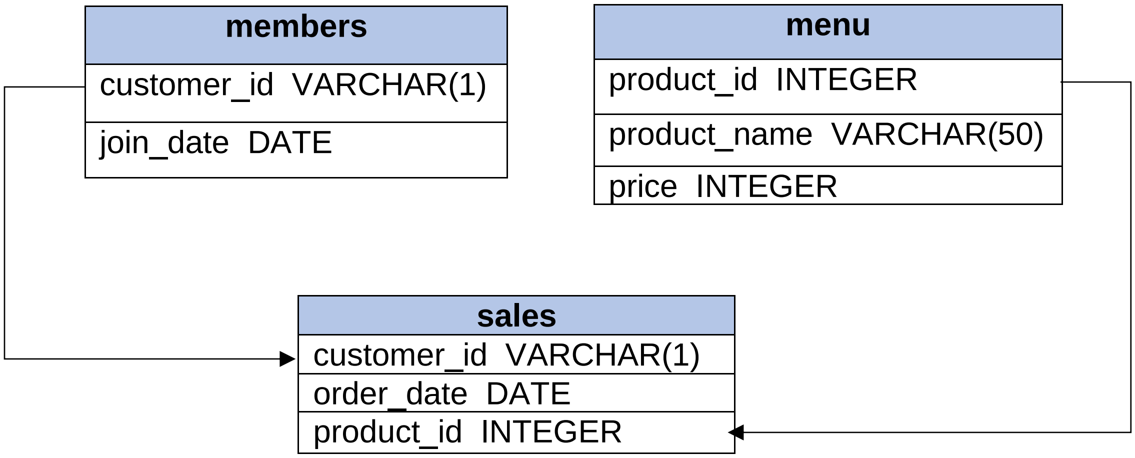
Hotel Visitors Analytics using PostgreSQL

Task: Hotel Manager 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.

Table Relationship:



members

customer_id VARCHAR(1)

join_date DATE

menu

product_id INTEGER

product_name VARCHAR(50)

price INTEGER

sales

customer_id VARCHAR(1)

order_date DATE

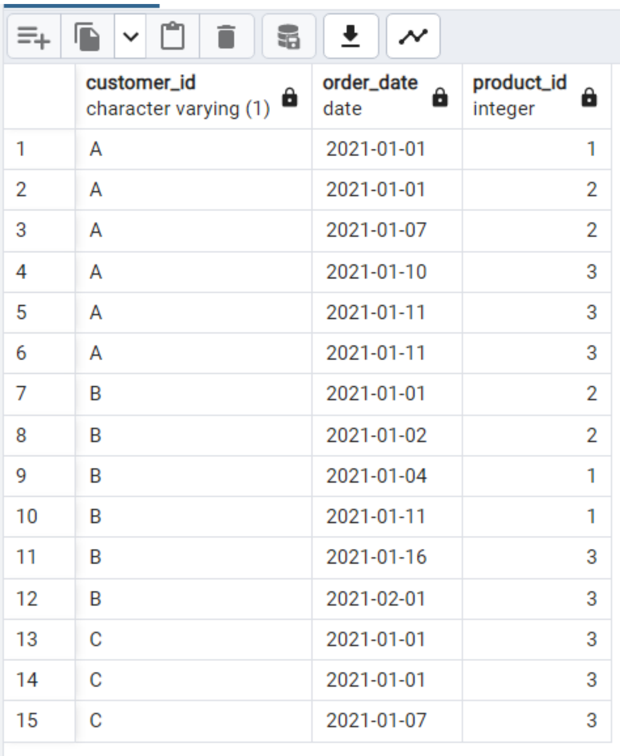
product_id INTEGER

Table Creation: -

```
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');
```

```
select * from sales;
```

