

LMS Database

▼ Table of Contents

[Table of Contents](#)

[Github Link:](#)

[Task 1](#)

[Task 2](#)

[Task 3](#)

[Honor Code](#)

▼ Github Link:

https://github.com/DhrutikS257/LMS_Database

▼ Task 1

▼ Creating Table

We decide to use `NOT NULL` for almost all the fields, except for the fields that were `PRIMARY KEY` or `FOREIGN KEY`. There was also one field in `BOOK_LOANS` called `return_date` that we needed to be `NULL` because of a specific query that required us to check if a book hadn't been returned which would mean that that field would be `NULL`. Here is how we created the tables for the database:

▼ `CREATE TABLE PUBLISHER`

```
CREATE TABLE PUBLISHER (  
  publisher_name VARCHAR(50) PRIMARY KEY,  
  phone VARCHAR(15) NOT NULL,
```

```
address VARCHAR(100) NOT NULL
);
```

▼ CREATE TABLE BOOK

```
CREATE TABLE BOOK (
    book_id INTEGER PRIMARY KEY AUTOINCREMENT,
    title VARCHAR(50) NOT NULL,
    book_publisher VARCHAR(50),
    FOREIGN KEY(book_publisher) REFERENCES PUBLISHER(publisher_name)
);
```

▼ CREATE TABLE BOOK_AUTHORS

```
CREATE TABLE BOOK_AUTHORS (
    book_id INTEGER,
    author_name VARCHAR(50),
    FOREIGN KEY(book_id) REFERENCES BOOK(book_id)
);
```

▼ CREATE TABLE LIBRARY_BRANCH

```
CREATE TABLE LIBRARY_BRANCH (
    branch_id INTEGER PRIMARY KEY AUTOINCREMENT,
    branch_name VARCHAR(100) NOT NULL,
    branch_address VARCHAR(100) NOT NULL
);
```

▼ CREATE TABLE BORROWER

```
CREATE TABLE BORROWER(
    card_no INTEGER PRIMARY KEY AUTOINCREMENT,
    name VARCHAR(50) NOT NULL,
    address VARCHAR(100) NOT NULL,
    phone VARCHAR(15) NOT NULL
);
```

▼ CREATE TABLE BOOK_COPIES

```
CREATE TABLE BOOK_COPIES(  
    book_id INTEGER,  
    branch_id INTEGER,  
    no_of_copies INT NOT NULL,  
    FOREIGN KEY(book_id) REFERENCES BOOK(book_id),  
    FOREIGN KEY(branch_id) REFERENCES LIBRARY_BRANCH(branch_id)  
);
```

▼ CREATE TABLE BOOK_LOANS

```
CREATE TABLE BOOK_LOANS (  
    book_id INTEGER,  
    branch_id INTEGER,  
    card_no INTEGER,  
    date_out DATE NOT NULL,  
    due_date DATE NOT NULL,  
    returned_date DATE NULL,  
    FOREIGN KEY(book_id) REFERENCES BOOK(book_id),  
    FOREIGN KEY(branch_id) REFERENCES LIBRARY_BRANCH(branch_id),  
    FOREIGN KEY(card_no) REFERENCES BORROWER(card_no)  
);
```

▼ Task 2

▼ Converting .xlsx to .csv

We used Python to convert the data given to us from .xlsx to .csv which will be used to insert into the table.

```
import pandas as pd, os
```

```
# converting .xlsx to .csv
curr_path = os.getcwd()
data_curr_path = os.path.join(curr_path, 'Data')
csv_curr_path = os.path.join(curr_path, 'CSVData')

readFile = pd.read_excel(os.path.join(data_curr_path, 'Book_Loans.xlsx'))

readFile.to_csv(os.path.join(csv_curr_path, 'idk.csv'), index=False)
```

▼ Inserting Data Into Table

We used Python to insert data into the database, by using the `sqlite3` library we were able to connect to the database, and execute queries. The following code shows the implementation of the data we inserted into the tables:

```
import sqlite3, csv, os

# inserting the data into the database
curr_path = os.getcwd()
curr_data_path = os.path.join(curr_path, 'CSVData')

#change the csv file name depending on which file you want to enter
with open(os.path.join(curr_data_path, 'Publisher.csv'), 'r') as f:
    readFile = csv.reader(f)

    connDB = sqlite3.connect(os.path.join(curr_path, 'Database.db'))
    connectDB = connDB.cursor()

    for row in readFile:
        #change the table name, fields, VALUES, and rows depending on the table you want to enter
        connectDB.execute('INSERT INTO PUBLISHER (publisher_name, phone, address) VALUES (?, ?, ?)', (row[0], row[1], row[2]))

    connDB.commit()
    connDB.close()
```

