**[ONLINE LIBRARY MANAGEMENT SYSTEM (DBMS)]**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

Aparajita Kundu(AP22110010247)

Darbha Sudha Vaishnavi(AP22110010201)

Ayush Kumar Mishra(AP22110010202)

**A picture containing text

Description automatically generated**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**[May, 2024]**

INDEX

* Abstract, Aim, Attributes, Entity……..3-4
* Realations……..5-6
* ER-Model Diagram……..7
* Relational Diagram……..8
* Normalization……..9-12
* Entity records……..13-21
* QUERIES AND VIEWS……..21-25

**Abstract**: The Library Management System is a database-driven software solution designed to streamline library operations. It manages books, borrowers, and loans efficiently. With features like book tracking, borrower management, and loan monitoring, it simplifies library tasks. The system includes stored procedures for generating reports and views for easy data retrieval. It enhances library efficiency by automating processes and providing a user-friendly interface. Overall, the Library Management System optimizes library management, ensuring smooth operations and improved user experience.

**Aim**: The aim is to develop a comprehensive Library Management System for efficient organization and tracking of books, borrowers, and loan transactions.

**Entity:** a “thing” or “object” in the enterprise that is distinguishable from other objects

The database comprises several entities essential for a Library Management System:

1. Publisher: Stores information about book publishers, including their name, address, and contact details.

2. Book: Represents individual books with attributes such as title and the name of the publisher. Each book is associated with a publisher through a foreign key relationship.

3. Library Branch: Contains details about different library branches, including their names and addresses.

4. Borrower: Stores information about library borrowers, including their name, address, and contact information.

5. Book Loans: Tracks loan transactions, including the book borrowed, the branch from which it was borrowed, the borrower, the date of loan, and the due date.

6. Book Copies: Records the number of copies available for each book at each library branch. This entity helps manage the inventory of books across different branches.

These entities collectively facilitate the organization, management, and tracking of library resources, borrowers, and loan activities within the system.

**Attributes** — characteristics of an entity and has an oval symbol.

There are different types of attributes:

* ***Key attribute:*** An attribute uniquely distinguishes the entity in an entity set.
* ***Simple attribute:*** An attribute that cannot be further subdivided into components.
* ***Composite attribute:*** An attribute that can be split into components.
* ***Single-valued attribute:*** The attribute which takes up only a single value for each entity instance.
* ***Multi-valued attribute:*** The attribute which takes up more than a single value for each entity instance.
* ***Stored attribute:*** An attribute that stores the data which can be used to get the derived attribute.
* ***Derived attribute:*** An attribute that can be derived from other attributes.

Sure, here are the attributes for each entity in the database:

1. \*\*Publisher\*\*:

- publisher\_PublisherName (Primary Key): Name of the publisher.

- publisher\_PublisherAddress: Address of the publisher.

- publisher\_PublisherPhone: Phone number of the publisher.

2. \*\*Book\*\*:

- book\_BookID (Primary Key): Unique identifier for each book.

- book\_Title: Title of the book.

- book\_PublisherName (Foreign Key): Name of the publisher associated with the book.

3. \*\*Library Branch\*\*:

- library\_branch\_BranchID (Primary Key): Unique identifier for each library branch.

- library\_branch\_BranchName: Name of the library branch.

- library\_branch\_BranchAddress: Address of the library branch.

4. \*\*Borrower\*\*:

- borrower\_CardNo (Primary Key): Unique identifier for each borrower.

- borrower\_BorrowerName: Name of the borrower.

- borrower\_BorrowerAddress: Address of the borrower.

- borrower\_BorrowerPhone: Phone number of the borrower.

5. \*\*Book Loans\*\*:

- book\_loans\_LoansID (Primary Key): Unique identifier for each loan.

- book\_loans\_BookID (Foreign Key): Identifier of the borrowed book.

- book\_loans\_BranchID (Foreign Key): Identifier of the library branch from which the book was borrowed.

- book\_loans\_CardNo (Foreign Key): Identifier of the borrower.

- book\_loans\_DateOut: Date when the book was borrowed.

- book\_loans\_DueDate: Due date for returning the book.

