College Data Management System

Database Management Systems

MSCS_542L_256_23S

Super Six

Marist College

School of Computer Science and Mathematics

Submitted To: Dr. Reza Sadeghi

Spring 2023

Project Report of College data management system

Team Name

Super Six

Team members

Pradeep Reddy Baireddy.	pradeepreddy.baireddy1@marist.edu (Team Head)
Prajakta kshirsagar	prajaktakshirsagar2022@gmail.com (Team member)
Deepthi Niharika Gadepalli	DeepthiNiharika.Gadepalli1@marist.edu (Team member)
Bharadwaja Thota	bharadwaja.thota1@marist.edu (Team member)
Pradeep Reddy Macha	pradeepreddy.macha1@marist.edu (Team member)
Chakradhar Reddy Marepally	ChakradharReddy.Marepally1@marist.edu (Team member)

Table of Contents

Table of figures

Objective:

The primary objective of the college data management system is to efficiently manage the data that includes a huge amount of data that related to student, courses, faculty, department, admissions and other vital information that need to be stored and retrieved. At this stage, this phase is presenting the conceptual design of college data management system through an Entity-Relationship (ER) model and an Enhanced ER (EER) model.

Review the Related Work:

After extensive research, the related work on college management systems has focused on their system design, implementation and usage. These systems are important for better communication, operations, and useful data for decision-making. But we need to address organizational and technical challenges to make these systems effective, such as having user-friendly interfaces, clear roles for managing data, and getting everyone involved. College data management systems can help in decision-making, but it is important to have the right tools to understand the data, such as data analytics. To ensure effective data management in colleges and universities, it is crucial to address privacy and security concerns that arise due to storing sensitive data. Integration with other systems is also important for data accuracy and access.

[1] This study compared different college data management systems based on features such as student information management, course management, and financial management. The study found that each system has its own strengths and weaknesses, and the choice of system should depend on the specific needs and requirements of the college or university.

[2] This system is designed to store and manage student information, academic records, and administrative data in a secure and efficient manner. The authors implemented the system using Amazon Web Services and evaluated its performance using various metrics. The results showed that the system is scalable, reliable, and cost-effective. The proposed system can be used by colleges and universities to manage their data more effectively.

Merits

- Data privacy and security in college data management systems able to provide security for sensitive data of students and faculty such as personal and financial information. This will ensure students and faculty that their sensitive data is secure.
- Increased efficiency in the college data management system helps to save time in data storing and retrieving and helps to automate some tasks like attendance.
- Data analysis in college data management system will help the institutions to analyze the previous data that has been stored and using it for better outcomes in future.
- An efficient accessibility in college data management system will allow the students, faculty and administration staff to access the data swiftly such as academic attendance, grades, academic records and schedules.

Entity Relationship Model (ER Model):

Entities:

Entity	Attributes
Student	Student_ID(PK), First_Name, Middle_Name, Last_Name, DOB, Gender, Age, Hostel_ID, Program_type
Faculty	Fac_IDI(PK), Faculty_Name, Designation, Office_hours, Dept_ID
Course	Course_ID(PK), Course_Name, Duration, Course_desc, Credits, Program_type
Department	Dept_ID(PK), Dept_Name, Courses_offered, HOD, Office_Location
Admissions	Student_ID(PK), Admission_Number, Application_date, Decision_Date, Test_written, Test_Score, Enrollment_Status
Programs	Program_ID (PK), program_Name, Credits_required, Duration, Number of Courses, Tution fee, Dept_Id
Grades and Attendence	Student_ID(PK), Session_ID, Assignment_score, Lab_score, Attendence, Quiz_Score
Student_Info	Student_ID(PK), Mail, Phone, Father_Name, Mother_name, Address (Door, Street, City, State, Zip)
Class Room	Session_ID, Class_Location, Course_ID, Faculty_ID, Timings, Duration
Hostel	Hostel_ID(PK), Hostel_Name, Amenities, Room_Number, Vacancy, Price_Range

