PROJECT FOR SQL MODULE

LIBRARY MANAGEMENT SYSTEM

Institute Name**:-** Itvedant Education Pvt .Ltd
Name:- Janhavi Arun Patole
Email Address:- janpatole99@gmail.com
Date of submission:-19 Nov 2023

Project Aims:

Database Design and Schema Creation:

Create an optimized and normalized database schema that accurately represents the relationships between different entities such as books, members, categories, and transactions.

Data Population and Integrity:

Populate the database with sample data to simulate real-world scenarios in a library environment.

Ensure data integrity through the use of constraints, such as primary keys, foreign keys, and unique constraints.

User Authentication and Authorization:

Implement a user authentication system to secure the database, allowing only authorized users to access and modify data.

Define and enforce role-based access control to restrict users' actions based on their roles (e.g., librarian, administrator).

Transaction Management:

Implement a transaction management system to handle the borrowing and return of books by library members.

Track transaction details, including dates, due dates, and return statuses.

Lost Book Tracking:

Develop a mechanism to track lost books, including recording the details of lost books and associated member information.

Implement a reporting system to identify and manage lost books effectively.

Category and Publisher Management:

Enable efficient categorization of books by implementing a category management system.

Incorporate publisher details to facilitate organization and retrieval of books based on their publishers.

Member Information Management:

Create a system for managing member information, including details such as names, contact information, and membership types.

Implement features for adding, updating, and deactivating member records.

Reporting and Analysis:

Develop SQL queries for generating various reports, such as a list of currently borrowed books, overdue books, and popular book categories.

Implement analysis queries to gather insights into library usage patterns.

User-Friendly Interface:

Consider the development of a user-friendly interface or integrate the database with an existing library management system for seamless interaction.**Scalability and Performance:**

Design the database with scalability in mind to accommodate potential future expansions of the library's collection and user base.

Optimize SQL queries and database indexing for improved performance.

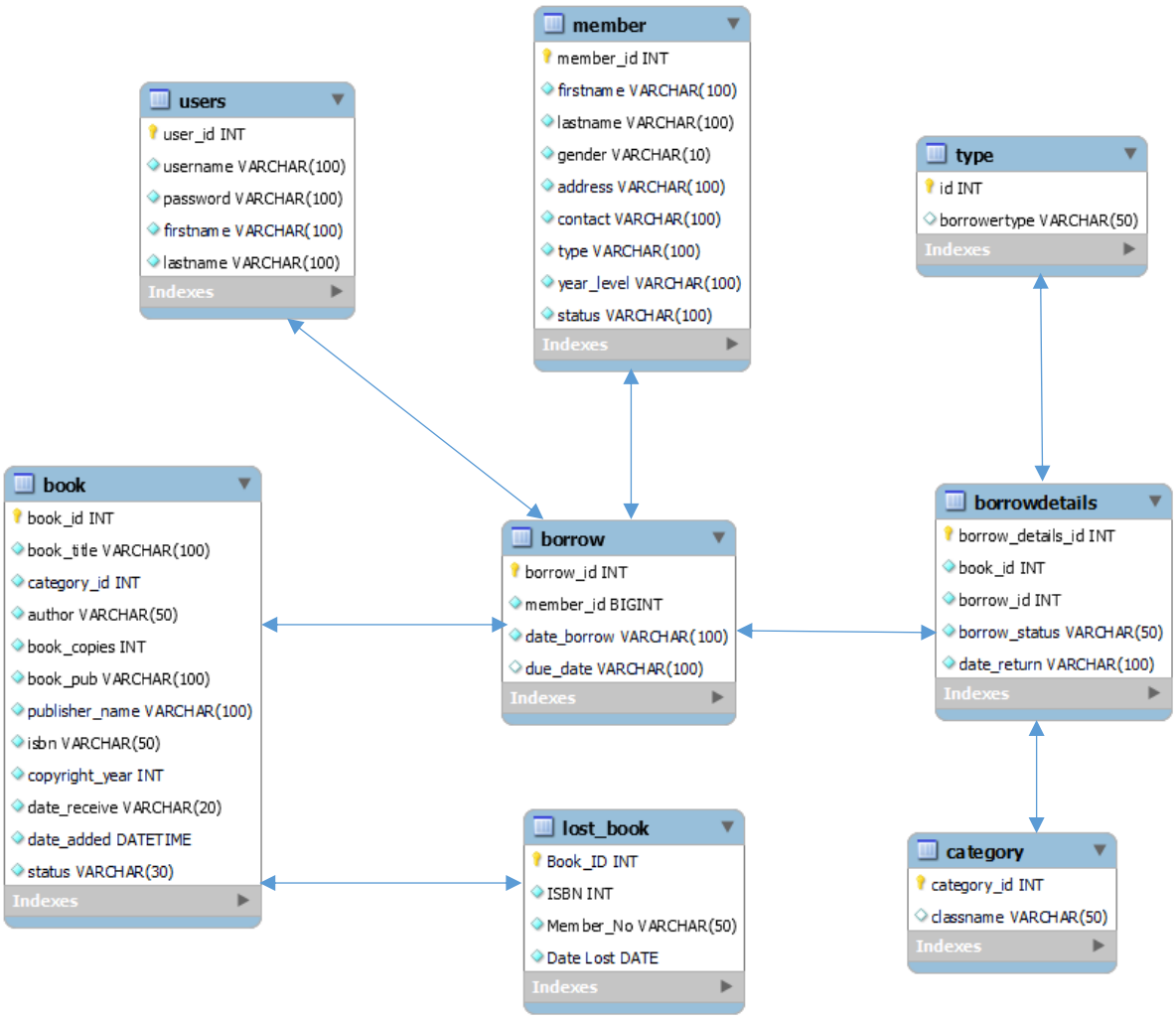
By achieving these aims, the project aims to create a well-organized, secure, and efficient database system that enhances the overall management of the library's resources and services.

