PostgreSQL

SQL CHALLENGE

By Kanishka Garg

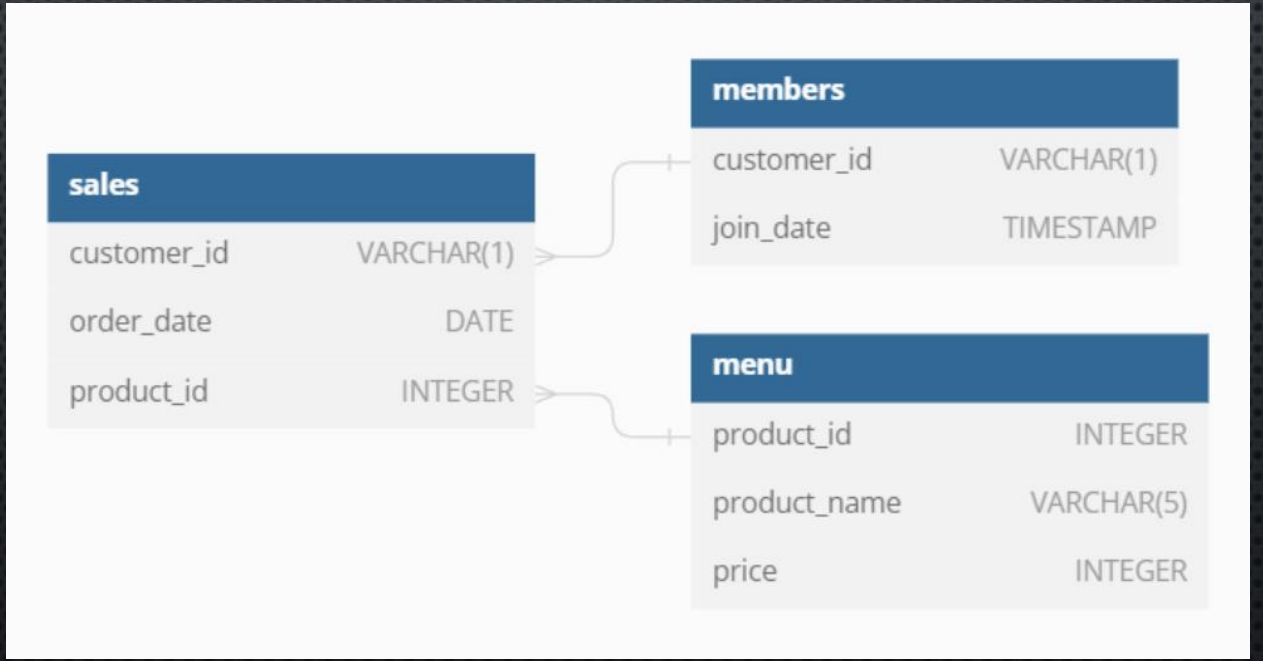
Introduction

- 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 favorite 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 favorite. Having this deeper connection with his customers will help him deliver a better and more personalized 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:
  1. sales
  2. Menu
  3. members

ENTITY RELATIONSHIP DIAGRAM



Q 1 : WHAT IS THE TOTAL AMOUNT EACH CUSTOMER SPENT AT THE RESTAURANT?

```sql
with sales_menu as (select * from sales s
join menu mn
on s.product_id = mn.product_id)

select customer_id, sum(price) as Amount_spent from sales_menu
group by customer_id
```

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

Q 2: HOW MANY DAYS HAS EACH CUSTOMER VISITED THE RESTAURANT?

```
select customer_id, count(distinct(order_date)) as no_of_days_visited from sales
group by customer_id;
```

