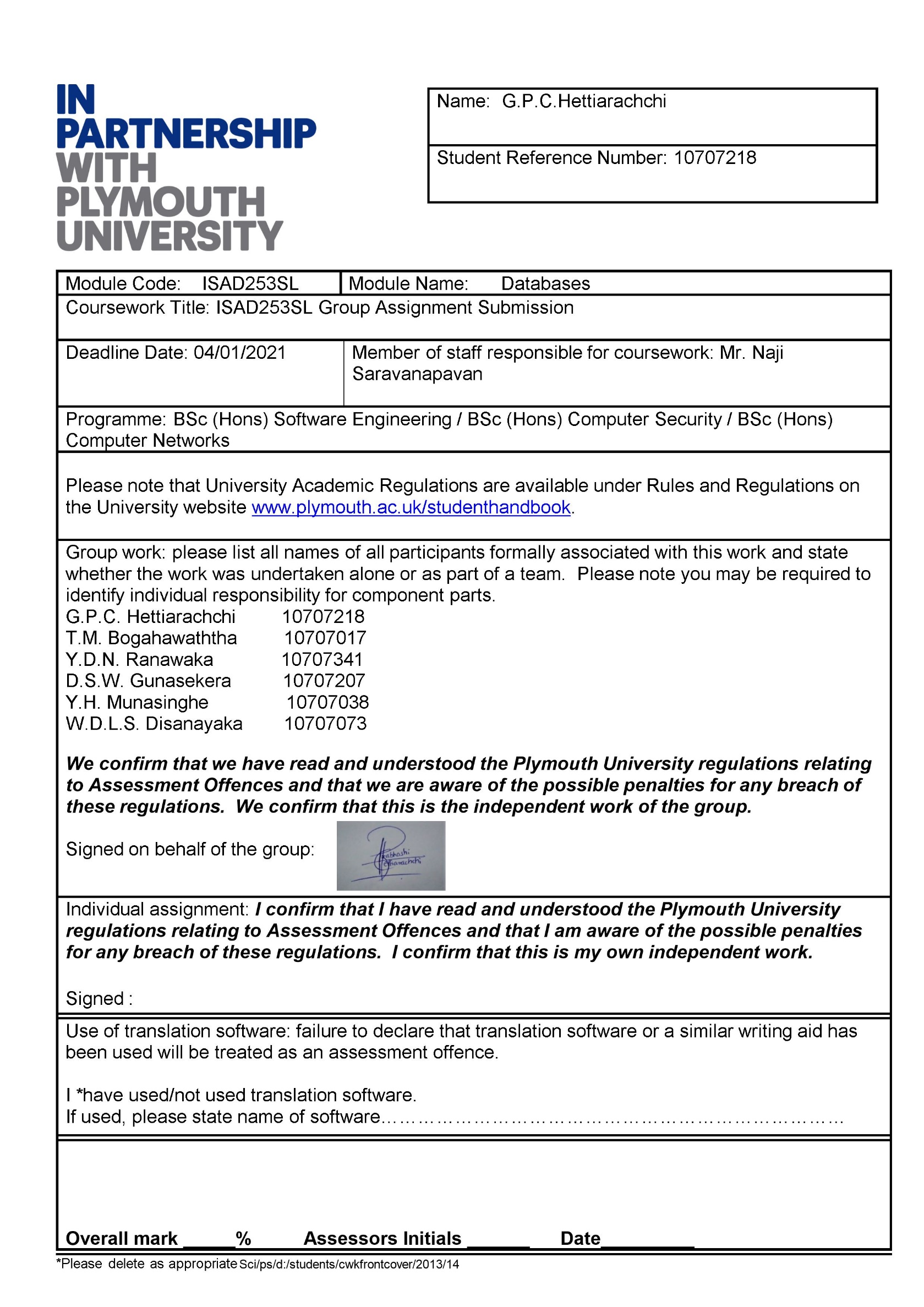
****

**ISAD253SL - Databases**

Coursework 2020 – 2021



**ACKNOWLEDGEMENT**

We would like to acknowledge our debt to each & every person associated in this Project Development. The Project Development required huge Commitment from all the individuals involved in it.

We are also indebted to **Mr.Naji Saravanapavan** who has guided us throughout the Project Development.

**TEAM 25**

**GITHUB LINK:** [**https://github.com/Pubzzz/LibraryManagementSystem**](mailto:https://github.com/Pubzzz/LibraryManagementSystem)

**Section 1**

* **Introduction to the scenario**

**Project name:** Library Management system

The main aim of this project is to provide a quality library service for everyone.

As we have learned C# programming language, we have used that programming language and Visual Studio IDE in developing the system and its interfaces.

Here, the database of the system is created using Microsoft SQL Server Management System which is high in data integrity and data validation properties along with the library management system in order to store details of the libraries, books, and the borrowers.

Here, Triggers, Mechanisms related to data validation and user defined functions are mainly used in the process of entering and management of data regarding the books, borrowers etc.

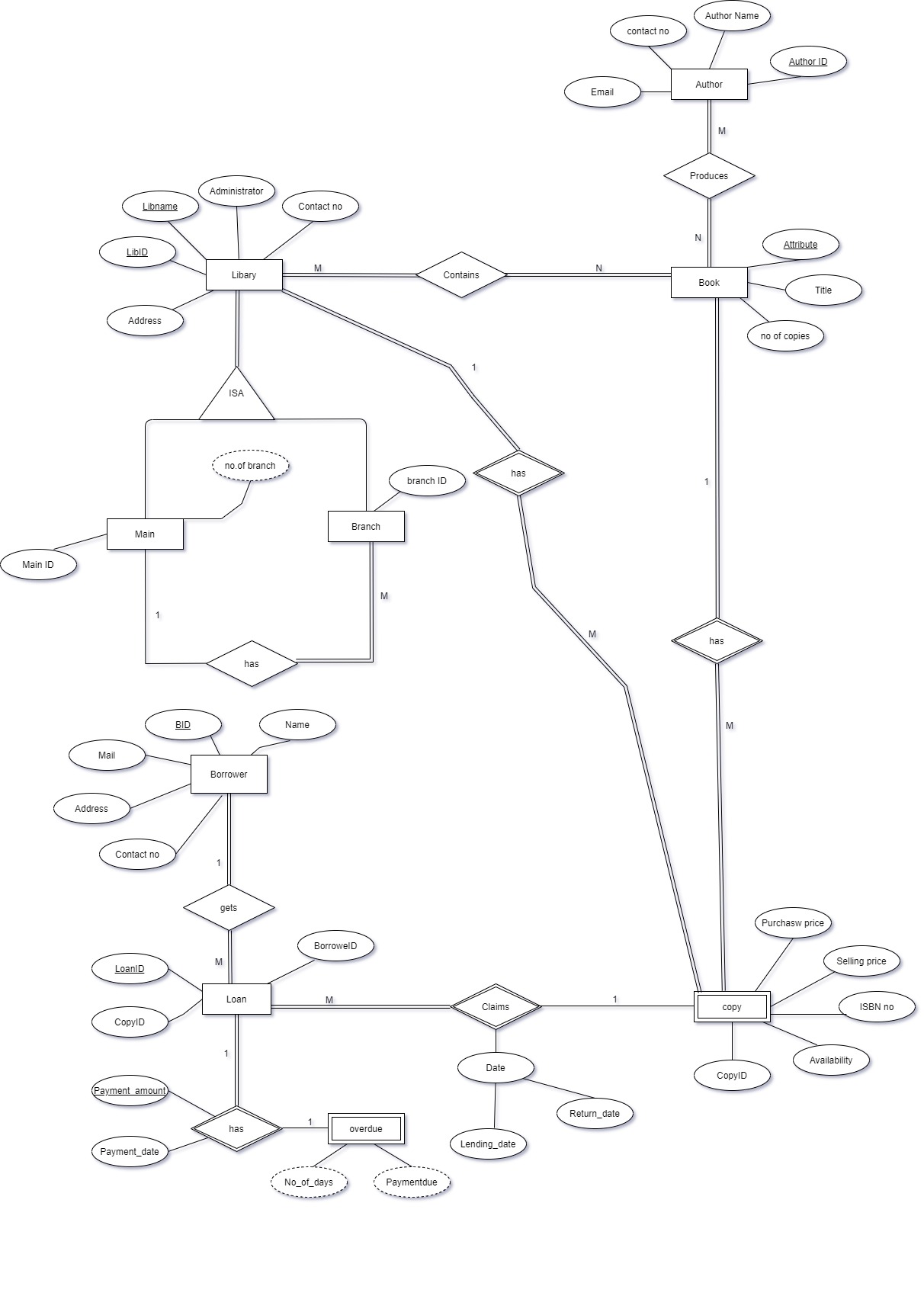
For the process of retrieving operational data, suitable views as well as suitable stored procedures are used allowing the authorized parties to access and take suitable decisions regarding the management.

