User Manual and Design Manual for University Database Application

CS 460 Final Project

Matthew Bell
Matthew Huestis
Alif Jakir
Conor Miller-Lynch

Table of Contents

User Manual

Introduction

This application allows users to access information in a university database. Administrator, professor, and student users have different permission levels that allow them to access relevant information.

Logging In

When the application starts, fields will appear for entering a username and password. Enter your username and password and click “Log In”. If you entered the login information correctly, you will be directed to the appropriate page for your user type. Click “Log Out” at any time to return to the login screen.

User Groups

- I.T. Administrator
    - Access User Account data, and add more users. Logs in using the Django admin page.
- University Administration
    - Roster: View a list of professors and sort them(1) by name (2) by department, (3) by salary.
      Additionally, view a list of departments and classes.
    - Salary: View salary of each professor and average salary of each department.
    - Performance: Given a professor's name, an academic year, and a semester, view the following for the professor: the number of sections taught during the semester, the number of students taught, the total dollar amount of funding through grants, and the list of publications for which the professor is an author.
- Professor
    - View the list of course sections and the number of students enrolled in each section that the professor taught in a given semester
    - View the list of students enrolled in a course section taught by the professor in a given semester.
- Student
    - View the list of sections offered by a department in a given year and semester.

Design Manual

Design Overview

The database stores data that might be of interest to administrators, instructors, and students at a university. The primary entities are **department**, **instructor**, **student**, **course**, **paper**, **fund_grants**. The **section** table describes an offering of a course in a specific year and semester. Many-to-many relationships between entities are implemented with the tables **takes** and **teaches**.

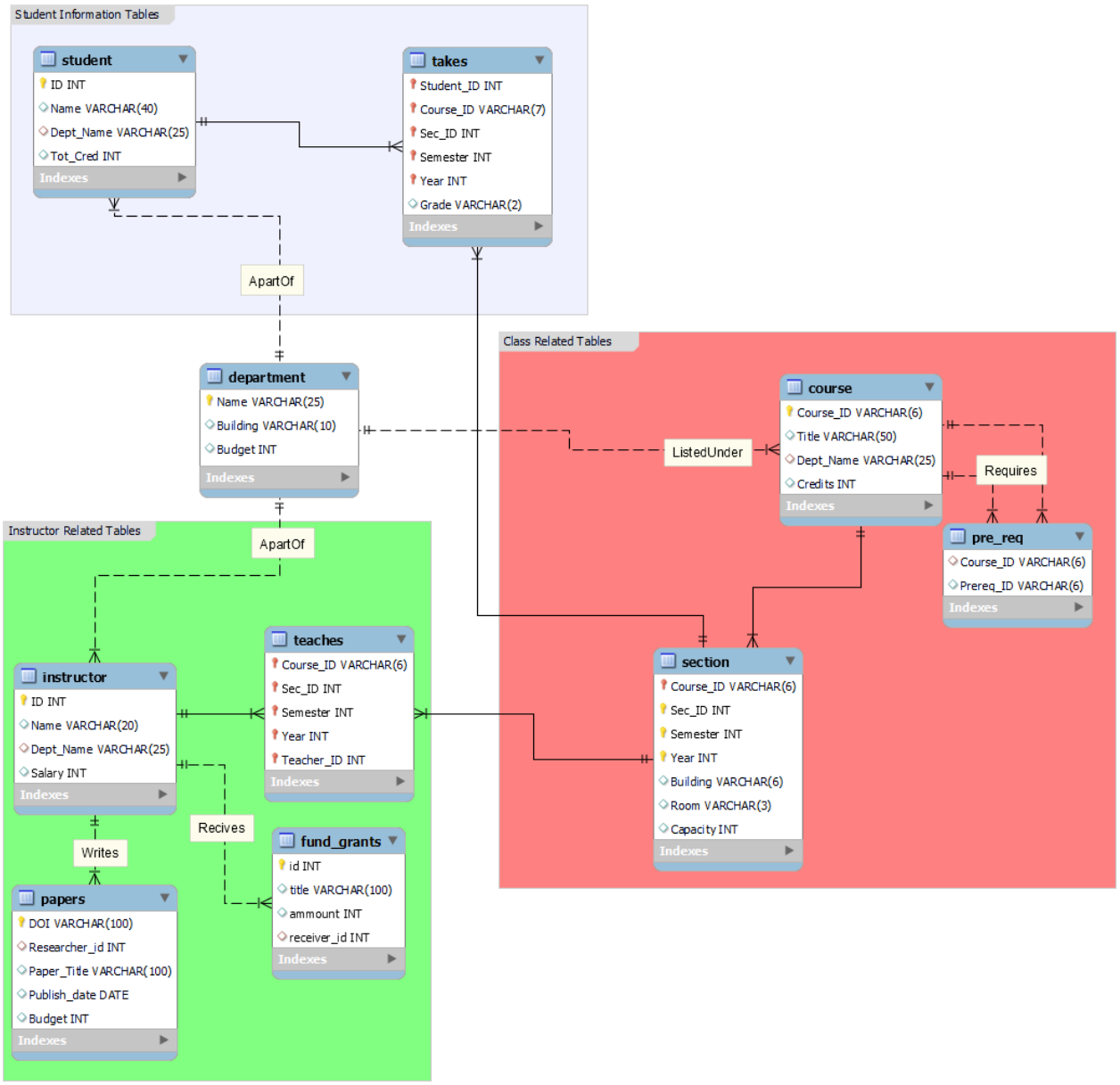
Information on each individual Instructor is stored on the **instructor** table. The Teaches table records what professor taught what class and section during each year and semester. Table **fund_grant** records the grants given to each instructor. **fund_grant** was used as a name due to SQL keyword conflicts. The **papers** table records information on publications.

The table **course** records information shared between all instances of a course at the university. The **pre_req** table records the prerequisites of courses that have them. The table **section** records the section number of the class along with the year and semester of the course.

The Table **student** records the information for each individual student. The table **takes** records the section and grade of the class a student has taken.

Tables that are used by Django to validate users and their information should be accessed through Django's administration system.

Schema / ER Diagram



Security Measures

If a user is not logged in or attempts to access a page they are not authorized to access, the server will redirect the user to the login screen or the user's home screen. This keeps forms and data private only to accessible users. Most queries are made using Django's query system.

User input for queried data is either done through drop down select boxes and user text input boxes. The drop down inputs are constrained for pre selected values and should not be a security concern. Text inputs for professor performance and other pages are designed to only query specific tables and specific columns. Additionally the views that display and query the database are designed to trip an error, and display no data if tripped. It will instead display an error message.