Indian Institute of Technology Gandhinagar

Alumni Database Management System

CS432 Project Report
1st February, 2023

Authors

Dhairya Shah 20110054
Bhavesh Jain 20110038
Inderjeet Singh 20110080
Rahul Chembakasseril 20110158
Joy Makwana 20110086
Kalash Kankaria 20110088
Kanishk Singhal 20110091
Harshvardhan Vala 20110075
Nokzendi Aier 20110173
Medhansh Singh 20110111

Under the guidance of

Professor. Mayank Singh

Responsibilities of Group 1

Answer 1:

Our alumni database management system is a database that stores information about former students (alumni) of the Indian Institute of Technology, Gandhinagar. The database will manage and track alumni relationships, communication, and engagement with the institute. The database will include personal details such as name, contact information, educational background, career information, relationship with the institute, volunteer activities and events attended, and much more.

The main stakeholder involved here is the Alumni Relations of IIT Gandhinagar. The professor in charge of Alumni Relations is Prof. Jaison Manjaly, and that's why we had a few meetings with him to discuss the features of our database. We learned that the current alumni database is static, and the staff members have to update information about the alumni manually. To solve this problem, we will implement our dynamic database system on Microsoft Dynamics 365, linking it with Linkedin Sales Navigator. Through this project, we aim to extend the database to other important stakeholders like students, alumni, and staff by creating different views.

Compared to the current database, the idea of a dynamic database greatly impacts the institute. There are many use cases, such as the following:

- Alumni Outreach: The database can be used to maintain contact with alumni and keep them informed about the latest developments at the institute, such as events, news, and reunions. LinkedIn Sales Navigator can find and target alumni on LinkedIn, while Microsoft Dynamics 365 manages and tracks alumni communication.
- Fundraising: The database can identify potential donors and target fundraising efforts. Microsoft Dynamics 365 to track giving history and identify high-value prospects.
- Career Services: The database can provide alumni career-related resources and networking opportunities.
- Event Management: The database can manage event planning, registration, and attendance. Microsoft Dynamics 365 can manage event invitations and attendance, while LinkedIn Sales Navigator can target alumni likely to attend specific events.

Corresponding to these use cases, our database has many functional requirements. Functional requirements are the specific capabilities that a system must have to meet its purpose. The functional requirements can include

- The ability to store, update, and retrieve alumni information dynamically, including personal details, educational background, and career information.

- Measures to ensure the confidentiality and security of alumni data, including user authentication and data encryption.
- The ability to search for alumni based on specific criteria, such as location, job title, year of graduation, et cetera.
- The ability to generate reports based on alumni data, such as alumni demographics, giving patterns, and event attendance.
- The ability to send targeted communications to alumni, including email, direct mail, and SMS.
- The ability to manage event planning, registration, and attendance, including the ability to send invitations, track attendance, and follow up after events.
- The ability to integrate alumni data with other systems, such as fundraising and financial management systems.
- The ability to manage user access to the system, including user roles and permissions.
- Import, export, and recover data.

Answer 2:

The primary stakeholder associated with the alumni database is the Alumni Relations of IIT Gandhinagar. Professor Jaison Manjaly heads it. We had two-three meetings with Professor Manjaly to understand how our database would look like.

We ideated with him about the use cases of the alumni database, which are mentioned in the previous answer. We inquired about the attributes that he felt were necessary and crucial for the database and sought information about the availability of the data to populate the database we would make. We also discussed the final version of the database, which must be set up in Microsoft Dynamics 365 with entries that dynamically keep updating via the LinkedIn profile using Linkedin Sales Navigator.

Given the use cases, after the comprehensive database is formed, we can also involve other stakeholders in our database. By setting up different views, and permissions, we can have faculties and staff as our stakeholders. We can also have our students and alumni as the stakeholders of the database, who could use it according to their requirements, as alumni can get access to job postings, and students can connect to specific alumni with certain attributes.