**Objectives**

Main objective is to generate a computerized library management system that is more efficient and more reliable than a manual library management system.

Objectives of a computerized library management system can be listed as follows,

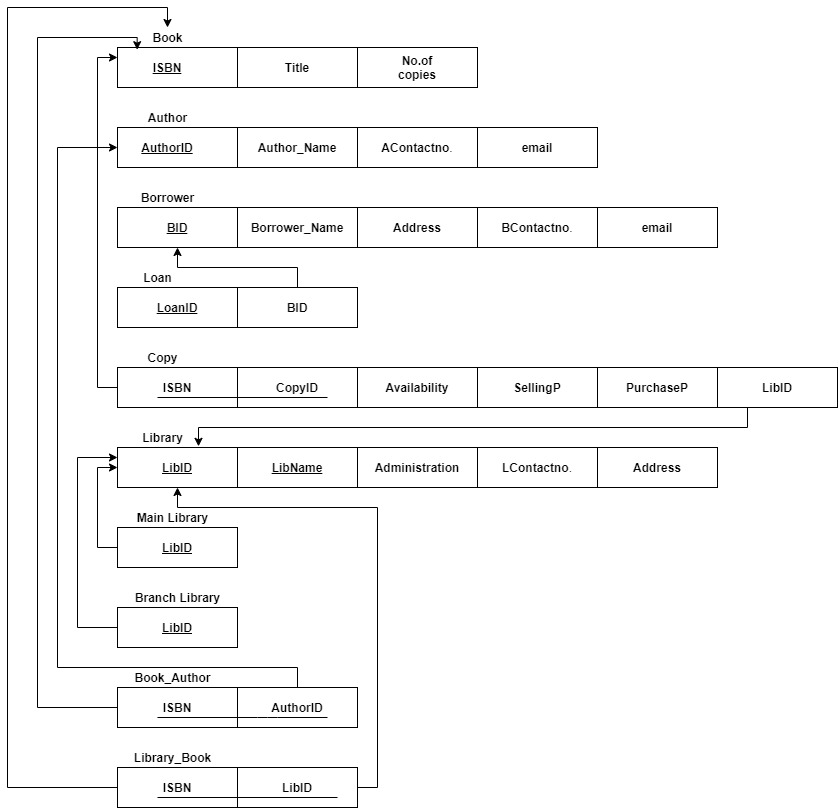
* Replacement of manual library management system.
* Developing as system that is efficient and reliable than a manual system.
* Making a user-friendly environment with attractive user interfaces.
* Development of a database with a proper storage of information about the details of books, borrowers, authors, libraries etc.
* **Extended Entity Relationship (EER)**

****

* **Additional Assumptions**

1. Assuming that one author has produced one or more books and assuming that there are books written by a single author as well as books written by a group of authors.
2. Assuming that a library contains a large collection of different types of books and a particular book can be located at least in one library or more libraries.
3. Assuming that some books do not have any copies or have more than one copy as well as assuming that a particular copy is exactly a copy of a single book.
4. Assuming that a copy is a weak entity related to the strong entity named as Book.
5. Assuming that a given library contains a number of copies of different books.
6. Assuming that only a single copy of a particular book can be borrowed by a loan in a particular occasion but the same copy can be obtained in several occasions.
7. Assuming that a copy will not take part in the loan or a particular copy can take part in number of loans of different borrowers.
8. The EER diagram was drawn according to the given scenario but, in the system the relationship between library and copy compatible only to one library(main library).

* **Relational Mapping**

****

* **Data Normalization**

**Book\_Author**

|  |  |
| --- | --- |
| **ISBN** | **AuthorID** |

**1NF**

It has only single atomic values.

So, this table is in 1NF.

**2NF**

There are no any partial dependencies.

So, this is in 2NF.

**3NF**

There are no partial dependencies.

So, this is in 3NF.

**Book**

|  |  |  |
| --- | --- | --- |
| **ISBN** | **Title** | **No.of copies** |

**1NF**

It has only single values.

So, this table is 1NF.

**2NF**

There are no partial dependencies.

So, this is in 2NF.

**3NF**

There are no any transitive dependencies.

So, this is in 3NF.

**Author**

|  |  |  |  |
| --- | --- | --- | --- |
| **AuthorID** | **Author\_Name** | **Contactno.** | **email** |

**1NF**

It has only single values.

So, this table is 1NF.

**2NF**

There are no partial dependencies.

So, this is in 2NF.

**3NF**

There are no any transitive dependencies.

So, this is in 3NF.

**Borrower**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BID** | **Name** | **Address** | **Contactno.** | **email** |

**1NF**

It has only single values.

So, this table is 1NF.

**2NF**

There are no partial dependencies.

So, this is in 2NF.

**3NF**

There are no any transitive dependencies.

So, this is in 3NF.

**Library\_Book**

|  |  |
| --- | --- |
| **ISBN** | **LibID** |

**1NF**

It has only single atomic values.

So, this table is in 1NF.

**2NF**

There are no any partial dependencies.

So, this is in 2NF.

**3NF**

There are no partial dependencies.

So, this is in 3NF.

**Loan**

|  |  |
| --- | --- |
| **LoanID** | **BID** |

**1NF**

It has only single atomic values.

So, this table is in 1NF.

**2NF**

There are no any partial dependencies.

So, this is in 2NF.

**3NF**

There are no partial dependencies.

So, this is in 3NF.

**Copy**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ISBN** | **CopyID** | **LibID** | **SellingP.** | **PurchaseP.** | **Availability** |

**1NF**

It has only single atomic values.

So, this table is in 1NF.

**2NF**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ISBN** | **CopyID** | **Availability** | **SellingP.** | **PurchaseP.** |

|  |
| --- |
| **LibID** |

There is a partial dependency here. There the no key attribute that is not connected with the primary key attribute is separated.

**3NF**

There are no partial dependencies.

So, this is in 3NF.

**Library**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LibID** | **LibName** | **Administration** | **Contactno.** | **Address** |

**1NF**

It has only single values.

So, this table is 1NF.

**2NF**

There are no partial dependencies.

So, this is in 2NF.

**3NF**

There are no any transitive dependencies. So, this is in 3NF.

**Main Library / Branch Library**

**1NF**

It has only single values.

So, this table is 1NF.

**2NF**

There are no partial dependencies.

So, this is in 2NF.

**3NF**

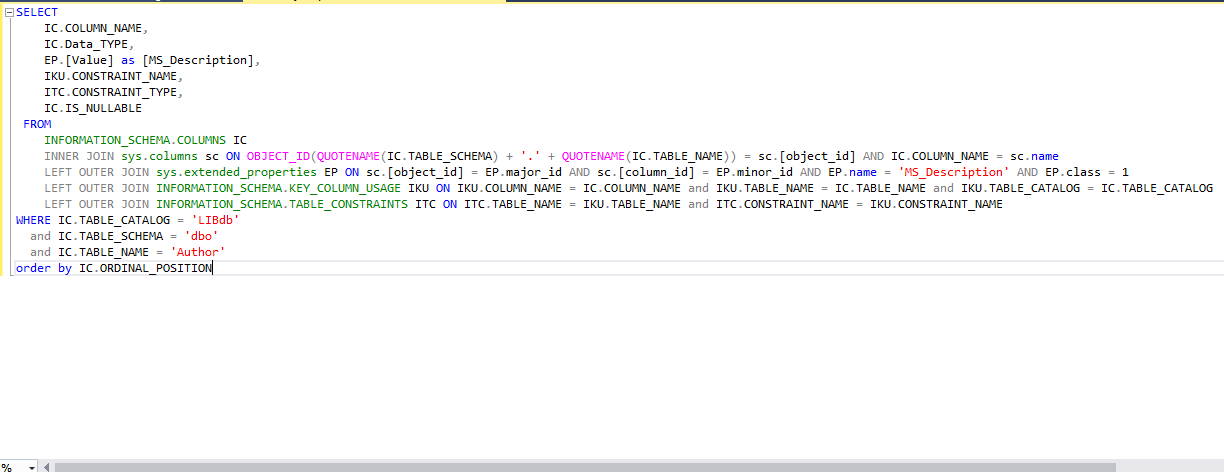
There are no any transitive dependencies.

So, this is in 3NF.

