**Classwork 1.1 Solution**

The relations represent necessary simplifications for presentation in a textbook. It is important to grasp the gap between toy examples in textbook and the complexity of real-world applications.

Examples of possible issues, limitations and questions:

* Class (1, 2) may not be meaningful enough.
* Course\_number (e.g. CS1310 may not be atomic)
* In the relation Course, if Course\_number is a candidate key, changing course name means the deletion of the old name.
* In the relation Course, Department name may be redundant as it may be inferred by Course\_number.
* In SECTION, using Instructor names as the foreign key may be problematic as instructor last name may not be unique.
* In SECTION, using Semester and Year to identify teaching section may not be sufficient. There may be more than one teaching section (Summer 1, Summer 2, Summer 3), etc. Using a composite key as a foreign key is not a good design practice.
* Year: 07: Y2K+1 problem. Many universities are more than 100 years old.
* PREREQUISUITE: can only have other courses but not generic description (such as a programming language)
* PREREQUISITE: cannot specific complicate condition, such as OR (C or Java)
* Many important information may be missing: e.g. student info (address, phone, email, DOB, etc), instructor information, course offering information, etc.
* Examples of assumptions and limitations
	+ At most one section per course in a given semester/year.
	+ GPA calculation not embedded in the relation schema.
* Examples of questions:
	+ Can a student take the same course again?
	+ Can the department change the title of a course?
	+ How many departments are there?
	+ Can a course be jointly offered by more than one department?
	+ Can an instructor change the grade of a student in a course?
	+ Are there any limitations of time for grade changes?

**Some lessons learnt:**

* DB relation schemas are relative to the business needs. One size does not fit all.
* Many questions can and should be asked.
* Modeling is a process of refinement and correction until there are sufficient accurate details for design and implementation.
* Given instances of relations, questions can be asked on the entire modeling and implementation lifecycle.