CptS -451 Introduction to Database Systems
Spring 2017

Project Description

In your semester long CptS 451course project you would develop a data search application for Yelp.com's business review data. The emphasis would be on the database infrastructure of the application.

Learner Objectives:

At the conclusion of this assignment you will gain experience in:

- ✓ Database modeling and design
- ✓ Populating the database with large datasets
- ✓ Querying large databases
- ✓ Optimizing query performance through indexes
- ✓ JSON parsing
- ✓ Database Application Development

Overview:

In 2013, Yelp.com has announced the "Yelp Dataset Challenge" and invited students to use this data in an innovative way and break ground in research. In your project you would query this dataset to extract useful information for local businesses and individual users.
The Yelp data is available in JSON format. The original Yelp dataset includes and **649K** tips by **687K** users for **86K** businesses from United States, Canada, UK, and Germany. (http://www.yelp.com/dataset_challenge/). In your project you will use a smaller dataset that your instructor created. This simplified dataset includes only **9,365** businesses, X users, and X tips written for those businesses.

You will be given sample code (C# and Python) to parse some of the Yelp JSON files (available on Blackboard). The Yelp JSON files that you will use in this project are available at the instructor's website at:
(http://www.eecs.wsu.edu/~arslanay/CptS451/project/yelp_dataset/YelpDataset_CptS451_2017.zip or http://www.eecs.wsu.edu/~arslanay/CptS451/links.html)
(Note: Please make sure to use the dataset available on the above link, not the one from the Yelp.com website)

See Appendix-B for an overview of the Yelp Academic Dataset.

Requirements:

You would develop a target application which runs queries on the Yelp data and extracts useful information. The primary users for this application will be potential customers seeking for businesses.
Using this application the users can gather information about:

- the businesses in a particular state, city, and/or zipcode,
- the businesses that belong to certain categories,
- detailed information about a business,
- ratings and popularity of businesses,
- etc.

You may design your application either as a standalone or a web-based application.

A detailed description of the application and example screenshots are available in Appendix-A. In evaluating your work instructor's primary focus will be primarily on how you design your database and how efficiently you

can search the database. However your GUI should provide the basic functionality for easy search of the business. Creativity is encouraged! Additional functionality will be considered for extra credit.

You will be given more detailed milestone descriptions when they are assigned.

Submission Instructions:

You will submit the deliverables for milestones on **Blackboard (learn.wsu.edu)**. For each milestone you will create a .zip files that contains all deliverables for that milestone, name the .zip files as *<yourname>_milestoneX.zip*, and submit it to the corresponding milestone dropbox on Blackboard. Specific submission details for each milestone will be provided under milestone descriptions.

Project Milestones:

I. Milestone-0: (no submission required)

Download and install PostgreSQL Database Server. You may download the latest version from the link https://www.postgresql.org/

II. Milestone-1: (Deadline Feb 9, 2017 11:59pm)

1) Parse JSON Data:

   Download the Yelp dataset from http://www.eecs.wsu.edu/~arslanay/CptS451/links.html. Look at each JSON file and understand what information the JSON objects provide. Pay attention to the data items in JSON objects that you will need for your application. The milestone-1 description will specify which data items you shouldn't parse in the *business* or *tip* or *checkin* or *user* JSON objects.

   Download the sample program from Blackboard (*Project*\ *Sample JSON Parsing Code*). You may either use the C# or the Python version of the sample parsing code. The sample code:
   - reads JSON objects form a file and extracts certain key and value pairs from JSON objects, and
   - writes the extracted data into a text file.

   Please note that the sample code includes examples of extracting simple key values only. In a JSON object the key value can be an array or another JSON object (for example: categories), therefore you need to recursively parse those objects until you extract all data stored in JSON objects. You will write the code for parsing business, user, tip, and checkin JSON objects.

