

# DBMS MINI PROJECT

SHAIL N PARIKH

U18CO082@COED.SVNIT.AC.IN



A decorative vertical bar in a vibrant magenta color runs down the left side of the page. The background features three overlapping circles: a large light blue circle at the top left, a large magenta circle at the bottom right, and a smaller light beige circle at the bottom center. The text is distributed within and around these circles.

**U18CO082**

**Mini project DBMS**

**SHAIL N PARIKH**

-----  
**COED-II@SVNIT**

**2020**

**Sem IV**

# LIBRARY SYSTEM

MANAGEMENT AND DATABASE  
HANDLING

## PROJECT DETAILS-- **CONTENT AND EXECUTION STAGES:**

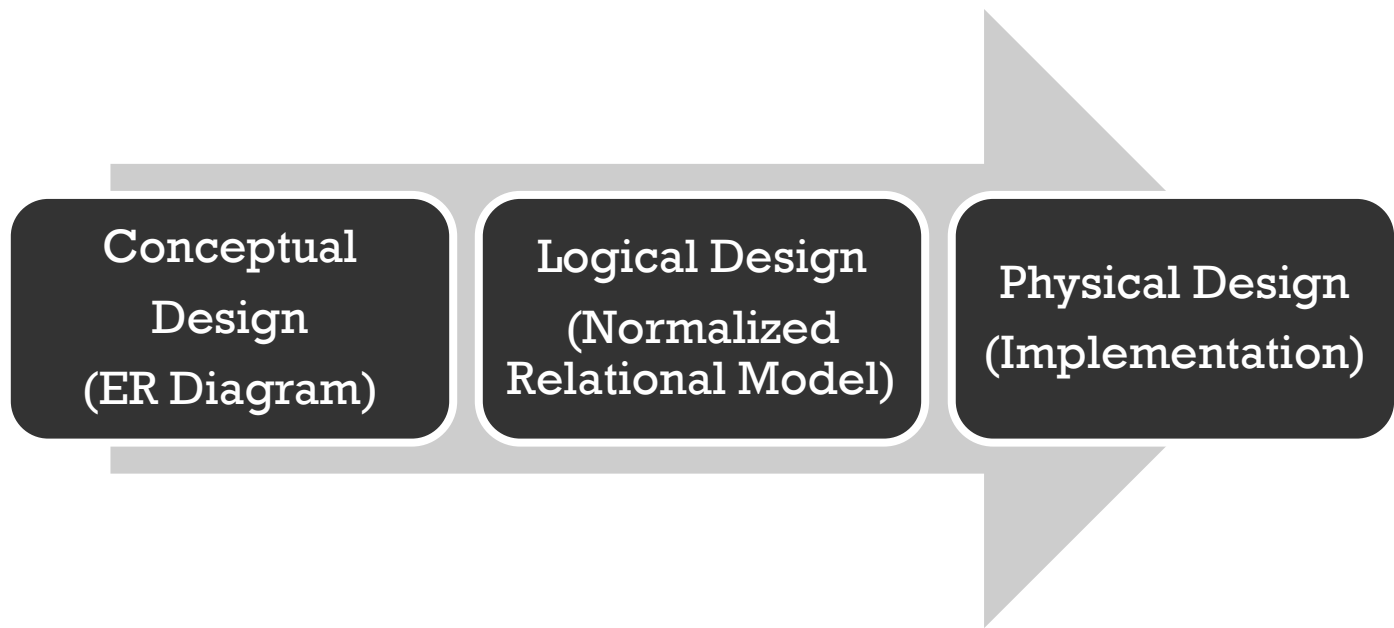
- Definition
- ER Diagram
- Conversion into Relational Model
- Normalization

# LIBRARY SYSTEM

MANAGEMENT AND DATABASE  
HANDLING

## DEFINITION:

- The process of creating Database and Managing the data in database without any redundancy or inconsistency consists of three phases i.e. Database Design Process.
- Project Outline



- Conceptual Designing involves understanding the requirements of the system and identifying the entities and creating an ER Diagram.
- In Logical Designing, Relational Schema is formulated using ER Diagram and Normalization Forms.
- Physical Designing is the last phase where the relations are created and various constraints are coded by using SQL statements and queries.
- Then various queries, procedures, functions, triggers, etc. are implemented using PL SQL codes.

