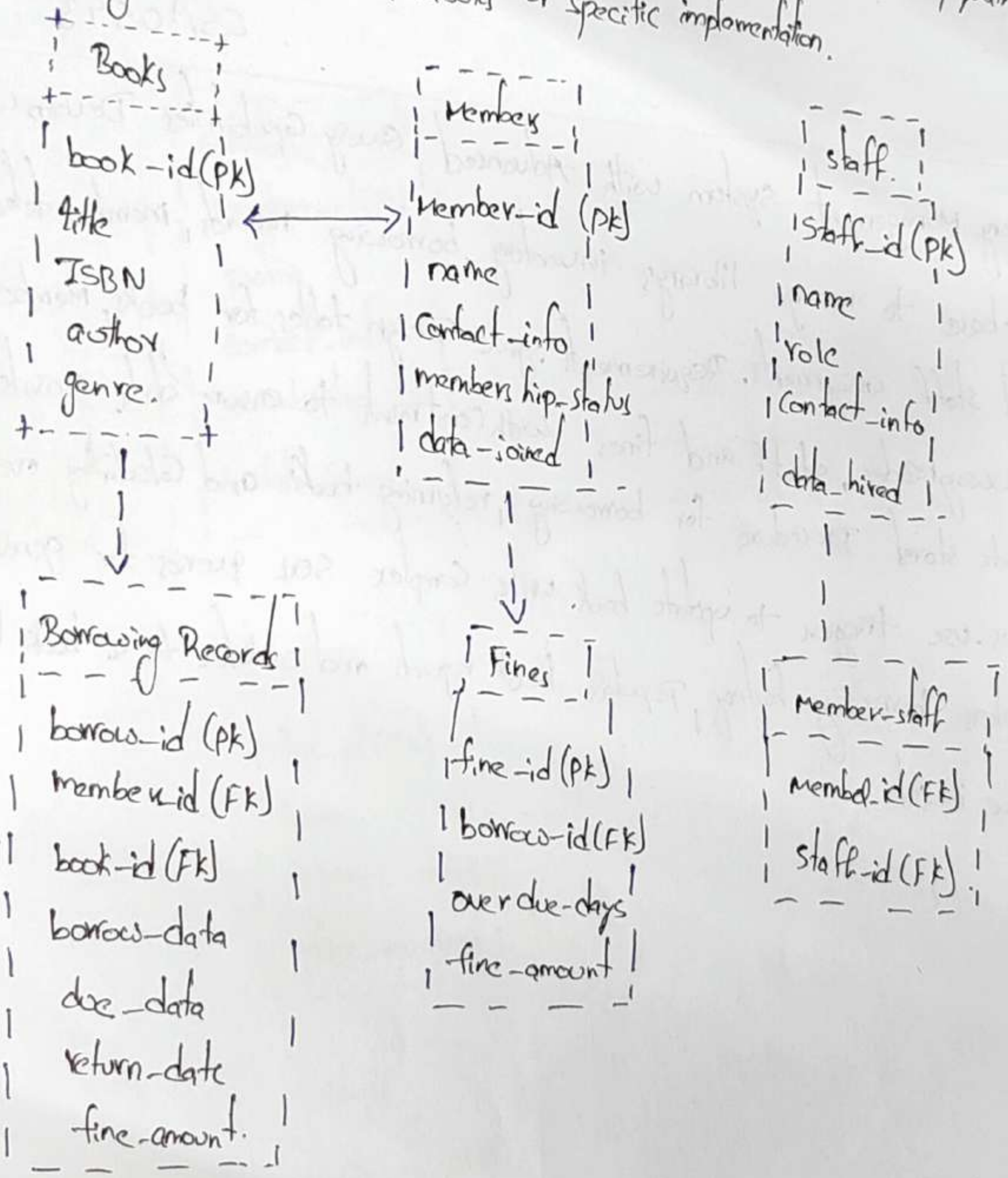
Assignment

D. Hari Prasanna
192325128
DBMS
CSA0593

1) Library Management system with Advanced Query Capabilities Develop with a database to manage library's inventory, borrowing records, membel details and staff assignments. Requirements: give Design tables for books, Membels Borrowingrecodes, staff and fines with constraints to ensure data consistency, write stored Procedure for borrowing, returning book and calculating overdues fines. use triggen to update book. write complex SQL queries for genelating members borrowing history, Popular book reports and overdue to a book lists give me list.

