Indian Institute of Technology Gandhinagar

Lab Management System

CS432 Assignment 1
29th January, 2024

Group Members

Kaushal Kothiya, 21110107
Sachin Jalan, 21110183
Manav Jain, 22110140
Anish Karnik, 21110098
Husain Malwat, 21110117
Srujan Kumar Shetty, 21110214
Ameya Tajne, 21110220

Responsibilities of G1:

Answer 1:

The lab management database streamlines academic and research operations by efficiently organizing information on lab equipment availability, laboratory schedules, and student details, including courses and projects, as well as professors' timetables, research projects, and lab grants, enabling universities and institutions to enhance class scheduling, staff assignments, and fund management for a more effective academic environment.

The lab management database at IIT Gandhinagar significantly enhances efficiency and transparency, optimizing resource allocation, providing real-time information for students to plan lab sessions effectively, enabling instructors to manage commitments and showcase contributions, and fostering collaborations with external organizations for a more organized and collaborative academic environment.

Primary stakeholders included lab administrators, whose insights were crucial in identifying specific requirements for efficient lab management. Through extensive discussions with administrators from key labs, we gained valuable information on their existing database systems and potential areas for improvement. Student involvement was essential as end-users, and their feedback through surveys and focus group discussions helped shape a user-friendly interface. We identified that most of the entries were done manually, so we wish to make the pipeline completely online to facilitate the students and lab admins in easily maintaining and accessing the lab.

Functional requirements encompass the specific capabilities of the system through which it serves its desired purpose. Our Lab Management system has functional requirements such as:

- Lab inventory management
- Lab booking system
- Student, Professor, and Staff Profiles
- Financial Tracking for Labs and Equipment
- Managing lab grants from organizations
- Keeping track of issued lab equipment from students

Answer 2:

1. Mention any kind of database that the lab staff keeps for lab operations.
    - Issue Register, Inventory, Employee Leave (Supin Bhai, Electrical Lab1)
    - Machine Specific Attenders, Different sections for different tools (Babloo Sir, Carpentry Trade Operator)

2. Mention the records that you keep online and offline. Do you feel the need to digitize the records that you keep offline?
    - Issue Register, Inventory, Employee Leave (Supin Bhai, Electrical Lab1)
    - Mostly, all the records are stored online. Employee recruitment is the only thing that happens offline in records. (Tushar Bhai, Electrical Lab1)
3. The equipment that students issue is mostly borrowed using QR codes. Name the details that students are asked in the QR code
    - Name, Email, Equipment name, quantity, Purpose, Return Date, Issue Date,Mobile Number, Discipline and Batch (Tarun Sharma, Mechanical Lab)
4. Do professors who buy equipment for the labs keep in the same database?
    - The equipment which is bought by professors are kept in the same database but we keep a record of professor bought equipment by labeling. (Tarun Sharma, Mechanical Lab)
5. What are the details needed to be stored while purchasing an item for labs?
    - Keeping the list of items, invoicing and purchases, billing details etc. (Pragnesh Bhai, Electrical Lab2)
6. Do billing details include attributes like vendor name, manufacturerâs name, etc?
    - Yes, also it includes the address and other minor details. (Pragnesh Bhai, Electrical Lab2)
7. Are the perishable and consumable items treated differently to maintain the database?
    - Yes, they are labeled differently as perishable and consumable items. (Tushar Bhai, Electrical Lab1)
8. Do students other than the IITGN students also visit the labs?
    - There are NEEV students, and students from nearby local colleges who visit the lab. (Raju Beerasant, Physics Lab)
9. What are the personal details of lab staff stored in the database?
    - Name, Email, Contacts, Salary, Children Education Allowance, Medical Insurance, Employee Leaves. (Pragnesh Bhai, Electrical Lab2)
10. Do you keep records of people who can access the lab?
    - We keep a record of students working under the professor and only those can access the labs. (Alok, Electron Microscope Lab)
11. How do you keep track of the updates that take place periodically?
    - We keep track of each computer in the computer lab like accessories, LAN connections, Softwares installed in them. At each maintenance date, we check all the PCs in the computer lab. (Rahulendra Bhaskar, IMS)
12. For labs receiving grants, what details are to be stored?
    - Year of Purchase, Equipment Details and quantity, Name and Contact details of Organization giving grants. (Prof. Joycee Mekie, VLSI Lab)

