



University of Barishal

Project Report

COURSE TITLE : Database Management System Lab

TOPIC : College Database Management System

COURSE CODE : CSE-2102

SUBMITTED TO : Md Samsuddoha
Assistant Professor
Dept. of Computer Science & Engineering
University of Barishal

SUBMITTED BY : Md Aminul Islam (21CSE042)

Dept. of Computer Science & Engineering
University of Barishal

SESSION : 2020-2021

SEMESTER : SEMESTER-03

DATE : 24-03-2024

TITLE : College Database Management System

DESCRIPTION

The College Database Management System (DBMS) project presents a comprehensive solution aimed at modernizing and optimizing administrative processes within educational institutions. Developed through SQL queries, the database schema encompasses interconnected tables representing key entities such as students, faculty, courses, and administrative resources. This schema facilitates efficient data management while adhering to normalization principles to ensure integrity and minimize redundancy. Features include student and faculty management functionalities, course cataloging, enrollment tracking, and administrative tools for reporting and resource management. Implemented with the InnoDB storage engine, the system maintains referential integrity through foreign key constraints, ensuring data consistency. Rigorous testing procedures validate the system's functionality, data integrity, and performance across various scenarios. The project's objective is to streamline operations, enhance decision-making, and improve services for students, faculty, and staff, with potential for future enhancements such as integration with analytics tools and learning management systems. Overall, the College DBMS project offers a robust solution tailored to meet the evolving needs of educational institutions in the digital age.

MOTIVATION

The motivation behind developing the College Database Management System (DBMS) stems from the recognition of the inefficiencies and complexities inherent in traditional administrative processes within educational institutions. Manual methods of managing student records, faculty details, course information, and administrative tasks often lead to errors, delays, and resource wastage.

The primary motivation for implementing the College DBMS is to address these challenges by leveraging technology to streamline and automate administrative workflows. By centralizing data management and providing intuitive interfaces, the system aims to enhance operational efficiency, improve data accuracy, and facilitate informed decision-making.

Moreover, the increasing reliance on digital solutions in education necessitates the adoption of modern tools to stay competitive and meet the expectations of students, faculty, and staff. The College DBMS aligns with this trend by offering a scalable, user-friendly platform that enhances collaboration, communication, and overall productivity within educational institutions.

Furthermore, the potential benefits of the College DBMS extend beyond administrative efficiency to include improved student experiences, enhanced academic outcomes, and better resource utilization. By optimizing processes such as enrollment management, course scheduling, and

academic support services, the system ultimately contributes to the overall success and reputation of the educational institution.

In essence, the motivation for developing the College DBMS lies in its potential to transform administrative operations, empower stakeholders, and foster a more conducive environment for teaching, learning, and research within educational institutions.

REQUIREMENTS

- User Management:
 - Implement authentication and authorization mechanisms.
 - Allow users to register, log in, and manage their profiles.
- Student Management:
 - Enable student registration and enrollment in courses.
 - Provide functionalities for viewing academic records, grades, and course schedules.
 - Allow submission of assignments and viewing of submissions.
- Faculty Management:
 - Enable faculty members to manage their profiles and teaching assignments.
 - Provide tools for managing course details, assignments, exams, and grades.
 - Enable communication between faculty and students.
- Course Management:
 - Implement a course catalog for adding, updating, and deleting courses.
 - Allow instructors to define course materials like assignments, exams, and readings.
 - Provide course scheduling functionalities for classrooms and timeslots.
- Enrollment Management:
 - Enable students to enroll in courses and manage their selections.
 - Implement waitlist functionality for oversubscribed courses.
 - Allow administrative staff to manage enrollments.
- Administrative Tools:
 - Generate reports on student enrollments, course schedules, and academic performance.
 - Manage resources such as classrooms, libraries, and facilities.
 - Provide communication tools for announcements and messaging.
- Security and Data Protection:
 - Ensure data security and protect sensitive information.
 - Enforce role-based access control.
 - Implement data encryption and secure transmission protocols.
- Integration and Compatibility:
 - Ensure compatibility with various devices and platforms.
 - Integrate with external systems such as learning management systems (LMS) or student information systems (SIS).

- Scalability and Performance:
 - Design for scalability to handle increasing data and user loads.
 - Optimize database queries and indexing for efficient data retrieval.
 - Monitor system performance and implement optimizations.
- Usability and Accessibility:
 - Design intuitive and accessible user interfaces.
 - Ensure compatibility with assistive technologies and accessibility standards.

