BOOK BANK MANAGEMENT SYSTEM

COURSE PROJECT REPORT

18CSC303J – DATABASE MANAGEMENT SYSTEMS

III Year/ VI Semester

Academic Year: 2023 -2024 (EVEN)

By

SAI KALYAN RAJU (RA2111026010330)

Under the guidance of

Dr. M Dhilsath Fathima

Assistant Professor

Department of Computational Intelligence

DEPARTMENT OF COMPUTATIONAL INTELLIGENCE

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Kancheepuram

APRIL 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

BONAFIDE CERTIFICATE

Certified that this project report titled "**BOOK BANK MANAGEMENT SYSTEM**" is a bona fide work done by **SAI KALYAN RAJU (RA2111026010330)** who carried out the project work under my supervision.Certified further that to the best of my knowledge the work reported herein does not form part of any otherproject work or dissertation.

SIGNATURE
Faculty In-Charge
Dr. M.Dhilsath Fathima
Assistant Professor
Department of Computational Intelligence
SRM Institute of Science and Technology
Kattankulathur

SIGNATURE
Head of Department
Dr. R. Annie Uthra
Professor and Head
Department of Computational Intelligence
SRM Institute of Science and Technology
Kattankulathur

ABSTRACT

The Book Bank Management System represents a sophisticated software solution meticulously crafted to meet the diverse needs of book banks and libraries. It embodies a suite of features meticulously designed to streamline library operations, optimize resource utilization, and enhance user experience. At its core, the system offers robust user management functionalities, enabling seamless registration, authentication, and profile management for library patrons. Concurrently, administrators are empowered with extensive catalog management capabilities, facilitating the efficient organization, classification, and tracking of library resources.

Moreover, the system seamlessly orchestrates the intricate processes of book borrowing and return, leveraging automation to simplify transactional workflows and enhance operational efficiency. Furthermore, it incorporates a sophisticated reservation system, enabling patrons to reserve coveted resources and administrators to effectively manage resource allocation.

In addition, the system integrates fine management mechanisms to handle overdue books with fines, ensuring adherence to library policies and fostering responsibility among users. Furthermore, the system boasts advanced reporting and analytics tools, empowering administrators with actionable insights into library usage patterns, circulation trends, and resource popularity.

Finally, the system stands out for its seamless integration capabilities, facilitating interoperability with existing library systems, such as RFID tagging, barcode scanning, and online databases, thereby ensuring a cohesive and streamlined library ecosystem. Overall, the Book Bank Management System serves as a catalyst for innovation, efficiency, and excellence in library management, ushering in a new era of accessibility, accountability, and user satisfaction in the realm of knowledge dissemination and acquisition.

INDEX

CONTENTS

LIST OF FIGURES

CONTENTS

CHAPTER I

INTRODUCTION

The Book Bank Management System represents a pivotal advancement in the realm of library management, aimed at revolutionizing the way libraries and book banks operate. In an era defined by digital transformation and technological innovation, the traditional paradigms of library management are being redefined, with a growing emphasis on leveraging technology to enhance accessibility, efficiency, and user experience. The Book Bank Management System emerges as a response to this evolving landscape, offering a comprehensive suite of features and functionalities designed to meet the diverse needs of modern libraries and book banks.

At its essence, the Book Bank Management System serves as a centralized platform for the systematic organization, cataloging, and dissemination of library resources. It embodies a user-centric approach, prioritizing the needs and preferences of library patrons while also addressing the operational requirements of library administrators. By integrating cutting-edge technologies, intuitive user interfaces, and robust administrative tools, the system aims to streamline library operations, optimize resource utilization, and foster a culture of lifelong learning and knowledge sharing within the community.

The significance of the Book Bank Management System lies in its ability to transcend the limitations of traditional library management practices. Gone are the days of manual cataloging, cumbersome transactional processes, and disjointed resource management. Instead, the system offers a seamless and cohesive solution that brings together users, resources, and administrators in a unified digital ecosystem.

This introduction serves as a prelude to a detailed exploration of the Book Bank Management System, delving into its key features, functionalities, and benefits. Through a comprehensive examination of its components and capabilities, this document aims to provide insights into the transformative potential of the system and its implications for the future of library management.

1.1Software

VS CODE (Frontend)

Visual Studio Code, often abbreviated as VS Code, is a popular and versatile source-code editor developed by Microsoft. Launched in 2015, it quickly gained traction among developers due to its lightweight yet powerful features, extensive customization options, and wide support for various programming languages and frameworks. Visual Studio Code has emerged as a go-to choose for developers worldwide, offering a powerful and customizable development environment that caters to a wide range of programming languages and workflows. Its simplicity, extensibility, and robust feature set make it a favourite among developers for building applications of all types and sizes.

Database – MySQL

The back-end database used in this project is **MySQL.**

It is a language used to interrogate and process data in a relational database. Originally developed by IBM for its mainframes, SQL commands can be used to interactively work with a database or can be embedded within a script or programming language as an interface to a database. Programming extensions to SQL have turned it into a full-blown database programming language, and all major database management systems (DBMSs) support it.

ANSI standardized SQL. But most DBMSs have some proprietary enhancement, which if used, makes SQL non- standard. Moving an application from one SQL database to another sometimes requires tweaking, the age-old problem in this business!

1.2 Advantages of MySQL

1. SQL Queries can be used to retrieve large amounts of records from a database quickly.
2. SQL is used to view the data without storing the data into the object
3. SQL joins two or more tables and show it as one object to user
4. SQL databases use long-established standard, which is being adopted by ANSI &amp; ISO. Non-SQL databases do not adhere to any clear standard.
5. MySQL can handle large volumes of data and scale as your ecommerce business grows.
6. As an open-source database, MySQL is cost-effective, making it accessible to businesses of all sizes.

CHAPTER II
PROJECT FEATURES AND OBJECTIVES

2.1 MAIN FEATURES AND FUNCTIONALITY

1. User Management
2. Book Management
3. Transaction Management
4. Admin Panel
5. Reporting and Analytics
6. Security
7. Integration and Scalability

2.2 OBJECTIVES

1. Efficient Book Management
2. User-Friendly Interface
3. Secure User Authentication
4. Effective Transaction Management
5. Comprehensive Reporting
6. Scalability and Flexibility
7. High Data Security and Integrity

2.3 IDENTIFICATION OF PROJECT MODULES

In our Book Bank Management System project, we organize the functionality into separate modules, allowing for better organization, maintenance, and scalability. Each module focuses on a specific aspect of the system, ensuring clarity and efficiency in development and execution. Here are the modules included in our project:

1. Admin Management
2. User Management
3. Book Catalog
4. Transaction Handling
5. Reporting and Analytics

2.4MODULE DESCRIPTION

2.4.1 Admin Management

- This module focuses on administrative tasks such as user management, book management, transaction monitoring, and system configuration. Administrators have privileged access to perform these tasks and oversee the overall operation of the system.

2.4.2 User Management

- Responsible for user authentication, registration, profile management, and access control. It ensures secure and seamless user interactions with the system, allowing users to borrow, reserve, and return books conveniently.

2.4.3 Book Catalog

- Manages the book inventory, including adding, updating, and deleting book records. This module categorizes books by genre, author, publication date, etc., making it easier for users to search and browse the available collection.

2.4.4 Transaction Handling

- Facilitates book borrowing, reservation, and return processes. It tracks transaction history, overdue books, fines, and notifications to users. This module ensures smooth and efficient management of book transactions within the system.

2.4.5 Reporting and Analytics

- U Generates reports and statistics on library usage patterns, book borrowing trends, popular genres, and user feedback. It provides valuable insights to administrators for decision-making, resource allocation, and system optimization.

CHAPTER III
BACK-END DESIGN, FRONT-END DESIGN

3.1 BACK-END DESIGN

Backend design is essential for building robust and efficient web applications. It involves defining clear requirements, selecting the appropriate technology stack, and architecting the system for scalability, performance, and security. With the right tools and methodologies, backend design lays the foundation for a stable and successful web application. Backend design requires careful consideration of various factors to ensure the reliability, scalability, security, and maintainability of the application. By following a structured approach and leveraging appropriate technologies and best practices, you can build a robust backend that meets your project requirements.

3.1.1 Conceptual Database Design (ER-Diagram)

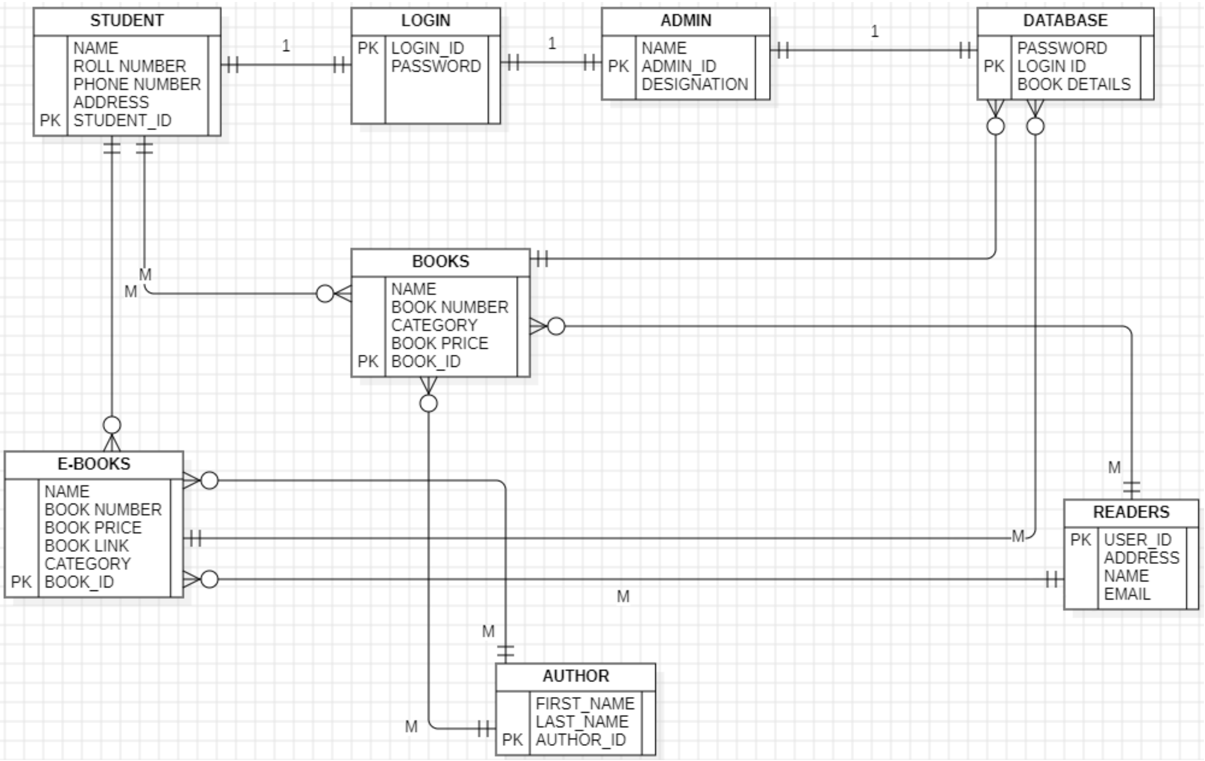


FIGURE 3.1.1 – ER DIAGRAM

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

Creating an Entity-Relationship (ER) diagram for a Book Bank Management System involves identifying the entities and their relationships with each other. Here's a ER diagram for the Book Bank Management System

3.1.2 Logical Database Design (ER Mapping)

SCHEMA DIAGRAM

ADMIN

Admin_Id	Name	Designation

AUTHOR

Author_Id	First_Name	Last_Name	Book_Id

BOOKS

Book_Id	Name	Book_Number	Category	Book_Price

EBOOKS

Book_Id	Name	Book_Number	Category	Book_Price	Book_Link

READERS

User_Id	Address	Name	Email

STUDENT

Student_Id	Name	Roll_Number	Phone_No	Address

- The entities are represented as tables.
- The tables contain the attributes.
- The attributes which are underlined are referred as primary keys.

3.2 FRONT-END DESIGN

3.2.1 Front-end web development details

- **HTML** provides the basic structure of sites, which is enhanced and modified by other technologies like CSS and JavaScript.
- **CSS** is used to control presentation, formatting, and layout.
- **JavaScript** is used to control the behaviour of different elements.

