Assignment 3

Nithin N Vusirikala (nnvusiri, 2000916423)

October 4th 2021

Question 1:

Given:

```
select e1.*
from   E1 e1
where  not exists (select distinct row() from F)
union
select e2.*
from   E2 e2
where  exists (select distinct row() from F);
```

We need to return $E1$ when F is empty and return $E2$ when F is not empty. RA Expression for this above statement would be below:

$$(\pi_{E1.*}(E_1 - \pi_{E_1.*}(E_1 X \pi_{()} F))) \vee (\pi_{E2.*}(E_2 X \pi_{()} F))$$

Question 2:

We need to return `True` when R has less than 2 elements and return `False` when it has more elements

$$\text{Let } F = \rho_{row}(\pi_{()}(\sigma_{R1.x \neq R2.x}(R1XR2)))$$

Then, below RA expression gives the required output of `True` or `False`

$$\pi_{\mathbf{True}}(\sigma_{row \neq '()'} F) \vee \pi_{\mathbf{False}}(\sigma_{row = '()'} F)$$

Question 3:

Let's take the special case first.

```
select L1(r)
from R r
where C1(r) [not] exists (select L2(s)
                          from S s
                          where C2(s,r)
                          [union | intersect | except]
                          select L3(t)
                          from T t
                          where C3(t,r))
```

Let's begin with `EXISTS` case. Given the below:

```
select L1(r)
from R r
where C1(r) exists (select L2(s)
                    from S s
                    where C2(s,r)
                    [union | intersect | except]
                    select L3(t)
                    from T t
                    where C3(t,r))
```

This can be translated to:

```
select L1(r) from (
                    select r.*
                    from R r, S s
                    where C1(r) and C2(s,r)
                    [union | intersect | except]
                    select r.* from R r, T t
                    where C1(r) and C3(t,r)) q
```

Which further can be translated to below using joins:

```
select L1(r) from (
                    select r.*
                    from R r cross join S s
                    where C1(r) and C2(s,r)
                    [union | intersect | except]
                    select r.* from R r cross join T t
                    where C1(r) and C3(t,r)) q
```

We can express the above SQL query in RA expression as below:

$$\pi_{L1(r)}(\pi_{r.*}(\sigma_{C1(r) \wedge C2(s,r)}(R \times S)) \;\; [\cup | \cap | -] \;\; \pi_{r.*}(\sigma_{C1(r) \wedge C3(t,r)}(R \times T)))$$

Given the General case:

```
select L1(r1,...,rn)
from R1 r1, ..., Rn rn
where C1(r1,...,rn) and exists (select L2(s1,...,sm)
                                from S1 s1,..., S1 sm
                                where C2(s1,...,sm,r1,...,rn)
                                [union | intersect | except]
                                select L3(t1,...,tk)
                                from T1 t1, ..., Tk tk
                                where C3(t1,...,tk,r1,...,rn))
```

In the similar way to above, the general case translates to the below:

```
WITH R1 as (select r1.* FROM R1 r1)
R2 AS (select r2.* FROM R2 r2)
...
select L1(r1,...,rn) from (
                            select r1.*,r2.*,....rn.*
                            from R1 r1 cross join...Rn rn cross join
                            S1 s1 cross join S2 s2 ... cross join Sm sm
                            where C1(r1,...rn) and C2(s1,...,sm,r1,...,rn)
                            [union | intersect | except]
                            select r1.*,r2.*,....rn.*
                            from R1 r1 cross join...Rn rn cross join
                            T1 t1 cross join T2 t2 ... cross join Tk tk
                            where C1(r1,...rn) and C3(t1,...,tk,r1,...,rn)) q
```

