**CHAPTER 1**

**INTRODUCTION**

The project Library management system creates software that stores and manages all the data needed to describe the personal and their framework within an agency. It has a database administration that has access to the entire database, in regards with viewing, deleting and update of information. The exclusive right is implemented using authorized access.

The historyof[libraries](https://en.wikipedia.org/wiki/Libraries) began with the first efforts to organize collections of [documents](https://en.wikipedia.org/wiki/Document). Topics of interest include accessibility of the collection, acquisition of materials, arrangement and finding tools, the book trade, the influence of the physical properties of the different writing materials, language distribution, role in education, rates of literacy, budgets, staffing, libraries for specially targeted audiences, architectural merit, patterns of usage, and the role of libraries in a nation's cultural heritage, and the role of government, church or private sponsorship. Since the 1960s, issues of computerization and digitization have arisen.

A college library management is a project that manages and stores books information electronically according to students needs. The system helps library manager to keep a constant track of all the books available in the library. It allows both the admin and the student to search for the desired book. It becomes necessary for colleges to keep a continuous check on the books issued and returned and even calculate fine. This task if carried out manually will be tedious and includes chances of mistakes. These errors are avoided by allowing the system to keep track of information such as issue date, last date to return the book and even fine information and thus there is no need to keep manual track of this information which thereby avoids chances of mistakes.  
Thus this system reduces manual work to a great extent allows smooth flow of library activities by removing chances of errors in the details.

**CHAPTER 2**

**HISTORICAL REVIEW**

* 1. **History of Libraries**

The historyof[libraries](https://en.wikipedia.org/wiki/Libraries) began with the first efforts to organize collections of [documents](https://en.wikipedia.org/wiki/Document). Topics of interest include accessibility of the collection, acquisition of materials, arrangement and finding tools, the book trade, the influence of the physical properties of the different writing materials, language distribution, role in education, rates of literacy, budgets, staffing, libraries for specially targeted audiences, architectural merit, patterns of usage, and the role of libraries in a nation's cultural heritage, and the role of government, church or private sponsorship. Since the 1960s, issues of computerization and digitization have arisen.

The first libraries consisted of [archives](https://en.wikipedia.org/wiki/Archive) of the [earliest form of writing](https://en.wikipedia.org/wiki/Writing#The_beginning_of_writing) – the [clay tablets](https://en.wikipedia.org/wiki/Clay_tablet) in [cuneiform script](https://en.wikipedia.org/wiki/Cuneiform_script) discovered in temple rooms in [Sumer](https://en.wikipedia.org/wiki/Sumer) some dating back to 2600 BC. About an inch thick, tablets came in various shapes and sizes. Mud-like clay was placed in the wooden frames, and the surface was smoothed for writing and allowed to dry until damp. After being inscribed, the clay dried in the sun, or for a harder finish, was baked in a kiln. For storage, tablets could be stacked on edge, side by side, the contents described by a title written on the edge that faced out and was readily seen. The first libraries appeared five thousand years ago in Southwest Asia's [Fertile Crescent](https://en.wikipedia.org/wiki/Fertile_Crescent), an area that ran from Mesopotamia to the Nile in Africa. Known as the cradle of civilization, the Fertile Crescent was the birthplace of writing, sometime before 3000 BC. (Murray, Stuart A.P.) These archives, which mainly consisted of the records of commercial transactions or inventories, mark the end of [prehistory](https://en.wikipedia.org/wiki/Prehistory) and the start of [history](https://en.wikipedia.org/wiki/History)

* 1. **History of Database Management System**

Following the technology progress in the areas of processors, computer memory, computer storage, and computer networks, the sizes, capabilities, and performance of databases and their respective DBMSs have grown in orders of magnitude. The development of database technology can be divided into three eras based on data model or structure: navigational, SQL/relational, and post-relational.[2] The two main early navigational data models were the hierarchical model, epitomized by IBM's IMS system, and the CODASYL model (network model), implemented in a number of products such as IDMS.

The relational model employs sets of ledger-style tables, each used for a different type of entity. Only in the mid-1980s did computing hardware become powerful enough to allow the wide deployment of relational systems (DBMSs plus applications). By the early 1990s, however, relational systems dominated in all large-scale data processing applications, and as of 2015 they remain dominant: IBM DB2, Oracle, MySQL, and Microsoft SQL Server are the top DBMS. The dominant database language, standardized SQL for the relational model, has influenced database languages for other data models.

**2.3 History of MySQL**

MySQL is an open-source relational database management system (RDBMS). MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer. MySQL works on many system platforms, including Linux, macOS, Microsoft Windows, NetBSD. MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary [Enterprise Server](https://en.wikipedia.org/wiki/MySQL_Enterprise). MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.[1]

Major features that are available in MySQL are a broad subset of [ANSI SQL 99](https://en.wikipedia.org/wiki/SQL:1999),as well as extensions, Cross-platform support, [Stored procedures](https://en.wikipedia.org/wiki/Stored_procedure), using a procedural language that closely adheres to [SQL/PSM](https://en.wikipedia.org/wiki/SQL/PSM), [Triggers](https://en.wikipedia.org/wiki/Database_trigger), [Cursors](https://en.wikipedia.org/wiki/Cursor_(databases)), Updatable [views](https://en.wikipedia.org/wiki/View_(SQL)), [Online DDL](https://en.wikipedia.org/wiki/Data_Definition_Language) when using the InnoDB Storage Engine. Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's Visual Studio and the JDBC driver for Java. In addition, an ODBC interface called MySQL Connector/ODBC allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion.

**CHAPTER 3**

**REQIREMENT SPECIFICATION**

**3.1 System Requirements**

The basic requirements for the development of this mini project are as follows

**3.1.1 Hardware Configuration**

1. Processor: Intel core i3 or above
2. Ram:512 MB
3. Hard disk:20 GB

**3.1.2 Software Configuration**

1. Front end tool: HTML
2. Back end tool: MySQL
3. Development tools: XAMPP server
4. IDE : Eclipse & IntelliJ
5. Documentation tool: Microsoft office 2003 or above

**3.2 Development Environment**

**3.2.1 Frontend-Java**

Java is an entire programming language resembling C or C++. It takes a sophisticated programmer to create Java code. And it requires a sophisticated programmer to maintain it. With Java, you can create complete applications. Or you can attach a small group of instructions, a Java "applet" that improves your basic HTML. A Java Applet can also cause text to change colour when you roll over it. A game, a calendar, a scrolling text banner can all be created with Java Applets. There are sometimes compatibility problems between Java and various browsers, operating systems or computers, and if not written correctly, it can be slow to load. Java is a powerful programming language with excellent security, but you need to be aware of the tradeoffs.

**3.2.2 Backend-MySQL**

Itis an open source relational database management system (RDBMS).

The MySQL development project has made its source code available under the terms of GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offered additional functionality. MySQL is central component of LAMP open-source web application software stack. LAMP is an acronym of “Linux, Apache, MySQL, Perl/PHP/Python”. Application that use the MySQL database include TTPO3, MODx, Joomal, WordPress, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile. Large-scale websites, including Google, Facebook, Twitter, Flickr, YouTube.

**XAMPP**

XAMPP is a small and light Apache distribution containing the most common web development technologies in a single package. Its contents, small size, and portability make it the ideal tool for students developing and testing applications in PHP and MySQL. XAMPP is available as a free download in two specific packages: full and lite. While the full package download provides a wide array of development tools, XAMPP Lite contains the necessary technologies that meet the Ontario Skills Competition standards.The light version is a small package containing Apache HTTP Server, PHP, MySQL, phpMyAdmin, Openssl, and SQLite.

Operating System Windows 10

|  |  |
| --- | --- |
| [[Screenshot of the default MySQL command-line banner and prompt](https://en.wikipedia.org/wiki/File:Mysql-screenshot.PNG)](https://en.wikipedia.org/wiki/File:Mysql-screenshot.PNG)  [Screenshot](https://en.wikipedia.org/wiki/Screenshot) of the default MySQL command-line banner and prompt | |
| [**Original author(s)**](https://en.wikipedia.org/wiki/Software_developer) | [MySQL AB](https://en.wikipedia.org/wiki/MySQL_AB) |
| [**Developer(s)**](https://en.wikipedia.org/wiki/Software_developer) | [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation) |
| **Initial release** | 23 May 1995; 22 years ago |
|  | |
| [**Stable release**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 5.7.20[[1]](https://en.wikipedia.org/wiki/MySQL#cite_note-mysql_release-1) / 16 October 2017; 37 days ago |
| [**Preview release**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 8.0.3 rc[[2]](https://en.wikipedia.org/wiki/MySQL#cite_note-2) / 21 September 2017; 2 months ago |
| [**Repository**](https://en.wikipedia.org/wiki/Repository_(version_control)) | <https://github.com/mysql/mysql-server>, <git://anongit.gentoo.org/proj/mysql-extras.git> |
| **Development status** | Active |
| **Written in** | [C](https://en.wikipedia.org/wiki/C_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B)[[3]](https://en.wikipedia.org/wiki/MySQL#cite_note-3) |
| [**Operating system**](https://en.wikipedia.org/wiki/Operating_system) | [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://en.wikipedia.org/wiki/Linux), [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)), [macOS](https://en.wikipedia.org/wiki/MacOS), [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD)[[4]](https://en.wikipedia.org/wiki/MySQL#cite_note-4) |
| **Available in** | [English](https://en.wikipedia.org/wiki/English_language) |
| [**Type**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [RDBMS](https://en.wikipedia.org/wiki/Relational_database_management_system) |
| [**License**](https://en.wikipedia.org/wiki/Software_license) | [GPL](https://en.wikipedia.org/wiki/GNU_General_Public_License) (version 2) or [proprietary](https://en.wikipedia.org/wiki/Proprietary_software)[[5]](https://en.wikipedia.org/wiki/MySQL#cite_note-5) |
| **Website** | [www.mysql.com](http://www.mysql.com/) |

