7CCSMDDW: Coursework 1

Lecturer: Natalia Criado

Coursework assigned: 7 October 2019
Coursework deadline: 4:00 pm, 11 November 2019
Late Submissions deadline (capped at 50%): 4:00 pm, 12 November 2019

Overview

The purpose of this coursework is to create a database for managing the information about the research conducted in a university based on a given ER diagram and a set of requirements. Then, you will have to perform different data manipulation and retrieval operations on the database. This coursework is formally assessed and is worth **10%** of your final mark. You will receive some feedback as part of the marking of the coursework. This coursework is broken down to 10 points distributed as follows:

(1) Create and implement a given design and specification in appropriate SQL table creation queries, including the implementation of keys and domain constraints (2 points)

(2) Insert the appropriate sample data using INSERT queries (1 point)

(3) Create the appropriate views (1 point)

(4) Create and output the appropriate SELECT queries (4 points)

(5) Update data using the appropriate queries (1 point)

(6) Remove data using the appropriate queries (1 point)

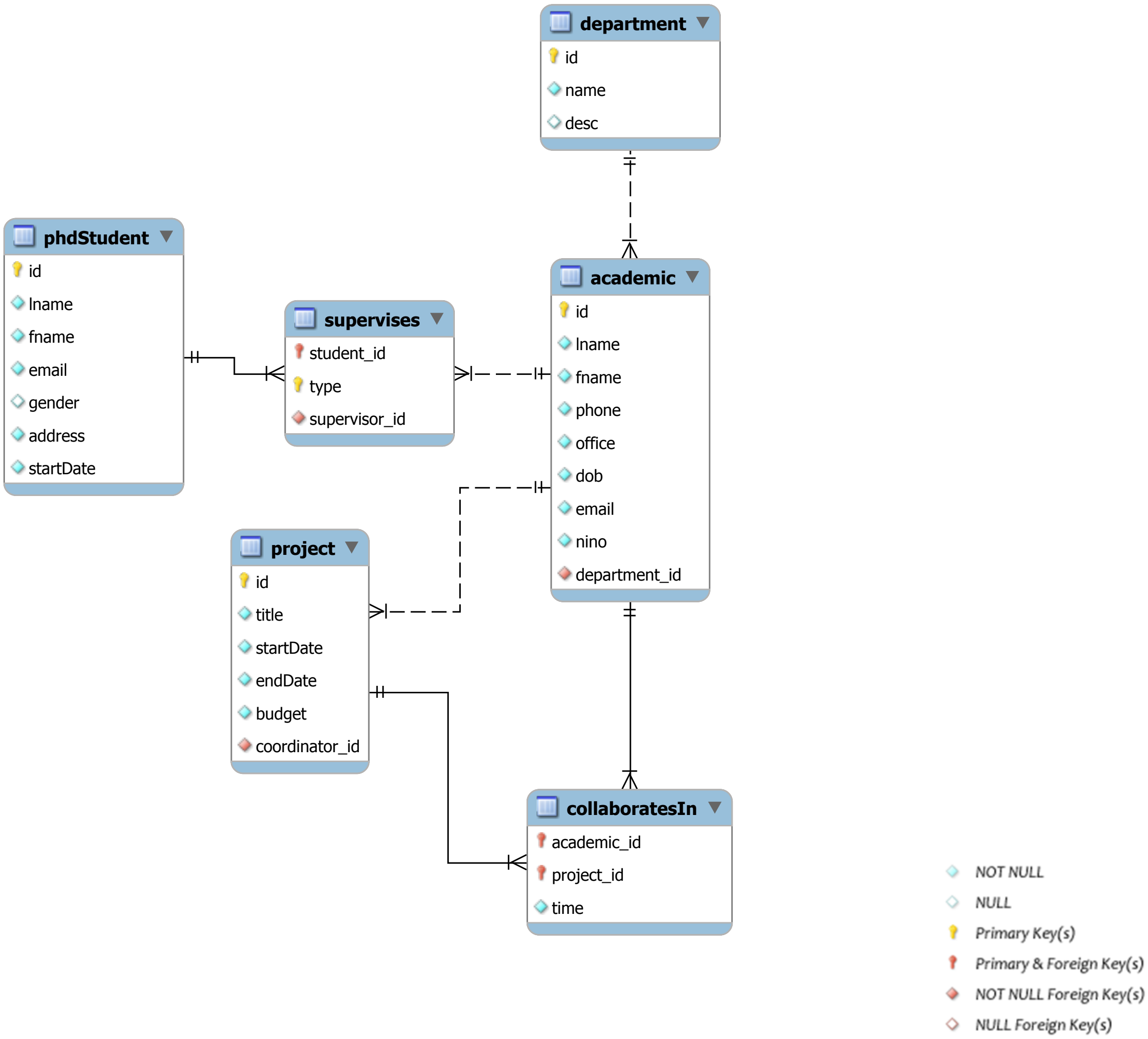
Requirements

A university has decided to create a simple database to register information about research activities. The research conducted in the university is structured in terms of research projects and PhD supervision. Each academic works in one department. Academics can coordinate research projects. In addition, academics can collaborate in research projects by devoting a percentage of their time to work on the project. Academics can also supervise PhD students. In particular, students can have a first and a second supervisor. When a student leaves the university there is a legal requirement to delete all information about this student. When an academic leaves the university, their personal information must be deleted, they no longer collaborate on any research project and other academics must replace them as project coordinators and PhD supervisors.
Here are the specifications for the database requirements:

- **Academic.** Every academic has an id, name, surname, phone, office, date of birth, email, and national insurance number.
- **PhD Student.** Every student has an id, name, surname, email, gender (some students may prefer not to disclose this information), address and a start date.

- **Project.** Each Project has an id, title, a budget, and a start date and end date.
- **Department.** Each department has an id, name and may have a description.

The following ER diagram contains the database model for the research conducted at the university:



Database Implementation and Manipulation Using SQL

Set-up. You will implement the research database in SQL. On KEATS you will find a .zip file which contains template SQL files to edit for your submission.

Be sure to:

1. Include your **NAME** at the top of the file in a SQL comment.
2. Edit these files as **text, not Word or proprietary SQL software**.
3. Include any comments using the SQL line comment (that is, lines beginning with `--`).

Submission and Evaluation. Include your edited versions of these files in the .zip of your submission. We will evaluate these SQL files using NMS's server (which is 5.5.50-MariaDB precisely). As a student you can utilize this server or your own database server installed on your personal computer to test your files and commands.

You can assume that these files are executed in the order that they appear in the description that follows:

1. `schema.sql`
2. `insert.sql`
3. `view.sql`
4. `select.sql`
5. `update.sql`
6. `delete.sql`

Any file that is missing, renamed, or does not run will result in 0 points for that section.

1 Schema Definition. Based on the design and requirements above, write the required SQL DDL (Data Definition Language) statements (i.e. CREATE TABLE) to create the corresponding tables. Ensure that:

- table and attribute names do not conflict with SQL reserved words
- attribute data types are core SQL data types
- your tables implement entity, referential and domain constraints as needed
- every table has a primary key specified as it corresponds to your relational model
- all foreign keys are properly declared

2 Populate Database with data. Time to get creative! Populate your database with some data that you will come up with on your own. Since you only require a small test sample of data, use SQL INSERT statements to populate your database. More precisely:

- Create at least two departments.
- Create at least 3 academics per each department.
- Create at least 10 students. The start date should be earlier than today. There should be at least one student who started last year.
- Include at least 1 supervisor per student.
- Create at least 2 projects per each academic. The end date should be greater than the start date. There should be at least one finished project (with an end date in the past) and one ongoing project (with an end date in the future).
- Include at least 4 collaboratesIn per academic.

Write your INSERT statements in the provided template file: `insert.sql`

3 Creating views. Write SQL statements to:

- Produce a view that returns ongoing projects (i.e., projects that will end in the future) and rejects any attempt to insert or update finished projects.

Write all of these SQL statements in the above order in the provided template file: `view.sql`

4 Querying the Data. Write the SELECT statements that to obtain the following queries:

- 4.1. Research Involvement. The university would like to know which academics are currently involved in research, either as supervisors of PhDs, as coordinators of an ongoing project or working in an ongoing project. List the name and surname of these academics.
- 4.2. Male Supervision Report. The university would like to know how many PhD male students are being first supervised by each supervisor. Include also the academics that are not supervising any male PhD students. 一個老師監督的第一位有幾個是男學生、和完全沒有監督男學生的
- 4.3. Number 1 and 2 by Research Income. The university would like to personally email its top 2 academics by their research income (i.e., the total budget on the projects coordinated by them). Write a SELECT query that lists the academic's name, surname and email. Sort the academics by their research income, showing the top-income academic first.
- 4.4. Most Overloaded Academic. List the information about the academic who is currently collaborating more hours on ongoing research projects. Note there may be more than one most overloaded academic.

Write all of these SELECT statements in the above order in the provided template file: `select.sql`

5 Updating the Data. Write the UPDATE (and possibly other) statements to perform the following operations:

- It is possible that some PhD students have a second supervisor but not a first supervisor. Create queries (it is possible that you may need to use something more than just update statements) to avoid this problem by making that the second supervisor becomes the first supervisor when there is no first supervisor.

Write these statements in the provided template file: `update.sql`

6 Removing Data from the Database. Write the DELETE statements to perform the following operation:

- PhD students without a second supervisor have decided to leave the university. Write the DELETE statement(s) that removes these students from the database. To avoid any future embarrassment in case of a data leak, make sure you remove all trace of these students from the database.

Write all of these DELETE statements in the provided template file: `delete.sql`.