Other people with whom we interacted:

- Rohit, Electron Microscope Lab
- Naresh, Nanomaterials and Nano Fluidics Lab

Responsibilities of G1 and G2:

Answer 1:

Entities and attributes:

1. **Student** with attributes (Name, Roll_Number, Degree, Batch, Amount_Due, Purpose)
2. **Inventory** with attributes (ID, Equipment_Name, Price, Vendor_Details, Manufacturer_Name, Status, Lab_Name, Quantity)
3. **Contact** with attributes (Email_ID, Phone_No)
4. **Lab** with attributes (Lab_Name, Room_no, Block_no, Amount_Allocated, Working_Hours, Capacity)
5. **Staff** with attributes (Name, Employee_ID, Salary, Role)
6. **Time_Slot** with attributes(Time_ID)
7. **Course** with attributes(Course_ID, Credits, Course_Name)
8. **Project** with attributes(Project_ID, Name, Grant_Aid)
9. **Professor** with attributes(Name, Employee_ID, Office_no, Grant_Aid)
10. **Department** with attributes(Dept_Name, No_of_Faculties)

Relationships:

1. **Lab_Grant :** Relating Labs and Grants
2. **Course_Slot :** Relating Labs and their Time Slots
3. **Works_In :** Relating Staff with the Labs they are working with
4. **Lab_department :** Relating Departments with the Labs they are working in
5. **Instructor :** Relating Professor with Course they are offering
6. **Accessed_tool** : Relating Student with Inventory(items issued)with Issued Time, Quantity Issued, Return Time as descriptive attributes
7. **Stud_Department :** Relating Student to their respective Department

8. **Student_Enrolled :** Relating Students with Courses they have taken
9. **Lab_tool :** Relating Inventory with Labs with Procured Through as descriptive attributes
10. **Staff_Contact :** Relating Staff with Contact
11. **Lab_Contact :** Relating Lab with Contact
12. **Prof_Department :** Relating Professor with their Department
13. **Prof_Contact :** Relating Professor with Contact
14. **Proj_Taken :** Relating Professor, Student and Project (To know which professors and students are involved in projects)
15. **Contact_Details :** Relating Student with Contact

Answer 2:

C: At least one Primary key and Foreign Key:

primary key :

- Student has attribute Roll_Number as the primary key, because we designed that each student will have a unique roll number as followed by our institute. No 2 students will have the same roll number. Also, the roll number of students will be fixed for his entire tenure.
- Inventory has attribute ID as the primary key; for example, every equipment or furniture in our institute has a unique id sticker attached; the ID of the equipment never changes and has no dependency. It will be fixed and will never get repeated for 2 pieces of equipment.

Foreign key :

- The Entity Grants is a weak entity set, it has the foreign key of Lab Name. The primary key of the grants entity is (Lab Name, Received From Which Organization, Amount, When). Reasoning, multiple labs can get the same grants from the same organization hence grants will not have a primary key of their own; thus, we have the lab name as a foreign key as to which lab received that grant to make the key unique.

D: At least one one-to-one relationship:

We kept the contact details as a separate attribute because there were various entities which had contact details as their attribute. So the one-to-one relations are **Contact_Details**,

Lab_Contact, **Staff_Contact**, **Prof_Contact** between contact details and students, contact details and labs, contact details and staff, contact details professors. These relationships are one-to-one, as each student will have a single entry in contact details and also each contact will correspond to a single person be it a student, staff or professor. The primary key of the contact details entity is the institute email id, as each entity (student, professor, staff, labs) will have a unique institute email id.

E: At least one one-to-many relationship :

Following are the one to many relationships:

- **Works_In :** The relationship between the Lab entity and the staff entity is one-to-many because a lab can have multiple staff members while a staff can work only in a single lab. A staff member cannot be working in 2 labs simultaneously.
- **Lab_tool :** The relationship between Lab and inventory is one-to-many because multiple inventories can be present in a single lab, but an inventory can belong only to a single lab. As each equipment has its unique id, hence it is impossible to have equipment shared between multiple labs.
- **Accessed_tool :** The relationship between Student and inventory is one-to-many because multiple inventories can be issued by a single student but an inventory can only be issued by a single student. This design is similar to the one used in the manufacturing lab by using Google Forms.

