



አዲስ አበባ
ካይንስ እና
ቴክኖሎጂ
ዩኒቨርሲቲ
ዩኒቨርሲቲ ለኢንዱስትሪ

ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING

DEPARTMENT OF SOFTWARE ENGINEERING

DATABASE SYSTEMS (SWE62108)

AASTU Library Management System DB

```
USE Section_A;
```

```
SELECT Name, ID_No  
FROM STUDENT  
WHERE Group = 1  
ORDER BY Name;
```

No.	Name	ID_No
1	Ararsa Derese	ETS0152/13
2	Birhanu Worku	ETS0279/13
3	Biruk Mesfin	ETS0290/13
4	Bisrat Kebere	ETS0306/13
5	Biyaol Mesay	ETS0309/13
6	Dagim Tezerawork	ETS0333/13

Submitted To:

Girma Neshir (Ph.D)

Submission Date: **August 12, 2022**

Table of Contents

Table of Contents	0
Introduction	1
Project Background	1
Description of Problem	1
The Current Working of the libraries data collecting method	2
Measuring the Benefits of Using Database System	2
Objective	2
Expected outcomes	2
Current Data Modeling	2
Conceptual Design	3
Basic Requirements	3
Entities and Attributes	3
Relationships and Cardinality	3
Assumptions	4
Entity Relationship Diagram	4
Logical Design	5
Constraints and Validations	5
Relational schema	5
Implementation	6
FAQ (Frequently Asked Queries)	7
Sample Trigger	7
Stored Procedure	7
Backup and Recovery Plan	8
Access levels and privileges	8
Security levels enforcement	8
Future Enhancements	9
Conclusion	9
Appendices	10
Appendix A - Detailed Logical Design	10
Appendix B - UML Relational schema	11
Appendix C – LMS Database State	12
Appendix D – Output table view for queries	13

Introduction

A prototype Library Management System is what this project seeks to create. This system's creation is specifically intended to handle library services and solve issues libraries encounter when managing their business. A computer-based library management system is currently not common in many libraries in this nation. Data loss, challenging data analysis, and the inability to access huge amounts of data all at once are just a few of the issues this will bring about. The Library of Addis Ababa Science and Technology University (AASTU) is also the victim of these problems of traditional data handling system.

The goal of this project is to develop a computerized system that will replace the manual data access. All information is currently kept in sizable dossiers that are housed in offices. This information can be combined using a computerized technique to create a database. We will feature a computerized system that guarantees data protection and accessibility for better management of AASTU Libraries. Additionally, the database will be assuring a book lend management to assist students to study with convenience.

Project Background

Information, data, and knowledge are actually more alive than ever in libraries. Within the chamber, there are countless books and databases where we may look for information to help us better ourselves and the world around us. Then their lives are confined inside the quarters on countless shelves. The database organization is done to make it simple for users to find the book they need easily and also borrow books from the libraries. Thus, in order to achieve so, it must be organized in the most efficient way possible to save time. The library serves the whole institution despite having few patrons and clients.

Description of Problem

But can this be possible with the current library management system? The records of the library have somewhat a file-based method, not simply the best formation for it. The book's data are compiled into PDF's that is difficult to be altered by CRUD (Create, Retrieve, Update and Delete) operations for simplicity. We were motivated to take this project because of these observed problems.

Modification of data is tough and might cause even extra risks, together with the provision of the identical data in numerous paperwork in several tables, book duplication, problems with updating, retrieving, and integrating books, etc. The prime database designs were meant to be time-saving. Due to capacity constraints, duplicate entries make it more difficult for the database to provide you with the information you need for reporting, transactions and other purposes. The database can only contain so much information before reaching its limit, since there is any duplication, the database would shrink significantly. The organization's carrying capacity is inadequate given the number of residents at the institution; other concerns include a lack of electricity and internet access, but these are more structural than database-related. The organizations system requirements might be lower than expected which can cause failure to do all the work required.

The Current Working of the libraries data collecting method

Commonly, the file-based system requires a separate application and program, the library has a program where it stores information concerning the institution. The books are organized in a simple listing with the books title, year, author, edition, publication, field of study and shelf number written and in order to search it requires you to state which field of study to pursue. To retrieve the data, it requires to ask for holding of the book but the file formats also can be incompatible. Deleting data is possible but since its separate program it only deletes the data from one program.

Measuring the Benefits of Using Database System

The organization's work will be carefully structured and planned. The database will appropriately store the data, enabling both the storing and retrieval of information. The suggested system will be more accurate overall. Every action would be conducted correctly, and it would guarantee the accuracy of any information coming from the center. Due to the aforementioned factors, the suggested system will be very reliable. The system will be more reliable since there will be better information storage going forward. Extreme care would be taken in the suggested system to ensure that no information is replicated anywhere, whether in storage or elsewhere. This would guarantee efficient storage space use and data consistency. The primary goal of the suggested approach is to enable an effective and speedy retrieval of information. Any kind of information would always be accessible to users.

