

Submission 1

1. Title

Library Management System (LMS)

2. Outline

LMS is a database system that will store and organize the library's data. The system can keep track the members, the libraries, the books, and their status. The books identified by their ISBN as a book unique ID number. Each member will sign up to get a library card therefore each member will provide his/her first name, last name, address, and phone number. The library will issue a library card with a unique member ID number for each person. The checkout operation will require both the member ID number and the library card number. Each library has a unique name, address and a phone number.

3. Description

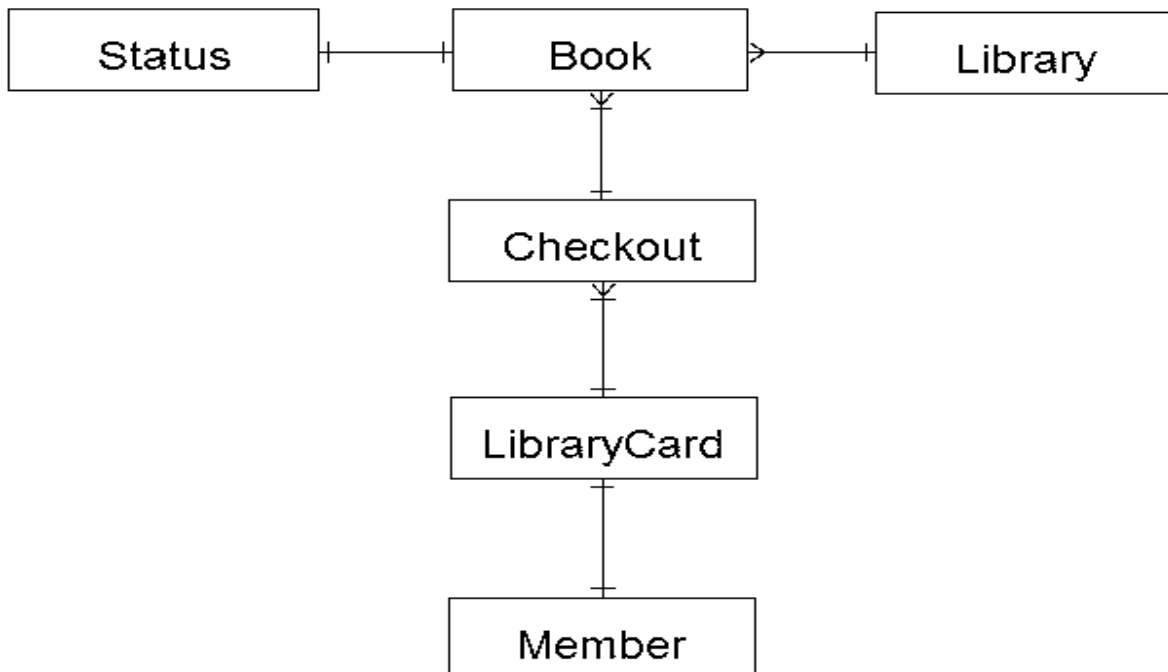
The member functions are:

1. Open the Library Management System (LMS)
2. Search for book(s) by ISBN, title, author, year, and/or price
3. Use the library card number to check out the required book(s)
4. Update personal information such as name, phone number and/or address

The employee functions are:

1. Add books to the inventory
2. Delete books out of the inventory
3. Add member IDs
4. Delete member IDs
5. Issue library cards
6. Stop library cards
7. Check out books
8. Update member information

4. ER diagram



Submission 2

1. Table schemas

Member (memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber)

Book (bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice)

Checkout (cardNumber, bookISBN, confirmationNumber, borrowDate, returnDate)

Library (libraryName, libraryAddress, libraryPhoneNumber)

LibraryCard (cardNumber, memberID, expireDate)

Status (bookISBN, bookStatus, statusComment)

2. Explanation of tables are normalized

1. The tables are in the First Normal Form (1NF) since they do not contain any non-atomic value such as list, array, or other composite data type and no repeated attributes from any table.
2. The tables are in the Second Normal Form (2NF) since they do not contain any partial key dependency.
3. The tables are in the Third Normal Form (3NF) since they do not contain any transitive dependencies.
4. The tables are in the Boyce-Codd Normal Form (BCNF) since every determinant is a candidate key.
5. The tables are in the Fourth Normal Form (4NF) since they do not contain more than one independent multi-valued dependency in each table.
6. The tables are in the Fifth Normal Form (5NF) they do not contain any cyclic dependencies.

3. Data Definition Language (DDL) to create the tables

DROP TABLE IF EXISTS Member;

```
CREATE TABLE Member (memberID INTEGER NOT NULL,  
    memberFirstName CHAR (16) NOT NULL,  
    memberLastName CHAR (16) NOT NULL,  
    memberAddress VARCHAR (64) NOT NULL,  
    memberPhoneNumber CHAR (16) NOT NULL,  
    PRIMARY KEY (memberID));
```

DROP TABLE IF EXISTS Book;

```
CREATE TABLE Book (bookISBN VARCHAR (16) NOT NULL,
                    libraryName CHAR (64) NOT NULL,
                    bookTitle VARCHAR (64) NOT NULL,
                    bookAuthor CHAR (32) NOT NULL,
                    bookYear INTEGER NOT NULL,
                    bookPrice DOUBLE NOT NULL,
                    PRIMARY KEY (bookISBN),
                    FOREIGN KEY (libraryName) REFERENCES Library);
```

DROP TABLE IF EXISTS Checkout;

```
CREATE TABLE Checkout (cardNumber INTEGER NOT NULL,  
                        bookISBN VARCHAR (16) NOT NULL,  
                        confirmationNumber INTEGER NOT NULL,  
                        borrowDate DATE NULL,  
                        returnDate DATE NULL,  
                        PRIMARY KEY (confirmationNumber),  
                        FOREIGN KEY (bookISBN) REFERENCES Book,  
                        FOREIGN KEY (cardNumber) REFERENCES LibraryCard);
```

DROP TABLE IF EXISTS Library;

[illegible]

```
libraryPhoneNumber CHAR(16) NOT NULL,  
PRIMARY KEY (libraryName));
```

```
DROP TABLE IF EXISTS LibraryCard;  
CREATE TABLE LibraryCard (cardNumber INTEGER NOT NULL,  
                           memberID INTEGER NOT NULL,  
                           expireDate DATE NULL,  
                           PRIMARY KEY (cardNumber),  
                           FOREIGN KEY (memberID) REFERENCES Member);
```

```
DROP TABLE IF EXISTS Status;  
CREATE TABLE Status (bookISBN VARCHAR (16) NOT NULL,  
                      bookStatus CHAR (16) NOT NULL,  
                      statusComment VARCHAR (32) NOT NULL,  
                      PRIMARY KEY (bookStatus),  
                      FOREIGN KEY (bookISBN) REFERENCES Book);
```