F: At least one many-to-many relationship :

There are many many-to-many relationships we are mentioning few of them :

- **Student_Enrolled :** The relationship between student and course is many to many because, similar to our institute, many students will be taking a single course, and many courses will also be taken by a single student; hence it is a many-to-many relationship. For example, many students have taken database courses, and I have taken multiple courses.
- **Prof_Department :** The relationship between professor and department is many to many because in our institute, multiple professors are present in a single department, like CSE has multiple professors, and a single professor is also present in multiple departments, for example, joint faculty.
- **Course_Slot :** The relationship between time slots and labs is many to many because, as in our timetable, multiple labs can be scheduled in the same time slot (K1, K2

slots). And also, a single lab can be scheduled in multiple time slots (for example, manufacturing lab, VLSI lab).

G: At least one total participation or partial participation:

Given are a few examples of total participation and partial participation in our database.

- **Prof_Department :** The relationship between professor and department is of complete participation from both entities because we assume that each professor will belong to a department in our institute and each department will have at least one professor. This is true in our institute.

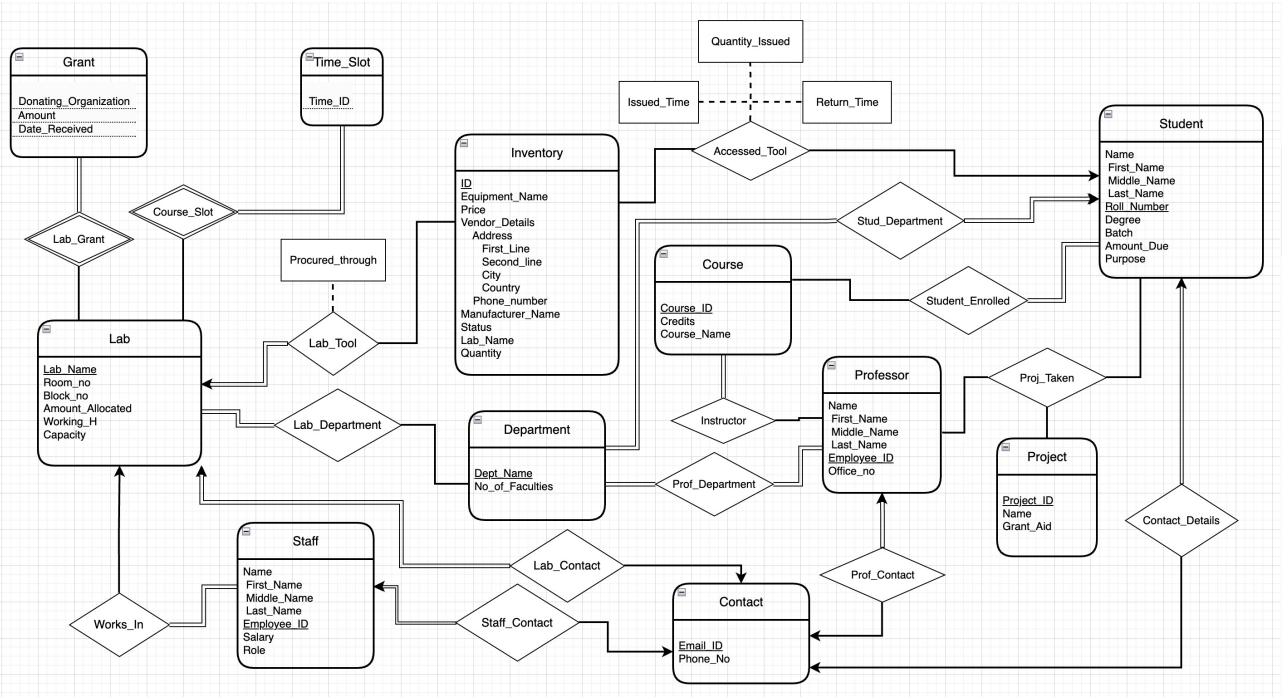
- **Stud_Department :** The relationship between student and department is of complete participation, because we have similar behavior, each student will be present in at least one department in our institute, and also each department will have at least one student. This is very realistic behavior followed at all institutes.