Objective

We anticipate replacing the old manual system, which is file-based, with a computerized system and database, to guarantee that data access rights in the database are authenticated to a certain kind of user only, to begin and enhance the book borrow and lend management, to assist in conducting backup and recovery for the system, to ensure that the data is always available and can be restored in the case of a system crash, to reduce the time needed for the user to access and retrieve data, to develop a quick method for adding the data , and to ensure that the database is well protected.

Expected outcomes

After this system has been developed, it is expected that this system will make management run like a well-oiled machine. The procedures can be completed in a short amount of time with minimal effort. This will end those long and unpleasant queues at hospital lobbies. The usage of this system would curb the use of paperwork. This effectively means storage spaces will be free for library rather than tons of unusable paper.

Current Data Modeling

The selection and purchase of these items may occasionally be a part of the collection-building process, but more often than not, it involves the deliberate planning of the growth of the current collection. The library's budget, staffing levels, and space, as well as any policies the library has established about the types and numbers of clients it serves or the implied intents and objectives of the institutional setting.

Conceptual Design

Basic Requirements

- ✓ The University is organized into colleges and departments. Thus, the books in the library are also categorized based on these departments. Each Book has a unique ID number, Title, Publication Year, Author, Publisher, Edition and Shelf No, Number of copies found and Department it is categorized into.
- ✓ The main users of these library are students and teachers. The database will store each user's name, ID number, sex (gender), Department based on the information that can be found from registrar.
- ✓ The library is managed by librarians. Each librarian will have unique ID number, name, sex(gender) and email address stored on the database. They manage and approve the book transactions in the library.
- ✓ As the main purpose of the database, the database also stores each borrow transaction having a unique ID number, borrower's ID, Book ID, the borrow date and return date as well as returns status. A user can borrow one or more books. It is required to keep track of the lent books if they are returned or not.

Entities and Attributes

We can identify the following five entity types, one corresponding to each of the items in the basic requirement specification:

1. An entity type **BOOK** with attributes Book ID, Title, Publication Year, Authors, Department, Publisher, Edition, Shelf No and Number of Copies. Authors is the only multivalued attribute. We can specify that Book ID is key attribute since it is unique for each.
2. An entity type **DEPARTMENT** with attributes Name and Number. Both being unique.
3. An entity type **USER** with attributes ID, Name, Sex, Email Address and Department. Both Name and ID Number are (separate) key attributes.
4. An entity type **LIBRARIAN** with attributes ID, Name, Sex and Email Address. ID Number is a key attribute.
5. An entity type **BORROW** will be derived from the M:N (**many-to-many**) between **USER** and **BOOKS**. It contains the attributes unique borrow ID, borrower ID, borrow Date, librarian ID, return Date and return Status.

