# Programming Assignment #2
# Due Wednesday May 13

As with Programming Assignment #1, you will turn this assignment in electronically by e-mailing your final script to `cs145ta@cs`. On-time assignments must be received by the TA before Wednesday midnight. The late policy for this assignment is the same as for Programming Assignment #1.

Before beginning this assignment please read Handout #25: *Oracle 7.3.2 SQL*. In particular, be sure to understand the sections on "Basic SQL Features", "Indexes", and "Timing SQL Commands". We also recommend doing Written Assignment #4 before doing this programming assignment, since the written assignment will familiarize you with most constructs of SQL.

---

1. Develop and test:

   (a) At least eight SQL data retrieval (`select`) commands.

   (b) At least two each of the four types of SQL data modification commands: `insert` a single tuple, `insert` a subquery, `delete`, `update`.

   Please note:

   - For this assignment you will be invoking your SQL commands interactively through `sqlplus`, as described in Handout #10: *Introductory Guide to Oracle*. Of course you should certainly build a script file, rather than typing the queries in each time you run them.
   - Please write "interesting" queries. You should try to use most or all of the SQL constructs discussed in class and in the textbook (subqueries, aggregates, set operators, etc.). You will not receive full credit if your queries and modifications are extremely simple.
   - We suggest that you experiment with your SQL commands on your small hand-created database before running them on the large database for which you generated data. Initial debugging is much easier when you're operating on small amounts of data. Once you're confident that your commands are working, run them on your complete database.
   - If you discover that most or all of your "interesting" queries return an empty answer on your large database, then you probably didn't follow the instructions in Programming Assignment #1 for generating data values that join properly. You will need to modify your data generator accordingly.
   - Turn in a copy of all of your SQL commands, along with a script illustrating their execution. Your script should be sufficient to convince us that your commands run successfully. Please do not, however, turn in query results that are thousands (or hundreds of thousands) of lines long!

2. In Problem #1 you probably discovered that some queries run very slowly over your large database. As discussed in class, one principal technique for improving the performance of queries is to create *indexes*. An index on an attribute $A$ of relation $R$ allows the database to quickly find all tuples in $R$ with a given value for attribute $A$ (which is useful when evaluating selection

or join conditions involving attribute $A$). An index can be created on any attribute of any relation, or on several attributes combined. The syntax for creating indexes in Oracle is given in Handout #25: *Oracle 7.3.2 SQL*. Create at least four useful indexes for your PDA. Run your queries from Problem #1 on your large database with the indexes and without the indexes. Turn in a script showing your commands to create indexes, and showing the relative times of query execution with and without indexes.

Please note:

- As mentioned in Handout #25: *Oracle 7.3.2 SQL*, Oracle automatically creates indexes for attributes declared as keys. Since you have not yet declared keys in your schema, you may consider creating indexes on key attributes as appropriate, and such indexes should be declared as `unique`.
- As described in Handout #25: *Oracle 7.3.2 SQL*, in order to set the system to show query execution times you must issue the command "`set timing on;`" at the `sqlplus` prompt.
- Your timings will be affected by external factors such as system load. However, for some of your queries, with appropriate indexes you should see a consistent dramatic difference between the execution times with indexes and the times without. If others of your queries do not show performance improvement even when relevant indexes are created, please include a short note suggesting why this might be the case.