

How can we design a comprehensive database system to efficiently manage a library's resources, including books, members, transactions, and administrative details?

How can we design a comprehensive database system to efficiently manage a library's resources, including books, members, transactions, and administrative details?

Arnaldo Fajardo

CUNY Lehman

Database Systems CMP 420

2023 Fall Term

How can we design a comprehensive database system to efficiently manage a library's resources, including books, members, transactions, and administrative details?

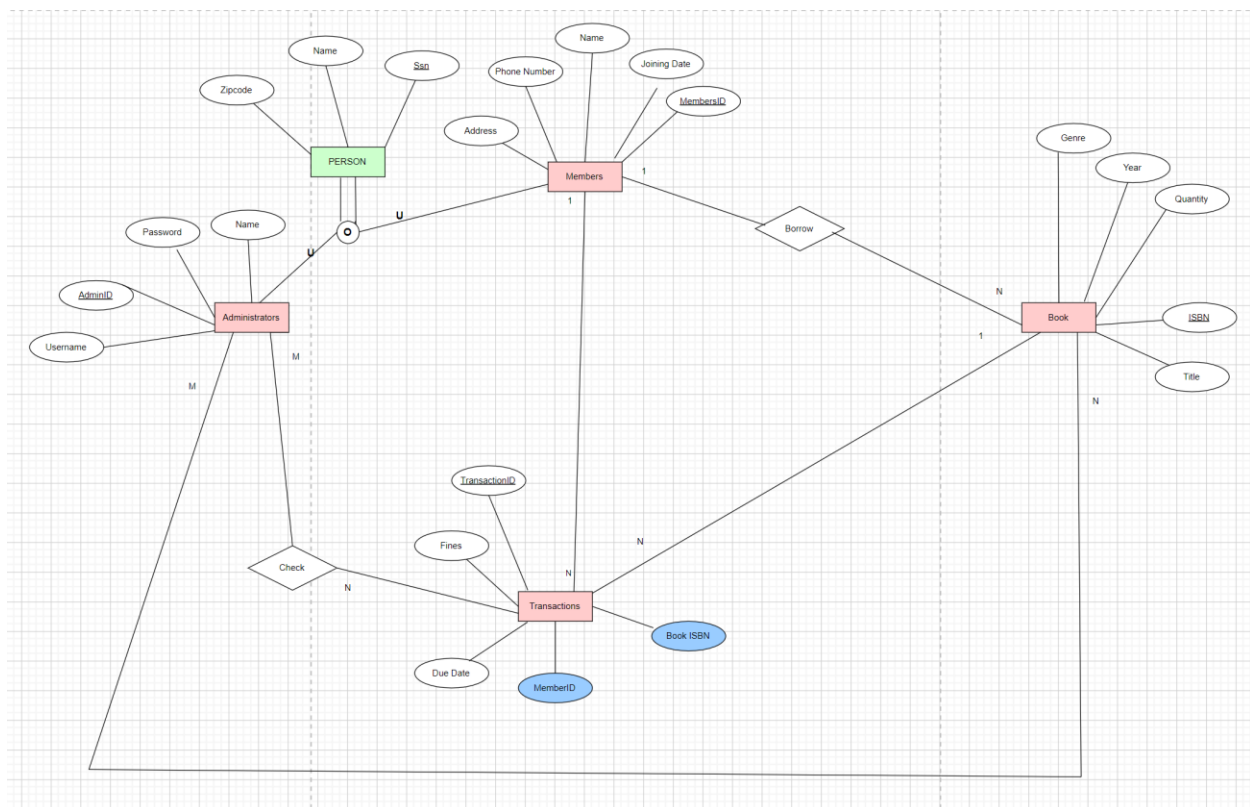
How can we design a comprehensive database system to efficiently manage a library's resources, including books, members, transactions, and administrative details?

- Each **Book** has different attributes like ISBN, Title, Author, Genre, Year, and Quantity.
 - Libraries also has members who can borrow books for a period of time. **Members** has attributes like Member ID, Name (inheritance from Person), Address, Phone Number, and Joining Date.
 - Moreover, Transactions are important in library to keep track of any book order. **Transactions** have Transactions ID, Book ISBN, Member ID, Due Date, Fines.
 - A library needs administrators to be able to keep it secure and in order. **Administrators** have Admin ID, Name (inheritance from Person), Username, and Password.
-
- ❖ Superclass **Person**: where we have attributes that defines this **Person** such as Name, Zipcode, and SSN as the primary key.
 1. Subclass **Member**: where a **Person** can be a Member, and he needs the following attributes Address, Phone Number, Joining Date and MemberID
 2. Subclass **Administrators**: Moreover, a **Person** can be an Administrator, and he needs an AdminID and Password.
 - When a member joins the membership can **Borrow** as many **Books** that she/he wants, but keeps in mind to return them on time.
 - However, when she/he is a Administrator must have the responsibilities to **Check** all the **Transactions** to fix mistakes and correct them.