2) Build a very simple database application:

   Download the "Milestone1DB.csv" file from the link http://www.eecs.wsu.edu/~arslanay/CptS451/links.html. Create a database on PostgreSQL with name "milestone1DB" and create a table named "business". You will import the CSV file into this table. Detailed instructions are available in Milestone-1 specification. (Note that the schema of this table should comply with the columns of the CSV file.)

   Write a simple application (either web or standalone) which connects to the milestone1DB database and runs simple queries on the business table. The goal of this exercise is to get you started in database programming early on. In Milestone3 you will develop a larger application with all required features.

   The instructor will provide a video which explains how to establish connectivity with PostgreSQL in C# using Npgsql. Instructor will provide the queries you need to run on your table (see Milestone 1 specification).

Milestone-1 Deliverables:

1. (40%) Source code for parsing all JSON data. Only submit your source code, **not the data files**.
2. (60%) Source code for your application. Only submit your source code, **not the data files**.

Create a zip archive *"<your-last-name>_milestone1.zip*" that includes your source code for JSON parsing and your sample application. Upload your milestone-1 submission on Blackboard until the deadline.

III. Milestone-2: (Deadline TBA)

1) Design a database schema that models the database for the described application scenario in Appendix-A and provide the ER diagram for your database design. Your database schema doesn't necessarily need to include all the data items provided in the JSON files. Your schema should be precise but yet complete. It should be designed in such a way that all queries/data retrievals on/from the database run efficiently and effectively. In Milestone3 you may revise your ER model.

   Translate your ER model into relations and produce DDL SQL statements for creating the corresponding tables in a relational DBMS. Note the constraints, including key constraints, referential integrity constraints, not NULL constraints, etc. needed for the relational schema to capture and enforce the semantics of your ER design.

   Populate your database with the Yelp data. Generate INSERT statements for your tables and run those to insert data into your DB. You will also write and additional scripts to update the information stored in your database.

   Write triggers and assertions to ensure the validity and consistency of the information stored in your database. Details will be available in Milestone2 specification.

Milestone-2 Deliverables:
(Weights of the deliverables are TBA)

1. The E-R diagram for your database design. To create your ER diagram, you are free to use whatever tool you are most comfortable with – you can use an ER modeling tool that you get from the web, your favorite drawing tool (e.g., Visio, Word, PowerPoint). **Should be submitted in .pdf format.** Name this file *"<your-last-name>_ER_v1.pdf*"
2. SQL script file containing all SQL statements (i.e., CEATE TABLE statements, UPDATE statements, and TRIGGERS) . Name this file *"<your-last-name>_SQL.sql*"
3. SQL script files containing samples of SQL INSERT statements for populating the database (20 samples from each table). You don't need to submit the complete files. *"<your-last-name>_Inserts.sql*"

Create a zip archive *"<your-last-name>_milestone2.zip*" that includes your ER diagram and SQL script files. Upload your milestone-2 submission on Blackboard until the deadline.
You will demonstrate your Milestone1 and Milestone-2 (together) to the TA.

IV. Milestone-3: (Deadline: TBA)

In this milestone you would:

1) Implement an application (either web or standalone) where the users can search for information and statistic about local businesses. A detailed description of the application requirements is provided in Appendix-A.
2) Propose and implement an additional search feature. Creativity is encouraged!

Milestone-3 Deliverables:
(Weights of the deliverables are TBA.)

1. The source code of your application. **Please only upload your source code, not your DB files.**
2. SQL script file that contains:
    a. your main SELECT query for searching businesses and reviews,
    b. CREATE INDEX statements for the indexes you used.

Create a zip archive *"<your-last-name>_milestone3.zip*" that includes your source code and the SQL script file. Upload your milestone-3 submission on Blackboard until the deadline.

You will demonstrate your final project to the instructor. The demonstration schedule will be announced in mid-April.

References:

1. Yelp Dataset Challenge, http://www.yelp.com/dataset_challenge/
2. Samples for users of the Yelp Academic Database, https://github.com/Yelp/dataset-examples
3. Yelp Challenge, University of Washington Student Paper 1 http://courses.cs.washington.edu/courses/cse544/13sp/final-projects/p08-fants.pdf
4. Yelp Challenge, University of Washington Student Paper 2, http://courses.cs.washington.edu/courses/cse544/13sp/final-projects/p10-michelmj.pdf

Appendix-A

Application Specification

The primary users for this application will be potential customers seeking for businesses. Using this application the users can gather information about:

- the businesses in a particular state, city, and/or zipcode,
- the businesses that belong to certain categories,
- detailed information about businesses,
- ratings and popularity of businesses,
- the businesses that their friends visited and reviewed, etc.

You may design your application either as a standalone or a web-based application. Below you will find screenshots to help you visualize the required functionality.

The application will have 2 main windows:

A. **User Information:**

Use Case:

1. The user enters his/her own user id and retrieves his/her user profile information including, name, average stars, date he/she joined yelp, number of fans, average stars, and count of votes. The list of the user's friends and the latest tip each friend posted are displayed. User may rate one of his/her friends or remove a friend. (See Figure-1)

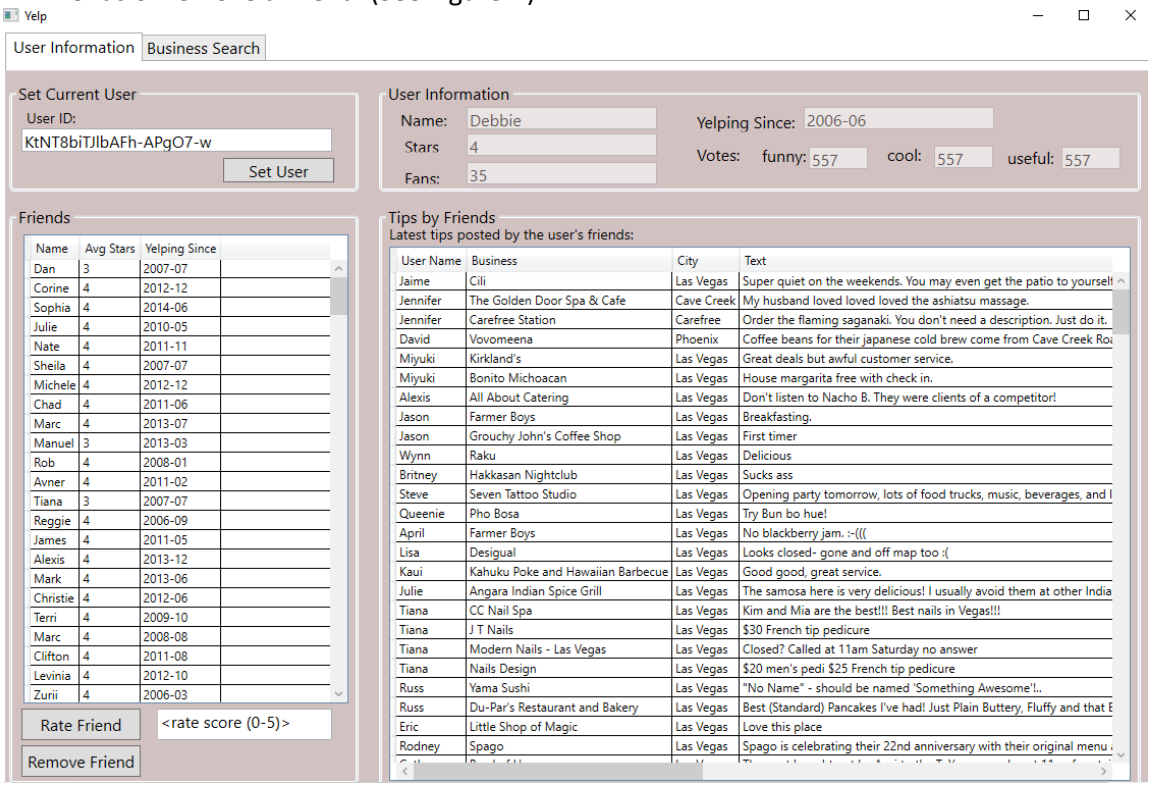

Figure 1 – User Information Window

B. **Business Search:**

Users can search for businesses which are within a certain state, city, and zip and which belong to the selected categories. The application allows users to display some statistics about the businesses in the search results and to retrieve various information about a selected business (See Figure-2)

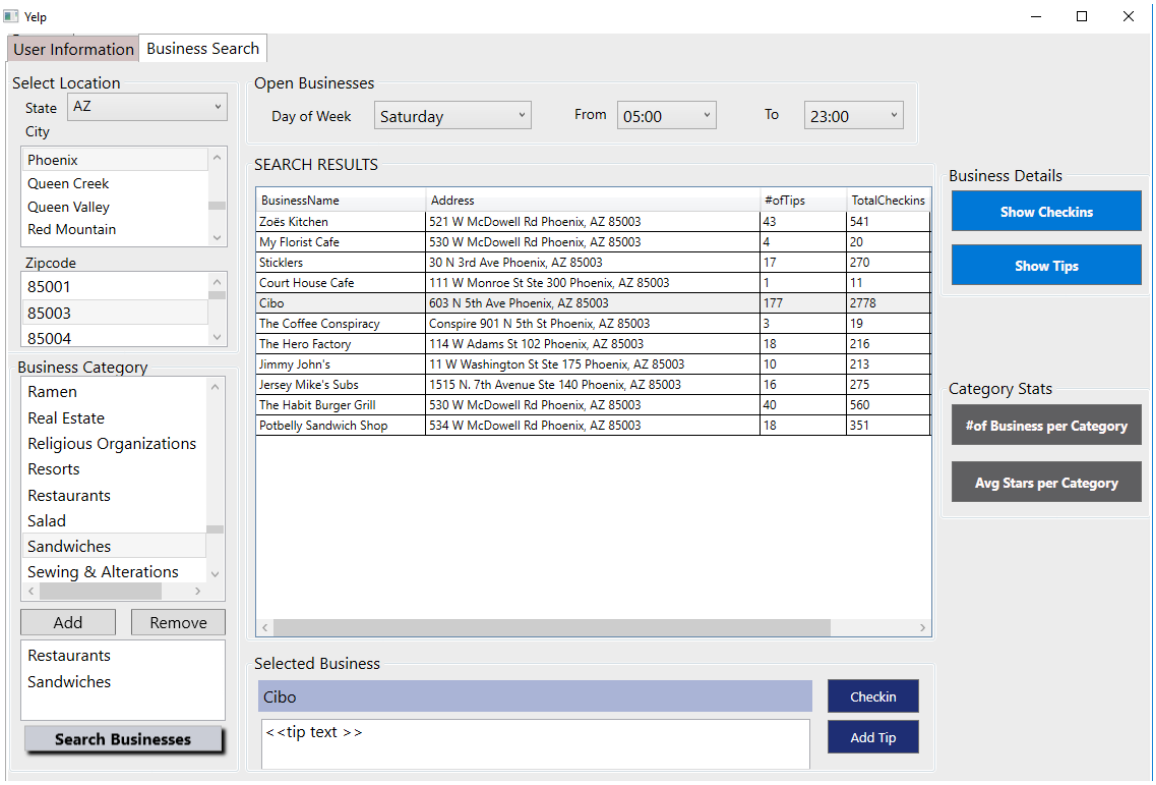

Figure 2 - Search Businesses

Use Cases:

1. User selects a state, city, and/or zipcode and specifies one or more categories. When search button is pressed the businesses in that state/city/zipcode which belong **to ALL specified categories** will be returned (i.e., AND condition). The following information should be provided for each business:
   - Business name
   - Address
   - # of tips provided for the business
   - Total number of check-ins

   (Note: You should query the tips table to calculate the number of tips and number of check-ins for each business and update those attributes in the business table.)

