BU EDGE

Project on

Creation of a Database for Geologic Field Work

**Submitted to**

Md Samsuddoha

Assistant Professor

Department of Computer Science Engineering

University of Barishal

**Submitted by**

Shimul Alrazi

Batch:DB-01

ID:22

The University of Barishal has a well-established faculty of science and engineering which consists of several departments among them Geology and Mining is a well-furnished department.

It offers theoretical courses along with lab and fieldwork. Every year it conducts fieldwork

to get hand-to-hand field experience in the region of the geologic enrichment site.

To conduct fieldwork it needs proper management. Through the help of DBMS, it can run the fieldwork without any hassle.

To create DB we need some steps

1. Drawing EDR
2. Reducing to schema

Steps of drawing ERD

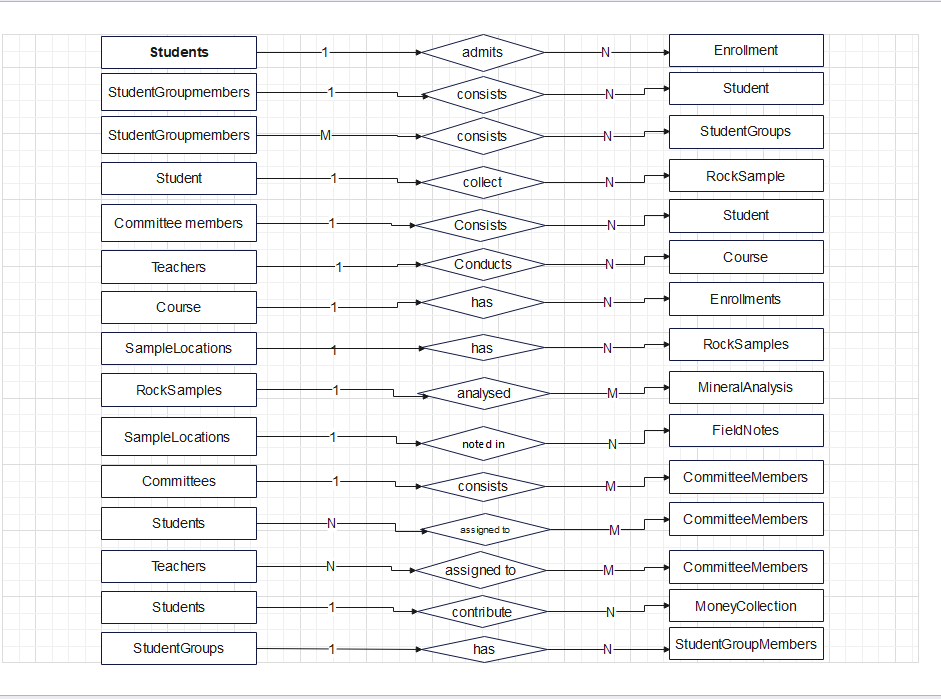
1. Identify the entities
2. Identify the attributes and primary key
3. Identify the relationship
4. Identify the condition ratio and perticipation
5. Draw diagram

