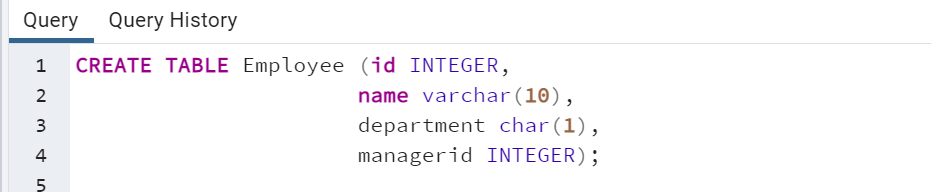
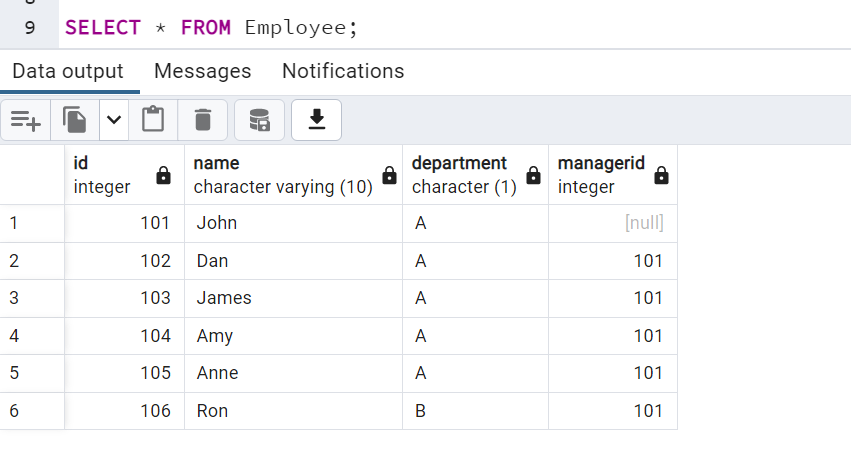
SQL Assignment

1. Employee Table





SELECT Name

FROM Employee

WHERE id =

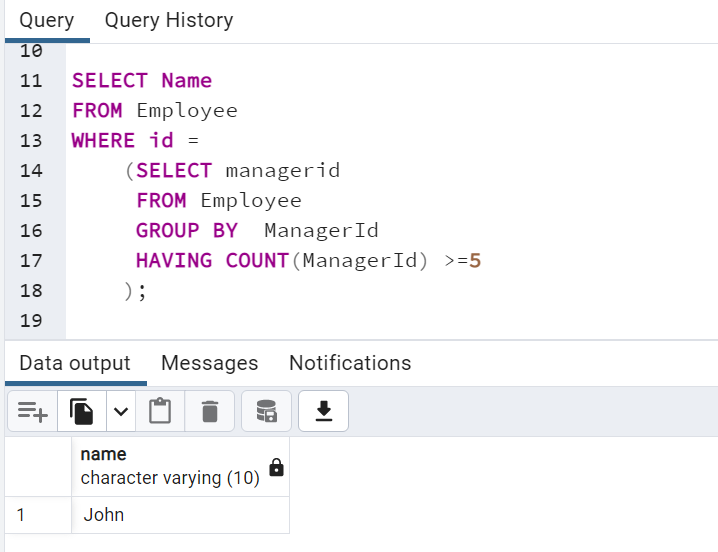
(SELECT managerid

FROM Employee

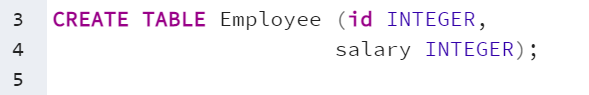
GROUP BY ManagerId

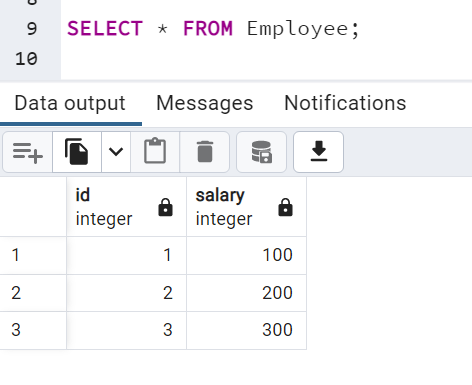
HAVING COUNT(ManagerId) >=5

);



2. Employee Table





CREATE FUNCTION getNthHighestSalary(N INT) RETURNS INT

BEGIN

DECLARE M INT;

SET M = N - 1;

