CSCI 4333 Design of Database Systems
Spring 2024
Section 1 Suggested Solution to Final Examination

[1] (a) For example:

```
SELECT DISTINCT s.schoolName,
       COUNT(f.facId) AS `Number of advising faculty members`
FROM school AS s INNER JOIN department AS d USING (schoolCode)
       INNER JOIN faculty AS f USING (deptCode)
       INNER JOIN student AS st ON (f.facId = st.advisor)
GROUP BY 1
ORDER BY 2 DESC;
```

(b)

```
SELECT DISTINCT s.stuId,
       CONCAT(s.fname, ' ', s.lname) AS student,
       COUNT(c.classId) AS `# of CSCI classes`
FROM student AS s INNER JOIN enroll AS e USING (stuId)
       INNER JOIN class AS c USING (classId)
       INNER JOIN course AS co USING (courseId)
WHERE co.rubric = 'CSCI'
AND s.major = 'CSCI'
GROUP BY 1, 2
HAVING `# of CSCI classes` >= 3
ORDER BY `# of CSCI classes` DESC;
```

(c)

```
WITH t1 AS
(SELECT sc.schoolCode, COUNT(s.major) AS numMajors
 FROM school AS sc LEFT JOIN department AS d USING (schoolCode)
       LEFT JOIN student AS s ON (s.major = d.deptCode)
 GROUP By 1),
 t2 AS (SELECT sc.schoolCode, COUNT(s.major) AS numMinors
       FROM school AS sc LEFT JOIN department AS d USING (schoolCode)
       LEFT JOIN student AS s ON (s.minor = d.deptCode)
 GROUP By 1)
SELECT s.schoolCode, s.SchoolName, t1.numMajors, t2.numMinors
FROM school AS s INNER JOIN t1 USING (schoolCode)
       INNER JOIN t2 USING (schoolCode);
```

(2)

(a) F (b) F (c) T (d) F (e) F

(f) T (g) T (h) F (i) F (j) T

(k) Anything is acceptable. This is the famous liar paradox.

(3)

(a) R(A,B,C,D) with {BC->D, D->AB}: Canonical Cover: same
CK: [1] BC, [2] CD; prime: B, C, D; Highest NF: 1NF; D->A violate 2NF.

(b) R(A,B,C,D) with {BC->D, D->A}: Canonical Cover: same
CK: [1] BC; prime: B, C; Highest NF: 2NF; D->A violates 3NF

(c) R(A,B,C,D) with {BC->D, D->ABC}: Canonical Cover: same
CK: [1] BC, [2] D; prime: B, C, D; Highest NF: BCNF

(4) For R(A,B,C,D,E) {A->B, AB->D, AD->E}

(a) Canonical cover: {A->BDE}
(b) Candidate Key: [1] AC; Prime attributes: A, C
(c) 1NF, as A->BCE violate 2NF
(d) R1(A,B,D,E) {A->BDE} in BCNF, and R2(A.C) {} in BCNF.

(5) For example:

```
form = cgi.FieldStorage()
dcode = form.getfirst('dcode')

print('<h3Department information</h3>')

#      SQL
query = '''
SELECT f.facId,
    CONCAT(f.fname, ' ', f.lname) AS faculty,
    COUNT(c.classId) AS n_classes
FROM faculty f LEFT JOIN class c ON (f.facId = c.facId)
WHERE f.deptCode = %s
GROUP BY f.facId, faculty;
'''
cursor.execute(query, (dcode,))

if (cursor.rowcount > 0):
    print('<h3>Department "' + dcode + '" has the following faculty members.</h3>')
    print('<ol>')
    for (facId, faculty, n_classes) in cursor:
        print('<li>(id# ' + str(facId) + ') ' + faculty + ": teaches " +
str(n_classes) + ' classes</li>')
    print('</ol>')
else:
    print('<h3>Department ' + dcode + ' does not exist or has no faculty member.')
```

(6) For example:

```
use toyu;
db.student.find(
       { "minor": {$ne: null},
         "major": "ITEC"
       },
       {  "stuId": 1,
          "name": { $concat: ["$fname", " ", "$lname"] },
```

```
            "minor": 1,
            "advisor": {"$ifNull": ["$advisor", "no advisor"]},
            "credits": "$ach",
            "_id": 0    }
)
```

(7) (a) D

Since D->A, and A is a CK, D must be a CK itself.

(b) (i)

AId -> ALName, AFName
BId -> BTitle, Publisher, Edition, Year
BTitle, Publisher, Edition -> BId

(ii) CK: {AId, BId} and {AId, BTitle, Publisher, Edition}
(iii) 1NF since AId -> ALName violates 2NF.