Revision: 02/13/2019

COMS 363 Section 2
Spring 2019

Project: What did our state legislatures say on Twitter?

Percentage in your final grade: 30%
Total points: 300

Submission due dates:

Part I: Tuesday March 12, 2019 by 11:50pm
Part II: Thursday April 18, 2019 by 11:50pm
Part III: Friday May 3, 2019 by 11:50pm

Per course policy, late submissions are not graded

If your team member does not contribute to the project, inform the instructor as early as possible, but no later than March 12, 2019.

This project asks a two-student team to develop a web database application for social scientists and journalists (clients) to examine Twitter communications of legislators and presidential candidates around the time of the 2016 presidential election till December 31, 2018. The team will be given tweets of Twitter user accounts of the 2016 presidential candidates, state's senators, state's house representatives, state's reporters, and state's Senate.

Tasks

1. (25 points) Design an ER diagram to model the data. Use the notations we studied in class.
2. (35 points) Design a relational database to store the given data. Import the data into the database. The DBMS to be used is MySQL Server 8.0.
3. Write a web-database application that performs all the functionality listed in Table 1. The application is to be implemented as Java Server Page (JSP) pages running on Apache Tomcat 8. These pages issue SQL statements to the backend MySQL database of your group on cs363winservdb through JDBC API.

   Grading criteria on the application (100 points):

   1. The application must take advantage of MySQL as much as possible to provide the correct output. The application that brings irrelevant data from the database and uses Java code to filter out irrelevant data will lose major points even the output is correct.
   2. The application must ensure that each functionality in Table 1 is its own transaction.
   3. The application must prevent against SQL injection caused by invalid input.
   4. Code must be documented. Each JSP page must include the author(s) of the page.

   Other requirements for the application (40 points):

- The application must not use root account or account that has privileges to access other data beyond the database of the project. The password for the account must not be in the top 25 most popular passwords according to this site.
    - https://www.welivesecurity.com/2018/12/17/most-popular-passwords-2018-revealed/
- The application must have a login page for only authorized users to perform certain actions. Users with the administrator privilege can do every functionality, but users with the read-only privilege can only view query results, but cannot add data (Table 1 functionality I) into the database or delete data (Table 1 functionality D) from the database.
- The passwords for authorized users must be kept in the database in an encrypted format (SHA or SHA2).

User interface is not graded, but it needs to be obvious for users to do the required functionality.

4. (60 points) Optimize your queries by doing physical database design and rewriting queries in Table 1 except functionalities I and D. Submit the design and an average query response times over 5 runs of the each of these selected queries.

5. (40 points) Individual contribution; both members of the team must be involved in every part of the project. The submission requires a project work log and individual work logs in a text format. Members who do not submit their individual work log will lose points on their contribution. The team will also lose points if starting the project in the last two weeks of the semester.

Database Requirements

- Tweets have properties: id, retweet_count (the number of retweets of this tweet), retweeted (whether this tweet has been retweeted), tweet text, created_at (timestamp---the number of milliseconds since 1/1/1970---in which the tweet was posted), day, month, and year.
- Tweets have zero or more hashtag and a hashtag must be used in at least one tweet.
- Tweets have zero or more url. A url must appear in at least one tweet.
- Users have the following properties: name, screen name, followers (indicating the number of followers), following (indicating the number of people this user follows), sub_category, category, location, and name. The sub_category indicates the party to which the user belongs: ‘GOP’, ‘Democrat’, ‘na’, or null. The category property is either a Senate account (senate_group), presidential_candidate, reporter, Senator, General, or null. The name property can have an empty string as a value. The screen name is unique.
- Each user has at most one state he/she belongs. Presidential candidates are not associated with any state.

Revision: 02/13/2019

Table 1: Functionalities of the application

ID	Description
Q1	List *k* most retweeted tweets in a given month and a given year; show the retweet count, the tweet text, the posting user's screen name, the posting user's category, the posting user's sub-category in descending order of the retweet count **Input:** value of k (e.g., 10), month (e.g., 1), and year (e.g., 2016) **Rationale:** This query finds *k* most influential tweets in a given time frame and the users who posted them.
Q2	In a given month of a given year, find *k* users who used a given hashtag in a tweet with the most number of retweets; show the user's screen name, user's category, tweet text, and retweet count in descending order of the retweet count. **Input:** value of *k*; hashtag, month, and year **Rationale:** This query finds *k* most influential users who used a hashtag of interest that may represent a certain agenda.
Q3	Find *k* hashtags that appeared in the most number of states in a given year; list the total number of states the hashtag appeared, the list of the distinct states it appeared, and the hashtag itself in descending order of the number of states the hashtag appeared. **Input:** value of k, year **Rationale:** This query finds *k* hashtags that are spread across the most number of states, which could indicate a certain agenda that is widely discussed. **Hint:** Use concat() to create a list
Q6	Find *k* users who used a certain set of hashtags in their tweets. Show the user's screen name and the state to which the user belongs in descending order of the number of followers. **Input:** value of k, hashtags (e.g., GOPDebate, DemDebate) **Rationale:** This is to find *k* users who share similar interests.
Q10	Find the list of distinct hashtags that appeared in one of the states in a given list in a given month of a given year; show the list of the hashtags and the names of the states in which they appeared. **Input from user:** list of the state, (e.g., Ohio, Alaska, Alabama), month, year **Rationale:** This is to find common interest among the users in the states of interest.
Q15	Find users in a given sub-category along with the list of URLs used in the user's tweets in a given month of a given year. Show the user's screen name, the state the user belongs, and the list of URLs **Input:** sub-category (e.g., GOP), month, year **Rationale:** This query finds URLs shared by a party.
Q23	Find k most used hashtags with the count of tweets it appeared posted by a given sub-category of users in a list of months. Show the hashtag name and the count in descending order of the count. **Input:** sub-category (e.g., GOP), a list of months (e.g., 1, 2, 3), year=2016, value of k
Q27	Given a month and two selected years, report the screen names of influential users (based on top k retweet counts in that month in the two selected years). **Input:** value of k (e.g., 10), month (e.g., 1), year1 (e.g., 2016), year2 (e.g., 2018)
I	Insert information of a new user into the database. **Input:** All relevant attribute values of a user
D	Delete a given user and all the tweets the user has tweeted, relevant hashtags, and users mentioned **Input:** screen name of the user to delete Must check that a user is valid before doing so. If the user's screen name is not valid, abort the transaction.

