**Library management System**

**Index:**

1. **Introduction**
2. **Tables**

* **author**
* **book**
* **book\_author**
* **book\_copy**
* **borrower\_acount**
* **category**
* **check\_out**
* **hold**
* **notifications**
* **publisher**
* **wait\_list**

1. **ER Diagram**
2. **SQL commands (Select, Update, Insert, Delete)**

**Introduction:**

The Database Library System is intended to Automate the library activities such as creating a new borrower, giving books to the borrowers, maintaining the details of all the item that were available in the books . This also helps the librarians by providing information such as total copies available each book, list of books that belong to a particular category.

**Tables:**

* Books, Authors and Book\_Author:

As books are the core element in the library system, our database needs to contain information about them and their authors. The most intuitive way is to create two tables: book and author.

* To summarize, the table book will have the following columns:
* id: the unique identifier and the primary key.
* title: the title of the book.
* The table author will have the following columns:
* id: the unique identifier and the primary key.
* name: the name of the author.
* The table book\_author will have the following columns:
* book\_id: the id of the book.
* author\_id: the id of the author.
* Categories:

In our library, each book is assigned to a category. A book may belong to only one category.

* We create a new table category. This table needs only two columns:
* id: the unique row identifier and the primary key.
* name: the name of the category, for example, "Adventure."
* Book Copies and Publisher:

Our library may have multiple copies of the same book. A book may also be published by multiple publishers. To model this, we create two tables: book\_copy and publisher.

* The table publisher represents the publisher of the book. It consists of two columns:
* id: the unique row identifier and the primary key.
* name: the name of the publisher.
* The table book\_copy has the following columns:
* id: the unique row identifier and the primary key.
* year\_published: the year in which the book was published.
* book\_id: a foreign key pointing to the corresponding row in the book
* publisher\_id: a foreign key, the id of the publisher of the book in the table publisher.
* Borrower Accounts:

The book part is done. Now, we need people to check out our books! We need to store information about library books borrower, such as first name, surname, and email. Each member has his or her own library card and an account which may be active or blocked.

* We will use one table to store all this data: Borrower\_Account. It consists of the following columns:
* card\_number: the unique card number; it is also the primary key.
* first\_name: the patron's first name.
* surname: the patron's last name.
* email: the borrower's email used for sending notifications.
* status: active or blocked.
* Checkouts and Holds:

We need a way for the borrowers to check out or place a hold on (i.e., reserve) specific books. We will store records of checkouts and holds in two tables: checkout and hold.

* The table checkout has the following columns:
* id: the unique row identifier and the primary key.
* start\_time: the timestamp when the book is checked out.
* end\_time: the timestamp when the book is returned.
* book\_copy\_id: the id of the book copy that has been checked out.
* borrower\_account\_id: the id of the borrower who has checked out the book.
* is\_returned: a Boolean field indicating whether or not the book has been successfully returned.
* The table hold has the following columns:
* id: the unique row identifier and the primary key.
* start\_time: the timestamp when the book is placed on hold.
* end\_time: the timestamp when the hold ends. This is either when the hold expires or when the book has been checked out.
* book\_copy\_id: the id of the book copy that has been placed on hold.
* borrower\_account\_id: the id of the borrower who has placed the hold.
* Waiting List:

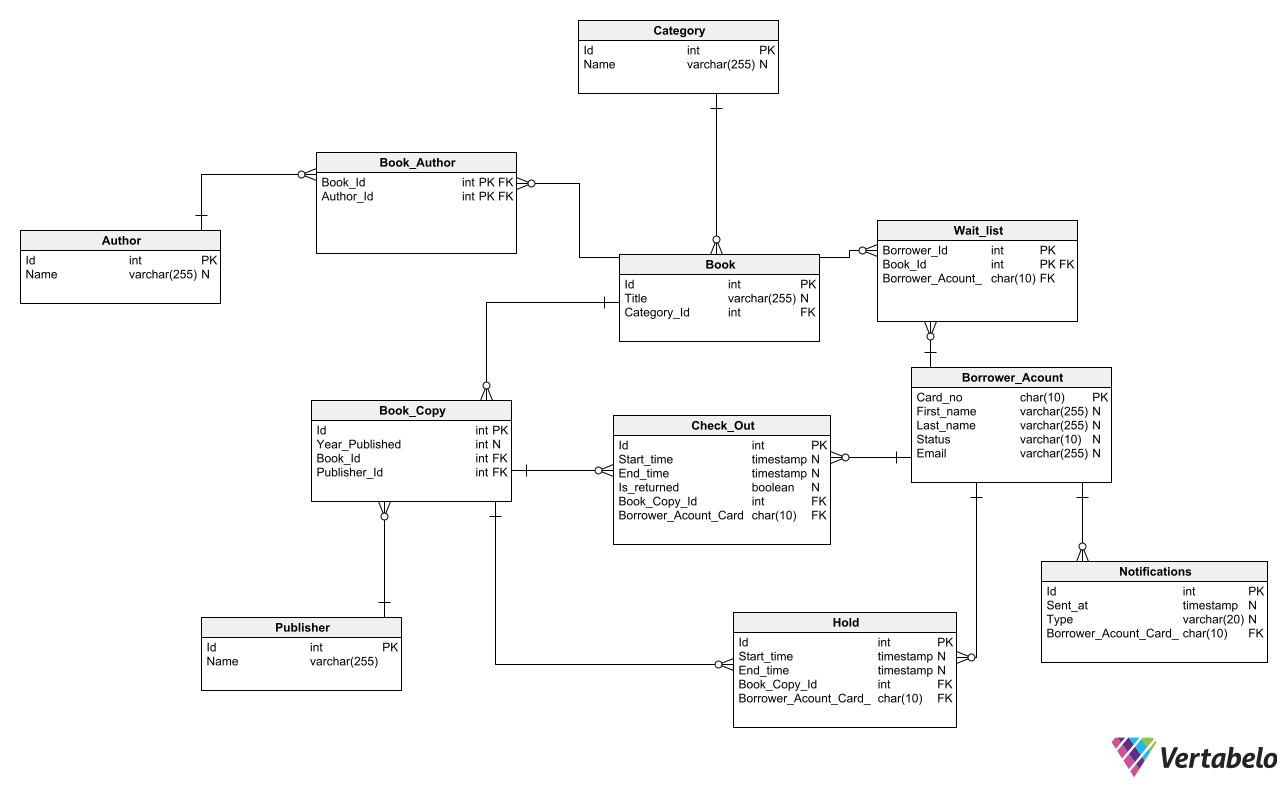
At times, all copies of a given book may be checked out or be on hold. The borrowers may want to be placed on a waiting list and get notified when it becomes available. We are going to create a table to save waiting list entries. Let's call it waitlist.

* This table will have only two columns:
* borrower\_id: the id of the borrower who is checking out the book and not returned.
* book\_id: a foreign key, the id of the book.
* Borrower\_Acount\_card\_no: a foreign key, the id of the borrower who wishes to check out the book.
* Notifications:

Last but not least, we need to send notifications to our borrowers: to remind them a book should be returned or to let them know the book they have been waiting for is now available in the library.We need to create one additional table, notifications.

* This table will consist of the following columns:
* id: the unique row identifier and the primary key.
* sent\_at: the timestamp when the notification is sent.
* type: the notification type. For now, it is either a reminder to return the book or a notification that a certain book is available. In the future, we may have more options here.
* Borrower\_account\_id: the ID pointing to the borrower to be notified by email.

**ER Diagram:**



**SQL commands (Select, Update, Insert, Delete):**

-- Created by Vertabelo (http://vertabelo.com)

-- Last modification date: 2023-01-10 17:09:48.498

show databases

use librarymanagement

show tables;

-- tables

-- Table: Author

CREATE TABLE Author (

Id int NOT NULL,

Name varchar(255) NULL,

CONSTRAINT Author\_pk PRIMARY KEY (Id)

);

-- Inserting data into Table: Author

Insert into Author Values(001, "R.N.Tagore");

Insert into Author Values(002, "Shakespeare");

Insert into Author Values(003, "Kalidas");

-- Check data in tables:

Select \* from Author;

-- Table: Book

CREATE TABLE Book (

Id int NOT NULL,

Title varchar(255) NULL,

Category\_Id int NOT NULL,

CONSTRAINT Book\_pk PRIMARY KEY (Id)