▼ Task 3

▼ Query Questions

▼ **Question 1:** Insert **yourself** as a New Borrower. Do not provide the Card_no in your query.

```
INSERT INTO BORROWER (card_no, name, address, phone)
VALUES (NULL, 'Dhrutik Solanki', '1234 House Street, State, ST, 76019', '123-456-7890');
-- Return nothing because it is INSERT INTO, the screen shot is just an example showing that the number was updated. One row was affected
```

card_no	name	address	phone
111111	Alex Kim	983 Sine St, Arizona, AR 70451	678-784-5563
121212	Chloe Park	345 Shark St, Arizona, AR 72213	755-905-5572
123456	John Smith	456 Oak St, Arizona, AR 70010	205-555-5555
222222	Rachel Lee	999 Apple Ave, Arizona, AR 70671	231-875-5564
232323	William Chen	890 Sting St, New York, NY 10459	406-755-5580
234567	Emily Lee	389 Oaklay St, Arizona, AR 70986	231-678-5560
333333	William Johnson	705 Paster St, New Jersey 32002	235-525-5567
343434	Olivia Johnson	345 Pine St, New Jersey, NJ 32095	662-554-5575
345678	Bob Johnson	12 Elm St, Arizona, AR 70345	545-234-5557
444444	Ethan Martinez	466 Deeplm St, New York, NY 10321	555-555-5569
454545	Dylan Kim	567 Cowboy way St, New Jersey, NJ 32984	435-254-5578
456789	Laura Chen	345 Mapman Ave, Arizona, AR 70776	565-985-9962
555555	Grace Hernandez	315 Babes St, Arizona, AR 70862	455-567-5587
565656	Sophia Park	678 Dolphin St, New York, NY 10062	675-455-5568
567890	Tom Lee	678 S Oak St, New York, NY 10045	209-525-5559
676767	Olivia Lee	345 Spine St, New York, NY 10092	435-878-5569
787878	Noah Thompson	189 GreenOak Ave, New Jersey, NJ 32453	245-555-5571
789012	Jane Doe	789 Maple Ave, New Jersey, NJ 32542	555-235-5556
890123	Michael Park	123 Pinewood St, New Jersey, NJ 32954	655-890-2161
901234	Sarah Kim	345 Pine St, New York, NY 10065	515-325-2158
989898	Olivia Smith	178 Elm St, New Jersey, NJ 32124	325-500-5579
989899	Dhrutik Solanki	1234 House Street, State, ST, 76019	123-456-7890

▼ **Question 2:** Update your phone number to (837) 721-8965

```
UPDATE BORROWER
SET phone = '837-721-8965'
WHERE name = 'Dhrutik Solanki';
--Return nothing because it is UPDATE, the screen shot is just an example showing that the number was updated. One row was affected
```

card_no	name	address	phone
111111	Alex Kim	983 Sine St, Arizona, AR 70451	678-784-5563
121212	Chloe Park	345 Shark St, Arizona, AR 72213	755-905-5572
123456	John Smith	456 Oak St, Arizona, AR 70010	205-555-5555
222222	Rachel Lee	999 Apple Ave, Arizona, AR 70671	231-875-5564
232323	William Chen	890 Sting St, New York, NY 10459	406-755-5580
234567	Emily Lee	389 Oaklay St, Arizona, AR 70986	231-678-5560
333333	William Johnson	705 Paster St, New Jersey 32002	235-525-5567
343434	Olivia Johnson	345 Pine St, New Jersey, NJ 32095	662-554-5575
345678	Bob Johnson	12 Elm St, Arizona, AR 70345	545-234-5557
444444	Ethan Martinez	466 Deeplm St, New York, NY 10321	555-555-5569
454545	Dylan Kim	567 Cowboy way St, New Jersey, NJ 32984	435-254-5578
456789	Laura Chen	345 Mapman Ave, Arizona, AR 70776	565-985-9962
555555	Grace Hernandez	315 Babes St, Arizona, AR 70862	455-567-5587
565656	Sophia Park	678 Dolphin St, New York, NY 10062	675-455-5568
567890	Tom Lee	678 S Oak St, New York, NY 10045	209-525-5559
676767	Olivia Lee	345 Spine St, New York, NY 10092	435-878-5569
787878	Noah Thompson	189 GreenOak Ave, New Jersey, NJ 32453	245-555-5571
789012	Jane Doe	789 Maple Ave, New Jersey, NJ 32542	555-235-5556
890123	Michael Park	123 Pinewood St, New Jersey, NJ 32954	655-890-2161
901234	Sarah Kim	345 Pine St, New York, NY 10065	515-325-2158
989898	Olivia Smith	178 Elm St, New Jersey, NJ 32124	325-500-5579
989899	Dhrutik Solanki	1234 House Street, State, ST, 76019	837-721-8965