4. Run the DDL to create the tables in a DBMS

```
mysql> DROP TABLE IF EXISTS Member;  
Query OK, 0 rows affected (0.02 sec)  
mysql> CREATE TABLE Member (memberID INTEGER NOT NULL,  
-> memberFirstName CHAR (16) NOT NULL,  
-> memberLastName CHAR (16) NOT NULL,  
-> memberAddress VARCHAR (64) NOT NULL,  
-> memberPhoneNumber CHAR (16) NOT NULL,  
-> PRIMARY KEY (memberID));  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> DROP TABLE IF EXISTS Book;  
Query OK, 0 rows affected (0.02 sec)  
mysql> CREATE TABLE Book (bookISBN VARCHAR (16) NOT NULL,  
-> libraryName CHAR (64) NOT NULL,  
-> bookTitle VARCHAR (64) NOT NULL,  
-> bookAuthor CHAR (32) NOT NULL,  
-> bookYear INTEGER NOT NULL,  
-> bookPrice DOUBLE NOT NULL,  
-> PRIMARY KEY (bookISBN),  
-> FOREIGN KEY (libraryName) REFERENCES Library);
```

```
Query OK, 0 rows affected (0.00 sec)  
mysql> CREATE TABLE Checkout (cardNumber INTEGER NOT NULL,  
-> bookISBN VARCHAR (16) NOT NULL,  
-> confirmationNumber INTEGER NOT NULL,  
-> borrowDate DATE NULL,  
-> returnDate DATE NULL,  
-> PRIMARY KEY (confirmationNumber),  
-> FOREIGN KEY (bookISBN) REFERENCES Book,  
-> FOREIGN KEY (cardNumber) REFERENCES LibraryCard);  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> DROP TABLE IF EXISTS Library;  
Query OK, 0 rows affected (0.00 sec)  
mysql> CREATE TABLE Library (libraryName VARCHAR (64) NOT NULL,
```

```
-> libraryAddress VARCHAR (64) NOT NULL,  
-> libraryPhoneNumber CHAR(16) NOT NULL,  
-> PRIMARY KEY (libraryName));  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> DROP TABLE IF EXISTS LibraryCard;  
Query OK, 0 rows affected (0.02 sec)  
mysql> CREATE TABLE LibraryCard (cardNumber INTEGER NOT NULL,  
-> memberID INTEGER NOT NULL,  
-> expireDate DATE NULL,  
-> PRIMARY KEY (cardNumber),  
-> FOREIGN KEY (memberID) REFERENCES Member);  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> DROP TABLE IF EXISTS Status;  
Query OK, 0 rows affected (0.00 sec)  
mysql> CREATE TABLE Status (bookISBN VARCHAR (16) NOT NULL,  
-> bookStatus CHAR (16) NOT NULL,  
-> statusComment VARCHAR (32) NOT NULL,  
-> PRIMARY KEY (bookStatus),  
-> FOREIGN KEY (bookISBN) REFERENCES Book);  
Query OK, 0 rows affected (0.00 sec)
```

5. Add sample data to the tables

```
INSERT INTO Member  
(memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber) VALUES  
(12345, 'Amir', 'Yousef', '12345 Tree Ave, Tampa, FL 12345', '813-123-4567');
```

```
INSERT INTO Member  
(memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber) VALUES  
(23456, 'Cathy', 'Powel', '23456 Citrus Ave, Clearwater, FL 23456', '813-234-5678');
```

```
INSERT INTO Member  
(memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber) VALUES  
(34567, 'April', 'Silver', '34567 Tree Ave, Oldsmar, FL 34567', '813-345-6789');
```

```
INSERT INTO Member  
(memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber) VALUES  
(45678, 'Justin', 'Jones', '45678 Book Ave, Palm Harbor, FL 45678', '813-456-7890');
```

```
INSERT INTO Member  
(memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber) VALUES  
(56789, 'Mary', 'Edward', '56789 Deer Ave, Brandon, FL 56789', '813-567-8910');
```

```
INSERT INTO Book  
(bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES  
('978-0123456789', 'Tampa Public Library', 'Computer Science', 'Alex Loui', '2000', 10.00);
```

```
INSERT INTO Book  
(bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES  
('978-1234567890', 'Clearwater Public Library', 'Computer Engineering', 'Michael Lee', '2003', 30.00);
```

```
INSERT INTO Book
(bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES
('978-2345678901', 'Oldsmar Public Library', 'Mathematics', 'George White', '2005', 50.00);
```

```
INSERT INTO Book
(bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES
('978-3456789012', 'Palm Harbor Public Library', 'Physics', 'Lucy Black', '2007', 70.00);
```

```
INSERT INTO Book
(bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES
('978-4567890123', 'Brandon Public Library', 'Public Speech', 'Dave Sink', '2009', 90.00);
```

```
INSERT INTO Checkout (cardNumber, bookISBN, confirmationNumber, borrowDate, returnDate)
VALUES (11111, '978-0123456789', 1234567, '2013-05-01', '2013-06-01');
```

```
INSERT INTO Checkout (cardNumber, bookISBN, confirmationNumber, borrowDate, returnDate)
VALUES (22222, '978-1234567890', 2345678, '2013-07-09', '2013-08-09');
```

```
INSERT INTO Checkout (cardNumber, bookISBN, confirmationNumber, borrowDate, returnDate)
VALUES (33333, '978-2345678901', 3456789, '2013-05-02', '2013-06-02');
```

```
INSERT INTO Checkout (cardNumber, bookISBN, confirmationNumber, borrowDate, returnDate)
VALUES (44444, '978-3456789012', 4567890, '2013-07-03', '2013-08-03');
```

```
INSERT INTO Checkout (cardNumber, bookISBN, confirmationNumber, borrowDate, returnDate)
VALUES (55555, '978-4567890123', 5678901, '2012-12-01', NULL);
```

```
INSERT INTO LibraryCard (cardNumber, memberID, expireDate)
VALUES (11111, 12345, '2016-01-01');
```

```
INSERT INTO LibraryCard (cardNumber, memberID, expireDate)
VALUES (22222, 23456, '2017-01-01');
```

```
INSERT INTO LibraryCard (cardNumber, memberID, expireDate)
VALUES (33333, 34567, '2018-01-01');
```

```
INSERT INTO LibraryCard (cardNumber, memberID, expireDate)
VALUES (44444, 45678, '2019-01-01');
```

```
INSERT INTO LibraryCard (cardNumber, memberID, expireDate)
VALUES (55555, 56789, '2020-01-01');
```

```
INSERT INTO Library (libraryName, libraryAddress, libraryPhoneNumber)
VALUES ('Tampa Public Library', '112233 Tampa RD, Tampa, FL 112233', '813-112-2334');
```

```
INSERT INTO Library (libraryName, libraryAddress, libraryPhoneNumber)
VALUES ('Clearwater Public Library', '221133 Clearwater RD, Clearwater, FL 221133', '813-221-1334');
```

```
INSERT INTO Library (libraryName, libraryAddress, libraryPhoneNumber)
```

```
VALUES ('Oldsmar Public Library', '332211 Oldsmar RD, Oldsmar, FL 332211', '813-332-2113');
```

```
INSERT INTO Library (libraryName, libraryAddress, libraryPhoneNumber)
```

```
VALUES ('Palm Harbor Public Library', '123123 Palm RD, Palm Harbor, FL 123123', '813-123-1233');
```

```
INSERT INTO Library (libraryName, libraryAddress, libraryPhoneNumber)
```

```
VALUES ('Brandon Public Library', '321321 Brandon RD, Brandon, FL 321321', '813-321-3214');
```

```
INSERT INTO Status (bookISBN, bookStatus, statusComment)
```

```
VALUES ('978-0123456789', 'Available', 'In Stock Since 2013-06-01');
```

```
INSERT INTO Status (bookISBN, bookStatus, statusComment)
```

```
VALUES ('978-1234567890', 'Not Available', 'Borrowed until 2013-08-09');
```

```
INSERT INTO Status (bookISBN, bookStatus, statusComment)
```

```
VALUES ('978-2345678901', 'On Hold', 'Damaged Since 2013-06-02');
```

```
INSERT INTO Status (bookISBN, bookStatus, statusComment)
```

```
VALUES ('978-3456789012', 'Returned', 'In Processing Since 2013-08-03');
```

```
INSERT INTO Status (bookISBN, bookStatus, statusComment)
```

```
VALUES ('978-4567890123', 'Lost/Stolen', 'Not Found Since 2013-01-03');
```

