## University Course Management System - Detailed Specification

**1. Overview**

\* A mobile application designed to streamline the management of university courses for students. It provides a centralized platform to access course materials, submit assignments, receive grades, and communicate with instructors and peers. The app aims to enhance the learning experience and improve student outcomes.

**2. Target Audience**

\* Undergraduate and graduate students  
\* University instructors and teaching assistants

**3. Key Features**

* **3.1 User Authentication and Roles**
  + **Authentication:**
    - Secure login and registration using university credentials (or integration with university authentication systems).
      * The system should support industry-standard authentication protocols (e.g., OAuth 2.0, OpenID Connect) for seamless integration with existing university systems.
      * Multi-factor authentication (MFA) can be implemented for enhanced security.
    - Password recovery and reset functionality.
      * Users should be able to recover their accounts using email or SMS verification.
      * Password reset processes should adhere to security best practices, such as requiring strong passwords and using secure tokens.
    - Session management.
      * The system should manage user sessions securely, with appropriate timeouts and session invalidation mechanisms.
      * Support for single sign-on (SSO) to allow users to access the app with their university credentials.
  + **Roles:**
    - **Student:** Access courses, view materials, submit assignments, check grades, participate in discussions.
      * Students can view course catalogs, search for courses, and manage their enrolled courses.
      * Personalized dashboards should display relevant course information, deadlines, and announcements.
      * Students can track their progress in each course and receive notifications for updates.
    - **Instructor:** Create and manage courses, upload materials, create and grade assignments, post announcements, facilitate discussions, communicate with students.
      * Instructors can create new courses, define course structure, and manage course settings.
      * A rich text editor should be provided for creating and formatting course materials and assignments.
      * Instructors can monitor student activity, track attendance, and generate reports on student performance.
    - **Admin (Optional):** Manage users, courses, and system settings. (May not be a core feature for the initial project scope)
      * System administrators can manage user accounts, roles, and permissions.
      * Administrators can configure system-wide settings, such as authentication methods, notification preferences, and data storage.
      * The system should provide tools for monitoring system health, performance, and security.
* **3.2 Course Management**
  + **Course Browsing:**
    - Students can view a list of available courses, including details such as course name, description, instructor, schedule, and prerequisites.
      * The course catalog should be searchable by keywords, course code, subject area, and instructor.
      * Filtering options should allow students to narrow down courses based on criteria such as level, term, and delivery method.
      * Course details pages should provide comprehensive information, including learning outcomes, syllabus, and required textbooks.
    - Search and filter functionality.
      * Implement search suggestions and auto-completion to help students find courses quickly.
      * Provide advanced filtering options, such as filtering by course credits, meeting times, and campus location.
  + **Course Enrollment:**
    - Students can enroll in courses.
      * The app should support different enrollment methods, such as self-enrollment, instructor approval, and enrollment through the university's registration system.
      * The system should handle enrollment limits, waitlists, and enrollment deadlines.
    - Integration with the university's enrollment system (if available) or a simplified enrollment process within the app.
      * If direct integration is not possible, the app should provide a user-friendly interface for students to initiate enrollment requests.
      * The app should synchronize enrollment data with the university's system to ensure consistency.
  + **Course Content Delivery:**
    - Instructors can upload and organize course materials, including documents, lecture notes, videos, and presentations.
      * Support for a wide range of file formats, including PDF, DOCX, PPTX, MP4, and others.
      * Instructors should be able to embed external resources, such as YouTube videos, interactive simulations, and online articles.
    - Support for various file formats.
      * The system should automatically convert files to formats suitable for web and mobile viewing.
      * Provide clear indicators of file sizes and download times.
    - A structured content organization (e.g., modules, topics).
      * Instructors can organize content into modules, units, or topics to create a logical learning path.
      * The system should support nested structures and allow for easy reordering of content.
    - Offline access to downloaded materials (using Hive).
      * Students can download course materials for offline viewing.
      * The app should efficiently manage storage space and provide options for selectively downloading content.
      * The app should automatically synchronize downloaded materials when the device is online.
* **3.3 Assignment Management**
  + **Assignment Creation:**
    - Instructors can create assignments with details such as title, description, due date, instructions, and grading criteria.
      * Instructors can specify the assignment type (e.g., essay, quiz, project) and configure submission settings.
      * The system should support different grading scales (e.g., points, percentages, letter grades) and allow instructors to define custom rubrics.
    - Support for different assignment types (e.g., essays, quizzes, projects).
      * **Essays:** Support for file uploads, text submissions, and plagiarism detection integration.
      * **Quizzes:** Support for various question types (e.g., multiple choice, true/false, short answer), automatic grading, and time limits.
      * **Projects:** Support for group submissions, peer review, and progress tracking.
  + **Assignment Submission:**
    - Students can submit their assignments through the app, including file uploads and text submissions.
      * The app should support various file types and provide clear instructions for submitting assignments.
      * Students should be able to track their submission status and view submission history.
    - Submission confirmation and history.
      * The system should provide students with confirmation of successful submission, including a timestamp and submission ID.
      * Students should be able to view their previous submissions and any feedback received.
  + **Grading:**
    - Instructors can grade assignments, provide feedback, and record grades.
      * The app should provide a user-friendly interface for grading assignments, with options for providing written and audio feedback.
      * Instructors should be able to use rubrics to ensure consistent and objective grading.
    - Support for different grading schemes (e.g., points, percentages).
      * The system should support various grading schemes and allow instructors to customize them.
      * The app should automatically calculate grades based on the chosen scheme and grading criteria.
    - Students can view their grades and feedback.
      * Students should be able to access their grades and feedback as soon as they are released by the instructor.
      * The app should provide clear explanations of grades and feedback.
* **3.4 Communication and Collaboration**
  + **Announcements:**
    - Instructors can post announcements to the entire class.
      * Instructors can format announcements using a rich text editor and include attachments.
      * The system should allow instructors to schedule announcements for future release.
    - Push notifications for important announcements.
      * Students should receive push notifications for urgent or important announcements, such as changes to the syllabus or assignment deadlines.
      * Users should be able to customize their notification preferences.
  + **Discussion Forums:**
    - Asynchronous discussion forums for students to interact with each other and the instructor.
      * The system should support threaded discussions, allowing for organized and focused conversations.
      * Instructors can create forums, set participation requirements, and moderate discussions.
    - Support for threads, replies, and notifications.
      * Users should be able to subscribe to threads and receive notifications for new replies.
      * The app should provide tools for searching and filtering discussion posts.
  + **Direct Messaging (Optional):**
    - A feature for students and instructors to communicate privately.
      * The system should support one-on-one messaging between students and instructors, as well as student-to-student messaging (optional).
      * The app should provide a user-friendly interface for sending, receiving, and managing messages.
  + **Real-time Notifications:**
    - Utilize push notifications to alert users of new announcements, assignment due dates, grades, and forum activity.
      * The system should allow users to customize their notification preferences, choosing which events trigger notifications and how they are delivered (e.g., push, email).
      * The app should provide a clear and organized view of all notifications.
* **3.5 Gradebook**
  + **Grade Display:**
    - Students can view their grades for individual assignments and their overall course grade.
      * The gradebook should display grades clearly and concisely, with options for viewing detailed feedback and grading rubrics.
      * Students should be able to track their progress throughout the course and see how their grades are calculated.
    - Grade Calculation:
      * The app automatically calculates grades based on the grading criteria set by the instructor.
        + The system should support various grading methods, such as weighted averages, points-based systems, and custom formulas.
        + Instructors should be able to define grading rules and policies.
    - Instructor Grade Management:
      * Instructors can manage and update grades.
        + Instructors should be able to enter, edit, and delete grades, as well as provide feedback to students.
        + The system should provide tools for importing and exporting grades from/to external systems (e.g., spreadsheets).
        + Instructors should be able to generate grade reports and analyze student performance.

