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

[1] (a) For example:

```
SELECT DISTINCT s.stuId, CONCAT(s.fName, ' ', s.lName) AS student,
    COUNT(DISTINCT c.courseId) AS `# CSCI courses`
FROM student s LEFT JOIN enroll e ON (s.stuId = e.stuId)
    LEFT JOIN class c ON (e.classId = c.classId)
    LEFT JOIN course co ON (c.courseId = co.courseId)
WHERE co.rubric = 'CSCI'
GROUP BY s.stuId, student
ORDER BY `# CSCI courses` DESC;
```

(b)

```
CREATE OR REPLACE VIEW DepartmentSummary AS
WITH t1 AS
(SELECT co.rubric AS deptCode, COUNT(c.classId) AS numClasses
 FROM course AS co INNER JOIN class AS c USING (courseId)
 GROUP BY deptCode),
t2 AS
(SELECT deptCode, COUNT(facId) AS numFaculty
 FROM faculty
 GROUP BY deptCode)
SELECT d.deptCode, d.deptName AS department,
    IFNULL(t2.numFaculty, 0) AS numFaculty,
    IFNULL(t1.numClasses, 0) AS numClasses
FROM department AS d LEFT JOIN t1 USING (deptCode)
    LEFT JOIN t2 USING (deptCode);
```

(c)

```
DELIMITER //
CREATE FUNCTION numCommonClasses(
    sid_1 INT,
    sid_2 INT) RETURNS INT
BEGIN
    DECLARE result INT;
    SELECT COUNT(DISTINCT e1.classId) INTO result
    FROM enroll AS e1 INNER JOIN enroll AS e2 USING (classId)
    WHERE e1.stuId = sid_1
    AND e2.stuId = sid_2;

    RETURN result;
END //
DELIMITER ;
```

(2)

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

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

(k) T

(3)

(a) R(A,B,C,D) with {A->B, B->C, C->A}: Canonical Cover: same
CK: [1] AD, [2] BD, [3] CD ; prime: A, B, C, D; Highest NF: 3NF; All FD violate BCNF.

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

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

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

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

(5) For example:

```
form = cgi.FieldStorage()
fid = form.getfirst('fid') or '1011'

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

#      SQL
query = '''
WITH t1 AS (
    SELECT facId, COUNT(classId) AS n_classes
    FROM class
    WHERE facId = %s
    GROUP BY facId
),
t2 AS (
SELECT advisor AS facId,
    GROUP_CONCAT(CONCAT('<li>', fName, ' ', LName, '</li>') SEPARATOR '') AS advisees
    FROM student
    WHERE advisor = %s
    GROUP BY advisor
)
SELECT CONCAT(f.fName, ' ', f.LName) AS faculty,
    IFNULL (t1.n_classes, 0) AS n_classes,
    IFNULL (t2.advisees, '') AS advisees
FROM faculty AS f LEFT JOIN t1 ON (f.facId = t1.facId)
     LEFT JOIN t2 ON (f.facId = t2.facId)
WHERE f.facId = %s;
'''
cursor.execute(query, (fid, fid, fid))
(faculty, n_classes, advisees) = cursor.fetchone()
print(f"{fid} ({faculty}): instructor of {str(n_classes)} classes.
Advisees:\n<ol>\n{advisees}\n</ol>")

print('</body></html>')
```

(6) For example:

```
use toyu;
db.student.find(
       { $or: [
         { "major": {$in: ['ENGL', 'ARTS']}},
         { "minor": {$in: ['ENGL', 'ARTS']}}
       ]
       },
       {  "stuId": 1,
          "major": {$ifNull: ["$major", "undeclared"]},
          "minor": {$ifNull: ["$minor", "undeclared"]},
        "status": {$cond: {if: {$gte: ["$ach", 60]}, then: "upper", else: "lower"}},
          "_id": 0    }
)
```

(7) (a) There is only one candidate key: AB. Superkeys are AB, ABC, ABD and ABCD.

(b) (i)

1. FD1. OrganizationId -> OrganizationName,
2. FD2. RoleId -> RoleName,
3. FD3. RoleName -> RoleId
4. FD4. Two choices with different assumptions, both acceptable.
    1. StudentId, OrganizationId, RoleId -> StartDate;
    2. StudentId, OrganizationId -> StartDate;

(ii) CK (for both FD4.1 or FD4.2): [1] { StudentId, OrganizationId, RoleId }, [2] { StudentId, OrganizationId, NameId }

(iii) 1NF, FD1 violates 2NF.