6. Description of the user interface

The user interface based on the manual query method. Each member will be able search for a book and find out if the book is available or not based on the status and after the member picked the needed book, member can scan/enter the ISBN number of the book and checkout the book for himself or the library's employee can do that for the member. The member will be able to find out exactly where is the location of the book to go to directly to that library location. The only functions not available for members are insert or delete books into inventory, update the book status, lookup the memberID or member card information to find who has the book if its borrowed since these functions are all for the library employees. Finally the member can have a receipt after check out process to find out the borrow date and the return date in case if forget what time suppose to return at the mean time that receipt includes receipt number, library card number, and the return date will help the library employee to update the status of the book. The user interface may implement easily by using PHP or JAVA since both can access and interact with the SQL.

7. Implementing the user interface

```
mysql> INSERT INTO Member
```

```
-> (memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber)
```

```
VALUES
```

```
-> (12345, 'Amir', 'Yousef', '12345 Tree Ave, Tampa, FL 12345', '813-123-4567');
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Member
```

```
-> (memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber)
```

```
VALUES
```

```
-> (23456, 'Cathy', 'Powel', '23456 Citrus Ave, Clearwater, FL 23456', '813-234-5678');
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Member
```

```
-> (memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber)
```

```
VALUES
```

-> (34567, 'April', 'Silver', '34567 Tree Ave, Oldsmar, FL 34567', '813-345-6789');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Member
-> (memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber)
VALUES
-> (45678, 'Justin', 'Jones', '45678 Book Ave, Palm Harbor, FL 45678', '813-456-7890');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Member
-> (memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber)
VALUES
-> (56789, 'Mary', 'Edward', '56789 Deer Ave, Brandon, FL 56789', '813-567-8910');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Member;

memberID	memberFirstName	memberLastName	memberAddress	memberPhoneNumber
12345	Amir	Yousef	12345 Tree Ave, Tampa, FL 12345	813-123-4567
23456	Cathy	Powel	23456 Citrus Ave, Clearwater, FL 23456	813-234-5678
34567	April	Silver	34567 Tree Ave, Oldsmar, FL 34567	813-345-6789
45678	Justin	Jones	45678 Book Ave, Palm Harbor, FL 45678	813-456-7890
56789	Mary	Edward	56789 Deer Ave, Brandon, FL 56789	813-567-8910

5 rows in set (0.00 sec)

mysql> INSERT INTO Book
-> (bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES
-> ('978-0123456789', 'Tampa Public Library', 'Computer Science', 'Alex Loui', '2000', 10.00);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Book
-> (bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES
-> ('978-1234567890', 'Clearwater Public Library', 'Computer Engineering', 'Michael Lee', '2003', 30.00);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Book
-> (bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES
-> ('978-2345678901', 'Oldsmar Public Library', 'Mathematics', 'George White', '2005', 50.00);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Book
-> (bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES
-> ('978-3456789012', 'Palm Harbor Public Library', 'Physics', 'Lucy Black', '2007', 70.00);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Book
-> (bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES
-> ('978-4567890123', 'Brandon Public Library', 'Public Speech', 'Dave Sink', '2009', 90.00);

Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Book;

bookISBN	libraryName	bookTitle	bookAuthor	bookYear	bookPrice
978-0123456789	Tampa Public Library	Computer Science	Alex Loui	2000	10
978-1234567890	Clearwater Public Library	Computer Engineering	Michael Lee	2003	30
978-2345678901	Oldsmar Public Library	Mathematics	George White	2005	50
978-3456789012	Palm Harbor Public Library	Physics	Lucy Black	2007	70
978-4567890123	Brandon Public Library	Public Speech	Dave Sink	2009	90

5 rows in set (0.00 sec)

mysql> INSERT INTO LibraryCard (cardNumber, memberID, expireDate)

-> VALUES (11111, 12345, '2016-01-01');

Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO LibraryCard (cardNumber, memberID, expireDate)

-> VALUES (22222, 23456, '2017-01-01');

Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO LibraryCard (cardNumber, memberID, expireDate)

-> VALUES (33333, 34567, '2018-01-01');

Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO LibraryCard (cardNumber, memberID, expireDate)

-> VALUES (44444, 45678, '2019-01-01');

Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO LibraryCard (cardNumber, memberID, expireDate)

-> VALUES (55555, 56789, '2020-01-01');

Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM LibraryCard;

cardNumber	memberID	expireDate
11111	12345	2016-01-01
22222	23456	2017-01-01
33333	34567	2018-01-01
44444	45678	2019-01-01
55555	56789	2020-01-01

5 rows in set (0.00 sec)

mysql> INSERT INTO Status (bookISBN, bookStatus, statusComment)

-> VALUES ('978-0123456789', 'Available', 'In Stock Since 2013-06-01');