Answer 3:

Entities and attributes:

1. **Donation:** with attributes (Transaction ID, Amount, Date)-D
2. **Scholarship:** with attributes (Scholarship name, Amount, Eligibility)-H
3. **Companies:** with attributes (Company name, Company Domain)-J
4. **Visits:** with attributes (Purpose, Date)-N
5. **Job Posting:** with attributes (Post_ID, Role, Salary, Location)-D
6. **Alumni**: with attributes (Roll number, Name, CPI, Mobile no., Picture, Relationship Status, Address, Income, Branch, Degree, Place of Birth, Guardian Name, Email, DoB, JEE Rank, Year of Graduation)-H
7. **Education**: with attributes (Institution Name, Discipline, Degree, Institute City)-J
8. **Residence**: with attributes (Hostel name, Room number)-N
9. **Instructor**: with attributes (Instructor ID, Name, Work_Email)-D
10. **Projects**: with attributes (Title, Duration, Outcome)-H
11. **Courses**: with attributes (Course ID, Course name, Credits)-J
12. **Achievements**: with attributes (Description, Purpose, Date)-N
13. **Department**: with attributes (Department name, No. of Faculty)-J
14. **Extracurricular:** with attributes (Name, Domain)-N

The relationships are listed below:

1. **Achieved**: relating achievements and alumni
2. **Belongs To**: relating instructor and department
3. **Dual Major**: relating department and alumni
4. **F.A.**: relating instructor and alumni
5. **Got**: relating scholarship and alumni with attributes (Year)

6. **Lived**: relating residence and alumni with attributes (Semester)
7. **Project_guide**: relating instructor, projects, and alumni
8. **Minor**: relating department and alumni
9. **Paid**: relating donation and alumni
10. **Participates**: relating extra-curricular and alumni with attributes (Role, Achievement)

11. **Posts**: relating alumni and job posting
12. **Don_extracurricular:** relating donation and extra-curricular.
13. **Studied in**: relating education and alumni with attributes (Start Year, End Year)
14. **Takes**: relating courses and alumni with attributes (Grade)
15. **Teaches**: relating instructor and courses

16. **Scholar_don**: relating donation and scholarship
17. **Visited**: relating visits and alumni
18. **Works in**: relating companies and alumni with attributes (Role, Salary, Start Year, End Year)

19. **Role_in:** companies and job posting.

Answer 4:

C. Following entities have :

1. Mobile_Number (Roll_Number , Contact_Number)
   PRIMARY KEY = (Roll_Number, Contact_Number)
   FOREIGN KEY = Roll_Number
   Justification: Mobile number has Roll_Number as an attribute and Roll_Number is the primary attribute of Alumni, so it is a foreign key for Mobile_Number.

D. Our database design does not have any one-to-one relationship because of the following reasons:

1. Most of the attributes/features that could have been a one-to-one relationship has been added as an attribute of the alumni relation.
2. Even if we explicitly made it different from the alumni relation, there would not be any other relationship from it to another entity resulting in it being finally merged with the alumni relation once we start reducing it to schemas.
3. Eg: email address and phone number.

E. Many-to-One OR One-to-Many

1. **Paid**: This is a *many-to-one* relationship between **Donation** and **Alumni**. Multiple donations can be made by an alumnus. However, multiple alumni can’t make the same donation. This assumes that no two or more alumnus makes a joint donation, and such cases are always differentiated and entered into the database.
2. **Dual Major**: This is a *many-to-one* relationship between **Alumni** and **Department**. An alumnus could have done a double major in one department, and many alumni could have done that double major. So, it’s many-to-one from alumni to department.
3. **F.A.**: This is a *one-to-many* relationship between **Instructor** and **Alumni**. A particular alumnus will have one final faculty advisor. A particular instructor could have been the FA to many alumni. So, it’s a Many-To-One relationship from Alumni to Instructor.

F. many-many:

1. **Lived:** (residence and alumni)
   This is a *many-to-many* relationship between **Residence** and **Alumni** because a hostel room may have belonged to multiple alumni over their college years and an Alumni may have lived in multiple different hostel rooms.

2. **Teaches:** (instructor, courses)
   This is a *many-to-many* relationship between **Instructor** and **Courses** because instructor takes multiple courses, and a course is also taken by multiple instructors

3. **Takes:** (courses, alumni)

This is a *many-to-many* relationship between **Courses** and **Alumni** because each alumni takes multiple courses, and each course is taken by multiple alumni.

G.

