Group members: Bach Xuan Phan – 934553922 – phanx@oregonstate.edu
Hyuntaek Oh – 934540212 – ohhyun@oregonstate.edu

1.
(a)
Relational Algebra:
$\pi_{ename}[\sigma_{w1.eid=emp.eid}\ [\sigma_{w1.eid=w3.eid}\ [\sigma_{w1.eid=w2.eid}\ [\rho\ (\text{w1},$
$\sigma_{dept.did=works.did\ and\ dept.dname='Hardware'}(works \times dept)) \times \rho\ (\text{w2},$
$\sigma_{dept.did=works.did\ and\ dept.dname='Software'}(works \times dept))\ ] \times \rho\ (\text{w3},$
$\sigma_{dept.did=works.did\ and\ dept.dname='Research'}(works \times dept))] \times emp]]$

Non-recursive Datalog:
Q(ename) :- emp(eid,ename,age,salary), works(eid,did,pct_time), dept(did, 'Hardware', budget, managerid), works(eid,did,pct_time), dept(did, ' Software', budget, managerid), works(eid,did,pct_time), dept(did, ' Software', budget, managerid)

Relational Calculus:
Q(ename) = $\forall_{ename}$ emp (eid, ename, age, salary) ∧ [($\exists_{dname}$ dept(did, 'Hardware', budget, manageid) ∧ $\forall_{dname}\forall_{did}$ (works∧ dept ⇒ w1.did=d1.did)∨ $\exists_{dname}$ dept (did, 'Software', budget, manageid)∧ $\forall_{dname}\forall_{did}$(works ∧ dept ⇒ w1.did=d1.did) ∨ $\exists_{dname}$dept(did, 'Research', budget, manageid) $\forall_{dname}\forall_{did}$(works∧ dept ⇒ w1.did=d1.did)]

SQL:
```
SELECT e.ename FROM (SELECT w.* FROM works w
INNER JOIN dept d1 ON d1.did = w.did AND d1.dname IN ('Hardware')) w1
INNER JOIN (SELECT w.* FROM works w
INNER JOIN dept d1 ON d1.did = w.did AND d1.dname IN ('Software')) w2
ON w1.eid = w2.eid
INNER JOIN (SELECT w.* FROM works w
INNER JOIN dept d1 ON d1.did = w.did AND d1.dname IN ('Research')) w3
ON w1.eid = w3.eid
INNER JOIN emp e ON w1.eid = e.eid
```

(b)
Relational Algebra:
$\pi_{dname}[\sigma_{w.eid\ is\ NULL}[(\text{dept} \cap \text{works}) \cup (\text{dept-works})]]$

Non-recursive Datalog:
Q(dname) :- dept(did, dname, budget, managerid), works(NULL, did, pct_time)

Relational Calculus:
Q(dname) = $\exists_{dname}$[w.eid is 'NULL' ⇒ (dept (did, dname, budget, managerid)∧ works(eid, did, pct_time))∨ ¬(works(eid, did, pct_time))]

SQL:
```
SELECT DISTINCT d.dname FROM dept d
LEFT JOIN works w ON d.did = w.did
WHERE w.eid IS NULL
```

(c)
SQL:

```sql
SELECT DISTINCT d1.managerid FROM dept d1
WHERE d1.budget > 1500000 AND d1.managerid NOT IN
(SELECT UNIQUE d1.managerid FROM dept d1
WHERE d1.budget <= 1500000)
```

(d)

“Whose salary is less than or equal to the salary of every employee” equals to “whose salary is the lowest”

SQL:

```sql
SELECT e1.ename FROM emp e1 WHERE e1.salary <= (SELECT MIN(e2.salary) FROM emp
e2)
```

(e)

SQL:

```sql
SELECT e.ename FROM dept d1
INNER JOIN emp e ON d1.managerid = e.eid
WHERE d1.budget = (SELECT MAX(d2.budget) FROM dept d2)
```

(f)

SQL:

```sql
SELECT d.dname, AVG(e.salary)AS 'average employee salary' FROM dept d
INNER JOIN works w ON w.did = d.did
INNER JOIN emp e ON e.eid = w.eid
WHERE d.budget >= 50
GROUP BY w.did
```

(g)

SQL:

```sql
SELECT d.managerid FROM dept d
GROUP BY d.managerid
HAVING SUM(d.budget) = (SELECT MAX(d2.sum) FROM (SELECT SUM(d1.budget) AS
sum FROM dept d1
GROUP BY d1.managerid) d2)
```

(h)

SQL:

```sql
SELECT e1.ename FROM emp e1, works w1
WHERE e1.eid = w1.eid AND w1.did = (SELECT d1.did FROM dept d1 WHERE d1.dname =
'Hardware') AND w1.eid NOT IN
(SELECT w1.eid FROM works w1
WHERE w1.did <> (SELECT d1.did FROM dept d1 WHERE d1.dname = 'Hardware'))
```

2.

< Relational Algebra to Non-recursive Datalog >

$$1.\ P \times Q \coloneqq ans\left(\overrightarrow{x}, \overrightarrow{y}\right) \leftarrow P\left(\overrightarrow{x}\right), Q\left(\overrightarrow{y}\right)$$

$$2.\ \sigma_F(R) \coloneqq ans\left(\overrightarrow{x}\right) \leftarrow R\left(\sigma_F\left(\overrightarrow{y}\right)\right)$$

$$3.\ \pi_{j_1, \cdots, j_n}(R) := ans(x_{j_1}, \cdots, x_{j_n}) \leftarrow R(x_1, \cdots, x_m)$$

< Non-recursive Datalog to Relational Algebra >

$$1.\ ans\left(\overrightarrow{x}\right) \leftarrow R_1\left(\overrightarrow{x_1}\right), \cdots, R_n\left(\overrightarrow{x_n}\right) := R_1 \times R_2 \times \cdots \times R_n$$

$$2.\ ans\left(\overrightarrow{x}\right) \leftarrow R\left(\sigma_F\left(\overrightarrow{x_1}, \cdots, \overrightarrow{x_n}\right)\right) := \sigma_F(R)$$

$$3.\ ans\left(\overrightarrow{x_1}, \cdots, \overrightarrow{x_n}\right) \leftarrow R(x_1, \cdots, x_n) := \pi_{1, \cdots, n}(R)$$

For example, $H(x, y) :- R(x, u, v), S(y, u)$ be a rule.

It can be expressed $H = \pi_{1,4}\,[\sigma_{2=2}(R \times S)]$ as a form of Relational Algebra (RA) with cartesian product, selection, and projection.

In the process of calculating this RA, $R \times S$ would be show the result of product of relation R and S, which means there are many redundant constants in the result.

x	u	v	y	u
x_1	u_1	v_1	y_1	u_1
x_2	u_2	v_2	y_1	u_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_n	u_n	v_n	y_n	u_n

< The table of $R \times S$ >

Then, by using selection, the result of the product operation will be filtered because selection operation works for separating between factual distinct data and redundant one.

x	u	v	y	u
x_1	u_1	v_1	y_1	u_1
x_2	u_2	v_2	y_2	u_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_n	u_n	v_n	y_n	u_n

< The table of $\sigma_{2=2}(R \times S)$ >

Finally, projection operation can help generate target variables and constants.

x	y
x_1	y_1
$\vdots$	$\vdots$
x_n	y_n

< The table of $\pi_{1,4}[\sigma_{2=2}(R \times S)]$ >