These requirements collectively form the framework for developing a robust and functional College DBMS tailored to the needs of educational institutions and their stakeholders.

Entity Relationship Diagram Step by Step:

❖ Step 1. Entities:

1. Student
2. Teacher
3. Faculty
4. Department
5. Course
6. Enrollment
7. Schedule
8. Assignment
9. Submission
10. Exam
11. Exam Result
12. Classroom
13. Campus
14. Library
15. Book Loan

❖ Step 2. Attributes & primary key for each entities:

Student (StudentID, FirstName, LastName, Email, DeptID)

Teacher (TeacherID, FirstName, LastName, Email, DeptID)

Faculty (FacultyID, FirstName, LastName)

Department (DeptID, DeptName, DeanID)

Course (CourseID, CourseName, DeptID, Credits, TeacherID)

Enrollment (EnrollmentID, StudentID, CourseID, EnrollmentDate, Grade)

Schedule (ScheduleID, RoomID, TeacherID, CourseID, StartTime, EndTime)

Assignment (AssignmentID, CourseID, Title, Description, DueDate)

Submission (SubmissionID, AssignmentID, StudentID, SubmissionDate, Grade)

Exam (ExamID, CourseID, Title, Description, ExamDate)

Exam_Result (ExamResultID, ExamID, StudentID, Course)

Classroom (RoomID, RoomNumber, Capacity, CampusID)

Campus (CampusID, CampusName, Location)

Library (BookID, Title, Author, DeptID)

Book Loan (LoanID, BookID, StudentID, LoanDate, ReturnDate)

❖ Step 3. Relationship:

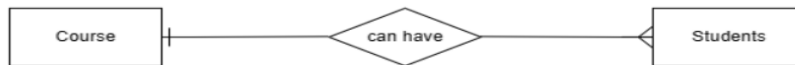
- **Student** can be enrolled in multiple **Courses**
- **Course** can have multiple **Students**
- **Teacher** can teach multiple **Courses**
- **Course** can be taught by multiple **Teachers**
- **Department** can have multiple **Courses**
- **Course** can have multiple **Enrollments**
- **Course** can have multiple **Assignments**
- **Assignment** can have multiple **Submissions**
- **Faculty** can be associated with multiple **Courses**
- **Course** can have multiple **Exams**
- **Exam** can have multiple **Exam_Results**
- **Library** can have multiple **Books**
- **Department** can have multiple **Books**
- **Book** can be loaned to multiple **Students**
- **Department** and its **Dean**
- **Campus** can have multiple **Classrooms**

❖ Step 4. Cardinality ratio and participation:

1.Student can be enrolled in multiple **Courses** (*one to many*)



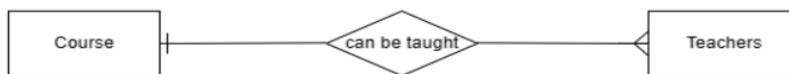
2. Course can have multiple **Students** (*one to many*)



3. Teacher can teach multiple **Courses** (*one to many*)



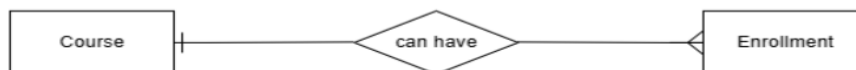
4. Course can be taught by multiple **Teachers** (*one to many*)



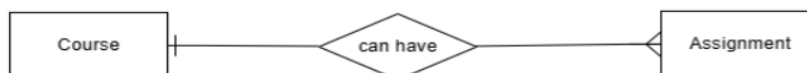
5. Department can have multiple **Courses** (*one to many*)



6. Course can have multiple **Enrollments** (*one to many*)



7. Course can have multiple **Assignments** (*one to many*)



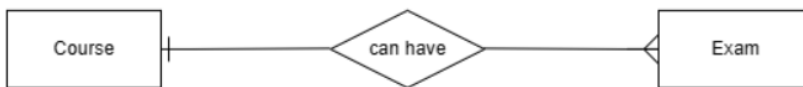
8. Assignment can have multiple **Submissions** (*one to many*)



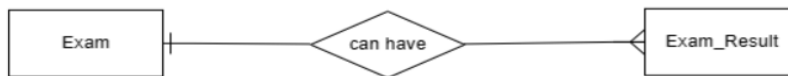
9. Faculty can be associated with multiple **Courses** (*one to many*)



