

# REPORT

# PRINCIPLE OF DATABASE MANAGEMENT

LECTURER: NGUYEN THI THANH SANG

# LIBRARY

MEMBERS

NGUYEN VIET THANG HO HAI NGUYEN TRAN DUY BAO  
BUI NGOC THANH HIEN PHAM MINH QUANG

# TABLE OF CONTENTS

I.	Introduction .....	2
II.	ER Diagram .....	3
III.	Relational Database Schema .....	4
IV.	Database in SQL.....	5
V.	Program .....	9
VI.	Conclusions .....	22

## MEMBER'S CONTRIBUTION

MEMBERS	ID	JOBS
Nguyễn Viết Thắng	ITITIU15062	ERD and Schema
Trần Duy Bảo	ITITIU15076	Program
Phạm Minh Quang	ITITIU15066	Report
Bùi Ngọc Thanh Hiền	BTFTIU15001	ERD and Database
Hồ Hải Nguyên	BTBCIU15007	Database and Report

# INTRODUCTION

SQL is a standard language for accessing and manipulating databases. The goals of this project is to further exercise us in database concepts in *realistic, holistic contexts*.

Our group's project is to design a program with the database of a **library**. Each book and each student have their own ID to easily identify. The task of the staff is managing relationship between books and students throughout the program written in Java.

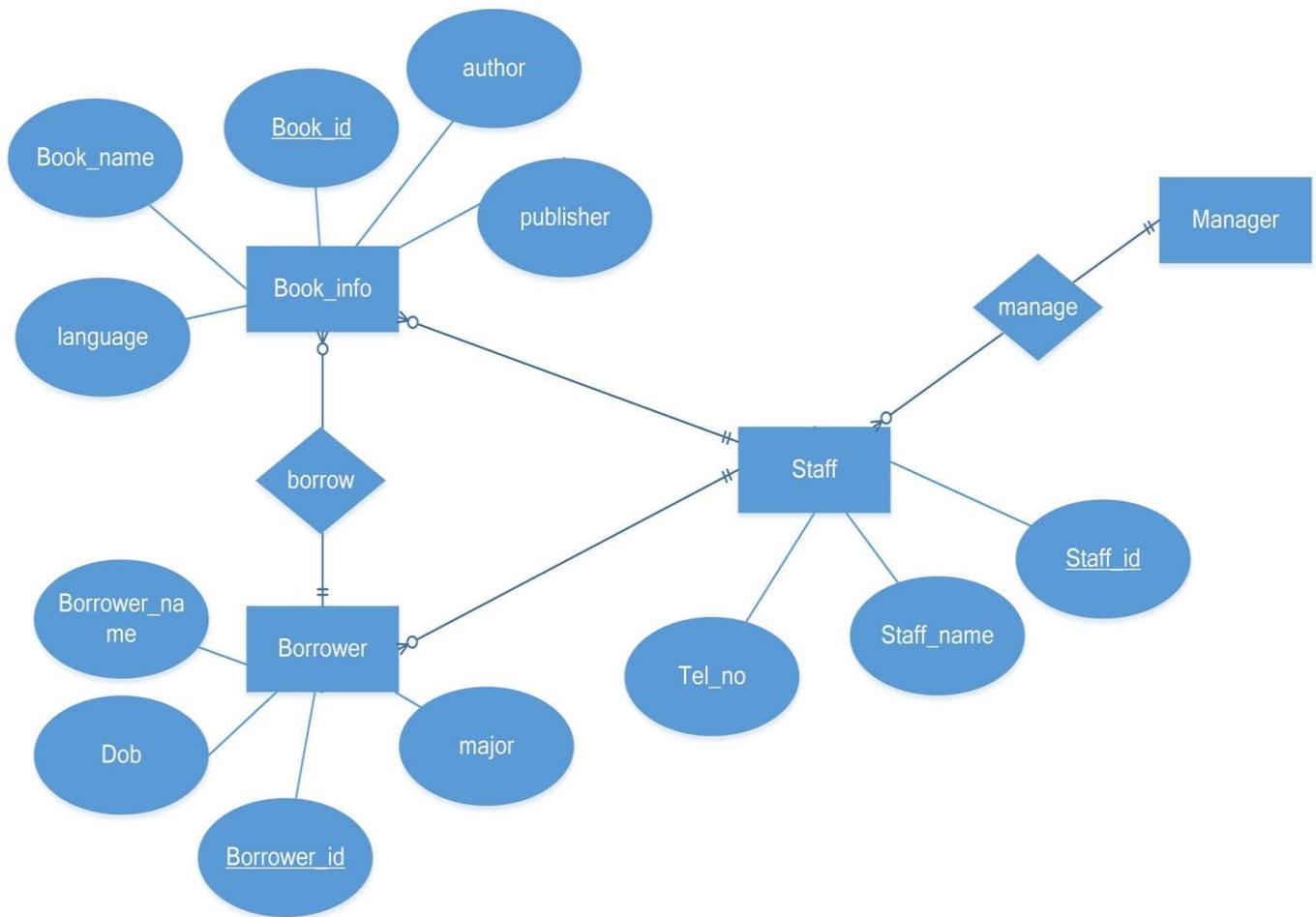
# ER DIAGRAM (ERD)

**Primary-key:** Borrower\_ID, Book\_ID, Staff\_ID

**One-to-many:** Staff -> Book\_Info

Staff -> Borrower

Manager -> Staff



# RELATIONAL DATABASE SCHEMA

**Book\_Info-Schema** = (book\_id, book\_name, author, publisher, language)

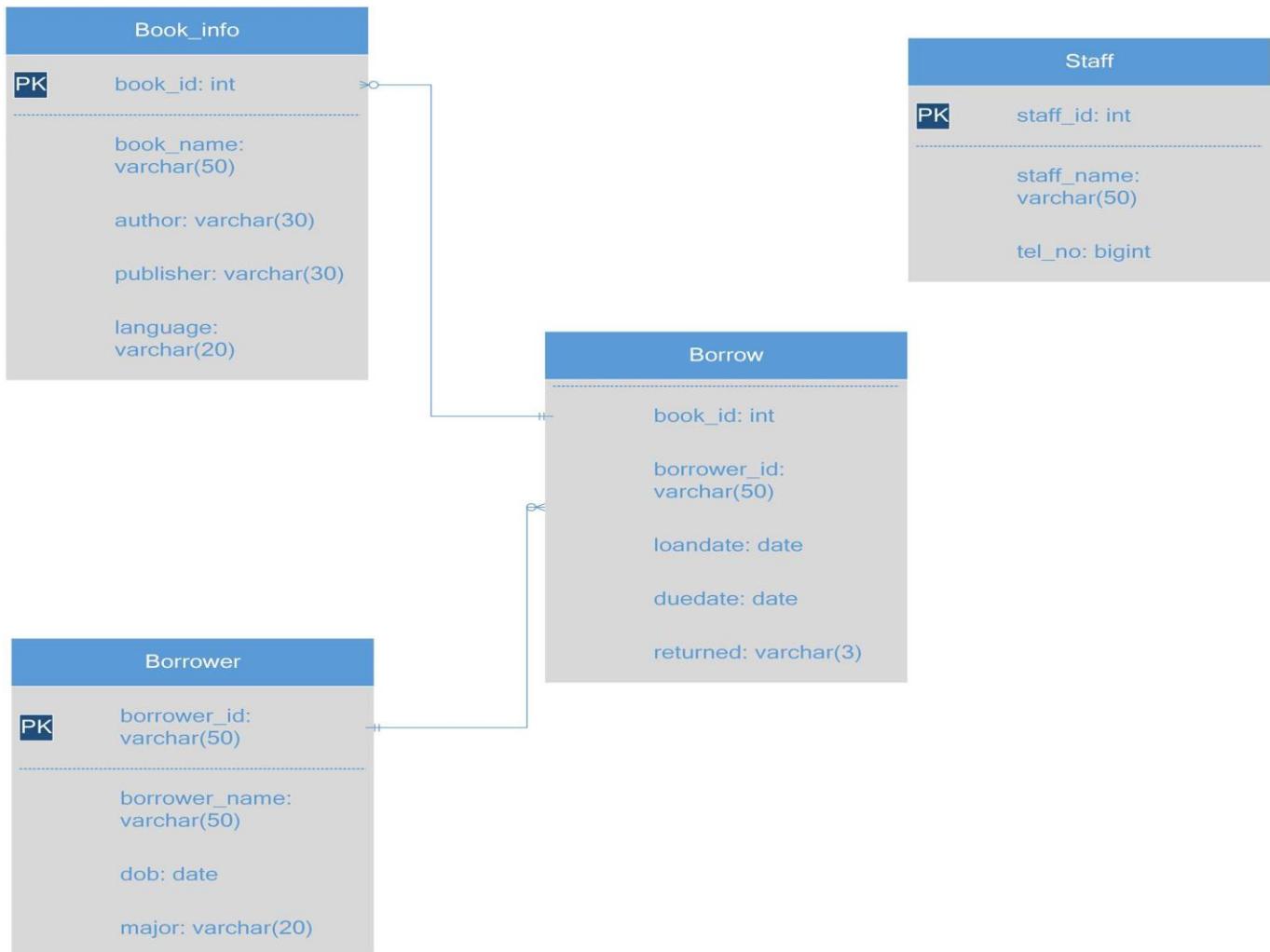
**Staff-Schema** = (staff\_id, staff\_name, tel\_no)

**Borrow-Schema** = (book\_id, borrower\_id, loandate, duedate)

- From Borrow\_Schema.book\_id to Book\_Info-Schema.book\_id

**Borrower-Schema** = (borrower\_id, borrower\_name, dob, major)

- From Borrower-Schema.borrower\_id to Borrow-Schema.borrower\_id



# DATABASE IN SQL

```

CREATE DATABASE Library;
CREATE TABLE Borrower (
    Borrower_ID varchar(50) NOT NULL,
    Borrower_Name varchar(50) NOT NULL,
    Major varchar(50) NOT NULL,
    DoB date NULL,
    Primary key(Borrower_ID)
);

```

```

    INSERT INTO Borrower(Borrower_ID, Borrower_Name, Major, DoB)
VALUES ( 'BABAIU15108', 'Jason Mars', 'Business', '1997-05-18');

    INSERT INTO Borrower(Borrower_ID, Borrower_Name, Major, DoB)
VALUES ( 'BABAWE16109', 'Nick Jason', 'Business', '1998-05-15');

    INSERT INTO Borrower(Borrower_ID, Borrower_Name, Major, DoB)
VALUES ( 'BAFNIU13023', 'Dannie Kai', 'Business', '1995-10-10');

    INSERT INTO Borrower(Borrower_ID, Borrower_Name, Major, DoB)
VALUES ( 'BTBCIU13089', 'Sarah Mannie', 'Biology Tech', '1995-09-08');

    INSERT INTO Borrower(Borrower_ID, Borrower_Name, Major, DoB)
VALUES ( 'BTBTIU16078', 'Nina Walson', 'Biology Tech', '1998-08-28');

    INSERT INTO Borrower(Borrower_ID, Borrower_Name, Major, DoB)
VALUES ( 'BTFTIU15009', 'Charlize Theron', 'Biology Tech', '1997-03-08');

    INSERT INTO Borrower(Borrower_ID, Borrower_Name, Major, DoB)
VALUES ( 'BTFTIU16005', 'Han Sika', 'Biology Tech', '1998-02-01');

    INSERT INTO Borrower(Borrower_ID, Borrower_Name, Major, DoB)
VALUES ( 'EEEEIU14019', 'Mark Landon', 'Electrical Eng', '1996-12-05');

    INSERT INTO Borrower(Borrower_ID, Borrower_Name, Major, DoB)
VALUES ( 'EEEEIU14034', 'Sam Seith', 'Electrical Eng', '1996-11-24');

    INSERT INTO Borrower(Borrower_ID, Borrower_Name, Major, DoB)
VALUES ( 'ITITIU12062', 'Anna Dell', 'Information Tech', '1994-01-30');

```

Column Name	Data Type	Allow Nulls
borrower_id	varchar(50)	<input type="checkbox"/>
borrower_name	varchar(50)	<input type="checkbox"/>
dob	date	<input checked="" type="checkbox"/>
major	varchar(20)	<input type="checkbox"/>

borrower_id	borrower_name	dob	major
BABAIU15108	Jason Mars	1997-05-18	Business
BABAWE16109	Nick Jason	1998-05-15	Business
BAFNIU13023	Dannie Kai	1995-10-10	Business
BTBCIU13089	Sarah Mannie	1995-09-08	Biology Tech
BTBTIU16078	Nina Walson	1998-08-28	Biology Tech
BTFTIU15009	Charlize Theron	1997-03-08	Biology Tech
BTFTIU16005	Han Sika	1998-02-01	Biology Tech
EEEEIU14019	Mark Landon	1996-12-05	Electrical Eng
EEEEIU14034	Sam Seith	1996-11-24	Electrical Eng
ITITIU12062	Anna Dell	1994-01-30	Information Tech
*	NULL	NULL	NULL

# DATABASE IN SQL

```
CREATE TABLE Book_info(
Book_ID int NOT NULL,
Book_Name nvarchar(50) NOT NULL,
Author nvarchar(50) NOT NULL,
Language nvarchar(50) NOT NULL,
Publisher nvarchar(50) NOT NULL,
Primary key(Book_id)
);
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists the database structure, including 'MAYTINH-8FIMONL.SQL Server 13.' and its 'Tables' folder containing 'dbo.book\_info'. The 'dbo.book\_info' table is selected in the center pane, showing its schema:

Column Name	Data Type	Allow Nulls
book_id	int	<input type="checkbox"/>
book_name	varchar(50)	<input type="checkbox"/>
author	varchar(30)	<input checked="" type="checkbox"/>
publisher	varchar(30)	<input checked="" type="checkbox"/>
language	varchar(20)	<input checked="" type="checkbox"/>

```
INSERT INTO Books.dbo.Book_info(Book_ID,Book_Name,Author,Language,Publisher)
VALUES (1,'Artificial Intelligence','Muhamad Ali','Indian','Kim Đong');
INSERT INTO Books.dbo.Book_info(Book_id,Book_Name,Author,Language,Publisher)
VALUES (2, 'Intro to Sql','Nguyen Viet Thang','Egyptian','National University');
INSERT INTO Books.dbo.Book_info(Book_id,Book_Name,Author,Language,Publisher)
VALUES (3, 'Seven Habits','Tony Stark','English','Education center');
INSERT INTO Books.dbo.Book_info(Book_id,Book_Name,Author,Language,Publisher)
VALUES (4, 'Professional Desinger','Alex Thanh Hien','Chinese','Shounen Jump');
INSERT INTO Books.dbo.Book_info(Book_id,Book_Name,Author,Language,Publisher)
VALUES (5, 'Gold is stable', 'Coloreone','Spanish','Young generation');
INSERT INTO Books.dbo.Book_info(Book_id,Book_Name,Author,Language,Publisher)
VALUES (6, 'Introduction to C', 'Bui Ngoc Thanh Hien','English','AI magazine');
```

The screenshot shows the Microsoft SQL Server Management Studio interface after executing the INSERT statements. The 'Object Explorer' on the left shows the database structure. The 'MAYTINH-8FIMONL...y - dbo.book\_info' table in the center pane displays the following data:

book_id	book_name	author	publisher	language
1	Artificial Intelligence	Muhamad Ali	Kim Đong	Indian
2	Intro to Sql	Nguyen Viet Thang	National University	Egyptian
3	Seven Habits	Tony Stark	Education center	English
4	Professional Desinger	Alex Thanh Hien	Shounen Jump	Chinese
5	Gold is stable	Coloreone	Young generation	Spanish
6	Introduction to C	Bui Ngoc Thanh Hien	AI magazine	English
*	NULL	NULL	NULL	NULL

# DATABASE IN SQL

```
CREATE TABLE staff(
staff_id int,
staff_name varchar(50),
tel_no bigint
);
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'MAYTINH-8FIMONL' is selected, and under its 'Tables' folder, a new table 'staff' is being defined. The table structure is as follows:

Column Name	Data Type	Allow Nulls
staff_id	int	<input type="checkbox"/>
staff_name	varchar(50)	<input type="checkbox"/>
tel_no	bigint	<input checked="" type="checkbox"/>

```
INSERT INTO Staff(Staff_ID, Staff_Name, tel_no)
VALUES ( 1001, 'Harry Kit', '0938333253');

INSERT INTO Staff(Staff_ID, Staff_Name, tel_no)
VALUES ( 1002, 'Robbert Sookie', '0938462351');

INSERT INTO Staff(Staff_ID, Staff_Name, tel_no)
VALUES ( 1003, 'Anne Fanny', '0906556130');

INSERT INTO Staff(Staff_ID, Staff_Name, tel_no)
VALUES ( 1004, 'Marry Nancy', '0903591119');

INSERT INTO Staff(Staff_ID, Staff_Name, tel_no)
VALUES ( 1005, 'Richard Polly', '0902941001');

INSERT INTO Staff(Staff_ID, Staff_Name, tel_no)
VALUES ( 1006, 'Phillip Hitch', '0938102220');

INSERT INTO Staff(Staff_ID, Staff_Name, tel_no)
VALUES ( 1007, 'James Nell', '0903339110');
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'Tables' folder of the 'MAYTINH-8FIMONL' database is selected. A grid on the right displays the data from the 'staff' table:

staff_id	staff_name	tel_no
1001	Harry Kit	938333253
1002	Robbert Sookie	938462351
1003	Anne Fanny	906556130
1004	Marry Nancy	903591119
1005	Richard Polly	902941001
1006	Phillip Hitch	938102220
1007	James Nell	903339110
NULL	NULL	NULL

# DATABASE IN SQL

```
CREATE TABLE borrow(
book_id int,
borrower_id varchar(50),
loandate date NOT NULL,
duedate date NOT NULL,
returned varchar(3)
);
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "MAYTINH-8FIMONL.library - dbo.borrow - Microsoft SQL Server Management Studio". The Object Explorer sidebar shows the database structure, including the 'library' database and its tables. The main pane displays the table definition for 'borrow' with columns: book\_id (int), borrower\_id (varchar(50)), loandate (date), duedate (date), and returned (varchar(3)).

```
INSERT INTO borrow(book_id, borrower_id, loandate, duedate, returned)
VALUES (1, 'BABAIU15108', '2016-05-18', '2016-05-30', 'Yes');

INSERT INTO borrow(book_id, borrower_id, loandate, duedate, returned)
VALUES (3, 'BABAWE16109', '2017-06-12', '2016-06-25', 'No');

INSERT INTO borrow(book_id, borrower_id, loandate, duedate, returned)
VALUES (4, 'BAFNIU13023', '2016-08-20', '2016-08-30', 'Yes');
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "MAYTINH-8FIMONL.library - dbo.borrow - Microsoft SQL Server Management Studio". The Object Explorer sidebar shows the database structure. The main pane displays the data inserted into the 'borrow' table:

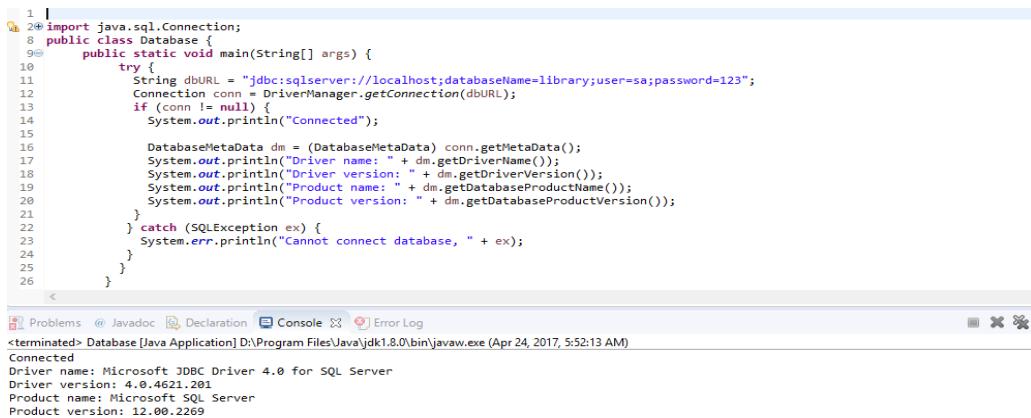
book_id	borrower_id	loandate	duedate	returned
1	BABAIU15108	2016-05-18	2016-05-30	Yes
3	BABAWE16109	2017-06-12	2016-06-25	No
4	BAFNIU13023	2016-08-20	2016-08-30	Yes
*	NULL	NULL	NULL	NULL

# PROGRAM

After designing the ERD and Schema , we are going to create the database and JDBC GUI :

In our project , the database name is “library” , the user is sa and password is 123 so the connection url is :

“jdbc:sqlserver://localhost;”+”databaseName=library;user=sa;password=123” , let’s check the connection :



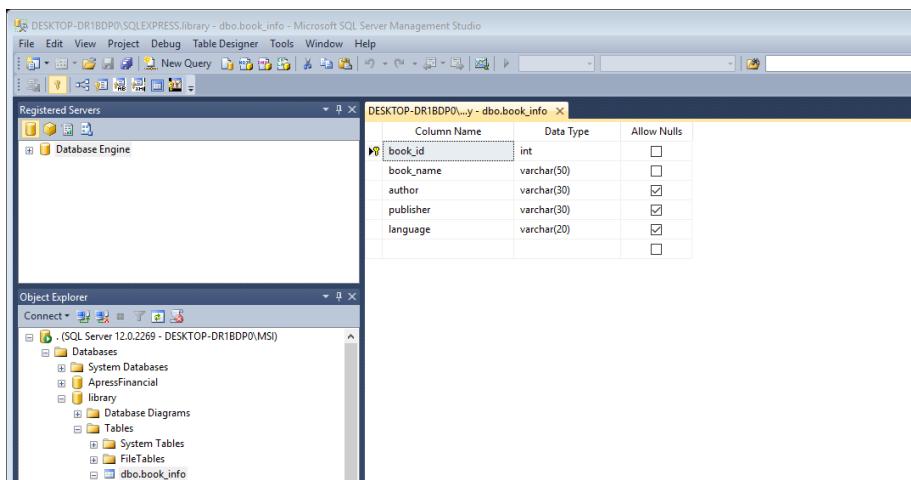
```
1 |
2 28 import java.sql.Connection;
3 public class Database {
4     public static void main(String[] args) {
5         try {
6             String dbURL = "jdbc:sqlserver://localhost;databaseName=library;user=sa;password=123";
7             Connection conn = DriverManager.getConnection(dbURL);
8             if (conn != null) {
9                 System.out.println("Connected");
10            }
11            DatabaseMetaData dm = (DatabaseMetaData) conn.getMetaData();
12            System.out.println("Driver name: " + dm.getDriverName());
13            System.out.println("Driver version: " + dm.getDriverVersion());
14            System.out.println("Product name: " + dm.getDatabaseProductName());
15            System.out.println("Product version: " + dm.getDatabaseProductVersion());
16        } catch (SQLException ex) {
17            System.err.println("Cannot connect database, " + ex);
18        }
19    }
20 }
21 }
```

Connected  
Driver name: Microsoft JDBC Driver 4.0 for SQL Server  
Driver version: 4.0.4621.201  
Product name: Microsoft SQL Server  
Product version: 12.00.2269

So JDBC is connected to sqlserver , we go to create database :

-> By using SQLServer, we have the database as below :

- Book\_info :



# PROGRAM

- Borrow:

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to 'DESKTOP-DR1BDP0\SQL EXPRESS.library - dbo.book\_info'. The 'Table Designer' tab is selected. The 'book\_info' table is displayed with the following columns:

Column Name	Data Type	Allow Nulls
book_id	int	<input type="checkbox"/>
book_name	varchar(50)	<input type="checkbox"/>
author	varchar(30)	<input checked="" type="checkbox"/>
publisher	varchar(30)	<input checked="" type="checkbox"/>
language	varchar(20)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

The 'Object Explorer' pane on the left shows the database structure, including the 'library' database and its tables.

- Borrower:

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to 'DESKTOP-DR1BDP0\SQL EXPRESS.library - dbo.borrower'. The 'Table Designer' tab is selected. The 'borrower' table is displayed with the following columns:

Column Name	Data Type	Allow Nulls
borrower_id	varchar(50)	<input type="checkbox"/>
borrower_name	varchar(50)	<input type="checkbox"/>
dob	date	<input checked="" type="checkbox"/>
major	varchar(20)	<input type="checkbox"/>
		<input type="checkbox"/>

The 'Object Explorer' pane on the left shows the database structure, including the 'library' database and its tables.

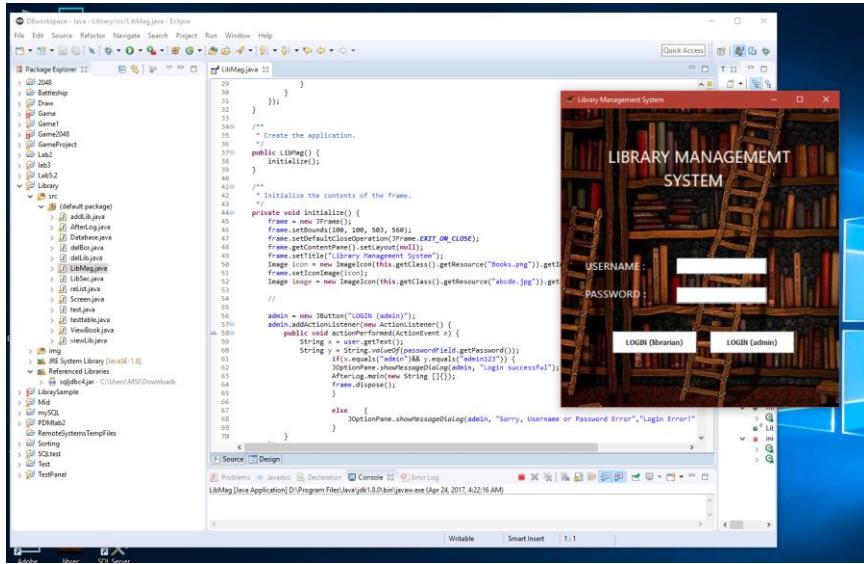
- Staff:

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to 'DESKTOP-DR1BDP0\SQL EXPRESS.library - dbo.staff'. The 'Table Designer' tab is selected. The 'staff' table is displayed with the following columns:

Column Name	Data Type	Allow Nulls
staff_id	int	<input type="checkbox"/>
staff_name	varchar(50)	<input type="checkbox"/>
tel_no	bigint	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

The 'Object Explorer' pane on the left shows the database structure, including the 'library' database and its tables.

# PROGRAM

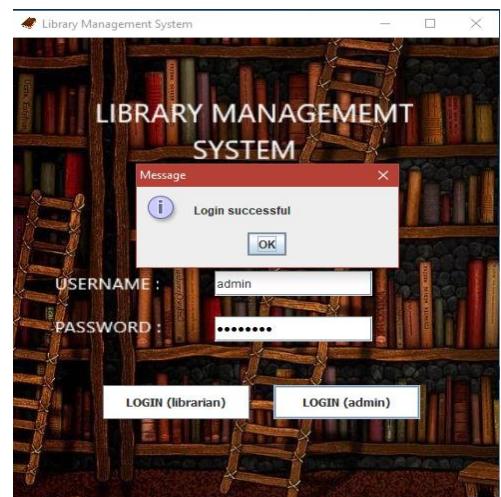


We use this database for our JDBC GUI application on Java Eclipse, let's have the first look :

It contains many classes linked together with different functions to connect the database and use it efficiently , the class LibMag.java below is the login frame .

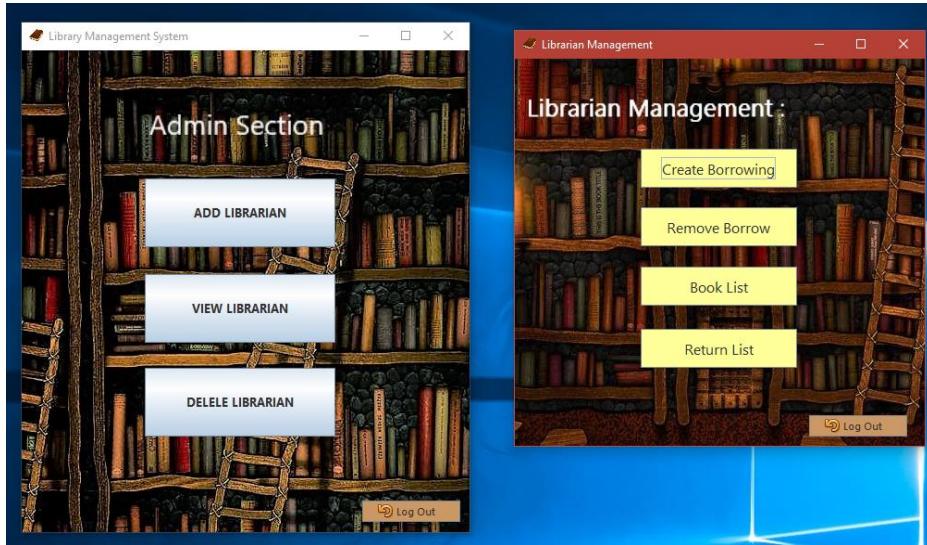
As you can see , there are textpane and passwordfield in order to login and 2 buttons Login(admin) and Login(librarian) , which is designed for 2 kinds of user : the administrator and the librarian. If you use the wrong user name or password , it will pops-up :

So librarian and admin have to remember their user name and password to login, in this case , we set the default : Librarian (username: librarian, pass : librarian 123) and Admin (username:admin, pass:admin123). After login success , it notify :



# PROGRAM

Then we come to the main screen of GUI , librarian and admin has the different main GUI :

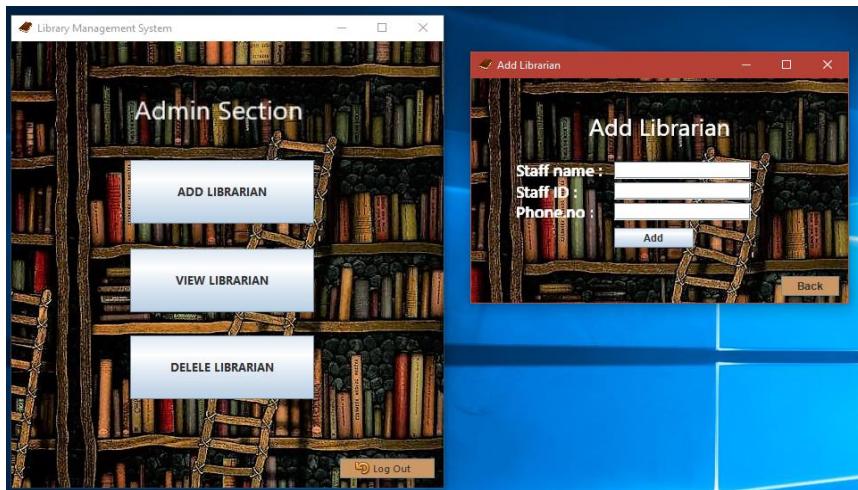


Admin can modify the librarian : add,view and delete librarian. Librarian can modify the borrower , create borrow list , view the list and also add some new book's . All of these connect and work with the database we created on sql server and both have log out button to go back to the login screen. Now we go to detail.

a) First we will see how the Admin Section work :

There are 3 options : Add Librarian , View Librarian and Delete Librarian

- Click the Add Librarian, a new window is appeared :



# PROGRAM

At beginning ,we have the initial database info below:

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer displays a connection to 'DESKTOP-DR1BDP0\SQLEXPRESS.library - dbo.staff'. In the center, a table named 'staff' is shown with three columns: 'staff\_id', 'staff\_name', and 'tel\_no'. There is one row present in the table.

staff_id	staff_name	tel_no
1	Bao	123413456
NULL	NULL	NULL

Assume we add Librarian with name = John, ID = 15 and tel no is 0920202020, we fill information and click add, a message shown :

The screenshot shows a Windows application window titled 'Add Librarian' with a bookshelf background. It contains three input fields: 'Staff name : John', 'Staff ID : 15', and 'Phone no : 0920202020'. Below the fields is an 'Add' button. To the right, a message box displays 'John has been added to database!' with an 'OK' button. In the bottom left corner of the application window, there is a navigation menu with buttons for 'ADD LIBRARIAN', 'VIEW LIBRARIAN', and 'DELETE LIBRARIAN'. In the bottom right corner, there is a 'Log Out' button. The bottom half of the screen shows the Microsoft SQL Server Management Studio interface, which is identical to the one in the first screenshot, displaying the 'staff' table with the new row added.

# PROGRAM

Now we go to check the database and see that John has been added as below:

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, there's a sidebar with icons for 'Management System', 'Admin', 'ADD LIBRARY', 'VIEW LIBRARY', and 'DELETE LIBRARY'. The main area has a title bar 'SQLQuery53.sql - (local).master (DESKTOP-DR1BDP0\MSI (53)) - Microsoft SQL Server Management Studio'. Below the title bar is a toolbar with various icons. The 'Object Explorer' pane on the left shows a tree structure of databases, including 'ApressFinancial' and 'library'. The 'Tables' node under 'library' is expanded, showing tables like 'System Tables', 'FileTables', 'dbo.book\_info', 'dbo.borrow', 'dbo.borrower', and 'dbo.staff'. The 'Columns' node under 'dbo.staff' is also expanded, showing columns 'staff\_id', 'staff\_name', and 'tel\_no'. The 'Results' tab in the bottom right shows a table with two rows of data. A red oval highlights the second row, which corresponds to the newly added staff member 'John'. The table has columns 'staff\_id', 'staff\_name', and 'tel\_no'. The data is as follows:

	staff_id	staff_name	tel_no
1	1	Bao	123413456
2	15	John	920202020

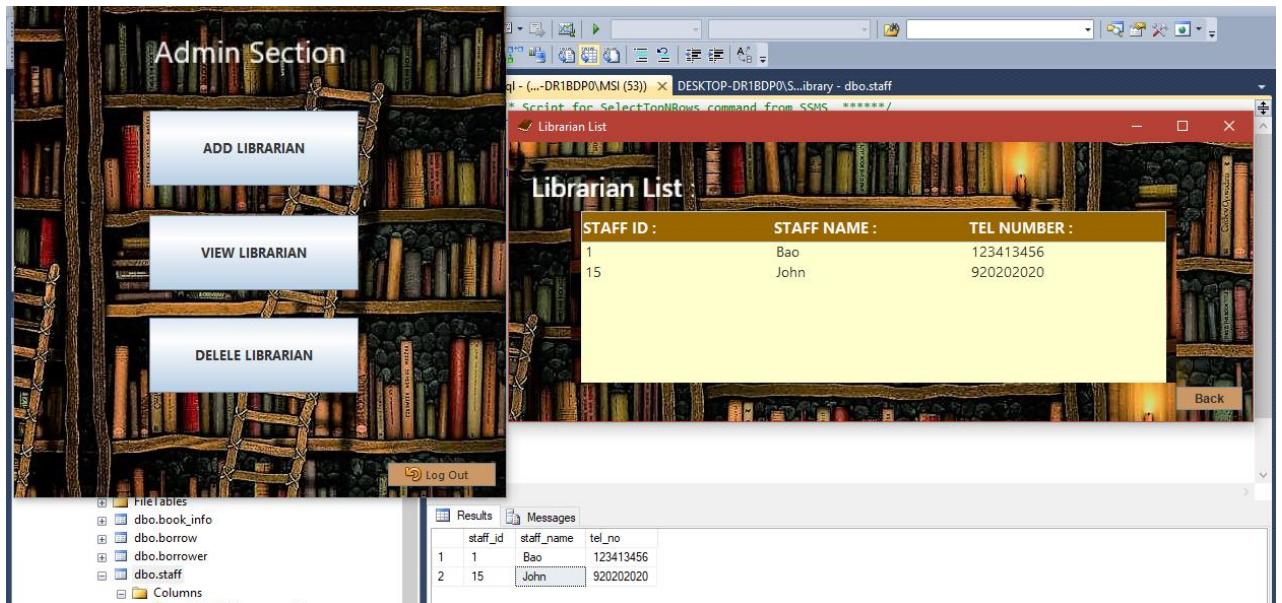
In Eclipse , we use a code below to connect and execute the query :

```
try {
    Connection con = DriverManager.getConnection("jdbc:sqlserver://localhost;"+"databaseName=library;user=sa;password=123");
    Statement st = con.createStatement();
    String sql = "insert into staff "+"(staff_id,staff_name,tel_no) "
    +" values ("+staffid.getText()+","+staffname.getText()+","+phoneno.getText()+"") ";
    st.executeQuery(sql);

}
catch (Exception exc) {
    exc.printStackTrace();
}
 JOptionPane.showMessageDialog(btnAdd, staffname.getText()+" has been added to database!");
staffname.setText(null); staffid.setText(null); phoneno.setText(null);
});
```

# PROGRAM

Next , we will go to see the list of librarian has been added, it select from the database staff in sqlserver ,click View Librarian , the list is shown as below :



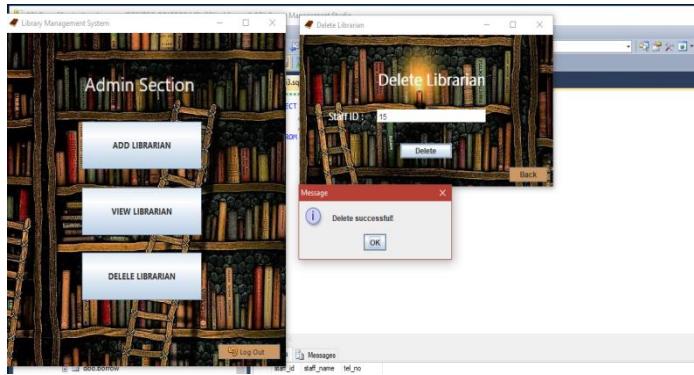
The list uses SELECT query and display on JTextArea by using the code :

```
//function
try {
    Connection con = DriverManager.getConnection("jdbc:sqlserver://localhost;"+"databaseName=library;user=sa;password=123");
    Statement st = con.createStatement();
    String sql = "select * from staff ";
    ResultSet rs = st.executeQuery(sql);

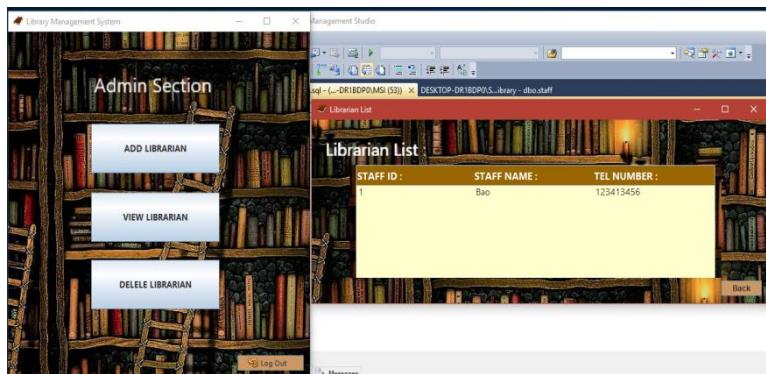
    while(rs.next()) {
        result.append(" "+rs.getString("staff_id")+"\t\t"+rs.getString("staff_name")+"\t\t"+rs.getString("tel_no")+"\n");
    }
}
catch (Exception exc) {
    exc.printStackTrace();
}
```

# PROGRAM

Then we will try to delete some librarian , assume that we want to delete John from the list , we only need to fill John ID = 15 in the Delete Librarian frame and click delete button as below:



We check the Librarian List and Database again to confirm the deletion:



Thus, John has been remove from both the database and the librarian list by using the delete query , here is the code to make it :

```
try {
    Connection con = DriverManager.getConnection("jdbc:sqlserver://localhost;"+"databaseName=library;user=sa;password=123");
    Statement st = con.createStatement();
    String sql = "delete from staff "+"where staff_id='"+staffid.getText();
    st.executeQuery(sql);

}
catch (Exception exc) {
    exc.printStackTrace();
}
 JOptionPane.showMessageDialog(btnDelete, "Delete successful!");
staffid.setText(null);
}
```

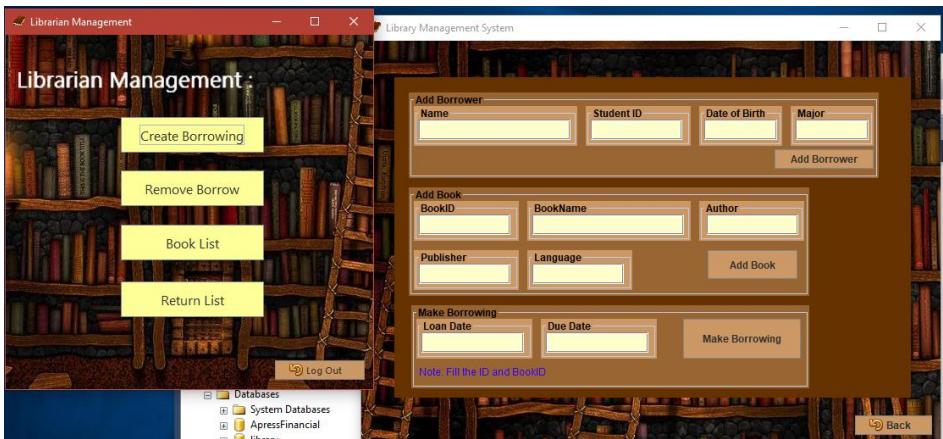
So that's all the options that administrator can do, they can manage/modify the librarian through GUI connected to Database conveniently.

# PROGRAM

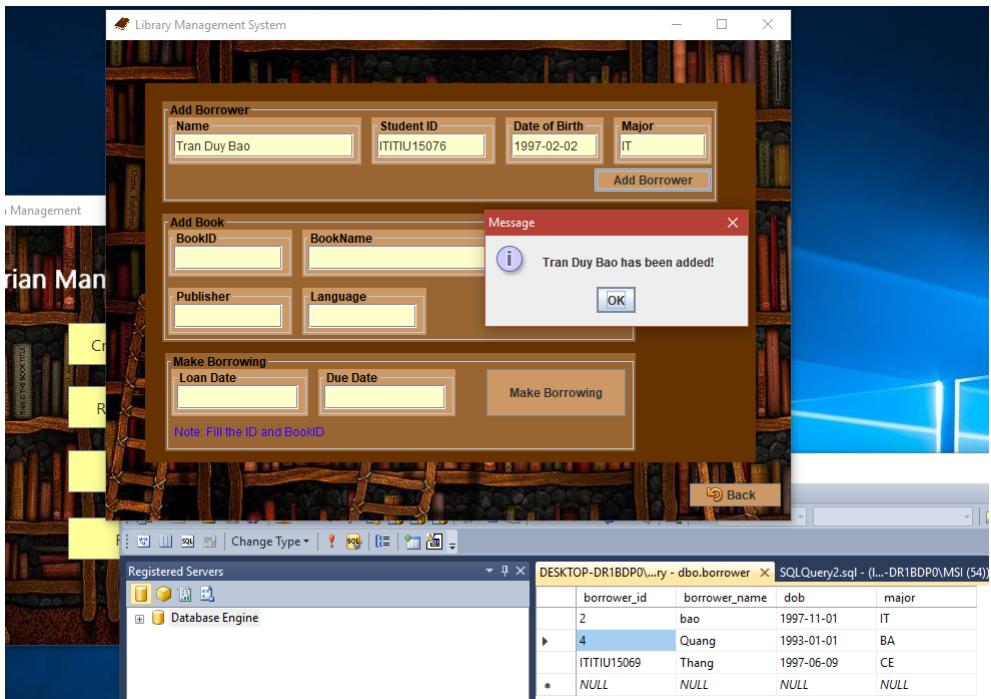
b) Now we go to see what Librarian can do :

There are 4 options : create borrowing ,remove the borrow , view the borrow list and return list

Click Create borrowing , you will come to a screen :



They are 3 forms with 3 different functions, the first one use to add a new person information to database , this can be used for the one who has never used the library service before . Let's take an example :



# PROGRAM

Now we check the database and found that Tran Duy Bao has been added:

The image shows a library management application interface. On the left, there's a sidebar with bookshelves labeled 'Ilan', 'Cr', and 'R'. The main area has two forms: 'Add Book' and 'Make Borrowing'. The 'Add Book' form has fields for BookID, BookName, Author, Publisher, and Language, with an 'Add Book' button. The 'Make Borrowing' form has fields for Loan Date and Due Date, with a 'Make Borrowing' button. A note at the bottom says 'Note: Fill the ID and BookID'. On the right, there's a screenshot of SQL Server Management Studio showing the 'borrower' table in the 'library' database. The table has columns: borrower\_id, borrower\_name, dob, and major. There are four rows of data, with the fourth row (Tran Duy Bao) circled in red.

	borrower_id	borrower_name	dob	major
1	2	bao	1997-11-01	IT
2	4	Quang	1993-01-01	BA
3	ITITIU15000	Trung	1997-08-08	CE
4	ITITIU15076	Tran Duy Bao	1997-02-02	IT

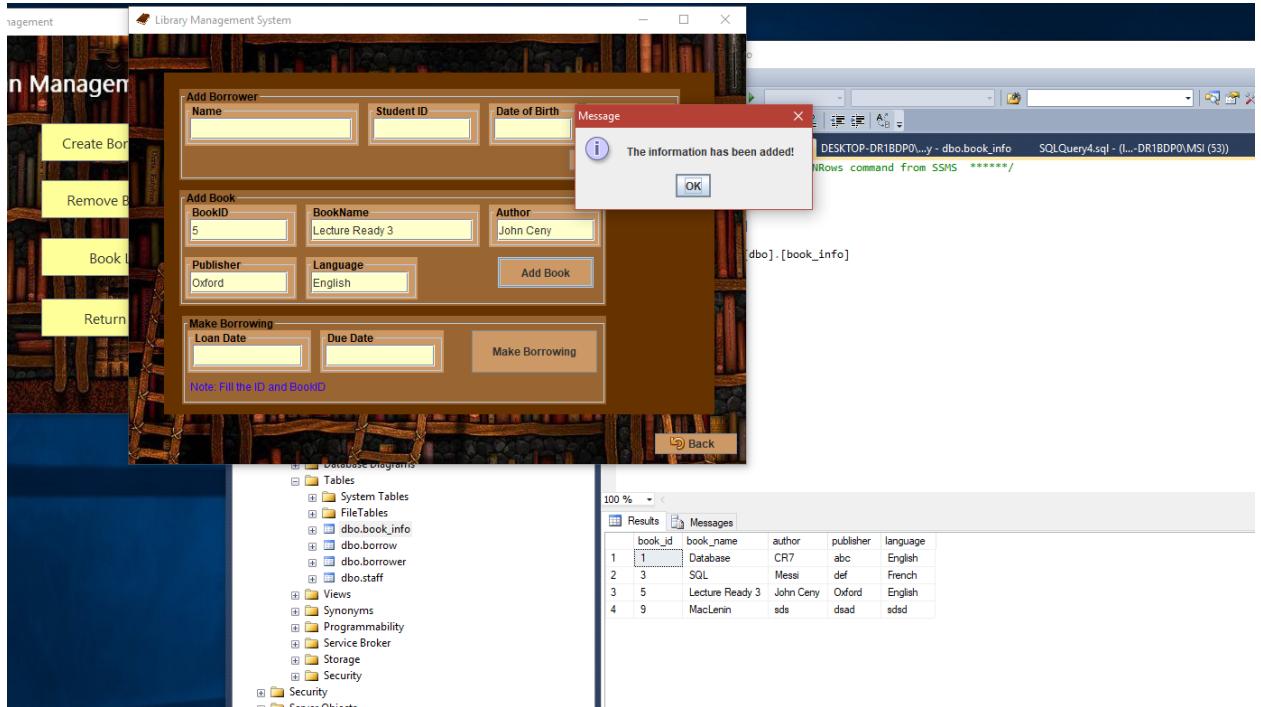
And here is the first form query used in JDBC :

```
public void actionPerformed(ActionEvent e) {
    try {
        Connection con = DriverManager.getConnection("jdbc:sqlserver://localhost;"+"databaseName=library;user=sa;password=123");
        Statement st = con.createStatement();

        String sql = "insert into borrower "+"(borrower_id,borrower_name,dob,major) "+"values ("+"'" +id.getText() + "','" +
                +name.getText() + "','" +dob.getText() + "','" +major.getText() + ")";
        st.executeQuery(sql);
    }
    catch (Exception exc) {
        exc.printStackTrace();
    }
    JOptionPane.showMessageDialog(addbor, name.getText()+" has been added!");
    dob.setText(null);major.setText(null);name.setText(null);
}
});
```

# PROGRAM

The second form use to create a new book information which some textfield to fill in the information same as the first . Let's take a example :



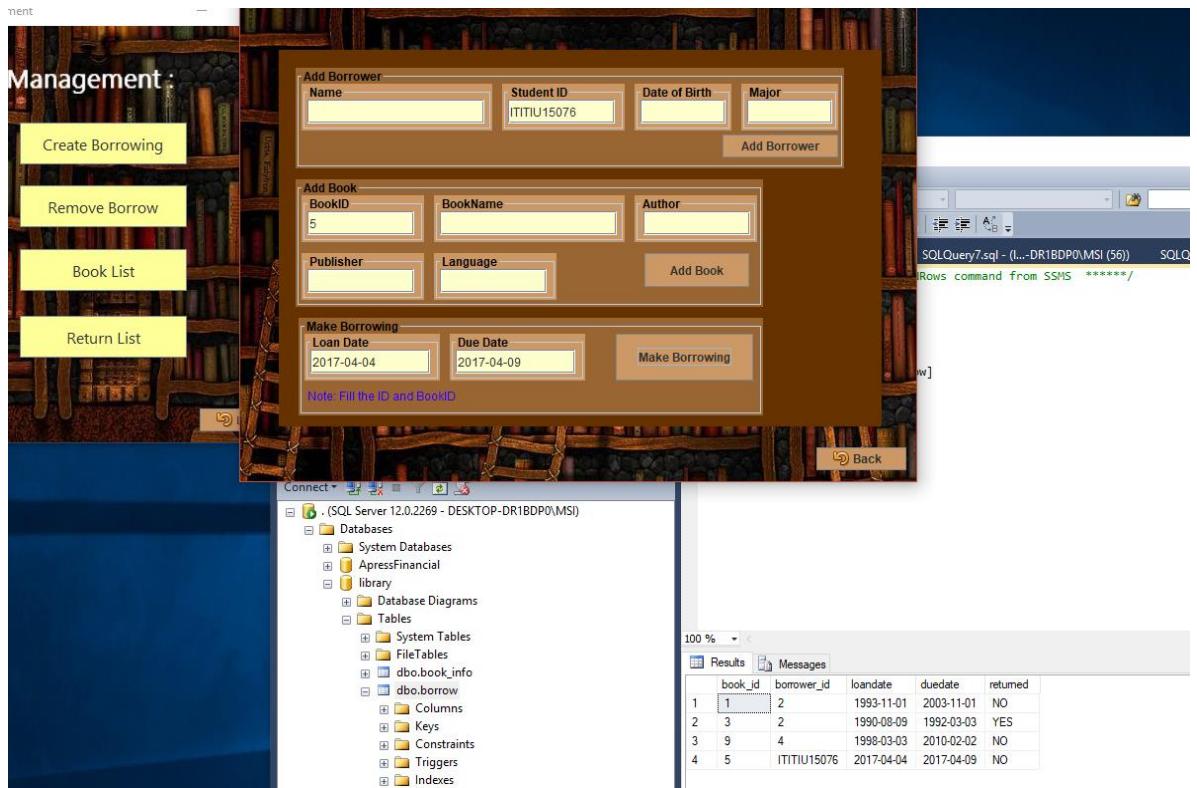
As you see , the book Lecture Ready 3 with some infomation has been add to the database, you can fill only the BookID and BookName because we has set Author, Publisher and Language NULL in the database

The function is JDBC :

```
public void actionPerformed(ActionEvent arg0) {
    try{
        Connection con = DriverManager.getConnection("jdbc:sqlserver://localhost;"+"databaseName=library;user=sa;password=123");
        Statement st2 = con.createStatement();
        String sql2 = "insert into book_info "+"(book_id,book_name,author,publisher,language) "
                    +" values ("+bookid.getText()+","+title.getText()+","+author.getText()+"','"+pub.getText()+"','"+lan.getText()+"') ";
        st2.executeQuery(sql2);
    }
    catch (Exception exc) {
        exc.printStackTrace();
    }
    JOptionPane.showMessageDialog(addbook, "The information has been added!");
    pub.setText(null);title.setText(null);author.setText(null);lan.setText(null);
}
```

# PROGRAM

The first form is use to create a borrowing , you have to fill the Student ID and Book ID to make it successful , and those ID must been added in database before , it means that you to finish 1<sup>st</sup> and 2<sup>nd</sup> form before using this form or use the information added in database , let's make a borrowing with the book Lecture Ready 3 ( Book ID = 5 ) by Tran Duy Bao (ID = ITITIU15076)



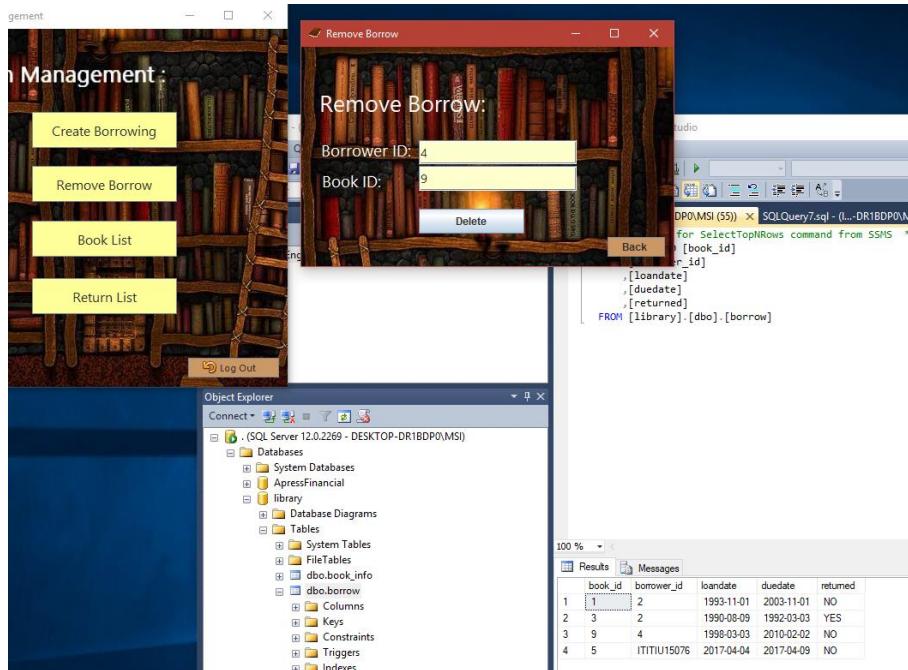
And this borrowing has been added to the database , and the below code has been used to make it :

```
try{
    Connection con = DriverManager.getConnection("jdbc:sqlserver://localhost;"+"databaseName=library;user=sa;password=123");
    Statement st3 = con.createStatement();
    String sql3 = "insert into borrow "+"(book_id,borrower_id,loandate,duedate) "
        +" values ("+bookid.getText()+","+id.getText()+","+loan.getText()+","+due.getText()+"')";
    st3.executeQuery(sql3);
}
catch (Exception exc) {
    exc.printStackTrace();
} JOptionPane.showMessageDialog(addbook, "The information has been added!");
}
```

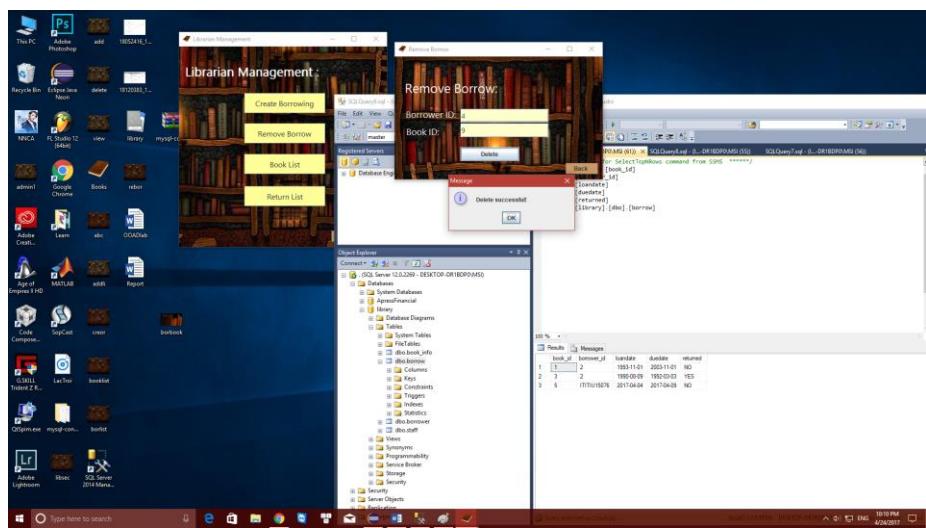
# PROGRAM

Now we will come to some more options that librarian can do , back to the main screen , click remove Borrow , a new window is appeared :

If someone has returned the book to the library , the librarian should remove the borrowing from database , in order to use it , let's take an example with ID = 4 and Book ID = 9.



Here is the result :



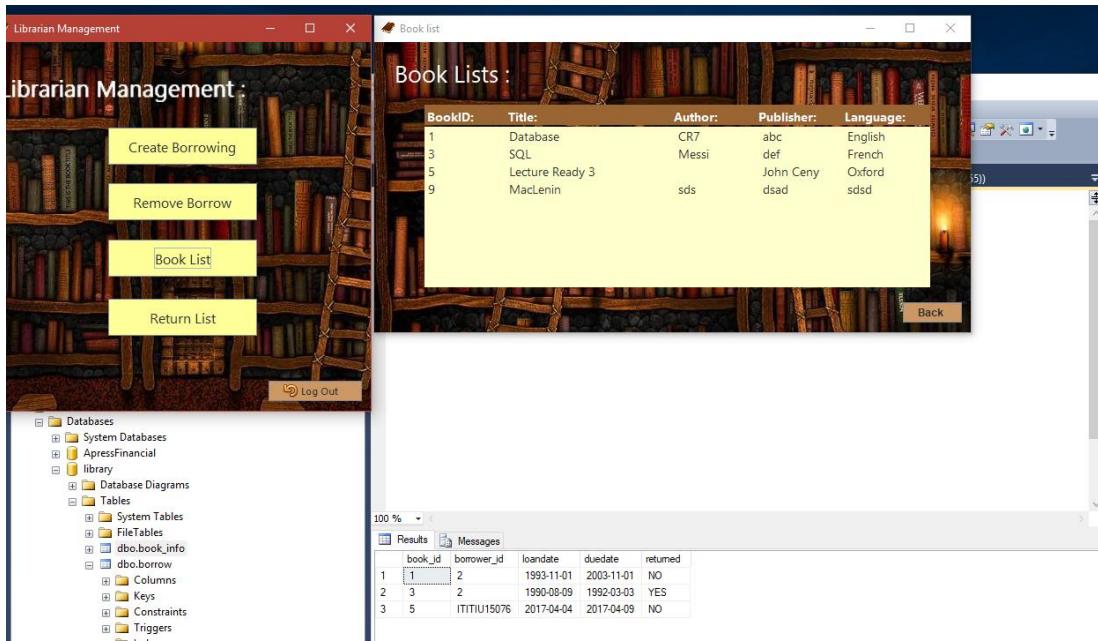
# PROGRAM

Here is the code :

```
private void jMenuItem1ActionPerformed (ActionEvent e) {
    public void actionPerformed(ActionEvent e) {
        try {
            Connection con = DriverManager.getConnection("jdbc:sqlserver://localhost;"+"databaseName=library;user=sa;password=123");
            Statement st = con.createStatement();
            String sql = "delete from borrow "+"where borrower_id='"+borid.getText()+"' and book_id='"+bookid.getText()+"';
            st.executeQuery(sql);

        }
        catch (Exception exc) {
            exc.printStackTrace();
        }
        JOptionPane.showMessageDialog(delete, "Delete successful!");
        bookid.setText(null); borid.setText(null);
    }
});
```

Another option show the list of all books in the library, the function is used as same as the View Staff of Admin :



Here is the function in JDBC :

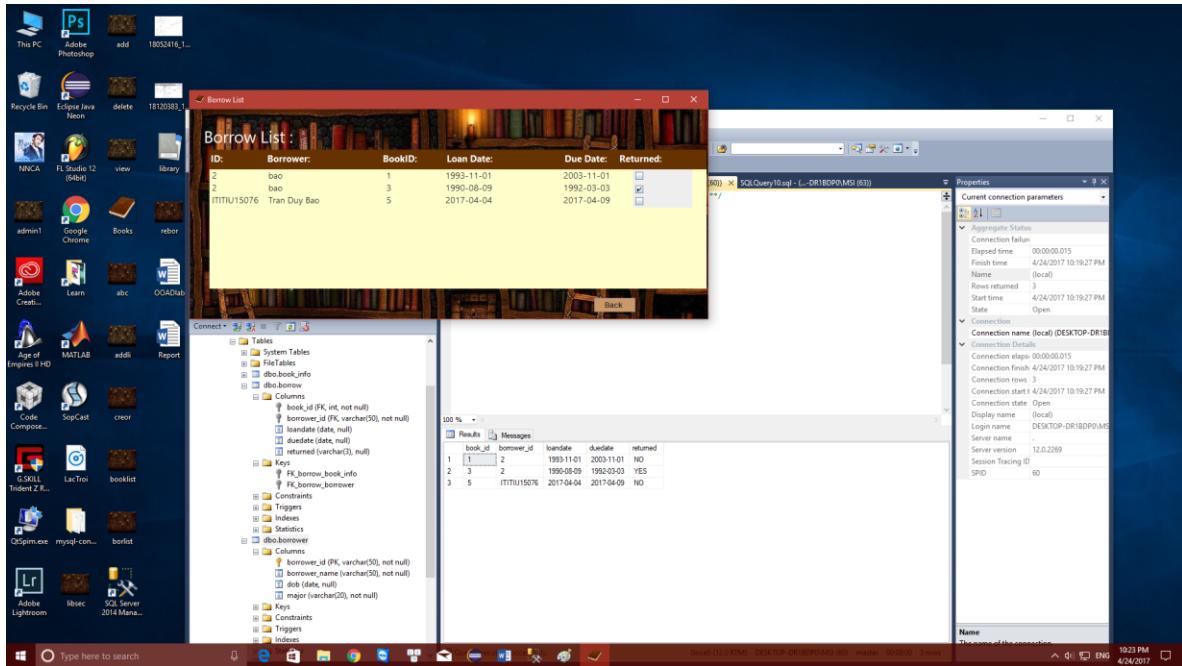
```
try {
    Connection con = DriverManager.getConnection("jdbc:sqlserver://localhost;"+"databaseName=library;user=sa;password=123");
    Statement st = con.createStatement();
    String sql = "select * from book_info ";
    ResultSet rs = st.executeQuery(sql);

    while(rs.next()) {
        result.append (" "+rs.getString("book_id")+"\t"+rs.getString("book_name")+"\t"+rs.getString("author")+"\t"+rs.getString("publisher")+
        +rs.getString("language")+"\n");
    }

} catch (Exception exc) {
    exc.printStackTrace();
}
result.setEditable(false);
```

# PROGRAM

And the last options is really important , it show the list of borrowing, contains the borrower and the book , the new point is that it shown the checkbox to see who has returned the book already:



The code below contains the query to select the list :

```
//function
try {
    Connection con = DriverManager.getConnection("jdbc:sqlserver://localhost;"+"databaseName=library;user=sa;password=123");
    Statement st = con.createStatement();
    String sql = "select distinct * from borrow,borrower where borrow.borrower_id=borrower.borrower_id";
    ResultSet rs = st.executeQuery(sql);

    while(rs.next()) {
        borlist.append(" "+rs.getString("borrower_id")+"\t"+rs.getString("borrower_name")+"\t\t"
        +rs.getString("book_id")+"\t"+rs.getString("loandate")+"\t"+rs.getString("duedate")+"\n");
    }
}
catch (Exception exc) {
    exc.printStackTrace();
}
```

**So that is what Librarian can use to manage the library , and that is all for JDBC  
The full code of this project contains many classes and each class has many lines ,  
so you can download the full project from the link below :**

[https://drive.google.com/open?id=0B6b\\_RfOtAEzkWEFRdjNVV3ISMXc](https://drive.google.com/open?id=0B6b_RfOtAEzkWEFRdjNVV3ISMXc)

# CONCLUSIONS

This project introduced a new program in order to manage the borrowing system in libraries. The system is made to help libraries' members to find and borrow books conveniently. It also help libraries' staffs to manage the borrowed books and the borrowers by linked their names and ID number.

The program provides an environment where the staffs can decide and update the borrowing date and returning date of books and the readers can search and ask for borrowing.