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’;





Optimized:





1. 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

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';





Optimized:







1. 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







Optimized:







1.  :

select distinct r1.a

from R r1, R r2, R r3

where r1.b = r2.a and r2.b = r3.a;

 :

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 | Q3 runtime (ms) | Q4 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 (Q4) on average runs way faster than Q3 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.

1.  :

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));

 :

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 | Q5 runtime (ms) | Q6 runtime (ms) |
| (10, 10, 100) | (10, 100) | 0.095 ms | 1.720 ms |
| (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 Q6 is faster, but from the raw data collected we see that it is not in the case of my laptop being likely corrupted.

1.  :

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));

:



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 | Q7 runtime (ms) | Q8 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) | – – | – – |

I had a similar result for Q8 that I had for Q6. My intuition is telling me that it should be faster than Q7 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.