Output: -

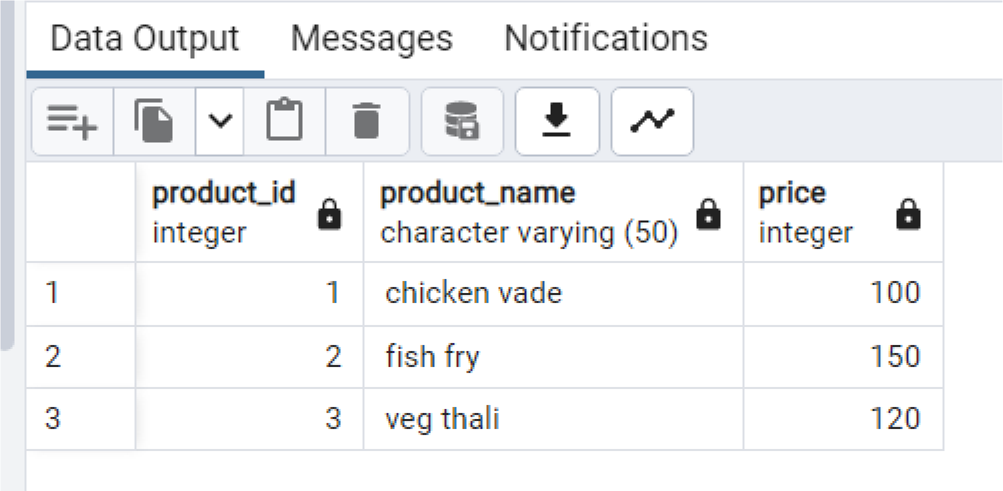


	customer_id character varying (1)	order_date date	product_id integer
1	A	2021-01-01	1
2	A	2021-01-01	2
3	A	2021-01-07	2
4	A	2021-01-10	3
5	A	2021-01-11	3
6	A	2021-01-11	3
7	B	2021-01-01	2
8	B	2021-01-02	2
9	B	2021-01-04	1
10	B	2021-01-11	1
11	B	2021-01-16	3
12	B	2021-02-01	3
13	C	2021-01-01	3
14	C	2021-01-01	3
15	C	2021-01-07	3

```
create table menu (
  "product_id" INTEGER,
  "product_name" VARCHAR(50),
  "price" INTEGER
);
insert into menu
  ("product_id", "product_name", "price")
values
  ('1', 'chicken vade', '100'),
  ('2', 'fish fry', '150'),
  ('3', 'veg thali', '120');

select * from menu;
```

Output: -

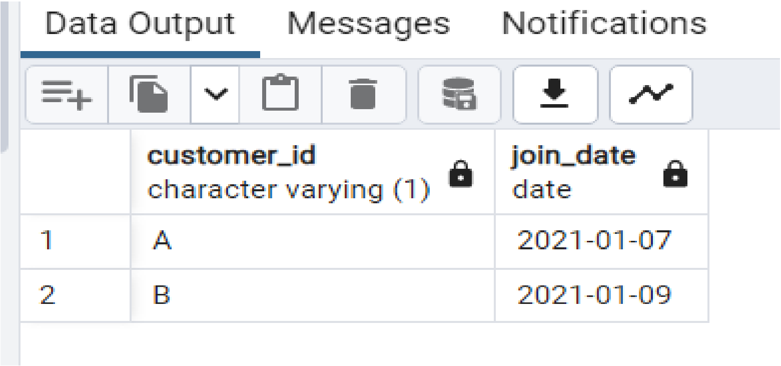


	product_id integer	product_name character varying (50)	price integer
1	1	chicken vade	100
2	2	fish fry	150
3	3	veg thali	120

```
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');
```

```
select * from members;
```

Output: -

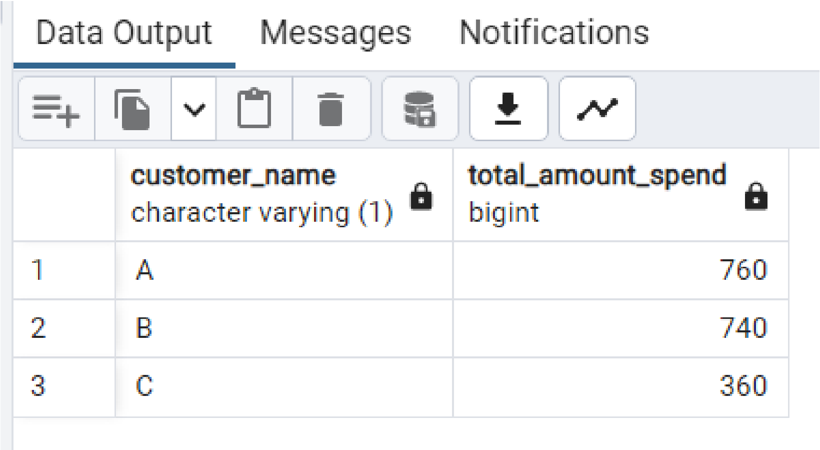


	customer_id character varying (1)	join_date date
1	A	2021-01-07
2	B	2021-01-09

1) what is the total amount each customer spends at the restaurant?

```
select sales.customer_id as customer_name,
                  sum(menu.price) as total_amount_spend
from sales
                  left join menu on sales.product_id=menu.product_id
group by sales.customer_id
order by sales.customer_id;
```