Fig 3.1 MySQL

|  |  |
| --- | --- |
| XAMPP | |
| [Xampp logo.svg](https://en.wikipedia.org/wiki/File:Xampp_logo.svg) | |
| [XAMPP In Action on OSX.png](https://en.wikipedia.org/wiki/File:XAMPP_In_Action_on_OSX.png)  Fig 3.2 A screenshot of XAMPP running all servers on Mac OS X | |
| [**Developer(s)**](https://en.wikipedia.org/wiki/Software_developer) | | Apache Friends | |
| **Initial release** | | May 22, 2002; 15 years ago | |
|  | | | |
| [**Stable release**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | | 7.1.11 - [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) 7.1.11 - [Linux](https://en.wikipedia.org/wiki/Linux) 7.1.11 - [macOS](https://en.wikipedia.org/wiki/MacOS) / November 10, 2017; 7 days ago | |
| [**Repository**](https://en.wikipedia.org/wiki/Repository_(version_control)) | | <https://github.com/ApacheFriends> | |
| **Development status** | | Active | |
| **Written in** | | Various Languages | |
| [**Operating system**](https://en.wikipedia.org/wiki/Operating_system) | | [Cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [Linux](https://en.wikipedia.org/wiki/Linux) [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)) [macOS](https://en.wikipedia.org/wiki/MacOS) | |
| [**Platform**](https://en.wikipedia.org/wiki/Computing_platform) | | [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) - [2008](https://en.wikipedia.org/wiki/Windows_Server_2008), [2012](https://en.wikipedia.org/wiki/Windows_Server_2012), [Vista](https://en.wikipedia.org/wiki/Windows_Vista), [7](https://en.wikipedia.org/wiki/Windows_7), [8](https://en.wikipedia.org/wiki/Windows_8), [10](https://en.wikipedia.org/wiki/Windows_10) x32 Bit [Linux](https://en.wikipedia.org/wiki/Linux) - [Debian](https://en.wikipedia.org/wiki/Debian), [RedHat](https://en.wikipedia.org/wiki/Red_Hat_Linux), [CentOS](https://en.wikipedia.org/wiki/CentOS), [Ubuntu](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)), [Fedora](https://en.wikipedia.org/wiki/Fedora_(operating_system)), [Gentoo](https://en.wikipedia.org/wiki/Gentoo_Linux), [Arch](https://en.wikipedia.org/wiki/Arch_Linux), [SUSE](https://en.wikipedia.org/wiki/SUSE) x32 or x64 Bit [macOS](https://en.wikipedia.org/wiki/MacOS) - [10.6](https://en.wikipedia.org/wiki/Mac_OS_X_Snow_Leopard) or later x64 Bit | |
| [**Size**](https://en.wikipedia.org/wiki/File_size) | | [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) x32 Bit - 120 Mb [Linux](https://en.wikipedia.org/wiki/Linux) x64 Bit - 137 Mb [macOS](https://en.wikipedia.org/wiki/MacOS) x64 Bit - 137 Mb | |
| **Available in** | | 11 languages | |
| [[show]](https://en.wikipedia.org/wiki/XAMPP)  **List of languages** | | | |
|  | | | |
| [**Type**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | | [WAMP](https://en.wikipedia.org/wiki/WAMP), [MAMP](https://en.wikipedia.org/wiki/MAMP), [SAMP](https://en.wikipedia.org/wiki/SAMP_(Sun_Web_Stack)), [LAMP](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) | |
| [**License**](https://en.wikipedia.org/wiki/Software_license) | | [GNU General Public Licence](https://en.wikipedia.org/wiki/GNU_General_Public_License) | |
| [**Alexa**](https://en.wikipedia.org/wiki/Alexa_Internet)**rank** | | Decrease10,525[[1]](https://en.wikipedia.org/wiki/XAMPP#cite_note-1) | |
| **Website** | | [apachefriends.org](https://www.apachefriends.org/) | |

**Normalization**

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly. To overcome these anomalies we need to normalize the data. There are 4 basic types of normalizations. They are:

* First normal form(1NF)
* Second normal form(2NF)
* Third normal form(3NF)
* Boyce & Codd normal form (BCNF)

A table is said to be in 2NF if the two conditions stated are satisfied. The table is in First normal form and all the non-prime attribute are dependent on the proper subset of any candidate key of table. The attribute that is not part of any candidate key are known as non-prime attribute.

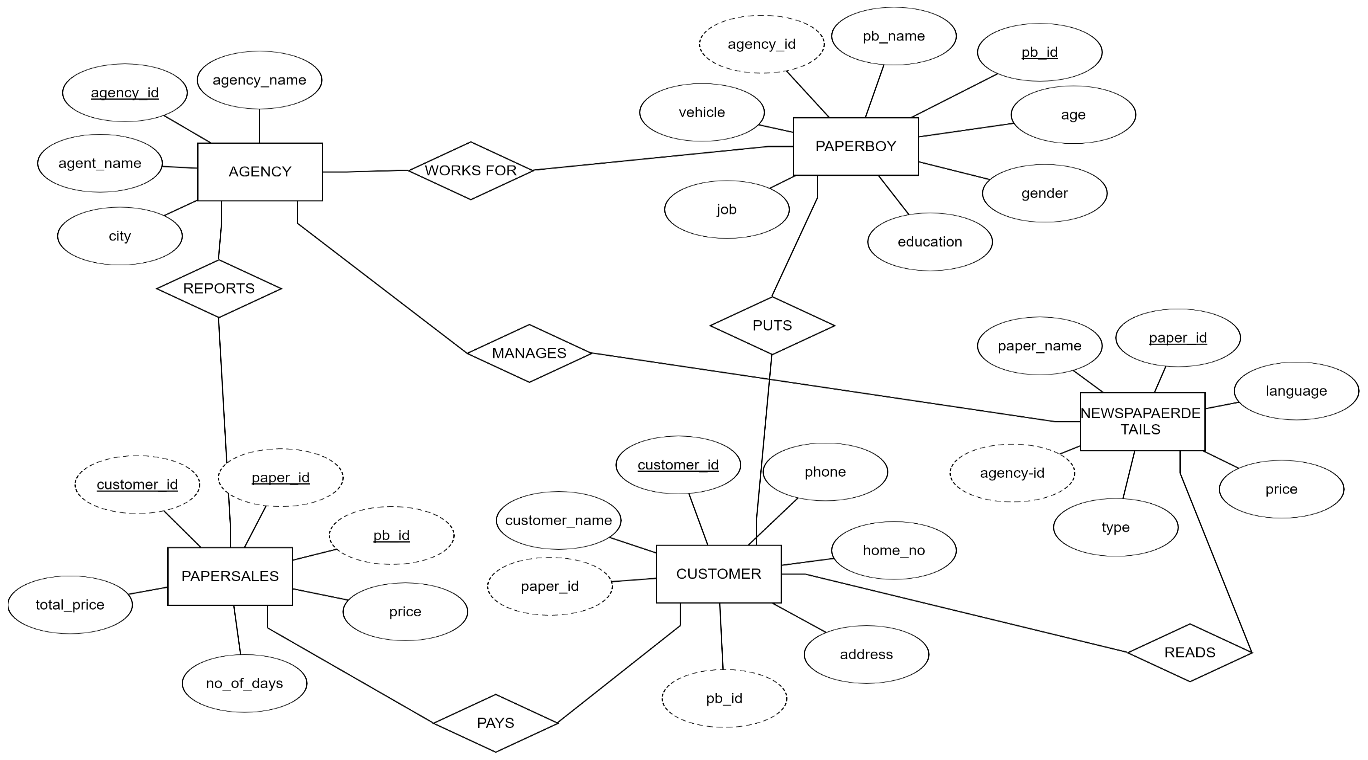
A table design is said to be in 3NF if the table is in 2NF and [Transitive functional dependency](https://beginnersbook.com/2015/04/transitive-dependency-in-dbms/) of non-prime attribute on any super key are removed.

A table design is said to be in BCNF if there is only one super key.

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 E-R Diagram**



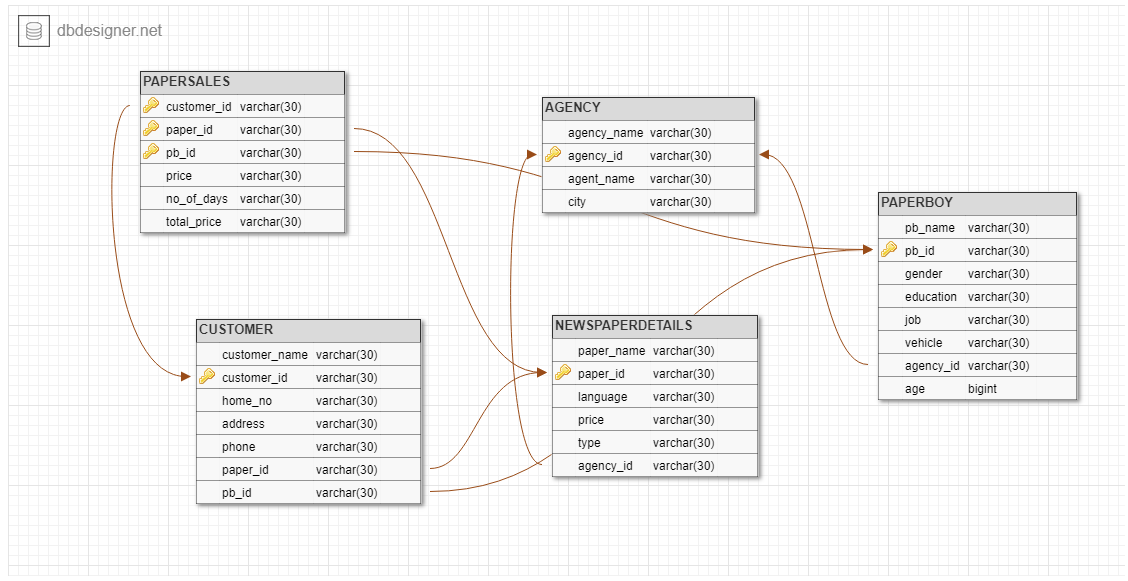
**Figure 4.1 E-R Diagram of Library Management System**

An entity–relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

In software engineering an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database, typically a relational database.

**4.2 Schema Diagram**

The database schema of a database system is its structure described in a formal language supported by the database management system (DBMS). The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). The formal definition of a database schema is a set of formulas (sentences) called integrity constraints imposed on a database.

