Project 1: TinyHub
Data Models and Query Language

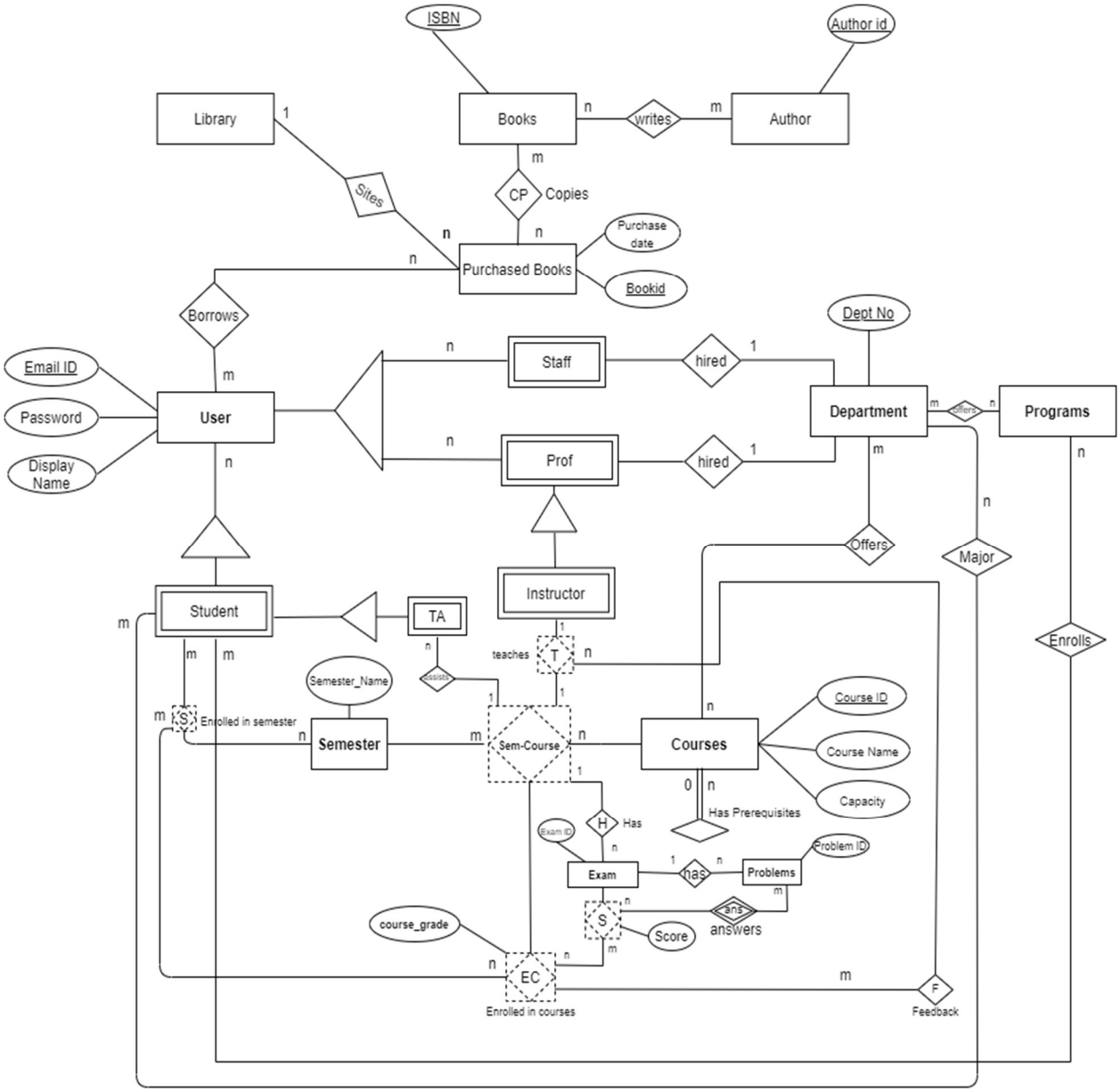
Anand (anand6)

Rahul Patil (rpatil)

Jing (jchen445)

E/R Schema:

The E/R model for the given task is as follows:



We're designing a database schema for TinyHub – a course enrollment system. The design of this database involves two important functions:

- User Management
- Department Management

- Course Management
- Library Management

USER MANAGEMENT

- The user is created with an email, which is unique for each user, and cannot be changed.
- The user is password protected
- The user can have an optional display name which can be changed as far as the display ID is unique.
- The user management system serves as a hub for any staff, professor or student to further proceed with other functions.
- The user attribute has 3 different entities: student, staff and professor.

