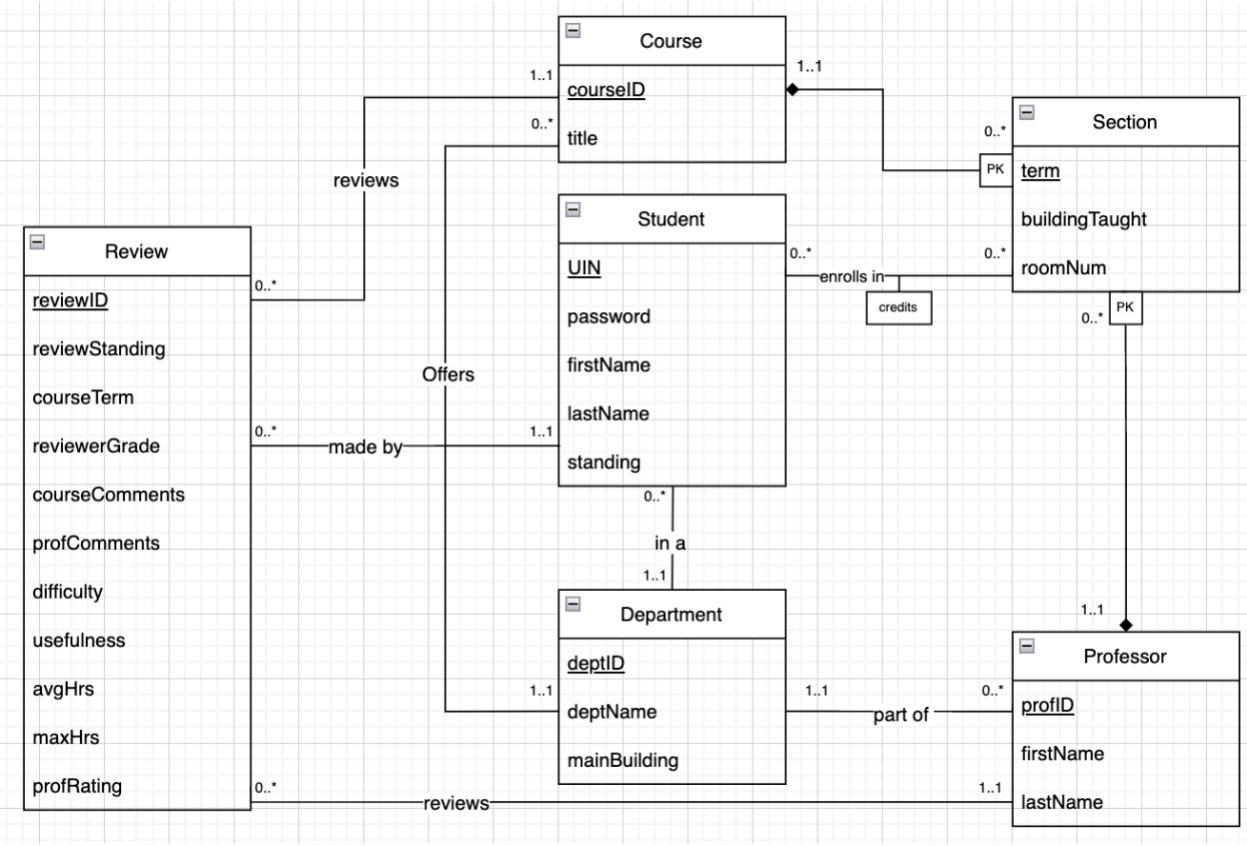
PT1 Stage 2 Report

UML Diagram



Relational Schemas

1. Course(courseID: varchar(8) [PK], title: varchar(255), deptID: varchar(255) [FK to Department.deptID])

2. Student(UIN [PK], password: varchar(255), firstName: varchar(255), lastName: varchar(255), deptID: varchar(255) [FK to Department.deptID], standing: int)
    a. Enrollments(studentUIN [PK, FK to Student.UIN], sectionCourseID [PK, FK to Course.courseID], sectionProf [PK, FK to Professor.profID], sectionTerm [PK], credits: int)

3. Section(studentUIN [PK, FK to Student.UIN], sectionCourseID [PK, FK to Course.courseID], sectionProf [PK, FK to Professor.profID], sectionTerm [PK], buildingTaught: varchar(255), roomNum: int)
    a. Weak entity

4. Professor(profID: int [PK], firstName: varchar(255), lastName: varchar(255), deptID: varchar(255) [FK to Department.deptID])

5. Department(deptID: varchar(255) [PK], deptName: varchar(255), mainBuilding: varchar(255))

6. Review(reviewID: int [PK], reviewerUIN: int [FK to Student.UIN], courseID: varchar(10) [FK to Course.courseID], profID: int [FK to Professor.profID],reviewStanding: int, courseTerm: varchar(4), profComments: varchar(5000), courseComments: varchar(5000), reviewerGrade: varchar(2), difficulty: int, usefulness: int, avgHrs: real, maxHrs: real, profRating: int)

