CSCI 4333 Design of Database Systems
Spring 2024
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 (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 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 the numbers of advising faculty members (a faculty member who advises at least one student) by school in descending order of this number in the following manner.

```
+------------------------------+-----------------------------------+
| schoolName                   | Number of advising faculty members |
+------------------------------+-----------------------------------+
| Science and Engineering      |                                 6 |
| Human Sciences and Humanities |                                 2 |
+------------------------------+-----------------------------------+
2 rows in set
```

(b) List the students majoring in CSCI who have enrolled in three or more CSCI courses in the following manner.

```
+--------+-----------+-------------------+
| stuId  | student   | # of CSCI classes |
+--------+-----------+-------------------+
| 100000 | Tony Hawk |                 4 |
+--------+-----------+-------------------+
1 row in set
```

(c) For each school, list the number of departments in the school, the number of majors and the number of minors in the following manner.

```
+------------+--------------------------------+-----------+-----------+
| schoolCode | SchoolName                     | numMajors | numMinors |
+------------+--------------------------------+-----------+-----------+
| BUS        | Business                       |         0 |         0 |
| CSE        | Science and Engineering        |         7 |         6 |
| EDU        | Education                      |         0 |         0 |
| HSH        | Human Sciences and Humanities  |         2 |         1 |
+------------+--------------------------------+-----------+-----------+
4 rows in set
```

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

(a) [T F] {A->B, A->C} is a canonical cover of {A->B, AB->C}

(b) [T F] In SQL, a trigger can be defined for the event of creating a table.

(c) [T F] In SQL, a stored procedure can have side effects.

(d) [T F] The ACID property of DBMS ensures the absence of any logical inconsistency.

(e) [T F] If A+ = B+ in R(A,B,C), A and B are both prime attributes.

(f) [T F] In Python, an array is also an object.

(g) [T F] The decomposition of R into R1, R2, and R3 is lossless if R = R1 |x| R2 |x| R3, in which |x| is the natural join operator.

(h) [T F] In MongoDB, the find() method always returns a single document.

(i) [T F] In SQL injection, Python code is inserted by the attackers to use SQL to attack the backend database.

(j) [T F] MySQL allows the creation of an index on a column that is not unique.

(k) (Bonus) [T F] This statement is false.

(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 {BC->D, D->AB}

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

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

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

(a) Provide a canonical cover.

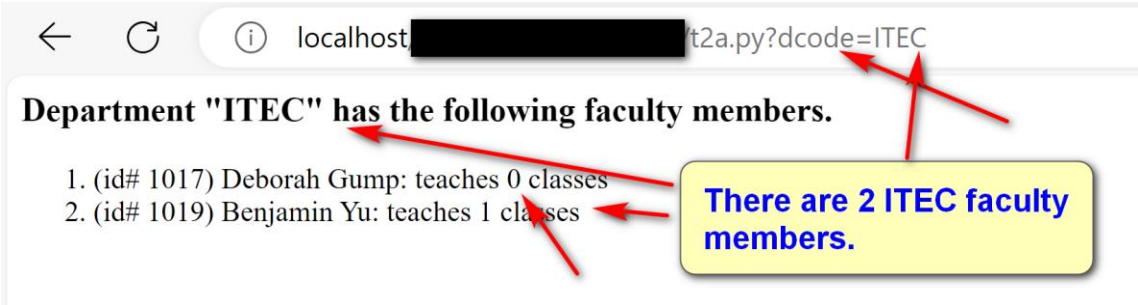
(b) Show all candidate keys.

(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 *dcode* (department code) and display the faculty members in the department and the courses they teach in the following manner.

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



There is no need for error checking of the user input parameter *dcode*. 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: dcode

# 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 in ITEC, and with a declared minor: stuId, student name, minor, and ach (credits) and advisor Id, in the following manner. If there is no assigned advisor, show "no advisor." Answer on the back of the previous page if needed.

Tip: MongoDB support null as a value.
The expression: '"abc": { $ifNull: ["$xyz", "unknown"] }' returns the value of the field "xyz" if "xyz" is not null. Otherwise, it returns "unknown".
Also, '"abc": { $concat: ["hello", " my", " friend"]} sets the field "abc" to "hello my friend"

```
[
  {
    stuId: 100003,
    minor: 'CINF',
    name: 'Catherine Lim',
    advisor: 'no advisor',
    credits: 20
  }
]
```

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

1. There are three candidate keys: A, BC, and a third one.
2. E is the only non-prime attribute.
3. D -> AB is true.

What is the third candidate key?

(b) [9 points] Consider the relation Author with an example portion of an instance below. For example, the first row stores the information an author, with AId A100 and Name 'Jenny Smith' and a book she authored: B1111 with the title 'Python for All,' third edition in the year 2019 by Springer. Note that edition 4 of 'Python for All' by Springer has two authors. A specific edition of a book has a unique BId and is published in a specific year. A book has a unique title when published by the same publisher. If a title changes, it starts a new sequence of editions. Two books (with various editions) by different publishers may have the same titles. For example, Linda Johnson may also author a book 'Python for All' by another publisher, 'Summer Hill.'

AId	ALName	AFName	BId	BTitle	Publisher	Edition	Year
A100	Jenny	Smith	B1111	Python for All	Springer	3	2019
A100	Jenny	Smith	B1112	Python for All	Springer	4	2020
A113	James	Smith	B1112	Python for All	Springer	4	2020
A237	Linda	Johnson	B1001	Python for All	Summer Hill	1	2017
A237	Linda	Johnson	B1005	Loving Java	Summer Hill	2	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 Author relation? Why?