**University Management System**

This system will manage all core academic data in a university including students, faculty, courses, departments, enrollments, exams, grades, and more.

**Requirements:**

**Students should be able to:**

* Register and enroll in courses
* View their results and fee status

**Faculty should be able to:**

* Assign grades
* View course assignments
* Take attendance

**Admin should be able to:**

* Manage students, faculty, courses
* Schedule classes
* View reports

**Entities (Tables):**

**1. Students**

* StudentID (PK)
* FirstName
* LastName
* Email (Unique)
* Phone
* Gender
* DOB
* DepartmentID (FK)
* Address

**2. Faculty**

* FacultyID (PK)
* FirstName
* LastName
* Email (Unique)
* Phone
* Designation
* DepartmentID **(FK)**

**3. Departments**

* DepartmentID (PK)
* DepartmentName (Unique)
* Building
* HeadOfDepartmentID(FK to FacultyID, optional)

**4. Courses**

* CourseID (PK)
* CourseName
* CreditHours
* DepartmentID(FK)
* FacultyID (FK — teacher assigned to course)

**5. Enrollments**

* EnrollmentID (PK)
* StudentID (FK)
* CourseID (FK)
* EnrollmentDate

**6. Results**

* ResultID (PK)
* StudentID (FK)
* CourseID (FK)
* ExamID (FK)
* Grade
* Remarks

**7. Exams**

* ExamID (PK)
* CourseID (FK)
* ExamType (Midterm, Final, Quiz)
* ExamDate
* TotalMarks

**8. Assignments**

* AssignmentID (PK)
* CourseID (FK)
* FacultyID (FK)
* Title
* Description
* DueDate

**9. Classrooms**

* ClassroomID (PK)
* RoomNumber
* Capacity
* Building

**10. Schedules**

* ScheduleID (PK)
* CourseID (FK)
* FacultyID (FK)
* ClassroomID (FK)
* Day
* StartTime
* EndTime

**11. Fees**

* FeeID (PK)
* StudentID (FK)
* Amount
* DueDate
* PaidDate
* Status(Paid, Unpaid, Partial)

**12. Attendance**

* AttendanceID (PK)
* StudentID (FK)
* CourseID (FK)
* Date
* Status (Present, Absent, Late)

**13. Admins**

* AdminID (PK)
* Username (Unique)
* PasswordHash
* Role
* Email

**Normalization:**

Every entity in the database is normalized to 3nf already.

**Assumptions:**

1. **Department–Faculty**
   * Relation Name: *"Has Faculty"*
   * Type: One-to-Many (1:N)
   * Description: A department can have multiple faculty members, but each faculty belongs to one department.
2. **Department–Course**
   * Relation Name: *"Offers Course"*
   * Type: One-to-Many (1:N)
   * Description: Each department offers several courses; a course belongs to one department.
3. **Faculty–Course**
   * Relation Name: *"Teaches Course"*
   * Type: One-to-Many (1:N)
   * Description: A faculty member can teach multiple courses, but each course is taught by one faculty member.
4. **Student–Enrollment**
   * Relation Name: *"Enrolls In"*
   * Type: One-to-Many (1:N)
   * Description: A student can enroll in many courses. Each enrollment record is linked to one student.
5. **Course–Enrollment**
   * Relation Name: *"Has Enrollment"*
   * Type: One-to-Many (1:N)
   * Description: A course can have multiple enrolled students. Each enrollment is for one course.
6. **Enrollment–Result**
   * Relation Name: *"Produces Result"*
   * Type: One-to-One (1:1) or One-to-Many (1:N)
   * Description: Each enrollment can result in one or multiple results, depending on assessment type.
7. **Exam–Result**
   * Relation Name: *"Generates Result"*
   * Type: One-to-Many (1:N)
   * Description: One exam can produce results for many students. Each result is tied to one exam.
8. **Student–Result**
   * Relation Name: *"Receives Result"*
   * Type: One-to-Many (1:N)
   * Description: A student can have multiple exam results. Each result belongs to one student.
9. **Course–Assignment**
   * Relation Name: *"Has Assignment"*
   * Type: One-to-Many (1:N)
   * Description: Each course can have multiple assignments. Each assignment is for one course.
10. **Student–Assignment (via Submission)**

* Relation Name: *"Submits Assignment"*
* Type: Many-to-Many (M:N)
* Description: Many students can submit many assignments. A Submissions table is required here.

