\*\*Schema (PostgreSQL v13)\*\*

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'),

('B', '2021-01-09');

---

\*\*Query #1\*\*

SELECT

s.customer\_id,

sum(m.price) as Total

FROM sales s join menu m

on s.product\_id=m.product\_id

group by customer\_id

order by 2 desc;

| customer\_id | total |

| ----------- | ----- |

| A | 76 |

| B | 74 |

| C | 36 |

---

\*\*Query #2\*\*

select customer\_id,

count(distinct order\_date) as Visits

from sales

group by customer\_id

order by 2 desc;

| customer\_id | visits |

| ----------- | ------ |

| B | 6 |

| A | 4 |

| C | 2 |

---

\*\*Query #3\*\*

With q1 as(

select customer\_id, product\_name,

row\_number() over(partition by customer\_id order by order\_date, s.product\_id) as rk

from sales s join menu m

on s.product\_id=m.product\_id)

select customer\_id, product\_name from q1

where rk = 1;

| customer\_id | product\_name |

| ----------- | ------------ |

| A | sushi |

| B | curry |

| C | ramen |

---

\*\*Query #4\*\*

select product\_name,

count(s.product\_id) as ordered

from sales s join menu m

on s.product\_id=m.product\_id

group by product\_name

order by ordered desc

limit 1;

| product\_name | ordered |

| ------------ | ------- |

| ramen | 8 |

---

\*\*Query #5\*\*

with q1 as

(select customer\_id, product\_name,

count(s.product\_id) as counts

from sales s join menu m

on s.product\_id=m.product\_id

group by customer\_id, product\_name

order by customer\_id, counts desc)

, q2 as

(select customer\_id, product\_name, counts,

rank() over(partition by customer\_id order by counts desc) as rk

from q1

)

select customer\_id, product\_name, counts as times\_ordered

from q2 where rk = 1;

| customer\_id | product\_name | times\_ordered |

| ----------- | ------------ | ------------- |

| A | ramen | 3 |

| B | ramen | 2 |

| B | curry | 2 |

| B | sushi | 2 |

| C | ramen | 3 |

---

\*\*Query #6\*\*

with q1 as(

select s.customer\_id, order\_date, product\_id

from sales s join members m

on s.customer\_id=m.customer\_id

where s.order\_date >= m.join\_date)

, q2 as (

select q1.customer\_id, product\_name,

rank() over (partition by q1.customer\_id order by q1.order\_date) as rk

from q1 join menu me

on q1.product\_id = me.product\_id

)

select customer\_id, product\_name

from q2

where rk=1;

| customer\_id | product\_name |

| ----------- | ------------ |

| A | curry |

| B | sushi |

---

\*\*Query #7\*\*

with q1 as(

select s.customer\_id, order\_date, product\_id

from sales s join members m

on s.customer\_id=m.customer\_id

where s.order\_date < m.join\_date)

, q2 as (

select q1.customer\_id, product\_name,

rank() over (partition by q1.customer\_id order by q1.order\_date desc, q1.product\_id desc) as rk

from q1 join menu me

on q1.product\_id = me.product\_id

)

select customer\_id, product\_name

from q2

where rk=1;

| customer\_id | product\_name |

| ----------- | ------------ |

| A | curry |

| B | sushi |

---

\*\*Query #8\*\*

select s.customer\_id, count(s.product\_id) as BeforeMemberordered, sum(price) as BeforeMembertotal

from sales s join members m

on s.customer\_id=m.customer\_id

join menu me

on s.product\_id=me.product\_id

where s.order\_date < m.join\_date

group by 1;

| customer\_id | beforememberordered | beforemembertotal |

| ----------- | ------------------- | ----------------- |

| B | 3 | 40 |

| A | 2 | 25 |

---

\*\*Query #9\*\*

select customer\_id,

sum(case when s.product\_id = 1 then price\*20

else price\*10 end) as RewardPoints

from sales s join menu m

on s.product\_id=m.product\_id

group by 1;

| customer\_id | rewardpoints |

| ----------- | ------------ |

| B | 940 |

| C | 360 |

| A | 860 |

---

\*\*Query #10\*\*

select s.customer\_id,sum(

case when (order\_date >= join\_date) and (order\_date < join\_date + integer '7') then price\*20