- **Accessed_tool :** The relationship between inventory and students is partial from both sides, because it is not necessary that every tool is being used by some student and also it is not necessary that each student is using some tool. This design choice was made by discussing the working of the labs with the stakeholders. Common examples can be computers in computer labs.

- **Works_In :** The relationship between staff and labs is partial on the side of labs and complete on the side of staff because there are some labs which are maintained by professors and there is no staff (for example major cse labs). But the relationship is of complete participation on the staff side, because each staff must be associated with some lab.

ER Diagram:

The google drive link for the same: here



Responsibility of G2:

Entities Relational Schemas:

- **Students:** (Roll_Number, First_Name, Middle_Name, Last_Name, Batch, Degree, Amount_Due, Purpose)

  PRIMARY KEY = Roll_Number

  Constraints:
  Roll_Number: NOT_NULL and UNIQUE(Roll Number must be present and a unique number)
  First_Name : NOT_NULL (First name must be present)
  Amount_Due: DEFAULT(The default value for due will be Rs. 0)

  First_Name : varchar(255)
  Middle_Name : varchar(255)
  Last_Name : varchar(255)
  Batch : integer(4)
  Degree : varchar(255)

Amount_Due : float(24)
Purpose : varchar(255)

- **Inventory:**(ID, Equipment_Name, Price, Vendor_Add_First_Line, Vendor_Add_Second_Line, Vendor_Add_City, Vendor_Add_Country, Vendor_Phone_Number, Manufacturer_Name, Status, Lab_Name, Quantity)

  PRIMARY KEY = ID FOREIGN KEY = Lab_Name

  Constraints:
  ID: NOT_NULL and UNIQUE(ID must be present and a unique number)
  Equipment_Name: NOT_NULL (Name should be present)
  Lab_Name : NOT_NULL (The designated lab name should be present)

  ID : integer (10)
  Equipment_Name : varchar(255)
  Price : float(24)
  Vendor_Add_First_Line : varchar(255)
  Vendor_Add_Second_Line : varchar(255)
  Vendor_Add_City : varchar(255)
  Vendor_Add_Country : varchar(255)
  Vendor_Phone_Number : varchar(255)
  Manufacturer's_Name : varchar(255)
  Status : varchar(255)
  Lab_Name : varchar(255)
  Quantity : integer

- **Contact:** (Email_ID, Phone_number)

  PRIMARY_KEY = Email_ID

  Constraints:
  Phone_Number: CHECK(<10000000000 and >= 1000000000)

  Email_ID : varchar(255)
  Phone_Number: integer(10)

- **Lab:** (Lab_Name, Room_no, Block_no, Amount_Allocated, Working_Hours, Capacity)

  PRIMARY_KEY = Lab_Name

  Constraints:
  Lab_Name : NOT_NULL and UNIQUE(Lab Name must be present and a unique number)
  Amount_Allocated : DEFAULT (The default value for amount allocated will be 0)
  Lab_Name : varchar(255)

Room_no : integer
Block_no : integer
Amount_Allocated : float(24)
Working_Hours: integer
Capacity: integer

- **Staff:** (Employee_ID, First_Name, Middle_Name, Last_Name, Salary, Role)

  PRIMARY_KEY = Employee_ID

  Constraints:
  Employee_ID: NOT_NULL and UNIQUE(Employee_ID must be present and a unique number)
  First_Name: NOT_NULL (First name must be present)

  Employee_ID : integer (10)
  First_Name : varchar(255)
  Middle_Name : varchar(255)
  Last_Name : varchar(255)
  Salary : float(24)
  Role : varchar(255)

- **Department:** (Dept_Name, No_of_Faculties)

  PRIMARY_KEY = Dept_Name

  Constraints:
  Dept_Name: NOT_NULL (The designated Department name should be present)
  No_of_Faculties:DEFAULT(The default value for No. of faculties will be 0)

  Dept_Name: varchar(255)
  No_of_Faculties : integer

- **Professor:** (Employee_ID, First_Name, Middle_Name, Last_Name, Grant_Aid, Office_Number)

  PRIMARY_KEY = Employee_ID

  Constraints:
  Employee_ID: NOT_NULL and UNIQUE(Employee_ID must be present and a unique number)
  First_Name: NOT_NULL (First name must be present)
  Grant_Aid: DEFAULT (As all professors donât receive grants, default value will be 0

Rs.)

