MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
NATIONAL TECHNICAL UNIVERSITY

“KHARKIV POLYTECHNIC INSTITUTE”

Department of Software Engineering and Intelligent Management Technology

- Author -

Ягизджан Явуз From Group КН221-ia.e

Ібрагім Чагтюрк Караджан From Group КН221-ia.e

Батухан Увер From Group КН221-ia.e

- Supervisor -

Андрій Михайлович Копп

Дмитро Миколайович Ковальч

Kharkiv – 2023

CONTENT

INTRODUCTION

In the context of library management, the process typically involves a Reader placing an order for a book, which is then searched for in the Catalog. The Librarian either gives the reader access to the book on a subscription basis or allows them to read it within the library's reading room. It is important to note that failure to return the borrowed book within the specified timeframe may result in the Reader being blacklisted by the Administrator. With this in mind, a robust library system that manages these processes efficiently is essential to ensuring smooth and effective library operations.

Additionally, to provide a visual representation of the library system, an example screenshot Figure 1.1 of the program is available for reference. This screenshot showcases the various features and functionalities of the program, including the ability to search and edit records, paginate through data sets, and monitor system performance. By viewing this example screenshot, users can gain a better understanding of how the program operates and how it can be utilized to manage library operations effectively.

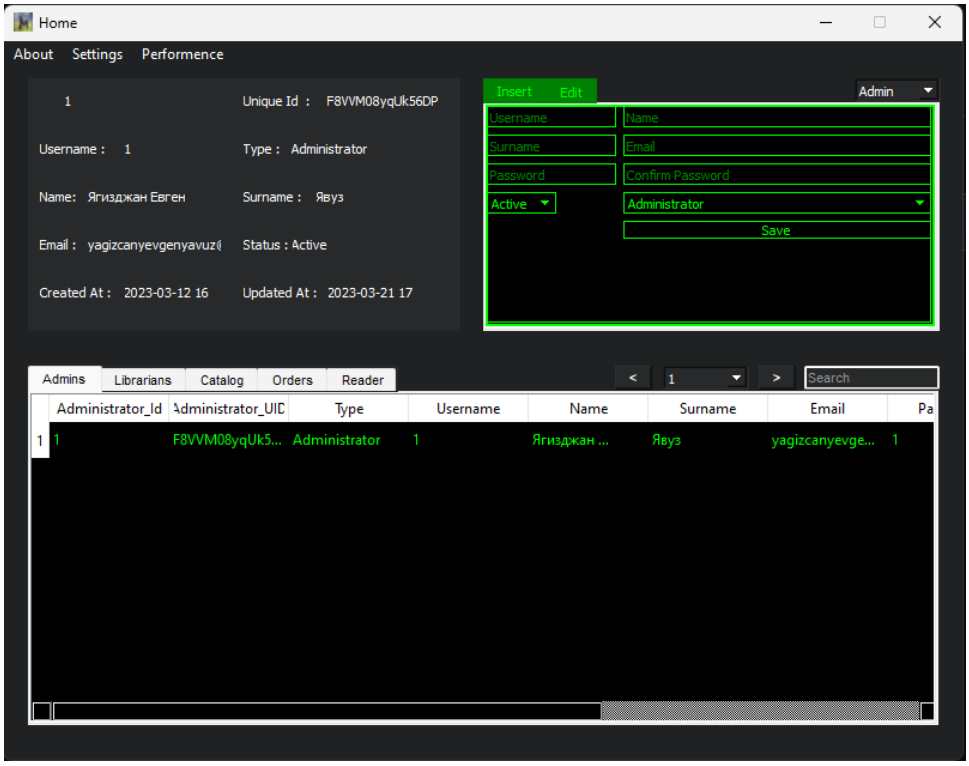

Figure 1.1

1 ANALYSIS OF THE SUBJECT DOMAIN

1. After conducting thorough research and analysis of the subject domain, a system of library rules has been established. The following library rules have been identified:
2. The first step in using the library system is to manually register an Administrator or Librarian in the database to enable access to the application.
3. In order to place an Order, a book must first be registered in the Catalog table and a user must be registered in the Reader table. Once these steps are complete, the Registered User can then place an Order for the desired book.
4. If a user fails to return a book, the Librarian or Administrator has the authority to block the user from borrowing additional books.
5. When a user orders a book, it is automatically deducted from the total number of available books. When the book is returned, the number of available books is recalculated and updated accordingly.
6. It is necessary to select the appropriate Type for insertion or editing actions.
7. When editing a record from a table, the user must select the appropriate Type based on the table and click once on the record to be edited.
8. An instant search feature is available to search records based on the selected table.
9. These rules have been designed to ensure the efficient and effective operation of the library system.
10. Additionally, the library system allows users to paginate through records using the dropdown menu or by tapping the Next and Previous buttons. This feature provides a convenient way for users to navigate through large sets of data without having to scroll through multiple pages.
11. Furthermore, users can monitor performance changes while using the program by clicking on the Performance button located in the Menu tab. This feature provides valuable insights into the system's performance and helps users optimize their workflow for maximum efficiency. By analyzing system performance data, users can identify areas for improvement and make necessary adjustments to improve overall productivity.

12. Moreover, users have the ability to modify database connections by accessing the Connection tab located in the Menu bar. This feature enables users to easily switch between different databases or modify connection settings as needed. By providing a flexible and user-friendly interface for managing database connections, the system enhances usability and allows users to work more efficiently.
13. Users can access the instructions for the library management system by clicking on the "Settings" option in the menu bar and selecting "Instructions" from the dropdown list. This feature provides users with a detailed guide on how to navigate and use the various functions of the application. By following the instructions, users can easily perform tasks such as adding new books, creating orders, searching for books or orders, and managing user accounts. This enhances the user experience and ensures efficient usage of the library management system.

1.1 Modeling of the subject area

To provide a clear and concise representation of the subject area model, it has been visualized in the notation of an Application workflow. The model is displayed in Figure 1.2 for reference. This workflow diagram highlights the various steps and processes involved in managing library operations, including registering an Administrator or Librarian, adding books to the Catalog and Readers to the Reader table, placing Orders, and managing User accounts. By viewing this workflow diagram, users can gain a better understanding of the logical flow of the library system and how it operates to facilitate effective management of library operations.

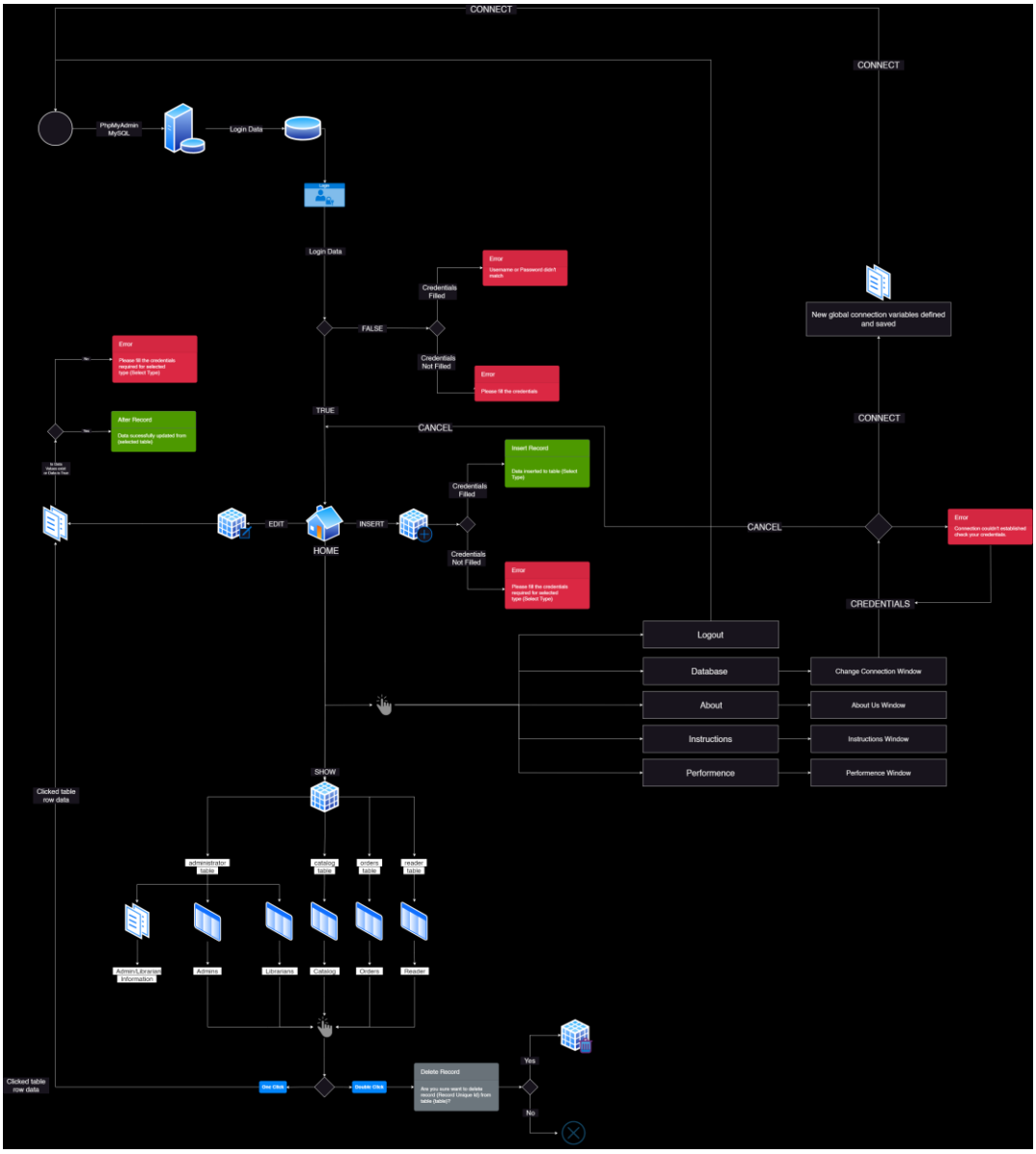

Figure 1.2

1.2 Problem Statement

To achieve the goal of the work, the following tasks must be performed:

- Develop a logical model of the database using IDEF1X notation.
- Develop a physical database model by converting the logical model into a physical database schema.
- Describe the structure of the database by documenting the tables, attributes, and relationships in the database.
- Implement the database using the MySQL database management system (DBMS) by executing Data Definition Language (DDL) commands.
- Populate the database with initial records and develop queries using SQL (Structured Query Language).
- Develop a desktop database application using Python and PyQt5, incorporating the MySQL database as the data source.
- Demonstrate the usage example of the database application by performing various operations, such as adding new books, creating orders, and searching for books or orders.

The ultimate goal of the project is to develop a functional and user-friendly library management system that enables efficient and accurate management of library operations, including book cataloging, order processing, reader management, and user account administration. By performing the above tasks, we can achieve the desired outcome and develop a reliable and effective library system that meets the needs of library users and administrators.

2 DATABASE DESIGN AND DEVELOPMENT

2.1 Development of logical and physical data models

To ensure the effective development of the library system, both logical and physical data models were created. The logical data model, displayed in Figure 1.3, has been visualized using the IDEF1X notation. This model highlights the relationships between the various entities and attributes involved in managing library operations, including the Catalog, Orders, Readers, and User accounts. By utilizing the IDEF1X notation, the logical data model provides a clear and concise representation of the data structure of the library system, enabling efficient management of data and facilitating effective decision-making.

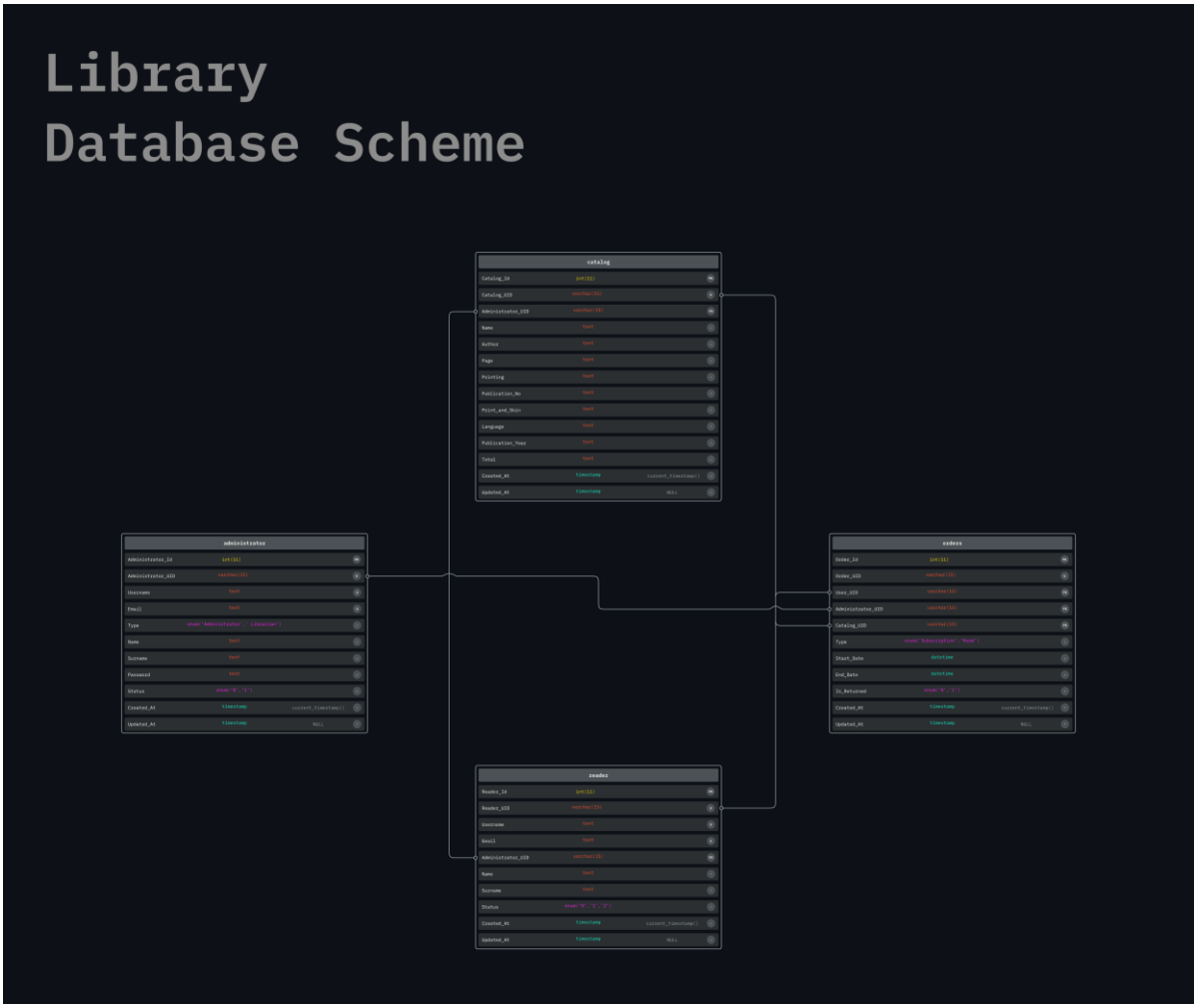

Figure 1.3

The relationship diagram of the database is illustrated in Figure 1.4 below. This diagram depicts the various tables in the database and their relationships, including the primary and foreign keys used to establish connections between the tables. By analyzing this diagram, users can gain a better understanding of how data is organized and stored in the database, and how different tables are interrelated. This can assist users in developing queries and performing various operations on the database.

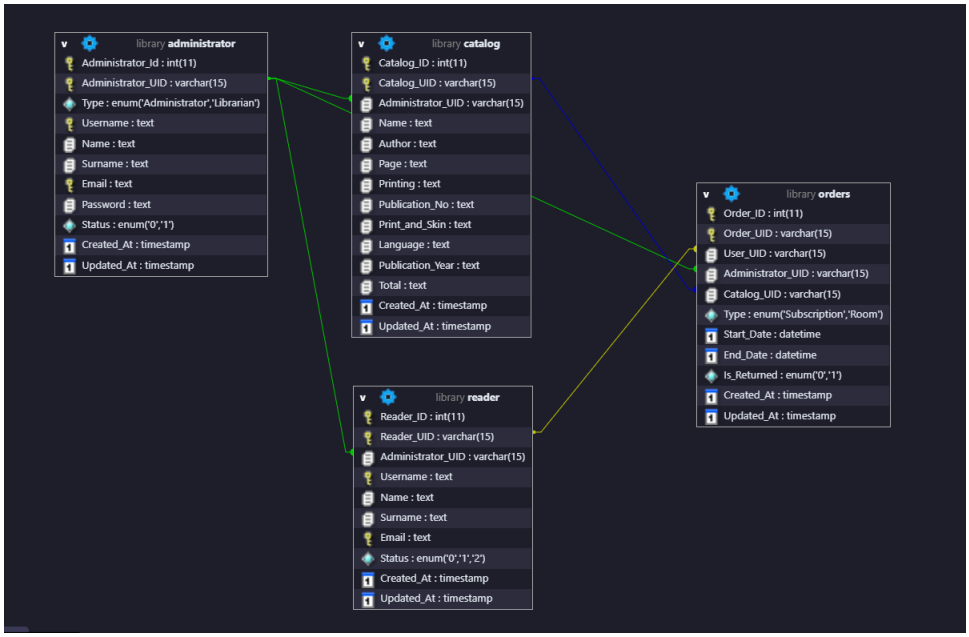

Figure 1.4

2.2 Development of logical and physical data models

The "Administrators" table is intended for storing information on admins. Each record of the table consists of the following fields, the description of which is given in Figure 2.1.

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
Administrator_Id	int(11)			No	*None*	Administrator Id	AUTO_INCREMENT
Administrator_UID	varchar(15)	utf8mb4_general_ci		No	*None*	Administrator Unique Id	
Type	enum('Administrator', 'Librarian')	utf8mb4_general_ci		No	*None*	Administrator Type	
Username	text	utf8mb4_general_ci		No	*None*	Administrator Username	
Name	text	utf8mb4_general_ci		No	*None*	Administrator Name	
Surname	text	utf8mb4_general_ci		No	*None*	Administrator Surname	
Email	text	utf8mb4_general_ci		No	*None*	Administrator Email	
Password	text	utf8mb4_general_ci		No	*None*	Administrator Password	
Status	enum('0', '1')	utf8mb4_general_ci		No	*None*	0 - Inactive 1 - Active	
Created_At	timestamp			No	current_timestamp()		
Updated_At	timestamp			Yes	*NULL*		

Figure 2.1

The "Catalog" table is intended for storing information on Books. Each record of the table consists of the following fields, the description of which is given in Figure 2.2.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Catalog_ID	int(11)			No	*None*	Catalog ID	AUTO_INCREMENT
2	Catalog_UID	varchar(15)	utf8mb4_general_ci		No	*None*	Catalog Unique Id	
3	Administrator_UID	varchar(15)	utf8mb4_general_ci		No	*None*	Book Created by Administrator	
4	Name	text	utf8mb4_general_ci		No	*None*	Book Name	
5	Author	text	utf8mb4_general_ci		No	*None*	Book Author	
6	Page	text	utf8mb4_general_ci		No	*None*	Book Page Count	
7	Printing	text	utf8mb4_general_ci		No	*None*	Book Printing	
8	Publication_No	text	utf8mb4_general_ci		No	*None*	Book Publication Number	
9	Print_and_Skin	text	utf8mb4_general_ci		No	*None*	Book Print & Skin	
10	Language	text	utf8mb4_general_ci		No	*None*	Book Language	
11	Publication_Year	text	utf8mb4_general_ci		No	*None*	Book Publication Year	
12	Total	text	utf8mb4_general_ci		No	*None*	Count of Book	
13	Created_At	timestamp			No	current_timestamp()		
14	Updated_At	timestamp			Yes	*NULL*		

Figure 2.2

The "Orders" table is intended for storing information on Orders taken by Administrators or Librarians from Readers. Each record of the table consists of the following fields, the description of which is given in Figure 2.3.

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
Order_ID	int(11)			No	*None*	Order Id	AUTO_INCREMENT
Order_UID	varchar(15)	utf8mb4_general_ci		No	*None*	Order Unique Id	
User_UID	varchar(15)	utf8mb4_general_ci		No	*None*	Ordered by User	
Administrator_UID	varchar(15)	utf8mb4_general_ci		No	*None*	Created by Admin	
Catalog_UID	varchar(15)	utf8mb4_general_ci		No	*None*	Ordered Book	
Type	enum('Subscription', 'Room')	utf8mb4_general_ci		No	*None*	Subscription or in the reading room	
Start_Date	datetime			No	*None*	Order Start Date (Can be created for Future Date)	
End_Date	datetime			No	*None*	Order Timeout Date	
Is_Returned	enum('0', '1')	utf8mb4_general_ci		No	*None*	Does this book returned by Reader?	
Created_At	timestamp			No	current_timestamp()		
Updated_At	timestamp			Yes	*NULL*		

Figure 2.3

The "Reader" table is intended for storing information on Readers. Each record of the table consists of the following fields, the description of which is given in Figure 2.4.

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
Reader_ID	int(11)			No	*None*	Reader Id	AUTO_INCREMENT
Reader_UID	varchar(15)	utf8mb4_general_ci		No	*None*	Reader Unique Id	
Administrator_UID	varchar(15)	utf8mb4_general_ci		No	*None*	Reader Created by Administrator	
Username	text	utf8mb4_general_ci		No	*None*	Reader Username	
Name	text	utf8mb4_general_ci		No	*None*	Reader Name	
Surname	text	utf8mb4_general_ci		No	*None*	Reader Surname	
Email	text	utf8mb4_general_ci		No	*None*	Reader Email	
Status	enum('0', '1', '2')	utf8mb4_general_ci		No	*None*	0 - Not Subscribed 1 - Subscribed 2 - Blocked	
Created_At	timestamp			No	current_timestamp()		
Updated_At	timestamp			Yes	*NULL*		

Figure 2.4

2.3 Implementation of the database in the MySQL DBMS

The data scheme for the database, which is implemented by MySQL database, is shown in Figure 2.5

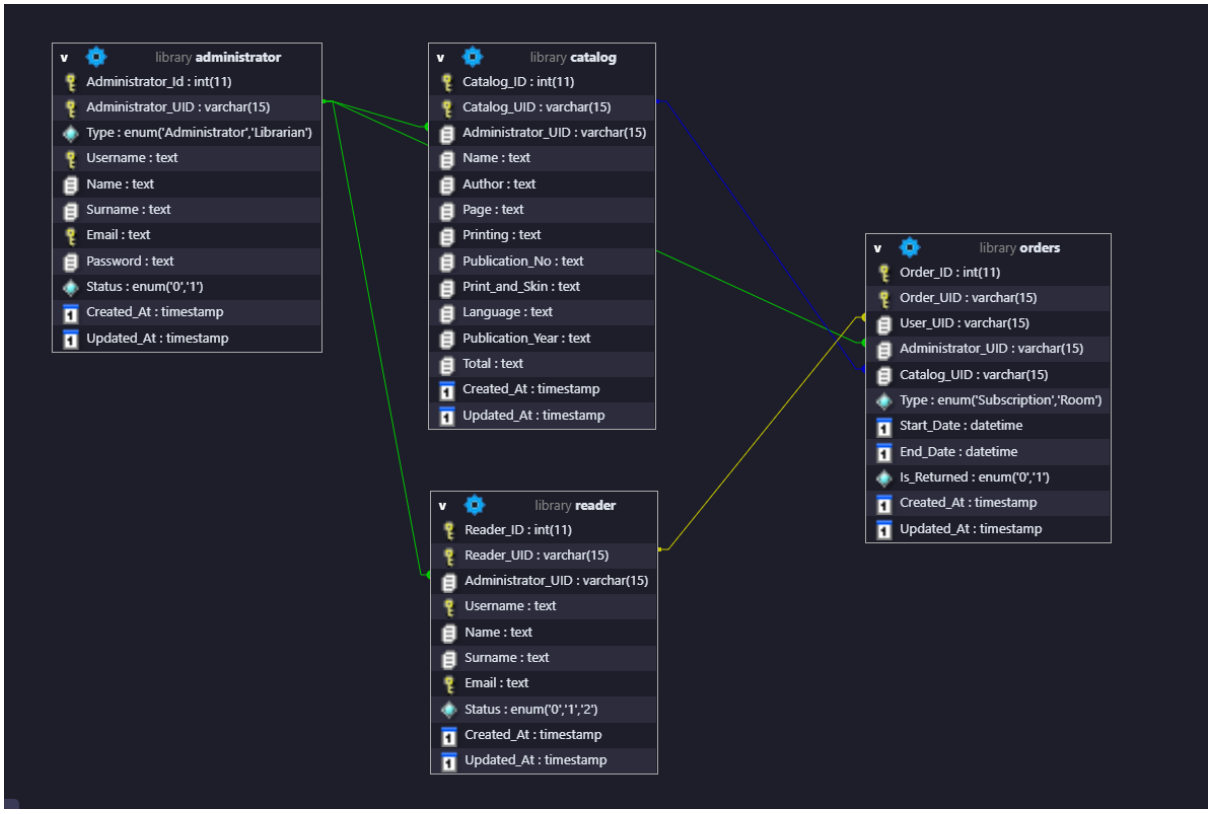

Figure 2.5 – Scheme of the developed database

The following DDL commands were used to create a database in MySQL:

```
CREATE TABLE `administrator` (
  `Administrator_Id` int(11) NOT NULL COMMENT 'Administrator Id',
  `Administrator_UID` varchar(15) NOT NULL COMMENT 'Administrator
Unique Id',
  `Type` enum('Administrator','Librarian') NOT NULL COMMENT
'Administrator Type',
  `Username` text NOT NULL COMMENT 'Administrator Username',
  `Name` text NOT NULL COMMENT 'Administrator Name',
  `Surname` text NOT NULL COMMENT 'Administrator Surname',
  `Email` text NOT NULL COMMENT 'Administrator Email',
  `Password` text NOT NULL COMMENT 'Administrator Password',
  `Status` enum('0','1') NOT NULL COMMENT '0 - Inactive\r\n1 -
Active',
  `Created_At` timestamp NOT NULL DEFAULT current_timestamp(),
  `Updated_At` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;


CREATE TABLE `catalog` (
  `Catalog_ID` int(11) NOT NULL COMMENT 'Catalog ID',
  `Catalog_UID` varchar(15) NOT NULL COMMENT 'Catalog Unique Id',
  `Administrator_UID` varchar(15) NOT NULL COMMENT 'Book Created by
Administrator',
  `Name` text NOT NULL COMMENT 'Book Name',
```

```
  `Author` text NOT NULL COMMENT 'Book Author',
  `Page` text NOT NULL COMMENT 'Book Page Count',
  `Printing` text NOT NULL COMMENT 'Book  Printing',
  `Publication_No` text NOT NULL COMMENT 'Book Publication Number',
  `Print_and_Skin` text NOT NULL COMMENT 'Book Print & Skin',
  `Language` text NOT NULL COMMENT 'Book  Language',
  `Publication_Year` text NOT NULL COMMENT 'Book Publication Year',
  `Total` text NOT NULL COMMENT 'Count of Book',
  `Created_At` timestamp NOT NULL DEFAULT current_timestamp(),
  `Updated_At` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;


CREATE TABLE `orders` (
  `Order_ID` int(11) NOT NULL COMMENT 'Order Id',
  `Order_UID` varchar(15) NOT NULL COMMENT 'Order Unique Id',
  `User_UID` varchar(15) NOT NULL COMMENT 'Ordered by User',
  `Administrator_UID` varchar(15) NOT NULL COMMENT 'Created by Admin',
  `Catalog_UID` varchar(15) NOT NULL COMMENT 'Ordered Book',
  `Type` enum('Subscription','Room') NOT NULL COMMENT 'Subscription
or in the reading\r\nroom',
  `Start_Date` datetime NOT NULL COMMENT 'Order Start Date (Can be
created for Future Date)',
  `End_Date` datetime NOT NULL COMMENT 'Order Timeout Date',
  `Is_Returned` enum('0','1') NOT NULL COMMENT 'Does this book
returned by Reader?',
  `Created_At` timestamp NOT NULL DEFAULT current_timestamp(),
  `Updated_At` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;


CREATE TABLE `reader` (
  `Reader_ID` int(11) NOT NULL COMMENT 'Reader Id',
  `Reader_UID` varchar(15) NOT NULL COMMENT 'Reader Unique Id',
  `Administrator_UID` varchar(15) NOT NULL COMMENT 'Reader Created by
Administrator',
  `Username` text NOT NULL COMMENT 'Reader Username',
  `Name` text NOT NULL COMMENT 'Reader Name',
  `Surname` text NOT NULL COMMENT 'Reader Surname',
  `Email` text NOT NULL COMMENT 'Reader Email',
  `Status` enum('0','1','2') NOT NULL COMMENT '0 - Not Subscribed\r\n1
- Subscribed\r\n2 - Blocked',
  `Created_At` timestamp NOT NULL DEFAULT current_timestamp(),
  `Updated_At` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

2.4 Filling Database with initial records

The following commands allow you to fill in data about Administratos in the created database (Figure 2.6 & Figure 2.7) :

```
INSERT INTO `administrator` (`Administrator_Id`,
`Administrator_UID`, `Type`, `Username`, `Name`, `Surname`, `Email`,
`Password`, `Status`, `Created_At`, `Updated_At`) VALUES
(1, 'F8VVM08yqUk56DP', 'Administrator', '1', 'Ягизджан Евгений',
'Явуз', 'yagizcanyevgenyavuz@gmail.com', '1', '1', '2023-03-12
13:45:18', '2023-03-21 14:23:08'),
(6, 'SJYS23B94LSNINO', 'Librarian', 'librarian', 'librarian',
'librarian', 'librarian', 'librarian', '1', '2023-04-26 07:01:35',
NULL);
```

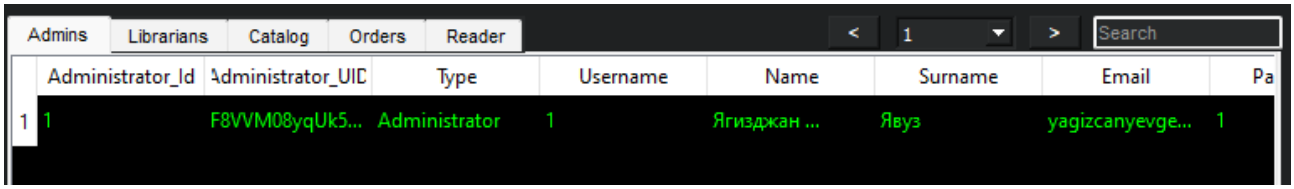

Figure 2.6 - Administrators Table

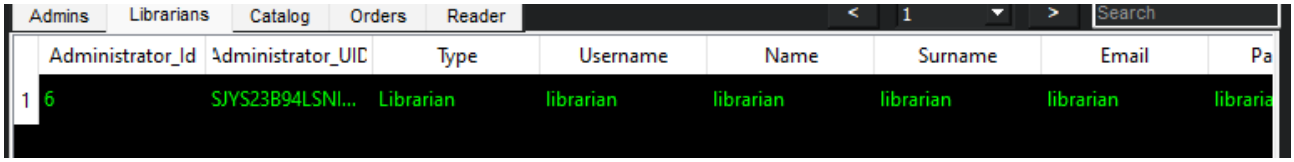

Figure 2.7 - Librarians Table

The following commands allow you to fill in data about Catalog in the created database (Figure 2.8) :

```
INSERT INTO `catalog` (`Catalog_ID`, `Catalog_UID`,
`Administrator_UID`, `Name`, `Author`, `Page`, `Printing`,
`Publication_No`, `Print_and_Skin`, `Language`, `Publication_Year`,
`Total`, `Created_At`, `Updated_At`) VALUES
(2, 'U4COL15RHUZZBIP', 'F8VVM08yqUk56DP', 'test', 'test', 'test',
'test', 'test', 'test', 'test', 'test', '49', '2023-03-21 23:55:57',
'2023-04-26 07:02:26');
```

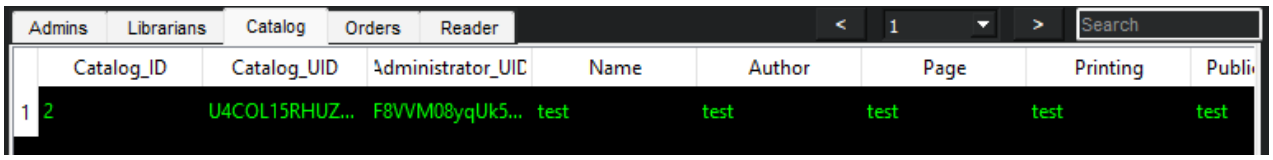

Figure 2.8 - Catalog Table

The following commands allow you to fill in data about Orders in the created database (Figure 2.9) :