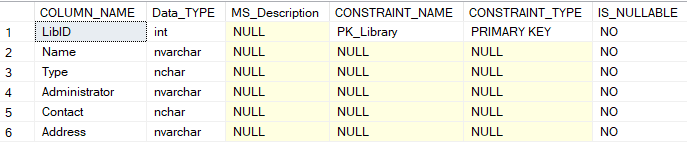
* **Data Dictionary**

Data dictionary is useful in understanding data and using data

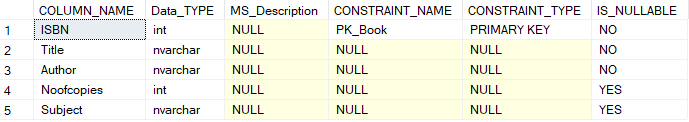
Here. A detailed information regarding the content of the database is provided by the below SQL code. They can be text descriptions, names of various variables and formats of data.



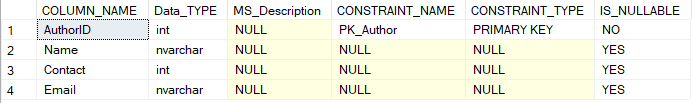
Library Table



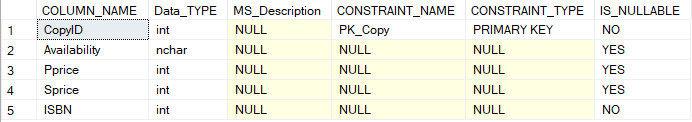
Book Table



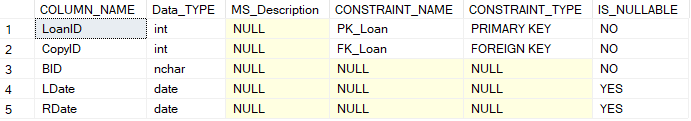
Author Table



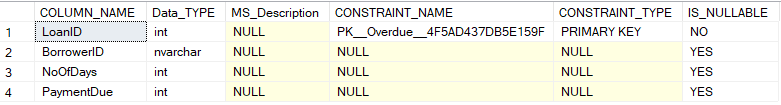
Copy Table



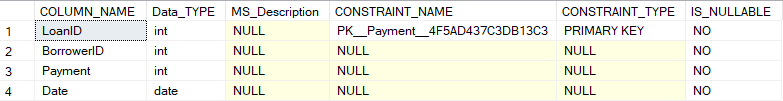
Loan Table



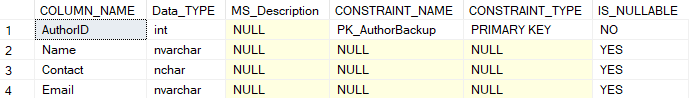
Overdue Table



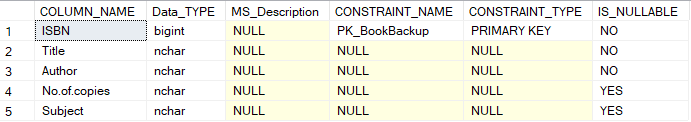
Payment Table

****

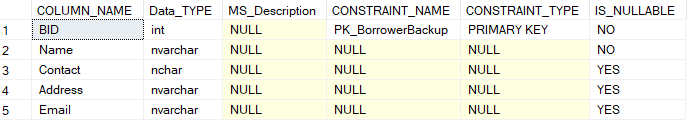
Author Backup Table



Book Backup Table



Author Backup Table



