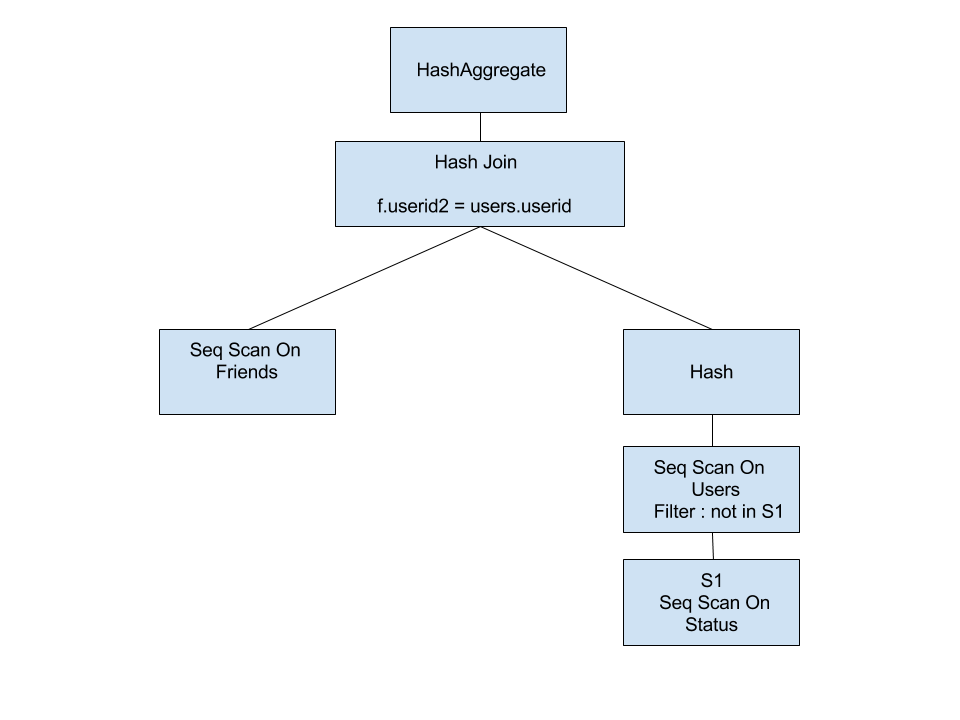
**Answer to Question 1**

Join doesn’t filter the resulting table which is why you are getting everyone regardless of what their name is. In addition, count(\*) counts all the rows so you filter the count so it doesn’t count the rows with NULL.

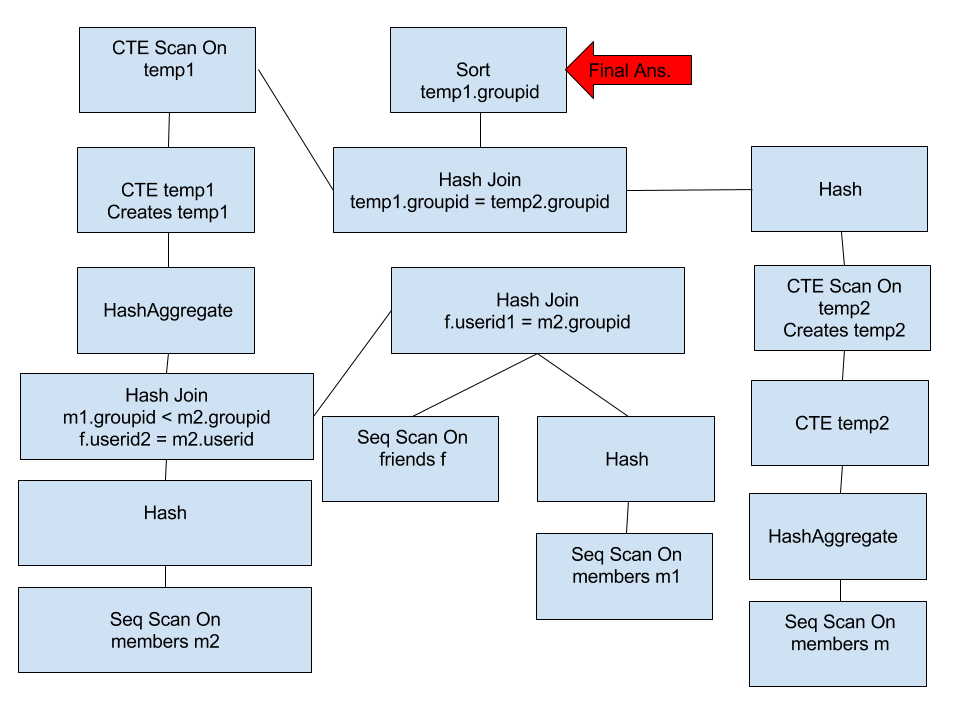
The updated query is therefore:

select u.userid, name, count(s.userid) from users u left outer join status s on (u.userid = s.userid) where u.name like 'Anthony%' group by u.userid, name order by u.userid;

**Answer to Question 2**

****

**Answer to Question 3**

****

**Answer to Question 4**

The cardinality estimates made by the optimizer aren’t even close to the actual rows generated for half of the operations. For example, the sequence scan on status is estimated to have 21 rows but in actuality it has 602 rows. In addition, the hash join operator at the top of the tree estimates only 14 rows but when the query is actually run there are 748 tuples. Some of the estimates are correct though. For example, the aggregate operator is very close to the actual tuples it receives.

**Answer to Question 5**

**Create NumberOfStatusUpdates Table**

create table NumberOfStatusUpdates(userid, user\_name, num\_updates)

select u.userid, name, count(\*)

from users u join status s on (u.userid = s.userid)

group by u.userid, name

order by u.userid;

**Create function used in trigger**

create or replacte function update\_row()

returns trigger as

$BODY$

declare

current\_updates integer;

current\_name char(20);

begin

if (TG\_OP = ‘INSERT’) then

select num\_updates into current\_update from NumberOfStatusUpdates where userid = NEW.userid;

select name into current\_name from Users where userid = NEW.userid;

if current\_count is null then

insert into NumberOfStatusUpdates values(NEW.userid, current\_name, 1);

else

update NumberOfStatusUpdates set num\_updates = num\_updates + 1 where userid = NEW.userid;

end if;

elsif (TG\_OP = ‘DELETE’) then

select num\_updates into current\_count from NumberOfStatusUpdates where userid = OLD.userid;

if current\_count = 1 then

delete from NumberOfStatusUpdates where userid = OLD.userid;

else

update NumberOfStatusUpdates set num\_updates = num\_updates - 1 where userid = OLD.userid;

end if;

end if;

return null;

end;

$BODY$

Language plpgsql;

**Create Trigger**

CREATE TRIGGER update\_trigger

AFTER INSERT OR DELETE

ON Status

FOR EACH ROW

EXECUTE PROCEDURE update\_row();

**Answer to Question 1**