We can express the above SQL query in RA as below:

$$\pi_{L1(\mathbf{r})}(\pi_{\mathbf{r}.*}(\sigma_{C1(\mathbf{r})\wedge C2(\mathbf{s},\mathbf{r})}(\mathbf{R}\times\mathbf{S}))\ \ [\cup|\cap|-]\ \ \pi_{\mathbf{r}.*}(\sigma_{C1(\mathbf{r})\wedge C3(\mathbf{t},\mathbf{r})}(\mathbf{R}\times\mathbf{T}))))$$

where:
$\mathbf{r} = r_1, r_2, ..., r_n$
$\mathbf{s} = s_1, s_2, ...s_m$
$\mathbf{t} = t_1, t_2, ..., t_k$
$\mathbf{r.*} = r_1*, r_2*, ...r_n.*$
$\mathbf{R} = R_1 \times R_2 \times ... \times R_n$
$\mathbf{S} = S_1 \times S_2 \times ... \times S_m$
$\mathbf{T} = T_1 \times T_2 \times ... \times T_k$

Now, let's take the `NOT EXISTS` case.
For the special case, given as:

```
select L1(r)
from R r
where C1(r) not exists (select L2(s)
                        from S s
                        where C2(s,r)
                        [union | intersect | except]
                        select L3(t)
                        from T t
                        where C3(t,r))
```

This can be translated to :

```
select L1(r) FROM
        (select r.* from R r except (
                                    select r.*
                                    from R r, S s
                                    where C1(r) and C2(s,r)
                                    [union | intersect | except]
                                    select r.* from R r, T
                                    where C1(r) and C3(t,r)))
```

Enabling `JOIN` the above query translates to:

```
select L1(r) FROM
        (select r.* from R r except (
                                    select r.*
                                    from R r cross join S s
                                    where C1(r) and C2(s,r)
                                    [union | intersect | except]
                                    select r.* from R r cross join T t
                                    where C1(r) and C3(t,r)))
```

The RA expression for the above looks like:

$$\pi_{L1(r)}(\pi_{r.*}(R)-(\pi_{r.*}(\sigma_{C1(r)\wedge C2(s,r)}(R\times S))\ \ [\cup|\cap|-]\ \ \pi_{r.*}(\sigma_{C1(r)\wedge C3(t,r)}(R\times T))))$$

Now, the general case for `NOT EXISTS`. Given:

```
select L1(r1,...,rn)
from R1 r1, ..., Rn rn
where C1(r1,...,rn) and
not exists (select L2(s1,...,sm)
            from S1 s1,..., S1 sm
            where C2(s1,...,sm,r1,...,rn)
            [union | intersect | except]
            select L3(t1,...,tk)
            from T1 t1, ..., Tk tk
            where C3(t1,...,tk,r1,...,rn))
```

This can be translated to:

```
WITH R1 as (select r1.* FROM R1 r1)
R2 AS (select r2.* FROM R2 r2)
.
.
select L1(r1,...,rn) from (
select r1.*,r2.*,....rn.*
from R1 r1 cross join R2 r2 cross join ... Rn rn
except(
        select r1.*,r2.*,....rn.*
        from R1 r1 cross join R2 r2 cross join ... Rn rn
        cross join S1 s1 cross join S2 s2 ....cross join Sm sm
        where C1(r1,...rn) and C2(s1,...,sm,r1,...,rn)
        [union | intersect | except]
        select r1.*,r2.*,....rn.*
        from R1 r1 cross join R2 r2 cross join ... Rn rn
        cross join  T1 t1 cross join T2 t2 ....cross join Tk tk
        where C1(r1,...rn) and C3(t1,...,tk,r1,...,rn))) qq
```

We can express the above SQL query in RA as below:

$$\pi_{L1(\mathbf{r})}(\pi_{\mathbf{r}.*}\mathbf{R}-(\pi_{\mathbf{r}.*}(\sigma_{C1(\mathbf{r})\wedge C2(\mathbf{s},\mathbf{r})}(\mathbf{R}\times\mathbf{S}))\ \ [\cup|\cap|-]\ \ \pi_{\mathbf{r}.*}(\sigma_{C1(\mathbf{r})\wedge C3(\mathbf{t},\mathbf{r})}(\mathbf{R}\times\mathbf{T}))))$$

where:

$\mathbf{r} = r_1, r_2, ..., r_n$
$\mathbf{s} = s_1, s_2, ...s_m$
$\mathbf{t} = t_1, t_2, ..., t_k$
$\mathbf{r.*} = r_1*, r_2*, ...r_n.*$
$\mathbf{R} = R_1 \times R_2 \times ... \times R_n$
$\mathbf{S} = S_1 \times S_2 \times ... \times S_m$
$\mathbf{T} = T_1 \times T_2 \times ... \times T_k$