## PROJECT OUTLINE:

- Library Data has to be stored in an efficient way so that it can be easily retrieved and there is no use of excess memory as well as time in fetching the required data.
- This project aims at developing an efficient and effective Database Management System, which stores and presents the data without any inconsistency and redundancy.
- There may be several **branches** of the same library across cities or states or even countries. So, that data can also be handled and managed by the Database Management System developed here.
- It stores all the data about Authors, Books and their Publishers, **number of copies** of a certain book available at a certain branch as well as the total copies.
- If a book demanded by a Member is not available at a particular branch, then it's availability is searched across other branches and lent to the Member if present at any of the branches (if possible).
- It also stores information related to the Publishers and of the Members.
- The issue, return and due dates are stored in a table and fine is calculated using function, if returned late.
- **Salient Feature** of this model is that it handles data of multiple branches and provides best possible way to satisfy the needs of Members.

## Code Snippet:

```
27
28 create or replace procedure book_availability(
29     book_id in varchar2,
30     branch_id in varchar2,
31     x out number
32 )
33 is
34 begin
35     x := 0;
36     select No_of_Copies into x from Book_Copies where Book_ID = book_id and Branch_ID = branch_id;
37     if x = 0 then
38         dbms_output.put_line('Oops! No Copy of the Book available at this Branch. ');
39         dbms_output.put_line('You will find this Book at Branches listed below: ');
40         select Library_Branch.Branch_Name, Library_Branch.Address from Book_Copies natural Library_Branch
41         where Book_Copies.Book_ID = book_id and Book_Copies.Branch_ID = branch_id and Book_Copies.No_of_Copies>0;
42     else
43         dbms_output.put_line('Yaay! Book is available here. ');
44     end if;
45 end;
46 /
47
```

- Similarly, Triggers have been created for checks during insertion, deletion or update. Functions are also formulated to calculate the fine and various functionalities.



