CptS -451 Introduction to Database Systems
Spring 2016

Project Milestone-2

Due Date: Tuesday March 22th, 11:59pm

Summary:

In this milestone you will:

- design the database schema for your application and provide the ER diagram for your database design,
- translate your entity relationship model into relations and produce DDL SQL statements for creating the corresponding tables in a relational DBMS,
- populate your database with the Yelp data and get to practice generating INSERT statements and running those to insert data into your DB,
- write triggers to enforce additional constraints,
- start developing your application GUI . In Milestone3 you will develop the full final application with all required features.

Milestone Description:

You need to complete the following in milestone-2:

1) (25%)Design a database schema that models the database for the described application scenarios in the project description 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.

   (Note: In your application you need to provide the list of business categories on the "Business Search" tab. In addition, when the user selects one or more categories, you need to list the attributes associated with those categories. In the JSON data, there is no explicit relationship between categories and attributes. However, each business has a list of categories and a list of attributes. You may assume that if an attribute is associated with a business, then it is associated to all categories of that business. In your database, you should have a table that associates attributes with categories.)

2) (5%) Translate your ER model into relations and produce DDL SQL (CREATE TABLE) statements for creating the corresponding tables in a relational DBMS. Note the constraints, including primary key constraints, foreign key constraints, not NULL constraints, etc. needed for the relational schema to capture and enforce the semantics of your ER design. Write your CREATE TABLE statements to a file named "*"<your-team-name>*_DDL.sql".

3) (50%) Populate your database with the Yelp data.

   a. Generate INSERT statements for your tables and run those to insert data into your DB. You will use your JSON parsing code from Milestone-1 and use the data you extracted from JSON objects to generate the INSERT statements for your tables.

   b. You may populate your DB with data in 2 different ways: (i) You may write the INSERT statements to a SQL script file and then run this (large) script file. (You will find some information about how to generate and run SQL scripts in Appendix-A of this document). (ii) Alternatively, you may embed your INSERT statement inside your software code and execute them one by one as you generate them. Please note that due to foreign key constraints, the order of tables you insert data to matters. The INSERTs to referenced tables should be run before INSERTs to referencing tables. Please do not create any INDEXES for your tables until you insert all the data.

4) After you insert all your data, retrieve the number of tuples in each of your tables and copy/paste that information into a text file. (Simply run the query "SELECT COUNT(*) FROM tablename" for each table in your database.) Copy the results to *"<your-team-name>*_TableSizes.txt file.

5) (5%) Re-calculate and update the "stars" and "review_count" information for each business.

   a. "Stars" value for a business should be updated to the average of the star ratings of all reviews for that business. Similarly, "review_count" should be updated to the number of reviews provided for that business. You should query the review table to calculate the average stars and number of reviews for each business. In grading, points will be deducted if you don't update these values and use the average stars and number of reviews information directly extracted from the business JSON objects.
   b. Write your UPDATE statements to a file named ""*<your-team-name>*_UPDATE.sql".

6) (10%) Create triggers to enforce the following constraints in your database:

   a. Whenever a new review is provided for a business, the "stars" and "review_count" values for that business should be automatically updated.

   b. New reviews can be provided for "open" (i.e., active) businesses only. Please note a business is active if its "open" key value is true (i.e., the "open" key in business objects.)

   Write your TRIGGER statements to a file named ""*<your-team-name>*_TRIGGER.sql".

7) (5%)Create a preliminary version of your GUI, where you list the business categories and let the user select one or more categories. When the user selects the categories, you list the attributes which are associated with those selected business categories. (See Appendix-C for an example screenshot.)

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-team-name>_ER.pdf*"
2. SQL script file containing all CREATE TABLE statements. Name this file *"<your-team-name>_DDL.sql*"
3. Text file containing all table sizes. Name this file *"<your-team-name>_* TableSizes.txt *l*"
4. SQL script file containing all UPDATE TABLE statements. Name this file *"<your-team-name>_UPDATE.sql*"
5. SQL script file containing all TRIGGER statements. Name this file *"<your-team-name>_TRIGGER.sql*"
6. Source code for the preliminary version of your application. Only submit your source code, not the data files.

Create a zip archive *"<your-team-name>_milestone2.zip*" that includes all the 6 items above. Upload your milestone-2 submission on Blackboard until the deadline. One submission per team is sufficient. Either of the team members can submit it.

You will demonstrate your Milestone-2 to the TA after spring break.

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 – How to create and run a SQL script file in MySQL

Simply open a text editor and put the following line at the beginning of the file:
use db_name;
(db_name is the name of your database)

Then write all of your queries, separating them using empty lines. Please make sure that each query is terminated by a ';'.
As an example, suppose that you have the following two queries, and your database name is yelpDB:
Q1:

```
select * from  reviewTable;
```

Q2:

```
select name from businessTable
where  star>3;
```

Your script file should then look like as follows:

```
use yelpDB;
select * from reviewTable;
select name from businessTable
where  star>3;
```

You should save this file with a ".sql" extension.

Running The Script

Start a console window (on Windows: run cmd), and browse to the installation directory of MySQL (for example: "C:\Program Files\MySQL\MySQL Server 5.x\bin\mysql")

Assuming that you have saved the script file in the folder c:\myfolder, run the following in command line:

```
shell> mysql  -u root –p db_name  <  c:\myfolder\your_script_file.sql
```

If you would be running mysql with another username (other than root), replace root with that username. You will be asked to enter your password for the username you specify.

If you already placed a "use db_name" statement in the beginning of your script file, it is unnecessary to specify the database name on the command line. You can simply run the following:

```
shell> mysql  -u root –p  <  c:\myfolder\your_script_file.sql
```

How to run the script and create the output file
Let's assume you have created your sql script file and saved it as your_script_file.sql.
If you would like to save query results into a file, then run a statement like the following in the command line. This would save the query results into your_output_file.txt:

```
shell> mysql -u root –p db_name < c:\myfolder\your_script_file.sql >
C:\myfolder\your_output_file.txt
```

Appendix B – How to list tables and their schema descriptions at the MySQL Command Line Client

The MySQL Command Line client is useful for running queries as well as displaying what tables are in a MySQL database. After logging into the MySQL command line client and selecting a database, you can list all the tables in the selected database with the following command:

```
mysql> use testDB;
mysql> show tables;
```

(mysql> is the command prompt, testDB is the name of the database, and "show tables;" is the actual query in the above example).

This returns the following:

```
+----------------+
| Tables_in_testDB |
+----------------+
| table1         |
| table2         |
+----------------+
2 rows in set (0.00 sec)
```

This shows us that there are two tables in the “testDB” database called "table1" and "table2". We can show the schema of the table using the "desc" command.

```
mysql> desc table1;
```

The test database table returns a result like the following, showing there are 4 columns and what types they have, etc.

```
+--------------+-----------------+------+-----+-----------------+----------------+
| Field        | Type            | Null | Key | Default         | Extra          |
+--------------+-----------------+------+-----+-----------------+----------------+
| id           | integer                | NO   | PRI | NULL            | auto_increment |
| name         | varchar(50)     | NO   |     | NULL            |                |
| value        | varchar(50)     | NO   |     | NULL            |                |
| updatetime   | datetime        | YES  |     | NULL            |                |
+--------------+-----------------+------+-----+-----------------+----------------+
4 rows in set (0.00 sec)
```

Appendix C – Sample Screenshot for Preliminary Application