DEPARTMENT MANAGEMENT

- The department is uniquely identified by department number.
- Each staff member is related to one department
- Each professor is related to one department
- Departments offer different programs.
- Student can major in different departments.
- A student can have any number of programs but can major in one department only.

COURSE MANAGEMENT

- A course can be identified by its course name and course ID where course ID has to be unique.
- A course also has cannot exceed a defined maximum strength.
- A course may or may not be offered in a particular semester
- A semester is defined by the year it is being offered and the season the course is offered.
- A course can be conducted multiple times in the same semester.
- Each course can have one or more TAs and only one instructor who can be changed in middle of the semester.
- An instructor must be a professor.
- A TA must be a student.
- Each course might or might not have a pre-requisite to be completed.
- A student is allowed to register to courses in different semester if they're being offered.

STUDENT COURSE MANAGEMENT

- A student can enroll in different courses if the following holds true:
    - Student is registered for the particular semester.
    - A course is offered in that semester
    - The department in which the course is offered is the same department in which the student is majoring
    - Course has not exceeded its maximum capacity for the semester.
- A student registered in the course is eligible to provide feedback to the instructor
- Each course will have exams and each exam has questions which carry scores.
- At the end of the course, each student is offered a grade F/D/C/B/A.

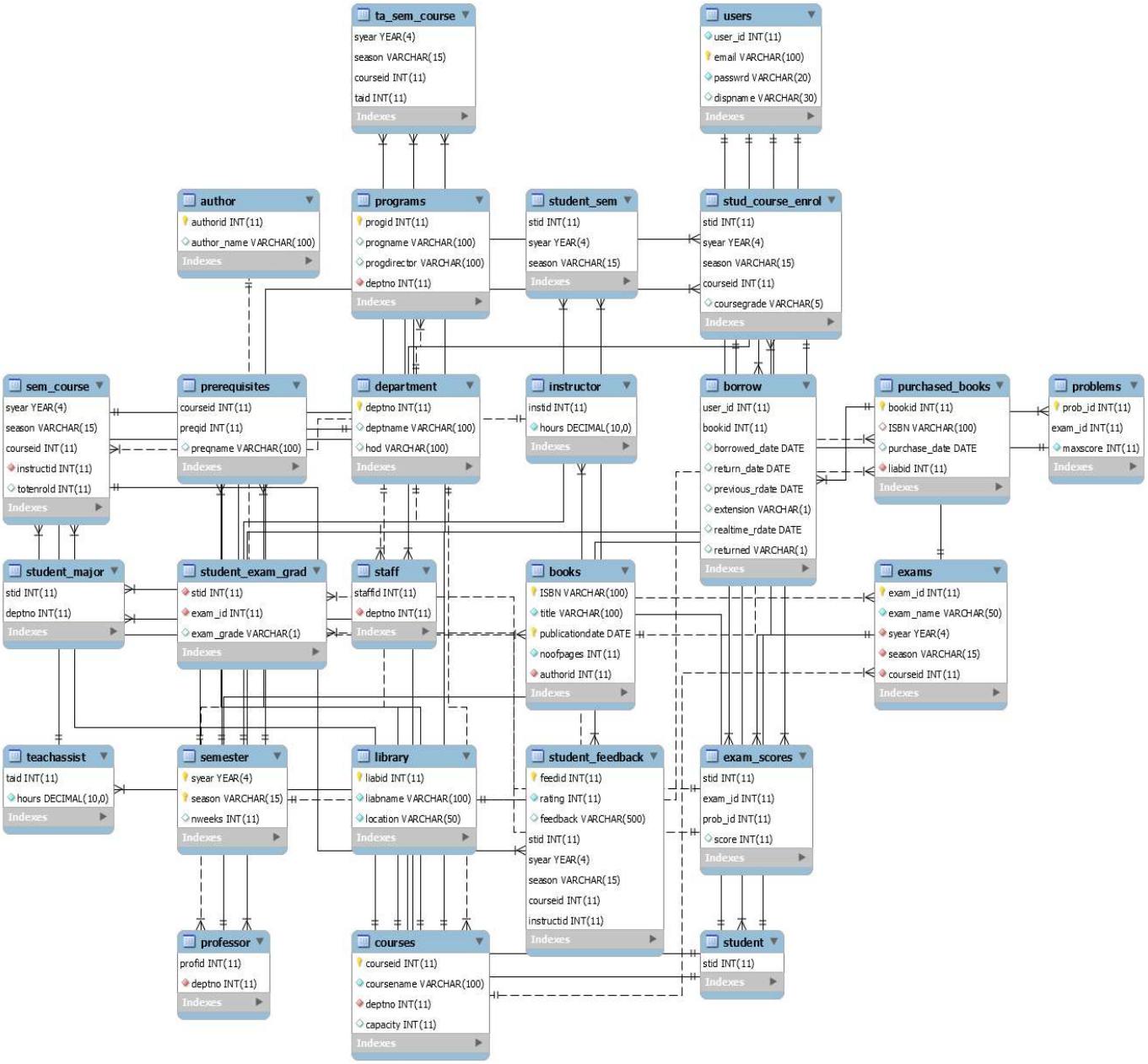
LIBRARY MANAGEMENT

- Multiple libraries each have a unique code based on their location.
- Books are identified through ISBN, although they also have title, author(s), number of pages, publication date.
- An author can write multiple books
- Books can be in multiple libraries
- A book can have different purchase dates and different prices
- Users can borrow books which needs to be returned within two weeks of the date of borrow, although there is an option for a week's extension.
- If the borrowed book(s) is not returned before due date, then no books are further allowed to be borrowed.
- A user borrowing the book has to return the same copy of the book and no other copy will be accepted.

RELATIONAL SCHEMA

The schema consists of the following tables:

- Author
- Books
- Borrow
- Courses
- Department
- Exam_Scores
- Exams
- Instructor
- Library
- Prerequisites
- Problems
- Professor
- Programs
- Purchased_Books
- Sem_Course
- Semester
- Staff
- Stud_Course_Enrol
- Student
- Student_Feedback
- Student_Major
- Student_Sem
- Ta_Sem_Course
- Teachassist
- Users



SCHEMA

1. USER MANAGEMENT:

a. Users:

Users table is the main table for this database which refers to all the members of this database. It consists of 4 columns, user_id which is a integer number generated for each new member and is referenced throughout the database to identify an individual. Email is the email ID of each individual which is unique, since each of these users are password protected, and a display name which is a optional field.

