Database System and Applications

Project Report

Student ID: 2202130025 Name: Abderrahim Talibi

Introduction

We're all aware that professors face challenges in managing student grades, involving tasks such as adding, deleting, and updating student information. This process is not only time-consuming but also presents difficulties in tracking and searching for specific details, leading to an increased likelihood of errors. To address this issue, we suggest developing software that streamlines these tasks for professors.

In more detail, our proposed mini application involves each professor having a dedicated account. Professors will be associated with a specific number of classes, each containing a certain number of students. Students, in turn, may belong to multiple classes, and each student will have a grade assigned in a particular class.

Upon launching the application, professors will encounter an interface prompting them to either log in or sign up if they don't have an account. If the professor lacks an account, they will proceed to create one; otherwise, they will enter their account information. Following this procedure, the application will guide the professor to another interface designed to effortlessly manage and display the classes they are responsible for.

**E/R** **Diagram**

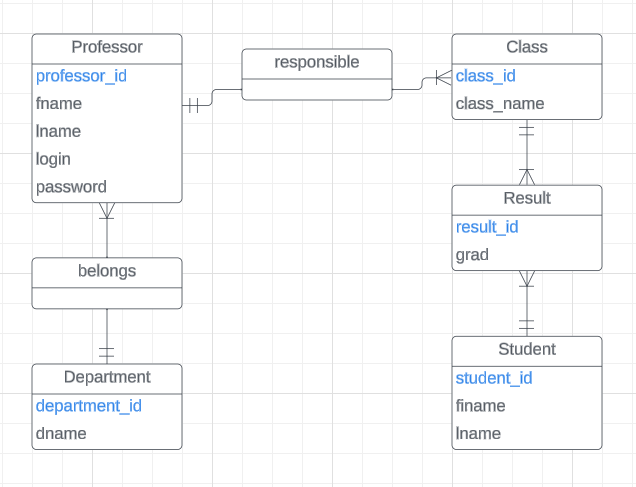


Figure 1 - E/R diagram for student’s management system

The diagram presented depicts the connections among various entities within our system:

* The relationship between Department and Professor is one-to-many, indicating that each department accommodates multiple professors, and each professor is affiliated with only one department.
* The connection between Professor and Class is one-to-many, signifying that a professor can oversee multiple classes, and each class is overseen by only one professor.
* The association between Class, Result, and Student is one-to-many, implying that each student has one result for each class.

**Relational** **Schema**

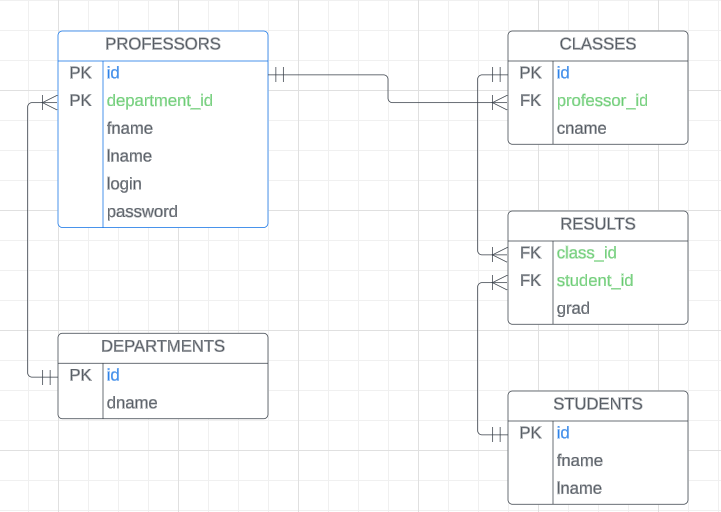
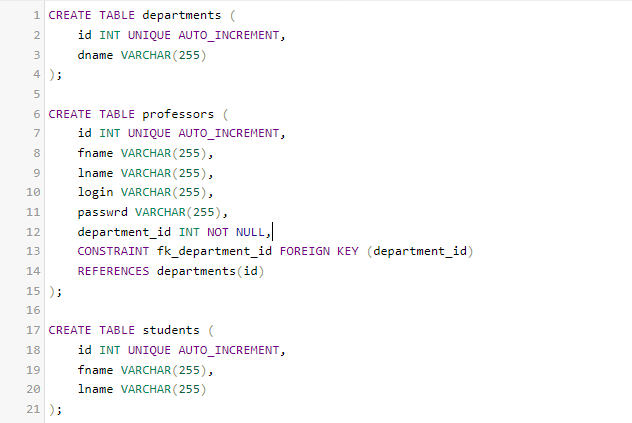


Figure 2 - Relation Schema for student’s management system

At each relation we can easily verify that 1NF, 2NF, and 3NF is verified.