1. **Takes:**
   Courses: Total
   Alumni: Total
   Each alumnus must have taken at least one course. Each course has been offered to at least one student/batch.
2. **Scholar_Don:**
   Donation: Partial
   Scholarship: Partial
   Not all donations will be for scholarships. Not all scholarships will be funded by donations.
3. **Participates:**
   Extracurricular: Total
   Alumni: Partial
   Extracurricular activities exist because students did them so it will always be total.
   However, not all students may engage in such activities.

Responsibilities of Group 2

Answer 1, 2, 3, 4:
Entities Relational Schemas:

- Alumni (Roll_Number, First_Name, Middle_Name, Sur_Name, Branch, CPI , Degree , Picture , Income, Guardian_Name, Personal_Email, College_Email, Date_of_Birth , Year_of_Graduation , JEE_Rank , Relationship_Status , Street_No , City , State , Country , Pincode)

  PRIMARY KEY = Roll_Number

  Constraints:

  Name - NOT NULL (Name of the alumni must be present to have a coherent database)

  College_Email - UNIQUE (Every Email ID is unique)

  Personal_Email - UNIQUE (Every Email ID is unique)

- Mobile_Number (Roll Number , Contact Number)

  PRIMARY KEY = (Roll_Number, Contact_Number)

  FOREIGN KEY = Roll_Number

  Constraints:

  Mobile_Number - CHECK(<10000000000 and >= 1000000000)

- Department (Dept_Name, No_of_Faculties)

  PRIMARY KEY = Dept_Name

- Courses (Course_ID, Course_Name, Credits)

  PRIMARY KEY = Course_ID

  Constraints:

  Course_Name - NOT NULL (Name of the alumni must be present to have a coherent database)

- UNIQUE (Every Course should have a unique name)

- Instructor (Name, Instructor_ID, Work_Email)

  PRIMARY KEY = Instructor_ID

  Constraints:

  Name - NOT NULL (Name of the instructor must be present to have a coherent database)

  Work_Email - UNIQUE (Every Email ID is unique)

- Projects (Title , Duration , Outcome)

  PRIMARY KEY = Title

  Constraints:

  Duration - CHECK(>0) (Project Duration should be greater than 0)

- Donation (Transaction ID, Date, Amount)

  PRIMARY KEY = Transaction ID

  Constraints:

  Amount - NOT NULL (No sense of Donation if the amount is not mentioned)

- Companies (Company_Name, Company_Domain)

  PRIMARY KEY = Company_Name

- Education (Degree, Discipline, Institution_Name, City)

  PRIMARY KEY = (Degree , Discipline , Institution_Name)

  Constraints:

  City - NOT NULL (There must necessarily be a city, suppose one wants to filter the database based on alumni studying in specific cities)

- Residence (Room_Number , Hostel_Name)

  PRIMARY KEY = (Room No. , Hostel Name)

  Constraints:

  Room_No - CHECK(Room_No >100 and Room_No<500) (Room numbers in IITGN only range in between these numbers since all hostels have only 4 floors)

- Scholarship (Scholarship_Name , Amount)

  PRIMARY KEY = Scholarship Name

  Constraints:

  Amount - NOT NULL (No sense of Scholarship if the amount is not mentioned)

- Eligibility (Scholarship Name , Eligibility)

  PRIMARY KEY = (Scholarship Name , Eligibility)

  FOREIGN KEY = Scholarship Name

- Extra-Curricular (Name, Domain)

  PRIMARY KEY = Name

- Visits(Roll_Number, Purpose , Date)

  Visits was a weak entity set with Partial Key (Discriminator) = (Purpose , Date). While reducing, primary key of the owner entity set was added i,e Roll_Number

  PRIMARY KEY = Roll_Number

- Achievements (Roll_Number, Purpose , Date, Description)

Achievements was a weak entity set with Partial Key (Discriminator) = (Purpose , Date, Description). While reducing, primary key of the owner entity set was added Roll_Number

PRIMARY KEY = Roll_Number

- Job Posting (<u>Post_Id</u>, Role , Salary, Location)

  Primary Key = <u>Post_ID</u>

Relationship Sets Relational Schemas:

- Paid(<u>Transaction_ID</u>, Roll_Number)
  PRIMARY KEY = Transaction_ID
  FOREIGN KEY = Transaction_ID, Roll_Number
  WHY IT IS NEEDED- To establish a connection between an alumnus and all the donations he/she has made

  MAPPING CARDINALITY- Multiple donations can be made by an alumnus. However, multiple alumni can't make the same donation. This assumes that no two or more alumnus makes a joint donation, and such cases are always differentiated and entered into the database. So, this is a Many-to-one relationship.