b. Student:

Student table refers to the users who are students, with column stid which is student_id referenced from the 'user_id' column of 'Users' table.

c. Professor:

The professor table consists of profid which is the user_id of the professor and deptno which enlists the department they belong to.

The primary key is profid referenced from the 'user_id' column of 'users' table and deptno is the department number referenced from the 'deptno' column of departments table.

d. Staff:

This table represents the staff in each dept. The column staffed is the primary key which is the user_id of the staff member referenced as a foreign key from the 'Users' table and 'deptno' is the department number referenced from the 'departments' table.

2. DEPARTMENT MANAGEMENT:

a. Department:

Department table lists the different departments in the university along with the department number being primary key, department name and head of the department.

b. Courses:

This table describes different courses offered by the university. Columns are coursid, coursename, deptno and capacity. Courseid is the primary key and deptno is the department number referenced from departments table.

c. Programs:

The programs table lists the different programs offered by departments. Here, the column progid is the primary key, with progname and progdirectors being remaining columns. The column deptno is the foreign key referenced from the 'departments' table which identifies the department to which a specific program belongs.

3. COURSE MANAGEMENT:

a. Sem_course:

This table describes the courses offered in different semesters. The columns are syear for the year, season, courseid, instructed and totenrold which is the strength of total class. Syear, season and courseid are the primary keys, with foreign keys being syear and season from 'semester' table, courseid from 'courses' table and instructid from 'instructor' table.

b. Instructor:

Instructors table consists of instid being the primary key and the other column is the number of hours worked.

c. Semester:

This table describes the different semesters. Columns being syear which the the semester year, season viz. Spring, Summer or Fall, and nweeks which are the no. of weeks in the semester. Syear and season are the primary key.

d. TA_sem_course:

This table references to the TAs assigned to each course of every semester. It consists of columns syear, season, courseid and taid which is the primary key and student id referenced as Foreign key from 'students' table.

e. Teachassist:

This table refers to the number of hours employed by each TA. It has two columns taid which is the primary key is a foreign key referenced from TA_Sem_course table and hours is a integer table.

f. Prerequisites:

Prerequisites table lists the different prerequisites for any course. The columns consisted are courseid, preqid and preqname. Here, preqid is the primary key and courseid is the foreign key referenced from the courses table.

g. Student_sem:

This table references the semester the student belongs to. It has columns stid, syear and season being the primary keys with stid being referenced as foreign key from student table and syear and season referenced from semester table.