Question 4:
Given to prove:

$$\pi_{a,d}(R \bowtie_{c=d} S) = \pi_{a,d}(\pi_{a,c}(R) \bowtie_{c=d} \pi_d(S)).$$

Let's put the predicate logic of LHS and try to prove the equality directly :

$\pi_{a,d}(R \bowtie_{c=d} S)$
$= \pi_{a,d}(\sigma_{c=d}(R \times S))$
$= \{(a,d)|\exists b, \exists c : R(a,b,c) \wedge S(d,e) \wedge c = d\}$
$= \{(a,d)|\exists b \exists c R(a,b,c) \wedge \exists e S(d,e) \wedge c = d\}$
$= \{(a,d)|(a,c) \in \pi_{(a,c)}(R) \wedge d \in \pi_d(S) \wedge c = d\}$
$= \pi_{(a,d)}(\sigma_{c=d}(\pi_{(a,c)}(R) \times \pi_d(S)))$
$= \pi_{(a,d)}(\pi_{(a,c)}(R) \bowtie_{c=d} \pi_d(S))$

=RHS

Question 5:
Given, $R(a,b,c), S(\underline{d},e)$. Now, if $c \in R$ is the foreign key of $d \in S$, then to maintain referential integrity, all the $c \in R$ should be present in $d \in S$.

Taking this information, we can observe that $\pi_{a,d}(R \bowtie_{c=d} S)$ returns all the tuples (a,d) where $a \in R \wedge d \in S \wedge c = d$.

So, the rewrite rule can simply be $\pi_{(\mathbf{a},\mathbf{c})}R$. Say,

$(x,y) \in \pi_{a,d}(R \bowtie_{c=d} S)$
$\Rightarrow y \in d \, \forall (x \in a)$
$\Rightarrow y \in c \, \forall (x \in a)$ since $c = d$ and $(a,c) \in A_R$
$\Rightarrow (x,y) \in \pi_{(a,c)}R$

So given $R(a,b,c), S(\underline{d},e)$ then $\pi_{a,d}(R \bowtie_{c=d} S) = \underline{\pi_{(\mathbf{a},\mathbf{c})}\mathbf{R}}$

Question 6:

The RA SQL expression is:

```
select distinct q.cname,q.headquarter from (select c.cname, c.headquarter
from company c natural join (select * from worksfor  where salary<55000) w
natural join (select * from person where city <>'Bloomington') p ) q;
```

The RA expression of the final output after translation is:

$$\pi_{C.cname,C.headquarter}(C \bowtie \sigma_{w.salary<55000}W \bowtie \sigma_{p.city<>\mathbf{Bloomington}}P)$$

Projection over joins:

$$\pi_{C.cname,C.headquarter}(C \bowtie \pi_{W.cname,W.pid}(\sigma_{W.salary<55000}W) \bowtie \pi_{P.pid}(\sigma_{p.city<>\mathbf{Bloomington}}(P)))$$

Converting to semi join:

$\pi_{C.cname,C.headquarter}(C \ltimes \pi_{W.cname,W.pid}(\sigma_{W.salary<55000}W) \ltimes \pi_{P.pid}(\sigma_{p.city<>\textbf{Bloomington}}(P)))$

Question 7:
The RA SQL expression post the translation is:

```
with F as (select p.pid,s1.skill as s1skill,s2.skill as s2skill from person p
cross join skill s1
cross join skill s2
where s1.skill<>s2.skill)

select distinct q.pid from (
(
select pid from
(select  p2.pid,s2.skill
from person p2 cross join skill s2
except
select  p.pid,s.skill
from person p
cross join skill s
join personSkill ps on p.pid = ps.pid and s.skill=ps.skill
) pp
)
INTERSECT
(
select p2.pid
from person p2
except
(
select  pid from (
(select  * from F
except
select  F.* from F join personSkill ps on F.pid=ps.pid and F.s1skill=ps.skill)
intersect
(select  * from F
except
select  F.* from F join personSkill ps on F.pid=ps.pid and F.s2skill=ps.skill)
) qq
)
)
)q;
```

