Assignment 5

Ojaas Hampiholi

19 October, 2020

Question 1

CASE 1: Let us consider a case with EXISTS and UNION operator as below:

SELECT L1(r)
FROM R r
WHERE C1(r)
AND EXISTS (SELECT L2(s)
FROM S s
WHERE C2(s,r)
UNION
SELECT L3(t)
FROM T t
WHERE C3(t,r))

The optimized query in RA for the above query can be written as follows:

$$\pi_{L1(r)}(R \bowtie_{C1(r) \wedge C2(s,r)} S)$$

CASE 2: Let us consider a case with NOT EXISTS and UNION operator as below:

SELECT L1(r)
FROM R r
WHERE C1(r)
AND NOT EXISTS (SELECT L2(s)
FROM S s
WHERE C2(s,r)
UNION
SELECT L3(t)
FROM T t
WHERE C3(t,r))

The optimized query in RA for the above query can be written as follows:

$$\pi_{L1(r)}(\sigma_{C1(r)}(R)) - \pi_{L1(r)}(R \bowtie_{C1(r) \wedge C2(s,r)} S)$$

CASE 3: Let us consider a case with EXISTS and INTERSECT operator as below:

SELECT L1(r)
FROM R r
WHERE C1(r)
AND EXISTS (SELECT L2(s)
FROM S s
WHERE C2(s,r)
INTERSECT
SELECT L3(t)
FROM T t
WHERE C3(t,r))

The optimized query in RA for the above query can be written as follows:

$$\pi_{L1(r)}(\sigma_{C1(r)}(R)) \cap \pi_{L1(r)}(R \bowtie_{C1(r) \wedge C2(s,r)} (S \bowtie_{S.s=T.t} T))$$

CASE 4: Let us consider a case with NOT EXISTS and INTERSECT operator as below:

SELECT L1(r)
FROM R r
WHERE C1(r)
AND NOT EXISTS (SELECT L2(s)
FROM S s
WHERE C2(s,r)
INTERSECT
SELECT L3(t)
FROM T t
WHERE C3(t,r))

The optimized query in RA for the above query can be written as follows:

$$\pi_{L1(r)}(\sigma_{C1(r)}(R)) - (\pi_{L1(r)}(\sigma_{C1(r)}(R)) \cap \pi_{L1(r)}(R \bowtie_{C1(r) \wedge C2(s,r)} (S \bowtie_{S.s=T.t} T)))$$

CASE 5: Let us consider a case with EXISTS and EXCEPT operator as below:

SELECT L1(r)
FROM R r
WHERE C1(r)
AND EXISTS (SELECT L2(s)
FROM S s

WHERE C2(s,r)
EXCEPT
SELECT L3(t)
FROM T t
WHERE C3(t,r))

The optimized query in RA for the above query can be written as follows:

$$\pi_{L1(r)}(\sigma_{C1(r)}(R)) - \pi_{L1(r)}(R \bowtie_{C1(r) \wedge C2(s,r)} (S \bowtie_{S.s=T.t} T))$$

CASE 6: Let us consider a case with NOT EXISTS and EXCEPT operator as below:

SELECT L1(r)
FROM R r
WHERE C1(r)
AND NOT EXISTS (SELECT L2(s)
FROM S s
WHERE C2(s,r)
EXCEPT
SELECT L3(t)
FROM T t
WHERE C3(t,r))

The optimized query in RA for the above query can be written as follows:

$$\pi_{L1(r)}(\sigma_{C1(r)}(R)) - (\pi_{L1(r)}(\sigma_{C1(r)}(R)) - \pi_{L1(r)}(R \bowtie_{C1(r) \wedge C2(s,r)} (S \bowtie_{S.s=T.t} T)))$$

Question 2

We can see that

$$\begin{aligned} RHS &= \pi_{a,b}(R \overline{\ltimes} S) \\ &= \pi_{a,b}(\pi_{a,b}(R) - \pi_{a,b}(R \bowtie S)) \\ &= \pi_{a,b}(\pi_{a,b}(R) - \pi_{a,b}(S)) \\ &= \pi_{a,b}(R) - \pi_{a,b}(S) \\ &= LHS \end{aligned}$$

