8WEEKSQLCHALLENGE.COM

CASE STUDY #1

THE TASTE OF SUCCESS

DATAWITHDANNY.COM

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

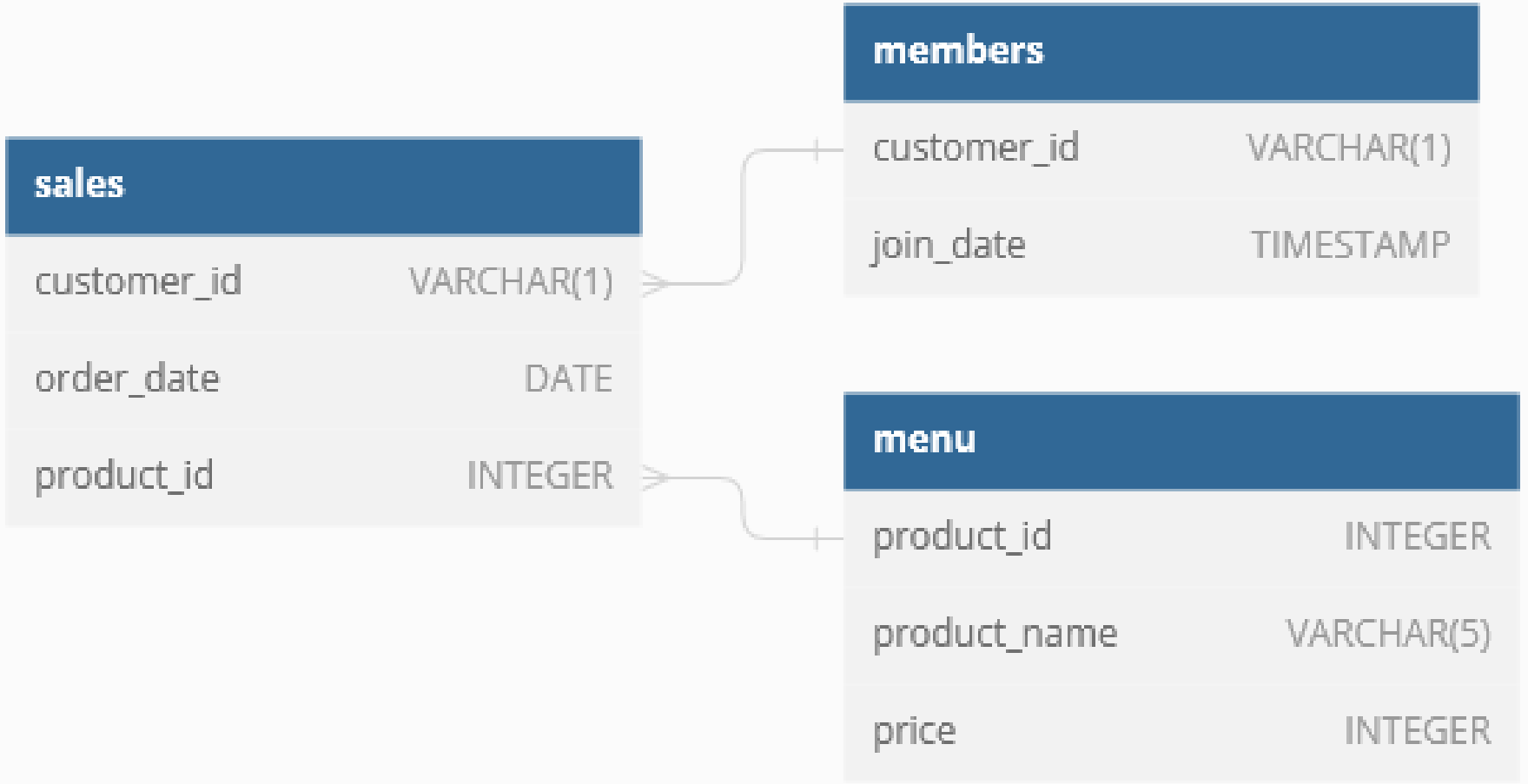
Danny wants to use the data to answer a few simple questions about his customers, especially about their visiting patterns, how much money they've spent and also which menu items are their favourite. Having this deeper connection with his customers will help him deliver a better and more personalised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

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


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

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?
2. How many days has each customer visited the restaurant?
3. What was the first item from the menu purchased by each customer?
4. What is the most purchased item on the menu and how many times was it purchased by all customers?
5. Which item was the most popular for each customer?
6. Which item was purchased first by the customer after they became a member?
7. Which item was purchased just before the customer became a member?
8. What is the total items and amount spent for each member before they became a member?
9. If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?
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?