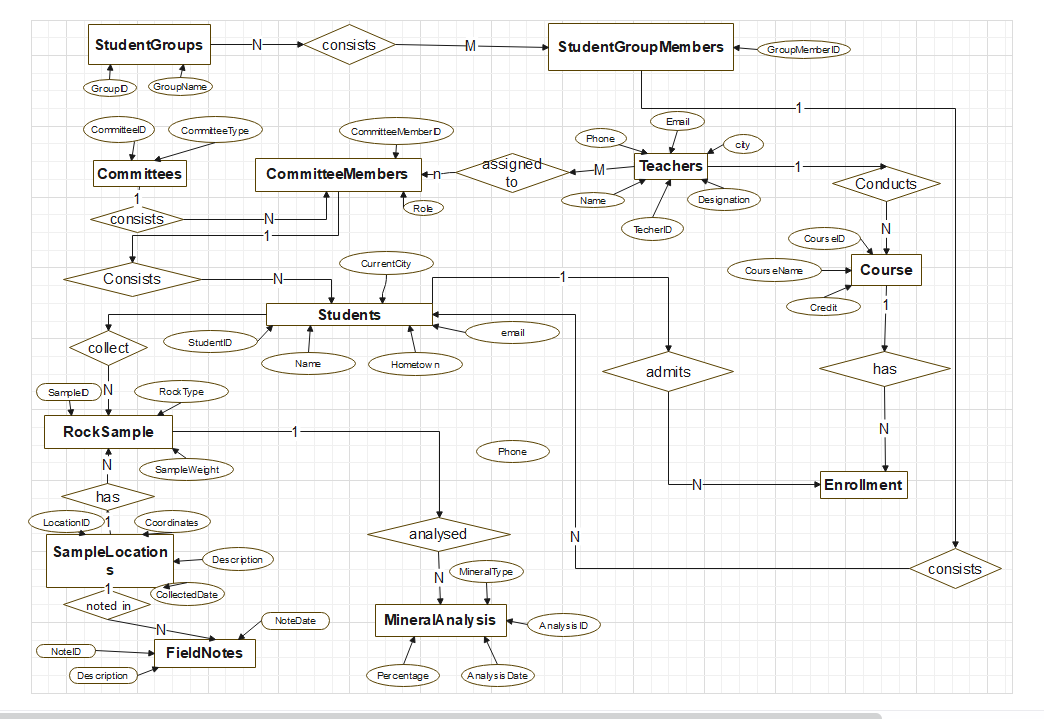
Step1+2: Identify the entities Identify the attributes and primary key

1. **Students=** (**StudentID** (Primary Key),**Name**, **Email**,**Phone, Hometown,Current\_City)**
2. **StudentGroups= (GroupID** (Primary Key),**GroupName)**
3. **StudentGroupMembers=** (**GroupMemberID** (Primary Key))
4. **Teachers=** (**TeacherID** (Primary Key), **Name**,**Email**, **Phone,Designation, City)**
5. **Courses**= (**CourseID** (Primary Key), **CourseName**,**Credits)**
6. **Table: Enrollments**= (**EnrollmentID** (Primary Key), **Grade,Date)**
7. **SampleLocations**= (**LocationID (Primary Key)**, **Coordinates, Description**

**CollectedDate)**

1. **RockSamples= (SampleID** (Primary Key), **RockType**, **SampleWeight)**
2. **MineralAnalysis=** (**AnalysisID** (Primary Key), **MineralType**, **Percentage**, **AnalysisDate**
3. **FieldNotes= (NoteID** (Primary Key), **NoteDate, Description)**
4. **Committees**= (**CommitteeID** (Primary Key), **CommitteeType**)
5. **CommitteeMembers**= (**CommitteeMemberID** (Primary Key),**Role)**
6. **FieldBudgets=** (**BudgetID** (Primary Key), **Amount**, **Description)**
7. **MoneyCollections**= (**CollectionID** (Primary Key), **Amount**, **CollectedDate)**
8. **Students admit Enrollments**: One-to-Many (Each student can enroll in multiple courses).
9. **StudentGroupMembers consists Student**:One-to-Many(Each group has many students)
10. **StudentGroupMembers consists StudentGroups:**Many to many
11. **Student collect RockSample:**One-to-Many
12. **Committee members consists Student**:One to many
13. **Teachers conducts Courses**: One-to-Many (Each teacher can teach multiple courses).
14. **Courses has Enrollments**: One-to-Many (Each course can have multiple students enrolled).
15. **SampleLocations has RockSamples**: One-to-Many (Each sample location can have multiple rock samples).
16. **RockSamples analyzed MineralAnalysis**: One-to-Many (Each rock sample can have multiple mineral analyses).
17. **SampleLocations noted FieldNotes**: One-to-Many (Each sample location can have multiple field notes).
18. **Committees cosists CommitteeMembers**: One-to-Many (Each committee can have multiple members).
19. **Students and Teachers assigned to CommitteeMembers**: Many-to-Many (Each student and teacher can be part of multiple committees and vice versa).
20. **Students contribute MoneyCollections**: One-to-Many (Each student can contribute multiple times).
21. **StudentGroups has StudentGroupMembers**: One-to-Many (Each student group can have multiple members).