Employee_ID : integer(10)
First_Name : varchar(255)
Middle_Name : varchar(255)
Last_Name : varchar(255)
Grant_Aid : float(24)
Office_Number: varchar(255)

- **Project:** (Project_ID, Name, Grant_Aid)

  PRIMARY_KEY = Project_ID

  Constraints: Project_ID: NOT_NULL and UNIQUE(Project_ID must be present and Unique) Name: NOT_NULL (The designated Project name should be present) Grant_Aid: DEFAULT (As all projects donât receive grants, the default value will be 0 Rs.)

  Project_ID: varchar(255)
  Name: varchar(255)
  Grant_Aid: float(24)

- **Course:** (Course_ID, Credits, Course_Name)

  PRIMARY KEY=Course_ID

  Constraints:
  Course_ID: NOT_NULL and UNIQUE(Course ID must be present and unique for each course. We are keeping track of courses corresponding to students and professors, which would update each semester so Course_ID can be considered unique.)

  Course_ID: varchar(255)
  Credits: integer
  Course_Name: varchar(255)

- **Time_Slot:**(Time_ID) PRIMARY KEY = (Time_ID)

  Constraints:
  Time_ID: NOT_NULL and UNIQUE(Time_ID must be present for each slot and must have some values associated)

  Time_ID: varchar(255)

- **Grants:** (Lab_Name,Donor, Donating_organization ,Amount, Receiving_Date)

  Grants was a weak entity set with partial key (Discriminator) =(Donor, Donating_organization, Amount, Receiving_Date)

While reducing the primary key of the Labs entity set(i.e. Lab_Name) was added.

PRIMARY_KEY=(Lab_Name,Donor, Donating_organization ,Amount, Receiving_Date)

Constraints:
Lab_Name: NOT_NULL and UNIQUE
Amount: NOT_NULL (Any donating organization will give some amount of money as a grant; otherwise, it wonât make any sense.)

Lab_Name: char(255)
Donor: char(255)
Donating_organization: char(255)
Amount: float(24)
Receiving_Date: date

Relationship Sets Relational Schemas:

- **Lab_Grant**(Lab_Name, Donating_Organization, Amount, Date_Received)

  PRIMARY_KEY=(Lab_Name, Donating_Organization, Amount, Date_Received)
  FOREIGN_KEY=Lab_Name

  Why It Is Needed : To link the laboratory with the organization that provided funding, specifying the amount and date of receipt.

  Mapping Cardinality:
  Many to many. Each laboratory can receive grants from multiple organizations, and donating entities may contribute grants to several labs. The total participation is from the grant-giving entities' side, as non-donating organizations are not included in the records.

- **Course_Slot**(Lab_Name,Time_ID)

  PRIMARY_KEY=(Lab_Name,Time_ID)
  FOREIGN_KEY=Lab_Name

  Why It Is Needed :To link on which time slot which lab is occupied.

  Mapping Cardinality :
  Many to many. Each Time_ID will have multiple associated labs, indicating that it's also a weak entity set. A single lab can occupy multiple time slots.

- **Works_In** (Employee_ID , Lab_Name)

  PRIMARY KEY= Employee_ID
  FOREIGN KEY= Employee_ID, Lab_Name

Why It Is Needed : To establish a connection between the staff who are employees in the lab.

Mapping Cardinality :
A person can be an employee in a single lab. However, a lab can have multiple employees. Here, we assume that a single person is not associated with two different labs. So, this relationship is a Many-to-one relationship.

The staff will have full participation in this relationship as staff without a designated lab is nugatory.

- **Lab_Department**(Dept_Name, Lab_Name)

  PRIMARY KEY= (Dept_Name, Lab_Name)
  FOREIGN KEY= Dept_Name, Lab_Name

  Why It Is Needed : To establish a connection between the departments which use/work in the labs .

  Mapping Cardinality :
  Multiple departments can use a single lab. Also, multiple labs can be used by people of a single department. Hence, this is a Many-to-Many relationship.

  Here we assume that every lab is designated with a particular department. There are no labs having no departments assigned. Thus labs will have full participation in this relationship.

