Library Management System Documentation

Overview

The Library Management System (LMS) is designed to manage library operations such as maintaining records of books, borrowers, branches, and loans. This document provides a comprehensive guide to the database schema, stored procedures, and usage instructions.

Database Schema

Tables

1. **tbl_publisher**

   - publisher_PublisherName: VARCHAR(100) PRIMARY KEY NOT NULL

   - publisher_PublisherAddress: VARCHAR(200) NOT NULL

   - publisher_PublisherPhone: VARCHAR(50) NOT NULL

2. **tbl_book**

   - book_BookID: INT AUTO_INCREMENT PRIMARY KEY

   - book_Title: VARCHAR(100) NOT NULL

   - book_PublisherName: VARCHAR(100) NOT NULL

   - FOREIGN KEY REFERENCES `tbl_publisher(publisher_PublisherName)` ON UPDATE CASCADE ON DELETE CASCADE

3. **tbl_library_branch**

   - library_branch_BranchID: INT AUTO_INCREMENT PRIMARY KEY

   - library_branch_BranchName: VARCHAR(100) NOT NULL

   - library_branch_BranchAddress: VARCHAR(200) NOT NULL

4. **tbl_borrower**

   - borrower_CardNo: INT AUTO_INCREMENT PRIMARY KEY

- borrower_BorrowerName: VARCHAR(100) NOT NULL

- borrower_BorrowerAddress: VARCHAR(200) NOT NULL

- borrower_BorrowerPhone: VARCHAR(50) NOT NULL

5. **tbl_book_loans**

- book_loans_LoansID: INT AUTO_INCREMENT PRIMARY KEY

- book_loans_BookID: INT NOT NULL

  - FOREIGN KEY REFERENCES ‘tbl_book(book_BookID)’ ON UPDATE CASCADE ON DELETE CASCADE

- book_loans_BranchID: INT NOT NULL

  - FOREIGN KEY REFERENCES `tbl_library_branch(library_branch_BranchID)` ON UPDATE CASCADE ON DELETE CASCADE

- book_loans_CardNo: INT NOT NULL

  - FOREIGN KEY REFERENCES `tbl_borrower(borrower_CardNo)` ON UPDATE CASCADE ON DELETE CASCADE

- book_loans_DateOut: DATE NOT NULL

- book_loans_DueDate: DATE NOT NULL

6. **tbl_book_copies**

- book_copies_CopiesID: INT AUTO_INCREMENT PRIMARY KEY

- book_copies_BookID: INT NOT NULL

  - FOREIGN KEY REFERENCES `tbl_book(book_BookID)` ON UPDATE CASCADE ON DELETE CASCADE

- book_copies_BranchID: INT NOT NULL

  - FOREIGN KEY REFERENCES `tbl_library_branch(library_branch_BranchID)` ON UPDATE CASCADE ON DELETE CASCADE

- book_copies_No_Of_Copies: INT NOT NULL

7. **tbl_book_authors**

- book_authors_AuthorID: INT AUTO_INCREMENT PRIMARY KEY

- book_authors_BookID: INT NOT NULL

- FOREIGN KEY REFERENCES `tbl_book(book_BookID)` ON UPDATE CASCADE ON DELETE CASCADE

- book_authors_AuthorName: VARCHAR(50) NOT NULL

Stored Procedures

1. **bookCopiesAtAllSharpstown**

Retrieves the number of copies of a specific book at the Sharpstown branch.

QUERY:

```
CREATE PROCEDURE bookCopiesAtAllSharpstown (
    IN bookTitle VARCHAR(70) DEFAULT 'The Lost Tribe',
    IN branchName VARCHAR(70) DEFAULT 'Sharpstown'
)
BEGIN
    SELECT copies.book_copies_BranchID AS BranchID,
        branch.library_branch_BranchName AS BranchName,
        copies.book_copies_No_Of_Copies AS NumberOfCopies,
        book.book_Title AS BookTitle
    FROM tbl_book_copies AS copies
    INNER JOIN tbl_book AS book ON copies.book_copies_BookID = book.book_BookID
    INNER JOIN tbl_library_branch AS branch ON copies.book_copies_BranchID = branch.library_branch_BranchID
    WHERE book.book_Title = bookTitle AND branch.library_branch_BranchName = branchName;
END;
```

2. **LoanersInfo**

Fetches information about borrowers with loans due today at a specified branch.

