# University Database Management System

**CS315 Project Report**
Indian Institute of Technology, Kanpur

**Akarsh Goyal** 160063

**Bharat Jindal** 160192

**Siddhant Sarkar** 160682

**Simran Sundrani** 160691

**Yash Choudhary** 160819

## Abstract

Our objective with this project was to make a single centralised database that could incorporate database requirements of a well functioning Educational Institute upto maximum extent.

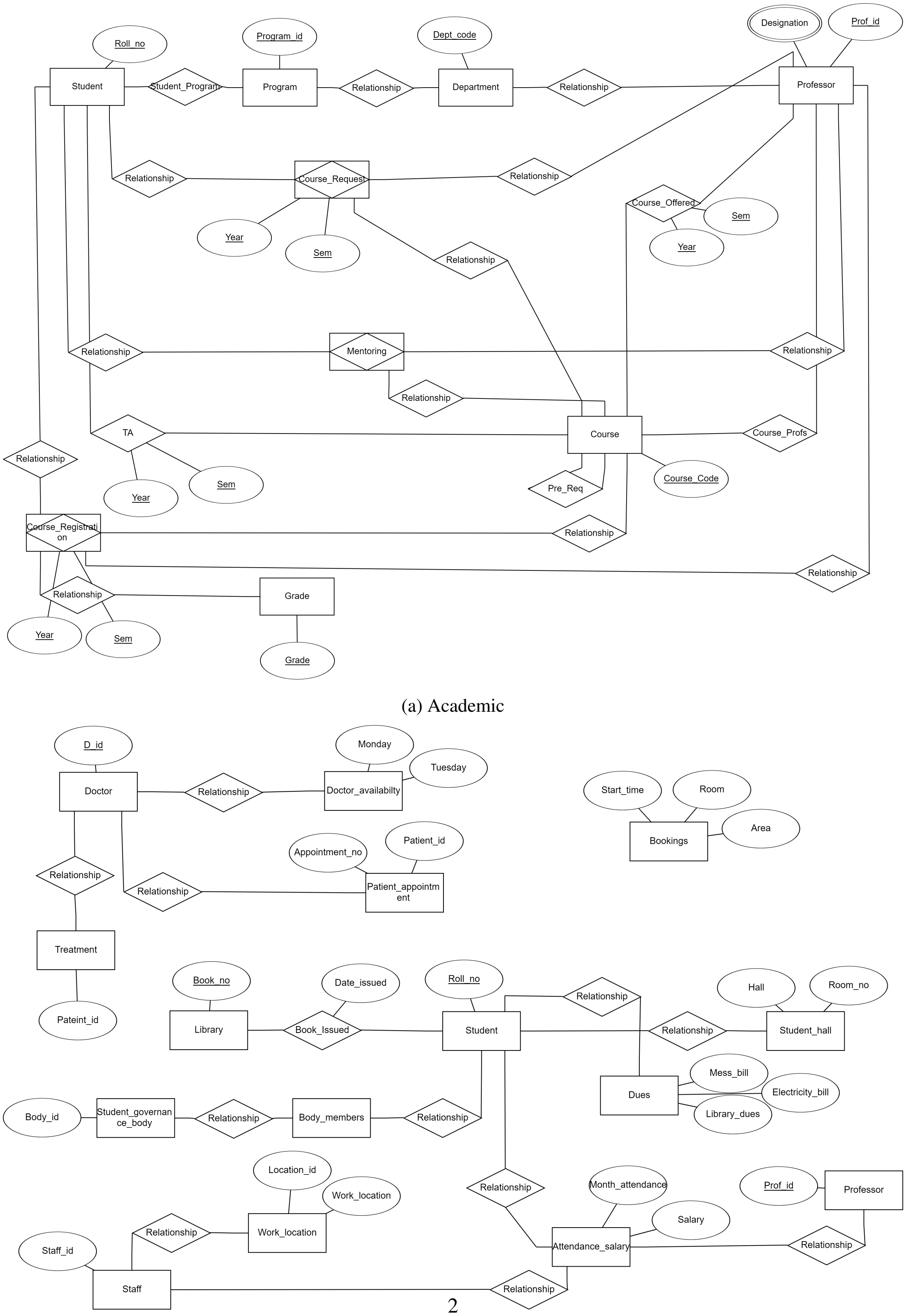
## 1 Database Description

The Normalisation Form is concentrated towards BNF, while only a few schematics in 3NF form. Breakdown of database in form of entities and relations is provided below along with brief description.