HTML

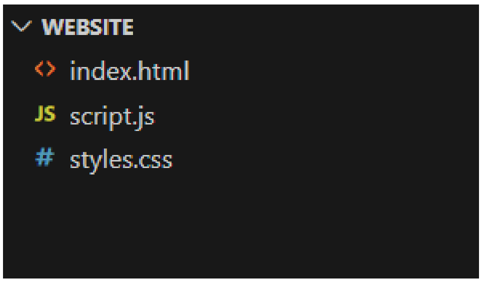
HTML is at the core of every web page, regardless the complexity of a site or number of technologies involved. It's an essential skill for any web professional. It's the starting point for anyone learning how to create content for the web. And, luckily for us, it's surprisingly easy to learn.

CSS

CSS stands for Cascading Style Sheets. This programming language dictates how the HTML elements of a website should actually appear on the frontend of the page.

JavaScript

JavaScript is a more complicated language than HTML or CSS, and it wasn't released in beta form until 1995. Nowadays, JavaScript is supported by all modern web browsers and is used on almost every site on the web for more powerful and complex functionality. In this project, JavaScript is mainly used for connecting the main web page with the order details page.

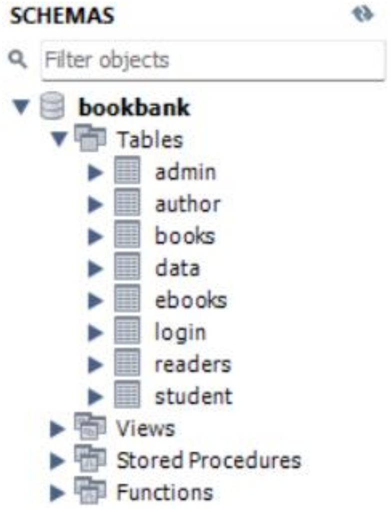


```
WEBSITE
  index.html
  script.js
  styles.css
```

FIGURE 3.2.1 – FILES LIST

CHAPTER IV
MODULE IMPLEMENTATION

4.1 CONSTRUCTION OF RELATIONAL TABLE FROM THE ER DIAGRAM



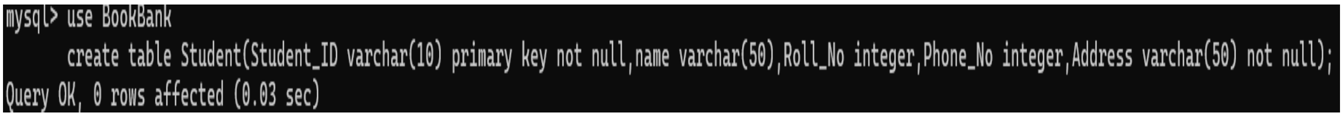
```
SCHEMAS
Filter objects
▼ bookbank
  ▼ Tables
    ▶ admin
    ▶ author
    ▶ books
    ▶ data
    ▶ ebooks
    ▶ login
    ▶ readers
    ▶ student
  ▶ Views
  ▶ Stored Procedures
  ▶ Functions
```

FIGURE 4.1 – TABLES LIST

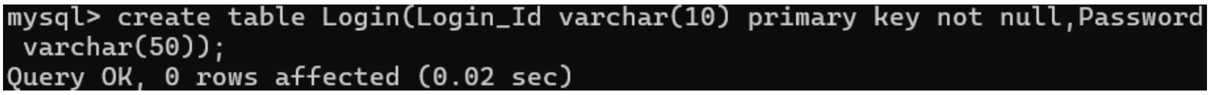
4.1.1 DDL, DML, DCL, TCL of Book Bank Management System

DDL Commands:

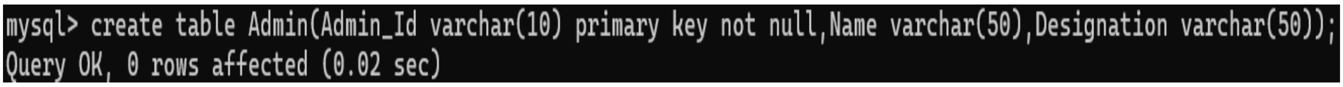
DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.



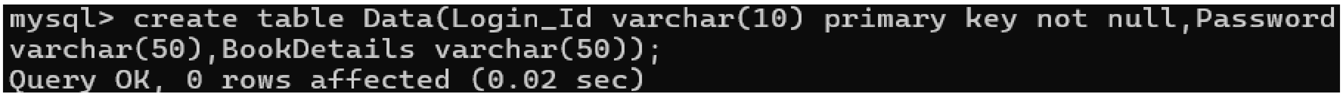
```
mysql> use BookBank
     create table Student(Student_ID varchar(10) primary key not null,name varchar(50),Roll_No integer,Phone_No integer,Address varchar(50) not null);
Query OK, 0 rows affected (0.03 sec)
```



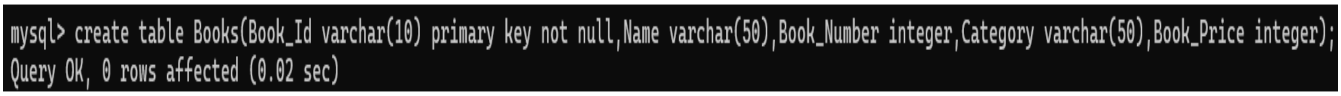
```
mysql> create table Login(Login_Id varchar(10) primary key not null,Password
 varchar(50));
Query OK, 0 rows affected (0.02 sec)
```



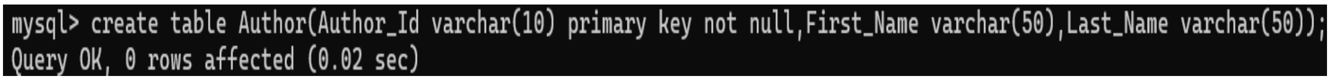
```
mysql> create table Admin(Admin_Id varchar(10) primary key not null,Name varchar(50),Designation varchar(50));
Query OK, 0 rows affected (0.02 sec)
```



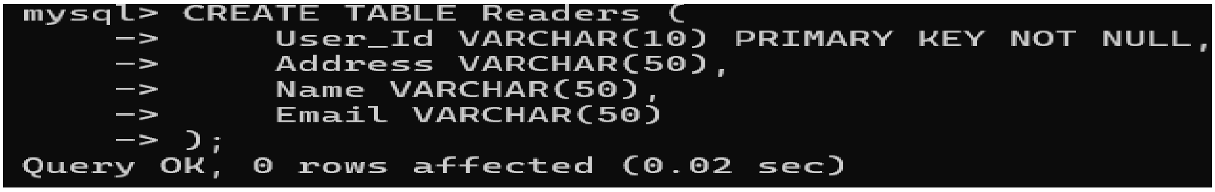
```
mysql> create table Data(Login_Id varchar(10) primary key not null,Password
varchar(50),BookDetails varchar(50));
Query OK, 0 rows affected (0.02 sec)
```



```
mysql> create table Books(Book_Id varchar(10) primary key not null,Name varchar(50),Book_Number integer,Category varchar(50),Book_Price integer);
Query OK, 0 rows affected (0.02 sec)
```



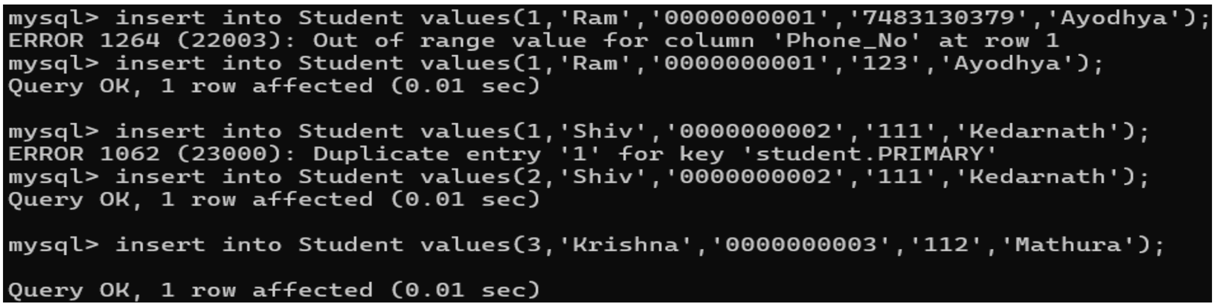
```
mysql> create table Author(Author_Id varchar(10) primary key not null,First_Name varchar(50),Last_Name varchar(50));
Query OK, 0 rows affected (0.02 sec)
```



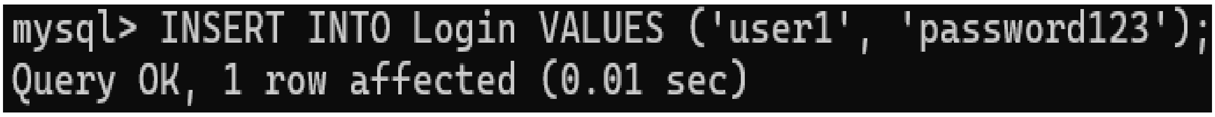
```
mysql> CREATE TABLE Readers (
    ->     User_Id VARCHAR(10) PRIMARY KEY NOT NULL,
    ->     Address VARCHAR(50),
    ->     Name VARCHAR(50),
    ->     Email VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.02 sec)
```

DML Commands:

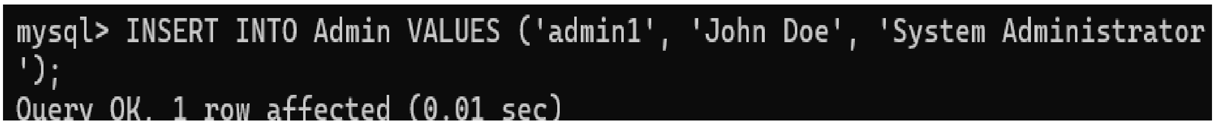
The SQL commands that deal with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements. It is the component of the SQL statement that controls access to data and to the database.



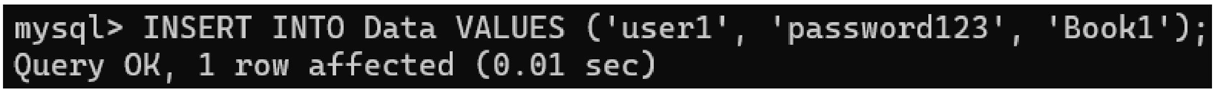
```
mysql> insert into Student values(1,'Ram','0000000001','7483130379','Ayodhya');
ERROR 1264 (22003): Out of range value for column 'Phone_No' at row 1
mysql> insert into Student values(1,'Ram','0000000001','123','Ayodhya');
Query OK, 1 row affected (0.01 sec)

mysql> insert into Student values(1,'Shiv','0000000002','111','Kedarnath');
ERROR 1062 (23000): Duplicate entry '1' for key 'student.PRIMARY'
mysql> insert into Student values(2,'Shiv','0000000002','111','Kedarnath');
Query OK, 1 row affected (0.01 sec)

mysql> insert into Student values(3,'Krishna','0000000003','112','Mathura');

Query OK, 1 row affected (0.01 sec)
```



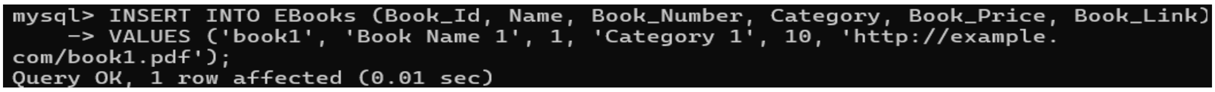
```
mysql> INSERT INTO Login VALUES ('user1', 'password123');
Query OK, 1 row affected (0.01 sec)
```



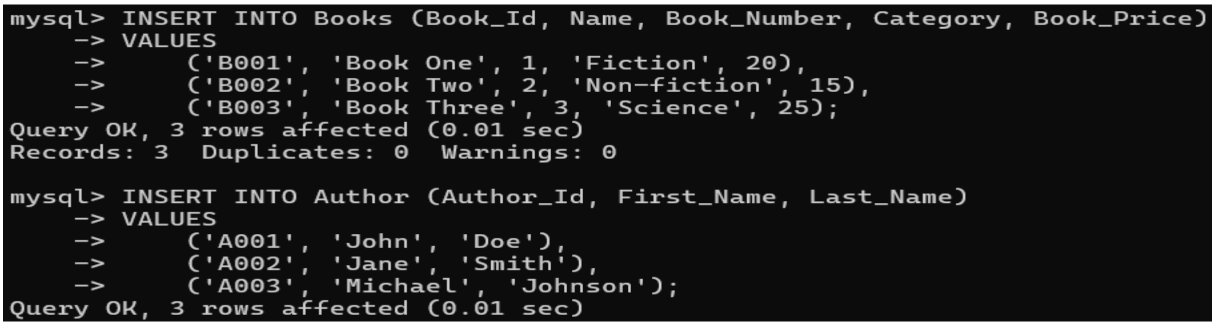
```
mysql> INSERT INTO Admin VALUES ('admin1', 'John Doe', 'System Administrator
');
Query OK, 1 row affected (0.01 sec)
```



