

Explanation

General Course: There is one course (primary key) for each (Department, Course #, Name) in General Course. Similarly, (Department, Course #, Name) is the foreign key link each section from **Section Attributes** to the general courses. Many to exactly one relationship.

Section Attributes: The key for each section is (CRN, Semester, Year) *this can potentially be combined to (CRN, YearTerm) if we use the YearTerm format like GPA History*. The foreign key (Course#, Name, Department) is not unique in **Section Attributes**, but since the Primary Key in **General Course** is, that is fine. **User Feedback** can link to **Section Attributes** using (CRN, semester, year) as the foreign key. *Concern: aggregating data for a course with multiple sections and CRNs (lecture, discussion, lab) for one registration.*

User Feedback: The key for **User Feedback** should be (Year, CRN, Semester, Email) since an individual should only have one review per course per semester (*still run into some of the multiple section issues*). Many to one relationship from **User Feedback** to **Section Attributes** with (CRN, Semester, Year) on both since we can have multiple feedback entries on one section, but they can each only be about one section.

Professors: The **Professors** and **Section Attributes** is a Many to many relationship since professors might teach multiple courses, and one course might have multiple professors. Professors should be unique by NetId/email. Linked by (LastName First Initial, Department) in **Section Attributes** to (Last Name, First Name, Department) in **Professors**.

GPA: (Term, CRN) in the **GPA** dataset corresponds to unique entries, and they can foreign key to **General Course** as (Subject, Number, Course Title) to (Department, Course #, Name). This is a many to one relationship because each GPA entry should only correspond to one General Course while one course will have many GPA entries. There are more nodes than what is shown on the entity for grades from A+ to F, including every other letter grade step in between.

Users: Each user is unique by their email (will have to be UIUC email they verify or that we would get from Shibboleth). The Email field on **User Feedback** is a foreign key for the entries in **Users** in a many to one relationship since each user can have many feedback forms, but a feedback form must only be from one user.

NOTE: Name of Primary Instructors is truncated in Wade's dataset

Concerns: Professor info in Section attributes is a bit concerning for linking those to a specific professor (if there is a professor in the same department with the same first initial and last name. Unlikely, but not impossible). The issue with using CRN for a course with multiple sections and getting the "Primary Instructors" is going to be an interesting challenge. Might be able to overcome this by linking by Professor instead of CRN since the user wouldn't really care about the data from each section, just how the professor was as a whole for a class. **Potential Breakthrough:** the .xml data includes what type of section it is with a short identifier, something that is also used in the GPA history. If we only do LEC (lectures), LCD (lecture-discussions), and ONL (online sections); that would likely get us what we need (potentially taking DIS {discussions} if the others are available).

Functional Dependencies:

Course #, Name, Department -> Course Description, Prereqs, Credit hours

CRN, Semester, Year -> Professor name, Course Department, Course Name, Course #

User email -> Password, user name,

User feedback form email, CRN, Semester, Year -> Testimony, Rating, Difficulty, Time commitment, Timestamp,

Professor NetId -> Professor First Name, Professor Last Name, Professor Department

Course History Term, CRN year -> # of each grade, Subject, Course Title, Course #, Course Instructor

Normalization:

Since every left side is a key for the table it is a part of, we are in 3NF format. 3NF seemed like the most simple way to go as we were defining how we wanted to structure our data, so that's why we decided to use it.