# LIBRARY SYSTEM

MANAGEMENT AND DATABASE  
HANDLING

Part 2

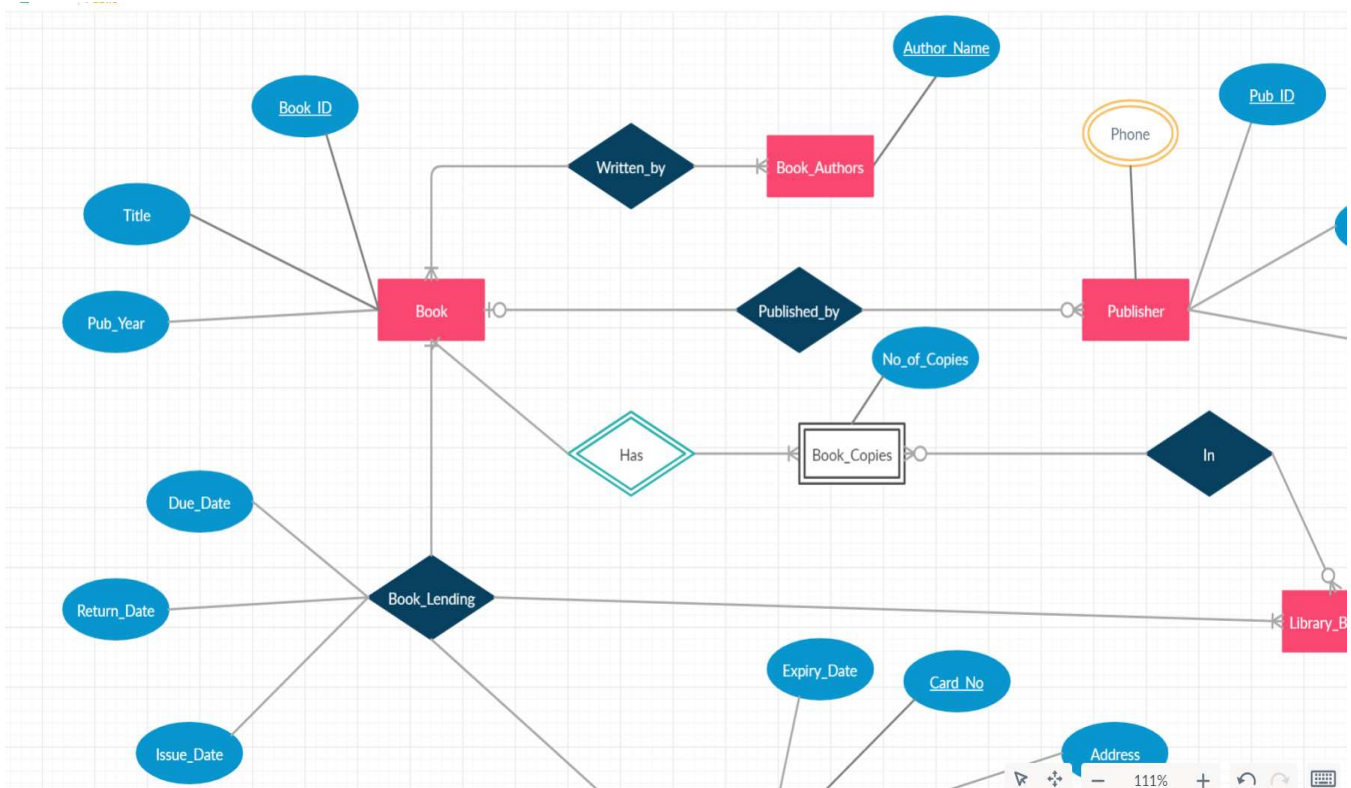
Conceptual Designing

## ER DIAGRAM:

- Entities
- Relationships
- Attributes of Entities
- ER Diagram of Library Management System

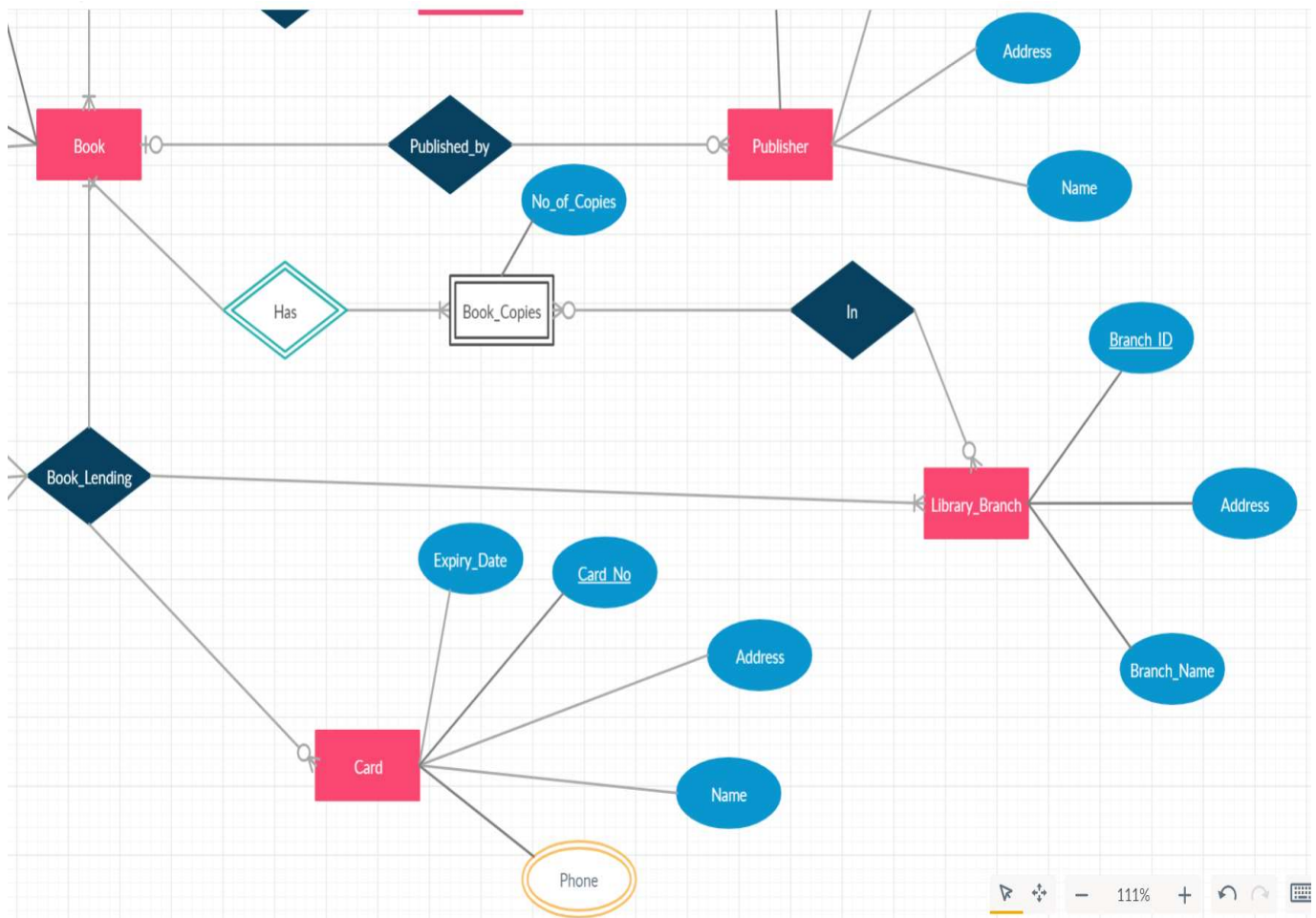
## ENTITIES:

- **Book**
- **Book\_Authors**
- **Card**
- **Publisher**
- **Library\_Branch**
- **Book\_Copies** (weak entity)



## RELATIONSHIPS:

- **Written\_by**
- **Published\_by**
- **Book\_Lending**
- **Has** (weak relationship)
- **In**



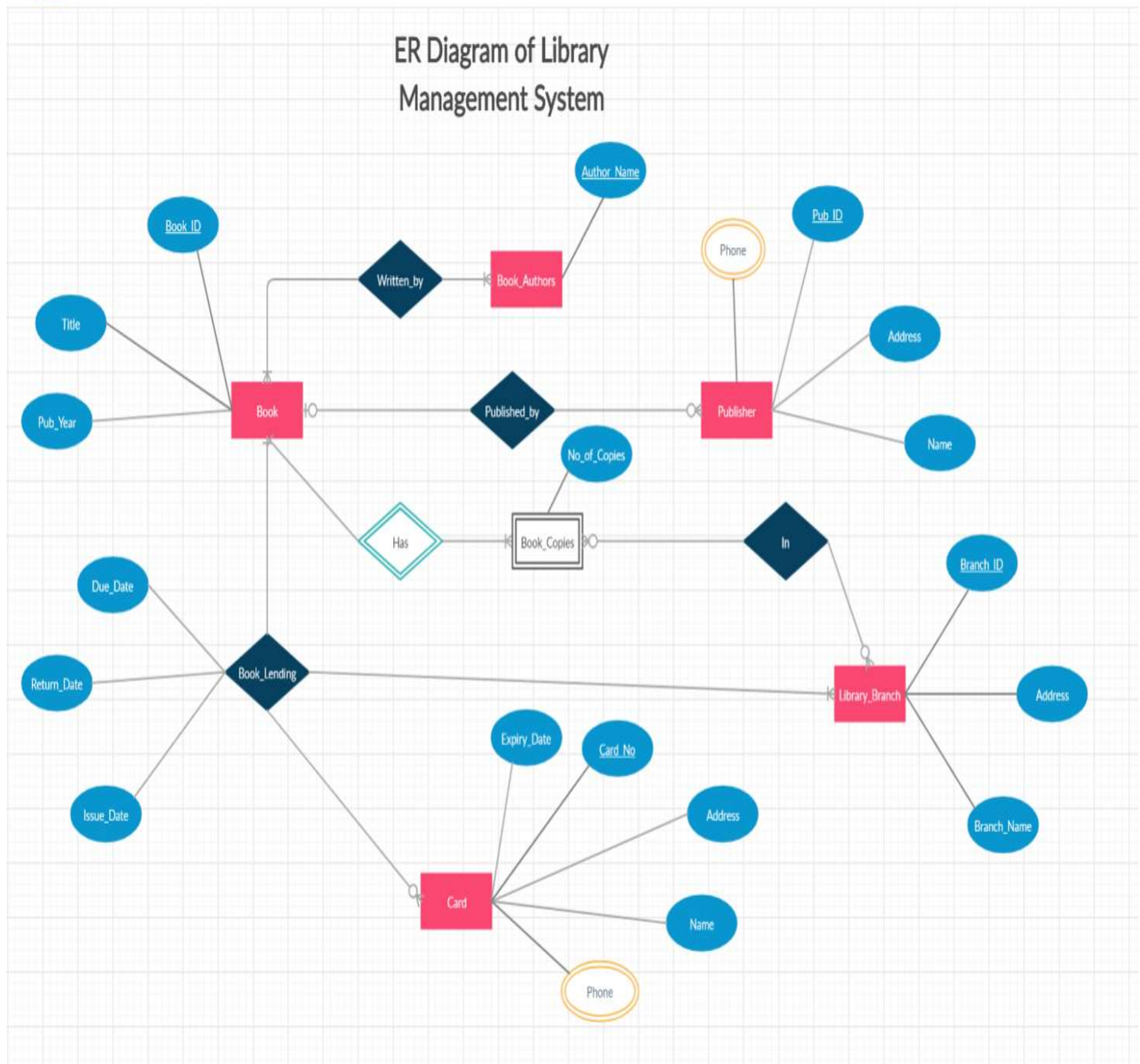
## ATTRIBUTES OF ENTITIES:

- **Book** -- Book ID, Title and Pub\_Year
- **Book\_Authors** -- Author Name
- **Book\_Copies** -- No\_of\_Copies
- **Publisher** -- Pub ID, Name, Address and Phone
- **Card** -- Card No, Name, Address and Expiry\_Date
- **Library\_Branch** -- Branch ID, Address and  
Branch\_Name

## ATTRIBUTES OF A RELATIONSHIP:

- **Book\_Lending** -- Issue\_Date, Return\_Date and  
Due\_Date

## ER DIAGRAM OF LIBRARY MANAGEMENT SYSTEM



# LIBRARY SYSTEM

MANAGEMENT AND DATABASE  
HANDLING

Logical Designing

## CONVERSION INTO RELATIONAL MODEL:

- Analysis of ER Diagram and Conversion into Relational Model

## ANALYSIS OF ER DIAGRAM AND CONVERSION INTO RELATIONAL MODEL:

- Since a book can be written by one or more authors, and an author in the database of Library must have minimum one book describing his contribution. So, the relationship is 'many-many' with relations –

*Book ( Book ID, Title, Pub\_Year ) ,*  
*Book\_Authors ( Book ID, Author Name )*

Here, since Book\_Authors has only one attribute ,we have replaced the Written\_by relation with the Book\_Authors relation.

- A book can be published by one or more publishers ,and a publisher should have published minimum one book. So, this is also 'many-many' relationship with relations –

*Published\_by ( Book ID, Pub ID ) ,*  
*Publisher ( Pub ID, Name, Address, Phone )*

- Copies of a particular book will only exist if there exists that book in any of the branches of the library, hence it is a weak entity.

The number of copies at a specific branch is also stored in that relation as –

*Book\_Copies ( Book\_ID, Branch\_ID, No\_of\_Copies )*

- The details of all the Members is stored by a unique Card\_No provided to each as shown –

*Card ( Card No, Name, Address, Phone, Expiry\_Date )*

- Details of all the branches is stored in the relation Library\_Branch –

*Library\_Branch ( Branch ID, Branch\_Name, Address )*

- All transactions related to lending are recorded by –

*Book\_Lending ( Book ID, Card No, Issue\_Date, Return\_Date, Due\_Date )*

Fine (if any) due to late returning of book is calculated using functions coded in PL SQL.



# LIBRARY SYSTEM

MANAGEMENT AND DATABASE  
HANDLING

Part 4

Logical Designing

## **NORMALIZATION:**

- Functional Dependencies and Normalization
- Normalized Relational Schema

## FUNCTIONAL DEPENDENCIES AND NORMALIZATION:

- Here, only the Phone attribute in Publisher and Card entity may contain more than one values. It is a multi-value attribute.

So, to avoid data redundancy we decompose the relations Publisher and Card as –

*Publisher ( Pub\_ID, Name, Address )*

*Pub\_Phone ( Pub\_ID, Phone )*

*Card ( Card\_No, Name, Address, Expiry\_Date )*

*Card\_Phone ( Card\_No, Phone )*

- Now, all the attributes in every relation have atomic domain. Hence, the relations are in **1<sup>st</sup> Normal Form**.
- Functional Dependencies are –

