CECS 521 - Database Architecture
Fall 2016
Homework Assignment 3
Due: Friday, Nov 4
@11:59pm

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

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 this 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 42,153 businesses, 252,898 users, and 1,125,458 reviews from Phoenix (AZ), Las Vegas (NV), Madison (WI) in United States and Waterloo (ON) and Edinburgh (ON) in Canada. (http://www.yelp.com/dataset_challenge/). In your project you will use a smaller and simplified dataset. This simplified dataset includes only **20,544** businesses, the reviews that are written for those businesses only, and the users that wrote those reviews.

The Yelp JSON files that you will use in this project are available on beachboard.
(Note: Please make sure to use the dataset available on beachboard, not the one from the Yelp.com website)

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

Overview & 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 and users that match their search criteria. Your application will have a user interface that provides the user the available business categories (main and sub-categories) and the checkin attribute along with business review and yelp user information associated with each business category. Using this application the user will search for the businesses from various business categories that have the properties (attributes) the user is looking for.

The user can filter the search results by checkin (from/to and Day/Hours), reviews (To/From, No. of stars/votes) and users information. The application should also allow the user to view the reviews provided for each business.

You will be designing your application a standalone Java application.

Example screenshots of a possible application are available in Appendix-B. 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 and pull out the information. However your GUI should provide the basic functionality for easy browsing of the business categories and attributes (as illustrated in Appendix-B). Creativity is encouraged!

Project Details:

I. Part 1

- Download the Yelp dataset from beachboard. 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 (For example, categories, attributes,…etc.)
- You may have to modify your database design from Homework 2 to model the database for the described application scenario on page-1. 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.
- 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.

II. Part 2

Implement the application for searching local businesses as explained in section "Overview & Requirements". In this milestone you would:

- Write the SQL queries to search your database.
- Establish connectivity with the DBMS.
- Embed/execute queries in/from the code. Retrieve query results and parse the returned results to generate the output that will be displayed on the GUI.
- Implement a GUI where the user can,
  - Search for either a business or users that match the criteria given. Please note that at any given time, the application can search for only a business or users, but not both.
  - Browse through main and sub-categories for the businesses; select the business categories that user wants to search for; (*note: The list of the main categories is given in Appendix-C. All other categories that appear in the business objects are sub-categories. Such a distinction is made for easier browsing of the business categories.*)
  - Search for the businesses that belong to the main and sub-categories that user specifies along with checkin/review information (Fig 1).
  - Search for users with attributes shown in Fig. 2 (The application should be able to search for the users that have either all the specified attributes (AND condition) or that have any of the attributes specified (OR condition))
  - Select a certain business in the search results and list all the reviews for that business. (*note: The review list should also include the names of the users who provided those reviews*)
  - Select a user in the search results and list all the reviews for that user.

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.

Required .sql files:

You are required to create two .sql files:

1. createdb.sql: This file should create all required tables. In addition, it should include constraints, indexes, and any other DDL statements you might need for your application.
2. dropdb.sql: This file should drop all tables and the other objects once created by your createdb.sql file.

Required Java Programs:

You are required to implement two Java programs:

1. populate.java: This program should get the names of the input files as command line parameters and populate them into your database. It should be executed as:
   "> java populate yelp_business.json yelp_review.json yelp_checkin.json yelp_user.json".
   Note that every time you run this program, it should remove the previous data in your tables; otherwise the tables will have redundant data.

2. hw3.java: This program should provide a GUI, similar to figure 1, to query your database. The GUI should include:
    a. List of main business categories.
    b. For the Business section, list of sub-categories associated with the selected main category(ies).
    c. Checkin and Review sections along with selectable attributes for business selection
    d. For Users section, list of selectable attributes and dropdowns
    e. List of results (Users or Businesses)
    f. List of the reviews provided for a specific business/User.

Grading guideline:

Points	
5	Creating/Dropping database tables
10	Populating database
10	JSON parsing
20	GUI containing all of requirements mentioned for user interfaces.
10	Listing of Category/Subcategory
25	Filtering Search/All/Any attributes, Checkin, Review, Users
10	Listing of reviews for a business/user
10	List of results

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

Appendix-A

Yelp's Academic Dataset

Yelp has made available a dataset which contains user reviews for 42,153 businessesin Phoenix (AZ), Las Vegas (NV), Madison (WI) in United States and Waterloo (ON) and Edinburgh (ON) in Canada. The purpose was to provide a real-world data set to promote research in various areas of research. The dataset includes 5 types of data objects: *business*, *review*, *user*, *tip*, and *check-in*. 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*.

