Exercise in English to UML

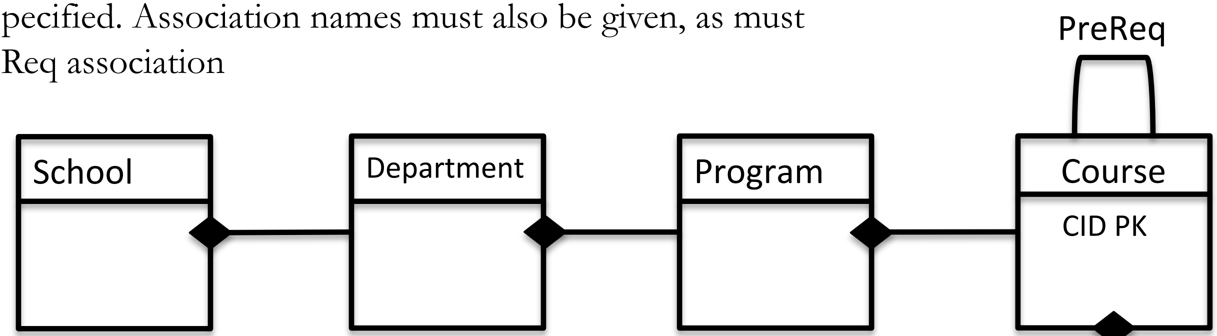
The standard model for a university's database would include a snippet
like this – that

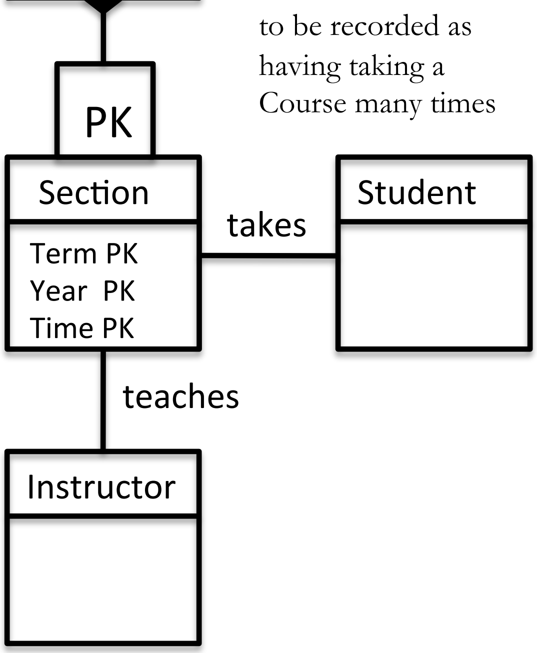
- Student are recorded as taking sections (of courses),
- Instructors are recorded as teaching sections (of courses),
- Courses are associated with Programs (e.g., CS, CompE, EE, American Studies),
- Programs are associated with Departments (e.g., EECS),
- Departments are associated with Schools (e.g., Eng, A&S, Peabody)

The UML models on the following pages get farther and farther away from the simple (and incomplete) specification above. I hope you are not confused by that. Each design reconceives the design task in a modest way

This UML approximates functionality of the spec on the previous page (with the addition of prerequistes) and leaving many of the multiplicity (aka cardinality) constraints to be still specified. Association names must also be given, as must role names for the PreReq association

◆ i.e., 1..1

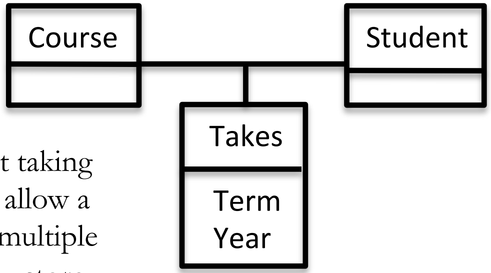




This construct allows a student to be recorded as having taking a Course many times

- Student are recorded as taking sections (of courses),
- Instructors are recorded as teaching sections (of courses),
- Courses are associated with Programs (e.g., CS, CompE, EE, American Studies),
- Programs are associated with Departments (e.g., EECS),
- Departments are associated with Schools (e.g., Eng, A&S, Peabody)