1. **Course–Schedule**

* Relation Name: *"Has Schedule"*
* Type: One-to-Many (1:N)
* Description: Each course can have multiple scheduled sessions.

1. **Classroom–Schedule**

* Relation Name: *"Holds Schedule"*
* Type: One-to-Many (1:N)
* Description: Each classroom can host many scheduled sessions. Each schedule uses one classroom.

1. **Schedule–Attendance**

* Relation Name: *"Has Attendance"*
* Type: One-to-Many (1:N)
* Description: Each scheduled session can have many student attendance records.

1. **Student–Attendance**

* Relation Name: *"Marks Attendance"*
* Type: One-to-Many (1:N)
* Description: A student has multiple attendance records.

1. **Student–Fee**

* Relation Name: *"Pays Fee"*
* Type: One-to-Many (1:N)
* Description: Each student can have multiple fee payments.

1. **Admin–Department *(Optional)***

* Relation Name: *"Manages Department"*
* Type: One-to-Many (1:N)
* Description: An admin may manage multiple departments.

1. **Admin–Faculty/Student *(Optional)***

* Relation Name: *"Manages User"*
* Type: One-to-Many (1:N)
* Description: An admin can manage many users.

**Functional Dependencies:**

**1. Student**

**Attributes:** StudentID, Name, Email, Phone, DepartmentID **FDs:**

* StudentID → Name, Email, Phone, DepartmentID
* Email → StudentID

**2. Department**

**Attributes:** DepartmentID, DepartmentName, Location **FDs:**

* DepartmentID → DepartmentName, Location

**3. Faculty**

**Attributes:** FacultyID, Name, Email, DepartmentID **FDs:**

* FacultyID → Name, Email, DepartmentID
* Email → FacultyID

**4. Course**

**Attributes:** CourseID, CourseName, Credits, DepartmentID **FDs:**

* CourseID → CourseName, Credits, DepartmentID

**5. Enrollment**

**Attributes:** EnrollmentID, StudentID, CourseID, Semester, Grade **FDs:**

* EnrollmentID → StudentID, CourseID, Semester, Grade
* StudentID, CourseID → Grade

**6. Classroom**

**Attributes:** RoomID, BuildingName, Capacity **FDs:**

* RoomID → BuildingName, Capacity

**7. Schedule**

**Attributes:** ScheduleID, CourseID, FacultyID, RoomID, TimeSlot **FDs:**

* ScheduleID → CourseID, FacultyID, RoomID, TimeSlot

**8. Attendance**

**Attributes:** AttendanceID, StudentID, ScheduleID, Date, Status **FDs:**

* AttendanceID → StudentID, ScheduleID, Date, Status
* StudentID, ScheduleID, Date → Status

**9. Exam**

**Attributes:** ExamID, CourseID, Date, TotalMarks **FDs:**

* ExamID → CourseID, Date, TotalMarks

**10. Result**

**Attributes:** ResultID, StudentID, ExamID, MarksObtained **FDs:**

* ResultID → StudentID, ExamID, MarksObtained
* StudentID, ExamID → MarksObtained *(alternative key)*

**11. Admin**

**Attributes:** AdminID, Name, Email, Role  
**FDs:**

* AdminID → Name, Email, Role
* Email → AdminID

**12. Library**

**Attributes:** BookID, Title, Author, Publisher, AvailableCopies **FDs:**

* BookID → Title, Author, Publisher, AvailableCopies

**Bottom of Form**