## University Course Management System - MySQL Database Schema

This document outlines the MySQL database schema for the University Course Management System, including table definitions and their relationships.

### 1. Users Table

Stores information about all users of the system (students, instructors, and administrators).

CREATE TABLE Users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(255) UNIQUE NOT NULL,

password\_hash VARCHAR(255) NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

role ENUM('Student', 'Instructor', 'Admin') NOT NULL,

first\_name VARCHAR(100),

last\_name VARCHAR(100),

created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,

last\_login DATETIME

);

**Relationships:**

* One-to-Many with Courses (an instructor teaches many courses).
* One-to-Many with Enrollments (a student has many enrollments).
* One-to-Many with Course\_Materials (a user uploads many materials).
* One-to-Many with Assignments (an instructor creates many assignments).
* One-to-Many with Submissions (a student makes many submissions).
* One-to-Many with Grades (an instructor grades many submissions).
* One-to-Many with Announcements (an instructor posts many announcements).
* One-to-Many with Discussion\_Forums (a user creates many forums).
* One-to-Many with Discussion\_Posts (a user makes many posts).
* One-to-Many with Messages (a user sends/receives many messages).

### 2. Courses Table

Stores details about university courses.

CREATE TABLE Courses (

course\_id INT AUTO\_INCREMENT PRIMARY KEY,

course\_name VARCHAR(255) NOT NULL,

course\_code VARCHAR(50) UNIQUE NOT NULL,

description TEXT,

instructor\_id INT NOT NULL,

schedule VARCHAR(255),

prerequisites TEXT,

start\_date DATE,

end\_date DATE,

credits DECIMAL(3,1),

level VARCHAR(50),

term VARCHAR(50),

delivery\_method VARCHAR(50),

syllabus\_url VARCHAR(255),

required\_textbooks TEXT,

FOREIGN KEY (instructor\_id) REFERENCES Users(user\_id)

);

**Relationships:**

* Many-to-One with Users (an instructor teaches one or more courses).
* One-to-Many with Enrollments (a course has many enrollments).
* One-to-Many with Course\_Modules (a course has many modules).
* One-to-Many with Course\_Materials (a course has many materials).
* One-to-Many with Assignments (a course has many assignments).
* One-to-Many with Announcements (a course has many announcements).
* One-to-Many with Discussion\_Forums (a course has many discussion forums).

### 3. Enrollments Table

Manages the many-to-many relationship between Users (students) and Courses.

CREATE TABLE Enrollments (

enrollment\_id INT AUTO\_INCREMENT PRIMARY KEY,

student\_id INT NOT NULL,

course\_id INT NOT NULL,

enrollment\_date DATETIME DEFAULT CURRENT\_TIMESTAMP,

status ENUM('Enrolled', 'Waitlisted', 'Dropped') DEFAULT 'Enrolled',

FOREIGN KEY (student\_id) REFERENCES Users(user\_id),

FOREIGN KEY (course\_id) REFERENCES Courses(course\_id),

UNIQUE (student\_id, course\_id) -- Ensures a student can only enroll in a course once

);

**Relationships:**

* Many-to-One with Users (many enrollments belong to one student).
* Many-to-One with Courses (many enrollments belong to one course).

### 4. Course\_Modules Table

Organizes course content into logical modules or units.