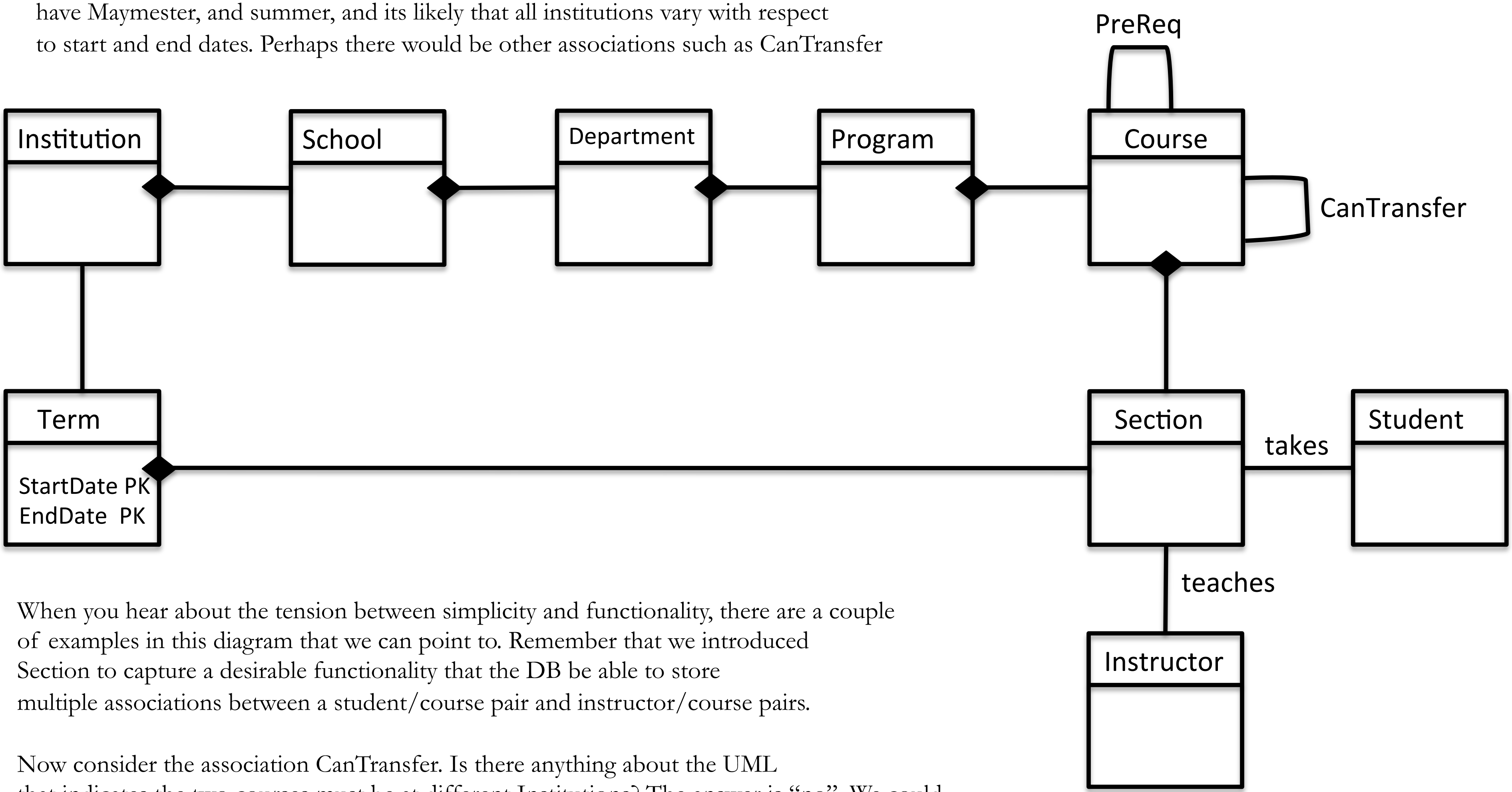
Note that if a student could only be recorded once for taking a course, and a instructor could only be recorded once as teaching a course, then we could probably exclude the Section class and include a snippet such as:



This construct would only allow one record to be recorded of a given student taking a given course. Policy might well allow a Student to take the same course multiple times, but the database could only store one (e.g., the most recent). It is probably the case that the University has Sections for Courses BECAUSE of a desire to store all instances of a student taking a course, or an instructor teaching a Course.

In the snippet above, the Primary Key for Section would be (CID, Term, Year, Time). What would we have to change if we wanted Instructor to be part of the primary key of Section? What would be the pros of having a unique Section ID across all of a university's offerings?

