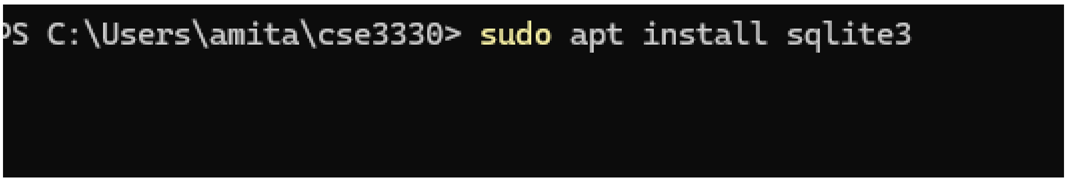
Readme

In order to use this program you will need to have the latest version of python installed within your system.
If it is not installed you can download the latest version here
https://www.python.org/downloads/
Ensure that tkinter is installed along with your python.

Installation Instructions

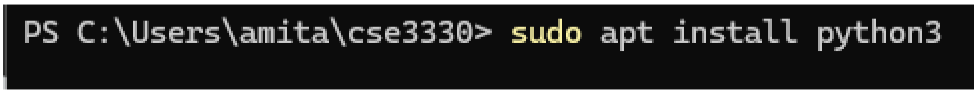
1.Download all files within the folder(Create tables, csv data, code.py)

2.Open your command console and open the directory to where the files were downloaded
3.Sudo apt install sqlite3



```
PS C:\Users\amita\cse3330> sudo apt install sqlite3
```

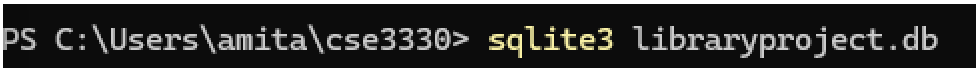
3. Sudo apt install python3



```
PS C:\Users\amita\cse3330> sudo apt install python3
```

