**README.md**

**CSE 3330 – 004 (Project 2 – Part 3)**

**GROUP 14 (Araohat Kokate, Inshaad Merchant , Athrva Arora)**

**HONOR CODE-**

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code

1. **TOOLS USED FOR THE PROJECT:**

* Tkinter for GUI
* Python
* Command Prompt
* Sqlite3
* Gridview for the GUI

**STEPS TO RUN THE LIBRARY MANAGEMENT SYSTEM –**

1. Move LMS.sql and all the provided Dataset in Project Part 2 and Code.py in the same directory-
2. Open the Command Prompt and open the sqlite3 shell and create and open the lms.db and fill it using the command “.read LMS.sql”

A computer screen with white text

Description automatically generated

1. It is important to understand that we need to make sure that our database is updated with all the changes in Project 2 up until the queries in Project2 Part3 Task1
2. Source Code for
3. LMS.sql

.read createTables.sql

.read import.sql

.read query1.sql

.read query2.sql

.read query3.sql

.read query4a.sql

.read query4b.sql

.read query5.sql

.read query6.sql

.read query7.sql

.read query8.sql

.read query9.sql

.read query10.sql

.read part3.sql

.read part3q2.sql

.read part3q3.sql

1. createTables.sql

DROP TABLE IF EXISTS LIBRARY\_BRANCH;

CREATE TABLE LIBRARY\_BRANCH(

        Branch\_id           INTEGER             PRIMARY KEY AUTOINCREMENT,

        Branch\_name         VARCHAR(20)         NOT NULL,

        Address             TEXT                NOT NULL

        );

DROP TABLE IF EXISTS PUBLISHER;

CREATE TABLE PUBLISHER(

        Publisher\_name      CHAR(20)            NOT NULL,

        Phone\_no            VARCHAR(13)         NOT NULL,

        Address             TEXT                NOT NULL,

        PRIMARY KEY (Publisher\_name));

DROP TABLE IF EXISTS BOOK;

CREATE TABLE BOOK(

        Book\_id             INTEGER             PRIMARY KEY AUTOINCREMENT,

        Title               VARCHAR(20)         NOT NULL,

        Publisher\_name      VARCHAR(50)         NOT NULL,

        FOREIGN KEY (Publisher\_name) REFERENCES PUBLISHER(Publisher\_name)

        ON UPDATE CASCADE

        ON DELETE CASCADE);

DROP TABLE IF EXISTS BOOK\_AUTHORS;

CREATE TABLE BOOK\_AUTHORS(

        Book\_id             INTEGER             PRIMARY KEY AUTOINCREMENT,

        Author\_name         VARCHAR(50)         NOT NULL

        );

DROP TABLE IF EXISTS BOOK\_COPIES;

CREATE TABLE BOOK\_COPIES(

        Book\_id             INT                 NOT NULL,

        Branch\_id           INT                 NOT NULL,

        No\_of\_Copies        INT                 NOT NULL,

        FOREIGN KEY (Book\_id) REFERENCES BOOK(Book\_id)

        ON UPDATE CASCADE

        ON DELETE CASCADE,

        FOREIGN KEY (Branch\_id) REFERENCES LIBRARY\_BRANCH(Branch\_id)

        ON UPDATE CASCADE

        ON DELETE CASCADE);

DROP TABLE IF EXISTS BORROWER;

CREATE TABLE BORROWER(

        Card\_no             INTEGER             PRIMARY KEY AUTOINCREMENT,

        Name                VARCHAR(30)         NOT NULL,

        Address             TEXT                NOT NULL,

        Phone               VARCHAR(13)         NOT NULL

        );

DROP TABLE IF EXISTS BOOK\_LOANS;

CREATE TABLE BOOK\_LOANS(

        Book\_id             INT                 NOT NULL,

        Branch\_id           INT                 NOT NULL,

        Card\_no             INT                 NOT NULL,

        Date\_out            DATE                NOT NULL,

        Due\_Date            DATE                NOT NULL,

        Returned\_date       DATE                NULL,

        FOREIGN KEY (Book\_id)   REFERENCES BOOK(Book\_id)

        ON UPDATE CASCADE

        ON DELETE CASCADE,

        FOREIGN KEY (Branch\_id) REFERENCES LIBRARY\_BRANCH(Branch\_id)

