



## Programming Assignment #1 CAS CS 460

### Restaurant Reviewing Application

Due: **Task 1: 10/6** (11:59pm) and **Task 2: 10/29** (11:59pm)

This programming assignment is for groups of two.

In this project you are asked to create a web application, which communicates with a local database that runs in your computer.

First, you will need to download the preconfigured Ubuntu machine available [here](#). This virtual machine has **PostgreSQL** already installed so you do not need to spend time on installation. The password for Ubuntu is 'cs460'. PostgreSQL is an open source relational database management system (DBMS) similar to MySQL that we covered in the first lab. The documentation for PostgreSQL can be found in this [link](#).

In order to run this image, you can download the open source virtual machine container Virtual Box from [here](#). A tutorial on how to open the image in virtual box on Windows is available [here](#). The procedure is the same for other Operating Systems as well. You can discuss in Piazza, Labs and TA OH any problems you might have with the installation.

The csv files, which can be found [here](#), contain the data you will need to insert into your database. By studying the data you will then have to create the DDL. Please choose data types carefully.

To create your database and insert the data you can either use the terminal or 'DBeaiver'. DBeaiver is already installed in your machines and provides a GUI environment for PostgreSQL. A demonstration of how DBeaiver works will take place in the third lab.

Your first task is to **clean the provided data** and make them **directly usable** for your application and your database system. **Data cleaning** is the process of detecting and correcting (or removing) corrupt or inaccurate records from a data collection. It generally refers to first identifying incomplete, inaccurate, or irrelevant parts of the data and then replacing, modifying, or deleting this part of the dataset.

For our restaurant reviewing application, the provided data contain four tables: "Business", "Reviews", "Users", and "Checkins". First, you will have to clean the last file ("Checkins") by eliminating different dates of the same day. For example "Friday\_1", "Friday\_2", ... "Friday\_31" should all be named as "Friday". In addition, you will clean any other columns from the remaining tables that are not going to be used in the context of our application.

Your web application should be able to connect to the database and retrieve data from it. You can use the tools you are most comfortable with. The image has

preinstalled the PostgreSQL system and a web server. The final deliverable is expected to:

- Have a web-based graphical interface that will facilitate the communication with the database and the presentation of the contents.
- Have the functionality to insert and delete data from the database. Specifically:
  - Insert/delete/update a restaurant
  - Insert/delete/update a user
  - Insert/delete/update a review
  - Insert/delete a checkin
- Enable the interested user to ask arbitrary SQL queries.
- Have a web-based interface that will execute the queries of part 2 and visualize the results.

### **Task 1: [15%] Modeling, Data Cleaning, and Loading**

Design the ER diagram of the modified database you created after cleaning the data as shown during the second lab. Your design should facilitate the insertion of new checkins from users. For this purpose, you might add new attributes (e.g. 'Date') or modify existing ones.

### **Task 2: [85%]**

For task 2, you will build a web application that allows users to execute queries. The interface of the application will have to run on a web browser. In this application you will need to have some fields that allow users to insert or delete data from the database you have already developed with PostgreSQL. In addition, you will need to have a field where users can execute SQL queries and retrieve data. For this purpose, you might also add buttons with empty fields that could execute the queries dynamically, based on user input. The way you organize the user interface is completely up to you. If fully functional you will receive full credit.

After implementing the system, you will have to translate the following queries from English to SQL and write the queries in a separate .txt file. Please follow the format shown precisely (note that Query 1 below is not the answer, but only an example). Your submission should be a ZIP file with both your source code and .txt file.

-----  
*Query1: select \* from Business;*  
*Query2: select...*  
...

-----

Queries:

1. Display all users that have made 50 or more reviews.
2. Display the name of the users that have made 3 reviews or less.



3. Display all inactive businesses.
4. Display the names of all pizza restaurants that have a rating of 4 stars or above.
5. Display the number of stores that had at least one check-in on Friday.
6. Display the text of all reviews made for the "Adobe Dam Family Golf Center".
7. Display the names of the restaurants that have taken at least one 1-star or 2-star review.
8. Display the average rating and total number of reviews of all KFC stores.
9. Display a list of the ids of the top 10 stores in terms of the number of reviews they have received.
10. Display the name of the user that has made the most reviews.