Programming Assignment #1
Due Wednesday April 29

Programming assignments will be turned in electronically by e-mailing your final script to `cs145ta@cs`. Assignments must be *received* by the TA before Wednesday midnight, so we suggest that you mail your assignment from a Stanford machine and allow at least 15 minutes leeway in case of sluggish mail delivery. Assignments less than 24 hours late will be penalized 10% or will use one of your four "chits". Assignments more than 24 but less than 48 hours late will be penalized 30% or will use two of your four "chits". No assignments will be accepted more than 48 hours late—i.e., all programming assignments must be received by the TA before Friday midnight. For programming assignments the turn-in procedure and late policy is identical for on-campus and SITN students.

You should complete Problem #7 on Written Assignment #3 before beginning this programming assignment.

---

1. Familiarize yourself with the Oracle relational DBMS by reading Handout #10: *Introductory Guide to Oracle*, logging into Oracle, trying some of the examples in the handout, and experimenting with the `help` command. You don't need to turn anything in for this problem.

2. Create relations for your PDA based on the final relational schema you came up with once you finished Problem #7 on Written Assignment #3. Turn in a script showing an Oracle session in which your relations are successfully created.

3. 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 Handout #11: *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. 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 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

   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 `course-no` 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: You will be using both your small (Problem #3) and large (Problem #4) databases for the rest of the course. The idea is to use the small database to experiment on meaningful-looking data, and the large one to experiment on something of more realistic size. You can keep the two databases going in parallel either by reloading the data each time you use it, or by creating two sets of relations with slightly different names. As described in Handout #10, in order to get rid of a table called *Foo* in Oracle, issue the command:

```
drop table foo;
```

If you want to get rid of all tuples in *Foo* without deleting the table itself, issue the command:

```
delete from foo;
```