Database Systems
Fall 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 deptCode and the number of minors in the program in the following manner. List only those deptCode with *2 or fewer* minors. Note that the result is shown in the descending order of the number of minoring students.

```
+----------------+-----------------------------+
| Degree Program | Number of minoring students |
+----------------+-----------------------------+
| ITEC           |                           2 |
| CSCI           |                           1 |
| ENGL           |                           1 |
| MATH           |                           0 |
| ACCT           |                           0 |
| ARTS           |                           0 |
+----------------+-----------------------------+
6 rows in set
```

(b) List the number of faculty members and the number of majoring students of each department in the following manner.

```
+------------------------------+------------------+-----------------------------+
| department                   | Number of faculty | Number of majoring students |
+------------------------------+------------------+-----------------------------+
| Accounting                   |                1 |                           0 |
| Arts                         |                1 |                           2 |
| Computer Information Systems |                2 |                           2 |
| Computer Science             |                4 |                           3 |
| English                      |                1 |                           0 |
| Information Technology       |                2 |                           2 |
| Mathematics                  |                0 |                           0 |
+------------------------------+------------------+-----------------------------+
7 rows in set
```

(c) Write a function n_enrolled_classes(*sid, rub*) to return the number of classes a student with stuId *sid* has enrolled in. Count only the classes of the course rubric *rub*.

```
CREATE OR REPLACE FUNCTION n_enrolled_classes(
   sid INT,
   rub varchar(4))
RETURNS INT … your code here.
```

```
SELECT n_enrolled_classes(100000, 'CSCI');
+-------------------------------------+
| n_enrolled_classes(100000, 'CSCI') |
+-------------------------------------+
|                                   4 |
+-------------------------------------+
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] In relational theory, for a relation R, a set of attributes X is a candidate key of R if X+ = R.

(b) [T or F] A relation that is not in BCNF is also not in 3NF.

(c) [T or F] In MySQL, a transaction cannot be rollback if it is already committed.

(d) [T or F] In functional dependency, if A->C, then AB->C.

(e) [T or F] It is possible for R(A,B,C,D) to have 16 superkeys.

(f) [T or F] In DBMS, the 'A' in the ACID property ensures that all relations are in 1NF.

(g) [T or F] If the relation R has only two columns and is in 1NF, it is in 3NF

(h) [T or F] A MongoDB database can have many collections.

(i) [T or F] Wide column databases are considered as a kind of NoSQL database.

(j) [T or F] Parametrized SQL query helps mitigate SQL injection.

(k) [T or F] (Bonus) SQL stands for Somewhat Questionable Language or the Sun rises in the East.

(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->C, AB->C, C->AD}

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

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

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

(a) Provide a canonical cover.

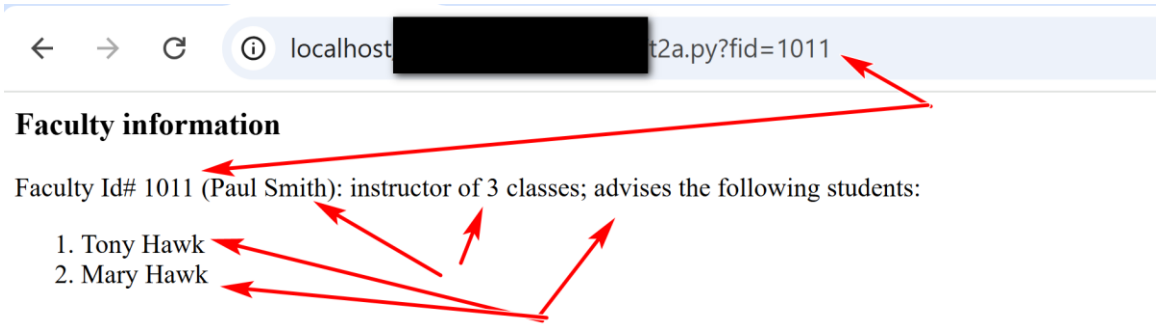
(b) Show all candidate keys.

(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 one HTTP parameter *fid* (facId) and display comparison information of a faculty member, including the facId, names, number of classes taught and advising information.

For example, for http://.../t2a.py?fid=1011, the following result specifies the required output:



There is no need for error checking of the input parameter. A skeleton for t2a.py is provided for you. Assume that dbconfig.py and dbconfig.ini are correctly set.

