LIBRARY MANAGEMENT SYSTEM

Project for SQL Module

By Nihal singh

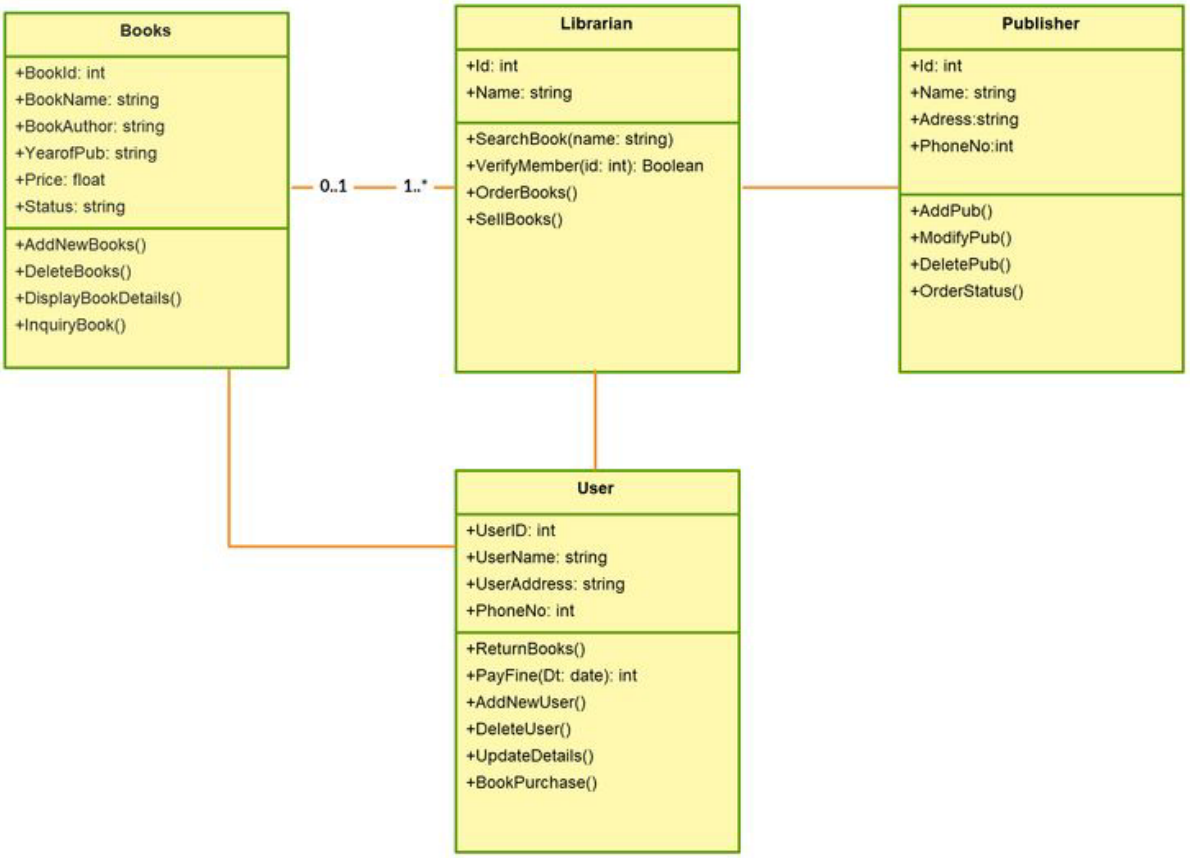
>Introduction

In this article, we will build Library Management System using MYSQL. We will build the database, which includes tables. Imagine that you go to the library, take a book, and just enter that book into the computer rather than entering your details and book details in a register. Isn't it easy and convenient? Here comes the library management system. This system is handy for people going to the library and searching for their desired book and for the librarian to guide them and take care of these things. It ensures that everything works in systematic order, given that each person taking or returning books needs to enter the record in the system.

>Purpose of Library Management System

Library Management System is a system that shows all the available books and their count and also books taken by people, the date on which they took that particular book, expected date of return, book details, borrower details, and so on. Everything will be crystal clear. There will be no ambiguity. It will be beneficial for both students and librarians.

