

Group Project (version 0.2)

MIS 6326.001 – Data Management

Due Date: TBD

Description

The project involves the creation of a client-server application to be utilized by librarians (not book borrowers). The project can be performed by groups of up to 3 people or individually. All work, design, and coding must be done by members of the group (no outside help).

High level functionality supported by the application:

- User interface to perform all listed functions unless stated otherwise
- Load initial data (provided) using SQL Developer Import utility (no need to develop UI)
- Add authors
- Add Books
- Add borrowers
- Search books by various parameters: ISBN, title, author
- Check-out (loan) books
- Check-in books

Database and Tools

- Database: Oracle Database 12c
- User Interface: Oracle Application Express 18.1: <http://ec2-18-219-60-86.us-east-2.compute.amazonaws.com:8080/ords>
- ER Diagrams: MS Visio or other tool

Initial Load Files

There are four files provided for initial load in order to help with creating more “natural” feel to application data and in order to help practice important concepts:

- Ability to load various file formats into DB using SQL developer guided import interface
- Ability to understand “raw” data
- Ability to translate given data into normalized version and match application DB design

Files:

- Books file: Books information, such as ISBN, authors, title, etc.
- Book copies file: number of copies of each book per library location
- Borrowers file: borrowers information, such as name, SSN (dummy), address, etc.
- Library branches file: branches address information

Grading (points distribution might change)

You will be required to demonstrate your application on the AWS APEX workspace created for your group.

An online sign-up mechanism will be made available to reserve a specific time with the TA for evaluation and grading. As a courtesy to the TAs and those waiting behind you, please be on time for your scheduled slot and have your application already launched and ready to go. Each group will have 10-15 minutes to demonstrate their system and execute the test cases provided at grading time. Test cases will be designed to validate use cases from the stated requirements.

Functional Requirements

1) Graphical User Interface, Overall Design, Architecture [30 points]

All user interaction with the Library Circulation Desk DBMS (queries, updates, deletes, etc.) must be done from a graphical user interface. The initial import of the dataset is considered to be an administration task and can be performed by utilizing SQL Developer Import functionality while data normalization should be performed using SQL scripts. UI design will be judged primarily on function and usability, not look-and-feel. Deliverables:

- ER diagram
- Database objects create statements: create table, foreign keys, indexes, etc.
- UI

1) Data Load, Normalization, data generation [10 points]

1. Load dataset files provided with the project requirements utilizing SQL Developer guided interface. Pay attention to file format/delimiter, SQL Developer is able to recognize proper delimiter when selecting "text" in input file options.
2. Write SQL scripts to populate target application tables while normalizing the data loaded in #1.
3. Write SQL to generate:
 - At least 500 books check-outs for at least 200 different borrowers and 100 different books
 - At least 50 fines records for 20 different borrowers

2) Book Search and Availability [20 points]

Book search functionality should allow setting a specific library branch location context or performing it globally for all locations.

Using UI, provide a single search field to locate a book given any combination of ISBN, title, and/or Author(s). Your query should support substring matching. For example, search for “will” should return results whose title include “will” or “willing”, or whose author name contains “Will”, “Willy”, “William”, etc.

User will have an option to limit search to one specified by user field: ISBN, title, and/or Author(s).

Search should display the following information for each book in the result set:

- ISBN
- Book title
- All book author(s)
- Availability status at the currently selected branch (if branch is selected)

3) Book Loans [10 points]

Checking Out Books:

- Using UI, be able to check out a book at the current location, which will create a new record in the BOOK_LOANS table. The Date_out should be today’s date. The Due_date should be 14 days after the Date_out. A checkout requires selecting a combination of BOOK and a BORROWER. How you do this is your design decision. Date_in should be NULL at checkout time. The system should not allow checking-out book copy that is not available.
- Each BORROWER is permitted a maximum of 3 BOOK_LOANS. If a BORROWER already has 3 active BOOK_LOANS, then the checkout (i.e. create new BOOK_LOANS tuple) should fail and display an appropriate error message.
- BORROWERS are not permitted to check out a book if they have either an unpaid fine or a currently overdue book. The system should reject such attempts and display an appropriate error message.

Checking in (returning) Books:

- Using UI, being able to check in a book. Be able to locate BOOK_LOANS tuples by searching on any book information or BORROWER information. Once an active BOOK_LOANS record is located, provide a way of checking in the book, i.e. updating the Date_in value from NULL to the current date.
- If an overdue book is checked in, a FINES record should be created (see Requirement #5)

4) Borrower Management [10 points]

- Using UI, being able to create new a BORROWER record in the system.
- BORROWER Name, SSN, and Address attribute values are required to create a new account (i.e. values must be NOT NULL).
- Borrowers are allowed to possess exactly one library card. If a new borrower is attempted with a duplicate SSN, the system should reject the new BORROWER and return an appropriate error message.

5) Fines [10 points]

- Display FINES information for a specified BORROWER. You should provide a mechanism to display paid, unpaid, or both.
- Display all overdue books for a specified BORROWER, i.e. BOOK_LOANS whose Due_date has passed, but whose Date_in is NULL
- Fines may not be assessed (i.e. create a record in the FINES table) until a book is returned.
- Fines are assessed at a rate of \$0.25/day (twenty-five cents per day).
- Provide a mechanism for librarians to enter payment of fines (i.e. update a FINES record to set Paid attribute to be true.

6) Reports [10 points]

Design and implement at least 3 reports that will be initiated from UI. The reports can accept input parameters, for example branch or date.