*Book\_ID -> { Title, Pub\_Year }*

*{ Book\_ID, Branch\_ID } -> No\_of\_Copies*

*Pub\_ID -> { Name, Address }*

*{ Book\_ID, Card\_No } -> { Issue\_Date, Return\_Date, Due\_Date }*

*Card\_No -> { Name, Address, Expiry\_Date }*

*Branch\_ID*  $\rightarrow$  { *Branch\_Name*, *Address* }

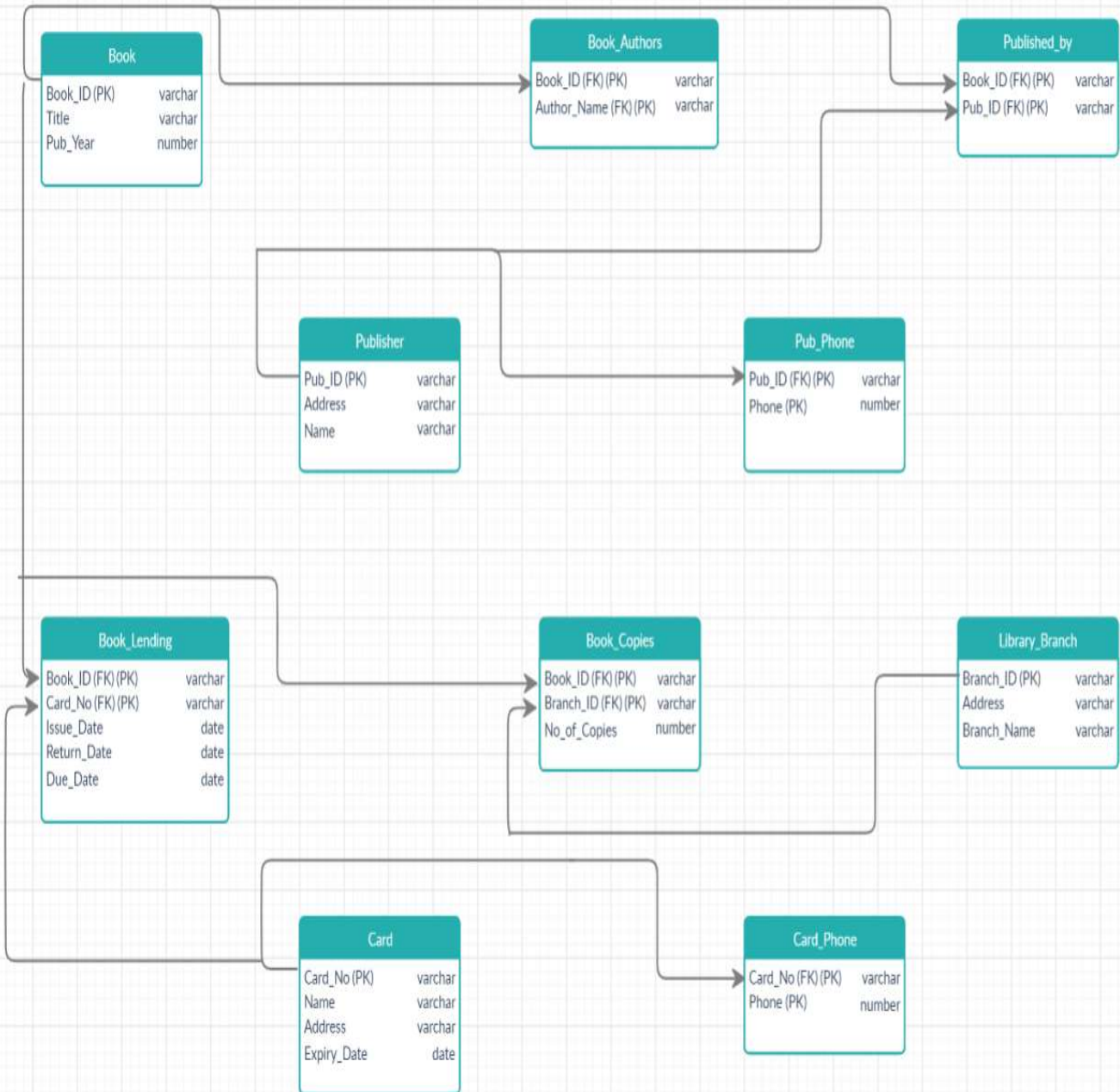
- It can be observed from the functional dependencies listed above that there is no partial dependency in any of the relations. Hence, the relations are in **2<sup>nd</sup> Normal Form**.
- No Transitive dependency, so the relations are in **3<sup>rd</sup> Normal Form**.
- The Left Hand Side (L.H.S.) of every Functional Dependency (FD) is Super Key. Hence, all relations satisfy **Boyce-Codd Normal Form**.
- There is no multivalued dependency in any of the relations. In Pub\_Phone and Card\_Phone relations, there are only two attributes which do not violate the rule. Hence, all relations are in **4<sup>th</sup> Normal Form**.
- Natural Join of every relation with common attributes is lossless and every FD is preserved. Since, the common attributes functionally determine every other attribute of both the relations.

Hence, Join Dependency is preserved. Hence, all the relations are in **5<sup>th</sup> Normal Form**.

- Hence, the relations are **completely Normalized**.
- There are no extraneous attributes or dependencies hence, the set of FDs is the **Canonical Cover**.

# NORMALIZED RELATIONAL SCHEMA

## SCHEMA DIAGRAM FOR LIBRARY MANAGEMENT SYSTEM DATABASE



***END***