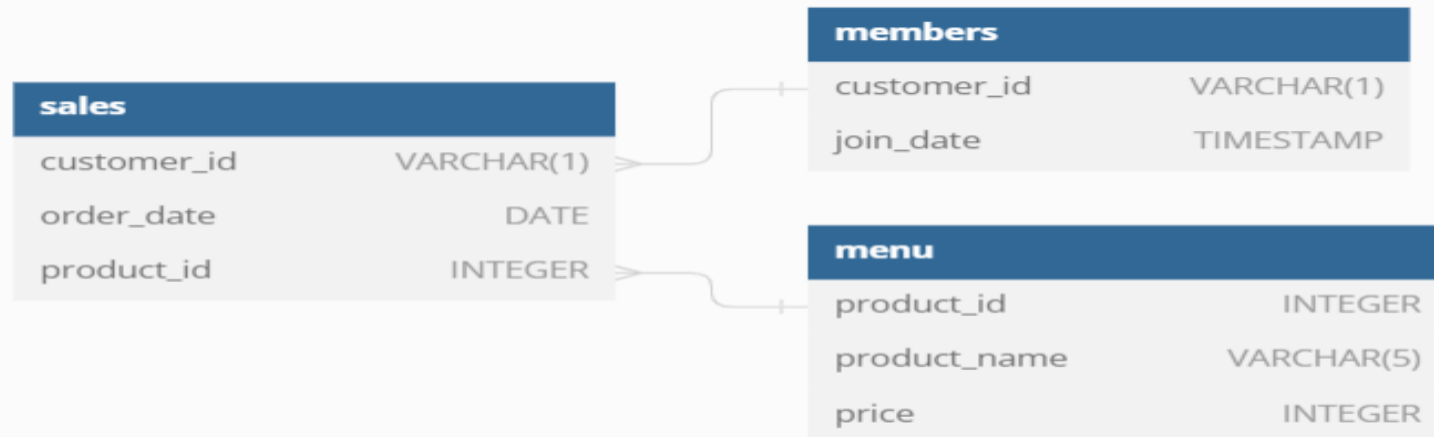
- Danny wants to use the data to answer a few simple questions about his customers, especially about their visiting patterns, how much money they've spent and also which menu items are their favourite.
- Having this deeper connection with his customers will help him deliver a better and more personalised experience for his loyal customers.

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



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

```
select customer_id, sum(price) as total_amount
from sales
join menu
on sales.product_id=menu.product_id
group by customer_id;
```

	customer_id	total_amount
▶	A	76
	B	74
	C	36

How many days has each customer visited the restaurant?

```
select customer_id, count(distinct order_date) as visit_count  
from sales  
group by customer_id;
```

	customer_id	visit_count
▶	A	4
	B	6
	C	2

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

```
• With cte as
(
  Select S.customer_id,
        M.product_name,
        S.order_date,
        DENSE_RANK() OVER (PARTITION BY S.Customer_ID Order by S.order_date) as rn
  From Menu m
  join Sales s
  On m.product_id = s.product_id
  group by S.customer_id, M.product_name,S.order_date
)
Select Customer_id, product_name
From cte
Where rn = 1;
```

	Customer_id	product_name
▶	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?

```
select M.product_name,count(M.product_name) as most_purchased
from sales S
join menu M
on S.product_id=M.product_id
group by M.product_name
order by most_purchased desc
limit 1;
```

	product_name	most_purchased
▶	ramen	8

Which item was the most popular for each customer?

```
• with cte as
(
  select S.customer_id,
  M.product_name,
  COUNT(S.product_id) as count,
  dense_rank() over(partition by S.customer_id order by count(S.product_id) desc) as rn
  from menu M
  join sales S
  on M.product_id=S.product_id
  group by S.customer_id,S.product_id,M.product_name
)
select customer_id,product_name
from cte
where rn=1;
```

	customer_id	product_name
▶	A	ramen
	B	curry
	B	sushi
	B	ramen
	C	ramen

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

```
with orders as(  
  select S.customer_id,M.product_name,S.order_date,Mem.join_date,  
         dense_rank() over (partition by customer_id order by order_date) as rn  
  from sales S  
  join menu M  
  on S.product_id=M.product_id  
  join members mem  
  on s.customer_id=mem.customer_id  
  where order_date>join_date  
)  
select customer_id,product_name  
from orders  
where rn=1;
```