  As the relation between *Donation* and *Alumni* is Many-to-one, only Transaction_ID is the primary key.

- Scholar_Don (<u>Transaction_ID</u>, <u>Scholarship_Name</u>)
  PRIMARY KEY = (Transaction_ID, Scholarship_Name)
  FOREIGN KEY = (Transaction_ID, Scholarship_Name)
  WHY IT IS NEEDED- To establish a connection between all scholarships which have been funded via donations.

  MAPPING CARDINALITY-

As the relation between *Donation* and *Scholarship* is Many-to-Many, both Transaction_ID and Scholarship_name are the primary key.

- Works_In (Roll_Number, Company_Name, Role, Salary, Start_year , End_year)
  PRIMARY KEY = (Roll_Number, Company_Name, Role)
  FOREIGN KEY = (Roll_Number, Company_Name)
  Constraints:
  Start_year = NOT NULL (Start Year needs to be defined to ensure validity)

  WHY IT IS NEEDED- To establish a connection between alumni and the companies they are working/have worked in and the roles they are working at.

  MAPPING CARDINALITY- A particular alumnus could have worked in many companies in his/her career, and a particular company could have multiple IITGN alumni as employees. Also, a particular alumnus could have taken multiple roles in the same company with career growth, so the Role attribute is also required to uniquely identify a relationship in the Works_In relationship set.

  As the relation between *Alumni* and *Companies* is Many-to-Many, Roll number, Company name, Role are the primary key.

- Got (Roll_Number, Scholarship Name, Year)
  PRIMARY KEY = Roll_Number, Scholarship_Name, Year
  FOREIGN KEY = Roll_Number, Scholarship_Name
  WHY IT IS NEEDED- To establish a relationship between alumni and the scholarships they received as students.

  MAPPING CARDINALITY- An alumnus, while he/she was a student, could have received multiple scholarships. The same scholarship (name) could have been received by multiple alumni. So, it’s a many-to-many relationship. The same alumnus could have gotten the same scholarship over multiple years. So, Year is necessary to uniquely identify a relationship in the Got relationship set.

As the relation between *Alumni* and *Scholarship* is Many-to-Many, Roll_Number, Scholarship_Name, Year are the primary key.

- Studied_In (Roll_Number, Name, Discipline, Degree, Start_year , End_year)
  PRIMARY KEY = (Roll_Number, Name, Discipline, Degree)
  FOREIGN KEY = Roll_Number, (Name, Discipline, Degree)
  Constraints:
  Start_year = NOT NULL (Start Year needs to be defined to ensure validity)

  WHY IT IS NEEDED- To link alumni with all the other educational institutions they have studied in.

  MAPPING CARDINALITY- A particular alumnus could have studied in more than one universities after graduation from IITGN, and multiple alumni could have studied in the same university. So, it’s a many-to-many relationship.

  As the relation between *Alumni* and *Education* is Many-to-Many, Roll_Number, Name, Discipline, Degree are the primary key.

- Dual Major(Roll_Number, Dept_Name)
  PRIMARY KEY = Roll_Number
  FOREIGN KEY = Roll_Number, Dept_Name
  WHY IT IS NEEDED- To link those alumni who have taken a double major to the department in which they took the double major.

  MAPPING CARDINALITY- An alumnus could have done a double major in one department and that double major could have been done by many alumni. So, it’s many-to-one from alumni to department.

As the relation between *Alumni* and *Department* is Many-to-One, Roll_Number are the primary key.

- Minor(Roll_Number, Dept_Name)
  PRIMARY KEY = (Roll_Number, Dept_Name)
  FOREIGN KEY = Roll_Number, Dept_Name
  WHY IT IS NEEDED- To link those alumni who have taken minor(s) to the department(s) in which they took the double major.

  MAPPING CARDINALITY- An alumnus could have done a minor in many departments and multiple alumni could have done a minor in any given department. So, it is many-to-many

  As the relation between *Alumni* and *Department* is Many-to-Many, Roll number, Dept_name are the primary key.

