CREATE DATABASE IF NOT EXISTS academic\_manager;

USE academic\_manager;

-- User table for authentication

CREATE TABLE user\_ (

id INT AUTO\_INCREMENT PRIMARY KEY,

email VARCHAR(100) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL, role Enum('student','teacher','admin') ,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

is\_active BOOLEAN DEFAULT TRUE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Departments

CREATE TABLE departments (

department\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

code VARCHAR(10) NOT NULL UNIQUE,

description TEXT

);

-- Student profile information

CREATE TABLE students (

student\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT NOT NULL,

date\_of\_birth DATE,

gender ENUM('male', 'female', 'other'),

address TEXT,

phone VARCHAR(20),

major VARCHAR(100),

program ENUM('bachelors', 'masters', 'phd', 'diploma', 'certificate') NOT NULL,

enrollment\_date DATE NOT NULL,

current\_semester INT,

FOREIGN KEY (user\_id) REFERENCES user\_(id) ON DELETE CASCADE,

UNIQUE KEY (user\_id)

);

-- Teacher

CREATE TABLE teacher (

teacher\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT NOT NULL,

birth\_date DATE,

gender ENUM('male', 'female', 'other'),

address TEXT,

phone VARCHAR(20),

salary INT NOT NULL,

teaching\_program ENUM('bachelors', 'masters', 'phd', 'diploma', 'certificate') NOT NULL,

enrollment\_date DATE NOT NULL,

current\_semester INT,

FOREIGN KEY (user\_id) REFERENCES user\_(id) ON DELETE CASCADE,

UNIQUE KEY (user\_id)

);

-- Courses

CREATE TABLE courses (

course\_id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(100) NOT NULL,

content\_file\_path VARCHAR(255),

credits INT NOT NULL,

department\_id INT,

syllabus TEXT,

FOREIGN KEY (department\_id) REFERENCES departments(department\_id)

);

-- Semesters

CREATE TABLE semesters (

semester\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(50) NOT NULL,

start\_date DATE NOT NULL,

end\_date DATE NOT NULL,

max\_credits INT NOT NULL,

is\_current BOOLEAN DEFAULT FALSE,

UNIQUE KEY (name, start\_date)

);

-- Sections (course offerings in specific semesters)

CREATE TABLE sections (

section\_id INT AUTO\_INCREMENT PRIMARY KEY,

course\_id INT NOT NULL,

semester\_id INT NOT NULL,

instructor\_id INT,

max\_capacity INT NOT NULL,

time\_start TIME NOT NULL,

time\_end TIME NOT NULL,

day ENUM('monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday') NOT NULL,

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

FOREIGN KEY (semester\_id) REFERENCES semesters(semester\_id),

FOREIGN KEY (instructor\_id) REFERENCES user\_(id)

);

-- Student enrollment in sections

CREATE TABLE enrollments (

enrollment\_id INT AUTO\_INCREMENT PRIMARY KEY,

student\_id INT NOT NULL,

section\_id INT NOT NULL,

course\_id INT NOT NULL,

enrollment\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

status ENUM('registered', 'dropped', 'completed') DEFAULT 'registered',

FOREIGN KEY (student\_id) REFERENCES students(student\_id) ON DELETE CASCADE,

FOREIGN KEY (section\_id) REFERENCES sections(section\_id) ON DELETE CASCADE,

FOREIGN KEY (course\_id) REFERENCES courses(course\_id) ON DELETE CASCADE,

UNIQUE KEY (student\_id, section\_id)

);

-- Assignments

CREATE TABLE assignments (

assignment\_id INT AUTO\_INCREMENT PRIMARY KEY,

section\_id INT NOT NULL,

title VARCHAR(100) NOT NULL,

description TEXT,

due\_date DATETIME NOT NULL,

max\_points DECIMAL(5,2) NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (section\_id) REFERENCES sections(section\_id) ON DELETE CASCADE

);

-- Student assignment submissions

CREATE TABLE assignment\_submission (

submit\_id INT AUTO\_INCREMENT PRIMARY KEY,

assignment\_id INT NOT NULL,

student\_id INT NOT NULL,

submit\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

file\_path VARCHAR(255),

grades DECIMAL(5,2),

FOREIGN KEY (assignment\_id) REFERENCES assignments(assignment\_id) ON DELETE CASCADE,

FOREIGN KEY (student\_id) REFERENCES students(student\_id) ON DELETE CASCADE,

UNIQUE KEY (assignment\_id, student\_id)

);