>ER Daigram:-



COMMANDS:-

```
CREATE DATABASE LibraryManagement;
CREATE TABLE LibraryManagement.tbl_publisher (
publisher_PublisherName VARCHAR(100) PRIMARY KEY NOT NULL,
publisher_PublisherAddress VARCHAR(200) NOT NULL,
publisher_PublisherPhone VARCHAR(50) NOT NULL
);
CREATE TABLE LibraryManagement.tbl_book (
   book_BookID INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
   book_Title VARCHAR(100) NOT NULL,
   book_PublisherName VARCHAR(100) NOT NULL,
```

```
   CONSTRAINT fk_publisher_name1 FOREIGN KEY (book_PublisherName)
REFERENCES tbl_publisher(publisher_PublisherName)
      ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE LibraryManagement.tbl_library_branch (
library_branch_BranchID INT AUTO_INCREMENT  PRIMARY KEY NOT NULL ,
library_branch_BranchName VARCHAR(100) NOT NULL,
library_branch_BranchAddress VARCHAR(200) NOT NULL
);

CREATE TABLE LibraryManagement.tbl_book_copies (
   book_copies_CopiesID INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
   book_copies_BookID INT NOT NULL,
   book_copies_BranchID INT NOT NULL,
   book_copies_No_Of_Copies INT NOT NULL,
   CONSTRAINT fk_book_id2 FOREIGN KEY (book_copies_BookID) REFERENCES
tbl_book(book_BookID) ON UPDATE CASCADE ON DELETE CASCADE,
   CONSTRAINT fk_branch_id2 FOREIGN KEY (book_copies_BranchID)
REFERENCES tbl_library_branch(library_branch_BranchID) ON UPDATE
CASCADE ON DELETE CASCADE
);

CREATE TABLE LibraryManagement.tbl_borrower (
borrower_CardNo INT AUTO_INCREMENT PRIMARY KEY NOT NULL ,
```

borrower_BorrowerName VARCHAR(100) NOT NULL,

borrower_BorrowerAddress VARCHAR(200) NOT NULL,

borrower_BorrowerPhone VARCHAR(50) NOT NULL

);

SELECT * FROM LibraryManagement.tbl_borrower;

```
MariaDB [(none)]> SELECT * FROM LibraryManagement.tbl_borrower;
+-----------------+-----------------------+----------------------------------------+-----------------------+
| borrower_CardNo | borrower_BorrowerName | borrower_BorrowerAddress               | borrower_BorrowerPhone |
+-----------------+-----------------------+----------------------------------------+-----------------------+
|               1 | Joe Smith             | 1321 4th Street, New York, NY 10014    | 212-312- 1234          |
|               2 | Jane Smith            | 1321 4th Street, New York, NY 10014    | 212-931- 4124          |
|               3 | Tom Li                | 981 Main Street, Ann Arbor, MI 48104   | 734-902-7455           |
|               4 | Angela Thompson       | 2212 Green Avenue, Ann Arbor, MI 48104 | 313-591-2122           |
|               5 | Harry Emnace          | 121 Park Drive, Ann Arbor, MI 48104    | 412-512-5522           |
|               6 | Tom Haverford         | 23 75th Street, New York, NY 10014     | 212-631-3418           |
|               7 | Haley Jackson         | 231 52nd Avenue New York, NY 10014     | 212-419-9935           |
|               8 | Michael Horford       | 653 Glen Avenue, Ann Arbor, MI 48104   | 734-998-1513           |
+-----------------+-----------------------+----------------------------------------+-----------------------+
```

CREATE TABLE LibraryManagement.tbl_book_loans (

book_loans_LoansID INT AUTO_INCREMENT PRIMARY KEY NOT NULL,

book_loans_BookID INT NOT NULL,

book_loans_BranchID INT NOT NULL,

book_loans_CardNo INT NOT NULL,

book_loans_DateOut DATE NOT NULL,

book_loans_DueDate DATE NOT NULL,

CONSTRAINT fk_book_id1 FOREIGN KEY (book_loans_BookID) REFERENCES tbl_book(book_BookID) ON UPDATE CASCADE ON DELETE CASCADE,

CONSTRAINT fk_branch_id1 FOREIGN KEY (book_loans_BranchID) REFERENCES tbl_library_branch(library_branch_BranchID) ON UPDATE CASCADE ON DELETE CASCADE,

CONSTRAINT fk_cardno FOREIGN KEY (book_loans_CardNo) REFERENCES tbl_borrower(borrower_CardNo) ON UPDATE CASCADE ON DELETE CASCADE