****

**Figure 4.2** **Schema Diagram**

* 1. **Pseudo codes for Library management system**

**4.3.1 Algorithm for Table Display**

**Step1: BEGIN**

**Step 2:** Establish connection with the database

**Step 3:** Define Result Set to return all the values from the query executed.

**Step 4:** Define the select query to retrieve all the values from the DBMS.

**Step 5:** The tables are stored in the result set object.

**Step 6:** **END**

**4.3.2 Algorithm for Insert**

**Step 1: BEGIN**

**Step 2:** Get all the necessary values required for insertion into variable defined in the method.

**Step 3:** Define the query for insertion as stated.

**Step 4:** Execute the Query.

**Step 5:** **END**

**4.3.3 Algorithm for update**

**Step 1: BEGIN**

**Step 2:** Get all the necessary values required for updating the values from the text field.

**Step 3:** Define the Query for Updating as stated above.

**Step 4:** Execute the Query.

**Step 5:** **END**

* + 1. **Algorithm for Delete**

**Step 1: BEGIN**

**Step 2:** Get the identification number of the entity which is to be deleted.

**Step 3:** Display the details and ask whether to confirm the deletion process.

**Step 4:** Execute the Query and show Confirm Message Dialog Box.

**Step 5:** **END**

**CHAPTER 5**

**SYSTEM IMPLEMENTATION**

**5.1 Modules Description**

**\*Create,** create table statement is used to create table to store data. Integrity constraints like primary key, foreign key, unique key, can be defined while creating the table.

**Create code for every table**

**5.1.1 Account Table**

CREATE TABLE `account` (

`username` varchar(20) NOT NULL,

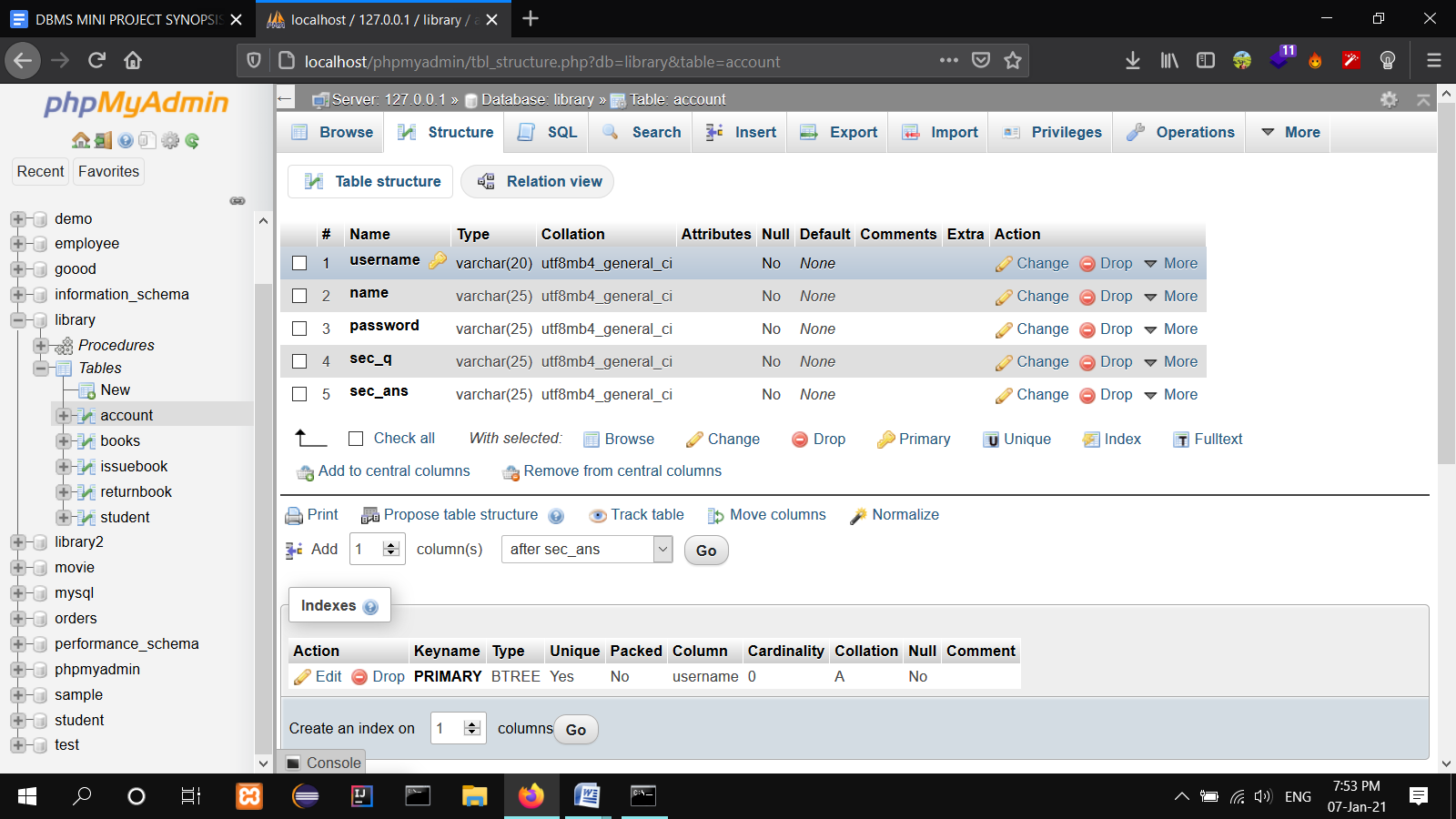
`name` varchar(25) NOT NULL,

`password` varchar(25) NOT NULL,

`sec\_q` varchar(25) NOT NULL,

`sec\_ans` varchar(25) NOT NULL,

PRIMARY KEY (`username`);



**Figure 5.1.1: Account Table Description.**

**5.1.2 Books table**

CREATE TABLE `books` (

`book\_id` varchar(10) NOT NULL,

`name` varchar(40) DEFAULT NULL,

`isbn` varchar(20) DEFAULT NULL,

`publisher` varchar(30) DEFAULT NULL,

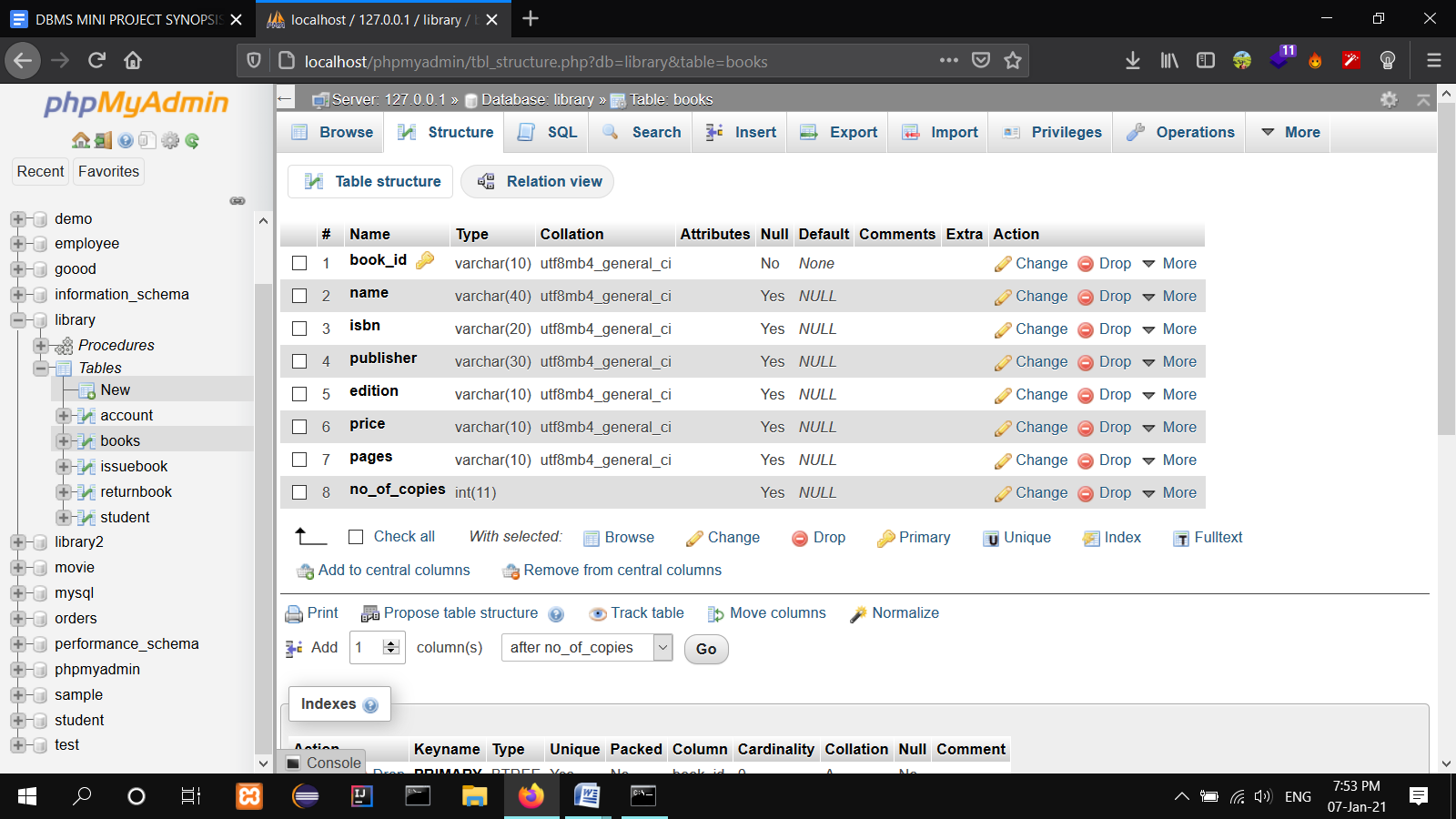
`edition` varchar(10) DEFAULT NULL,

`price` varchar(10) DEFAULT NULL,

`pages` varchar(10) DEFAULT NULL,

`no\_of\_copies` int(11) DEFAULT NULL,

PRIMARY KEY (`book\_id`);

****

**Figure 5.1.2:** Books table description.

**5.1.3 Students Details table**

CREATE TABLE `student` (

`student\_id` varchar(10) NOT NULL,

`name` varchar(25) NOT NULL,

`father` varchar(25) NOT NULL,

`course` varchar(10) DEFAULT NULL,

`branch` varchar(10) DEFAULT NULL,

`year` varchar(10) DEFAULT NULL,

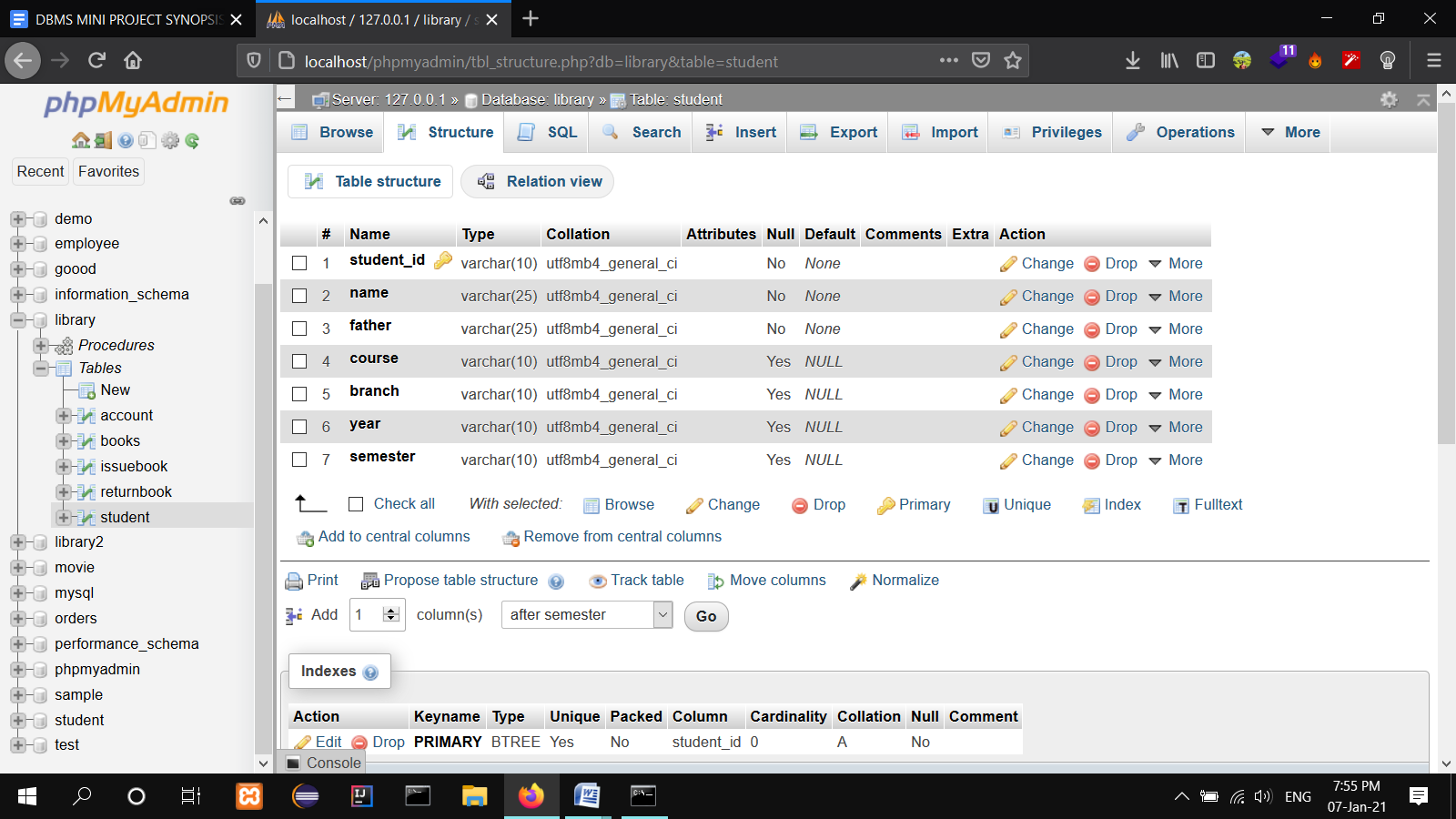
`semester` varchar(10) DEFAULT NULL,

PRIMARY KEY (`student\_id`),

CONSTRAINT `CONSTRAINT\_1` CHECK (`name` <> ''),

CONSTRAINT `CONSTRAINT\_2` CHECK (`father` <> '')

);

****

**Figure 5.1.3:** Student Details table description.

**5.1.4 Issued Books table**

CREATE TABLE `issuebook` (

`book\_id` varchar(10) DEFAULT NULL,

`student\_id` varchar(10) DEFAULT NULL,

`bname` varchar(40) DEFAULT NULL,

`sname` varchar(40) DEFAULT NULL,

`course` varchar(20) DEFAULT NULL,

`branch` varchar(10) DEFAULT NULL,

`dateOfIssue` varchar(20) DEFAULT NULL,

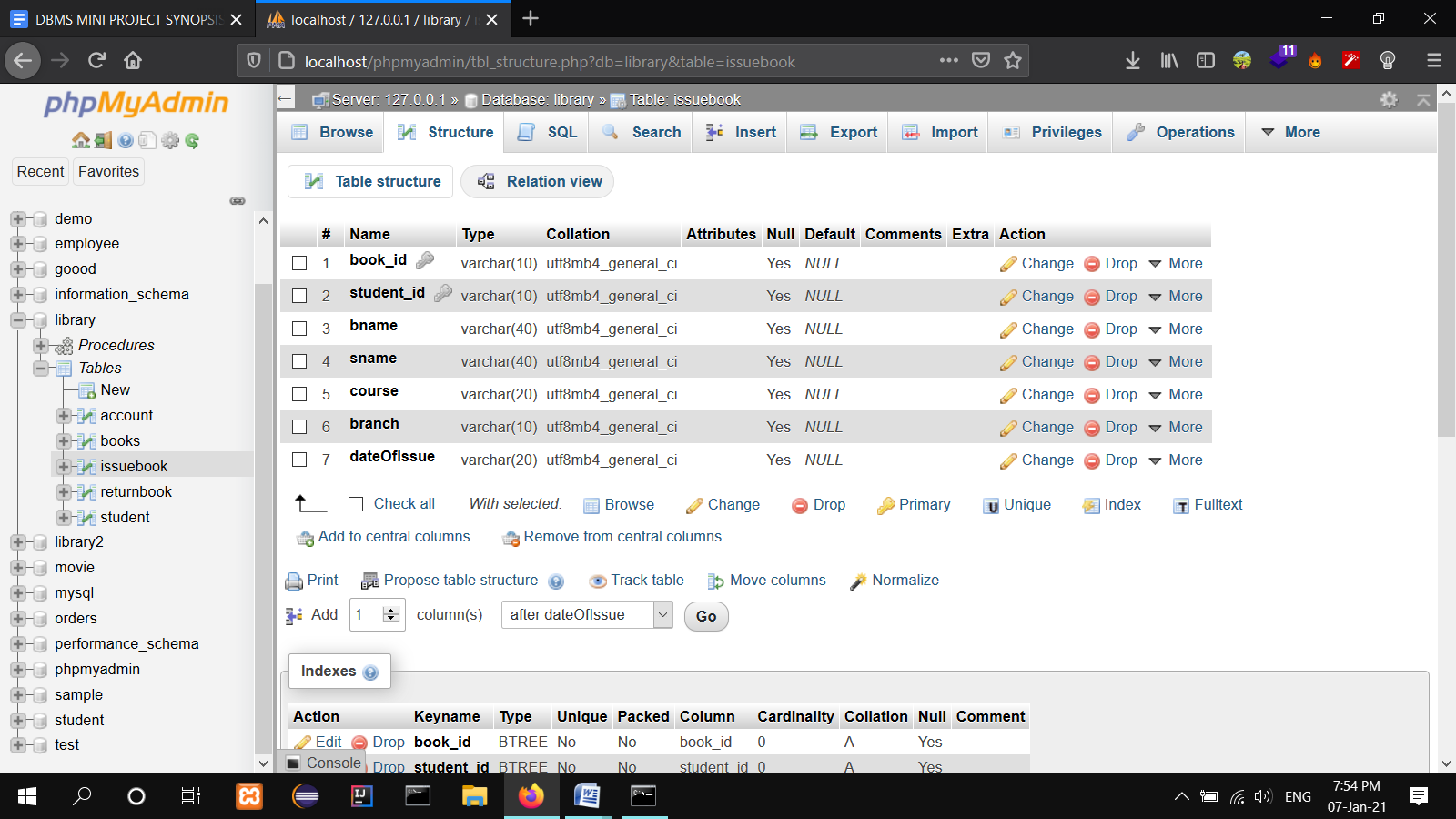
KEY `book\_id` (`book\_id`),

KEY `student\_id` (`student\_id`),

CONSTRAINT `issuebook\_ibfk\_1` FOREIGN KEY (`book\_id`) REFERENCES `books` (`book\_id`) ON DELETE CASCADE,

CONSTRAINT `issuebook\_ibfk\_2` FOREIGN KEY (`student\_id`) REFERENCES `student` (`student\_id`) ON DELETE CASCADE

);