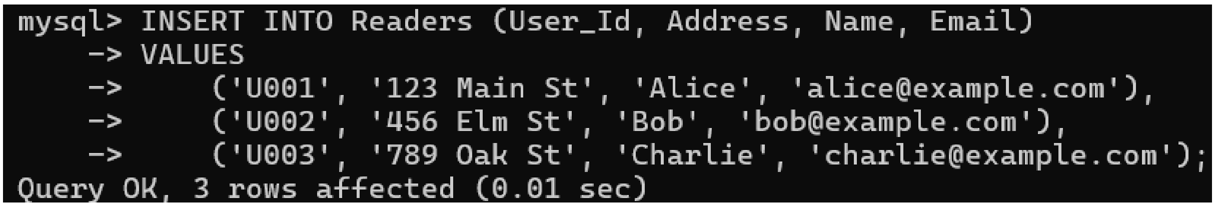
```
mysql> INSERT INTO Data VALUES ('user1', 'password123', 'Book1');
Query OK, 1 row affected (0.01 sec)
```



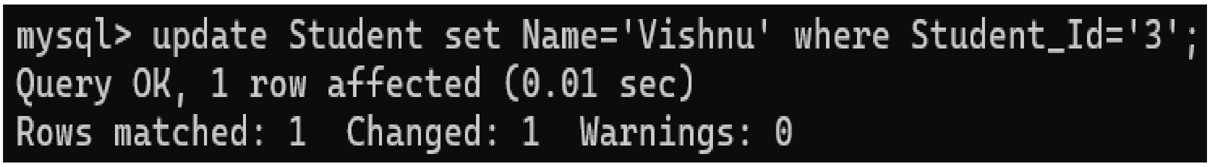
```
mysql> INSERT INTO EBooks (Book_Id, Name, Book_Number, Category, Book_Price, Book_Link)
    -> VALUES ('book1', 'Book Name 1', 1, 'Category 1', 10, 'http://example.
com/book1.pdf');
Query OK, 1 row affected (0.01 sec)
```



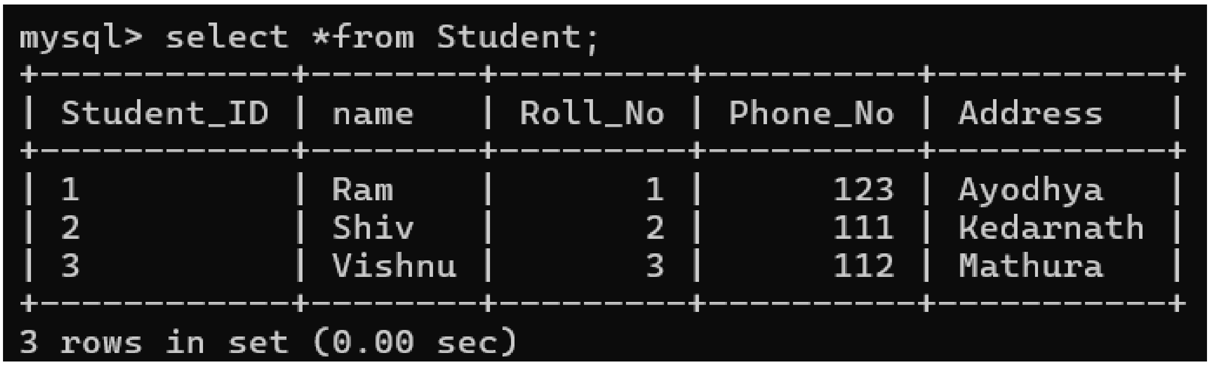
```
mysql> INSERT INTO Books (Book_Id, Name, Book_Number, Category, Book_Price)
    -> VALUES
    ->     ('B001', 'Book One', 1, 'Fiction', 20),
    ->     ('B002', 'Book Two', 2, 'Non-fiction', 15),
    ->     ('B003', 'Book Three', 3, 'Science', 25);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Author (Author_Id, First_Name, Last_Name)
    -> VALUES
    ->     ('A001', 'John', 'Doe'),
    ->     ('A002', 'Jane', 'Smith'),
    ->     ('A003', 'Michael', 'Johnson');
Query OK, 3 rows affected (0.01 sec)
```



```
mysql> INSERT INTO Readers (User_Id, Address, Name, Email)
    -> VALUES
    ->     ('U001', '123 Main St', 'Alice', 'alice@example.com'),
    ->     ('U002', '456 Elm St', 'Bob', 'bob@example.com'),
    ->     ('U003', '789 Oak St', 'Charlie', 'charlie@example.com');
Query OK, 3 rows affected (0.01 sec)
```



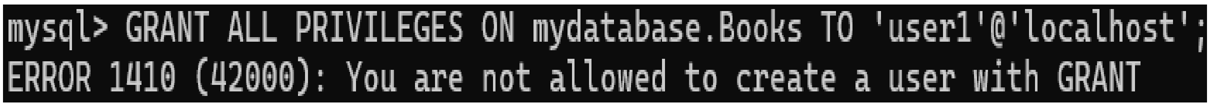
```
mysql> update Student set Name='Vishnu' where Student_Id='3';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```



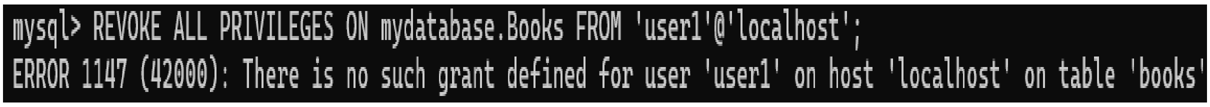
```
mysql> select *from Student;
+------------+--------+---------+----------+-----------+
| Student_ID | name   | Roll_No | Phone_No | Address   |
+------------+--------+---------+----------+-----------+
| 1          | Ram    |       1 |      123 | Ayodhya   |
| 2          | Shiv   |       2 |      111 | Kedarnath |
| 3          | Vishnu |       3 |      112 | Mathura   |
+------------+--------+---------+----------+-----------+
3 rows in set (0.00 sec)
```

DCL Commands:

DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.



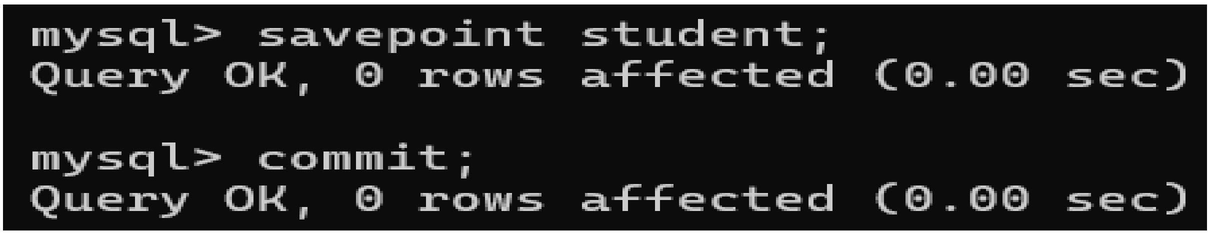
```
mysql> GRANT ALL PRIVILEGES ON mydatabase.Books TO 'user1'@'localhost';
ERROR 1410 (42000): You are not allowed to create a user with GRANT
```



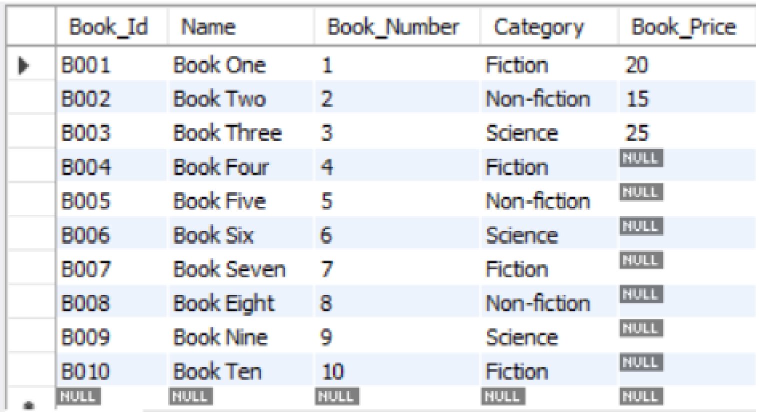
```
mysql> REVOKE ALL PRIVILEGES ON mydatabase.Books FROM 'user1'@'localhost';
ERROR 1147 (42000): There is no such grant defined for user 'user1' on host 'localhost' on table 'books'
```

TCL Commands:

Transactions group a set of tasks into a single execution unit. Each transaction begins with a specific task and ends when all the tasks in the group are successfully completed.



```
mysql> savepoint student;
Query OK, 0 rows affected (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)
```



	Book_Id	Name	Book_Number	Category	Book_Price
▶	B001	Book One	1	Fiction	20
	B002	Book Two	2	Non-fiction	15
	B003	Book Three	3	Science	25
	B004	Book Four	4	Fiction	NULL
	B005	Book Five	5	Non-fiction	NULL
	B006	Book Six	6	Science	NULL
	B007	Book Seven	7	Fiction	NULL
	B008	Book Eight	8	Non-fiction	NULL
	B009	Book Nine	9	Science	NULL
	B010	Book Ten	10	Fiction	NULL
*	NULL	NULL	NULL	NULL	NULL

Figure 4.1.1 – BOOKS TABLE

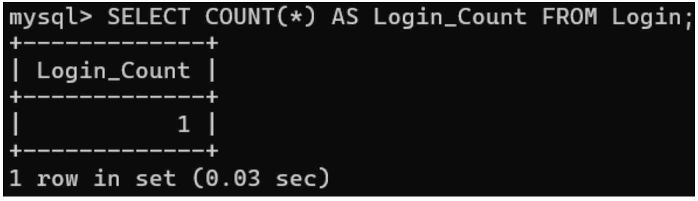
4.1.2 In- Built functions of Book Bank Management System

Built-in functions in SQL are predefined operations that can be applied to data within a database. They encompass a variety of functionalities, including manipulating strings, performing mathematical calculations, summarizing data, and working with date and time values.

MYSQL has several built-in functions that can be used to manipulate data.

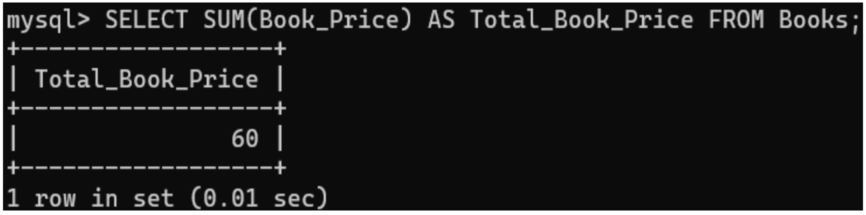
- Aggregate Functions: Perform calculations on sets of values.
- String Functions: Manipulate string values.
- Date Functions: Work with date and time values.
- Numeric Functions: Operate on numeric values.
- Conditional Functions: Control the flow of data based on conditions.

Count the number of login Id:



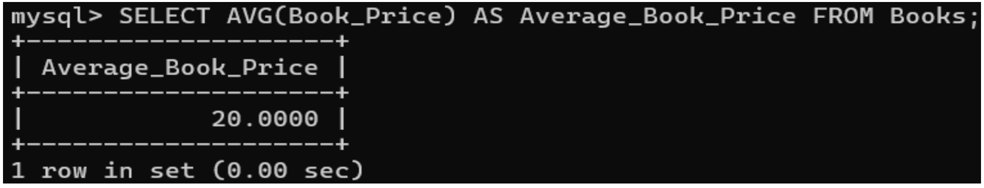
```
mysql> SELECT COUNT(*) AS Login_Count FROM Login;
+-------------+
| Login_Count |
+-------------+
|           1 |
+-------------+
1 row in set (0.03 sec)
```

Calculate the total sum of all book prices:



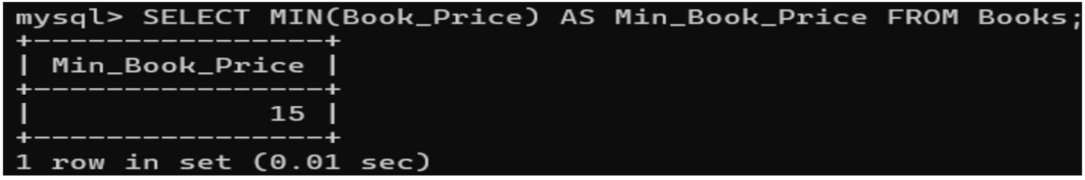
```
mysql> SELECT SUM(Book_Price) AS Total_Book_Price FROM Books;
+------------------+
| Total_Book_Price |
+------------------+
|               60 |
+------------------+
1 row in set (0.01 sec)
```

Calculate the average book price from the Books table:

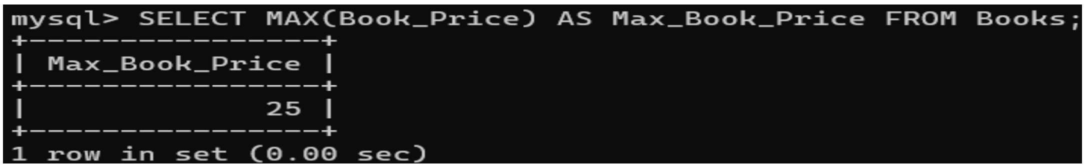


```
mysql> SELECT AVG(Book_Price) AS Average_Book_Price FROM Books;
+--------------------+
| Average_Book_Price |
+--------------------+
|            20.0000 |
+--------------------+
1 row in set (0.00 sec)
```

To find the minimum and maximum book prices from the Books table:



```
mysql> SELECT MIN(Book_Price) AS Min_Book_Price FROM Books;
+----------------+
| Min_Book_Price |
+----------------+
|             15 |
+----------------+
1 row in set (0.01 sec)
```



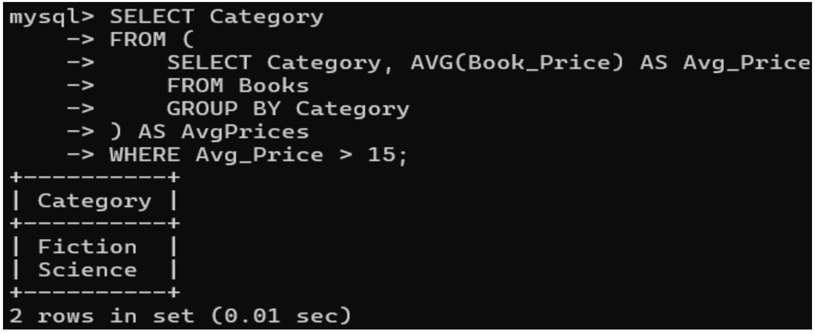
```
mysql> SELECT MAX(Book_Price) AS Max_Book_Price FROM Books;
+----------------+
| Max_Book_Price |
+----------------+
|             25 |
+----------------+
1 row in set (0.00 sec)
```

Figure 4.1.2 – INBUILT FUNCTIONS

4.1.3 Nested Queries of Book Bank Management System

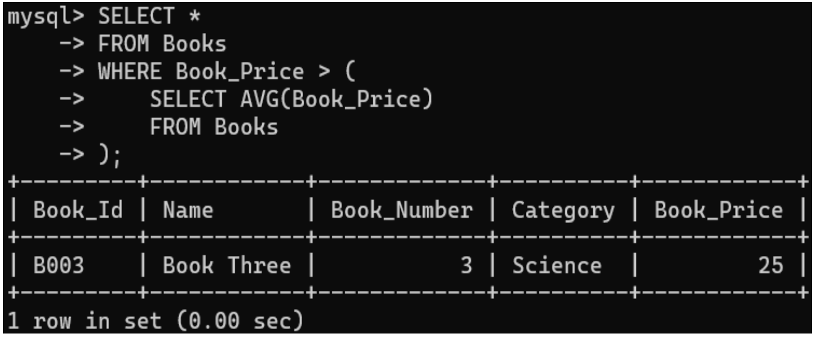
A MySQL subquery is a query that is enclosed within another query, such as SELECT, INSERT, UPDATE, or DELETE. These subqueries can themselves contain additional subqueries, creating nested levels of queries. The query containing the subquery is referred to as the outer query, while the subquery itself is known as the inner query. This nesting allows for complex and flexible data retrieval and manipulation within MySQL databases.

- **Find the categories with an average price higher than a certain threshold (let's say $15):**



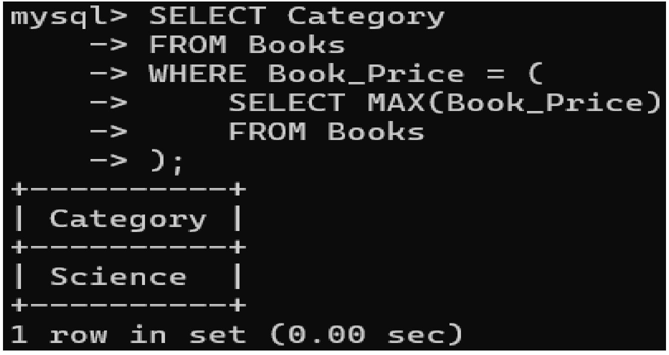
```
mysql> SELECT Category
    -> FROM (
    ->     SELECT Category, AVG(Book_Price) AS Avg_Price
    ->     FROM Books
    ->     GROUP BY Category
    -> ) AS AvgPrices
    -> WHERE Avg_Price > 15;
+----------+
| Category |
+----------+
| Fiction  |
| Science  |
+----------+
2 rows in set (0.01 sec)
```

- **Find books with prices higher than the average book price:**



```
mysql> SELECT *
    -> FROM Books
    -> WHERE Book_Price > (
    ->     SELECT AVG(Book_Price)
    ->     FROM Books
    -> );
+---------+------------+-------------+----------+------------+
| Book_Id | Name       | Book_Number | Category | Book_Price |
+---------+------------+-------------+----------+------------+
| B003    | Book Three |           3 | Science  |         25 |
+---------+------------+-------------+----------+------------+
1 row in set (0.00 sec)
```

- **Find the category with the most expensive book:**



```
mysql> SELECT Category
    -> FROM Books
    -> WHERE Book_Price = (
    ->     SELECT MAX(Book_Price)
    ->     FROM Books
    -> );
+----------+
| Category |
+----------+
| Science  |
+----------+
1 row in set (0.00 sec)
```

Figure 4.1.3 – NESTED QUERIES

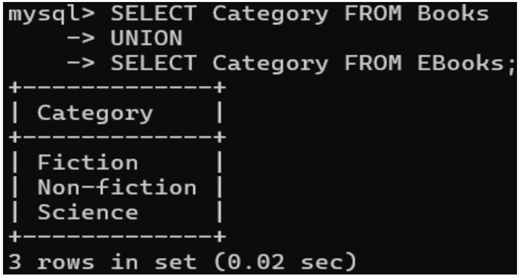
4.1.4 Set Operators & Views of Book Bank Management System

The SET operators in MySQL are utilized to merge the results of multiple SELECT statements and present them as a unified result set. There are four main types of set operators in SQL:

- **UNION:** Merges two or more result sets into a single set, excluding duplicates.
- **UNION ALL:** Merges two or more result sets into a single set, including duplicates.

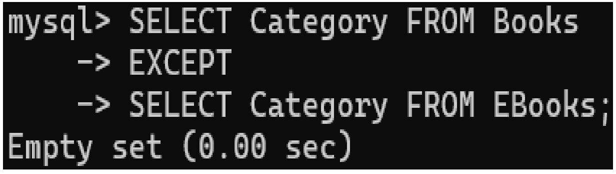
- **INTERSECT:** Combines two result sets and outputs the data that is common to both result sets.
- **EXCEPT:** Combines two result sets and retrieves the data from the first result set that is not present in the second result set.

- **Using UNION to show all the books:**



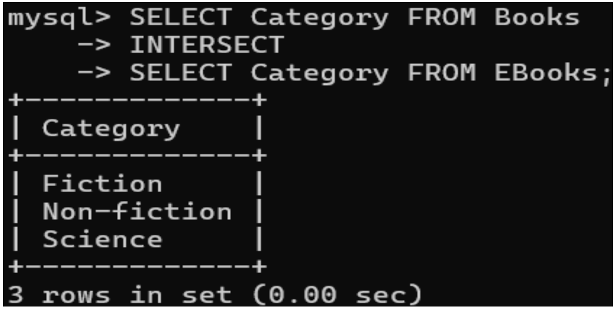
```
mysql> SELECT Category FROM Books
    -> UNION
    -> SELECT Category FROM EBooks;
+-------------+
| Category    |
+-------------+
| Fiction     |
| Non-fiction |
| Science     |
+-------------+
3 rows in set (0.02 sec)
```

- **Find categories in Books but not in E-Books**



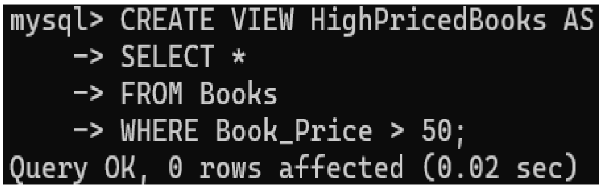
```
mysql> SELECT Category FROM Books
    -> EXCEPT
    -> SELECT Category FROM EBooks;
Empty set (0.00 sec)
```

- **Find common categories between Books and E-Books tables:**



```
mysql> SELECT Category FROM Books
    -> INTERSECT
    -> SELECT Category FROM EBooks;
+-------------+
| Category    |
+-------------+
| Fiction     |
| Non-fiction |
| Science     |
+-------------+
3 rows in set (0.00 sec)
```

Views are virtual tables that are defined by a SELECT query. They can simplify complex queries and provide a layer of abstraction over your data.



