Library Management System Database

University of Texas at Arlington

Department of Computer Science and Engineering

CSE 3330 – 004: Database Systems and File Structures

Dr. Nadra Guizani

Project 2 Part 1

Fall 2024

Chime Nguyen

Ivan Ko

Trung Nguyen

# HONOR CODE

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or that I contribute to group

collaborations, and I will appropriately reference any work from other sources. I will follow the

highest standards of integrity and uphold the spirit of the Honor Code.

# TABLE OF CONTENTS

I.) Honor Code ……………………………………………………………… 2

II.) Introduction ……………………………………………………………… 4

III.) Mini World Description .….….…………………………………………… 5

I.) Entities

II.) Relationships

III.) Example Scenarios

III.) Assumptions ……………………………………………………………… 7

I.) No Author Entity

II.) Assuming The ReturnDate Attribute Is The Due Date

III.) Creating A BorrowFrom Entity From The Relationship

IV.) Entity Relationship Diagram ……………………………………………… 9

I.) Publisher Entity

II.) Book Entity

III.) Book\_Copies Weak Entity

IV.) Library\_Branches Entity

V.) Borrows\_From Weak Entity

VI.) Borrower Entity

V.) Relational Database Schema ……………………………………………… 12

I.) Book Table

II.) Publisher Table

III.) Book\_Copies Table

IV.) Library\_Branches Table

V.) Borrower Table

VI.) Borrows\_From Table

VI.) References .………………………………………………………………… 15

# INTRODUCTION

We developed an Entity-Relationship (ER) diagram as well as a relational database schema based upon the “Library Management System” detailed in the requirements for the database application. These visualizations of the structuring of data illustrate the entities involved – books, publishers, book copies, library branches, borrowers, and book loans – along with their attributes and relationships between them. Following the creation of the ER diagram, we translated the ER diagram into a relational database schema for a structured format suitable for implementation toward a database management system.

The ER diagram we produced offers a graphical depiction of the system’s entities and their relevant attributes, seeing how the metadata is structured within the library’s data ecosystem. Translating the ER diagram to a relational database schema, we mapped the entities and relationships into tables, defined the primary and foreign keys, and formatted them such that the schema could be designed for optimized implementation in a database system.

# MINI WORLD DESCRIPTION

## Entities:

1. BOOK:
   1. BookID: Unique identifier for books
   2. PublisherName: Name of the book’s publisher
   3. Author: Book author’s name
   4. Title: Title of the book
2. PUBLISHER:
   1. Name: Unique name of the publisher
   2. PhoneNumber: publisher’s phone number
   3. Address: Publisher’s address
3. BOOK\_COPIES:
   1. BookID: Identifier for the book
   2. BranchID: Identifier for the branch
   3. No\_of\_copies: Number of copies of a specific book at a specific branch
4. LIBRARY\_BRANCH:
   1. BranchID: Unique identifier for branches
   2. Address: Adress of the branch
   3. Name: Name of the branch
5. BORROWER:
   1. Card\_No: Unique identifying number for book borrowers
   2. Name: Name of the borrower
   3. PhoneNumber: Phone number of the borrower
   4. Address: Address of the borrower
6. BORROWS\_FROM:
   1. BookID: Identifier for the book being borrowed
   2. BranchID: Branch where the book belongs
   3. Card\_No: Identifier for the borrower of the book
   4. CheckedOut: The date when the book was checked out
   5. ReturnDate: The due date of the book
   6. Returned: Boolean value for if the book has been returned yet

## Relationships:

1. PUBLISHER - BOOK: publishers can publish many books (one to many)
2. BOOK – BOOK\_COPIES: Each book can have many book copies (one to many)
3. LIBRARY\_BRANCH – BOOK\_COPES: Many library branches can own many copies of a book (many to many)
4. BORROWER – LIBRARY\_BRANCH: Many borrowers can borrow from a library branch. (many to one)

## Example Scenarios:

1. A new library branch named “Lorem Branch” is opened
   1. LIBRARY\_BRANCH entry added
      1. BranchID: 12
      2. Address: “789 Ipsum Dr, Arlington, TX”
      3. Name: “Lorem Branch”
2. Lorem Branch acquires multiple copies of a new book:
   1. BOOK entry added
      1. BookID: 1001
      2. PublisherName: “Joe’s Publishing”
      3. Author: “John Doe”
      4. Title: “Fundamentals of Database Systems”
   2. PUBLISHER entry added
      1. Name: “Joe’s Publishing”
      2. PhoneNumber: “123 - 456 – 7890"
      3. Address: 123 Publishing St, Arlington, TX
   3. BOOK\_COPIES entry added
      1. BookID: 1001
      2. BranchID: 12
      3. No\_of\_copies: 10
3. A new borrower registers at Lorem Branch and borrows a book
   1. BORROWER entry added
      1. Card\_No: 2001
      2. Name: “John Smith”
      3. PhoneNumber: “234 - 567 – 8901"
      4. Address: “456 House Ave, Arlington, TX”
   2. BORROWS\_FROM entry added
      1. BookID: 1001
      2. BranchID: 12
      3. Card\_No: 2001
      4. CheckedOut: “10 - 31 – 2024"
      5. ReturnDate: “11 - 10 – 2024"
      6. Returned: 0 (false)
4. John returns the book he borrowed
   1. BORROWS\_FROM entry altered
      1. BookID: 1001
      2. BranchID: 12
      3. Card\_No: 2001
      4. Returned: 1 (true)

# ASSUMPTIONS

In developing the ER diagram for the Library Management System, we made some assumptions regarding ambiguities and some design choices:

* (1) There is no particular author entity mentioned in the assignment details, so we are assuming that the user is manually entering the author into the DB for each book, else, there is a missing requirement regarding the author,
* (2) Assuming ReturnDate is the "Due Date" of when you need to return the book, rather than when the book gets returned, and
* (3) BorrowsFrom relationship acts like an entity since the relationship has attributes and 3 foreign keys (Card\_No, BranchID, BookID).

## No Author Entity

One of the significant assumptions we made concerning the handling of entities, was the omission of the author entity that would have normally been a separate entity, but we decided to have it “combined” with the Book entity.

This approach may simplify the database structure, but it is worth nothing that it may limit the system’s overall ability to efficiently manage or query strictly author-related information

## Assuming The ReturnDate Attribute Is The Due Date

When deciding on the attributes on book’s being loaned, we interpreted the ‘ReturnDate’ field as the due date for the book rather than the actual date of return. This assumption was listed since we had some concerns on if it meant ‘date of return’ or ‘date to return by’ - and came to a consensus that having it be the “date to return by” follows the typical label a book would have. Following this assumption, we allow the management of loans to be tracked for overdue loans.

This interpretation means that the system may not inherently track the actual return date of the books, which could be a small oversight on a more statistical or operational level.

## Creating A BorrowFrom Entity From The Relationship

While creating the ER diagram and translating it to a schema, we decided to treat the ‘BorrowsFrom’ relationship as an entity. Working through the thought process, having the relationship be an associative entity was driven by the presence of attributes specified for “borrowing a book from a library branch” needing to link three separate entities (Borrower, Branch, and Book) through foreign keys (Carn\_no, BranchID, and BookID).

This approach allows for a more flexible representation of borrowing the books, enabling the system to store “transaction” data into an explicitly described database table that avoids assuming implications of connected entities.

# ENTITY RELATIONSHIP DIAGRAM

A diagram of a company

Description automatically generated with medium confidence

**Figure 1**

In the ER diagram above for the Library Management System, we have 6 entities (2 of which are weak entities) and 4 relationships (1 of which is an identifying relationship).

This section will not go in depth on the cardinality ratio; however, it is mentioned in the Mini World Description.

## Publisher Entity

For the Publisher entity, we attached the Name, Address, and PhoneNumber attributes to the Publisher because it was mentioned in one of our requirements, as mentioned below:

Each PUBLISHER has a phone number and an address associated with it.

Our primary key is the Name of the Publisher as the PublisherName is included as one of the attributes in the Book entity. The relationships that are connected to the Publisher entity are the Published By relationship, which relates the Publisher and Book entities by the Publisher’s primary key.

## Book Entity

For the Book entity, we attached the BookID, Title, Author, and PublisherName attributes to the Book entity because it was mentioned in one of our requirements, as mentioned below:

The database keeps track of BOOKs. Each Book has a unique IdNo (assume this is a unique integer generated by the system for each new BOOK such as 1, 2, 3, ...), a title (assume this is a string, such as “The mind of the leader” or “Atomic Habits”), an author, and a publishers name (also a string).

Our primary key is the BookID since the BookID is unique compared to the other attributes in the Book entity. The relationships that are connected to the Book entity are the Published By relationship, which relates the Publisher and Book entities by the Publisher’s primary key, and the Has Multiple relationship, which relates the Book and the Book\_Copies by the Book’s primary key. It is also being used in the Borrow\_From entity as a foreign key.

## Book\_Copies Weak Entity

For the Book\_Copies entity, we attached the BookID, BranchID, and No\_of\_copies attributes to the Book entity because it was mentioned in one of our requirements, as mentioned below:

There are multiple BOOK\_COPIES of every BOOK at different LIBRARY\_BRANCHES.

The foreign keys are both the BookID from the Book entity as well as the BranchID from the Library\_Branches entity. The foreign keys have the dotted underline below as well as the dotted border around the attributes. Our primary key is the combination of both the two foreign keys as they both create the link between the Book\_Copies entity and the Book and Library\_Branches entities. Because No\_of\_copies was the only other attribute that we could choose from which could not be a primary key, we had to get foreign keys from other entities, making this a weak entity. The relationships that are connected to the Book\_Copies entity are the Has Multiple relationship, which relates the Book entity with the Book\_Copies entity, and the Has identifying relationship, which relates the Book\_Copies entity to the Library\_Branches entity.