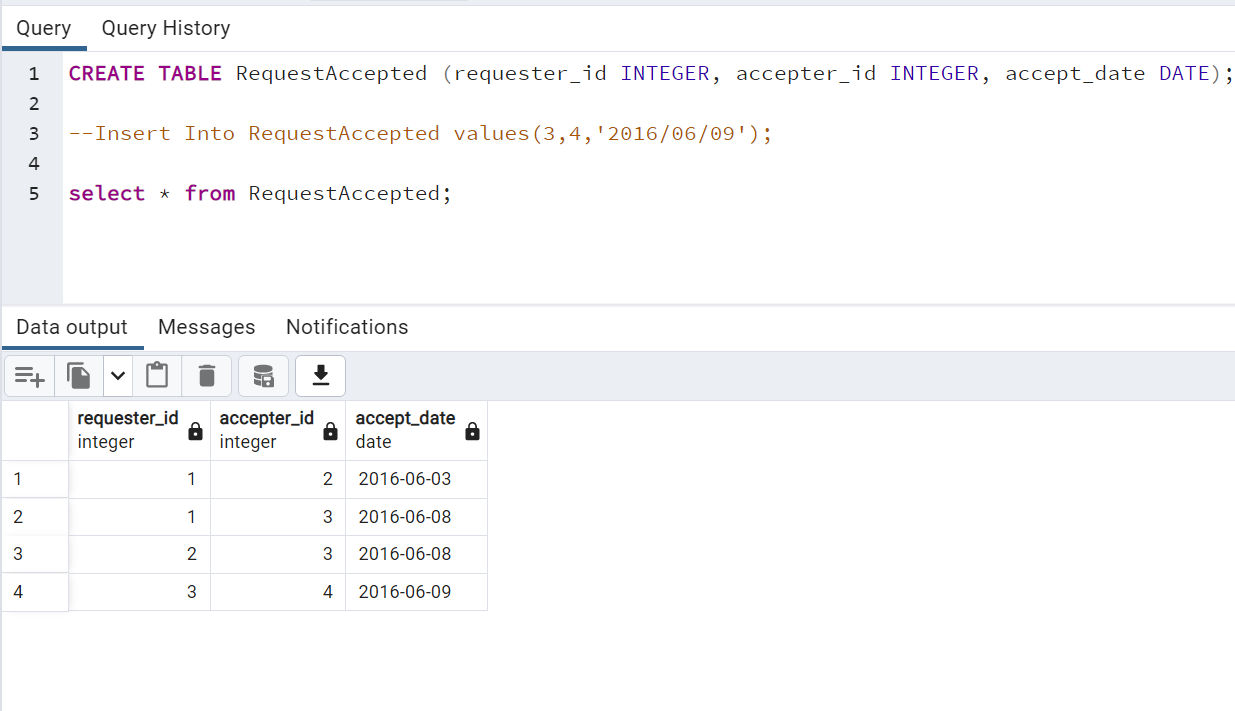
RETURN (

SELECT DISTINCT Salary FROM Employee ORDER BY Salary DESC LIMIT M, 1

);

