LIBRARY MANAGEMENT SYSTEM
PXP170009

1. Introduction

1.1 Document overview

This document describes the architecture of Library Management system.

- Description of the system
- Software used to develop the system
- SQL queries to run the database

1.2 Description of the system

- System contains a GUI to interact with it. Where you can search for a book through the text box and you can check out a book of your interest and you will be provided with an unique card number and librarian can add borrowers details and for every late check in appropriate fine will be calculated and stored and it will be updates periodically.

1.3 Softwares Required

The below applications are required to run the application successfully

- MySQL workbench/command line.
- Servlets
- MySQL connector for java (JDBC).
- Maven plugins to import the files as Maven document
- Apache Tomcat Server
- Eclipse
- JSP
- pom files contains all the jar dependencies.

Run the following SQL commands in the MySQL workbench to fetch the results of the query:

Query1

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';


-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------
DROP SCHEMA IF EXISTS `mydb` ;

-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;

-- -----------------------------------------------------
-- Table `mydb`.`book`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`book` ;

CREATE TABLE IF NOT EXISTS `mydb`.`book` (
  `ISBN` VARCHAR(10) NOT NULL  ,
  `TITLE` VARCHAR(200) NULL,
  PRIMARY KEY (`ISBN`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`authors`
-- -----------------------------------------------------
```

```
DROP TABLE IF EXISTS `mydb`.`authors` ;


CREATE TABLE IF NOT EXISTS `mydb`.`authors` (
  `AUTHOR_ID` INT NOT NULL AUTO_INCREMENT,
  `NAME` VARCHAR(50) NULL,
  PRIMARY KEY (`AUTHOR_ID`))
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `mydb`.`book_authors`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`book_authors` ;


CREATE TABLE IF NOT EXISTS `mydb`.`book_authors` (
  `AUTHOR_ID` INT NOT NULL,
  `ISBN` VARCHAR(10) NOT NULL,
  PRIMARY KEY (`ISBN`, `AUTHOR_ID`),
  INDEX `fk_BOOK_AUTHORS_2_idx` (`AUTHOR_ID` ASC),
  CONSTRAINT `fk_BOOK_AUTHORS_1`
    FOREIGN KEY (`ISBN`)
    REFERENCES `mydb`.`book` (`ISBN`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_BOOK_AUTHORS_2`
```

```
    FOREIGN KEY (`AUTHOR_ID`)
    REFERENCES `mydb`.`authors` (`AUTHOR_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`borrower`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`borrower` ;

CREATE TABLE IF NOT EXISTS `mydb`.`borrower` (
  `CardID` VARCHAR(6) NOT NULL,
  `SSN` VARCHAR(11) NOT NULL,
  `BNAME` VARCHAR(50) NOT NULL,
  `ADDRESS` VARCHAR(100) NULL,
  `PHONE` VARCHAR(14) NULL,
  `ID` INT NULL,
  PRIMARY KEY (`CardID`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`book_loans`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `mydb`.`book_loans` ;


CREATE TABLE IF NOT EXISTS `mydb`.`book_loans` (
  `LOAN_ID` INT NOT NULL AUTO_INCREMENT,
  `ISBN` VARCHAR(10) NOT NULL,
  `CARD_ID` VARCHAR(6) NOT NULL,
  `DATE_OUT` DATE NOT NULL,
  `DUE_DATE` DATE NOT NULL,
  `DATE_IN` DATE NULL,
  PRIMARY KEY (`LOAN_ID`),
  INDEX `fk_BOOK_LOANS_1_idx` (`ISBN` ASC),
  INDEX `fk_book_loans_2_idx` (`CARD_ID` ASC),
  CONSTRAINT `fk_BOOK_LOANS_1`
    FOREIGN KEY (`ISBN`)
    REFERENCES `mydb`.`book` (`ISBN`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_book_loans_2`
    FOREIGN KEY (`CARD_ID`)
    REFERENCES `mydb`.`borrower` (`CardID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------

