Conceptual Modeling

The DataBasers™

1. Introduction

1.1 Project Overview

The objective of this project is to design and implement a relational database to be used by the staff and clientele of a small public library. It is meant to aid the daily needs of both library staff and clients, providing clients with a convenient source of information on the library's collection and staff with the means to record and track media loans, borrowing trends, and more.

1.2 Scope (Project Boundaries)

This project will include a Project Plan, which summarizes the project objectives and team organization; a Database Requirements document, which indicates the project requirements and preliminary design components; a Conceptual Modeling document, which details the complete conceptual design of the database, including an ER diagram; a Logical Relational Model document, which details the relational schema of the database; and a SQL-based Physical Database Design, which will be represented in a document displaying the process, testing, and results of the database implementation. The project will culminate in a full demonstration of the capabilities of the finished database.

1.3 Glossary

Database: A highly structured and organized digital collection of data.
DBMS: Short for Database Management System, a software that is used to store and manage data through the use of databases.
DCL: Short for Data Control Language, a computer language used to manage privilege and access to users with respect to a database.
DDL: Short for Data Definition Language, a computer language used to create and modify a database system
DML: Short for Data Manipulation Language, a computer language used to add, delete, and manipulate data in a database.
ENUM: A data type that works as a list of constants, often strings of characters.
ER Model: Short for Entity-Relationship model, which is used to create and model the conceptual design of a database system, highlighting key data entities, their attributes, and relationships with other entities.
INT: Short for integer, a whole number.
ISBN: Short for International Standard Book Number, a unique identifier for virtually all books.
ISSN: Short for International Standard Serial Number, a unique identifier for publication issues.
Primary Key: An attribute or set of attributes used to distinguish one entity from another.
Query: A request prepared by a user of a database to a database system. The DBMS interprets the query and prepares and returns the results of the query.

Relational Schema: Describes the logical layout of the database, defining the tables of the database, their columns, and the data types of each column.
SQL: Short for Structured Query Language, a computer language that includes DDL and DML elements. It is used for creating, modifying, and using a database.
varChar: A string of text characters of varying length. Also known simply as a string.

2. ER Modeling Components

2.1 Entities

The data entities that will be included and implemented in this database are included below:

User: A general entity representing all users of the database.
Member: A specialization of the User that represents clients of the library.
Staff: A specialization of the User that represents members of the library staff.
Administrator: A privileged specialization of the User.
Media; An entity representing all specific pieces of media a library contains.
Book: A specialization of the Media entity that represents physical books.
Magazine: A specialization of the Media entity that represents magazine issues.
DigitalMedia: A specialization of the Media entity that represents digital media.
Loan: An entity that represents a loan a member places on a piece of media.
Fine: An entity that represents a fine levied on a user.
Payment: An entity representing the payment of a specific fine.
Reservation: An entity representing a reservation placed by a user on media.

2.2 Attributes

The attributes of each entity are included below:

User:

userID: Unique INT greater than or equal to, primary key of the entity.
name: varChar.
phoneNumber: varChar in a valid phone number format.
emailAddress: varChar in a valid email address format.
physicalAddress: varChar in a valid mailing address format.
userType: ENUM, either “admin”, “staff”, or “member”.
accountStatus: ENUM, either “active” or “inactive”.

Member:

membershipType: Composite attribute, composed of attributes shown below.
typeName: ENUM, either “regular”, “student”, or “senior”.
borrowingLimit: INT greater than zero and set by library policy.
lateFeeRate: DECIMAL greater than zero and set by library policy.

Staff:
The Staff entity needs no additional attributes other than those from the User entity. Privileges will be granted to users of this type in the implementation phase.

Administrator:
The Administrator entity needs no additional attributes other than those from the User entity. Privileges will be granted to users of this type in the implementation phase.

Media:

itemID: unique INT greater than or equal to zero, primary key of the entity.
title: varChar.
itemType: ENUM, either “book”, “digital”, or “magazine”.
publicationYear: INT greater than zero.
availabilityStatus: ENUM, either “available” or “unavailable”.
specialPremium: DECIMAL greater than or equal to zero, set by policy.
specialRestriction: ENUM, either “common” or “rare”, set by policy.