Query OK, 1 row affected (0.00 sec)


```
mysql> INSERT INTO Status (bookISBN, bookStatus, statusComment)
-> VALUES VALUES ('978-1234567890', 'Not Available', 'Borrowed until 2013-08-09');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Status (bookISBN, bookStatus, statusComment)
-> VALUES ('978-2345678901', 'On Hold', 'Damaged Since 2013-06-02');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Status (bookISBN, bookStatus, statusComment)
-> VALUES ('978-3456789012', 'Returned', 'In Processing Since 2013-08-03');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Status (bookISBN, bookStatus, statusComment)
-> VALUES ('978-4567890123', 'Lost/Stolen', 'Not Found Since 2013-01-03');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM Status;
+-----+-----+-----+
| bookISBN | bookStatus | statusComment |
+-----+-----+-----+
| 978-0123456789 | Available | In Stock Since 2013-06-01 |
| 978-1234567890 | Not Available | Borrowed until 2013-08-09 |
| 978-2345678901 | On Hold | Damaged Since 2013-06-02 |
| 978-3456789012 | Returned | In Processing Since 2013-08-03 |
| 978-4567890123 | Lost/Stolen | Not Found Since 2013-01-03 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> INSERT INTO Library (libraryName, libraryAddress, libraryPhoneNumber)
-> VALUES ('Tampa Public Library', '112233 Tampa RD, Tampa, FL 112233', '813-112-2334');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Library (libraryName, libraryAddress, libraryPhoneNumber)
-> VALUES ('Clearwater Public Library', '221133 Clearwater RD, Clearwater, FL 221133', '813-221-1334');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Library (libraryName, libraryAddress, libraryPhoneNumber)
-> VALUES ('Oldsmar Public Library', '332211 Oldsmar RD, Oldsmar, FL 332211', '813-332-2113');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Library (libraryName, libraryAddress, libraryPhoneNumber)
-> VALUES ('Palm Harbor Public Library', '123123 Palm RD, Palm Harbor, FL 123123', '813-123-1233');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Library (libraryName, libraryAddress, libraryPhoneNumber)
-> VALUES ('Brandon Public Library', '321321 Brandon RD, Brandon, FL 321321', '813-321-3214');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM Library;
+-----+-----+-----+
| libraryName | libraryAddress | libraryPhoneNumber |
+-----+-----+-----+
```

libraryName	libraryAddress	libraryPhoneNumber
Tampa Public Library	112233 Tampa RD, Tampa, FL 112233	813-112-2334
Clearwater Public Library	221133 Clearwater RD, Clearwater, FL 221133	813-221-1334
Oldsmar Public Library	332211 Oldsmar RD, Oldsmar, FL 332211	813-332-2113
Palm Harbor Public Library	123123 Palm RD, Palm Harbor, FL 123123	813-123-1233
Brandon Public Library	321321 Brandon RD, Brandon, FL 321321	813-321-3214

5 rows in set (0.00 sec)

```
mysql> INSERT INTO Checkout (cardNumber, bookISBN, confirmationNumber, borrowDate, returnDate)
-> VALUES (11111, '978-0123456789', 1234567, '2013-05-01', '2013-06-01');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Checkout (cardNumber, bookISBN, confirmationNumber, borrowDate, returnDate)
-> VALUES (22222, '978-1234567890', 2345678, '2013-07-09', '2013-08-09');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO Checkout (cardNumber, bookISBN, confirmationNumber, borrowDate, returnDate)
-> VALUES (33333, '978-2345678901', 3456789, '2013-05-02', '2013-06-02');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Checkout (cardNumber, bookISBN, confirmationNumber, borrowDate, returnDate)
-> VALUES (44444, '978-3456789012', 4567890, '2013-07-03', '2013-08-03');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Checkout (cardNumber, bookISBN, confirmationNumber, borrowDate, returnDate)
-> VALUES (55555, '978-4567890123', 5678901, '2012-12-01', NULL);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM Checkout;
```

cardNumber	bookISBN	confirmationNumber	borrowDate	returnDate
11111	978-0123456789	1234567	2013-05-01	2013-06-01
22222	978-1234567890	2345678	2013-07-09	2013-08-09
33333	978-2345678901	3456789	2013-05-02	2013-06-02
44444	978-3456789012	4567890	2013-07-03	2013-08-03
55555	978-4567890123	5678901	2012-12-01	NULL

5 rows in set (0.00 sec)

8. Submit a “How’s it going” statement

I have created all the required table, inserted all the data type values, and normalized all of the tables after added all of the required foreign keys...etc. Here are the screen shots of the tables showing the table structures such as columns or attributes and the attribute values for each table.

```
mysql> SHOW COLUMNS IN Book;
```

Field	Type	Null	Key	Default	Extra
bookISBN	varchar(16)	NO	PRI		
libraryName	char(64)	NO	MUL		
bookTitle	varchar(64)	NO			

```

| bookAuthor | char(32) | NO | | | |
| bookYear   | int(11)  | NO | | | |
| bookPrice  | double   | NO | | | |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

mysql> SELECT * FROM Book;

```

+-----+-----+-----+-----+-----+-----+
| bookISBN      | libraryName      | bookTitle      | bookAuthor      | bookYear | bookPrice |
+-----+-----+-----+-----+-----+-----+
| 978-0123456789 | Tampa Public Library | Computer Science | Alex Loui      | 2000     | 10        |
| 978-1234567890 | Clearwater Public Library | Computer Engineering | Michael Lee    | 2003     | 30        |
| 978-2345678901 | Oldsmar Public Library | Mathematics      | George White   | 2005     | 50        |
| 978-3456789012 | Palm Harbor Public Library | Physics          | Lucy Black     | 2007     | 70        |
| 978-4567890123 | Brandon Public Library | Public Speech     | Dave Sink      | 2009     | 90        |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

mysql> SHOW COLUMNS IN Member;

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| memberID   | int(11)   | NO   | PRI |         |       |
| memberFirstName | char(16)  | NO   |     |         |       |
| memberLastName | char(16)  | NO   |     |         |       |
| memberAddress  | varchar(64) | NO   |     |         |       |
| memberPhoneNumber | char(16)  | NO   |     |         |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

mysql> SELECT * FROM Member;

```

+-----+-----+-----+-----+-----+-----+
| memberID | memberFirstName | memberLastName | memberAddress      | memberPhoneNumber |
+-----+-----+-----+-----+-----+-----+
| 12345 | Amir          | Yousef         | 12345 Tree Ave, Tampa, FL 12345 | 813-123-4567      |
| 23456 | Cathy          | Powel          | 23456 Citrus Ave, Clearwater, FL 23456 | 813-234-5678      |
| 34567 | April          | Silver         | 34567 Tree Ave, Oldsmar, FL 34567 | 813-345-6789      |
| 45678 | Justin         | Jones          | 45678 Book Ave, Palm Harbor, FL 45678 | 813-456-7890      |
| 56789 | Mary          | Edward         | 56789 Deer Ave, Brandon, FL 56789 | 813-567-8910      |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

mysql> SHOW COLUMNS IN LibraryCard;

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| cardNumber | int(11)   | NO   | PRI |         |       |

```

```
| memberID | int(11) | NO | MUL | | |
| expireDate | date | YES | | NULL | |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM LibraryCard;
+-----+-----+-----+
| cardNumber | memberID | expireDate |
+-----+-----+-----+
| 11111 | 12345 | 2016-01-01 |
| 22222 | 23456 | 2017-01-01 |
| 33333 | 34567 | 2018-01-01 |
| 44444 | 45678 | 2019-01-01 |
| 55555 | 56789 | 2020-01-01 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> SHOW COLUMNS IN Status;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| bookISBN | varchar(16) | NO | MUL | | |
| bookStatus | char(16) | NO | PRI | | |
| statusComment | varchar(32) | NO | | | |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM Status;
+-----+-----+-----+
| bookISBN | bookStatus | statusComment |
+-----+-----+-----+
| 978-0123456789 | Available | In Stock Since 2013-06-01 |
| 978-1234567890 | Not Available | Borrowed until 2013-08-09 |
| 978-2345678901 | On Hold | Damaged Since 2013-06-02 |
| 978-3456789012 | Returned | In Processing Since 2013-08-03 |
| 978-4567890123 | Lost/Stolen | Not Found Since 2013-01-03 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> SHOW COLUMNS IN Library;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| libraryName | char(32) | NO | PRI | | |
| libraryAddress | char(64) | NO | | | |
| libraryPhoneNumber | char(16) | NO | | | |
+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

