



CENG 351

Data Management and File Structures

Fall 2016-2017

Programming Assignment 1

Due date: 19 12 2016, Monday, 23:55

1 Introduction

In the scope of this assignment, you are asked to create a system by designing queries and implementing pre-defined functions to administer a database for a stackoverflow which is designed to store information about the users, articles, comments and reputation information about users. For this, you have certain tasks and well-defined interfaces. What you will do is to implement the provided interfaces to accomplish the given tasks. All necessary data files, classes and the interface you will implement will be provided as source files. Do not confuse interface with graphical user interface (GUI). You will not design any GUI in the scope of this assignment. You will be familiar with interface which is a data type like class and enum. The first thing you should do is implementing functions which create the necessary tables corresponding to the schema given in Section 3. Then, you should design queries to accomplish the given tasks. Lastly, you should implement the interface using the queries you have designed as they give the desired results when defined parameters are given. You will not implement Evaluation class. It will be implemented by me to manipulate the database through the pre-defined interface and evaluate your implementations. Your task is to build up classes which implement the provided interfaces.

2 Objectives

This assignment aims to help you get familiar with

- Java programming language basics,
- Object oriented programming concepts,
- Connecting and querying to MySQL Server using JDBC.

3 Schema

You will use (strictly) the schema given below in the scope of this assignment.

```
user(userID:char(10), username:char(30), registrationDate:date, lastLoginDate:date),  
article(articleID:char(10), userID:char(10), name:char(80), description:char(130), date:date, rating:int),  
comment(commentID:char(10), articleID:char(10), userID:char(10), message:char(130), date:date, rating:int),  
reputation(reputationID:char(10), userID:char(10), weeklyReputation:int, monthlyReputation:int, yearlyReputation:int,  
alltimeReputation:int)
```

Your task is to generate a class named STACKOVERFLOWDB (should belong to package ceng.ceng351.stackoverflowdb) which implements ISTACKOVERFLOWDB interface. You can create any additional classes if necessary. STACKOVERFLOWDB class should be able to accomplish the following tasks:

- Creating the database tables,
- Inserting data into user table,
- Inserting data into article table,
- Inserting data into comment table,
- Inserting data into reputation table,

- Find the article which has the highest rating,
- Find the user whose comment message does not contain 'mysql' and the comment has the lowest rating,
- List the userID, username, registration date and weekly reputation of users whose registration date is after a given date,
- List the username, message, and rating of comments which does not contain given string in their message and it's rating is more than given rating,
- Multiply rating of the comment by 2 whose date is after for a given date,
- List the userID, username and last login date of user/s who comment all the articles commented by a given user. (given userID),
- List username, comment message and rating of comments whose comment rating is more than given number of ratings for a given article (given articleID),
- List the article name, username, article date and article rating of articles commented by this user such that none of these articles are commented by any other user,
- Find the article whose date is before given date and whose user has the highest weeklyReputation,
- Delete the articles whose date is after a given date and list the username, date and rating of these articles,
- Dropping the database tables.

Tasks are explained in more detail below. For each task, there is a corresponding method in ISTACKOVERFLOWDB interface. Necessary data files (for populating the tables) to accomplish these tasks will be provided. In **data** folder there are 4 txt files corresponding to each table. You will use these tables when you are inserting data. Data files, interfaces and classes for fulfilling these tasks will be provided as source files. You can assume all information will be complete and consistent, i.e. all necessary data will be inserted before executing a query. You can find detailed description about the usage of the functions in provided source files. Do not forget to define **foreign keys** when you are creating tables. Please do not forget to use DISTINCT keyword when appropriate in your MySQL queries.

3.1 Creating the database tables

You will create all the tables according to the schema declared in Section 3.

You can assume that tables will be created before executing any other database operation.

Returns the number of tables that are created successfully.

3.2 Inserting data into user table

You will insert data into user table.

Returns the number of rows inserted successfully.

3.3 Inserting data into article table

You will insert data into article table.

Returns the number of rows inserted successfully.

3.4 Inserting data into comment table

You will insert data into comment table.

Returns the number of rows inserted successfully.

3.5 Inserting data into reputation table

You will insert data into reputation table.

Returns the number of rows inserted successfully.

3.6 Find the article which has the highest rating

Output the username, date and rating of article which has the highest rating (In tab delimited).