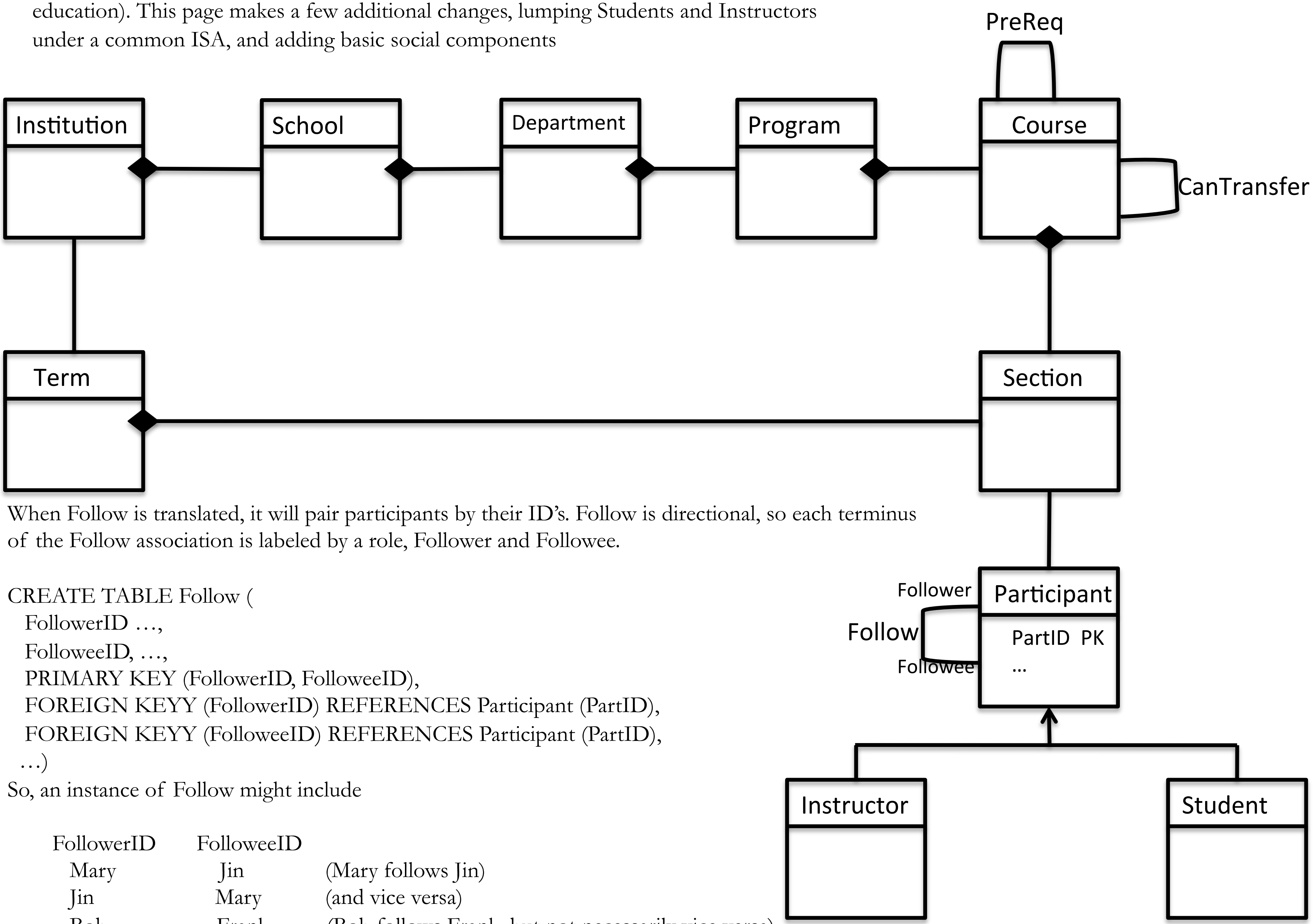
When we consider a database, not for a single institution (e.g., Vanderbilt), but to store records across many institution, we would likely add an Institution class, as well as possibly others, such as Term (reflecting that some courses are on semester systems, some on quarters, some have Maymester, and summer, and its likely that all institutions vary with respect to start and end dates. Perhaps there would be other associations such as CanTransfer



When you hear about the tension between simplicity and functionality, there are a couple of examples in this diagram that we can point to. Remember that we introduced Section to capture a desirable functionality that the DB be able to store multiple associations between a student/course pair and instructor/course pairs.

Now consider the association CanTransfer. Is there anything about the UML that indicates the two courses must be at different Institutions? The answer is “no”. We could probably redesign the DB UML so that a constraint that the tow coures participating in the roles of CanTransfer were clear by the UML itself, but it would likely be quite a bit more complicated, with other classes and associations required. Rather, we might simply choose to annotate the UML with these latter requirements on CanTransfer, and enforce the constraints with assertions or in-table checks when we translated to SQL.