The fields of objects are given below:

Business Objects
Business objects contain basic information about local businesses.

```
{
     'business_id': (encrypted business id),
     'full_address': (localized address),
     ‘hours’: (the days of the week when business is open; the opening and closing times on those days)
     'open': True / False (corresponds to closed, not business hours),
     ‘categories’: (categories associated with the business)
     'city': (city),
     'state': (state),
     'latitude': latitude,
     'longitude': longitude,
     'review_count': review count,
     'name': (business name),
     'neighborhoods': [(hood names)],
     'stars': (star rating, rounded to half-stars),
     ‘attributes’: (business properties),
     'type': 'business'
}
```

Review Objects
Review objects contain the review text, the star rating, and information on votes Yelp users have cast on the review. Use user_id to associate this review with others by the same user. Use business_id to associate this review with others of the same business.

```
{
     'votes': {
         'useful': (count of useful votes),
         'funny': (count of funny votes),
         'cool': (count of cool votes)
       }
     'user_id': (the identifier of the authoring user),
     'review_id': (the identifier of the reviewed business),
     'stars': (star rating, integer 1-5),
     'date': (date, formatted like '2011-04-19'),
     'text': (review text),
     'type': 'review',
     'business_id': (the identifier of the reviewed business)
}
```

User Objects
User objects contain aggregate information about a single user across all of Yelp (including businesses and reviews not in this dataset).

```
{
     ‘yelping_since’: (the date when user account was created)
     'votes': {
         'useful': (count of useful votes across all reviews),
         'funny': (count of funny votes across all reviews),
         'cool': (count of cool votes across all reviews)
```

```
    }
    'review_count': (review count),
    'name': (first name, last initial, like 'Matt J.'),
    'user_id': (unique user identifier),
    ‘friends’: (friends of the user),
    ‘fans’: (number fans of the user),
    'average_stars': (floating point average, like 4.31),
    'type': 'user',
    ‘compliments’: (comments from other users),
    ‘elite’: ()
}
```

Checkin

```
{
    '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 list
}
```

Tip

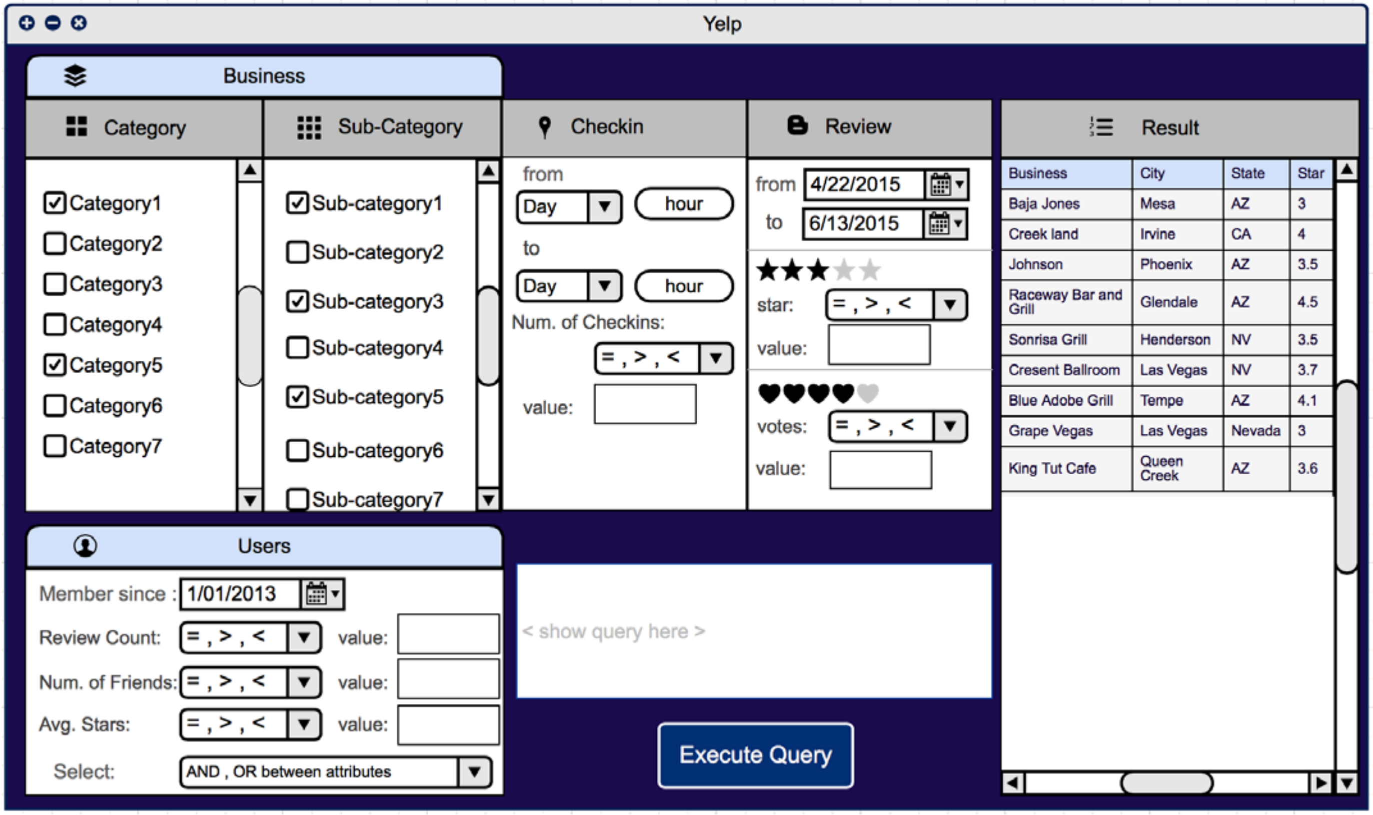
```
{
    'user_id': (encrypted user id),
    'text': (),
    'business_id': (encrypted user id),
    ‘likes’: (),
    ‘date’: (),
    ‘type': 'tip'
}
```

