CSCI3170 (2021-2022 Fall)
Introduction to Database Systems
Project – Library Inquiry System

TA In Charge: HAN, Chenxia (cxhan@cse.cuhk.edu.hk)

Group Registration Deadline: 23:59 4th October 2021
Phase 1 Deadline: 23:59 18th October 2021
Phase 2 Deadline: 23:59 29th November 2021

1. Introduction

You are required to implement a Library Inquiry System for a library so that all information about books in the library, library users, and loans are stored. The system shall support interactive operations from operators. You are required to use Java JDBC API to access the database. Our tutors will give a tutorial on how to use the JDBC API. You are required to implement a Java command line application to realize all system functions stated in this specification.

This project is divided into two phases:
In phase 1, you are required to design the database for the system (including an ER-diagram and a relational schema). After the deadline of Phase 1, a suggested solution will be provided. **You are required to use the suggested solution to complete Phase 2.**

In Phase 2, you are required to implement the Library Inquiry System as a **Java** command-line program. Our tutors will give tutorials on how to connect to a **MySQL** database system with JDBC API and deploy your work on the required platform.

This is a group project, and each group should have at most three members. ONLY one copy of solution is required for each group. Please fill out the group registration form in the Blackboard system before the group registration deadline.

2. Milestones

Preparation

- Read the document thoroughly and make sure you understand all the assumptions and regulations stated in Section 4.

Phase 1 (20 %)

- According to the data specifications in Section 3, design an ER-diagram and transform it into a relational schema **without any redundant fields and tables**.

Phase 2 (80 %)

- According to the **suggested solution** of Phase 1, implement a Java application that fulfills all requirements stated in Section 5.
- Debug your system with different datasets and user inputs.
- Write a readme file to describe the compilation and deployment of your system.

3. Data Specification

All data files of the system are in Linux text file format (i.e., Newline character is **\n**) encoded in ASCII. Your Java command line application is required to read records stored in the files and inserts them into appropriate tables of the provided MySQL DBMS via JDBC API. There are five input files, a list of user categories, a list of library users, a list of book categories, a list of books and a list of check-out records. Each line of each input file is a sequence of attributes delimited by a tab (**\t**) character. The definition of each attribute in each input file is defined in the corresponding subsections. The order of the attributes within a line of each input file follows that of the attribute in the corresponding subsection. A sample data set will be provided after the deadline of Phase 1.

3.1. User Categories – user_category.txt

Library users are divided into different categories. The loan period and the maximum number of books that can be borrowed by a library user are determined by his or her category. Each user category has a unique user category ID.

Attribute Name	Format	Description
User Category ID	Non-empty positive integer with 1 digit	A unique identifier for a user category.
Max books	Non-empty positive integer with at most 2 digits	The maximum number of books that can be borrowed by the library user in the corresponding category.
Loan period	Non-empty positive integer with at most 2 digits	The maximum number of days for the library user in the corresponding category to borrow a book.

3.2. Library Users – user.txt

Each library user has a unique user ID and belongs to exactly one category.

Attribute Name	Format	Description
User ID	Non-empty string with 10 characters	A unique identifier of the library user.
Name	Non-empty string with at most 25 characters	The name of the library user.
Age	Non-empty positive integer with 3 digits	The age of the library user.
Address	Non-empty string with at most 100 characters	The address of the library user.
User Category ID	Non-empty positive integer with 1 digit	It indicates to which category the library user belongs.

3.3. Book Categories – book_category.txt

Books are divided into different categories. Each book category has a unique book category ID.

Attribute Name	Format	Description
Book Category ID	Non-empty positive integer with 1 digit	A unique identifier for a book category.
Book Category Name	Non-empty string with at most 30 characters	The name of the book category.

3.4. Books – book.txt

book (callnum, title, publish, rating, tborrowed, bcid)

Each book is a literary composition written by one or more authors and a library stores a number of physical copies for each book. Each book has a unique call number

Attribute Name	Format	Description
Call number	Non-empty string with 8 characters	It is used for the library users to search for the book.
Number of copies	Non-empty positive integer with 1 digit	The number of identical copies of the book.
Title	Non-empty string with at most 30 characters	The title of the book.
Author(s)	Non-empty string with at most 25 characters for each author	Author name(s) of the book concatenated as a string with comma character as the delimiter.
Date of publication	Date format see 4.1	The date that the book is published.
Rating	Non-negative floating-point number	The rating of the book rated by the users.
Number of times borrowed	Non-empty non-negative integer with 2 digits	The number of times the book has been borrowed.
Book Category ID	Non-empty positive integer with 1 digit	A unique identifier for a book category.

0 1 2 3 4 5 6 7

3.5. Checked-Out Records - check_out.txt

Each check-out record shows a borrow history of a library user.