);

```
CREATE TABLE LibraryManagement.tbl_book_authors (
   book_authors_AuthorID INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
   book_authors_BookID INT NOT NULL,
   book_authors_AuthorName VARCHAR(50) NOT NULL,
   CONSTRAINT fk_book_id3 FOREIGN KEY (book_authors_BookID)
REFERENCES tbl_book(book_BookID) ON UPDATE CASCADE ON DELETE
CASCADE
);


INSERT INTO LibraryManagement.tbl_publisher
(publisher_PublisherName, publisher_PublisherAddress,
publisher_PublisherPhone)
VALUES
('DAW Books','375 Hudson Street, New York, NY 10014','212-366-2000'),
('Viking','375 Hudson Street, New York, NY 10014','212-366-2000'),
('Signet Books','375 Hudson Street, New York, NY 10014','212-366-2000'),
('Chilton Books','Not Available','Not Available'),
('George Allen & Unwin','83 Alexander Ln, Crows Nest NSW 2065,
Australia','+61-2-8425-0100'),
('Alfred A. Knopf','The Knopf Doubleday Group Domestic Rights, 1745
Broadway,
New York, NY 10019','212-940-7390'),
('Bloomsbury','Bloomsbury Publishing Inc., 1385 Broadway, 5th Floor, New
York,
```

NY 10018','212-419-5300'),

('Shinchosa','Oga Bldg. 8, 2-5-4 Sarugaku-cho, Chiyoda-ku, Tokyo 101-0064

Japan','+81-3-5577-6507'),

('Harper and Row','HarperCollins Publishers, 195 Broadway, New York, NY

10007','212-207-7000'),

('Pan Books','175 Fifth Avenue, New York, NY 10010','646-307-5745'),

('Chalto & Windus','375 Hudson Street, New York, NY 10014','212-366-2000'),

('Harcourt Brace Jovanovich','3 Park Ave, New York, NY 10016','212-420-5800'),

('W.W. Norton',' W. W. Norton & Company, Inc., 500 Fifth Avenue, New York, New

York 10110','212-354-5500'),

('Scholastic','557 Broadway, New York, NY 10012','800-724-6527'),

('Bantam','375 Hudson Street, New York, NY 10014','212-366-2000'),

('Picador USA','175 Fifth Avenue, New York, NY 10010','646-307-5745');

SELECT * FROM LibraryManagement.tbl_publisher;

```
MariaDB [(none)]> SELECT * FROM LibraryManagement.tbl_publisher;
+--------------------------+-------------------------------------------------------------------------------+--------------------------+
| publisher_PublisherName  | publisher_PublisherAddress                                                    | publisher_PublisherPhone |
+--------------------------+-------------------------------------------------------------------------------+--------------------------+
| Alfred A. Knopf          | The Knopf Doubleday Group Domestic Rights, 1745 Broadway,
New York, NY 10019 | 212-940-7390           |
| Bantam                   | 375 Hudson Street, New York, NY 10014                                         | 212-366-2000             |
| Bloomsbury               | Bloomsbury Publishing Inc., 1385 Broadway, 5th Floor, New York,
NY 10018     | 212-419-5300          |
| Chalto & Windus          | 375 Hudson Street, New York, NY 10014                                         | 212-366-2000             |
| Chilton Books            | Not Available                                                                 | Not Available            |
| DAW Books                | 375 Hudson Street, New York, NY 10014                                         | 212-366-2000             |
| George Allen & Unwin     | 83 Alexander Ln, Crows Nest NSW 2065,
Australia                          | +61-2-8425-0100          |
| Harcourt Brace Jovanovich | 3 Park Ave, New York, NY 10016                                                | 212-420-5800             |
| Harper and Row           | HarperCollins Publishers, 195 Broadway, New York, NY
10007                | 212-207-7000           |
| Pan Books                | 175 Fifth Avenue, New York, NY 10010                                          | 646-307-5745             |
| Picador USA              | 175 Fifth Avenue, New York, NY 10010                                          | 646-307-5745             |
| Scholastic               | 557 Broadway, New York, NY 10012                                              | 800-724-6527             |
| Shinchosa                | Oga Bldg. 8, 2-5-4 Sarugaku-cho, Chiyoda-ku, Tokyo 101-0064
Japan           | +81-3-5577-6507          |
| Signet Books             | 375 Hudson Street, New York, NY 10014                                         | 212-366-2000             |
| Viking                   | 375 Hudson Street, New York, NY 10014                                         | 212-366-2000             |
| W.W. Norton              |  W. W. Norton & Company, Inc., 500 Fifth Avenue, New York, New
York 10110    | 212-354-5500           |
+--------------------------+-------------------------------------------------------------------------------+--------------------------+
```

```
INSERT INTO LibraryManagement.tbl_book
(book_Title, book_PublisherName)
VALUES
('The Name of the Wind', 'DAW Books'),
('It', 'Viking'),
('The Green Mile', 'Signet Books'),
('Dune', 'Chilton Books'),
('The Hobbit', 'George Allen & Unwin'),
('Eragon', 'Alfred A. Knopf'),
('A Wise Mans Fear', 'DAW Books'),
('Harry Potter and the Philosophers Stone', 'Bloomsbury'),
('Hard Boiled Wonderland and The End of the World', 'Shinchosa'),
('The Giving Tree', 'Harper and Row'),
('The Hitchhikers Guide to the Galaxy', 'Pan Books'),
('Brave New World', 'Chalto & Windus'),
('The Princess Bride', 'Harcourt Brace Jovanovich'),
('Fight Club', 'W.W. Norton'),
('Holes', 'Scholastic'),
('Harry Potter and the Chamber of Secrets', 'Bloomsbury'),
('Harry Potter and the Prisoner of Azkaban', 'Bloomsbury'),
('The Fellowship of the Ring', 'George Allen & Unwin'),
('A Game of Thrones', 'Bantam'),
('The Lost Tribe', 'Picador USA');
```

