Course Enrollment Management System

Author: Yuchen Kuang **Date:** October 26, 2025 **Team:** Individual Project
Course: 5200 DAMG6210 DB Management & DB Design

01．Project Overview

1.1Project Description

This database system manages university course enrollments by tracking departments, courses, instructors, students, semesters, and enrollment records. The system provides a comprehensive solution for academic administration, enabling efficient course management and student registration processes.

1.2Database Requirements

The Course Enrollment Management System will track the following information:

Departments
Department identification and basic information
Organizational structure of the university
Relationship with courses and instructors
Courses
Course catalog information including code, name, credits, and description
Department affiliation
Instructor assignments
Availability across different semesters
Instructors
Faculty member information and credentials
Teaching assignments to courses
Contact information
Students
Student demographic information
Academic records including major and GPA
Complete enrollment history
Semesters
Academic term definitions (Fall, Spring, Summer)
Temporal organization of course offerings
Start and end dates for each term

Enrollments
Student-course registrations
Grade records and enrollment status tracking
Links between students, specific courses, and specific semesters

1.3 Key Functionalities

For Administrators:
Manage department, course, instructor, and student records
Create and configure academic semesters
Generate enrollment reports and statistics
Monitor course capacities and enrollment trends
For Instructors:
View assigned courses and enrolled students
Enter and update student grades
Access course rosters by semester
For Students:
Browse available courses by semester
Enroll in courses and drop courses when needed
View current enrollments and academic transcripts
Access grades for completed courses

02． Database Technology Selection

2.1 Selected Database: PostgreSQL (SQL)

This project will use PostgreSQL as the database management system.

2.2 Why SQL Database

Strong Relational Structure
The course enrollment domain has clearly defined relationships that are naturally expressed in a relational model:

- **One-to-many relationships:** Department to Course, Course to Enrollment, Student to Enrollment
- **Many-to-many relationships:** Course and Semester (implemented through Enrollment entity)
- These relationships are efficiently managed through foreign keys and JOIN operations

Data Integrity Requirements
Academic data requires strict consistency and integrity:

- ACID Compliance ensures enrollment transactions are atomic (a student either successfully enrolls or doesn't, with no partial states)

Referential Integrity through foreign key constraints prevents orphaned records (e.g., enrollments without valid students or courses)

Constraint Enforcement ensures business rules like "each course must belong to exactly one department" are enforced at the database level

Complex Query Requirements
The system needs sophisticated queries such as:

- Finding all students enrolled in courses taught by a specific instructor
- Calculating enrollment statistics by department and semester
- Generating student transcripts requiring multi-table JOINs
- Aggregating enrollment data across multiple semesters

Transaction Management
Enrollment operations require robust transaction support. For example, when a student enrolls:

- Check course capacity
- Register the student
- Update enrollment count

All steps must succeed together or all must fail together

03． Technology Stack

Primary Database: PostgreSQL 15.0 or higher
Operating System: Windows
Database Client: DBeaver Community Edition
Primary Language: Python 3.10 or higher and SQL
Web Framework:
Flask 3.0.0 - Lightweight web framework
Flask-SQLAlchemy 3.0.0 - ORM for database operations (optional)
Web Application
Frontend: HTML5, CSS3, JavaScript
Backend: Flask framework
Styling: Bootstrap 5 (Maybe for responsive design)
Development Tools
IDE: Visual Studio Code or PyCharm Community Edition
Version Control: Git for source code management
ER Diagram Tool: draw.io
Documentation: Markdown for README and documentation files

04. Personal Interest and Motivation

As a graduate student in computer science, I interact with course enrollment systems on a daily basis. This project provides a unique opportunity to reverse-engineer a familiar system and understand the database architecture behind tools I use regularly. By building this system from scratch, I can identify design patterns and best practices in educational software while recognizing the complexity behind seemingly simple enrollment operations.

