CSE 4/560 Project1: TinyTwitter

Due 23:59 10/03/2018 EST

September 25, 2018

1 Submission

Failure to comply with the submission specifications will incur penalties for EACH violation.

- What to submit: A zip file has to be submitted through the 'submit_cse460' ('submit_cse560') submit script by 10/03/2018 11:59PM EST. Only zip extension will be accepted, please **don't** use any other compression methods such as tar or 7zip.
- Zip file naming: Use *ubit_proj1* (**NO SPACE!**) for the filename, for example: *jsmith_proj1.zip*, where *jsmith* is the ubit. The project is an **INDIVIDUAL** project, the filename should contain only one ubit.
- Sub-structure of zip file: On unzipping the zip file, there should be a folder named with your ubit *ubit_proj1*, under the folder *ubit_proj1*, there should be two files: (1) a .pdf report, and (2) a .sql SQL file.

2 Description

You are to design and implement the database schema for TinyTwitter, which is a social network website provides simple functions, main functions of TinyTwitter are as the following:

- User management: user sign up, user login/logout.
- Tweet management: create and post new tweet, delete a tweet.
- Friendship management: follow, unfollow.
- User-Tweet relationship management: like, comment, re-tweet.

Your DB schema must be able to support all the functions listed above. You are required to use E/R modeling (an online E/R diagram tool is `https://www.draw.io/`) to design and present your database, and map your E/R model to a relational database schema and implement the mapped schema use a RDBMS by a set of CREATE TABLE statements. Recommended RDBMSes are:

- PostgreSQL (`https://www.postgresql.org/download/`)
- MySQL(`https://dev.mysql.com/downloads/`), you must install MySQL Community Server to use MySQL, and you need one of MySQL Shell and MySQL WorkBench for SQL development. If you need a GUI, you may prefer to use MySQL WorkBench.

You also need to explain how your modeling supports all the required functions. The final product of this is a report (.pdf file), of which the structure will be given later, and a SQL file (.sql file type) which contains all the CREATE TABLE statements you used for implementing your RDB schema.

Note: you ONLY need to design and implement the database schema for the data needed to support the functions, you are NOT required to implement the functions.

3 Requirements of functions

The specific requirements for each function of TinyTwitter are given in this section. **Note that you need to analyze and design the entity sets, attributes and relationship sets by yourself**, while this project description only give some of them for clarifying the system requirements.

3.1 User management

Users sign up use their *email addresses*, the email addresses are used as the TinyTwitter *usernames* of the users, users need to set their *passwords* and *display names*, where:

- username, password and display name are `strings`.
- one account, i.e. one username, has only one display name, and one display name corresponds to only one username.
- one email address can be used to register only one account.

3.2 Tweet management

Users can create a new **tweet**, which is a `string` consists of up to 140 characters (including special characters and white spaces).

Users can choose to post their new tweets or save a tweet as a draft (without posting it).

The basic set of attributes of tweets are:

- time: the posted time of a tweet

- content: the tweet's content, i.e. a string consists of up to 140 characters (including special characters and white spaces)

You need to think about and model the **tweet drafting** function by yourself.

3.3 User relationship management

A user can follow other users, by following other users, users can see the posted tweets from their following users.

3.4 User-Tweet relationship management

Users can like a tweet and can see the details of "likes" of a tweet: every user can see the number of "likes" of a tweet, users can also explore who liked the tweet.

Users can comment a tweet and can see the details of comments of a tweet: every user can see the number of comments of a tweet, the content of a comment and the user who posted a comment.

Users can also re-tweet others' tweets, either re-tweeting with additional comments or without additional comments.

4 Report template

Here's a template of the report, your report must contain each of the following sections, you can extend this template to include any necessary sections for your design.

- E/R schema: in this section you need to introduce your E/R schema and include the picture of your E/R schema.
- Relational database schema: in this section, you must
  - discuss briefly how you map the E/R schema to your relational database schema. If any design choice is made in the mapping process, illustrate and explain it briefly.
  - discuss how your relational database schema satisfies all the requirements listed in section 3.
- Further discussion: in this section, you need to discuss the advantages and disadvantages of your design.