****

**Figure 5.1.4:** Issued Books table description.

**5.1.5 Returned Books table**

CREATE TABLE `returnbook` (

`book\_id` varchar(10) DEFAULT NULL,

`student\_id` varchar(10) DEFAULT NULL,

`bname` varchar(40) DEFAULT NULL,

`sname` varchar(40) DEFAULT NULL,

`course` varchar(20) DEFAULT NULL,

`branch` varchar(10) DEFAULT NULL,

`dateofissue` varchar(20) DEFAULT NULL,

`dateofreturn` varchar(20) DEFAULT NULL,

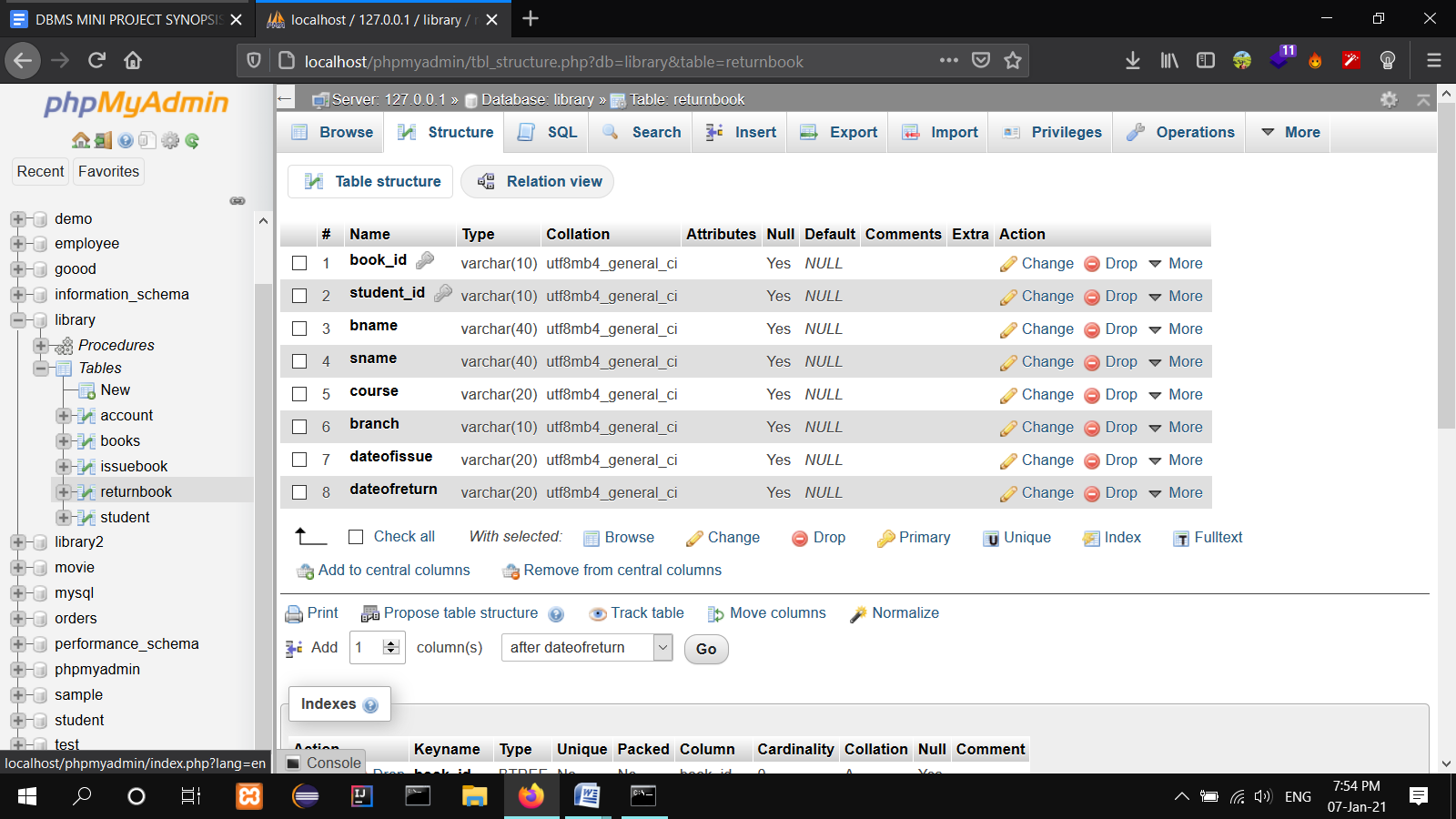
KEY `book\_id` (`book\_id`),

KEY `student\_id` (`student\_id`),

CONSTRAINT `returnbook\_ibfk\_1` FOREIGN KEY (`book\_id`) REFERENCES `books` (`book\_id`) ON DELETE CASCADE,

CONSTRAINT `returnbook\_ibfk\_2` FOREIGN KEY (`student\_id`) REFERENCES `student` (`student\_id`) ON DELETE CASCADE

);

****

**Figure 5.1.5:** Returned Books table description

**5.2 OPERATIONS PERFORMED:**

**\*Insertion,** insertion is use to insert the tuple or row to the table. We insert value from frontend by making use HTTP the value inserted from to frontend will be going to store in backend database in XAMPP Server.

**\*Update,** update will help to edit the tables in the database. In this project we have given update option for table package, to update the place column in package table.

**\*Delete,** delete will help us to delete a tuple or row from the table. In this project we have delete option for table employee to delete the particular row or employee information from the table.

**\* Trigger,** a trigger is a special kind of stored procedure that automatically executes when an event occurs in the database server. DML triggers execute when a user tries to modify data through a data manipulation language(DML) event. DML events are INSERT, UPDATE, OR DELETE statements on a table or view. In this project we added trigger on trigger on table package to give discount of 10% if the group has more than 10 travelers

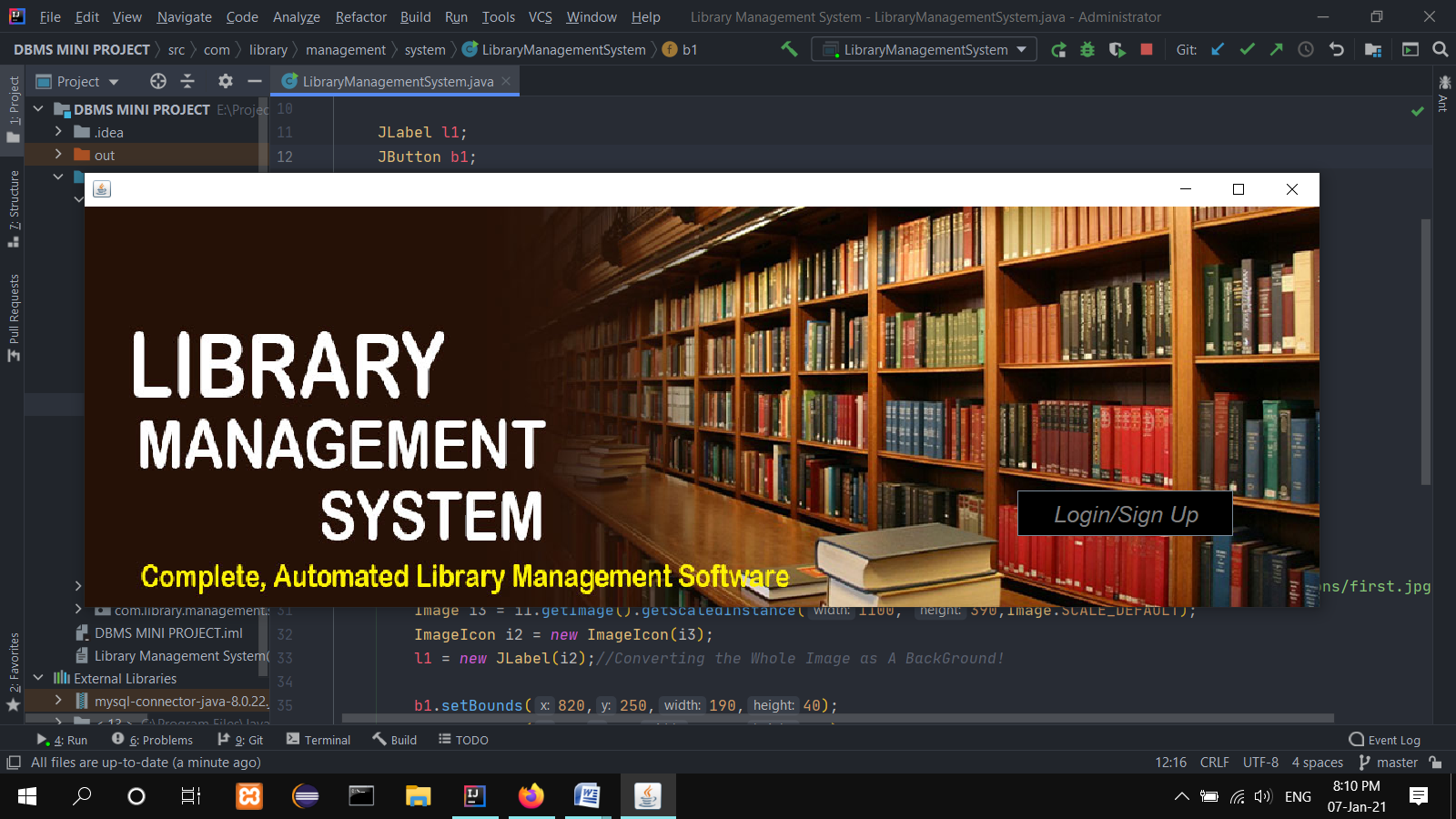
**\*Stored procedure,** a stored procedure is a set of Structure Query Language(SQL) statements with an assigned name, which are stored in a relational database management system as a group, so it can be reused and shared by multiple programs. Stored procedure can access or modify data in a database, in this project we added stored procedure for table package to find income of our site.