END

3. Table RequestAccepted

SELECT id, COUNT(\*) AS num

FROM (

SELECT requester\_id AS id FROM requestaccepted

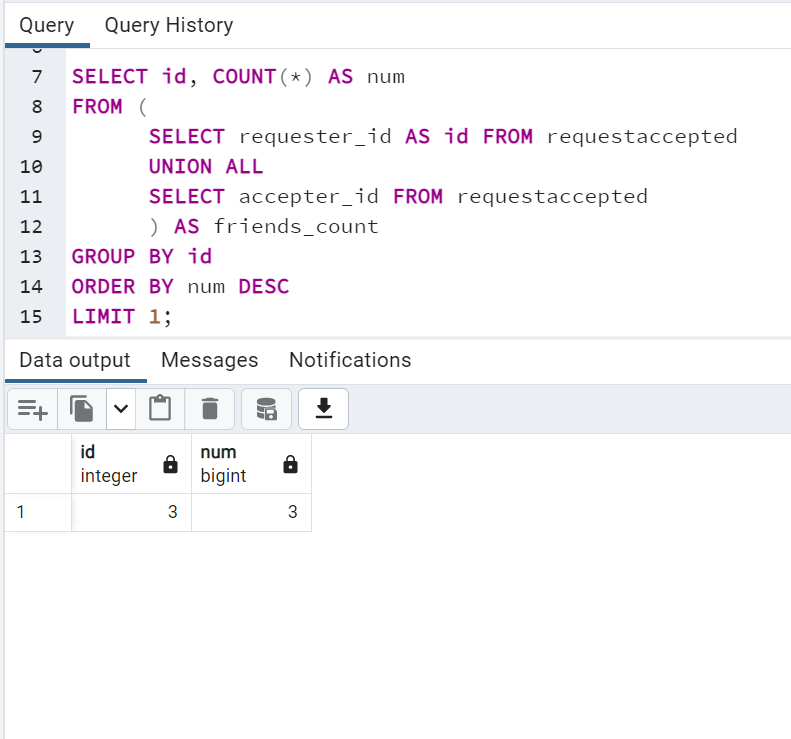
UNION ALL

SELECT accepter\_id FROM requestaccepted

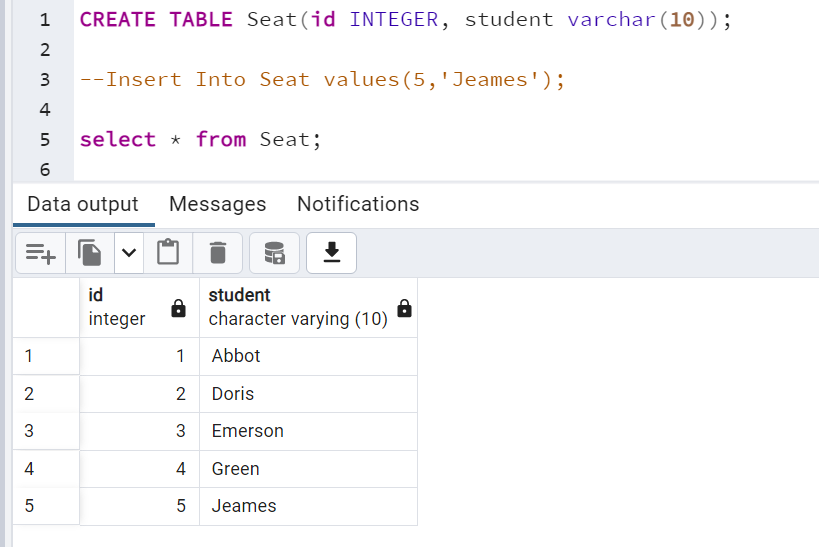
) AS friends\_count

GROUP BY id

ORDER BY num DESC LIMIT 1;



4. Table SEAT

 SELECT

(CASE

WHEN MOD(id, 2) != 0 AND counts != id THEN id + 1

WHEN MOD(id, 2) != 0 AND counts = id THEN id

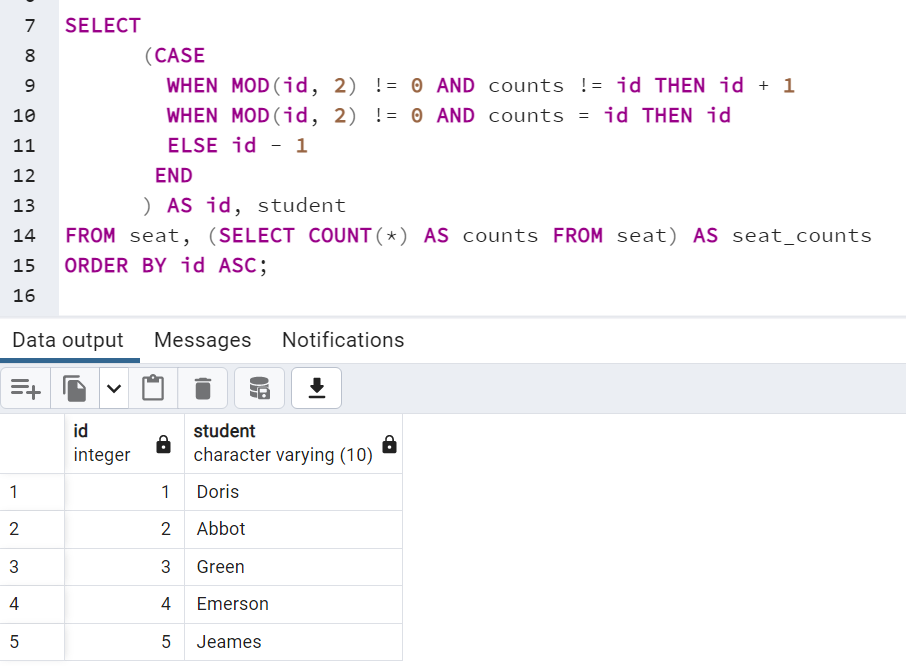
ELSE id - 1

END

) AS id, student

FROM seat, (SELECT COUNT(\*) AS counts FROM seat) AS seat\_counts

ORDER BY id ASC;



5. Table Customer

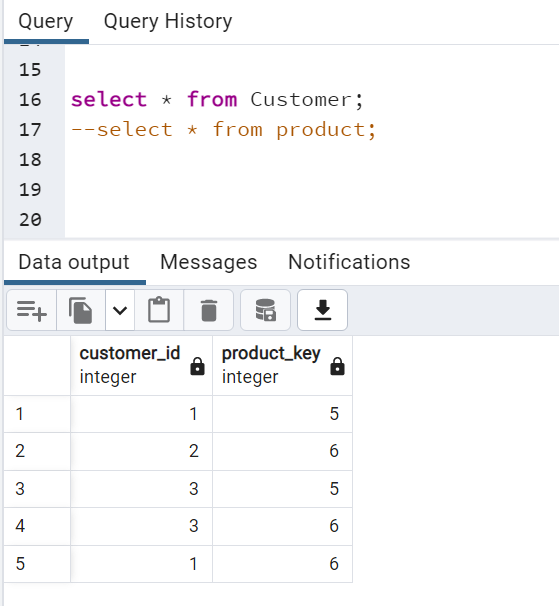
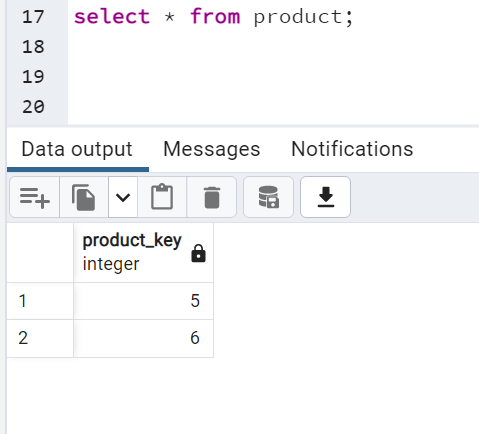