▼ **Question 3:** Increase the number of book_copies by 1 for the **East Branch**

```
UPDATE BOOK_COPIES
SET no_of_copies = no_of_copies +
  1
WHERE branch_id = 3;
```

--Return nothing because it is UPD ATE, the screen shot is just an example showing that the number was updated. Nine rows were affected

book_id	branch_id	no_of_copies
1	1	3
2	1	2
3	2	1
4	3	5
5	1	5
6	2	3
7	2	2
8	3	2
9	1	4
10	2	2
11	1	3
12	3	3
13	3	2
14	1	5
15	3	2
16	2	3
17	3	3
18	3	3
19	1	5
20	3	2
21	3	2

▼ **Question 4-a:** Insert a new BOOK with the following info: Title: **Harry Potter and the Sorcerer's Stone** ; Book_author: **J.K. Rowling**


```
INSERT INTO BOOK (book_id,title,book_publisher)
VALUES (NULL,'Harry Potter and the Sorcerer''s Stone','J.K. Rowling');

-- Return nothing because it is INSERT INTO, the screen shot is just an example showing that the book was updated. One row was affected

--Also we assumed that 'J.K. Rowling' was publisher_name because there is no workaround for us to enter 'J.K. Rowling' as an author and also connect book_id without hardcoding the value
```

book_id	title	book_publisher
1	To Kill a Mockingbird	HarperCollins
2	1984	Penguin Books
3	Pride and Prejudice	Penguin Classics
4	The Great Gatsby	Scribner
5	One Hundred Years of Solitude	Harper & Row
6	Animal Farm	Penguin Books
7	The Catcher in the Rye	Little, Brown and Company
8	Lord of the Flies	Faber and Faber
9	Brave New World	Chatto & Windus
10	The Picture of Dorian Gray	Ward, Lock and Co.
11	The Alchemist	HarperCollins
12	The God of Small Things	Random House India
13	Wuthering Heights	Thomas Cautley Newby
14	The Hobbit	Allen & Unwin
15	The Lord of the Rings	Allen & Unwin
16	The Hitchhiker's Guide to the Galaxy	Pan Books
17	The Diary of a Young Girl	Bantam Books
18	The Da Vinci Code	Doubleday
19	The Adventures of Huckleberry Finn	Penguin Classics
20	The Adventures of Tom Sawyer	American Publishing Company
21	A Tale of Two Cities	Chapman and Hall
22	Harry Potter and the Sorcerer's Stone	J.K. Rowling

▼ **Question 4-b:** You also need to insert the following branches:

North Branch	456 NW, Irving, TX 76100
UTA Branch	123 Cooper St, Arlington TX 76101

```

INSERT INTO LIBRARY_BRANCH (branch_id,branch_name,branch_address)
VALUES (NULL,'North Branch','456 NW, Irving, TX 76100');

INSERT INTO LIBRARY_BRANCH (branch_id,branch_name,branch_address)
VALUES (NULL,'UTA Branch','123 Cooper St, Arlington TX 76101')

-- Return nothing because it is INSERT INTO, the screen shot is just an example showing that the library_branch was updated. Two rows were affected

```

branch_id	branch_name	branch_address
1	Main Branch	123 Main St, New York, NY 10003
2	West Branch	456 West St, Arizona, AR 70622
3	East Branch	789 East St, New Jersey, NY 32032
4	North Branch	456 NW, Irving, TX 76100
5	UTA Branch	123 Cooper St, Arlington TX 76101

▼ **Question 5:** Return all Books that were loaned between March 5, 2022 until March 23, 2022. List Book title and Branch name, and how many days it was borrowed for.

```

SELECT b.title AS Title, lb.branch_name AS Branch_Name,
       julianday(returned_date) - julianday(date_out) AS Days_Borrowed
FROM BOOK as b, LIBRARY_BRANCH as lb
JOIN BOOK_LOANS AS bl ON bl.book_id = b.book_id AND bl.branch_id = lb.branch_id
WHERE bl.date_out BETWEEN '2022-03-05' AND '2022-03-23';
--Returns two books with title, branch_name, and Days_borrowed

```

Title	Branch_Name	Days_Borrowed
The Diary of a Young Girl	East Branch	7.0
The Hitchhiker's Guide to the Galaxy	West Branch	-9.0

