Middle East Technical University Department of Computer Engineering

CENG 351

Data Management and File Structures

Fall 2019-2020

Programming Assignment 1

Due date: 06.12.2019, 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 book store. 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.

author(author_id:int, author_name:varchar(60))

publisher(publisher_id:int, publisher_name:varchar(50))

book(isbn:char(13), book_name:varchar(120), publisher_id:int, first_publish_year:char(4), page_count:int, category:varchar(25), rating:float) **REFERENCES** publisher(publisher_id)

author_of(isbn:char(13),author_id:int) **REFERENCES** book(isbn) author(author_id)

phw1(isbn:char(13), book_name:varchar(120), rating:float)

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

- Creating the database tables

- Inserting data into tables
- Retrieving the required SQL queries for the given questions.
- Performing the required DML operations.
- Dropping the database tables

Tasks are explained in more detail below. For each task, there is a corresponding method in IBOOKDB interface. You need to implement them in BOOKDB class. 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 (10 pts)

```
public int createTables();
```

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)

```
public int insertAuthor(Author[] authors);
public int insertBook(Book[] books);
public int insertPublisher(Publisher[] publishers);
public int insertAuthor_of(Author_of[] author_ofs);
```

You will insert data into appropriate tables.
Returns the number of rows inserted successfully. In this task, please do **not** insert any data inside **phw1** table.

3.3 Query 1 (5 pts)

```
public QueryResult.ResultQ1[] functionQ1();
```

List isbn, first_publish_year, page_count and publisher_name of the book(s) which have the maximum number of pages.
Order the results by isbn in ascending order.

3.4 Query 2 (5 pts)

```
public QueryResult.ResultQ2[] functionQ2(int author_id1,int author_id2);
```

For the publishers who have published books that were co-authored by both of the given authors(author1 and author2); list publisher_id(s) and average page_count(s) of all the books these publishers have published.
Order the results by publisher_id in ascending order.

3.5 Query 3 (10 pts)

```
public QueryResult.ResultQ3[] functionQ3(String author_name);
```

List book_name, category and first_publish_year of the earliest published book(s) of the author(s) whose author_name is given.
Order the results by book_name, category and first_publish_year in ascending order.

3.6 Query 4 (10 pts)

```
public QueryResult.ResultQ4[] functionQ4();
```

For publishers whose name contains at least 3 words (i.e., "Koc Universitesi Yayinlari"), have published at least 3 books and average rating of all their books are greater than(>) 3; list their publisher_id(s) and distinct category(ies) they have published.
PS: You may assume that each word in publisher_name is seperated by a space.
Order the results by publisher_id and category in ascending order.

3.7 Query 5 (10 pts)

```
public QueryResult.ResultQ5[] functionQ5(int author_id);
```

List author_id and author_name of the authors who have worked with all the publishers that the given author_id has worked.
Order the results by author_id in ascending order.

3.8 Query 6 (10 pts)

```
public QueryResult.ResultQ6[] functionQ6();
```

List author_id(s) and isbn(s) of the book(s) written by "Selective" authors. "Selective" authors are those who have worked with publishers that have published their books only.(i.e., they haven't published books of different authors)
Order the results by author_id and isbn in ascending order.

3.9 Query 7 (10 pts)

```
public QueryResult.ResultQ7[] functionQ7(double rating);
```

List publisher_id and publisher_name of the publishers who have published at least 2 books in 'Roman' category and average rating of their books are more than (>) the given value.
Order the results by publisher_id in ascending order.

3.10 Query 8 (Bulk insert) (10 pts)

```
public QueryResult.ResultQ8[] functionQ8();
```

1. Some of the books in the store have been published more than once: although they have same names(book_name), they are published with different isbns. For each multiple copy of these books, find the book_name(s) with the lowest rating for each book_name and store their isbn, book_name and rating into **phw1** table using a single BULK insertion query. If there exists more than 1 with the lowest rating, then store them all.
2. After the bulk insertion operation, list isbn, book_name and rating of all rows in **phw1** table.
   Order the results by isbn in ascending order.

3.11 Query 9 (Update) (5 pts)

```
public double functionQ9(String keyword);
```

1. For the books that contain the given **keyword** anywhere in their names, increase their ratings by one. Please note that, the maximum rating cannot be more than 5, therefore if the previous rating is greater than 4, do not update the rating of that book.
2. After the update operation, return sum of the ratings of all books.

3.12 Query 10 (Delete) (5 pts)

```
public int function10();
```

1. Delete publishers in publisher table who haven't published a book yet.
2. After the delete operation, return count of all rows of the publisher table.

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 Odtu Class 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 OdtuClass. Create a compressed file named ceng.tar.gz that contains BOOKDB 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 'bookdb' which contains your source files.

```
ceng
└── ceng351
    └── bookdb
        ├── BOOKDB.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