# CASE STUDY # 1


# 8 WEEKS SQL CHALLANGE

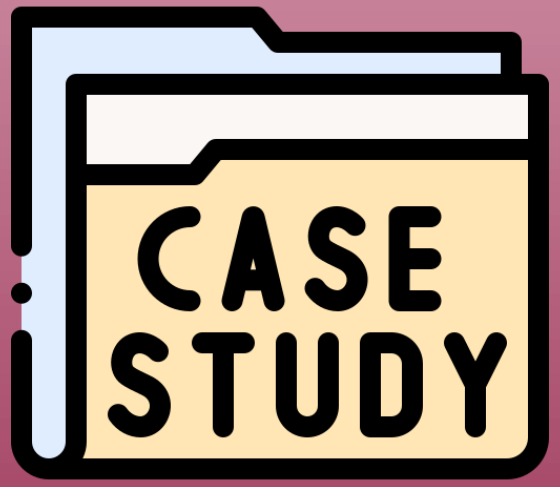
# Contents

- Who are the Danny’s Diner ? About the Case Study
- Objective of the Analysis
- Knowing the Dataset (ER)
- Data Analysis (Q & A )

# Who are the Danny’s Diner ? About the Case Study

- Danny a passionate lover of 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.

- Now, Danny's Diner seeks your aid to navigate their operational data effectively and ensure the restaurant's success