6. \*\*Book Copies\*\*:

- book\_copies\_CopiesID (Primary Key): Unique identifier for each copy.

- book\_copies\_BookID (Foreign Key): Identifier of the book.

- book\_copies\_BranchID (Foreign Key): Identifier of the library branch where the copy is available.

- book\_copies\_No\_Of\_Copies: Number of copies available at the branch.

Categorizing the different types of attribute found in the schema:

the classification of attributes based on the provided categories:

1. \*\*Key attribute\*\*:

- publisher\_PublisherName (Publisher)

- book\_BookID (Book)

- library\_branch\_BranchID (Library Branch)

- borrower\_CardNo (Borrower)

- book\_loans\_LoansID (Book Loans)

- book\_copies\_CopiesID (Book Copies)

2. \*\*Simple attribute\*\*:

- publisher\_PublisherAddress (Publisher)

- publisher\_PublisherPhone (Publisher)

- book\_Title (Book)

- library\_branch\_BranchName (Library Branch)

- library\_branch\_BranchAddress (Library Branch)

- borrower\_BorrowerName (Borrower)

- borrower\_BorrowerAddress (Borrower)

- borrower\_BorrowerPhone (Borrower)

- book\_loans\_DateOut (Book Loans)

- book\_loans\_DueDate (Book Loans)

- book\_copies\_No\_Of\_Copies (Book Copies)

3. \*\*Composite attribute\*\*:

- None in the provided schema.

4. \*\*Single-valued attribute\*\*:

- publisher\_PublisherName (Publisher)

- publisher\_PublisherAddress (Publisher)

- publisher\_PublisherPhone (Publisher)

- book\_BookID (Book)

- book\_Title (Book)

- library\_branch\_BranchID (Library Branch)

- library\_branch\_BranchName (Library Branch)

- library\_branch\_BranchAddress (Library Branch)

- borrower\_CardNo (Borrower)

- borrower\_BorrowerName (Borrower)

- borrower\_BorrowerAddress (Borrower)

- borrower\_BorrowerPhone (Borrower)

- book\_loans\_LoansID (Book Loans)

- book\_loans\_BookID (Book Loans)

- book\_loans\_BranchID (Book Loans)

- book\_loans\_CardNo (Book Loans)

- book\_loans\_DateOut (Book Loans)

- book\_loans\_DueDate (Book Loans)

- book\_copies\_CopiesID (Book Copies)

- book\_copies\_BookID (Book Copies)

- book\_copies\_BranchID (Book Copies)

- book\_copies\_No\_Of\_Copies (Book Copies)

5. \*\*Multi-valued attribute\*\*:

- None in the provided schema.

6. \*\*Stored attribute\*\*:

- All attributes in the schema are stored attributes since they directly store data.

7. \*\*Derived attribute\*\*:

- There are no explicitly defined derived attributes in the provided schema. Derived attributes are usually calculated or derived from other attributes in the database, but there are none specified in this schema.

**Relationship:**

A relationship is an association among several entities.

In the Library Management System:

* Books published by Publisher: Each book is published by a publisher.
* Books available at Library Branch: Books are available in multiple copies at different library branches.
* Borrowers borrowing Books: Borrowers borrow books from library branches.
* Book Loans associated with Borrowers, Books, and Library Branches: Each loan is associated with a specific book, borrower, and library branch.
* Borrowers placing Orders for Books: Borrowers request to borrow books from library branches.
* Library Branches managing Orders for Books: Library branches handle orders placed by borrowers for books.
* Customers (Borrowers) paying Late Fees: Customers pay late fees for overdue books.
* Late Fees paid by Customers (Borrowers): Customers settle late fees for overdue book loans.
* Library has multiple Branches: The library consists of multiple branches.
* Branches exist within the library: Branches are part of the library's infrastructure.

**Relations:**

The relationships entail entities interacting within the system, such as items being created, owned, or managed by other entities, transactions occurring between entities, and entities being associated or linked with each other based on specific criteria or events. The relationships in the Library Management System involve books being published by publishers, available in multiple copies across library branches, borrowed by readers, with each loan associated with specific books, borrowers, and branches.

