CSCE 315 Team Project 1 - Application
Spring 2017

For your database application, you are to put together an application that allows people to explore the Yelp review data.

Your code should make use of the database system provided by another team, not your own team's database system. As a reminder, teams should not give source code to another team, only the header files and .dll/.lib files.

Data source:
The data should be taken from the Yelp Dataset Challenge data. You can obtain this data from: https://www.yelp.com/dataset_challenge

Note that the data set is part of a larger contest focused on challenging students to see what types of insight they can generate into the data. You are not required to work in that direction; we are just using this as a data source. IF you wish to, you may of course do more with the dataset than is required here.

When you follow the links to download data, you will find two data files. You should obtain the 1.8GB data file (the main one), and you do not need to get the "Photos Auxiliary File". That auxiliary file contains images that correspond with the main dataset. You may use it if you wish, but it is not required.

When you download the data, you will need to uncompress it (it is a tar file). When uncompressed, you will find that the data is in a set of five different .json files. The format for these files is given on the Yelp Challenge website, but follows the JSON data format (which is a very common format for exchanging data from web services).

It is unlikely that the other team whose database you are using has made their database efficient enough to handle the gigabytes of data in the Yelp dataset. As a result, you should feel free to limit the data you actually enter to a subset of the overall dataset. As an example, you could use only a fraction of the user and business data, and then use only review data that matches both.

Operation:
Your program should read the data from these files and put it into a database system. You should create a program that will allow a user to explore that data. For example, you should allow them to get information about a user, or about a business, or about individual reviews of a business.

You will be graded in part by how "fully featured" you make your system. That is, the more information you can provide, the better. You will be graded based on:

- The types of queries you allow
- The thoroughness of the results that you allow the user of your application to see
- How well you make use of the database functionality that is provided
- How well you present the results (it should be presented in a clear and orderly way)

While the specific queries and operation are up to you, you should, **at minimum**, include queries/reports that let you:

- Display information about a user. Note that users have only their first name listed.

- Display information about a business.
- Combine information from more than one table. An example would be to show the reviews for a user, or for a business.

There are many ways to expand from there. To give you some ideas (these are not meant to be comprehensive, just a starting point), you could include information such as:

- Summary information about reviews/users for a business
- Summary information about reviews (such as average star rating) and compliments for a user
- Information about how ratings and customers correlate
- Information about ranges of ratings/reports

Four person teams will be expected to have more options for reports than 3-person teams.

Other Libraries:

You may wish to use external libraries to assist with parts of this. For example, you may wish to use a library to read JSON data, or to provide a user interface. This is allowed, but you need to be very clear about what you use (i.e. clearly state in all documentation) and ensure you are following license rules.

Turnin:

When you turn in your application, you should turn in a Readme or other document file that explains:

- How to use the system. What needs to be done to run queries, etc. If you provide functionality but don't say that you did so, we won't know and you won't get any credit for doing it!
- A description of how you managed to make use of the database functionality that was provided to you. Mention which database functionality you did and did not use in your system
- Any notes about parts of the database system that you tried to use but were unable to, and what this prevented you from doing.
- Any external libraries used.

Note that although the documentation is a relatively low part of the overall project grade, it is important to provide enough information that your code can be fully evaluated.

You will have a separate team report; this will be a chance to "evaluate" the database given to you. You do not need to provide that evaluation within this Readme file.

Grading:

Note that this application counts for 25% of the overall project grade. The grading for this portion will be as follows. Note that 75% of the grade will judge whether you have met the basic requirements, and the final 25% will be based on how many features are added into the system to provide more information.

20% Reading in JSON data and entering it into DB system
20% Use of DB system to retrieve data for display in application
10% Providing information about users
10% Providing information about businesses
15% Combining information from more than one table in the display
25% Overall level of features in the program (beyond the minimum)