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

Last Name: _____ **First Name:** _____ **Student Id:** _____

Number: _____

Time allowed: *2 hours.* Total score: 100 points. *Closed* book examination. Two information sheets (letter size, both sides) prepared by yourself are allowed. Answer all questions. Turn in everything: question and answer papers, information sheet and sketch papers. They will be stapled together.

Academic honesty policy will be followed strictly. Cheating will be pursued vigorously and will result in a failing grade of D or below, a permanent academic record, and possibly other more serious penalties.

Use the toyu db in the supplementary sheet for questions on SQL and Python.

(1) [24 points] Construct SQL statements for the following queries. Make sure that your answers generate the exact results, including column names and orders (if ordered).

(a) List the stuId and names of the students and the number of CSCI courses they have enrolled in. The results should be ordered in the descending number of CSCI courses in the following manner. Note that Tony Hawk has enrolled in only three courses even though he has enrolled in 4 CSCI classes. He has enrolled in one CSCI course twice in two different class offerings.

```
+--------+------------+-----------------+
| stuId  | student    | # CSCI courses |
+--------+------------+-----------------+
| 100000 | Tony Hawk  |               3 |
| 100001 | Mary Hawk  |               2 |
| 100002 | David Hawk |               2 |
+--------+------------+-----------------+
3 rows in set
```

(b) Define a view DepartmentSummary to return deptId, deptName, the number of faculty, and the number of classes the department offered. Executing "SELECT * FROM DepartmentSummary;" will output the following.

```
+----------+------------------------------+------------+------------+
| deptCode | department                   | numFaculty | numClasses |
+----------+------------------------------+------------+------------+
| ACCT     | Accounting                   |          1 |          0 |
| ARTS     | Arts                         |          1 |          1 |
| CINF     | Computer Information Systems |          2 |          3 |
| CSCI     | Computer Science             |          4 |          6 |
| ENGL     | English                      |          1 |          2 |
| ITEC     | Information Technology       |          2 |          2 |
| MATH     | Mathematics                  |          0 |          0 |
+----------+------------------------------+------------+------------+
7 rows in set
```

(c) Write a SQL stored function numCommonClasses to take two stuId (type INT) and return the number of classes the students have enrolled in together.

```
MariaDB [toyu]> SELECT numCommonClasses(100000, 100001);
+-----------------------------------+
| numCommonClasses(100000, 100001) |
+-----------------------------------+
|                                 2 |
+-----------------------------------+
1 row in set
```

(2) [20 points + 2 Bonus] True or False. *Circle* one choice, or *clearly* write 'T' or 'F'.

(a) [T or F] A string is also an object in Python.

(b) [T or F] It is possible to define an index on three fields in MySQL.

(c) [T or F] SQL injection is based on attacker's attempts to change the structures of backend SQL statements in Web database applications.

(d) [T or F] It is possible to use OUT parameters in a MySQL stored functions.

(e) [T or F] The HTTP protocol is used for communications between Web browsers and Web servers.

(f) [T or F] ACID properties are fully supported by NoSQL DB.

(g) [T or F] If A and B are the only prime attributes in R(A,B,C,D), R may still have three candidate keys.

(h) [T or F] In SQL, a stored procedure cannot have any DELETE statement.

(i) [T or F] MongoDB is more object-oriented than relational DB.

(j) [T or F] It is possible for R(A,B,C,D) to have three canonical covers for a set of functional dependencies.

(k) (Bonus) [T F] Cats have nine lives or cat are mammals.

(3) [9 points] Short Questions. State the candidate keys and the highest normal forms of the following relations. Assume the relations are at least in 1NF.

(a) R(A,B,C,D) with {A->B, B->C, C->A}

(b) R(A,B,C,D) with {A->BC, B->CD}

(c) R(A,B,C,D) with {A->BD, B->C, C->A}

(4) [9 points] Consider the relation R(A,B,C,D,E)) {B->C, BC->A, AB->CD}

(a) Provide a canonical cover.

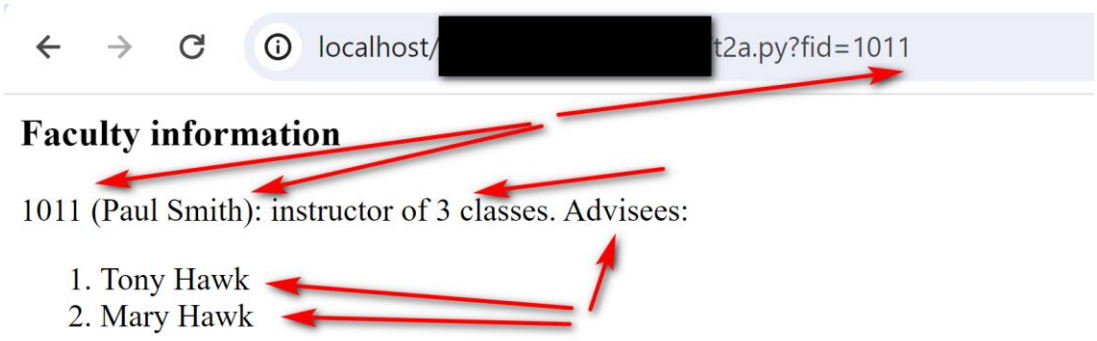
(b) Show all candidate keys.

(c) What is the highest normal form (up to BCNF)? Why?

(d) If it is not in BCNF, can you losslessly decompose R into component relations in BCNF while preserving functional dependencies? If yes, how? If no, why not?