-- Attendance records

CREATE TABLE attendance (

attendance\_id INT AUTO\_INCREMENT PRIMARY KEY,

section\_id INT NOT NULL,

student\_id INT NOT NULL,

course\_id INT NOT NULL,

status ENUM('present', 'absent', 'late', 'excused') NOT NULL,

date DATE NOT NULL,

duration DECIMAL(4,2) COMMENT 'Duration in hours',

FOREIGN KEY (section\_id) REFERENCES sections(section\_id) ON DELETE CASCADE,

FOREIGN KEY (student\_id) REFERENCES students(student\_id) ON DELETE CASCADE,

FOREIGN KEY (course\_id) REFERENCES courses(course\_id) ON DELETE CASCADE,

UNIQUE KEY (section\_id, student\_id, date)

);

-- Exams

CREATE TABLE exam\_course (

exam\_id INT AUTO\_INCREMENT PRIMARY KEY,

course\_id INT NOT NULL,

section\_id INT NOT NULL,

student\_id INT NOT NULL,

type ENUM('midterm', 'final', 'quiz', 'project') NOT NULL,

marks DECIMAL(5,2) NOT NULL,

total\_marks DECIMAL(5,2) NOT NULL,

FOREIGN KEY (course\_id) REFERENCES courses(course\_id) ON DELETE CASCADE,

FOREIGN KEY (section\_id) REFERENCES sections(section\_id) ON DELETE CASCADE,

FOREIGN KEY (student\_id) REFERENCES students(student\_id) ON DELETE CASCADE

);

-- Transcript records

CREATE TABLE transcript (

transcript\_id INT AUTO\_INCREMENT PRIMARY KEY,

student\_id INT NOT NULL,

course\_id INT NOT NULL,

semester\_id INT NOT NULL,

grade VARCHAR(2) NOT NULL,

points DECIMAL(3,2) NOT NULL,

type ENUM('core', 'elective') NOT NULL,

FOREIGN KEY (student\_id) REFERENCES students(student\_id) ON DELETE CASCADE,

FOREIGN KEY (course\_id) REFERENCES courses(course\_id) ON DELETE CASCADE,

FOREIGN KEY (semester\_id) REFERENCES semesters(semester\_id) ON DELETE CASCADE,

UNIQUE KEY (student\_id, course\_id, semester\_id)

);

-- Financial records

CREATE TABLE financial\_record (

record\_id INT AUTO\_INCREMENT PRIMARY KEY,

student\_id INT NOT NULL,

semester\_id INT NOT NULL,

amount DECIMAL(10,2) NOT NULL,

due\_date DATE NOT NULL,

payment\_method VARCHAR(50),

status ENUM('pending', 'paid', 'overdue', 'cancelled') NOT NULL,

FOREIGN KEY (student\_id) REFERENCES students(student\_id) ON DELETE CASCADE,

FOREIGN KEY (semester\_id) REFERENCES semesters(semester\_id) ON DELETE CASCADE

);

-- Internship opportunities

CREATE TABLE internship (

inter\_id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(100) NOT NULL,

company VARCHAR(100) NOT NULL,

description TEXT NOT NULL,

location VARCHAR(100),

start\_date DATE,

end\_date DATE,

application\_deadline DATE NOT NULL,

posted\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

is\_active BOOLEAN DEFAULT TRUE

);

-- Library books

CREATE TABLE books (

book\_id INT AUTO\_INCREMENT PRIMARY KEY,

isbn VARCHAR(20) UNIQUE,

title VARCHAR(255) NOT NULL,

author VARCHAR(100) NOT NULL,

publisher VARCHAR(100),

year INT,

category VARCHAR(50),

total\_copies INT DEFAULT 1,

available\_copies INT DEFAULT 1

);

-- Book reservations/borrowing

CREATE TABLE book\_reserve (

reserve\_id INT AUTO\_INCREMENT PRIMARY KEY,

book\_id INT NOT NULL,

student\_id INT NOT NULL,

type ENUM('borrow', 'reserve') NOT NULL,

due\_date DATE,

return\_date DATE,

status ENUM('active', 'returned', 'overdue', 'cancelled') DEFAULT 'active',

FOREIGN KEY (book\_id) REFERENCES books(book\_id) ON DELETE CASCADE,

FOREIGN KEY (student\_id) REFERENCES students(student\_id) ON DELETE CASCADE

);

-- Password reset tokens

CREATE TABLE password\_reset\_tokens (

token\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT NOT NULL,

token VARCHAR(255) NOT NULL,

expires\_at DATETIME NOT NULL,

used BOOLEAN DEFAULT FALSE,

FOREIGN KEY (user\_id) REFERENCES user\_(id) ON DELETE CASCADE

);