```
mysql> SELECT * FROM Library;
+-----+-----+-----+
| libraryName | libraryAddress | libraryPhoneNumber |
+-----+-----+-----+
```

Tampa Public Library	112233 Tampa RD, Tampa, FL 112233	813-112-2334
Clearwater Public Library	221133 Clearwater RD, Clearwater, FL 221133	813-221-1334
Oldsmar Public Library	332211 Oldsmar RD, Oldsmar, FL 332211	813-332-2113
Palm Harbor Public Library	123123 Palm RD, Palm Harbor, FL 123123	813-123-1233
Brandon Public Library	321321 Brandon RD, Brandon, FL 321321	813-321-3214

5 rows in set (0.00 sec)

```
mysql> SHOW COLUMNS IN Checkout;
```

Field	Type	Null	Key	Default	Extra
cardNumber	int(11)	NO	MUL		
bookISBN	varchar(16)	NO	MUL		
confirmationNumber	int(11)	NO	PRI		
borrowDate	date	YES		NULL	
returnDate	date	YES		NULL	

5 rows in set (0.00 sec)

```
mysql> SELECT * FROM Checkout;
```

cardNumber	bookISBN	confirmationNumber	borrowDate	returnDate
11111	978-0123456789	1234567	2013-05-01	2013-06-01
22222	978-1234567890	2345678	2013-07-09	2013-08-09
33333	978-2345678901	3456789	2013-05-02	2013-06-02
44444	978-3456789012	4567890	2013-07-03	2013-08-03
55555	978-4567890123	5678901	2012-12-01	NULL

5 rows in set (0.00 sec)

Submission 3

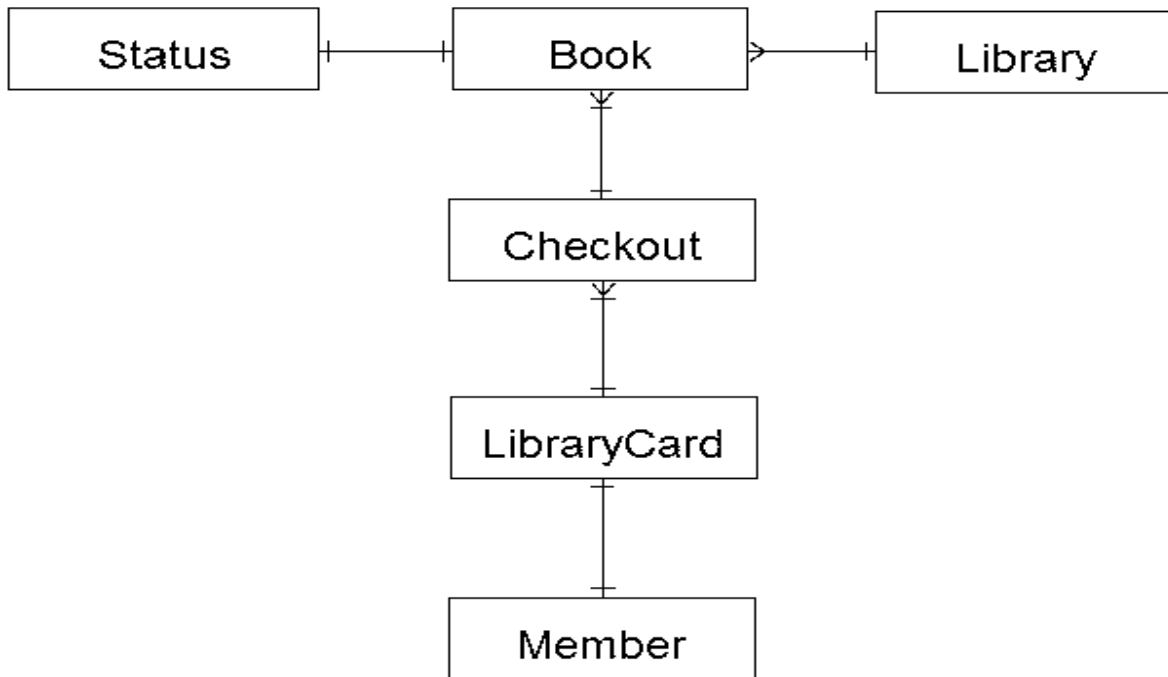
1. Title

Library Management System (LMS)

2. Outline

LMS is a database system that will store and organize the library's data. The system can keep track the members, the libraries, the books, and their status. The books identified by their ISBN as a book unique ID number. Each member will sign up to get a library card therefore each member will provide his/her first name, last name, address, and phone number. The library will issue a library card with a unique member ID number for each person. The checkout operation will require both the member ID number and the library card number. Each library has a unique name, address and a phone number.

3. ER Diagram



4. Table schemas

Member (memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber)

Book (bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice)

Checkout (cardNumber, bookISBN, confirmationNumber, borrowDate, returnDate)

Library (libraryName, libraryAddress, libraryPhoneNumber)

LibraryCard (cardNumber, memberID, expireDate)

Status (bookISBN, bookStatus, statusComment)

5. Explanation of tables are normalized

1. The tables are in the First Normal Form (1NF) since they do not contain any non-atomic value such as list, array, or other composite data type and no repeated attributes from any table.
2. The tables are in the Second Normal Form (2NF) since they do not contain any partial key dependency.
3. The tables are in the Third Normal Form (3NF) since they do not contain any transitive dependencies.
4. The tables are in the Boyce-Codd Normal Form (BCNF) since every determinant is a candidate key.
5. The tables are in the Fourth Normal Form (4NF) since they do not contain more than one independent multi-valued dependency in each table.
6. The tables are in the Fifth Normal Form (5NF) they do not contain any cyclic dependencies.

6. DDL to create the tables

DROP TABLE IF EXISTS Member;

```

CREATE TABLE Member (memberID INTEGER NOT NULL,
                      memberFirstName CHAR (16) NOT NULL,
                      memberLastName CHAR (16) NOT NULL,
                      memberAddress VARCHAR (64) NOT NULL,
                      memberPhoneNumber CHAR (16) NOT NULL,
                      PRIMARY KEY (memberID));
  
```

DROP TABLE IF EXISTS Book;

```

CREATE TABLE Book (bookISBN VARCHAR (16) NOT NULL,
                   libraryName CHAR (64) NOT NULL,
                   bookTitle VARCHAR (64) NOT NULL,
                   bookAuthor CHAR (32) NOT NULL,
  
```

