DANNY MA 8 WEEK SQL challenge

This is solved both in SQL and Pandas for improving skills.

SQL Schema :

**CREATE** **SCHEMA** dannys\_diner;

**SET** search\_path = dannys\_diner;

**CREATE** **TABLE** sales (

"customer\_id" VARCHAR(1),

"order\_date" DATE,

"product\_id" INTEGER

);

**INSERT** **INTO** sales

("customer\_id", "order\_date", "product\_id")

**VALUES**

('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');

**CREATE** **TABLE** menu (

"product\_id" INTEGER,

"product\_name" VARCHAR(5),

"price" INTEGER

);

**INSERT** **INTO** menu

("product\_id", "product\_name", "price")

**VALUES**

('1', 'sushi', '10'),

('2', 'curry', '15'),

('3', 'ramen', '12');

**CREATE** **TABLE** members (

"customer\_id" VARCHAR(1),

"join\_date" DATE

);

**INSERT** **INTO** members

("customer\_id", "join\_date")

**VALUES**

('A', '2021-01-07'),

Danny’s Diner :

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

select customer\_id,sum(price) FROM menu

INNER JOIN sales ON menu.product\_id=sales.product\_id

group by customer\_id;

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

select customer\_id, count(distinct(order\_date)) FROM menu

INNER JOIN sales ON menu.product\_id=sales.product\_id

group by customer\_id;

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

with cte1 as

(

select sales.customer\_id,sales.order\_date,sales.product\_id,menu.price,menu.product\_name, dense\_rank()over(partition by sales.customer\_id order by order\_date) as date\_rank

from sales

inner join menu on sales.product\_id=menu.product\_id

)

select customer\_id,order\_date,product\_name

from cte1

where date\_rank=1;

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

select count(product\_name) as cnt,product\_name,customer\_id from menu

inner join sales on menu.product\_id=sales.product\_id

group by product\_name

order by cnt desc

LIMIT 1;

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

with most\_popular\_cte as (

SELECT customer\_id,product\_name,count(product\_name) as order\_count,DENSE\_RANK() OVER(PARTITION BY customer\_id

order by count(product\_name)desc) as product\_rank

from sales

inner join menu on sales.product\_id=menu.product\_id

group by customer\_id,product\_name

)

select customer\_id,product\_name,order\_count from most\_popular\_cte

where product\_rank=1;

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

SELECT \*

FROM (

SELECT sales.customer\_id, menu.product\_id, menu.product\_name,

sales.order\_date, members.join\_date,

DENSE\_RANK() OVER (PARTITION BY sales.customer\_id ORDER BY DATEDIFF(sales.order\_date, members.join\_date)) AS date\_rank

FROM menu

INNER JOIN sales ON sales.product\_id = menu.product\_id

INNER JOIN members ON sales.customer\_id = members.customer\_id

WHERE sales.order\_date > members.join\_date

) AS ranked\_data

WHERE date\_rank = 1

ORDER BY customer\_id;

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

SELECT \*

FROM (

SELECT sales.customer\_id, menu.product\_id, menu.product\_name,

sales.order\_date, members.join\_date,

DENSE\_RANK() OVER (PARTITION BY sales.customer\_id ORDER BY DATEDIFF(sales.order\_date, members.join\_date) desc) AS date\_rank

FROM menu

INNER JOIN sales ON sales.product\_id = menu.product\_id

INNER JOIN members ON sales.customer\_id = members.customer\_id

WHERE sales.order\_date < members.join\_date

) AS ranked\_data

WHERE date\_rank = 1

ORDER BY customer\_id;

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

select sales.customer\_id,count(product\_name),sum(price) from sales

inner join menu on sales.product\_id=menu.product\_id

inner join members on sales.customer\_id=members.customer\_id

where order\_date<join\_date

group by sales.customer\_id

order by sales.customer\_id;

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

select \*, sum(

case

when menu.product\_name='sushi' then price\*20

else price\*10

end) as points

from members

right join sales on members.customer\_id=sales.customer\_id

inner join menu on menu.product\_id=sales.product\_id

group by sales.customer\_id

order by sales.customer\_id;

**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?**

***This can be improved but as a beginner understanding about CTE and syntax. I personally feel it is a good habit to break things down run each cte and see how things are looking.***

with cte1 as

(

select sales.customer\_id,sales.order\_date,members.join\_date,sales.product\_id,menu.price,menu.product\_name, datediff(order\_date,join\_date) as n\_days

, price\*10 as base\_points

from members

right join sales on sales.customer\_id=members.customer\_id

inner join menu on sales.product\_id=menu.product\_id

),

cte2 as

(

select \*,

case

when n\_days>=0 and n\_days<=6 then base\_points\*2

else base\_points

end as first\_week\_points, month(order\_date) as n\_months

from cte1),

cte3 as

( select \*,

case

when cte2.product\_name='sushi' then cte2.base\_points\*2

else first\_week\_points

end as total\_points

from cte2)

select cte3.customer\_id,sum(total\_points) from cte3

where cte3.n\_months=1

group by cte3.customer\_id

order by cte3.customer\_id

LIMIT 2;