        ON UPDATE CASCADE

        ON DELETE CASCADE,

        FOREIGN KEY (Card\_no)   REFERENCES BORROWER(Card\_no)

        ON UPDATE CASCADE

        ON DELETE CASCADE);

1. import.sql

.mode csv

.import --skip 1 Book.csv BOOK

.import --skip 1 Book\_Authors.csv BOOK\_AUTHORS

.import --skip 1 Book\_Copies.csv BOOK\_COPIES

.import --skip 1 Book\_Loans.csv BOOK\_LOANS

.import --skip 1 Borrower.csv BORROWER

.import --skip 1 Library\_Branch.csv LIBRARY\_BRANCH

.import --skip 1 Publisher.csv PUBLISHER

1. query1.sql

INSERT INTO BORROWER(Name, Address, Phone)

VALUES('Araohat Kokate','587 Spaniolo Dr, Arlington TX, 76010','682-340-0275');

1. query2.sql

UPDATE BORROWER

SET Phone = '837-721-8965'

WHERE Name like 'Araohat Kokate';

1. query3.sql

UPDATE BOOK\_COPIES

SET No\_of\_Copies = No\_of\_Copies + 1

WHERE Branch\_id = (SELECT LB.Branch\_id FROM LIBRARY\_BRANCH AS LB WHERE LB.Branch\_name = 'East Branch');

1. query4a.sql

INSERT INTO BOOK(Title, Publisher\_name)

VALUES('Harry Potter and the Sorcerer"s Stone','Oxford Publishing');

INSERT INTO BOOK\_AUTHORS(Author\_Name)

VALUES('J.K.Rowling');

1. query4b.sql

INSERT INTO LIBRARY\_BRANCH(Branch\_name, Address)

VALUES('North Branch','456 NW, Irving, TX 76100');

INSERT INTO LIBRARY\_BRANCH(Branch\_name, Address)

VALUES('UTA Branch','123 Cooper St, Arlington TX 76101');

1. query5.sql

SELECT B.Title, LB.Branch\_Name,

CAST(JULIANDAY(BL.Returned\_date) AS INTEGER) - CAST(JULIANDAY(BL.Date\_out) AS INTEGER) AS Borrowed\_Days

FROM    BOOK AS B, LIBRARY\_BRANCH  AS LB, BOOK\_LOANS AS BL

WHERE   B.Book\_Id = BL.Book\_Id AND

        BL.Branch\_Id = LB.Branch\_Id AND

        BL.Date\_Out BETWEEN '2022-03-05' AND '2022-03-23' AND

        BL.Returned\_date IS NOT NULL;

1. query6.sql

SELECT B.Name

FROM   BORROWER AS B

JOIN   BOOK\_LOANS AS BL ON BL.Returned\_date = 'NULL'

WHERE  B.Card\_no = BL.Card\_no;

1. query7.sql

SELECT LIBRARY\_BRANCH.Branch\_Name,

    COUNT(BOOK\_LOANS.Book\_id) AS Num\_of\_Book\_Borrowed,

    COUNT(CASE WHEN DATE('now') >= BOOK\_LOANS.Returned\_date THEN BOOK\_LOANS.Book\_id END) AS

    Num\_of\_Books\_Returned,

    COUNT(CASE WHEN BOOK\_LOANS.Returned\_date IS NULL OR BOOK\_LOANS.Returned\_date = 'NULL' THEN

    BOOK\_LOANS.Book\_id END) AS Num\_of\_Books\_Still\_Borrowed,

    COUNT(CASE WHEN BOOK\_LOANS.Returned\_date > BOOK\_LOANS.Due\_Date AND

    BOOK\_LOANS.Returned\_date IS NOT 'NULL' THEN BOOK\_LOANS.Book\_id END) AS

    Num\_of\_Late\_Returns

FROM LIBRARY\_BRANCH, BOOK\_LOANS

WHERE LIBRARY\_BRANCH.Branch\_Id = BOOK\_LOANS.Branch\_Id

GROUP BY LIBRARY\_BRANCH.Branch\_Name;

1. query8.sql

SELECT B.Title,

    IFNULL(CAST(JULIANDAY(BL.Returned\_date) AS INTEGER) - CAST(JULIANDAY(BL.Date\_out) AS INTEGER), 0) AS Borrowed\_Days