```
INSERT INTO `orders` (`Order_ID`, `Order_UID`, `User_UID`,
`Administrator_UID`, `Catalog_UID`, `Type`, `Start_Date`,
`End_Date`, `Is_Returned`, `Created_At`, `Updated_At`) VALUES
(12, 'Y9OXMZJBEG72OKM', '98YWQONHIO73U7W', 'F8VVM08yqUk56DP',
'U4COL15RHUZZBIP', 'Room', '2000-01-01 00:00:00', '2000-01-01
00:00:00', '0', '2023-04-26 07:02:26', NULL);
```

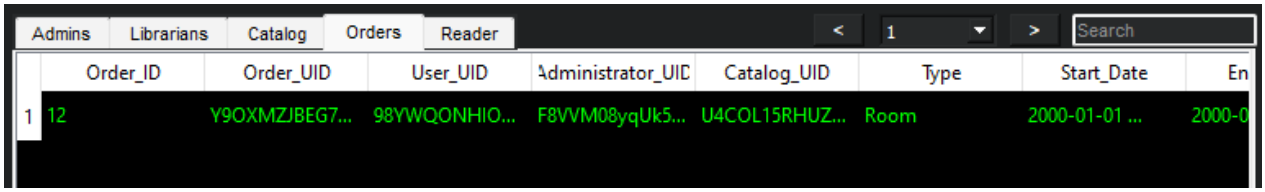

Figure 2.9 - Orders Table

The following commands allow you to fill in data about Readers in the created database (Figure 2.10) :

```
INSERT INTO `reader` (`Reader_ID`, `Reader_UID`,
`Administrator_UID`, `Username`, `Name`, `Surname`, `Email`,
`Status`, `Created_At`, `Updated_At`) VALUES
(3, '98YWQONHIO73U7W', 'F8VVM08yqUk56DP', 'test', 'test', 'test',
'test', '2', '2023-04-26 07:01:04', '2023-04-26 07:03:22');
```

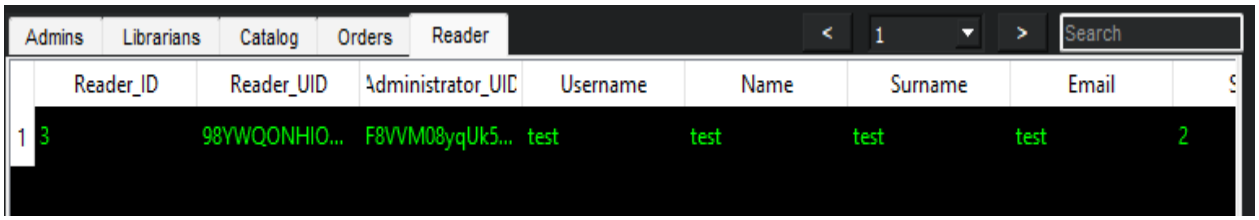

Figure 2.10 - Readers Table

3.Used programs

In the development of the library system project, the following programs were utilized:

- XAMPP - XAMPP is a free and open-source cross-platform web server solution stack that includes the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming

languages. In the project, XAMPP was used to host the MySQL database, which was used to store and manage data related to books, orders, readers, and user accounts.

- PyCharm - PyCharm is an integrated development environment (IDE) used for computer programming in Python. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems, and supports web development with Django. PyCharm was used in the development of the project for writing, testing, and debugging the Python code.
- MySQL - MySQL is a free and open-source relational database management system that uses SQL for querying and managing data. In the project, MySQL was used as the database management system for the library system, allowing for the acquisition and storage of data related to library operations.
- PyQt5 - PyQt5 is a Python binding of the cross-platform GUI toolkit Qt, which is used for creating desktop applications with graphical user interfaces. PyQt5 was used in the development of the project to create a user-friendly interface for managing library operations, providing a visual appearance and intuitive user experience for users of the library system.

4.CONCLUSION

In conclusion, the development of a library management system involved a thorough analysis of the subject domain, the creation of logical and physical data models, and the implementation of a database using MySQL and Python programming language. The resulting application allows users to efficiently manage various library operations, such as adding new books, creating orders, managing user accounts, and generating reports. Through the use of XAMPP, PyCharm, and PyQt5, the application provides a user-friendly interface with easy navigation and a comprehensive set of instructions. Overall, the library management system presented in this report can enhance the efficiency and effectiveness of library operations, ultimately providing a better user experience for both librarians and administrators.