Let

$$P_{opt} = \pi_{pid}P$$

$$F = \pi_{pid,s1.skill,s2.skill}(\sigma_{s1.skill \neq s2.skill}(P \times S1 \times S2)$$

$$F_{opt} = \rho_{pid,skill1,skill2}(\pi_{pid,s1.skill,s2.skill}(\sigma_{s1.skill \neq s2.skill}(P_{opt} \times S1 \times S2))$$

The RA expression of the main output can be written as:

$$\pi_{pid}(A \wedge B)$$

Where

$$A = \pi_{pid}(\pi_{pid,skill}(P \times S) - \pi_{pid,ps.skill}(P \times S \bowtie_{P.pid=pS.pid \wedge s.skill=pS.skill} pS)$$

$$B = P_{opt} - [(\pi_{pid}(F_{opt} - (F_{opt} \bowtie_{pid=pS.pid \wedge skill1=pS.skill} pS))$$
$$\wedge(\pi_{pid}(F_{opt} - (F_{opt} \bowtie_{pid=pS.pid \wedge skill2=pS.skill} pS)] = B_{opt}$$

Let's take A and try to optimize that.

$$A = \pi_{pid}(\pi_{pid,skill}(P \times S) - \pi_{pid,ps.skill}(P \times S \bowtie_{P.pid=pS.pid \wedge s.skill=pS.skill} pS)$$

Since $pS \subseteq P \times S$ then $(P \times S \bowtie_{P.pid=pS.pid \wedge s.skill=pS.skill} pS)$ can be replaced with pS itself as join will result in the pids and skills that are there in pS itself. So, A can be rewritten as,

$$A_{opt} = \pi_{pid}(\pi_{pid,skill}(P_{opt} \times S) - pS)$$

So, finally the result would be

$$\pi_{pid}(A_{opt} \wedge B_{opt})$$

Question 8:

The RA SQL expression post the translation is:

```
select q.pid, q.pname from(
(select p.pid,p.pname,p.city
from person p
natural join worksFor w
join (select * from companyLocation where city ='Bloomington') c ON
c.cname=w.cname)

intersect

(select p.pid,p.pname,p.city from
person p

EXCEPT

select p.pid,p.pname,p.city from person p
join knows k on p.pid=k.pid1
join (select * from person where city='Chicago') p2 on
k.pid2 = p2.pid
)

) q;
```

The RA expression of the same would be:

$$\pi_{pid,pname}(E \wedge (P - F))$$

where

$$E = \pi_{p.*}(P \bowtie W \bowtie_{w,cname=cL.cname} (\sigma_{city=\textbf{Bloomington}}(cL))$$

$$F = \pi_{p1.*}(P1 \bowtie_{P1.pid=K.pid1} K \bowtie_{K.pid2=P2.pid} (\sigma_{city=\textbf{Chicago}}(P2)))$$

$$\pi_{pid,pname}(E \wedge (P - F))$$

$$= \pi_{pid,pname}((E \wedge P) - (E \wedge F))$$

Now, $E \wedge P = E$ since $E \subseteq P$, Thus, the above expression changes to,

$$= \pi_{pid,pname}(E - (E \wedge F))$$
$$= \pi_{pid,pname}((E - E) \vee (E - F))$$
$$= \pi_{pid,pname}((E - F))$$

Now Let's take E,

$$E = \pi_{p.*}(P \bowtie W \bowtie_{w,cname=cL.cname} (\sigma_{city=\textbf{Bloomington}}(cL))$$

Using projections over joins and semijoins properties, the above translates to,

$$E = (P \ltimes (\pi_{cname,pid}(W) \ltimes \pi_{cname}(\sigma_{city=\textbf{Bloomington}}(cL)))) = E_{opt}$$

Similarly, let's take F

$$F = \pi_{p1.*}(P1 \bowtie_{P1.pid=K.pid1} K \bowtie_{K.pid2=P2.pid} (\sigma_{city=\textbf{Chicago}}(P2)))$$