QUERY:

```
CREATE PROCEDURE LoanersInfo (
    IN DueDate DATE,
    IN LibraryBranchName VARCHAR(50) DEFAULT 'Sharpstown'
)
BEGIN
    SELECT Branch.library_branch_BranchName AS BranchName,
        Book.book_Title AS BookName,
        Borrower.borrower_BorrowerName AS BorrowerName,
        Borrower.borrower_BorrowerAddress AS BorrowerAddress,
        Loans.book_loans_DateOut AS DateOut,
        Loans.book_loans_DueDate AS DueDate
    FROM tbl_book_loans AS Loans
    INNER JOIN tbl_book AS Book ON Loans.book_loans_BookID = Book.book_BookID
    INNER JOIN tbl_borrower AS Borrower ON Loans.book_loans_CardNo =
Borrower.borrower_CardNo
    INNER JOIN tbl_library_branch AS Branch ON Loans.book_loans_BranchID =
Branch.library_branch_BranchID
    WHERE Loans.book_loans_DueDate = DueDate AND Branch.library_branch_BranchName
= LibraryBranchName;
END;
```

3. **TotalLoansPerBranch**

Provides the total number of books loaned out from each branch.

QUERY:

```
CREATE PROCEDURE TotalLoansPerBranch()
BEGIN
    SELECT Branch.library_branch_BranchName AS BranchName,
        COUNT(Loans.book_loans_BranchID) AS TotalLoans
    FROM tbl_book_loans AS Loans
    INNER JOIN tbl_library_branch AS Branch ON Loans.book_loans_BranchID =
Branch.library_branch_BranchID
    GROUP BY Branch.library_branch_BranchName;
END;
```

4. **BookbyAuthorandBranch**

Retrieves titles and the number of copies of books authored by a specific author at a specific branch.

QUERY:

```
CREATE PROCEDURE BookbyAuthorandBranch (
    IN BranchName VARCHAR(50),
    IN AuthorName VARCHAR(50)
)
BEGIN
    SELECT Branch.library_branch_BranchName AS BranchName,
        Book.book_Title AS Title,
        Copies.book_copies_No_Of_Copies AS NumberOfCopies
    FROM tbl_book_authors AS Authors
    INNER JOIN tbl_book AS Book ON Authors.book_authors_BookID = Book.book_BookID
    INNER JOIN tbl_book_copies AS Copies ON Authors.book_authors_BookID =
Copies.book_copies_BookID
```

```
    INNER JOIN tbl_library_branch AS Branch ON Copies.book_copies_BranchID =
Branch.library_branch_BranchID

    WHERE Branch.library_branch_BranchName = BranchName AND
Authors.book_authors_AuthorName = AuthorName;

END;
```

Usage

1. Create Tables : Use the provided SQL scripts to create the necessary tables in your database.

2. Insert Data: Populate the tables with initial data as required for your library operations.

3. Execute Procedures: Utilize the stored procedures to perform specific queries and manage data.

Example Procedure Calls

```
-- Number of copies of "The Lost Tribe" at Sharpstown branch

    CALL bookCopiesAtAllSharpstown('The Lost Tribe', 'Sharpstown');


-- Information of borrowers with loans due today at Sharpstown branch

    CALL LoanersInfo(CURDATE(), 'Sharpstown');


-- Total number of books loaned out from each branch

    CALL TotalLoansPerBranch();


-- Titles and number of copies of books by Stephen King at Central branch

    CALL BookbyAuthorandBranch('Central', 'Stephen King');
```