CREATE TABLE IF NOT EXISTS course\_section\_teacher (

id INT AUTO\_INCREMENT PRIMARY KEY,

course\_id INT NOT NULL,

section\_id INT NOT NULL,

teacher\_id INT NOT NULL,

semester\_id INT NOT NULL,

assigned\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE KEY unique\_course\_section (course\_id, section\_id),

FOREIGN KEY (semester\_id) REFERENCES semesters(semester\_id) ON DELETE CASCADE,

FOREIGN KEY (course\_id) REFERENCES courses(course\_id) ON DELETE CASCADE,

FOREIGN KEY (section\_id) REFERENCES sections(section\_id) ON DELETE CASCADE,

FOREIGN KEY (teacher\_id) REFERENCES teacher(teacher\_id) ON DELETE CASCADE

);

CREATE TABLE IF NOT EXISTS course\_content (

content\_id INT AUTO\_INCREMENT PRIMARY KEY,

section\_id INT NOT NULL,

teacher\_id INT NOT NULL,

title VARCHAR(255) NOT NULL,

description TEXT,

file\_path VARCHAR(255),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (section\_id) REFERENCES sections(section\_id) ON DELETE CASCADE,

FOREIGN KEY (teacher\_id) REFERENCES teacher(teacher\_id) ON DELETE CASCADE

);

-- 2. Announcements

CREATE TABLE IF NOT EXISTS announcements (

announcement\_id INT AUTO\_INCREMENT PRIMARY KEY,

section\_id INT NOT NULL,

teacher\_id INT NOT NULL,

title VARCHAR(100) NOT NULL,

message TEXT NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (section\_id) REFERENCES sections(section\_id) ON DELETE CASCADE,

FOREIGN KEY (teacher\_id) REFERENCES teacher(teacher\_id) ON DELETE CASCADE

);

-- 3. Feedback for Assignments

CREATE TABLE IF NOT EXISTS feedback (

feedback\_id INT AUTO\_INCREMENT PRIMARY KEY,

submit\_id INT NOT NULL,

teacher\_id INT NOT NULL,

comments TEXT,

grade DECIMAL(5,2),

graded\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (submit\_id) REFERENCES assignment\_submission(submit\_id) ON DELETE CASCADE,

FOREIGN KEY (teacher\_id) REFERENCES teacher(teacher\_id) ON DELETE CASCADE

);

-- 4. Office Hours

CREATE TABLE IF NOT EXISTS office\_hours (

hour\_id INT AUTO\_INCREMENT PRIMARY KEY,

teacher\_id INT NOT NULL,

day ENUM('monday', 'tuesday', 'wednesday', 'thursday', 'friday') NOT NULL,

start\_time TIME NOT NULL,

end\_time TIME NOT NULL,

location VARCHAR(100),

FOREIGN KEY (teacher\_id) REFERENCES teacher(teacher\_id) ON DELETE CASCADE

);

-- 5. Student Meetings

CREATE TABLE IF NOT EXISTS student\_meetings (

meeting\_id INT AUTO\_INCREMENT PRIMARY KEY,

teacher\_id INT NOT NULL,

student\_id INT NOT NULL,

meeting\_time DATETIME NOT NULL,

purpose TEXT,

status ENUM('scheduled', 'completed', 'cancelled') DEFAULT 'scheduled',

FOREIGN KEY (teacher\_id) REFERENCES teacher(teacher\_id),

FOREIGN KEY (student\_id) REFERENCES students(student\_id)

);

-- 6. Research Papers

CREATE TABLE IF NOT EXISTS research\_papers (

paper\_id INT AUTO\_INCREMENT PRIMARY KEY,

teacher\_id INT NOT NULL,

title VARCHAR(255) NOT NULL,

abstract TEXT,

published\_date DATE,

journal VARCHAR(100),

file\_path VARCHAR(255),

FOREIGN KEY (teacher\_id) REFERENCES teacher(teacher\_id) ON DELETE CASCADE

);

CREATE TABLE exam\_results (

exam\_id INT AUTO\_INCREMENT PRIMARY KEY,

student\_id INT NOT NULL,

course\_id INT NOT NULL,

semester\_id INT NOT NULL,

mid1 float ,

mid2 float ,

project float ,

quizzes float ,

assignments float ,

final\_exam float ,

final\_result float ,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (student\_id) REFERENCES students(student\_id) ON DELETE CASCADE,

FOREIGN KEY (course\_id) REFERENCES courses(course\_id) ON DELETE CASCADE,

FOREIGN KEY (semester\_id) REFERENCES semesters(semester\_id) ON DELETE CASCADE,

UNIQUE KEY unique\_exam\_record (student\_id, course\_id, semester\_id)

)