The value of *k* is between 1 and 100.

Revision: 02/13/2019

Submissions per group

1. Submission for ER diagram

   Due: Tuesday March 12, 2019 by 11:50pm

   - ER diagram of the database in pdf format

   Checklist to avoid point deduction:

   - Use the notations studied in class.
   - Make sure that the primary key of each entity set is indicated in the diagram.
   - Make sure that no relationship sets have a primary key attribute because each relationship can already be uniquely identified by the entities participating in the relationship.
   - If there is any candidate key, write the name of the attribute(s) that form the candidate key and the name of the entity set. For example, University ID is a candidate key of the Student entity set.
   - Make sure that all the given constraints (e.g., key constraint, overlapping constraint, covering constraint) are specified in the ER diagram.
   - Make sure that arrows (if any) are pointed toward the diamond notation for the corresponding relationship set.
   - If you think some constraints are missing, you may add it in as long as the added constraint does not conflict with the given constraints. **But make sure to add in the assumption for your added constraint as part of your database requirement.**
   - Do not provide an extended ER diagram that includes types of attributes, indexing, etc. Such a diagram includes physical database design choices in it.

2. Submission of scripts of that create relational databases with data populated.
   Due: April 18, 2019 by 11:50pm

   1. The SQL DDL script file (projectDDL.sql) to create the relations with necessary constraints.
   2. The SQL script (projectinsert.sql) or your program to insert the data from the given .csv files. Consult this page for bulk loading of data (https://dev.mysql.com/doc/refman/8.0/en/load-data.html).
   3. The SQL script that implements the functionalities in Table 1 (except I and D) before optimization. In the script, input from user is given in a variable.
      set @k=10
   4. The database of your group in cs363winservdb must have the data populated.
   5. The SQL script (createuser.sql) that creates a user “cs363@%1” with the standard authentication method. This user has the privilege only to view, drop, create, insert, delete the data in your group database.

   Checklist to avoid point deduction

   - SQL statements in projectDDL.sql and projectinsert.sql must run successfully on cs363winservdb. The content of this file projectDDL.sql must not include physical database design choices such as indexing, or which MySQL engine to use.
   - In this file projectDDL.sql, the same table name must be dropped before it is recreated.
   - Your design at this stage must aim to reduce redundancy as the main objective.

- Ensure that the script createuser.sql run on your MySQL server 8.0 that you install on the VM.

3. Submission of JSP code and query optimization result
Due: May 3, 2019 by 11:50pm

1. The SQL script named “physicaldesign.sql” that implements the physical database design. The design is to use index provided by MySQL and rewrite queries to speed up the query performance. The goal is to reduce the query response time and keep redundancy at a minimum.

2. A pdf file named “perf.pdf” that has Table 2 filled up and the optimization made for each query in reference to the conceptual design you submit earlier.

Table 2: Average execution time measured by the server of 5 runs for each query

Buffer pool size: __________ GBytes

Query ID	**Optimization method**	**Before optimization (ms)**	**After optimization (ms)**
Q1			
Q2			
Q3			
Q6			
Q10			
Q15			
Q23			
Q27			

1. Source code: Project folder with JSP source files. Make sure that your code works in the test environment to be announced by the instructor. Separate styles from JSP. Indent the code so that it is easily read. Comment the code and provide the authors of the code.

2. TeamWorklog.txt has the record of what each member of the team has done for each week. Meeting during project work time in class or outside of class needs to be recorded.

3. Two individual work logs, each is named <netid>Worklog.txt; the file has a record of what each team member with that netid has done each week and approximate work hours. These documents are very important as they are used for grading individual contributions of the team members.

Example TeamWorklog.txt

Week of Feb. 11, 2019

- Team received the project description and the data
- Team meeting (1 hour) and individual tasks assigned
  - James's task:
    - design the ER diagram;
    - check the schemas designed by Liam
  - Liam's task:
    - proofread the ER diagram
    - design the relational schemas from the ER diagram

Example of individual Worklog.txt

James:

Week of Feb. 11, 2019

- Designed the ER diagram (4 hours). Sent it to Liam for review.