SELECT * FROM LibraryManagement.tbl_book WHERE book_PublisherName = 'George Allen & Unwin';

book_BookID	book_Title	book_PublisherName
5	The Hobbit	George Allen & Unwin
18	The Fellowship of the Ring	George Allen & Unwin

INSERT INTO LibraryManagement.tbl_library_branch

(library_branch_BranchName, library_branch_BranchAddress)

VALUES

('Sharpstown','32 Corner Road, New York, NY 10012'),

('Central','491 3rd Street, New York, NY 10014'),

('Saline','40 State Street, Saline, MI 48176'),

('Ann Arbor','101 South University, Ann Arbor, MI 48104');

SELECT * FROM LibraryManagement.tbl_library_branch;

```
MariaDB [(none)]> SELECT * FROM LibraryManagement.tbl_library_branch;
```

library_branch_BranchID	library_branch_BranchName	library_branch_BranchAddress
1	Sharpstown	32 Corner Road, New York, NY 10012
2	Central	491 3rd Street, New York, NY 10014
3	Saline	40 State Street, Saline, MI 48176
4	Ann Arbor	101 South University, Ann Arbor, MI 48104

INSERT INTO LibraryManagement.tbl_borrower

(borrower_BorrowerName, borrower_BorrowerAddress,

borrower_BorrowerPhone)

VALUES

('Joe Smith','1321 4th Street, New York, NY 10014','212-312- 1234'),

('Jane Smith','1321 4th Street, New York, NY 10014','212-931- 4124'),

('Tom Li','981 Main Street, Ann Arbor, MI 48104','734-902-7455'),

('Angela Thompson','2212 Green Avenue, Ann Arbor, MI 48104','313-591-2122'),

('Harry Emnace','121 Park Drive, Ann Arbor, MI 48104','412-512-5522'),

('Tom Haverford','23 75th Street, New York, NY 10014','212-631-3418'),

('Haley Jackson','231 52nd Avenue New York, NY 10014','212-419-9935'),

('Michael Horford','653 Glen Avenue, Ann Arbor, MI 48104','734-998-1513');

