Project 1 Project Description

In your first homework, you were asked to design a database that stores some information about a university. This project is to implement that design using a relational data model. Specifically, you are asked to write the following SQL scripts.

1. CreateTables.sql [Points: 15]

This script creates the following tables. Each table must be created with the table name, attribute names and corresponding types and length as specified. Also, make sure to specify primary key, candidate key and foreign key (if any), accordingly.

- **students**

  a) Attribute, type and length: *snum: integer, ssn: integer, name: varchar(20), gender: varchar(1), dob: varchar(10), c_addr: varchar(20), c_phone: varchar(10), p_addr: varchar(20), p_phone: varchar(10)*
  b) Primary key: *ssn*
  c) Candidate key: *snum*
  d) Foreign key: *N/A*

- **departments**

  a) Attribute, type and length: *code: integer, name: varchar(50), phone: varchar(10), college: varchar(20)*
  b) Primary key: *code*
  c) Candidate key: *name*
  d) Foreign key: *N/A*

- **degrees**

  a) Attribute, type and length: *name: varchar(50), level: varchar(5),* department_code: integer
  b) Primary key: *name, level*
  c) Candidate key: *N/A*
  d) Foreign key: *department_code refers to code in table departments*

- **courses**

  a) Attribute, type and length: *number: integer, name: varchar(50), description: varchar(50), credithours: integer, level: varchar(20), department_code: integer*
  b) Primary key: *number*
  c) Candidate key: *N/A*
  d) Foreign key: *department_code refers to code in table departments*

- **register**

  a) Attribute, type and length: *snum: integer, course_number: integer, regtime: varchar(20), grade: integer*
  b) Primary key: *snum, course_number*
  c) Candidate key: *N/A*

   d) Foreign key: *snum refers to snum in table students, course_number refers to number in table courses*

- **major**
   a) Attribute, type and length: *snum: integer, name: varchar(50), level: varchar(5)*
   b) Primary key: *snum, name, level*
   c) Candidate key: *N/A*
   d) Foreign key: *snum refers to snum in table students, name & level refer to name & level in table degrees*
- **minor**
   a) Attribute, type and length: *snum: integer, name: varchar(50), level: varchar(5)*
   b) Primary key: *snum, name, level*
   c) Candidate key: *N/A*
   d) Foreign key: *snum refers to snum in table students, name & level refer to name & level in table degrees*

2. InsertRecords.sql [Points: 0]

This script inserts the records to the appropriate tables created by CreateTables.sql. You must load the data from csv file to database, here is the reference link for load data from local. Kindly refer the csv file uploaded into canvas to load the data.

- **students :** students.csv
- **departments :** departments.csv
- **degrees :** degrees.csv
- **major :** major.csv
- **minor :** minor.csv
- **courses :** courses.csv
- **register :** register.csv

3. Query.sql [Points: 68]

This script prints out the following information

1) (3pts) The campus addresses of the students whose name is "Amy"
2) (5pts) The major name and major level of the students whose name is "Gail"
3) (5pts) The numbers and names of all courses offered by the department of Computer Science, order by course number
4) (5pts) The name of the students enrolled in Fall2022 semester.

5) (5pts) All degree names and levels offered by the department Computer Science, order by degree level
6) (5pts) The snum and names of all students who have a minor, order by student snum
7) (5pts) The names and snums of all non-undergraduate students enrolled in course “database”, order by snum. (“non-undergraduate students” means the major degrees of these students are MS or PhD levels)
8) (5pts) The name, snum and SSN of the students whose name contains letter “n” or “N”, order by snum
9) (5pts) The name, snum and SSN of the students whose name is between “Amy” and “Christopher”, order by name
10) (5pts) The course number, name and the number of students registered for each course, order by course number (if a course has no student registered, the count should be 0)
11) (6pts) The count of female students who major or minor in Software Engineering degrees at any level
12) (7pts) The degree name, degree level, and number of students of the most popular majors (I.e., the major with the highest number of students), order by degree name if there is a tie
13) (7pts) The degree name, degree level, and number of students of the most popular degrees (I.e., the degree program with the highest number of students taking it as major or minor), order by degree name if there is a tie

4. ModifyRecords.sql [10]

This script modifies the following information

1) Change the name of the student with ssn = 144673371 to Scott
2) Change the major of the student with ssn = 144673371 to Computer Science, Master.
3) Delete all registration records that were in “Summer2024”,

5. DropTables.sql [7]

This script deletes all tables, and resulting in an empty database.

Submission Instruction

Submit all your scripts to your Canvas account. The sql scripts must be named exactly how its showing below and all the files must be submitted in one run to trigger grading (if you didn’t finish a certain file, submit it as an empty file):

1) CreateTables.sql,

2) *InsertRecords.sql,*
3) *ModifyRecords.sql,*
4) *DropTables.sql.*
5) *Query1.sql,*
6) *Query2.sql,*
7) *Query3.sql,*
8) *Query4.sql,*
9) *Query5.sql,*
10) *Query6.sql,*
11) *Query7.sql,*
12) *Query8.sql,*
13) *Query9.sql,*
14) *Query10.sql,*
15) *Query11.sql,*
16) *Query12.sql,*
17) *Query13.sql.*

Additional Note: Do not include create database [db] or use [db] commend in any files. Make sure InsertRecords.sql script includes the load data from csv. All table names should be lower case, how its mentioned in this document. You will get immediate feedback after submitting all files. You have unlimited attempts.

We are not grading *InsertRecords.sql* **by running it, rather we are only randomly checking formatting, not necessarily the query corrects, so make sure the query runs in your local. But still its mandate to upload** *InsertRecords.sql* **while submission.**