Attribute Name	Format	Description
Call number	Non-empty 8 characters	The call number of the checked-out book copy.
Copy number	Non-empty 1-digit Positive integer	The copy number of the checked-out book copy.
User ID	Non-empty 10 characters	The user ID of the borrower.
Check-out date	Date format see 4.1	The date that the book is checked out.
Return date	Date format see 4.1 & 4.8	The date that the book is returned.

borrow (libuid, callnum, copynum, checkout, return)

4. Assumption and Regulations

4.1. System

- All numerical values will not be larger than the maximum integer value that can be handled by Java.
- The system is case sensitive.
- All dates follow the format "DD/MM/YYYY", e.g., "16/09/2021".
- There is no duplicate row in any input files.
- You may assume that any user inputs to the system are **correct in format only**.
- You may assume that all data files are **correct in format and content**.

4.2. User Categories

- Each user category has a unique *user category ID,* and it can be used to identify a user category.
- Some user categories may have the same *max books* or *loan period*.

4.3. Library Users

- Each library user has a unique *user ID,* and it can be used to identify a library user.
- Some library users may have the same *name* or *address*.
- A user can only belong to one *user category*.

4.4. Book Categories

- Each book category has a unique *book category ID*, and it can be used to identify a book category.
- It's possible that none of the books fall into a certain book category.

4.5. Books

- Each book has a unique *call number,* and it can be used to identify a book.
- Each book must have at least one *author*; Some books may have the same *title* or *author*.
- The number of copies is guaranteed to be less than ten.
- The rating ranges from 0 to 10. The higher, the better. If a book has not been rated by any user, the *rating* is set to **NULL**.
- The *number of times borrowed* is zero if the book has never been borrowed by any user.
- A book can only belong to one *book category*.

4.6. Book Copies

- A book copy refers to a physically existing book copy that can be borrowed by a library user in the library.
- Each book copy of a book has a *copy number*. If a book has n book copies, the *copy number* of those n book copies ranges from 1 to n. For example, if a book has four copies, these four book copies have the same *call number*, and their *copy numbers* are 1, 2, 3 and 4.
- Each book copy has a unique pair of *call number* and *copy number* and they can be used jointly to identify a book copy.

4.7. Authors

- An author can be uniquely identified by his/her *name*.
- An author may write more than one book.

4.8. Checked-Out Records

- Each check-out record has a unique set of {*Call number*, *Copy number*, *User ID* and *Check-out date*} and this set of attributes can be used to identify a check-out record.
- Some library users may have never checked out any book copies.
- Some book copies may have never been checked out.
- A user may borrow the same book copy for more than one time.
- The *return date* of a book copy is **NULL** if the book copy is not returned.
- For each book copy, there is only one check-out record with a NULL return date.
- Increase the *number of times borrowed* by one if a user successfully borrows and returns the book.
- Every time a user returns a book, he/she is required to rate the borrowed book.

5. System Function Requirements

You are required to write a simple command line application in Java. After performing a function, the program should **display the last appeared menu**. The following sections describe the functionalities of the system.

In the system, you're required to ask the user to choose one of the three interfaces – Administrator, Library User, and Librarian.

```
Welcome to Library Inquiry System!

-----Main menu-----
What kinds of operations would you like to perform?
1. Operations for Administrator
2. Operations for Library User
3. Operations for Librarian
4. Exit this program
Enter Your Choice:
```

Figure 1: Example main menu.

5.1. Administrator

The system should let administrators to perform the following operations:

- **Create table schemas in the database**: This function creates all the tables for this system based on the relational schema given.

```
-----Main menu-----
What kinds of operations would you like to perform?
1. Operations for Administrator
2. Operations for Library User
3. Operations for Librarian
4. Exit this program
Enter Your Choice: 1

-----Operations for administrator menu-----
What kind of operation would you like to perform?
1. Create all tables
2. Delete all tables
3. Load from datafile
4. Show number of records in each table
5. Return to the main menu
Enter Your Choice: 1
Processing...Done. Database is initialized.
```

Figure 2: Example interactive input and output while creating table schemas.

- **Delete table schemas in the database**: This function deletes all existing tables in the system.

```
-----Operations for administrator menu-----
What kind of operation would you like to perform?
1. Create all tables
2. Delete all tables
3. Load from datafile
4. Show number of records in each table
5. Return to the main menu
Enter Your Choice: 2
Processing...Done. Database is removed.
```

Figure 3: Example interactive input and output while deleting table schemas.