- **Instructor** (Course_ID , Employee_ID)

  PRIMARY KEY= (Course_ID, Employee_ID)
  FOREIGN KEY= Course_ID, Employee_ID

  Why It Is Needed : To establish a connection between professors who are instructors of that specific course.

  Mapping Cardinality :
  A course can have multiple instructors. Also, a professor can take multiple courses. Hence, this is a Many-to-Many relationship.

  A course should always have an instructor, so it should have full participation. Assuming it is possible that a professor doesn't take any course, it has partial participation.

- **Accessed_Tool**(ID, Roll_Number, Issued_Time, Quantity_Issued, Return_Time)

  PRIMARY KEY= (ID, Roll_Number)
  FOREIGN KEY= Roll_Number

Why It Is Needed : To establish the relationship between students and the inventory, three descriptive attributes are utilized: Issued_Time (indicating when the equipment was issued), Quantity_Issued (representing the amount issued), and Return_Time (specifying when the item/s need to be returned).

Constraints:
Quantity_Issued : NOT_NULL(The issued quantity must be stored to keep the count of equipment in inventory.)

Issued_Time : date
Quantity_Issued : integer
Return_Time : date

Mapping Cardinality :
Many to one. The relation will be one on the student side because any equipment can be issued by just one student. Also, there will be many on the Inventory side because one student can issue more than one equipment.

- **Stud_Department** (Dept_Name , Roll_Number)

  PRIMARY KEY= (Dept_Name, Roll_Number)
  FOREIGN KEY= Dept_Name, Roll_Number

  Why It Is Needed : To establish a connection between students of a particular department.

  Mapping Cardinality :
  A department can have multiple students. Assuming here that a student cannot be enrolled in multiple departments, we can consider this relationship as a Many-to-One relationship.

  As a department wonât have zero students and the students will necessarily be assigned a department, both entities will have full participation.

- **Student_Enrolled** (Course_ID , Roll_Number)

  PRIMARY_KEY=(Course_ID,Roll_number)
  FOREIGN_KEY=(Course_ID,Roll_number)

  Why It Is Needed: This links which student is enrolled in which course

  Mapping Cardinality :
  Many to many. Due to the fact that one student can enroll in multiple courses and vice versa, with each course having multiple students, there is total participation from the student side as every student must be registered, while partial participation exists from the course side since there can be courses not registered by any student.

All the students will be in a specific department and all the departments would contain some number of students. Hence, there will be total participation from both the entities in this relation.

- **Prof_department**(Dept_Name, Employee_ID)

  PRIMARY KEY= (Dept_Name , Employee_ID)
  FOREIGN KEY= (Dept_Name , Employee_ID)

  Why It Is Needed :This department links professors with their respective departments.

  Mapping Cardinality :
  It is a many-to-many relationship as one professor can be part of multiple departments, and a department contains multiple professors. There is total participation from both sides as a professor will always have a department, and a department has at least one professor; otherwise, that department wonât make any sense.

  All departments have professors assigned to them and assuming that all the professors will be also related to a particular department, there will be full participation from both the entities in this relation.

- **Prof_Contact** (Employee_ID , Email_ID)

  PRIMARY KEY= Employee_ID
  FOREIGN KEY= (Roll Number, Email_ID)

  Why It Is Needed : This relationship gives contact details to link contacts with professors.

  Mapping Cardinality :
  It is a one-to-one relationship because one professor will have only one email associated with him/her and vice-versa. There will be total participation from the professor's side as every professor will have contact details.

  There will be total participation from the professorâs side, as every professor will have contact details.

- **Proj_Taken**(Employee_ID , Roll_Number , Project_ID)

  PRIMARY KEY= (Employee_ID ,Roll Number, Project_ID)
  FOREIGN KEY= (Employee_ID , Roll_Number , Project_ID)

  Why It Is Needed : This relationship links students, the professor under which student is pursuing the project and the project. It is a trinary relationship.

  Mapping Cardinality :
  This relationship is a many-to-many-to-many relationship as one student can take multiple projects under multiple professors, one professor can offer multiple projects

to multiple students and in one project many students and many professors can collaborate.

- **Contact_Details**(Roll_Number, Email_ID)

  PRIMARY KEY= Roll Number
  FOREIGN KEY= (Roll Number, Email_ID)

  Why It Is Needed :To link contacts with students, this relationship gives students contact details.

