Ben Reichert
Assignment 5
B461 – Gucht

1. select p1.pid, p1.name
   from   person p1, worksfor w1
   where p1.pid = w1.pid and w1.cname = 'Google' and
       exists (select 1
           from person p2, worksfor w2
           where p2.pid = w2.pid and
               (p1.pid,p2.pid) in (select k.pid1,k.pid2 from knows k) and
               w1.salary < w2.salary);

RA:

select distinct p1.pid, p1.name
from person p1 inner join worksfor w1 on (p1.pid = w1.pid)
   inner join worksfor w2 on (w1.salary < w2.salary)
   inner join person p2 on (p2.pid = w2.pid)
   inner join knows k on ((p1.pid, p2.pid) = (k.pid1, k.pid2))
where w1.cname = 'Google';

$$\pi_{p_1.pid,p_1.name}(P_1 \bowtie_{p_1.pid=w_1.pid} W_1 \bowtie w_1.salary < w_2.salary W_2 \bowtie w_2.pid = p_2.pid$$
$$P_2 \bowtie_{(p_1.pid,p_2.pid)=(k.pid1,k.pid2)} K(\sigma_{w_1.cname='Google'}))$$

Optimized:
$$\pi_{pid,name}(\pi_{pid,name}(P) \bowtie \pi_{pid}(W_1 \times W_2 \times P_2) \bowtie \pi_{pid1,pid2}(\sigma_{(pid,pid)=(pid1,pid2)}(K \times P_1 \times P_2)$$
$$\wedge cname =' Google' \wedge salary < salary(W_1 \times W_2)))$$

2. select p.pid
   from   person p
   where p.pid = SOME (select ps.pid
               from   personSkill ps
               where ps.skill = 'Programming' or ps.skill = 'Networks') and
           p.pid <> ALL (select w.pid
               from   worksFor w
               where w.cname = 'Amazon') and
           not exists (select p1.pid
               from   person p1
               where p1.city = 'Indianapolis' and
                   p1.pid in (select k.pid2 from knows k where k.pid1 = p.pid));

select distinct p.pid
from person p, worksfor w, personskill ps
where w.cname = 'Amazon' and p.pid = ps.pid and (ps.skill = 'Programming' or ps.skill =
'Networks') and p.pid != w.pid
   except
select p.pid
from person p
where exists (select p1.pid
           from person p1, knows k
           where p1.city = 'Indianapolis' and p1.pid = k.pid2 and k.pid1 = p.pid);

RA:
select distinct p.pid

Ben Reichert
Assignment 5
B461 – Gucht

from person p inner join personskill ps on (p.pid = ps.pid)
   inner join worksfor w on (p.pid <> w.pid)
where w.cname = 'Amazon' and ps.skill = 'Programming' or ps.skill = 'Networks'
except
select distinct p.pid
from person p inner join knows k on (k.pid1 = p.pid)
   inner join person p1 on (p1.pid = k.pid2)
where p1.city = 'Indianapolis';

$$\pi_{p.pid}(P \bowtie_{p.pid=ps.pid} S \bowtie_{p.pid \neq w.pid} W(\sigma_{w.cname='Amazon' \wedge ps.skill='Programming' \vee ps.skill='Networks'}))$$
$$-\pi_{p.pid}(P \bowtie_{p.pid=k.pid1} K \bowtie_{k.pid2=p1.pid} P_1(\sigma_{p_1.city='Indianapolis'}))$$

Optimized:
$$\pi_{pid}(\pi_{pid}(\sigma_{pid=pid}(P \times S)) \bowtie \pi_{pid}(\sigma_{pid \neq pid}(P \times W)) \wedge$$
$$cname =' Amazon' \wedge skill =' Programming' \vee skill =' Networks') - \pi_{pid}$$
$$(\pi_{pid}(\sigma_{pid=pid1}(P \times K)) \bowtie \pi_{pid}(\sigma_{pid2=pid}(K \times P_1)) \wedge city =' Indianapolis')$$

3. select p1.pid, p2.pid
   from   person p1, person p2
   where (p1.pid, p2.pid) in (select k.pid1, k.pid2 from knows k) and
      not p2.birthyear > SOME (select p.birthyear
               from   person p
              where p.pid in (select k.pid2
                    from   knows k
                    where k.pid1 = p1.pid));

RA:
select p1.pid, p2.pid
from person p1, person p2, knows k
where p1.pid = k.pid1 and p2.pid = k.pid2
  except
select p1.pid, p2.pid
from person p1, person p2, knows k, person pp, knows kk
where p1.pid = k.pid1 and p2.pid = k.pid2 and pp.pid = kk.pid2 and kk.pid1 = p1.pid and
p2.birthyear > pp.birthyear

$$\pi_{p_1.pid,p_2.pid}(P_1 \bowtie_{p_1.pid=k.pid1} K \bowtie_{k.pid2=p_2.pid} P_2)-$$
$$\pi_{p_1.pid,p_2.pid}(P_1 \bowtie_{p_1.pid=k.pid1} K \bowtie_{k.pid2=p_2.pid} P_2$$
$$\bowtie_{p_1.pid=k_2.pid} K_2 \bowtie_{k_2.pid2=p_3.pid} P_3(\sigma_{p_2.birthyear>p_3.birthyear}))$$