**CHAPTER 6**

**SAMPLE OUTPUT**

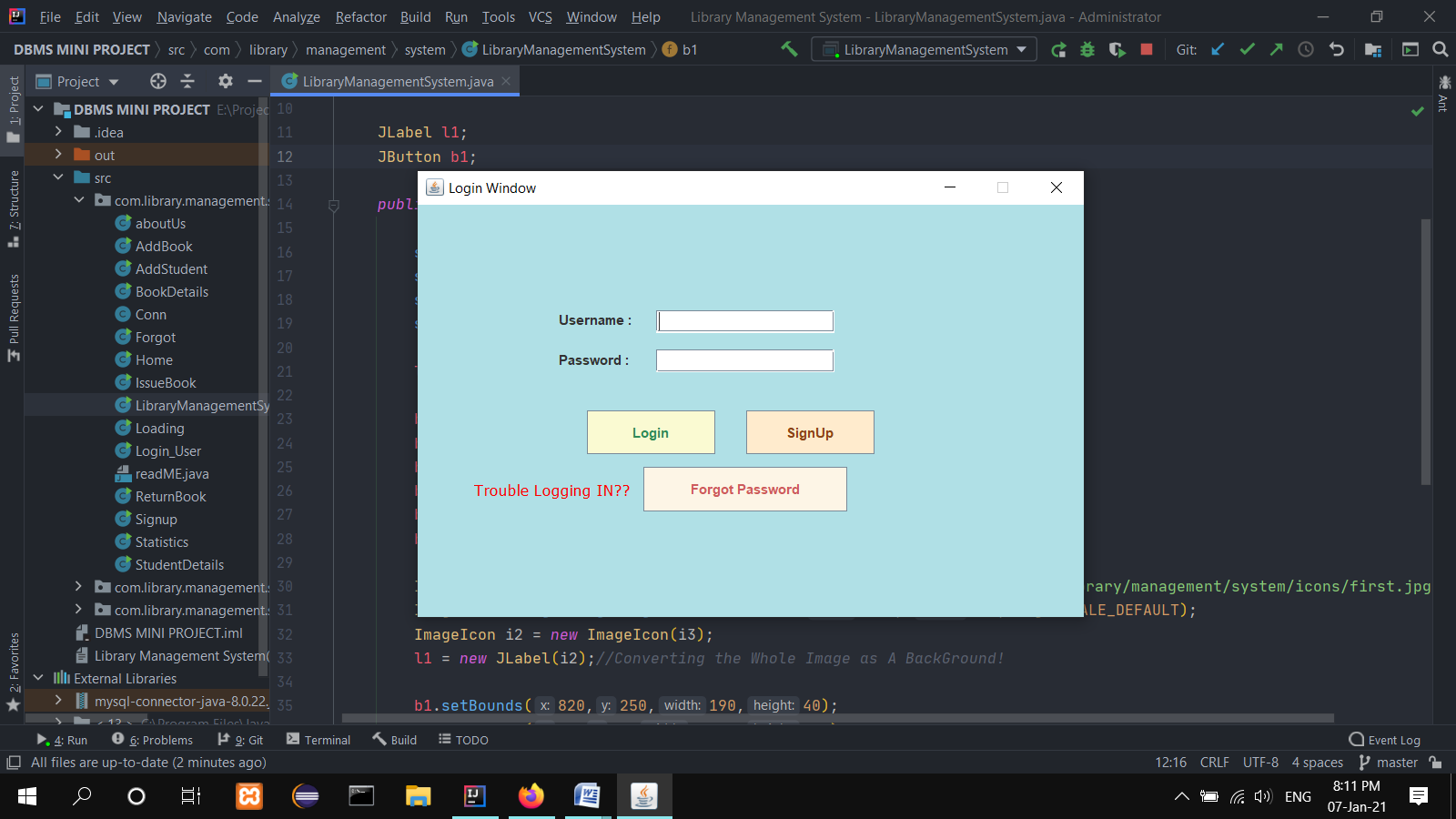
**6.1 Screenshots**

**6.1.1** **Welcome Page**

****

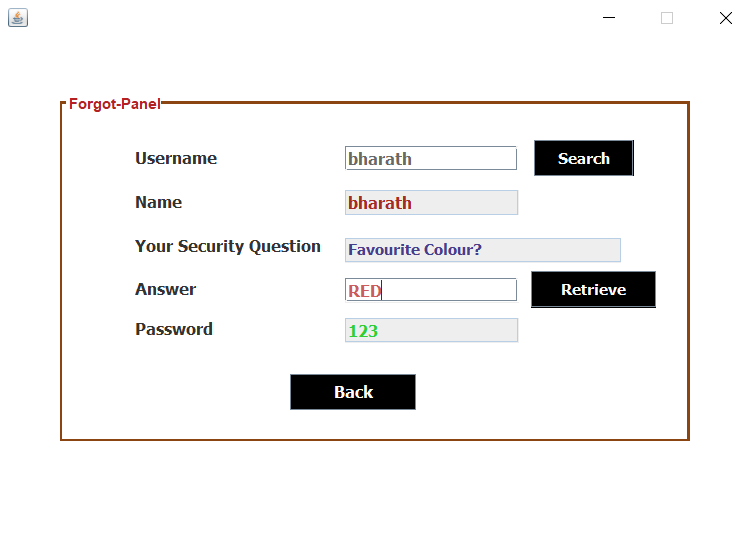
**Fig 6.1.1** Welcome page

**6.1.2 Login Frame**

****

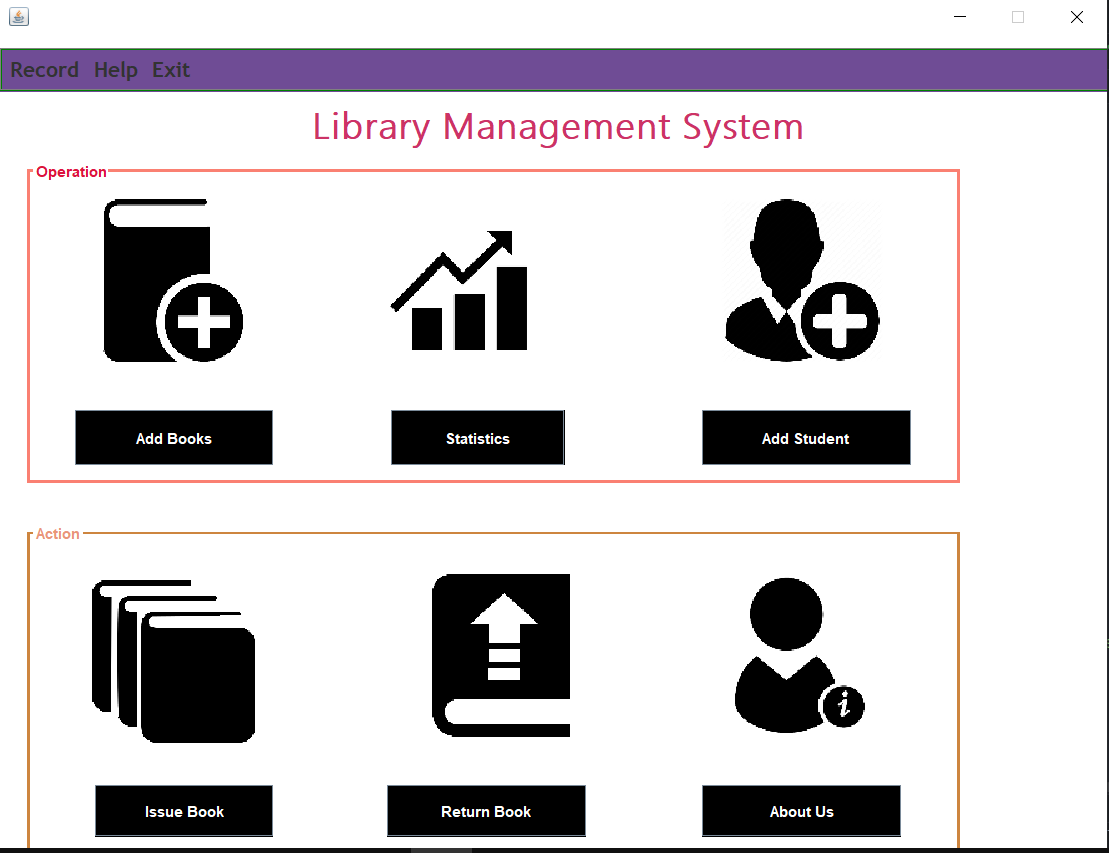
**Fig 6.1.2** Login Page

**6.1.3 Forgot Panel**



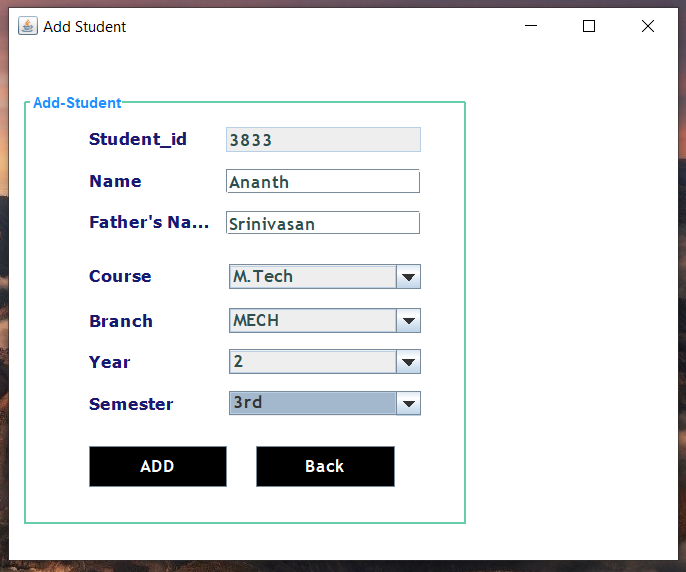
**Fig 6.1.3** ForgotPanel

**6.1.4 Home Page**



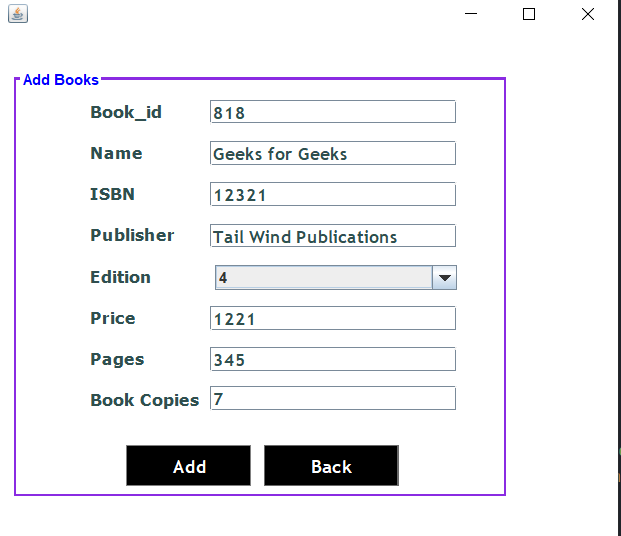
**Fig 6.1.4** Home Page

**6.1.5 Add Students Page**

****

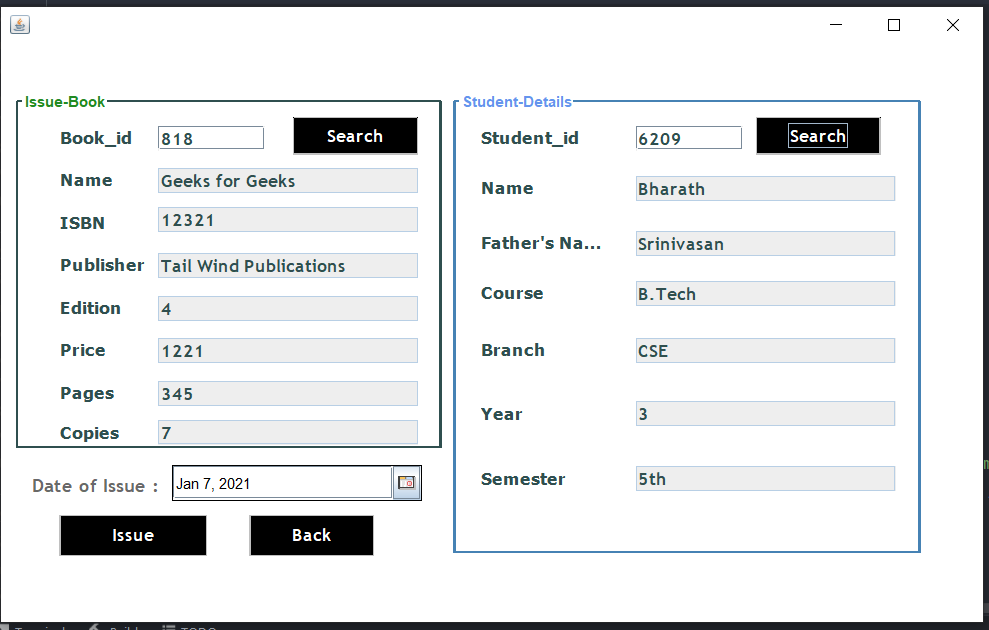
**Fig 6.2.2** Add Students Frame

**6.1.6 Add Books Page**

****

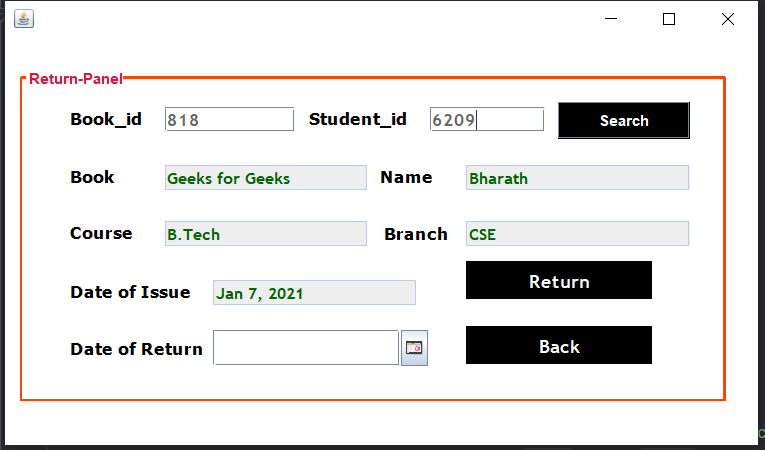
**Fig 6.1.7** AddBooks Frame

**6.1.7 Issue Books Page**

****

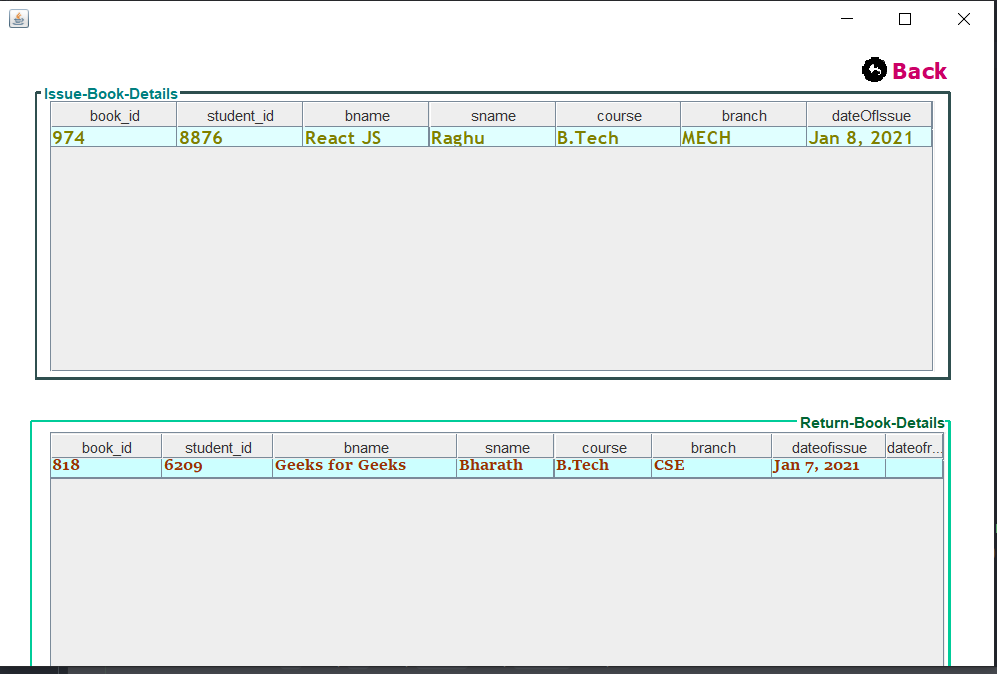
