IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Name: G.P.C.Hettiarachchi
Student Reference Number: 10707218

Module Code: ISAD253SL	Module Name: Databases
Coursework Title: ISAD253SL Group Assignment Submission	
Deadline Date: 04/01/2021	Member of staff responsible for coursework: Mr. Naji Saravanapavan
Programme: BSc (Hons) Software Engineering / BSc (Hons) Computer Security / BSc (Hons) Computer Networks	

Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook.

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

G.P.C. Hettiarachchi 10707218
T.M. Bogahawaththa 10707017
Y.D.N. Ranawaka 10707341
D.S.W. Gunasekera 10707207
Y.H. Munasinghe 10707038
W.D.L.S. Disanayaka 10707073

We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.

Signed on behalf of the group: 

Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***

Signed :

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I *have used/not used translation software.
If used, please state name of software……

Overall mark _____% Assessors Initials _____ Date_____

*Please delete as appropriate Sci/ps/d:/students/cwkfrontcover/2013/14

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

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.

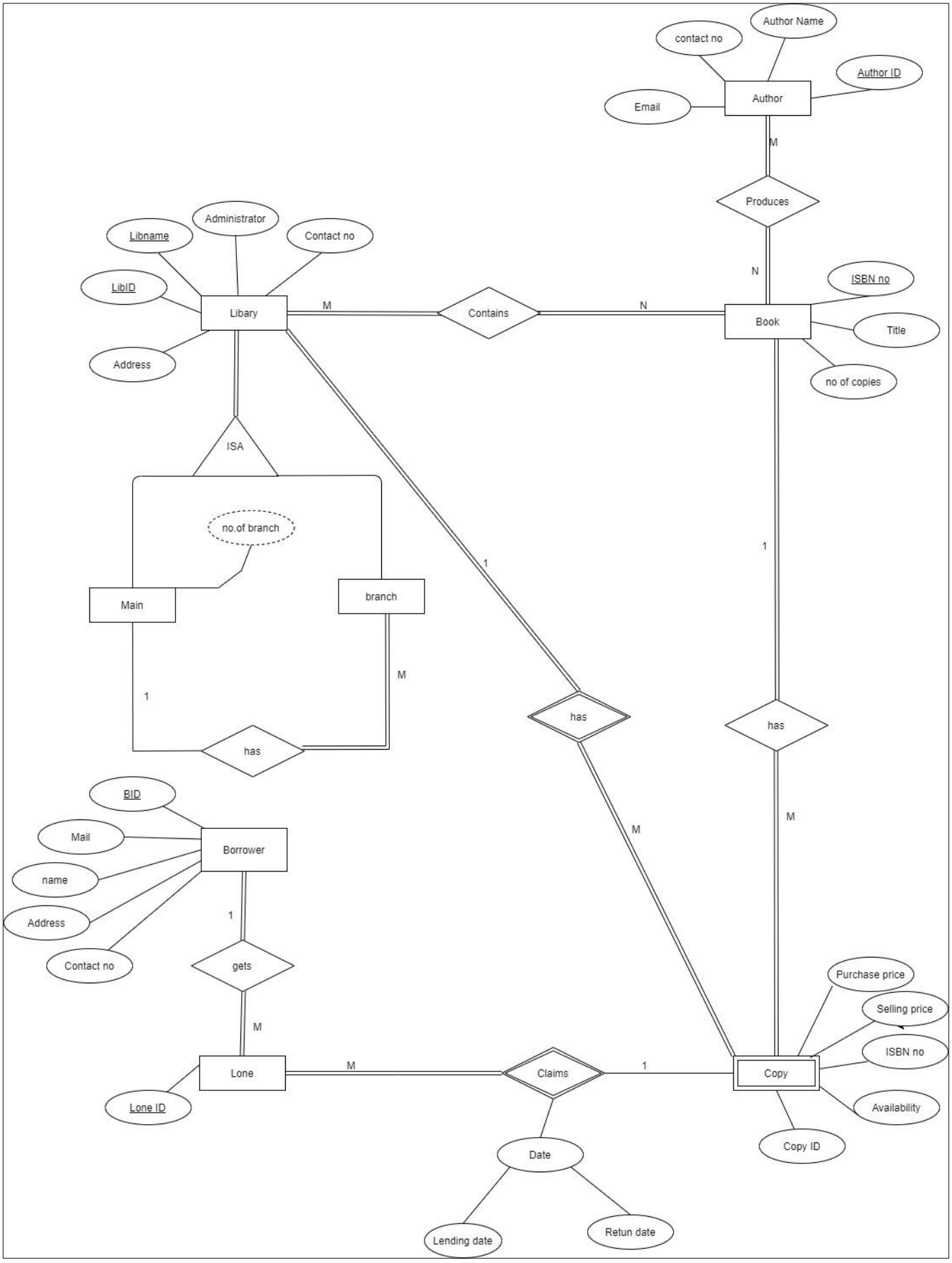
Objectives

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 each borrower can have either one or more books of different types and assuming that each loan can have more than one borrower.

4) 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.

5) Assuming that a copy is a weak entity related to the strong entity named as Book.

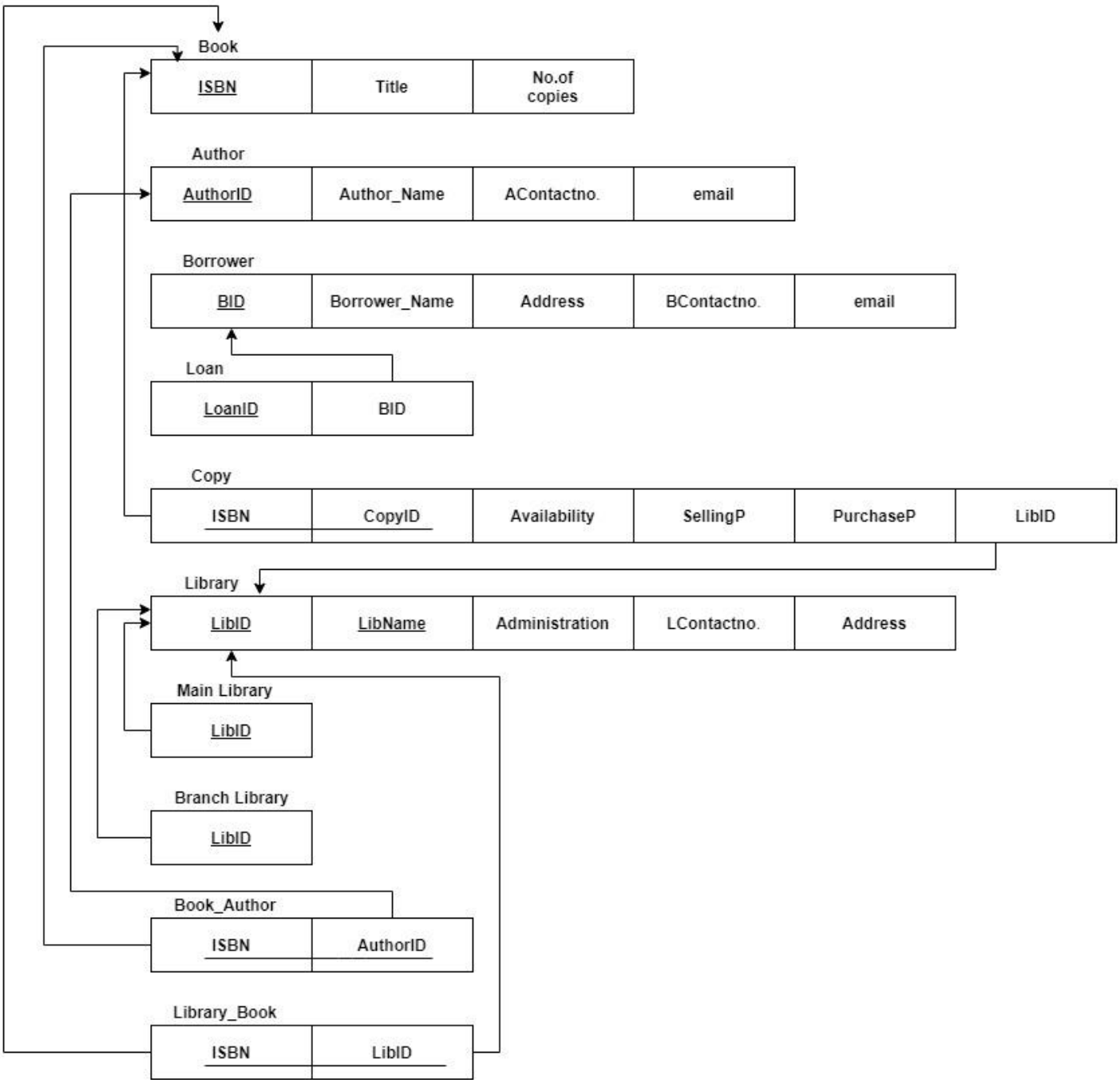
6) Assuming that a given library contains a number of copies of different books.

7) 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.

8) 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.

9) 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.

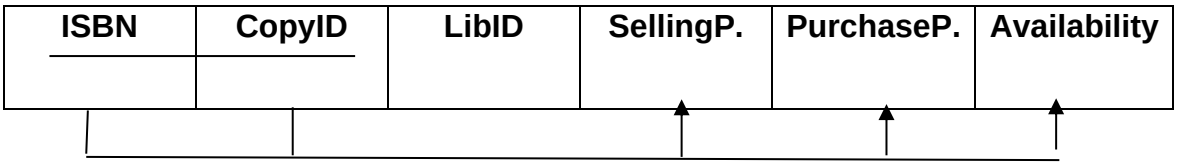
So, this is in 2NF.

