Simple Library Database Implementation

CMPE 321 - Project 3

September 14, 2020

Ramiz Dündar
2016400012

1 Introduction

This is a simple implementation of library database. Each book has ISBN, title and author. People can borrow books for 2 weeks and one person can borrow at most 8 books.

2 Constraints

- Each book has one copy
- TC is integer (TC $< 2^{32}$)
- ISBN is integer (ISBN $< 2^{32}$)

3 Relation definitions

- books (isbn INT PRIMARY KEY, title VARCHAR(255), author VARCHAR(255), borrowed BOOLEAN)
- borrows (tc INT, isbn INT, date VARCHAR(255))
- 1NF: books and borrows are in 1NF because all attributes are atomic(single valued)
- 2NF: books and borrows are in 2NF because there is no partial dependency between attributes.
- 3NF and BCNF: books and borrows are in BCNF beacause only isbn -> title and isbn -> author. Nothe that isbn is primary key.

4 SQL DML

4.1 Insert a book

```python
def insert(isbn, title, author):
    mycursor = mydb.cursor()

    borrowed = 0

    sql = "INSERT INTO books (isbn, title, author, borrowed) VALUES (%s, %s, %s, %s)"
    val = (isbn, title, author, borrowed)
    mycursor.execute(sql, val)

    mydb.commit()

    print("Added book:", val)
    return val
```

4.2 Delete a book

```python
def delete(isbn):
    mycursor = mydb.cursor()

    sql = "DELETE FROM books WHERE isbn = %s"
    val = (isbn, )

    mycursor.execute(sql, val)

    print("Deleted book:", val)

    mydb.commit()

    return val
```

4.3 Search a book

```python
def search(value, field):
    mycursor = mydb.cursor()

    sql = f"SELECT * FROM books WHERE {field} = %s"
    val = (value, )

    mycursor.execute(sql, val)

    myresult = mycursor.fetchall()

    return myresult
```

4.4 Search a person

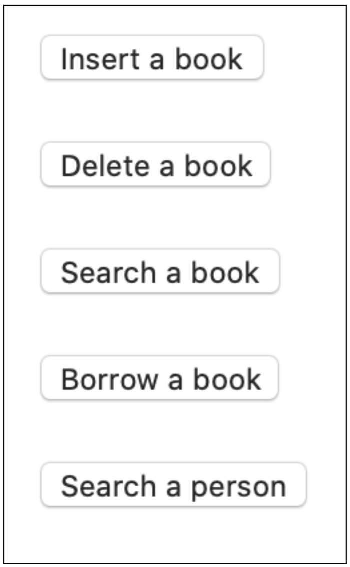
```python
def search_person(tc):

    mycursor = mydb.cursor()

    sql = "SELECT * FROM borrows WHERE tc = %s"
    val = (tc, )

    mycursor.execute(sql, val)

    myresult = mycursor.fetchall()

    return myresult
```

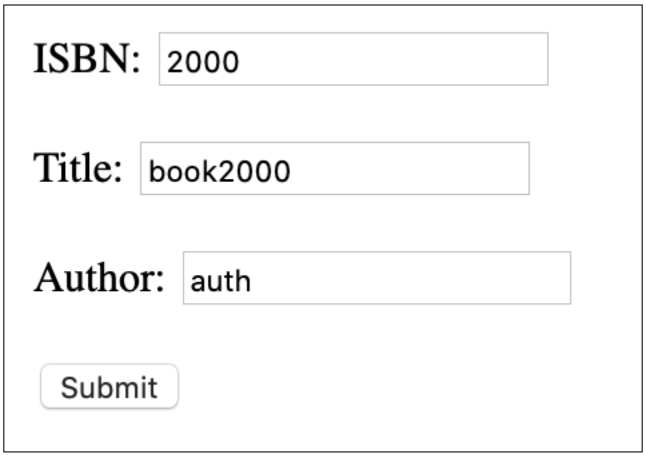
4.5 Borrow a book

```python
def borrow(tc, isbn):
    book = search(isbn, "isbn")

    if len(book) == 0:
        print("Book not found")
        return "Book not found"

    if book[0][3] == 1:
        print("Book is not available")
        return "Book is not available"

    person = search_person(tc)

    if len(person) == 8:
        print("You can't borrow more than 8 books")
        return "You can't borrow more than 8 books"

    mycursor = mydb.cursor()
    # Added this part to trigger
    # sql = "UPDATE books SET borrowed = %s WHERE isbn = %s"
    # val = (1, isbn)

    # mycursor.execute(sql, val)

    sql = "INSERT INTO borrows (tc, isbn, date) VALUES (%s, %s, %s)"
    val = (tc, isbn, (datetime.now()+timedelta(days=14)).strftime("%d-%m-%Y"))
    mycursor.execute(sql, val)

    mydb.commit()

    print(f"TC: {tc} borrowed ISBN: {isbn}")
    return f"TC: {tc} borrowed ISBN: {isbn}"
```