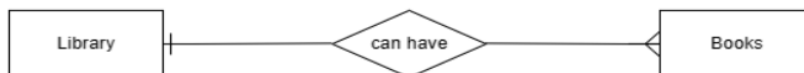
10. Course can have multiple **Exams** (*one to many*)



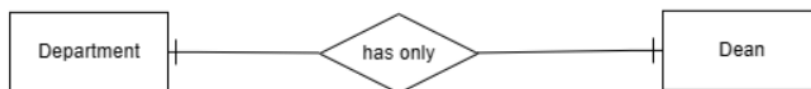
11. Exam can have multiple **Exam_Results** (*one to many*)



12. Department can have multiple **Books** (*one to many*)



13. Department and its **Dean** (*one to one*)



❖ **Step 5: Draw ER diagram:**

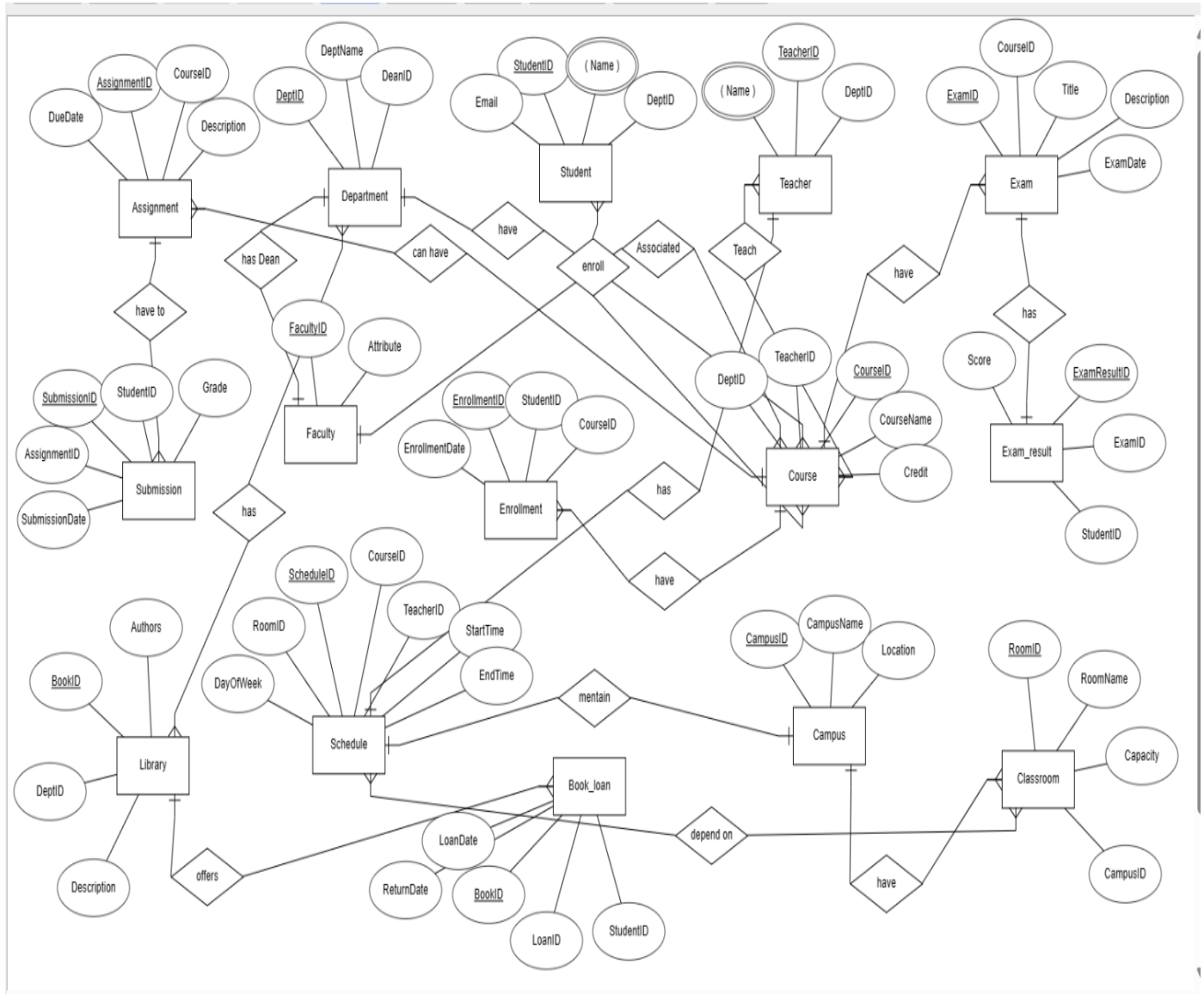


Fig. ER-Diagram of College Database Management System

❖ Step 6 : Schema Diagram

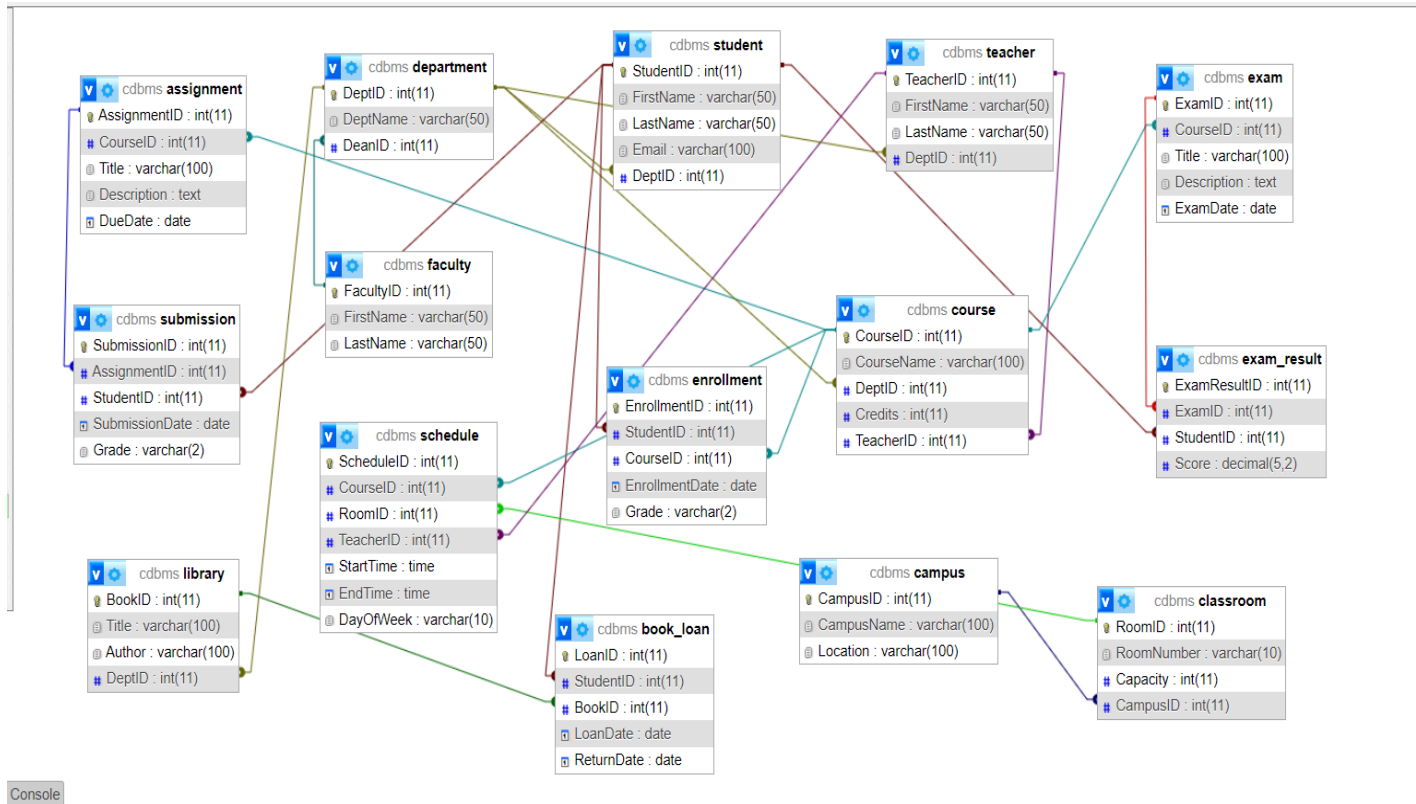


Fig. Schema Diagram of College Database Management System

CONCLUSION

The College Database Management System (CDBMS) project has successfully developed a comprehensive database schema tailored to academic institutions' needs. It efficiently manages various college operations such as course enrollment, assignment submission, exam scheduling, and library resource management. With a user-friendly interface and robust security measures, it ensures secure data access and manipulation. The system's analytical capabilities empower administrators and faculty to make informed decisions. Ongoing maintenance and improvement efforts are crucial to sustaining its effectiveness in meeting evolving college requirements and technological advancements.