**Section 2**

* **Microsoft SQL Server Create Table statements with related Constraints**

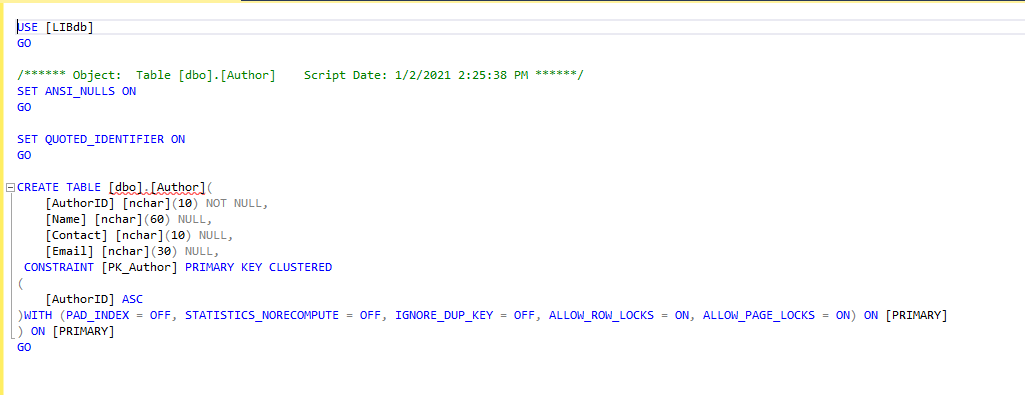
Library Table



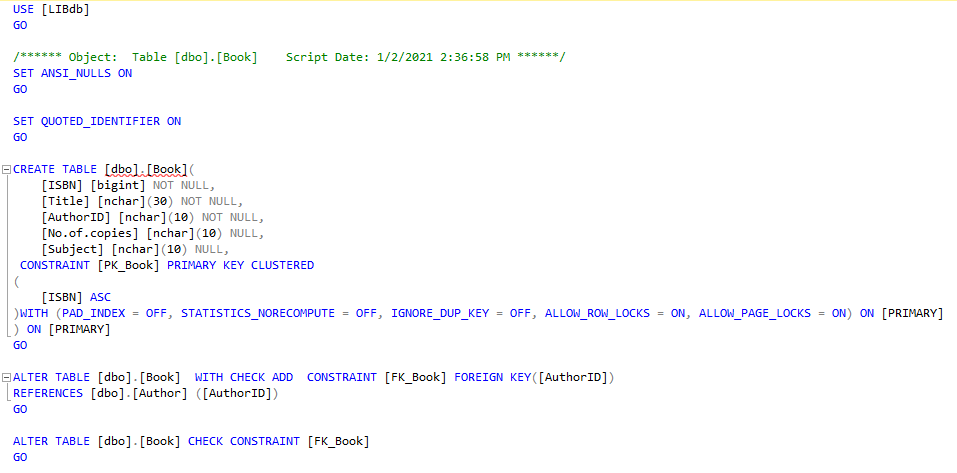
User Table



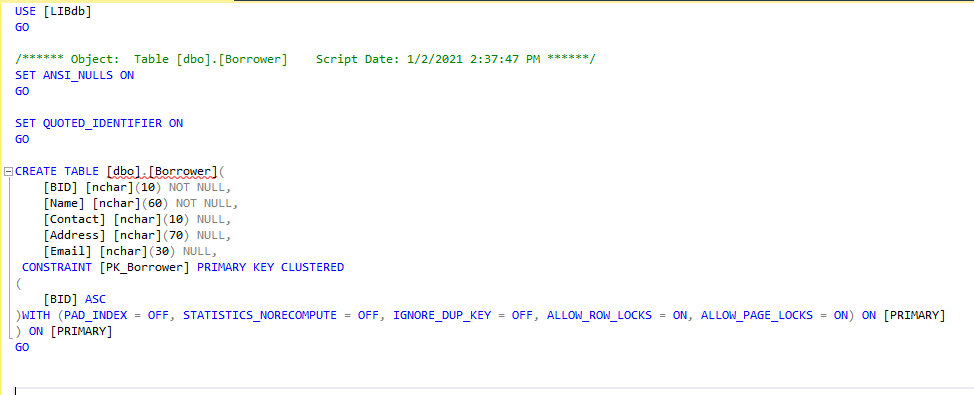
Author Table



Book Table



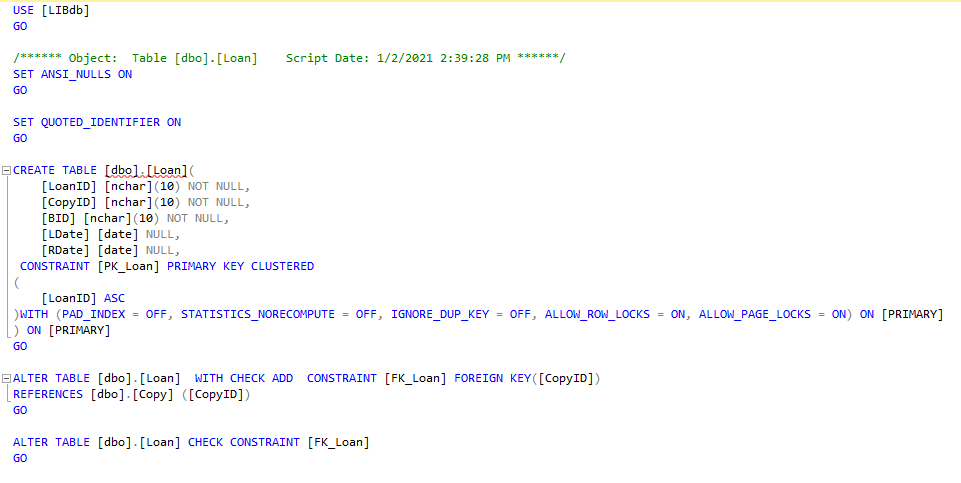
Borrower Table



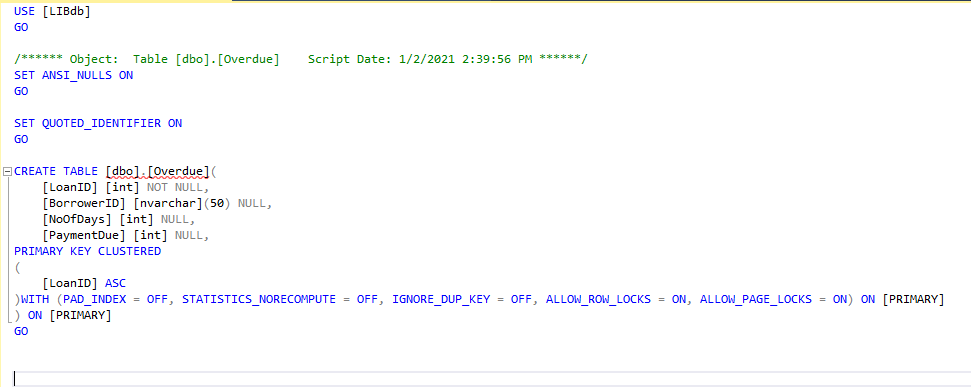
Copy Table



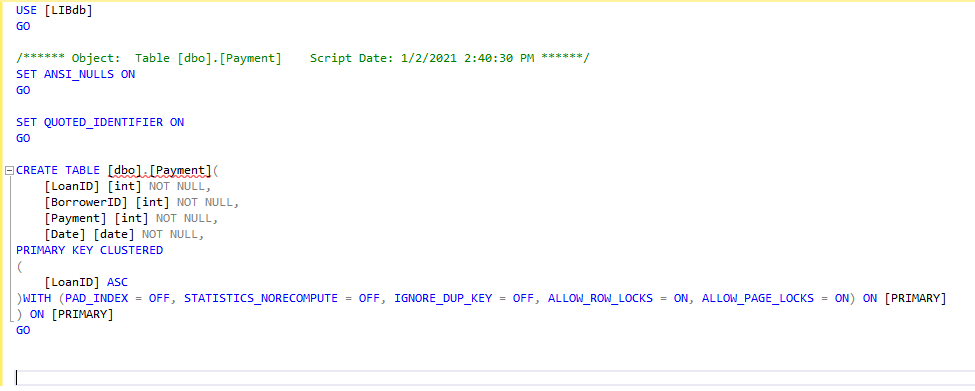
Loan Table



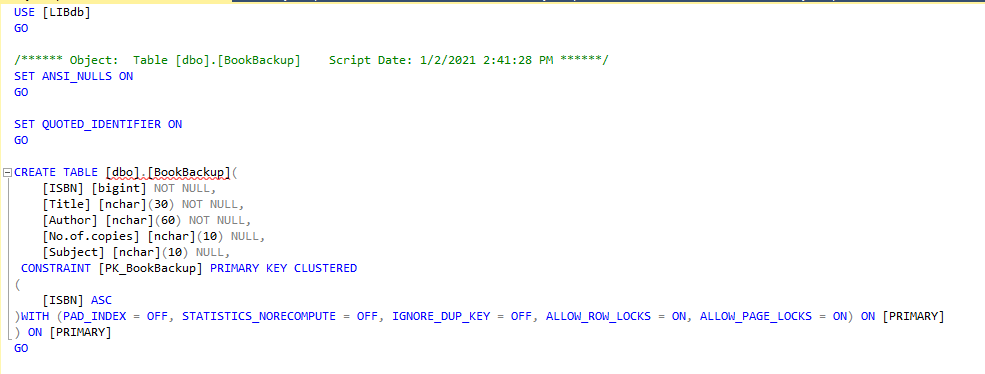
Overdue Table



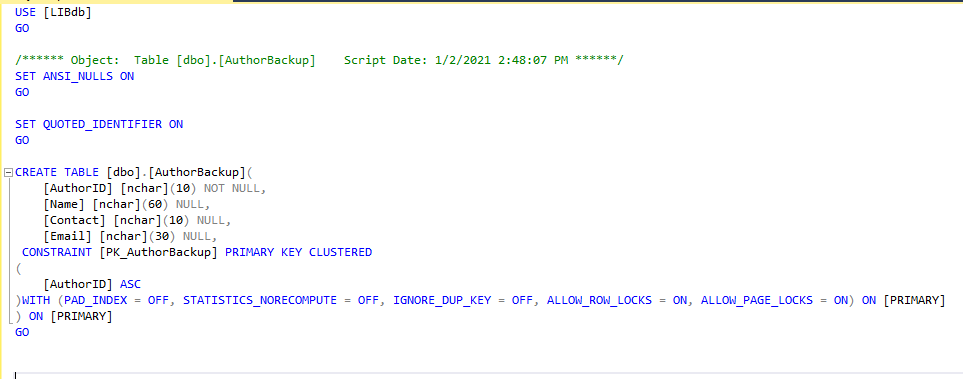
Payment Table



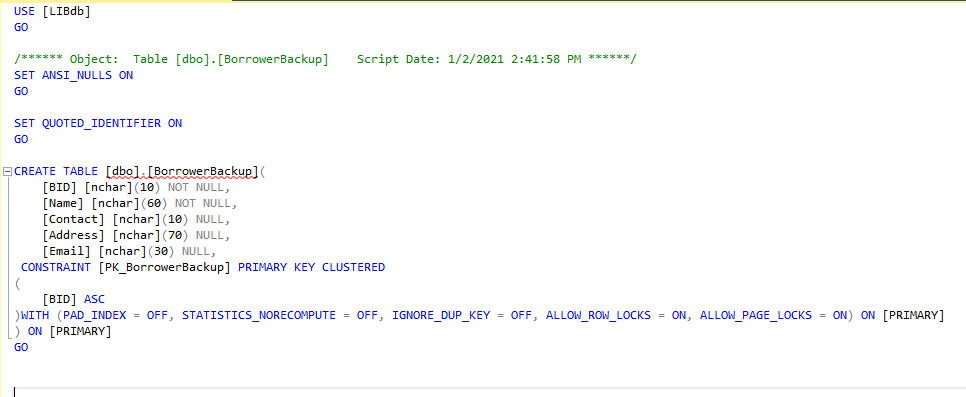
Book Backup Table



Author Backup Table



Borrower Backup Table

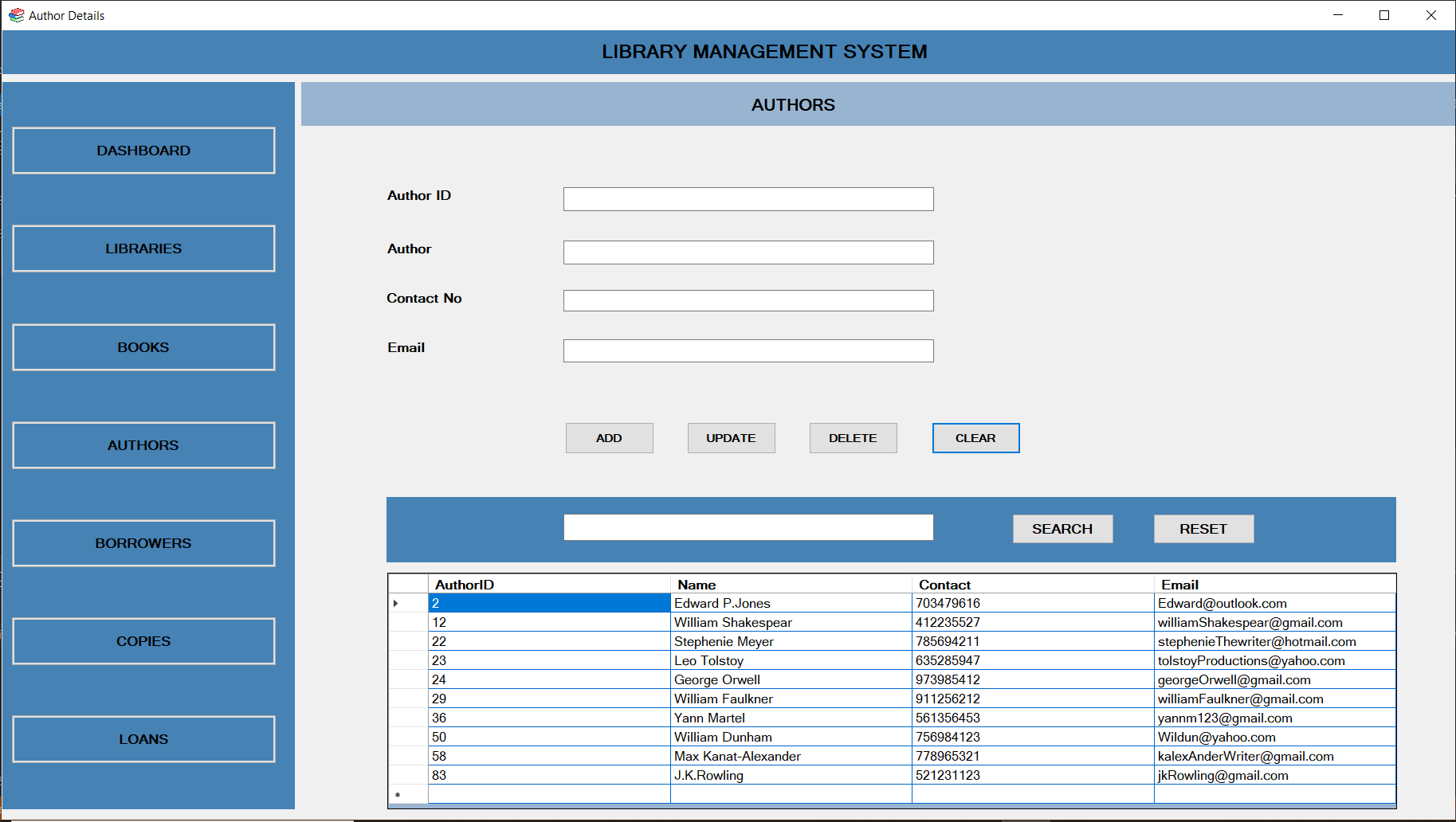


* **Database Diagram**

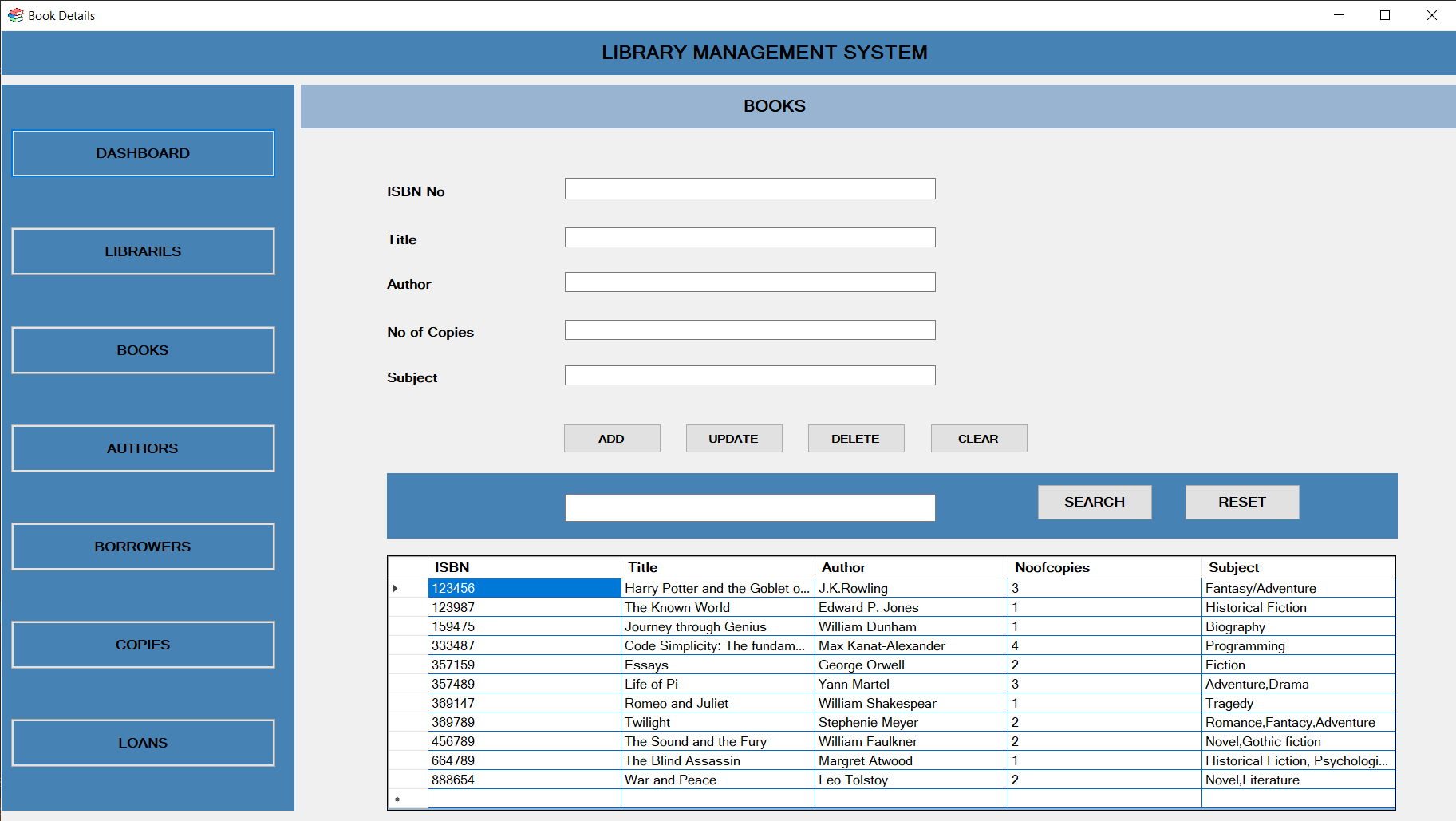


* **Sample Records**

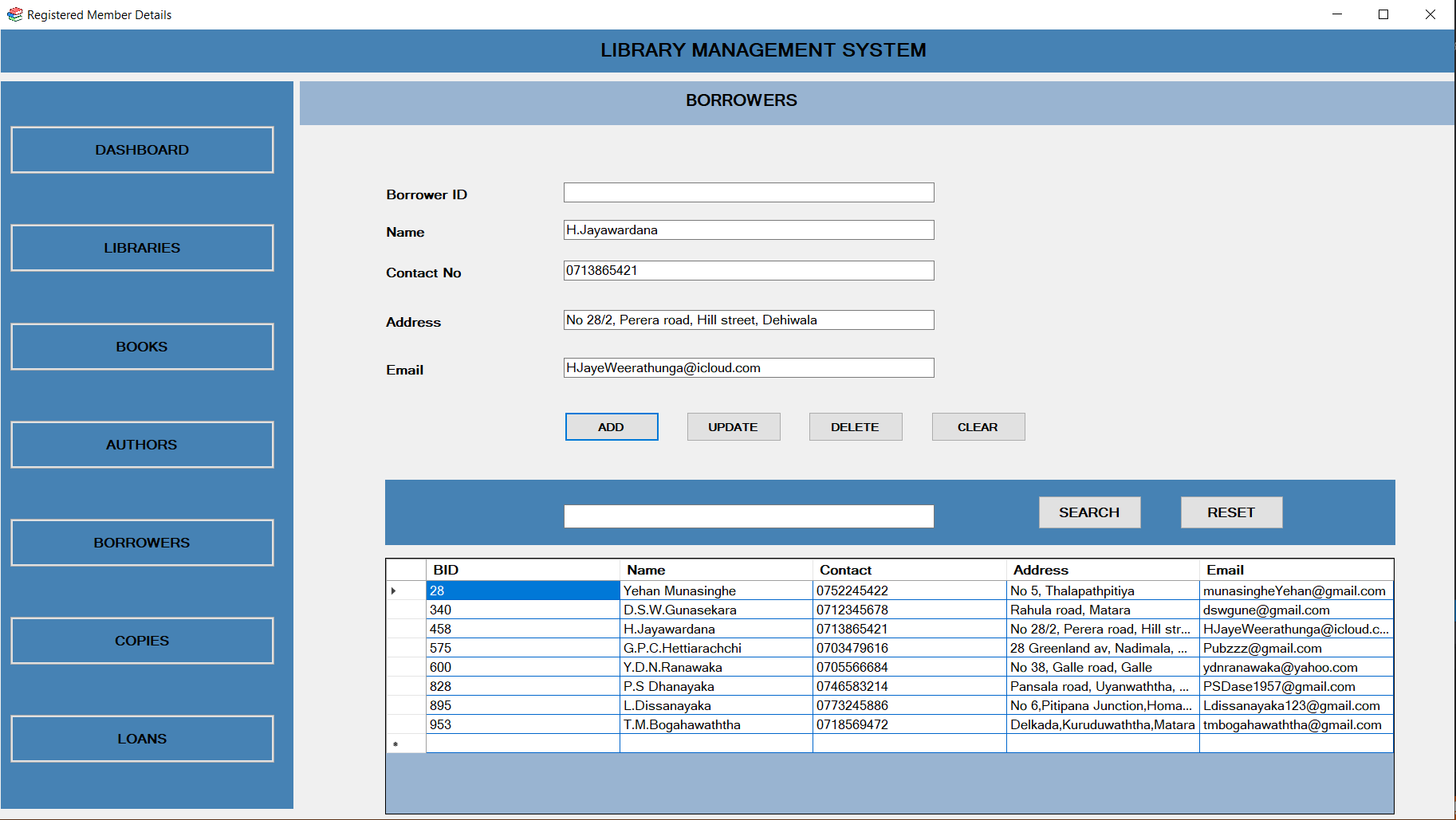
Author Table



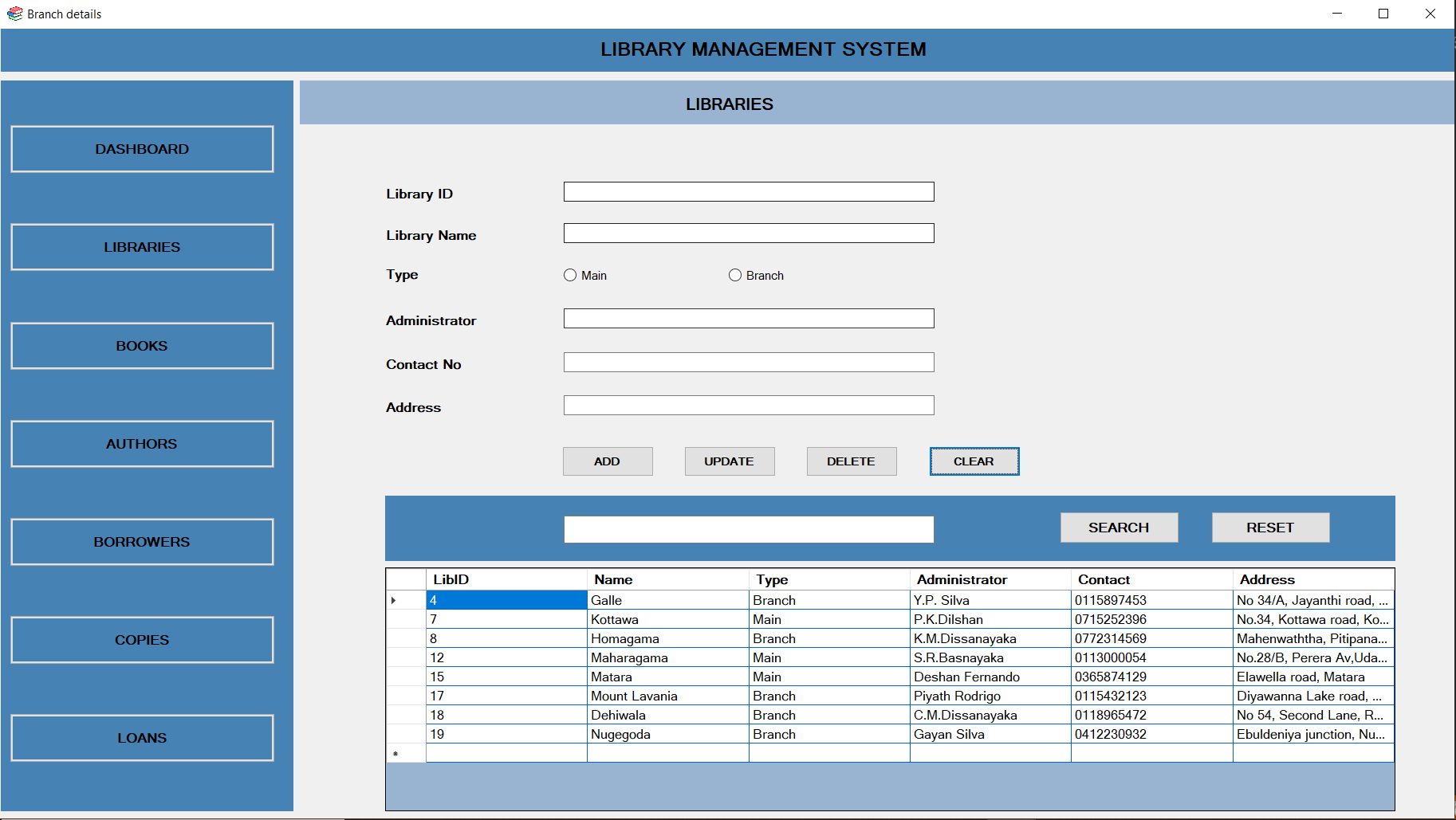