FROM BOOK AS B

JOIN BOOK\_LOANS AS BL ON B.Book\_Id = BL.Book\_Id;

1. query9.sql

SELECT BOOK.Title, BOOK\_AUTHORS.Author\_Name, CAST(JULIANDAY(Returned\_date) AS INTEGER) - CAST(JULIANDAY(Date\_out) AS INTEGER) as Days\_Borrowed,

CASE WHEN BOOK\_LOANS.Returned\_date IS NULL THEN 'YES' WHEN CAST(JULIANDAY(Returned\_date) AS INTEGER) - CAST(JULIANDAY(Due\_Date)AS INTEGER) > 0 THEN 'YES' ELSE 'NO'

END AS Late

FROM  BORROWER ,BOOK , BOOK\_AUTHORS

JOIN  BOOK\_LOANS ON BORROWER.Card\_no=BOOK\_LOANS.Card\_No

WHERE BORROWER.Name = 'Ethan Martinez' AND BOOK\_LOANS.Book\_id=BOOK.Book\_id AND BOOK.Book\_id=BOOK\_AUTHORS.Book\_Id

ORDER BY BOOK\_LOANS.Date\_Out;

1. query10.sql

SELECT B.Name, B.Address

FROM BORROWER AS B, LIBRARY\_BRANCH AS LB,  BOOK\_LOANS AS BL

WHERE BL.branch\_id = LB.branch\_id AND B.card\_no = BL.card\_no AND LB.branch\_name = 'West Branch';

1. part3.sql

ALTER TABLE BOOK\_LOANS

ADD COLUMN Late INT;

UPDATE BOOK\_LOANS

SET Late = CASE

    WHEN Returned\_Date IS NULL OR Due\_Date IS NULL THEN NULL

    WHEN Returned\_Date > Due\_Date THEN 1

    WHEN Returned\_Date > Due\_Date THEN 1

    ELSE 0

END;

1. part3q2.sql

ALTER TABLE LIBRARY\_BRANCH

ADD COLUMN LateFee INT;

UPDATE LIBRARY\_BRANCH

SET LateFee = CASE

    WHEN Branch\_id = 1 THEN 1

    WHEN Branch\_id = 2 THEN 2

    WHEN Branch\_id = 3 THEN 3

    WHEN Branch\_id = 4 THEN 4

    WHEN Branch\_id = 5 THEN 5

    ELSE 0

END;

1. part3q3.sql

-- Drop the existing view

DROP VIEW IF EXISTS vBookLoanInfo;

-- Recreate the view with the added column Book\_id

CREATE VIEW vBookLoanInfo AS

SELECT

    bl.Card\_no,

    b.Name AS "Borrower Name",

    bl.Date\_out,

    bl.Due\_Date,

    bl.Returned\_date,

    CAST((julianday(bl.Returned\_date) - julianday(bl.Date\_out)) AS INTEGER) AS TotalDays,

    bo.Title AS "Book Title",

    ba.Book\_id, -- Added column Book\_id

    CASE

        WHEN bl.Returned\_date <= bl.Due\_Date THEN 0

        ELSE CAST((julianday(bl.Returned\_date) - julianday(bl.Due\_Date)) AS INTEGER)

    END AS "Number of days returned late",

    bl.Branch\_id,

    CASE

        WHEN bl.Returned\_date <= bl.Due\_Date THEN 0

        ELSE CAST((julianday(bl.Returned\_date) - julianday(bl.Due\_Date)) AS INTEGER) \* lb.LateFee

    END AS LateFeeBalance

FROM

    BOOK\_LOANS bl

JOIN BORROWER b ON bl.Card\_no = b.Card\_no

JOIN BOOK bo ON bl.Book\_id = bo.Book\_id

JOIN BOOK\_AUTHORS ba ON bo.Book\_id = ba.Book\_id -- Joining BOOK\_AUTHORS to get Book\_id

JOIN LIBRARY\_BRANCH lb ON bl.Branch\_id = lb.Branch\_id;

1. Now the database lms.db have been filled and we can run our GUI application which will access data from lms.db directly.
2. To Compile and run the code in CMD using the command “python gui.py”



1. Main Window will pop up, then you can click on query to perform it.

A screenshot of a computer

Description automatically generated