With projections over joins,

$$F = \pi_{p1.*}(P1 \bowtie_{P1.pid=K.pid1} K \bowtie_{K.pid2=P2.pid} \pi_{P2.pid}(\sigma_{city=\textbf{Chicago}}(P2))) = F_{opt}$$

So finally, the given expression updates to:

$$\pi_{pid,pname}(E_{opt} - F_{opt})$$

with

$$E_{opt} = (P \ltimes (\pi_{cname,pid}(W) \ltimes \pi_{cname}(\sigma_{city=\textbf{Bloomington}}(cL))))$$

$$F_{opt} = \pi_{p1.*}(P1 \bowtie_{P1.pid=K.pid1} K \bowtie_{K.pid2=P2.pid} \pi_{P2.pid}(\sigma_{city=\textbf{Chicago}}(P2)))$$

Question 9:

The RA SQL output of the translated output is below:

```
select q.cname,q.headquarter from (
   (select c.cname,c.headquarter
from company c natural join worksfor w
    )
intersect
(select c.cname,c.headquarter from company c
 except
 select pp.cname,pp.headquarter from
(select w.cname as wname,w.salary,w.pid,c.cname,c.headquarter from company c
natural join (select * from worksfor w where w.salary<=70000) w
except
 (
select w.cname as wname,w.salary,w.pid,c.cname,c.headquarter from company c
natural join worksfor w natural join (select * from personSkill ps where skill='Programming') ps
intersect
select w.cname as wname,w.salary,w.pid,c.cname,c.headquarter from company c
natural join worksfor w natural join (select * from personSkill ps where skill='AI') ps
) )pp )) q;
```

The RA expression of the above can be written as below:

$$\pi_{cname,headquarter}(E \wedge (C - \pi_{cname,headquarter}(F - (G \wedge H))$$

where

$$E = \pi_{c.*}(C \bowtie W) = C \ltimes W = E_{opt}$$

$$F = \pi_{c.*,w.*}(C \bowtie \sigma_{w.salary<=70000}(W)) = F_{opt}$$

$$G = \pi_{c.*,w.*}(C \bowtie (W \bowtie \sigma_{skill=\textbf{Programming}}(pS)))$$
$$= \pi_{c.*,w.*}(C \bowtie (W \ltimes \sigma_{skill=\textbf{Programming}}(pS))) = G_{opt}$$

$$H = \pi_{c.*,w.*}(C \bowtie (W \bowtie \sigma_{skill=\textbf{AI}}(pS)))$$
$$= \pi_{c.*,w.*}(C \bowtie (W \ltimes \sigma_{skill=\textbf{AI}}(pS))) = H_{opt}$$

So, finally with optimized expressions:

$$\pi_{cname,headquarter}(E_{opt} \wedge (C - \pi_{cname,headquarter}(F_{opt} - (G_{opt} \wedge H_{opt}))$$

Question 10:

```
The RA SQL output of the translated output is below:

(select pid , true from
(select * from person p
join hasmanager hm1 on p.pid=hm1.mid
join hasmanager hm2 on p.pid = hm2.mid and hm1.eid<>hm2.eid) as pp)
union
(select pid, false from
(select p.* from person p
except
(select p.* from person p
 join hasmanager hm1 on hm1.mid=p.pid
 join hasmanager hm2 on hm2.mid=p.pid and hm1.eid<>hm2.eid)) qq) order by 1;
```

The RA Expression of the same will look like:

$$\pi_{pid,\mathbf{True}}(A) \vee \pi_{pid,\mathbf{False}}(P - A)$$

where

$$A = \pi_{P.*}(P \bowtie_{P.pid=hm1.mid} hm1 \bowtie_{p.pid=hm2.mid \wedge hm1.eid \neq hm2.eid} hm2)$$

with projection over joins,

$= \pi_{P.pid}(\pi_{pid} P \bowtie_{P.pid=hm1.mid} hm1 \bowtie_{P.pid=hm2.mid \wedge hm1.eid \neq hm2.eid} hm2) = A_{opt}$

Similarly, if we do for our full expression, it becomes:

$$\pi_{pid,\mathbf{True}}(A_{opt}) \vee \pi_{pid,\mathbf{False}}(\pi_{pid} P - A_{opt})$$