Question 3

3.A

The given query is

```
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);
```

The following query can be translated to RA Query using temporary view as follows:

```
with googleworker as (select pid, name, salary from worksfor where cname='Google')
select pid, name
from (googleworker g1
join worksfor g2
on (g1.salary < g2.salary)) subquery1
join (select p1.pid, p2.pid
from person p1 join knows on (p1.pid=k.pid1)
join person 2 on (k.pid2 = p2.pid))
on (p1.pid = g1.pid and p2.pid = g2.pid);
```

$$\texttt{googleWorker} = \pi_{pid,cname,salary}(\sigma_{cname='Google'}(worksfor))$$

$$\texttt{E} = \pi_{*}(p \bowtie_{k.pid1=p.pid} knows \bowtie_{k.pid2=p1.pid} p1)$$

$$\texttt{F} = \pi_{*}(googleWorker_1 \bowtie_{googleWorker_1.salary<worksfor.salary} worksfor)$$

$$\pi_{p.pid,p.name}(E \bowtie_{p.pid=googleWorker_1.pid \wedge p1.pid=googleWorker_2.pid} F)$$

3.B

We can optimize the above expression as follows:

```
select distinct pid, name
from (select distinct w1.pid, w2.pid
from (select pid, salary from worksfor where cname = 'Google') w1
join (select pid, salary from worksfor) w2 on w1.salary < w2.salary) E
join (select pid, name, pid2
```

from (select pid, name from person) p1
join knows k on p1.pid = k.pid1) F
on (pid = w1.pid and pid2 = w2.pid);

$$\texttt{E} = \pi_{w1.pid,w2.pid}(\sigma_{cname='Google'}(w1) \bowtie_{w1.salary<w2.salary} w2)$$

$$\texttt{F} = \pi_{name,pid1,pid2}(p \bowtie_{p..pid=k.pid1} k)$$

$$\pi_{pid1,name}(E \bowtie_{pid1=w1.pid \wedge pid2=w2.pid} F)$$

Question 4

4.A

The given Query is as follows:

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));

We can write the modified RA Query as follows:

with progNetworks as (select pid, skill
from personSkill
where skill = 'Programming' or skill = 'Networks'),
personIndy as (select pid, name, city, birthyear
from person
where city = 'Indianapolis'),
amazonWorker as (select pid, cname, salary
from worksFor
where cname = 'Amazon')

select pid
from ((select pid, name, city, birthyear, skill
from person natural join progNetworks
except

```
select p.pid, name, city, birthyear, skill
from person p
natural join progNetworks
natural join amazonWorker)
intersect
(select pid, name, city, birthyear, skill
from person natural join progNetworks
except
select p.pid, p.name, p.city, p.birthyear, skill
from person p
natural join progNetworks
join(personIndy p1 join knows k on p1.pid = k.pid2)
on k.pid1 = p.pid)) subquery;
```

$\texttt{progNetworks} = \pi_{pid,skill}(\sigma_{skill='Programming' \vee skill='Networks'}(personSkill)$

$\texttt{personIndy} = \pi_{*}(\sigma_{city='Indianapolis'}(p)$

$\texttt{amazonWorker} = \pi_{pid,cname,salary}(\sigma_{cname='Amazon'}(w)$

$\texttt{E} = \pi_{*}(p \bowtie progNetworks)$

$\texttt{F} = \pi_{*}(E \bowtie amazonWorker)$

$\texttt{G} = \pi_{*}(E \bowtie_{p.pid=k.pid1} (k \bowtie_{k.pid2=personIndy.pid} personIndy))$

