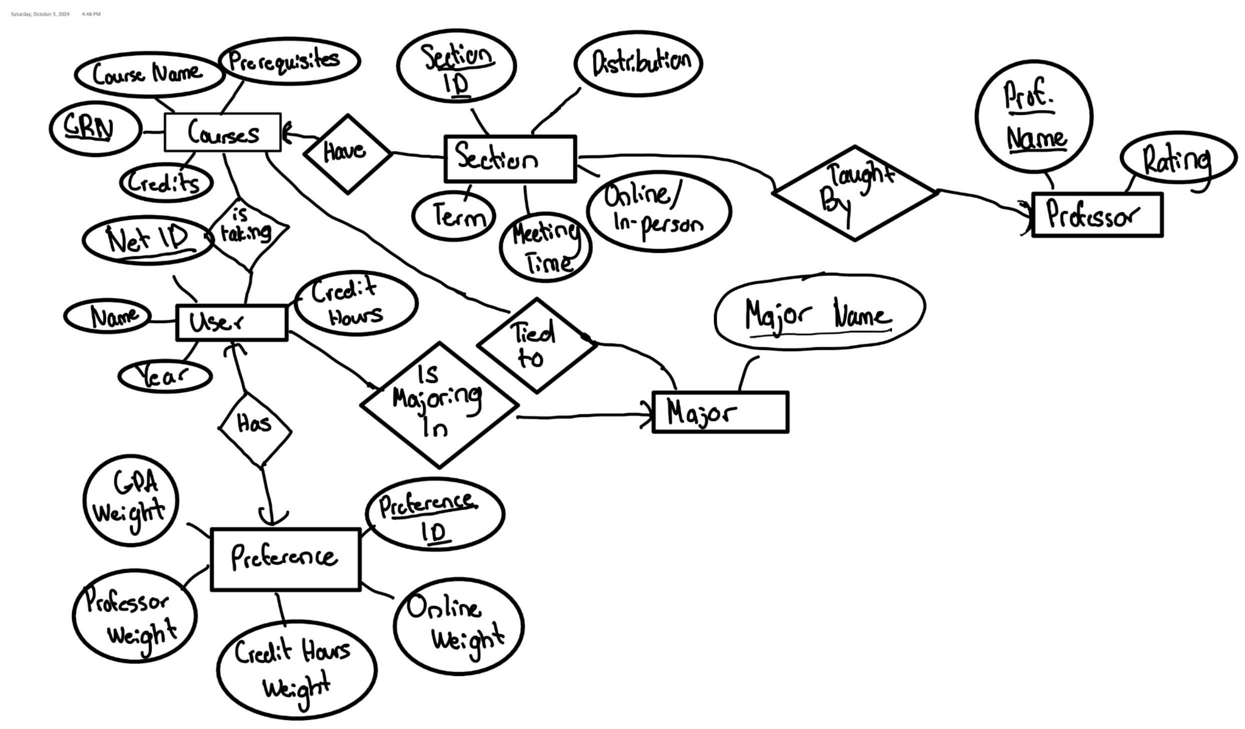
Entity-Relationship Diagram:



Entities and Attributes:

Courses:

- <u>CRN</u>
- Course Name
- Credits
- Prerequisites

Description: The Courses entity houses all courses available in the UIUC’s Course Catalog and includes important details about each course: CRN, course name, any prerequisites needed, and how many credits the course is. We modeled Courses as an entity because we will need all courses available when we take into consideration the courses a user has already taken, what courses are required per major, and what sections pertain to what course. The Courses entity has a relationship “have” with Sections to denote the many sections one course can have. The relationship “is taking” with User demonstrates the courses that a user has taken, which helps us determine what classes or prerequisites they have taken. The “tied to” relationship with Major displays the courses that are required for each major, which will be important when determining the best classes to take.

Section:

- <u>SectionID</u>

- Term
- Meeting Time
- Online/In-person
- Distribution

Description: The Section entity contains all relevant information for each course's section. Since a course can have multiple sections and since these sections have different attributes—meeting times, term, if the class is online, and grade distribution—it would not suffice to have sections as an attribute of the Courses entity. Additionally, we need the Section entity to recognize the relationships between the Professor entity via "taught by." The relationship allows us to see which professor teaches what section and what that professor's rating is on RateMyProfessor. Section also has a relationship with Courses, which we have already discussed above.

User:
- Name
- <u>NetID</u>
- Year
- Credit Hours

Description: The User entity represents the user's information that will be used to predict courses. This needs to be a separate entity because it holds key attributes that identify each user, such as their name, NetID, academic year, and total credits earned. Keeping this as a separate entity allows the system to efficiently track user-specific data independently of other entities, such as courses or departments. Also, this allows for easier updates and deletions when new users are added.