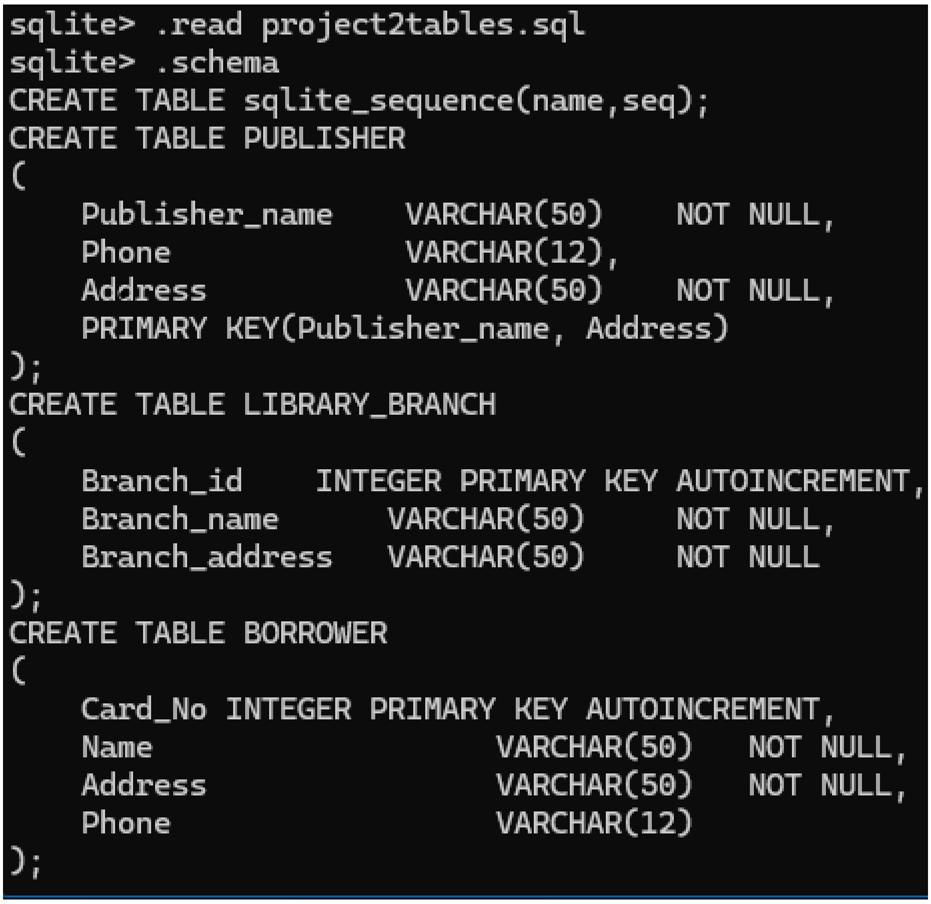
Run Instructions

1. Sqlite3 libraryproject.db



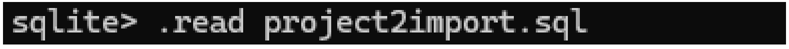
```
PS C:\Users\amita\cse3330> sqlite3 libraryproject.db
```

2. .read project2tabl.sql



```
sqlite> .read project2tables.sql
sqlite> .schema
CREATE TABLE sqlite_sequence(name,seq);
CREATE TABLE PUBLISHER
(
    Publisher_name     VARCHAR(50)    NOT NULL,
    Phone              VARCHAR(12),
    Address            VARCHAR(50)    NOT NULL,
    PRIMARY KEY(Publisher_name, Address)
);
CREATE TABLE LIBRARY_BRANCH
(
    Branch_id    INTEGER PRIMARY KEY AUTOINCREMENT,
    Branch_name       VARCHAR(50)     NOT NULL,
    Branch_address    VARCHAR(50)     NOT NULL
);
CREATE TABLE BORROWER
(
    Card_No INTEGER PRIMARY KEY AUTOINCREMENT,
    Name                       VARCHAR(50)   NOT NULL,
    Address                    VARCHAR(50)   NOT NULL,
    Phone                      VARCHAR(12)
);
```

3. read project2import.sql

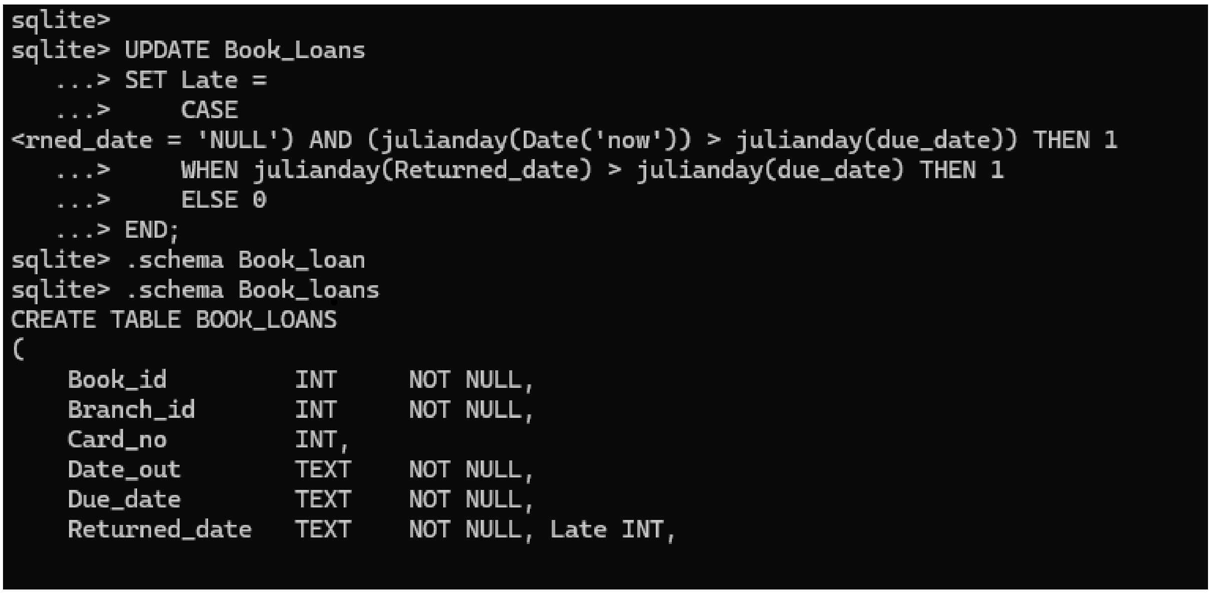


```
sqlite> .read project2import.sql
```

