University Student Management System
(Ege Tan - Kutluhan Berke Kılıçkaya - Efe Sözer)

This database system is designed to manage various aspects of an educational institution, encompassing student information, courses, faculty, departments, classroom infrastructure, attendance, and student projects. There are 8 entities , which are Student , Course , Professor , Department , Attendance , Building , Room , Project. **Student** entity stores personal and academic information about students. Each record includes a unique StudentID, the student's Name, and Grade Point Average (GPA).**Course** entity represents the academic courses offered by the institution. Each course has a unique CourseID and cname. **Professor** entity contains information on faculty members, including a unique ProfessorID, the professor's Name.**Department** entity defines the various academic departments within the institution. Each department is identified by a unique DepartmentID and has a Name that describes its academic focus. **Attendance** entity tracks student attendance in courses. Each record, identified by a unique AttendanceID, is linked to a specific Student and Course (via foreign keys) and the student's Attendance Status (Present/Absent).**Building** entity represents the physical structures on campus. Each building is identified by a unique BuildingID and includes information on the Building's Name. **Room** entity details the individual rooms within buildings, useful for scheduling and resource allocation. Each room has a unique RoomID, is associated with a Building via a foreign key, and includes a RoomNumber. **Project** entity captures information on student projects, including a unique ProjectID and pdeadline.

Relationships:

-Student-Course Enrollment: Illustrates a many-to-many relationship where a single student can enroll in multiple courses,also each course can be enrolled by any student .Each course must have at least one student.

-Course-Attendance Have: A one-to-many relationship, indicating that each course can have multiple attendance records that shows if the course has full capacity, but each attendance record is specific to one course. Additionally since attendance is not mandatory in some courses, there is no participation constraint.

-Professor-Course Teaching: A many-to-many relationship is realized through a junction table (not explicitly mentioned), allowing professors to teach multiple courses and courses to be taught by multiple professors. But each course must have a professor, therefore there is a participation constraint.

-Department-Professor Include: A one-to-many relationship, where a department can include multiple professors, but each professor is associated with only one department. In terms of to be a department, there should be at least one professor.

-Courses-Building Hold_in: Many-to-one relationships that allow multiple course instances to be associated with a single building, but each course instance is associated with only one building at any given time.

-Building-Room Include: A one-to-many relationship, where a building can contain multiple rooms, but each room is located in a single building. Each building must have at least one room.

-Student-Project Assigned: A many-to-one relationship, allowing students to work on multiple projects, but each project is associated with a single student.There is no participation constraint for this relation.

Code:

```
CREATE TABLE Students (
    studentid INT,
    sname VARCHAR(255),
    sgpa DECIMAL(3, 2),
    PRIMARY KEY (studentid)
);

CREATE TABLE Courses (
    courseid INT,
    cname VARCHAR(255),
    UNIQUE(cname),
    PRIMARY KEY (courseid)

);

CREATE TABLE Enroll(
  studentid INT,
  courseid INT,
  PRIMARY KEY (studentid, courseid),
  FOREIGN KEY (studentid) REFERENCES Students(studentid) ON DELETE CASCADE,
  FOREIGN KEY (courseid) REFERENCES Courses(courseid) ON DELETE CASCADE
);
```

```sql
CREATE TABLE Project_Assigned(
   projectid INT,
   pdeadline DATE,
   studentid INT,
   PRIMARY KEY (projectid),
   FOREIGN KEY (studentid) REFERENCES Students(studentid) ON DELETE CASCADE
);


CREATE TABLE Departments (
    departmentid INT,
    dname VARCHAR(255),
    PRIMARY KEY (DepartmentID),
    UNIQUE(departmentid)
);


CREATE TABLE Professors(
    professorid INT,
    departmentid INT NOT NULL,
    pname VARCHAR(255),
    PRIMARY KEY (professorid),
    FOREIGN KEY (departmentid) REFERENCES Departments(departmentid) ON DELETE CASCADE
);
```

```sql
CREATE TABLE Room_include (
    roomid INT,
    buildingid INT,
    PRIMARY KEY (roomid),
    FOREIGN KEY (buildingid) REFERENCES Buildings(buildingid) ON DELETE CASCADE
);



CREATE TABLE Course_holdin (
    courseid INT ,
    coursename VARCHAR(255),
    buildingid INT,
    PRIMARY KEY (courseid),
    FOREIGN KEY (buildingid) REFERENCES Buildings(buildingid) ON DELETE CASCADE
);
```

studentid	sname	sgpa
1	John Doe	3.75
2	Jane Smith	3.90
3	Alice Johnson	3.60
4	Bob Williams	3.80
5	Eva Brown	3.95
6	Charlie Davis	3.70
7	Grace Lee	3.85
8	David Taylor	3.45
9	Sophia White	3.92
10	Michael Jackson	3.55
NULL	NULL	NULL

courseid	cname
7	Biology
6	Chemistry
2	Computer Science
9	Economics
5	English
4	History
1	Mathematics
3	Physics
10	Political Science
8	Psychology
NULL	NULL

studentid	courseid
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
NULL	NULL