Output: -



	customer_name character varying (1)	total_amount_spend bigint
1	A	760
2	B	740
3	C	360

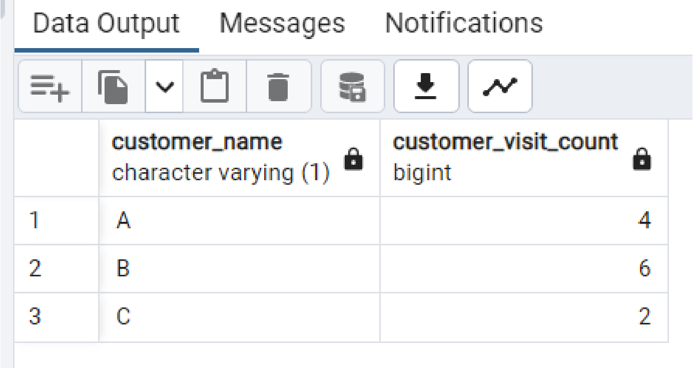
- Customer A spent 760.

- Customer B spent 740.
- Customer C spent 360.

2) How many days each customer visited the restaurant?

```
select customer_id as customer_name,
       count(distinct(order_date)) as customer_visit_count
from sales
group by customer_id
order by customer_id;
```

Output: -



	customer_name character varying (1)	customer_visit_count bigint
1	A	4
2	B	6
3	C	2

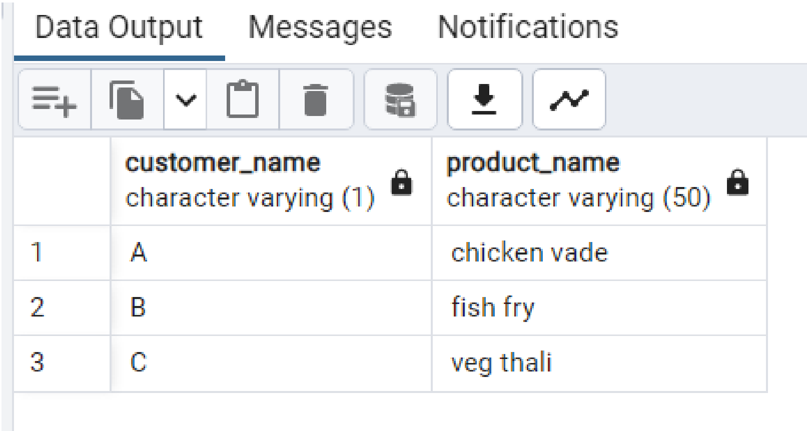
- Customer A visited 4 times.
- Customer B visited 6 times.
- Customer C visited 2 times.

3) what is the first item from the menue purchased by each customer?

```
select customer_name,
       product_name
from (
       select sales.customer_id as customer_name,
              menu.product_name,
              row_number() over (partition by sales.customer_id order by sales.order_date asc) as rn
       from sales
```

```
        left join menu on sales.product_id=menu.product_id
        ) x
where rn=1;
```

Output: -



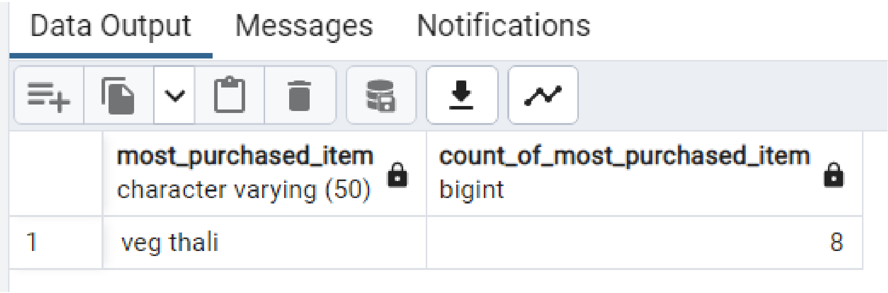
	customer_name character varying (1)	product_name character varying (50)
1	A	chicken vade
2	B	fish fry
3	C	veg thali

- Customer A’s first order is chicken vade.
- Customer B's first order is fish fry.
- Customer C's first order is veg thali.

