COEN 280 - Database Systems
Spring 2016
Homework Assignment 2
Due: Friday, Apr 29
@11:59pm

Part 1: Map the EER diagram into Oracle RDBMS model (20 points)
Convert your EER conceptual schema into tables and then implement these tables in the Oracle database. You can change your EER design freely during you conversion since your schema might not optimal. You will get full credit for part 1 and part 2 if your query is working properly.

Note: You are required to populate your database with the given data and test data with the queries in Part2. Use the excel data files for this. The excel file has tabs in the bottom for each data section. Also, go through the queries in part 2 to make reasonable assumptions regarding the attributes unavailable in the files provided and fill them out.

IMPORTANT Notes:

▪ **The following procedure must be followed to access the Oracle database server:**

//$ is the system prompt
$ sqlplus //Use sqlplus to issue sql statements

▪ **Reduction Guidelines for Oracle RDBMS:**

- Do not use triggers.
- Use reference for foreign keys and specify what action should be taken in case of update and/or deletion of the referenced tuple/row (i.e., cascade, reject, or setdefault/null).

▪ **Reference**:

Refer to Oracle SQL reference manual for information on how to create tables, indexes, insert data, etc. (http://docs.oracle.com/cd/E11882_01/server.112/e41084.pdf).

Part 2: Queries on the database (75 points)
Write the following queries in Oracle SQL and run them on your database developed as mentioned in Part 2 of this assignment. Depend on the data, your query might not return any data but it does not means your query is wrong.

1) Get the businesses in San Jose, CA that are still "in business". Results should be sorted by review counts in descending order. Return for the top 7 businesses, the business id, name, review count, and average rating of the business.
2) Get the review counts for businesses in each business category. Results should be sorted by the review counts in descending order.
3) Find the average rating across all reviews written by a particular user.
4) Get the businesses in San Jose, CA that have been reviewed by more than 5 `elite' users. Users who have written more than 10 reviews are called `elite' users. Results should be ordered by the `elite' user count in descending order. Return for the top 5 businesses, the business id, business name, business review count, average rating, and the count of the `elite' users for the particular business.
5) Get the businesses in San Jose, CA that have the highest percentage of reviews with rating of 5, and have been reviewed at least 5 times. Results should be ordered by the percentage in descending order. Return for the top 5 businesses, the business id, business name, review count, and percentage of reviews with rating of 5.
6) Find top two businesses, which have highest average ratings in each business category. For each business, list business id, business name and business category name.
7) Find the oldest user who never reviewed any businesses.
8) Find every business in CA that has the word "Coffee" in its name.
9) List all 5-star businesses that have been reviewed by any users between the ages of 15 and 25. A 5-star business is a business which has an average review rating of 5.
10) List users that have been to at least 3 distinct states. Order by number of states traveled to (decreasing), break ties with user id (increasing). For each user, list his/her user id, name and list of states traveled to.

Submission Guidelines

1. Your submission of part 1 and part 2 should include one createdb.sql file, one dropdb.sql file, ten .sql files for queries described in part 2 (named q1.sql to q10.sql), and one readme.txt file.

2. **createdb.sql** file should create required types, tables, indexes if required, generate primary keys, constraints, ... , and populate all the provided data based on the skeleton data provided. There is **60 points penalty** if this file is missing since it is not possible for us to check your queries without any data.

3. The **dropdb.sql** file should drop all types and tables that are created by createdb.sql. There is **10 points penalty** if this file is missing from your submission or if it does not drop all of your database objects.

4. **q1.sql ~ q10.sql** query files should contain SQL statements for queries Q1 to Q10 described in part 2 respectively. If you need to write two or more SQLs for ONE step, then they should be written after each other in ONE file.

5. Make sure to properly test created.sql, dropdb.sql and the query files (q1.sql…q10.sql) before submission. There will be **penalty** for resubmission if one the mentioned files do not execute properly.

6. The **readme.txt** file must have your name, the name of the database, tables that your createdb.sql file generates and the execution result of query files (q1.sql…q10.sql). There is **10 points penalty** if this file or some of the required information is missing from your submission.

7. You must make a .zip file to include all of your files in one file (<your_name>_hw2.zip:
   Your zip file should contain **createdb.sql dropdb.sql readme.txt q1.sql q2.sql q3.sql q4.sql q5.sql q6.sql q7.sql q8.sql q9.sql q10.sql** files.

8. You need to submit the 1st and 2nd part of your assignment to Camino
9. Start working on your assignment early.