Note that the DB design of the previous page allows both students and instructors to be associated with courses at different Institutions, so we have a start on a DB design for the "unbundled university" (a favorite topic of mine, as one who works with online education). This page makes a few additional changes, lumping Students and Instructors under a common ISA, and adding basic social components



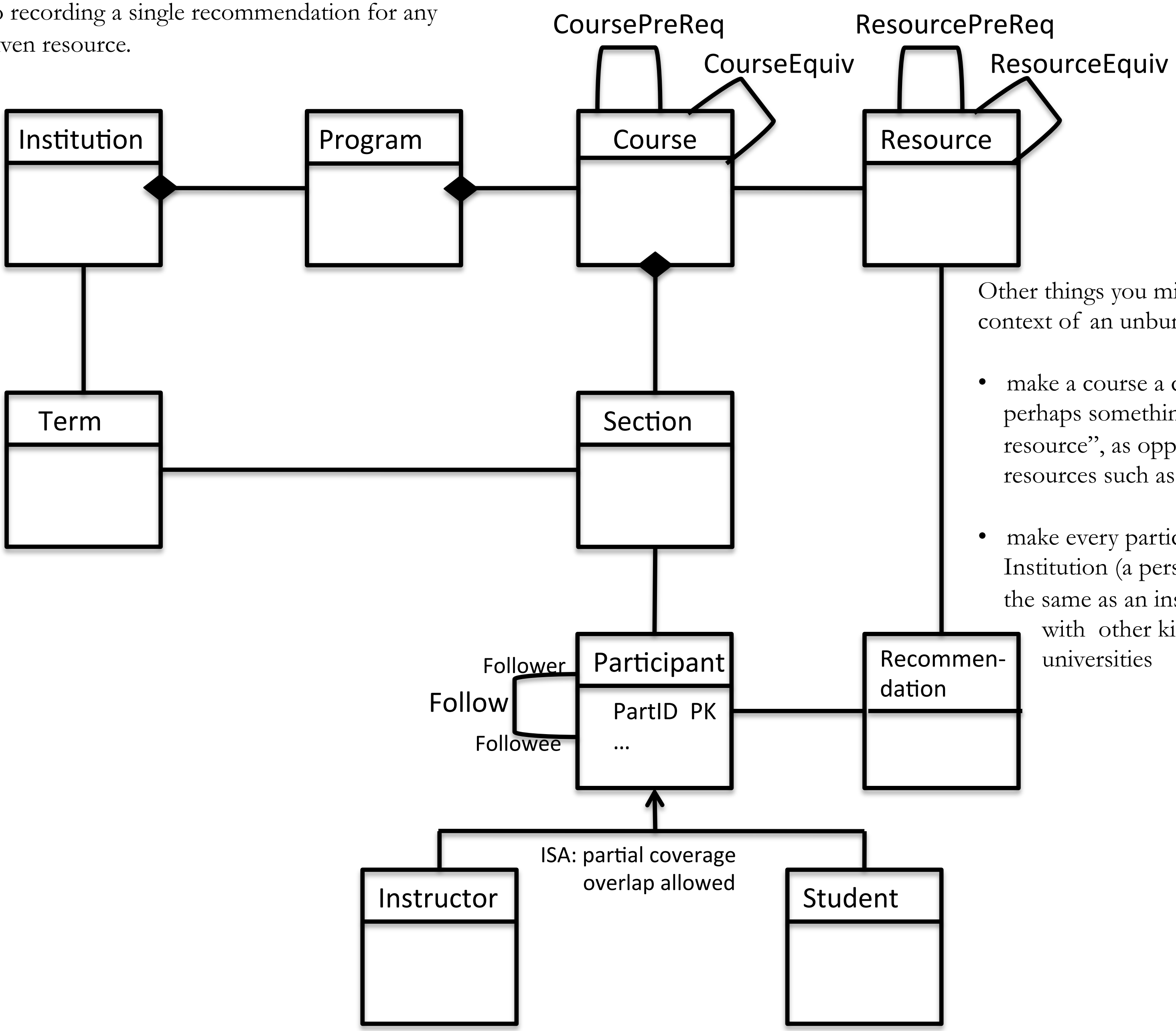
When Follow is translated, it will pair participants by their ID's. Follow is directional, so each terminus of the Follow association is labeled by a role, Follower and Followee.

```
CREATE TABLE Follow (
  FollowerID …,
  FolloweeID, …,
  PRIMARY KEY (FollowerID, FolloweeID),
  FOREIGN KEYY (FollowerID) REFERENCES Participant (PartID),
  FOREIGN KEYY (FolloweeID) REFERENCES Participant (PartID),
 …)
```

So, an instance of Follow might include

FollowerID	FolloweeID	
Mary	Jin	(Mary follows Jin)
Jin	Mary	(and vice versa)
Bob	Frank	(Bob follows Frank, but not necessarily vice versa)

This diagram takes things a bit further, representing courses as composed of resources, adding recommendations by participants for resources (note that we could simply have Recommendation as an association, rather than a Class, but that would limit a Participant to recording a single recommendation for any given resource.



Other things you might consider in the context of an unbundled university:

- make a course a child (ISA) of a resource, perhaps something called a "composite resource", as opposed to "Primitive" resources such as videos, etc

- make every participant and kind (isa) of Institution (a person as an institution is not the same as an institution as a person), along with other kinds of Institutions, like universities