projectid	pdeadline	studentid
1	2022-03-01	1
2	2022-03-02	2
3	2022-03-03	3
4	2022-03-04	4
5	2022-03-05	5
6	2022-03-06	6
7	2022-03-07	7
8	2022-03-08	8
9	2022-03-09	9
10	2022-03-10	10
NULL	NULL	NULL

department...	dname
1	Computer Science
2	Mathematics
3	Physics
4	History
5	English
6	Chemistry
7	Biology
8	Psychology
9	Economics
10	Political Science
NULL	NULL

professorid	departmentid	pname
1	1	Dr. Smith
2	2	Dr. Johnson
3	3	Dr. Davis
4	4	Dr. Williams
5	5	Dr. Brown
6	6	Dr. Lee
7	7	Dr. Taylor
8	8	Dr. White
9	9	Dr. Jackson
10	10	Dr. Moore
NULL	NULL	NULL

professorid	courseid
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
NULL	NULL

attendanceid	courseid	astatus
1	1	1
2	2	1
3	3	1
4	4	0
5	5	1
6	6	0
7	7	1
8	8	0
9	9	1
10	10	1
NULL	NULL	NULL

buildingid	bname
1	Building A
2	Building B
3	Building C
4	Building D
5	Building E
6	Building F
7	Building G
8	Building H
9	Building I
10	Building J
NULL	NULL

roomid	buildingid
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
NULL	NULL

courseid	coursename	buildingid
1	Math101	1
2	CS200	2
3	Physics301	3
4	History101	4
5	English200	5
6	Chemistry301	6
7	Biology200	7
8	Psychology101	8
9	Economics200	9
10	PoliticalScience301	10
NULL	NULL	NULL

```sql
INSERT INTO Students (studentid, sname, sgpa) VALUES
(1, 'John Doe', 3.75),
(2, 'Jane Smith', 3.90),
(3, 'Alice Johnson', 3.60),
(4, 'Bob Williams', 3.80),
(5, 'Eva Brown', 3.95),
(6, 'Charlie Davis', 3.70),
(7, 'Grace Lee', 3.85),
(8, 'David Taylor', 3.45),
(9, 'Sophia White', 3.92),
(10, 'Michael Jackson', 3.55);

INSERT INTO Courses (courseid, cname) VALUES
(1, 'Mathematics'),
(2, 'Computer Science'),
(3, 'Physics'),
(4, 'History'),
(5, 'English'),
(6, 'Chemistry'),
(7, 'Biology'),
(8, 'Psychology'),
(9, 'Economics'),
(10, 'Political Science');

INSERT INTO Enroll (studentid, courseid) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8),
(9, 9),
(10, 10);

INSERT INTO Project_Assigned (projectid, pdeadline, studentid) VALUES
(1, '2022-03-01', 1),
(2, '2022-03-02', 2),
(3, '2022-03-03', 3),
(4, '2022-03-04', 4),
(5, '2022-03-05', 5),
(6, '2022-03-06', 6),
(7, '2022-03-07', 7),
(8, '2022-03-08', 8),
(9, '2022-03-09', 9),
(10, '2022-03-10', 10);
```

```sql
INSERT INTO Departments (departmentid, dname) VALUES
(1, 'Computer Science'),
(2, 'Mathematics'),
(3, 'Physics'),
(4, 'History'),
(5, 'English'),
(6, 'Chemistry'),
(7, 'Biology'),
(8, 'Psychology'),
(9, 'Economics'),
(10, 'Political Science');
```

```sql
INSERT INTO Professors (professorid, departmentid, pname) VALUES
(1, 1, 'Dr. Smith'),
(2, 2, 'Dr. Johnson'),
(3, 3, 'Dr. Davis'),
(4, 4, 'Dr. Williams'),
(5, 5, 'Dr. Brown'),
(6, 6, 'Dr. Lee'),
(7, 7, 'Dr. Taylor'),
(8, 8, 'Dr. White'),
(9, 9, 'Dr. Jackson'),
(10, 10, 'Dr. Moore');
```

```sql
INSERT INTO Teaching (professorid, courseid) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8),
(9, 9),
(10, 10);
```

```sql
INSERT INTO Attendance_Have (attendanceid, courseid, astatus) VALUES
(1, 1, true),
(2, 2, true),
(3, 3, true),
(4, 4, false),
(5, 5, true),
(6, 6, false),
(7, 7, true),
(8, 8, false),
(9, 9, true),
(10, 10, true);
```

```sql
INSERT INTO Buildings (buildingid, bname) VALUES
(1, 'Building A'),
(2, 'Building B'),
(3, 'Building C'),
(4, 'Building D'),
(5, 'Building E'),
(6, 'Building F'),
(7, 'Building G'),
(8, 'Building H'),
(9, 'Building I'),
(10, 'Building J');

INSERT INTO Room_include (roomid, buildingid) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8),
(9, 9),
(10, 10);

INSERT INTO Course_holdin (courseid, coursename, buildingid) VALUES
(1, 'Math101', 1),
(2, 'CS200', 2),
(3, 'Physics301', 3),
(4, 'History101', 4),
(5, 'English200', 5),
(6, 'Chemistry301', 6),
(7, 'Biology200', 7),
(8, 'Psychology101', 8),
(9, 'Economics200', 9),
(10, 'PoliticalScience301', 10);
```