);

-- Inserting data into Table: Book

Insert into Book Values(001,"Geetanjali",001);

Insert into Book Values(002,"The Merchant of Venice",002);

Insert into Book Values(003,"Chitra",002);

Insert into Book Values(004,"Shakuntala",002);

Insert into Book Values(005,"Bisarjan",002);

-- Check data in tables:

Select \* from Book;

-- Table: Book\_Author

CREATE TABLE Book\_Author (

Book\_Id int NOT NULL,

Author\_Id int NOT NULL,

CONSTRAINT Book\_Author\_pk PRIMARY KEY (Book\_Id,Author\_Id)

);

-- Inserting data into Table: Book\_Author

Insert into Book\_Author Values(001,001);

Insert into Book\_Author Values(002,002);

Insert into Book\_Author Values(003,001);

Insert into Book\_Author Values(004,001);

Insert into Book\_Author Values(005,001);

-- Check data in tables:

Select \* from Book\_Author;

-- Table: Book\_Copy

CREATE TABLE Book\_Copy (

Id int NOT NULL,

Year\_Published int NULL,

Book\_Id int NOT NULL,

Publisher\_Id int NOT NULL,

CONSTRAINT Book\_Copy\_pk PRIMARY KEY (Id)

);

-- Inserting data into Table:Book\_Copy

Insert into Book\_Copy Values(001,1913,001,001);

Insert into Book\_Copy Values(002,1913,001,001);

Insert into Book\_Copy Values(003,1596,002,002);

Insert into Book\_Copy Values(004,1596,002,002);

Insert into Book\_Copy Values(005,1913,003,001);

Insert into Book\_Copy Values(006,1913,003,001);

Insert into Book\_Copy Values(007,1791,004,001);

Insert into Book\_Copy Values(008,1791,004,001);

Insert into Book\_Copy Values(009,1912,005,001);

Insert into Book\_Copy Values(010,1912,005,001);

-- Check data in tables:

Select \* from Book\_Copy;

-- Table: Borrower\_Acount

CREATE TABLE Borrower\_Acount (

Card\_no char(10) NOT NULL,

First\_name varchar(255) NULL,

Last\_name varchar(255) NULL,

Status varchar(10) NULL,

Email varchar(255) NULL,

CONSTRAINT Borrower\_Acount\_pk PRIMARY KEY (Card\_no)

);

-- Inserting data into table: Borrower\_Acount

Insert into Borrower\_Account Values(001,"Akshay","Kumar","Active","akshay@gmail.com");

Insert into Borrower\_Account Values(002,"Bhushan","Dasar","Blocked","bhushan@gmail.com");

Insert into Borrower\_Account Values(003,"Dinesh","Singh","Active","dinesh@gmail.com");

Insert into Borrower\_Account Values(004,"Riya","Sharma","Blocked","riya@gmail.com");

Insert into Borrower\_Account Values(005,"Sneha","Yadav","Active","sneha@gmail.com");

-- Check data in tables:

Select \* from Borrower\_Acount;

-- Table: Category

CREATE TABLE Category (

Id int NOT NULL,

Name varchar(255) NULL,

CONSTRAINT Category\_pk PRIMARY KEY (Id)

) COMMENT 'In our library, each book is assigned to a category. A book may belong to only one category';

-- Inserting data into table: Category

Insert into Category Values(001,"Poetic");

Insert into Category Values(002,"Play");

Insert into Category Values(003,"Literature");

-- Check data in tables:

Select \* from Category;

-- Table: Check\_Out

CREATE TABLE Check\_Out (

Id int NOT NULL,

Start\_time timestamp NULL,

End\_time timestamp NULL,

Is\_returned boolean NULL,

Book\_Copy\_Id int NOT NULL,

Borrower\_Acount\_Card\_no char(10) NOT NULL,

CONSTRAINT Check\_Out\_pk PRIMARY KEY (Id)

);

-- Inserting data into table:Check\_Out

Insert into Check\_Out Values(001,'2022-12-03','2023-01-10', True, 005,001);

Insert into Check\_Out Values(002,'2022-12-08','2023-01-11', True, 006,005);

