**Entities and Tables**

1. **Users**
   * Stores data for all users (students, tutors, and admins).
   * **Attributes:**
     + user\_id (PK, INT, AUTO\_INCREMENT)
     + first\_name (VARCHAR)
     + last\_name (VARCHAR)
     + email (VARCHAR, UNIQUE)
     + password (VARCHAR, ENCRYPTED)
     + role (ENUM: 'student', 'tutor', 'admin')
     + created\_at (DATETIME)
     + updated\_at (DATETIME)
2. **Courses**
   * Information about programming courses.
   * **Attributes:**
     + course\_id (PK, INT, AUTO\_INCREMENT)
     + course\_name (VARCHAR)
     + description (TEXT)
     + difficulty\_level (ENUM: 'Beginner', 'Intermediate', 'Advanced')
     + language (VARCHAR) — e.g., Python, Java, C++
     + duration (INT, in hours)
     + created\_at (DATETIME)
3. **Classes**
   * Tracks individual class sessions conducted by tutors.
   * **Attributes:**
     + class\_id (PK, INT, AUTO\_INCREMENT)
     + course\_id (FK, References Courses)
     + tutor\_id (FK, References Users)
     + class\_date (DATE)
     + class\_time (TIME)
     + duration (INT, in minutes)
     + meeting\_link (VARCHAR)
     + recording\_link (VARCHAR, NULLABLE)
     + created\_at (DATETIME)
4. **Enrollments**
   * Manages student enrollments into courses.
   * **Attributes:**
     + enrollment\_id (PK, INT, AUTO\_INCREMENT)
     + student\_id (FK, References Users)
     + course\_id (FK, References Courses)
     + enrollment\_date (DATETIME)
     + status (ENUM: 'active', 'completed', 'dropped')
     + progress (FLOAT, percentage)
5. **Assignments**
   * Contains assignments or coding challenges for each course.
   * **Attributes:**
     + assignment\_id (PK, INT, AUTO\_INCREMENT)
     + course\_id (FK, References Courses)
     + title (VARCHAR)
     + description (TEXT)
     + due\_date (DATETIME)
     + created\_at (DATETIME)
6. **Submissions**
   * Tracks assignment submissions by students.
   * **Attributes:**
     + submission\_id (PK, INT, AUTO\_INCREMENT)
     + assignment\_id (FK, References Assignments)
     + student\_id (FK, References Users)
     + submitted\_at (DATETIME)
     + code\_link (VARCHAR)
     + grade (FLOAT, NULLABLE)
     + feedback (TEXT, NULLABLE)
7. **Payments**
   * Tracks payments made by students for enrolling in courses.
   * **Attributes:**
     + payment\_id (PK, INT, AUTO\_INCREMENT)
     + student\_id (FK, References Users)
     + course\_id (FK, References Courses)
     + amount (FLOAT)
     + payment\_date (DATETIME)
     + payment\_status (ENUM: 'success', 'failed', 'pending')
     + transaction\_id (VARCHAR, UNIQUE)
8. **Feedback**
   * Allows students to provide feedback for tutors or courses.
   * **Attributes:**
     + feedback\_id (PK, INT, AUTO\_INCREMENT)
     + student\_id (FK, References Users)
     + tutor\_id (FK, References Users)
     + course\_id (FK, References Courses)
     + rating (INT, range 1-5)
     + comment (TEXT)
     + created\_at (DATETIME)
9. **Certificates**
   * Certificates issued to students upon course completion.
   * **Attributes:**
     + certificate\_id (PK, INT, AUTO\_INCREMENT)
     + student\_id (FK, References Users)
     + course\_id (FK, References Courses)
     + issued\_date (DATETIME)
     + certificate\_link (VARCHAR)

**Relationships**

1. **Users → Courses → Enrollments**
   * Students enroll in courses.
   * Tutors create and manage courses.
2. **Courses → Classes**
   * Each course has multiple classes.
3. **Classes → Users (tutors)**
   * Classes are conducted by tutors.
4. **Courses → Assignments → Submissions**
   * Courses have assignments; students submit solutions.
5. **Users → Payments**
   * Students make payments for course enrollments.
6. **Users → Feedback → Courses/Tutors**
   * Students provide feedback for courses and tutors.
7. **Courses → Certificates → Users**
   * Certificates are awarded to students who complete courses.