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

s

Appendix-B

Sample Application



Category: ☑ Category1 ☐ Category2 ☐ Category3 ☐ Category4 ☑ Category5 ☐ Category6 ☐ Category7

Sub-Category: ☑ Sub-category1 ☐ Sub-category2 ☑ Sub-category3 ☐ Sub-category4 ☑ Sub-category5 ☐ Sub-category6 ☐ Sub-category7

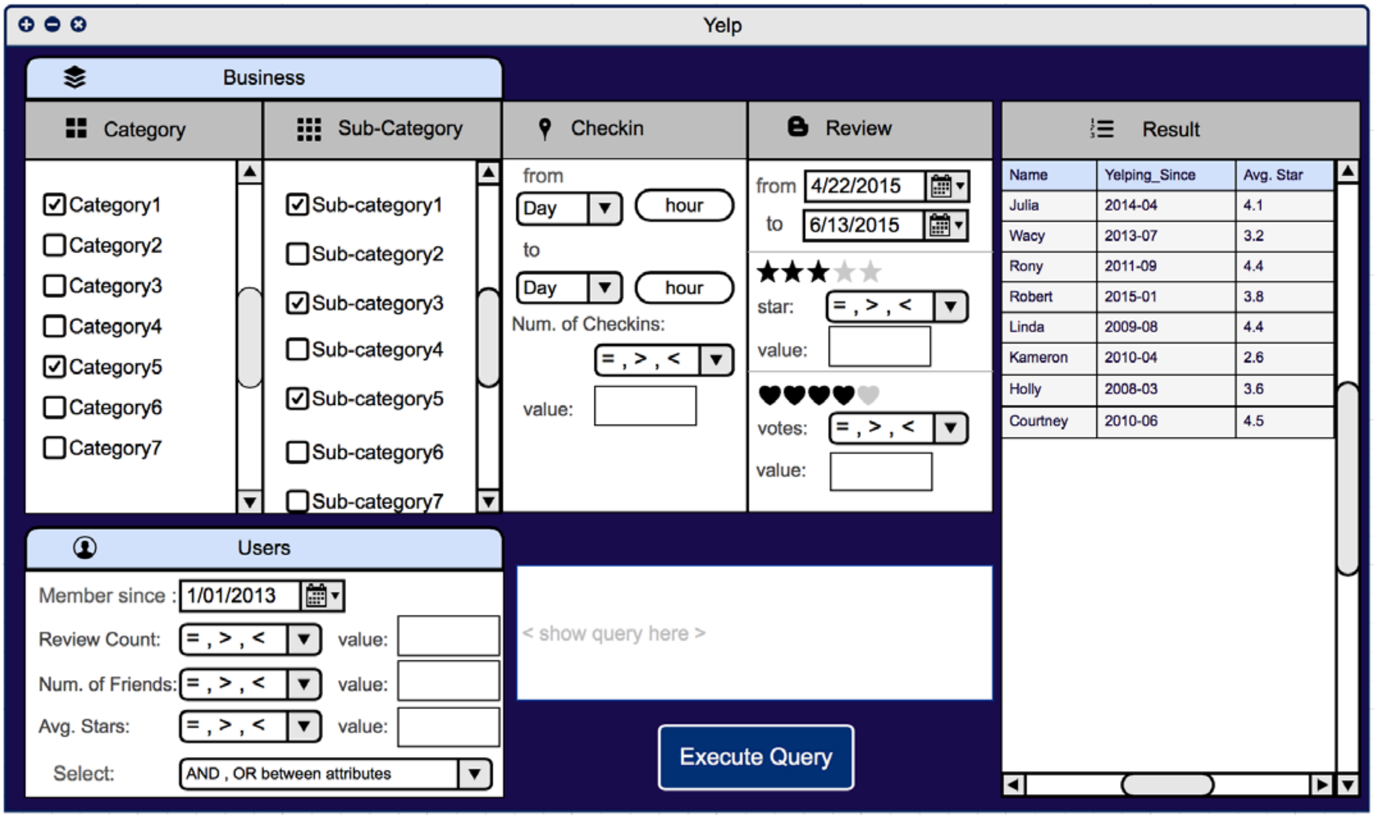
Checkin: from Day ▼ hour; to Day ▼ hour; Num. of Checkins: = , > , < ▼ value:

Review: from 4/22/2015 to 6/13/2015; ★★★☆☆ star: = , > , < ▼ value:; ♥♥♥♥♡ votes: = , > , < ▼ value:

Business	City	State	Star
Baja Jones	Mesa	AZ	3
Creek land	Irvine	CA	4
Johnson	Phoenix	AZ	3.5
Raceway Bar and Grill	Glendale	AZ	4.5
Sonrisa Grill	Henderson	NV	3.5
Cresent Ballroom	Las Vegas	NV	3.7
Blue Adobe Grill	Tempe	AZ	4.1
Grape Vegas	Las Vegas	Nevada	3
King Tut Cafe	Queen Creek	AZ	3.6

Users:
Member since : 1/01/2013
Review Count: = , > , < ▼ value:
Num. of Friends: = , > , < ▼ value:
Avg. Stars: = , > , < ▼ value:
Select: AND , OR between attributes ▼

Figure 1- Yelp Application Main UI (Business Search)



Category: ☑ Category1 ☐ Category2 ☐ Category3 ☐ Category4 ☑ Category5 ☐ Category6 ☐ Category7

Sub-Category: ☑ Sub-category1 ☐ Sub-category2 ☑ Sub-category3 ☐ Sub-category4 ☑ Sub-category5 ☐ Sub-category6 ☐ Sub-category7

Checkin: from Day ▼ hour; to Day ▼ hour; Num. of Checkins: = , > , < ▼ value:

Review: from 4/22/2015 to 6/13/2015; ★★★☆☆ star: = , > , < ▼ value:; ♥♥♥♥♡ votes: = , > , < ▼ value:

Name	Yelping_Since	Avg. Star
Julia	2014-04	4.1
Wacy	2013-07	3.2
Rony	2011-09	4.4
Robert	2015-01	3.8
Linda	2009-08	4.4
Kameron	2010-04	2.6
Holly	2008-03	3.6
Courtney	2010-06	4.5

Users:
Member since : 1/01/2013
Review Count: = , > , < ▼ value:
Num. of Friends: = , > , < ▼ value:
Avg. Stars: = , > , < ▼ value:
Select: AND , OR between attributes ▼

Figure 2- Yelp Application Main UI (User Search)

Appendix-C

Main Business Categories

1. Active Life
2. Arts & Entertainment
3. Automotive
4. Car Rental
5. Cafes
6. Beauty & Spas
7. Convenience Stores
8. Dentists
9. Doctors
10. Drugstores
11. Department Stores
12. Education
13. Event Planning & Services
14. Flowers & Gifts
15. Food
16. Health & Medical
17. Home Services
18. Home & Garden
19. Hospitals
20. Hotels & Travel
21. Hardware Stores
22. Grocery
23. Medical Centers
24. Nurseries & Gardening
25. Nightlife
26. Restaurants
27. Shopping
28. Transportation