Entity Relationship Diagram

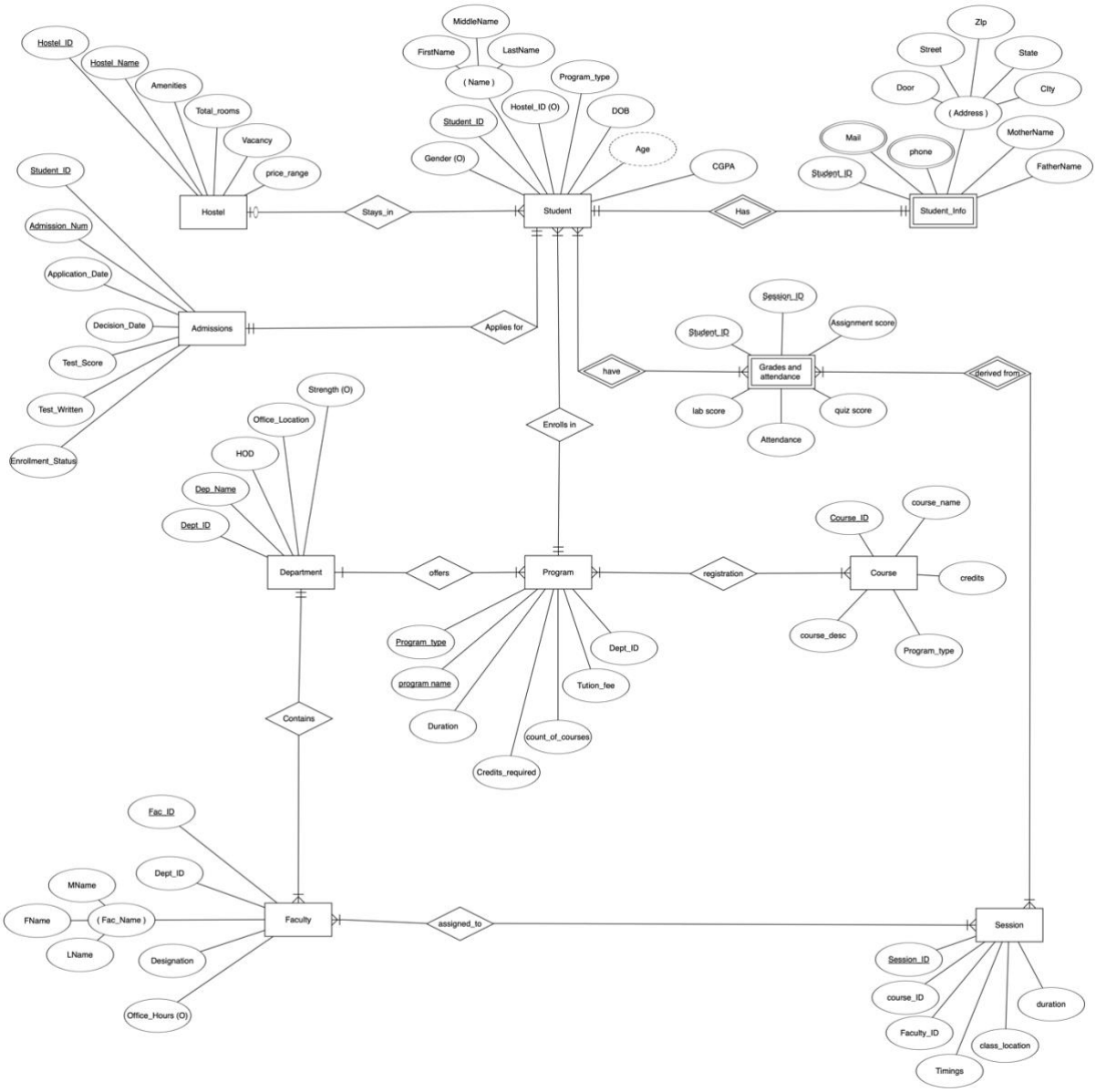

Figure 1 Conceptual design of Entity Relationships

Figure 1 shows the conceptual design of Entity Relationships using multiple entities, attributes and other important relations described below

Implementation of ER Diagram:

To carry out this operation, we use online ER-Diagram program named ERDplus. Since students make up the majority of the college management system, we began with the student entity. Student ID serves as the primary key for the student entity, while Name serves as a composite property. In addition, we have other attributes like DOB, gender, program type, and age, which is a derived attribute from DOB.