CREATE TABLE Course\_Modules (

module\_id INT AUTO\_INCREMENT PRIMARY KEY,

course\_id INT NOT NULL,

module\_name VARCHAR(255) NOT NULL,

module\_order INT, -- To define the display order of modules

FOREIGN KEY (course\_id) REFERENCES Courses(course\_id)

);

**Relationships:**

* Many-to-One with Courses (many modules belong to one course).
* One-to-Many with Course\_Materials (a module can contain many materials).

### 5. Course\_Materials Table

Stores details about course materials (documents, videos, presentations, etc.).

CREATE TABLE Course\_Materials (

material\_id INT AUTO\_INCREMENT PRIMARY KEY,

course\_id INT NOT NULL,

module\_id INT, -- Optional: links to a specific module

title VARCHAR(255) NOT NULL,

description TEXT,

file\_path VARCHAR(255), -- Path to the stored file

file\_type VARCHAR(50), -- e.g., 'PDF', 'DOCX', 'MP4'

external\_url VARCHAR(255), -- For embedded resources (e.g., YouTube links)

uploaded\_by INT NOT NULL,

upload\_date DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (course\_id) REFERENCES Courses(course\_id),

FOREIGN KEY (module\_id) REFERENCES Course\_Modules(module\_id),

FOREIGN KEY (uploaded\_by) REFERENCES Users(user\_id)

);

**Relationships:**

* Many-to-One with Courses (many materials belong to one course).
* Many-to-One with Course\_Modules (many materials can belong to one module).
* Many-to-One with Users (many materials are uploaded by one user).

### 6. Assignments Table

Defines assignments for courses.

CREATE TABLE Assignments (

assignment\_id INT AUTO\_INCREMENT PRIMARY KEY,

course\_id INT NOT NULL,

instructor\_id INT NOT NULL,

title VARCHAR(255) NOT NULL,

description TEXT,

due\_date DATETIME NOT NULL,

instructions TEXT,

grading\_criteria TEXT,

assignment\_type ENUM('Essay', 'Quiz', 'Project', 'Other') NOT NULL,

max\_points DECIMAL(10,2),

grading\_scale VARCHAR(100), -- e.g., "Points", "Percentage", "Letter Grade"

FOREIGN KEY (course\_id) REFERENCES Courses(course\_id),

FOREIGN KEY (instructor\_id) REFERENCES Users(user\_id)

);

**Relationships:**

* Many-to-One with Courses (many assignments belong to one course).
* Many-to-One with Users (many assignments are created by one instructor).
* One-to-Many with Submissions (an assignment has many submissions).

### 7. Submissions Table

Records student submissions for assignments.

CREATE TABLE Submissions (

submission\_id INT AUTO\_INCREMENT PRIMARY KEY,

assignment\_id INT NOT NULL,

student\_id INT NOT NULL,

submission\_date DATETIME DEFAULT CURRENT\_TIMESTAMP,

file\_path VARCHAR(255), -- Path to the submitted file

text\_content LONGTEXT, -- For direct text submissions

status ENUM('Submitted', 'Late', 'Graded', 'Draft') DEFAULT 'Submitted',

FOREIGN KEY (assignment\_id) REFERENCES Assignments(assignment\_id),

FOREIGN KEY (student\_id) REFERENCES Users(user\_id),

UNIQUE (assignment\_id, student\_id) -- Ensures a student has only one active submission per assignment (can be modified for multiple attempts/versions)

);

**Relationships:**

* Many-to-One with Assignments (many submissions belong to one assignment).
* Many-to-One with Users (many submissions are made by one student).
* One-to-One with Grades (each submission can have one grade).

### 8. Grades Table

Stores grades and feedback for submitted assignments.

CREATE TABLE Grades (

grade\_id INT AUTO\_INCREMENT PRIMARY KEY,

submission\_id INT UNIQUE NOT NULL, -- Ensures one grade per submission

score DECIMAL(5,2),

feedback\_text TEXT,

feedback\_audio\_url VARCHAR(255),

graded\_by INT NOT NULL,

grade\_date DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (submission\_id) REFERENCES Submissions(submission\_id),

FOREIGN KEY (graded\_by) REFERENCES Users(user\_id)

);

**Relationships:**

* One-to-One with Submissions (a grade is for one specific submission).
* Many-to-One with Users (many grades are given by one instructor).

### 9. Announcements Table

Stores announcements posted by instructors for courses.

CREATE TABLE Announcements (

announcement\_id INT AUTO\_INCREMENT PRIMARY KEY,

course\_id INT NOT NULL,

instructor\_id INT NOT NULL,

title VARCHAR(255) NOT NULL,

content TEXT NOT NULL,

post\_date DATETIME DEFAULT CURRENT\_TIMESTAMP,

scheduled\_date DATETIME, -- For announcements to be released in the future

FOREIGN KEY (course\_id) REFERENCES Courses(course\_id),

FOREIGN KEY (instructor\_id) REFERENCES Users(user\_id)

);

**Relationships:**

* Many-to-One with Courses (many announcements belong to one course).
* Many-to-One with Users (many announcements are posted by one instructor).

### 10. Discussion\_Forums Table

Defines discussion forums within courses.

CREATE TABLE Discussion\_Forums (

forum\_id INT AUTO\_INCREMENT PRIMARY KEY,

course\_id INT NOT NULL,

title VARCHAR(255) NOT NULL,

description TEXT,

created\_by INT NOT NULL,

created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (course\_id) REFERENCES Courses(course\_id),

FOREIGN KEY (created\_by) REFERENCES Users(user\_id)

);

**Relationships:**

* Many-to-One with Courses (many forums belong to one course).
* Many-to-One with Users (many forums are created by one user).
* One-to-Many with Discussion\_Posts (a forum has many posts).

