

SQL Programming Project

CS-6360 Database Design

Description

1. Programming Language(s) and Frameworks

GUI interface:

I implemented a web interface with HTML.

My application GUI is programmed with Python and framework Flask.

As for my database, I used MySQL.

2. Data

Baseline data to initialize the database is provided in the eLearning programming assignment folder:

books.csv

borrowers.csv

library_branch.csv

book_copies.csv

book_loans.csv

3. Functions

a. GUI

Designed a website interface for the library management system with HTML.

Created the database system from command line: source create-library-data-for-MySQL.sql

Inserted the data from csv files by run insert.py and insert_book_loans_and_fine.py from command line

b. Book Search and Availability

Since we need to search any combination of ISBN, title, and any substring of author's name, I chose 'text' as input type in html and considered seven situations as following:

ISBN	Title	Author name
not empty	empty	empty
empty	not empty	empty
empty	empty	not empty
not empty	not empty	empty
not empty	empty	not empty
empty	not empty	not empty
not empty	not empty	not empty

In order to search substring of author's name, I splited the name string by ' ' and used LIKE in query. And in order to find how many copies are still available, I first counted the number of ISBN in BOOK_LOANS table and subtracted it from the total number of copies.

c. Book Loans

Check Out:

All three inputs, ISBN, branch id, and card no cannot be empty, otherwise, it would cause error.

Before checking out, I need to count the number of loan id with the input card no, and check if it is more than or equal to 3 and also need to ensure the available number of copies is not zero.

If all the check out conditions are satisfied, create a new tuple in BOOK_LOANS table.

Check in:

I need to consider several situations for the input as following:

book ISBN	card no	borrower name
not empty	empty	not empty
not empty	empty	empty
empty	not empty	not empty
empty	not empty	empty
not empty	not empty	not empty
not empty	not empty	empty

And I chose the same method to search borrower's name as the method I chose to search author's name.

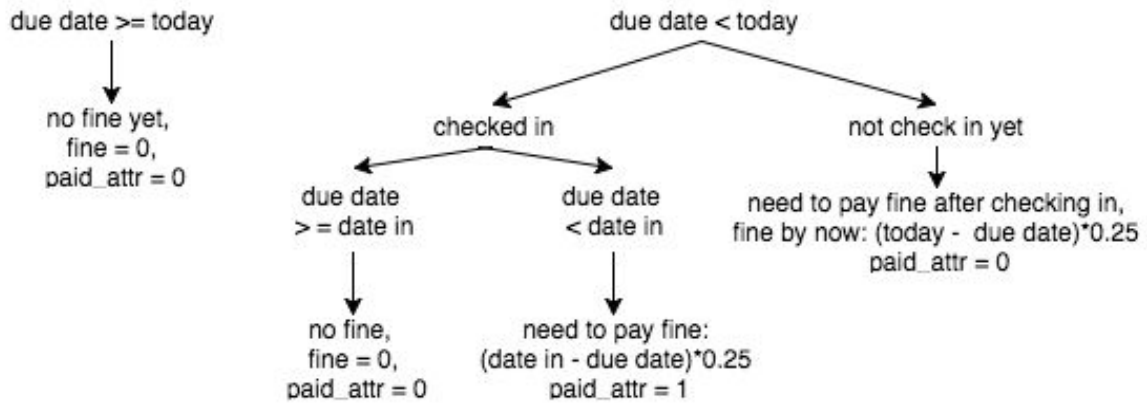
In order to check in, I added a radio button to each record shown in html and after checking in, I updated the date_in in BOOK_LOANS table for that loan_id.

When checking in, I need not only update the date_in value, but also need to update paid_attr from 0 to 1, if Est_amt != 0.

d. Borrower management

To sign up a new borrower, I need to check if the SSN has already existed; if not, I generate a new borrower id: "ID00" + total number of borrowers

e. Fine



When creating the Fine table, I added an attribute `paid_attr` to decide whether payment is available, that is, the loan book need to be paid and has been returned, just as shown above. Only returned book can be paid, fine record with `paid_attr = 0` cannot be paid. When update the fine record, in case that some books are returned after the last update, all book loan records need to be updated.