CMSC 23500 — Introduction to Database Systems
Discussion Session #2

April 7, 2008

The purpose of this discussion session, like the previous one, is for you to design a database schema. However, unlike the previous discussion session, you will not be given a description of the "mini-world" you have to model. Instead, you will have to extract the information necessary to design the database from a "client": your Teaching Assistant. The client will start with a brief presentation of his data storage requirements to give you an idea of what the general domain is. After that, you will have to ask the client questions to understand all the details.

When asking questions, you may want to take the following into account:

- The client does not know anything about database theory. If you refer to "entities", "relationships", "primary keys", etc. he will have no idea what you're talking about.
- Data storage requirements are usually driven by a set of applications that will depend on a database. Asking about this can help you scope the domain of your database, since your database must model, at least, the information consumed by those applications. Furthermore, this is a good starting point for determining the entities in your model. For example, if the client says "We will need an application to manage all the grants under the supervision of our scientists", then *grants* and *scientists* will probably end up being entities in your database. Ask the client for more details about the information associated with them.
- For the purposes of this discussion, we will assume that the database won't be redesigned in the future to accomodate more applications (in practice, you do want to model information that will possibly be used by future applications).
- Your schema will model only a small subset of the client's "mini-world". Make sure you don't get caught up in irrelevant details; you may end up modelling data that the client will have no use for.
- Be careful when modelling relationships, which tend to be trickier than entities. For example, the statement "a department has several employees" could be interpreted multiple ways:

- Each department has several employees, and each employee is assigned to a single department.
- Each department has several employees, and each employee can be assigned to more than one department.
- Each department has several employees, and each employee can be assigned to more than one department, and there is a distinction between the "primary department" and all the "secondary departments".

So, make sure you probe the client about the exact nature of a relationship between entities.