

# COEN 280 - Database Systems

Fall 2015

## Homework Assignment 2

**Due: Friday, Oct 16**

**@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 your conversion since your schema might not be 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 Part 2. 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](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 mean your query is wrong.

1. Count the number of businesses in California ("CA").
2. Find every business in California that has the word "Coffee" (case-sensitive) in its name but is not classified as a coffee place (i.e. has no word "Coffee" in any of its categories). List the business id, bid, and name in ascending bid order.
3. For each state, give the most popular bar in that state. "Popular" means that it has the highest count of reviews, among all establishments that have one category as "Bars", in said state. Sort by state name (increasing). In the remote case of a tie in first place, list all bars, in ascending bid order. More specifically, for each bar, print its bid (i.e. business id), name, number of reviews and state.
4. List the top 10 businesses in San Jose, California for breakfast or brunch. "Top" means that it has the highest average number of stars among all businesses that have one category as "Breakfast & Brunch". Sort by their scores and break ties with number of reviews received (decreasing), then bid (increasing). For each business, print its bid (business id), name, average number of stars, and number of reviews.
5. List users that have been to more than 5 distinct states. Order by number of states traveled to (decreasing), break ties with uid (user id, increasing). For each user, list his/her uid, name and number of states traveled to.
6. List the top 5 burger restaurants (i.e. have the highest average number of stars) in San Jose, California. "Top" means that it has the highest score among all businesses in San Jose, California that have one category as "Burgers". Sort by score (highest, first) and break ties with number of reviews by travelers (decreasing), then bid (increasing). For each restaurant, print its bid, name, score, and number of traveler reviews received.
7. Find the user with the highest sum of the 'useful' votes that his/her reviews have attracted. If there is a tie in first place, list all such users, ordered by uid (i.e. user id). For each user, print his/her uid (i.e. userid), name, and usefulness count.
8. List California businesses that have more than 10 reviews and all of them are "5 stars". Order by the number of reviews received (decreasing), and bid (i.e. business id, increasing). For each business, print bid, name and number of reviews.
9. Find the businesses that got more than 50 5-star reviews from users that have only voted once. For each business, print its bid (i.e. business id), name and the number of reviews from such users. Order by the number of reviews received (decreasing), then business name (increasing) then bid (increasing).
10. Find the businesses whose average rating was raised by more than 1 stars from May 2011 to June 2011. Order your results by the magnitude of the jump (largest, first), then business names (increasing) and id (increasing). Print the first 10 if there are more than 10. For each business, print its bid (i.e. business id), name and jump magnitude.

## 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, ... , and populate sufficient data based on the skeleton data provided. “Sufficient data” means enough data such that your queries return something, but not everything. 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 Q11 described in part 3 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. The **readme.txt** file must have your name, the name of the database and tables that your **createdb.sql** file generates. There is 25 points penalty if this file or some of the required information is missing from your submission.
6. 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.
7. You need to submit the 1st and 2nd part of your assignment to Camino
8. Start working on your assignment early.