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

```
select  menu.product_name as most_purchased_item,
        count(sales.product_id) as count_of_most_purchased_item
from sales
        left join menu on sales.product_id=menu.product_id
group by menu.product_name
order by count_of_most_purchased_item desc
limit 1;
```

Output: -



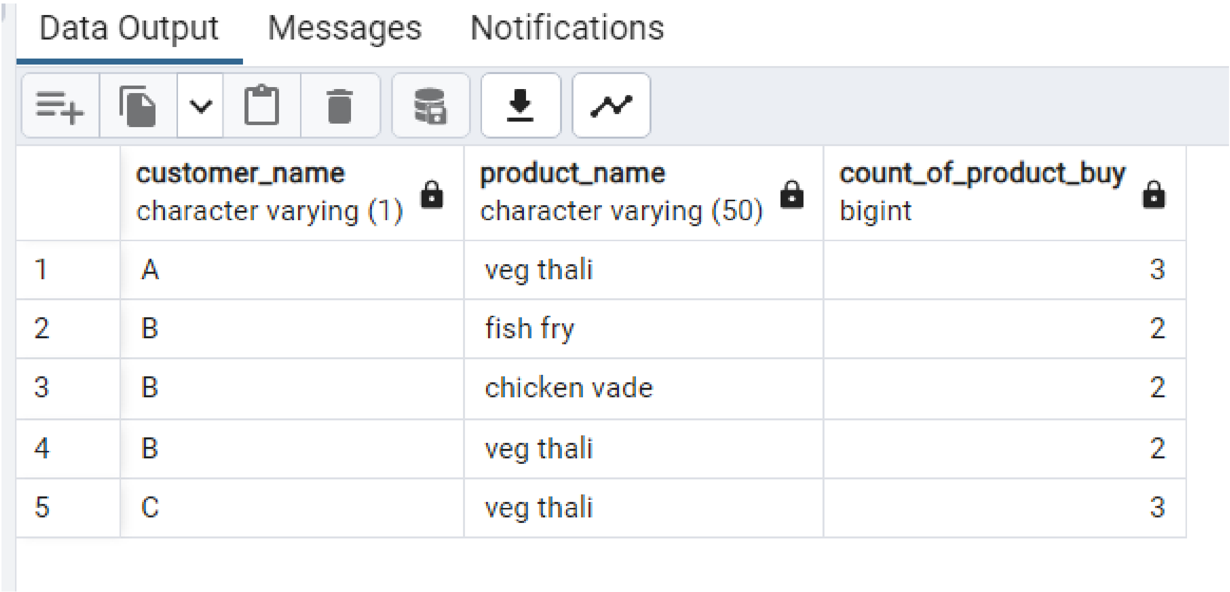
	most_purchased_item character varying (50)	count_of_most_purchased_item bigint
1	veg thali	8

- Most purchased item on the menu is veg thali which is 8 times. Yummy!

5) Which item was the most popular for each customer?

```
with most_popular_cust as (
        select sales.customer_id as customer_name,
                menu.product_name ,
                count(menu.product_name) as count_of_product_buy,
                dense_rank() over(partition by sales.customer_id order by count(sales.customer_id)
desc) as rn
        from sales
        left join menu on sales.product_id=menu.product_id
        group by sales.customer_id ,menu.product_name
        order by sales.customer_id
        )
select customer_name,
        product_name,
        count_of_product_buy
from most_popular_cust
where rn=1;
```

Output: -



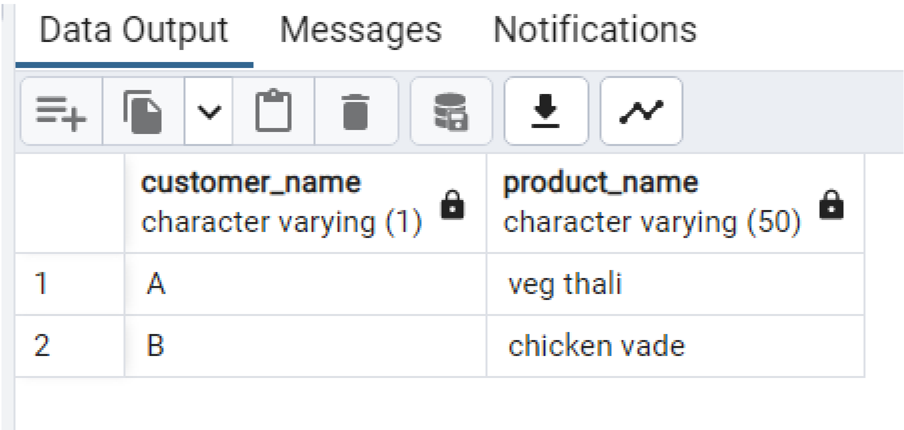
	customer_name character varying (1)	product_name character varying (50)	count_of_product_buy bigint
1	A	veg thali	3
2	B	fish fry	2
3	B	chicken vade	2
4	B	veg thali	2
5	C	veg thali	3

