CSCI 4333 Design of Database Systems
Spring 2025
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) Show the code, name, and number of major students for every department in the following manner.

```
+----------+------------------------------+-------------------+
| deptCode | department                   | # enrolled majors |
+----------+------------------------------+-------------------+
| ACCT     | Accounting                   |                 0 |
| ARTS     | Arts                         |                 2 |
| CINF     | Computer Information Systems |                 2 |
| CSCI     | Computer Science             |                 3 |
| ENGL     | English                      |                 0 |
| ITEC     | Information Technology       |                 2 |
| MATH     | Mathematics                  |                 0 |
+----------+------------------------------+-------------------+
7 rows in set
```

(b) Show the ids and names of faculty members of the school 'CSE' who have advised two or more students in the following manner.

```
+-------+------------+--------------------+
| facId | faculty    | Number of advisees |
+-------+------------+--------------------+
|  1011 | Paul Smith |                  2 |
+-------+------------+--------------------+
1 row in set (0.003 sec)
```

(c) List the id, name, and number of advisees of all faculty members who are the instructors of two or more classes of CSCI courses.

```
+-------+------------+--------------------+
| facId | faculty    | number of advisees |
+-------+------------+--------------------+
|  1011 | Paul Smith |                  2 |
|  1012 | Mary Tran  |                  1 |
+-------+------------+--------------------+
2 rows in set
```

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

(a) [T or F] In MySQL, the default InnoDB storage engine does not support the ACID property of transactions.

(b) [T or F] It is possible that "SELECT * FROM R NATURAL JOIN S" returns an empty set for the relations R and S.

(c) [T or F] MongoDB is an example of a wide column NoSQL DB.

(d) [T or F] In Python, a function is not an object.

(e) [T or F] The function ‘IS NOT NULL’ is a binary operator in SQL.

(f) [T or F] In SQL injection, a syntactically correct SELECT statement in SQL is entered into a text field of a Web page by the attackers.

(g) [T or F] For R(A,B,C), the following SQL statement contains an error.

```
SELECT A, COUNT(B)
FROM R
WHERE COUNT(B) >= 10
GROUP BY A;
```

(h) [T or F] In relational theory, for a relation R, if X+ = Z, then XW -> Z,

(i) [T or F] It is possible that R(A,B) is not in BCNF.

(j) [T or F] If a relation R is in 3NF, it is also in BCNF.

(k) [T or F] (Bonus) Tomorrow is 5/1/2024.

(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 {B->D, C->D, D->A}

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

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

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

(a) Provide a canonical cover.

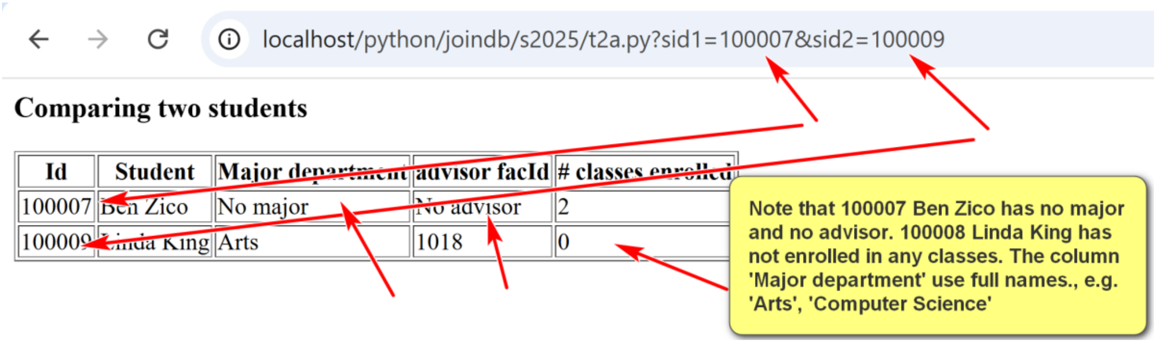
(b) Show all candidate keys.

(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 two HTTP Get parameters *sid1*and *sid2* (both student ids) and display comparison information of the two students, including their ids, names, majors (department names, not department code), adviosrs' facId, and numbers of classes enrolled.

For example, for http://.../t2a.py? sid1=100007&sid2=100009, the following result specifies the required output:



localhost/python/joindb/s2025/t2a.py?sid1=100007&sid2=100009

Comparing two students

Id	Student	Major department	advisor facId	# classes enrolled
100007	Ben Zico	No major	No advisor	2
100009	Linda King	Arts	1018	0

Note that 100007 Ben Zico has no major and no advisor. 100008 Linda King has not enrolled in any classes. The column 'Major department' use full names., e.g. 'Arts', 'Computer Science'

There is no need for error checking of the user input parameters. A skeleton for t2a.py is provided for you.

```
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()

# 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 query in JS to show the information of all students majoring or minoring in 'CINF' and with 15 or more ach credits in the following format. Answer in the back of the previous page if needed.

```
[
  {
    stuId: 100000,
    major: 'CSCI',
    minor: 'CINF',
    student: 'Tony Hawk',
    'ach credits': 40
  },
  {
    stuId: 100001,
    major: 'CSCI',
    minor: 'CINF',
    student: 'Mary Hawk',
    'ach credits': 35
  },
  {
    stuId: 100003,
    major: 'ITEC',
    minor: 'CINF',
    student: 'Catherine Lim',
    'ach credits': 20
  },
  {
    stuId: 100006,
    major: 'CINF',
    minor: 'ITEC',
    student: 'Lillian Johnson',
    'ach credits': 18
  }
]
```

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

1. There are two candidate keys. One of them is A.
2. D and E are non-prime attributes.
3. There are 20 superkeys.

What is the second candidate key?

(b) [9 points] Consider the relation Tutoring(TutorId, TutorEMail, StudentId, StudentEMail, SubjectId, SubjectName), which stores tutoring information about a tutor tutoring a student in a subject. For example (‘T11’, ‘karl404@gmail.com’, ‘S21’, ‘paul503@gmail.com’, ‘CSCI’, ‘Computer Science’) stores the information that the tutor ‘T11’, (with an email address of ‘karl404@gmail.com’), tutors the student ‘S21’ (with an email address of ‘paul503@gmail.com’) on the subject CSCI (with code ‘CSCI’ and name ‘Computer Science’).

It is known that TutorId, StudentId and SubjectId are unique identifiers of tutors, students, and subjects respectively. Information about the tutors, students, and subjects are stored in other relations. Only one email is stored for a tutor or a student. Both the code and name of a subject are unique. If necessary, make reasonable assumptions.

(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?