### 11. Discussion\_Posts Table

Stores individual posts and replies within discussion forums.

CREATE TABLE Discussion\_Posts (

post\_id INT AUTO\_INCREMENT PRIMARY KEY,

forum\_id INT NOT NULL,

parent\_post\_id INT, -- NULL for original threads, references post\_id for replies

user\_id INT NOT NULL,

content TEXT NOT NULL,

post\_date DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (forum\_id) REFERENCES Discussion\_Forums(forum\_id),

FOREIGN KEY (parent\_post\_id) REFERENCES Discussion\_Posts(post\_id), -- Self-referencing for replies

FOREIGN KEY (user\_id) REFERENCES Users(user\_id)

);

**Relationships:**

* Many-to-One with Discussion\_Forums (many posts belong to one forum).
* Many-to-One with Users (many posts are made by one user).
* Self-referencing for parent\_post\_id to create threaded discussions.

### 12. Messages Table

Handles direct private messages between users.

CREATE TABLE Messages (

message\_id INT AUTO\_INCREMENT PRIMARY KEY,

sender\_id INT NOT NULL,

receiver\_id INT NOT NULL,

content TEXT NOT NULL,

sent\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,

read\_status BOOLEAN DEFAULT FALSE,

FOREIGN KEY (sender\_id) REFERENCES Users(user\_id),

FOREIGN KEY (receiver\_id) REFERENCES Users(user\_id)

);

**Relationships:**

* Many-to-One with Users (many messages are sent by one user).
* Many-to-One with Users (many messages are received by one user).

## Sample Data for University Course Management System

This document provides INSERT statements for populating the MySQL tables of the University Course Management System. The data is generated to ensure consistency and adhere to foreign key constraints.

### 1. Users Table Records

INSERT INTO Users (username, password\_hash, email, role, first\_name, last\_name, created\_at, last\_login) VALUES

('johndoe', 'hashed\_password\_1', 'john.doe@university.edu', 'Student', 'John', 'Doe', '2023-01-10 09:00:00', '2024-05-28 10:30:00'),

('janesmith', 'hashed\_password\_2', 'jane.smith@university.edu', 'Student', 'Jane', 'Smith', '2023-01-15 10:00:00', '2024-05-29 11:00:00'),

('alicebrown', 'hashed\_password\_3', 'alice.brown@university.edu', 'Student', 'Alice', 'Brown', '2023-02-01 11:00:00', '2024-05-27 09:15:00'),

('bobwhite', 'hashed\_password\_4', 'bob.white@university.edu', 'Student', 'Bob', 'White', '2023-02-05 12:00:00', '2024-05-29 14:00:00'),

('charliegreen', 'hashed\_password\_5', 'charlie.green@university.edu', 'Student', 'Charlie', 'Green', '2023-03-01 13:00:00', '2024-05-28 16:00:00'),

('dr.evans', 'hashed\_password\_6', 'e.evans@university.edu', 'Instructor', 'Emily', 'Evans', '2022-08-01 08:00:00', '2024-05-29 09:00:00'),

('prof.davis', 'hashed\_password\_7', 'd.davis@university.edu', 'Instructor', 'David', 'Davis', '2022-09-10 09:30:00', '2024-05-29 13:00:00'),

('mrs.taylor', 'hashed\_password\_8', 't.taylor@university.edu', 'Instructor', 'Sarah', 'Taylor', '2022-10-01 10:00:00', '2024-05-28 11:00:00'),

('adminuser', 'hashed\_password\_9', 'admin@university.edu', 'Admin', 'System', 'Admin', '2022-07-01 07:00:00', '2024-05-29 15:00:00'),

('frankblack', 'hashed\_password\_10', 'frank.b@university.edu', 'Student', 'Frank', 'Black', '2023-03-15 10:00:00', '2024-05-27 10:00:00'),

('gracekelly', 'hashed\_password\_11', 'grace.k@university.edu', 'Student', 'Grace', 'Kelly', '2023-04-01 11:00:00', '2024-05-28 12:00:00'),

('dr.wilson', 'hashed\_password\_12', 'w.wilson@university.edu', 'Instructor', 'William', 'Wilson', '2022-11-01 09:00:00', '2024-05-29 10:00:00');

### 2. Courses Table Records

*(Assuming user\_id 6, 7, 8, 12 are instructors)*

INSERT INTO Courses (course\_name, course\_code, description, instructor\_id, schedule, prerequisites, start\_date, end\_date, credits, level, term, delivery\_method, syllabus\_url, required\_textbooks) VALUES

('Introduction to Computer Science', 'CS101', 'Fundamentals of programming and algorithms.', 6, 'Mon/Wed/Fri 10:00-10:50 AM', 'None', '2024-09-02', '2024-12-13', 3.0, 'Undergraduate', 'Fall 2024', 'In-person', 'http://example.com/cs101\_syllabus.pdf', 'Python Crash Course'),

('Calculus I', 'MA201', 'Introduction to differential and integral calculus.', 7, 'Tue/Thu 09:00-10:15 AM', 'Algebra', '2024-09-02', '2024-12-13', 4.0, 'Undergraduate', 'Fall 2024', 'Hybrid', 'http://example.com/ma201\_syllabus.pdf', 'Calculus: Early Transcendentals'),

('Introduction to Psychology', 'PS101', 'Overview of psychological principles and research methods.', 8, 'Mon/Wed 14:00-15:15 PM', 'None', '2024-09-02', '2024-12-13', 3.0, 'Undergraduate', 'Fall 2024', 'Online', 'http://example.com/ps101\_syllabus.pdf', 'Psychology: The Science of Mind and Behavior'),