Entities, Assumptions, and Comments

1. Course
    a. This entity stores information about different courses that are offered at UIUC
        i. Even if a course is no longer taught, its information can be kept
    b. A course is labeled with an ID like ‘cs411’
    c. This ID is unique since it contains both the department identifier and the course number in a single attribute
    d. When user’s search for a particular course to see reviews for, this is what they will search
    e. We assume a course has a FK relation with Department
2. Section
    a. This entity stored information about different sections that are and have been offered at UIUC
    b. Each section is uniquely identified by the course it is allocated to (courseID), the professor that taught it (profID), and the term that it was offered.
    c. Though this could be a relationship for a general case, in our case we treat it as a main entity since it includes information from three different tables
    d. We assume it is possible for multiple professors to teach the same course during the same semester (for larger classes) hence profID is part of the primary key
    e. However, we assume that only one professor can teach a single section
        i. E.g can’t have one professor teach the first half of a term and a second professor teach the second half
    f. ‘Term’ is of the format ‘sp20’ or ‘fa21’, this specifies when the course was offered. This is also part of the key since we assume the same professor could have taught the same course at some point in the past so it was still part of a different section
        i. This is an important consideration since reviews will not be limited to the current term since that largely defeats the purpose of the app
    g. We could provide a unique identifier to each section, but when a review is made the student would only know the name of the course, the prof that taught it when they took it, and the term they took the course. It would be inconvenient to have to search up a CRN for example
        i. We also do not know if CRN is unique over multiple semesters so we would need to generate our own unique ID for each section which adds complexity when parsing reviews which do not have this identifier
    h. For the reasons mentioned in *g*, we leave Section as a weak entity since it can only be uniquely identified by by using the primary key of multiple different tables
    i. From this, we assume a Section has a FK relationship with Course, Professor, and Student
3. Professor
    a. This entity stores information about the professors that teach at UIUC
        i. This information is kept even after a professor leaves UIUC
    b. We assume that each section is taught by only a single professor

        i. E.g can't have one professor teach the first half of a term and a second professor teach the second half
    c. We assume that each professor is only associated with a single department rather than being listed under multiple departments
    d. From this, we assume a Professor has a FK relationship with Department
4. Student
    a. This entity stores information about students who are enrolled in UIUC
        i. This information is kept even after a student graduates since their reviews will still be relevant
    b. Student refers exclusively to UIUC students who have a UIN
    c. Each student will log in via their password to access the application
    d. We assume that the lower bound for the number of sections a student is enrolled in is zero (ie they can be enrolled in no sections) e.g. because they are taking a gap semester.
    e. We assume a student is only listed under a single department, regardless of their standing
    f. We assume a Student has a FK relation with Department
5. Department
    a. This entity stores information about departments at UIUC
    b. We assume that both professors and students can only be listed under one department
    c. We assume that each course can only be listed under one department
6. Review
    a. This entity stores information about a review for a particular course and the corresponding professor at the time the reviewer took the class. Students specify the course, the professor, and the term they took the course as well as the grade they got in the course and their standing at the time of taking it. They then provide a written review for the course and for the professor (courseComments and profComments). They also provide data like the average and maximum number of hours they spent working on the class (to their best estimate). Lastly, they provide a rating of the professor on a scale of 0 to 5 and a rating for both the difficulty and usefulness of the course also on a scale of 0 to 5
        i. courseComments and profComments are separate so that students can focus on both separately rather than blending them into a single comment.
        ii. This will be useful later if users search only for a professor, we do not want to show reviews that only talk about the course and barely about the professor themself
        iii. The ratings and time commitment are by no means objective, the student just provides and estimate so other students can have an idea of the approximate workload for the course and see if it was deemed a difficult and valuable course

    1. E.g the avgHrs and maxHrs are entered based on the following thought process: ‘I spend roughly *avgHrs* hours a week on class Y, during the most hectic week i spent *maxHrs* hours’
b. We assume that a review can only be made by existing UIUC students who are able to authenticate as mentioned for the Student entity. Furthermore, a review can only be made for an existing class, existing professor and valid term (ie can’t make a review for ‘sp26’ when its ‘sp22’). This will be ensured through the application interface so students can only submit a review if these requirements are met (via a drop down menu or something similar when selecting the course and professor)
c. We assume that each review is uniquely identified by its own ID
d. We assume that each review has a FK relationship with Student, Professor, and Course

Relationships and Cardinality

1. A section can have many students, and a student can be enrolled in many sections. [many-many]
2. A professor can teach many sections, and a section can be taught by only one professor [one-many]
3. A course hase multiple sections, and each section corresponds to a single course [one-many]
4. A professor can belong to one department, and a department can have many professors [one-many]
5. Courses only belong to one department, and departments can have many courses [many-one]
6. Each department has multiple students, and each student is in a single department [one-many]
7. A student can write many reviews, and a review belongs to a singular student [one-many]
8. A review is made for one course, and a course can have many reviews [many-one]
9. A review is made for one professor, and a professor can have many reviews [many-one]