## Library\_Branches Entity

For the Library\_Branches entity, we attached the BranchID, Name, and Address attributes

In this system there are several LIBRARY\_BRANCHES, each branch has a unique Id, a name, and an address.

Our primary key is the BranchID since the BranchID is both unique while the other attributes in the Library\_Branches entity are not unique. The relationships that are connected to the Library\_Branches entity is the Has identifying relationship, which relates the Book\_Copies entity to the Library\_Branches entity, and the BorrowsFrom relationship, which relates the Borrowers and the Library\_Branches entities as well as the Borrows\_From weak entity.

## Borrows\_From Weak Entity

For the Borrows\_From entity, we attached the BookID, BranchID, Card\_No, CheckedOut, Returned, and ReturnDate attributes to the Book entity because it was mentioned in one of our requirements, as mentioned below:

BORROWERs can borrow BOOKs from any LIBRARY\_BRANCH. The Library Management System keeps track of the BOOKs’ date checked out and the return date and if returned or not.

While it does not directly say what the table is linked to, it suggests that there is another entity that can be created from the three entities mentioned above,

The foreign keys are the BookID, the BranchID, and the Card\_No from the Book, Library\_Branches, and Borrower entities respectively. Our primary key is the combination of all three foreign keys which allows this table to exist since the CheckedOut, Returned, and ReturnDate are not unique to become a primary key. Because of this, this entity is considered a weak entity. The relationships that are connected to the Borrows\_From entity is the BorrowsFrom relationship, which connects to both the Library\_Branches and the Borrowers entities through their primary keys.

## Borrower Entity

For the Borrower entity, we attached the Card\_No, Name, Address, and Phone attributes to the Book entity because it was mentioned in one of our requirements, as mentioned below:

The Library Management System also keeps track of BORROWERs of BOOKs, each BORROWER has a card, name, address, and phone number.

Our primary key is the Card\_No because the attribute is unique and is associated to only one borrower. The relationships that are connected to the Borrower entity are the Library\_Branches by the BorrowsFrom relationship (Not to be confused by the Borrows\_From entity).

# RELATIONAL DATABASE SCHEMA

A screenshot of a computer

Description automatically generated**Figure 2**

After creating the ER diagram described in figure 1, we translated the same design choices into a relational database schema for the Library Management System illustrated in Figure 2. Each entity, such as “BOOK” or “PUBLISHER”, is represented as a table with its columns showing the attributes of the respective entity. For example, the table “LIBRARY\_BRANCH” has columns “BranchId”, “Address,” and “Name” which is a one-to-one mapping of the entity-attribute notation from the ER diagram. The primary keys and foreign keys of each table are identified by underlines beneath the column names. Distinguishing between the two types of keys is through the arrow notation which maps a foreign key back to the primary key it referenced from another table. This design can be seen in the “BOOK\_COPIES” table where its foreign key, “BookId”, has an arrow pointing from it to the “BookId” column of the “BOOK” table which is the primary key it referenced.

In the relational database schema for the Library Management System, we have 6 tables with both primary and foreign keys underlined with the arrows going towards the primary key from the foreign key(s).

## Book Table

* Primary Key: BookID
* Attributes: PublisherName, Author, Title

This table stores information about individual books, each book is uniquely identified by the book’s ID. PublisherName serves as a foreign key linking to the PUBLISHER table, while Author and Title provide details about the book.

## Publisher Table

* Primary Key: Name
* Attributes: PhoneNumber, Address

This table contains information about the book publishers. The Name of the publisher serves as the primary key, while PhoneNumber and Address are included as contact details.

## Book\_Copies Table

* “Composite” Primary Key: BookID, BranchID
* Attribute: No\_of\_copies

The table represents the inventory of books across different library branches. The combination of BookID and BranchID uniquely identifies each record, allowing tracking of how many copies of a specific book are at each branch.

## Library\_Branches Table

* Primary Key: BranchID
* Attributes: Address, Name

This table stores information about different library branches. Each branch is uniquely identified by its BranchID and includes the branch’s Address and Name in it.

## Borrower Table

* Primary Key: Card\_No
* Attributes: Name, PhoneNumber, Address

This table contains information about the library’s borrowers. Each borrower is uniquely identified by their Card\_No and includes personal details of their Name, PhoneNumber, and Address.

## Borrows\_From Table

* “Composite” Primary Key: BookID, BranchID, Card\_No
* Attributes: CheckedOut, ReturnDate, Returned

This table represents the borrowing “transaction” of books. It looks books (by their BookID), library branches (by their BranchID) and borrowers (by their Card\_No). Additional attributes of CheckedOut and ReturnDate are represented by dates and Returned marks the book with a boolean value.

# REFERENCES

* Elmasri, R., & Navathe, S. (2016). *Fundamentals of Database Systems Chapter 3 Slides* (7th ed.). Pearson.
* Elmasri, R., & Navathe, S. (2016). *Fundamentals of database systems* (7th ed.). Pearson.
* Guizani, N. (2024, September 5). *Live Lecture* [Live video recording].