Student Record-Keeping Database Design Documentation

Group #12: Chen Jin, Naimo Yasin, Anthony Siprak, Bassil Ahmad

Revision History

Date	Version	Description	Author
2021/11/27	1.0	Initial Design, 7 relations (Student, Courses, Academic Standing, major/minor, department, faculty, course info).	Group 12
2021/11/29	1.1	Resolved issues in version 1.0. Simplified relations, improved key and relation design. Added Scholarships table. (Student, Course_Attendance, Courses, Programs, Faculty, Scholarships)	Group 12
2021/11/30	1.2	Implemented past and current courses, and bridges. (Current_Courses, Past_Courses, Courses, Student, Faculty, Majors, Minors, Student_Major, Student_Minor, Scholarships, Bridge_table)	Group 12
2021/12	1.3 (Current)	Simplified and restructured relations, removed bridges. (Current_Courses, Past_Courses, Student, Scholarships, Program, Courses, Faculty)	Group 12

Table of Contents

1. Introduction

2. Detailed Database design

2.1. Database design (Relational Database)

2.1.1. Conceptual diagram

2.1.2. Description

2.1.3. Purpose of Tables

2.1.3.1. Student

2.1.3.2. Courses

2.1.3.3. Current_Courses

2.1.3.4. Past_Courses

2.1.3.5. Faculty

2.1.3.6. Program

2.1.3.7. Scholarships

3. References

1. Introduction

The student record-keeping database management system is a system that is mainly designed to store and maintain student information including but not limited to personal information, contact information and academic information such as student gpa and their program of study.

The data in the database is stored in MySQL server 8.0. Dummy data was used during the implementation of the database. Data can be accessed through the terminal. This design is to allow easy data entry and access through querying.

The main objective of the database design was to have an efficient, normalized database that would allow easy data entry and access. The entity relational diagram, the relational schema, the physical data with the data types used in the relations are also included in this document.