Professor:
- <u>Prof. Name</u>
- Rating

Description: The Professor entity represents the instructor for each course section. It will have a unique professor name and an associated rating which will be pulled from the RateMyProfessor database. Creating a distinct Professor entity, we can easily track and manage instructor-related information across multiple course sections. It allows students to consider the professor's identity and reputation when choosing courses.

Preference:
- <u>Preference ID</u>
- GPA Weight
- Professor Weight
- Credit Hours Weight
- Online Weight

Description: A preference entity stores all the weights provided by the user for generating personalized recommendations and is uniquely identified by its Preference ID. Each preference is linked to a specific user, and each user has a corresponding preference entity. By separating preferences into their own entity, users can easily update their preferences without altering any user-related attributes.

Major:

- <u>Major Name</u>

Description: A major entity simply contains a unique Major name identifier, which is the major name. We made this its own separate entity because both student users and courses need to be associated with a major. By having a distinct major entity, we ensure that changes or updates to a major can be managed independently, making it easier to manage relationships between users, courses, and majors.

Relationships:

- Courses → Section (one-to-many): Each course has multiple sections from which a student can choose.
- User → Courses (many-to-many): Users can have multiple completed courses, and multiple users can take each course.
- Professor → Section (one-to-many): Each section has one professor teaching it, and a professor can teach multiple sections.
- User → Preference (one-to-one): Each user can have one set of preferences, and a certain preference is tailored to one specific user.
- User → Major (many-to-one): A user has one major, but multiple users can have the same major.

Normalization of Database:

<u>Functional Dependencies:</u>

CRN → Course Name, Credits, Prerequisites

SectionID → Term, Meeting Time, Online/In-person, Distribution, ProfName, CRN

NetID → Name, Year, Credit Hours, PreferenceID, Major Name

ProfName → Rating

PreferenceID → GPA Weight, Professor Weight, Credit Hours Weight, Online Weight, NetID

BCNF:

-CRN → Course Name, Credits, Prerequisites

Courses (CRN, CourseName, Credits, Prerequisites)

This is in BCNF since the relation is decomposed into Courses and CRN+ is a superkey for the attributes Courses.

-SectionID → Term, Meeting Time, Online/In-person, ProfName, Distribution, CRN

Section(SectionID , Term, Meeting Time, Online/In-person, ProfName, Distribution, CRN)

This is in BCNF since SectionId+ is a superkey for Section, and uniquely determines each of the associated dependencies.

-NetID → Name, Year, Credits, PreferenceID, Major Name

User(NetID, Name, Year, Credits, PreferenceID, Major Name)

This is in BCNF since NetID+ is a superkey for User, and uniquely determines each of the associated dependencies. It also uniquely identifies a single PreferenceID and a single Major.

-ProfName→ Rating

Professor (ProfName, Rating)

This is in BCNF since the Professor , and now ProfName+ is a superkey for Professor.

-PreferenceId → GPA Weight, Professor Weight, Credit Hours Weight, Online Weight, NetID

Preference(PreferenceID, GPAWeight, ProfessorWeight, CreditsWeight, OnlineWeight, NetID)

This is in BCNF since PreferenceId+ is a superkey for Preference, and uniquely determines each of the associated dependencies. It also uniquely identifies a single NetID.

Conversion to Relational Schema:

Courses (CRN: INT [PK], CourseName: VARCHAR(255), Prerequisites: VARCHAR(255), Credits: INT)

User (NetID: VARCHAR(10) [PK], Name: VARCHAR(255), Year: INT, CreditHours: INT, PreferenceID: INT [FK to Preference.PreferenceID], MajorName: VARCHAR(255) [FK to Major.MajorName])

Enrollments (NetID: VARCHAR(10) [FK to User.NetID], CRN: INT [FK to Courses.CRN])

Preference (PreferenceID: INT [PK], GPAWeight: INT, ProfessorWeight: INT, CreditsWeight: INT, OnlineWeight: INT, NetID: VARCHAR(10) [FK to User.NetID])

Major (MajorName: VARCHAR(255) [PK])

TiedTo (MajorName: INT [FK to Major.MajorName], CRN: INT [FK to Courses.CRN])

Section(SectionID: VARCHAR(10) [PK], MeetingTime: VARCHAR(10), Online/In-Person: VARCHAR(10), Term: VARCHAR(10), ProfName: VARCHAR(255) [FK to Professor.ProfName], Distribution: INT, CRN: INT [FK to Courses.CRN])

Professor (ProfName: VARCHAR(255) [PK], Rating: INT)

ProfessorSection((ProfName, SectionID) [PK], ProfName: VARCHAR(255) [FK to Professor.ProfName], SectionID: VARCHAR(255) [FK to Section.SectionID] **)**