Middle East Technical University　　Department of Computer Engineering

CENG 351

Data Management and File Structures

Fall 2018-2019

Programming Assignment 1

Due date: 12.12.2018, Wednesday, 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 digital music streaming service. 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 are 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:int, userName:varchar(60), email:varchar(30), password:varchar(30))
song(songID:int, songName:varchar(60), genre:varchar(30), rating:double, artistID:int, albumID:int)
artist(artistID:int, artistName:varchar(60))
album(albumID:int, title:varchar(60), albumGenre:varchar(30), albumRating:double, releaseDate:date, artistID:int)
listen(userID:int, songID:int, lastListenTime:timestamp, listenCount:int)

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

- Creating the database tables
- Inserting data into tables
- List songs which have the highest rating
- Find the most recent album for a given artist
- List songs that are listened by both users whose usernames are given
- List artists and number of times his/her songs have been listened by the given user

- List users who have listened all songs of a given artist
- List users and number of songs listened by this user such that none of these songs are listened by any other user
- List artists who have sung pop music at least once and whose average rating of all songs is greater than the given rating
- Retrieve songs with the lowest rating in pop category, in case of more than one song exist, retrieve the least listened ones
- Multiply rating of albums by 1.5 whose release dates are after for a given date
- Delete the song for the given song name
- Dropping the database tables

Tasks are explained in more detail below. For each task, there is a corresponding method in IMUSICDB interface. You need to implement them in MUSICDB class. Necessary data files (for populating the tables) to accomplish these tasks will be provided. In **data** folder there are 5 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 (10 pts)

You will create all the tables according to the schema described above.
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 tables (5 pts)

You will insert data into appropriate tables.
Returns the number of rows inserted successfully.

3.3 List songs which have the highest rating (5 pts)

List the artistName, songName, genre and rating of the songs which have the highest rating. (In tab delimited)
Order the results by artistName in ascending order.

3.4 Find the most recent album of given artist (5 pts)

Return the title, releaseDate and rating of an album of an artist whose name is given.(In tab delimited)

3.5 List songs that are listened by both users whose usernames are given (10 pts)

List the artistName, songName, genre and rating of the songs that are listened by both users whose usernames are given. (In tab delimited)
Order by rating in descending order.

3.6 List artists and number of times his/her songs have been listened by the given user (10 pts)

List the artistName and number of times his/her songs have been listened by the given user. (In tab delimited)
Order by artistName in ascending order.

3.7 List users who have listened all songs of a given artist (10 pts)

List the userID and userName, email and password of users who have listened all songs of an artist whose name is given. (In tab delimited)
Order by userID in ascending order.

3.8 List users and number of songs listened by this user such that none of these songs are listened by any other user (10 pts)

List the userID, userName and number of songs listened by this user such that none of these songs are listened by any other user. (In tab delimited)
Order by userID in ascending order.

3.9 List artists who have sung pop music at least once and whose average rating of all songs is greater than the given rating (10 pts)

List the artistID and artistName of artists who have sung pop music at least once and whose average rating of all songs is greater than the given rating. (In tab delimited)
Order by artistID in ascending order.

3.10 Retrieve songs with the lowest rating in pop category, in case of more than one song exist, retrieve the least listened ones (10 pts)

Retrieve the song with the lowest rating in pop category. If there exists multiple songs that hold this condition, retrieve the ones that are listened the least number of times.
Return the songID, songName, rating, genre, artistID and albumID of these songs. (In tab delimited)
Order by songID in ascending order.

3.11 Multiply rating of the albums by 1.5 whose release dates are after for a given date (5 pts)

Update the rating of the albums by multiplying 1.5 whose release dates are after for a given date.
Return the number of rows affected.

3.12 Delete the song for the given song name (5 pts)

Delete the song for the given song name.
Return the songID, songName, rating, genre, artistID and albumID of these songs. (In tab delimited)

3.13 Dropping the database tables (5 pts)

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 must use JDBC driver to connect to the database. Your final submission must 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 is 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: It is GUARANTEED that input files are correctly formatted and sample data will be given to you. There will be no surprises about the data, similar (and larger) data will be used while evaluating homeworks. Your program will be evaluated automatically using ”black-box” technique so make sure to obey the specifications. Please, be noticed that you have to accomplish 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 MUSICDB 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 'musicdb' which contains your source files.

```
ceng
└── ceng351
    └── musicdb
        ├── MUSICDB.java
        ├── AnotherClassIfYouNeed1.java
        ├── AnotherClassIfYouNeed2.java
        ├── ..
        ├── ...
        └── AnotherClassIfYouNeedN.java
```

6 Useful Links

- Java Documentation:
  http://docs.oracle.com/javase/tutorial/java/index.html
- MySQL Reference Manual:
  http://dev.mysql.com/doc/refman/8.0/en/
- Basic MySQL Tutorial:
  http://www.mysqltutorial.org/basic-mysql-tutorial.aspx