4. Add column Late to Book_Loan with query

```
ALTER TABLE Book_Loans
ADD Late INT;

UPDATE Book_Loans
SET Late =
    CASE
    WHEN (Returned_date IS NULL OR Returned_date = 'NULL') AND
(julianday(Date('now')) > julianday(due_date)) THEN 1
    WHEN julianday(Returned_date) > julianday(due_date) THEN 1
    ELSE 0
END;
```

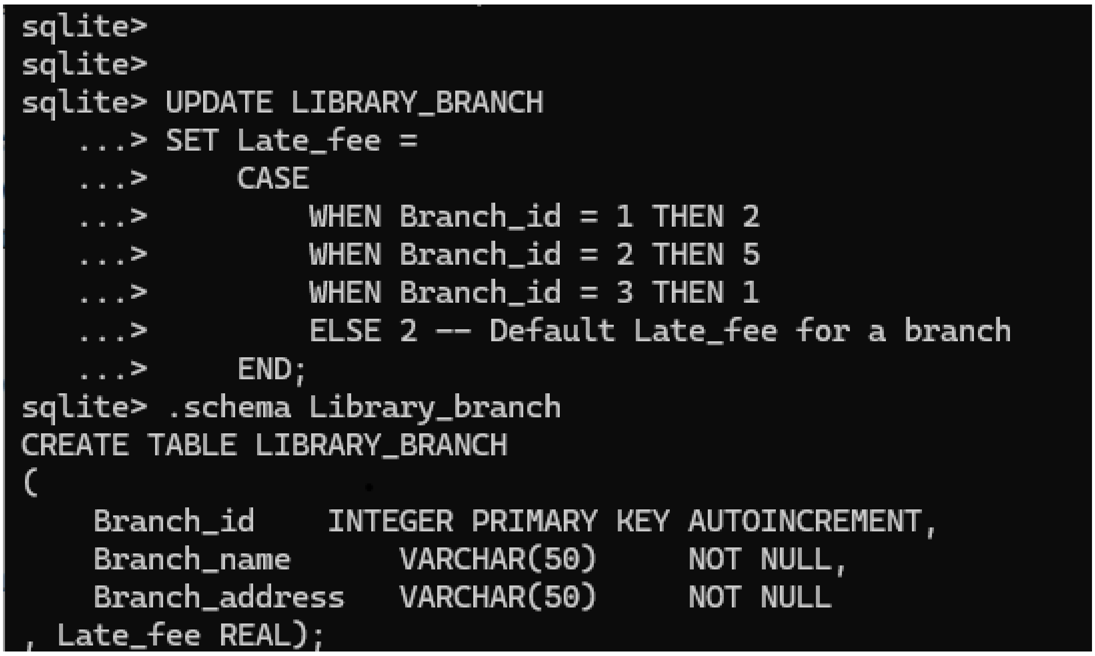


```
sqlite>
sqlite> UPDATE Book_Loans
   ...> SET Late =
   ...>     CASE
<rned_date = 'NULL') AND (julianday(Date('now')) > julianday(due_date)) THEN 1
   ...>     WHEN julianday(Returned_date) > julianday(due_date) THEN 1
   ...>     ELSE 0
   ...> END;
sqlite> .schema Book_loan
sqlite> .schema Book_loans
CREATE TABLE BOOK_LOANS
(
    Book_id         INT     NOT NULL,
    Branch_id       INT     NOT NULL,
    Card_no         INT,
    Date_out        TEXT    NOT NULL,
    Due_date        TEXT    NOT NULL,
    Returned_date   TEXT    NOT NULL, Late INT,
```