('Database Systems', 'CS305', 'Design and implementation of relational databases.', 6, 'Tue/Thu 13:00-14:15 PM', 'CS101', '2024-09-02', '2024-12-13', 3.0, 'Undergraduate', 'Fall 2024', 'In-person', 'http://example.com/cs305\_syllabus.pdf', 'Database Management Systems'),

('Linear Algebra', 'MA310', 'Vector spaces, linear transformations, eigenvalues.', 7, 'Mon/Wed/Fri 11:00-11:50 AM', 'MA201', '2024-09-02', '2024-12-13', 3.0, 'Undergraduate', 'Fall 2024', 'In-person', 'http://example.com/ma310\_syllabus.pdf', 'Linear Algebra and Its Applications'),

('Abnormal Psychology', 'PS320', 'Study of psychological disorders and their treatment.', 8, 'Tue/Thu 10:30-11:45 AM', 'PS101', '2024-09-02', '2024-12-13', 3.0, 'Undergraduate', 'Fall 2024', 'Hybrid', 'http://example.com/ps320\_syllabus.pdf', 'Abnormal Psychology: An Integrative Approach'),

('Web Development Fundamentals', 'CS220', 'Introduction to HTML, CSS, and JavaScript.', 12, 'Mon/Wed 15:00-16:15 PM', 'None', '2024-09-02', '2024-12-13', 3.0, 'Undergraduate', 'Fall 2024', 'Online', 'http://example.com/cs220\_syllabus.pdf', 'Eloquent JavaScript'),

('Data Structures and Algorithms', 'CS202', 'Advanced data structures and algorithm analysis.', 6, 'Mon/Wed/Fri 09:00-09:50 AM', 'CS101', '2024-09-02', '2024-12-13', 3.0, 'Undergraduate', 'Fall 2024', 'In-person', 'http://example.com/cs202\_syllabus.pdf', 'Introduction to Algorithms'),

('Differential Equations', 'MA320', 'Methods for solving ordinary differential equations.', 7, 'Tue/Thu 12:00-13:15 PM', 'MA201', '2024-09-02', '2024-12-13', 3.0, 'Undergraduate', 'Fall 2024', 'In-person', 'http://example.com/ma320\_syllabus.pdf', 'Elementary Differential Equations'),

('Social Psychology', 'PS210', 'Study of how individuals are influenced by others.', 8, 'Mon/Wed 11:00-12:15 PM', 'PS101', '2024-09-02', '2024-12-13', 3.0, 'Undergraduate', 'Fall 2024', 'Online', 'http://example.com/ps210\_syllabus.pdf', 'Social Psychology'),

('Operating Systems', 'CS401', 'Principles of operating system design and implementation.', 6, 'Tue/Thu 15:00-16:15 PM', 'CS202', '2024-09-02', '2024-12-13', 3.0, 'Undergraduate', 'Fall 2024', 'In-person', 'http://example.com/cs401\_syllabus.pdf', 'Operating System Concepts'),

('Introduction to Statistics', 'ST200', 'Basic statistical methods and data analysis.', 7, 'Mon/Wed/Fri 13:00-13:50 PM', 'None', '2024-09-02', '2024-12-13', 3.0, 'Undergraduate', 'Fall 2024', 'Hybrid', 'http://example.com/st200\_syllabus.pdf', 'Statistics for Dummies');

### 3. Enrollments Table Records

*(Assuming user\_id 1-5, 10-11 are students; course\_id 1-12 are courses)*

INSERT INTO Enrollments (student\_id, course\_id, enrollment\_date, status) VALUES

(1, 1, '2024-08-20 09:00:00', 'Enrolled'), -- John Doe in CS101

(1, 2, '2024-08-20 09:05:00', 'Enrolled'), -- John Doe in MA201

(2, 1, '2024-08-21 10:00:00', 'Enrolled'), -- Jane Smith in CS101

(2, 3, '2024-08-21 10:05:00', 'Enrolled'), -- Jane Smith in PS101

(3, 4, '2024-08-22 11:00:00', 'Enrolled'), -- Alice Brown in CS305

(3, 5, '2024-08-22 11:05:00', 'Enrolled'), -- Alice Brown in MA310

(4, 6, '2024-08-23 12:00:00', 'Enrolled'), -- Bob White in PS320

(4, 7, '2024-08-23 12:05:00', 'Enrolled'), -- Bob White in CS220

(5, 8, '2024-08-24 13:00:00', 'Enrolled'), -- Charlie Green in CS202

(5, 9, '2024-08-24 13:05:00', 'Enrolled'), -- Charlie Green in MA320

(10, 1, '2024-08-25 14:00:00', 'Enrolled'), -- Frank Black in CS101

(10, 10, '2024-08-25 14:05:00', 'Enrolled'), -- Frank Black in PS210

(11, 11, '2024-08-26 15:00:00', 'Enrolled'), -- Grace Kelly in CS401

(11, 12, '2024-08-26 15:05:00', 'Enrolled'), -- Grace Kelly in ST200

(1, 8, '2024-08-27 09:00:00', 'Enrolled'), -- John Doe in CS202

(2, 10, '2024-08-27 10:00:00', 'Enrolled'); -- Jane Smith in PS210

### 4. Course\_Modules Table Records

*(Assuming course\_id 1-12 are courses)*

INSERT INTO Course\_Modules (course\_id, module\_name, module\_order) VALUES

(1, 'Module 1: Introduction to Programming', 1),

(1, 'Module 2: Data Types and Variables', 2),

(1, 'Module 3: Control Flow', 3),

(2, 'Module 1: Limits and Continuity', 1),

(2, 'Module 2: Differentiation', 2),

(3, 'Module 1: History and Approaches', 1),

(3, 'Module 2: Research Methods', 2),

(4, 'Module 1: Relational Model', 1),

(4, 'Module 2: SQL Fundamentals', 2),

(5, 'Module 1: Vector Spaces', 1),

(6, 'Module 1: Anxiety Disorders', 1),

(7, 'Module 1: HTML Basics', 1),

(8, 'Module 1: Array Data Structures', 1),

(9, 'Module 1: First-Order Equations', 1),

(10, 'Module 1: Social Cognition', 1);