SQL Query

Create DATABASE Geological\_Field\_Work\_Management\_System;

--Creating the Students table

CREATE TABLE Students (

StudentID INT PRIMARY KEY,

Name VARCHAR(100),

Email VARCHAR(100),

Phone VARCHAR(20)

);

-- Creating the StudentGroups table

CREATE TABLE StudentGroups (

GroupID INT PRIMARY KEY,

GroupName VARCHAR(100)

);

-- Creating the StudentGroupMembers table

CREATE TABLE StudentGroupMembers (

GroupMemberID INT PRIMARY KEY,

StudentID INT,

GroupID INT,

FOREIGN KEY (StudentID) REFERENCES Students(StudentID),

FOREIGN KEY (GroupID) REFERENCES StudentGroups(GroupID)

);

-- Creating the Teachers table

CREATE TABLE Teachers (

TeacherID INT PRIMARY KEY,

Name VARCHAR(100),

Email VARCHAR(100),

Phone VARCHAR(20),

City Varchar(30),

Designation VARCHAR(30)

);

-- Creating the Courses table

CREATE TABLE Courses (

CourseID INT PRIMARY KEY,

CourseName VARCHAR(100),

Credits INT,

TeacherID INT,

FOREIGN KEY (TeacherID) REFERENCES Teachers(TeacherID)

);

-- Creating the Enrollments table

CREATE TABLE Enrollments (

EnrollmentID INT PRIMARY KEY,

StudentID INT,

CourseID INT,

Grade CHAR(2),

FOREIGN KEY (StudentID) REFERENCES Students(StudentID),

FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)

);

--Creating the samplelocation table

CREATE TABLE SampleLocations (

LocationID INT PRIMARY KEY,

Latitude DECIMAL(9, 6),

Longitude DECIMAL(9, 6),

Description TEXT,

CollectedDate DATE

);

---- Creating the RockSamples table

CREATE TABLE RockSamples (

SampleID INT PRIMARY KEY,

LocationID INT,

RockType VARCHAR(100),

SampleWeight FLOAT,

CollectedBy INT,

FOREIGN KEY (LocationID) REFERENCES SampleLocations(LocationID),

FOREIGN KEY (CollectedBy) REFERENCES Students(StudentID)

);

-- Creating the MineralAnalysis table

CREATE TABLE MineralAnalysis (

AnalysisID INT PRIMARY KEY,

SampleID INT,

MineralType VARCHAR(100),

Percentage FLOAT,

AnalysisDate DATE,

FOREIGN KEY (SampleID) REFERENCES RockSamples(SampleID)

);

-- Creating the FieldNotes table

CREATE TABLE FieldNotes (

NoteID INT PRIMARY KEY,

LocationID INT,

NoteDate DATE,

NoteText TEXT,

FOREIGN KEY (LocationID) REFERENCES SampleLocations(LocationID)

);

-- Creating the Committees table

CREATE TABLE Committees (

CommitteeID INT PRIMARY KEY,

CommitteeName VARCHAR(100),

CommitteeType VARCHAR(100)

);

-- Creating the CommitteeMembers table

CREATE TABLE CommitteeMembers (

CommitteeMemberID INT PRIMARY KEY,

CommitteeID INT,

MemberID INT,

Role VARCHAR(100),

FOREIGN KEY (CommitteeID) REFERENCES Committees(CommitteeID)

);

-- Creating the FieldBudgets table

CREATE TABLE FieldBudgets (

BudgetID INT PRIMARY KEY,

Amount DECIMAL(10, 2),

Description TEXT

);

-- Creating the MoneyCollections table

CREATE TABLE MoneyCollections (

CollectionID INT PRIMARY KEY,

StudentID INT,

Amount DECIMAL(10, 2),

CollectedDate DATE,

FOREIGN KEY (StudentID) REFERENCES Students(StudentID)

);

-- Inserting values into Students

INSERT INTO Students (StudentID, Name, Email, Phone) VALUES

(1, 'John Doe', 'johndoe@example.com', '123-456-7890'),

(2, 'Jane Smith', 'janesmith@example.com', '098-765-4321'),

(3, 'Alice Johnson', 'alicejohnson@example.com', '555-555-5555');