Consider the PROFESSORS relation as an illustration, where each attribute exhibits functional dependencies with the primary key, ensuring atomicity (1NF). Moreover, all attributes are elementary, dependent on the entirety of the key (2NF). Lastly, all attributes are directly dependent on the key, indicating that their determination is not through transitivity.

**Creating** **Database** **(SQL)**



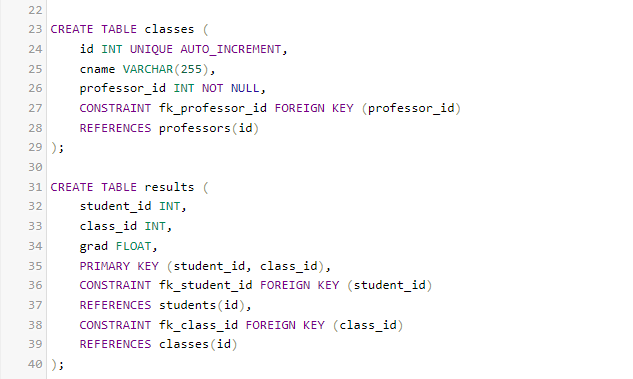


Figure 3 - SQL query to create tables

Results in the interface of phpMyAdmin:

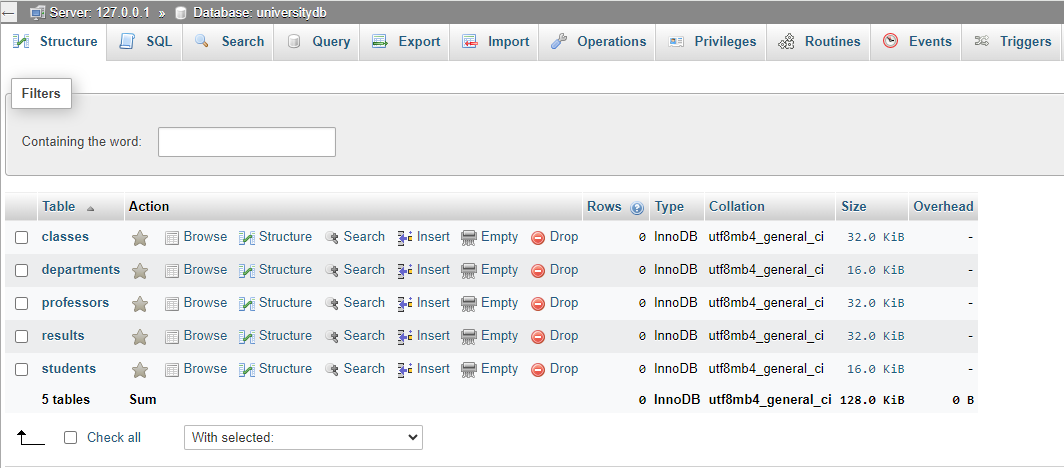


Figure 4 - phpMyAdmin database interface

**User** **Interface**

We the user interface using swing API (Application Programming Interface) the code implemented using the Java language.

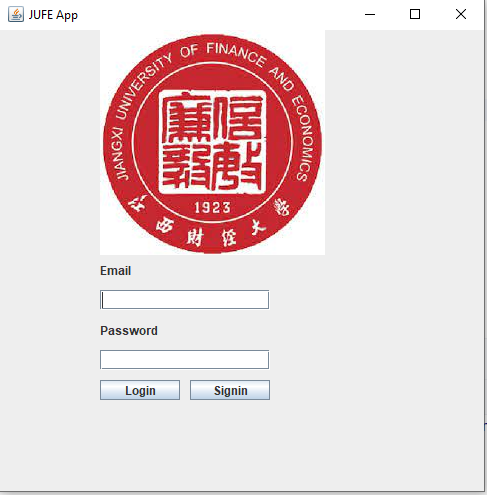


Figure 5 - Professor’s login window

We have in the frame two button the first one for the login if the professors have an account the app will pass him to the business interface for manipulating students’ data.

If the professor does not have an account, he can create one by clicking on the button signing.