The student entity then has a one-to-one cardinal relationship with the student info entity (1:1). One record must be used for each student's personal information, and this record is linked to other records using student ID as a foreign key. Additionally, the connection is seen as a "identifying relationship." The student info contains details like the names of the parents, an email address, an address, and a phone number. Phone number is a multi-valued attribute, while address is a composite attribute.

The hostel entity and the student entity have some connections. Due to the fact that not all students must stay in the hostel, there is only a limited participation. The hostel's Hostel ID and Hostel Name serve as distinctive identifiers, and additional attributes include the number of rooms, vacancies, and pricing ranges. Given that multiple students can lodge in a single hostel, the relationship is one of many to one (N:1). Admission number serves as the main key for the admission entity, which is also tied to the student entity in a (1:1) connection.

Due to the fact that one program can accommodate numerous students, the connection between the student and program entities is N:1 many-to-one. Program ID, Program Name, credits needed, number of courses, department ID, and tuition are some of the attributes that the program entity has. Program Name and Program ID are referred to as primary keys. Given that multiple programs may be offered by a single department, the relationship between the program and department entities is many-to-one. Other attributes include department name, courses offered, HOD, and office location. The dept ID serves as the primary key for all of these. With a key attribute called faculty ID as well as other elements like name, designation, office hours, and department ID, the faculty entity is tied to the department entity. In this manner, the remaining entities are created in accordance with the demands of the college management system.

Multivalued Attributes:

For the Student_info which is a weak entity, there are 2 multivalued attributes that are mail and phone. A single student can have multiple phone numbers and mail addresses to be contacted so for our database design, these 2 are the multivalued attributes.

Composite Attributes:

For this design, we have Student name, Faculty name and Address as the composite attributes. Both student name and faculty name are further sub-divided as first name, middle name and last name. For address the sub attributes are door, street, zip, state and city.

Derived Attributes:

We have taken only one derived attribute that is age. Age is a derived attribute from DOB which is itself an attribute for the student entity

Weak Entity:

We have taken Student_info and Grades and Attendance as our weak entities for this database design. Both these weak entities are related to the student entity.

Strong entity:

The student, program, course entities are examples of strong entities. They do not rely on any other entities for representation and identification as they can exist uniquely.

Participations:

There are two kinds of participation, the partial participation does not require the involvement of entire entities in the relationship whereas the total participation required every record in the entity to be related with the other. The relationship between hostel and student entities can be partial as every student is not required to stay in hostel but student and student_Info entities are under total participation as every student must has his own record of information.

Cardinality Ratios

The above Entity Relationships diagram covers all the cardinality ratios such as1-1 cardinality ratio, 1-N cardinality ratio, and M-N cardinality ratio. Student and Student info have 1-1 cardinality ratio, student and program have 1-N cardinality ratio, and department and Faculty have M-N cardinality ratio.

Identifying and Non-Identifying Relationships

The student and student_Info entities are in identifying relationship because the student_Info do not have a unique identifier to identify the entity uniquely. The admission entity and the student entity are in non-identifying relationship as both entities can be represented uniquely without dependency.

Enhanced Entity Relationship Model (EER Model):