Book Table



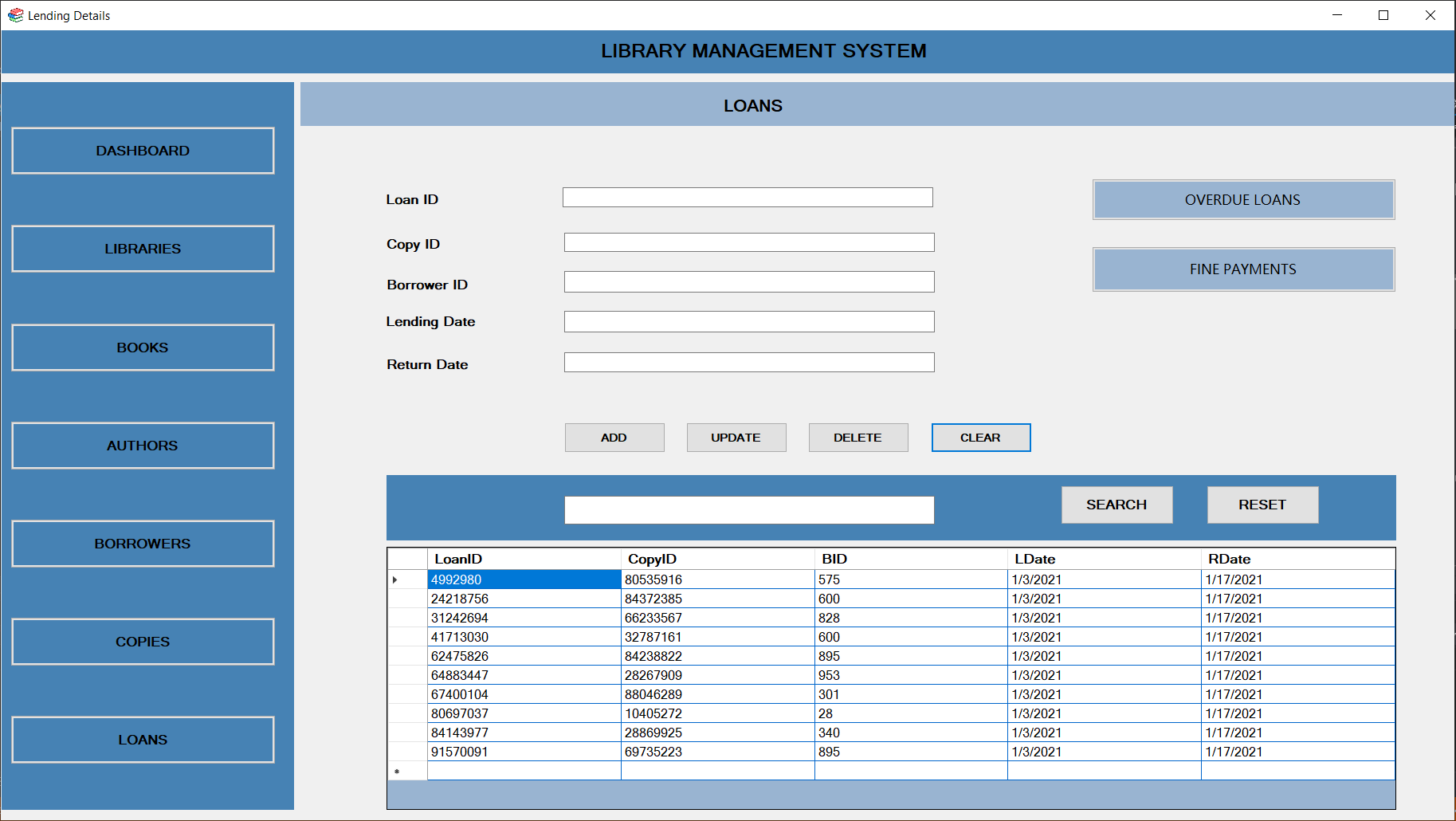
Borrower Table



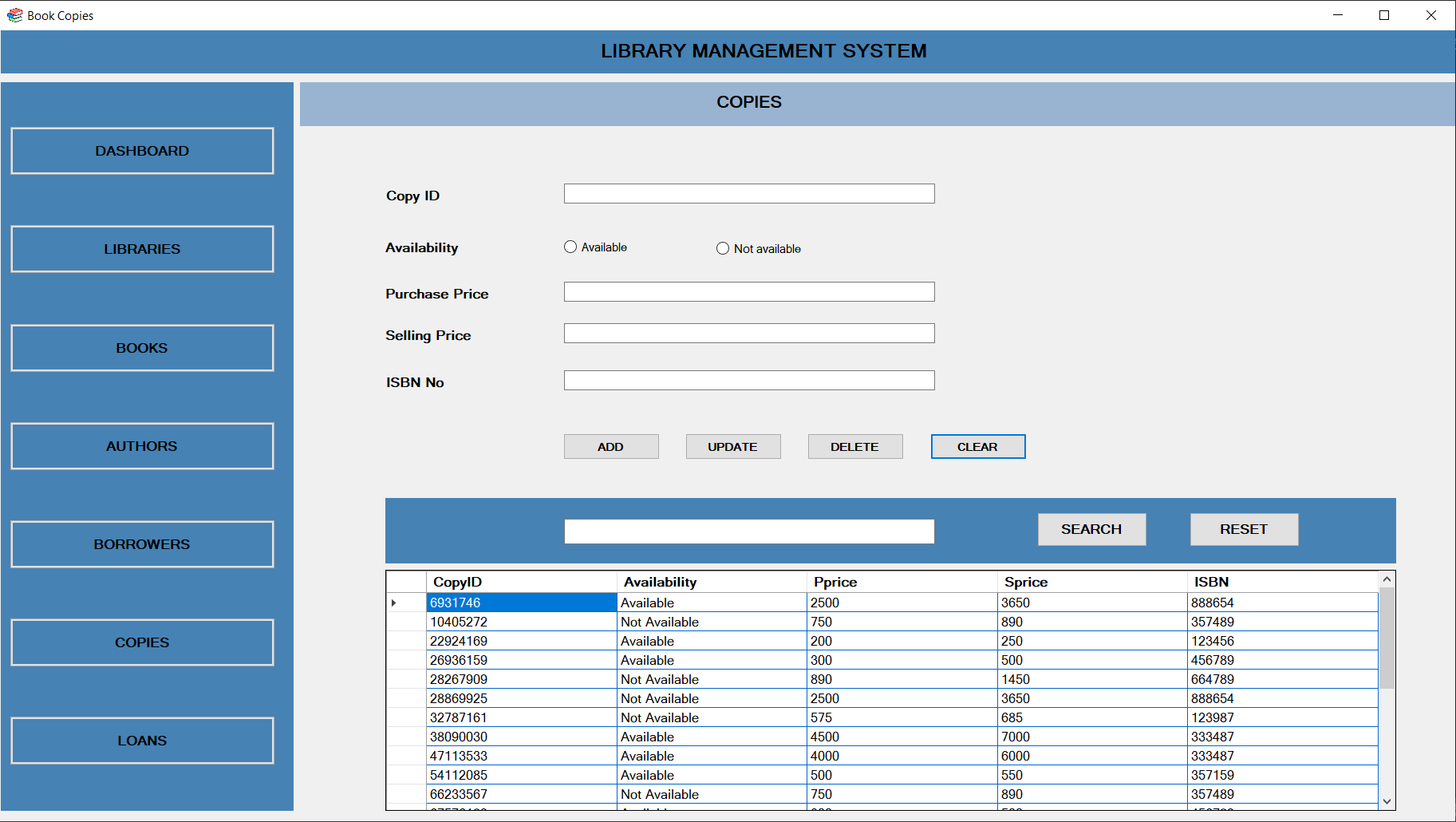
Library Table



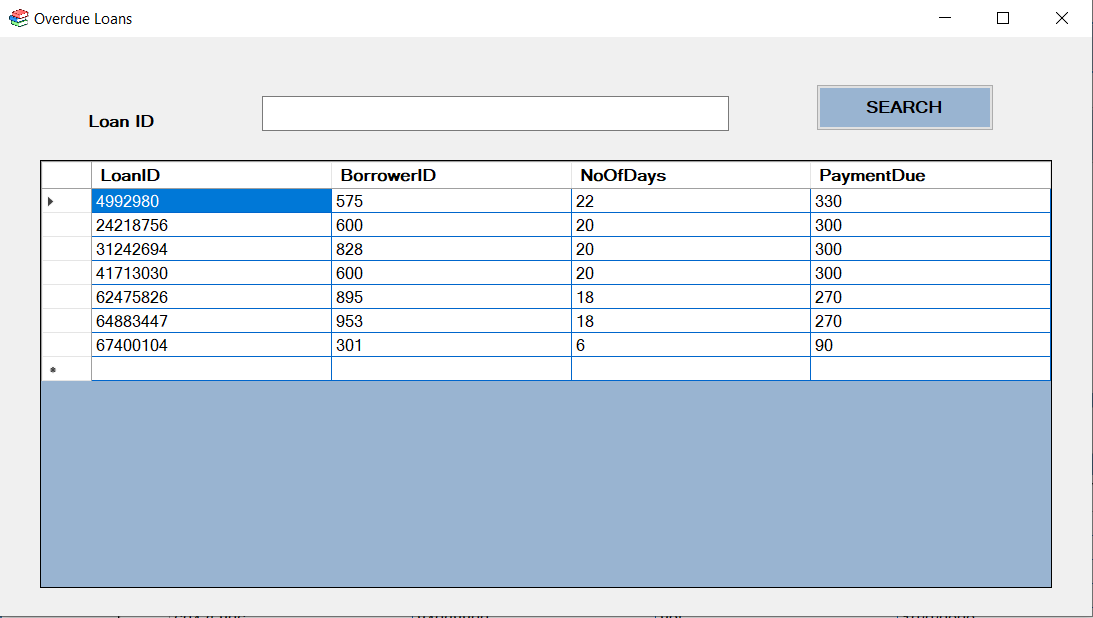
Loan Table



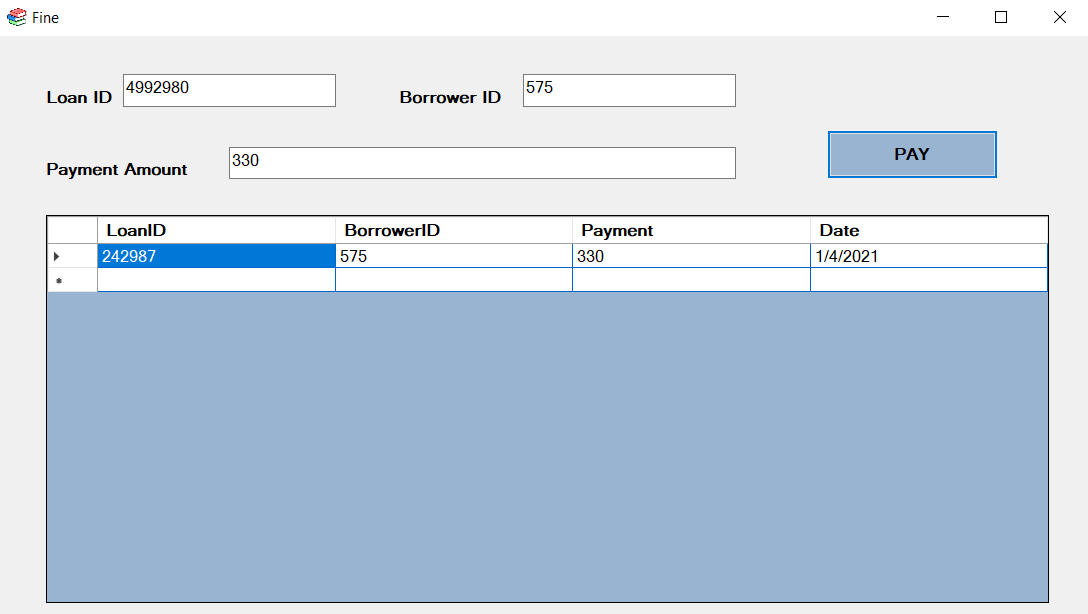
Copy Table



Overdue Table



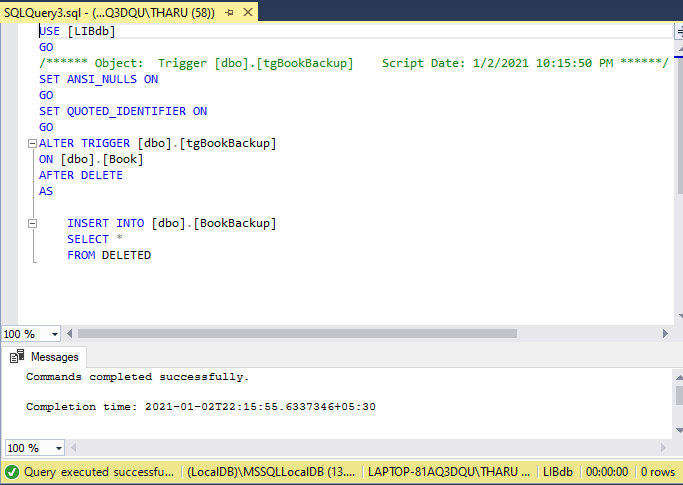
Payment Table



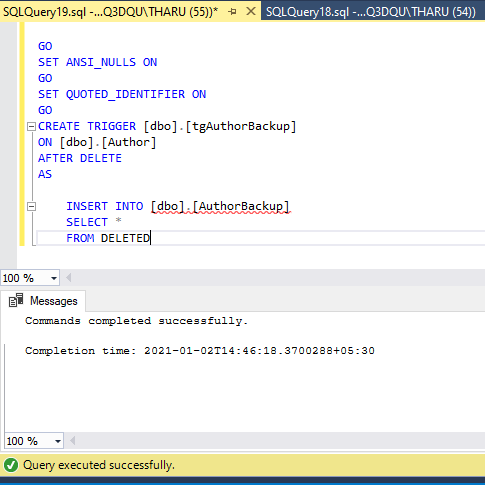
**Section 3**

* **Microsoft SQL Server Trigger statements**

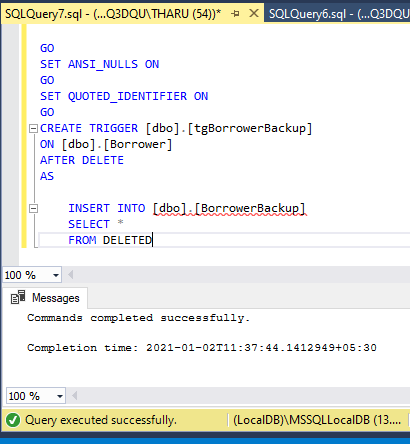
Trigger statement for Book Backup



Trigger statement for Author Backup

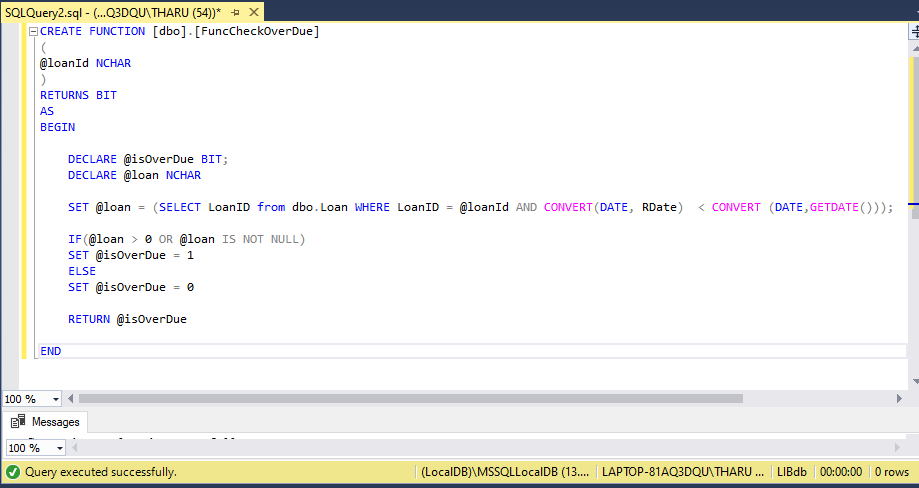
****

Trigger statement for Borrower Backup

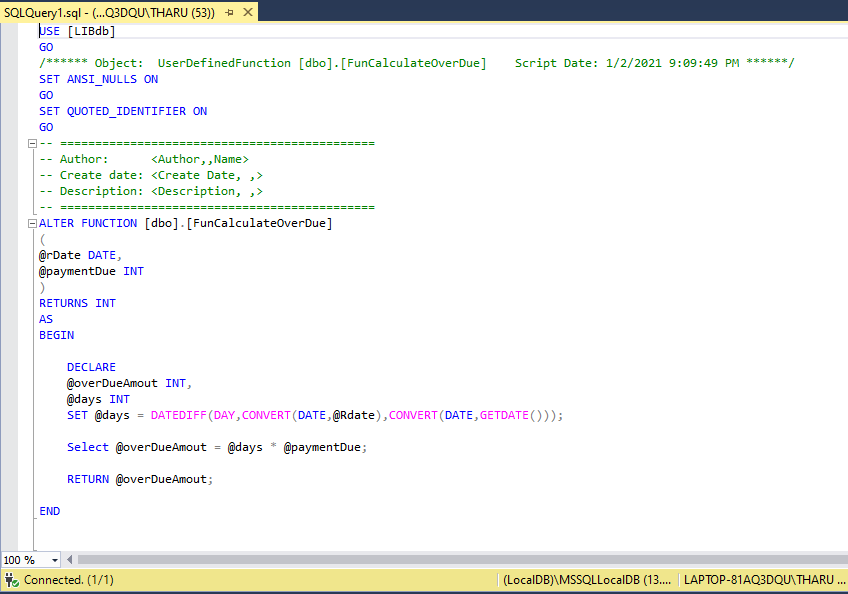
****

* **Microsoft SQL Server User Defined Functions**

Function to check Overdue

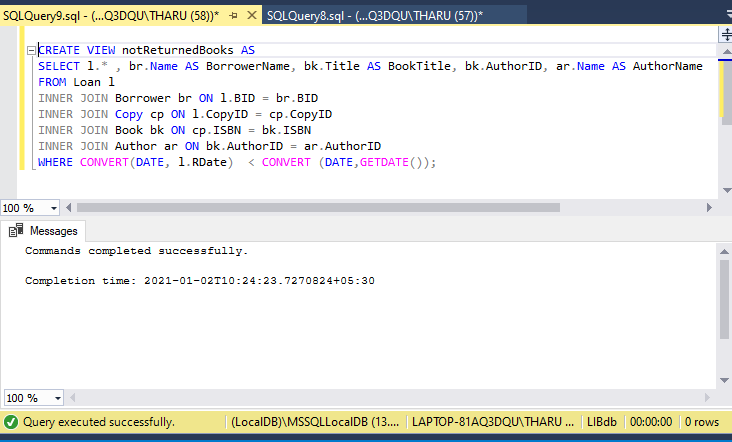
****

Function to calculate Overdue



* **Microsoft SQL Server View statements**

