CS145 Programming Assignment #1

Due Friday April 23*

IMPORTANT: *Start early—this programming assignment requires a fair amount of work. Some helpful notes and hints can be found toward the end of this handout.*

1. **Getting Started.**

   Familiarize yourself with the Oracle RDBMS by reading *Getting Started With Oracle*.† Log into Oracle, change your password, and try some of the examples in the guide. You don't need to turn anything in for this part.

2. **Creating the Database Schema.**

   Before you start this problem, you should complete Problem #4 of Written Assignment #3.

   Create relations for your PDA based on the final relational schema you came up with once you finished Problem #4 of Written Assignment #3. Use the `CREATE TABLE` command described in *Getting Started With Oracle*. Pick a suitable datatype for each attribute. Page 286 of the textbook gives you the principal options regarding types. If any of your attributes is a date or time, check out *Oracle Dates and Times*.‡

   Turn in the list of commands you used to create your database schema (it is a good idea to keep a copy of this list for the rest of the course). Also turn in a script showing an Oracle session in which your relations are successfully created. For instructions on how to record an Oracle session, see *Getting Started With Oracle*.

3. **Creating a Small Database Instance.**

   For each relation in your PDA, create a file containing a few (approximately 5–10) records of "meaningful" data for your PDA. Then use the Oracle bulk-loading facilities to insert those records as tuples into your relations. Refer to *Using the Oracle Bulk Loader*§ for file format and how to load records into Oracle. Turn in a listing showing the files you created and the successful loading of the data into Oracle.

4. **Creating a Large Database Instance.**

   In Problem #3 you hand-created a small amount of data for your PDA. In this problem you will generate a large database. Write a program in any programming language you like that creates large files of records for each of your PDA relations. If you have available "real" data for your PDA, then your program will need to transform the data into files of records conforming to your PDA schema. Otherwise you will write a program to *fabricate* data: your program will generate either random or non-random (e.g., sequential) records conforming to your schema. Note that it is expected that your data values—strings especially—may be meaningless gibberish. The point of generating large

*Please refer to CS145 Course Information Page (`http://www.stanford.class/cs145/info.html`) for submission instructions and late policy.

†`http://www.stanford.edu/class/cs145/or-intro.html`

‡`http://www.stanford.edu/class/cs145/or-time.html`

§`http://www.stanford.edu/class/cs145/or-load.html` amounts of data is so that you can experiment with a database of realistic size. The data you generate and load should be on the order of:

- At least two relations with thousands of tuples
- At least one relation with hundreds of tuples

Your Oracle account has a disk quota of 10 megabytes, which should be enough. If your application naturally includes relations that are expected to be relatively small (e.g., schools within a university), then 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:

(a) Although you have not (yet) declared keys in your Oracle relations, 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.

(b) Your PDA almost certainly includes relations that are expected to join with each other. For example, you may have a `Student` relation with attribute `courseNum` that's expected to join with attribute `number` in relation `Course`. When generating data, be sure to generate values that actually do join—otherwise all of your interesting queries will have empty results! There are a couple of ways to properly generate joining values. One is to generate records for multiple relations at once. Another is to generate the values in one relation R_1, then use the generated values in R_1 to select joining values for R_2.

Turn in your program code for generating or transforming data, a small sample of the records generated for each relation (5 or so records per relation), and a script showing the successful loading of your data into Oracle.

Notes and Hints:

- You will be using both your small (Problem #3) and large (Problem #4) database instances for the rest of the course. The idea is to use the small instance to experiment on meaningful-looking data, and the large one to experiment on something of more realistic size. You can keep the two going in parallel either by reloading the data each time you use it, or by creating two sets of relations with slightly different names.

- If you find yourself typing the same SQL commands over and over again, put them into a `.sql` file and use "`@`" to execute it in `sqlplus` (see "Executing SQL From a File" in *Getting Started With Oracle*). For example, you might have one `.sql` file containing all your `CREATE TABLE` commands, and another one with all your `DROP TABLE` commands.