MIE353 Assignment #2

Analyzing The Course Management System

October 2024

Preliminaries

In this assignment, you will be responsible for modifying a database and its supporting Java-based software to extend the data model for a course management system of a University. All the code you need, as well as the required dependencies and libraries, is included in the github assignment repository. To ensure everyone has a consistent environment for running and modifying their code, the code has been provided alongside its own development container [1]. You will need to ensure you have Docker [2] and Visual Studio Code [3] installed to open and run the code. The Java code and the database have been connected using Hibernate/JPA [4]. When opening the project in VS Code, ensure you open it in a dev container and have docker running.

Submission

All files including your modified code should be submitted via the github repository you created from the github classroom link. Ensure you have filled out the form for your github username so we can associate your code to your student id. Files can be either committed through git or via drag-and-drop file upload through github. The starter repository for github will also have a submit folder. Any written answers should be combined into a single .pdf and placed into the submit file. Additionally, any data files should also be uploaded as .csv files and uploaded into the submit folder.

Grading

Full marks will be given for (1) working, readable, reasonably efficient, documented, and original* code that achieves the assignment goals, (2) for providing appropriate answers to

[1] To learn more about devcontainers, please see `https://code.visualstudio.com/docs/devcontainers/containers`

[2] `https://www.docker.com/`

[3] `https://code.visualstudio.com/`

[4] `https://www.baeldung.com/learn-jpa-hibernate`

the written questions in your .pdf write-up, (3) for correct and correctly-formatted data results provided in your .csv files.

* Any multi-line code snippets from external sources **must** be cited in a comment above the code indicating where you obtained it from. This is permissible for boilerplate or minor functionality but **not** acceptable for significant portions of your implementations.

1 Exploring The Course Management System (CMS)

This first question will familiarize you with the use of Java for connecting to data sources using JDBC and Hibernate/JPA. You are not required to modify or add any code in order to answer this first set of questions.

Q1. How many classes are there in the data model of `CMS`?

Q2. What is the relationship between the following classes: `Person`, `Student`, `Professor`?

Q3. Open the `CrsSystem.mdb` database and examine it. How many relations are there in the database?

Q4. How are these three classes (`Person`, `Student`, `Professor`) translated into into the database `database.mdb`?

Q5. From this translation, can you determine whether Professors and Students satisfy any disjoint and/or covering conditions?

Q6. What format is the database stored in?

Q7. What is the purpose of the `hibernate.cfg.xml` file?

Q8. What functionalities of the `lombok` library are being used for CMS?

2 Fixing Software Problems

The `CMS` designer missed an important piece of implementation when writing the Java code: they did not store information about the `Teaches` table in the database. **Your Task:** Add this information into the database by modifying the existing model class(es) as necessary.

3 Extending the Data Model

The University wishes to modify CMS to incorporate information about teaching assistants. **Your Task:** Modify CMS by adding a new class, TeachingAssistant.java, according to the following constraints given to you by the University administrator:

- All teaching assistants are students.
- All teaching assistants assist at least one course.

Additionally, you should also add functionality to CMS to obtain a list of all Teaching Assistants to be stored in memory as part of the `main` function (similarly to the existing functionality for students, courses, professors, and transcript). You should populate the teaching assistant table with at least 4 individuals for testing purposes. Populating this new table must be done using SQL, as functions in the `DataUtil.java` file of CMS, and called through the `main` function for testing purposes.

4 Writing Queries

The university has asked you to come up with statistic about CMS. To do so, you should write SQL queries to solve the following problems:

1. What was the highest grade received in mie302h1s?
2. What are all the names of first-year courses? (first year courses have a number code beginning with 1 after their department code)
3. What are the full names of professors only teaching a single Course?
4. What are the full names of all students that have taken at least one course taught by Patrick Ward?
5. Which students (full names) have never taken a course taught by Kelsey Lewis?
6. Is Jason Martinez's average across all courses they have taken higher than that of all the other students who took those same courses?
7. List all the students (full names) who have taken the "Hum.Cent.Syst.Design" course and failed at least one *other* course. (A failing grade is 49 or below).

The SQL queries you write should be implemented as functions in the `DataUtil.java` file of CMS, and called through the `main` function for testing purposes. Additionally, each query result should be saved into a .csv containg the resulting header and rows, following the naming convention: `query1.csv, query2.csv, ... queryX.csv`