- **Load data from a dataset**: After a user enters the path of the folder that contains the data files, the system reads all data files from the user-specified folder and inserts the records into the appropriate table in the database. (Your program can assume that the user-specified folder must contain all 5 data files. These 5 input files are named user_category.txt, user.txt, book_category.txt, book.txt and check_out.txt. Each data file stores the data corresponding to its filename.)

```
-----Operations for administrator menu-----
What kind of operation would you like to perform?
1. Create all tables
2. Delete all tables
3. Load from datafile
4. Show number of records in each table
5. Return to the main menu
Enter Your Choice: 3

Type in the Source Data Folder Path: sample_data
Processing...Done. Data is inputted to the database.
```

Figure 4: Example interactive input and output while loading table schemas from the database

- **Show the number of records in each table:** For each table in the database, display the number of records in it.

```
-----Operations for administrator menu-----
What kind of operation would you like to perform?
1. Create all tables
2. Delete all tables
3. Load from datafile
4. Show number of records in each table
5. Return to the main menu
Enter Your Choice: 4
Number of records in each table:
table1: xxx
table2: xxx
table3: xxx
table4: xxx
```

Figure 5: Example interactive input and output while showing number of records in each table.

Note: Please replace words (i.e., table1: xxx, table2: xxx, table3: xxx, table4: xxx) in Figure 5 with the tables in relational schema given in the suggested solution of phase 1. The number of tables may not be the same as shown in Figure 5.

5.2. Library User

- **Search for books**: The system is required to provide an interface to allow a library user to search for the books in the library in three different ways:
  - By call number (exact matching)
  - By title (partial matching)
  - By author (partial matching)

  You can assume that only one searching method can be selected by the library user for each query and the whole string entered by the library user is considered as one search word (e.g., When a user entered "Database Concept", The system will consider "Database Concept" as one and only one search keyword instead of two search keywords "Database" and "Concept"). After the library user entered the search keyword, the program should perform the query and return all matching books in terms of their *call number*, *title, book category name, authors, rating* and *number of available copies*. The results of the query should be sorted in ascending order of *call number* and outputted as a table as follows:

```
-----Main menu-----
What kinds of operations would you like to perform?
1. Operations for Administrator
2. Operations for Library User
3. Operations for Librarian
4. Exit this program
Enter Your Choice: 2

-----Operations for library user menu-----
What kind of operation would you like to perform?
1. Search for Books
2. Show loan record of a user
3. Return to the main menu
Enter Your Choice: 1
Choose the Search criterion:
1. call number
2. title
3. author
Choose the search criterion: 1
Type in the Search Keyword:G3632000
|Call Num|Title|Book Category|Author|Rating|Available No. of Copy|
|G3632000|Database system implementation|Database|Hector Garcia-Molina, Jeffrey D. Ullma
n, Jennifer Widom|8.5|5|
End of Query
```

Figure 6: Example input and output while searching for books

- **Show all check-out records of a library user:** The system is required to provide an interface to allow a library user to show all his/her check-out records of with a given *user ID*. After the library user enters his/her *user ID*, the program will perform the query and return all the matching check-out records in terms of *call number*, *copy number*, *title*, *authors*, *check out date* and *whether the book copy of the corresponding check-out record is returned*. The check-out records should be sorted in descending order of *check-out date* and outputted as a table as follows:

```
-----Operations for library user menu-----
What kind of operation would you like to perform?
1. Search for Books
2. Show loan record of a user
3. Return to the main menu
Enter Your Choice: 2
Enter The User ID: user000007
Loan Record:
|CallNum|CopyNum|Title|Author|Check-out|Returned?|
|S5583200|5|Operating system concepts|Abraham Silberschatz, Greg Gagne, Peter Baer Galvi
n|2021-03-21|Yes|
|D3184199|2|OO database system|Hiroshi Ishikawa|2021-02-09|Yes|
End of Query
```

Figure 7: Example input and output while showing all check-out records of a library user

5.3. Librarian

- **Borrow a book copy:** A librarian can perform the book borrowing procedure through the Library Inquiry System. First, he/she needs to input *call number* and *copy number* of the book copy being borrowed and the *user ID* of the library user. Then the system should check whether that book copy is available to be borrowed (i.e., There is no check out record of the specified book copy with NULL *return date*). If the book copy is available, it is then borrowed and a new check-out record of the specified book copy and user with NULL *return date* should be added to the database accordingly. Finally, there should be an informative message whether the book copy can be lent successfully in layman terms.

```
-----Main menu-----
What kinds of operations would you like to perform?
1. Operations for Administrator
2. Operations for Library User
3. Operations for Librarian
4. Exit this program
Enter Your Choice: 3

-----Operations for librarian menu-----
What kind of operation would you like to perform?
1. Book Borrowing
2. Book Returning
3. List all un-returned book copies which are checked-out within a period
4. Return to the main menu
Enter Your Choice: 1
Enter The User ID: user000001
Enter The Call Number: S5583200
Enter The Copy Number: 1
Book borrowing performed successfully.
```