5. Add column LateFee to Library_Branch

```
ALTER Table LIBRARY_BRANCH
ADD Late_fee REAL;

UPDATE LIBRARY_BRANCH
SET Late_fee =
    CASE
        WHEN Branch_id = 1 THEN 2
        WHEN Branch_id = 2 THEN 5
        WHEN Branch_id = 3 THEN 1
        ELSE 2 -- Default Late_fee for a branch
    END;
```



```
sqlite>
sqlite>
sqlite> UPDATE LIBRARY_BRANCH
   ...> SET Late_fee =
   ...>     CASE
   ...>         WHEN Branch_id = 1 THEN 2
   ...>         WHEN Branch_id = 2 THEN 5
   ...>         WHEN Branch_id = 3 THEN 1
   ...>         ELSE 2 -- Default Late_fee for a branch
   ...>     END;
sqlite> .schema Library_branch
CREATE TABLE LIBRARY_BRANCH
(
    Branch_id     INTEGER PRIMARY KEY AUTOINCREMENT,
    Branch_name       VARCHAR(50)     NOT NULL,
    Branch_address    VARCHAR(50)     NOT NULL
, Late_fee REAL);
```

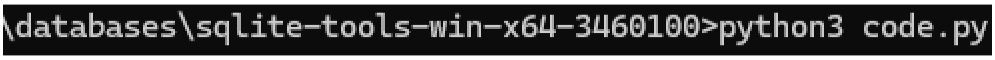
6.

Create the view BookLoanInfo for the database using .read create_view.sql or running the following code in the sqlite3 terminal

