**Library DBMS**

This project is of an advanced Library Management System. This system will be able to keep track of all the books in the library, throughout all of the various branches of the library. The information about all books such as their ID’s, name, author, publication, status etc. can be stored in the database. A set of employees is present – these employees are all associated via their employee ID’s, names and salary. The employees manage the books present in the Library branches. Each of the branches of the library has an associated manager, address and branch ID. The manager is one of the employees stored in the Employee database.

Any customer can also sign up for the Library service in any branch. Customer has to specify their name and address – the ID, number of books issued and registration date is stored within the Customer database. The customer can issue and return books at any time. Every issue transaction has an associated transaction ID, the ID of the customer and book issued and an issue date. A Required return date value is also stored; if the return transaction’s date exceeds this date, a fine is levied on the customer.

Using this intelligent system, a user will find it easy to check the status of the book(s) in the library automatically, without having to physically go and check the book availability in the library branches. The manager of the Library System can also check on the employees hired, and whether they have been on schedule. If they have been following their schedule, their salary is paid to them.

**Entities and Attributes**

**Book**

* ID (primary key, distinct, NOT null)
* Name
* Author
* Publisher
* Edition
* Status (issued/free)
* Genre/Category

**Customer**

* Customer ID (primary key)
* Number of books issued
* Name
* Address
* Registration date
* Fine (if any)

**Branch**

* Manager ID
* Branch ID (primary key)
* Address  
  - Contact number  
  - Branch number  
  - Street  
  - City  
  - State  
  - Zipcode

**Employee**

* Employee ID (primary key)
* Employee Name
* Salary
* Position (manager or staff)

**Transaction Status**

* Customer ID (foreign key from Customer)
* Book ID (foreign key from Book)
* Issue or Return? (transtype)
* Issue/Return date
* Transaction ID (prim key)

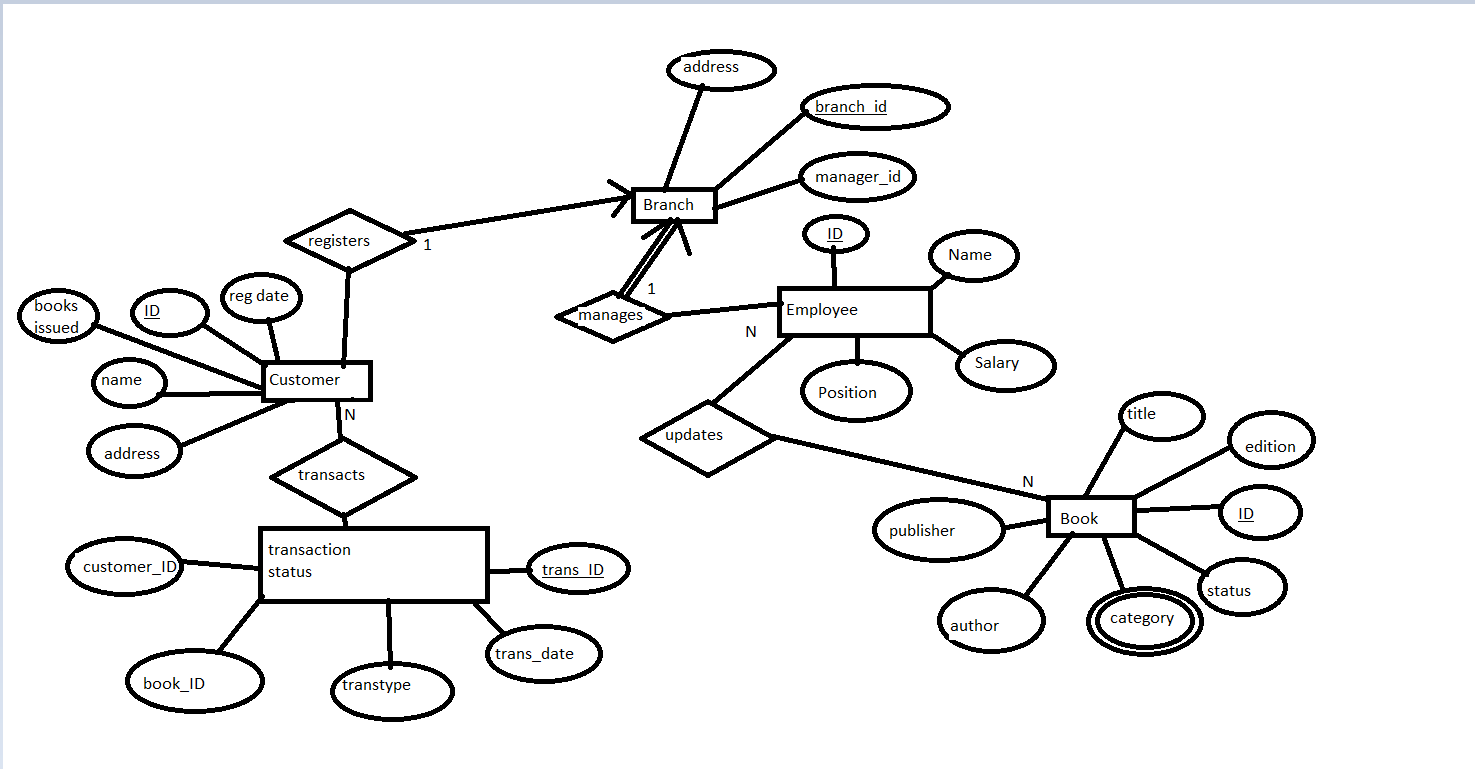
**Relations**

* MANAGER manages the Branch (1-N)
* CUSTOMER registers in respective Branch (N-1)
* CUSTOMER transacts Books (N-1)
* EMPLOYEE updates Books (N-N)

**Data Types to be used**

* Integer (for ID’s)
* Varchar (for alphanumeric information, like address and name)
* Date (for specifying the return/issue dates)

**E-R Diagram**



**Schema**

Book (title, edition, ID, status, category, author, publisher)

Employee (ID, name, salary)

Branch (address, branch\_id, manager\_id)

Customer (regdate, ID, books\_issued, name, address)

TransactionStatus(customerID, bookID, transtype, trans\_date, transID)

**Functional Requirements**

We intend to develop the system using the following technology –

* **Frontend**: HTML/CSS and PHP
* **Backend**: MySQL