

Assignment 7

5.

- a.

```
SELECT p1.pid, p1.pname, p2.pid
FROM Person p1 JOIN Person p2 ON (p1.pid <> p2.pid)
WHERE p1.city = 'Bloomington'
EXCEPT
SELECT p1.pid, p1.pname, p2.pid
FROM Person p1 JOIN Person P2 ON (p1.pname = 'Eric') JOIN worksFor w ON
(w.salary > 100000 AND NOT p2.pid = w.pid);
```
- b.

```
SELECT m.mid
FROM Manages m JOIN worksFor w1 ON (m.mid=w1.pid) JOIN worksfor w2
ON (m.eid = w2.pid)
WHERE w1.salary < w2.salary AND w1.cname = 'Google';
```

6.

- a. RA optimization image under these translations
 - i. **Push down Bloomington condition onto P1 + attribute elimination on P1 since we no longer need city attribute**

```
WITH btown AS (SELECT pid, pname FROM Person WHERE city =
'Bloomington')
SELECT b.pid, b.pname, p2.pid
FROM btown b JOIN person p2 on (b.pid <> p2.pid)
EXCEPT
SELECT p1.pid, p1.pname, p2.pid
FROM Person p1 JOIN Person p2 ON (p1.pname = 'Eric') JOIN
worksFor w ON (w.salary > 100000 AND NOT p2.pid = w.pid);
```

ii. **Pushing condition of salary onto worksFor + attribute elimination of cname**

```
WITH btown AS (SELECT pid, pname FROM Person where city =
'Bloomington'),
      w AS (SELECT pid FROM worksFor WHERE salary > 100000)
SELECT b.pid, b.pname, p2.pid
FROM btown b JOIN person p2 on (b.pid <> p2.pid)
EXCEPT
SELECT p1.pid, p1.pname, p2.pid
FROM Person p1 JOIN Person p2 ON (p1.pname = 'Eric') JOIN W w ON
(NOT p2.pid = w.pid);
```

iii. **Distribution rewrite rule for selection, pulling pname = Eric into btown view**

```
WITH btown AS (SELECT pid, pname FROM Person where city =
'Bloomington' AND pname <> 'Eric'),
      w AS (SELECT pid FROM worksFor WHERE salary > 100000)
SELECT b.pid, b.pname, p2.pid
FROM btown b JOIN person p2 on (b.pid <> p2.pid)
EXCEPT
SELECT b.pid, b.pname, p2.pid
FROM btown b NATURAL JOIN Person p2 JOIN W w ON (p2.pid <>
w.pid);
```

iv. **Rewrite rule for difference set operation since btown view is used in both clauses**

```
WITH btown AS (SELECT pid, pname FROM Person where city =
'Bloomington' AND pname <> 'Eric'),
      w AS (SELECT pid FROM worksFor WHERE salary > 100000)
SELECT b.pid, b.pname, q.pid
FROM btown b JOIN (SELECT pid FROM person
EXCEPT
SELECT pid FROM W)q ON q.pid <> b.pid;
```

$$\pi_{P1.pid} \left(\sigma_{\substack{P1.city = Boston \\ P1.pname = Eric}} (P1 \bowtie P2) \right) - \pi_{\substack{P1.pid \\ P1.pname \\ P2.pid}} \left(P1 \bowtie P2 \bowtie W \right)$$

$\wedge \substack{W.salary > 100000 \\ \neg (P2.pid = W.pid)}$

pushing selection + attribute elimination on P1

$$\equiv \pi_{\substack{P1.pid, \\ P2.pname, \\ P2.pid}} \left(\left(\pi_{\substack{P1.city = Boston \\ P1.pname = Eric}} (\pi_{P1.pid} \sigma (P1)) \right) \bowtie P2 \right) - \pi_{\substack{P1.pid \\ P1.pname \\ P2.pid}} \left(P1 \bowtie P2 \bowtie W \right)$$

$\wedge \substack{W.salary > 100000 \\ \neg (P2.pid = W.pid)}$

pushing selection + attribute elimination on W

$$\equiv \pi_{\substack{P1.pname \\ P1.pid \\ P2.pid}} \left(\left(\pi_{\substack{P1.city = Boston \\ P1.pname = Eric}} (\pi_{P1.pid} \sigma (P1)) \right) \bowtie P2 \right) - \pi_{\substack{P1.pid \\ P1.pname \\ P2.pid}} \left(P1 \bowtie P2 \bowtie (\pi_{P2.pid} \sigma (W)) \right)$$