Table Product

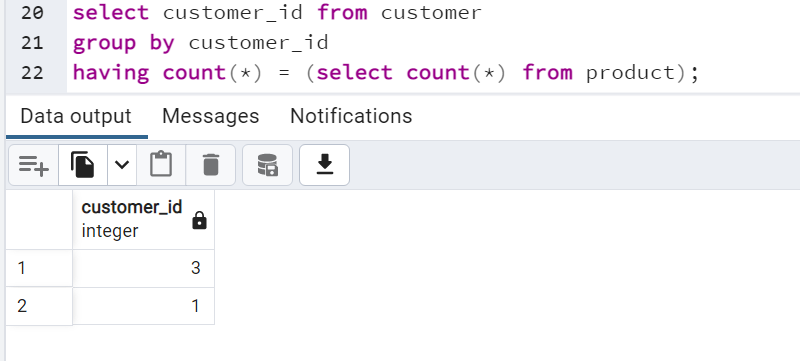


SELECT customer\_id

FROM customer

GROUP BY customer\_id

HAVING COUNT( DISTINCT product\_key) = (SELECT COUNT(\*) FROM product)



6. Table Users

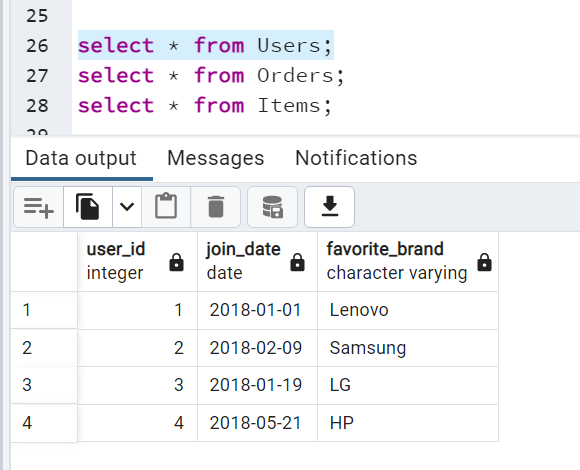


Table Orders

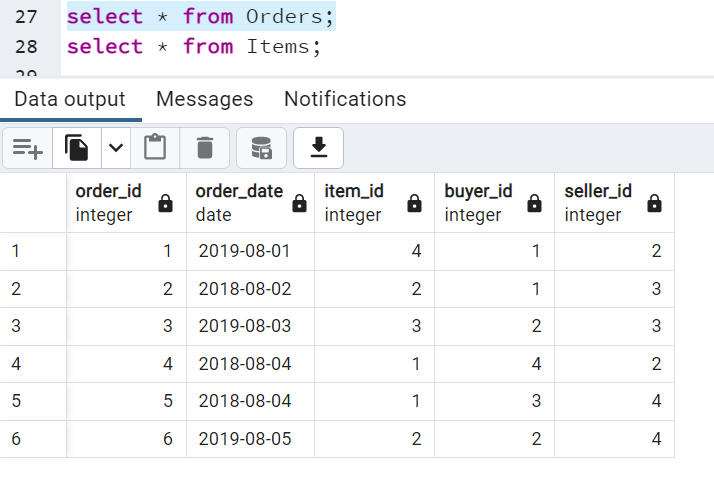
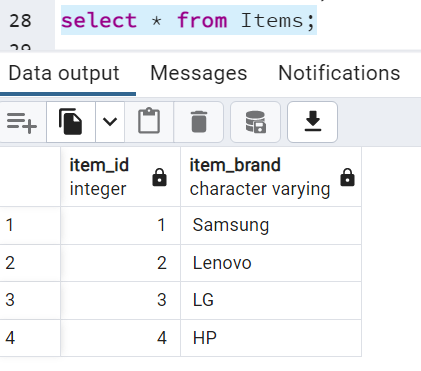


Table Items



SELECT U.user\_id AS buyer\_id, U.join\_date,

SUM(CASE WHEN extract(YEAR from O.order\_date) = '2019' THEN 1 ELSE 0 END) AS orders\_in\_2019