```
bookYear INTEGER NOT NULL,  
bookPrice DOUBLE NOT NULL,  
PRIMARY KEY (bookISBN),  
FOREIGN KEY (libraryName) REFERENCES Library);
```

DROP TABLE IF EXISTS Checkout;

```
CREATE TABLE Checkout (cardNumber INTEGER NOT NULL,  
bookISBN VARCHAR (16) NOT NULL,  
confirmationNumber INTEGER NOT NULL,  
borrowDate DATE NULL,  
returnDate DATE NULL,  
PRIMARY KEY (confirmationNumber),  
FOREIGN KEY (bookISBN) REFERENCES Book,  
FOREIGN KEY (cardNumber) REFERENCES LibraryCard);
```

DROP TABLE IF EXISTS Library;

```
CREATE TABLE Library (libraryName VARCHAR (64) NOT NULL,  
libraryAddress VARCHAR (64) NOT NULL,  
libraryPhoneNumber CHAR(16) NOT NULL,  
PRIMARY KEY (libraryName));
```

DROP TABLE IF EXISTS LibraryCard;

```
CREATE TABLE LibraryCard (cardNumber INTEGER NOT NULL,  
memberID INTEGER NOT NULL,  
expireDate DATE NULL,  
PRIMARY KEY (cardNumber),  
FOREIGN KEY (memberID) REFERENCES Member);
```

DROP TABLE IF EXISTS Status;

```
CREATE TABLE Status (bookISBN VARCHAR (16) NOT NULL,  
bookStatus CHAR (16) NOT NULL,  
statusComment VARCHAR (32) NOT NULL,  
PRIMARY KEY (bookStatus),  
FOREIGN KEY (bookISBN) REFERENCES Book);
```

7-10 Query, Interacting, Screen shots, and Reports

Insert: Add More Members

```
INSERT INTO Member  
(memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber) VALUES  
(11111, 'Ihab', 'Rizk', '11111 Tree Ave, Tampa, FL 11111', '813-111-1111');
```

```
INSERT INTO Member  
(memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber) VALUES  
(22222, 'Nancy', 'Gergis', '22222 Tree Ave, Tampa, FL 22222', '813-222-2222');
```

```
INSERT INTO Member  
(memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber) VALUES  
(33333, 'Pola', 'Abnop', '33333 Tree Ave, Tampa, FL 33333', '813-333-3333');
```

```
mysql> INSERT INTO Member
-> (memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber)
VALUES
-> (11111, 'Ihab', 'Rizk', '11111 Tree Ave, Tampa, FL 11111', '813-111-1111');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Member
-> (memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber)
VALUES
-> (22222, 'Nancy', 'Gergis', '22222 Tree Ave, Tampa, FL 22222', '813-222-2222');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Member
-> (memberID, memberFirstName, memberLastName, memberAddress, memberPhoneNumber)
VALUES
-> (33333, 'Pola', 'Abnop', '33333 Tree Ave, Tampa, FL 33333', '813-333-3333');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM Member;
```

memberID	memberFirstName	memberLastName	memberAddress	memberPhoneNumber
12345	Amir	Yousef	12345 Tree Ave, Tampa, FL 12345	813-123-4567
23456	Cathy	Powel	23456 Citrus Ave, Clearwater, FL 23456	813-234-5678
34567	April	Silver	34567 Tree Ave, Oldsmar, FL 34567	813-345-6789
45678	Justin	Jones	45678 Book Ave, Palm Harbor, FL 45678	813-456-7890
56789	Mary	Edward	56789 Deer Ave, Brandon, FL 56789	813-567-8910
11111	Ihab	Rizk	11111 Tree Ave, Tampa, FL 11111	813-111-1111
22222	Nancy	Gergis	22222 Tree Ave, Tampa, FL 22222	813-222-2222
33333	Pola	Abnop	33333 Tree Ave, Tampa, FL 33333	813-333-3333

```
8 rows in set (0.00 sec)
```

Insert: Add More Books

```
INSERT INTO Book
(bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES
('978-1111111111', 'Tampa Public Library', 'Computer Network', 'Alex Loui', '2001', 11.00);
```

```
INSERT INTO Book
(bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES
('978-2222222222', 'Tampa Public Library', 'Computer Hardware', 'Alex Loui', '2002', 12.00);
```

```
INSERT INTO Book
(bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES
('978-3333333333', 'Tampa Public Library', 'Computer Architecture', 'Alex Loui', '2003', 13.00);
```

```
mysql> INSERT INTO Book
-> (bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES
-> ('978-1111111111', 'Tampa Public Library', 'Computer Network', 'Alex Loui', '2001', 11.00);
Query OK, 1 row affected (0.00 sec)
mysql> INSERT INTO Book
```



```
-> (bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES
-> ('978-2222222222', 'Tampa Public Library', 'Computer Hardware', 'Alex Loui', '2002', 12.00);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Book
-> (bookISBN, libraryName, bookTitle, bookAuthor, bookYear, bookPrice) VALUES
-> ('978-3333333333', 'Tampa Public Library', 'Computer Architecture', 'Alex Loui', '2003', 13.00);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM Book;
+-----+-----+-----+-----+-----+
| bookISBN | libraryName | bookTitle | bookAuthor | bookYear | bookPrice |
+-----+-----+-----+-----+-----+
| 978-0123456789 | Tampa Public Library | Computer Science | Alex Loui | 2000 | 10 |
| 978-1234567890 | Clearwater Public Library | Computer Engineering | Michael Lee | 2003 | 30 |
| 978-2345678901 | Oldsmar Public Library | Mathematics | George White | 2005 | 50 |
| 978-3456789012 | Palm Harbor Public Library | Physics | Lucy Black | 2007 | 70 |
| 978-4567890123 | Brandon Public Library | Public Speech | Dave Sink | 2009 | 90 |
| 978-1111111111 | Tampa Public Library | Computer Network | Alex Loui | 2001 | 11 |
| 978-2222222222 | Tampa Public Library | Computer Hardware | Alex Loui | 2002 | 12 |
| 978-3333333333 | Tampa Public Library | Computer Architecture | Alex Loui | 2003 | 13 |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

Insert: Add More Library Cards

```
INSERT INTO LibraryCard (cardNumber, memberID, expireDate)
VALUES (66666, 15973, '2014-01-01');
```

```
INSERT INTO LibraryCard (cardNumber, memberID, expireDate)
VALUES (77777, 35719, '2015-01-01');
```

```
INSERT INTO LibraryCard (cardNumber, memberID, expireDate)
VALUES (88888, 85246, '2016-01-01');
```

```
mysql> INSERT INTO LibraryCard (cardNumber, memberID, expireDate)
-> VALUES (66666, 15973, '2014-01-01');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO LibraryCard (cardNumber, memberID, expireDate)
-> VALUES (77777, 35719, '2015-01-01');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO LibraryCard (cardNumber, memberID, expireDate)
-> VALUES (88888, 85246, '2016-01-01');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM LibraryCard;
+-----+-----+-----+
| cardNumber | memberID | expireDate |
+-----+-----+-----+
| 11111 | 12345 | 2016-01-01 |
| 22222 | 23456 | 2017-01-01 |
```