-- Inserting values into StudentGroups

INSERT INTO StudentGroups (GroupID, GroupName) VALUES

(1, 'Group A'),

(2, 'Group B'),

(3, 'Group C');

-- Inserting values into StudentGroupMembers

INSERT INTO StudentGroupMembers (GroupMemberID, StudentID, GroupID) VALUES

(1, 1, 1),

(2, 2, 2),

(3, 3, 3);

-- Inserting values into Teachers

INSERT INTO Teachers (TeacherID, Name, Email, Phone, City, Designation) VALUES

(1, 'Dr. Smith', 'drsmith@example.com', '111-222-3333' , 'Barishal' , 'Associate Professsor'),

(2, 'Prof. Johnson', 'profjohnson@example.com', '444-555-6666' , 'Dhaka' , 'Professor'),

(3, 'Dr. Williams', 'drwilliams@example.com', '777-888-9999' 'Rangpur' , 'Lecturer');

-- Inserting values into Courses

INSERT INTO Courses (CourseID, CourseName, Credits, TeacherID) VALUES

(1, 'Geology 101', 3, 1),

(2, 'Mineralogy', 4, 2),

(3, 'Petrology', 3, 3);

-- Inserting values into Enrollments

INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES

(1, 1, 1, 'A'),

(2, 2, 2, 'B'),

(3, 3, 3, 'A');

-- Inserting values into SampleLocations using converted decimal degrees

INSERT INTO SampleLocations (LocationID, Latitude, Longitude, Description, CollectedDate) VALUES

(1, 40.4461, -79.9822, 'Observation point 1', '2024-01-01'),

(2, 34.0522, -118.2437, 'Observation point 2', '2024-02-01'),

(3, 51.5074, -0.1278, 'Observation point 3', '2024-03-01');

-- Inserting values into RockSamples

INSERT INTO RockSamples (SampleID, LocationID, RockType, SampleWeight, CollectedBy) VALUES

(1, 1, 'Granite', 5.5, 1),

(2, 2, 'Sandstone', 2.3, 2),

(3, 3, 'Claystone', 4.1, 3);

-- Inserting values into MineralAnalysis

INSERT INTO MineralAnalysis (AnalysisID, SampleID, MineralType, Percentage, AnalysisDate) VALUES

(1, 1, 'Quartz', 60.0, '2024-04-01'),

(2, 2, 'Feldspar', 25.0, '2024-05-01'),

(3, 3, 'Mica', 15.0, '2024-06-01');

-- Inserting values into FieldNotes

INSERT INTO FieldNotes (NoteID, LocationID, NoteDate, NoteText) VALUES

(1, 1, '2024-01-01', 'Observed granite formations.'),

(2, 2, '2024-02-01', 'Sandy area with some quartz grains.'),

(3, 3, '2024-03-01', 'Claystone samples collected.');

-- Inserting values into Committees

INSERT INTO Committees (CommitteeID, CommitteeName, CommitteeType) VALUES

(1, 'Food Committee', 'Food'),

(2, 'Core Committee', 'Core'),

(3, 'Transport Committee', 'Transport');

-- Inserting values into CommitteeMembers

INSERT INTO CommitteeMembers (CommitteeMemberID, CommitteeID, MemberID, Role) VALUES

(1, 1, 1, 'Member'),

(2, 2, 2, 'Member'),

(3, 3, 3, 'Member');

-- Inserting values into FieldBudgets

INSERT INTO FieldBudgets (BudgetID, Amount, Description) VALUES

(1, 10000.00, 'Budget for food and accommodation.'),

(2, 5000.00, 'Budget for transport.'),

(3, 3000.00, 'Budget for equipment.');

-- Inserting values into MoneyCollections

INSERT INTO MoneyCollections (CollectionID, StudentID, Amount, CollectedDate) VALUES

(1, 1, 100.00, '2024-01-01'),

(2, 2, 150.00, '2024-02-01'),

(3, 3, 200.00, '2024-03-01');

