**a.**

```
SELECT emp.ename
FROM emp
WHERE emp.eid = (SELECT t1.eid
                 FROM (SELECT works.eid
                       FROM dept, works
                       WHERE dept.did = works.did AND dept.dname = 'Hardware') AS t1,

                       (SELECT works.eid
                       FROM dept, works
                       WHERE dept.did = works.did AND dept.dname = 'Software') AS t2,
                       (SELECT works.eid
                       FROM dept, works
                       WHERE dept.did = works.did AND dept.dname = 'Research')
                       AS t3
                 WHERE t1.eid = t2.eid AND t2.eid = t3.eid AND t1.eid= t3.eid)
```

Relational algebra

$$(\Pi_{emp.ename}(\sigma_{emp.eid = works.eid \wedge works.did = dept.did \wedge dept.dname = "Hardware"}(dept \times emp \times works))) \cap$$
$$(\Pi_{emp.ename}(\sigma_{emp.eid = works.eid \wedge works.did = dept.did \wedge dept.dname = "Software"}(dept \times emp \times works))) \cap$$
$$(\Pi_{emp.ename}(\sigma_{emp.eid = works.eid \wedge works.did = dept.did \wedge dept.dname = "Research"}(dept \times emp \times works))) \cap$$

Datalog

$$Q1(ename) :- emp(ename, eid, age1, salary1),\ works(eid, did, pctime1),\ dept(did, Hardware, budget1, managerid1))$$
$$Q2(ename) :- emp(ename, eid, age2, salary2),\ works(eid, did, pctime2),\ dept(did,\ Software, budget2, managerid2)$$
$$Q3(ename) :- emp(ename, eid, age3, salary3),\ works(eid, did, pctime3),\ dept(did, Research, budget3, managerid3)$$
$$Q4(ename) :- Q1(ename), Q2(ename), Q3(ename)$$

**b.**

```
SELECT DISTINCT dept.dname
FROM dept, works
WHERE dept.did NOT IN (SELECT works.did
                       FROM works
                       GROUP BY works.did)
```

Relational algebra

$$\Pi_{dept.dname}(dept) - (\Pi_{dept.dname}(\sigma_{emp.eid = works.eid \wedge works.did = dept.did}(dept \times emp \times works)))$$

Datalog

$$Q(dname) :- emp(ename, eid, age, salary),\ works(eid, did, pctime),\ dept(did, dname, budget, managerid)$$
$$Y(dname) :- dept(dname)\ not\ Q(dname)$$

**c.**

```
SELECT t1.managerid
FROM (SELECT MIN(budget) as min_bud ,managerid
        FROM dept
        GROUP BY managerid) AS t1
WHERE t1.min_bud> 1500000
```

**d.**

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

**e.**

```
SELECT emp.ename
FROM emp
WHERE emp.eid = ANY(SELECT t2.managerid
                    FROM (SELECT SUM(budget) as all_budget, managerid
                          FROM dept
                          GROUP BY dept.managerid) as t2
                          WHERE t2.all_budget >= ALL (SELECT SUM(budget) as
                    total_budget FROM `dept` GROUP BY dept.managerid))
```

**f.**

```
SELECT dept1.dname, AVG(emp.salary)
FROM (SELECT dept.did, dept.dname
        FROM dept
        WHERE dept.budget >= 50
        GROUP by dept.did) AS dept1, works, emp
WHERE dept1.did = works.did AND works.eid = emp.eid
GROUP BY dept1.did
```

**g.**

```
SELECT t1.managerid
FROM (SELECT dept.managerid, SUM(dept.budget) as budget
         FROM dept
         GROUP BY dept.managerid) AS t1
WHERE budget = (SELECT MAX(t1.budget)
                FROM (SELECT dept.managerid, SUM(dept.budget) as budget
                      FROM dept
                      GROUP BY dept.managerid) AS t1)
```

**h.**

```
SELECT DISTINCT emp.ename
FROM(
        SELECT dept1.did, AVG(emp.salary) AS salary
        FROM (SELECT dept.did, dept.dname
        FROM dept
        WHERE dept.budget >= 50
        GROUP by dept.did) AS dept1, works, emp
        WHERE dept1.did = works.did AND works.eid = emp.eid
        GROUP BY dept1.did) as dept_avg ,works, emp
WHERE dept_avg.did = works.did AND works.eid = emp.eid AND emp.salary <= dept_avg.salary
```

**i.**

```
SELECT emp.ename
FROM emp,    (SELECT works.eid
              FROM works
              GROUP BY works.eid
              HAVING COUNT(works.did) = 1) as count_emp
WHERE count_emp.eid = emp.eid AND emp.eid IN (SELECT works.eid
                                              FROM works
                                              WHERE works.did = (SELECT dept.did
                                          FROM dept WHERE dept.dname = “Hardware”) )
```

---

2 : Query Containment

(a) Explain whether the homomorphism theorem holds for conjunctive queries with equality, e.g., (T (s) : − R(x, y), x = y). If your answer is no, then modify the theorem to hold for these queries. (1 point)

the answer is no.

#1

Let q,q’ be two conjunctive queries that have the same distinguished variables
and let T,T’ be their tableaux. assume homomorphism with $Q_1, Q_2$

$Q_1(x) \ :- \ R(x,y)\ x = y$

$Q_2(x) \ :- \ R(x,y),\ R(z,'Joe')$

note that T, ignoring the declaration of distinguished variables, can be seen as a database instance
note also that $x \ \in \ Q_1(T)$
since $Q_1 \subseteq Q_2$ it follows that $x \in Q_2(T)$

$h(y) = y$

But, $R(z, 'Joe')$ does not match with $Q_1$

Since $Q_1 \nsubseteq Q_2$ , this query does not holds homomorphism

head variable treat ‘x’ as constants
q = (T,u) / q :- R(x,y) x=y
q’ = (T’,u’) / $q' :- R'(x', y'), x' = y'$
suppose exist a homomorphism θ from q’ and q.
$I$ is an instance over R. $q(I) \subseteq q'(I)$
a mapping $\delta : T(q') \rightarrow T(q)$
$\delta(c) = c$ for every constant c
$\delta(x) = x$ for every variables x of q’
if $\delta(x) = \delta(y)$ , it holds homomorphism

(b) Explain whether the homomorphism theorem holds for conjunctive queries with ≤, e.g., (T(s) : − R(x,y),x ≤ y). (1 point)

yes, homomorphism hold for this conjunctive queries

q1:- R(x,y), R(y,z)

q2:- R(x,y), R(y,z), R(z,t), x ≤ y,

homomorphism holds for q2⊆q1

h: q1→q2 :h(y) = y, h(z) = z, q2 ⊆ q1