DBMS Project :

Names : Dinky , Aastha , Gurvinder

Topic : Library Management System

**OMNI LIB**

(B2C)

Assumptions:

--All city/state libraries are registered or are partners in our app

--A Global library Card (ID) through which a user can Renew/Hold a Book

--2 ways for Payment –

🡪 Global Membership where user has to pay once a year (or once a lifetime in Government Libraries Rs.500/-) + Some Deposit so that a Book doesn’t get damaged

🡪City Specific Membership where user needs to pay depending upon the city it lives in + some deposit.

Features:

--A Global library Card (ID) through which a user can Renew/Hold a Book in any city/state library which is registered with the app

--2 ways for Payment –

🡪 Global Membership where user has to pay once a year (or once a lifetime in Government Libraries Rs.500/-) + Some Deposit so that a Book doesn’t get damaged.

--Multi user(On Library Admin Side)

--Issuing and Collection/Return a Book (Library Admin)

--Renew a book(User)

--Put Hold on a book(User)

--Search/Sort Books based on an Author Name , Book Name ,Genre , New Releases , Popular (Based on Ratings given by other users) -🡪(For Users)

--Search based on city and if a book is not available in my locality , then user can search for nearby locality library(User)

--Credit System -🡪 If on time or prior return of book , then user gets some positive points and can get some rewards based on those points (Voucher ,etc.)

🡪If there is a delay in book collection/return , then the user gets some negative points and can opt to pay either Cash or Online Payment.(Admin Access)

Entity Sets --- Attributes

Relationship Sets

Roles

User ---

Local Admin ---

City Admin—

State Admin---

Country Admin ---

-Work : Grants/Revokes permission to state admin

**Country Admin**

-Attributes :::

Name of Admin, email id , country id , phone number, city , state

Direct Relations :::(One to Many)

-With State Admins

Keeps a table of State Admins

-Work : Grants/Revokes permission to city admin

**State Admin**

-Attributes :::

Name of state admin, email id , state id , phone number, city , state , country id

Direct Relations :::(One to Many)

-With City Admins

Keeps a table of all the city admins in that state

-Work : Grants/Revokes permission to local admins

**City Admin**

-Attributes :::

Name of City Admin, email id , city id , phone number, city , state , state id , country id

Direct Relations :::(One to Many)

-With Local Admins

Keeps a table of all the local admins in a city

-Work : Grants/Revokes permission to a user

**Local Admin**

-Attributes :::

Name of Admin, email id , local / library id , phone number, city , state , city id , state id , country id

Direct Relations :::(One to Many)

-With Users

(Keeps a table of all the users in the library of the local admin)

-With Books/Magazines Table

Functions/Duties :

-Issue a book

-Collect a book

-Renew a book as notified by the user

-Keep a hold on a book as notified by the user

-Updates the Books and authors i.e. if there is a new launch , it adds those to the table of book

--------pending :::: collection of late fine

**User**

-Attributes :::

Name of User, email id , user id , phone number, city , state,

Area , Merits , Demerits ,ISBN, Book Count , Membership Type

Direct Relations :::(One to Many)

-With Books/Magazine Table

-With Local Admin

**Books/Magazine**

Attributes :::::

Book Name , Edition , Genre , Type , ISBN , Area ID, City ID , State ID , Lib ID , Publisher ID

Direct Relations ::::::

With Publishers

With Users

With Library Table

With Return/Issue Table

**Publisher**

Attributes : Publisher ID , Year of Publication , Name

Direct Relation::::

-With Book Table

**Library Table (Diamond )**

**It connects Users with Local Admin**

Attributes ::::

Lib Name , Lib ID , Area ID , City ID , State , Country , Local Admin ID

Direct Relation:::::

--With Users

--With Local Admin

**Return/Issue Table**

Attributes:::::

Actual Return date, Issue/Renew Date , Hold Date (Derived Attribute can be No. of Days to Hold), User Return Date ,ISBN

Direct Relations::::

-With Book/Magazine Table

-With User Table

**Author Table**

Attributes ::::

Author Name , Book Name , Author ID

No primary key

Direct Relations ::::

--With book Table

**Authorization**

Attributes::::

User ID/Email id , Password

Direct Relations:::::

All Admins + User

::::::::::::::::::**Notes** ::::::::::::::::

🡪Create a separate table for membership of user where the admins can keep track of how many users are there from a particular state , city , etc.

🡪Create a separate table for user authentication like username/userid , password for users (Basically log in details)

🡪Create a separate table for Authors as there can be many authors of a book and there can be many books from a single author …… i.e. many to many relationship persists.

-It can be said that author can act as a multivalued attribute in book/magazine table which should have a separate table as the no. of authors are not fixed. Book name/ISBN from book/magazine table can act as foreign key in author table

🡪Table for Membership --- Attributes … Membership Type like Monthly , Quarterly , Half Yearly , Yearly

--- Membership Amount

🡪Table for all admins with position so that redundancy can be avoided

🡪Add Membership Type in User Table

🡪Add Funds attribute in library table

🡪Think about pk in user table

🡪Think about payment options for dbms

🡪Think about employee table in library