- Each report will have information presented from at least 3 tables
- Each report will have aggregation and accumulation functionality

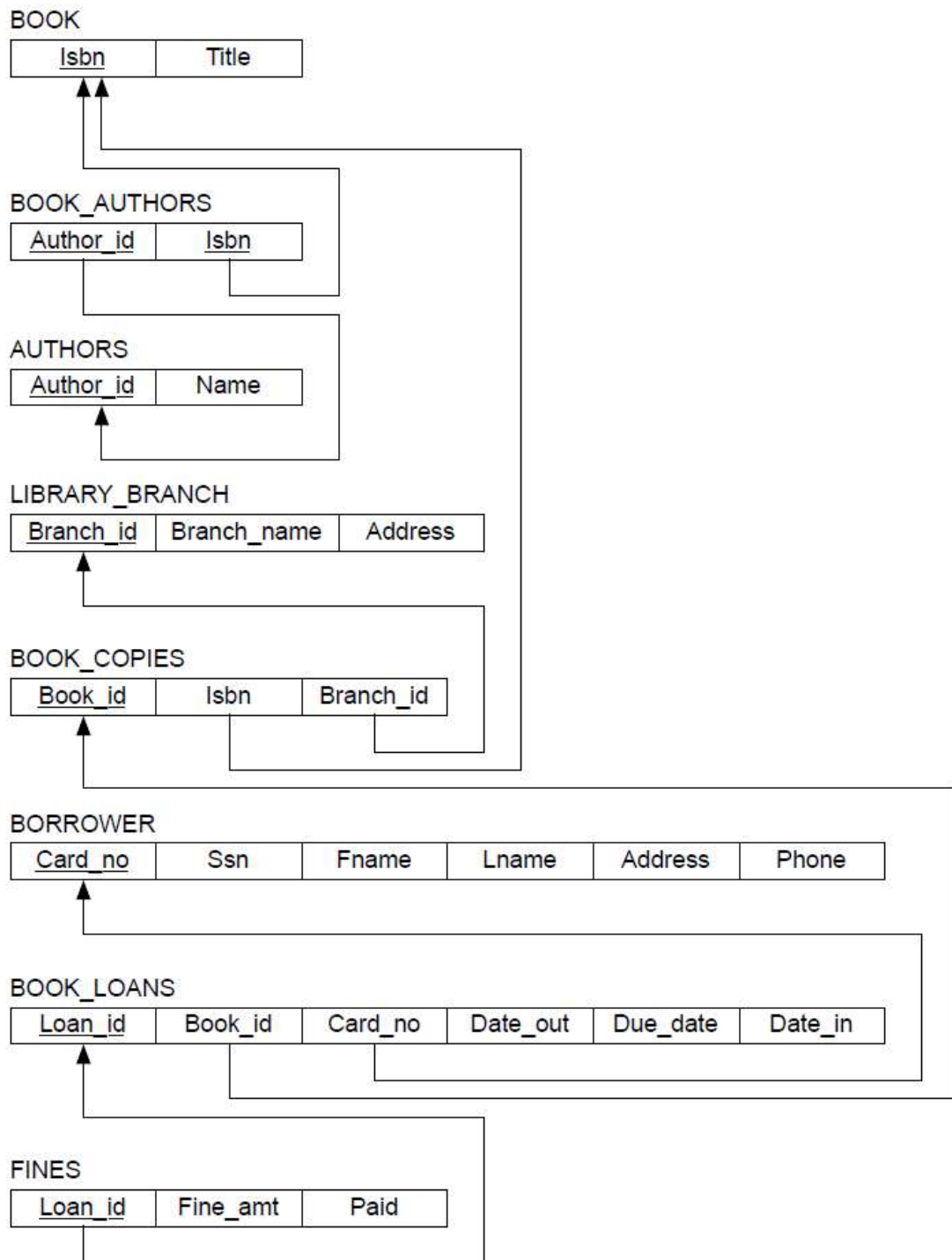
Examples of the reports:

- Top 10 most popular books in the last month
- Top 10 books that were checked in late

Schema

The schema for this project is provided below. You are permitted to modify or augment this schema provided that your system:

- backwards compatible with the given schema
- adheres to the written requirements



Data

- Baseline data to initialize your database is provided in the eLearning programming assignment folder. All data is provided in plain text comma or tab delimited format.
- Not all files may use the same CSV format. For example, some files may use comma delineation while others use tab delimiters. Some files may use quotes around strings, some may not.
- Note that there is not a one-to-one, file-to-table correspondence. Some of the data contains “noise”. Part of your system design task to clean and normalize these data, i.e. map it onto your schema and tables.
- You may use either ISBN-10 or ISBN-13 as the BOOK.ISBN primary key. Note that some ISBN-10 values may contain leading zeroes. These are part of the attribute value and should not be truncated!

Submission

You will be required to submit the following files for grading:

- A Design Document that contains a high-level description of your system architecture including system ER diagram, design decisions and assumptions. It should be 2-3 pages 12 point font text, not including any schema diagrams or system architecture figures. Clearly indicate any and all third party libraries and/or software used. Document format should be PDF.
- A Quick Start Guide (a brief User Manual) for librarian system users (1-2 pages). Document format should be PDF. Include URL and detailed info on how to access/start application.
- DB objects creation scripts
- Any scripts to clean or normalize data.
- All submission files must be zipped together into a single file. This file should be named <netid+netid+netid>_library.zip, where < netid+netid+netid > is list NETIDs of the team members. Example: cid021000+ cfx022000+dfv041000_library.zip