Domain Knowledge Interest

The course enrollment domain is particularly interesting because it involves clear business rules (such as preventing duplicate enrollments and enforcing enrollment capacity limits), complex relationships (many-to-many relationships between students and courses, temporal dimensions through semesters, and hierarchical structures from departments to courses), and strict data integrity requirements (consistent grade records, no enrollment conflicts, and accurate historical tracking).

05. Entity-Relationship Diagram

5.1 System Entities

Department: Represents academic departments within the university. Departments serve as the primary organizational unit for courses and faculty members.

Course: Represents courses in the academic catalog. Each course belongs to one department, can be taught by instructors, and is offered across multiple semesters.

Instructor: Represents faculty members who teach courses. Instructors can teach multiple courses and may be affiliated with departments.

Student: Represents enrolled students in the university. Students can register for multiple courses and are tracked across multiple semesters.

Semester: Represents academic terms such as Fall, Spring, and Summer. Semesters provide the temporal dimension for the system and contain multiple course offerings.

Enrollment: An associative entity that connects Students, Courses, and Semesters. Enrollment represents a student's registration in a specific course during a specific semester. This entity contains attributes such as grade, enrollment date, and enrollment status.

5.2 Relationship

Department ↔ Course

A department can offer zero or more courses; each course must belong to exactly one department.

Course ↔ Semester

Many-to-many relationship - a course can be offered in multiple semesters; a semester can offer multiple courses.

Course ↔ Enrollment

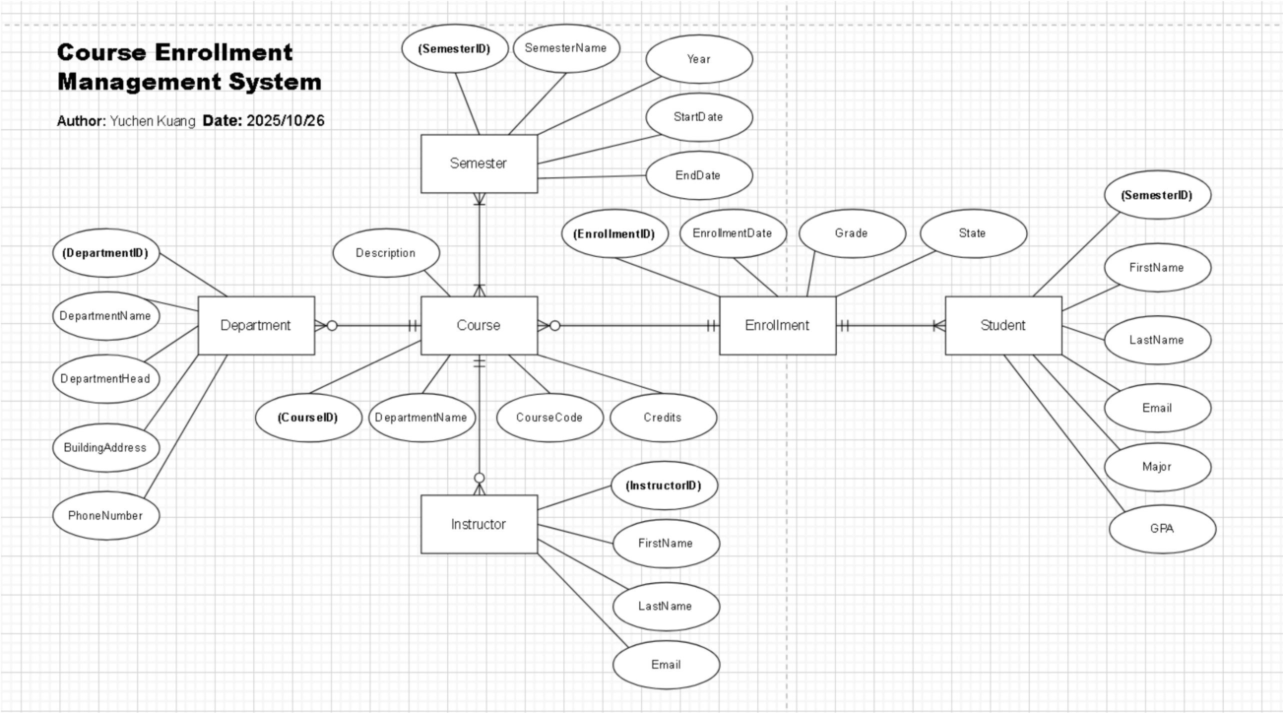
A course can have zero or more course enrollment records; each course enrollment record must correspond to exactly one course.