  Mapping Cardinality :
  It is a one-to-one relationship because one student will have only one email associated with him/her and vice-versa. There will be total participation from the student side as every student will have contact details.

  There will be total participation from the student's side, as every student will have contact details.

- **Lab_Tool**(ID , Lab_Name , Procured_Through)

  PRIMARY KEY=ID, Lab_Name
  FOREIGN KEY=Lab_Name

  Why It Is Needed: This links the tool from inventory with the corresponding lab. The details from whom the lab equipment were taken from inventory is also important, and they are stored in the descriptive attribute Procured_Through.

  Procured_Through : varchar(255)

  Mapping Cardinality :
  There can be multiple tools associated with a lab. However, a single tool cannot be a part of two labs at the same time. Hence, this is a Many-to-One Relationship.

  Assuming that all the labs have some equipment, there is full participation of labs. It is possible that an item from the inventory is not designated a particular lab, so it has partial participation.

- **Staff_Contact** (Employee_ID, Email_ID)

  PRIMARY KEY=Employee_ID
  FOREIGN KEY= Employee_ID

  Why It Is Needed: To provide contact details of lab staff
  Mapping Cardinality :
  Each staff member will have a unique and single contact email and vice versa. So, the relationship will be a One-to-One relationship.

There will be total participation from the lab staff's side, as every lab staff will have contact details.

- **Lab_Contact** (Lab_Name, Email_ID)

  PRIMARY KEY = (Lab_Name, Email_ID)
  FOREIGN KEY = Lab_Name

  Why It Is Needed: It provides the contact details associated with a lab.

  Mapping Cardinality :
  Every lab will have a unique email ID and phone number, the vice versa also holds true . So, the relationship will be a One-to-One relationship.

  There will be total participation for lab contact, as every lab will have contact details.

Contributions:

1. Anish Karnik:(Member of G2)
   - Brainstorming entities,their attributes and relationships between those entities.
   - Decided the primary and foreign keys.
   - Wrote the justification for mapping cardinalities and participation in the relationship.
   - Documented the questions and the answers by stakeholders asked by members of G1 team.
   - Analyzed if some relationships needed descriptive attributes, and what descriptives attributes can be assigned.
2. Kaushal Kothiya:(Member of G2)
   - Ideated about entities, attributes and relationships.
   - Decided the primary and foreign keys.
   - Wrote the constraints of the attributes in different entity and relationship sets.
   - Analyzed if some relationships needed descriptive attributes, and what descriptive attributes can be assigned.
   - Made the Latex Document.
3. Manav Jain: (Member of G1)

- Involved in gaining insights about labs from pertinent stakeholders to enhance the relevance of the database.
- Brainstormed about entities and their attributes.
- Contributed to the selection of primary keys of entities, and categorizing them as strong and weak entities.
- Helped in formulating the answers for Q1.

4. Sachin Jalan : (Member of G1)
    - Arranged meetings with the stakeholders, discussed with them the functioning of the labs and brainstormed new ideas which could be implemented.
    - Worked along with the team to decide about the entities, attributes and relationships in the database.
    - Provided justifications for total participation, partial participation, and many-to-many relationships.
    - Suggested improvements in the final E-R diagram.
5. Husain Malwat: (Member of G1)
    - Brainstormed the information necessary for the labs management database. Coordinated with stakeholders, i.e., lab personnel from diverse labs, to engage in discussions regarding challenges in the maintenance of data.
    - Worked with the team to decide about the entities, attributes, and relationships in the database.
    - Contributed in framing the answer for Q1
    - Suggested improvements in the final E-R diagram.
6. Ameya Tajne :(Member of G2)
    - Collaborated with team members to draft mapping cardinalities for Relationship Sets in Relational Schemas.
    - Contributed to determining constraints.
    - Participated in discussions regarding the necessity of various elements within the schema.
    - Engaged in brainstorming sessions to identify and define descriptive attributes.
7. Srujan Kumar Shetty: (Member of G1)
    - Ideated the list of entities, attributes and relations.

- Made the ER diagram using the online drawing software (draw.io) Visited the labs, meeting and asking questions to respective stakeholders about lab database requirements.
- Wrote the purpose of our database for Q1
- Made the Latex Document.