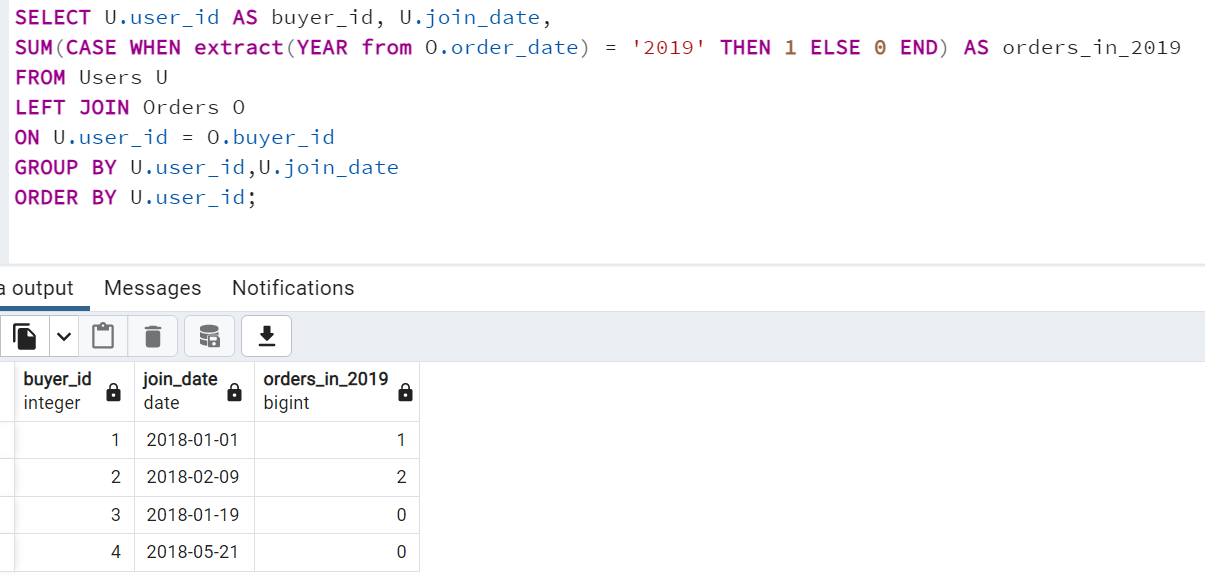
FROM Users U

LEFT JOIN Orders O

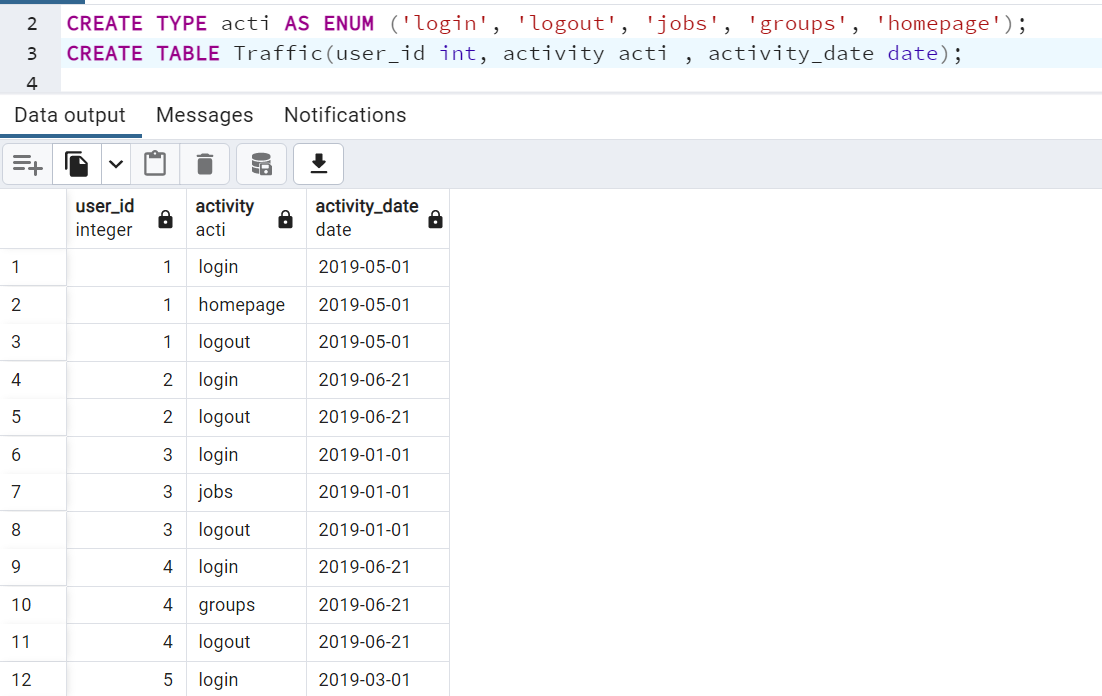
ON U.user\_id = O.buyer\_id

GROUP BY U.user\_id,U.join\_date

ORDER BY U.user\_id;



7. Table Traffic



SELECT login\_date, COUNT(user\_id) AS user\_count

FROM

(SELECT user\_id, MIN(activity\_date) AS login\_date

FROM Traffic

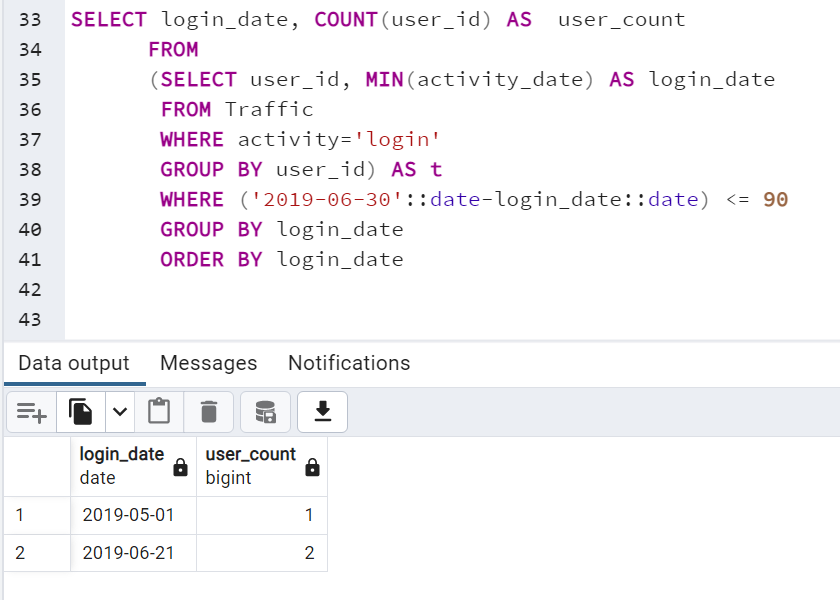
WHERE activity='login'