3NF

There are no partial dependencies.

So, this is in 3NF.

Copy

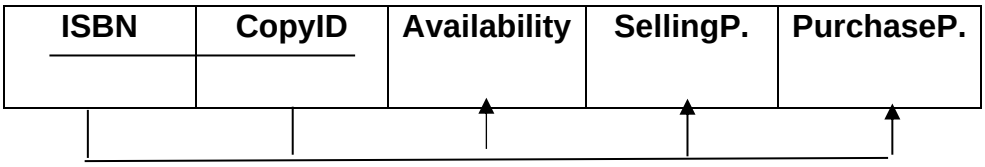


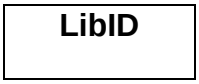
1NF

It has only single atomic values.

So, this table is in 1NF.

2NF





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

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.

```
SELECT
    IC.COLUMN_NAME,
    IC.Data_TYPE,
    EP.[Value] as [MS_Description],
    IKU.CONSTRAINT_NAME,
    ITC.CONSTRAINT_TYPE,
    IC.IS_NULLABLE
 FROM
    INFORMATION_SCHEMA.COLUMNS IC
    INNER JOIN sys.columns sc ON OBJECT_ID(QUOTENAME(IC.TABLE_SCHEMA) + '.' + QUOTENAME(IC.TABLE_NAME)) = sc.[object_id] AND IC.COLUMN_NAME = sc.name
    LEFT OUTER JOIN sys.extended_properties EP ON sc.[object_id] = EP.major_id AND sc.[column_id] = EP.minor_id AND EP.name = 'MS_Description' AND EP.class = 1
    LEFT OUTER JOIN INFORMATION_SCHEMA.KEY_COLUMN_USAGE IKU ON IKU.COLUMN_NAME = IC.COLUMN_NAME and IKU.TABLE_NAME = IC.TABLE_NAME and IKU.TABLE_CATALOG = IC.TABLE_CATALOG
    LEFT OUTER JOIN INFORMATION_SCHEMA.TABLE_CONSTRAINTS ITC ON ITC.TABLE_NAME = IKU.TABLE_NAME and ITC.CONSTRAINT_NAME = IKU.CONSTRAINT_NAME
WHERE IC.TABLE_CATALOG = 'LIBdb'
  and IC.TABLE_SCHEMA = 'dbo'
  and IC.TABLE_NAME = 'Author'
order by IC.ORDINAL_POSITION
```

Library Table

	COLUMN_NAME	Data_TYPE	MS_Description	CONSTRAINT_NAME	CONSTRAINT_TYPE	IS_NULLABLE
1	LibID	int	NULL	PK_Library	PRIMARY KEY	NO
2	Name	nvarchar	NULL	NULL	NULL	NO
3	Type	nchar	NULL	NULL	NULL	NO
4	Administrator	nvarchar	NULL	NULL	NULL	NO
5	Contact	nchar	NULL	NULL	NULL	NO
6	Address	nvarchar	NULL	NULL	NULL	NO

Book Table

	COLUMN_NAME	Data_TYPE	MS_Description	CONSTRAINT_NAME	CONSTRAINT_TYPE	IS_NULLABLE
1	ISBN	int	NULL	PK_Book	PRIMARY KEY	NO
2	Title	nvarchar	NULL	NULL	NULL	NO
3	Author	nvarchar	NULL	NULL	NULL	NO
4	Noofcopies	int	NULL	NULL	NULL	YES
5	Subject	nvarchar	NULL	NULL	NULL	YES

Author Table

	COLUMN_NAME	Data_TYPE	MS_Description	CONSTRAINT_NAME	CONSTRAINT_TYPE	IS_NULLABLE
1	AuthorID	int	NULL	PK_Author	PRIMARY KEY	NO
2	Name	nvarchar	NULL	NULL	NULL	YES
3	Contact	int	NULL	NULL	NULL	YES
4	Email	nvarchar	NULL	NULL	NULL	YES

Copy Table

	COLUMN_NAME	Data_TYPE	MS_Description	CONSTRAINT_NAME	CONSTRAINT_TYPE	IS_NULLABLE
1	CopyID	int	NULL	PK_Copy	PRIMARY KEY	NO
2	Availability	nchar	NULL	NULL	NULL	YES
3	Pprice	int	NULL	NULL	NULL	YES
4	Sprice	int	NULL	NULL	NULL	YES
5	ISBN	int	NULL	NULL	NULL	NO

Loan Table

	COLUMN_NAME	Data_TYPE	MS_Description	CONSTRAINT_NAME	CONSTRAINT_TYPE	IS_NULLABLE
1	LoanID	int	NULL	PK_Loan	PRIMARY KEY	NO
2	CopyID	int	NULL	FK_Loan	FOREIGN KEY	NO
3	BID	nchar	NULL	NULL	NULL	NO
4	LDate	date	NULL	NULL	NULL	YES
5	RDate	date	NULL	NULL	NULL	YES

Overdue Table

	COLUMN_NAME	Data_TYPE	MS_Description	CONSTRAINT_NAME	CONSTRAINT_TYPE	IS_NULLABLE
1	LoanID	int	NULL	PK__Overdue__4F5AD437DB5E159F	PRIMARY KEY	NO
2	BorrowerID	nvarchar	NULL	NULL	NULL	YES
3	NoOfDays	int	NULL	NULL	NULL	YES
4	PaymentDue	int	NULL	NULL	NULL	YES

Payment Table

	COLUMN_NAME	Data_TYPE	MS_Description	CONSTRAINT_NAME	CONSTRAINT_TYPE	IS_NULLABLE
1	LoanID	int	NULL	PK__Payment__4F5AD437C3DB13C3	PRIMARY KEY	NO
2	BorrowerID	int	NULL	NULL	NULL	NO
3	Payment	int	NULL	NULL	NULL	NO
4	Date	date	NULL	NULL	NULL	NO

Author Backup Table

	COLUMN_NAME	Data_TYPE	MS_Description	CONSTRAINT_NAME	CONSTRAINT_TYPE	IS_NULLABLE
1	AuthorID	int	NULL	PK_AuthorBackup	PRIMARY KEY	NO
2	Name	nvarchar	NULL	NULL	NULL	YES
3	Contact	nchar	NULL	NULL	NULL	YES
4	Email	nvarchar	NULL	NULL	NULL	YES

Book Backup Table

	COLUMN_NAME	Data_TYPE	MS_Description	CONSTRAINT_NAME	CONSTRAINT_TYPE	IS_NULLABLE
1	ISBN	bigint	NULL	PK_BookBackup	PRIMARY KEY	NO
2	Title	nchar	NULL	NULL	NULL	NO
3	Author	nchar	NULL	NULL	NULL	NO
4	No.of.copies	nchar	NULL	NULL	NULL	YES
5	Subject	nchar	NULL	NULL	NULL	YES

Author Backup Table

	COLUMN_NAME	Data_TYPE	MS_Description	CONSTRAINT_NAME	CONSTRAINT_TYPE	IS_NULLABLE
1	BID	int	NULL	PK_BorrowerBackup	PRIMARY KEY	NO
2	Name	nvarchar	NULL	NULL	NULL	NO
3	Contact	nchar	NULL	NULL	NULL	YES
4	Address	nvarchar	NULL	NULL	NULL	YES
5	Email	nvarchar	NULL	NULL	NULL	YES

Section 2

➢ Microsoft SQL Server Create Table statements with related Constraints

Library Table

```
USE [LIBdb]
GO

/****** Object:  Table [dbo].[Library]    Script Date: 1/2/2021 2:38:58 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Library](
    [LibID] [nchar](10) NOT NULL,
    [Name] [nchar](60) NOT NULL,
    [Type] [nchar](10) NOT NULL,
    [Administrator] [nchar](60) NOT NULL,
    [Contact] [nchar](10) NOT NULL,
    [Address] [nchar](70) NOT NULL,
 CONSTRAINT [PK_Library] PRIMARY KEY CLUSTERED
(
    [LibID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

User Table

```
USE [LIBdb]
GO

/****** Object:  Table [dbo].[Users]    Script Date: 1/2/2021 2:40:58 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Users](
    [Username] [nchar](10) NOT NULL,
    [Password] [nchar](10) NOT NULL,
PRIMARY KEY CLUSTERED
(
    [Username] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Author Table

```
USE [LIBdb]
GO

/****** Object:  Table [dbo].[Author]    Script Date: 1/2/2021 2:25:38 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Author](
    [AuthorID] [nchar](10) NOT NULL,
    [Name] [nchar](60) NULL,
    [Contact] [nchar](10) NULL,
    [Email] [nchar](30) NULL,
 CONSTRAINT [PK_Author] PRIMARY KEY CLUSTERED
(
    [AuthorID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Book Table

```
USE [LIBdb]
GO

/****** Object:  Table [dbo].[Book]    Script Date: 1/2/2021 2:36:58 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Book](
    [ISBN] [bigint] NOT NULL,
    [Title] [nchar](30) NOT NULL,
    [AuthorID] [nchar](10) NOT NULL,
    [No.of.copies] [nchar](10) NULL,
    [Subject] [nchar](10) NULL,
 CONSTRAINT [PK_Book] PRIMARY KEY CLUSTERED
(
    [ISBN] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Book]  WITH CHECK ADD  CONSTRAINT [FK_Book] FOREIGN KEY([AuthorID])
REFERENCES [dbo].[Author] ([AuthorID])
GO

ALTER TABLE [dbo].[Book] CHECK CONSTRAINT [FK_Book]
GO
```

Borrower Table

```
USE [LIBdb]
GO

/****** Object:  Table [dbo].[Borrower]    Script Date: 1/2/2021 2:37:47 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Borrower](
    [BID] [nchar](10) NOT NULL,
    [Name] [nchar](60) NOT NULL,
    [Contact] [nchar](10) NULL,
    [Address] [nchar](70) NULL,
    [Email] [nchar](30) NULL,
 CONSTRAINT [PK_Borrower] PRIMARY KEY CLUSTERED
(
    [BID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Copy Table

```
USE [LIBdb]
GO

/****** Object:  Table [dbo].[Copy]    Script Date: 1/2/2021 2:38:20 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Copy](
    [CopyID] [nchar](10) NOT NULL,
    [Availability] [nchar](10) NULL,
    [Pprice] [nchar](10) NULL,
    [Sprice] [nchar](10) NULL,
    [ISBN] [nchar](10) NOT NULL,
 CONSTRAINT [PK_Copy] PRIMARY KEY CLUSTERED
(
    [CopyID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Loan Table

```
USE [LIBdb]
GO

/****** Object:  Table [dbo].[Loan]    Script Date: 1/2/2021 2:39:28 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Loan](
    [LoanID] [nchar](10) NOT NULL,
    [CopyID] [nchar](10) NOT NULL,
    [BID] [nchar](10) NOT NULL,
    [LDate] [date] NULL,
    [RDate] [date] NULL,
 CONSTRAINT [PK_Loan] PRIMARY KEY CLUSTERED
(
    [LoanID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Loan]  WITH CHECK ADD  CONSTRAINT [FK_Loan] FOREIGN KEY([CopyID])
REFERENCES [dbo].[Copy] ([CopyID])
GO

ALTER TABLE [dbo].[Loan] CHECK CONSTRAINT [FK_Loan]
GO
```

Overdue Table

```
USE [LIBdb]
GO

/****** Object:  Table [dbo].[Overdue]    Script Date: 1/2/2021 2:39:56 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Overdue](
    [LoanID] [int] NOT NULL,
    [BorrowerID] [nvarchar](50) NULL,
    [NoOfDays] [int] NULL,
    [PaymentDue] [int] NULL,
PRIMARY KEY CLUSTERED
(
    [LoanID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Payment Table

```
USE [LIBdb]
GO

/****** Object:  Table [dbo].[Payment]    Script Date: 1/2/2021 2:40:30 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Payment](
    [LoanID] [int] NOT NULL,
    [BorrowerID] [int] NOT NULL,
    [Payment] [int] NOT NULL,
    [Date] [date] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [LoanID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Book Backup Table

```
USE [LIBdb]
GO

/****** Object:  Table [dbo].[BookBackup]    Script Date: 1/2/2021 2:41:28 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[BookBackup](
    [ISBN] [bigint] NOT NULL,
    [Title] [nchar](30) NOT NULL,
    [Author] [nchar](60) NOT NULL,
    [No.of.copies] [nchar](10) NULL,
    [Subject] [nchar](10) NULL,
 CONSTRAINT [PK_BookBackup] PRIMARY KEY CLUSTERED
(
    [ISBN] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

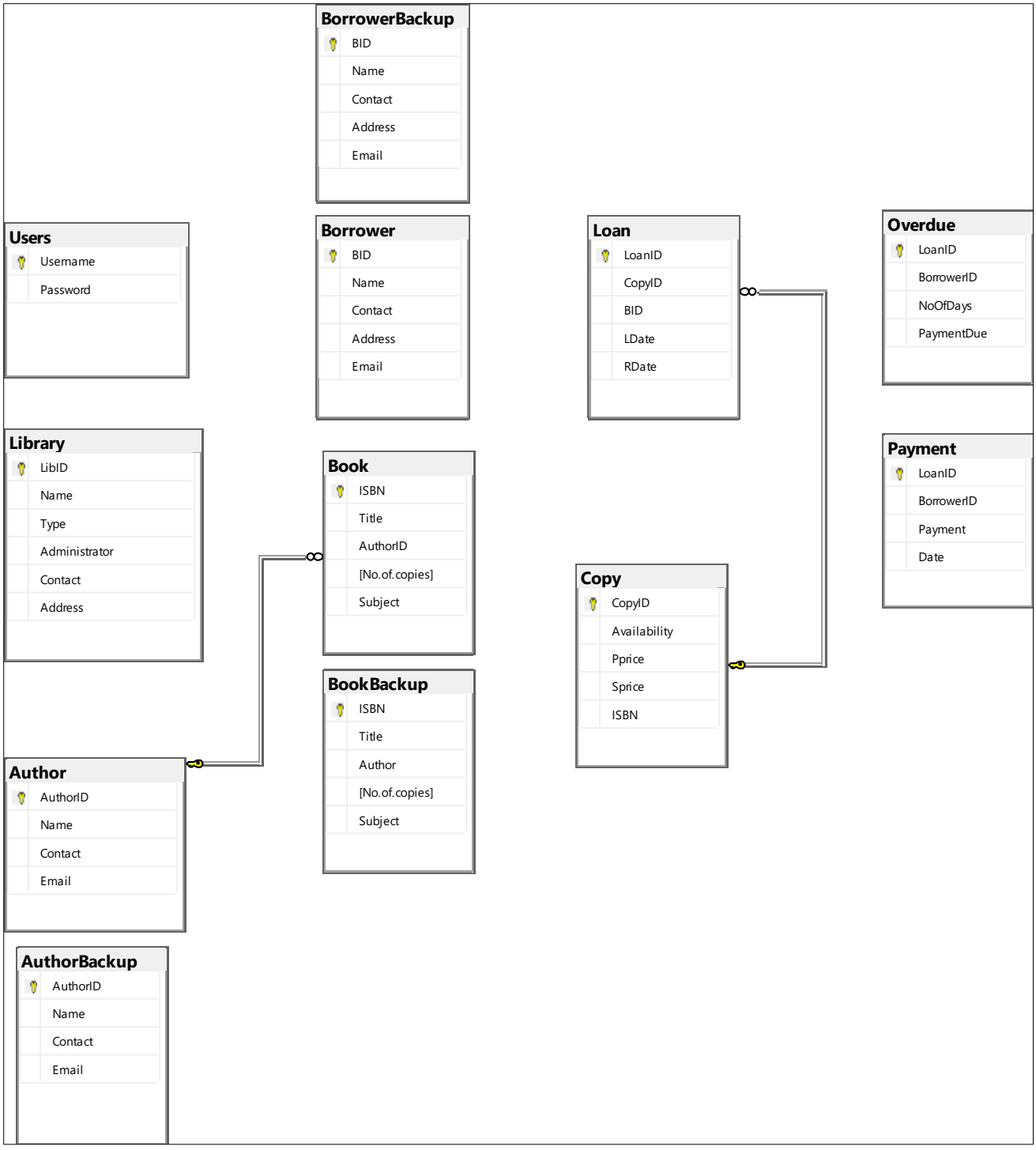
Author Backup Table

```
USE [LIBdb]
GO

/****** Object:  Table [dbo].[AuthorBackup]    Script Date: 1/2/2021 2:48:07 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[AuthorBackup](
    [AuthorID] [nchar](10) NOT NULL,
    [Name] [nchar](60) NULL,
    [Contact] [nchar](10) NULL,
    [Email] [nchar](30) NULL,
 CONSTRAINT [PK_AuthorBackup] PRIMARY KEY CLUSTERED
(
    [AuthorID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Borrower Backup Table

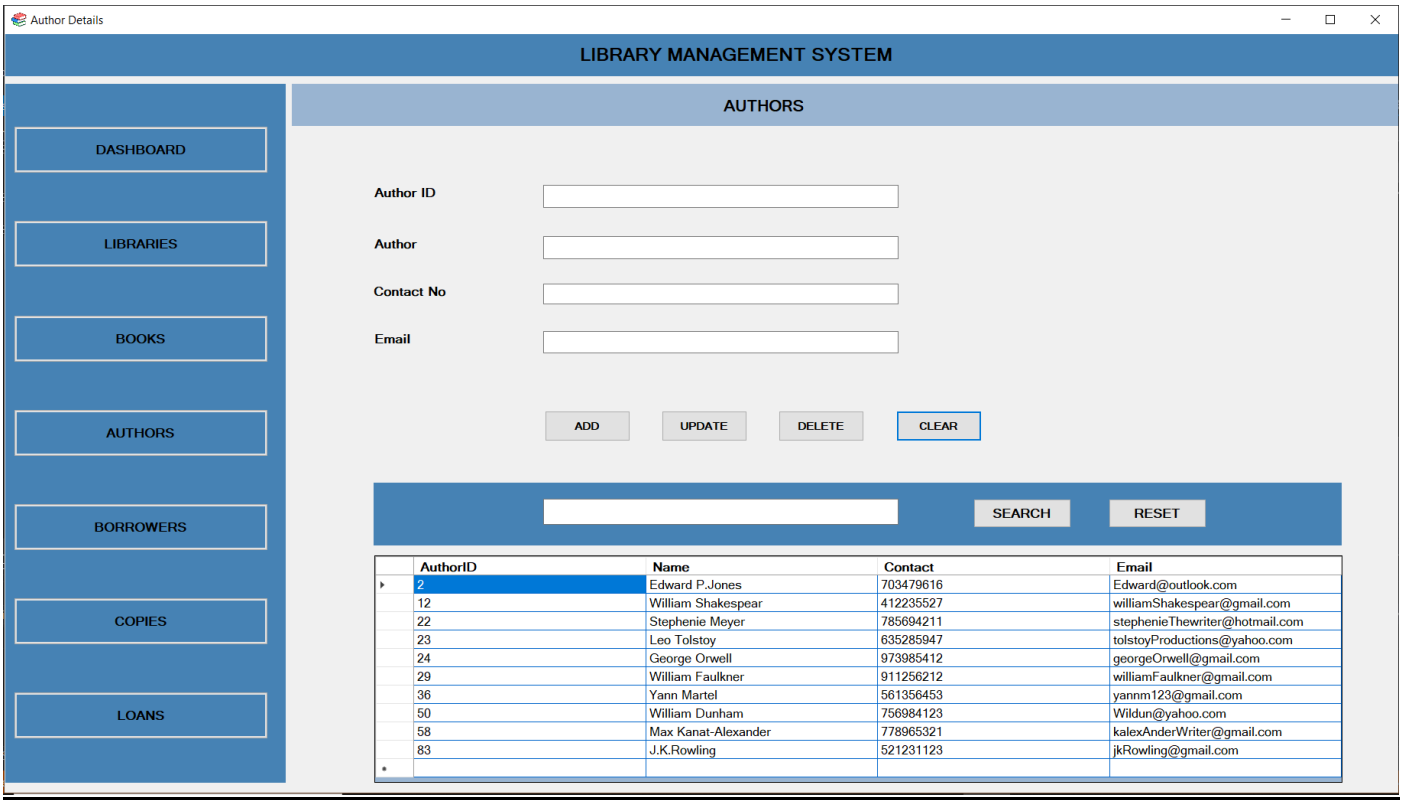
```
USE [LIBdb]
GO

/****** Object:  Table [dbo].[BorrowerBackup]    Script Date: 1/2/2021 2:41:58 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[BorrowerBackup](
    [BID] [nchar](10) NOT NULL,
    [Name] [nchar](60) NOT NULL,
    [Contact] [nchar](10) NULL,
    [Address] [nchar](70) NULL,
    [Email] [nchar](30) NULL,
 CONSTRAINT [PK_BorrowerBackup] PRIMARY KEY CLUSTERED
(
    [BID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

➢ Database Diagram



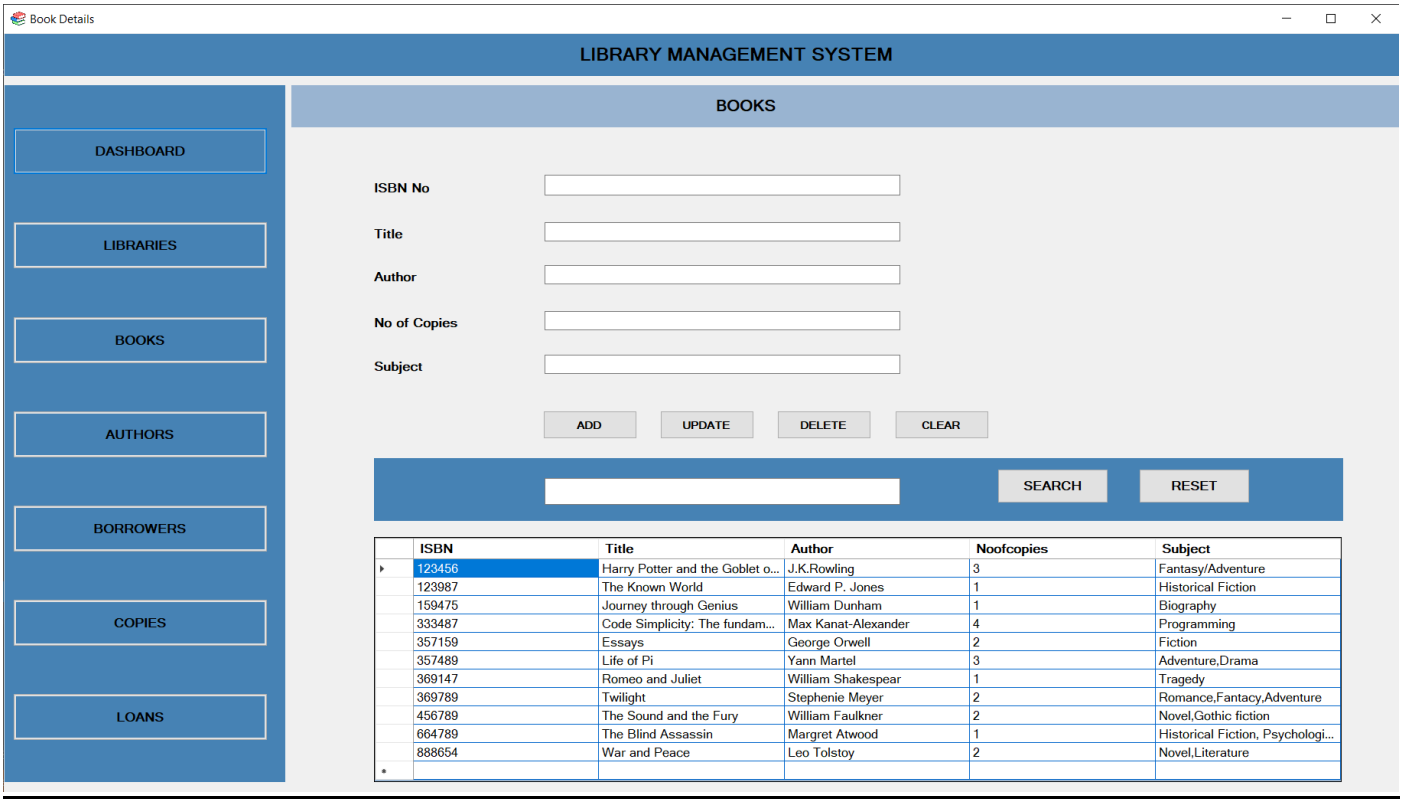
➢ Sample Records

Author Table



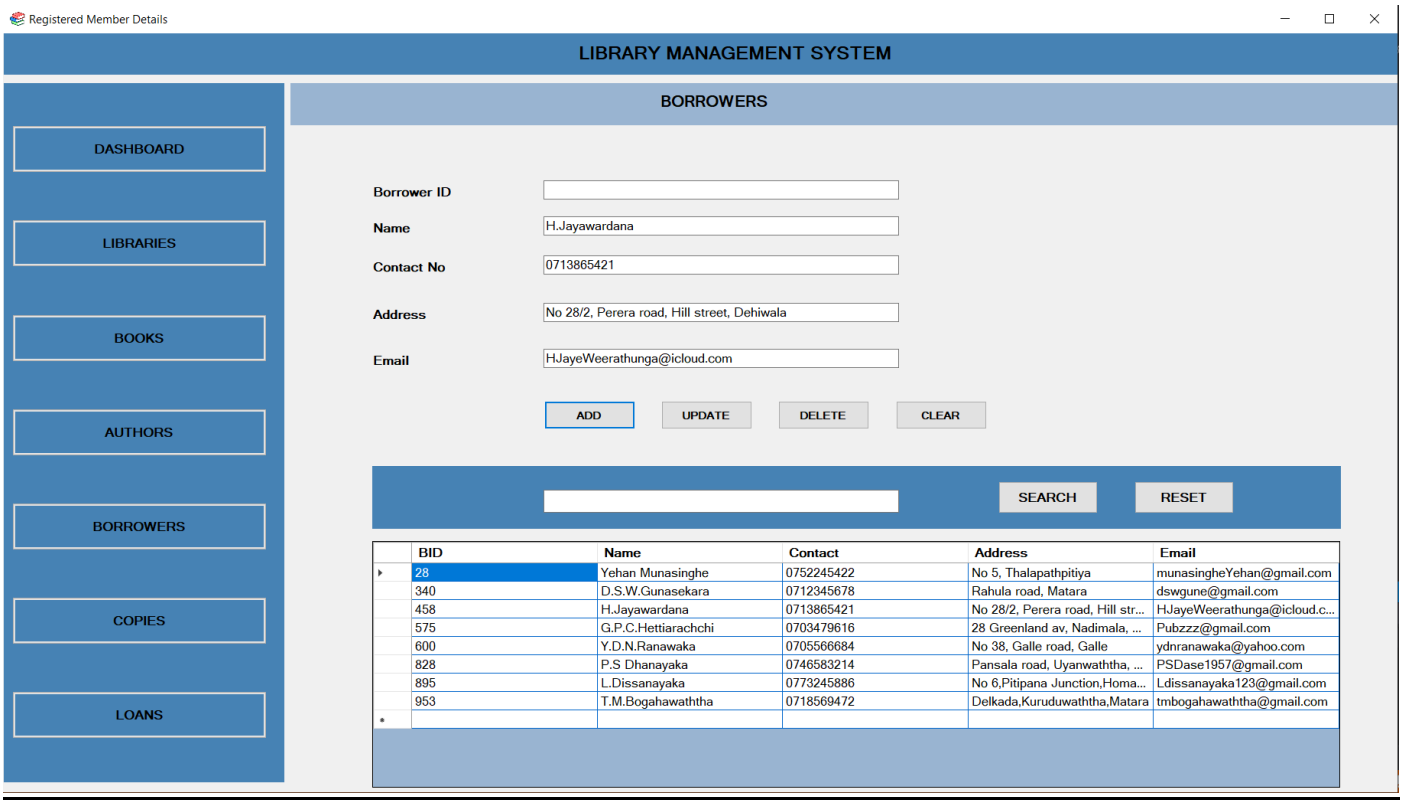
AuthorID	Name	Contact	Email
2	Edward P.Jones	703479616	Edward@outlook.com
12	William Shakespear	412235527	williamShakespear@gmail.com
22	Stephenie Meyer	785694211	stephenieThewriter@hotmail.com
23	Leo Tolstoy	635285947	tolstoyProductions@yahoo.com
24	George Orwell	973985412	georgeOrwell@gmail.com
29	William Faulkner	911256212	williamFaulkner@gmail.com
36	Yann Martel	561356453	yannm123@gmail.com
50	William Dunham	756984123	Wildun@yahoo.com
58	Max Kanat-Alexander	778965321	kalexAnderWriter@gmail.com
83	J.K.Rowling	521231123	jkRowling@gmail.com

Book Table

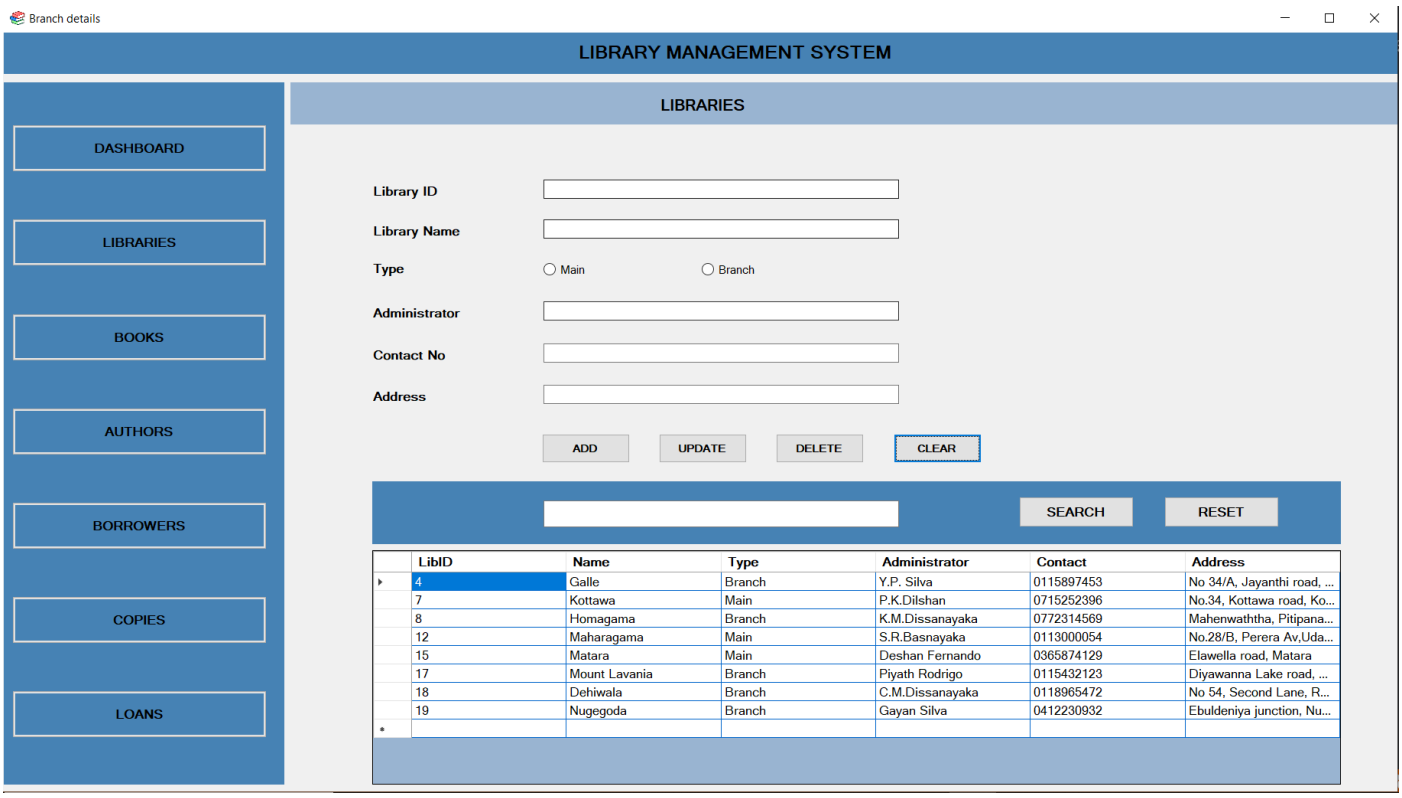


ISBN	Title	Author	Noofcopies	Subject
123456	Harry Potter and the Goblet o...	J.K.Rowling	3	Fantasy/Adventure
123987	The Known World	Edward P. Jones	1	Historical Fiction
159475	Journey through Genius	William Dunham	1	Biography
333487	Code Simplicity: The fundam...	Max Kanat-Alexander	4	Programming
357159	Essays	George Orwell	2	Fiction
357489	Life of Pi	Yann Martel	3	Adventure,Drama
369147	Romeo and Juliet	William Shakespear	1	Tragedy
369789	Twilight	Stephenie Meyer	2	Romance,Fantacy,Adventure
456789	The Sound and the Fury	William Faulkner	2	Novel,Gothic fiction
664789	The Blind Assassin	Margret Atwood	1	Historical Fiction, Psychologi...
888654	War and Peace	Leo Tolstoy	2	Novel,Literature

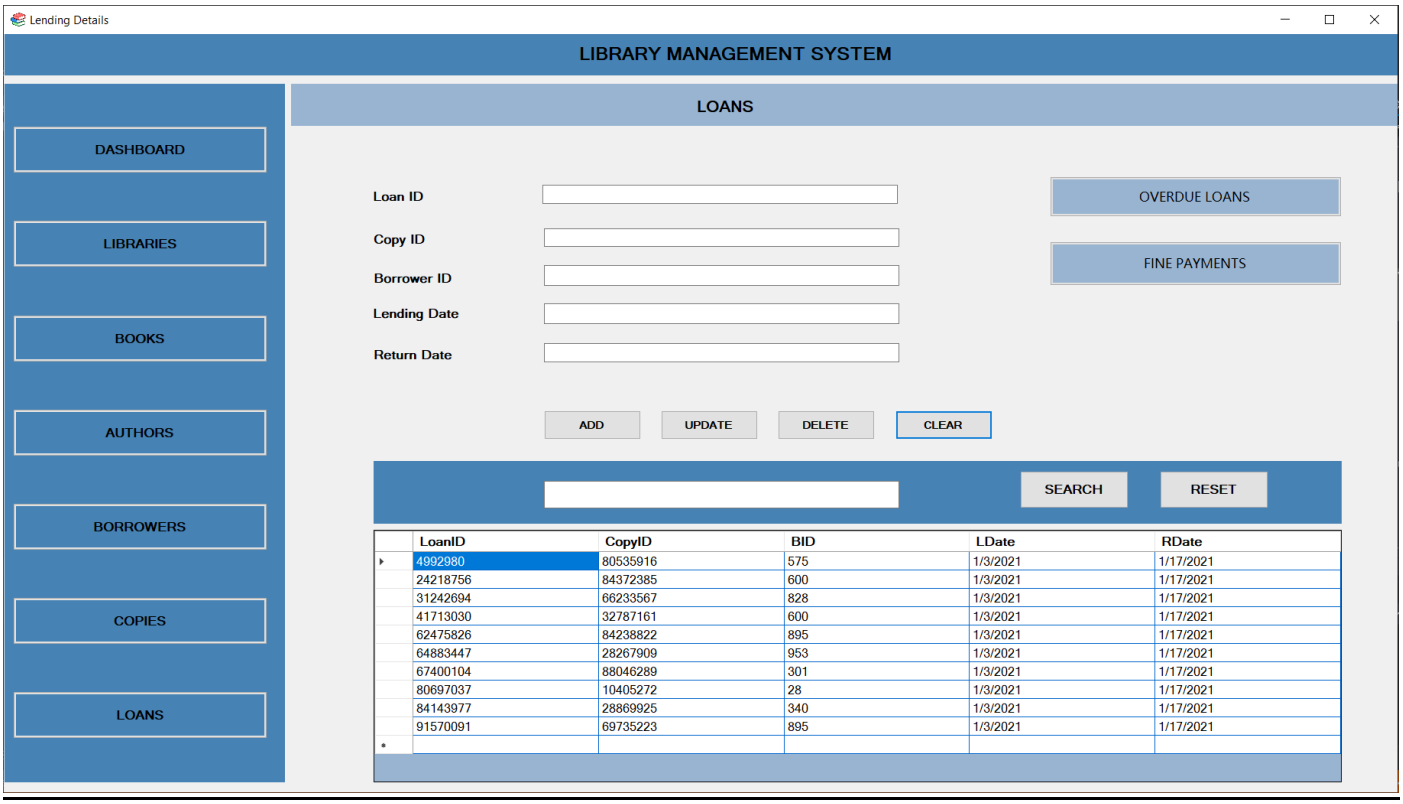
Borrower Table



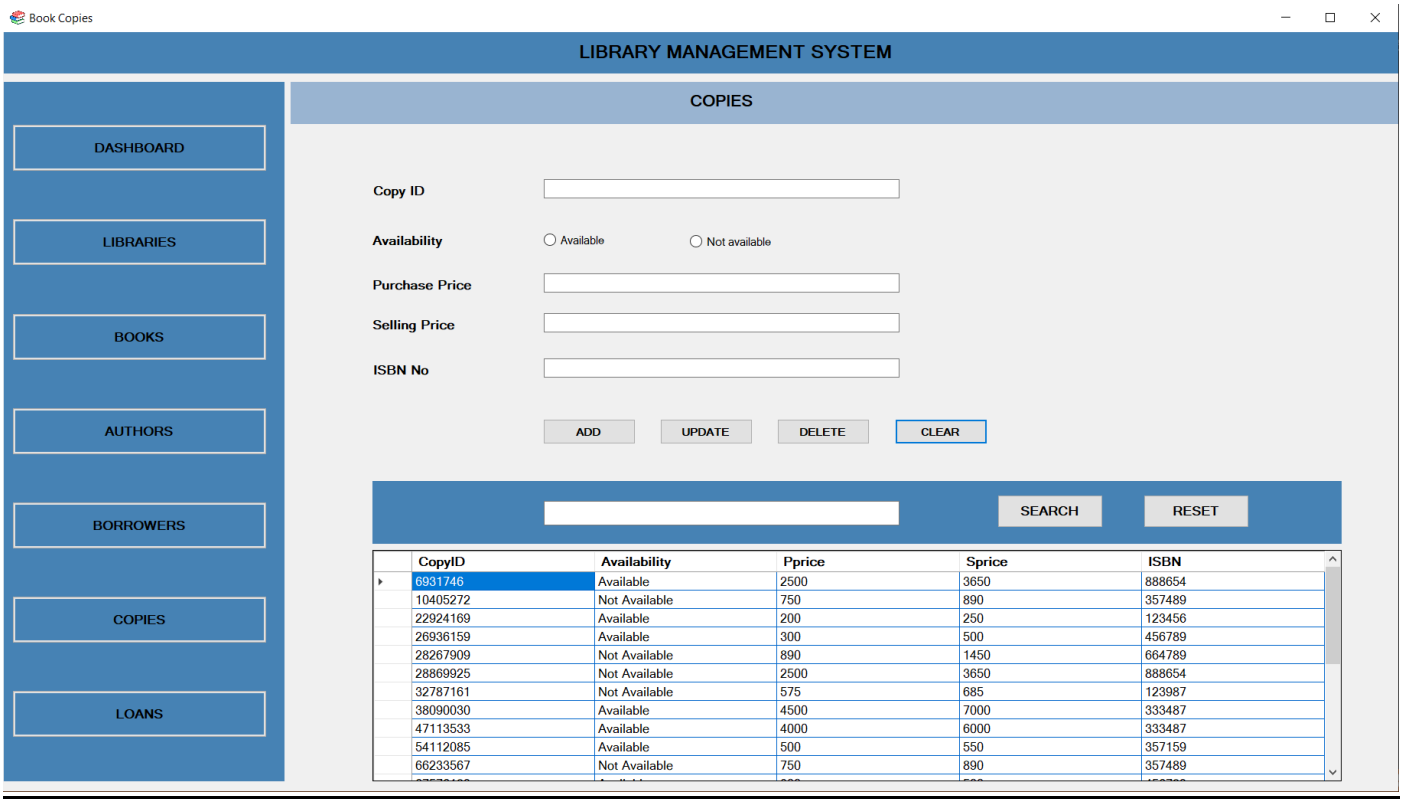
Library Table



Loan Table



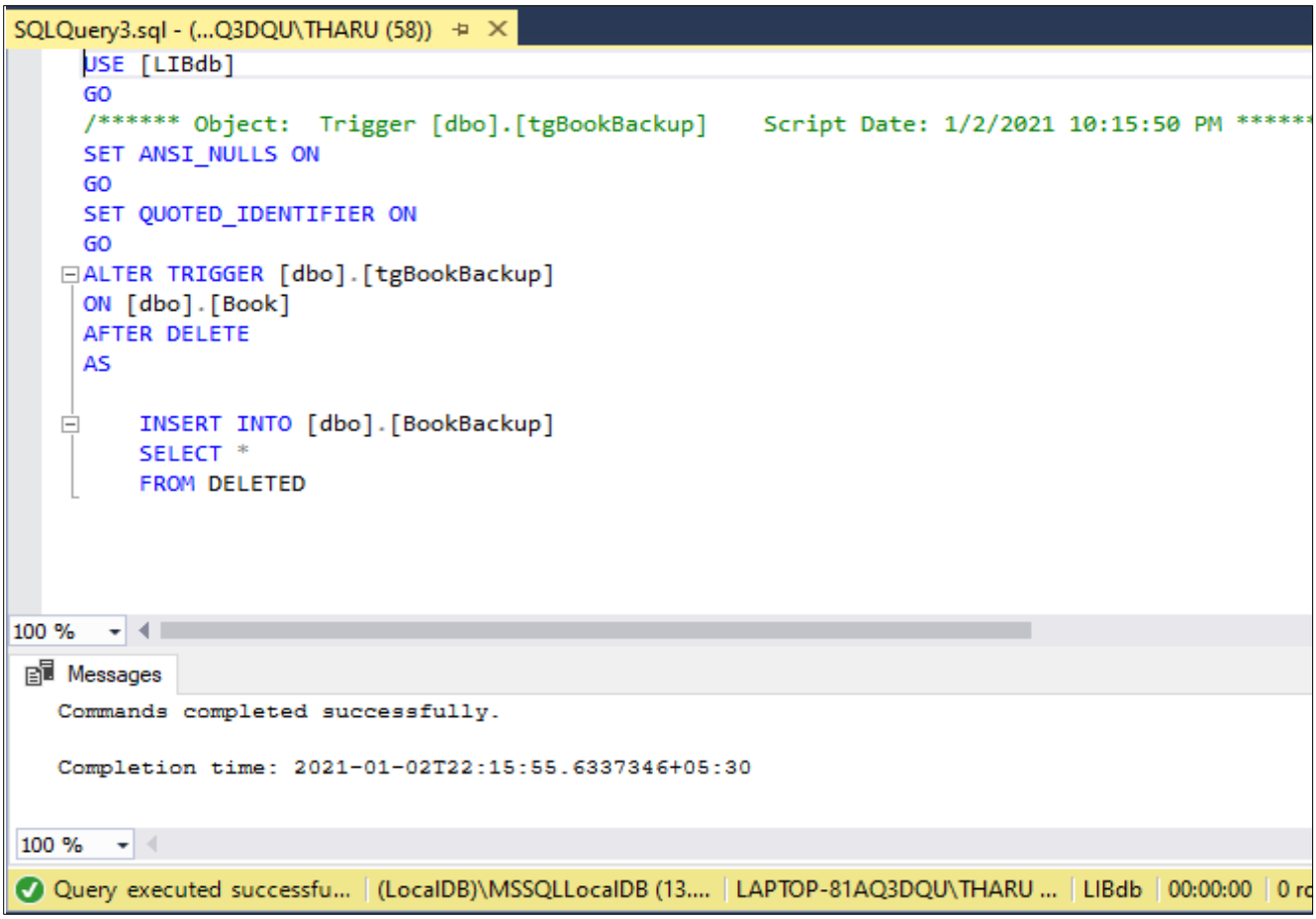
Copy Table



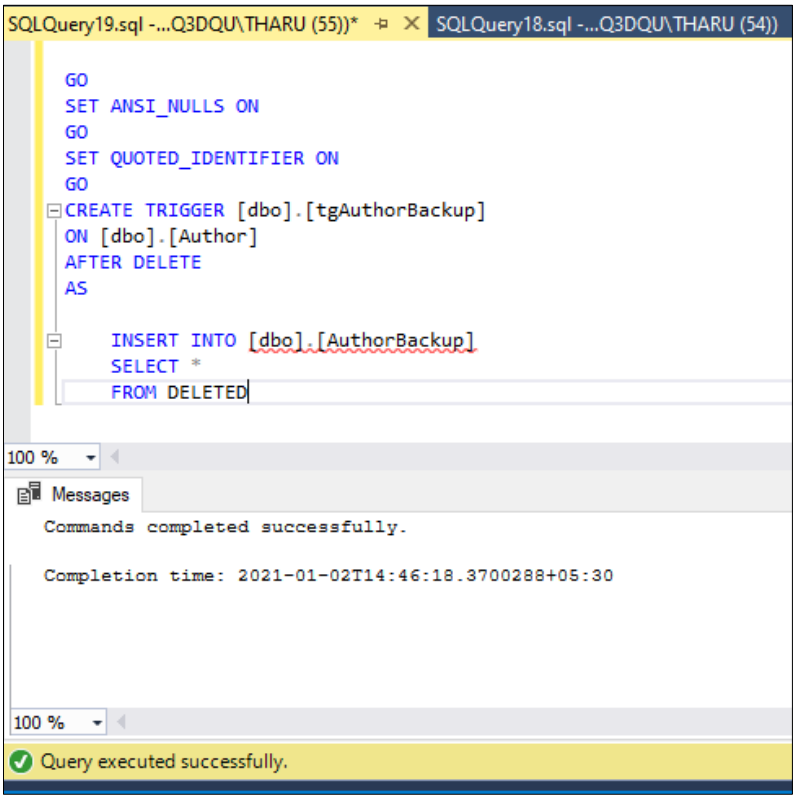
Section 3

➢ Microsoft SQL Server Trigger statements

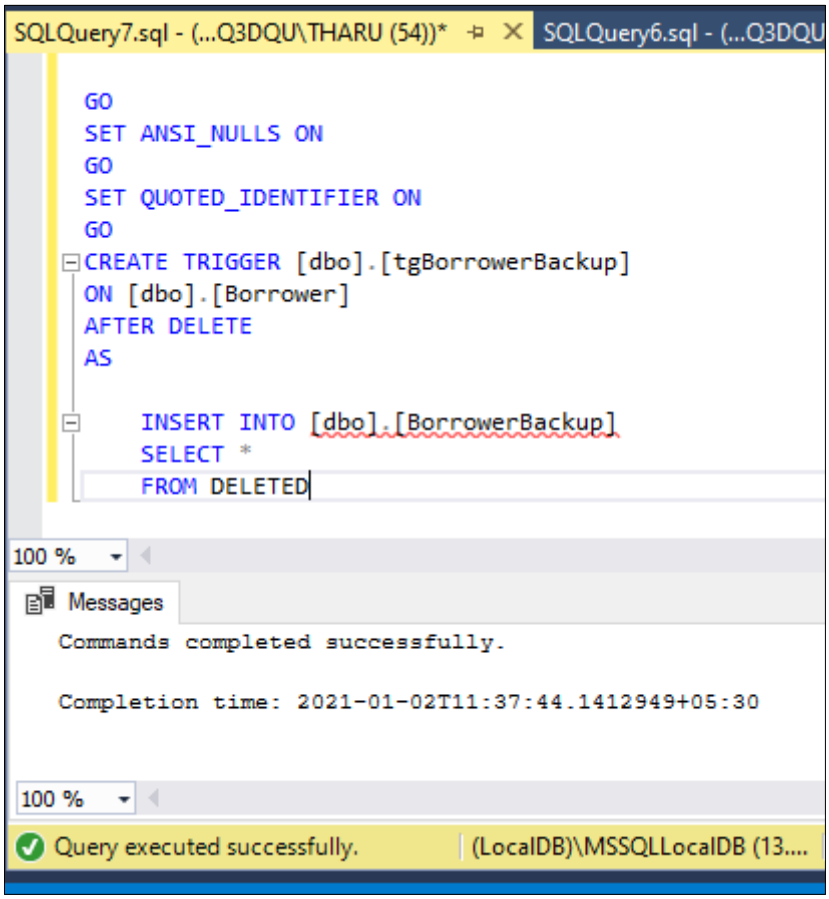
Trigger statement for Book Backup



Trigger statement for Author Backup



Trigger statement for Borrower Backup



➢ Microsoft SQL Server User Defined Functions

Function to check Overdue

SQLQuery2.sql - (...Q3DQU\THARU (54))*

```
CREATE FUNCTION [dbo].[FuncCheckOverDue]
(
@loanId NCHAR
)
RETURNS BIT
AS
BEGIN

    DECLARE @isOverDue BIT;
    DECLARE @loan NCHAR

    SET @loan = (SELECT LoanID from dbo.Loan WHERE LoanID = @loanId AND CONVERT(DATE, RDate)  < CONVERT (DATE,GETDATE()));

    IF(@loan > 0 OR @loan IS NOT NULL)
    SET @isOverDue = 1
    ELSE
    SET @isOverDue = 0

    RETURN @isOverDue

END
```

100 %

Messages

100 %

Query executed successfully. | (LocalDB)\MSSQLLocalDB (13.... | LAPTOP-81AQ3DQU\THARU ... | LIBdb | 00:00:00 | 0 rows

Function to calculate Overdue

SQLQuery1.sql - (...Q3DQU\THARU (53))

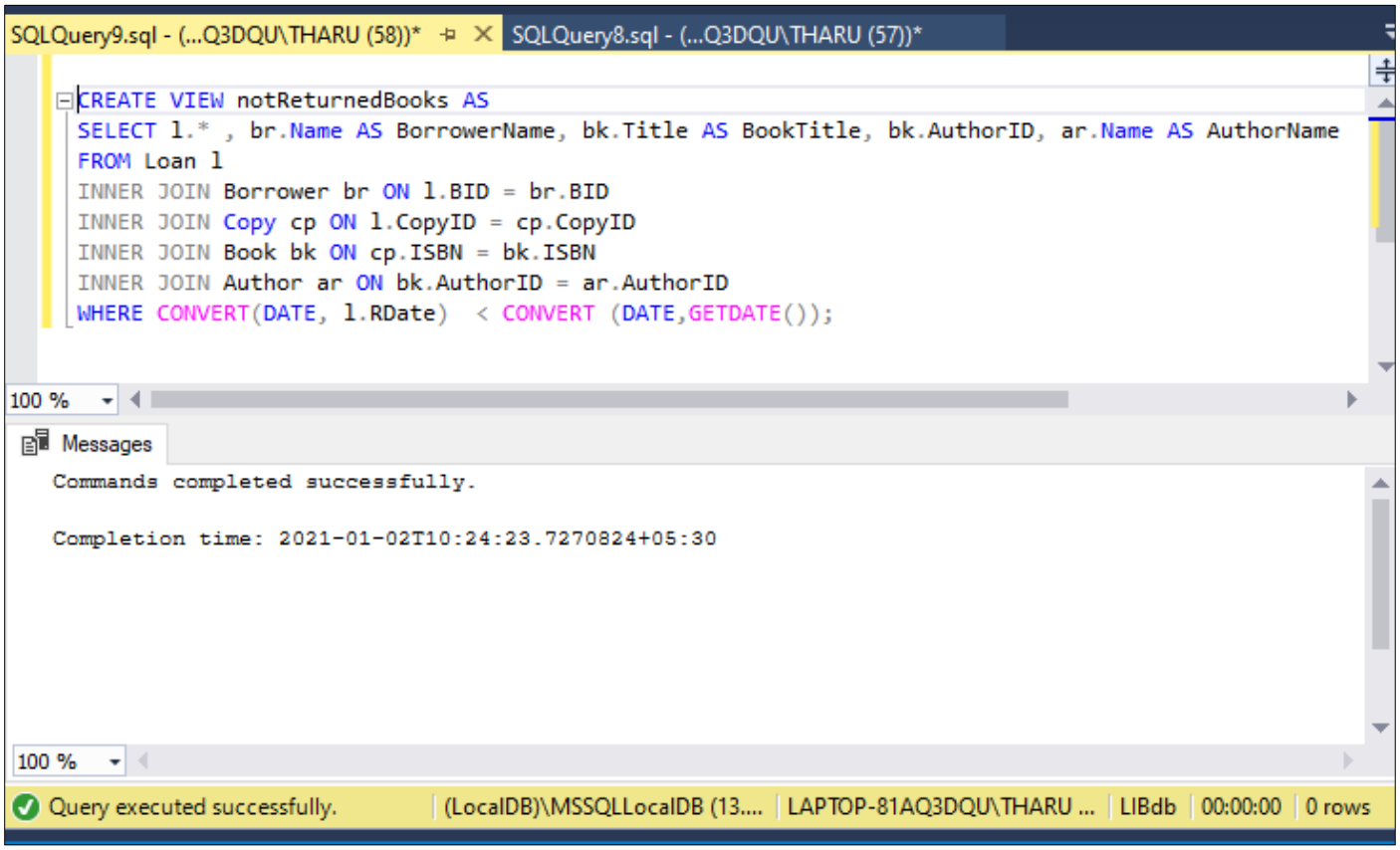
```
USE [LIBdb]
GO
/****** Object:  UserDefinedFunction [dbo].[FunCalculateOverDue]    Script Date: 1/2/2021 9:09:49 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Author,,Name>
-- Create date: <Create Date, ,>
-- Description: <Description, ,>
-- =============================================
ALTER FUNCTION [dbo].[FunCalculateOverDue]
(
@rDate DATE,
@paymentDue INT
)
RETURNS INT
AS
BEGIN

    DECLARE
    @overDueAmout INT,
    @days INT
    SET @days = DATEDIFF(DAY,CONVERT(DATE,@Rdate),CONVERT(DATE,GETDATE()));

    Select @overDueAmout = @days * @paymentDue;

    RETURN @overDueAmout;

END
```

100 %

Connected. (1/1) | (LocalDB)\MSSQLLocalDB (13.... | LAPTOP-81AQ3DQU\THARU ...

➢ Microsoft SQL Server View statements

View statement for not returned books



➢ Microsoft SQL Server Stored Procedures

Data Insert Stored Procedures

```
USE [LIBdb]
GO
/****** Object:  StoredProcedure [dbo].[spInsertAuthors]    Script Date:
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
ALTER PROCEDURE [dbo].[spInsertAuthors]
@authorId NCHAR,
@name NCHAR,
@contact NCHAR,
@email NCHAR

AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO dbo.Author([AuthorID],[Name],[Contact],[Email])
    VALUES (@authorId,@name,@contact,@email)
END
```

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================
-- Author:  <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE PROCEDURE [dbo].[spInsertBooks]
@ISBN BIGINT,
@title NCHAR,
@author NCHAR,
@copies NCHAR,
@subject NCHAR
AS
BEGIN

    INSERT INTO dbo.Book([ISBN],[Title], [Author], [No.of.copies], [Subject])
    VALUES (@ISBN, @title, @author, @copies,@subject)

END
GO
```

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================
-- Author:  <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE PROCEDURE [dbo].[spInsertBorrowers]
@BID NCHAR,
@name NCHAR,
@contact NCHAR,
@address NCHAR,
@email NCHAR

AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO dbo.Borrower([BID],[Name],[Contact],[Address],[Email])
    VALUES (@BID,@name,@contact,@address,@email);
END
GO
```

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================
-- Author:  <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE PROCEDURE [dbo].[spInsertCopies]
@copyId NCHAR,
@Availability NCHAR,
@pPrice NCHAR,
@sPrice NCHAR,
@isbn NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO dbo.Copy([CopyID],[Availability],[Pprice],[Sprice],[ISBN])
    VALUES(@copyId,@Availability,@pPrice,@sPrice,@isbn);
END
GO
```

```
USE [LIBdb]
GO
/****** Object:  StoredProcedure [dbo].[spInsertLibrary]    Script Date: 1/3/2021 8:27:
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
ALTER PROCEDURE [dbo].[spInsertLibrary]
@libraryId NCHAR,
@name NCHAR,
@type NCHAR,
@administrator NCHAR,
@contact NCHAR,
@address NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO dbo.Library([LibID],[Name],[Type],[Administrator],[Contact],[Address])
    VALUES(@libraryId,@name,@type,@administrator,@contact,@address)
END
```

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================
-- Author:  <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE PROCEDURE [dbo].[spInsertLoans]
@loanId NCHAR,
@copyId NCHAR,
@BID NCHAR,
@LDate DATE,
@RDate DATE
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO dbo.Loan([LoanID],[CopyID],[BID],[LDate],[RDate])
    VALUES(@loanId,@copyId,@BID,@LDate,@RDate)
END
GO
```

Data Update Stored Procedures

```
SET QUOTED_IDENTIFIER ON
GO

-- ================================================...
CREATE PROCEDURE [dbo].[spUpdateAuthors]
@authorId NCHAR,
@name NCHAR,
@contact NCHAR,
@email NCHAR

AS
BEGIN
    SET NOCOUNT ON;
    UPDATE dbo.Author
    SET
    [AuthorID] = @authorId,
    [Name] = @name,
    [Contact] = @contact,
    [Email] = @email
    WHERE [AuthorID] = @authorId
```

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- ================================================...
CREATE PROCEDURE [dbo].[spUpdateBooks]
@ISBN BIGINT,
@title NCHAR,
@authorId NCHAR,
@copies NCHAR,
@subject NCHAR

AS
BEGIN
    SET NOCOUNT ON;
    UPDATE dbo.Book
    SET
    [Title] = @title,
    [AuthorID] = @authorId,
    [No.of.copies] = @copies,
    [Subject] = @subject
    WHERE [ISBN] = @ISBN;
END
GO
```

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- ================================================...
CREATE PROCEDURE [dbo].[spUpdateBorrowers]
@BID NCHAR,
@name NCHAR,
@contact NCHAR,
@address NCHAR,
@email NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    Update [dbo].[Borrower]
    SET
        [BID]=@BID,
        [name]=@name,
        [contact]=@contact,
        [address]=@address,
        [email]=@email
    WHERE [BID]=@BID
END
GO
```

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO


-- ================================================...
CREATE PROCEDURE [dbo].[spUpdateCopies]
@copyId NCHAR,
@Availability NCHAR,
@pPrice NCHAR,
@sPrice NCHAR,
@isbn NCHAR

AS
BEGIN
    SET NOCOUNT ON;
    UPDATE [dbo].[Copy]
    SET
    [copyId] = @copyId,
    [Availability] = @Availability,
    [pPrice] = @pPrice,
    [sPrice] = @sPrice
    WHERE [copyId] = @copyId;
END
GO
```

```
SET QUOTED_IDENTIFIER ON
GO

-- ================================================...
CREATE PROCEDURE [dbo].[spUpdateLibrary]
@libraryId NCHAR,
@name NCHAR,
@type NCHAR,
@administrator NCHAR,
@contact NCHAR,
@address NCHAR

AS
BEGIN
    SET NOCOUNT ON;
    UPDATE [dbo].[Library]
    SET
    [LibId] = @libraryId,
    [name] = @name,
    [type] = @type,
    [administrator] = @administrator
    WHERE [LibId] = @libraryId;
```

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- ================================================...
CREATE PROCEDURE [dbo].[spUpdateLoans]
@loanId NCHAR,
@copyId NCHAR,
@BID NCHAR,
@LDate DATE,
@RDate DATE
AS
BEGIN
    SET NOCOUNT ON;
    Update dbo.Loan
    SET
        [CopyID]=@copyId,
        [BID]=@BID,
        [LDate]=@LDate,
        [RDate]=@RDate
    WHERE [LoanID]=@loanId
END
GO
```

Data Delete Stored Procedures

```
GO

/****** Object:  StoredProcedure [dbo].[spDeleteAuthors]
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================...
CREATE PROCEDURE [dbo].[spDeleteAuthors]
@authorID NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    DELETE FROM dbo.Author
    WHERE authorID = @authorId;
END
GO
```

```
GO

/****** Object:  StoredProcedure [dbo].[spDeleteBooks]
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================...
CREATE PROCEDURE [dbo].[spDeleteBooks]
@ISBN NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    DELETE FROM dbo.Book
    WHERE [ISBN] = @ISBN;
END
GO
```

```
GO

/****** Object:  StoredProcedure [dbo].[spDeleteBorrowe
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================...
CREATE PROCEDURE [dbo].[spDeleteBorrowers]
@borrowerId NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    DELETE FROM dbo.Borrower
    WHERE BID = @borrowerId;
END
GO
```

```
GO

/****** Object:  StoredProcedure [dbo].[spDeleteCopies]
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================...
CREATE PROCEDURE [dbo].[spDeleteCopies]
@copyId NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    DELETE FROM dbo.Copy
    WHERE CopyID = @copyId;
END
GO
```

```
GO

/****** Object:  StoredProcedure [dbo].[spDeleteLibrary]
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================...
CREATE PROCEDURE [dbo].[spDeleteLibrary]
@libraryId NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    DELETE FROM dbo.Library
    WHERE LibID = @libraryId;
END
GO
```

```
GO

/****** Object:  StoredProcedure [dbo].[spDeleteLoans]
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================...
CREATE PROCEDURE [dbo].[spDeleteLoans]
@loanId NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    DELETE FROM dbo.Loan
    WHERE LoanId = @loanId;
END
GO
```

Data Search Stored Procedures

```sql
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE PROCEDURE spSearchAuthors
@authorId NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    SELECT * FROM [dbo].[Author]
    WHERE AuthorID = @authorId;
END
GO
```

```sql
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE PROCEDURE spSearchBooks
@isbn BIGINT
AS
BEGIN
    SET NOCOUNT ON;
    SELECT * FROM [dbo].[Book]
    WHERE ISBN = @isbn
END
GO
```

```sql
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE PROCEDURE spSearchBorrowers
@bid NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    SELECT * FROM [dbo].[Borrower]
    WHERE BID = @bid;
END
GO
```

```sql
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE PROCEDURE spSearchCopies
@copyId NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    SELECT * FROM [dbo].[Copy]
    WHERE CopyID = @copyId;
END
GO
```

```sql
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE PROCEDURE spSearchLibraryData
@libId NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    SELECT * FROM [dbo].[Library]
    WHERE LibID = @libId
END
GO
```

```sql
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE PROCEDURE spSearchLoans
@loanId NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    SELECT * FROM [dbo].[Loan]
    WHERE LoanID = @loanId;
END
GO
```

Check Data Validity Stored Procedure

```
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE PROCEDURE [dbo].[spCheckUserValidity]
@userName NCHAR,
@password NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    SELECT COUNT(UserName) FROM dbo.Users
    WHERE UserName = @userName and Password = @password;
END
```

Process Overdue Stored Procedure

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================...
CREATE PROCEDURE [dbo].[spProcessOverDue]
@loanId NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE
    @isOverDue BIT,
    @borrowerId NCHAR,
    @days INT,
    @paymentDue INT,
    @isRecordExist INT

    SET @isOverDue = (SELECT dbo.FuncCheckOverDue(@loanId));
    SET @borrowerId = (SELECT BID from dbo.notReturnedBooks WHERE LoanID = @loanId);
    SET @days = DATEDIFF(DAY,(SELECT RDate from dbo.notReturnedBooks WHERE LoanID = @loanId),GETDATE());
    SET @paymentDue = (SELECT  dbo.FunCalculateOverDue((SELECT RDate from dbo.notReturnedBooks WHERE LoanID = @loanId),15));
    SET @isRecordExist = (SELECT COUNT(LoanID) from dbo.OverDue WHERE LoanID = @loanId);

    IF @isOverDue = 1 AND @isRecordExist = 0
    INSERT INTO [dbo].[Overdue]
    VALUES (@loanId,@borrowerId,@days,@paymentDue);

END
GO
```

Not Returned Books Stored Procedure

```
GO

/****** Object:  StoredProcedure [dbo].[spNotReturnedBookLoanID]
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================...
CREATE PROCEDURE [dbo].[spNotReturnedBookLoanID]
@loanId NCHAR
AS
BEGIN
    SET NOCOUNT ON;
    SELECT * FROM dbo.notReturnedBooks nrb
    WHERE nrb.LoanID =@loanId;
    EXEC dbo.spProcessOverDue @loanId
END
GO
```

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.