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

```sql
SELECT
    customer_id, SUM(price)
FROM
    sales a
        INNER JOIN
    menu b ON a.product_id = b.product_id
GROUP BY customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	SUM(price)
A	76
B	74
C	36

Q2. How many days has each customer visited the restaurant?

```sql
SELECT
    customer_id, COUNT(DISTINCT (order_date)) no_of_days
FROM
    sales
GROUP BY customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	no_of_days
A	4
B	6
C	2

Q3. What was the first item from the menu purchased by each customer?

```sql
select * from (
select a.customer_id,a.order_date,b.product_name,
rank() over(partition by customer_id order by order_date) first_order
from sales a inner join menu b on a.product_id=b.product_id ) z
where first_order = 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	order_date	product_name	first_order
A	2021-01-01	sushi	1
A	2021-01-01	curry	1
B	2021-01-01	curry	1
C	2021-01-01	ramen	1
C	2021-01-01	ramen	1

Q4. What is the most purchased item on the menu and how many times was it purchased by all customers?

```sql
SELECT
    product_name, COUNT(product_name) max_orders
FROM
    sales a
        INNER JOIN
    menu b ON a.product_id = b.product_id
GROUP BY product_name
ORDER BY max_orders DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

product_name	max_orders
ramen	8

Q5. Which item was the most popular for each customer?

```sql
select customer_id,product_name from (
select * from (
select * ,
rank() over(partition by customer_id order by popular_item desc) rnk
from(
select customer_id,product_name,count(product_name) popular_item from (
select a.customer_id,b.product_name from sales a inner join menu b on a.product_id=b.product_id) z
group by customer_id,product_name )x ) c
where rnk =1 ) k ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

Q6. Which item was purchased first by the customer after they became a member?

```sql
select * from (
select *,
rank() over (partition by customer_id order by order_date) rnk
 from(
select * from (
select b.customer_id,a.join_date,b.order_date,b.product_id,c.product_name from members a left join sales b on a.customer_id=b.customer_id
 inner join menu c on b.product_id = c.product_id) z
where join_date <=order_date ) x) c
where rnk =1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	join_date	order_date	product_id	product_name	rnk
A	2021-01-07	2021-01-07	2	curry	1
B	2021-01-09	2021-01-11	1	sushi	1

Q7. Which item was purchased just before the customer became a member?

```sql
select customer_id,product_name from (
select *,
rank() over(partition by customer_id order by order_date desc) first_order_before_joining
 from (
select * from (
select a.customer_id,a.join_date,b.order_date,b.product_id,c.product_name from members a left join sales b on a.customer_id=b.customer_id
 inner join menu c on b.product_id=c.product_id ) z
 where order_date < join_date ) x) c
 where first_order_before_joining =1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	product_name
A	sushi
A	curry
B	sushi

Q8. What is the total items and amount spent for each member before they became a member?

```sql
SELECT
    a.customer_id, COUNT(product_name) total_items, SUM(price) amount_spent
FROM
    members a
        LEFT JOIN
    sales b ON a.customer_id = b.customer_id
        INNER JOIN
    menu c ON b.product_id = c.product_id
WHERE
    order_date < join_date
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	total_items	amount_spent
B	3	40
A	2	25

Q9. 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) points
FROM
    (SELECT
        a.customer_id,
            b.product_id,
            b.product_name,
            b.price,
            CASE
                WHEN product_name = 'sushi' THEN price * 20
                ELSE price * 10
            END AS points
    FROM
        sales a
    INNER JOIN menu b ON a.product_id = b.product_id) z
GROUP BY customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	points
A	860
B	940
C	360

Q10. 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
SELECT
    customer_id, SUM(new_price) * 10 as points
FROM
    (SELECT
        a.customer_id, order_date,product_name,price,
            CASE
                WHEN product_name = 'sushi' THEN 2 * price
                WHEN order_date BETWEEN join_date AND (join_date + INTERVAL 6 DAY) THEN 2 * price
                ELSE price
            END AS new_price
    FROM
        sales a
    JOIN menu b ON a.product_id = b.product_id
    JOIN members c ON a.customer_id = c.customer_id
    WHERE
        order_date <= '2021-01-31') z
GROUP BY customer_id;
```

THANK YOU