

Project 2 Phase 3

Edosa Aigbuza & Brian Shamayev

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.

CONTRIBUTIONS

Edosa Aigbuza – Query Completion, Python script, GUI management

Brian Shamayev – Query Completion, Python script, GUI management

Task 1: Execute the following queries on the LMS database tables

Query 1:

```
ALTER TABLE BOOK_LOANS ADD Late int;  
UPDATE BOOK_LOANS  
SET Late = 1  
WHERE Returned_date > Due_Date;  
UPDATE BOOK_LOANS  
SET Late = 0  
WHERE Returned_date <= Due_Date;
```

Query 2:

```
ALTER TABLE LIBRARY_BRANCH ADD LateFee float;  
UPDATE LIBRARY_BRANCH  
SET LateFee = 1.50  
WHERE Branch_Id = 1;  
UPDATE LIBRARY_BRANCH  
SET LateFee = 2.50  
WHERE Branch_Id = 2;  
UPDATE LIBRARY_BRANCH  
SET LateFee = 3.00  
WHERE Branch_Id = 3;  
UPDATE LIBRARY_BRANCH  
SET LateFee = 4.50  
WHERE Branch_Id = 4;  
UPDATE LIBRARY_BRANCH  
SET LateFee = 5.00  
WHERE Branch_Id = 5;
```

Query 3:

```
CREATE VIEW vBookLoanInfo
AS SELECT BOR.Card_No, BOR.Name AS Borrower_Name, BL.Date_Out, BL.Due_Date,
BL.Returned_date,
CASE
    WHEN BL.Returned_date IS NOT NULL AND BL.Returned_date != ' NULL'
        THEN (julianday(BL.Returned_date) - julianday(BL.Date_Out))
    ELSE (julianday('now') - julianday(BL.Date_out))
END AS TotalDays,
B.Title AS Book_Title,
CASE
    WHEN BL.Returned_date IS NOT NULL AND BL.Returned_date != 'NULL' AND
julianday(BL.Returned_date) > julianday(BL.Due_Date)
        THEN (julianday(BL.Returned_date) - julianday(BL.Due_Date))
    ELSE 0
END AS Days_Late,
BL.Branch_Id,
CASE
    WHEN BL.Returned_date IS NOT NULL AND BL.Returned_date != 'NULL' AND
julianday(BL.Returned_date) > julianday(BL.Due_Date)
        THEN (julianday(BL.Returned_Date) - julianday(BL.Due_Date)) * LB.LateFee
    ELSE 0
END AS LateFeeBalance

FROM BOOK_LOANS BL
JOIN BOOK B ON BL.Book_Id = B.Book_Id
JOIN BORROWER BOR ON BL.Card_No = BOR.Card_No
JOIN LIBRARY_BRANCH LB ON BL.Branch_Id = LB.Branch_Id;

SELECT * FROM vBookLoanInfo;
```