	customer_id	product_name
▶	A	ramen
	B	sushi

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

```
with items as(
  select S.customer_id,M.product_name,S.order_date,Mem.join_date,
  row_number() over(partition by customer_id order by order_date) as rn
  from sales S
  join menu M
  on S.product_id=M.product_id
  join members Mem
  on S.Customer_id=Mem.customer_id
  where order_date < join_date
)
select customer_id,product_name
from items
where rn=1;
```

	customer_id	product_name
▶	A	sushi
	B	curry

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

```
• Select S.customer_id,count(S.product_id ) as quantity ,Sum(M.price) as total_sales
  From Sales S
  Join Menu M
  ON m.product_id = s.product_id
 JOIN Members Mem
  ON Mem.Customer_id = S.customer_id
 Where S.order_date < Mem.join_date
 Group by S.customer_id;
```

	customer_id	quantity	total_sales
▶	B	3	40
	A	2	25

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 S.customer_id,M.price,  
    case  
        when M.product_id= 1 then M.price*10*2  
        else M.price*10  
    end as points  
    from menu M  
    join sales S  
    on M.product_id=S.product_id  
)  
select customer_id,sum(points)  
from CTE  
group by customer id;
```

	customer_id	sum(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 customer A and B have at the end of January?

```
SELECT
  s.customer_id,
  SUM(
    CASE
      WHEN (DATEDIFF(s.order_date, me.join_date) BETWEEN 0 AND 7) OR (m.product_ID = 1)
      THEN m.price * 20
      ELSE m.price * 10
    END
  ) AS Points
FROM
  members AS me
JOIN sales AS s ON s.customer_id = me.customer_id
JOIN menu AS m ON m.product_id = s.product_id
WHERE s.order_date < '2021-01-31'
GROUP BY s.customer_id
LIMIT 50000;
```

	customer_id	Points
▶	B	940
	A	1370

Bonus Questions

Join All The Things

```
• select S.customer_id,S.order_date,M.product_name,M.price,  
  case when S.order_date >= mem.join_date then 'Y'  
  ELSE 'N'  
  end as member_status  
from menu M  
LEFT join sales S  
on M.product_id=S.product_id  
LEFT join members mem  
on S.customer_id=mem.customer_id  
order by S.customer_id,S.order_date;
```

	customer_id	order_date	product_name	price	member_status
▶	A	2021-01-01	sushi	10	N
	A	2021-01-01	curry	15	N
	A	2021-01-07	curry	15	Y
	A	2021-01-10	ramen	12	Y
	A	2021-01-11	ramen	12	Y
	A	2021-01-11	ramen	12	Y
	B	2021-01-01	curry	15	N
	B	2021-01-02	curry	15	N
	B	2021-01-04	sushi	10	N
	B	2021-01-11	sushi	10	Y
	B	2021-01-16	ramen	12	Y
	B	2021-02-01	ramen	12	Y
	C	2021-01-01	ramen	12	N
	C	2021-01-01	ramen	12	N
	C	2021-01-07	ramen	12	N

Rank All The Things

```
with cte as
(
  SELECT S.customer_id,S.order_date,M.product_name,M.price,
  case when S.order_date>=mem.join_date then 'Y'
  else 'n'
  end as member_status

  from menu M
  left join Sales S
  on M.product_id=S.product_id
  left join members mem
  on S.customer_id=mem.customer_id
)
select *,CASE
  WHEN member_status = 'N' then NULL
  ELSE RANK () OVER(
    PARTITION BY customer_id, member_status
    ORDER BY order_date) END AS ranking
FROM cte;
```

	customer_id	order_date	product_name	price	member_status	ranking
▶	A	2021-01-01	sushi	10	n	NULL
	A	2021-01-01	curry	15	n	NULL
	A	2021-01-07	curry	15	Y	1
	A	2021-01-10	ramen	12	Y	2
	A	2021-01-11	ramen	12	Y	3
	A	2021-01-11	ramen	12	Y	3
	B	2021-01-01	curry	15	n	NULL
	B	2021-01-02	curry	15	n	NULL
	B	2021-01-04	sushi	10	n	NULL
	B	2021-01-11	sushi	10	Y	1
	B	2021-01-16	ramen	12	Y	2
	B	2021-02-01	ramen	12	Y	3
	C	2021-01-01	ramen	12	n	NULL
	C	2021-01-01	ramen	12	n	NULL
	C	2021-01-07	ramen	12	n	NULL