Project Objective:

The objective of this SQL project is to design and implement a comprehensive database system for a library management system. The primary focus is on creating

a robust and efficient database schema to manage various aspects of library operations, including book cataloging, member information, borrowing and return transactions, and tracking lost books. The project aims to enhance the overall efficiency and organization of the library by providing a centralized and well-structured database.

In this ER diagram:



Each rectangle represents an entity. Lines between the entities represent the relationships between them.

The lines between the entities represent primary keys and foreign keys connecting the tables.

The ER diagram showcases the relationships between the different tables, such as users, orders, products, categories, and more, as described in the project description.

Table description:

1. **Book**

Field	Type	Null	Key	Default	Extra
book_id	int	NO	PRI	NULL	auto_increment
book_title	varchar(100)	NO		NULL	
category_id	int	NO		NULL	
author	varchar(50)	NO		NULL	
book_copies	int	NO		NULL	
book_pub	varchar(100)	NO		NULL	
publisher_name	varchar(100)	NO		NULL	
isbn	varchar(50)	NO		NULL	
copyright_year	int	NO		NULL	
date_receive	varchar(20)	NO		NULL	
date_added	datetime	NO		NULL	
status	varchar(30)	NO		NULL	

2. **Borrow**

Field	Type	Null	Key	Default	Extra
borrow_id	int	NO	PRI	NULL	auto_increment
member_id	bigint	NO	MUL	NULL	
date_borrow	varchar(100)	NO		NULL	
due_date	varchar(100)	YES		NULL	

3. **Borrowdetails**

Field	Type	Null	Key	Default	Extra
borrow_details_id	int	NO	PRI	NULL	auto_increment
book_id	int	NO		NULL	
borrow_id	int	NO		NULL	
borrow_status	varchar(50)	NO		NULL	
date_return	varchar(100)	NO		NULL	

4. **Category**

Field	Type	Null	Key	Default	Extra
category_id	int	NO	PRI	NULL	auto_increment
classname	varchar(50)	YES		NULL	

5. **Lost_book**

Field	Type	Null	Key	Default	Extra
Book_ID	int	NO	PRI	NULL	auto_increment
ISBN	int	NO		NULL	
Member_No	varchar(50)	NO		NULL	
Date Lost	date	NO		NULL	

6. Member

Field	Type	Null	Key	Default	Extra
member_id	int	NO	PRI	NULL	auto_increment
firstname	varchar(100)	NO		NULL	
lastname	varchar(100)	NO		NULL	
gender	varchar(10)	NO		NULL	
address	varchar(100)	NO		NULL	
contact	varchar(100)	NO		NULL	
type	varchar(100)	NO		NULL	
year_level	varchar(100)	NO		NULL	
status	varchar(100)	NO		NULL	

7. Type

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
borrowertype	varchar(50)	YES	MUL	NULL	

8. Users

Field	Type	Null	Key	Default	Extra
user_id	int	NO	PRI	NULL	auto_increment
username	varchar(100)	NO		NULL	
password	varchar(100)	NO		NULL	
firstname	varchar(100)	NO		NULL	
lastname	varchar(100)	NO		NULL	

