Joshua A. Barrs | 2294736

Brandon Kleinman | 2291703

Final Report: Assignments Database

1. Problem To Address

One of the most indelible memories of my time as a student occurred when I happened to glance over at the Google Calendar “Month” view of one of my peers. The page was covered with tiles marking the due date of an assignment or the date of a special event. Indeed, Google Calendar, Outlook, or other online calendars have played a very important role in helping students organize their very-busy schedules. Even with such tools like these, oftentimes it becomes easy to lose track of an assignment or an important event, only to realize we forgot about it after the deadline has passed. In order to address this situation, we created a simple yet relevant database application that allows students or school administrators to keep track of the projects or assignments currently issued by faculty in order to best assist the student body in reminding them of the important tasks they must complete by a certain deadline.

1. Related Applications/Work in Area

In addition to the applications previously mentioned, other assignment trackers are also in place. There are several assignment tracker applications that can be installed on a mobile device or on the computer that assist with keeping track of currently-issued assignments and projects. Some of these applications are free of cost; others require a fee. myStudyLife, myHomework, iHomework, myHomeworkStudentPlanner, and Google Socratic are a few of the current planner applications available for download currently.1 However, as we will explain in greater detail, our application serves as both a student planner and an administrative tool, in that administrative staff can also use the application to track current assignments broken down by Course, Faculty, School, and/or Department.

1. Elements of Our Solution (Application Design)

In order to address the problems described above, we created a database application that keeps track of current assignments/projects/tasks. Our application stores assignments in a CourseAssignmentDB database, which contains five different tables responsible for storing information of its respective type: Assignments, Courses, Faculty, Department, and universitySchools (to keep track of schools/colleges within the particular university utilizing this application). Each of these tables maintains a relationship with the others via a foreign key.

Once an assignment has been assigned, the student/administrator can enter the assignment into the Assignments table, which requires the name of the assignment, the assignment’s due date, and the course to which this assignment corresponds (the assigning course). The assignmentID, serving as a primary key to uniquely identify each assignment in the Assignments table, is auto-generated/auto-incremented whenever an assignment is inputted into the table. The course to which the assignment corresponds is referenced by a CourseID, which serves as a foreign key linking the Assignments table to the Courses table. (In order to assign an assignment to a course, that course must already be in the Courses table.)

In inputting an entry into the Courses table, the following attributes are required by the user: the name of the course (CourseName), the course code (as in “CPSC 408”), the department under which this particular course falls (such as “Computer Science”), and the FacultyID of the course’s instructor. The course ID serves as the primary key to uniquely identify each record in the Courses table, but is also auto-generated/auto-incremented whenever a new course is added to the table. The FacultyID of the course’s instructor (or CourseInstructorID) serves as a foreign key linking the Courses table to the Faculty table.

Within the Faculty table, the following attributes are required in inputting a new faculty member: name of the faculty member, rank of the faculty member, whether or not the faculty member is tenured, the education level of the faculty member, and the department to which the faculty member is assigned. The primary key FacultyID is auto-generated/auto-incremented for each new faculty member entry. The department to which the faculty member is assigned (DepartmentID) serves as a foreign key linking the Faculty table to the Department table.