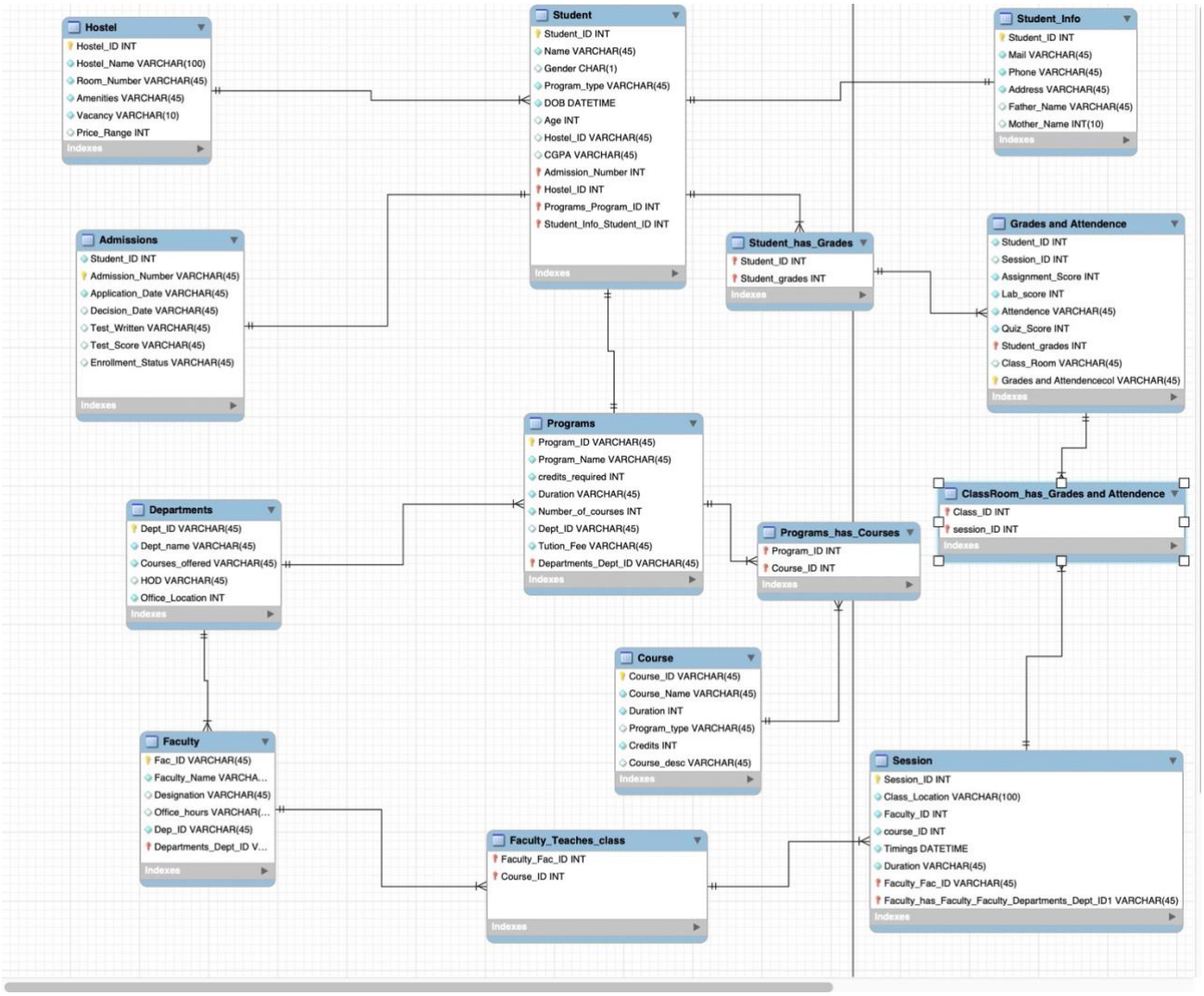

Figure 2 Enhanced Entity Relationship

Implementation of EER:

For creating an EER diagram, we used MySQL Workbench. Instead of going to the query tab we can go to the 2nd option in left menu of the Workbench through which we can design and view the EER diagram for our database. Using different tools, we can create the EER design as per our requirement [3].

The entity Student has an attribute hostel_id, which uniquely identifies each of the row in the entity Hostel, Program_type attribute is refferd as program_ID attribute in the "Program" entity, similarly each of the table interrelated to "Student" Entity can be uniquely related to other. Program. By the same pattern every entity can be related.

Every entity provided in the EER model is interrelated to each of the other entity, we are relating them using the primary and composite keys to manage the attributes in the entities. All the entities in the table except for "Grades and Attendance" has Primary Key. "Grades and Attendance" entity can be identified with the composite key by taking attributes "Student_ID" and "Session_ID". So, every entity in the model is uniquely identified, by means of which we can manage the data.

References:

[1] Patel et al. (2018) *"A Comparative Study of College Data Management Systems".*

[2] Lin et al. (2019) *"Design and Implementation of College Data Management System Using Cloud Computing".*

[3] *https://www.edrawsoft.com/article/what-is-eer-diagram.html*

GitHub repository address:

https://github.com/PradeepReddy-Baireddy/MSCS_542L_256_23S_College-Data-Management-System_Super-Six