(5) [16 points] Write a Python CGI program, t2a.py, to accept a HTTP Get parameter *fid* (faculty id) and display information of the faculty member: the number of classes taught and advisees.

For example, for http://.../t2a.py?fid=10011, the following result specifies the required output:



There is no need for error checking of the user input parameter *fid*. A skeleton for t2a.py is provided for you. You do not need to write this skeleton again in your answer.

```
from dbconfig import *
import pymysql
import cgi
import cgitb
cgitb.enable()

print("Content-Type: text/html;charset=utf-8")
print()
print ("<html>\n<head></head>\n<body>")

db = get_mysql_param()
cnx = pymysql.connect(user=db['user'], password=db['password'],
                      host=db['host'], database=db['database'])
cursor = cnx.cursor()
#      Get HTTP parameter: fid

# your code here. Write in the back of the previous page if needed.




print('</body></html>')
cursor.close()
cnx.close()
quit()
```

(6) [10 points] Consider the collection ‘student’ in the db ‘toyu’ as stored in MongoDB:

```
[ { _id: ObjectId("63c19f66c1fb90601512c759"), stuId: 100000, fname: 'Tony',
       lname: 'Hawk', major: 'CSCI', minor: 'CINF', ach: 40, advisor: 1011 },
  { _id: ObjectId("63c19f66c1fb90601512c75a"), stuId: 100001, fname: 'Mary',
    lname: 'Hawk', major: 'CSCI', minor: 'CINF', ach: 35, advisor: 1011 },
  { _id: ObjectId("63c19f66c1fb90601512c75b"), stuId: 100002, fname: 'David',
    lname: 'Hawk', major: 'CSCI', minor: 'ITEC', ach: 66, advisor: 1012 },
  { _id: ObjectId("63c19f66c1fb90601512c75c"), stuId: 100003, fname: 'Catherine',
    lname: 'Lim', major: 'ITEC', minor: 'CINF', ach: 20, advisor: null },
  { _id: ObjectId("63c19f66c1fb90601512c75d"), stuId: 100004, fname: 'Larry',
    lname: 'Johnson',  major: 'ITEC', minor: null, ach: 66, advisor: 1017 },
  { _id: ObjectId("63c19f66c1fb90601512c75e"), stuId: 100005, fname: 'Linda',
    lname: 'Johnson', major: 'CINF', minor: 'ENGL', ach: 13, advisor: 1015 },
  { _id: ObjectId("63c19f66c1fb90601512c75f"), stuId: 100006, fname: 'Lillian',
    lname: 'Johnson', major: 'CINF', minor: 'ITEC', ach: 18, advisor: 1016 },
  { _id: ObjectId("63c19f66c1fb90601512c760"), stuId: 100007, fname: 'Ben',
    lname: 'Zico', major: null, minor: null, ach: 16, advisor: null },
  { _id: ObjectId("63c19f66c1fb90601512c761"), stuId: 100008, fname: 'Bill',
    lname: 'Ching', major: 'ARTS', minor: null, ach: 90, advisor: null },
  { _id: ObjectId("63c19f66c1fb90601512c762"), stuId: 100009, fname: 'Linda',
    lname: 'King', major: 'ARTS', minor: 'CSCI', ach: 125, advisor: 1018 },
  { _id: ObjectId("63c19f66c1fb90601512c763"), stuId: 100111, fname: 'Cathy',
    lname: 'Johanson', major: null, minor: null, ach: 0, advisor: 1018 }
]
```

Construct Mongosh code to show the following information of all students majoring or minoring in ENGL or ARTS in the following manner. Note that the projected output field “status” is “upper” if the field “ach” is greater than or equal to 60. Furthermore, if the values of major or minor are null, the result should show ‘undeclared’. Answer on the back of the previous page if needed.

Tips:

1. You may use the $cond operator of the format: { $cond: { if: <boolean-expression>, then: <true-case>, else: <false-case> } } with obvious meaning.
2. Mongo supports set membership operator: { field: { $in: [<value1>, <value2>, ... <valueN>] } } returns true if the field is in the array […].
3. {$gte: {“$x”, 10}} if the field x >= 10.

```
[
  { stuId: 100005, major: 'CINF', minor: 'ENGL', status: 'lower' },
  {
    stuId: 100008,
    major: 'ARTS',
    minor: 'undeclared',
    status: 'upper'
  },
  { stuId: 100009, major: 'ARTS', minor: 'CSCI', status: 'upper' }
]
```

(7) [12 points] (a) [3 points] Three facts are known for R(A,B,C,D):

1. D is a non-prime attribute.
2. A and B are prime attributes.
3. There are exactly four superkeys.

List the candidate keys.

(b) [9 points] Membership(StudentId, OrganizationId, OrganizationName, RoleId, RoleName, StartDate). The relation stores the information of student's participations in organizations and their roles. For example, the row

('S11', 'O22, 'DB League', 'R10', 'member', '07-04-2024')

represents the fact that the student 'S11' joined the organization 'O22' with the name "DB League" in the role of 'member' (with the role id 'R10') starting from '07-04-2024'.

It is known that a student can join many organizations and may have many roles in an organization. There are separate tables (Student, Organization, and Role) to store detailed information of students, organizations, and roles respectively. The RoleId is always the same for the same role name, and vice versa.

(i) List the functional dependencies representing the specification above.

(ii) What are the candidate keys?

(iii) What is the highest normal form for the Membership relation? Why?