1. Publisher PUBLISHES Book: One Publisher creates many Books.

Cardinality :(One-to-Many)

Publishes

PUBLISHER

BOOKS

1 M

2. Library Branch HOLDS Book Copies : One Library Branch holds many Book Copies.

Cardinality:(One-to-Many)

HOLDS

BOOK COPIES

LIBRARY BRANCH

1 M

3.Book is WRITTEN BY Book Authors: One Book is written by many Authors.

Cardinality:(One-to-Many)

BOOK

AUTHORS

WRITTEN

BOOK

1 M

4. Borrower TAKES Book Loans: One Borrower takes many Book Loans.

Cardinality: (One-to-Many)

BORROWS

BOOK

LOANS

BORROWER

1 M

5. Book is LOANED OUT IN Book Loans: One Book is loaned out in many Book Loans.

Cardinality:(One-to-One)

LOANED OUT

BOOK

BOOK LOAN

1 1

6. Library Branch ISSUES Book Loans: One Library Branch issues many Book Loans.

Cardinality:(One-to-Many)

BOOK LOANS

ISSUES

LIBRARY

1 M

12. \*\*Book HAS Book Copies\*\*: One Book has many copies.

Cardinality: (One-to-Many)

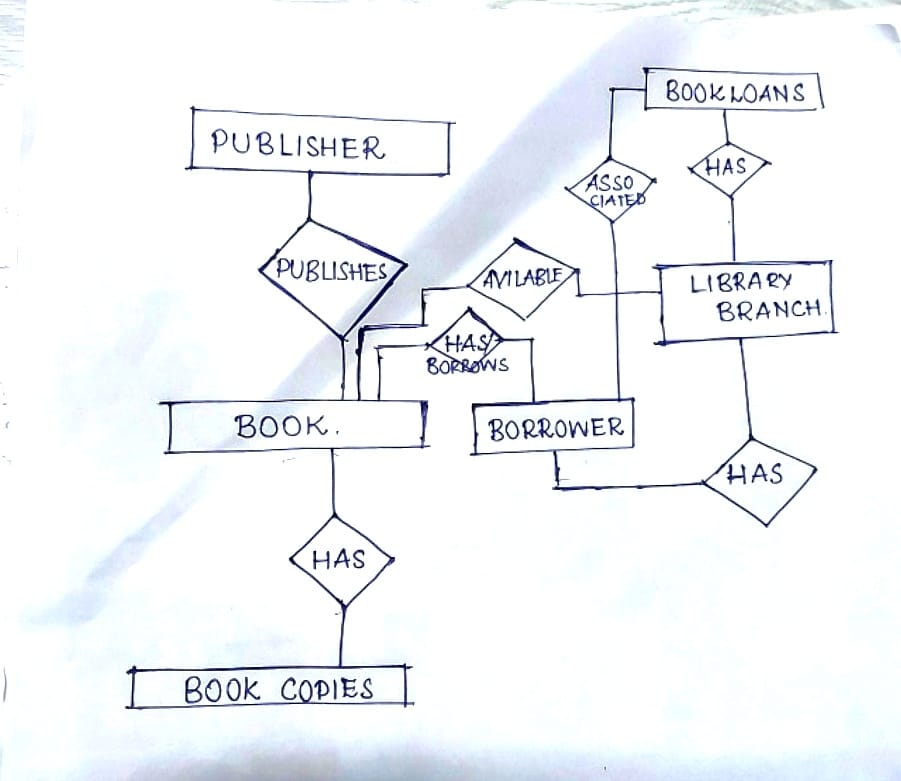
HAS

COPIES

OF THE BOOK

BOOK

1 M

****

**E-R Model:**

ER model stands for an Entity-Relationship model.

It is a high-level data model. This model is used to define the data elements and relationships for a specified system.It develops a conceptual design for the database. It also develops a very simple and easy-to-design view of data.

A diagram of a flowchart

Description automatically generated

**Relational Diagram**:

Converting ER model to tables/relations, commonly used, flexible.

Each and every column header is called an attribute. The row header is called a tuple.