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

Q 3: WHAT WAS THE FIRST ITEM FROM THE MENU PURCHASED BY EACH CUSTOMER?

```sql
with sales_menu as (select s.customer_id, s.order_date, mn.product_name,
dense_rank() over (partition by customer_id order by s.order_date) as rank from sales s
join menu mn
on s.product_id = mn.product_id
group by s.customer_id, s.order_date,mn.product_name
order by s.order_date asc)

select customer_id, order_date, product_name as first_purchased from sales_menu
where rank = 1;
```

	customer_id character varying (1)	order_date date	first_purchased character varying (5)
1	A	2021-01-01	curry
2	A	2021-01-01	sushi
3	B	2021-01-01	curry
4	C	2021-01-01	ramen

Q 4: WHAT IS THE MOST PURCHASED ITEM ON THE MENU AND HOW MANY TIMES WAS IT PURCHASED BY ALL CUSTOMERS?

```sql
with sales_menu as (select * from sales s
join menu mn
on s.product_id = mn.product_id)

select product_name, count(product_name) as no_of_times_purchased from sales_menu
group by product_name
order by count(product_name) desc
limit 1
```

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

Q 5: WHICH ITEM WAS THE MOST POPULAR FOR EACH CUSTOMER?

```sql
with sales_menu as (select s.customer_id, mn.product_name, count(mn.product_name) as count_of_product,
                    dense_rank() over (partition by customer_id order by count(product_name) desc ) as rank
                    from sales s
                    join menu mn
                    on s.product_id = mn.product_id
                    group by s.customer_id, mn.product_name
                   )
select customer_id, product_name, count_of_product from sales_menu
where rank = 1;
```

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

Q 6: WHICH ITEM WAS PURCHASED FIRST BY THE CUSTOMER AFTER THEY BECAME A MEMBER?

```
with combined_table as (select min(s.order_date) as after_joindate, s.customer_id, mn.product_name from sales s
                        natural join menu mn
                        natural join members mm
                       group by customer_id, order_date, join_date, product_name
                       having join_date < order_date)


select min(after_joindate), customer_id, product_name from combined_table
group by customer_id, product_name
limit 2
```

	min date	customer_id character varying (1)	product_name character varying (5)
1	2021-01-11	B	sushi
2	2021-01-10	A	ramen

Q 7: WHICH ITEM WAS PURCHASED JUST BEFORE THE CUSTOMER BECAME A MEMBER?

```
with combined_table as (select min(s.order_date) as after_joindate, s.customer_id, mn.product_name from sales s
                        natural join menu mn
                        natural join members mm
                       group by customer_id, order_date, join_date, product_name
                       having join_date > order_date)


select min(after_joindate), customer_id, product_name from combined_table
group by customer_id, product_name
limit 3
```

	min date	customer_id character varying (1)	product_name character varying (5)
1	2021-01-04	B	sushi
2	2021-01-01	A	curry
3	2021-01-01	A	sushi

Q 8: WHAT IS THE TOTAL ITEMS AND AMOUNT SPENT FOR EACH MEMBER BEFORE THEY BECAME A MEMBER?

```
with combined_table as (select s.order_date, s.customer_id, mn.product_name, mm.join_date, mn.price from sales s
                        natural join menu mn
                        natural join members mm
                        where mm.join_date > s.order_date)


select customer_id, sum(price) as amount_spent, count(product_name) as total_items from combined_table
group by customer_id
```

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

Q 9: IF EACH $1 SPENT EQUATES TO 10 POINTS AND SUSHI HAS A 2X POINTS MULTIPLIER - HOW MANY POINTS WOULD EACH CUSTOMER HAVE?

```
with sales_menu as (select *, CASE
    WHEN product_name = 'sushi' THEN 20
    ELSE 10
END as num
from sales s
join menu mn
on s.product_id = mn.product_id
)

select customer_id, sum(price*num) as total_points from sales_menu
group by customer_id
```

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

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?

```sql
select customer_id, sum(price*num) as total_points from sales_menu
group by customer_id;
-- select * from sales_menu

--10
select s.customer_id, sum(20*mn.price)
from sales s
natural join menu mn
natural join members mm
where mm.join_date <= s.order_date and order_date < '2021-01-31'
group by s.customer_id
```

	customer_id character varying (1)	sum bigint
1	B	440
2	A	1020