GROUP BY user\_id) AS t

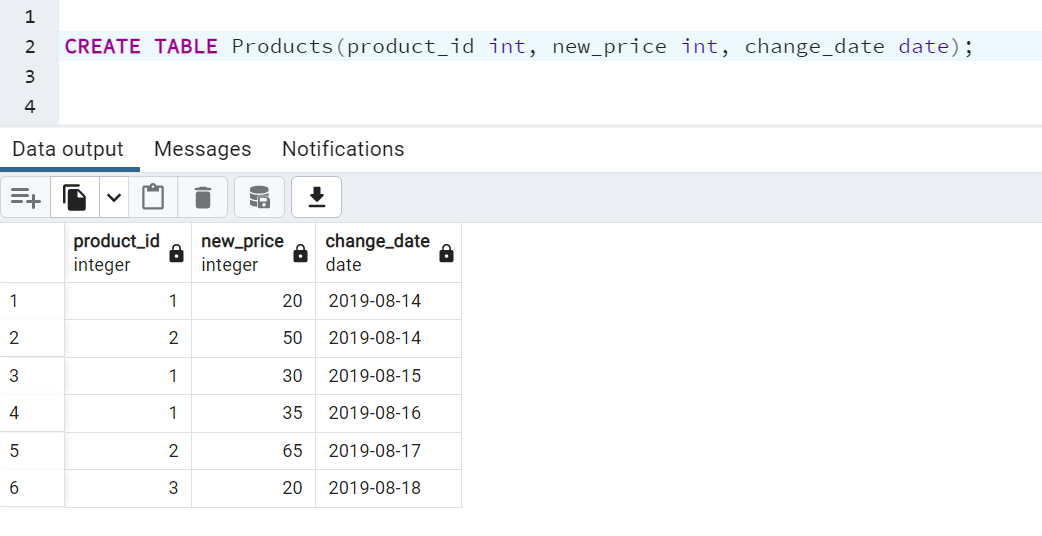
WHERE ('2019-06-30'::date-login\_date::date) <= 90

GROUP BY login\_date

ORDER BY login\_date



8. Table Products



select \* from (

select product\_id, new\_price as price from Products

where (product\_id, change\_date) in (

select product\_id, max(change\_date) from Products

where change\_date <= '2019-08-16'

group by product\_id

)

union

select distinct product\_id, 10 as price from Products

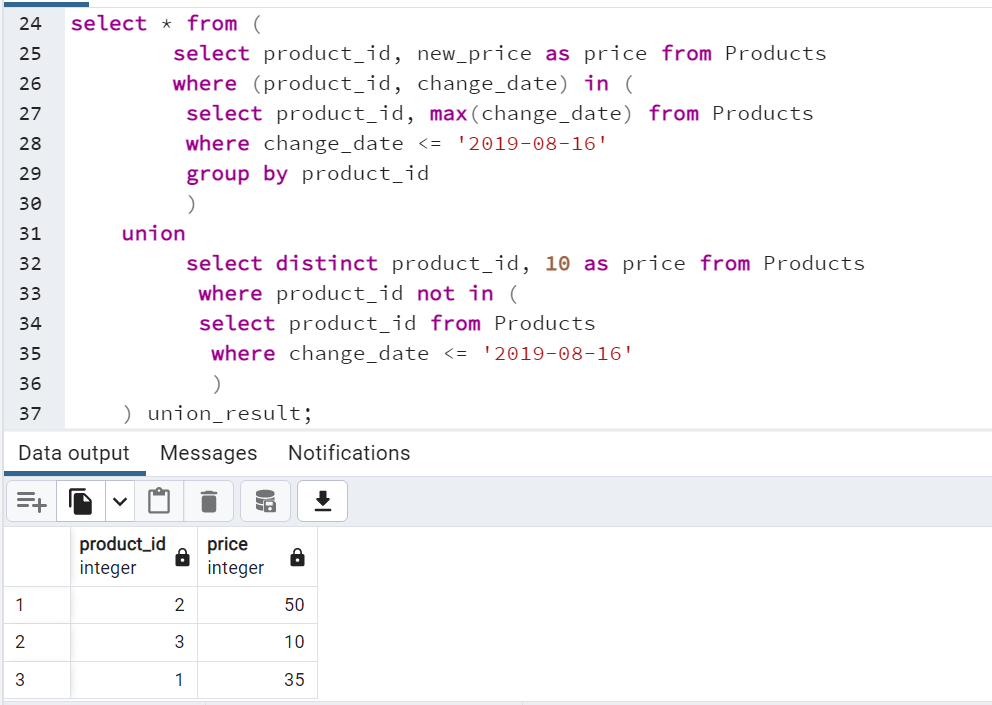
where product\_id not in (

select product\_id from Products

where change\_date <= '2019-08-16'