Publisher (publisher\_PublisherName [PK], publisher\_PublisherAddress, publisher\_PublisherPhone)

Book (

book\_BookID [PK],

book\_Title,

book\_PublisherName [FK]

)

LibraryBranch (

library\_branch\_BranchID [PK],

library\_branch\_BranchName,

library\_branch\_BranchAddress

)

Borrower (

borrower\_CardNo [PK],

borrower\_BorrowerName,

borrower\_BorrowerAddress,

borrower\_BorrowerPhone

)

BookLoans (

book\_loans\_LoansID [PK],

book\_loans\_BookID [FK],

book\_loans\_BranchID [FK],

book\_loans\_CardNo [FK],

book\_loans\_DateOut,

book\_loans\_DueDate

)

BookCopies (

book\_copies\_CopiesID [PK],

book\_copies\_BookID [FK],

book\_copies\_BranchID [FK],

book\_copies\_No\_Of\_Copies

)

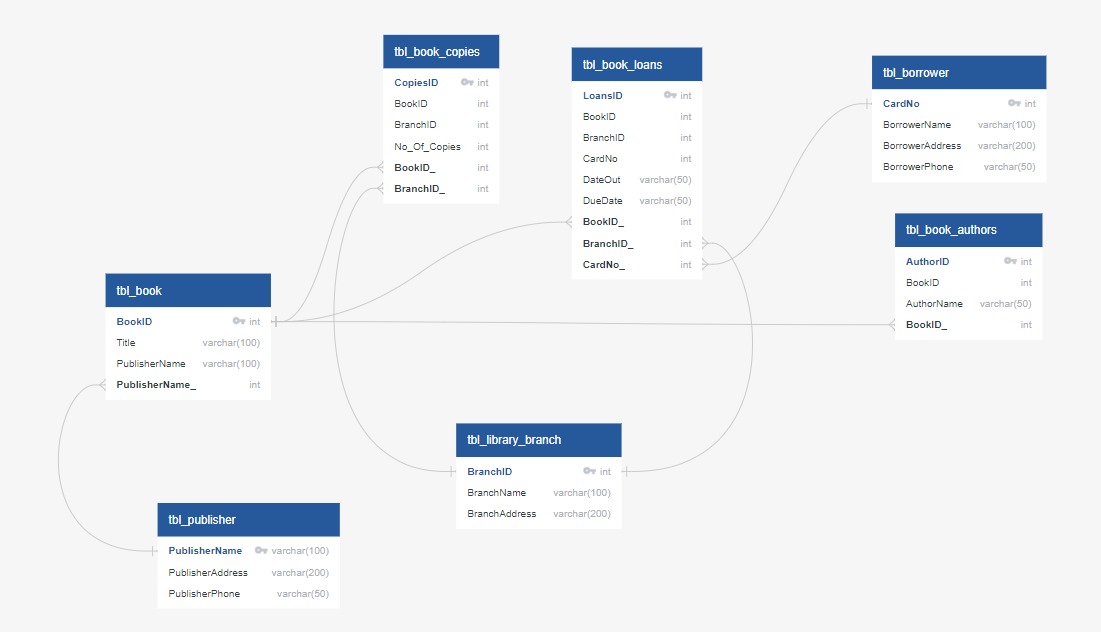
BookAuthors (

book\_authors\_AuthorID [PK],

book\_authors\_BookID [FK],

book\_authors\_AuthorName.

)



Relationships:

One-to-Many relationships exist between:

tbl\_book and tbl\_book\_loans (one book can have many loans).

tbl\_library\_branch and tbl\_book\_loans (one branch can have many loans).

tbl\_borrower and tbl\_book\_loans (one borrower can have many loans).

tbl\_book and tbl\_book\_copies (one book can have multiple copies in different branches).

tbl\_library\_branch and tbl\_book\_copies (one branch can have multiple copies of different books).

**Normalization:**

Normalization is used to minimize the redundancy from a relation or set of relations.

1. First Normal Form ( 1NF ):

A relation is said to be in its First Normal form if it has got no non-atomic attribute.

(Non-atomic attribute means the attribute which can’t be subdivided).

1. Second Normal Form (2NF):

A relation that is in 1NF is said to have a second normal form if it satisfies any one of the following conditions.

1. The primary key contains only one attribute.
2. There exist no non-key attributes.
3. Every non-key attribute present in the relation should functionally depend upon a full set of the primary key.
4. Third Normal Form (3NF).