33333	34567	2018-01-01
44444	45678	2019-01-01
55555	56789	2020-01-01
66666	15973	2014-01-01
77777	35719	2015-01-01
88888	85246	2016-01-01

+-----+-----+-----+

8 rows in set (0.00 sec)

Update Existing Members

UPDATE Member

SET memberAddress = '98765 Deer Ave, Tampa, FL 98765'

WHERE memberID = 12345;

mysql> UPDATE Member

-> SET memberAddress = '98765 Deer Ave, Tampa, FL 98765'

-> WHERE memberID = 12345;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT * FROM Member WHERE memberID = 12345;

memberID	memberFirstName	memberLastName	memberAddress	memberPhoneNumber
12345	Amir	Yousef	98765 Deer Ave, Tampa, FL 98765	813-123-4567

1 row in set (0.00 sec)

Successfully updated "memberID 12345" Address from "12345 Tree Ave, Tampa, FL 12345" to "98765 Deer Ave, Tampa, FL 98765". By same way we can update the firstName, lastName, and phoneNumber.

Update Existing Books

UPDATE Book

SET bookPrice = 22.00

WHERE bookISBN = '978-1111111111';

mysql> UPDATE Book

-> SET bookPrice = 22.00

-> WHERE bookISBN = '978-1111111111';

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT * FROM Book WHERE bookISBN = '978-1111111111';

bookISBN	libraryName	bookTitle	bookAuthor	bookYear	bookPrice
978-1111111111	Tampa Public Library	Computer Network	Alex Loui	2001	22

1 row in set (0.00 sec)

Successfully updated “bookISBN = '978-1111111111'” bookPrice from “11” to “22”. By same way, we can update the libraryName, bookTitle, bookAuthor, and bookYear.

Update Existing LibraryCards

```
UPDATE LibraryCard
SET expireDate = '2020-02-02'
WHERE cardNumber = 66666;
```

```
mysql> UPDATE LibraryCard
-> SET expireDate = '2020-02-02'
-> WHERE cardNumber = 66666;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> SELECT * FROM LibraryCard WHERE cardNumber = 66666;
+-----+-----+-----+
| cardNumber | memberID | expireDate |
+-----+-----+-----+
| 66666 | 15973 | 2020-02-02 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Successfully updated “cardNumber = 66666” expireDate from “2014-01-01” to “2020-02-02”. By same way, we can update the memberID.

Delete Existing Members

This is the table before delete:

```
mysql> SELECT * FROM Member;
+-----+-----+-----+-----+-----+
| memberID | memberFirstName | memberLastName | memberAddress | memberPhoneNumber |
+-----+-----+-----+-----+-----+
| 12345 | Amir | Yousef | 98765 Deer Ave, Tampa, FL 98765 | 813-123-4567 |
| 23456 | Cathy | Powel | 23456 Citrus Ave, Clearwater, FL 23456 | 813-234-5678 |
| 34567 | April | Silver | 34567 Tree Ave, Oldsmar, FL 34567 | 813-345-6789 |
| 45678 | Justin | Jones | 45678 Book Ave, Palm Harbor, FL 45678 | 813-456-7890 |
| 56789 | Mary | Edward | 56789 Deer Ave, Brandon, FL 56789 | 813-567-8910 |
| 11111 | Ihab | Rizk | 11111 Tree Ave, Tampa, FL 11111 | 813-111-1111 |
| 22222 | Nancy | Gergis | 22222 Tree Ave, Tampa, FL 22222 | 813-222-2222 |
| 33333 | Pola | Abnop | 33333 Tree Ave, Tampa, FL 33333 | 813-333-3333 |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

Now apply delete:

```
DELETE FROM Member
WHERE memberID = 11111;
```

```
DELETE FROM Member
WHERE memberID = 22222;
```

```
DELETE FROM Member
WHERE memberID = 33333;
```

```
mysql> DELETE FROM Member
-> WHERE memberID = 11111;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> DELETE FROM Member
-> WHERE memberID = 22222;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> DELETE FROM Member
-> WHERE memberID = 33333;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM Member;
```

```
+-----+-----+-----+-----+
| memberID | memberFirstName | memberLastName | memberAddress |
memberPhoneNumber |
+-----+-----+-----+-----+
| 12345 | Amir | Yousef | 98765 Deer Ave, Tampa, FL 98765 | 813-123-4567 |
| 23456 | Cathy | Powel | 23456 Citrus Ave, Clearwater, FL 23456 | 813-234-5678 |
| 34567 | April | Silver | 34567 Tree Ave, Oldsmar, FL 34567 | 813-345-6789 |
| 45678 | Justin | Jones | 45678 Book Ave, Palm Harbor, FL 45678 | 813-456-7890 |
| 56789 | Mary | Edward | 56789 Deer Ave, Brandon, FL 56789 | 813-567-8910 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Successfully deleted 3 members (rows/tuples) from the table.

Delete Existing Books

This is the table before delete:

```
mysql> SELECT * FROM Book;
```

```
+-----+-----+-----+-----+-----+-----+
| bookISBN | libraryName | bookTitle | bookAuthor | bookYear | bookPrice |
+-----+-----+-----+-----+-----+-----+
| 978-0123456789 | Tampa Public Library | Computer Science | Alex Loui | 2000 | 10 |
| 978-1234567890 | Clearwater Public Library | Computer Engineering | Michael Lee | 2003 | 30 |
| 978-2345678901 | Oldsmar Public Library | Mathematics | George White | 2005 | 50 |
| 978-3456789012 | Palm Harbor Public Library | Physics | Lucy Black | 2007 | 70 |
| 978-4567890123 | Brandon Public Library | Public Speech | Dave Sink | 2009 | 90 |
| 978-1111111111 | Tampa Public Library | Computer Network | Alex Loui | 2001 | 22 |
| 978-2222222222 | Tampa Public Library | Computer Hardware | Alex Loui | 2002 | 12 |
| 978-3333333333 | Tampa Public Library | Computer Architecture | Alex Loui | 2003 | 13 |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

```
DELETE FROM Book
WHERE bookISBN = '978-1111111111';
```

```
DELETE FROM Book
WHERE bookISBN = '978-2222222222';
```

```
DELETE FROM Book
WHERE bookISBN = '978-3333333333';
```

```
mysql> DELETE FROM Book
-> WHERE bookISBN = '978-1111111111';
Query OK, 1 row affected (0.00 sec)
```

```
mysql>
mysql> DELETE FROM Book
-> WHERE bookISBN = '978-2222222222';
Query OK, 1 row affected (0.00 sec)
```

```
mysql>
mysql> DELETE FROM Book
-> WHERE bookISBN = '978-3333333333';
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM Book;
```

bookISBN	libraryName	bookTitle	bookAuthor	bookYear	bookPrice
978-0123456789	Tampa Public Library	Computer Science	Alex Loui	2000	10
978-1234567890	Clearwater Public Library	Computer Engineering	Michael Lee	2003	30
978-2345678901	Oldsmar Public Library	Mathematics	George White	2005	50
978-3456789012	Palm Harbor Public Library	Physics	Lucy Black	2007	70
978-4567890123	Brandon Public Library	Public Speech	Dave Sink	2009	90

5 rows in set (0.00 sec)

Successfully deleted 3 books out of book table, bookISBN = 978-1111111111, 978-2222222222, and 978-3333333333.