```
from dbconfig import *
import pymysql
import cgi

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 ‘faculty’ in the db ‘toyu’ as stored in MongoDB:

```
[ { _id: ObjectId("63c19f66c1fb90601512c73b"), facId: 1011,
      fname: 'Paul', lname: 'Smith', deptCode: 'CSCI', rank: 'Professor'},
  { _id: ObjectId("63c19f66c1fb90601512c73c"), facId: 1012,
      fname: 'Mary', lname: 'Tran', deptCode: 'CSCI', rank: 'Associate Professor'},
  { _id: ObjectId("63c19f66c1fb90601512c73d"), facId: 1013,
    fname: 'David', lname: 'Love', deptCode: 'CSCI', rank: null},
  {_id: ObjectId("63c19f66c1fb90601512c73e"), facId: 1014,
    fname: 'Sharon', lname: 'Mannes', deptCode: 'CSCI', rank: 'Assistant Professor'},
  { _id: ObjectId("63c19f66c1fb90601512c73f"), facId: 1015,
    fname: 'Daniel', lname: 'Kim', deptCode: 'CINF', rank: 'Professor'},
  { _id: ObjectId("63c19f66c1fb90601512c740"), facId: 1016,
    fname: 'Andrew', lname: 'Byre', deptCode: 'CINF', rank: 'Associate Professor'},
  { _id: ObjectId("63c19f66c1fb90601512c741"), facId: 1017,
     fname: 'Deborah', lname: 'Gump', deptCode: 'ITEC', rank: 'Professor'},
  {  _id: ObjectId("63c19f66c1fb90601512c742"), facId: 1018,
     fname: 'Art', lname: 'Allister', deptCode: 'ARTS', rank: 'Assistant Professor'},
  { _id: ObjectId("63c19f66c1fb90601512c743"), facId: 1019,
     fname: 'Benjamin', lname: 'Yu', deptCode: 'ITEC', rank: 'Lecturer'},
  { _id: ObjectId("63c19f66c1fb90601512c744"), facId: 1020,
    fname: 'Katrina', lname: 'Bajaj', deptCode: 'ENGL', rank: 'Lecturer'},
  { _id: ObjectId("63c19f66c1fb90601512c745"), facId: 1021,
    fname: 'Jorginlo', lname: 'Neymar', deptCode: 'ACCT', rank: 'Assistant Professor'}
]
```

Construct Mongosh query in JS to show Show the information of all faculty members of "CINF" or ITEC" who are tenure-track faculty: Professor, Associate Professor or Assistant Professor. Answer in the back of the previous page if needed.

```
[
  {
    facId: 1015,
    deptCode: 'CINF',
    faculty: 'Daniel Kim',
    'faculty rank': 'Professor'
  },
  {
    facId: 1016,
    deptCode: 'CINF',
    faculty: 'Andrew Byre',
    'faculty rank': 'Associate Professor'
  },
  {
    facId: 1017,
    deptCode: 'ITEC',
    faculty: 'Deborah Gump',
    'faculty rank': 'Professor'
  }
]
```

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

1. C is not a prime attribute.
2. B is a prime attribute.
3. B is a superkey.
4. AD is not a superkey.

What are the candidate key(s)?

(b) Consider the relation Participation(StudentId, OrganizationId, RoleId, RoleName StartDate). The relation stores information of student's participations in organizations and their roles. For example, the row

('S1234', 'O17, 'R1', 'member', '05-04-2025')

represents the fact that the student 'S1234' joined the organization 'O17' in the role of the name 'member' (with the role id 'R1') starting from '05-04-2025'. It is known that the role name and role id can identify each other. It is also known that a student can join many organizations but can only have one role in an organization.

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

(ii) What are the candidate keys?

(iii) What is the highest normal form of the relation Participation? Why?