Optimized:
$$\pi_{pid,pid}(\pi_{pid,pid}(P_1 \times P_2) \bowtie \pi_{pid1}(\sigma_{pid1=pid}(K \times P_1)) \bowtie \pi_{pid2}\sigma_{pid2=pid}(K \times P_2))-$$
$$\pi_{pid,pid}(\pi_{pid,pid}(P_1 \times P_2) \bowtie \pi_{pid1}(\sigma_{pid1=pid}(K \times P_1)) \bowtie \pi_{pid2}\sigma_{pid2=pid}(K \times P_2)$$
$$\bowtie\pi_{pid}(\sigma_{pid=pid}(P_1 \times K_2)) \bowtie \pi_{pid}(\sigma_{pid=pid2}(P_3 \times K_2) \wedge p_2.birthyear > p_3.birthyear))$$

4. Q_3:

```
select distinct r1.a
    from  R r1, R r2, R r3
    where  r1.b = r2.a and r2.b = r3.a;
```

Q_4 :

```
select distinct r1.a
from R r1 inner join R r2 on (r1.b = r2.a)
inner join R r3 on (r2.b = r3.a);
```

makerandomR	Q_3 runtime (ms)	Q_4 runtime (ms)
(10, 10, 100)	8.496 ms	0.305 ms
(100, 100, 1000)	41.078 ms	1.965 ms
(500, 500, 1000)	4.802 ms	3.148 ms
(1000, 1000, 5000)	771.005 ms	18.864 ms
(5000, 5000, 100000)	–	–

The optimized query (Q_4) on average runs way faster than Q_3 does on average. As expected, it runs way faster and scales way better. As I got up to the biggest values on the table my SQL file took forever to run so I stopped the file.

5. Q_5 :

```
select ra.a
    from  Ra ra
    where  not exists (select r.b
                        from Rr
                        where r.a = ra.a and r.b not in (select s.b from S s));
```

Q_6 :

```
select q1.rra
from (select ra.a as rra

   from Ra ra

     except

   select q2.rra

   from (select ra.a as rra, r.b

       from Ra ra inner join R a on (r.a = ra.a)

         except

       select ra.a, r.b

       from Ra ra inner join R r on (r.a = r.a)

       inner join S s on (r.b = s.b)) q2) q1
order by 1;
```

makerandomR	makerandomS	Q_5 runtime (ms)	Q_6 runtime (ms)
(10, 10, 100)	(10, 100)	0.095 ms	1.720 ms

Ben Reichert
Assignment 5
B461 – Gucht

(100, 100, 1000)	(100, 1000)	0.288 ms	50.713 ms
(500, 500, 1000)	(500, 1000)	0.564 ms	223.277 ms
(1000, 1000, 5000)	(1000, 5000)		
(5000, 5000, 100000)	(5000, 100000)		

The conclusions I can draw from this experiment is interesting because it was heavily inconsistent. My intuition is that the optimized query should be and is faster, but for some reason the explain analyze is returning that it is terribly slower sometimes. This is especially interesting because when I first tested these functions it was initially running way faster. I kept getting errors for there being multiple instances of the database running, plus I was running into issues having port 5432 already running a db. I think that my conclusion should be that Q_6 is faster, but from the raw data collected we see that it is not in the case of my laptop being likely corrupted.

6. Q_7:

```
select ra.a
  from   Ra ra
  where  not exists (select s.b
                     from Ss
                     where s.b not in (select r.b
                                       from Rr
                                       where r.a = ra.a));
```

Q_8:

```
select q1.rra
from (select ra.a as rra
    from Ra ra
      except
    select q2.rra
    from (select ra.a as rra, s.b
        from Ra ra inner join R r on (r.a = r.a)
        inner join S s on (s.b = s.b)
          except
        select ra.a, s.b
        from Ra ra inner join R r on (r.a = ra.a)
        inner join S s on (s.b = r.b)) q2) q1
order by 1;
```

makerandomR	makerandomS	Q_7 runtime (ms)	Q_8 runtime (ms)
(10, 10, 100)	(10, 100)	0.826 ms	7.663 ms
(100, 100, 1000)	(100, 1000)	10.433 ms	4917.327 ms
(500, 500, 1000)	(500, 5000)	43.340 ms	– –
(1000, 1000, 5000)	(1000, 5000)	– –	– –
(5000, 5000, 100000)	(5000, 100000)	– –	– –

Ben Reichert
Assignment 5
B461 – Gucht
I had a similar result for Q_8 that I had for Q_6. My intuition is telling me that it should be faster than Q_7 but it is not retuning that that is the case. It is especially weird again because it was returning that it was faster earlier, and for some reason is not being so now.