sqlite> SELECT * FROM vbookLoanInfo;									
Card_No	Borrower_Name	Date_Out	Due_Date	Returned_date	TotalDays	Book_Title	Days_Late	Branch_Id	LateFeeBalance
123456	John Smith	2022-01-01	2022-02-01	2022-02-01	31.0	To Kill a Mockingbird	0	1	0
789012	Jane Doe	2022-01-02	2022-02-02		1214.13740148162	1984	0	1	0
345678	Bob Johnson	2022-01-03	2022-02-03		1213.13740148162	Pride and Prejudice	0	2	0
901234	Sarah Kim	2022-01-04	2022-02-04	2022-02-04	31.0	The Great Gatsby	0	3	0
567890	Tom Lee	2022-01-05	2022-02-05	2022-02-09	35.0	One Hundred Years of Solitude	4.0	1	6.0
234567	Emily Lee	2022-01-06	2022-02-06	2022-02-10	35.0	Animal Farm	4.0	2	10.0
890123	Michael Park	2022-01-07	2022-02-07	2022-03-08	60.0	The Catcher in the Rye	29.0	2	72.5
456789	Laura Chen	2022-01-08	2022-02-08	2022-03-10	61.0	Lord of the Flies	30.0	3	90.0
111111	Alex Kim	2022-01-09	2022-02-09	2022-02-06	28.0	Brave New World	0	1	0
222222	Rachel Lee	2022-01-10	2022-02-10	2022-02-07	28.0	The Picture of Dorian Gray	0	2	0
333333	William Johnson	2022-03-01	2022-03-08	2022-03-08	7.0	The Alchemist	0	1	0
444444	Ethan Martinez	2022-03-03	2022-03-10	2022-03-10	7.0	The God of Small Things	0	3	0
555555	Grace Hernandez	2022-02-03	2022-03-03	2022-02-18	15.0	Wuthering Heights	0	3	0
666666	Sophia Park	2022-01-14	2022-02-14	2022-03-31	76.0	The Hobbit	45.0	1	67.5
676767	Olivia Lee	2022-01-15	2022-02-15	2022-02-21	37.0	The Lord of the Rings	6.0	3	18.0
787878	Noah Thompson	2022-03-05	2022-03-12	2022-03-24	19.0	The Hitchhiker's Guide to the Galaxy	12.0	2	30.0
989898	Olivia Smith	2022-03-23	2022-03-30	2022-03-30	7.0	The Diary of a Young Girl	0	3	0
121212	Chloe Park	2022-01-18	2022-02-18	2022-02-18	31.0	The Da Vinci Code	0	3	0
232323	William Chen	2022-03-24	2022-03-31	2022-03-31	7.0	The Adventures of Huckleberry Finn	0	1	0
343434	Olivia Johnson	2022-01-21	2022-02-21	2022-02-21	31.0	The Adventures of Tom Sawyer	0	3	0
454545	Dylan Kim	2022-01-24	2022-02-24	2022-02-24	31.0	A Tale of Two Cities	0	3	0
555555	Grace Hernandez	2025-04-06	2025-05-06		24.1374014816247	To Kill a Mockingbird	0	1	0

Task 2: Create a GUI for the LMS database

Requirement 1:

```
INSERT INTO Book_Loans VALUES(:Book_Id, :Branch_Id, :Card_No, :Date_Out, :Due_Date,
:Returned_date, :Late)
```

```
        """ , {
            'Book_Id' : int(book_Id.get()),
            'Branch_Id' : int(branch_Id.get()),
            'Card_No' : int(card_number.get()),
            'Date_Out' : date_out.get(),
            'Due_Date' : due_date.get(),
            'Returned_date': None,
            'Late': None,
        })
    submit_cur.execute("""
        SELECT * FROM Book_Copies
        WHERE Book_Id = ? AND Branch_Id = ?
    """, (
        book_Id.get(),
        branch_Id.get()
    ))
```

```
CREATE TRIGGER IF NOT EXISTS update_book_copies
```

```
    AFTER INSERT ON BOOK_LOANS
```

```
    FOR EACH ROW
```

```
    BEGIN
```

```
        UPDATE BOOK_COPIES
```

```
        SET No_Of_Copies = No_Of_Copies - 1
```

```
        WHERE Book_Id = NEW.Book_Id AND Branch_Id = NEW.Branch_Id;
```

```
    END;
```

The screenshot shows a GUI window titled "Library Database". It contains a form with the following fields and values:

Card Number:	222222
Name:	
Address:	
Phone:	
BookID:	1
Book Title:	
Branch_Id:	1
Date Out:	2025-04-06
Due Date:	2025-05-06
Due Date Start:	
Due Date End:	
Publisher:	
Author:	
# of Copies:	

Below the form are several buttons: "Checkout Book", "Add New Borrower", "Add Books", "Find Branch Copies", "Days Late", "Borrower Late Fees", and "Book Information".

At the bottom, there is a "Book Checkout Report" section with the text: "Book ID: 1, Branch ID: 1, Copies Left: 1".

Requirement 2:

```
submit_cur.execute("INSERT INTO BORROWER(Name, Address, Phone) VALUES(:name,  
:address, :phone) ",
```

```
{  
    #'card_number': card_number.get(),  
    'name': name.get(),  
    'address': address.get(),  
    'phone': phone.get(),  
})
```

```
submit_cur.execute("SELECT Card_No FROM BORROWER WHERE Name = ? AND  
Address = ? AND Phone = ?",(name.get(), address.get(), phone.get(),))
```

The screenshot shows a window titled "Library Database" with a sidebar menu on the left and a main content area. The sidebar menu includes the following items: "Checkout Book", "Add New Borrower", "Add Books", "Find Branch Copies", "Days Late", "Borrower Late Fees", and "Book Information". The "Add New Borrower" item is currently selected. The main content area contains a form with the following fields: "Card Number:", "Name:" (with the value "New Borrower"), "Address:" (with the value "1345 Random St"), "Phone:" (with the value "222-555-9512"), "BookID:", "Book Title:", "Branch_Id:", "Date Out:", "Due Date:", "Due Date Start:", "Due Date End:", "Publisher:", "Author:", and "# of Copies:". Below the form, there is a button labeled "New Borrower ID" which displays the value "989900".

Requirement 3:

```
submit_cur.execute("INSERT INTO Book(Title, Publisher_Name) VALUES(:Title,
:Publisher_Name) ",
{
    #'Book_Id': book_Id.get(),
    'Title' : booktitle.get(),
    'Publisher_Name' :publisher.get(),
})

new_book_id = submit_cur.lastrowid

submit_cur.execute("INSERT INTO Book_Authors(Author_Name)
VALUES(:Author_Name)",
{
    #'Book_Id' : book_Id.get(),
    'Author_Name' : author.get(),
})

# Add to all 5 branches with 5 copies
for branch_id in range(1, 6):
    submit_cur.execute("INSERT INTO Book_Copies VALUES(:Book_Id, :Branch_Id,
:NO_Of_Copies)",
    {
        'Book_Id': new_book_id,
        'Branch_Id': branch_id,
        'No_Of_Copies': 5,
    })
```

The screenshot shows a window titled "Library Database" with a sidebar on the left containing several buttons: "Checkout Book", "Add New Borrower", "Add Books", "Find Branch Copies", "Days Late", "Borrower Late Fees", and "Book Information". The main area of the window contains a form with the following fields and values:

Card Number:	
Name:	
Address:	
Phone:	
BookID:	
Book Title:	New Book
Branch_Id:	
Date Out:	
Due Date:	
Due Date Start:	
Due Date End:	
Publisher:	New Publisher
Author:	New Author
# of Copies:	1

Requirement 4:

```
iq_cur.execute("""
```

```
    SELECT B.Branch_Id, COUNT(*) AS Copies_Loaned
```

```
    FROM Book_Loans BL
```

```
    JOIN Book Bk ON BL.Book_Id = Bk.Book_Id
```

```
    JOIN Library_Branch B ON BL.Branch_Id = B.Branch_Id
```

```
    WHERE Bk.Title = ?
```

```
    GROUP BY B.Branch_Id
```

```
""", (booktitle.get(),))
```

The screenshot shows a window titled "Library Database" with a sidebar of buttons and a main content area. The sidebar buttons are: "Checkout Book", "Add New Borrower", "Add Books", "Find Branch Copies", "Days Late", "Borrower Late Fees", and "Book Information". The main content area contains a form with the following fields: "Card Number:", "Name:", "Address:", "Phone:", "BookID:", "Book Title:" (with the text "To Kill a Mockingbird" entered), "Branch_Id:", "Date Out:", "Due Date:", "Due Date Start:", "Due Date End:", "Publisher:", "Author:", and "# of Copies:". Below the form is a button labeled "Checkout Book". At the bottom of the window, there is a box titled "Copies Loaned Report" containing the text "Branch ID 1: 2 copies loaned out".

Requirement 5:

```
dl_cur.execute("""  
    SELECT  
    Book_Id,  
    Card_No,  
    Branch_Id,  
    Due_Date,  
    Returned_date,  
    julianday(Returned_date) - julianday(Due_Date) AS Days_Late  
    FROM BOOK_LOANS  
    WHERE Late = 1 AND Due_Date BETWEEN ? AND ? AND Returned_Date IS NOT  
    NULL  
    """, (due_date_start.get(), due_date_end.get()))
```

The screenshot shows a web-based application titled "Library Database". On the left, there is a vertical menu with buttons for "Checkout Book", "Add New Borrower", "Add Books", "Find Branch Copies", "Days Late", "Borrower Late Fees", and "Book Information". The main area contains a form with the following fields: Card Number, Name, Address, Phone, BookID, Book Title, Branch_Id, Date Out, Due Date, Due Date Start (pre-filled with "2022-01-01"), Due Date End (pre-filled with "2022-03-31"), Publisher, Author, and # of Copies. Below the form is a "Late Books Report" box containing a list of books with their respective branch IDs, card numbers, and days late.

Book ID	Branch ID	Card #	Days Late
5	567890	1	4.0
6	234567	2	4.0
7	890123	2	29.0
8	456789	3	30.0
14	565656	1	45.0
15	676767	3	6.0
16	787878	2	12.0

Requirement 6a:

```
query = """
    SELECT
    Card_No,
    Borrower_Name,
    printf('$%.2f', LateFeeBalance)
    FROM vBookLoanInfo
    """

params = []

if card_number.get():
    query += " WHERE Card_No = ?"
    params.append(card_number.get())
if name.get():
    query += " WHERE Borrower_Name LIKE ?"
    params.append(f"%{name.get()}%")
```

Library Database

Card Number: 123456 (optional)

Name: John Smith (optional)

Address:

Phone:

BookID:

Book Title:

Branch_Id:

Date Out:

Due Date:

Due Date Start:

Due Date End:

Publisher:

Author:

of Copies:

Checkout Book

Add New Borrower

Add Books

Find Branch Copies

Days Late

Borrower Late Fees

Book Information

Late Fee Report

Card #: 123456, Name: John Smith, LateFeeBalance: \$0.00

Requirement 6b:

```
query = ""
SELECT
    V.Card_No AS "Borrower ID",
    V.Borrower_Name AS "Borrower Name",
    V.Book_Title AS "Book Title",
    V.Branch_Id AS "Branch ID",
    V.Date_Out AS "Date Out",
    V.Due_Date AS "Due Date",
    V.Returned_date AS "Date Returned",
    V.TotalDays AS "Total Days",
    V.Days_Late AS "Days Late",
    CASE
        WHEN V.LateFeeBalance = 0 THEN 'Non-Applicable'
        ELSE printf('%.2f', V.LateFeeBalance)
    END AS "Late Fee"
FROM vBookLoanInfo V, BOOK_LOANS BL, BOOK B
WHERE BL.Book_Id = B.Book_Id AND BL.Card_No = V.Card_No
""
```

```
params = []
```

```
if card_number.get():
    query += " AND BL.Card_No = ?"
    params.append(card_number.get())
```

```
if book_Id.get():
    query += " AND BL.Book_Id = ?"
    params.append(book_Id.get())
```

```
if booktitle.get():
    query += " AND B.Title LIKE ?"
    params.append(f"%{booktitle.get()}%")
```

```
query += " ORDER BY V.LateFeeBalance DESC"
```

The screenshot shows a web application titled "Library Database". On the left is a sidebar with a list of fields: Card Number, Name, Address, Phone, BookID, Book Title, Branch, Id, Date Out, Due Date, Due Date Start, Due Date End, Publisher, Author, # of Copies, Checkout Book, Add New Borrower, Add Books, Find Branch Copies, Days Late, Borrower Late Fees, and Book Information. The main area contains a form with input fields for these fields. The "Card Number" field is filled with "123456 (optional)". The "Book Title" field is filled with "To Kill a Mockingbird (optional)". Below the form are several buttons: "Checkout Book", "Add New Borrower", "Add Books", "Find Branch Copies", "Days Late", "Borrower Late Fees", and "Book Information". At the bottom, there is a "Book Information" section displaying the following data: Borrower ID: 123456, Borrower Name: John Smith, Book Title: To Kill a Mockingbird, Book ID: 1, Date Out: 2022-01-01, Due Date: 2022-02-01, Returned: 2022-02-01, Total Days: 31.0, Days Late: 0, LateFeeBalance: Non-Applicable.