

Phase 2 Team 13 Members: Michael Yoo, Warren Nguyen

Prof. James Taylor

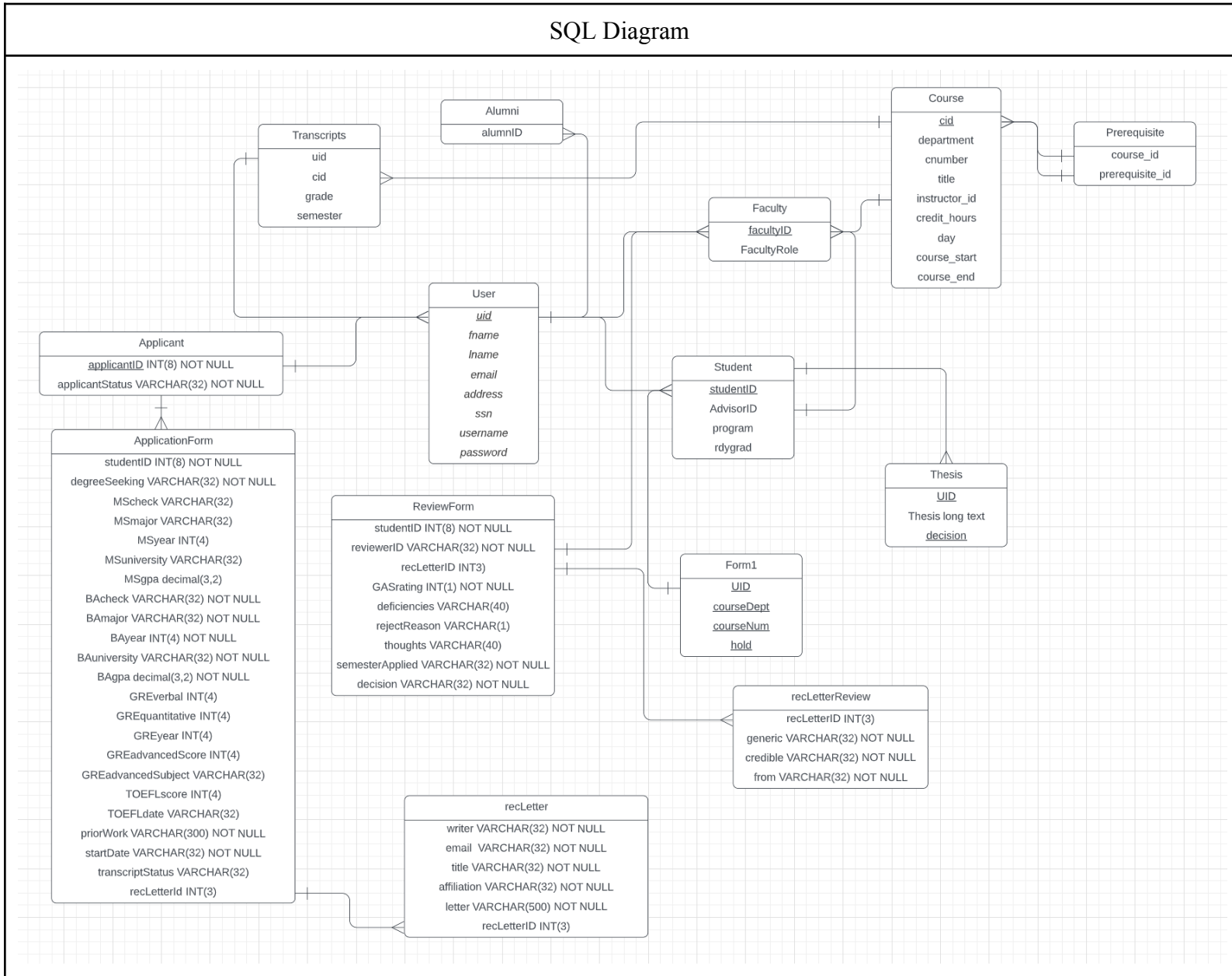
CSCI 2451W

5 May 2023

Registration System: Phase 1 Report

DB Design.....	2
Visual Overview.....	5
Design Justification.....	6
Special Features.....	8
Work Breakdown.....	9

DB Design



The ARGs System's Tables meet the following forms:

1. User
 - a. The User table satisfies the requirements of 3NF because there are no transitive dependencies between non-key columns. This means that all non-key columns are dependent only on the primary key and not on any other non-key columns.

2. faculty
 - a. The faculty table satisfies the requirements of 3NF because there are no transitive dependencies between non-key columns. There are no other columns in the table, so there are no transitive dependencies to consider.
3. Student
 - a. The Student table satisfies the requirements of 3NF because there are no transitive dependencies between non-key columns. The advisorid column is a foreign key and it is not a part of the primary key. However, it is not functionally dependent on any other column in this table, and therefore, it does not create any transitive dependency.
4. Alumni
 - a. 3NF: same reason as faculty
5. Applicant
 - a. 3NF: same reason as faculty
6. Course
 - a. The Course table also satisfies the requirements of 3NF because there are no transitive dependencies between non-key columns. All non-key columns are directly dependent on the primary key (cid).
7. Preq
 - a. The Preq table satisfies the requirements of 3NF because there are no transitive dependencies between non-key columns. There is only one non-key column in the table, and it is directly dependent on both courseID and preqID columns.
8. Transcript
 - a. 3NF: Same reason as Course
9. Form1
 - a. The Form1 table also satisfies the requirements of 2NF because it has a composite primary key (id, courseDept, courseNum), and all non-key columns are dependent on the primary key. The id, courseDept, and courseNum columns together uniquely identify each row, and the holding column is dependent on the primary key. However, it does not satisfy the requirements of 3NF because the holding column is not functionally dependent on the entire primary key. It is only

dependent on the id column, which means there is a transitive dependency between the id column and the holding column.