$\texttt{Therefore} \quad \pi_{pid1}((E - F) \cap (E - G)$

4.B

```
with progNetworks as (select pid, skill
from personSkill
where skill = 'Programming' or skill = 'Networks'),
personIndy as
(select pid from person
where city = 'Indianapolis'),
amazonWorker as (select pid from worksFor
where cname = 'Amazon')

select pid
from (select pid, skill
from progNetworks
except
(select pid, skill
```

from progNetworks natural join amazonWorker
union
select distinct p.pid, skill
from progNetworks p
join (select distinct pid1
from personIndy join knows k on pid = k.pid2)
k on pid1 = p.pid)) subquery;

$\texttt{progNetworks} = \pi_{pid,skill}(\sigma_{skill='Programming' \vee skill='Networks'}(personSkill)$

$\texttt{personIndy} = \pi_{*}(\sigma_{city='Indianapolis'}(p)$

$\texttt{amazonWorker} = \pi_{pid,cname,salary}(\sigma_{cname='Amazon'}(w)$

$\texttt{E} = \pi_{*}(p \bowtie progNetworks)$

$\texttt{F} = \pi_{*}(E \bowtie amazonWorker)$

$\texttt{G} = \pi_{*}(E \bowtie_{p.pid=k.pid1} (k \bowtie_{k.pid2=personIndy.pid} personIndy))$

$\texttt{Therefore} \quad \pi_{pid1}(E - (F \cup G))$

Question 5

5.A

The given query is 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));

We can write the RA SQl for the same using temporary views as follows

select p1pid, p2pid
from (select *
from person p1
join knows k on p1.pid = k.pid1
join person p2 on p2.pid = k.pid2
except
select p1.*, k.*, p2.*
from person p1

join knows k on p1.pid = k.pid1
join person p2 on p2.pid = k.pid2
join (person p3 join knows k3 on (p3.pid = k3.pid2)) on
(p2.birthyear > p3.birthyear and k3.pid1 = p1.pid)) subquery;

$\texttt{E} = \pi_{p1.*,k.*,p2.*}(p1 \bowtie_{p1.pid=k.pid1} k \bowtie_{k.pid2=p2.pid} p2)$

$\texttt{F} = \pi_{p1.*,k.*,p2.*}(E \bowtie_{p1.pid=k2.pid \wedge p2.birtyear>p3.birthyear} (p3 \bowtie_{p3.pid=k2.pid2} k2))$

$\texttt{Therefore} \quad \pi_{p1.pid1,p2.pid}(E - F)$

5.B

with P as (select pid, birthyear from person),
p1Knowsp2 as (select p1.pid as pid1, p2.pid as pid2, p2.birthyear
from person p1
join knows k on p1.pid = k.pid1
join person p2 on p2.pid = k.pid2)
select pid1, pid2
from (select p1Knowsp2.*
from p1Knowsp2
except
select p1Knowsp2.*
from p1Knowsp2
join (select distinct birthyear, pid1
from person p3 join knows k3 on p3.pid = k3.pid2) pk3
on (p1Knowsp2.birthyear > pk3.birthyear and pk3.pid1 = p1Knowsp2.pid1))
subquery;

$\texttt{P} = \pi_{p.pid,p.birtyear}(person)$

$\texttt{p1Knowsp2} = \pi_{p1.pid,p2.pid,p2.birtyear}(p1 \bowtie_{p1.pid=k.pid1} k \bowtie_{k.pid2=p2.pid} p2)$

$\texttt{E} = \pi_{k.pid1,p.birthyear}(p \bowtie_{p.pid=k.pid2} (k))$

$\texttt{F} = \pi_{*}(p1Knowsp2 \bowtie_{p1Knowsp2.birthyear>E.birthyear \wedge p1Knowsp2.pid1=E.pid1} E)$

$\texttt{Therefore} \quad \pi_{pid1,pid2}(p1Knowsp2 - \pi_{pid1,pid2}(F))$

Question 6

6.A

We can see that the original query is

```
select p.pid
from person p
where exists (select 1
from person p1
where p1.pid in (select ps.pid from personSkill ps
where ps.skill = 'Programming'
intersect
select ps.pid from personSkill ps
where ps.skill = 'Databases') and
(p.pid, p1.pid) in (select k.pid1, k.pid2 from knows k));
```

