contains

Shelf

Requests

adds

Librarian

Book

Library Card

Person

1 1

has

1

M

Checks Out

M N

M 1

M

1

Library Card

|  |  |  |
| --- | --- | --- |
| Id | Pin # | dateIssued |

Person

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | DOB | Fname | Lname | Pin # | libraryCardNum |

Book

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | Author  FName | Author  Lname | Title | Desc | Genre | Checked  OutBy | dateOut | dueDate | addedBy | shelf |

Request

|  |  |  |
| --- | --- | --- |
| id | pid | bid |

Librarian

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id | Fname | Lname | startDate | DOB | Pin # | username |

Shelf

|  |  |  |
| --- | --- | --- |
| id | Location | floor |

**OUTLINE:**

This is a database based on a library. Patrons and librarians can both create and manage their account. Patrons can check out books, request books, and return books. Librarians can add and delete books as well as shelves in the library. Both can view and search the inventory based on title, author name, genre, and description. It is helpful and interesting to track this information, because it gives a library a way to keep track of their inventory, patrons, and librarians as well as the interaction between all of these things.

**DATABASE OUTLINE:**

People have library cards. Each person has one library card and each library card belongs to only one person. People can exist without a library card, but a library card must belong to a person. People can check out books. A person can check out many books, but a book can only be checked out by one person. People can also request books. People can request many books and a book can be requested by many different people. Books are added by librarians. A book is added by one librarian. Each librarian can add many books. A book must be added by a librarian but a librarian does not need to have added any books. Books are contained on shelves. A book is held on one shelf, while one shelf can hold many books. If a book exists it must be on a shelf but a shelf does not need to hold any books.

**CREATE TABLES:**

CREATE TABLE libraryCard(

id INT AUTO\_INCREMENT ,

pinNum INT(4) UNIQUE NOT NULL ,

dateIssued DATE,

PRIMARY KEY ( id, pinNum )

) ENGINE = INNODB;

CREATE TABLE Person (

id int AUTO\_INCREMENT PRIMARY KEY,

DOB date,

fname varchar(255) NOT NULL,

lname varchar(255) NOT NULL,

libraryCardNum int DEFAULT NULL UNIQUE,

pinNum(4) int DEFAULT NULL UNIQUE,

FOREIGN KEY (libraryCardNum, pinNum) references libraryCard (id, pinNum) ON DELETE SET NULL ON UPDATE CASCADE) ENGINE = INNODB;

CREATE TABLE Librarian (

id int PRIMARY KEY AUTO\_INCREMENT,

DOB date,

fname varchar(255) NOT NULL,

lname varchar(255) NOT NULL,

startDate date,

pinNum int NOT NULL UNIQUE,

userName varchar(255) NOT NULL UNIQUE)ENGINE = INNODB;

CREATE TABLE Shelf(

id int PRIMARY KEY AUTO\_INCREMENT,

location ENUM('North Wing', 'East Wing', 'South Wing', 'West Wing'),

floorNum int

)ENGINE = INNODB;

CREATE TABLE Book(

id int PRIMARY KEY AUTO\_INCREMENT,

authorFName varchar(255) NOT NULL,

authorLName varchar(255) NOT NULL,

title varchar(255) UNIQUE NOT NULL,

genre varchar(255) NOT NULL,

description text,

checkedOutBy int DEFAULT NULL,

dateOut Date,

dueDate Date,

addedBy int NOT NULL,

shelf int NOT NULL,

FOREIGN KEY (checkedOutBy) references Person (id) ON DELETE SET NULL,

FOREIGN KEY (addedBy) references Librarian (id),

FOREIGN KEY (shelf) references Shelf (id)

)ENGINE = INNODB;

CREATE TABLE request(

id int PRIMARY KEY AUTO\_INCREMENT ,

pid INT NOT NULL ,

bid INT NOT NULL ,

FOREIGN KEY ( pid ) REFERENCES Person( id ) ,

FOREIGN KEY ( bid ) REFERENCES Book( id )

) ENGINE = INNODB;

**CREATE LIBRARIAN:**

INSERT INTO Librarian (DOB, fName, lName, startDate, pinNum, username)

VALUES ('1990-03-07', 'Rory', 'Gilmore', CURDATE(), '1234', 'rgilmore');

**CREATE LIBRARY CARD:**

INSERT INTO libraryCard( pinNum, dateIssued )

VALUES ( 6543, CURDATE( ) ) ;

**CREATE PERSON:**

INSERT INTO Person (DOB, fname, lname, pinNum, libraryCardNum)

VALUES ('1967-10-18', 'Ross', 'Geller', '6543', (SELECT id FROM libraryCard WHERE pin=’6543’);

**SELECT ID:**

SELECT id FROM Person WHERE libraryCardNum=X

**DELETE CARD:**

DELETE FROM libraryCard WHERE id=X(“

**UPDATE PERSON WITH CARD NUMBER: -- for replace card**

UPDATE Person SET libraryCardNum = (SELECT libraryCard.id FROM libraryCard WHERE pinNum=4444), pinNum = 4444 WHERE id = 3

**GET CARD NUM TO PRINT TO USER**

SELECT id FROM libraryCard WHERE pinNum=X

**SEARCH FOR PERSON CARD NUMBR AND PIN NUM: -- to validate log in**

SELECT libraryCardNum, pinNum FROM Person;

**SEARCH FOR LIBRARIAN USERNAME AND PIN: --to validate log in**

SELECT username, pinNum FROM Librarian;

**SEARCH FOR PERSON NAME: --to display profile**

SELECT fname, lname

FROM Person

WHERE libraryCardNum =2;

**CREATE SHELF:**

INSET INTO Shelf (‘location’, floorNum’)

VALUES (‘North Wing’, 1);

**Find books to move**

SELECT id FROM Book WHERE genre=X

“**Move” Books**

UPDATE Book SET shelf=(SELECT max(id) FROM shelf)

WHERE id ='".$bookRow['id']

**CREATE BOOK:**

INSERT INTO Book(authorFName, authorLName, title, genre, description, addedBy, shelf) VALUES(?, ?, ?, ?, ?, (SELECT id FROM Librarian WHERE username= ?), ?