10. Thesis

- a. 2NF: Similar reason as Form1

11. ApplicationForm

- a. This ApplicationForm table is in 1st normal form (1NF) because all the columns contain atomic values and there are no repeating groups or arrays.

12. ReviewForm

- a. 1NF: Similar reason as ApplicationForm

Overall, the schema is in at least 3NF (third normal form) because the it meets the requirements below:

First Normal Form (1NF): Every column in a table should be atomic, meaning that it should contain only one value per cell. Looking at the schema, each column in every table appears to contain atomic values.

Second Normal Form (2NF): The table should be in 1NF, and every non-key column in the table should be fully dependent on the primary key. From the schema, all tables have primary keys, and every non-key column in the table seems to be dependent on the primary key.

Third Normal Form (3NF): The table should be in 2NF, and there should be no transitive dependencies in the table. Based on the schema, all tables seem to be in 3NF since there are no transitive dependencies between columns.

Visual Overview

Videos are in the Visual Overview folder in the repository

- Check Applicant.mp4 for Applicant Video
- Check Faculty_Reviewer.mp4 for Faculty Reviewer Video
- Check Chair.mp4 for Chair Video
- Check Registration.mp4 for form 1 demonstration, registration demonstration, and grading demonstration
- Check Graduation.mp4 for graduation demonstration and thesis demonstration
- Check Faculty.mp4 for remaining demonstration of Graduate Secretary, Instructor/Advisor, and Admin roles

Design Justification

APP- Maya

For the integration of the APP system with the REG system, it was essential to admit a student when their application status became accepted. This status can be updated by the graduate secretary or chair. When the application status becomes accepted, we need to remove the applicant from the applicant table and insert their applicant id into the users table. The applicant id is created when a new user is created from the homepage. We include checks to see if that id already exists so there won't be multiple of the same user. The default state of a new user is going to contain the applicant id they submit and application status automatically is set to "missing information". This allows the workflow to begin and the process of an applicant to be accepted and hopefully become an alumni one day.

REG- Warren

For the integration of the REG system, there was an overlap within the transcript, courses, and student table. The REG system is connected to the APP system and ADV system through the students table where a new student can register. The form1 will limit their registration ability which is described in further detail in ADS portion. For transcripts, we combined aspects of the advising schema(transcript) and the regs schema(enrollment) into 1 table. We then were able to update grades in the transcript whenever an instructor or higher level faculty updates the grade of a class that a student has taken or is in progress. The overlap in courses can be seen in our new courses and preq tables which is a combination of the ADS schema(courses) and REGs schema(courses, preq). The functionality of the 2 tables worked fluidly with no issues while maintaining the highest level of normal form possible.

ADS- Michael

The integration of advising into the project was a challenging task. First, our team needed to understand where the overlap occurs for the ADS and REGs systems. We saw that there was mainly an overlap within the student and form1. The students table had an overlap where the ADS schema(students(program and rdygrad)) did not exist in the REGs schema(student). We added those 2 fields which allowed for the students graduation status to be visible to the graduate secretary where students can then be graduated. When a student has graduated, they are deleted from the students table and inserted into the alumni table. The ADS schema(transcript) was linked to the studentid of a student but we had to change this to the userid in user which allowed for the ability to always view a user's transcript. Finally, we integrated the use of form1 to allow students to have a registration hold, which stops students from registering for classes by not having access to the registration link. The form1 registration hold can only be lifted once the student's faculty advisor or higher faculty approves their form1.

Assumptions

- Each individual in faculty has one role
- All advisors are/can be instructors, but not all instructors are advisors
- Registration and Grading occurs for the same semester: Spring 2023
- Applicant status is auto set to “missing info”
- New instructors that are created by the admin do not advise anybody until the gsec or admin assigns them
- A new form1 of a student is automatically set to a registration hold
- A student must write a thesis (unless ringo starr)
- Registration only opens once form1 has been approved
- Student must be advised for them to register

Special Features

- GS, instructors, and administrators all have uniform advisor portals
- Instructors and students have uniform grading/registration portals
- Instructors have grading and advisor portals and can switch between them to demonstrate different roles
- Administrator can create all types of users

Work Breakdown

Michael's Contributions:

- Creating the database, populating tables, and setting up primary and foreign keys
 - At first with issues but later resolved as a team
- Implemented the reset sql script
- Combined aspects of group members log in function from phase 1
- Updated Implemented all graduate secretary student related routes (seen in student advising portal)
- Updated Implemented all graduate secretary advising related routes (seen in graduate secretary advising portal)
- Updated Implemented all instructor advising related routes (seen in instructor advising portal)
- Updated Implemented all admin advising related routes (seen in admin advising portal)
- Implemented the creation of a new user by the admin
- Implemented the creation of a new user through the register in the homepage
- Implemented the logic and functionality of admitting a new student through the chair/cac and graduate secretary roles
- Added the idea of an advising hold restricting users from registering for classes
- Styling of some advising specific pages

Warren's Contributions

- Implemented the registration portal for students
 - Catalog
 - Schedule
 - Registration: Adding courses and error checking scheduling issues and prerequisites
- Implemented the grading portal for graduate secretaries and teachers
 - List of courses available for grading
 - List of students in selected course to grade
- Organized the overall routing of the html files to show a cohesive integration of the project sections
- Assisted in transcript calculation of GPA
- Designed the HTML and CSS:
 - The registration portal for students and all associated pages
 - The grading portal for graduate secretary and instructor and all associated pages
 - The advising portal of students and all associated pages
 - The advising portal of graduate secretary and instructor
 - The login page
 - The main page
 - The profile pages of various user types