Enrollment ↔ Student

Each course enrollment record corresponds to exactly one student; each student can have multiple course enrollment records.

Course ↔ Instructor

Each course must have an instructor; an instructor can teach zero or more courses.



06. User Interaction Flow

6.1 System Overview

The Course Enrollment Management System supports three user roles with distinct capabilities:

1. Administrator - Full system access for data management
2. Instructor - View courses and manage grades
3. Student - Browse courses and manage enrollments

Each role has a dedicated workflow optimized for their specific responsibilities and access levels.

6.2 Administrator Workflow

Manage Departments

Step 1: Administrator selects "Department Management" from main menu
Step 2: Choose operation: Add New Department, View All Departments, Update Department, or Delete Department
Step 3: System performs selected operation (CREATE/READ/UPDATE/DELETE) and displays confirmation

Manage Courses

Step 1: Administrator selects "Course Management" from main menu
Step 2: Choose operation: Add New Course, View All Courses, Update Course, or Delete Course
Step 3: For add/update operations, select department and instructor from dropdown lists, then save. System validates and displays confirmation

Manage Instructors and Students
Step 1: Administrator selects "Instructor Management" or "Student Management"
Step 2: Choose operation: Add, View, Update, or Delete records
Step 3: Enter or modify information, then save. System performs validation and displays confirmation

Create Semesters
Step 1: Administrator selects "Semester Management"
Step 2: Enter semester name, year, start date, and end date
Step 3: System validates dates (end date after start date, no overlapping) and saves to database

View Reports
Step 1: Administrator selects "Reports and Analytics"
Step 2: Choose report type: Enrollment Statistics, Course Popularity, Department Summary, or Grade Distribution
Step 3: System performs complex queries with JOINs and GROUP BY, then displays formatted results

6.3 Instructor Workflow

View My Courses
Step 1: Instructor logs in and selects "View My Courses"
Step 2: System queries courses where InstructorID matches current instructor
Step 3: Display table showing course code, name, semester, and enrolled student count

View Course Roster
Step 1: Instructor selects a course from "My Courses" list
Step 2: System performs JOIN between Enrollment and Student tables for selected course
Step 3: Display roster with student ID, name, major, and current grade

Enter/Update Grades
Step 1: From course roster, instructor selects a student
Step 2: Enter or modify grade (validates format: A-F or 0-100)
Step 3: System updates Enrollment table and displays confirmation message

6.4 Student Workflow

Browse Available Courses
Step 1: Student logs in and selects "Browse Available Courses"
Step 2: Select semester and optionally filter by department

Step 3: System performs JOIN across Course, Department, and Instructor tables, displaying course catalog with code, name, credits, department, and instructor

Enroll in Course

Step 1: From course catalog, student selects a course

Step 2: System validates: checks if already enrolled, verifies course capacity

Step 3: If validation passes, insert new Enrollment record with status 'Enrolled' and display confirmation

View My Enrollments

Step 1: Student selects "View My Enrollments"

Step 2: Choose view option: Current Semester, All Semesters, or Specific Semester

Step 3: System performs JOIN across Enrollment, Course, Semester, and Instructor tables, displaying enrollments with grades and status

Drop Course

Step 1: From "My Enrollments", student selects course to drop

Step 2: System displays confirmation prompt

Step 3: If confirmed, update Enrollment status to 'Dropped' (soft delete) and display confirmation

View My Transcript

Step 1: Student selects "View My Transcript"

Step 2: System queries completed enrollments (Status = 'Completed') with JOINs

Step 3: Calculate GPA, display transcript organized by semester showing courses, grades, and cumulative GPA