-- Table `mydb`.`fines`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`fines` ;


CREATE TABLE IF NOT EXISTS `mydb`.`fines` (

  `LOAN_ID` INT NOT NULL,

  `FINE_AMT` DECIMAL(5,2) NULL,

  `PAID` TINYINT(1) NULL,

  PRIMARY KEY (`LOAN_ID`),

  CONSTRAINT `fk_FINES_1`

    FOREIGN KEY (`LOAN_ID`)

    REFERENCES `mydb`.`book_loans` (`LOAN_ID`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION)

ENGINE = InnoDB;



SET SQL_MODE=@OLD_SQL_MODE;

SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;

SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Query2

```
CREATE OR REPLACE VIEW dummy AS
    SELECT
        t1.isbn AS ISBN,
        t1.title AS TITLE,
        t3.NAME AS AUTHOR_NAME
    FROM
        book t1
            LEFT JOIN
        book_authors t2 ON t1.ISBN = t2.ISBN
            LEFT JOIN
        authors t3 ON t2.AUTHOR_ID = t3.AUTHOR_ID;

CREATE or replace view dummy2 as
select isbn, title, group_concat(AUTHOR_NAME) as author_name from dummy group by ISBN, TITLE;

CREATE OR REPLACE VIEW searchview AS
    SELECT
        isbn,
        title,
        author_name,
        (CASE
            WHEN
                ((SELECT
                        COUNT(*)
```

```
          FROM
            book_loans
          WHERE
            book_loans.isbn = dummy2.isbn and book_loans.DATE_IN is null) = 1)
      THEN
        'NO'
      ELSE 'YES'
    END) AS isavailable
  FROM
    dummy2;


create or replace view checkinview as
select borrower.id, book_loans.Card_ID as cardno,
book_loans.isbn as isbn, borrower.ssn as ssn, borrower.bname as borrower_name, book_loans.DATE_IN as date_in
from book_loans join borrower on book_loans.card_ID = borrower.cardID;



CREATE OR REPLACE VIEW overdue AS
  SELECT
    loan_id, CARD_ID,
    (CASE
      WHEN DATE_IN IS NULL THEN DATEDIFF(CURDATE(), DUE_DATE)
      ELSE DATEDIFF(DATE_IN, DUE_DATE)
    END) AS daysoverdue,
    (CASE WHEN DATE_IN IS NULL THEN false ELSE true END) as isbookreturned
  FROM
    book_loans
  WHERE
    CURDATE() > DUE_DATE AND DATE_IN IS NULL
      OR DATE_IN > DUE_DATE;



CREATE OR REPLACE VIEW displayfines AS
  SELECT
    GROUP_CONCAT(overdue.LOAN_ID) AS LOAN_IDS,
    CARD_ID,
    SUM(FINE_AMT) AS TOTAL_FINE
  FROM
    overdue
      JOIN
    fines ON overdue.loan_id = fines.LOAN_ID
  WHERE
    paid = FALSE AND isbookreturned = TRUE
  GROUP BY CARD_ID;
```

Query3: to run the query 3 change the paths of the normalized csv files according to your path location

```
------------------------------------------------------------------------------------------------------------------------------------
```

```
use mydb;

LOAD DATA INFILE 'E:/db/books_normalized.csv'
INTO TABLE book
FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

LOAD DATA INFILE 'E:/db/authors_normalized.csv'
INTO TABLE authors
CHARACTER SET UTF8
(NAME);

set foreign_key_checks = 0;
LOAD DATA INFILE 'E:/db/book_authors_normalized.csv'
INTO TABLE book_authors
FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(@ISBN, @AUTHOR_NAME)
set
        AUTHOR_ID = (select AUTHOR_ID FROM authors WHERE NAME = @AUTHOR_NAME limit 1),
  ISBN = @ISBN;
set foreign_key_checks = 1;
```

```
LOAD DATA INFILE 'E:/db/borrowers_normalized.csv'
INTO TABLE borrower
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```