Insert into Check\_Out Values(003,'2022-12-15','2023-01-05', True, 005,003);

Insert into Check\_Out(Id, Start\_time, Is\_returned, Book\_Copy\_Id, Borrower\_Acount\_Card\_no) Values(004,'2022-12-23', False, 005,001);

-- Updating data into table:

Update Check\_Out set Book\_Copy\_Id = '008' where Id = '003';

Update Check\_Out set Book\_Copy\_Id = '003' where Id = '001';

-- We can delete records using delete command

Delete from Check\_Out where Id = '002'

-- Check data in tables:

Select \* from Check\_Out;

-- Table: Hold

CREATE TABLE Hold (

Id int NOT NULL,

Start\_time timestamp NULL,

End\_time timestamp NULL,

Book\_Copy\_Id int NOT NULL,

Borrower\_Acount\_Card\_no char(10) NOT NULL,

CONSTRAINT Hold\_pk PRIMARY KEY (Id)

);

-- Inserting data into table: Hold

Insert into Hold Values(001,'2022-12-03','2022-12-23',005,001);

Insert into Hold Values(002,'2022-12-05','2022-12-15',008,003);

Insert into Hold Values(003,'2022-12-10','2022-12-31',001,005);

-- Check data in tables:

Select \* from Hold;

-- Table: Notifications

CREATE TABLE Notifications (

Id int NOT NULL,

Sent\_at timestamp NULL,

Type varchar(20) NULL,

Borrower\_Acount\_Card\_no char(10) NOT NULL,

CONSTRAINT Notifications\_pk PRIMARY KEY (Id)

);

-- Inserting data into table: Notificatios

Insert into Notifications Values(001,'2022-12-14 10:42:13',"Book is available",003);

Insert into Notifications Values(002,'2022-12-22 13:40:13',"Book is available",001);

Insert into Notifications Values(003,'2023-01-10 09:23',"Rem: Return Book",001);

-- Check data in tables:

Select \* from Notifications;

-- Table: Publisher

CREATE TABLE Publisher (

Id int NOT NULL,

Name varchar(255) NOT NULL,

CONSTRAINT Publisher\_pk PRIMARY KEY (Id)

);

-- Inserting data into table: Publisher

Insert into Publisher Values(001, "R.N.Tagore");

Insert into Publisher Values(002, "Shakespeare");

Insert into Publisher Values(003, "Kalidas");

-- Check data in tables:

Select \* from Publisher;

-- Table: Wait\_list

CREATE TABLE Wait\_list (

Borrower\_Id int NOT NULL,

Book\_Id int NOT NULL,

Borrower\_Acount\_Card\_no char(10) NOT NULL,

CONSTRAINT Wait\_list\_pk PRIMARY KEY (Borrower\_Id,Book\_Id)

);

-- Inserting data into Tables: Wait\_list

Insert into Wait\_list Values(001,003,005);

-- Check data in tables:

Select \* from Wait\_list;

-- foreign keys

-- Reference: Book\_Author\_Author (table: Book\_Author)

ALTER TABLE Book\_Author ADD CONSTRAINT Book\_Author\_Author FOREIGN KEY Book\_Author\_Author (Author\_Id)

REFERENCES Author (Id);

-- Reference: Book\_Author\_Book (table: Book\_Author)

ALTER TABLE Book\_Author ADD CONSTRAINT Book\_Author\_Book FOREIGN KEY Book\_Author\_Book (Book\_Id)

REFERENCES Book (Id);

-- Reference: Book\_Category (table: Book)

ALTER TABLE Book ADD CONSTRAINT Book\_Category FOREIGN KEY Book\_Category (Category\_Id)

REFERENCES Category (Id);

-- Reference: Book\_Copy\_Book (table: Book\_Copy)

ALTER TABLE Book\_Copy ADD CONSTRAINT Book\_Copy\_Book FOREIGN KEY Book\_Copy\_Book (Book\_Id)

REFERENCES Book (Id);

-- Reference: Book\_Copy\_Publisher (table: Book\_Copy)

ALTER TABLE Book\_Copy ADD CONSTRAINT Book\_Copy\_Publisher FOREIGN KEY Book\_Copy\_Publisher (Publisher\_Id)

