HOMEWORK 3 SOLUTION
CS430

Each one cost 9 points except question i) for 10 points. All queries are tested by Jason; you can play with your query using xing.sql.

a) SELECT E.classnum
   FROM Enrollment E, Student S
   WHERE S.ssn = E.student_ssn AND S.sname LIKE "A%";

b) SELECT S.ssn, S.sname
   FROM Student S
   WHERE (SELECT COUNT(*)
            FROM Enrollment E
            WHERE E.student_ssn = S.ssn) = 1

c) SELECT (SELECT COUNT(DISTINCT student_ssn)
            FROM Enrollment INNER JOIN Class ON Enrollment.classnum = Class.classnum
            WHERE dcode = 'CS') / (SELECT COUNT(*) FROM Student)*100 as PERCENT;

d) SELECT F.fname
   FROM Class C, Faculty F
   WHERE C.instructor_ssn=F.ssn
   GROUP BY F.ssn
   HAVING COUNT(*)>=2
e) SELECT S.ssn, S.sname,

(SELECT COUNT(*)

FROM Enrollment E
WHERE S.ssn = E.student_ssn)

AS CourseCount

FROM Student S

ORDER BY CourseCount, S.ssn ASC

f) SELECT F.fname

FROM Faculty F

WHERE (SELECT COUNT(C1.classnum)

FROM Class C1
WHERE C1.instructor_ssn = F.ssn) >=

2*(SELECT COUNT(C2.classnum)

FROM Class C2 INNER JOIN Faculty F2 ON C2.instructor_ssn = F2.ssn
WHERE F2.fname = 'Smith')

AND (SELECT COUNT(C3.classnum)

FROM Class C3
WHERE C3.instructor_ssn = F.ssn) >= 1

Trick: If Prof. Smith has no teaching courses, for X >= 2 * 0, X can be 0 which means that all teachers can be selected even they don’t have teaching courses as Prof. Smith. We ask for at least one course should be taught.

g) SELECT P.dcode, P.cnum

FROM Prereq P
GROUP BY P.cnum

HAVING COUNT(*) = (SELECT MIN(numPrereqs) as minPrereqs

FROM ((SELECT COUNT(*) as numPrereqs

FROM Prereq

GROUP BY Prereq.cnum) as pre))

h) SELECT COUNT(*)

From Department D

WHERE D.chair_ssn IS NULL

i) Let units = credits, here is a solution (Inspired by Matthew Dunlap). Any reasonable queries are accepted including taking units as classes, etc.

SELECT ssn, sname

FROM Student s

WHERE IFNULL((SELECT SUM(credits) FROM Enrollment INNER JOIN Class ON

Enrollment.classnum = Class.classnum INNER JOIN Course ON

Class.dcode = Course.dcode AND Class.cnum = Course.cnum

WHERE student_ssn = s.ssn),0) + IFNULL((SELECT SUM(credits) FROM Transcript

INNER JOIN Course ON Transcript.dcode = Course.dcode AND

Transcript.cnum = Course.cnum WHERE student_ssn = s.ssn),0) >= 9

j) My database design takes Student.major = Department.dcode, so there is an easy answer:

SELECT D.dcode, COUNT(*) as NumGrads
FROM Student S, Department D
WHERE S.status = 'Grad' AND S.major = D.dcode
GROUP BY D.dcode
UNION

SELECT D.dcode, COUNT(*) as NumUndergrads
FROM Student S, Department D
WHERE S.status <> 'Grad' AND S.major = D.dcode
GROUP BY D.dcode;

If you say we need to find students through Class, Enrollment and Student, the answer may not be true if a student does not take any course (not been enrolled). Credits are still given for such query since each one has different database design.

k) SELECT D.dcode, (SELECT F.rank
    FROM Faculty F
    WHERE F.dcode = D.dcode
    ORDER BY rank DESC LIMIT 1) AS highestRank,
    (SELECT COUNT(*)
    FROM Faculty F2
    WHERE F2.dcode = D.dcode AND F2.rank = highestRank)
    as numberOfFaculty FROM Department D