### 5. Course\_Materials Table Records

*(Assuming course\_id 1-12, module\_id 1-15, user\_id 6, 7, 8, 12 are instructors)*

INSERT INTO Course\_Materials (course\_id, module\_id, title, description, file\_path, file\_type, external\_url, uploaded\_by, upload\_date) VALUES

(1, 1, 'CS101 Lecture 1 Slides', 'Introduction to CS concepts.', '/materials/cs101/lec1.pdf', 'PDF', NULL, 6, '2024-09-01 10:00:00'),

(1, 2, 'Python Variables Cheatsheet', 'Quick reference for Python variables.', '/materials/cs101/vars\_cheatsheet.docx', 'DOCX', NULL, 6, '2024-09-08 11:00:00'),

(1, NULL, 'Introduction to Python (Video)', 'Video lecture on Python basics.', NULL, 'MP4', 'https://www.youtube.com/watch?v=example1', 6, '2024-09-05 14:00:00'),

(2, 4, 'MA201 Limits Notes', 'Detailed notes on limits.', '/materials/ma201/limits\_notes.pdf', 'PDF', NULL, 7, '2024-09-03 09:30:00'),

(3, 6, 'PS101 Research Methods Reading', 'Chapter 2 on research methodologies.', '/materials/ps101/chapter2.pdf', 'PDF', NULL, 8, '2024-09-04 10:00:00'),

(4, 8, 'SQL Joins Tutorial', 'Interactive tutorial on SQL joins.', NULL, 'HTML', 'https://www.w3schools.com/sql/sql\_join.asp', 6, '2024-09-10 15:00:00'),

(7, 11, 'HTML Tags Reference', 'Comprehensive list of HTML tags.', '/materials/cs220/html\_ref.pdf', 'PDF', NULL, 12, '2024-09-05 11:00:00'),

(8, 12, 'Array Operations (Video)', 'Video explanation of array operations.', NULL, 'MP4', 'https://www.youtube.com/watch?v=example2', 6, '2024-09-12 10:00:00'),

(1, 3, 'CS101 Control Flow Practice', 'Exercises for control flow statements.', '/materials/cs101/control\_flow\_practice.pdf', 'PDF', NULL, 6, '2024-09-15 13:00:00'),

(2, 5, 'MA201 Differentiation Examples', 'Solved examples for differentiation.', '/materials/ma201/diff\_examples.pdf', 'PDF', NULL, 7, '2024-09-18 14:00:00'),

(3, 7, 'PS101 Approaches to Psychology', 'Overview of different psychological approaches.', '/materials/ps101/approaches.pptx', 'PPTX', NULL, 8, '2024-09-20 10:00:00');

### 6. Assignments Table Records

*(Assuming course\_id 1-12, user\_id 6, 7, 8, 12 are instructors)*

INSERT INTO Assignments (course\_id, instructor\_id, title, description, due\_date, instructions, grading\_criteria, assignment\_type, max\_points, grading\_scale) VALUES

(1, 6, 'Programming Assignment 1', 'Write a Python program to calculate factorial.', '2024-09-20 23:59:59', 'Submit a .py file.', 'Correctness, Efficiency, Readability', 'Project', 100.00, 'Points'),

(1, 6, 'Quiz 1: Python Basics', 'Multiple choice quiz on Python syntax.', '2024-09-25 23:59:59', 'Complete online quiz.', 'Automatic grading', 'Quiz', 20.00, 'Points'),

(2, 7, 'Calculus Homework 1', 'Solve problems on limits and continuity.', '2024-09-22 23:59:59', 'Upload PDF of solutions.', 'Correctness of solutions', 'Essay', 50.00, 'Points'),

(3, 8, 'Psychology Essay 1', 'Discuss the nature vs. nurture debate.', '2024-09-28 23:59:59', 'Min 1000 words, APA style.', 'Content, Structure, Argumentation', 'Essay', 75.00, 'Points'),

(4, 6, 'Database Design Project', 'Design a relational database schema for a library system.', '2024-10-15 23:59:59', 'Submit ERD and schema.', 'Completeness, Normalization, Efficiency', 'Project', 150.00, 'Points'),

(7, 12, 'Build a Personal Website', 'Create a simple personal website using HTML and CSS.', '2024-10-05 23:59:59', 'Submit a zip file of your project.', 'Design, Responsiveness, Code Quality', 'Project', 100.00, 'Points'),

(8, 6, 'Algorithm Analysis Homework', 'Analyze the time complexity of given algorithms.', '2024-10-10 23:59:59', 'Upload PDF of solutions.', 'Correctness of analysis', 'Essay', 60.00, 'Points'),

(9, 7, 'Differential Equations Problem Set 1', 'Solve first-order differential equations.', '2024-10-01 23:59:59', 'Upload PDF of solutions.', 'Correctness of solutions', 'Essay', 40.00, 'Points'),

(10, 8, 'Social Psychology Reflection', 'Write a reflection on a social psychology experiment.', '2024-10-07 23:59:59', 'Min 500 words.', 'Insight, Critical Thinking', 'Essay', 50.00, 'Points'),