```
mysql> CREATE VIEW HighPricedBooks AS
    -> SELECT *
    -> FROM Books
    -> WHERE Book_Price > 50;
Query OK, 0 rows affected (0.02 sec)
```

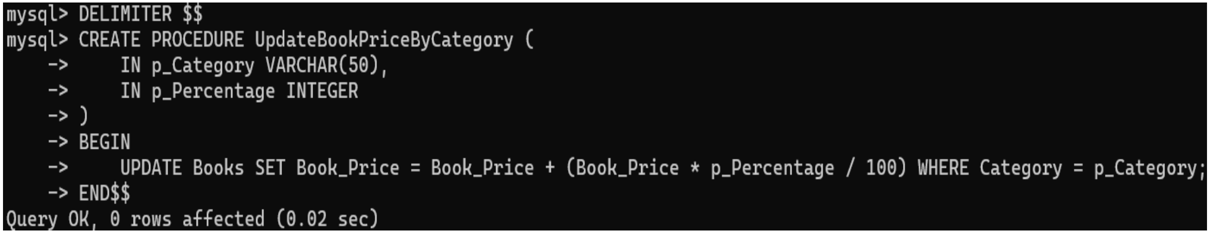
Figure 4.1.4 – SET OPERATORS AND VIEWS

4.1.5 PL/SQL Procedures and Functions of Book Bank Management System

PL/SQL subprograms are categorized as named PL/SQL blocks that can be called with a specified set of parameters. There are two types of PL/SQL subprograms:

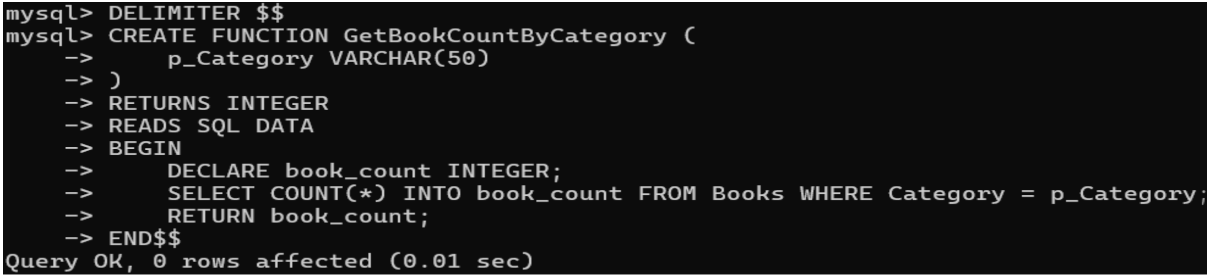
- **Functions:** These subprograms return a single value and are primarily utilized to compute and deliver a result.
- **Procedures:** These subprograms do not directly return a value; they are mainly employed to execute a specific action or set of actions.

- **Procedure to Update Book Price by Category:**



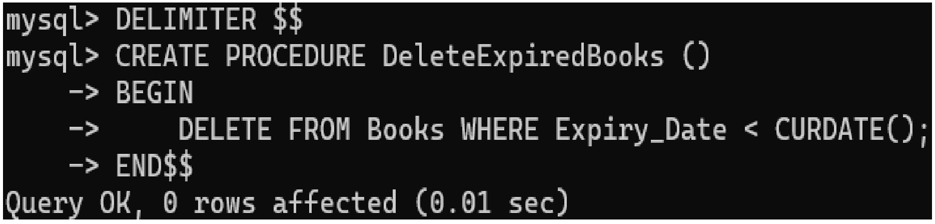
```
mysql> DELIMITER $$
mysql> CREATE PROCEDURE UpdateBookPriceByCategory (
    ->     IN p_Category VARCHAR(50),
    ->     IN p_Percentage INTEGER
    -> )
    -> BEGIN
    ->     UPDATE Books SET Book_Price = Book_Price + (Book_Price * p_Percentage / 100) WHERE Category = p_Category;
    -> END$$
Query OK, 0 rows affected (0.02 sec)
```

- **Function to Retrieve Book Count by Category:**



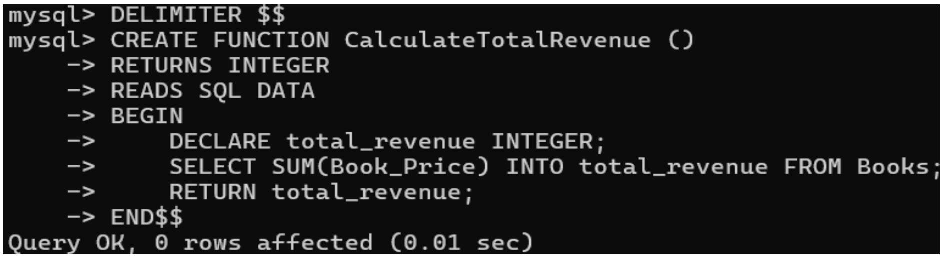
```
mysql> DELIMITER $$
mysql> CREATE FUNCTION GetBookCountByCategory (
    ->     p_Category VARCHAR(50)
    -> )
    -> RETURNS INTEGER
    -> READS SQL DATA
    -> BEGIN
    ->     DECLARE book_count INTEGER;
    ->     SELECT COUNT(*) INTO book_count FROM Books WHERE Category = p_Category;
    ->     RETURN book_count;
    -> END$$
Query OK, 0 rows affected (0.01 sec)
```

- **Procedure to Delete Expired Books:**



```
mysql> DELIMITER $$
mysql> CREATE PROCEDURE DeleteExpiredBooks ()
    -> BEGIN
    ->     DELETE FROM Books WHERE Expiry_Date < CURDATE();
    -> END$$
Query OK, 0 rows affected (0.01 sec)
```

- **Function to Calculate Total Revenue:**



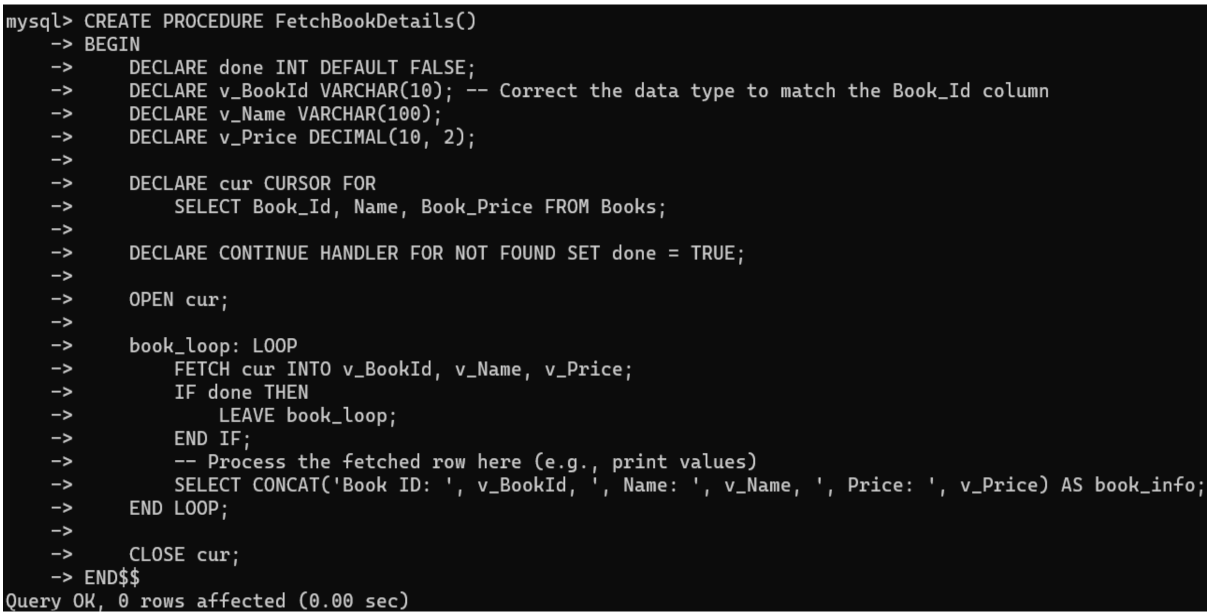
```
mysql> DELIMITER $$
mysql> CREATE FUNCTION CalculateTotalRevenue ()
    -> RETURNS INTEGER
    -> READS SQL DATA
    -> BEGIN
    ->     DECLARE total_revenue INTEGER;
    ->     SELECT SUM(Book_Price) INTO total_revenue FROM Books;
    ->     RETURN total_revenue;
    -> END$$
Query OK, 0 rows affected (0.01 sec)
```

Figure 4.1.5 – PROCEDURES AND FUNCTIONS

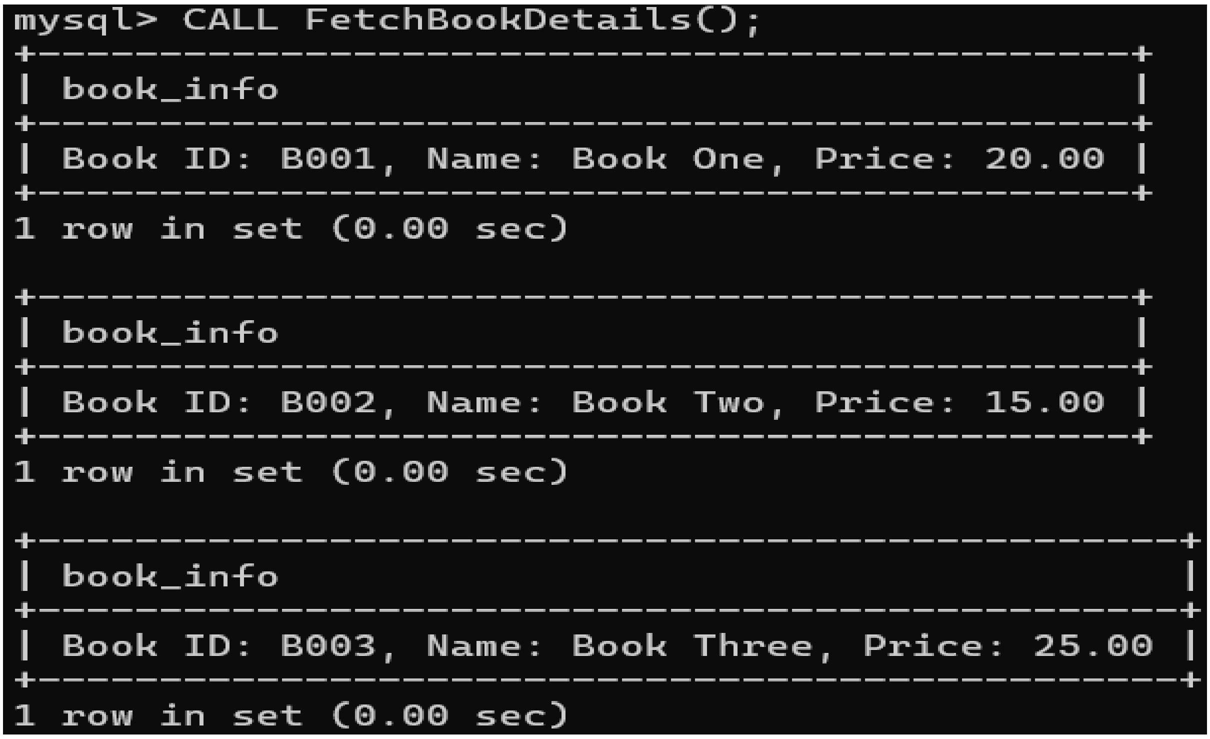
4.1.6 PL/SQL Cursors and Exceptional Handling of Book Bank Management System

Cursors serve the primary purpose of fetching data from a result set one row at a time, in contrast to SQL commands that operate on all rows in the result set simultaneously. They are employed when users require updating records individually or row by row within a database table.

- **Cursor to Fetch Book Details:**

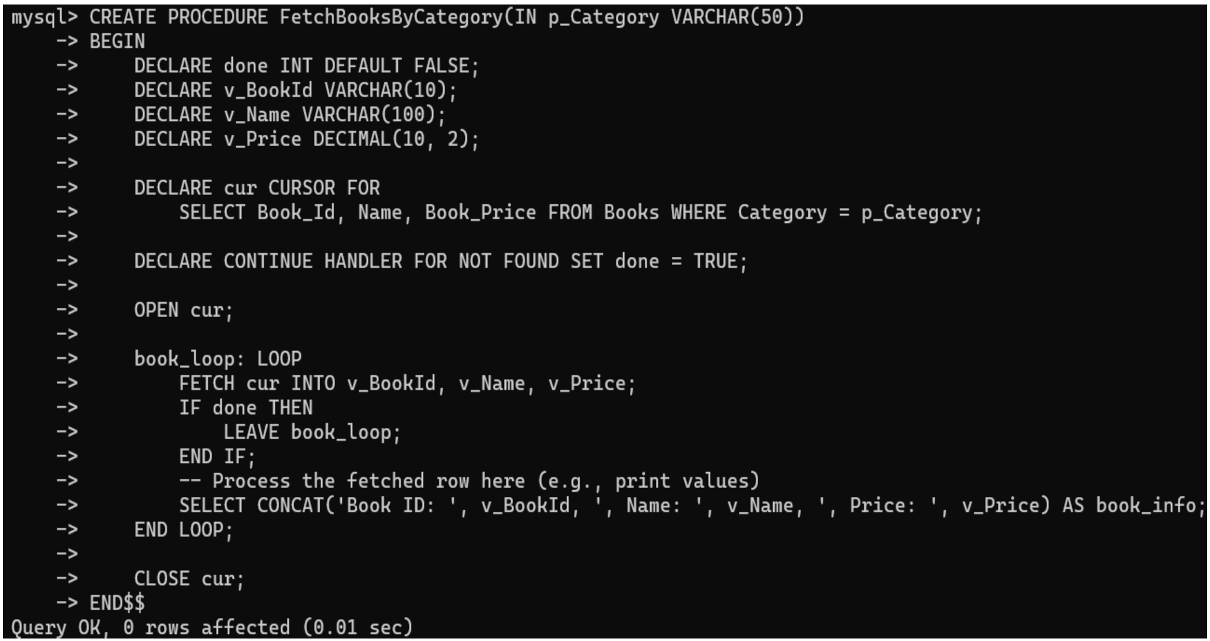


```
mysql> CREATE PROCEDURE FetchBookDetails()
    -> BEGIN
    ->     DECLARE done INT DEFAULT FALSE;
    ->     DECLARE v_BookId VARCHAR(10); -- Correct the data type to match the Book_Id column
    ->     DECLARE v_Name VARCHAR(100);
    ->     DECLARE v_Price DECIMAL(10, 2);
    ->
    ->     DECLARE cur CURSOR FOR
    ->         SELECT Book_Id, Name, Book_Price FROM Books;
    ->
    ->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
    ->
    ->     OPEN cur;
    ->
    ->     book_loop: LOOP
    ->         FETCH cur INTO v_BookId, v_Name, v_Price;
    ->         IF done THEN
    ->             LEAVE book_loop;
    ->         END IF;
    ->         -- Process the fetched row here (e.g., print values)
    ->         SELECT CONCAT('Book ID: ', v_BookId, ', Name: ', v_Name, ', Price: ', v_Price) AS book_info;
    ->     END LOOP;
    ->
    ->     CLOSE cur;
    -> END$$
Query OK, 0 rows affected (0.00 sec)
```

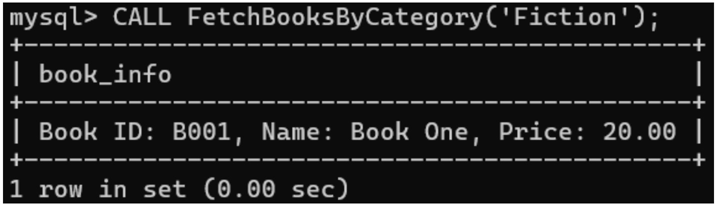


```
mysql> CALL FetchBookDetails();
+-------------------------------------------+
| book_info                                 |
+-------------------------------------------+
| Book ID: B001, Name: Book One, Price: 20.00 |
+-------------------------------------------+
1 row in set (0.00 sec)

+-------------------------------------------+
| book_info                                 |
+-------------------------------------------+
| Book ID: B002, Name: Book Two, Price: 15.00 |
+-------------------------------------------+
1 row in set (0.00 sec)

+---------------------------------------------+
| book_info                                   |
+---------------------------------------------+
| Book ID: B003, Name: Book Three, Price: 25.00 |
+---------------------------------------------+
1 row in set (0.00 sec)
```

- **Cursor with Parameters:**

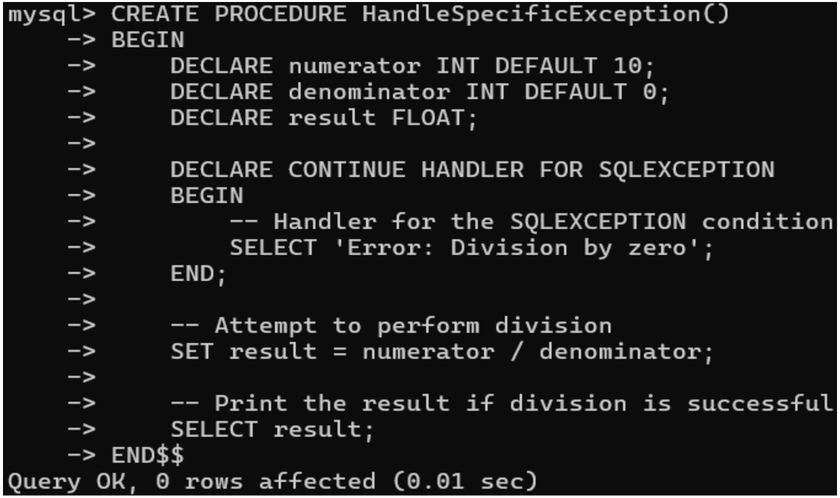


```
mysql> CREATE PROCEDURE FetchBooksByCategory(IN p_Category VARCHAR(50))
    -> BEGIN
    ->     DECLARE done INT DEFAULT FALSE;
    ->     DECLARE v_BookId VARCHAR(10);
    ->     DECLARE v_Name VARCHAR(100);
    ->     DECLARE v_Price DECIMAL(10, 2);
    ->
    ->     DECLARE cur CURSOR FOR
    ->         SELECT Book_Id, Name, Book_Price FROM Books WHERE Category = p_Category;
    ->
    ->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
    ->
    ->     OPEN cur;
    ->
    ->     book_loop: LOOP
    ->         FETCH cur INTO v_BookId, v_Name, v_Price;
    ->         IF done THEN
    ->             LEAVE book_loop;
    ->         END IF;
    ->         -- Process the fetched row here (e.g., print values)
    ->         SELECT CONCAT('Book ID: ', v_BookId, ', Name: ', v_Name, ', Price: ', v_Price) AS book_info;
    ->     END LOOP;
    ->
    ->     CLOSE cur;
    -> END$$
Query OK, 0 rows affected (0.01 sec)
```

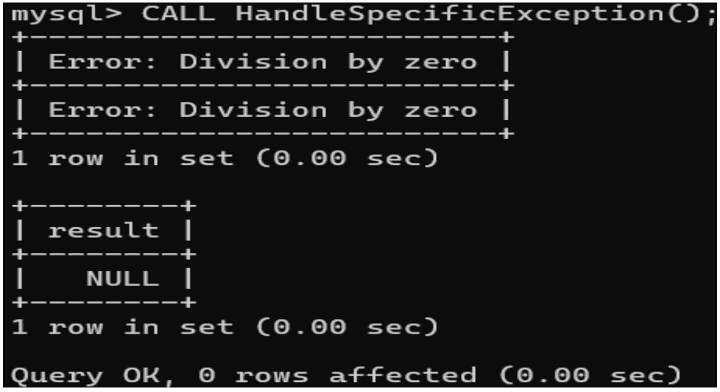


```
mysql> CALL FetchBooksByCategory('Fiction');
+---------------------------------------------+
| book_info                                   |
+---------------------------------------------+
| Book ID: B001, Name: Book One, Price: 20.00 |
+---------------------------------------------+
1 row in set (0.00 sec)
```

- **Handle Specific Exception:**

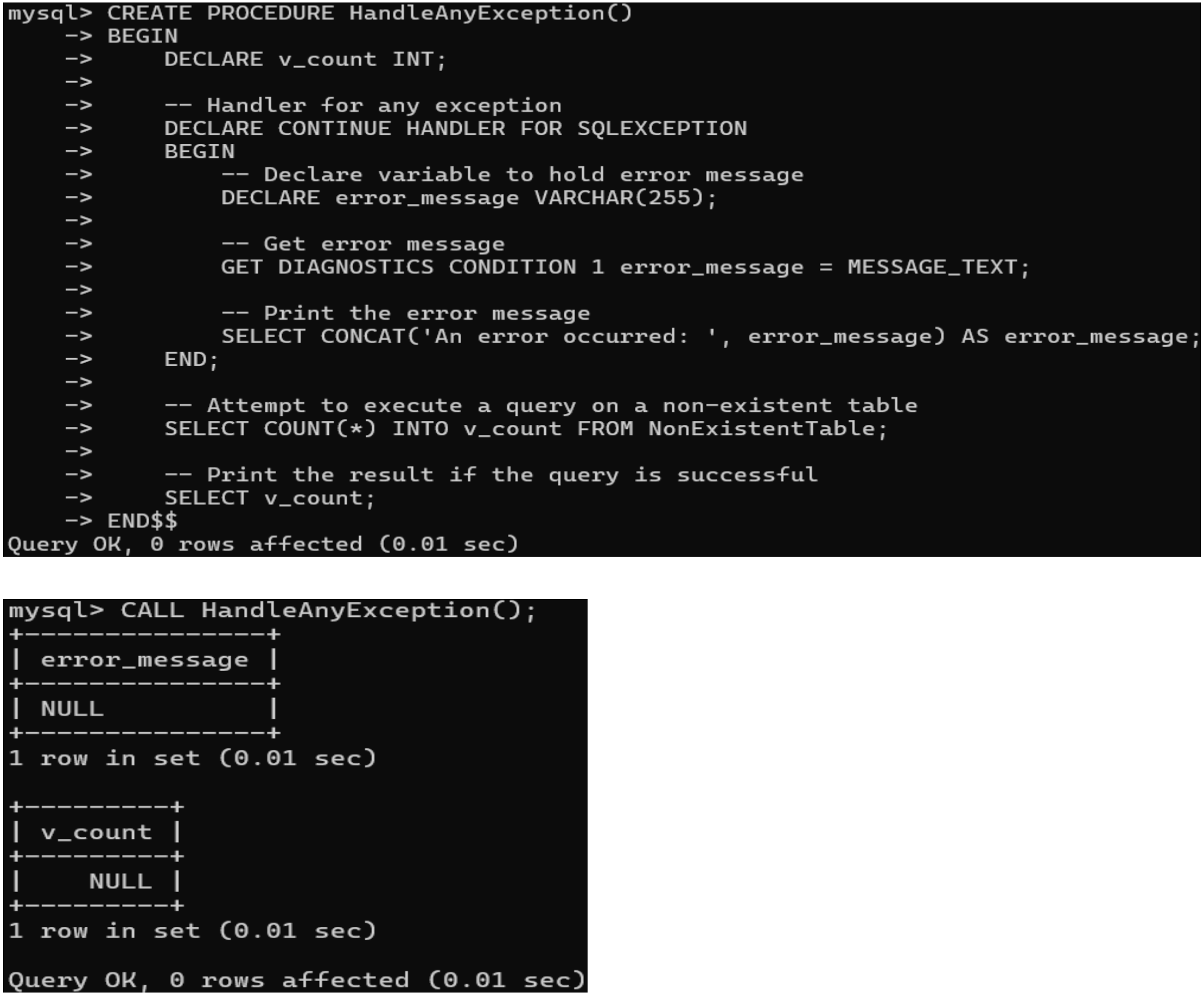


```
mysql> CREATE PROCEDURE HandleSpecificException()
    -> BEGIN
    ->     DECLARE numerator INT DEFAULT 10;
    ->     DECLARE denominator INT DEFAULT 0;
    ->     DECLARE result FLOAT;
    ->
    ->     DECLARE CONTINUE HANDLER FOR SQLEXCEPTION
    ->     BEGIN
    ->         -- Handler for the SQLEXCEPTION condition
    ->         SELECT 'Error: Division by zero';
    ->     END;
    ->
    ->     -- Attempt to perform division
    ->     SET result = numerator / denominator;
    ->
    ->     -- Print the result if division is successful
    ->     SELECT result;
    -> END$$
Query OK, 0 rows affected (0.01 sec)
```



```
mysql> CALL HandleSpecificException();
+-------------------------+
| Error: Division by zero |
+-------------------------+
| Error: Division by zero |
+-------------------------+
1 row in set (0.00 sec)

+--------+
| result |
+--------+
|   NULL |
+--------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

- **Handle Any Exception:**

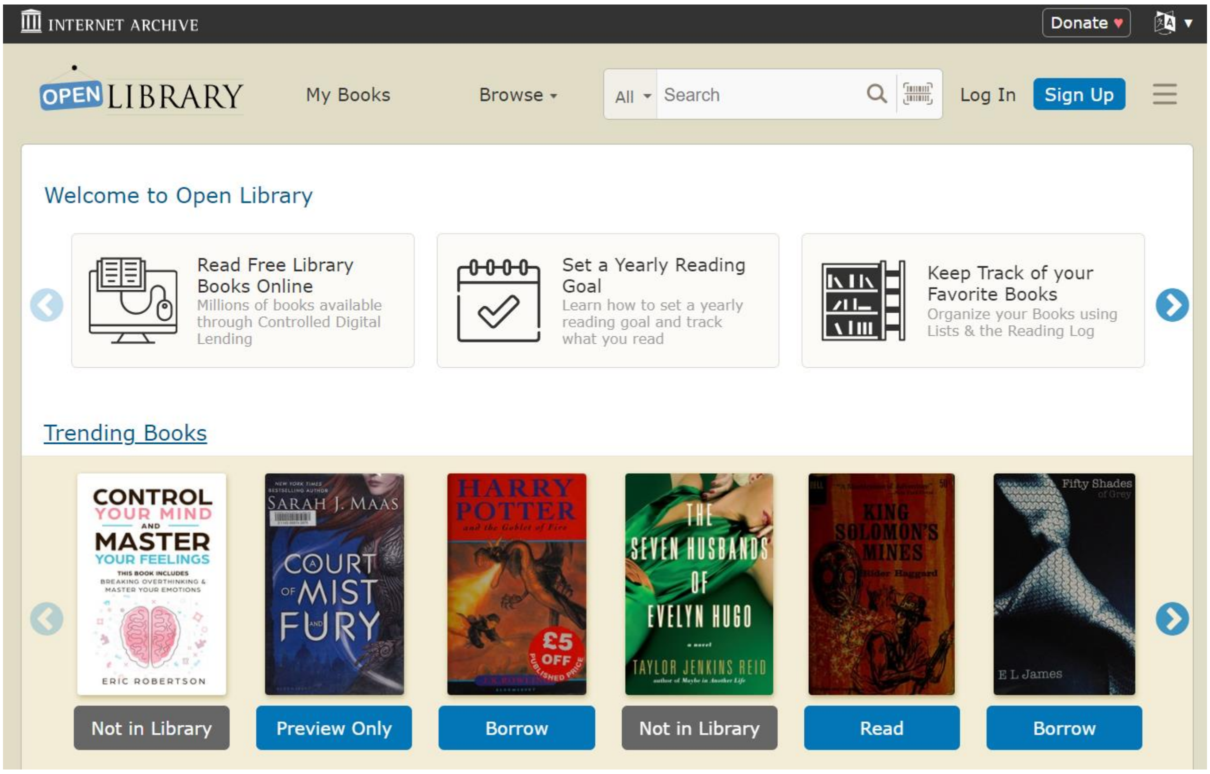


```
mysql> CREATE PROCEDURE HandleAnyException()
    -> BEGIN
    ->     DECLARE v_count INT;
    ->
    ->     -- Handler for any exception
    ->     DECLARE CONTINUE HANDLER FOR SQLEXCEPTION
    ->     BEGIN
    ->         -- Declare variable to hold error message
    ->         DECLARE error_message VARCHAR(255);
    ->
    ->         -- Get error message
    ->         GET DIAGNOSTICS CONDITION 1 error_message = MESSAGE_TEXT;
    ->
    ->         -- Print the error message
    ->         SELECT CONCAT('An error occurred: ', error_message) AS error_message;
    ->     END;
    ->
    ->     -- Attempt to execute a query on a non-existent table
    ->     SELECT COUNT(*) INTO v_count FROM NonExistentTable;
    ->
    ->     -- Print the result if the query is successful
    ->     SELECT v_count;
    -> END$$
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> CALL HandleAnyException();
+---------------+
| error_message |
+---------------+
| NULL          |
+---------------+
1 row in set (0.01 sec)

+---------+
| v_count |
+---------+
|    NULL |
+---------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)
```

Figure 4.1.6 – CURSORS AND EXCEPTION HANDLING

4.1.7 HOME



Welcome to Open Library

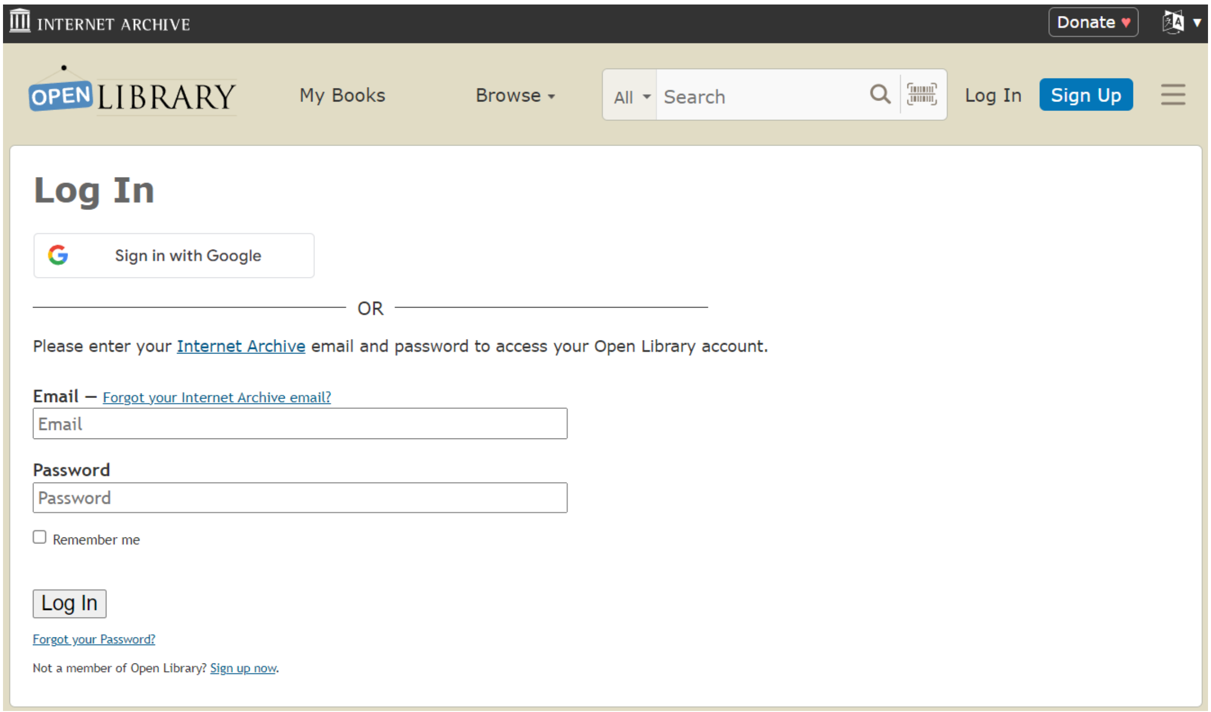
Read Free Library Books Online
Millions of books available through Controlled Digital Lending

Set a Yearly Reading Goal
Learn how to set a yearly reading goal and track what you read

Keep Track of your Favorite Books
Organize your Books using Lists & the Reading Log

Trending Books

LOGIN



Log In

Sign in with Google

OR

Please enter your Internet Archive email and password to access your Open Library account.

Email — Forgot your Internet Archive email?

Email

Password

Password

□ Remember me

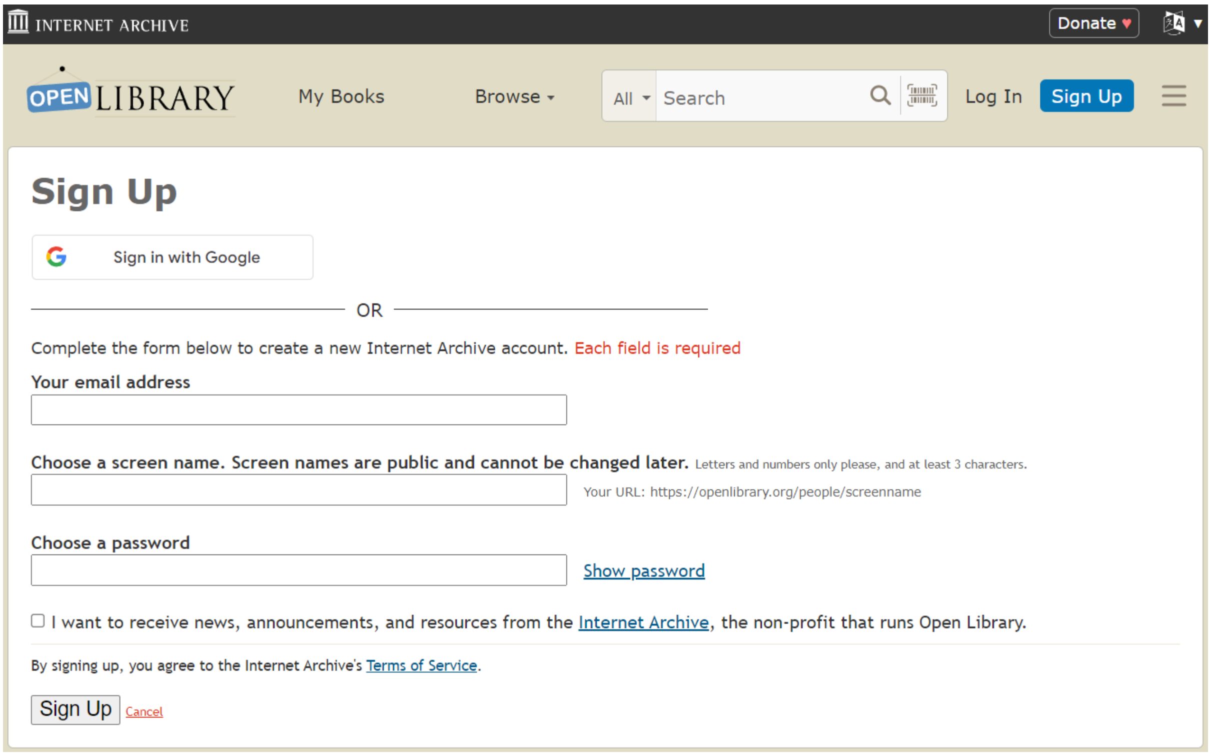
Log In

Forgot your Password?

Not a member of Open Library? Sign up now.

Figure 4.1.7 – HOME PAGE & LOGIN PAGE

4.1.8 SIGN UP



Sign Up

Complete the form below to create a new Internet Archive account. Each field is required

Your email address

Choose a screen name. Screen names are public and cannot be changed later. Letters and numbers only please, and at least 3 characters.

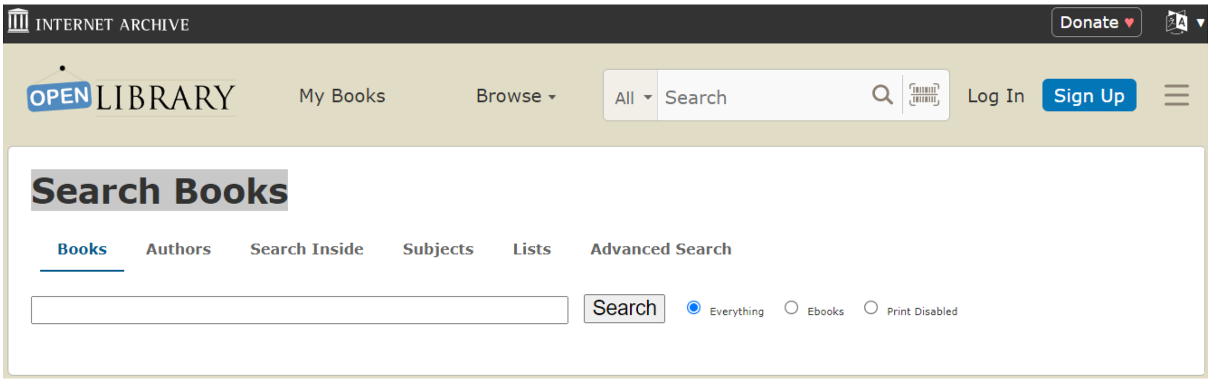
Your URL: https://openlibrary.org/people/screenname

Choose a password

□ I want to receive news, announcements, and resources from the Internet Archive, the non-profit that runs Open Library.

By signing up, you agree to the Internet Archive's Terms of Service.