**Fig 6.1.8** Issue Books Frame

**6.1.8 Return Books Page**

****

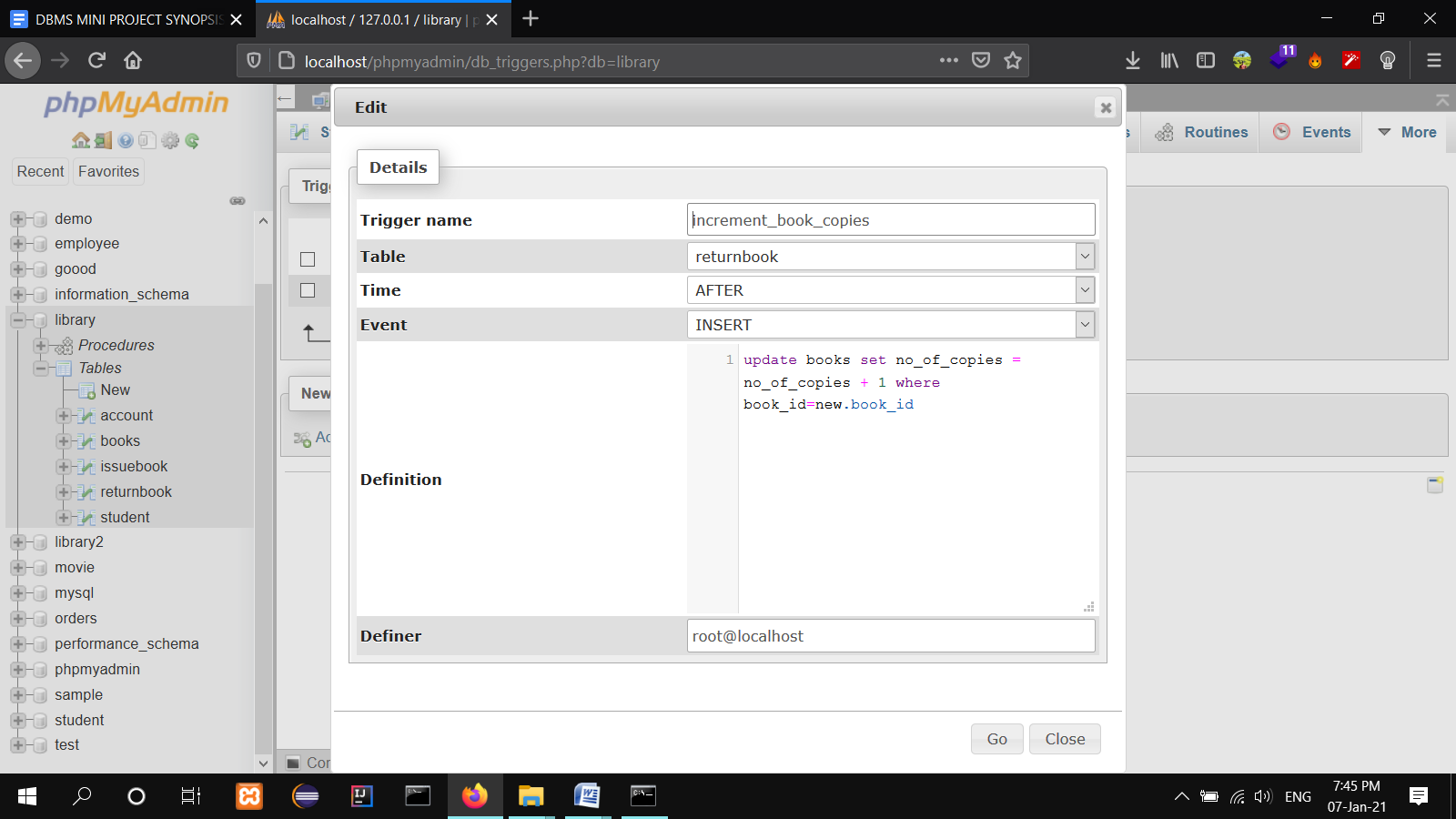
**Fig 6.1.9** Return Books Frame

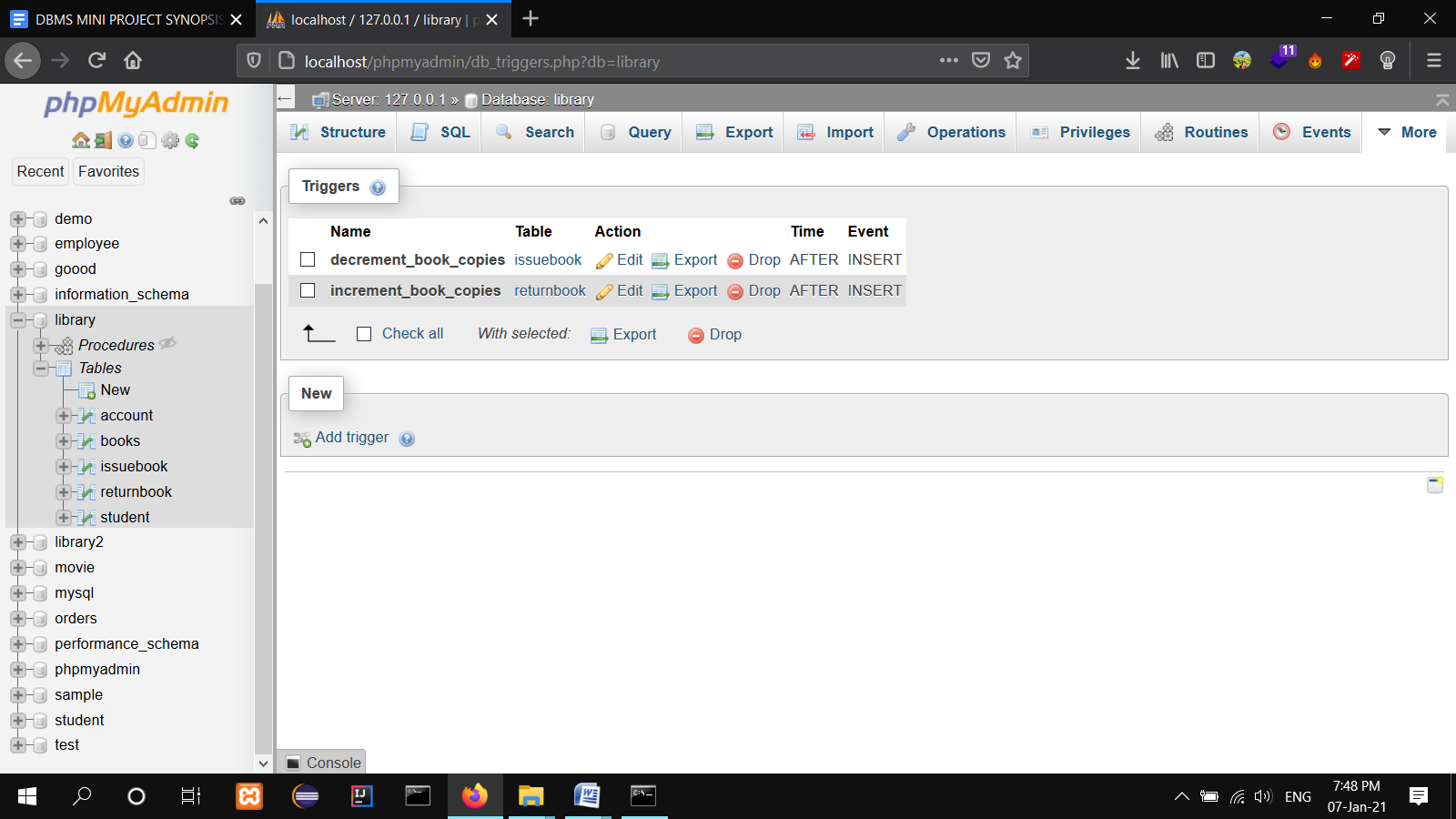
**6.1.9 Book Statistics**

****

**Fig 6.2.0** Book Statistics

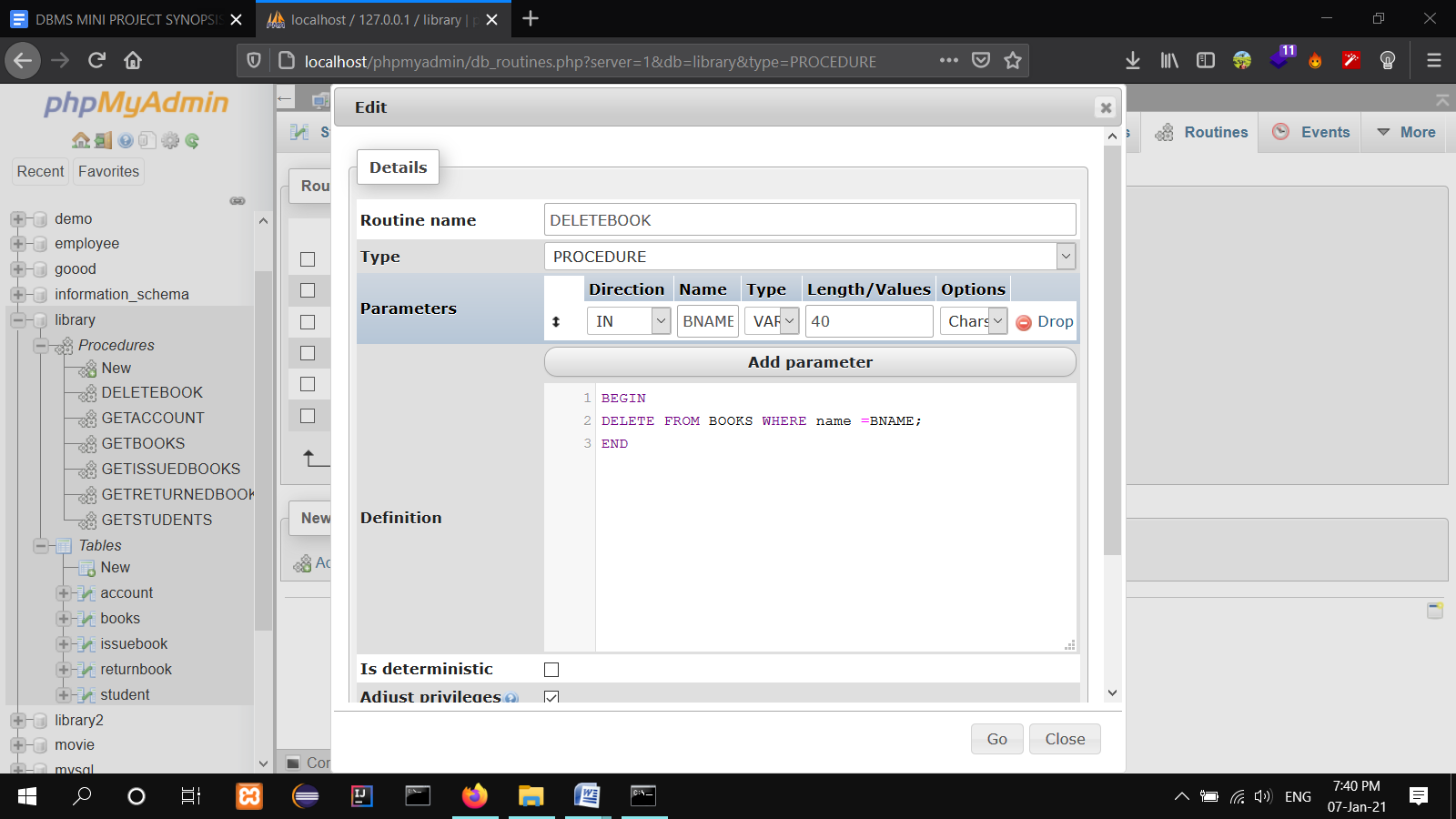
**6.2.0 TRIGGERS**

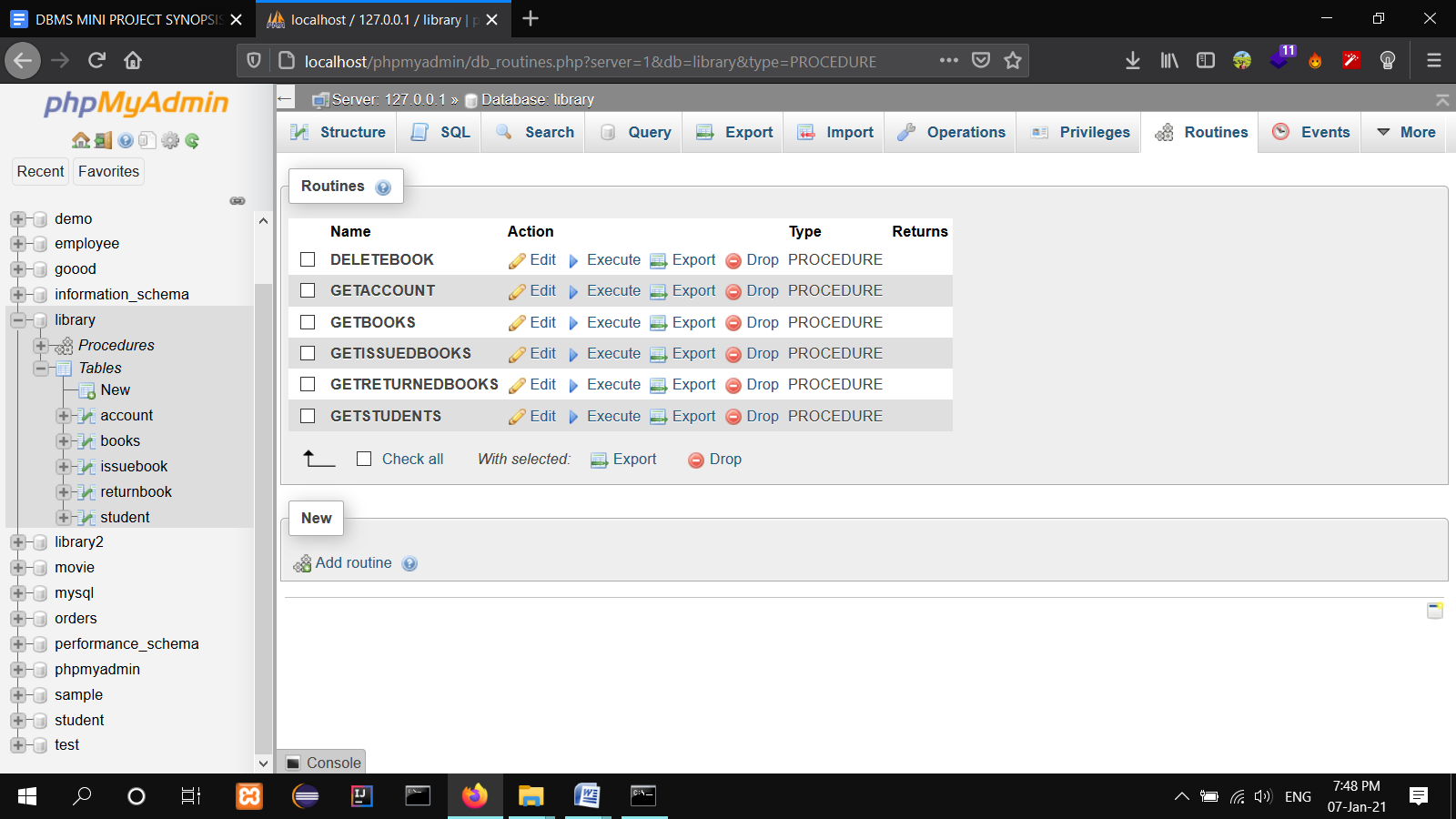




**Fig 6.2.1 TRIGGER**

**6.2.1 STORED PROCEDURES**

****

****

**Fig 6.2.2 STORED PROCEDURE**

**Chapter 7**

**CONCLUSION**

To conclude the description about the project, this was developed using XAMPP Server with Java & MySQL based on the requirement specification of the users and analysis of the existing system, with flexibility for future enhancement. Library Management System is very useful for agency managers. This software is designed for Librarian Admin who want to keep the record of the books in library that are circulated in the library.

Future work:

1. Implementation of algorithm sending message to students if returning of books are delayed

2. Uses feedback-based evolution of data base

3. Connecting other library branches across the city.

**BIBLIOGRAPHY**

[1] [www.w3schools.com](http://www.w3schools.com)

[2] [www.YouTube.com](http://www.YouTube.com)

[3] [www.quora.com](http://www.quora.com)

[4] [www.Wikipedia.org](http://www.Wikipedia.org)

[5] Ramez Elmasri and Shamkant B. Navathe, 7th edition, 2017, Pearson