(1, 6, 'Programming Assignment 2', 'Implement a sorting algorithm.', '2024-10-20 23:59:59', 'Submit a .py file.', 'Correctness, Efficiency', 'Project', 100.00, 'Points');

### 7. Submissions Table Records

*(Assuming assignment\_id 1-10, user\_id 1-5, 10-11 are students)*

INSERT INTO Submissions (assignment\_id, student\_id, submission\_date, file\_path, text\_content, status) VALUES

(1, 1, '2024-09-19 22:00:00', '/submissions/john\_doe/pa1.py', NULL, 'Submitted'),

(1, 2, '2024-09-20 23:00:00', '/submissions/jane\_smith/pa1.py', NULL, 'Submitted'),

(2, 1, '2024-09-24 18:00:00', NULL, '{"q1": "A", "q2": "C"}', 'Submitted'), -- Example JSON for quiz answers

(3, 1, '2024-09-21 15:00:00', '/submissions/john\_doe/calc\_hw1.pdf', NULL, 'Submitted'),

(3, 2, '2024-09-23 10:00:00', '/submissions/jane\_smith/calc\_hw1.pdf', NULL, 'Late'),

(4, 2, '2024-09-27 20:00:00', NULL, 'The nature vs. nurture debate is complex...', 'Submitted'),

(5, 3, '2024-10-14 17:00:00', '/submissions/alice\_brown/db\_project.zip', NULL, 'Submitted'),

(7, 4, '2024-10-04 21:00:00', '/submissions/bob\_white/website.zip', NULL, 'Submitted'),

(8, 5, '2024-10-09 19:00:00', '/submissions/charlie\_green/algo\_analysis.pdf', NULL, 'Submitted'),

(9, 5, '2024-09-30 22:00:00', '/submissions/charlie\_green/diff\_eq\_ps1.pdf', NULL, 'Submitted'),

(1, 10, '2024-09-20 20:00:00', '/submissions/frank\_black/pa1.py', NULL, 'Submitted'),

(2, 10, '2024-09-25 21:00:00', NULL, '{"q1": "B", "q2": "A"}', 'Submitted');

### 8. Grades Table Records

*(Assuming submission\_id 1-12, user\_id 6, 7, 8, 12 are instructors)*

INSERT INTO Grades (submission\_id, score, feedback\_text, feedback\_audio\_url, graded\_by, grade\_date) VALUES

(1, 95.00, 'Excellent work, clear and efficient code.', NULL, 6, '2024-09-25 10:00:00'),

(2, 88.00, 'Good attempt, minor issues with edge cases.', NULL, 6, '2024-09-26 11:00:00'),

(3, 18.00, 'Well done on the quiz.', NULL, 6, '2024-09-26 11:30:00'),

(4, 45.00, 'Solid understanding, review problem 3.', NULL, 7, '2024-09-28 14:00:00'),

(5, 30.00, 'Good effort, but submitted late. Content needs more depth.', NULL, 7, '2024-09-29 10:00:00'),

(6, 68.00, 'Interesting points, but structure could be improved.', NULL, 8, '2024-10-01 16:00:00'),

(7, 130.00, 'Comprehensive design, good use of normalization.', NULL, 6, '2024-10-20 09:00:00'),

(8, 92.00, 'Visually appealing and responsive.', NULL, 12, '2024-10-08 10:00:00'),

(9, 55.00, 'Correct analysis for most algorithms, some minor errors.', NULL, 6, '2024-10-15 11:00:00'),

(10, 38.00, 'Solutions are mostly correct, show more steps next time.', NULL, 7, '2024-10-05 13:00:00');

### 9. Announcements Table Records

*(Assuming course\_id 1-12, user\_id 6, 7, 8, 12 are instructors)*

INSERT INTO Announcements (course\_id, instructor\_id, title, content, post\_date, scheduled\_date) VALUES

(1, 6, 'Welcome to CS101!', 'Welcome to the course! Please review the syllabus.', '2024-09-01 08:00:00', NULL),

(1, 6, 'Assignment 1 Due Date Extension', 'Due date for PA1 extended to Sept 20th.', '2024-09-15 17:00:00', NULL),

(2, 7, 'Office Hours Change', 'My office hours for this week are changed to Thursday 11 AM.', '2024-09-18 09:00:00', NULL),

(3, 8, 'Guest Speaker Next Week', 'We will have a guest speaker on Oct 5th.', '2024-09-29 10:00:00', NULL),

(4, 6, 'Midterm Exam Schedule', 'Midterm exam for CS305 will be on Oct 25th.', '2024-10-01 14:00:00', NULL),

(7, 12, 'New Tutorial Posted', 'A new tutorial on CSS Flexbox has been uploaded to Module 2.', '2024-09-10 11:00:00', NULL),

(8, 6, 'Review Session for Data Structures', 'There will be a review session on Oct 18th.', '2024-10-12 16:00:00', NULL),

(1, 6, 'No Class on Friday', 'Due to a conference, CS101 class on Friday, Sept 27th is cancelled.', '2024-09-26 09:00:00', NULL),

(2, 7, 'Reminder: Calculus Homework 2', 'Just a reminder that Homework 2 is due this Sunday.', '2024-09-25 15:00:00', NULL),

(3, 8, 'Reading for Next Week', 'Please read Chapter 3 for next week\'s lecture.', '2024-09-27 11:00:00', NULL);

### 10. Discussion\_Forums Table Records

*(Assuming course\_id 1-12, user\_id 6, 7, 8, 12 are instructors, 1, 2, 3, 4, 5, 10, 11 are students)*