)

) union\_result;



9. Table Transactions

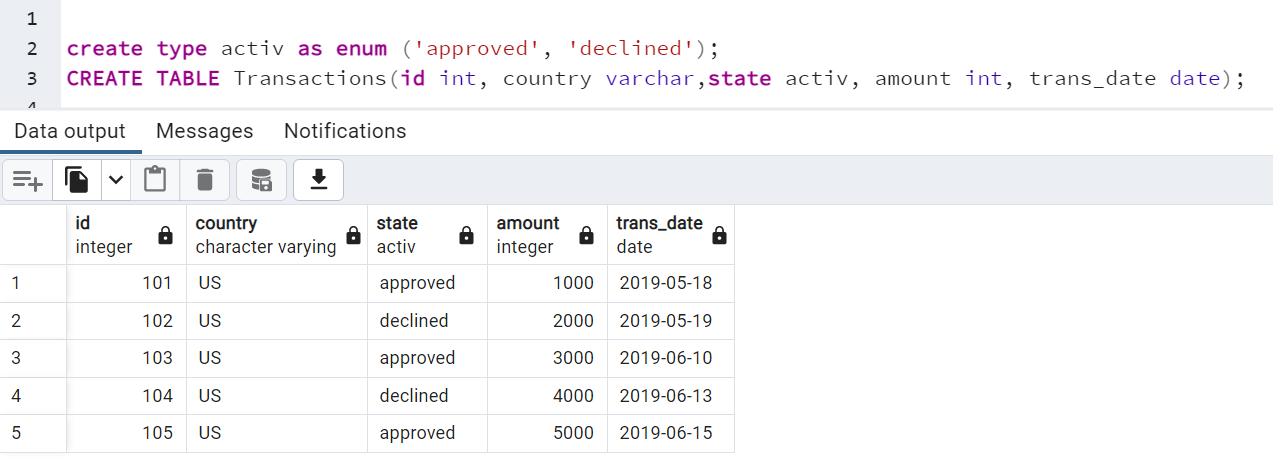
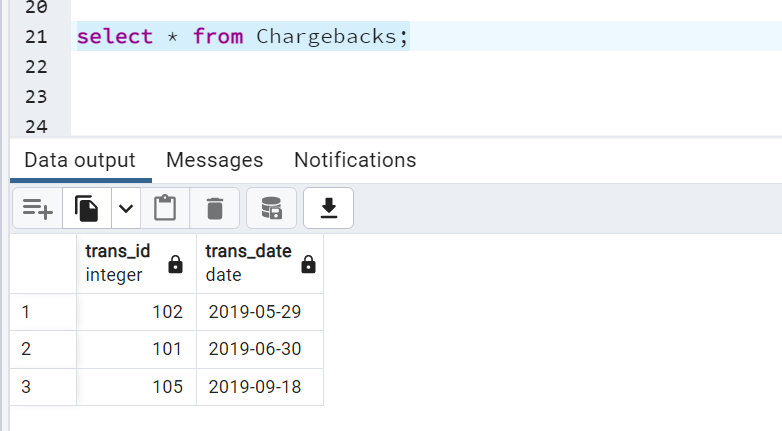


Table Chargebacks



select

month

,country

,sum(case when state = 'approved' then 1 else 0 end) as approved\_count

,sum(case when state = 'approved' then amount else 0 end) as approved\_amount

,sum(case when state = 'chargeback' then 1 else 0 end) as chargeback\_count

,sum(case when state = 'chargeback' then amount else 0 end) as chargeback\_amount

from ((

select

TO\_CHAR(trans\_date, 'YYYY-MM') as month

,country

,amount

,'approved' as state

from Transactions

where state = 'approved'

)