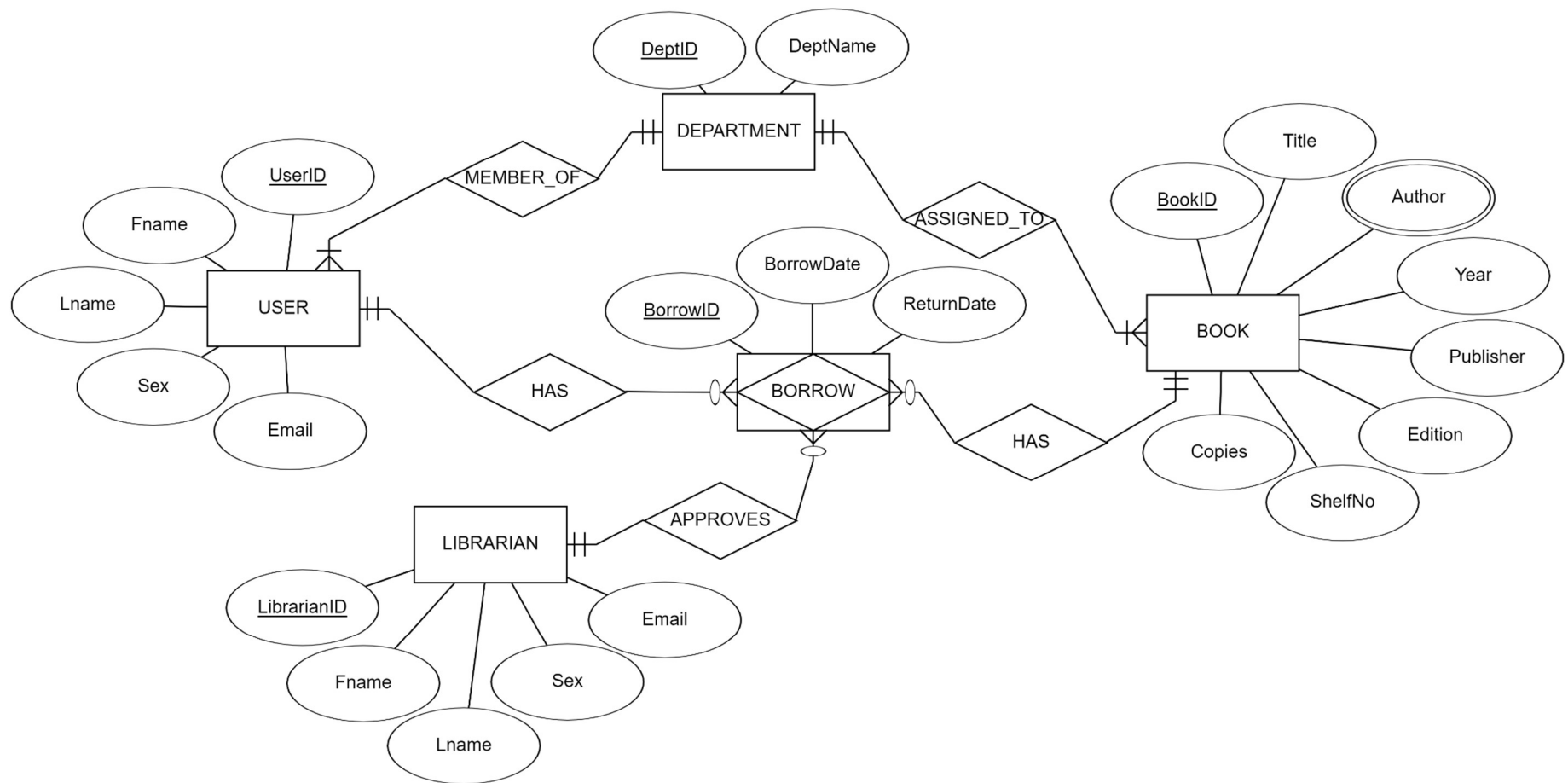
Relationships and Cardinality

- ∞ **BORROW**, a M:N (many-to-many) relationship type between **USER** and **BOOK**. Both participations are partial since a user may or may not borrow a book and a book can also be lent or not.
- ∞ **APPROVES**, 1:N (one-to-many) relationship type between **LIBRARIAN** and **BORROW**. The participation of **BORROW** is total.
- ∞ **ASSIGNED_TO**, 1:N (one-to-many) relationship type between **DEPARTMENT** and **BOOK**. Both participations are total.
- ∞ **MEMBER_OF**, 1:N (one-to-many) relationship type between **DEPARTMENT** and **USER**. Both participations are total.

Assumptions

- A book can have multiple Authors.
- There may be more than one copy of a book available in the library.
- A user can borrow more than one book.
- Every book in the library is assigned to a department.

