SQL Programming Project

CS-6360 Database Design

Document Owner : Ankit Raj

The primary contact for questions regarding this document is: Ankit Raj

Project Name : Library Management System

Phone: +1 (214)-718 -7920

Email: axr180133@utdallas.edu

1. Purpose of this document

Describe system architectures, design decisions and assumptions.

2. Architecture

Hardware Architecture:

It needs minimum 4 GB RAM to support server without performance issue. Apart from this, there is no special requirement.

Software Architecture:

It is a web-based application. Book Search is set as the home/default page. In other words, it is the entry point of the application. User can then navigate through different pages: check in, borrower management, Fine management through menu provided in all the pages. Each mentioned page is associated with different functionalities. Borrower management is to add a new user/borrower, fine management is to calculate fine and make payment.

Operating System: Supported on both 32 bit and 64-bit architecture.

Database Server: MySQL
Version: 8.0.18

Application Server: Django

Client: Web browser

Languages:

Database: All interaction with database is done using SQL (Structured Query Language)

Backend: Python. Application is based on MVC pattern (Model-view-controller).

Frontend: HTML 5, CSS 3, JavaScript and Jquery

3. Database

- All tables within schema are created according to the description provided in the project description.
- All key constraints like foreign keys, primary keys are implemented according to the project description.
- Tables are populated by importing data from books.csv and borrowers.csv files.
- Other constraints include:
    - A single user can issue only 3 books at a time.
    - Due date is 14 days after checkout date.
    - Check out date will be automatically populated based on date in local computer. Fines cannot be paid until book is checked in.

Assumptions:

1. Date is in YYYY-MM-DD format.
2. Check in date is set to null by default.
3. Borrower id or Card id is auto generated for a new user.
4. Loan id is auto generated whenever a book is loaned
5. There is an additional Availability attribute introduced in Book table which is defaulted to 1 during table creation to keep track of availability. 1- Available, 0- Unavailable.
6. Fine update mechanism is executed in the backend on button click.
7. Page is refreshed to display updated tuples.

4. Additional Features

- After search results are fetched from database, we can sort the table based on attributes (Alphabetically if string, otherwise based on numerical value).
- An addition search option is available to user to further filter out the table. It reduces load on database and improves searching experience.
- To handle situation where a search returns high number of tuples increasing the display table size and hence the web page, an option to set table length is provided to the user. This gives ability to users to adjust table size according to their convenience improving readability and user experience.

- If user tries to perform a conflicting action like checkout an unavailable book or pay fine without checking in book, appropriate user-friendly messages are displayed.

5. References

- https://www.w3schools.com/sql/
- https://www.geeksforgeeks.org/sql-tutorial/
- https://www.geeksforgeeks.org/mysql-group_concat-function/
- https://stackoverflow.com/questions/2469457/insert-on-duplicate-key-update-with-where
- https://dev.mysql.com/doc/refman/5.7/en/insert-on-duplicate.html
- https://stackoverflow.com/questions/35726910/bulk-update-mysql-with-where-statement
- https://www.geeksforgeeks.org/mysql-group_concat-function/
- https://docs.djangoproject.com/en/2.2/topics/db/sql/