CSE 460 Project1: TinyTwitter

Due 23:59 09/28/2017 EST

Date 09/07/2017

1 Submission

Failure to comply the submission specifications will incur penalties for EACH violations

1.1 What to submit

A zip file has to be submitted through the 'submit_cse460' submission script by 09/28/2017 11:59PM EST. Only zip extension will be accepted, please **don't** use any other compression methods such as tar or 7zip.

1.2 Zip file naming convention

Write your *ubit* (**NO SPACE!**) for the filename. The project is an **INDIVIDUAL** project, the filename contains only one ubit, for example: *jsmith.zip*, where *jsmith* is the ubit.

1.3 Sub-structure of zip file

- On unzipping the zip file, there should be a folder named with your ubit *ubit*.
- Under the folder *ubit*, 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:

- 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. Your are required to use E/R modeling (an online E/R diagram tool is `https://www.draw.io/`) to model the data, then map your E/R model to a relational database schema and implement the mapped schema use a DBMS by a set of CREATE TABLE statements. Recommended DBMSes 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 write how your relational DB schema supports all the required functions. The final product of this is a report (.pdf file) and a SQL file (.sql file type) which contains all the CREATE TABLE statements you used for implementing your DB schema. More details of the report will be given later.

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 the *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 an un-posted tweet as a draft.

The basic set of attributes of tweets are:

- time: the posted time of a tweet
- content: the tweet's content

You need to think about and model the 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. By default, the users see their own posted tweets and the posted tweets from their following users after they login, and all the tweets will be displayed by the order newest first (newsfeed style).

3.4 User-Tweet relationship management

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

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

4 Report template

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

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