2. The user might refine the results according to the times the business is open/closed on a certain day of the week. User specifies a day-of-week and a start and end time. All businesses that are closed during the given time-slot are excluded from the results. Please note that if a business is closed during part of the slot (but open during the rest), it should be excluded from the result. <u>All filtering on the results need to be implemented in SQL queries. No points will be given if you filter results in the GUI when you display them in the list-view</u>.
   If no time-slot is given, no filtering should be done.

3. The user may select a certain business in the search results (by simply clicking on a business) and display various information about the business, including:
   a. *Show Check-ins:* All check-ins for the business which are grouped by the day-of-the-week and the time-of-the-day. For simplicity, you are asked to aggregate the check-in information into morning (6am-12noon), afternoon (12noon-5pm), evening (5pm-11pm), night (11pm-6am) intervals. (Assume start time of each interval is inclusive and end time is exclusive.) Your application should visualize the number of check-ins for each day-of-the-week and time-of-the-day as a chart. Please see Figure-3 for an example.

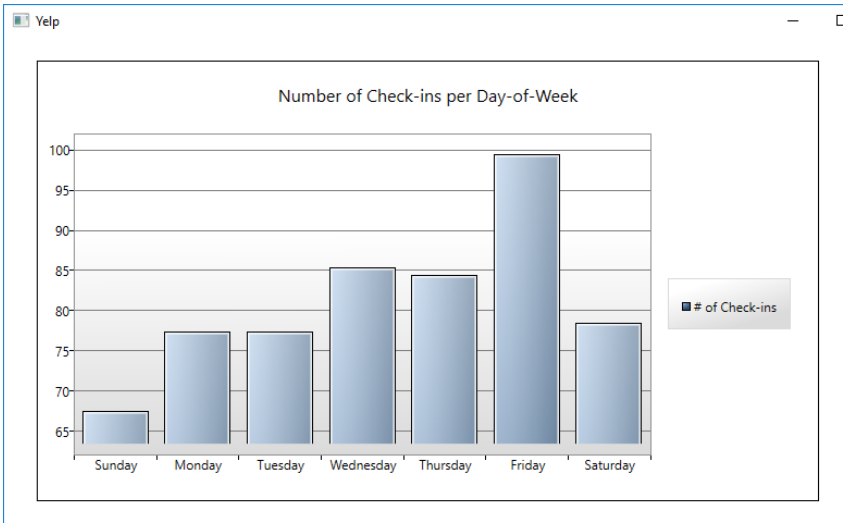

Figure 3 – Number of check-ins for the selected business

b. *Show Tips:* The tips provided for the selected business. For each tip, you should display the name of the user who provided the tip, the date tip is provided, the number likes for the tip and the tip text. You should display this information as a list (or table).

c. *#of Business per Category:* Number of business per category for the businesses that appear in the search results. You should display this information as a chart. (see Figure-4)

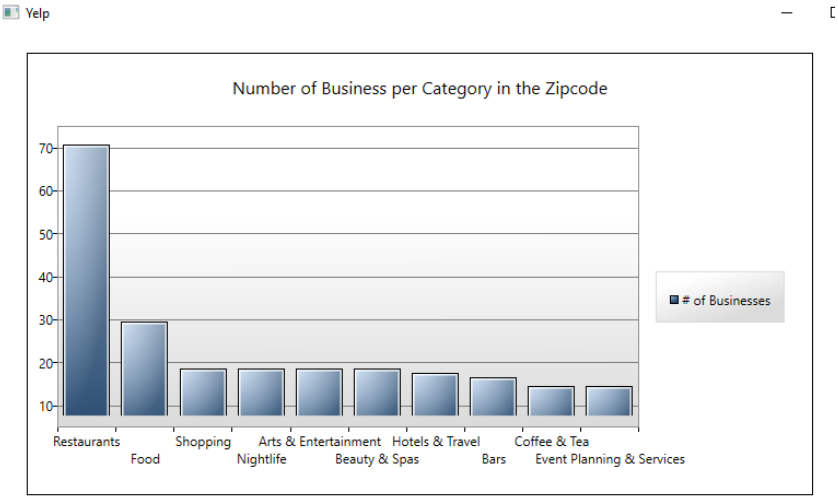

Figure 4 – Number of businesses per category

d. *Avg Stars per Category:* Average number of star ratings per category. Your application should display this information as a graph.