SEARCH BOOKS

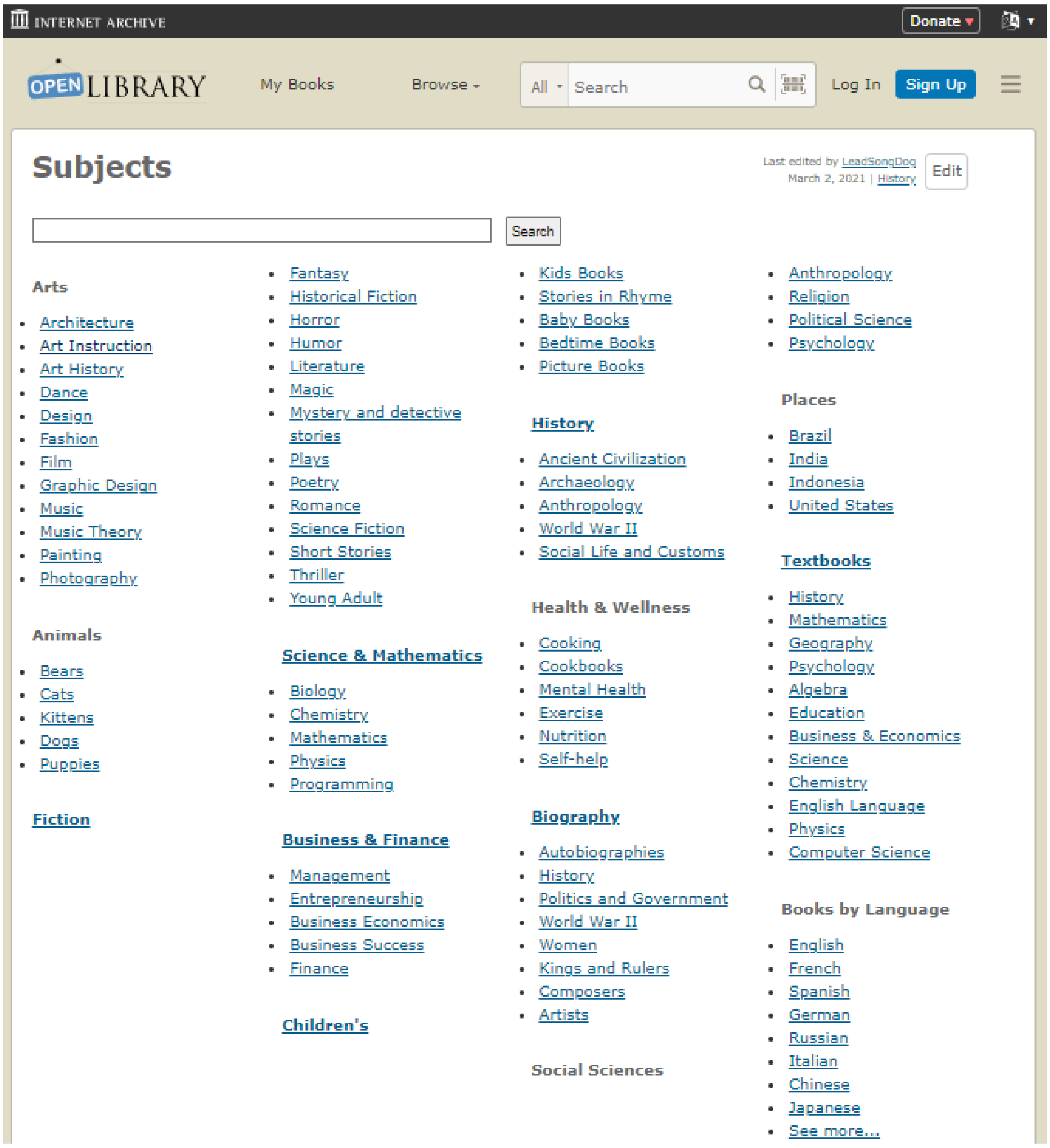


Search Books

● Everything ○ Ebooks ○ Print Disabled

Figure 4.1.8 – SIGN UP PAGE & SEARCH BOOK PAGE

4.1.9 BROWSE BOOKS



INTERNET ARCHIVE Donate

OPEN LIBRARY My Books Browse All Search Log In Sign Up

Subjects

Last edited by LeadSongDog March 2, 2021 | History Edit

Search

Arts

- Architecture
- Art Instruction
- Art History
- Dance
- Design
- Fashion
- Film
- Graphic Design
- Music
- Music Theory
- Painting
- Photography

Animals

- Bears
- Cats
- Kittens
- Dogs
- Puppies

Fiction

- Fantasy
- Historical Fiction
- Horror
- Humor
- Literature
- Magic
- Mystery and detective stories
- Plays
- Poetry
- Romance
- Science Fiction
- Short Stories
- Thriller
- Young Adult

Science & Mathematics

- Biology
- Chemistry
- Mathematics
- Physics
- Programming

Business & Finance

- Management
- Entrepreneurship
- Business Economics
- Business Success
- Finance

Children's

- Kids Books
- Stories in Rhyme
- Baby Books
- Bedtime Books
- Picture Books

History

- Ancient Civilization
- Archaeology
- Anthropology
- World War II
- Social Life and Customs

Health & Wellness

- Cooking
- Cookbooks
- Mental Health
- Exercise
- Nutrition
- Self-help

Biography

- Autobiographies
- History
- Politics and Government
- World War II
- Women
- Kings and Rulers
- Composers
- Artists

Social Sciences

- Anthropology
- Religion
- Political Science
- Psychology

Places

- Brazil
- India
- Indonesia
- United States

Textbooks

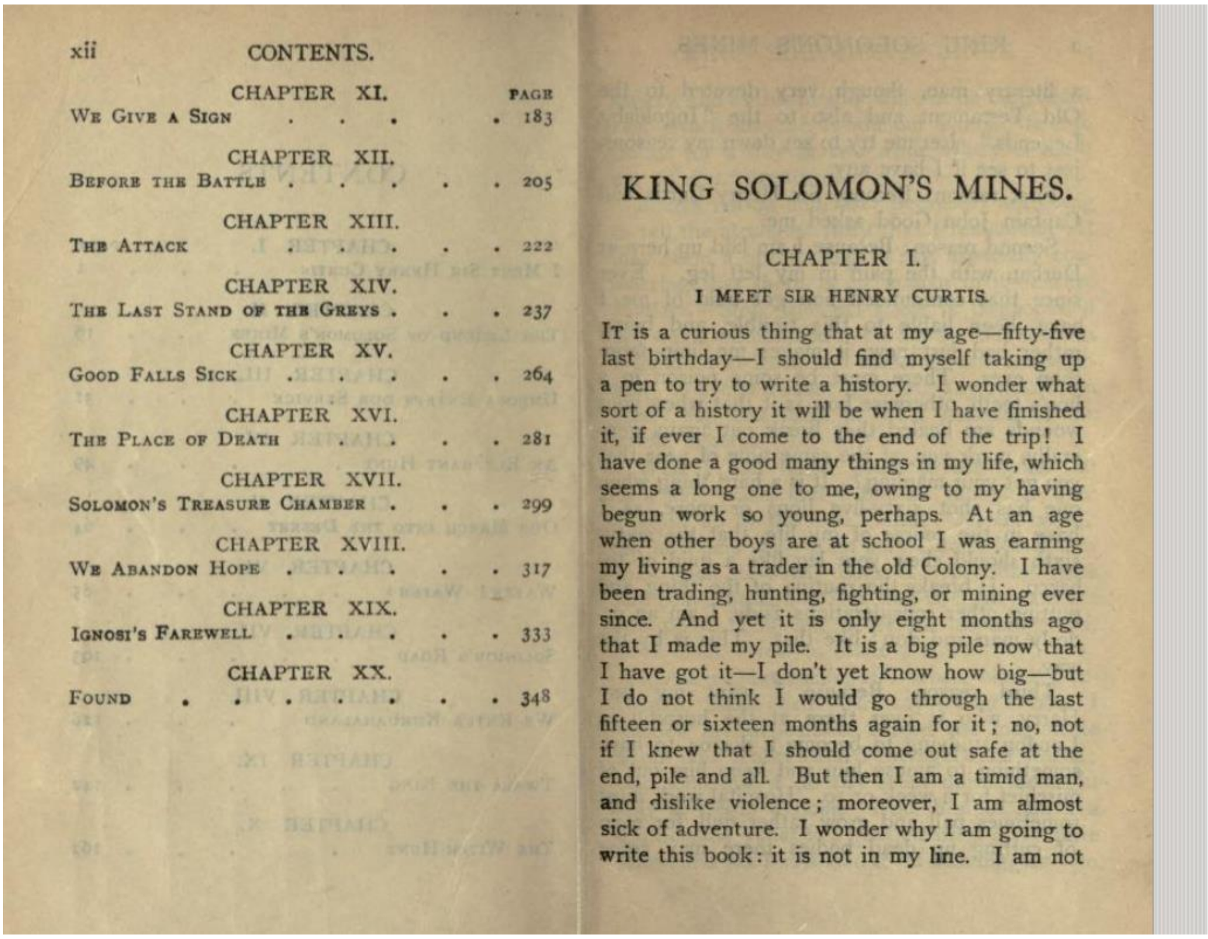
- History
- Mathematics
- Geography
- Psychology
- Algebra
- Education
- Business & Economics
- Science
- Chemistry
- English Language
- Physics
- Computer Science

Books by Language

- English
- French
- Spanish
- German
- Russian
- Italian
- Chinese
- Japanese
- See more...

Figure 4.1.9 – BROWSE BOOKS PAGE

4.1.10 ONLINE BOOK READING



KING SOLOMON'S MINES.

CHAPTER I.

I MEET SIR HENRY CURTIS.

It is a curious thing that at my age—fifty-five last birthday—I should find myself taking up a pen to try to write a history. I wonder what sort of a history it will be when I have finished it, if ever I come to the end of the trip! I have done a good many things in my life, which seems a long one to me, owing to my having begun work so young, perhaps. At an age when other boys are at school I was earning my living as a trader in the old Colony. I have been trading, hunting, fighting, or mining ever since. And yet it is only eight months ago that I made my pile. It is a big pile now that I have got it—I don't yet know how big—but I do not think I would go through the last fifteen or sixteen months again for it; no, not if I knew that I should come out safe at the end, pile and all. But then I am a timid man, and dislike violence; moreover, I am almost sick of adventure. I wonder why I am going to write this book: it is not in my line. I am not

Figure 4.1.9 – BOOK READING PAGE

APPLICATIONS

Book Bank System facilitates the lending and management of books, an integrated software solution plays a pivotal role. Here's how such a system can be applied:

- **Book Bank Management System:** The system allows the book bank to meticulously track and manage its inventory of available books. Through real-time updates and notifications, librarians can efficiently monitor the availability of books, ensuring that borrowers have access to the titles they require.
- **Lending Process Automation:** By automating lending processes such as book checkouts, renewals, and returns, the system streamlines operations for librarians and borrowers alike. Borrowers can easily browse available titles, request loans, and manage their borrowing history, while librarians can efficiently process these requests and maintain accurate records.
- **Reservation System:** The system can incorporate a reservation feature, enabling borrowers to reserve books that are currently on loan. This ensures equitable access to popular titles and helps manage demand for high-demand resources.
- **Fine Management:** Integrated fine management features allow the book bank to automate the calculation and collection of fines for overdue books. This ensures accountability among borrowers and encourages the timely return of borrowed materials.
- **Reporting and Analytics:** Robust reporting and analytics capabilities provide insights into book lending patterns, popular titles, and borrower demographics. This data-driven approach enables the book bank to make informed decisions regarding collection development, resource allocation, and service improvements.
- **User Profiles and Preferences:** By maintaining user profiles and preferences, the system can personalize the borrowing experience for individual borrowers. This includes recommendations based on past borrowing history, notifications for new arrivals in preferred genres, and tailored communication regarding overdue loans or reservation availability.

In essence, the book bank system serves as a comprehensive platform for efficient book management, borrower engagement, and data-driven decision-making. By leveraging technology to automate processes and enhance user experiences, the system optimizes operations and fosters a culture of lifelong learning and access to knowledge.

CHAPTER V

CONCLUSION

While developing this project I have learnt a lot about HTML/CSS/JS /MySQL and working with database management, I have also learnt how to make the application user- friendly (easy to use and handle) by hiding the complicated parts of it from the users.

During the development process, I studied carefully and understood the criteria for making a software more demanding, I also realized the importance of maintaining a minimal margin for errors. I inferred various modules that should be used in Ecommerce management system which makes the flow much simpler. The importance of database management system is that its needed for developing efficient web applications, MySQL is the most simplest and easier to learn and implement. MySQL workbench is an application which helps in developing MySQL codes and supports every functionality of SQL like views, PL/SQL procedures, functions, cursors, exception handling etc.

BIBLIOGRAPHY

It has been a matter of immense pleasure, honour and challenge to have this opportunity to take up this project and complete it successfully.
I have obtained information from various resources to design and implement our project.

I have acquired most of the knowledge from the Internet.

The following are some of the resources:

- Silva, A., & Oliveira, E. (2019). Design and Implementation of a Book Management System Using Java. International Journal of Computer Applications, 182(39), 25-29.
- Smith, J., & Johnson, M. (2020). Efficient Data Management Techniques for Book Inventory Systems. Journal of Information Systems, 15(2), 87-104.
- Wang, H., & Li, Q. (2018). A Comparative Study of Database Models for Book Cataloging Systems. Journal of Database Management, 29(3), 58-72. doi:10.4018/JDM.2018070104
- Jones, R., & Brown, S. (2017). Implementation Strategies for Book Checkout Systems in Libraries. Library Information Science Research, 39(2), 134-150.
- Patel, R., & Shah, S. (2021). Integration of Machine Learning Algorithms for Book Recommendation Systems. International Journal of Advanced Computer Science and Applications, 12(5), 310-317.
- Johnson, T., & Williams, E. (2016). User Interface Design Considerations for Book Management Systems. Journal of Human-Computer Interaction, 24(3), 210-225.
- Garcia, L., & Martinez, A. (2018). Security Measures in Book Management Systems: A Comparative Analysis. International Journal of Information Security and Privacy, 12(4), 45-60.
- Kim, Y., & Lee, S. (2019). Scalability and Performance Evaluation of Distributed Book Management Systems. IEEE Transactions on Parallel and Distributed Systems, 30(8), 1600-1612. doi:10.1109/TPDS.2019.2890479
- Wang, X., & Chen, Y. (2020). Blockchain Technology for Book Ownership and Distribution Management. International Journal of Blockchain and Distributed Ledger Technology, 2(1), 78-92.
- Liu, H., & Zhang, Q. (2017). Data Mining Techniques for Predictive Analysis in Book Management Systems. Journal of Big Data Research, 5(2), 112-125.