How can we design a comprehensive database system to efficiently manage a library's resources, including books, members, transactions, and administrative details?

Assumption.

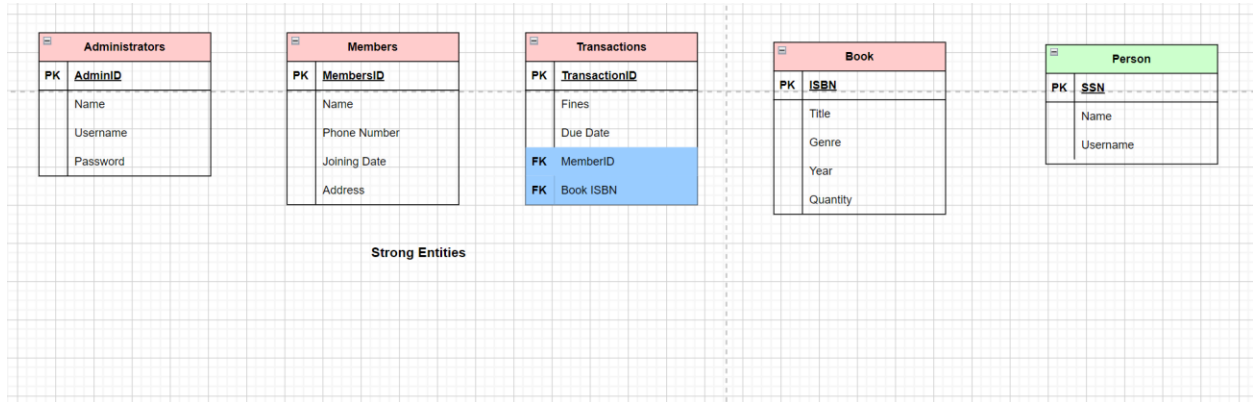
Every **Book** has a unique ISBN that makes easier to look the name of the book in library, as well as each **Member** has a unique MemberID to be able to track history of that member. The **Transactions** have a specific Transaction ID where it is easier to locate for specific rentals books. **Fines** are calculated based on the overdue days and predefined fine rate. Also, **Books** can have multiple copies and each copy is treated as a separate entity for tracking availability.



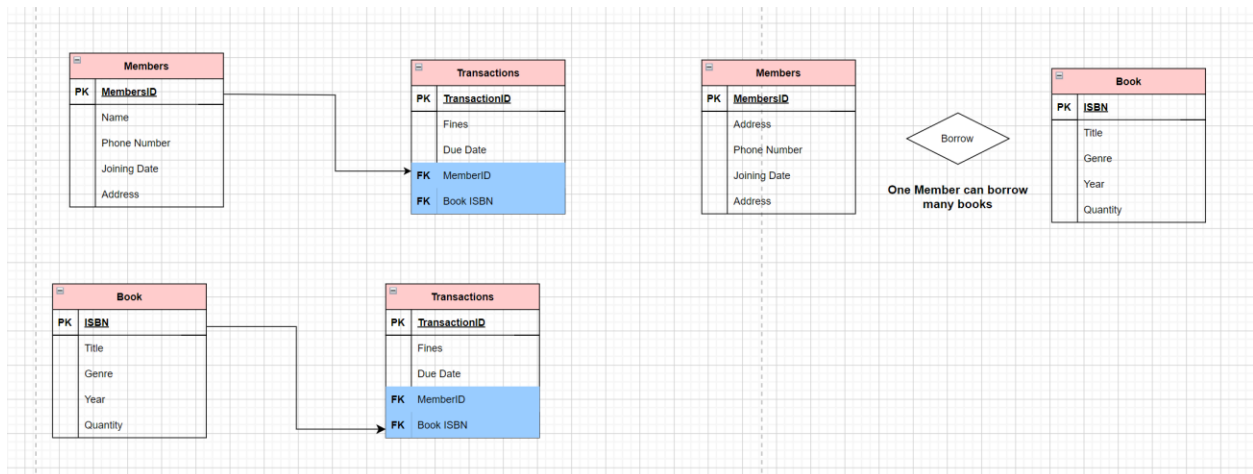
How can we design a comprehensive database system to efficiently manage a library's resources, including books, members, transactions, and administrative details?

Logical Modeling.

