

8WEEKSQLCHALLENGE.COM
CASE STUDY #1



THE TASTE OF SUCCESS



Anurag Dubey

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.



01

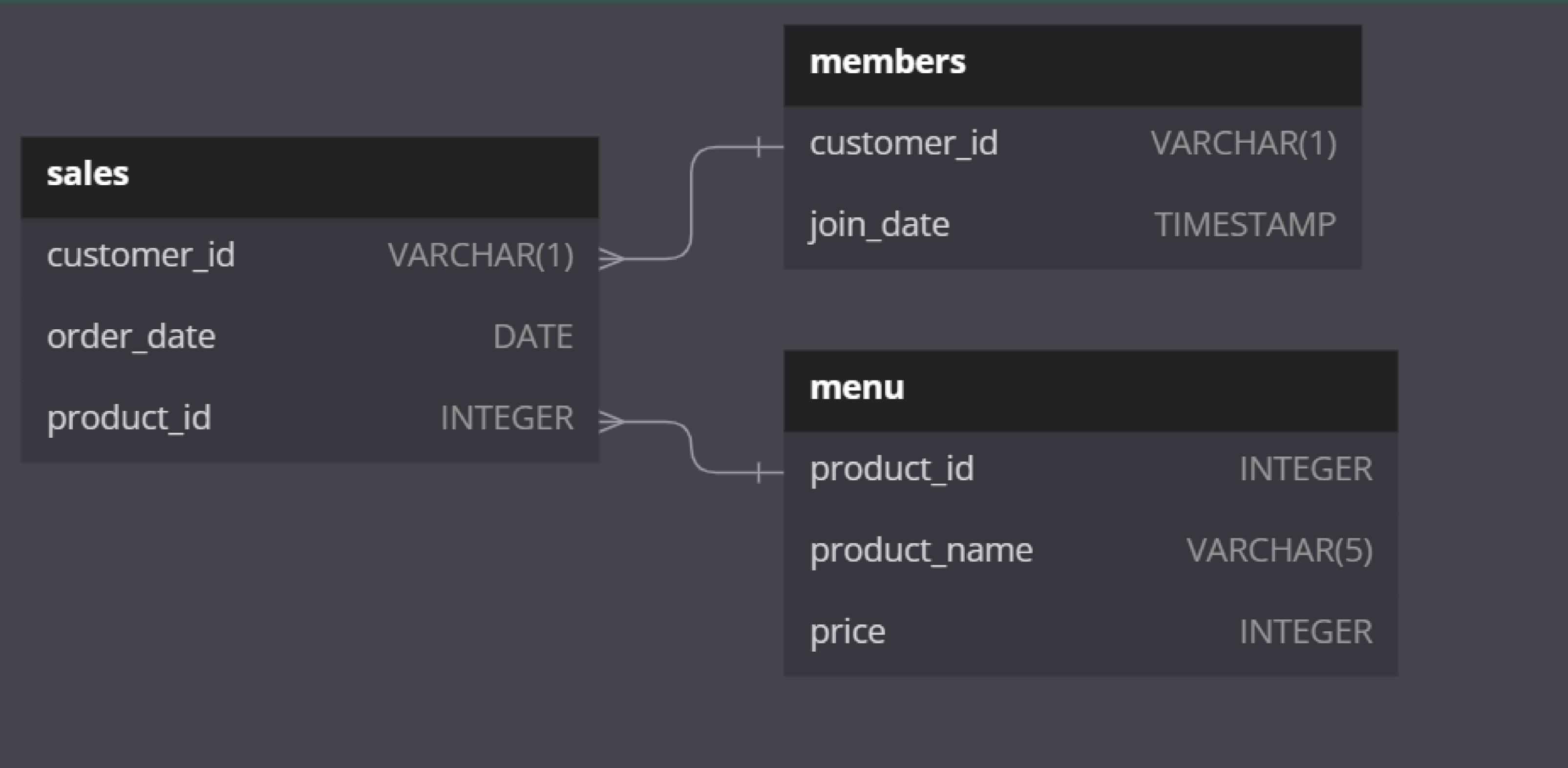
02

03

Tables – Member Sales Menu



Entity Relationship Diagram



:

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

```
select m.customer_id,sum(price) as 'Total Spent'  
from members m inner join sales s on m.customer_id=s.customer_id  
inner join menu on s.product_id=menu.product_id  
group by m.customer_id  
order by 2 desc;
```

| customer_id | Total Spent |
|-------------|----------------|
| A | 76 |
| B | 74 |

