**Digital Library System**

**Project introduction**

The digital library system allows members browse through books available at the library and related information about them. They can also make a reservation for a book to come and pick it up at a certain date or report a book if they find anything wrong with the book itself or the information provided about it.

**Relational Schema**

**DLS Database**: library members, books, Reservations, Managers, Librarian and Reports information.

It is defined by the following relational schema:

**member** (m\_id , m\_pwd, m\_name, m\_surname, m\_email)

**librarian** (l\_id, l\_pwd, l\_name, l\_email)

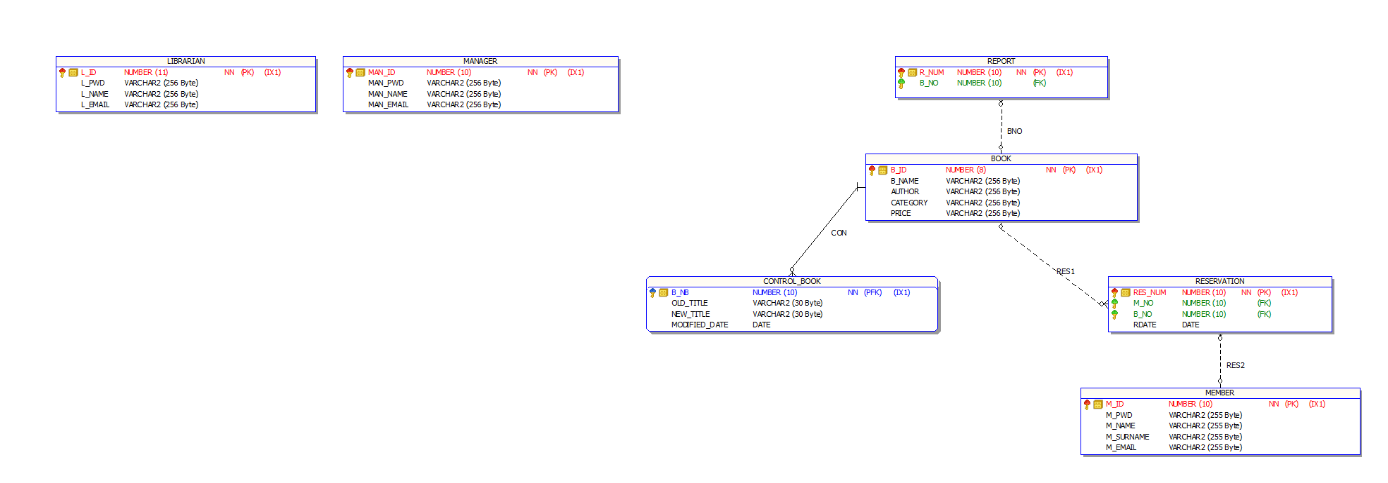
**manager** (man\_id, man\_pwd, man\_name, man\_email)

**book** (b\_id, b\_name, author, category, price)

**reservation** (res\_num,#m\_no, #b\_no, rdate)

**report** (r\_num, #b\_no)

**control\_book**(#b\_nb,#l\_no,old\_title,new\_title,modified\_date)

**Conceptual UML:**

**Interrogation implemented in the database:**

**RESERVED\_BOOKS :** A function that displays the number of books reserved by a member.

**B\_TITLE :** A procedure that displays the titles of the books that have been reserved by a specified member.

**REPORTED\_BOOK :** If the report number is 2, that means that the name of the title is wrong. This procedure will change the title.

**REMOVE\_MEMBER :** A procedure that deletes a member whose number is specified by the manager.

**EXCEP** : If a book is reported more than 3 times, an exception created by the user will display  ‘this book has been reported more than 3 times’

**AUDIT\_BOOK :** For each insert, update or delete on the table book, a trigger will insert in the table CONTROL\_BOOK the previous title and/or the new title and the modified date.

**RESERVE\_BOOK:** reserve a book for a certain member by inserting the book id and the member id.

**REPORT\_BOOK:** report a book by inserting the book id.

**DATE\_RES:** a trigger that displays and error message if the reservation date is before the system date

**BOOK\_INFO:** see all information about a book by inserting its id.

**VIEW\_MEMBER:** see all information about a member by inserting his or her id.

**ADD\_BOOK:** add a book to the book table by inserting its informations.

**Project Code:** view sql files

**Login and Password:**

**LOGIN:** library\_admin

**PASSWORD:** plibrary

**Conclusion :**

This project helped us organize books, members, librarians and managers in a single database. It allows for the manipulation of this data by reserving a book for a certain member, or adding one to the database by defining its information, reporting book for wrong information, removing a member or removing a librarian. It also includes procedure to see individual book information or see individual member information and a function to see the number of books reserved. It also contains a trigger that fires when there is an insertion, an update or a deletion of a book name to store the old and new data in a new control table. Finally, we put all of them into a package to make it easy to run them.