- ENTITIES

  1. Student
  2. Professor
  3. Doctor
     Tables Student, Professor and Doctor are incorporated in our database to store the primary details like institute-id, name, e-mail etc of all the three entities.
  4. Staff
     Across the institute we have staff, to facilitate easy maintenance and smooth working of the institute and each one of them contribute in a different way. Work-description stores details of work done by the staff member.

     It becomes easy for person in-charge of particular location to find the staff-id working under them by finding tuples with same location-id.
  5. Department
     Just the code and name linkage of departments.
  6. Course
     For each course its code, name and credits are stored.



(a) Academic

(b) Non Academic

Entity Relation Diagram.

* ERD is simplied, to prevent overcrowding of attributes

7. Grade
   Grade and its corresponding grade value is stored.This is included to allow flexibility in case of future structural changes in grade allocation rules.

8. Ongoing
   Storing current date, semester and year, this table is helpful in defined triggers and for pre registration processes (offering courses, current semester CPI) displayed in front end.

9. Library
   Treasure house of every educational institute.
   We are storing each Book with its accession number (for organisational ease) and total copies we have in a library.

10. Work_Locations
11. Student_Governance_Body
    Student bodies helps in cultivating qualities of leadership amongst students and also some bodies strives to ensure the welfare of the students by providing them emotional, academic and financial assistance.

    For each body we have its Id, name and funds allocated by the institute.

12. Bookings
    LHCs, VH rooms, Lab rooms, Auditorium multiple such facilities provide booking services for any suitable purposes, and to efficiently check and make availabilities and booking status this table is generated.

- Tables for Multi Valued Attributes
  1. Prof_Designations
     Professor might be serving multiple roles in a Department (for E.g. Admission Incharge, Lab Incharge, HOD, Professor).

  2. Pre_Req
     Course may have prerequisite(s) and a student had to have completed them all with E or better grade to request the same course.

- RELATIONS
  1. Program
     For completion of degree minimum SO, ESO, DE, IC,HSS etc credits are required which vary for all the programs.

  2. Student_Program
     Multiple Student Program enrollment is possible, hence requiring separate table.

  3. Course_Offered
     All the courses offered in each and every semester and year, till day are stored. Also the professor incharge id taking the course offered is stored.

     If a student wants to know the course type (OE, DE, DC etc) and whether or not they can have a time clash with any other course can be easily checked using type-offered and time-slot attribute respectively.

4. Course_Profs
   For each course offered in a particular sem only one professor will be in-charge but co-instructors might be there. This table is present for maintaining all the co-instructor-ids.

5. Course_Request
   For implementing a working system of pre registration this table was used to store momentary statuses of course requests applied via students.

6. Course_Registration
   Once, course request is accepted or rejected via professor, they got automatically added in Course_registration, with the help of trigger.
   This table contains info about all the courses registered, their grades in any semester.

7. Mentoring
   This table maintains the student list and respective instructor id under which the student is registered for courses taken as UGP or thesis.

8. TA
   Stores the id of students who serve as technical assistant for any course.

9. Doctor_Availability
   Bitmap corresponding to slots for each day is stored . If a doctor is available in a particular slot the values for that slot is stored as 1 else 0.

10. Treatment
    Stores the details of patient and the doctor-id who treated the patient.

11. Patient_Appointment
    Stores the details of the appointment taken by the patient.

12. Student_Hall
    Stores the hall and room no. of the students.

13. Book_Issued
    This table is helpful maintaining the total no. of book which has been issued. This also stores which book is issued by whom and the date when it was issued, the expected return date and the date when the book was actually returned. Due date and return date are maintained to calculate the extra days the person has kept a book and then fines are charged accordingly.

14. Dues
    Dues include the library, mess and electricity dues of a student.

15. Body_Members
    Stores roll no. of students holding different positions for each body.

16. Attendance_Salary
    Maintains the monthly attendance and salary of the employees

## 2 Challenges Encountered

- Establishing database connection with the javascript was a tiresome task.
  For accepting/rejecting particular requests, we had to build onclick JS function inside .php page so that we can apply query on a selected tuple from displayed table.

- Database if fully connected and is huge, in terms of number of tables and attributes.
  - Minor changes in one table required overhaul in many other tables.
  - Maintaining a good normalised structure of data required generation of many relations.

## 3 Tools and Software used

- *Backend:* MySQL
- *Front End:* PHP, JavaScript