View statement for not returned books

****

* **Microsoft SQL Server Stored Procedures**

Data Insert Stored Procedures

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Data Update Stored Procedures

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

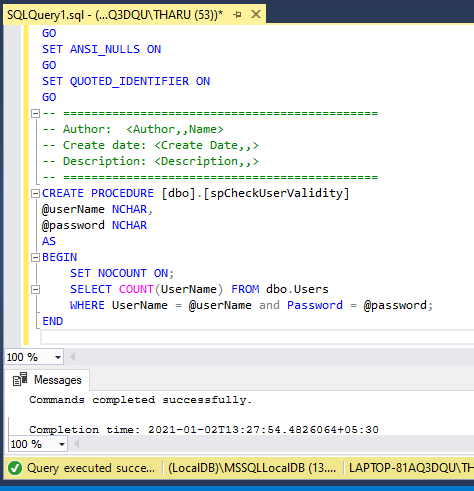
Data Delete Stored Procedures

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Data Search Stored Procedures

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Check Data Validity Stored Procedure



Process Overdue Stored Procedure



Not Returned Books Stored Procedure



**Section 4**

* **Critical Evaluation**

We can use this newly developed computerized library management system for the entry of a new book with the attributes ISBN number, Title, Number of copies and Name of the author and etc. This also records the information of the borrowers namely Borrower ID, Borrower name, Address, Contact number, Mail and etc. The lending dates and the return dates can be recorded easily by the ones who operate the system and can be found whether the return dates are overdue using the stored procedures. Here, by using user defined functions the fine amounts are automatically calculated when the dates are overdue. Triggers are used in the above system to keep backups of deleted data of books, borrowers as well as the authors.

User validity stored procedures are also used in the above system in order to validate the users of the library.

This is a user-friendly application along with a database that is developed in a way that can accept further modifications and in a way that any changes can be applied in the future. The interfaces will be modified in the future to make them more attractive.

* **Future Implementation**

This application is developed in a way that can accept further modifications.

Here, in the future all their information regarding books, their authors, the borrowers and etc. will be entered to the database of the public library in our hometown. According to the staff of the library further modifications will be applied for the system to make it more user friendly.

At present triggers are applied for only keeping backups of the books, borrowers (members) and the authors but this system can be developed further in order to keep backups of different other fields.

In the future this system will be developed further with a scanner that scans the ISBN code that is located at the end of the back cover of the book. So, there the system will be developed in to more user-friendly system that records all the information of the book as well as the borrowed and the return times and dates automatically.