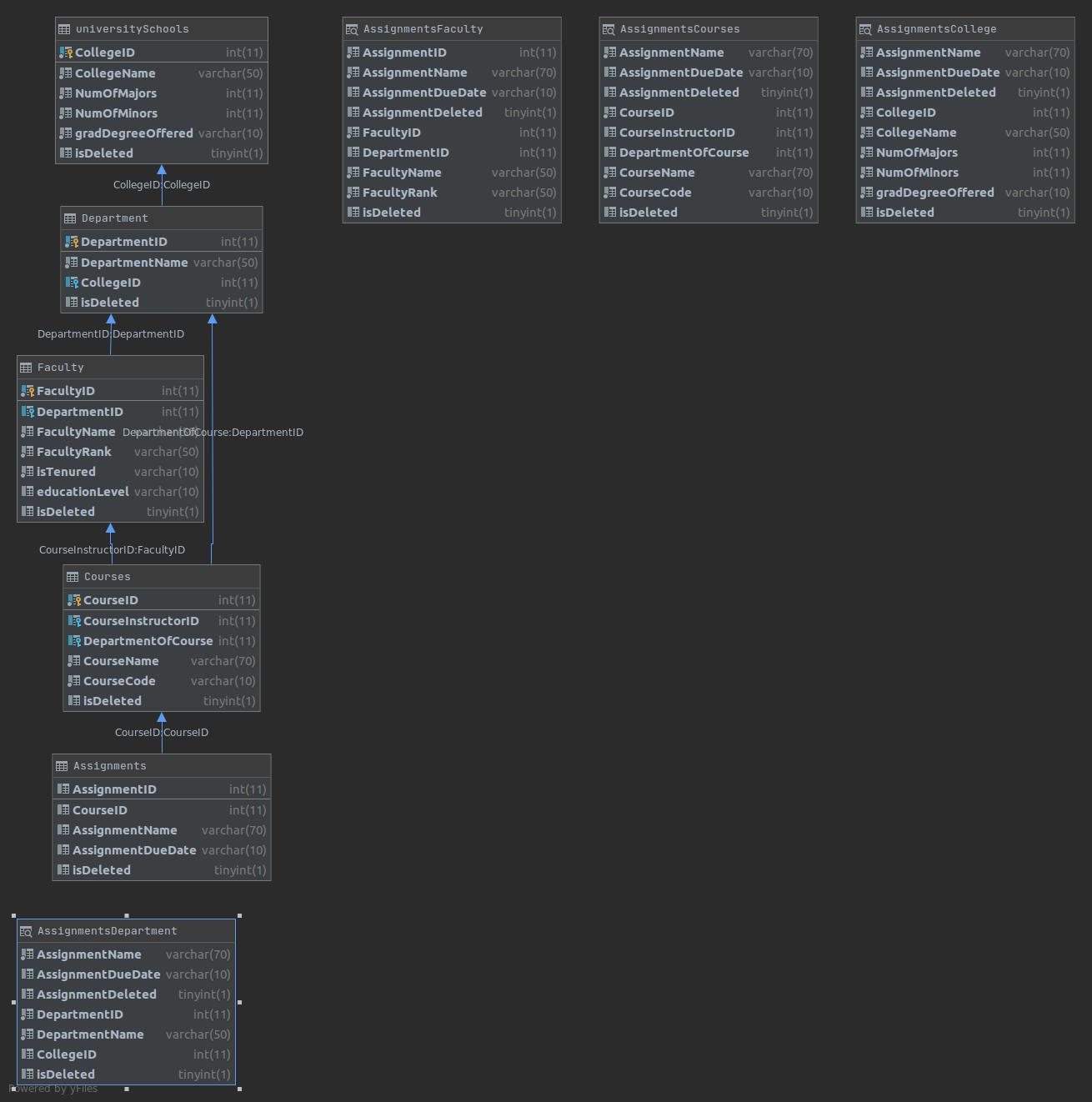
Within the Department table, the following attributes are required in inputting a new department: the department name and the school/college under which this department falls (such as “Fowler School of Engineering”). The primary key DepartmentID is auto-generated/auto-incremented for each new department entry. The school/college under which the department falls (CollegeID) serves as a foreign key linking the Department table to the universitySchools table.

Within the universitySchools table, the following attributes are required in inputting a new school or college: the college ID, the name of the school or college, the number of majors in this school/college, the number of minors in this school/college, and whether or not a graduate degree is offered in this school/college (a boolean value). The CollegeID primary key is auto-generated/auto-incremented for each new school/college entry. The universitySchools table has no foreign key in that it does not need to reference any other table in the CourseAssignmentDB database.

1. Results

The user will have the ability to make the following changes to the database: adding new records to tables, deleting records from tables (this will be a soft delete, so the record will still be in the table but won’t be displayed), creating new tables with specified information, undoing any previous changes, and committing all changes that have been made. After the user has completed and committed all desired actions in the database, there are two main ways that users can see results – displaying records from tables in the terminal, and generating CSV reports of tables in the database to be exported to the user’s computer. Within the former option, the user can either display entire tables, or display records from the tables after a filter has been applied. For example, the assignments can be searched by faculty member, by department, by course, or by the whole school or college. In addition, the user can also query the respective table data individually (i.e. see a list of all faculty members). Overall, we have created a relatively straightforward yet highly beneficial database application to assist administrators and students in managing their understandably-busy schedules.

1. Application Schema Diagrams



Works Cited

1. <https://justdomyhomework.com/blog/homework-does-not-have-to-be-hard>