Delete Existing LibraryCards

This is the table before delete:

```
mysql> SELECT * FROM LibraryCard ;
```

cardNumber	memberID	expireDate
11111	12345	2016-01-01
22222	23456	2017-01-01
33333	34567	2018-01-01
44444	45678	2019-01-01
55555	56789	2020-01-01
66666	15973	2020-02-02
77777	35719	2015-01-01
88888	85246	2016-01-01

8 rows in set (0.00 sec)

```
DELETE FROM LibraryCard
WHERE cardNumber = 66666;
```

```
DELETE FROM LibraryCard
WHERE cardNumber = 77777;
```

```
DELETE FROM LibraryCard
WHERE cardNumber = 88888;
```

```
mysql> DELETE FROM LibraryCard
-> WHERE cardNumber = 66666;
Query OK, 1 row affected (0.00 sec)
mysql> DELETE FROM LibraryCard
-> WHERE cardNumber = 77777;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> DELETE FROM LibraryCard
-> WHERE cardNumber = 88888;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM LibraryCard ;
+-----+-----+-----+
| cardNumber | memberID | expireDate |
+-----+-----+-----+
| 11111 | 12345 | 2016-01-01 |
| 22222 | 23456 | 2017-01-01 |
| 33333 | 34567 | 2018-01-01 |
| 44444 | 45678 | 2019-01-01 |
| 55555 | 56789 | 2020-01-01 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Successfully deleted library cards number 66666, 77777, 88888.

Produce Reports

To see all members information:

```
mysql> SELECT * FROM Member;
+-----+-----+-----+-----+-----+
| memberID | memberFirstName | memberLastName | memberAddress |
memberPhoneNumber |
+-----+-----+-----+-----+-----+
| 12345 | Amir | Yousef | 98765 Deer Ave, Tampa, FL 98765 | 813-123-4567 |
| 23456 | Cathy | Powel | 23456 Citrus Ave, Clearwater, FL 23456 | 813-234-5678 |
| 34567 | April | Silver | 34567 Tree Ave, Oldsmar, FL 34567 | 813-345-6789 |
| 45678 | Justin | Jones | 45678 Book Ave, Palm Harbor, FL 45678 | 813-456-7890 |
| 56789 | Mary | Edward | 56789 Deer Ave, Brandon, FL 56789 | 813-567-8910 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

To find out how many members:

```
mysql> SELECT COUNT(*) FROM Member;
+-----+
| COUNT(*) |
+-----+
|      5 |
+-----+
1 row in set (0.00 sec)
```

To sort the members by their last name:

```
mysql> SELECT *
-> FROM Member
-> GROUP BY memberLastName;
```

memberID	memberFirstName	memberLastName	memberAddress	memberPhoneNumber
56789	Mary	Edward	56789 Deer Ave, Brandon, FL 56789	813-567-8910
45678	Justin	Jones	45678 Book Ave, Palm Harbor, FL 45678	813-456-7890
23456	Cathy	Powel	23456 Citrus Ave, Clearwater, FL 23456	813-234-5678
34567	April	Silver	34567 Tree Ave, Oldsmar, FL 34567	813-345-6789
12345	Amir	Yousef	98765 Deer Ave, Tampa, FL 98765	813-123-4567

5 rows in set (0.00 sec)

To sort the members by their first name:

```
mysql> SELECT *
-> FROM Member
-> GROUP BY memberFirstName;
```

memberID	memberFirstName	memberLastName	memberAddress	memberPhoneNumber
12345	Amir	Yousef	98765 Deer Ave, Tampa, FL 98765	813-123-4567
34567	April	Silver	34567 Tree Ave, Oldsmar, FL 34567	813-345-6789
23456	Cathy	Powel	23456 Citrus Ave, Clearwater, FL 23456	813-234-5678
45678	Justin	Jones	45678 Book Ave, Palm Harbor, FL 45678	813-456-7890
56789	Mary	Edward	56789 Deer Ave, Brandon, FL 56789	813-567-8910

5 rows in set (0.00 sec)

To find the members with memberID number more than 30000:

```
mysql> SELECT *
-> FROM Member
-> GROUP BY memberFirstName
-> HAVING memberID > 30000;
```

```

+-----+-----+-----+-----+
| memberID | memberFirstName | memberLastName | memberAddress |
memberPhoneNumber |
+-----+-----+-----+-----+
| 34567 | April | Silver | 34567 Tree Ave, Oldsmar, FL 34567 | 813-345-6789 |
| 45678 | Justin | Jones | 45678 Book Ave, Palm Harbor, FL 45678 | 813-456-7890 |
| 56789 | Mary | Edward | 56789 Deer Ave, Brandon, FL 56789 | 813-567-8910 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

To search for only available books in stock:

```

SELECT *
FROM Status
GROUP BY bookISBN
HAVING bookStatus = 'Available';

```

```

mysql> SELECT *
-> FROM Status
-> GROUP BY bookISBN
-> HAVING bookStatus = 'Available';
+-----+-----+-----+
| bookISBN | bookStatus | statusComment |
+-----+-----+-----+
| 978-0123456789 | Available | In Stock Since 2013-06-01 |
+-----+-----+-----+
1 row in set (0.00 sec)

```

To find a book about Computer Science and sort by the library location:

```

SELECT *
FROM Book
GROUP BY libraryName
HAVING bookTitle = 'Computer Science';

```

```

mysql> SELECT *
-> FROM Book
-> GROUP BY libraryName
-> HAVING bookTitle = 'Computer Science';
+-----+-----+-----+-----+-----+-----+
| bookISBN | libraryName | bookTitle | bookAuthor | bookYear | bookPrice |
+-----+-----+-----+-----+-----+-----+
| 978-0123456789 | Tampa Public Library | Computer Science | Alex Loui | 2000 | 10 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```


This project was a simple SQL database that enters books, members' information, and libraries information into database and then allowing search, update, and/or delete for these data. Created the tables and make sure that they are all having unique identifiers as a primary key and also related to each other and linked by foreign keys. Performed most of the functions we have learned during the semester such insert, update, delete, group by, and having...etc.

In my opinion the project is successful since it does all of the functions that we need to manage simple book library but the system able to be bigger than that and accept more relations or tables and more functions to perform more functions such as paying fees for late return books. Personally, I like the COP4710 class and feels like this is the most needed class in the future careers and helped me to think about how to handle problems and think as a programmers. Hoping the project will be acceptable from my instructor, and at the mean time, I would like to thank him for all of his help and his support.

Thank you,

Amir Yousef
Summer13
COP 4710
Project