**Group Members**

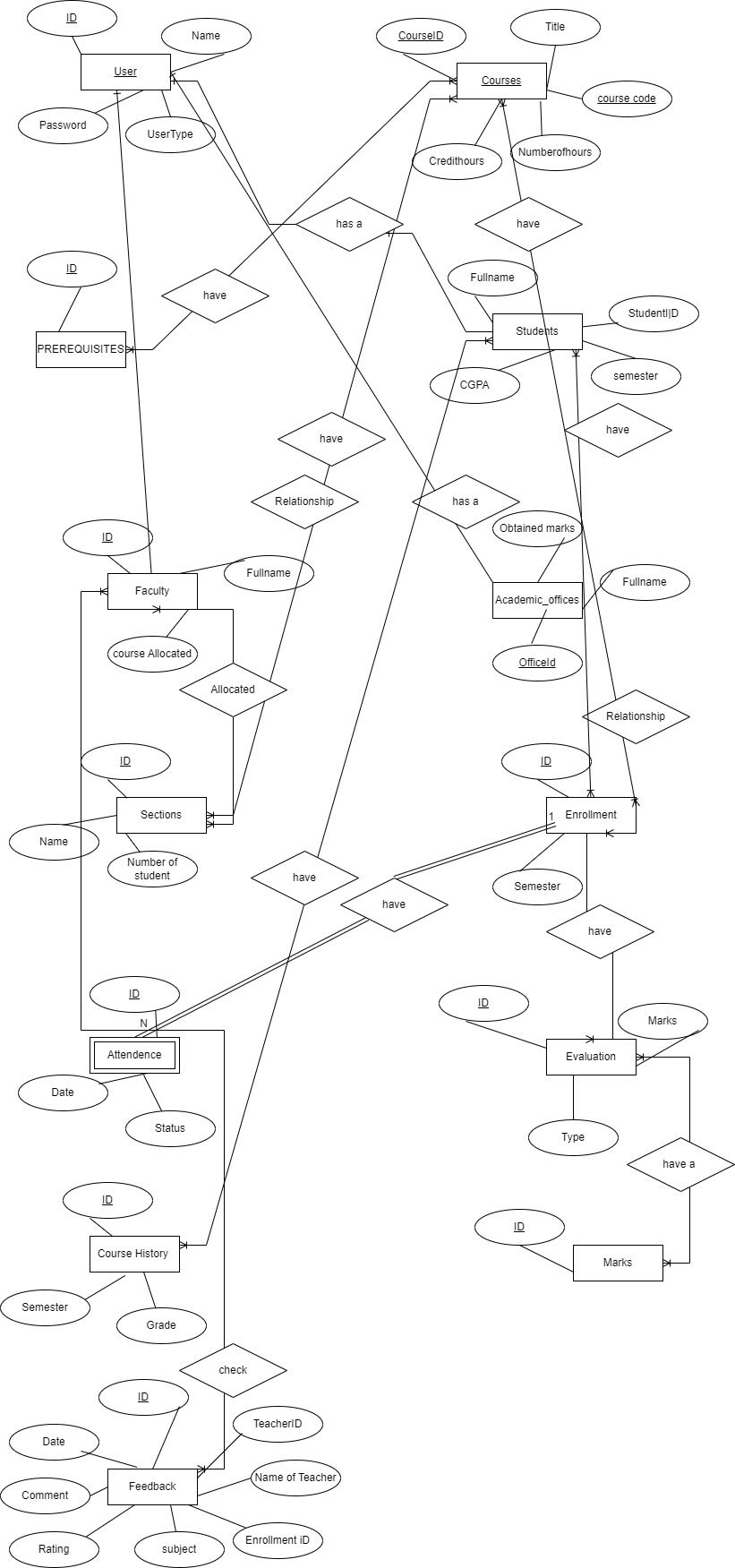
**20I-0854 Umais**

**20i-0840 Abdul Musawir**

**20i-0730 Bilal Ahmad**

**DB Report**

**ERD**

****

**Schema**

-- Create the flex database

CREATE DATABASE flex;

USE flex;

-- Create the USERS table

CREATE TABLE USERS (

UserId INT PRIMARY KEY,

UserName VARCHAR(255),

Password VARCHAR(255),

UserType VARCHAR(50)

);

-- Create the COURSES table

CREATE TABLE COURSES (

CourseId INT PRIMARY KEY,

CourseCode VARCHAR(50),

CourseTitle VARCHAR(255),

CreditHours INT,

NumberOfHours INT

);

-- Create the PREREQUISITES table

CREATE TABLE PREREQUISITES (

CourseId INT,

PrerequisiteCourseId INT,

PRIMARY KEY (CourseId, PrerequisiteCourseId),

FOREIGN KEY (CourseId) REFERENCES COURSES(CourseId),

FOREIGN KEY (PrerequisiteCourseId) REFERENCES COURSES(CourseId)

);

-- Create the STUDENTS table

CREATE TABLE STUDENTS (

StudentId INT PRIMARY KEY,

UserId INT,

FullName VARCHAR(255),

CGPA DECIMAL(3, 2),

FOREIGN KEY (UserId) REFERENCES USERS(UserId)

);

-- Create the FACULTY table

CREATE TABLE FACULTY (

FacultyId INT PRIMARY KEY,

UserId INT,

FullName VARCHAR(255),

Courses\_Alloted INT,

FOREIGN KEY (UserId) REFERENCES USERS(UserId)

);

-- Create the ACADEMIC\_OFFICE table

CREATE TABLE ACADEMIC\_OFFICE (

OfficeId INT PRIMARY KEY,

UserId INT,

FullName VARCHAR(255),

FOREIGN KEY (UserId) REFERENCES USERS(UserId)

