8WEEKSQLCHALLENGE.COM

CASE STUDY #1

THE TASTE OF SUCCESS

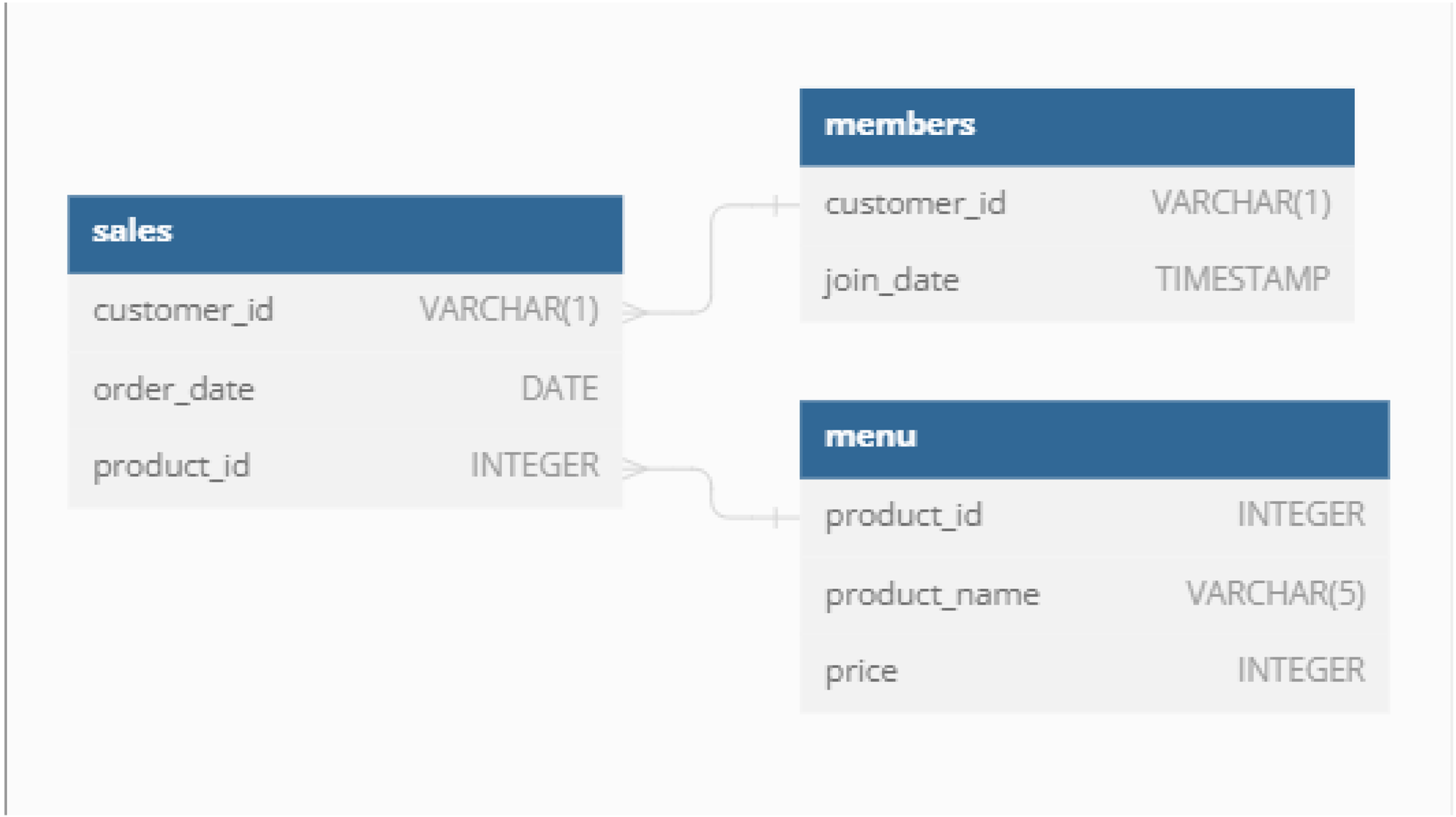
DATAWITHDANNY.COM

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.

