Project 1
Yelp: Intro to SQL

Out: September 27th, 2018
Due: October 3th, 2018, 11:59 P.M.

1 Introduction

In this assignment, you will familiarize yourself with SQL queries and learn how to interact with a database through Java. You will compose five SQL queries on Yelp business data.

2 Overview of the Data

You will be using a small subset of Yelp’s Academic Dataset, which provides data and reviews of businesses around 30 U.S universities, including Brown University. The TAs have already parsed the data into a SQLite database, which can be found in /course/cs1270/pub/yelp/yelp.db.

For more information about Yelp’s Academic Dataset, see https://www.yelp.com/academic_dataset.

2.1 Database Schema

Below is the database schema for three tables (business, review and user) along with the datatype for each column (field). For more information about datatypes in SQLite, you can refer to http://www.sqlite.org/datatype3.html.

2.1.1 business

'id': a unique identifier for this business (VARCHAR),
'name': the full business name (VARCHAR),
'full_address': localized address (VARCHAR),
'city': city (VARCHAR),
'state': state (VARCHAR),
'latitude': latitude (REAL),
'longitude': longitude (REAL),
'stars': star rating, rounded to half-stars (REAL),
'review_count': review count (INT),
'open': is the business still open for business? (INT),
'photo_url': photo url (VARCHAR)
2.1.2 review

'business_id': the identifier of the reviewed business (VARCHAR),
'user_id': the identifier of the authoring user (VARCHAR),
'stars': star rating, integer 1-5 (INT),
text': review text (TEXT),
'useful_votes': count of useful votes (INT),
'funny_votes': count of funny votes (INT),
'cool_votes': count of cool votes (INT)

2.1.3 user

'id': unique user identifier (VARCHAR),
'name': first name, last initial, like 'Matt J.' (VARCHAR),
'review_count': review count (INT),
'useful_votes': count of useful votes across all reviews (INT),
'funny_votes': count of funny votes across all reviews (INT),
'cool_votes': count of cool votes across all reviews (INT)

3 Tools

3.1 SQLite

SQLite is installed on all Sunlab machines. It can be accessed from the command line using sqlite3. To load a pre-existing database (for example, the Yelp database), simply run

```
sqlite3 /course/cs1270/pub/yelp/yelp.db
```

For more information on using SQLite from the command line, see [http://www.sqlite.org/cli.html](http://www.sqlite.org/cli.html)

3.2 SQLite Manager

SQLite Manager is a Firefox add-on. It allows you to view the tables in the database and execute test queries. To get the SQLite Manager add-on, see [https://addons.mozilla.org/en-us/firefox/addon/sqlite-manager/](https://addons.mozilla.org/en-us/firefox/addon/sqlite-manager/)

To connect the database to the SQLite Manager:

1. Open Firefox and navigate to Tools → SQLite Manager.
2. Navigate to Database → Connect Database.
3. Browse to /course/cs1270/pub/yelp, change the file type selector from *.sqlite to All Files, and open yelp.db.
3.3 JDBC

We will be using SQLite database through a JDBC driver (Java Database Connectivity). The JDBC SQLite driver is included as a part of your stencil code.

There will not be an official help session on how to use JDBC, but TAs will be happy to answer questions on hours or via email. Students are highly encouraged to check out [http://web.archive.org/web/20100814175321/http://www.zentus.com/sqlitejdbc/](http://web.archive.org/web/20100814175321/http://www.zentus.com/sqlitejdbc/) which has a simple tutorial on working with JDBC and SQLite.

4 Working on the Project

4.1 Getting Started

First, add the following line to your ~/.bashrc

```bash
export JAVA_HOME='''/pro/java/linux/jdk1.8.0_40''
```

After adding it, run `source ~/.bashrc` (you should only need to run this once).

To get started with the Java stencil, copy `/course/cs1270/pub/yelp/stencil.tgz` into your course directory, and unpack it with `tar -xvzf stencil.tgz`.

4.2 Importing into Eclipse

1. Expand the stencil code inside your course directory. That should create a directory named “yelp”
2. Open Eclipse. From the top menu bar, navigate to File → New Java Project.
3. From there, uncheck “Use default location.”
4. Browse to the yelp directory inside your course directory. Click Finish.

5 The GUI Application

5.1 How to Run

To launch the GUI, either run App.java in Eclipse or run with the command `ant run` in the project directory.
5.2 How to Use

On the left pane, the application displays business information. Only seven businesses are shown. Note that the business names are clickable and once clicked, it will open your default browser to load an actual business page on Yelp.

On the right pane, the application displays the corresponding reviews for the selected business. Only seven reviews are shown. The “User Avg” button serves to provide the average rating for all reviews written by the user. The user names are clickable and once clicked, it will open your default browser to load an actual user page on Yelp.

In the bottom panel, the three buttons serve to provide different ranking algorithms for displaying seven businesses on the left pane.

5.3 Demo

Students are highly encouraged to check out the demo before starting the assignment. The demo can be found in `/course/cs1270/pub/yelp/demo/demo.jar` and it can be run with the command `java -jar demo.jar`. You can compare the results of the demo to your own once you finish writing each query to check that you are on the right track.

