20(a) $\{ e.deptName, sm.major \mid \forall e \in employedBy \; \forall sm \in studentMajor$

$(e.sid = sm.sid \rightarrow e.salary >= 2000)\}$

$\vee$ [union operation]

$\{ e.deptName, sm.major \mid \forall e \in employedBy \; \forall sm \in studentMajor$

$(e.sid = sm.sid \rightarrow e.salary >= 2000)\}$

20) <b> { s1.sid, s2.sid | Student(s1) ∧ Student(s2) ∧
s1.sid ≠ s2.sid
∧
(¬∃h ∈ hasFriend ¬∃e ∈ employedBy (s1.sid = h.sid ∧
h.sid2 = e.sid ∧ e.deptName = 'CS')
— [difference operation]
(¬∃h ∈ hasFriend ¬∃e ∈ employedBy (s2.sid = h.sid ∧
h.sid2 = e.sid ∧ e.deptName = 'CS')
∧ [intersection operation]
(¬∃h ∈ hasFriend ¬∃e ∈ employedBy (s2.sid = h.sid1 ∧
h.sid2 = e.sid ∧ e.deptName = 'CS')
—
(¬∃h ∈ hasFriend ¬∃e ∈ employedBy (s1.sid = h.sid1 ∧
h.sid2 = e.sid ∧ e.deptName = 'CS')

20) <c> { m.Major | Major(m) ∧ ∃ sm ∈ StudentMajor (
(sm.major = m.major) ∧
(sm.major in ({ sm1.major | studentMajor(sm1) ∧
hasFriend(h) ∧ (h.sid1 = sm.sid) ∧ h.sid2 = sm1.sid }))
∧
∃sm1 ∈ studentMajor ∃h ∈ hasFriend (h.sid1 = sm.sid ∧
h.sid2 = sm1.sid ∧ sm1.major ≠ m.Major) }.

22) (a) { () | ∃ Major(m) ∧ ¬∃sm1 ∈ StudentMajor ¬ ∃sm2 ∈ studentMajor
((sm1.major = sm2.major) ∧ (m.major = sm1.major) ∧
(m.major = sm2.major) → (sm1.sid != sm2.sid)) }.

23) <a>
{() | ∀ (∀ eb1 ∈ employedBy ∧ employedBy (eb2) ∧ hasFriend (hf)
(hf.sid1 = eb1.sid) → (eb1.deptName = eb2.deptName ∧
eb1.salary = eb2.salary ∧ eb2.sid = hf.sid2 ∧ eb1.sid = hf.sid1))}

24) <a>

$\{() \mid \forall (studentMajor(sm1) \wedge studentMajor(sm2) \wedge employedBy(eb1) \wedge employedBy(eb2) \wedge$

$(eb1.sid \neq eb2.sid \wedge eb1.deptName = eb2.deptName \wedge$
$eb1.salary = eb2.salary \wedge sm1.sid = eb1.sid \wedge$
$sm2.sid = eb2.sid \wedge sm1.major = sm2.major \wedge$
$sm1.sid \neq sm2.sid))\}$