## CS 443 Database Systems

## MyDB 3 (50 points)

## Due Friday July 7th in the drop box

**MyDB 3: Step 3 of your Personal Database Application (MyDB)**

Note: This assignment is a modification of material developed by Stanford Database Group

1. (50 pts) Develop a substantial amount of data for your database and load it into your relations. To create the data, write a program in any programming language you like that creates large files of records in a format acceptable to the MySQL loader, then load the data into your MyDB relations. If you are using real data for your MyDB, your program will need to transform the data into files of records conforming to your MyDB schema. The rest of you will write a program to fabricate data: your program will generate either random or nonrandom (e.g., sequential) records conforming to your schema. Note that it is both fine and likely for your data values – strings especially – to be meaningless gibberish.

The point of generating large amounts of data is so that you can experiment with a database of realistic size, rather than the small “toy” databases often used in classes. The data you generate and load should be on the order of:

* At least three entities with thousands of tuples
* At least one entity with hundreds of tuples

If the semantics of your application includes relations that are expected to be relatively small (e.g., schools within a university), it is fine to use some small relations, but please ensure that you have relations of the sizes prescribed above as well. When writing a program to fabricate data, there are two important points to keep in mind:

1. In many cases, you probably know that an attribute or set of attributes in a relation will serve as a key. If so, be sure not to generate duplicate values for these attributes.
2. Your MyDB almost certainly includes relations that are expected to join with each other. For example, you may have a Student relation with attribute courseNo that is expected to join with attribute number in the relation Course. In generating data, be sure to generate values that do join—otherwise all your interesting queries will have empty results! One way to guarantee joinability is to generate the values in one relation, then use the generated values in one relation to select joining values for the other relation. For example, you could generate course numbers first (either sequentially or randomly), then use these numbers to fill in the courseNo values in the Student relation.

Turn in the following:

1. Your program code for generating or transforming data. A sample data file and a sample control file.
2. Do a describe statement and count (\*) on all your tables. Spool the output and turn it in (screen shots to show your work).
3. A small sample of the records generated from any two entities and a relationship relating them. Show (5 or so records per relation), and a screen shot showing the loading of your data into any one of your tables. You can use LIMIT clause to restrict the number of rows to be displayed. For example, by doing this you see the first 6 records.

SELECT \*

FROM Employee

Limit 6;

Don’t forget to save a copy of your MyDB 3 for reference as you do Step 4 of the MyDB.