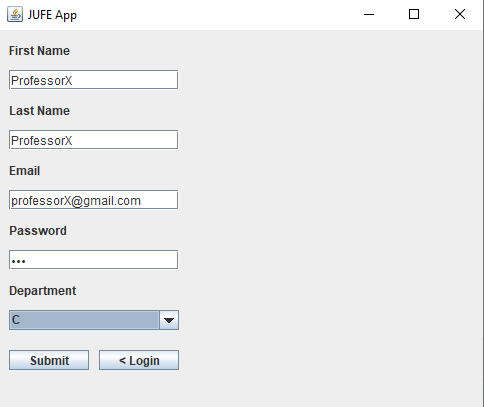


Figure 6 - Creating new account

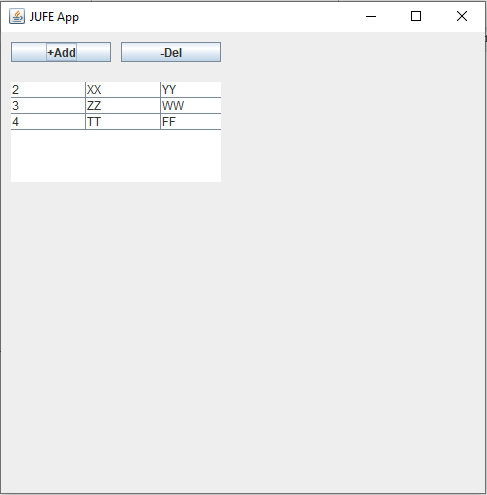


Figure 7 - Student manipulation

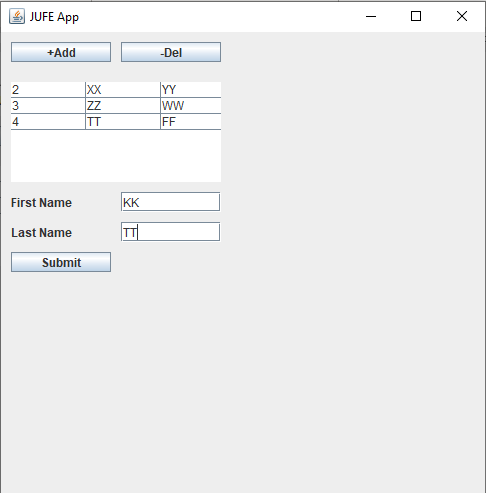


Figure 8 - Adding student to the database

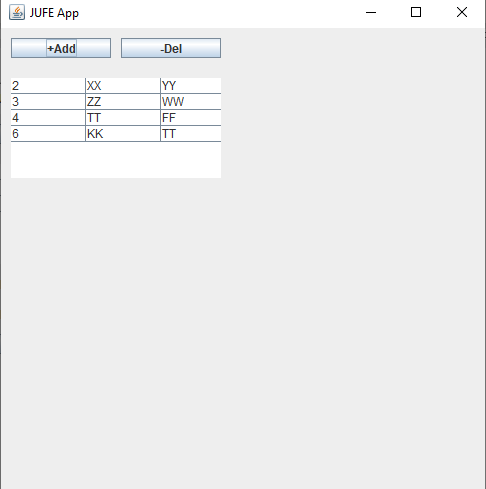


Figure 9 - Result of adding student

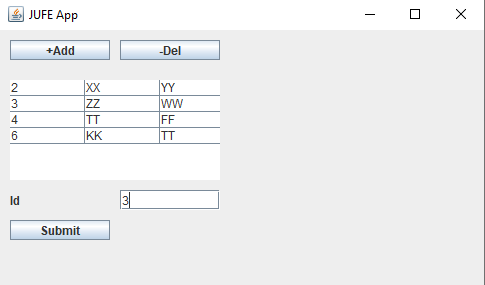


Figure 10 - Deleting Student

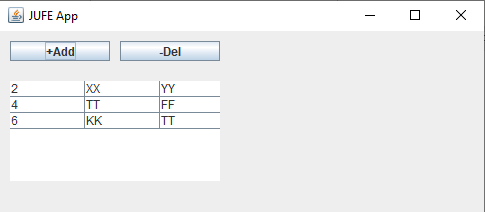


Figure 11 - Result of deleting student



Figure 12 - Professor table

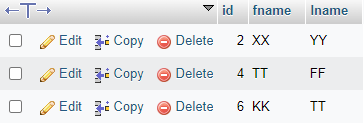


Figure 13 - Student table