else (case when s.product\_id=1 then price\*20

else price\*10 end) end) as RewardPoints

from sales s join menu m

on s.product\_id=m.product\_id

join members me on s.customer\_id=me.customer\_id

where order\_date <= date '2021-01-31'

group by 1;

| customer\_id | rewardpoints |

| ----------- | ------------ |

| A | 1370 |

| B | 820 |

---

\*\*Query #11\*\*

select s.customer\_id, order\_date, product\_name, price,

case when order\_date < join\_date then 'N'

when order\_date >= join\_date then 'Y'

else 'N' end as Member

from sales s join menu m

on s.product\_id=m.product\_id

left join members me

on s.customer\_id=me.customer\_id

order by customer\_id, order\_date;

| customer\_id | order\_date | product\_name | price | member |

| ----------- | ------------------------ | ------------ | ----- | ------ |

| A | 2021-01-01T00:00:00.000Z | sushi | 10 | N |

| A | 2021-01-01T00:00:00.000Z | curry | 15 | N |

| A | 2021-01-07T00:00:00.000Z | curry | 15 | Y |

| A | 2021-01-10T00:00:00.000Z | ramen | 12 | Y |

| A | 2021-01-11T00:00:00.000Z | ramen | 12 | Y |

| A | 2021-01-11T00:00:00.000Z | ramen | 12 | Y |

| B | 2021-01-01T00:00:00.000Z | curry | 15 | N |

| B | 2021-01-02T00:00:00.000Z | curry | 15 | N |

| B | 2021-01-04T00:00:00.000Z | sushi | 10 | N |

| B | 2021-01-11T00:00:00.000Z | sushi | 10 | Y |

| B | 2021-01-16T00:00:00.000Z | ramen | 12 | Y |

| B | 2021-02-01T00:00:00.000Z | ramen | 12 | Y |

| C | 2021-01-01T00:00:00.000Z | ramen | 12 | N |

| C | 2021-01-01T00:00:00.000Z | ramen | 12 | N |

| C | 2021-01-07T00:00:00.000Z | ramen | 12 | N |

---

\*\*Query #12\*\*

with q1 as(

select s.customer\_id, order\_date, product\_name, price,

case when order\_date < join\_date then 'N'

when order\_date >= join\_date then 'Y'

else 'N' end as Member

from sales s join menu m

on s.product\_id=m.product\_id

left join members me

on s.customer\_id=me.customer\_id)

select \*,

case when Member = 'N' then Null

else rank() over(partition by q1.customer\_id, q1.Member order by order\_date)

end as ranking

from q1

order by customer\_id, order\_date;

| customer\_id | order\_date | product\_name | price | member | ranking |

| ----------- | ------------------------ | ------------ | ----- | ------ | ------- |

| A | 2021-01-01T00:00:00.000Z | sushi | 10 | N | |

| A | 2021-01-01T00:00:00.000Z | curry | 15 | N | |

| A | 2021-01-07T00:00:00.000Z | curry | 15 | Y | 1 |

| A | 2021-01-10T00:00:00.000Z | ramen | 12 | Y | 2 |

| A | 2021-01-11T00:00:00.000Z | ramen | 12 | Y | 3 |

| A | 2021-01-11T00:00:00.000Z | ramen | 12 | Y | 3 |

| B | 2021-01-01T00:00:00.000Z | curry | 15 | N | |

| B | 2021-01-02T00:00:00.000Z | curry | 15 | N | |

| B | 2021-01-04T00:00:00.000Z | sushi | 10 | N | |

| B | 2021-01-11T00:00:00.000Z | sushi | 10 | Y | 1 |

| B | 2021-01-16T00:00:00.000Z | ramen | 12 | Y | 2 |

| B | 2021-02-01T00:00:00.000Z | ramen | 12 | Y | 3 |

| C | 2021-01-01T00:00:00.000Z | ramen | 12 | N | |

| C | 2021-01-01T00:00:00.000Z | ramen | 12 | N | |

| C | 2021-01-07T00:00:00.000Z | ramen | 12 | N | |

---

[View on DB Fiddle](https://www.db-fiddle.com/f/m8poJtm4GHX57Fz3WmGyTv/2)