The relation in 2Nf is said to be 3NF if there exists no transitive dependency of any non-key attribute on the set of the primary key.

Normalization of Database:

Applying the normalization process to the db\_LibraryManagement database schema from the previous prompts step by step:

Original Schema:

tbl\_book(Book\_ID (key), book\_Title, book\_PublisherName)

First Normal Form (1NF):

Meets the 1NF requirement as it has no non-atomic attribute.

Second Normal Form (2NF):

Meets the 2NF Rule-1: The primary key contains only one attribute. All non prime attributes are functionally dependent on primary attribute

Third Normal Form (3NF):

No transitive dependencies exist in this relation.

Normalization Steps:

Step 1:

The tbl\_book table already meets the requirements of 3NF, so no further normalization is needed for this table.

Now, let's apply normalization to other tables in the database:

tbl\_library\_branch(BranchID (key), BranchName, BranchAddress)

It does not need any normalization

tbl\_book\_copies(BookCopiesID (key), BookID (fk), BranchID (fk), NoOfCopies)

Step 2:

Normalize tbl\_book\_copies table to 3NF:

tbl\_book\_copies(BookCopiesID (key), NoOfCopies)

Step 3:

Create a new table tbl\_book\_branch to remove transitive dependency in tbl\_book\_copies table. There is transitive dependency between “BookId” and “No\_of\_Copies”

tbl\_book\_branch(BookID (fk), BranchID (fk))

Now, the final relations normalized to the Third Normal Form are:

tbl\_book(Book\_ID (key), book\_Title, book\_PublisherName)

tbl\_library\_branch(BranchID (key), BranchName, BranchAddress)

tbl\_book\_copies(BookCopiesID (key), NoOfCopies)

tbl\_book\_branch(BookID (fk), BranchID (fk))

|  |  |
| --- | --- |
|  |  |

**ENTITY RECORDS**

1: Publisher’s Table:

A close-up of a computer screen

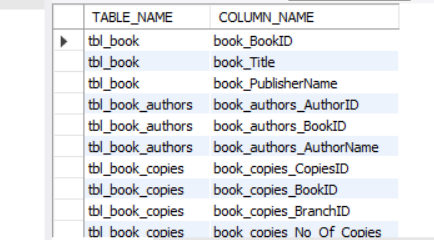
Description automatically generated

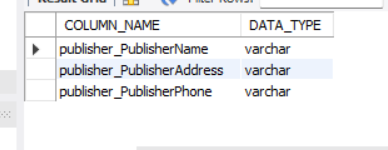
2:Book Table:

A screenshot of a computer

Description automatically generated

Sample Table:





🡪 QUERIES

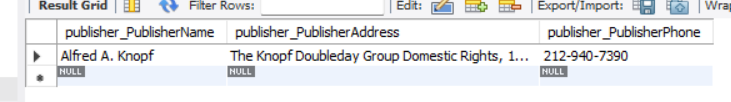
**Query 1: query will retrieve all records from the tbl\_publisher table where the publisher\_PublisherName column starts with the letter 'A'.**

**SQL Command:**

**SELECT \***

**FROM db\_LibraryManagement.tbl\_publisher**

**WHERE publisher\_PublisherName LIKE 'A%';**

****

**🡪VIEWS:**

CREATE VIEW BooksPublishedByNYPublishers AS

SELECT b.\*

FROM db\_LibraryManagement.tbl\_book b

JOIN db\_LibraryManagement.tbl\_publisher p ON b.book\_PublisherName = p.publisher\_PublisherName

WHERE p.publisher\_PublisherAddress LIKE '%New York%';

SELECT \*

FROM BooksPublishedByNYPublishers;

A screenshot of a computer

Description automatically generated

**QUERY 2:**

Names of pulisher starting with ‘b’.

SELECT \*

FROM db\_LibraryManagement.tbl\_book

WHERE book\_PublisherName IN (

SELECT publisher\_PublisherName

FROM db\_LibraryManagement.tbl\_publisher

WHERE publisher\_PublisherName LIKE 'B%'

);

A screenshot of a computer

Description automatically generated

THANK YOU