Commands:

```
create database library;

use library;


CREATE TABLE `book` (
  `book_id` int(11) NOT NULL AUTO_INCREMENT,
  `book_title` varchar(100) NOT NULL,
  `category_id` int(50) NOT NULL,
  `author` varchar(50) NOT NULL,
  `book_copies` int(11) NOT NULL,
  `book_pub` varchar(100) NOT NULL,
  `publisher_name` varchar(100) NOT NULL,
  `isbn` varchar(50) NOT NULL,
  `copyright_year` int(11) NOT NULL,
  `date_receive` varchar(20) NOT NULL,
  `date_added` datetime NOT NULL,
  `status` varchar(30) NOT NULL,
  PRIMARY KEY (`book_id`)
);



INSERT INTO `book` (`book_id`, `book_title`, `category_id`, `author`, `book_copies`, `book_pub`, `publisher_name`, `isbn`, `copyright_year`, `date_receive`, `date_added`, `status`) VALUES

(15, 'Natural Resources', 8, 'Robin Kerrod', 15, 'Marshall Cavendish Corporation', 'Marshall', '1-85435-628-3', 1997, '', '2013-12-11 06:34:27', 'New'),

(16, 'Encyclopedia Americana', 5, 'Grolier', 20, 'Connecticut', 'Grolier Incorporation', '0-7172-0119-8', 1988, '', '2013-12-11 06:36:23', 'Archive'),

(17, 'Algebra 1', 3, 'Carolyn Bradshaw, Michael Seals', 35, 'Pearson Education, Inc', 'Prentice Hall, New Jersey', '0-13-125087-6', 2004, '', '2013-12-11 06:39:17', 'Damage'),

(18, 'The Philippine Daily Inquirer', 7, '..', 3, 'Pasay City', '..', '..', 2013, '', '2013-12-11 06:41:53', 'New'),
```

(19, 'Science in our World', 4, 'Brian Knapp', 25, 'Regency Publishing Group', 'Prentice Hall, Inc', '0-13-050841-1', 1996, '', '2013-12-11 06:44:44', 'Lost'),

(20, 'Literature', 9, 'Greg Glowka', 20, 'Regency Publishing Group', 'Prentice Hall, Inc', '0-13-050841-1', 2001, '', '2013-12-11 06:47:44', 'Old'),

(21, 'Lexicon Universal Encyclopedia', 5, 'Lexicon', 10, 'Lexicon Publication', 'Pulication Inc., Lexicon', '0-7172-2043-5', 1993, '', '2013-12-11 06:49:53', 'Old'),

(22, 'Science and Invention Encyclopedia', 5, 'Clarke Donald, Dartford Mark', 16, 'H.S. Stuttman inc. Publishing', 'Publisher , Westport Connecticut', '0-87475-450-x', 1992, '', '2013-12-11 06:52:58', 'New'),

(23, 'Integrated Science Textbook ', 4, 'Merde C. Tan', 15, 'Vibal Publishing House Inc.', '12536. Araneta Avenue Corner Ma. Clara St., Quezon City', '971-570-124-8', 2009, '', '2013-12-11 06:55:27', 'New'),

(24, 'Algebra 2', 3, 'Glencoe McGraw Hill', 15, 'The McGrawHill Companies Inc.', 'McGrawhill', '978-0-07-873830-2', 2008, '', '2013-12-11 06:57:35', 'New'),

(25, 'Wiki at Panitikan ', 7, 'Lorenza P. Avera', 28, 'JGM & S Corporation', 'JGM & S Corporation', '971-07-1574-7', 2000, '', '2013-12-11 06:59:24', 'Damage'),

(26, 'English Expressways TextBook for 4th year', 9, 'Virginia Bermudez Ed. O. et al', 23, 'SD Publications, Inc.', 'Gregorio Araneta Avenue, Quezon City', '978-971-0315-33-8', 2007, '', '2013-12-11 07:01:25', 'New'),

(27, 'Asya Pag-usbong Ng Kabihasnan ', 8, 'Ricardo T. Jose, Ph . D.', 21, 'Vibal Publishing House Inc.', 'Araneta Avenue . Cor. Maria Clara St., Quezon City', '971-07-2324-3', 2008, '', '2013-12-11 07:02:56', 'New'),