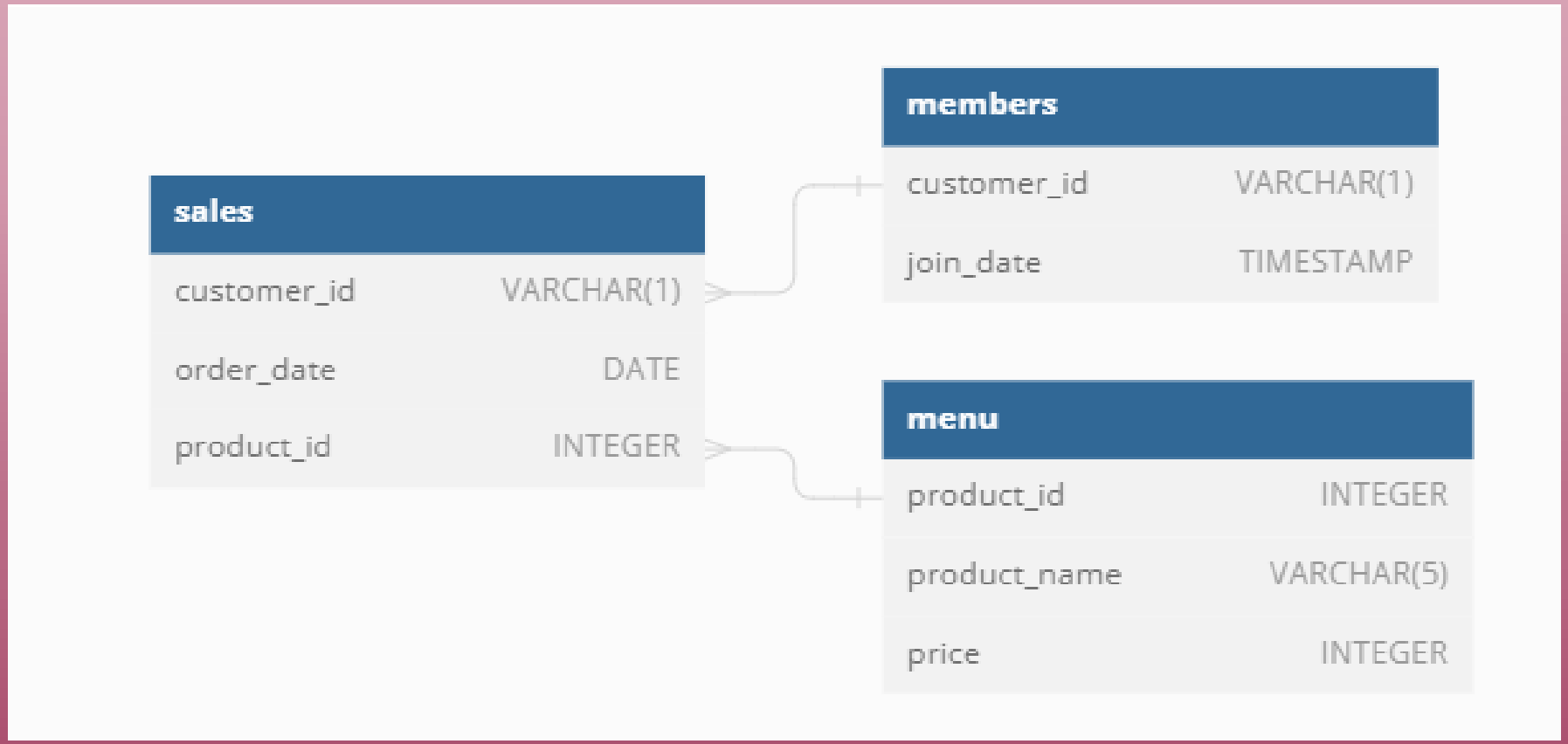
# Objective of the Analysis

- Danny aims to understand customer behavior,focusing on visiting patterns, expenditure, and favourite menu items to enhance personalized experiences. 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.

# Knowing the Dataset (ER)

- The dataset comprises of three tables namely : Sales , Members , Menu .
- The data specification and ER  are below :



# Data Analysis (Q & A )

## Q1-What is the total amount each customer spent at the restaurant?

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

| Customer_id | Total_amount |
|---|---|
| A | 76 |
| B | 74 |
| C | 36 |

# Data Analysis (Q & A )

## Q-2 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
order by customer_id ;
```

| Customer_id | No_of_Days |
|---|---|
| A | 4 |
| B | 6 |
| C | 2 |

# Data Analysis (Q & A )

## Q-3 What was the first item from the menu purchased by each customer?

```
select customer_id , product_name from
(
select s.customer_id , m.product_name ,
row_number() over(partition by customer_id  order by
order_date) as rnk
from sales as s join menu as m on s.product_id = m.product_id
)
where rnk =1;
```

| Customer_id | Product_Name |
|---|---|
| A | sushi |
| B | curry |
| C | ramen |

# Data Analysis (Q & A )

## Q-4 What is the most purchased item on the menu and how many times was it purchased by all customers?

```
select m.product_name , count(order_date) as total
from sales as s join menu as m on s.product_id = m.product_id
group by m.product_name
order by total desc
```

| Product_name | No_of_times |
|---|---|
| sushi | 8 |

# Data Analysis (Q & A )

## Q-5 Which item was the most popular for each customer?

```
select customer_id , product_name, no_times from
(
select s.customer_id , m.product_name,count(order_date) as
no_times ,
row_number() over(partition by customer_id  order by
count(order_date)desc) as rnk
from sales as s join menu as m on s.product_id = m.product_id
group by s.customer_id ,m.product_name
)
where rnk =1;
```

| Customer_id | Product_name | No_of_times |
|---|---|---|
| A | ramen | 3 |
| B | curry | 2 |
| C | ramen | 3 |

# Data Analysis (Q & A )

## Q-6 Which item was purchased first by the customer after they became a member?

```
with CTE as
(
        select s.customer_id,s.order_date,m.product_name,mb.join_date,
        rank() over(partition by s.customer_id order by order_date) as rnk
        from sales as s join menu as m on s.product_id =m.product_id
        join members as mb on s.customer_id = mb.customer_id
        where order_date>=join_date
)
select customer_id,product_name
from cte
where rnk=1
```

| Customer_id | Product_name |
|---|---|
| A | curry |
| B | sushi |

# Data Analysis (Q & A )

## Q-7 Which item was purchased just before the customer became a member?