- Takes (Roll_Number , Course ID)
  PRIMARY KEY = (Roll_Number, Course_ID)
  FOREIGN KEY = Roll_Number, Course_ID
  WHY IT IS NEEDED- To establish a connection between alumni and the courses they took as students.

  MAPPING CARDINALITY- An alumnus could take multiple courses and a particular course could have been taken by multiple alumni. So, it is many-to-many.

  As the relation between *Alumni* and *Courses* is Many-to-Many, Roll number, Course_ID are the primary key.

- Belongs_To (Instructor_ID, Dept_Name)
  PRIMARY KEY = (Instructor_ID, Dept_Name)
  FOREIGN KEY = Instructor_ID, Dept_Name

WHY IT IS NEEDED-To establish a connection between instructors and the departments they are associated with.

MAPPING CARDINALITY-

An instructor could be part of multiple departments and one department could have multiple instructors.

As the relation between *Instructor* and *Department* is Many-to-Many, Instructor_ID, Dept_Name are the primary key.

- Project_guide(Roll_Number, Title, Instructor_ID)
  PRIMARY KEY = (Roll_Number, Title, Instructor_ID)
  FOREIGN KEY = Roll_Number, Title , Instructor_ID
  WHY IT IS NEEDED- To establish a link between alumni and instructors and the projects they made together. This way we can understand the project work an alumnus did, who he/she did it with, and what the work was.

  MAPPING CARDINALITY- Many to Many from (Alumni,Instructor) to Project. To intuitively express the cardinality of a ternary relationship, I am clubbing the Alumni and Instructor entities and expressing an (A,B) to C cardinality.

  A particular instructor-alumnus pair could have worked on multiple projects. A particular project could have been worked on by multiple (Alumni, Instructor) pairs where either alumni could be changing (group project) or multiple instructors or both instructors and alumni are changing.

  As the relation between (*Alumni, Instructor)* and *Projects* is Many-to-Many, Roll number, Company name, Role are the primary key.

- F.A.(Roll_Number, Instructor_ID)
  PRIMARY KEY = Roll_Number

FOREIGN KEY = Roll_Number, Instructor_ID
WHY IT IS NEEDED- To establish a link between each alumnus and his/her faculty advisor.

MAPPING CARDINALITY- A particular alumnus will have one final faculty advisor. A particular instructor could have been the FA to many alumni. So, it's a Many-To-One relationship from Alumni to Instructor.

As the relation between *Alumni* and *Instructor* is Many-to-One from Alumni to Instructor, Roll number(many-side) is the primary key.

- Lived (Roll_Number , Hostel_Name, Room_Number, Semester)
  PRIMARY KEY = (Roll_Number , Hostel_Name, Room_Number, Semester)
  FOREIGN KEY = Roll_Number , (Hostel_Name, Room_Number)
  Constraints:
  Semester = CHECK(<=8 and >=1) (Only valid range of Semesters)

  WHY IT IS NEEDED- To establish a link between alumni and all the room(s) they have lived in during their stay at IITGN.

  MAPPING CARDINALITY- A particular alumnus could have lived in multiple different rooms over the course of degree, and the room could have been occupied by different alumni. So, the relationship is Many-to-Many. A particular alumnus could have lived in the same room over multiple semesters, so to uniquely identify an entity in the "lived" relationship, we need to include the Semester as a part of the primary key.

  As the relation between *Alumni* and *Residence* is Many-to-Many, Roll_Number , Hostel_Name, Room_Number, Semester are the primary key.

- Teaches(Course_ID, Instructor_ID)
  PRIMARY KEY = (Course_ID, Instructor_ID)
  FOREIGN KEY = Course_ID, Instructor_ID

WHY IT IS NEEDED-

MAPPING CARDINALITY- Many to Many. An instructor could be teaching multiple courses. A particular course could be taught by multiple instructors. The primary key is a union of primary keys of Instructor and Courses.

As the relation between *Course and Instructor* is Many-to-Many, Course_ID, Instructor_ID are the primary key.

- Participates (Roll_Number, Name, Role, Achievements)
  PRIMARY KEY = (Roll_Number, Name, Role, Achievements)
  FOREIGN KEY = Roll_Number, Name
  WHY IT IS NEEDED-To link alumni to the various extracurricular activities they were part of as students.