Entity Relationship Diagram



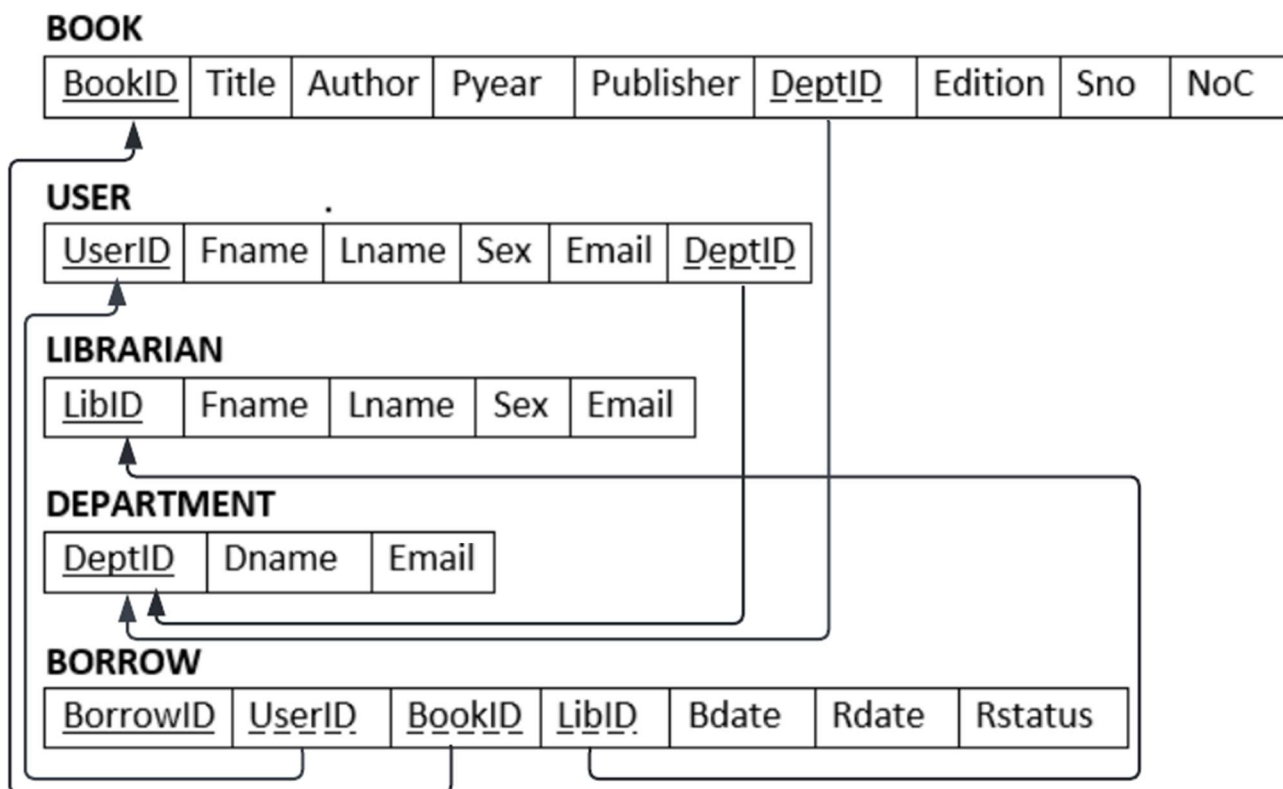
Logical Design

Constraints and Validations

- Primary keys, foreign keys, and name entries cannot be NULL
- The value for sex will only be either 'F' or 'M' for users as well as librarians.
- The value for Return Status (Rstatus) should be either 'Lent' or 'Returned'
- Email address of each user and librarian should be unique
- BorrowID will be an autogenerated and auto incrementing integer value
- Other primary key ID's will have a character value of less than 10 chars
- No of copies(NoC) and Edition of a book cannot be a negative value, only positive integer
- Return Date (Rdate) should be after Borrow Date (Bdate)
- On Update, all the foreign keys will also cascade with the changes made.
- However, Deletion of the values for BookID and UserID is restricted since these values are important borrow information for report.

➤ Detailed descriptions of the logical design are provided on [Appendix A.](#)

Relational schema



➤ UML relational schema is provided on [Appendix B.](#)

Implementation

```
DROP DATABASE IF EXISTS LMS;
CREATE DATABASE LMS;
USE LMS;

CREATE TABLE user(
    UserID char(10) PRIMARY KEY, Fname varchar(15) NOT NULL, Lname varchar(15) NOT NULL,
    Sex varchar(1), CHECK (Sex in ('F', 'M')), DeptID char(10), Email varchar(50) UNIQUE);

CREATE TABLE book( BookID char(10) PRIMARY KEY, Title varchar(100) NOT NULL,
    Author varchar(50), Pyear int, Publisher varchar(50), DeptID char(10),
    Edition tinyint, CHECK (Edition > 0), Sno varchar(5), NoC int , CHECK (NoC >= 0) );

CREATE TABLE librarian(
    LibID char(10) PRIMARY KEY, Fname varchar(15) NOT NULL, Lname varchar(15) NOT NULL,
    Sex varchar(1), CHECK (Sex IN ('F', 'M')), Email varchar(50) UNIQUE );

CREATE TABLE department(
    DeptID char(10) PRIMARY KEY, Dname varchar(50) UNIQUE NOT NULL, Email varchar(50) UNIQUE );

CREATE TABLE borrow( BorrowID int AUTO_INCREMENT PRIMARY KEY, UserID char(10), BookID char(10),
    LibID char(10), Bdate date NOT NULL, Rdate date NOT NULL, Rstatus varchar(15),
    CHECK (Rstatus IN ('Lent', 'Returned')), CHECK (Rdate > Bdate),
    FOREIGN KEY(UserID) REFERENCES user(UserID) ON DELETE RESTRICT ON UPDATE CASCADE,
    FOREIGN KEY(BookID) REFERENCES book(BookID) ON DELETE RESTRICT ON UPDATE CASCADE,
    FOREIGN KEY(LibID) REFERENCES librarian(LibID) ON DELETE CASCADE ON UPDATE CASCADE);

ALTER TABLE book ADD FOREIGN KEY(DeptID)
REFERENCES department(DeptID) ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE user ADD FOREIGN KEY(DeptID)
REFERENCES department(DeptID) ON DELETE SET NULL ON UPDATE CASCADE;

INSERT INTO department VALUES ('D00', 'Freshman', NULL),
('D01', 'Software Engineering', 'sweg@aastu.edu.et'), ('D02', 'Electrical Engineering', 'ele@aa
stu.edu.et'),('D03', 'Mechanical Engineering', 'mec@aastu.edu.et'), ('D04', 'Architecture',
'arch@aastu.edu.et'),('D05', 'Chemical Engineering', 'chem@aastu.edu.et'),
('D06', 'Civil Engineering', 'civil@aastu.edu.et');

INSERT INTO user VALUES
('ETS0306/13', 'Bisrat', 'Kebere', 'M', 'D01', 'bisrat.kebere@aastustudent.edu.et'),
('ETS0309/13', 'Biyaol', 'Mesay', 'M', 'D03', 'biyaol.mesay@aastustudent.edu.et'),
('ETS0152/13', 'Ararsa', 'Derese', 'M', 'D02', 'ararsa.derese@aastustudent.edu.et'),
('ETS0333/13', 'Dagim', 'Tezerawork', 'M', 'D04', 'dagim.tezerawork@aastustudent.edu.et'),
('ETS0290/13', 'Biruk', 'Mesfin', 'M', 'D01', 'biruk.mesfin@aastustudent.edu.et'),
('ETS0279/13', 'Birhanu', 'Worku', 'M', 'D01', 'birhanu.worku@aastustudent.edu.et'),
('ETS1306/13', 'Yeshi', 'Afewerk', 'F', 'D05', 'yeshi.afewerk@aastustudent.edu.et');

INSERT INTO book VALUES
('B0001', 'Fundamentals of Database Systems', 'Ramez Elmasri, Shamkant B. Navathe', 2016, 'Pear
son Education', 'D01', 7, 'S01', 13),('B0002', 'Fundamentals of Sustainability in Civil Enginee
ring', 'Andrew Braham', 1977, 'McGrawHill Education', 'D06', 2, 'S06', 23),('B0003', 'Distillat
ion Design', 'Henry Kister', 2000, 'Cengage', 'D05', 3, 'S05', 10),('B0004', 'Standard Handboo
k of Machine Design', 'Joseph E. Shigley and Charles R. Mischke', 2017, 'Springer Nature', 'D03
', 12, 'S03', 8),('B0005', 'The Architecture of the City', 'Aldo Rossi', 1999, 'John Wiley & So
ns', 'D04', 1, 'S04', 15),('B0006', 'Clean Code', 'Robert C. Martin', 2012, 'Penguin Random Hou
se', 'D01', 5, 'S01', 5);
```



```
INSERT INTO librarian VALUES ('L01', 'Aberash', 'Wegayewu', 'M', 'aberash@aastu.edu.et'),
('L02', 'Bereket', 'Dalebo', 'M', 'bekibek13@yahoo.com'), ('L03', 'Gelete', 'Umama', 'F', 'gelete.umama@aastu.edu.et'), ('L04', 'Fatuma', 'Gulilat', 'F', 'fatumaG@gmail.com'), ('L05', 'Hagos', 'Teklay', 'M', 'hagos teklay4@gmail.com'), ('L06', 'Abdulaziz', 'Mohamed', 'M', 'abdulaziz.mohamed@aastu.edu.et');
```

```
INSERT INTO borrow (UserID, BookID, LibID, Bdate, Rdate, Rstatus)
VALUES ('ETS0306/13', 'B0006', 'L01', '1973-07-22', '1973-07-28', 'Lent'),
('ETS0309/13', 'B0002', 'L03', '2022-02-22', '2022-03-03', 'Lent'),
('ETS0279/13', 'B0003', 'L04', '2021-07-12', '2021-07-18', 'Returned'),
('ETS0309/13', 'B0001', 'L06', '2022-06-20', '2022-07-01', 'Lent'),
('ETS0152/13', 'B0004', 'L02', '2022-04-10', '2022-04-11', 'Lent'),
('ETS0152/13', 'B0005', 'L02', '2022-04-10', '2022-04-11', 'Lent');
UPDATE borrow SET Rstatus = 'Returned' WHERE BorrowID = 1 OR BorrowID = 4;
```