We can wrote the optimized query as follows using temporary views:

```
with programming as (select * from personSkill where skill = 'Programming'),
databases as (select * from personSkill where skill = 'Databases')
select distinct pid
from (select p.*, p1.pid as pid1, p1.name, p1.city, p1.birthyear
from person p cross join
person p1 join programming ps on (p1.pid = ps.pid)
intersect
select p.*, p1.*
from person p cross join
person p1 join databases ps on (p1.pid = ps.pid)
intersect
select p.*, p1.*
from person p join knows k on (p.pid = k.pid1)
join person p1 on (p1.pid = k.pid2)) subquery;
```

$\texttt{programming} = \pi_*(\sigma_{skill='Programming'}(personSkill))$

$\texttt{databases} = \pi_*(\sigma_{skill='Databases'}(personSkill))$

$\texttt{p1Knowsp2} = \pi_{p1.pid,p2.pid,p2.birtyear}(p1 \bowtie_{p1.pid=k.pid1} k \bowtie_{k.pid2=p2.pid} p2)$

$\texttt{E} = \pi_*(p \times (p1 \bowtie programming))$

$\texttt{F} = \pi_*(p \times (p1 \bowtie databases))$

$\texttt{G} = \pi_*(p1 \bowtie_{p1.pid=k.pid1} k \bowtie_{k.pid2=p2.pid} p2)$

$\texttt{Therefore} \quad \pi_{p.pid}(E \cap F \cap G)$

6.B

```
select distinct pid1
from ( (select pid
from personSkill where skill = 'Programming'
intersect
select pid
from personSkill where skill = 'Databases') p
join knows on pid = pid2 ) subquery;
```

$\texttt{E} = \pi_{pid}(\sigma_{skill='Programming'}(personSkill)) \cap \pi_{pid}(\sigma_{skill='Databases'}(personSkill))$

$\texttt{Therefore} \quad \pi_{k.pid1}(E \bowtie_{E.pid=k.pid2} k)$

Question 7
7.A

Let the query Q3 be :
select distinct r1.a

from R r1, R r2, R r3

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

We can see that the translated RA SQL query Q4 can be written as follows:

```
select distinct a

from R natural

join (select distinct a as b

    from R natural join (select distinct a as b

                                from R) q1 ) q2

order by 1;
```

7.B

makerandomR	Q3 (ms)	Q4 (ms)
(100,100,1000)	62.131	1.477
(150,150,4000)	2185.853	6.957
(200,200,10000)	10523.178	16.669
(250,250,20000)	23968.634	29.887

Question 8

8.A

Let the given query Q5 be:

```
select ra.a

from Ra ra

where not exists (select r.b

from R r

where r.a = ra.a and

r.b not in (select s.b from S s));
```

We can write the translated query Q6 as follows:

```
(select distinct a
from Ra)
except
(select a
from (select a, b
        from R
        except
        select a, b
        from R natural join S) subquery );
```

8.B

makerandomR	makerandomS	Q5 (ms)	Q6 (ms)
(100,100,1000)	(100, 1500)	0.664	1.774
(150,150,4000)	(150, 2500)	3.013	6.563
(250,250,10000)	(250, 5000)	6.792	17.812
(500,500,20000)	(500, 10000)	6.517	42.357

Question 9

9.A

Let the query Q7 be:

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

We can define the translated query Q8 as follows:

```
select distinct a
```

```
from Ra
except
select a
from (select a, b
        from Ra cross join S
        except
        select a, b
        from R) subquery;
```

9.B

makerandomR	makerandomS	Q7 (ms)	Q8 (ms)
(100,100,1000)	(100, 1500)	30.742	11.494
(150,150,4000)	(150, 2500)	148.010	28.864
(250,250,10000)	(250, 5000)	757.113	102.284
(500,500,20000)	(500, 10000)	2129.636	454.571