Order the results by username in ascending order.

3.7 Find the user whose comment message does not contain 'mysql' and the comment has the lowest rating

Output the username, message, rating and alltimeReputation of user whose comment message does not contain 'mysql' and the comment has the lowest rating (In tab delimited).

Order by username in ascending order.

3.8 List the userID, username, registration date and weekly reputation of users whose registration date is after a given date

Output the userID, username, registration date and weekly reputation of users whose registration date is after a given date (In tab delimited).

Order by userID in ascending order.

3.9 List the username, message, and rating of comments which does not contain given string in their message and it's rating is more than given rating

Output the username, message, and rating of comments which does not contain given string in their message and it's rating is more than given rating (In tab delimited).

Order by username in ascending order.

3.10 Multiply rating of the comment by 2 whose date is after for a given date

Update the rating of the comment by multiplying 2 whose date is after for a given date (In tab delimited).

Output the number of rows affected.

3.11 List the userID, username and last login date of user/s who comment all the articles commented by a given user. (given userID)

Output the userID, username and last login date of user/s who comment all the articles commented by a given user (In tab delimited).

Order by userID in ascending order.

3.12 List username, comment message and rating of comments whose comment rating is more than given number of ratings for a given article (given articleID)

Output the username, comment message and rating of comments whose comment rating is more than given number of ratings for a given article (given articleID) (In tab delimited).

Order by username in ascending order.

3.13 List the article name, username, article date and article rating of articles commented by this user such that none of these articles are commented by any other user

Output the article name, username(who comments), article date and article rating of articles commented by this user such that none of these articles are commented by any other user (In tab delimited).

Order by article name in ascending order.

3.14 Find the article whose date is before given date and whose user has the highest weeklyReputation

Output the username, date and rating of articles whose date is before given date and whose user has the highest weeklyReputation (In tab delimited).

Order by username name in ascending order.

3.15 Delete the articles whose date is after a given date and list the username, date and rating of these articles

First delete rows of articles whose date is after a given date then, output the username, date and rating of these articles (In tab delimited).

Order by username in ascending order.

3.16 Dropping the database tables

You will drop all the tables (if they exist).

Returns the number of tables that are dropped successfully.

4 Regulations

1. Programming Language: Java.
2. Database: An account on the MySQL server on my office machine will be created for each of you and an e-mail including credentials and connection configuration will be sent to your ceng mail. You will use JDBC driver to connect to the database. Your final submission should connect to the MySQL server on my office machine. So, make sure that the connection information is correct before submitting your homework.
3. Attachments: Necessary source files and JDBC driver will be provided.
4. Input: All strings will be case-sensitive and they will not include any non-English characters.
5. Late Submission: Late submission policy is stated in the course syllabus.
6. Cheating: We have zero tolerance policy for cheating. People involved in cheating will be punished according to the university regulations.
7. Newsgroup: You must follow the newsgroup (news.ceng.metu.edu.tr) for discussions and possible updates on a daily basis.
8. Evaluation: Your program will be evaluated automatically using "black-box" technique so make sure to obey the specifications.
9. Evaluation: Please, be noticed that you have to achieve the tasks only within your sql queries not with any other java programming facilities .

5 Submission

Submission will be done via COW. Create a compressed file named ceng.tar.gz that contains STACKOVERFLOWDB class and all other classes, created by you. You will not submit interface and class files provided by me. So, be sure you do not modify them during implementation. Because evaluation will be held with unmodified versions of them. The compressed file should contain a directory tree same as the package tree. That is, you should compress the directory named 'ceng' which contains a directory named 'ceng351' which contains a directory named 'stackoverflowdb' which contains your source files.

```
ceng
├── ceng351
│   └── stackoverflowdb
│       ├── STACKOVERFLOWDB.java
│       ├── AnotherClassIfYouNeed1.java
│       ├── AnotherClassIfYouNeed2.java
│       ├── ..
│       ├── ...
│       └── AnotherClassIfYouNeedN.java
```

6 Useful Links

- Learning the Java Language:
<http://docs.oracle.com/javase/tutorial/java/index.html>
- MySQL Reference Manual:
<http://dev.mysql.com/doc/refman/5.6/en/>
- Basic MySQL Tutorial:
<http://www.mysqltutorial.org/basic-mysql-tutorial.aspx>