DASC 5333 Database Systems for Data Science
CSCI 4333 Design of Database Systems
Spring 2025
Homework #5 Even More SQL

This assignment uses the toyu database in MySQL or MariaDB:
https://dcm.uhcl.edu/yue/courses/joinDB/spring2025/notes/toyu/toyu.html.

Provide the MySQL commands for the following queries. Make sure that your queries produce the exact output as shown below (however, row orders can be different).

The solution should be put in an executable .sql file. Name the file h5.sql. If Canvas disallows the upload of .sql file, add .txt at the end.

[1] Using toyu, show the numbers of majors in departments and colleges in the following manner.

```
+---------+---------------------+------------------------------------+------------------------+
| college | # majors in college | department                         | # majors in department |
+---------+---------------------+------------------------------------+------------------------+
| BUS     |                   0 | ACCT: Accounting                   |                      0 |
| CSE     |                   7 | MATH: Mathematics                  |                      0 |
| CSE     |                   7 | ITEC: Information Technology       |                      2 |
| CSE     |                   7 | CSCI: Computer Science             |                      3 |
| CSE     |                   7 | CINF: Computer Information Systems |                      2 |
| EDU     |                   0 |                                    |                      0 |
| HSH     |                   2 | ENGL: English                      |                      0 |
| HSH     |                   2 | ARTS: Arts                         |                      2 |
+---------+---------------------+------------------------------------+------------------------+
8 rows in set
```

[2] List the number of FK of tables in toyu that have at least one foreign key. Tips: there a table REFERENTIAL_CONSTRAINTS in the MySQL information_schema database.

```
+------------+--------+
| table      | num_fk |
+------------+--------+
| class      |      2 |
| course     |      1 |
| department |      1 |
| enroll     |      3 |
| faculty    |      1 |
| student    |      3 |
+------------+--------+
6 rows in set
```

[3] Write a procedure foreign_keys of a table of an INNODB schema. Its behavior as illustrated by the following examples. Tips: you may use the tables information_schema.REFERENTIAL_CONSTRAINTS, information_schema.INNODB_SYS_FOREIGN, and information_schema.INNODB_SYS_FOREIGN_COLS. The prototype of the procedure is provided below.

```
MariaDB [hw]> SET @n_fk = 0;
```

```
Query OK, 0 rows affected (0.001 sec)

MariaDB [hw]> CALL foreign_keys('toyu', 'student', @n_fk);
+--------------------------------+
| table: number of foreign keys |
+--------------------------------+
| toyu.student: 3               |
+--------------------------------+
1 row in set (0.010 sec)

+---------+-------------------------+
| column  | referenced_table.column |
+---------+-------------------------+
| advisor | faculty.facId           |
| major   | department.deptCode     |
| minor   | department.deptCode     |
+---------+-------------------------+
3 rows in set (0.095 sec)

Query OK, 1 row affected (0.210 sec)

MariaDB [hw]> SELECT @n_fk;
+-------+
| @n_fk |
+-------+
|     3 |
+-------+
1 row in set (0.001 sec)

MariaDB [hw]>
MariaDB [hw]> CALL foreign_keys('toyu', 'enroll', @n_fk);
+--------------------------------+
| table: number of foreign keys |
+--------------------------------+
| toyu.enroll: 3                |
+--------------------------------+
1 row in set (0.008 sec)

+---------+-------------------------+
| column  | referenced_table.column |
+---------+-------------------------+
| classId | class.classId           |
| grade   | grade.grade             |
| stuId   | student.stuId           |
+---------+-------------------------+
3 rows in set (0.096 sec)

Query OK, 1 row affected (0.254 sec)

MariaDB [hw]> SELECT @n_fk;
+-------+
| @n_fk |
+-------+
|     3 |
+-------+
1 row in set (0.000 sec)
```

```
MariaDB [hw]>
MariaDB [hw]> CALL foreign_keys('toyu', 'school', @n_fk);
+-------------------------------+
| table: number of foreign keys |
+-------------------------------+
| toyu.school: 0                |
+-------------------------------+
1 row in set (0.006 sec)

Empty set (0.104 sec)

Query OK, 1 row affected (0.132 sec)

MariaDB [hw]> SELECT @n_fk;
+-------+
| @n_fk |
+-------+
|     0 |
+-------+
1 row in set (0.001 sec)
```

The procedure:

```
CREATE PROCEDURE foreign_keys(
    IN db_name VARCHAR(64),
    IN table_name VARCHAR(64),
    OUT n_fk INT
) …
```

[4] Write a function column_count to count the number of columns in a MySQL table. Its behavior is illustrated by the examples below.

```
MariaDB [hw]> SELECT column_count('toyu', 'student');
+---------------------------------+
| column_count('toyu', 'student') |
+---------------------------------+
|                               7 |
+---------------------------------+
1 row in set (0.011 sec)

MariaDB [hw]>
MariaDB [hw]> SELECT column_count('toyu', 'faculty');
+---------------------------------+
| column_count('toyu', 'faculty') |
+---------------------------------+
|                               5 |
+---------------------------------+
1 row in set (0.025 sec)

MariaDB [hw]>
MariaDB [hw]> SELECT column_count('information_schema', 'columns');
+-----------------------------------------------+
| column_count('information_schema', 'columns') |
+-----------------------------------------------+
|                                            22 |
+------------------------------------------------+
1 row in set (0.031 sec)
```

The function:

```
DROP FUNCTION IF EXISTS column_count $$
CREATE FUNCTION column_count(
    db_name VARCHAR(64),
    table_name VARCHAR(64)
) RETURNS INT
…
```