8WEEKSQLCHALLENGE.COM

CASE STUDY #1

THE TASTE OF SUCCESS

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`

Entity Relationship Diagram

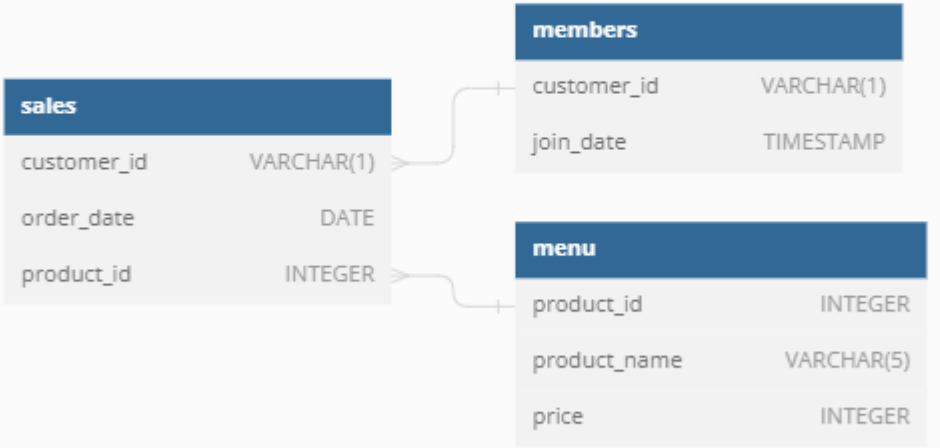

Table 1: sales

The `sales` table captures all `customer_id` level purchases with an corresponding `order_date` and `product_id` information for when and what menu items were ordered.

customer_id	order_date	product_id
A	2021-01-01	1
A	2021-01-01	2
A	2021-01-07	2
A	2021-01-10	3
A	2021-01-11	3
A	2021-01-11	3
B	2021-01-01	2
B	2021-01-02	2
B	2021-01-04	1
B	2021-01-11	1
B	2021-01-16	3
B	2021-02-01	3
C	2021-01-01	3
C	2021-01-01	3
C	2021-01-07	3

Table 2: menu

The `menu` table maps the `product_id` to the actual `product_name` and `price` of each menu item.

product_id	product_name	price
1	sushi	10
2	curry	15
3	ramen	12

Table 3: members

The final `members` table captures the `join_date` when a `customer_id` joined the beta version of the Danny's Diner loyalty program.

customer_id	join_date
A	2021-01-07
B	2021-01-09

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. 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 ?

```sql
/*What is the total amount each customer spent at the restaurant?*/
select customer_id,sum(price) from sales
Inner join menu on sales.product_id=menu.product_id
group by customer_id
```

Data Output | Messages | Notifications

	customer_id character varying (1)	sum bigint
1	B	74
2	C	36
3	A	76

```
/*How many days has each customer visited the restaurant?*/
select customer_id,count(distinct(order_date)) as total_days from sales
group by customer_id
```

Data Output | Messages | Notifications

	customer_id character varying (1)	total_days bigint
1	A	4
2	B	6
3	C	2

```sql
/*What was the first item from the menu purchased by each customer?*/
WITH CTE AS (
SELECT customer_id,product_name,order_date,
ROW_NUMBER() OVER(PARTITION BY customer_id ORDER BY order_date)as rn
from sales
Inner join menu on sales.product_id=menu.product_id
    )
SELECT customer_id,product_name from CTE
WHERE rn='1'
```

Data Output | Messages | Notifications

	customer_id character varying (1)	product_name character varying (5)
1	A	curry
2	B	curry
3	C	ramen

```sql
/*What is the most purchased item on the menu and how many times
 was it purchased by all customers?*/
select count(sales.product_id) as orders,product_name from sales
Inner join menu on sales.product_id=menu.product_id
group by product_name
limit 1
```

Data Output | Messages | Notifications

	orders bigint	product_name character varying (5)
1	8	ramen

```sql
/*Which item was the most popular for each customer?*/
WITH CTE AS (
select customer_id,count(sales.product_id) as orders,product_name,
RANK() OVER(PARTITION BY customer_id ORDER BY count(sales.product_id)DESC)as rnk
from sales
Inner join menu on sales.product_id=menu.product_id
group by customer_id,product_name
    )
SELECT customer_id,product_name from CTE
where rnk='1'
```

Data Output | Messages | Notifications

	customer_id character varying (1)	product_name character varying (5)
1	A	ramen
2	B	sushi
3	B	curry
4	B	ramen
5	C	ramen

```sql
/*Which item was purchased first by the customer after they became a member?*/
WITH CTE AS (
select sales.customer_id,order_date,join_date,product_name,
RANK() OVER(PARTITION BY sales.customer_id ORDER BY order_date)as rnk
from sales
Inner join members on sales.customer_id=members.customer_id
Inner join menu on sales.product_id=menu.product_id
where order_date>=join_date
    )
SELECT customer_id,product_name from CTE
WHERE rnk='1'
```

Data Output | Messages | Notifications

	customer_id character varying (1)	product_name character varying (5)
1	A	curry
2	B	sushi

```sql
/*Which item was purchased just before the customer became a member?*/
WITH CTE AS(
select sales.customer_id,product_name,order_date,join_date,
RANK() OVER(PARTITION BY sales.customer_id ORDER BY order_date DESC)as rnk
from sales
Inner join members on sales.customer_id=members.customer_id
Inner join menu on sales.product_id=menu.product_id
WHERE order_date<join_date
    )
SELECT customer_id,product_name from CTE
WHERE rnk='1'
```

Data Output | Messages | Notifications

	customer_id character varying (1)	product_name character varying (5)
1	A	sushi
2	A	curry
3	B	sushi

```sql
/*What is the total items and amount spent for each member before
  they became a member?*/
select sales.customer_id,count(product_name)as total_items,
sum(price)as total_amount_spent from sales
Inner join members on sales.customer_id=members.customer_id
Inner join menu on sales.product_id=menu.product_id
where order_date<join_date
group by sales.customer_id
order by customer_id
```

Data Output | Messages | Notifications

	customer_id character varying (1)	total_items bigint	total_amount_spent bigint
1	A	2	25
2	B	3	40

```sql
/*If each $1 spent equates to 10 points and sushi has a 2x points multiplier -
  how many points would each customer have?*/
Select customer_id, SUM(CASE WHEN product_name='sushi' THEN price*10*2
                        ELSE price*10 END) AS points
from menu
Inner join sales on sales.product_id=menu.product_id
GROUP BY customer_id
ORDER BY customer_id
```

Data Output | Messages | Notifications

	customer_id character varying (1)	points bigint
1	A	860
2	B	940
3	C	360

```sql
/*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?*/
select sales.customer_id,SUM(CASE
            WHEN product_name='sushi' THEN price*10*2
            WHEN product_name='curry' THEN price*10*2
            WHEN product_name='ramen' THEN price*10*2
            END) AS points
from sales
Inner join members on sales.customer_id=members.customer_id
Inner join menu on sales.product_id=menu.product_id
where order_date>=join_date
GROUP BY sales.customer_id
ORDER BY sales.customer_id
```

Data Output | Messages | Notifications

	customer_id character varying (1)	points bigint
1	A	1020
2	B	680