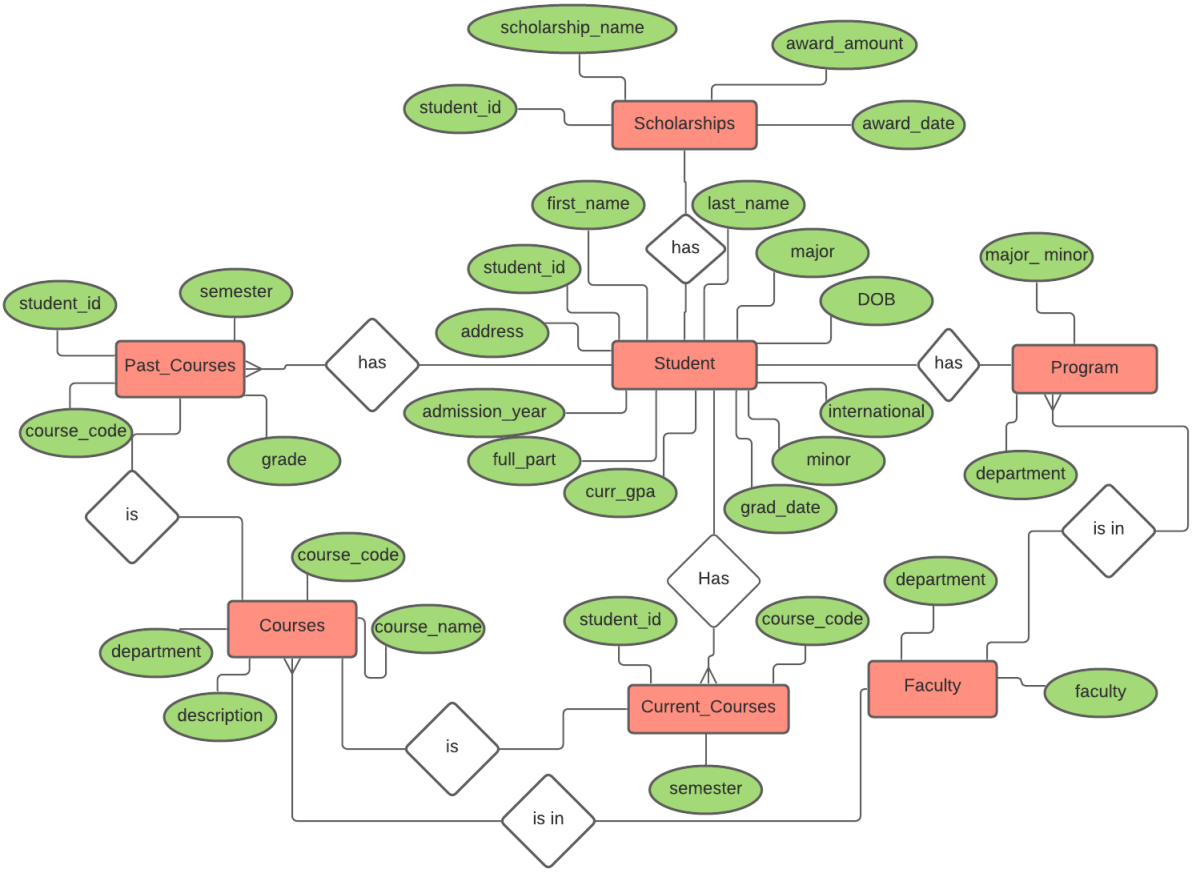
2. Detailed Database Design

At the physical level, this database is stored in a MySQL server 8.0. The relations in the structure are defined using primary keys, composite keys, and foreign keys, and the attributes used in the databases each are stored using a specified data type. MySQL systems use both trees and hash tables to store and retrieve data in databases.

The conceptual makeup of this database revolves around the student table, defined by student_id. Most tables are directly related to this table and those that aren't are just there to provide necessary function to tables that are.

At the logical level, there are seven total tables. There is the student table, holding information dependent solely on student_id. There are also tables storing different things students can have, like scholarships, programs, current courses, and past courses. For all of these, there is a many-to-many relationship to student_id, as there are multiple students and each student can have more than one of these entities. There also exists the faculty and courses table to hold necessary data on what courses are available and what faculties programs and courses can belong to.