Figure 8: Example input and output while a librarian processes a book borrowing request

- **Return a book copy:** A librarian can perform the book returning procedure through the Library Inquiry System. First, he/she needs to input *call number* and *copy number* of the book copy being borrowed and the *user ID* of the library user. Then the system should check if a check-out record corresponding to the specified *user ID*, *call number* and *copy number* exists. If such record is found, the book copy can be returned, and the *return date* of the check-out record found is updated to be the current date of

the database server. Besides, the *rating* of the book should be updated by the following formula, and the *number of times borrowed* should be increased by one.

$$new\ book\ rating = \frac{book\ rating \times number\ of\ times\ borrowed + user\ rating}{number\ of\ times\ borrowed + 1}$$

For example, assume there is a book that has the rating of 8.3 and has been borrowed by 3 times. If a user would like to rate it as 9, then the rating of the book is updated to $\frac{8.3\times3+9}{3+1} = 8.475$. Also, the number of times borrowed is increased to 4.

Finally, there should be an informative message whether the book copy can be returned successfully in layman terms. (For the sake of simplicity, you are not required to check whether the book is overdue or not)

```
-----Operations for librarian menu-----
What kind of operation would you like to perform?
1. Book Borrowing
2. Book Returning
3. List all un-returned book copies which are checked-out within a period
4. Return to the main menu
Enter Your Choice: 2
Enter The User ID: user000001
Enter The Call Number: 5S661996
Enter The Copy Number: 3
Enter Your Rating of the Book: 8
Book returning performed successfully.
```

Figure 9: Example input and output while a librarian processes a book returning request

- **List all un-returned book copies which are checked-out within a period:** The system is required to provide an interface to allow a librarian to list all un-returned book copies which are checked-out within a given period (e.g., from 20/03/2021 to 20/04/2021). After the librarian enters the period, the program will perform the query and return a list of all un-returned book copies in terms of *user ID, call number*, *copy number and check-out date* in descending order of *check-out date* within the inputted period *inclusively*.

```
-----Operations for librarian menu-----
What kind of operation would you like to perform?
1. Book Borrowing
2. Book Returning
3. List all un-returned book copies which are checked-out within a period
4. Return to the main menu
Enter Your Choice: 3
Type in the starting date [dd/mm/yyyy]: 20/03/2021
Type in the ending date [dd/mm/yyyy]: 20/04/2021
List of UnReturned Book:
|LibUID|CallNum|CopyNum|Checkout|
|user000006|QA762011|1|2021-03-26|
|user000020|QA76D3L5|4|2021-03-26|
|user000008|D3184199|1|2021-03-23|
|user000015|QA76D3L5|2|2021-03-20|
End of Query
```

Figure 10: Example input and output while listing out all unreturned books

5.4. Error Handling

If a run-time error occurs, the Library Inquiry System should output an information message in layman terms and in a new line as shown below. You are not required to handle all possible errors, just try some.

```
-----Operations for librarian menu-----
What kind of operation would you like to perform?
1. Book Borrowing
2. Book Returning
3. List all un-returned book copies which are checked-out within a period
4. Return to the main menu
Enter Your Choice: 2
Enter The User ID: user000010
Enter The Call Number: D3184199
Enter The Copy Number: 1
Enter Your Rating of the Book: 8
[Error]: An matching borrow record is not found. The book has not been borrowed yet.
```

Figure 11: Example input and output when an error occurs.

Please note that the outputs of examples in all figures are not model answers of the testing data.

6. Grading Policy

The marks are distributed as follows:

Phase	Content	Mark Distribution
1	ER-diagram	10%
	Relational schema (based on your ER-diagram)	10%
2	Java application	80%

- There will be a mark deduction if your application is terminated unexpectedly during the demonstration.
- You are not allowed to modify any source code during the demonstration.
- All members in the same group will receive the same marks for the project. In order to encourage every student to participate in the project, a question about this project may be asked in the final examination.

7. Demonstration

- Depending on pandemic situation, we will hold a face-to-face demonstration or online demonstration.
- All group members should attend the demonstration.
- The duration for the demonstration for each group is about 15 minutes.
- The Java application will be **complied** and **tested** in a Linux 64-bit machine of the CSE department.
- The dataset used in the demonstration may be different from the dataset provided for testing.

8. Submission Methods

8.1. Phase 1

- Submit a PDF file (one copy for each group) to the Blackboard system.
- The PDF file should include an ER diagram, a relational schema, the group number, the names, and the student IDs of all group members of your group.

8.2. Phase 2

- Submit a ZIP file (one copy for each group) to the Blackboard system. The ZIP file should include all your source codes and a README file (README.txt), which contains:
  - Your group number
  - The name and the student ID of each group member
  - Instructions on how to compile and run your system