$\wedge \substack{W.salary > 100000 \\ P2.pid \neq W.pid}$

distrib. rule on Eric P1 ($B = \pi_{P2.pid} \sigma (P1)$); $W = \pi_{P2.pid} \sigma (W)$)

$\wedge \substack{P1.city = Boston \\ P1.pname = Eric}$

$$= \pi_{\substack{B.pname \\ B.pid \\ P2.pid}} (B \bowtie P2) - \pi_{\substack{B.pid \\ B.pname \\ P2.pid}} (B \bowtie P2 \bowtie W)$$

rewrite rule on \neg set operation (since B is projected in both clauses)

$$\pi_{\substack{B.pname \\ B.pid \\ P2.pid}} \left(B \bowtie \left(\pi_{P2.pid} (P) - \pi_{P2.pid} (W) \right) \right)$$

b.

i. Pushing down selection of google over join + attribute elimination of cname from w1

SELECT m.mid

FROM Manages m JOIN (SELECT pid, salary FROM worksFor WHERE
cname = 'Google')w1 ON m.mid = w1.pid

JOIN worksFor w2 ON (m.eid = w2.pid)

WHERE w1.salary < w2.salary;

ii. Attribute elimination of cname from w2

```

SELECT m.mid
FROM Manages m JOIN (SELECT pid, salary FROM worksFor WHERE
cname = 'Google')w1 ON m.mid = w1.pid
JOIN (SELECT pid, salary FROM worksFor)w2 ON (m.eid =
w2.pid)
WHERE w1.salary < w2.salary;

```

iii. **Semi-join m and w1**

```

SELECT m.mid
FROM manages m natural join (select pid as mid, salary FROM worksFor
WHERE cname = 'Google')w1 JOIN (SELECT pid, salary FROM
worksFor)w2 ON w2.pid = m.eid
WHERE w1.salary < w2.salary;

```

$$\begin{aligned}
 & \pi_{m.mid} (\sigma_{(m \bowtie_{w1.pid=pid, w2.eid=p.id} w1 \bowtie_{w1.salary < w2.salary} w2)}) \\
 & \quad w1.cname = Google \\
 & \equiv \pi_{m.mid} (\sigma_{(m \bowtie_{w1.salary, mid=w1.cname} (\pi_{pid, salary} w1) \bowtie_{eid=pid} w2)}) \leftarrow \text{pushing down selection over join + attribute elim of cname (w1)} \\
 & \quad \leftarrow w2.sal, pid = Google \\
 & \equiv \pi_{m.mid} (\sigma_{w1.salary, w2.salary} (m \bowtie_{mid=cname} (\pi_{pid, salary} w1) \bowtie_{eid=pid} \pi_{pid, salary} w2)) \leftarrow \text{attribute elimination (eliminating cname from w2)} \\
 & \quad \leftarrow \text{semi-join because} \\
 & \equiv \pi_{m.mid} (\sigma_{w1.salary, w2.salary} (m \bowtie_{Google} (\sigma_{mid=pid} w1) \bowtie_{mid=pid} w2))
 \end{aligned}$$

- a. WITH E AS (SELECT pid, CARDINALITY(skills) AS skillCount FROM personHasSkills)
SELECT pid
FROM E
WHERE skillCount = (SELECT MAX(skillCount) FROM E);
- b. WITH managesWho AS (SELECT DISTINCT p.pid,
array(SELECT m.eid FROM Manages m WHERE m.mid = p.pid order by 1) as
manages FROM Person P order by 1)
SELECT p.pid, p.pname, CARDINALITY(m.manages) as number_managed
FROM Person p, managesWho m
WHERE p.pid = m.pid;

*I had to create my own relation and inserted values in order to test anything with Manages relation:

```
CREATE TABLE Manages(mid integer,  
                      eid integer,  
                      primary key(mid, eid),  
                      foreign key (mid) references Person(pid),  
                      foreign key (eid) references Person(pid));
```

```
INSERT INTO Manages VALUES
```

```
(1005, 1011),  
(1001, 1009),  
(1009, 1014),  
(1009, 1018),  
(1009, 1012),  
(1014, 1002),  
(1015, 1016),  
(1015, 1010),  
(1015, 1003),
```

(1015, 1006),

(1015, 1004),

(1006, 1008),

(1017, 1007);