Document providing guidance with issues about the project

1. To easily see the content of the tweets.csv, change the file extension to txt like tweets.txt. Run Excel and open this text file. Excel will ask to use a delimiter to separate a row into columns. Specify the semi-colon as the delimiter.

2. Load data from a csv file. There are multiple ways. The description here shows an example of importing the data from tweets.txt using the load data infile statement.

   Run forimport.sql to create a database named “test” and two relations, tweet and newtweet, in the newly created database. Then, import the data from tweets.txt.

   Use Load data infile statement. This method is the fastest but has a cryptic error message that is difficult to pinpoint where the error is. See https://dev.mysql.com/doc/refman/8.0/en/load-data.html. This requires that the data file you want to import is in a certain directory as indicated by the variable name 'secure_file_priv'. To know the directory name, execute the following statement.

   ```
   show variables like 'secure_file_priv';
   ```

   Put your file in the directory output as the result of the statement above.

   Let’s say that the output is "C:\ProgramData\MySQL\MySQL Server 8.0\Uploads".

   For mac user, refer to https://dba.stackexchange.com/questions/168768/mysql-on-macos-sierra-secure-file-priv-setting to update 'secure_file_priv'

   The following example loads the data in tweets.txt in C:\ProgramData\MySQL\MySQL Server 8.0\Uploads into the tweet relation, ignoring the last column, which is the posting_user. The IGNORE 1 lines is to ignore the header line. Set the default database to test first before executing the statement.

   ```
   LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/tweets.txt'
   INTO TABLE tweet
   FIELDS TERMINATED BY ';' OPTIONALLY ENCLOSED BY '"'
   LINES TERMINATED BY '\n'
   IGNORE 1 LINES
   (tid,textbody,retweet_count,retweeted,posted,@col6);
   ```

   The following example shows transformation of the data before they are entered into newtweet.

   Note that @col6, the posting_user, is not entered into the newtweet. Also, the posting_time is transformed to day, month, and year and inserted into the respective columns.

   ```
   LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/tweets.txt'
   INTO TABLE newtweet
   FIELDS TERMINATED BY ';' OPTIONALLY ENCLOSED BY '"'
   LINES TERMINATED BY '\n'
   IGNORE 1 LINES
   (tid,textbody,retweet_count,retweeted,@col5,@col6)
   set day_posted= day(str_to_date(@col5, '%Y-%m-%d %H:%i:%s')),
   month_posted= month(str_to_date(@col5, '%Y-%m-%d %H:%i:%s')),
   year_posted= year(str_to_date(@col5, '%Y-%m-%d %H:%i:%s'));
   ```

   If you do not want to use the load data infile to transform the input directly, you can load the data into the tweet table and do the following to do the data conversion. Either way is acceptable for the project.

```
INSERT into test.newtweet
SELECT tid, textbody, retweet_count, retweeted, day(posted), month(posted), year(posted)
FROM test.tweet;
```