ER DIAGRAM


sales
customer_id VARCHAR(1)
order_date DATE
product_id INTEGER
members
customer_id VARCHAR(1)
join_date TIMESTAMP
menu
product_id INTEGER
product_name VARCHAR(5)
price INTEGER

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

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

customer_id	amount_spent
A	76
B	74
C	36

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

```sql
select customer_id,
count( distinct order_date) as no_of_days
from dannys_diner.sales
group by customer_id
order by customer_id;
```

customer_id	no_of_days
A	4
B	6
C	2

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

```
2  select distinct customer_id,product_name,order_date from
3    (select customer_id,
4    m.product_name as product_name,
5    s.order_date as order_date,
6    dense_rank() over (partition by customer_id order by order_da
7    from dannys_diner.sales s
8    join dannys_diner.menu m
9    on s.product_id = m.product_id) as sub
10 where rnk = 1;
```

customer_id	product_name	order_date
A	curry	2021-01-01
A	sushi	2021-01-01
B	curry	2021-01-01
C	ramen	2021-01-01

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

```sql
with highest_selling_item as (
    select product_id,
      count(product_id) as cnt
      from dannys_diner.sales
      group by product_id
      order by cnt desc
      limit 1)
select s.customer_id,count(m.product_id) as no_of_times_purchased
from dannys_diner.sales s
join dannys_diner.menu m
on s.product_id = m.product_id
where m.product_id = (select product_id from highest_selling_item)
group by s.customer_id
order by s.customer_id;
```

customer_id	no_of_times_purchased
A	3
B	2
C	3

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

```sql
with get_highest_ordered_item as(
  select s.customer_id,
         m.product_name,
         count(s.*) as no_of_times_ordered,
         rank() over (partition by s.customer_id order by count(s.*) desc ) as rnk
         from dannys_diner.sales s
         join dannys_diner.menu m
         on s.product_id = m.product_id
         group by s.customer_id,m.product_name
)

select customer_id,product_name,no_of_times_ordered

from get_highest_ordered_item
where rnk = 1;
```

customer_id	product_name	no_of_times_ordered
A	ramen	3
B	sushi	2
B	curry	2
B	ramen	2
C	ramen	3

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

```sql
with get_first_order as
  (select s.customer_id as customer_id,
  mm.product_name as product_name,
  row_number() over (partition by s.customer_id order by s.order_date) as rnum
  from dannys_diner.members m
  join dannys_diner.sales s
  on s.customer_id = m.customer_id
  join dannys_diner.menu mm
  on s.product_id = mm.product_id
  where order_date > join_date)

select customer_id,product_name from get_first_order where rnum =1;
```

customer_id	product_name
A	ramen
B	sushi

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

```sql
with get_first_order as
  (select s.customer_id as customer_id,
  mm.product_name as product_name,
  s.order_date as order_date,m.join_date as join_date,
  rank() over (partition by s.customer_id order by s.order_date) as rnum
  from dannys_diner.members m
  join dannys_diner.sales s
  on s.customer_id = m.customer_id
  join dannys_diner.menu mm
  on s.product_id = mm.product_id
  where order_date <  join_date)

select customer_id,product_name,order_date,join_date
from get_first_order where rnum =1;
```

customer_id	product_name	order_date	join_date
A	sushi	2021-01-01	2021-01-07
A	curry	2021-01-01	2021-01-07
B	curry	2021-01-01	2021-01-09

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

New unsaved query*

```
1  select  s.customer_id,count(distinct m.product_name),sum(m.price) as amount_spent
2  from dannys_diner.menu m
3  join dannys_diner.sales s
4  on m.product_id = s.product_id
5  join dannys_diner.members mm
6  on s.customer_id = mm.customer_id
7  where s.order_date < mm.join_date
8  group by s.customer_id;
9
```

customer_id	count	amount_spent
A	2	25
B	2	40

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

```sql
select customer_id,sum(points) from
(
 select  s.customer_id,m.product_name,sum(m.price),
  case when product_name = 'sushi' then sum(m.price) * 20
       else sum(m.price) * 10 end as points
  from dannys_diner.sales s
  join dannys_diner.menu m
  on m.product_id = s.product_id
  group by s.customer_id,m.product_name
) get_points
group by customer_id
order by customer_id;
```

customer_id	sum
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?

```sql
WITH get_points
     AS (SELECT mm.customer_id AS customer_id,
                mm.join_date   AS join_date,
                s.order_date   AS order_date,
                m.price,
                m.product_name,
                CASE
                  WHEN s.order_date BETWEEN mm.join_date AND mm.join_date + 6
                THEN
                  m.price * 20
                  WHEN m.product_name = 'sushi' THEN m.price * 20
                  ELSE m.price * 10
                END            AS points
         FROM   dannys_diner.members mm
                JOIN dannys_diner.sales s
                  ON mm.customer_id = s.customer_id
                JOIN dannys_diner.menu m
                  ON m.product_id = s.product_id)
SELECT customer_id,
       Sum(points)
FROM   get_points
GROUP  BY customer_id;
```

customer_id	sum
A	1370
B	940