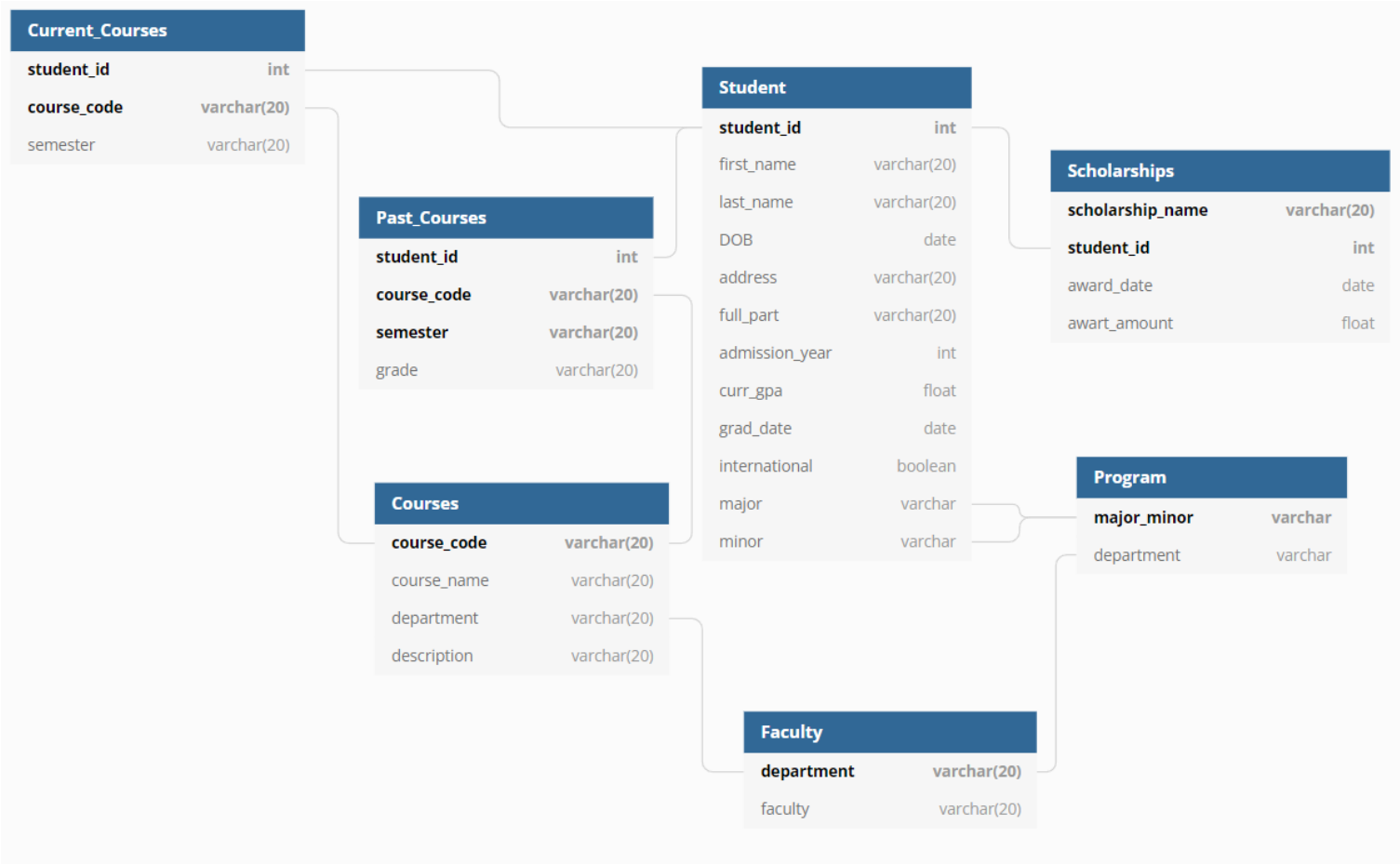
Meanwhile at the external level, all interaction with this database is handled via the MySQL text terminal. There is no implemented GUI, so all information is retrieved using different types of queries.



The above diagram shows the relationships of the entity sets stored in the database as well as the attributes associated with each entity. For more information about the purpose of each relation, please refer to section 2.1.3.

2.1 Database design (Relational Database)

2.1.1 Conceptual Diagram



2.1.2 Description

The purpose of this database was to create an effective method of storing and retrieving student information, with a significant amount of relational info to correlate the different tables. This allows for less redundancy, and faster retrieval of data.

2.1.3 Purpose of Tables

2.1.3.1 Student

Stores individual student data, including private information and some academic information such as the student gpa. Primary key is **student_id**, and foreign keys are **major** and **minor**.

2.1.3.2 Courses

Stores data about all courses including course name, department, and description that can be referenced. Course_code is the **primary key** and **department** is the foreign key referencing **Faculty(department)**.

2.1.3.3 Current_Courses

Stores data about courses that students are currently taking. **Student_id** and **course_code** combine to make a composite primary key. Individually, they are both foreign keys referencing **Student(student_id)** and **Courses(course_id)**.

2.1.3.4 Past_Courses

Stores data about past courses that students have taken. **Student_id**, **course_code**, and **semester** combine to create a composite primary key. **Student_id** and **course_code** are also foreign keys referencing **Student(student_id)** and **Courses(course_id)**.

2.1.3.5 Faculty

Stores data about each faculty and which departments belong to which faculties. **Department** is the primary key.

2.1.3.6 Program

Stores data regarding majors/minors and which department they belong to. **Major_minor** is the primary key and **department** is the foreign key referencing **Faculty(department)**.

2.1.3.7 Scholarships

Stores scholarship info including which students have scholarships. **Student_id** and **scholarship_name** make the composite primary key. **Student_id** is also the foreign key referencing **Student(student_id).**

3. References

Garcia-Molina, H., Ullman, J. D., & Widom, J. (2014). Database systems: The complete book. Pearson.

Gao, S. (2021). Relational Model [Slides]. MyLearningSpace.

https://mylearningspace.wlu.ca/d2l/le/content/399848/viewContent/2657741/View