**SELECT BOOKS: --to print Patron table**

SELECT Book.id, authorFName, authorLName, title, genre, description, checkedOutBy, dueDate, Shelf.location AS location, Shelf.floorNum AS floorNum, Person.libraryCardNum AS libNum

FROM Book

INNER JOIN Shelf ON Shelf.id = Book.shelf

LEFT JOIN Person on Person.id = Book.checkedOutBy

**CHECK OUT A BOOK:**

UPDATE Book SET checkedOutBy = ( SELECT id

FROM Person

WHERE libraryCardNum = '2' ), dateOut=curdate(), dueDate=date\_add(curdate(), Interval 10 DAY)

WHERE id =4

**RETURN BOOK:**

UPDATE Book SET checkedOutBy=NULL WHERE id=2

**PATRON BOOK SEARCH:**

SELECT Book.id, authorFName, authorLName, title, genre, description, checkedOutBy, dueDate, Shelf.location AS location, Shelf.floorNum AS floorNum, Person.libraryCardNum AS libNum  
FROM Book  
INNER JOIN Shelf ON Shelf.id = Book.shelf  
LEFT JOIN Person ON Person.id = Book.checkedOutBy  
WHERE authorfName LIKE '%shel%'

**REQUEST BOOK**

INSERT INTO request(pid, bid) VALUES ((SELECT id FROM Person WHERE libraryCardNum=2), 7)

**GET REQUEST INFO**

SELECT pid FROM request WHERE bid=4

**CANCEL REQUEST**

DELETE FROM request WHERE pid =3 AND bid =7

**FIRST REQUEST**

SELECT MIN(id), pid FROM request WHERE bid=4

**SELECT PERSON ID WITH CARD NUMBER**

SELECT id FROM Person WHERE libraryCardNum=4

**CHECKED OUT BOOK DATA FOR PATRON PROFILE**

SELECT Book.id, authorFName, authorLName, title, dueDate, dateOut FROM Book

WHERE checkedOutBy = (SELECT id FROM Person WHERE libraryCardNum='4')

**REQUEST BOOK DATA FOR PATRON PROFILE**

SELECT Book.id, authorFName, authorLName, title, dueDate, request.id FROM Book

INNER JOIN request ON request.bid = Book.id

WHERE pid = ( SELECT id FROM Person WHERE libraryCardNum =4 )

**FOR PATRON PROFILE INFO**

SELECT Person.id, fname, lname, libraryCardNum, Person.pinNum, libraryCard.dateIssued AS date

FROM Person

INNER JOIN libraryCard ON libraryCard.id = Person.libraryCardNum

WHERE libraryCardNum =4

**DELETE PERSON/CARD – QUIT LIBRARY**

DELETE FROM Person WHERE pinNum = ?

DELETE FROM Person WHERE pinNum = ?

**UPDATE LIBRARY CARD**

UPDATE libraryCard SET pinNum=? WHERE pinNum=?

**SELECT PIN TO DETERMINE IF NEW PIN IS UNIQUE**

SELECT pinNum FROM Person

**GET PIN FOR USER – TO TEST INPUT OF PIN**

SELECT pinNum FROM Person WHERE libraryCardNum=X

**GET Librarians name for profile**

SELECT fname, lname FROM Librarian WHERE username='

**SEARCH INVENTORY**

SELECT Book.id, authorFName, authorLName, title, genre, description, checkedOutBy, dueDate, dateOut, Shelf.location AS location, Shelf.floorNum AS floorNum, Person.fName AS pfname, Person.lname AS plname, Librarian.fname AS lfname, Librarian.lname AS llname

FROM Book

INNER JOIN Shelf ON Shelf.id = Book.shelf

LEFT JOIN Person ON Person.id = Book.checkedOutBy

INNER JOIN Librarian ON Librarian.id = Book.addedBy

**TO PRINT REQUESTS TO LIBRARIAN**

SELECT fname, lname, request.bid

FROM Person

LEFT JOIN request ON request.pid = Person.id

**DELETE BOOK**

DELETE FROM Book WHERE id=X

**Librarian profile info**

SELECT Librarian.id, fname, lname, username, startDate, pinNum FROM Librarian

WHERE username=X

**Print books added by a librarian**

SELECT Book.id AS id, authorFName, authorLName, title, genre, description, checkedOutBy, dueDate, dateOut, Shelf.location AS location, Shelf.floorNum AS floorNum, Person.fName AS pfname, Person.lname AS plname

FROM Book

INNER JOIN Shelf ON Shelf.id = Book.shelf

LEFT JOIN Person ON Person.id = Book.checkedOutBy

INNER JOIN Librarian ON Librarian.id = Book.addedBy

WHERE addedBy = (

SELECT id

FROM Librarian

WHERE username = 'rgilmore' )

**Select a list of genres**

SELECT DISTINCT genre

FROM Book

**Select Shelves**

SELECT id, location, floorNum FROM Shelf

**Select title to compare for add book**

SELECT title FROM Book

**Find shelf to put book on**

SELECT shelf FROM BOOK WHERE genre=X

**Get number of books checked out**

SELECT id FROM Book

WHERE checkedOutBy=(SELECT id FROM Person WHERE pinNum=X)