5 Screen shots of input and output

5.1 Insert a book

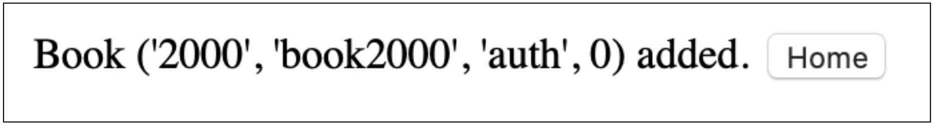




ISBN: 2000

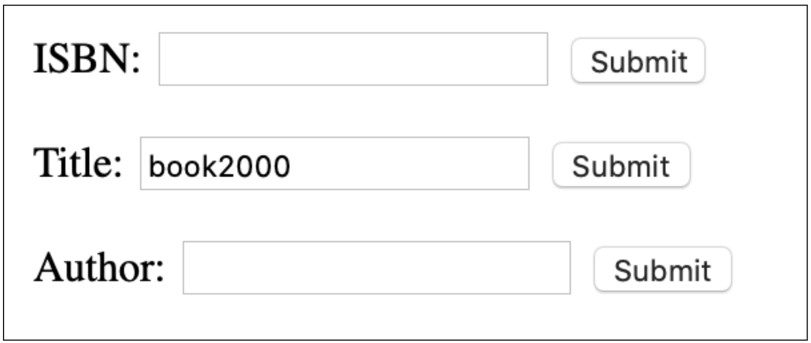
Title: book2000

Author: auth



Book ('2000', 'book2000', 'auth', 0) added.

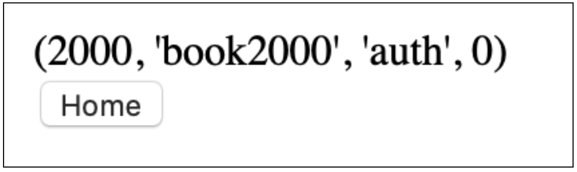
5.2 Search a book



ISBN:

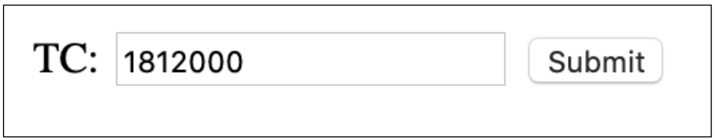
Title: book2000

Author:

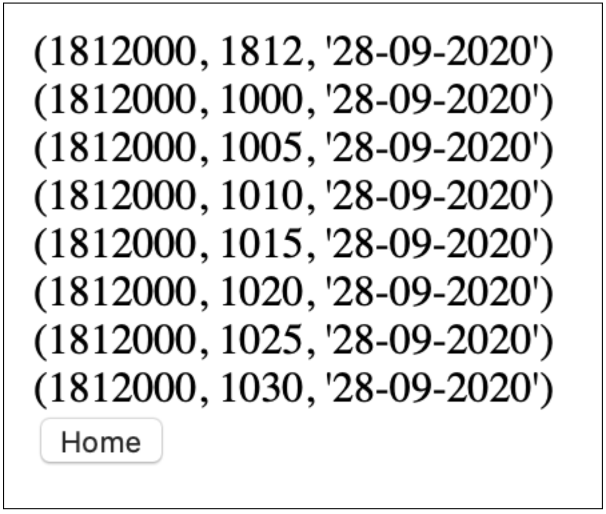


(2000, 'book2000', 'auth', 0)

5.3 Search a person



TC: 1812000



```
(1812000, 1812, '28-09-2020')
(1812000, 1000, '28-09-2020')
(1812000, 1005, '28-09-2020')
(1812000, 1010, '28-09-2020')
(1812000, 1015, '28-09-2020')
(1812000, 1020, '28-09-2020')
(1812000, 1025, '28-09-2020')
(1812000, 1030, '28-09-2020')
```

6 Conclusions

This simplistic implementation of library database features minimalistic web interface. Although it's features are limited it still provides the most essential needs for library databases. To improve further unique book limitations could be removed. Also returning a book should be added.