Book:

ISBN: INT greater than zero.
author: varChar.
genre: varChar.

DigitalMedia:

digitalMediaID: INT greater than zero.
creator: varChar.
genre: varChar.

Magazine:

ISSN: INT greater than zero.
publicationDate: DATE that is a valid date.

Loan:

loanID: unique INT greater than or equal to zero, primary key of the entity.
memberID: INT, key of the User entity.
itemID: INT, key of the Media entity.
checkoutDate: DATE.
dueDate: DATE, later than the checkoutDate.

returnDate: DATE, later than the checkoutDate.
lateFeeCharge: DECIMAL, greater than or equal to zero.

Reservation:

reservationID: unique INT greater than or equal to zero, key of the entity.
memberID: INT, key of the User entity.
itemID: INT, key of the Media entity.
reservationDate: DATE.
expirationDate: DATE, on or later than the expiration date as set by policy.
status: ENUM, either "active" or "inactive"

Fine:

fineID: unique INT greater than or equal to zero, key of the entity.
memberID: INT, key of the User entity.
loanID: INT, key of the Loan entity.
amount: DECIMAL, greater than or equal to zero.
issueDate: DATE.
status: ENUM, either "unpaid" or "paid".

Payment:

paymentID: unique INT greater than or equal to zero, key of the entity.
fineID: INT, key of the Fine entity.
memberID: INT, key of the User entity.
paymentDate: DATE.
amountPaid: DECIMAL greater than or equal to zero.

2.3 Relationships

The relationships of this database are defined as follows:

Member Makes Reservation:

There is a one-to-many relationship between the Member and the Reservation. The Member can make zero or more Reservations, and a Reservation can only be made by one Member.

Reservation For Media:

There is a one-to-many relationship between the Media entity and the Reservation entity. A Media item can be reserved many times, and a Reservation can be made for only one media item.

Media Is Loaned In Loan:

There is a one-to-many relationship between the Media entity and the Loan entity. A Media item can be loaned many times, but a given loan can only be associated with one Media item. However, a Media item cannot be loaned if it is currently already on loan.

Member Borrows Loan:
There is a one-to-many relationship between the Member and the Loan. A Member can place many loans in the account lifetime, but a given loan is associated with only one Member. The Member will also have limits on how many loans can be active at one time, as set by library policy.

Fine Is Triggered By Loan:
There is a one-to-one relationship between the Fine and the Loan. A Loan can trigger zero or one Fine, and a specific Fine is triggered by only one Loan. This Fine can grow as the Loan becomes more overdue, so additional Fine instances for one Loan are unnecessary.