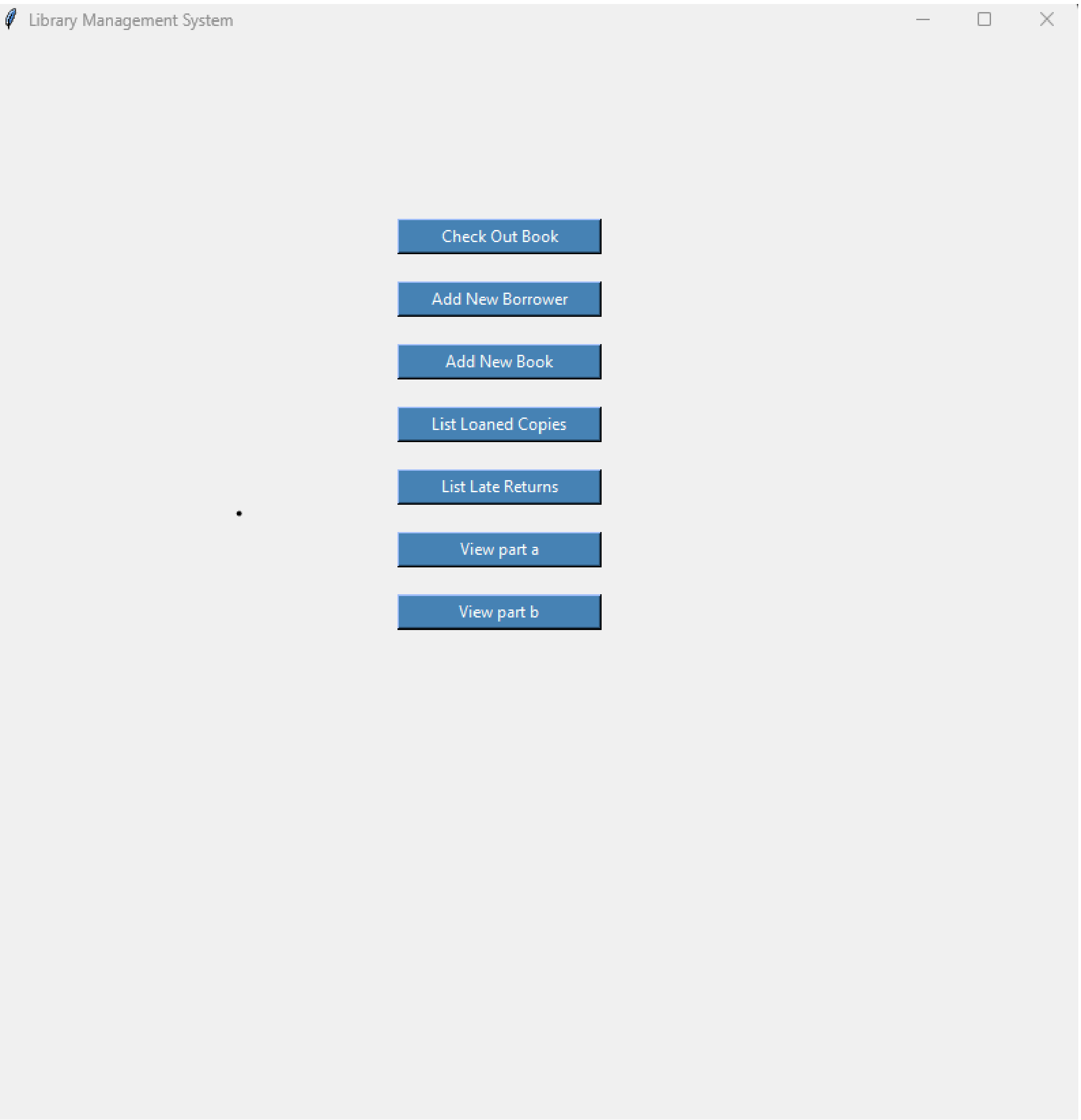
```
CREATE VIEW BookLoanInfo AS
SELECT BO.Card_No, BO.Name AS Borrower_name, BL.Date_out, BL.Due_Date,
BL.Returned_Date, B.Title AS Book_title,
(CASE
    WHEN (BL.Returned_date = 'NULL' OR BL.Returned_date IS NULL) THEN
julianday(Date('now')) - julianday(Date_out)
    WHEN Returned_date > Date_out
    THEN julianday(BL.Returned_date)- julianday(BL.Date_out)
    ELSE
    0
    END
) AS TotalDays,
(CASE
    WHEN (BL.Returned_date = 'NULL' OR BL.Returned_date IS NULL) AND
(julianday(Date('now')) > julianday(BL.Due_date)) THEN
(julianday(Date('now')) - julianday(BL.Due_date))
    WHEN
    julianday(BL.returned_date) > julianday(BL.due_date) THEN
julianday(BL.returned_date) - julianday(BL.Due_date)
    ELSE
    0
END) AS Late_days,
BL.Branch_id,
(   CASE
    WHEN (BL.Returned_date = 'NULL' OR BL.Returned_date IS NULL) AND
(julianday(Date('now')) > julianday(BL.Due_date)) THEN
(julianday(Date('now')) - julianday(BL.Due_date)) * LB.Late_fee
    WHEN
    julianday(BL.Returned_date) > julianday(BL.Due_date) THEN
    (julianday(BL.Returned_date) - julianday(BL.Due_date)) *
LB.Late_fee
    ELSE
    0
END) AS LateFeeBalance
FROM Library_Branch LB JOIN Book_Loans BL ON LB.Branch_id =
BL.Branch_id
JOIN Borrower BO ON BL.Card_no = BO.Card_no JOIN Book B ON BL.Book_id =
B.Book_id;
```

7. Run the python code through the terminal

```
Python3 code.py
```



```
\databases\sqlite-tools-win-x64-3460100>python3 code.py
```



Library Management System
Check Out Book
Add New Borrower
Add New Book
List Loaned Copies
List Late Returns
View part a
View part b