➤ Database state view on [Appendix C.](#)

FAQ (Frequently Asked Queries)

```
SELECT * FROM user; -- Show list of all users
-- Search for a book by title and Department
SELECT * FROM book WHERE Title LIKE 'Fundamental%' AND DeptID = 'D06';
-- Search for a book by Author
SELECT * FROM book WHERE Author LIKE '%Elmasri%';
-- Show list of unreturned books with borrower info and book title
SELECT b.UserID 'Student ID', Fname 'First Name', Lname 'Last Name', Dname Department, bo.Title, Bdate
'Borrowed on', Rdate 'Return on', u.Email FROM borrow as b join user as u on b.UserID = u.UserID
join book as c on c.BookID = b.BookID join department as d on d.DeptID = u.DeptID
WHERE Rstatus = 'Lent' ORDER BY Bdate;
-- No of books Borrowed and returned
SELECT count(BorrowID) FROM borrow WHERE Rstatus = 'Returned';
-- The Title of the books with more than 10 copies in the library
SELECT Title FROM book WHERE NoC > 10;
-- Total number of Book copies in the library
SELECT SUM(NoC) FROM book;
```

Sample Trigger

```
-- Sample Trigger to decrease the number of copies of a book after a borrow
DELIMITER $$
CREATE TRIGGER modify_book_count AFTER INSERT ON borrow FOR EACH ROW BEGIN
UPDATE book as c SET NoC = NoC - 1 WHERE c.BookID = NEW.BookID; END$$
DELIMITER ;
```

Stored Procedure

```
-- Stored Procedure to get 1.all borrows of a user, 2.all borrows of a book,
-- 3.unreturned borrows of a user 4. All unreturned borrows
DELIMITER $$
CREATE PROCEDURE get_query(c int, search char(10)) BEGIN IF c = 1 THEN SELECT BorrowID, Fname, BookID,
Rstatus FROM user as u JOIN borrow AS b ON b.UserID = u.UserID WHERE b.UserID = search;
ELSEIF c = 2 THEN SELECT BorrowID, Title, UserID, Rdate, Rstatus FROM book as c
JOIN borrow AS b ON b.BookID = c.BookID WHERE b.BookID = search;
ELSEIF c = 3 THEN SELECT BorrowID, u.UserID, Fname, BookID FROM borrow as b
JOIN user AS u ON b.UserID = u.UserID WHERE b.UserID = Search and Rstatus = 'Lent';
ELSE SELECT * FROM borrow WHERE Rstatus = 'Lent'; END IF; END$$
DELIMITER ;
CALL get_query(1, 'ETS0309/13'); -- Sample call to query all borrows of a particular user
CALL get_query(3, 'ETS0309/13'); -- Sample call to query unreturned borrows of a particular user
```

➤ Output table view for queries and procedures on [Appendix D.](#)

Backup and Recovery Plan

Data backup and recovery methods give you peace of mind. Protecting our system and database against data loss caused by human error, equipment failure and cyberattacks can prevent us from experiencing an unrecoverable data failure.