(28, 'Literature (the readers choice)', 9, 'Glencoe McGraw Hill', 20, '..', 'the McGrawHill Companies Inc', '0-02-817934-x', 2001, '', '2013-12-11 07:05:25', 'Damage'),

(29, 'Beloved a Novel', 9, 'Toni Morrison', 13, '..', 'Alfred A. Knoff, Inc', '0-394-53597-9', 1987, '', '2013-12-11 07:07:02', 'Old'),

(30, 'Silver Burdett Engish', 2, 'Judy Brim', 12, 'Silver Burdett Company', 'Silver', '0-382-03575-5', 1985, '', '2013-12-11 09:22:50', 'Old'),

(31, 'The Corporate Warriors (Six Classic Cases in American Business)', 8, 'Douglas K. Ramsey', 8, 'Houghton Miffin Company', '..', '0-395-35487-0', 1987, '', '2013-12-11 09:25:32', 'Old'),

(32, 'Introduction to Information System', 9, 'Cristine Redoblo', 10, 'CHMSC', 'Brian INC', '123-132', 2013, '', '2014-01-17 19:00:10', 'New');

CREATE TABLE `borrow` (

```
  `borrow_id` int(11) NOT NULL AUTO_INCREMENT,
  `member_id` bigint(50) NOT NULL,
  `date_borrow` varchar(100) NOT NULL,
  `due_date` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`borrow_id`),
  KEY `borrowerid` (`member_id`),
  KEY `borrowid` (`borrow_id`)
);



INSERT INTO `borrow` (`borrow_id`, `member_id`, `date_borrow`, `due_date`)
VALUES
(484, 55, '2014-03-20 23:50:27', '21/03/2014'),
(483, 55, '2014-03-20 23:49:34', '21/03/2014'),
(482, 52, '2014-03-20 23:38:22', '03/01/2014');



CREATE TABLE  `borrowdetails` (
  `borrow_details_id` int(11) NOT NULL AUTO_INCREMENT,
  `book_id` int(11) NOT NULL,
  `borrow_id` int(11) NOT NULL,
  `borrow_status` varchar(50) NOT NULL,
  `date_return` varchar(100) NOT NULL,
  PRIMARY KEY (`borrow_details_id`)
);



INSERT INTO `borrowdetails` (`borrow_details_id`, `book_id`, `borrow_id`,
`borrow_status`, `date_return`) VALUES
(164, 16, 484, 'pending', ''),
(162, 15, 482, 'pending', ''),
```

```
(163, 15, 483, 'returned', '2014-03-21 00:30:51');



CREATE TABLE  `category` (
  `category_id` int(11) NOT NULL AUTO_INCREMENT,
  `classname` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`category_id`),
  UNIQUE KEY `category_id` (`category_id`),
  KEY `classid` (`category_id`));
INSERT INTO `category` (`category_id`, `classname`) VALUES
(1, 'Periodical'),
(2, 'English'),
(3, 'Math'),
(4, 'Science'),
(5, 'Encyclopedia'),
(6, 'Filipiniana'),
(7, 'Newspaper'),
(8, 'General'),
(9, 'References');



CREATE TABLE `lost_book` (
  `Book_ID` int(11) NOT NULL AUTO_INCREMENT,
  `ISBN` int(11) NOT NULL,
  `Member_No` varchar(50) NOT NULL,
  `Date Lost` date NOT NULL,
  PRIMARY KEY (`Book_ID`)
);
```

```
CREATE TABLE `member` (
  `member_id` int(11) NOT NULL AUTO_INCREMENT,
  `firstname` varchar(100) NOT NULL,
  `lastname` varchar(100) NOT NULL,
  `gender` varchar(10) NOT NULL,
  `address` varchar(100) NOT NULL,
  `contact` varchar(100) NOT NULL,
  `type` varchar(100) NOT NULL,
  `year_level` varchar(100) NOT NULL,
  `status` varchar(100) NOT NULL,
  PRIMARY KEY (`member_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=66 ;



INSERT INTO `member` (`member_id`, `firstname`, `lastname`, `gender`, `address`, `contact`, `type`, `year_level`, `status`) VALUES
(52, 'Mark', 'Sanchez', 'Male', 'Talisay', '212010', 'Teacher', 'Faculty', 'Active'),
(53, 'April Joy', 'Aguilar', 'Female', 'E.B. Magalona', '00', 'Student', 'Second Year', 'Banned'),
(54, 'Alfonso', 'Pancho', 'Male', 'E.B. Magalona', '009', 'Student', 'First Year', 'Active'),
(55, 'Jonathan ', 'Antanilla', 'Male', 'E.B. Magalona', '0032', 'Student', 'Fourth Year', 'Active'),
(56, 'Renzo Bryan', 'Pedroso', 'Male', 'Silay City', '03030', 'Student', 'Third Year', 'Active'),
(57, 'Eleazar', 'Duterte', 'Male', 'E.B. Magalona', '90902', 'Student', 'Second Year', 'Active'),
(58, 'Ellen Mae', 'Espino', 'Female', 'E.B. Magalona', '123', 'Student', 'First Year', 'Active'),
(59, 'Ruth', 'Magbanua', 'Female', 'E.B. Magalona', '9340', 'Student', 'Second Year', 'Active'),
(60, 'Shaina Marie', 'Gabino', 'Female', 'Silay City', '132134', 'Student', 'Second Year', 'Active'),
```

