Lab 3: One more dataset
(dates and more)

Due date: Tuesday, October 9, **before the Lab period starts!!!**

Extra Credit: Lab assignments submitted by the end of the October 4 (Thursday) lab period will be assessed 15% extra credit.

Lab Assignment

Assignment Preparation

This is an individual lab. Each student has to complete all work required in the lab, and submit all required materials **exactly as specified** in this assignment.

For this assignment, you will be using your Oracle account. The accounts have been assigned in class on Tuesday, October 2. If you have not recieved your Oracle account information, please contact the instructor.

You will be using Oracle's SQL*plus client and the Oracle DBMS server `ora10g`. Please refer to your SQL*plus handout for more information about starting SQL*plus and working with it.

The Task

You will be working with a new course dataset, MARATHON. The MARATHON dataset consists of a single CSV file documenting the performance of participants of a half-marathon race. The performance is tracked for the entire race, as well as within each gender/age category. The dataset is available for download from the course web page:

http://www.csc.calpoly.edu/~dekhtyar

You will have to perform the same three tasks as in Lab 2: build an Oracle database and insert data into it, perform some changes in the structure of the database, and produce some formatted output.

Note: One new aspect of working with the MARATHON dataset is the fact that it contains information about times (time a participant ran a race, and participant's pace), which come in in formats that are different from Oracle SQL's default format for the DATE type. While we have not spent much time on this in class, please, consult the handout on dealing with DATE datatype to correctly process the information. I strongly recommend starting with some simple examples (create a table with a DATA attribute, try inserting different DATE values in it).

You have to complete the following list of tasks:

1. Create a relational database to store the MARATHON dataset date. As in Lab 2 assignment, the following rules must be followed:
   (a) The tables of the database must match the files of the dataset one for one.
   (b) You are allowed to choose any (hopefully meaningful and non-offensive) names for all relational tables and columns in them.
   (c) You must properly detect and declare all constraints, including primary key, candidate key (SQL's UNIQUE), and referential integrity/foreign key constraints.
2. Write and test an SQL script for creation of the MARATHON database (`MARATHON-setup.sql`).
3. Write and test an SQL script for deleting all tables from the database (`MARATHON-cleanup.sql`).
4. Prepare and test the SQL script for populating each database with the data avilable from the .csv file. Note, you may need to include some formatting instruction for the dates, which means that the scripts you used for Lab 2 might not work for this assignment. (`MARATHON-marathon.sql`.
5. Write and test the script for checking the contents of the database. (`MARATHON-test.sql`).
6. Alter the shape of the main table of the MARATHON dataset as follows. The `Group` column (note, this is not a proper column name, because `GROUP` is an SQL keyword), is basically a representation of a pair of numbers - the lower bound for the age in this age group, and the upper bound. The goal of this assignment is to replace this column in the database with a pair of integer columns, one for the lower and one for the upper bound. You need to
   - add the necessary columns to the database schema
   - instantiate the values in the columns (using some string operations and datatype conversion functions).
   - get rid of the old age group column (but only after the other columns are correct.

- output the contents of the new table.

Write the SQL script with these commands (`MARATHON-modify.sql`).

7. This part needs to be performed on the original data, not the modified version. Consider the following SQL query

```
SELECT <Group>, <Sex>, <GroupPlace>, <FirstName>, <LastName>, <Town>, <State>, <Place>
FROM <MarathonTable>
WHERE <GroupPlace> <= 5
ORDER BY <Group>, <Sex>, <GroupPlace>
```

   This query returns information about the first five finishers for each age/gender group and organizes this information by groups.

   Write an SQL script which outputs this query formatted as follows:

   - Column separator is " || " (note the spaces);
   - The headers for the columns are set as follows. The `group` column has the header "Age Category"; the `sex` column has the header "Gender"; the `GroupPlace` column has the header "Place in group"; first and last name columns have headers "First Name" and "Last Name".
   - All column headers are fully visible.
   - Each row is printed in one line
   - Each page of the report contains information about full age/gender groups (i.e., no page break comes between two rows with the same `<Group>` and `<Sex>` values). (note, this can be done in a number of ways).
   - The age range is displayed only once for each age/gender group.
   - There is one empty line between information about each group.

   Name the file `MARATHON-formatted.sql`.

Submission Instructions

You must submit all your files in a single archive. Accepted formats are `gzipped tar` (`.tar.gz`) or `zip` (`.zip`). The file you are submitting must be named `lab2-ilastname.ext`, where *i* stands for the initial of your first name, and *lastname* is your last name. E.g., if I were submitting this file, the name would be `lab2-adekhtyar.zip` or `lab2-adekhtyar.tar.gz`.

Your archive shall contain a single directory, `MARATHON`, inside which all files (see filenames above) are to be stored.