union all

(select TO\_CHAR(Chargebacks.trans\_date, 'YYYY-MM') as month,

country,

amount,

'chargeback' as state

from Transactions

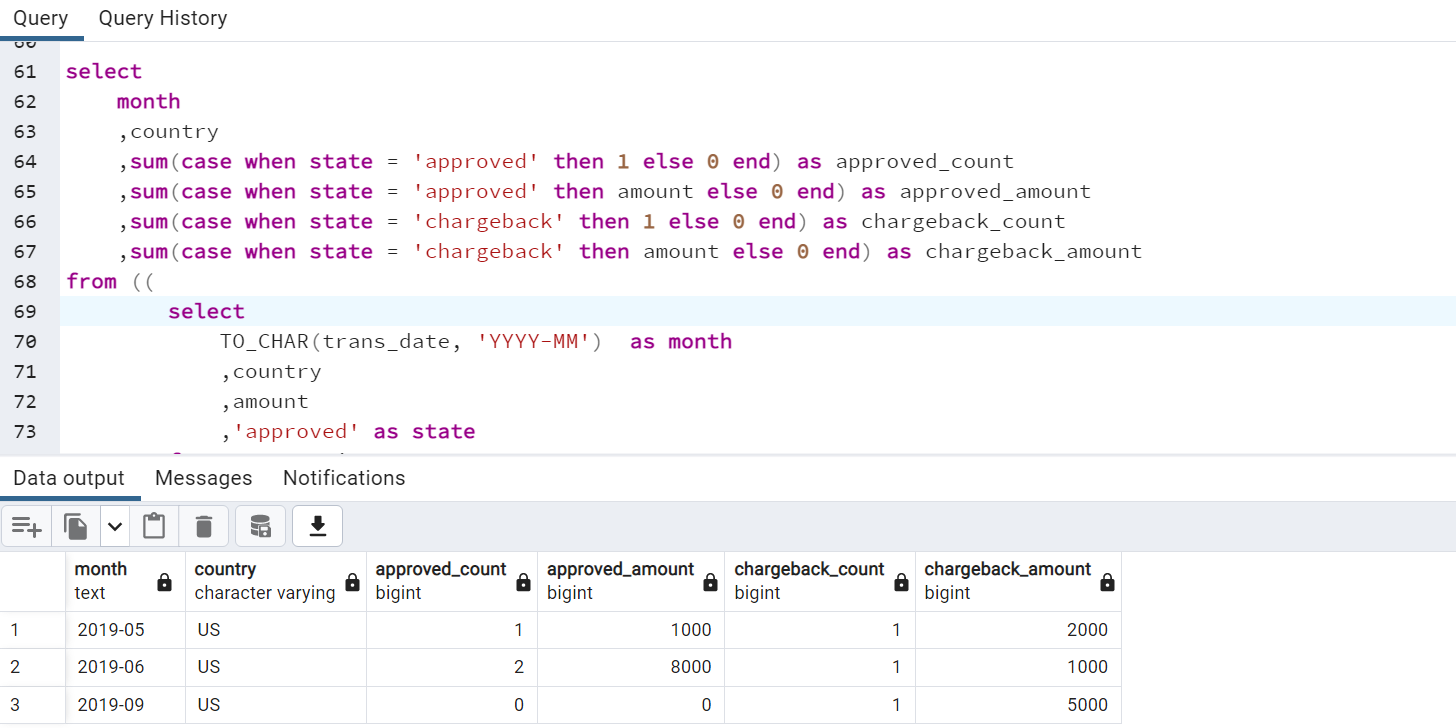
inner join Chargebacks

on Transactions.id = Chargebacks.trans\_id

)) temp

group by temp.month, temp.country

order by month;



10. Table Teams

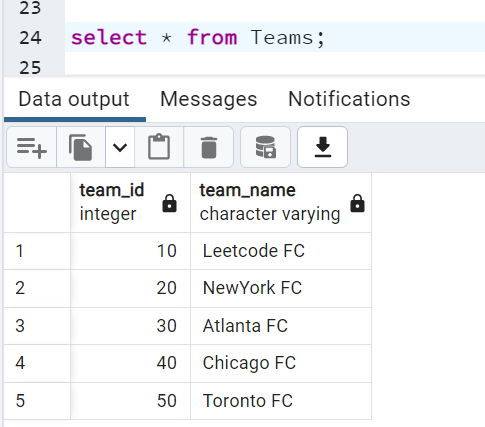
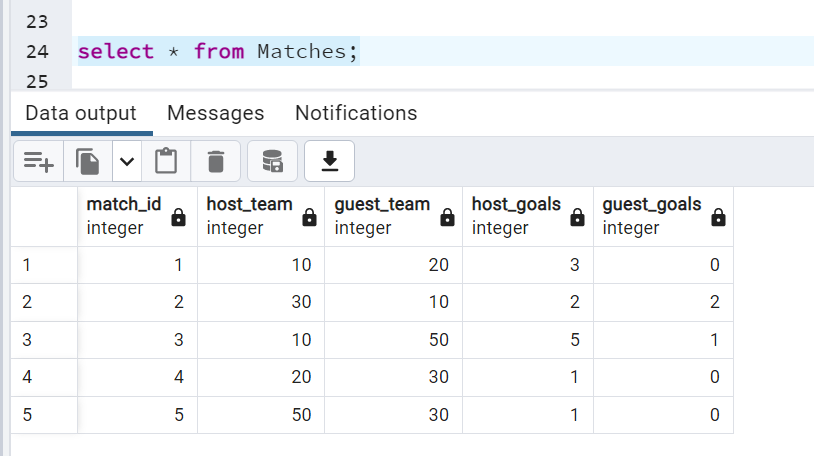


Table Matches



select t.team\_id, t.team\_name,

sum(case when t.team\_id = m.host\_team and m.host\_goals > m.guest\_goals then 3

when t.team\_id = m.host\_team and m.host\_goals = m.guest\_goals then 1

when t.team\_id = m.guest\_team and m.host\_goals < m.guest\_goals then 3

when t.team\_id = m.guest\_team and m.host\_goals = m.guest\_goals then 1

else 0 end) as num\_points

from Matches m

right join Teams t

on m.host\_team = t.team\_id or m.guest\_team = t.team\_id

group by team\_id, team\_name

order by num\_points desc, team\_id