6 Your Assignment

Your task is to set up a connection to the database and write five queries to answer the following questions. All queries should be composed of a single SQL statement. All of your Java code should be written in DBStudentController.java, a stencil file provided for you.

For each query method, you will have to do two things. First, you will have to write an appropriate SQL query and execute it to return a ResultSet. **You must use PreparedStatement to execute the queries.** PreparedStatement is an efficient way to execute queries, especially for the ones that require inputs. More information can be found in [http://web.archive.org/web/20100814175321/http://www.zentus.com/sqlitejdbc/](http://web.archive.org/web/20100814175321/http://www.zentus.com/sqlitejdbc/). Second, you will need to extract the data from the ResultSet, create an appropriate data type to store them, and return it accordingly.

6.1 Queries

1. Get the businesses in Providence, RI that are still open. Results should be sorted by review counts in descending order. Return top 7 businesses.

   **Input:** N/A
   
   **Output:** Six columns - the business id, name, full address, review count, photo url, and stars of the business.
2. Get the reviews for a particular business, given the business ID. Results should be sorted by the review’s useful vote counts in descending order. Return top 7 reviews.

**Input:** Business ID

**Output:** Four columns - the user id, name of the user, stars of the review, and text of the review.

3. Find the average star rating across all reviews written by a particular user.

**Input:** User ID

**Output:** One column - the average star rating.

4. This problem consists of two parts. Only part (b) will show up on the GUI. Part (a) is optional, but completing part (a) will keep you on the right track, and can potentially earn you partial credit.

(a) First, write a query to find the ‘elite’ users, users who have written more than 10 reviews. Store your query as a variable named "partA". (e.g. String partA = "SELECT ...";)

(b) Now, get the businesses in Providence, RI that have been reviewed by more than 5 ‘elite’ users. Results should be ordered by the ‘elite’ user count in descending order. Return top 7 businesses.

**Input:** N/A

**Output:** Seven columns - the business id, business name, business full address, review count, photo url, stars, and the count of the ‘elite’ users for the particular business.

5. Get the businesses in Providence, RI that have the highest percentage of five star reviews, and have been reviewed at least 20 times. Results should be ordered by the percentage in descending order. Return top 7 businesses.

**Input:** N/A

**Output:** Seven columns - the business id, business name, business full address, review count, photo url, stars, and percentage of five star reviews

6.2 Hint

To return top n rows in SQLite, you can use the LIMIT clause (http://www.sqlite.org/syntaxdiagrams.html#select-stmt). For example, SELECT * FROM business LIMIT 10; will return the top 10 rows in the business table.
7 Handin

You can handin your project by running the following command from the directory containing all your files:

/course/cs1270/bin/cs127_handin yelp

8 QA

Please check QA below before you post the question on Piazza.

1. Why SQLite Manager can’t open yelp.db?
   Please make sure the file extension should be db, not db.sql. Everyone should have read capabilities to the yelp.db database. As a workaround, try making a copy of the database from the /course/cs1270/pub/yelp directory to your home directory. Then, you should have total access to that copy of the database.

2. What parameters the DriverManager.getConnection() method is supposed to take on?
   Try this: "jdbc:sqlite:/course/cs127/pub/yelp/yelp.db"

3. Are we allowed to create new tables for the yelp project?
   New tables should not be needed. Please construct your answer using the existing tables.

4. Couldn’t run demo.jar on our own computer?
   The demo has a path to the database in the course directory. You can replicate the same path to run it locally. In other words, the demo should work on department PC’s that you are working on or ssh’d into. The demo references the yelp database at ’/course/cs127/pub/yelp/yelp.db’ so unless you have that exact path in your local file system the demo will not work.

5. For Query 4, will the elite user be counted twice in the condition (num_elite_users > 5 for a business) if he/she visited the business twice?
   You can assume each user writes at most one review per business.

6. Can we use string concatenation to pass a parameter to a SQL query or do we have to use the JDBC command paired with the question mark*?
   Using the JDBC command with the question mark is the better way to do it. Heres a short resource on why parameterization is preferable:
   [https://cran.r-project.org/web/packages/RODBCext/vignettes/Parameterized_SQL_queries.html](https://cran.r-project.org/web/packages/RODBCext/vignettes/Parameterized_SQL_queries.html)
7. Is Yelp graded based solely on accuracy, or also on the elegance of the query?
We expect your queries to run in a reasonable amount of time (probably no more than one minute at the very most). If you get the same output as the demo, that’s an excellent indicator your queries are correct. We will check your queries in terms of parameterization and to make sure none of the results are hard-coded, so the more readable those portions are the better! We will not check for style otherwise.

8. Could we resubmit the project files if we have handed in once? Would that cause the problem of duplicate files?
You can submit as many times up to the deadline as you like. However, if you last submission is after the deadline, it will be counted as a late submission.

9. Could we do this project on our personal computer?
Yes, you can, but we highly recommend you can do the project on the department machine. If you do the project on your personal computer, make sure you test it and have the same result on the department machine before you hand in.