```
with CTE as
(
        select s.customer_id,s.order_date,m.product_name,mb.join_date,
        rank() over(partition by s.customer_id order by order_date desc ) as rnk
        from sales as s join menu as m on s.product_id =m.product_id
        join members as mb on s.customer_id = mb.customer_id
        where order_date < join_date

)
select customer_id,product_name
from cte
where rnk=1;
```

| Customer_id | Product_name |
|---|---|
| A | sushi |
| A | curry |
| B | sushi |

# Data Analysis (Q & A )

## Q-8 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
product,sum(m.price)
from sales as s join menu as m on s.product_id =m.product_id
join members as mb on s.customer_id = mb.customer_id
where order_date < join_date
group by s.customer_id
order by s.customer_id
```

| Customer_id | Product_count | Price |
|---|---|---|
| A | 2 | 25 |
| B | 3 | 40 |

# Data Analysis (Q & A )

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

```
select s.customer_id,
sum(case
        when m.product_id='1' then m.price*2*10
        else m.price*10
        end) as points
from sales as s join menu as m on s.product_id =m.product_id
group by s.customer_id
Order by  s.customer_id ;
```

| Customer_id | Points |
|---|---|
| A | 860 |
| B | 940 |
| C | 360 |

# Data Analysis (Q & A )

**Q-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 CTE1 AS (
   SELECT
      S.customer_id, S.order_date, M.product_name, M.price,
      CASE
         WHEN product_name = 'sushi' THEN 2 * M.price
         WHEN order_date BETWEEN P.join_date AND
DATE_ADD(P.join_date, INTERVAL 6 DAY) THEN 2 * M.price
         ELSE M.price
      END AS points
   FROM Sales S JOIN Menu M ON S.product_id = M.product_id
JOIN Members P ON S.customer_id = P.customer_id
 WHERE DATE_FORMAT (order_date, '%Y-%m-01') = '2021-01-01'
)
```

```
SELECT
   customer_id, SUM(points) * 10 AS total_points
FROM CTE1
GROUP BY customer_id
order by customer_id;
where s.order_date >= me.join_date and s.order_date <=
CAST('2021-01-31' AS DATE)
Group by s.customer_id
```

| Customer_id | Total_points |
|---|---|
| A | 1370 |
| B | 820 |

# Bonus Questions-1

## Join All The Things

**Solution :**

```
SELECT
    S.customer_id,
    S.order_date,
    M.product_name,
    M.price,
    CASE
        WHEN order_date >= join_date THEN 'Y'
        ELSE 'N'
    END AS Member
FROM
    Sales S
        JOIN
    Menu M ON S.product_id = M.product_id
        LEFT JOIN
    Members P ON S.customer_id = P.customer_id
ORDER BY S.customer_id , S.order_date , M.product_name;
```

# Output :

| Customer_id | Order_date | Product_name | Price | Member |
|---|---|---|---|---|
| A | 2021-01-01 | curry | 15 | N |
| A | 2021-01-01 | sushi | 10 | N |
| A | 2021-01-07 | curry | 15 | Y |
| A | 2021-01-10 | 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 |

# Bonus Questions-2

## Rank All The Things

**Solution :**

```
with rank1 as
(
select s.customer_id , s.order_date,m.product_name,m.price,
case
        when s.order_date>=mb.join_date then 'Y'
        else 'N'
end as members
from sales as s join menu as m on s.product_id =m.product_id
left join members as mb on s.customer_id = mb.customer_id
Order by s.customer_id,s.order_date
)
select * ,
(
case
when members like'N' then Null
else
        rank() over(partition by customer_id ,members order by order_date)
end ) as ranking
from rank1;
```

# Output :

| Customer_id | Order_date | Product_name | Price | Member | Rank |
|---|---|---|---|---|---|
| A | 2021-01-01 | curry | 15 | N | null |
| A | 2021-01-01 | sushi | 10 | 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 |


THANK
YOU