REFERENCES Publisher (Id);

-- Reference: Check\_Out\_Book\_Copy (table: Check\_Out)

ALTER TABLE Check\_Out ADD CONSTRAINT Check\_Out\_Book\_Copy FOREIGN KEY Check\_Out\_Book\_Copy (Book\_Copy\_Id)

REFERENCES Book\_Copy (Id);

-- Reference: Check\_Out\_Borrower\_Acount (table: Check\_Out)

ALTER TABLE Check\_Out ADD CONSTRAINT Check\_Out\_Borrower\_Acount FOREIGN KEY Check\_Out\_Borrower\_Acount (Borrower\_Acount\_Card\_no)

REFERENCES Borrower\_Acount (Card\_no);

-- Reference: Hold\_Book\_Copy (table: Hold)

ALTER TABLE Hold ADD CONSTRAINT Hold\_Book\_Copy FOREIGN KEY Hold\_Book\_Copy (Book\_Copy\_Id)

REFERENCES Book\_Copy (Id);

-- Reference: Hold\_Borrower\_Acount (table: Hold)

ALTER TABLE Hold ADD CONSTRAINT Hold\_Borrower\_Acount FOREIGN KEY Hold\_Borrower\_Acount (Borrower\_Acount\_Card\_no)

REFERENCES Borrower\_Acount (Card\_no);

-- Reference: Notifications\_Borrower\_Acount (table: Notifications)

ALTER TABLE Notifications ADD CONSTRAINT Notifications\_Borrower\_Acount FOREIGN KEY Notifications\_Borrower\_Acount (Borrower\_Acount\_Card\_no)

REFERENCES Borrower\_Acount (Card\_no);

-- Reference: Wait\_list\_Book (table: Wait\_list)

ALTER TABLE Wait\_list ADD CONSTRAINT Wait\_list\_Book FOREIGN KEY Wait\_list\_Book (Book\_Id)

REFERENCES Book (Id);

-- Reference: Wait\_list\_Borrower\_Acount (table: Wait\_list)

ALTER TABLE Wait\_list ADD CONSTRAINT Wait\_list\_Borrower\_Acount FOREIGN KEY Wait\_list\_Borrower\_Acount (Borrower\_Acount\_Card\_no)

REFERENCES Borrower\_Acount (Card\_no);

-- Questions:-

use librarymanagement

-- 1. How many borrowers are there in the dataset?

Select count(\*) from Borrower\_Acount;

-- 2. How many books is issued per card in the Check\_out table?

Select distinct(Borrower\_Acount\_Card\_no), count(Book\_Copy\_Id) from Check\_Out group by Borrower\_Acount\_Card\_no;

-- 3. Which book has highest number of borrowers?

-- select \* from Check\_Out;

-- Select \* from Book;

-- Select \* from Book\_Copy;

-- Select Book\_Copy\_Id, Borrower\_Acount\_Card\_no from Check\_Out;

select distinct(B.Id) as Book\_Id, Title, count(C.Id) as Copy\_Id, count(Borrower\_Acount\_Card\_no) as card\_no from Book B left join Book\_Copy C on B.Id=C.Book\_Id left join Check\_Out on Book\_Copy\_Id=C.Id group by B.Id;

-- 4. Which borrower has the maximum books?

Select distinct(Borrower\_Acount\_Card\_No), count(Book\_Copy\_Id) from Check\_Out group by Borrower\_Acount\_Card\_No;

-- 5. Which borrower has not returned the book?Provide the book\_copy\_id which is yet to be return.

Select Borrower\_Acount\_Card\_no, Book\_Copy\_Id from Check\_Out where Is\_returned= False;

-- 6. Which author wrote the maximum no of books?

Select A.Id as Author\_Id, Name as Author\_Name, count(Book\_Id) from Author A left join Book\_Author B on A.Id=B.Author\_Id group by A.Id;

-- 7. Which category has the maximum number of books?

Select distinct(C.Id) as category\_Id, Name as Category\_Name, Count(B.Category\_Id) from Category C left join Book B on C.Id=B.Category\_Id group by C.Id;

-- 8. How many Notification has sent by the Libraray management System?

select count(\*) from Notifications;

-- End of file.