(62, 'Chairty Joy', 'Punzalan', 'Female', 'E.B. Magalona', '12423', 'Teacher', 'Faculty', 'Active'),

(63, 'Kristine May', 'Dela Rosa', 'Female', 'Silay City', '1321', 'Student', 'Second Year', 'Active'),

(64, 'Chinie marie', 'Laborosa', 'Female', 'E.B. Magalona', '902101', 'Student', 'Second Year', 'Active'),

(65, 'Ruby', 'Morante', 'Female', 'E.B. Magalona', '', 'Teacher', 'Faculty', 'Active');

```
CREATE TABLE `type` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `borrowertype` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `borrowertype` (`borrowertype`),
  KEY `id` (`id`)
);

INSERT INTO `type` (`id`, `borrowertype`) VALUES
(2, 'Teacher'),
(20, 'Employee'),
(21, 'Non-Teaching'),
(22, 'Student'),
(32, 'Contruction');

CREATE TABLE `users` (
  `user_id` int(11) NOT NULL primary key AUTO_INCREMENT,
  `username` varchar(100) NOT NULL,
  `password` varchar(100) NOT NULL,
  `firstname` varchar(100) NOT NULL,
  `lastname` varchar(100) NOT NULL);
```

INSERT INTO `users` (`user_id`, `username`, `password`, `firstname`, `lastname`) VALUES

(2, 'admin', 'admin', 'john', 'smith');

Joins Queries:

1. **List all books with their corresponding categories:**
   SELECT book.book_title, category.classname
   FROM book
   JOIN category ON book.category_id = category.category_id;
2. **Retrieve the details of borrowed books along with member information:**
   SELECT book.book_title, borrowdetails.date_return, member.firstname, member.lastname
   FROM borrowdetails
   JOIN book ON borrowdetails.book_id = book.book_id
   JOIN borrow ON borrowdetails.borrow_id = borrow.borrow_id
   JOIN member ON borrow.member_id = member.member_id;
3. **Show the books that are currently borrowed:**

   SELECT book.book_title, borrowdetails.date_return
   FROM borrowdetails
   JOIN book ON borrowdetails.book_id = book.book_id
   WHERE borrowdetails.borrow_status = 'pending';
4. **List the members who borrowed books and their due dates:**
   SELECT member.firstname, member.lastname, borrow.due_date
   FROM borrow
   JOIN member ON borrow.member_id = member.member_id;
5. **Display the books that are borrowed but not yet returned:**
   SELECT book.book_title, borrowdetails.date_return
   FROM borrowdetails
   JOIN book ON borrowdetails.book_id = book.book_id
   WHERE borrowdetails.borrow_status = 'pending';

6. **Retrieve the lost books along with member information:**
   SELECT lost_book.Book_ID, book.book_title, member.firstname, member.lastname
   FROM lost_book
   JOIN book ON lost_book.ISBN = book.isbn
   JOIN member ON lost_book.Member_No = member.contact;
7. **List all books and their publishers:**

   SELECT book.book_title, book.publisher_name

FROM book;

8. **Show the members and the types of users they belong to:**
   SELECT member.firstname, member.lastname, type.borrowertype
   FROM member
   JOIN type ON member.type = type.borrowertype;
9. **Retrieve the details of borrowed books and their return status:**
   SELECT book.book_title, borrowdetails.date_return,
   borrowdetails.borrow_status
   FROM borrowdetails
   JOIN book ON borrowdetails.book_id = book.book_id;

10. **List the books and their authors:**
    SELECT book.book_title, book.author
    FROM book;

Subqueries Queries:

1. **Find the number of books in each category:**
   SELECT classname, (SELECT COUNT(*) FROM book WHERE category_id = category.category_id) as num_books
   FROM category;
2. **Retrieve the books borrowed by a specific member:**
   SELECT book.book_title
   FROM borrowdetails
   JOIN book ON borrowdetails.book_id = book.book_id
   WHERE borrowdetails.borrow_id = (SELECT borrow_id FROM borrow
   WHERE member_id = 55);
3. **Show the members who borrowed more than 2 books:**
   SELECT firstname, lastname
   FROM member
   WHERE member_id IN (SELECT member_id FROM borrow GROUP BY member_id HAVING COUNT(*) > 2);
4. **List the books with a copyright year greater than the average:**
   SELECT book_title, copyright_year
   FROM book
   WHERE copyright_year > (SELECT AVG(copyright_year) FROM book);
5. **Retrieve the books with the same publisher:**
   SELECT book_title, publisher_name
   FROM book
   WHERE publisher_name IN (SELECT publisher_name FROM book GROUP BY publisher_name HAVING COUNT(*) > 1);

6. **Show the books with the same ISBN but different titles:**
   SELECT book_title, isbn
   FROM book

WHERE isbn IN (SELECT isbn FROM book GROUP BY isbn HAVING COUNT(DISTINCT book_title) > 1);

7. **List the members who have borrowed books but are currently banned:**
   SELECT firstname, lastname
   FROM member
   WHERE member_id IN (SELECT member_id FROM borrow WHERE member_id IN (SELECT member_id FROM member WHERE status = 'Banned'));
8. **Retrieve books borrowed on a specific date:**
   SELECT book_title
   FROM borrowdetails
   JOIN book ON borrowdetails.book_id = book.book_id
   WHERE borrowdetails.borrow_id IN (SELECT borrow_id FROM borrow WHERE date_borrow = '2014-03-20 23:50:27');
9. **Show members who have borrowed books with a due date in the future:**
   SELECT firstname, lastname
   FROM member
   WHERE member_id IN (SELECT member_id FROM borrow WHERE due_date > NOW());
10. **List books with the same author as a specific book:**
    SELECT book_title, author
    FROM book
    WHERE author IN (SELECT author FROM book WHERE book_title = 'Algebra 1');

Double Joins Queries:

1. **List books along with their categories and publishers:**
   SELECT book.book_title, category.classname, book.publisher_name
   FROM book
   JOIN category ON book.category_id = category.category_id
   JOIN borrowdetails ON book.book_id = borrowdetails.book_id;
2. **Retrieve books along with their borrow details and member information:**
   SELECT book.book_title, borrowdetails.borrow_status, member.firstname, member.lastname
   FROM book
   JOIN borrowdetails ON book.book_id = borrowdetails.book_id
   JOIN borrow ON borrowdetails.borrow_id = borrow.borrow_id
   JOIN member ON borrow.member_id = member.member_id;
3. **Show lost books along with their member information and categories:**
   SELECT lost_book.Book_ID, book.book_title, member.firstname, member.lastname, category.classname
   FROM lost_book
   JOIN book ON lost_book.ISBN = book.isbn
   JOIN member ON lost_book.Member_No = member.contact
   JOIN category ON book.category_id = category.category_id;
4. **List members with their borrow details and types:**

```
SELECT member.firstname, member.lastname, borrowdetails.borrow_status,
type.borrowertype
FROM member
JOIN borrow ON member.member_id = borrow.member_id
JOIN borrowdetails ON borrow.borrow_id = borrowdetails.borrow_id
JOIN type ON member.type = type.borrowertype;
```

5. **Retrieve books along with their authors, categories, and publishers:**

```
SELECT book.book_title, book.author, category.classname,
book.publisher_name
FROM book
JOIN category ON book.category_id = category.category_id
JOIN borrowdetails ON book.book_id = borrowdetails.book_id;
```

Conclusion

- The library management system is essential for colleges, schools, and many more places these days. A lot of manual work can be reduced with this library management system. And also, a lot of glitches like wrong borrow date and miscalculation of fine amount are avoided. As it is a computer-managed system and so these are all avoided. It is also efficient and cost-effective. The Library management system stores the details of books and also details of persons. So overall, we have seen-

- To build a database to maintain all the related information
- We built tables separately to store data.
- Learned the purpose of the library management system.
- What features are required for students and librarians to use LMS?
- We have seen all the implementations using MYSQL and how the software allows storing all the details related to the library.
- Finally, we tested the final database.


THANK
you