The database will be backed up with level 2 hybrid cloud backups which is the combination of both local and cloud backups. Storing data in multiple locations ensures that you always have a local copy, as well as a copy stored securely offsite.

A differential backup method will be implemented in order to backup the backbone database once then continuously update the data that has been created, updated or altered in any way and does not copy all of the data every time. Data Recovery will be done from the local or cloud storage in case of data loss.

Access levels and privileges

Different access levels are enforced in order to prevent data breach and loss due to unauthorized use of privileges. The minimum amount of access and privilege needed to achieve the intended work will be granted to users of the database. For example, librarians will only be granted to insert, update and delete the books and borrow table rather than being given the privilege to add and delete on all database or tables. Even database administrators will have a guided and limited amount privilege on database and tables.

Security levels enforcement

Data security is important to protect the data in an organization from unauthorized access, malicious attacks, or phishing activities. Security breaches may come from the data, user-level, and system level.

Data at-rest encryption methods will be used to avoid any tampering of the data. Encryption transforms the data to another form and only authorized access can decode it. If the backup falls in the wrong hands, sensitive data is still protected based on the encryption when the backup was done.

Multiple-factor authentication will be used for user-level security not only for the database but for the database backups as well. A good combination of user login (username and strong password) combined with security codes from mobiles is a good way to verify the access as a means to data protection.

System-Level Database security at the hardware, network and communication lines level which can also pose a threat to the database backups. To avoid this, there should be limited access and protection such as a firewall on the servers wherever the backups are stored. This minimizes the threat to backup servers or backup hosting via system-level threat.

Future Enhancements

The database system is open for modification and enhancements to meet future needs such as developing a website or mobile application to make better use of it. For example, we are planning to use this database on our [C++ project](#) that gives users the access to check books and also to request borrow remotely. Thus, using this database on the project will enhance the project capability.

Modifications such as additional column or attribute to a table, a password column to the users and librarians table can be an example or additional table to administer fee and reimbursement for lost and destroyed books during the borrow. Such modifications can easily be done by additional snippets of code without affecting the integrity of the database.

➤ Link to [AASTU LMS](#) C++ Project DEMO simulator. <https://bit.ly/3An90YE>

Conclusion

Library is a collection of organized information and resources which is made accessible to a well-defined community for borrowing or reference. The main aim of this system was to develop a new programmed system that will convey ever lasting solution to the manual base operation and to make available a channel through which staff can maintain the records easily and customers can access the information about the easily by eliminating the time-consuming manual process.

LMS allows the user to store the book details and the customer details. The implementation of the system in the organization will considerably reduce data entry, time and also provide readily calculated reports.

Even if developing and implementing such database is costly and somehow difficult, the outcome is remarkable since it reduces wastage of the most important resource of human beings, time. Also, this database helps students to have access to hardcopy books through the borrowing system.

Appendices

Appendix A - Detailed Logical Design

BOOK

Attribute	Abbreviation	Data type	Field size	Constraint	Validation
Book ID No	BookID	varchar	15	primary key	Unique
Title	Title	varchar	100	Single-valued	not null
Author	Author	varchar	50	Multi-valued	Default null
Year Published	Pyear	Int	4	Single-valued	Default null
Publisher	Publisher	varchar	50	Single-valued	Default null
Department ID No	DeptID	char	15	foreign key	not null
Edition	Edition	tinyint	2	Single-valued	Default null
Shelf No	Sno	varchar	5	Single-valued	not null
No of Copies	NoC	int	4	Single-valued	not null

USER

Attribute	Abbreviation	Data type	Field size	Constraint	Validation
User ID No	UserID	char	10	primary key	Unique
First Name	Fname	varchar	15	Single-valued	not null
Last Name	Lname	varchar	15	Single-valued	not null
Sex	Sex	char	1	'M' or 'F'	not null
Email Address	Email	varchar	50	Single-valued	Unique
Department ID No	DeptID	char	10	foreign key	not null

LIBRARIAN

Attribute	Abbreviation	Data type	Field size	Constraint	Validation
Librarian ID No	LibID	char	10	primary key	Unique
First Name	Fname	varchar	15	Single-valued	not null
Last Name	Lname	varchar	15	Single-valued	not null
Sex	Sex	char	1	'M' or 'F'	not null
Email Address	Email	varchar	50	Single-valued	Unique