Step 1: Strong entities

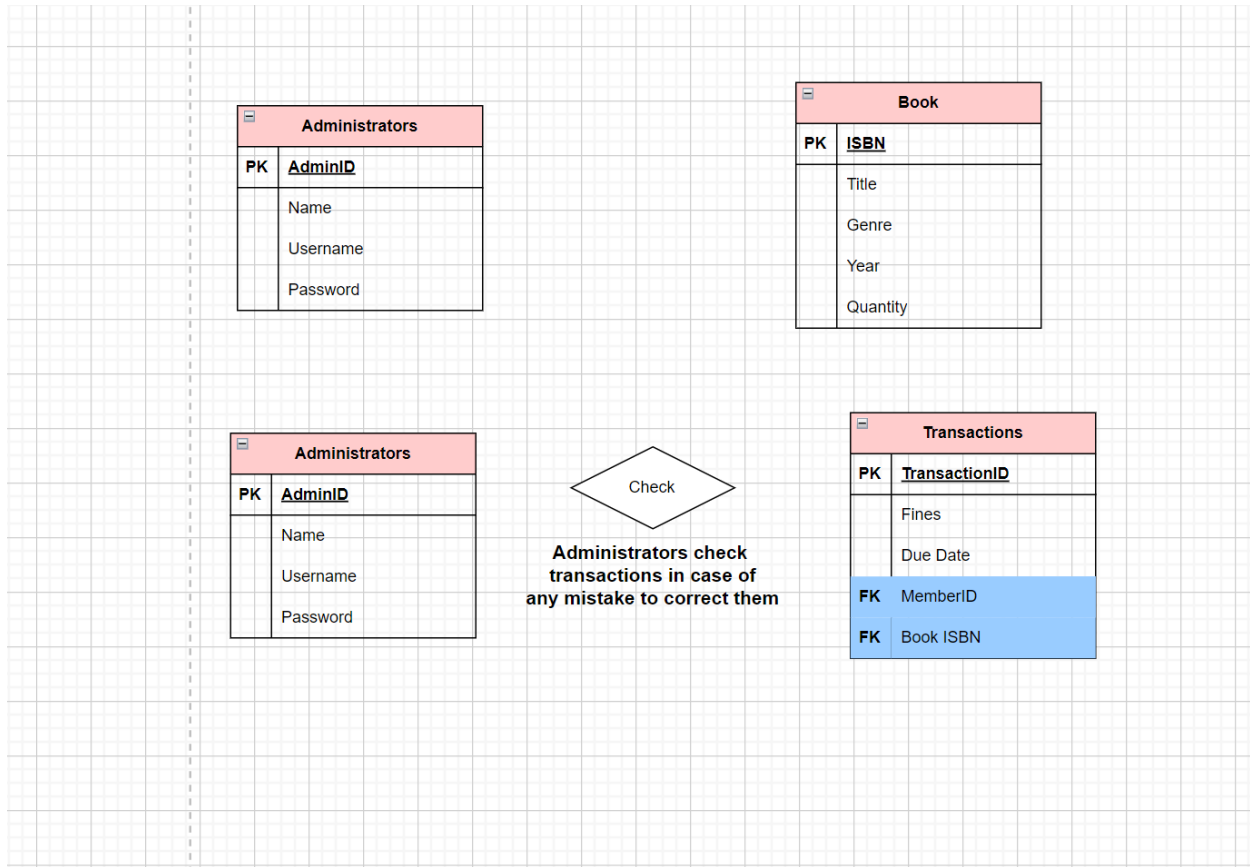


Step 2: Binary relationship 1:N



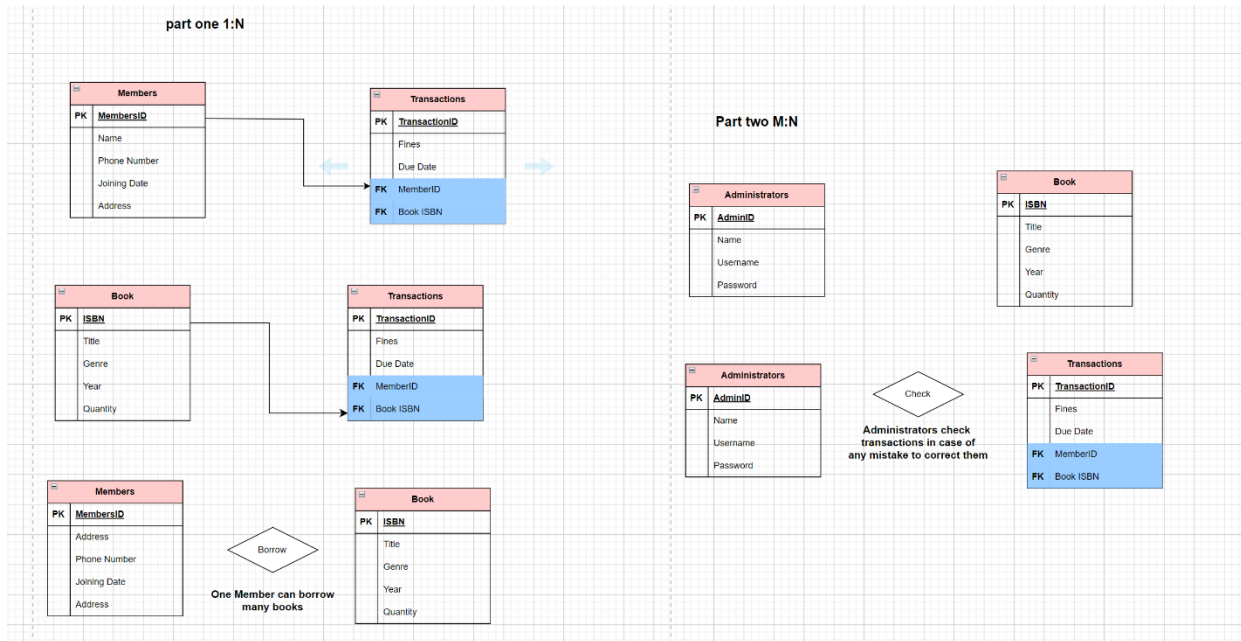
How can we design a comprehensive database system to efficiently manage a library's resources, including books, members, transactions, and administrative details?