  MAPPING CARDINALITY- The relationship between Alumni and Extracurricular activities is Many-to-Many because one alumnus can participate in multiple extracurricular activities and a particular activity can be undertaken by multiple alumni. So, the primary key is a union of the primary keys of both relationships as well as the roles and the different achievements that an alumnus makes in an activity. Alumnus could have multiple achievements.

  As the relation between *Alumni* and *Extra-curriculars* is Many-to-Many, Roll_Number, Name, Role, Achievements are the primary key.

- Don_Extracurricular (Transaction_ID, Name)
  PRIMARY KEY = Transaction_ID
  FOREIGN KEY = Transaction_ID, Name
  WHY IT IS NEEDED- To connect a donation(s) with the extracurricular activity(s) that it was used to fund

MAPPING CARDINALITY- Many to many. This depends on the purpose that a donation is used for. It’s possible that a large donation amount is used to fund multiple extra-curricular activities. It’s also possible that a particular extracurricular activity is funded via multiple donation amounts.

As the relation between *Donation* and *Extra Curriculars* and is Many-to-Many, Transaction_ID, Name are the primary key.

- Role_In (<u>Post_ID</u>, Company_Name)
  PRIMARY KEY = Post_ID
  FOREIGN KEY = Post_ID, Company_Name

  WHY IT IS NEEDED- To describe the relationship between a company and a particular job posting within that company. A company may make multiple job postings each with their Post_ID and so, the relationship between Companies and Job Postings is One-To-Many

  MAPPING CARDINALITY - One-To-Many from Company to Job Postings.

  As the relation between *Job Posting* and *Companies* is Many-to-One, Post_ID are the primary key.

- Posts (<u>Post_ID</u>, Roll_Number)
  PRIMARY KEY = Post_ID
  FOREIGN KEY = Post_ID, Roll_Number

  WHY IT IS NEEDED- To describe the relation between an alumnus and the job postings that he/she makes. A particular alumnus may make multiple job postings, so it’s a One-To-Many mapping.

  MAPPING CARDINALITY- One-To-Many from Alumni to Job_Postings

As the relation between *Alumni* and *Posts* is One-to-Many, Post_ID are the primary key.

- Achieved (dropped, reason given below)
- Visited(dropped, reason given below)

Answer 5:

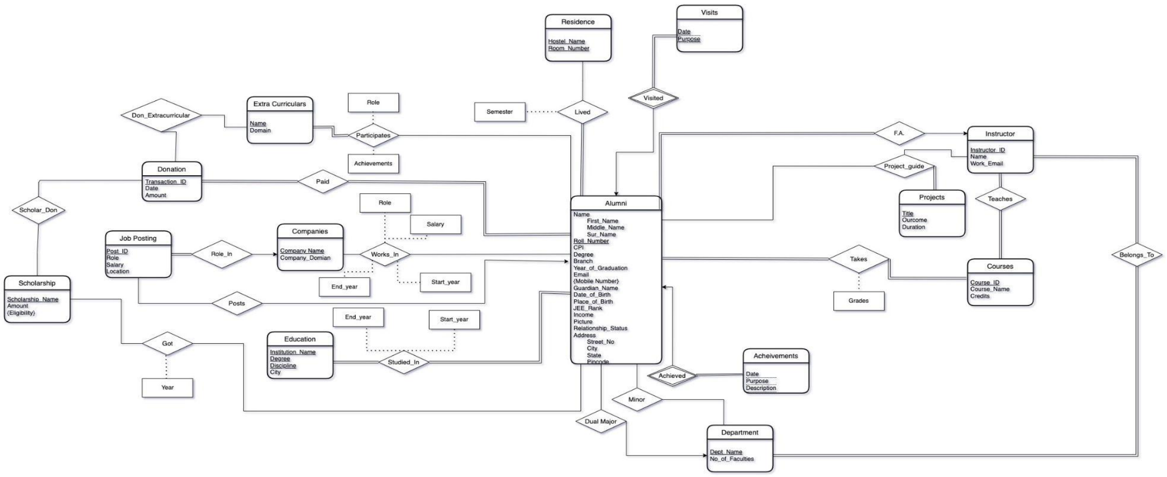
There were no entity sets which were removed from the final design schema. Two relationship sets were removed from the final design as they were linking strong entity sets to weak entity sets. They were redundant, as all their attributes would be the same as the final reduction of weak entity sets in relation schema.