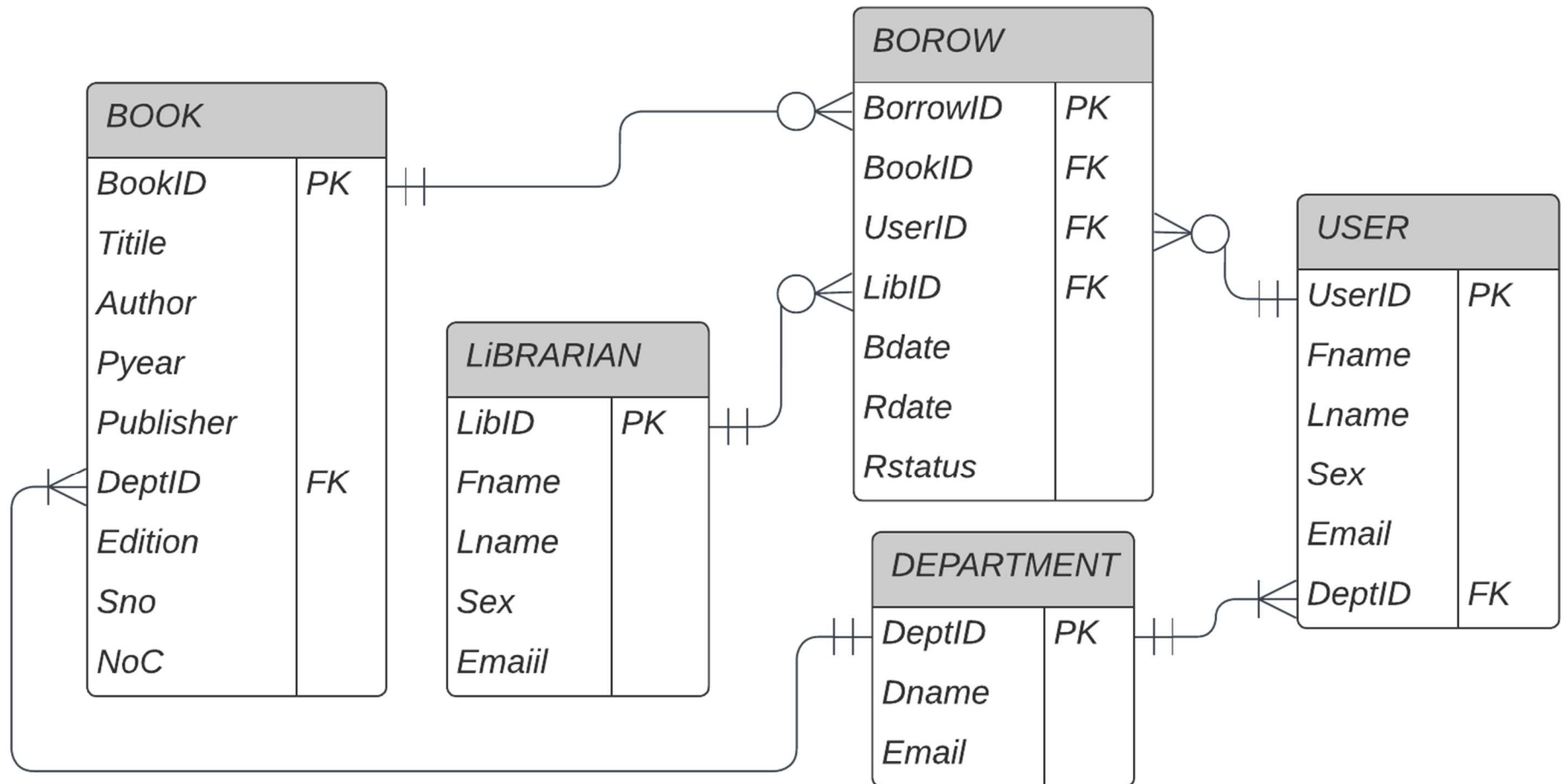
DEPARTMENT

Attribute	Abbreviation	Data type	Field size	Constraint	Validation
Department ID No	DeptID	char	10	primary key	Unique
Department Name	Dname	varchar	15	Single-valued	not null, unique
Email Address	Email	varchar	50	Single-valued	Unique

BORROW

Attribute	Abbreviation	Data type	Field size	Constraint	Validation
Borrow ID No	BorrowID	int	auto increment	primary key, autogenerated	Unique
Borrower ID No	UserID	char	10	foreign key	not null
Book ID No	BookID	char	10	foreign key	not null
Librarian ID No	LibID	char	10	foreign key	not null
Borrow Date	Bdate	Date		Single-valued	not null
Return Date	Rdate	Date		Single-valued	not null
Return Status	Rstatus	varchar	15	'Lent' or 'Returned'	not null

Appendix B - UML Relational schema



Appendix C – LMS Database State

BOOK

<u>BookID</u>	<u>Title</u>	<u>Author</u>	<u>Pyear</u>	<u>Publisher</u>	<u>DeptID</u>	<u>Edition</u>	<u>Sno</u>	<u>NoC</u>
B0001	Fundamentals of Database Systems	Ramez Elmasri, Shamkant B. Navathe	2016	Pearson Education	D01	7	S01	13
B0002	Fundamentals of Sustainability in Civil Engineering	Andrew Braham	1977	McGraw-Hill Education	D06	2	S06	23
B0003	Distillation Design	Henry Kister	2000	Cengage	D05	3	S05	10
B0004	Standard Handbook of Machine Design	Joseph E. Shigley and Charles R. Mischke	2017	Springer Nature	D03	12	S03	8
B0005	The Architecture of the City	Aldo Rossi	1999	John Wiley & S.	D04	1	S04	15
B0006	Clean Code	Robert C. Martin	2012	Penguin R.H	D01	5	S01	5