INSERT INTO Discussion\_Forums (course\_id, title, description, created\_by, created\_at) VALUES

(1, 'General Q&A', 'Ask any questions about the course material or assignments.', 6, '2024-09-01 12:00:00'),

(1, 'Programming Help', 'Discuss programming challenges and solutions.', 6, '2024-09-05 10:00:00'),

(2, 'Calculus Concepts', 'Discuss difficult calculus concepts.', 7, '2024-09-03 14:00:00'),

(3, 'Research Methods Discussion', 'Share thoughts on different research methodologies.', 8, '2024-09-06 11:00:00'),

(4, 'SQL Queries', 'Forum for discussing SQL query optimization and issues.', 6, '2024-09-12 09:00:00'),

(7, 'HTML/CSS Layouts', 'Share and get feedback on your website layouts.', 12, '2024-09-07 10:00:00'),

(8, 'Data Structures Problems', 'Discuss solutions to data structure problems.', 6, '2024-09-15 13:00:00'),

(10, 'Social Psychology Debates', 'Debate controversial topics in social psychology.', 8, '2024-09-10 14:00:00'),

(1, 'Module 1 Feedback', 'Provide feedback on Module 1 content.', 1, '2024-09-10 15:00:00'),

(2, 'Exam Prep', 'Collaborate on preparing for the Calculus midterm.', 2, '2024-10-01 10:00:00');

### 11. Discussion\_Posts Table Records

*(Assuming forum\_id 1-10, user\_id 1-5, 10-11 are students, 6, 7, 8, 12 are instructors)*

INSERT INTO Discussion\_Posts (forum\_id, parent\_post\_id, user\_id, content, post\_date) VALUES

(1, NULL, 1, 'When is the first programming assignment due?', '2024-09-02 10:30:00'), -- Post 1

(1, 1, 6, 'PA1 is due on September 20th, John. Please check the assignments section.', '2024-09-02 11:00:00'), -- Post 2 (Reply to Post 1)

(2, NULL, 2, 'I am having trouble with the loop in the factorial program.', '2024-09-06 14:00:00'), -- Post 3

(2, 3, 1, 'Jane, make sure your loop condition is correct and the variable is incrementing.', '2024-09-06 14:30:00'), -- Post 4 (Reply to Post 3)

(3, NULL, 3, 'Can someone explain L\'Hopital\'s Rule in simpler terms?', '2024-09-05 16:00:00'), -- Post 5

(3, 5, 7, 'L\'Hopital\'s Rule helps evaluate limits of indeterminate forms...', '2024-09-05 16:30:00'), -- Post 6 (Reply to Post 5)

(4, NULL, 4, 'What are the ethical considerations in psychological research?', '2024-09-08 09:00:00'), -- Post 7

(4, 7, 8, 'Excellent question, Bob. Ethical guidelines protect participants...', '2024-09-08 09:30:00'), -- Post 8 (Reply to Post 7)

(5, NULL, 5, 'How to optimize SQL queries for large datasets?', '2024-09-13 11:00:00'), -- Post 9

(5, 9, 6, 'Indexing is crucial. Also, avoid SELECT \* and use specific columns.', '2024-09-13 11:30:00'), -- Post 10 (Reply to Post 9)

(1, NULL, 10, 'Are there any extra credit opportunities?', '2024-09-18 10:00:00'), -- Post 11

(1, 11, 6, 'Not at the moment, Frank. Focus on the main assignments.', '2024-09-18 10:30:00'), -- Post 12 (Reply to Post 11)

(7, NULL, 11, 'What are some common mistakes in HTML semantic tagging?', '2024-09-10 14:00:00'), -- Post 13

(7, 13, 12, 'Overusing `div` instead of semantic tags like `header`, `nav`, `main`, `footer`.', '2024-09-10 14:30:00'); -- Post 14 (Reply to Post 13)

### 12. Messages Table Records

*(Assuming user\_id 1-5, 10-11 are students, 6, 7, 8, 12 are instructors, 9 is admin)*

INSERT INTO Messages (sender\_id, receiver\_id, content, sent\_at, read\_status) VALUES

(1, 6, 'Professor Evans, I have a question about PA1.', '2024-09-17 10:00:00', FALSE),

(6, 1, 'Hi John, please post your question on the discussion forum if it\'s general, or come to office hours.', '2024-09-17 10:15:00', TRUE),

(2, 8, 'Dear Professor Taylor, could I get an extension on the essay?', '2024-09-26 15:00:00', FALSE),

(8, 2, 'Jane, please provide a valid reason for the extension request.', '2024-09-26 15:30:00', TRUE),

(3, 7, 'Professor Davis, I need help with problem 5 on HW1.', '2024-09-21 11:00:00', FALSE),

(7, 3, 'Alice, I\'ll be in my office from 2-3 PM today if you want to stop by.', '2024-09-21 11:15:00', TRUE),

(4, 5, 'Hey Charlie, want to study for the CS220 midterm together?', '2024-10-01 16:00:00', FALSE),

(5, 4, 'Sure Bob, when are you free?', '2024-10-01 16:15:00', TRUE),

(10, 6, 'Professor Evans, I submitted PA1. Can you confirm?', '2024-09-20 23:00:00', FALSE),

(6, 10, 'Yes, Frank, your submission is recorded.', '2024-09-21 09:00:00', TRUE),

(11, 12, 'Professor Wilson, I am having trouble accessing the video lecture.', '2024-09-08 10:00:00', FALSE),

(12, 11, 'Grace, please try clearing your browser cache or using a different browser.', '2024-09-08 10:15:00', TRUE);

