CSCI 4333 Design of Database Systems
Fall 2023
Section 1 Final Examination

Family Name: _______________ **First Name:** ____________ **Student Id:** ________

Number: _________

Time allowed: *2 hours*. Total score: 100 points. *Closed* book examination. Two information sheets prepared by yourself are allowed. Answer all questions. Turn in both question and answer sheets (if needed).

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 penalty!

Use the toyu database in the supplementary sheet for questions including 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 all ids, names and department names of all faculty members who taught less than 2 classes. Include the number of classes taught and show the result in the descending order of the number of classes.

```
+-------+-----------------+------------------------------+--------------------------+
| facId | faculty         | department                   | Number of classes taught |
+-------+-----------------+------------------------------+--------------------------+
|  1013 | David Love      | Computer Science             |                        1 |
|  1015 | Daniel Kim      | Computer Information Systems |                        1 |
|  1019 | Benjamin Yu     | Information Technology       |                        1 |
|  1016 | Andrew Byre     | Computer Information Systems |                        1 |
|  1020 | Katrina Bajaj   | English                      |                        1 |
|  1017 | Deborah Gump    | Information Technology       |                        0 |
|  1021 | Jorginlo Neymar | Accounting                   |                        0 |
+-------+-----------------+------------------------------+--------------------------+
7 rows in set
```

(b) List the id, name and major of every student who has not enrolled in any class and has not declared minor in the following manner.

```
+--------+----------------+-------+
| stuId  | student        | major |
+--------+----------------+-------+
| 100111 | Cathy Johanson | NULL  |
+--------+----------------+-------+
1 row in set
```

(c) For all students, show their ids, names, the number of CSCI courses enrolled and the number of CINF courses enrolled in the following manner. Must use common table expressions.

```
+--------+-----------------+------------------------+------------------------+
| stuId  | student         | Number of CSCI classes | Number of CINF classes |
+--------+-----------------+------------------------+------------------------+
| 100000 | Tony Hawk       |                      4 |                      2 |
| 100001 | Mary Hawk       |                      2 |                      0 |
| 100002 | David Hawk      |                      2 |                      1 |
| 100003 | Catherine Lim   |                      0 |                      0 |
| 100004 | Larry Johnson   |                      0 |                      2 |
| 100005 | Linda Johnson   |                      0 |                      2 |
| 100006 | Lillian Johnson |                      0 |                      1 |
| 100007 | Ben Zico        |                      0 |                      0 |
| 100008 | Bill Ching      |                      0 |                      0 |
| 100009 | Linda King      |                      0 |                      0 |
| 100111 | Cathy Johanson  |                      0 |                      0 |
+--------+-----------------+------------------------+------------------------+
11 rows in set
```

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

(a) [T or F] Python is weakly typed.

(b) [T or F] Using prepared SQL statements is a useful technique in mitigating SQL injection.

(c) [T or F] If A+ = B+ in R(A,B,C), both A and B are candidate keys.

(d) [T or F] The decomposition of R(A,B,C,D) {A->B, C->D} into R1(A,B) and R2(A,C,D) is lossless.

(e) [T or F] In SQL, a stored function cannot have any UPDATE statement.

(f) [T or F] ACID is an important property for all DBMS.

(g) [T or F] In SQL, a trigger can be called directly by a stored procedure (even though it cannot be called by a stored function.)

(h) [T or F] A relation must have at least one superkey.

(i) [T or F] MongoDB is an example of an object-oriented DBMS.

(j) [T or F] If AB->C, then A->B.

(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, BC->D}

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

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

(4) [9 points] Consider the following relation

R(A,B,C,D,E) {B->A, BA->D, D->E}

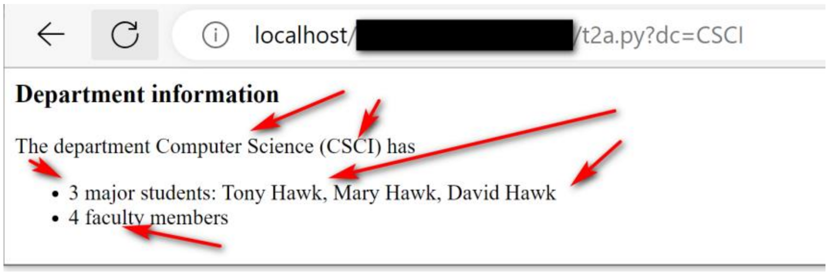
(a) Show all candidate keys?

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

(c) If it is not in BCNF, can you losslessly decompose R into component relations in BCNF while preserving functional dependencies? How?

(5) [16 points] Write a Python CGI program, t2a.py, to accept a HTTP Get parameter *dc* (department code) and display the department name, the number and names of students majoring in it, and the number of faculty members.

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



localhost/[redacted]/t2a.py?dc=CSCI

Department information

The department Computer Science (CSCI) has

- 3 major students: Tony Hawk, Mary Hawk, David Hawk
- 4 faculty members

There is no need for error checking of the user input parameter *dc*. 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>")

#      Get HTTP parameter: department code
form = cgi.FieldStorage()
dc = form.getfirst('dc')

db = get_mysql_param()
cnx = pymysql.connect(user=db['user'], password=db['password'],
                      host=db['host'], database=db['database'])
cursor = cnx.cursor()

# 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 to show the following information of all students with an advisor and with 15 to 35 ach credits: stuId, fname, lname, minor, and ach credits, in the following manner. Answer in the back of the previous page if needed.

Tip: MongoDB support null as a value. Furthermore, the expression: '"xyz": { $ifNull: ["$xyz", "not applicable"] }' returns the value of the field "xyz" is it is null. Otherwise, it returns "not applicable".

```
[
  {
    stuId: 100001,
    fname: 'Mary',
    lname: 'Hawk',
    advisor: 1011,
    minor: 'CINF',
    ach_credits: 35
  },
  {
    stuId: 100006,
    fname: 'Lillian',
    lname: 'Johnson',
    advisor: 1016,
    minor: 'ITEC',
    ach_credits: 18
  }
]
```

(7) [12 points + 2 points bonus] (a) [3 points] Four facts are known for R(A,B,C,D,E):

1. There are exactly 2 candidate keys.
2. One of the candidate keys is A.
3. B and E are prime attributes. C and D may or may not be prime attributes.
4. There are exactly 20 superkeys.

What is the second candidate key?

(b) [9 points] Consider the relation DormAssignment(StudentId, Major, DormId, DormName, Room, Semester, Year). StudentId is the id of a student with a major, which may be undeclared. A dorm has a unique id, DormId, and a unique name, DormName. Dorm assignment is semester-based. A row in the table stores the room of the dorm assigned to a student in a particular semester and year. There are separate tables for storing information about students, majors, and dorms. An example of a portion of the table indicating that the student S101 was assigned to three different rooms in three semesters is shown below. Rooms can be shared by students. Make reasonable assumptions.

StudentId	Major	DormId	DormName	Room	Semester	Year
S101	CSCI	D118	Jones Hall	2101	Spring	2022
S101	CSCI	D118	Jones Hall	3313	Fall	2022
S101	CSCI	D31	Roberts Hall	1024	Spring	2023
S211	MATH	D31	Roberts Hall	1024	Spring	2023

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

(ii) What are the candidate keys?

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

(c) [Bonus: 2 points] What is interesting about the phrase "modeling goldmine"?