);

-- Create the SECTIONS table

CREATE TABLE SECTIONS (

SectionId INT PRIMARY KEY,

CourseId INT,

FacultyId INT,

StudentCount INT,

FOREIGN KEY (CourseId) REFERENCES COURSES(CourseId),

FOREIGN KEY (FacultyId) REFERENCES FACULTY(FacultyId)

);

alter table SECTIONS

add Sec\_name varchar(50)

-- Create the ENROLLMENTS table

CREATE TABLE ENROLLMENTS (

EnrollmentId INT PRIMARY KEY,

StudentId INT,

CourseId INT,

Semester VARCHAR(50),

FOREIGN KEY (StudentId) REFERENCES STUDENTS(StudentId),

FOREIGN KEY (CourseId) REFERENCES COURSES(CourseId),

);

-- Create the ATTENDANCES table

CREATE TABLE ATTENDANCES (

AttendanceId INT PRIMARY KEY,

EnrollmentId INT,

Date DATE,

Status VARCHAR(50),

FOREIGN KEY (EnrollmentId) REFERENCES ENROLLMENTS(EnrollmentId)

);

-- Create the EVALUATIONS table

CREATE TABLE EVALUATIONS (

EvaluationId INT PRIMARY KEY,

EnrollmentId INT,

EvaluationType VARCHAR(50),

Marks DECIMAL(5, 2),

FOREIGN KEY (EnrollmentId) REFERENCES ENROLLMENTS(EnrollmentId)

);

--CREATE the course history table

CREATE TABLE COURSE\_HISTORY (

HistoryId INT PRIMARY KEY,

StudentId INT,

CourseId INT,

Semester VARCHAR(50),

Grade VARCHAR(10),

FOREIGN KEY (StudentId) REFERENCES STUDENTS(StudentId),

FOREIGN KEY (CourseId) REFERENCES COURSES(CourseId)

);

delete from COURSE\_HISTORY

-- Create the MARKS table

CREATE TABLE MARKS (

MarkId INT PRIMARY KEY,

EvaluationId INT,

ObtainedMarks DECIMAL(5, 2),

FOREIGN KEY (EvaluationId) REFERENCES EVALUATIONS(EvaluationId)

);

-- Create the FEEDBACK table

CREATE TABLE FEEDBACK (

FeedbackId INT PRIMARY KEY,

EnrollmentId INT,

TeacherId INT,

Date DATE,

Subject VARCHAR(100),

NameOfTeacher VARCHAR(100),

Q1Rating INT,

Q2Rating INT,

Q3Rating INT,

Q4Rating INT,

Q5Rating INT,

Q6Rating INT,

Q7Rating INT,

Q8Rating INT,

Q9Rating INT,

Q10Rating INT,

Q11Rating INT,

Q12Rating INT,

Q13Rating INT,

Q14Rating INT,

Q15Rating INT,

Q16Rating INT,

Q17Rating INT,

Q18Rating INT,

Q19Rating INT,

Q20Rating INT,

Comments VARCHAR(MAX),

FOREIGN KEY (EnrollmentId) REFERENCES ENROLLMENTS (EnrollmentId),

FOREIGN KEY (TeacherId) REFERENCES FACULTY (FacultyId)

);

delete from SECTIONS

BULK INSERT USERS

**Mapping Relationship detail**

1. **USERS:** Stores user information, including their ID, username, password, and user type.
2. **COURSES:** Contains information about courses, such as the course ID, course code, title, credit hours, and number of hours.
3. PREREQUISITES: Represents the prerequisites for courses. It has a composite primary key consisting of CourseId and PrerequisiteCourseId, and foreign key references to the COURSES table.
4. STUDENTS: Stores student-specific details like student ID, user ID, full name, and CGPA. It has a foreign key reference to the USERS table.
5. FACULTY: Contains information about faculty members, including faculty ID, user ID, full name, and the number of courses allotted to them. It also has a foreign key reference to the USERS table.
6. ACADEMIC\_OFFICE: Represents the academic office with details like office ID, user ID, and full name. It has a foreign key reference to the USERS table.
7. SECTIONS: Stores information about course sections, including section ID, course ID, faculty ID, student count, and section name. It has foreign key references to the COURSES and FACULTY tables.
8. ENROLLMENTS: Represents student enrollments in courses. It includes an enrollment ID, student ID, course ID, and semester. It has foreign key references to the STUDENTS and COURSES tables.
9. ATTENDANCES: Stores attendance records for students. It has an attendance ID, enrollment ID, date, and status. It references the ENROLLMENTS table.
10. EVALUATIONS: Contains evaluation details for students, such as evaluation ID, enrollment ID, evaluation type, and marks. It references the ENROLLMENTS table.
11. COURSE\_HISTORY: Stores the history of courses taken by students. It includes a history ID, student ID, course ID, semester, and grade. It has foreign key references to the STUDENTS and COURSES tables.
12. MARKS: Represents marks obtained by students in evaluations. It includes a mark ID, evaluation ID, and obtained marks. It references the EVALUATIONS table.
13. FEEDBACK: Stores feedback provided by students to teachers. It includes a feedback ID, enrollment ID, teacher ID, date, subject, teacher's name, ratings for various questions, comments, and references to the ENROLLMENTS and FACULTY tables
14. The "ATTENDANCES" entity depends on the existence of an "ENROLLMENT" record. It cannot exist without an associated "ENROLLMENT" record. The primary key of the "ATTENDANCES" table includes the foreign key "EnrollmentId", which references the primary key of the "ENROLLMENTS" table.

# Screen Shots:

A screenshot of a login form

Description automatically generated with medium confidenceA screenshot of a login form

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with low confidenceA screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence