CSE 3330 - 004
Project 2 Part 3

Bibliotech

Library Management System

Aniv Surana - 1001912967
Joshua Mora - 1002020919

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.

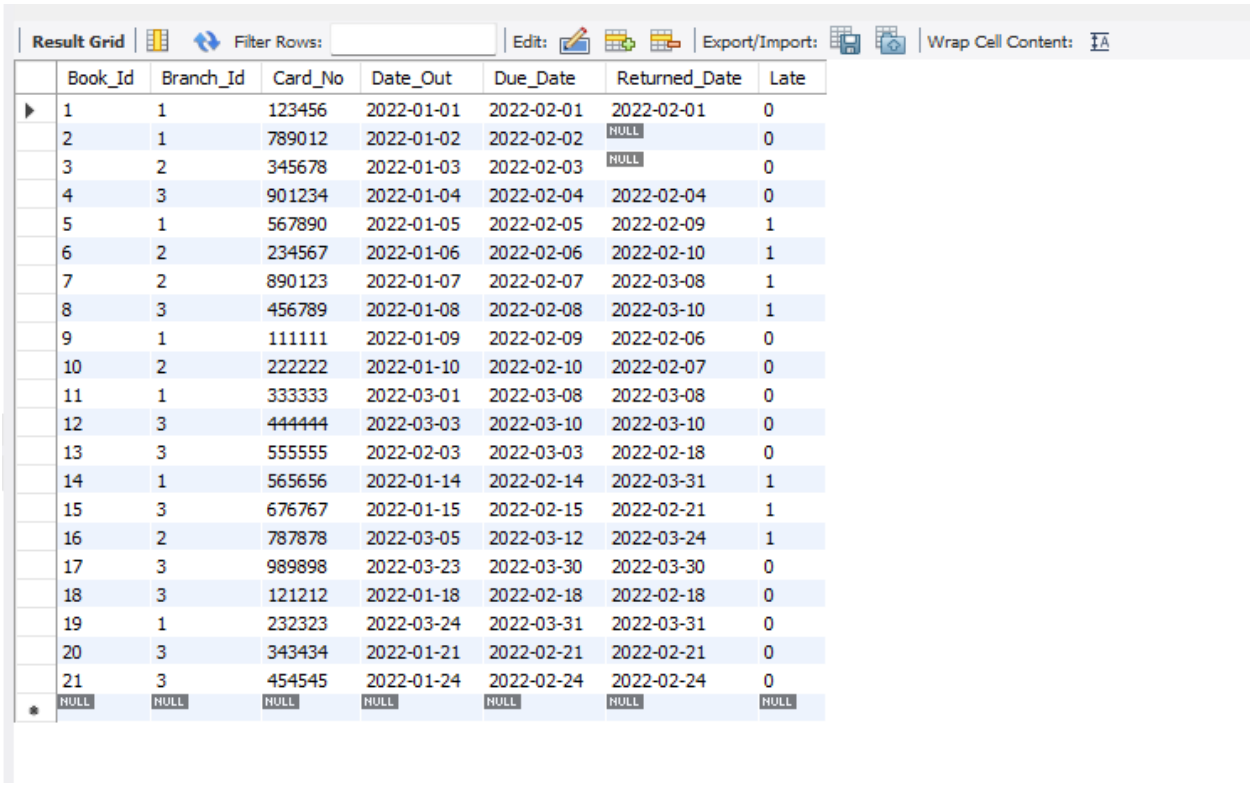
I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

Task 1

Query 1:

Add an extra column ‘Late’ to the Book_Loan table. Values will be 0-for non-late returns, and 1-for late returns. Then update the ‘Late’ column with '1' for all records that they have a return date later than the due date and with '0' for those were returned on time

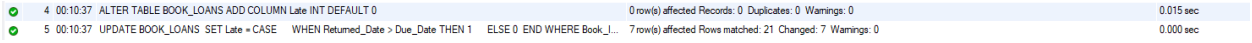
```
ALTER TABLE BOOK_LOANS ADD COLUMN Late INT DEFAULT 0;
UPDATE BOOK_LOANS
SET Late = CASE
   WHEN Returned_Date > Due_Date THEN 1
   ELSE 0
END
WHERE Book_Id > 0 AND Branch_Id > 0 AND Card_No > 0;

SELECT * FROM Book_Loans
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Book_Id	Branch_Id	Card_No	Date_Out	Due_Date	Returned_Date	Late
1	1	123456	2022-01-01	2022-02-01	2022-02-01	0
2	1	789012	2022-01-02	2022-02-02	NULL	0
3	2	345678	2022-01-03	2022-02-03	NULL	0
4	3	901234	2022-01-04	2022-02-04	2022-02-04	0
5	1	567890	2022-01-05	2022-02-05	2022-02-09	1
6	2	234567	2022-01-06	2022-02-06	2022-02-10	1
7	2	890123	2022-01-07	2022-02-07	2022-03-08	1
8	3	456789	2022-01-08	2022-02-08	2022-03-10	1
9	1	111111	2022-01-09	2022-02-09	2022-02-06	0
10	2	222222	2022-01-10	2022-02-10	2022-02-07	0
11	1	333333	2022-03-01	2022-03-08	2022-03-08	0
12	3	444444	2022-03-03	2022-03-10	2022-03-10	0
13	3	555555	2022-02-03	2022-03-03	2022-02-18	0
14	1	565656	2022-01-14	2022-02-14	2022-03-31	1
15	3	676767	2022-01-15	2022-02-15	2022-02-21	1
16	2	787878	2022-03-05	2022-03-12	2022-03-24	1
17	3	989898	2022-03-23	2022-03-30	2022-03-30	0
18	3	121212	2022-01-18	2022-02-18	2022-02-18	0
19	1	232323	2022-03-24	2022-03-31	2022-03-31	0
20	3	343434	2022-01-21	2022-02-21	2022-02-21	0
21	3	454545	2022-01-24	2022-02-24	2022-02-24	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Action Output Response:



4	00:10:37	ALTER TABLE BOOK_LOANS ADD COLUMN Late INT DEFAULT 0	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec
5	00:10:37	UPDATE BOOK_LOANS SET Late = CASE WHEN Returned_Date > Due_Date THEN 1 ELSE 0 END WHERE Book_I...	7 row(s) affected Rows matched: 21 Changed: 7 Warnings: 0	0.000 sec

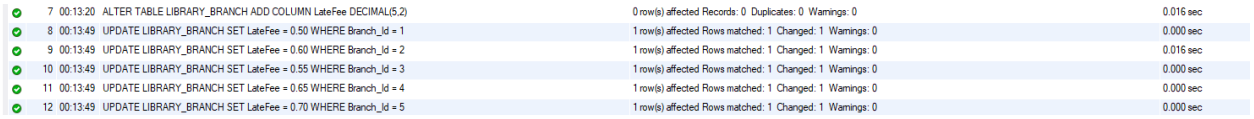
Query 2:

Add an extra column ‘LateFee’ to the Library_Branch table, decide late fee per day for each branch and update that column

```
ALTER TABLE LIBRARY_BRANCH ADD COLUMN LateFee DECIMAL(5,2);
UPDATE LIBRARY_BRANCH SET LateFee = 0.50 WHERE Branch_Id = 1;
UPDATE LIBRARY_BRANCH SET LateFee = 0.60 WHERE Branch_Id = 2;
UPDATE LIBRARY_BRANCH SET LateFee = 0.55 WHERE Branch_Id = 3;
UPDATE LIBRARY_BRANCH SET LateFee = 0.65 WHERE Branch_Id = 4;
UPDATE LIBRARY_BRANCH SET LateFee = 0.70 WHERE Branch_Id = 5;

SELECT * FROM Library_Branch
```

Branch_Id	Branch_Name	Branch_Address	LateFee
1	Main Branch	123 Main St, New York, NY 10003	0.50
2	West Branch	456 West St, Arizona, AR 70622	0.60
3	East Branch	789 East St, New Jersy, NY 32032	0.55
4	North Branch	456 NW, Irving, TX 76100	0.65
5	UTA Branch	123 Cooper St, Arlington TX 76101	0.70
NULL	NULL	NULL	NULL

Action Output Response:



7	00:13:20	ALTER TABLE LIBRARY_BRANCH ADD COLUMN LateFee DECIMAL(5,2)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
8	00:13:49	UPDATE LIBRARY_BRANCH SET LateFee = 0.50 WHERE Branch_Id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
9	00:13:49	UPDATE LIBRARY_BRANCH SET LateFee = 0.60 WHERE Branch_Id = 2	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
10	00:13:49	UPDATE LIBRARY_BRANCH SET LateFee = 0.55 WHERE Branch_Id = 3	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
11	00:13:49	UPDATE LIBRARY_BRANCH SET LateFee = 0.65 WHERE Branch_Id = 4	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
12	00:13:49	UPDATE LIBRARY_BRANCH SET LateFee = 0.70 WHERE Branch_Id = 5	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec

Query 3:

```
CREATE VIEW vBookLoanInfo AS
SELECT
   BL.Card_No,
   BR.Name AS 'Borrower Name',
   BL.Date_Out,
   BL.Due_Date,
   BL.Returned_Date,
   DATEDIFF(IFNULL(BL.Returned_Date, CURDATE()), BL.Date_Out) AS 'TotalDays',
   BK.Title AS 'Book Title',
   CASE
```

```
    WHEN BL.Returned_Date IS NULL OR BL.Returned_Date <= BL.Due_Date THEN
0
    ELSE DATEDIFF(BL.Returned_Date, BL.Due_Date)
  END AS 'Days Returned Late',
  BL.Branch_Id,
  CASE
    WHEN BL.Returned_Date IS NULL OR BL.Returned_Date <= BL.Due_Date THEN
0
    ELSE DATEDIFF(BL.Returned_Date, BL.Due_Date) * LB.LateFee
  END AS 'LateFeeBalance'
FROM
  BOOK_LOANS BL
JOIN
  BORROWER BR ON BL.Card_No = BR.Card_No
JOIN
  BOOK BK ON BL.Book_Id = BK.Book_Id
JOIN
  LIBRARY_BRANCH LB ON BL.Branch_Id = LB.Branch_Id;

SELECT * FROM vBookLoanInfo
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Card_No	Borrower Name	Date_Out	Due_Date	Returned_Date	TotalDays	Book Title	Days Returned Late	Branch_Id	LateFeeBalance
123456	John Smith	2022-01-01	2022-02-01	2022-02-01	31	To Kill a Mockingbird	0	1	0
789012	Jane Doe	2022-01-02	2022-02-02	NULL	701	1984	0	1	0
345678	Bob Johnson	2022-01-03	2022-02-03	NULL	700	Pride and Prejudice	0	2	0
901234	Sarah Kim	2022-01-04	2022-02-04	2022-02-04	31	The Great Gatsby	0	3	0
567890	Tom Lee	2022-01-05	2022-02-05	2022-02-09	35	One Hundred Years of Solitude	4	1	2.00
234567	Emily Lee	2022-01-06	2022-02-06	2022-02-10	35	Animal Farm	4	2	2.40
890123	Michael Park	2022-01-07	2022-02-07	2022-03-08	60	The Catcher in the Rye	29	2	17.40
456789	Laura Chen	2022-01-08	2022-02-08	2022-03-10	61	Lord of the Flies	30	3	16.50
111111	Alex Kim	2022-01-09	2022-02-09	2022-02-06	28	Brave New World	0	1	0
222222	Rachel Lee	2022-01-10	2022-02-10	2022-02-07	28	The Picture of Dorian Gray	0	2	0
333333	William Johnson	2022-03-01	2022-03-08	2022-03-08	7	The Alchemist	0	1	0
444444	Ethan Martinez	2022-03-03	2022-03-10	2022-03-10	7	The God of Small Things	0	3	0
555555	Grace Hernandez	2022-02-03	2022-03-03	2022-02-18	15	Wuthering Heights	0	3	0
565656	Sophia Park	2022-01-14	2022-02-14	2022-03-31	76	The Hobbit	45	1	22.50
676767	Olivia Lee	2022-01-15	2022-02-15	2022-02-21	37	The Lord of the Rings	6	3	3.30
787878	Noah Thompson	2022-03-05	2022-03-12	2022-03-24	19	The Hitchhiker's Guide to the ...	12	2	7.20
989898	Olivia Smith	2022-03-23	2022-03-30	2022-03-30	7	The Diary of a Young Girl	0	3	0
121212	Chloe Park	2022-01-18	2022-02-18	2022-02-18	31	The Da Vinci Code	0	3	0
232323	William Chen	2022-03-24	2022-03-31	2022-03-31	7	The Adventures of Huckleberr...	0	1	0
343434	Olivia Johnson	2022-01-21	2022-02-21	2022-02-21	31	The Adventures of Tom Sawyer	0	3	0
454545	Dylan Kim	2022-01-24	2022-02-24	2022-02-24	31	A Tale of Two Cities	0	3	0

Action Output Response:

```
15  00:20:26  CREATE VIEW vBookLoanInfo AS SELECT  BL.Card_No,  BR.Name AS 'Borrower Name',  BL.Date_Out,  BL.Due_Date, ...  0 row(s) affected  0.015 sec
16  00:20:54  SELECT * FROM vBookLoanInfo LIMIT 0, 1000  21 row(s) returned  0.015 sec / 0.000 sec
```

Task 2

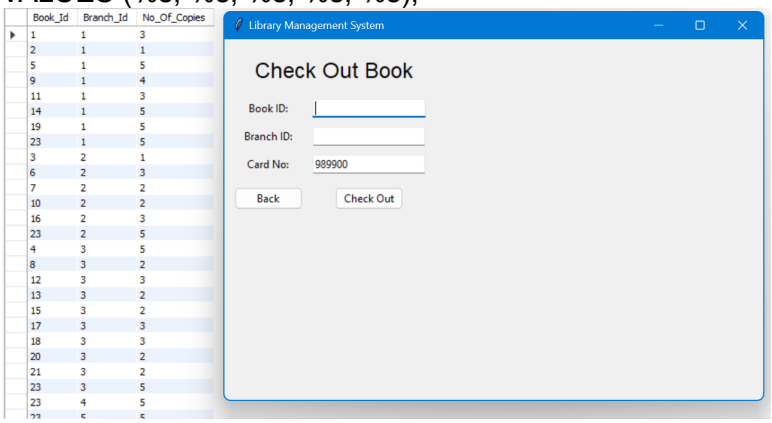
1. User checks out a book, add it to Book_Loan, the number of copies needs to be updated via trigger in the Book_Copies table. Show the output of the updated Book_Copies.

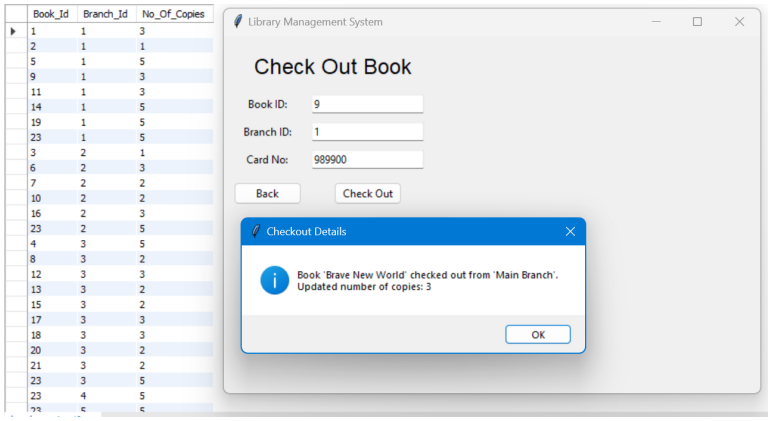
   Trigger to update the value of Book_Copies

   ```
   DELIMITER $$

   CREATE TRIGGER after_book_loan
   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$$

   DELIMITER ;
   ```

   Query to insert the new loan into the table:

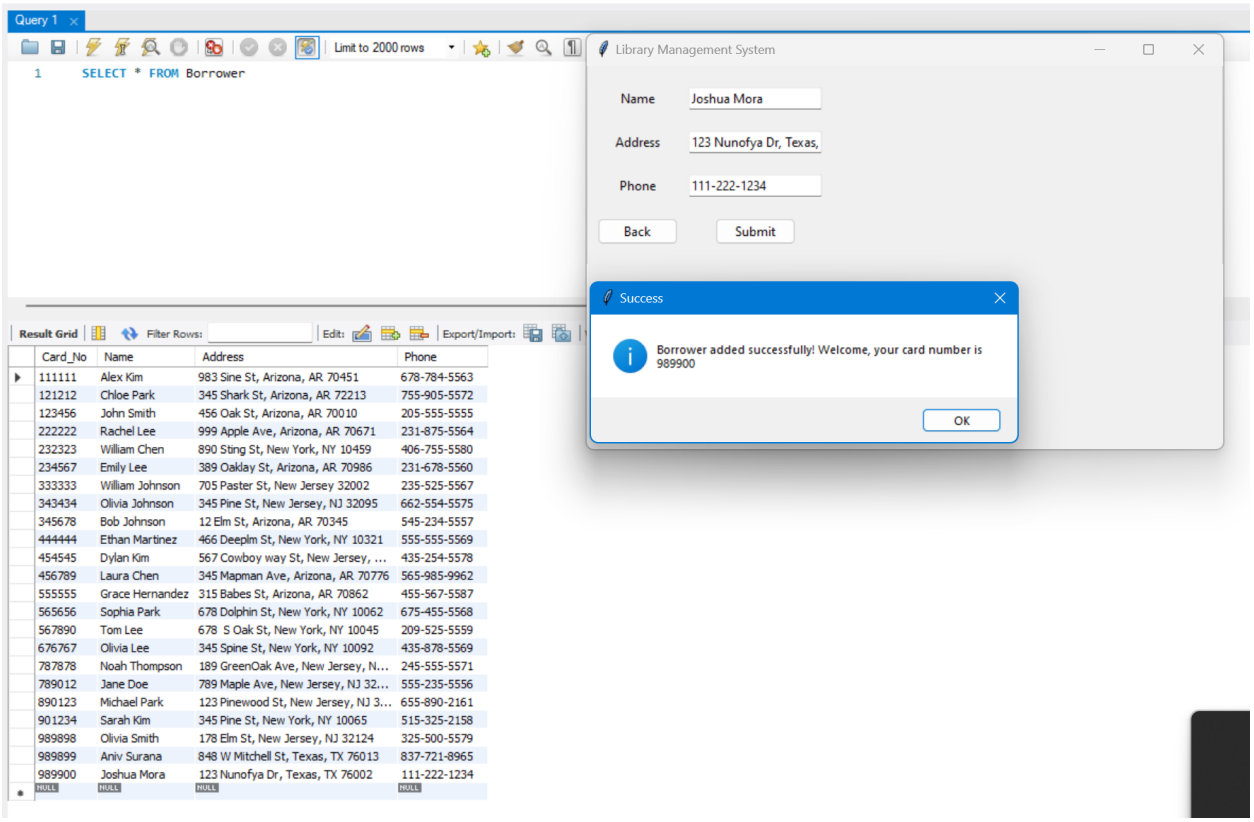
   ```
   INSERT INTO BOOK_LOANS (Book_Id, Branch_Id, Card_No, Date_Out, Due_Date)
   VALUES (%s, %s, %s, %s, %s);
   ```





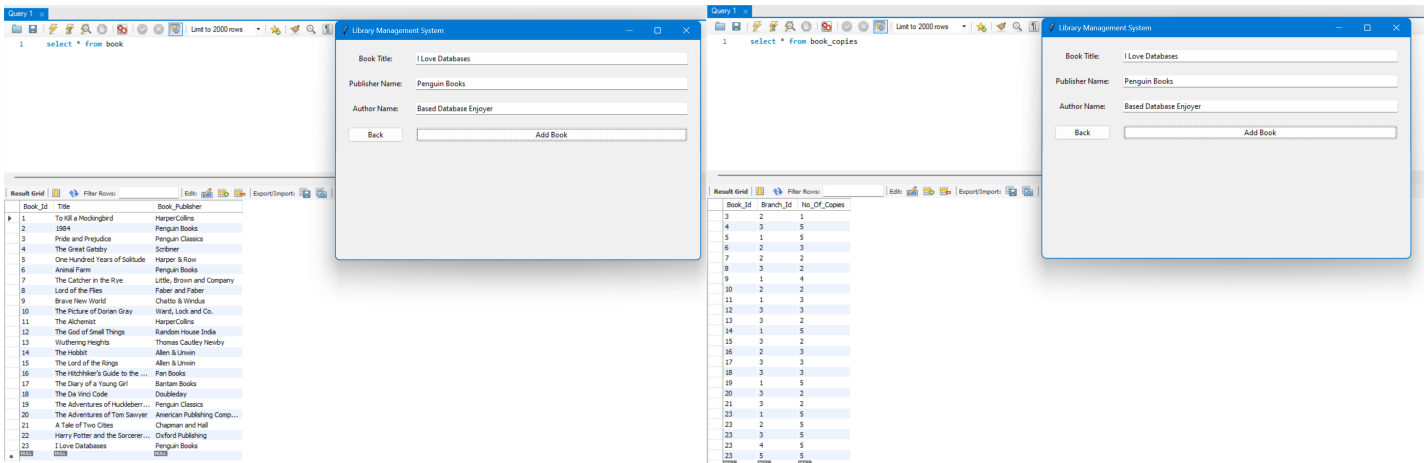
2. Add information about a new Borrower. Do not provide the CardNo in your query. Output the card number as if you are giving a new library card. Submit your editable SQL query that your code executes.

   INSERT INTO BORROWER (Name, Address, Phone) VALUES (%s, %s, %s);



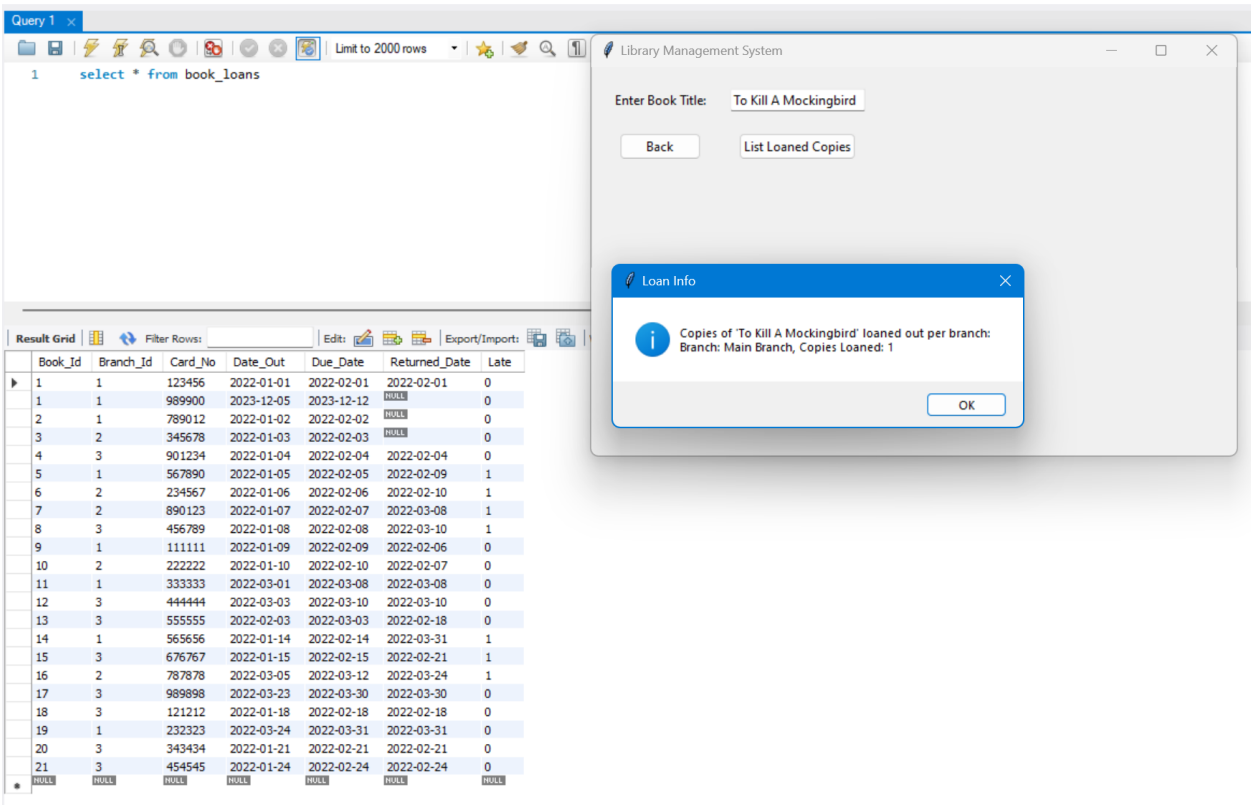
3. Add a new Book with publisher (user can use a publisher that already exists) and author information to all 5 branches with 5 copies for each branch. Submit your editable SQL query that your code executes.

   INSERT INTO BOOK (Title, Book_Publisher) VALUES (%s, %s);
   INSERT INTO BOOK_AUTHORS (Book_Id, Author_Name) VALUES (%s, %s);
   -- Repeat the following for each branch (1 to 5)
   INSERT INTO BOOK_COPIES (Book_Id, Branch_Id, No_Of_Copies) VALUES (%s, %s, 5);



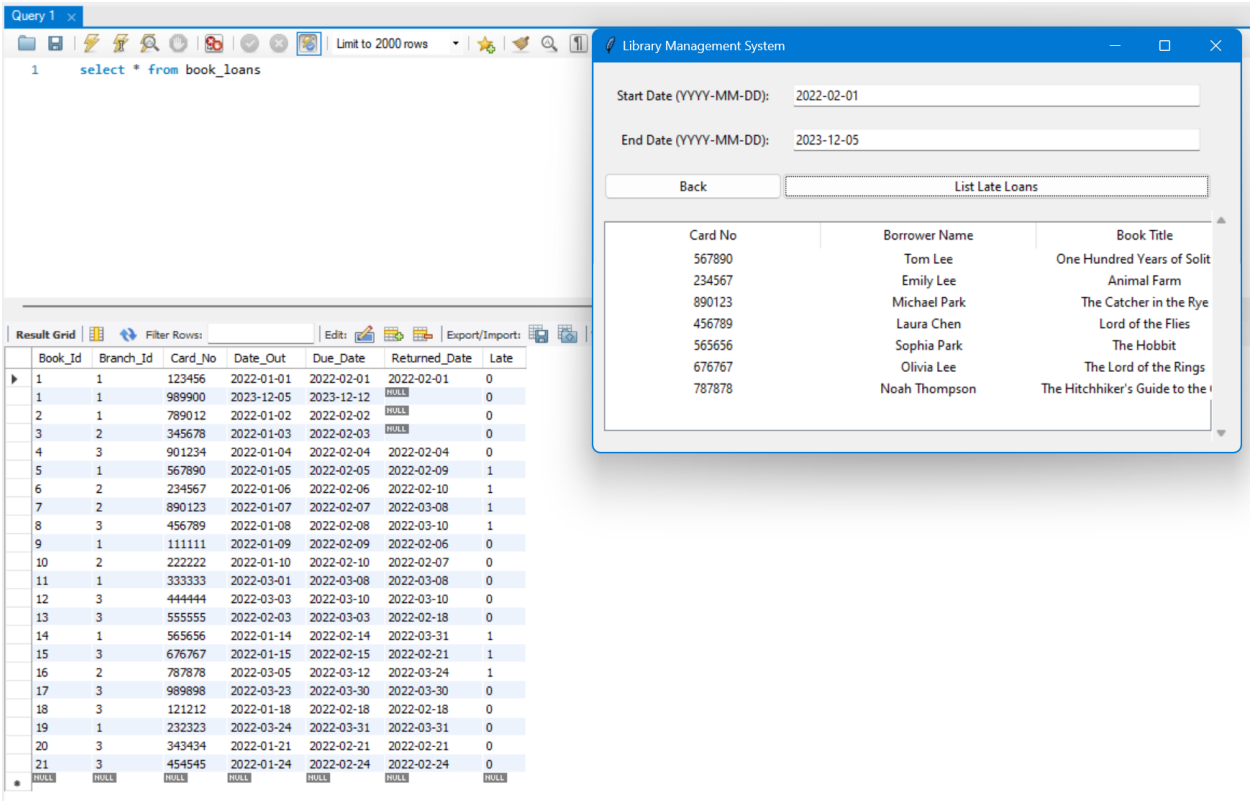
4. Given a book title list the number of copies loaned out per branch.

```
SELECT LB.Branch_Name, COUNT(*) AS Copies_Loaned
FROM BOOK_LOANS AS BL
JOIN BOOK AS B ON BL.Book_Id = B.Book_Id
JOIN LIBRARY_BRANCH AS LB ON BL.Branch_Id = LB.Branch_Id
WHERE B.Title = %s AND BL.Returned_Date IS NULL
GROUP BY LB.Branch_Name;
```



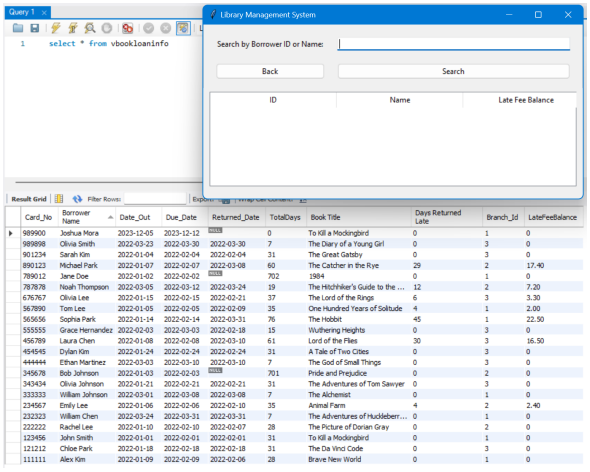
5. Given any due date range list the Book_Loans that were returned late and how many days they were late. Submit your editable SQL queries that your code executes.

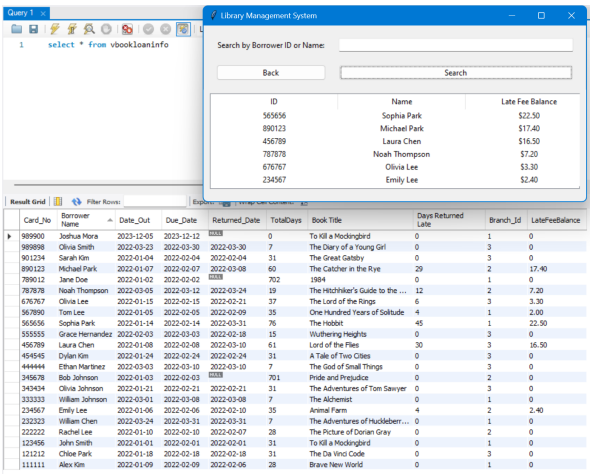
```
SELECT
    Card_No,
    `Borrower Name`,
    `Book Title`,
    Date_Out,
    Due_Date,
    Returned_Date,
    `Days Returned Late`,
    Branch_Id,
    `LateFeeBalance`
FROM
    vBookLoanInfo
WHERE
    Due_Date BETWEEN %s AND %s
    AND `Days Returned Late` > 0;
```

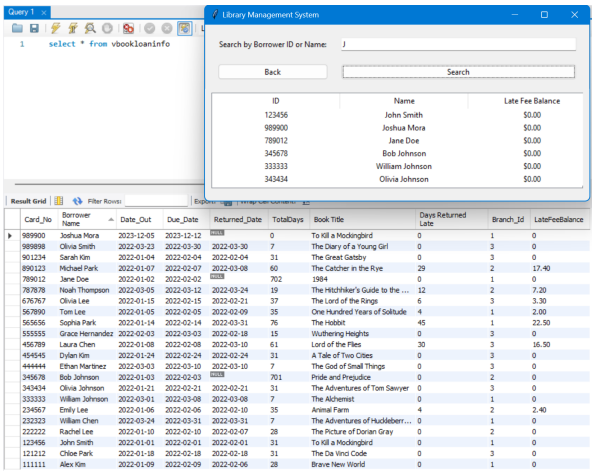


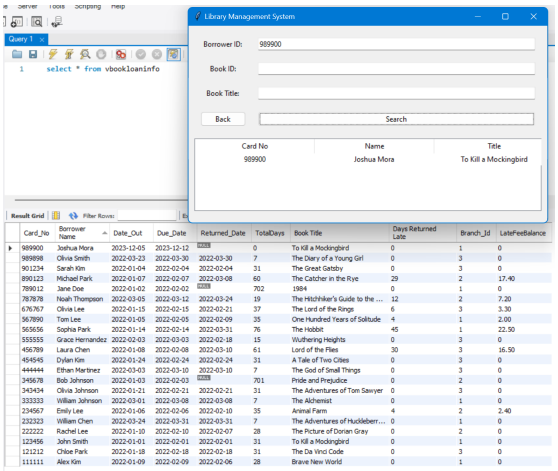
6. a. List for every borrower the ID, name, and if there is any lateFee balance. The user has the right to search either by a borrower ID, name, part of the name, or to run the query with no filters/criteria. The amount needs to be in US dollars. For borrowers with zero (0) or NULL balance, you need to return zero dollars ($0.00). Make sure that your query returns meaningful attribute names. In the case that the user decides not to provide any filters, order the results based on the balance amount.

```
SELECT
    Card_No AS 'ID',
    `Borrower Name` AS 'Name',
    COALESCE(`LateFeeBalance`, 0) AS 'LateFeeBalance'
FROM
    vBookLoanInfo
WHERE
    (`Card_No` LIKE %s OR `Borrower Name` LIKE %s)
ORDER BY
    `LateFeeBalance` DESC;
```



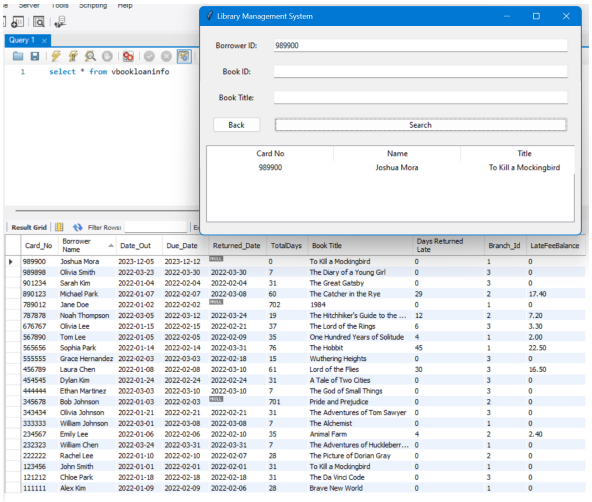


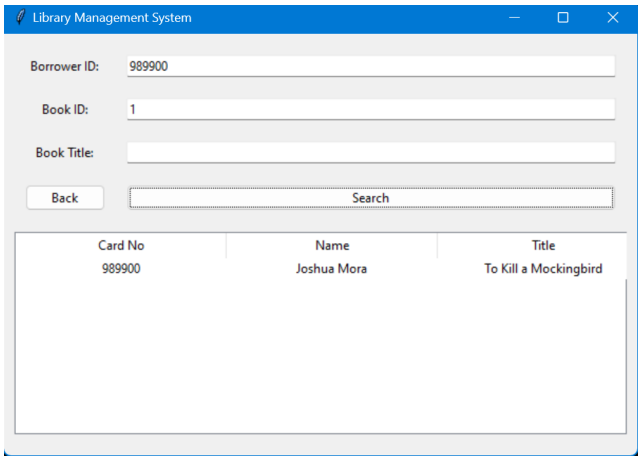


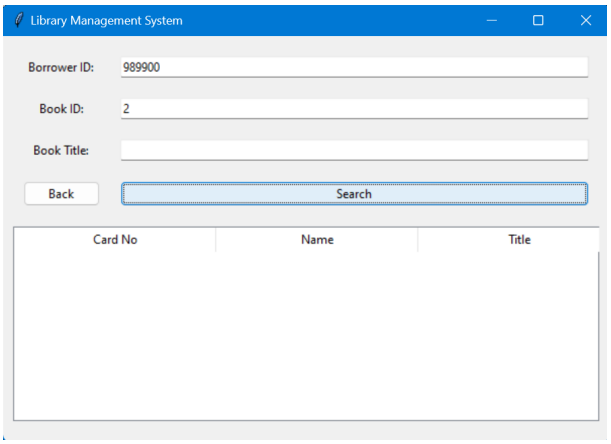


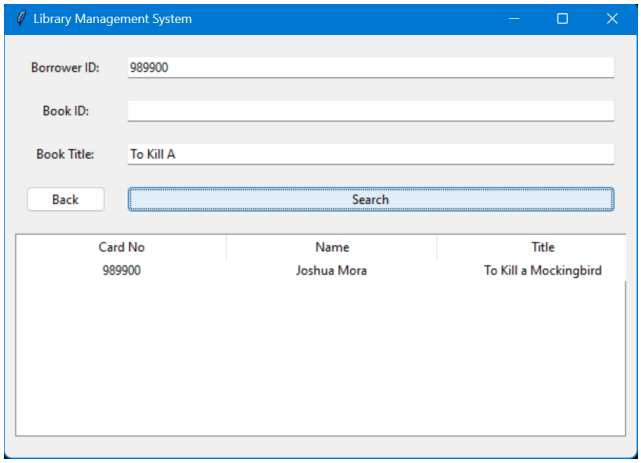
b. List book information in the view. The user must search with borrowerID and any of the following search items: book id, books title, part of book title, or to run the query with no filters/criteria. The late fee amount needs to be in US dollars. The late fee price amount needs to have two decimals as well as the dollar '$' sign. For books that they do not have any late fee amount, you need to substitute the NULL value with a 'Non-Applicable' text. Make sure that your query returns meaningful attribute names. In the case that the user decides not to provide any filters, order the results based on the highest late fee remaining. Submit your editable SQL query that your code executes.

```
SELECT
    v.Card_No AS 'Borrower ID',
    `Borrower Name` AS 'Borrower Name',
    `Book Title` AS 'Title',
    v.Branch_Id AS 'Branch ID',
    COALESCE(CONCAT('$', FORMAT(LateFeeBalance, 2)), 'Non-Applicable') AS
'Late Fee'
  FROM
    vbookloaninfo v JOIN BOOK_LOANS BL ON v.Card_No = BL.Card_No
WHERE
  v.Card_No = %s AND BL.Book_id = %s AND (`Book Title` LIKE %s)
ORDER BY
  `LateFeeBalance` DESC;
```









Contribution List

Aniv Surana:

- Task 1 Queries
- Tkinter Python Setup for GUI
- Task 2: Fulfillment of requirements 1, 2, 4, 5, 6a
- README
- Report

Joshua Mora:

- Task 2: Fulfillment of requirement 3, 6b
- Task 2 Screenshots