USER

<u>UserID</u>	<u>Fname</u>	<u>Lname</u>	<u>Sex</u>	<u>Email</u>	<u>DeptID</u>
ETS0152/13	Ararsa	Dereese	M	ararsa.dereese@aastustudent.edu.et	D02
ETS0279/13	Birhanu	Worku	M	birhanu.worku@aastustudent.edu.et	D01
ETS0290/13	Biruk	Mesfin	M	biruk.mesfin@aastustudent.edu.et	D01
ETS0306/13	Bisrat	Kebere	M	bisrat.kebere@aastustudent.edu.et	D01
ETS0309/13	Biyaol	Mesay	M	biyaol.mesay@aastustudent.edu.et	D03
ETS0333/13	Dagim	Tezerawork	M	dagim.tezerawork@aastustudent.edu.et	D04
ETS1306/13	Yeshi	Afewerk	F	yeshi.afewerk@aastustudent.edu.et	D05

LIBRARIAN

<u>LibID</u>	<u>Fname</u>	<u>Lname</u>	<u>Sex</u>	<u>Email</u>
L01	Aberash	Wegayewu	M	aberash@aastu.edu.et
L02	Bereket	Dalebo	M	bekibek13@yahoo.com
L03	Gelete	Umama	F	gelete.umama@aastu.edu.et
L04	Fatuma	Gulilat	F	fatumaG@gmail.com
L05	Hagos	Teklay	M	hagosteklay4@gmail.com

DEPARTMENT

<u>DeptID</u>	<u>Dname</u>	<u>Email</u>
D00	Freshman	NULL
D01	Software Engineering	sweg@aastu.edu.et
D02	Electrical Engineering	ele@aastu.edu.et
D03	Mechanical Engineering	mec@aastu.edu.et
D04	Architecture	arch@aastu.edu.et
D05	Chemical Engineering	chem@aastu.edu.et
D06	Civil Engineering	civil@aastu.edu.et

BORROW

<u>BorrowID</u>	<u>UserID</u>	<u>BookID</u>	<u>LibID</u>	<u>Bdate</u>	<u>Rdate</u>	<u>Rstatus</u>
1	ETS0306/13	B0006	L01	1973-07-22	1973-07-28	Returned
2	ETS0309/13	B0002	L03	2022-02-22	2022-03-03	Lent
3	ETS0279/13	B0003	L04	2021-07-12	2021-07-18	Returned
4	ETS0309/13	B0001	L01	2022-06-20	2022-07-01	Returned
5	ETS0152/13	B0004	L02	2022-04-10	2022-04-11	Lent
6	ETS0152/13	B0005	L02	2022-04-10	2022-04-11	Lent

Appendix D – Output table view for queries

-- Search for a book by title and Department

```
SELECT * FROM book WHERE Title LIKE 'Fundamental%' AND DeptID = 'D06';
```

BookID	Title	Author	Pyear	Publisher	DeptID	Edition	Sno	NoC
B0002	Fundamentals of Sustainability in Civil Engineering	Andrew Braham	1977	McGraw-Hill Education	D06	2	S06	23

-- Show list of unreturned books with borrower info and book title

BookID	Title	Author	Pyear	Publisher	DeptID	Edition	Sno	NoC
B0001	Fundamentals of Database Systems	Ramez Elmasri, Shamkant B. Navathe	2016	Pearson Education	D01	7	S01	13

-- Show list of unreturned books with borrower info and book title

BookID	Title	Author	Pyear	Publisher	DeptID	Edition	Sno
ETS0309/13	Biyaol	Mesay	Mechanical Engineering	Fundamentals of Sustainability in Civil Engineering	2022-02-22	2022-03-03	biyaol.mesay@aastustudent.edu.et
ETS0152/13	Ararsa	Derese	Electrical Engineering	Standard Handbook of Machine Design	2022-04-10	2022-04-11	ararsa.derese@aastustudent.edu.et
ETS0152/13	Ararsa	Derese	Electrical Engineering	The Architecture of the City	2022-04-10	2022-04-11	ararsa.derese@aastustudent.edu.et

-- No of books Borrowed and returned

```
SELECT count(BorrowID) FROM borrow WHERE Rstatus = 'Returned';
```

Count(BorrowID)
3

-- The Title of the books with more than 10 copies in the library

```
SELECT Title FROM book WHERE NoC > 10;
```

Title
Fundamentals of Database Systems
Fundamentals of Sustainability in Civil Engineering
The Architecture of the City

-- Total number of Book copies in the library

```
SELECT SUM(NoC) FROM book;
```

Sum(BoC)
74

-- Stored Procedure to get all borrows of a particular user

```
CALL get_query(1, 'ETS0309/13');
```

BorrowID	Fname	BookId	Rstatus
2	Biyaol	B0002	Lent
4	Biyaol	B0001	Returned

-- Stored Procedure to get unreturned borrows of a user

```
CALL get_query(1, 'ETS0309/13');
```

BorrowID	Fname	BookId	Rstatus
2	ETS0309/13	Biyaol	B0002