- Customer A and C's favorite item is veg thali.
- Customer B enjoys all items on the menu. He/she is a true foodie.

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

```
with cust_after_member as (
        select  members.customer_id,
                sales.product_id,
                row_number() over (partition by members.customer_id order by sales.order_date )as rn
        FROM members
                INNER JOIN sales ON members.customer_id = sales.customer_id
                                 AND sales.order_date > members.join_date
)
select  cust_after_member.customer_id as customer_name,
        menu.product_name
from    cust_after_member
        left join menu on cust_after_member.product_id=menu.product_id
where cust_after_member.rn = 1
order by cust_after_member.customer_id asc;
```

Output: -



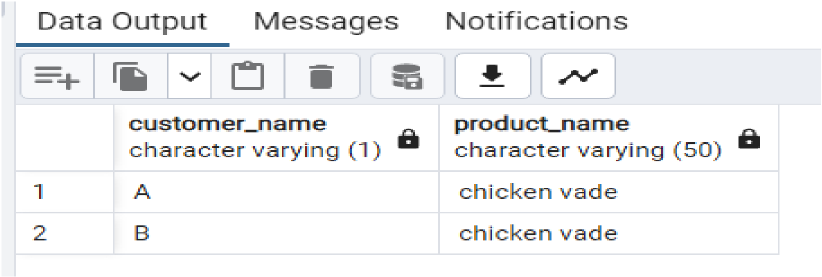
	customer_name character varying (1)	product_name character varying (50)
1	A	veg thali
2	B	chicken vade

- Customer A's first order as a member is veg thali.
- Customer B's first order as a member is chicken vade.

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

```
with purchased_item_before_member as (
        select  members.customer_id,
                sales.product_id,
                row_number() over(partition by members.customer_id order by sales.order_date desc ) as rn
        from members
                inner join sales on members.customer_id=sales.customer_id
                                and sales.order_date < members.join_date
        )
select  purchased_item_before_member.customer_id as customer_name,
        menu.product_name
from purchased_item_before_member
        left join menu on purchased_item_before_member.product_id=menu.product_id
where rn=1
order by purchased_item_before_member.customer_id asc;
```

Output: -



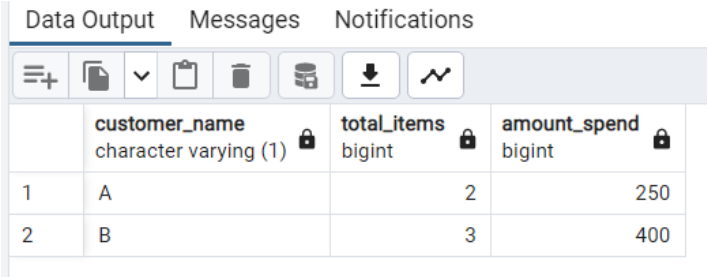
	customer_name character varying (1)	product_name character varying (50)
1	A	chicken vade
2	B	chicken vade

- Both customer’s last order before becoming members are chicken vade.

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

```
select  sales.customer_id as customer_name,
        count(sales.product_id) as total_items,
        sum(menu.price) as amount_spend
from sales
        inner join members on sales.customer_id=members.customer_id
                                    and sales.order_date < members.join_date
        left join menu on sales.product_id=menu.product_id
group by sales.customer_id
order by sales.customer_id;
```

Output: -



	customer_name character varying (1)	total_items bigint	amount_spend bigint
1	A	2	250
2	B	3	400