Please note that all data displayed on the GUI should be kept in the database and should be retrieved from it when needed. You are not allowed to create internal data structures to store data.
You may design your application either as a standalone or a web-based application.

Appendix-B

Yelp's Academic Dataset

Yelp has made available a dataset which contains user reviews **86K** businesses from United States, Canada, UK, and Germany. The purpose was to provide a real-world data set to promote research in various areas of research. The dataset includes 6 types of data objects: *business*, *review*, *user*, *tip*, *check-in, and photos*. Every object contains a 'type' field, which tells whether it is a *business*, a *user*, or a *review*. *Business* objects contain basic information about local businesses. *Review* objects contain the details of the reviews by users for the businesses. *Review*'s user_id associates the reviews with the *user* objects. Similarly, *review*'s business_id associates each review with the *businesses*.
You can download the compressed Yelp data files from http://www.yelp.com/dataset_challenge/. Each file is composed of one json-object per line.
The fields of objects are given below:

Business

```
{
     'type': 'business',
     'business_id': (encrypted business id),
     'name': (business name),
     'neighborhoods': [(hood names)],
     'full_address': (localized address),
     'city': (city),
     'state': (state),
     'latitude': latitude,
     'longitude': longitude,
     'stars': (star rating, rounded to half-stars),
     'review_count': review count,
     'categories': [(localized category names)]
     'open': True / False (corresponds to closed, not business hours), %(You don't need "open" information for this project)

     'hours': {
        (day_of_week): {
           'open': (HH:MM),
           'close': (HH:MM)
        },
        ...
     },
     'attributes': { %(You don't need "attributes" information for this project)

        (attribute_name): (attribute_value),
        ...
     },
}
```

Review **(You won't use review data in your project)**

```
{
'type': 'review',
    'business_id': (encrypted business id),
    'user_id': (encrypted user id),
    'stars': (star rating, rounded to half-stars),
    'text': (review text),
    'date': (date, formatted like '2012-03-14'),
    'votes': {(vote type): (count)},
}
```

User

```
{
    'type': 'user',
    'user_id': (encrypted user id),
    'name': (first name),
    'review_count': (review count),
    'average_stars': (floating point average, like 4.31),
    'votes': {(vote type): (count)},
    'friends': [(friend user_ids)],
    'elite': [(years_elite)], %(You don't need "elite" information for this project)
    'yelping_since': (date, formatted like '2012-03'),
    'compliments': { %(You don't need "compliments" information for this project)
        (compliment_type): (num_compliments_of_this_type),
        ...
```

```
    },
    'fans': (num_fans),
}
```

Check-in

```
{
    'type': 'checkin',
    'business_id': (encrypted business id),
    'checkin_info': {
        '0-0': (number of checkins from 00:00 to 01:00 on all Sundays),
        '1-0': (number of checkins from 01:00 to 02:00 on all Sundays),
        ...
        '14-4': (number of checkins from 14:00 to 15:00 on all Thursdays),
        ...
        '23-6': (number of checkins from 23:00 to 00:00 on all Saturdays)
    }, # if there was no checkin for a hour-day block it will not be in the dict
}
```

Tip

```
{
    'type': 'tip',
    'text': (tip text),
    'business_id': (encrypted business id),
    'user_id': (encrypted user id),
    'date': (date, formatted like '2012-03-14'),
    'likes': (count),
}
```

photos (from the photos auxiliary file) **(You won't use photos data in your project)**

```
[
    {
        "photo_id": (encrypted photo id),
        "business_id" : (encrypted business id),
        "caption" : (the photo caption, if any),
        "label" : (the category the photo belongs to, if any)
    },
    {...}
]
```

Usage of this dataset is governed by the Academic Dataset Terms of Use.