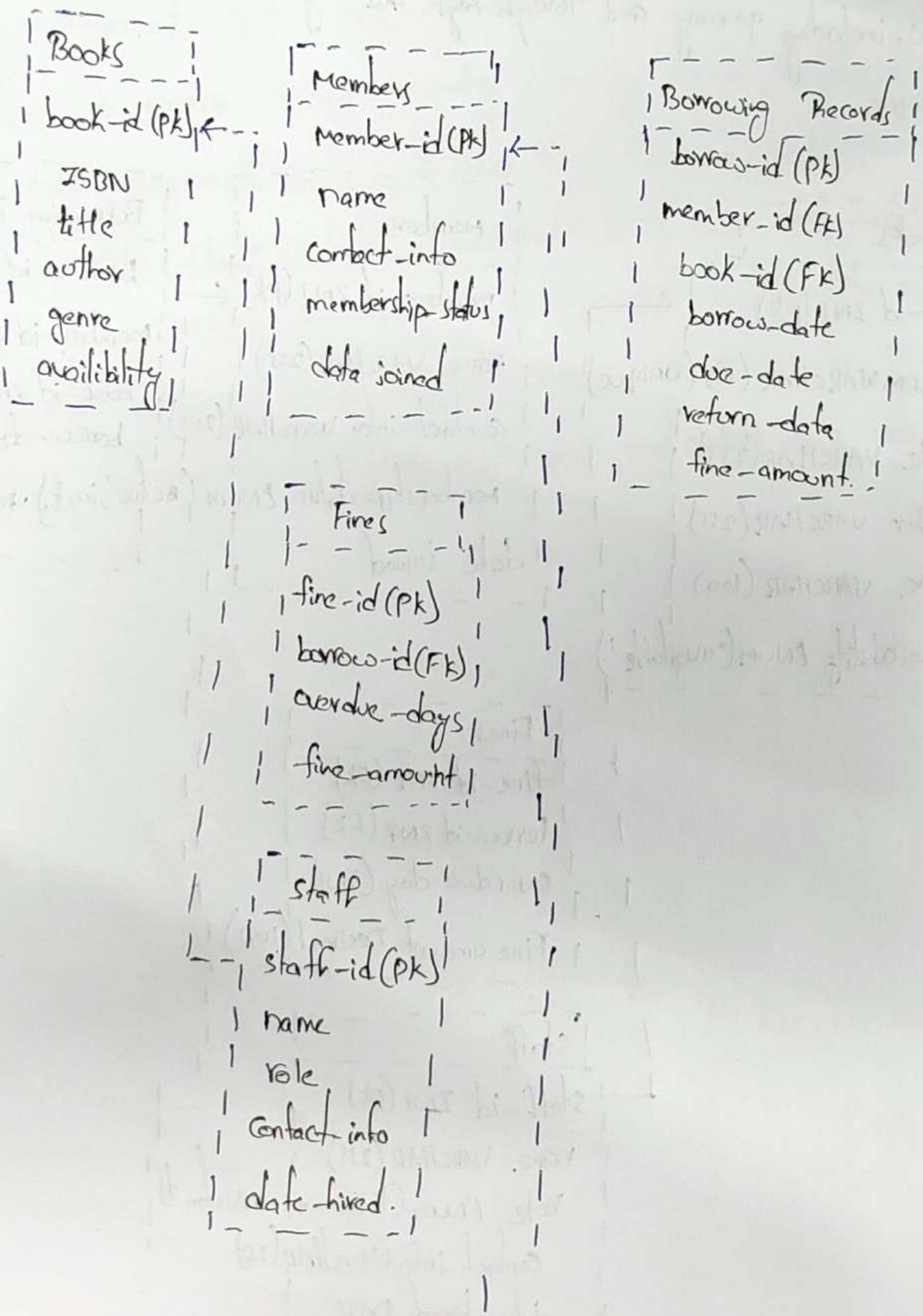
Conceptual ER diagram:-

This diagram outlines the entities and their basic relationship, with out focosing on data base attributes or specific implomentation.