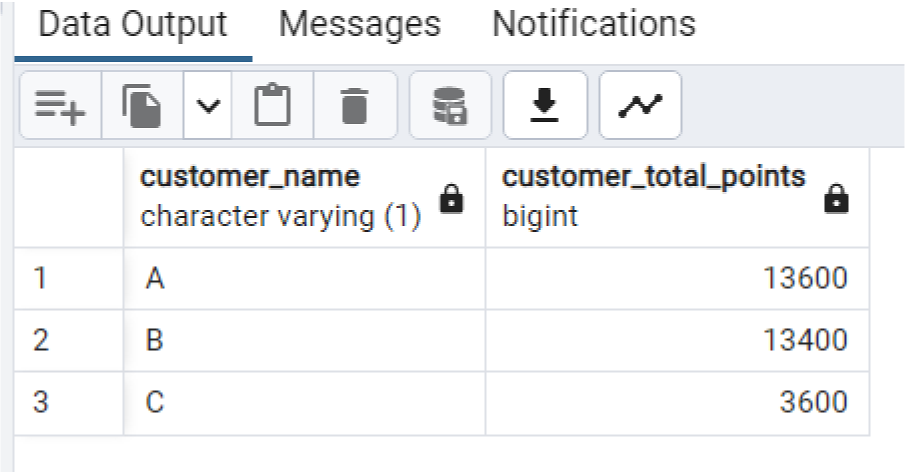
Before becoming members,

- Customer A spent 250 on 2 items.
- Customer B spent 400 on 3 items.

9) If each 50 spent equates to 10 points and fish fry has a 3x points multiplier — how many points would each customer have?

```
with points_table as (
        select  menu.product_id,
                case when product_id =2 then price * 30 else price * 10
                end as points
        from menu
        )
select  sales.customer_id as customer_name,
        sum(points_table.points) as customer_total_points
from sales
        left join points_table on sales.product_id = points_table.product_id
group by sales.customer_id
order by sales.customer_id;
```

Output: -



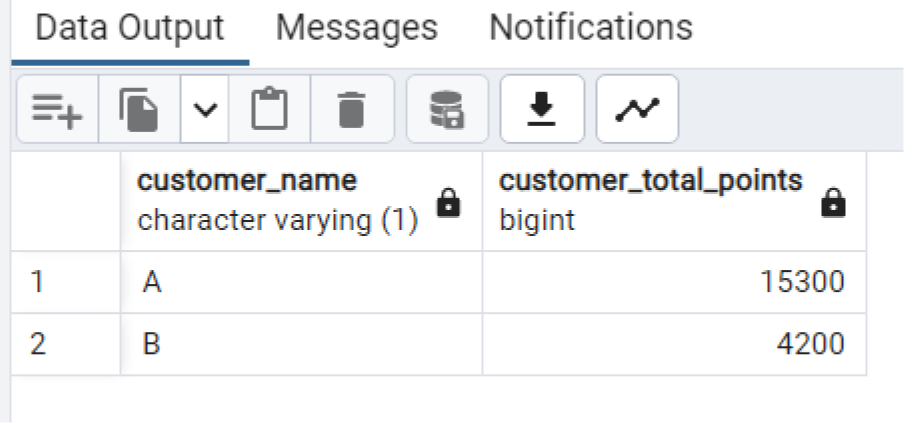
	customer_name character varying (1)	customer_total_points bigint
1	A	13600
2	B	13400
3	C	3600

- Total points for Customer A is 13600.
- Total points for Customer B is 13400.
- Total points for Customer C is 3600.

10) In the first week after a customer joins the program (including their join date) they earn 3x points on all items, not just fish fry — how many points do customer A and B have at the end of January?

```
with date_table as (
        select  customer_id,
                join_date,
                join_date + 6 as valid_date,
                (DATE_TRUNC('month', join_date) + interval '1 month' - interval '1 day' )::date as
last_date
        from members
        )
select  sales.customer_id as customer_name,
        sum(case when menu.product_name = 'fish fry' then 3 * 10 * menu.price
                  when sales.order_date between date_table.join_date and date_table.valid_date then
                  3 * 10 *          menu.price  else 10 * menu.price
               end) as customer_total_points
from sales
        inner join date_table on sales.customer_id=date_table.customer_id
                                and sales.order_date >= date_table.join_date
                                and sales.order_date <= date_table.last_date
        left join menu on sales.product_id = menu.product_id
group by sales.customer_id
order by sales.customer_id;
```

Output: -



	customer_name character varying (1)	customer_total_points bigint
1	A	15300
2	B	4200

- Total points for Customer A is 15300.
- Total points for Customer B is 4200.