4. STUDENT COURSE MANAGEMENT:

a. Stud_course_enrol:

This table shows the courses and the syudents enrolled to those courses. Te columns of this table are stid i.e student ID, syear i.e. semester year, season i.e Spring, Fall, Summer etc, courseid and course grade. Here stid, syear, season and courseid are the primary key which are referenced from the tables student, semester and sem_course respectively.

b. Student_exam_grade:

This table refers to the grades obtained by students in the exams they appeared. It consists of columns student_id, exam_id and exam_grade.

c. Student_feedback:

This table refers to the student feedback of the course and the instructor. It consists of columns stid, feedid i.e the feedback_id, rating provided by each student, feedback, syear i.e the semester year, season and courseid. Here, stid, syear, season, courseid, instructid, feedid are the primary keys, while stid, syear, season and courseid are foreign keys referenced from stud_course_enrol table, instructid is referenced from the sem_course table.

d. Student_major:

This table refers to the department in which students are majoring. It consists of columns stid from student table and deptno from the departments table.

e. Exams:

Exams table consists of records for the different exams being conducted each semester. The columns in exam table are exam_id, exam_name, syear, season and courseid. Exam_id is he primary key here, with syear and season the foreign keys referenced from table 'semester' and courseid from the 'courses' table.

f. Problems:

Problems table consists of the problems that appear in an exam. The columns in this table are prob_id, exam_id and maxscore. The primary scores are exam_id and prob_id and exam_id is a foreign key referenced from exams table.

g. Exam_scores:

Exam_scores table records the diffent scores stored by students for student management. It consists of columns student id, exam_id, problem_id, and score by the student.

Here, student_id, exam_id and problem_id are the primary keys, where student_id is referenced from table student_course_enrol, prob_id from table problems and exam_id from the exams table.

5. LIBRARY:

a. Library:

Library table describes the library location and different libraries in the university. The columns are liabrary ID, library location and library name. Here, liabid is the primary key.

b. Books:

Books is a table for the different books in the library. The table consists of columns ISBN, Publication Date, Title, No. of pages and authorid.

ISBN and Publication date are the primary keys and authorid is referenced as foreign key from table Author.

c. Author:

In this table, we are defining the authors for books in the library.

authorid is the primary key and the table consists of another column author_name

d. Purchased_books:

Purchased_books table lists the books present in the library and their date and price of purchase, the location of the library and the book_id. The primary key here being bookid, ISBN referenced from the 'books' table and 'liabid' referenced from the 'library' table are the foreign keys.

e. Borrow:

This table logs the books borrowed from the library. The table consists of user_id, bookid, borrowed_date, return_date, previous_date, extension, realtime_rdate, and returned columns.

User_id and bookid are the primary keys where userid is a foreign key from table 'users' and 'bookid' is a foreign key from table 'purchased_books'. The extension_date is a variable char like (Y/N) and return_date is calculated as 2 weeks from the borrowed date. Previous_rdate is a return date for books borrowed previously.

JUSTIFICATIONS AND APPLICATION HANDLING:

We've referenced tables in such a way that the specified constraints such as only a enrolled student in a particular semester can provide feedback etc. are handles. However, for more effective functioning of the database, the following can be implemented:

- Procedure to check for password strengths
- While creating a role, check the student, professor and staff tables if there is a pre-existing user or pre-assigned role.
- Restrict the programs available for students based on the department they belong to. This ensures they choose and major in the department they belong to.
- Only courses for a selected year and season that are offered are displayed to students while enrolling, details of which are available in the table student_sem -> syear, season.
- Check if all the prerequisite courses are completed, using student_course_enrol.courseid = prerequisites.preqid and student_course_enrol.coursegrade
- Check if the capacity of the course exceeds the maximum strength of the course from the course.capacity and sem_course.totenrold column and update the sem_course.totenrold column
- Restrict the book extension of the borrowed_books table to just once i.e. a flag to check if the extension is already chosen or not.
- A restriction on borrowing books, if any book is not returned post return date of the borrowed_date column. That is flag the borrower if (borrow.realtime_rdate-borrow.return_date)>0
- If borrow.bookid does not match the bookid of the returned book, then it must be rejected.
- Can use a stored procedure for batch processing such that:
    - Notification mails are sent to students to return books 3 days prior to the return date i.e (borrow.return_date – current_date)<= 3
    - Exam and course grades are calculated by stored procedure calls

ADVANTAGES:

- Discrepancies can be avoided as tables for course registration reference the table for sem_enrolled and table for feedback references the table for sem_enrolled. This prevents duplicate and unauthorized entries.