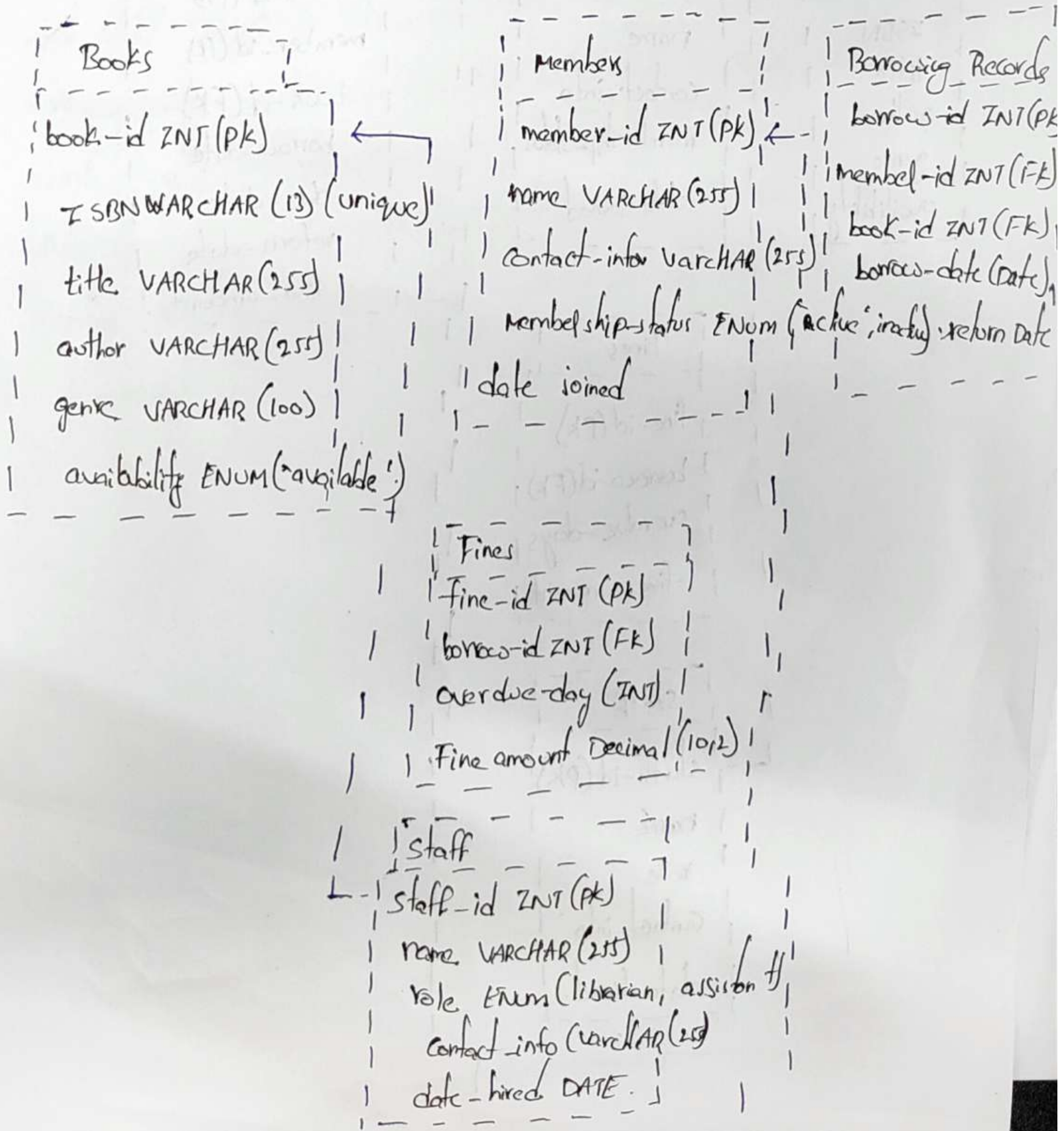
logical ER diagram:-

This diagram refines the conceptual model by adding specific attributes and specifying Primary and foreigh keys.



Physical ER diagram:-

In the physical ER diagram, we focus on how the entities will be implem and in a relational database. This includes detailed specification for each table, including primary and foreigh keys and any constraints such as NOTN or unique.



1. Create Tables:-

```
CREATE TABLE Books (
 BookID INT PRIMARY Key,
  Title VARCHAR(100),
  Author ID INT,
 Genre VARCHAR (50),
 published year (INT),
 CopiesAvailable INT,
 FOREIGN key(Author ID) REFERENCES Authors (AUTHor ID)
 );
  CREATE TABlE Authors (
  Author ID INT PRIMARY key
  NAME VARCHAR(100),
  Country VARCHAR(50)
  );
   Create Table Loans
  Loan ID INT PRIMARY Key,
   Book ID INT,
   Member ID INT,
   Loan Date DATAE,
   DUE DATE DATE,
   RETURNDate DATE,
  FOREIGN key(Book ID) REFERENCES Books (Book ID),
FOREIGN key (member ID) REFERENCES Members (member ID).
```

Insert Data sample:-

```
Insert INTO Author (Author ID, Name, Country)
Values(1, 'J.K. Rowling', 'united kingdom'),
      (2, 'George orwell', 'united kingdom'),
      (3, 'J.R.R. Tolkien', 'united kingdom');

INSERT INTO Members (Members ID, NAME, Address, Phone, Email, Membership Date)
Values(1, 'Alice Johnson', '123 Maple street', '1234567890', 'alice@example.com', '2023-01-15'

Insert into Loans (Loan ID, Book ID, Member ID, loan Date, Due Date, Return Date
values (1,1,1, '2024-10-1', '2024-10-15', Null),
```

Queries:-

```
SELECT Title, Genre, published year, Copies Available
From Books
WHERE Copies Available >0;

SELECT Member. Name, members. Email, Books. Title, Loans. LoanDate, loans. Due Date
from members
Join Loans on Members. member ID=loans. member ID
Join books on Loans. Book ID=Books. Book ID.
WHERE Loans. Return Date is NULL;

SELECT Members. Name, Books. Title, loans. Due Date
From loans
Join members ON Loans. Member ID= Member. Member ID
Join Books ON Loans. Book ID= Books. Book ID
WHERE Loans. Return Date IS NULL AND loans. Due Date < CURDATE( );
```