SELECT * FROM LibraryManagement.tbl_borrower;

```
MariaDB [(none)]> SELECT * FROM LibraryManagement.tbl_borrower;
+-----------------+-----------------------+-----------------------------------------+-----------------------+
| borrower_CardNo | borrower_BorrowerName | borrower_BorrowerAddress                | borrower_BorrowerPhone |
+-----------------+-----------------------+-----------------------------------------+-----------------------+
|               1 | Joe Smith             | 1321 4th Street, New York, NY 10014     | 212-312- 1234          |
|               2 | Jane Smith            | 1321 4th Street, New York, NY 10014     | 212-931- 4124          |
|               3 | Tom Li                | 981 Main Street, Ann Arbor, MI 48104    | 734-902-7455           |
|               4 | Angela Thompson       | 2212 Green Avenue, Ann Arbor, MI 48104  | 313-591-2122           |
|               5 | Harry Emnace          | 121 Park Drive, Ann Arbor, MI 48104     | 412-512-5522           |
|               6 | Tom Haverford         | 23 75th Street, New York, NY 10014      | 212-631-3418           |
|               7 | Haley Jackson         | 231 52nd Avenue New York, NY 10014      | 212-419-9935           |
|               8 | Michael Horford       | 653 Glen Avenue, Ann Arbor, MI 48104    | 734-998-1513           |
+-----------------+-----------------------+-----------------------------------------+-----------------------+
```

SOME QUERIES:-

Here are some important query’s which help of we can do easily

Some changes in our Table / Data.

>How to retrieve table

DESC TABLE NAME;

>How to retrieve all records from table

SELECT * FROM TABLENAME;

>Retrieve one particular or specific column from table

SELECT COLUMN NAME FROM TABLE NAME;

>Record filter in sql by using WHERE CLAUSE

SELECT COLUMN NAME FROM TABLE

WHERE CONDITION;

SELECT * FROM USERS WHERE user_id = 4;

Here are some common camparison operator

= (equal to), != (not equal to), < (less than),

> (greater than), <= (less than or equal to), >= (greater than or equal)

LOGICAL OPERATORS

AND (logical AND), OR (logical OR), NOT (logical NOT)

Ex.

SELECT * FROM Users

where user_id = 3 AND gender = 'female';

>INNER JOIN

An Inner join represented only matching columns in both table

SELECT COLUMNS FROM TABLE 1 INNER JOIN

TABLE 2 ON TABLE 1.COLUMN=TABLE 2.COLUMN;

>OUTER JOIN

An outer join show only mismatching columns in both table

SELECT COLUMNS FROM TABLE 1 OUTER JOIN

TABLE 2 ON TABLE 1.COLUMN=TABLE 2.COLUMN;

OUTER JOIN HAVE THREE PART

LEFT JOIN / LEFT OUTER JOIN = return all row from the left table

RIGHT JOIN /RIGHT OUTER JOIN = return all row from the right table

FULL JOIN / FULL OUTER JOIN = return all row from the both table

>UNION = this operator combine the result sets of multiple queries and remove duplicate row from the final result

SELECT COLUMNS FROM TABLE 1 UNION SELECT COLUMNS FROM TABLE 2;

>UNION ALL = operator, on the other hand, combines the result sets of multiple queries without removing duplicates.

SELECT columns FROM table1 UNION ALL SELECT columns FROM table2;

>Use UPDATE from modify existing records

UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;

Constraints in SQL

● NOT NULL - Restricts NULL value from being inserted into a column.

● CHECK - Verifies that all values in a field satisfy a condition.

● DEFAULT - Automatically assigns a default value if no value has been specified for the field.

● UNIQUE - Ensures unique values to be inserted into the field.

● PRIMARY KEY - Uniquely identifies each record in a table.

● FOREIGN KEY - Ensures referential integrity for a record in another table.

>delete and truncate

The `DELETE` command is used to remove specific rows from a table based on a specified condition

>The `TRUNCATE` command is used to remove all rows from a table in a more efficient way compared to `DELETE`.

>Aggregate and Scalar functions

Scalar Functions: Scalar functions operate on individual values in a row and return a single value for each row

Common scalar functions include

`UPPER`, `LOWER`, `CONCAT`, `LEN`, `ROUND`, `SUBSTRING` etc

SELECT UPPER(first_name) AS upper_first_name, LEN(last_name) AS last_name_length FROM TABLE NAME;

>Aggregate functions perform calculations on a set of values and return a single value that summarizes the entire set

Common aggregate functions include

`SUM`, `AVG`, `COUNT`, `MAX`, and `MIN`.

> For adding constraints

ALTER TABLE TABLE_NAME

MODIFY COLUMN_NAME VARCHAR(40);

>LIKE OPERATOR

Select * from table_name

Where f_name like ‘s’;

>DISTINCT CAUSE

Select distinct l_name

From table_name;

THANK YOU….