▼ **Question 6:** Return a list of `borrower_names`, that have books not been returned

```
SELECT bo.name, bo.card_no
FROM BOOK_LOANS AS bl
JOIN BORROWER as bo ON bl.card_no = bo.card_no
WHERE bl.returned_date IS NULL
GROUP BY bo.card_no;
--This query won't output anything because there are no returned_date values that are NULL
```

▼ **Question 7:** Create a report that will return all branches with the number of books borrowed per branch separated by if they have been returned, still borrowed, or late.

```
SELECT lb.branch_id, lb.branch_name,
       SUM(CASE WHEN bl.returned_date <= bl.due_date THEN 1 ELSE 0 END) AS Returned,
       SUM(CASE WHEN bl.returned_date IS NULL THEN 1 ELSE 0 END) AS Borrowed,
       SUM(CASE WHEN bl.returned_date > bl.due_date THEN 1 ELSE 0 END) AS Late
FROM BOOK_LOANS AS bl
JOIN LIBRARY_BRANCH AS lb ON bl.branch_id = lb.branch_id
JOIN BOOK AS b ON b.book_id = bl.book_id
GROUP BY lb.branch_id, lb.branch_name;
--Returns three rows along with their branch_name, books returned, borrowed, and late
```

branch_id	branch_name	Returned	Borrowed	Late
1	Main Branch	5	0	2
2	West Branch	3	0	2
3	East Branch	7	0	2

▼ **Question 8:** What is the maximum number of days a book has been borrowed.

```
SELECT MAX(julianday(returned_date) - julianday(date_out)) AS Max_Borrowed_Days
FROM BOOK_LOANS;
--Returns one rows with the max amount borrowed days for a single borrower
```

Max_Borrowed_Days
76.0

▼ **Question 9:** Create a report for Ethan Martinez with all the books they borrowed. List the book title and author. Also, calculate the number of days each book was borrowed for and if any book is late in return date. Order the results by the date_out.

```
SELECT b.title,ba.author_name,julianday(returned_date)-julianday(date_out) AS
Num_Day,
CASE WHEN bl.returned_date > bl.due_date THEN 1 ELSE 0 END AS Returned_Late
FROM BOOK_LOANS AS bl
JOIN BORROWER as bo ON bl.card_no = bo.card_no
JOIN BOOK AS b ON bl.book_id = b.book_id
JOIN BOOK_AUTHORS AS ba ON ba.book_id = b.book_id
WHERE bo.name = 'Ethan Martinez'
ORDER BY bl.date_out DESC;
--Returns one row for book borrowed by 'Ethan Martinez', and it's author, and
the number of days he kept it for and if it was returned late (if return_late
is 0, means it is returned on time, otherwise if it is one, then it is returned late)
```

title	author_name	Num_Day	Returned_Late
The God of Small Things	Arundhati Roy	7.0	0

▼ **Question 10:** Return all borrowers and their addresses that borrowed a book.

```
SELECT bo.name, bo.address
FROM BOOK_LOANS AS bl
JOIN BORROWER as bo ON bl.card_no = bo.card_no;
--Returns twenty one rows for borrowers and thier address for borrowers who bo
rrowed book and their address
```

name	address
John Smith	456 Oak St, Arizona, AR 70010
Jane Doe	789 Maple Ave, New Jersey, NJ 32542
Bob Johnson	12 Elm St, Arizona, AR 70345
Sarah Kim	345 Pine St, New York, NY 10065
Tom Lee	678 S Oak St, New York, NY 10045
Emily Lee	389 Oaklay St, Arizona, AR 70986
Michael Park	123 Pinewood St, New Jersey, NJ 32954
Laura Chen	345 Mapman Ave, Arizona, AR 70776
Alex Kim	983 Sine St, Arizona, AR 70451
Rachel Lee	999 Apple Ave, Arizona, AR 70671
William Johnson	705 Paster St, New Jersey 32002
Ethan Martinez	466 Deeplm St, New York, NY 10321
Grace Hernandez	315 Babes St, Arizona, AR 70862
Sophia Park	678 Dolphin St, New York, NY 10062
Olivia Lee	345 Spine St, New York, NY 10092
Olivia Smith	178 Elm St, New Jersey, NJ 32124
Noah Thompson	189 GreenOak Ave, New Jersey, NJ 32453
Chloe Park	345 Shark St, Arizona, AR 72213
William Chen	890 Sting St, New York, NY 10459
Olivia Johnson	345 Pine St, New Jersey, NJ 32095
Dylan Kim	567 Cowboy way St, New Jersey, NJ 32984

▼ 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