1) Relationship set achieved was dropped. It would have all the attributes of entity achievements (Discriminator Keys) as well as the primary key of students i.e. Roll_Number. However these will also be the attributes of the Achievement relation after the reduction. Therefore it will be redundant.

2) Relationship set visited was dropped. It would have all the attributes of entity visits (Discriminator Keys) as well as the primary key of students i.e. Roll_Number. However these will also be the attributes of the Visits relation after the reduction. Therefore it will be redundant.

Entity-Relationship Diagram

Link to the draw.io project:
https://drive.google.com/file/d/1i43KJjNNeOrZvOedPRNlJhb0BjXi4pEX/view?usp=sharing



Contributions

Dhairya Shah: Member of G1

- Leading the team, arranging the meetings with stakeholders, and ideating with them about the use cases of the database.
- Listing down the entities, relationships, attributes according to the use cases and formulating the E-R diagram from them.
- Formulated Answer 1 for Group 1.
- Worked on designing the ER Diagram on draw.io

Inderjeet Singh Bhullar : Member of G2

- Ideated the needs for entities, attributes and relationships in order to conceptualize the ER Diagram for our project.
- Worked on designing the ER Diagram on draw.io
- Worked on converting the ER diagram to Relational Schema.
- Helped in deciding the primary and foreign keys of various entity sets and relationship sets.

Kalash Kankaria: Member of G2

- Worked on drawing the ER Diagram on draw.io.

- Responsible for deciding all the constraints and key specifications in our relational schemas.
- Worked on deciding the primary and foreign keys for different entity sets and relationship sets.
- Helped in resolving the ER diagram into relational schemas.

Rahul Chembakasseril: Member of G2

- Involved in gaining insights from relevant stakeholders about the needs of their database.
- Ideating over the relevant entities in the database and the best set of attributes we could store.
- Conceptualizing and sketching out the ER diagram and iteratively improving it.
- Inferring the mapping cardinalities and the participation constraints among the relationship sets.
- Contributed to reduction of ER diagram to Relational schemas.

Kanishk Singhal: Member of G2

- Part of conversion of ER diagram to Relational Schema (Relationship sets)
- Brainstorming about entities and attributes in the ER diagram
- Part of putting together the ER diagram with relations, attributes, keys and cardinalities
- Figuring entitites as strong and weak with simple and complex attributes

Bhavesh Jain: Member of G2

- Brainstorming and formulating the ER diagram including attributes, entities, and relations.
- Converting ER diagram to relation schema.
- Responsible for looking after the redundancies and dropping redundancies.
- Helped in deciding primary keys and foreign keys, and analysing cardinalities.
- Deciding the structure and attributes of final relationship sets after reduction to relation schema.

Medhansh Singh: Member of G1

- Brainstorming and deciding the entities, attributes and relations in the ER diagram.
- Worked on deciding cardinality of relationships and total/partial participation in the relationships.
- Worked on deciding primary and foreign keys for attributes.
- Came up with the justification for cardinalities and participation of entities in the relationship.

Nokzendi Aier: Member of G1

- Brainstorming and deciding the entities, attributes and relations for our ER diagram.
- Worked on the cardinality of relationships, and total/partial participation in the relationships.
- Primary and foreign keys for entities.
- Justification for the cardinalities, and the participation of entities in the relationship.

Harshvardhan Vala: Member of G1

- Brainstormed together the information that is necessary for the alumni database.
- Met with the stakeholders (Alumni Relations, IIT Gandhinagar) and got the requirements and use cases. Took into account the use cases of the stakeholders and ensured that appropriate
  information was included in the database.
- Distributed the information into different entities and attributes for the database. Then finally formed the ER diagram for the database taking into account the constraints and criteria mentioned in the assignment were met.
- Also worked on draw.io for the final ER diagram with appropriate symbols and pointers.

Joy Makwana: Member of G1

- Brainstormed the entity, attributes, and relations in the E-R diagram
- Worked on E-R diagram
- Identified primary keys and foreign keys for all the entities
- Found the cardinality of various entities and given the appropriate justification
- Helped in finding participation constraints

References

- https://www2.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter2/node7.html
- https://drawio-app.com/tutorials/
- https://www.researchgate.net/figure/Entity-relationship-diagram-of-the-system_fig2_282532213