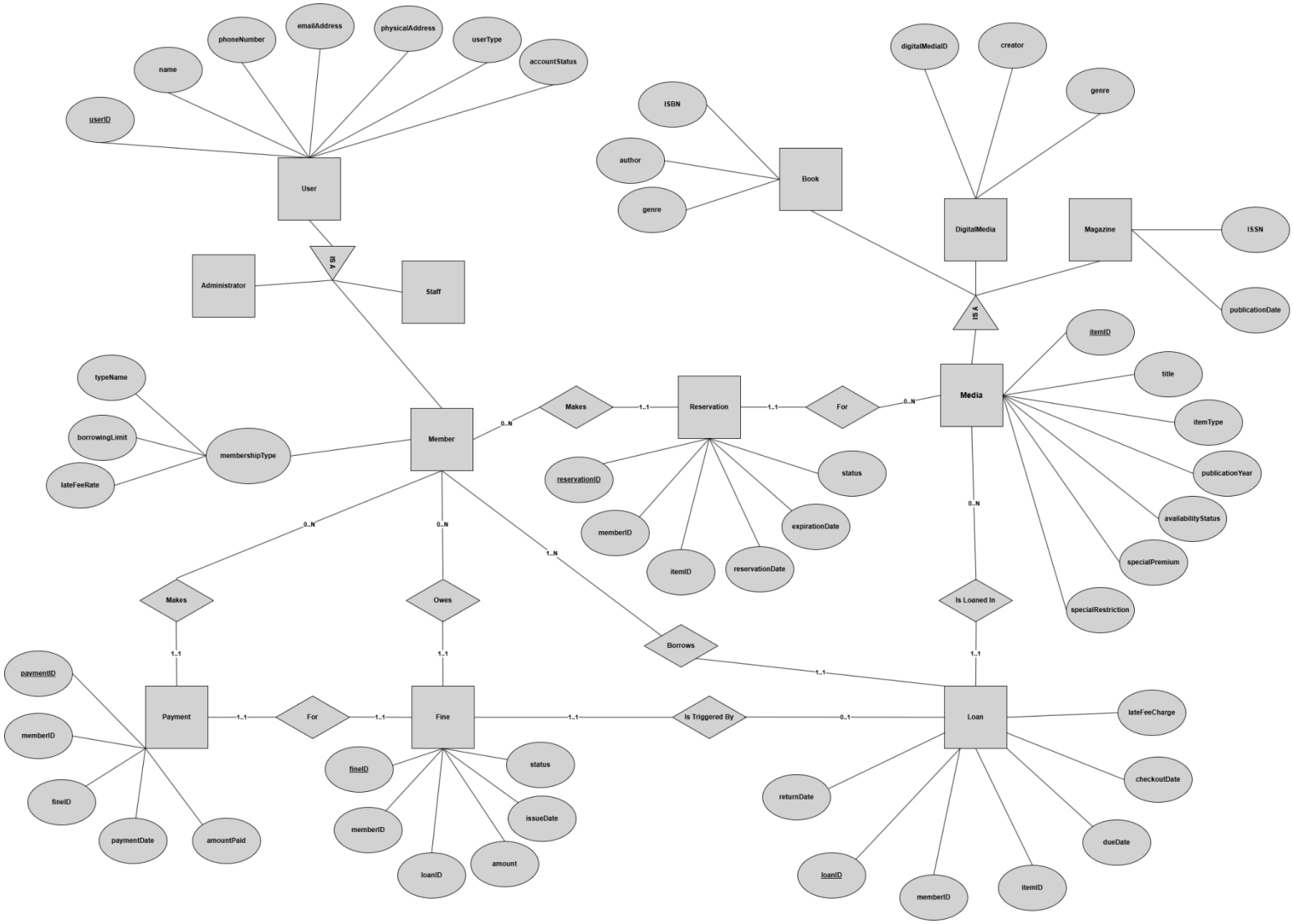
Member Owes Fine:
There is a one-to-many relationship between the Member and Fine entities. A Member can owe several Fines simultaneously, but a Fine can only be owed by one Member.

Member Makes Payment:
There is a one-to-many relationship between the Member and Payment entities. A Member can make several Payments, but a specific Payment can only be made by one Member.

Payment For Fine:
There is a one-to-one relationship between the Payment and Fine entities. One Payment will cover the entire balance of a specific Fine.

3. ER Model

The ER Model for this database is shown below:



4. Appendices

4.1 Library Policies

Borrowing limits based on the membershipType attribute of the Member entity are not typically defined by the database designers but by preexisting library policies. For context, the relevant policies are listed below:

> ***Regular Members:*** Regular members can borrow up to five items at any given time and have a set late fee rate of $4/day.
>
> ***Senior Members:*** Senior members can borrow up to seven items at any given time and have a set late fee rate of $2/day.
>
> ***Student Members:*** Student members can borrow up to ten items at any given time and have a set late fee rate of $1/day.

Additionally, according to library policy, reservations on a media item expire three days after the item has been returned to the library. Rarer items will have customized borrowing restrictions and late fee premiums that will reflect their rarity.

4.2 Staff and Administrators

While the Staff and Administrator entities appear in the ER Model, neither entity is connected with other entities in the model. An ER model is designed to represent the relevant relationships between different entities in a situational context. However, it focuses on relationships that require recording, specifically in tables in a database. For instance, while a Member may eventually use the database to search for a specific book, a "search" relationship is not shown in the ER diagram since recording the specific searches of each Member is undesired and unnecessary. Similarly, while the Staff and Administrators may perform actions on the database, such as performing the creation of a Loan record during the checkout process, maintaining records on which Staff members created which records is beyond the scope of the Requirements Specification. Since a registry of Staff and Administrator users is useful for the Database, the entities are included in the ER model. In the implementation phase, DCL will be used to give users in these tables the proper privileges to perform their functions.