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



```
select s.customer_id, count(distinct(order_date)) as 'Days Visited'  
from members m inner join sales s on m.customer_id=s.customer_id  
group by s.customer_id;
```

| customer_id | Days Visited |
|-------------|--------------|
| A | 4 |
| B | 6 |



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



```
WITH RankedSales AS (
    SELECT
        s.customer_id,
        m.product_name,
        ROW_NUMBER() OVER (PARTITION BY s.customer_id ORDER BY s.order_date) AS purchase_rank
    FROM
        sales s
    INNER JOIN menu m ON s.product_id = m.product_id
)
SELECT customer_id, product_name
FROM RankedSales
WHERE purchase_rank = 1;
```



| customer_id | product_name |
|-------------|--------------|
| 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?



```
with most_purchase_item as
(select product_name,count(*) Number_of_purchase,
dense_rank() over(order by count(*) desc) as rnk
from sales s inner join menu m on s.product_id=m.product_id
group by product_name)

select product_name,Number_of_purchase from most_purchase_item
where rnk=1;
```



| product_name | Number_of_purchase |
|--------------|--------------------|
| ramen | 8 |

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



```
SELECT customer_id, product_name, number_of_time_ordered
FROM (
    SELECT
        m.customer_id, product_name,
        COUNT(*) AS number_of_time_ordered,
        DENSE_RANK() OVER (PARTITION BY m.customer_id ORDER BY COUNT(*) DESC) AS rnk
    FROM members m
        INNER JOIN sales s ON m.customer_id = s.customer_id
        INNER JOIN menu ON s.product_id = menu.product_id
    GROUP BY m.customer_id, product_name
) ranked_items
WHERE rnk = 1;
```

| customer_id | product_name | number_of_time_ordered |
|-------------|--------------|------------------------|
| A | ramen | 3 |
| B | sushi | 2 |
| B | curry | 2 |
| B | ramen | 2 |



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



```
with item_first_purchase as (
    select m.customer_id, product_name,
    row_number() over(partition by m.customer_id order by order_date asc) as rn
    from members m inner join sales s on m.customer_id=s.customer_id
    inner join menu on menu.product_id=s.product_id
    where m.join_date < s.order_date)

select customer_id, product_name
from item_first_purchase
where rn=1;
```



| customer_id | product_name |
|-------------|--------------|
| A | ramen |
| B | sushi |

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

```
with item_first_purchase as (
    select m.customer_id,product_name,
    row_number() over(partition by m.customer_id order by order_date desc) as rn
    from members m inner join sales s on m.customer_id=s.customer_id
    inner join menu on menu.product_id=s.product_id
    where m.join_date>s.order_date)

select customer_id,product_name
from item_first_purchase
where rn=1;
```

| customer_id | product_name |
|-------------|--------------|
| A | sushi |
| B | sushi |

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



```
select m.customer_id,  
       count(product_name) as 'total items',  
       sum(price) as 'total amount spent'  
  from members m inner join sales s on m.customer_id=s.customer_id  
inner join menu on menu.product_id=s.product_id  
 where m.join_date>s.order_date  
 group by m.customer_id  
 order by 1;
```

| customer_id | total items | total amount spent |
|-------------|-------------|--------------------|
| A | 2 | 25 |
| B | 3 | 40 |

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



```
WITH PointsPerPurchase AS (
    SELECT
        s.customer_id, m.product_name, m.price,
        CASE WHEN m.product_name = 'sushi' THEN m.price * 20 ELSE m.price * 10 END AS points
    FROM sales s
    INNER JOIN menu m ON s.product_id = m.product_id
)
SELECT
    customer_id, SUM(points) AS total_points
FROM PointsPerPurchase
GROUP BY customer_id
ORDER BY customer_id;
```



| customer_id | 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?

```
WITH PointsPerPurchase AS (
    SELECT
        s.customer_id, product_name, price,
        CASE
            WHEN s.order_date <= m.join_date + INTERVAL 7 DAY THEN price * 20 ELSE price * 10 END AS points
        FROM sales s INNER JOIN menu ON s.product_id = menu.product_id
        INNER JOIN members m ON m.customer_id=s.customer_id
        WHERE month(s.order_date)=1 AND s.customer_id IN ('A','B')
)
SELECT
    customer_id, SUM(points) AS total_points
FROM PointsPerPurchase
GROUP BY customer_id
```

| customer_id | total_points |
|-------------|--------------|
| A | 1520 |
| B | 1240 |

THANK YOU!