Step 3: Binary relationship M:N



How can we design a comprehensive database system to efficiently manage a library's resources, including books, members, transactions, and administrative details?

Final Relational Diagram



How can we design a comprehensive database system to efficiently manage a library's resources, including books, members, transactions, and administrative details?

Database Implementation for 1 Relationship.

In this point I realize that I need to make some improvement of the variable name such as Phone Number to PhoneNumber, Joining Date to JoiningDate, Due Date to DueDate, and Book ISBN to BookISBN to have a better data integrity.

A **Member** wants to know the **DueDate** to return a book from a previous **Transactions**.

#Creating the Members table

```
CREATE TABLE Members (  
  MembersID INT PRIMARY KEY,  
  Name VARCHAR(75),  
  PhoneNumber VARCHAR(15),  
  JoiningDate VARCHAR(75),  
  Address VARCHAR(100)  
);
```

Insert 5 rows into the Members table

```
INSERT INTO Members VALUES  
(0050, 'Peter', '917-833-4520', '12-08-2021', '176 Dyckman st 8M'),  
(0296, 'Buck', '856-847-8820', '09-01-2006', '111 Burnside AV 3F'),  
(0145, 'Sam', '203-964-7521', '01-20-2020', '106 Santiago st 1N'),  
(2001, 'James', '456-852-7896', '04-11-2009', '202 Penn AV 7K'),  
(2003, 'Carl', '203-963-4521', '05-19-2020', '115 Flouch st 14J');
```

#Creating the Transactions table

```
CREATE TABLE Transactions (  
  TransactionID INT PRIMARY KEY,  
  Fines INT,  
  DueDate VARCHAR(100),  
  MemberID INT,  
  FOREIGN KEY (MemberID) REFERENCES Members(MembersID)  
  BookISBN VARCHAR(100)  
);
```

How can we design a comprehensive database system to efficiently manage a library's resources, including books, members, transactions, and administrative details?

Insert 5 rows into the Book table

```
INSERT INTO Transactions VALUES
(5896, 52, '12-15-2023', 2031, '10-85-456'),
(3457, 20, '12-28-2023', 3201, '45-78-654'),
(2477, 15, "12-30-2023", 4521, '15-85-963'),
(1045, 10, "12-19-2023", 1237, '80-90-785'),
(1132, 05, "12-23-2023", 5968, '52-73-854');
```

Query:

```
SELECT Transactions.DueDate
FROM Transactions
JOIN Members ON Transactions.MemberID = Members.MemberID
WHERE Members.Name = 'Peter';
```


How can we design a comprehensive database system to efficiently manage a library's resources, including books, members, transactions, and administrative details?

Conclusion.

What was easy about this project for you? What was difficult about this project for you?

The coding part I found easier because the Query part was not hard to code it. We have seen many examples like this query in class. The relational Diagram was hard because we need to do the EER first and make the relationships to be able to know the PK and FK. Also, I got a little confusing doing the superclass and subclass because at the beginning I was not understanding the concept, but then I could get it.

What was your experience learning about the fundamentals of database systems?

To be honest at the beginning I thought I was more theory than coding like (a lot of reading again), but when we get to the hackerank (coding part) was interested more because of the implementation of coding in this programming language. this programming language is similar as speaking English like `SELECT Pizza FROM Store`, that made it sort of easier to catch.

Any final comments and conclusions

I would like to say that, I would love to learn more data types in class and more answer question from hackerank. I like the idea of the professor sending us to solve problems and helping us at the same time. This build more confidence to us because the professor is there to help us when we are wrong. Thanks for that professor!!!