Introduction

Project Overview

This project aims to develop a database system for a small library, enhancing scalability, item management, membership tracking, borrow tracking, and report generation. The system will deliver these features through an intuitive and user-friendly interface.

Scope

The system will enable efficient cataloging and tracking of loanable items, enforce borrowing policies, and generate detailed reports to support library management. Its core functionalities will include item management, membership management, borrowing and returning processes, and query and report generation, all while ensuring database integrity.

Glossary

- **UI (User Interface)**
  - A User Interface is a high level application that allows an end user to interact with a system.
- **3 Schema Architecture**
  - A database design approach that splits data views into external, conceptual and internal layers.
- **Documentation**
  - An official piece of information that provides internal and external parties information about a topic.

- **Internal Stakeholders**
    - People or groups inside a company that have a direct interest in the success of a project.
- **External Stakeholders**
    - People or groups outside of a company that have an indirect interest in the success of a project.
- **Functional Requirements**
    - Requirements for a system that specify what the system should be able to do. This includes its functions, behaviors, and operations.
- **Data Entities**
    - Data entities are an abstraction away from fully implemented tables. They specify attributes, data-types, and constraints from a high level view.
- **Entity Attributes**
    - A characteristic or trait of an entity type that describes the entity, for example, the Person entity type has the Date of Birth attribute
- **ER (entity-relationship) Diagram**
    - An entity–relationship model which describes interrelated things of interest in a specific domain of knowledge.
- **End User**
    - A user that interacts with the top level form of an application. (e.g. a library user interacting with the library database by using a system)

Identify ER Modeling Components

Entities and Their Attributes:

1. **Book:** Represents books in the library system

Attribute	Data Type	Constraints
book_id	INT	PRIMARY KEY, UNIQUE
title	VARCHAR(500)	NOT NULL
author	VARCHAR(500)	NOT NULL
isbn	VARCHAR(13)* ISBNs are 13 digits long	UNIQUE, NOT NULL
publication_year	YEAR	NOT NULL
genre	VARCHAR(500)* *update with longest genre classification	NOT NULL
avaliability_status	ENUM(‘Available’, ‘CheckedOut’, ‘Reserved’)	DEFAULT ‘Available’
popularity	INT	*tracks amount of times book has been checked out

2. **Media:** Represents digital media in the library system

Attribute	Data Type	Constraints
media_id	INT	PRIMARY KEY, UNIQUE
title	VARCHAR(500)	NOT NULL
creator	VARCHAR(500)	NOT NULL
media_type	ENUM(‘E-book’, ‘Audiobook’, ‘Video’)	NOT NULL

isbn	VARCHAR(13)* ISBNs are 13 digits long	NOT NULL only if media_type = ‘E-book’ (CHECK)
publication_year	YEAR	NOT NULL
genre	VARCHAR(65535)* *update with longest genre classification	NOT NULL
avaliability_status	ENUM(‘Available’, ‘CheckedOut’, ‘Reserved’)	DEFAULT ‘Available’
popularity	INT	*tracks amount of times item has been checked out

3. **Magazine:** Represents magazines in the library system

Attribute	**Data Type**	**Constraints**
magazine_id	INT	PRIMARY KEY, UNIQUE
title	VARCHAR(500)	NOT NULL
issue_number	VARCHAR(20)	NOT NULL
publication_date	DATE	NOT NULL
avaliability_status	ENUM(‘Available’, ‘CheckedOut’, ‘Reserved’)	DEFAULT ‘Available’
popularity	INT	*tracks amount of times magazine has been checked out

4. **Client:** Represents clients in the library system

Attribute	**Data Type**	**Constraints**
client_id	INT	PRIMARY KEY, UNIQUE
name	VARCHAR(500)	NOT NULL
contact_info	VARCHAR(500)	NOT NULL
membership_type	ENUM(‘Regular’, ‘Student’, ‘Senior’)	FOREIGN KEY REFERENCES

		Membership_Type(Type), DEFAULT ‘Regular’
account_status	ENUM(‘Active’, ‘Inactive’)	DEFAULT ‘Active’

5. **Membership_Type:** Represents membership types in the library system

Attribute	**Data Type**	**Constraints**
type	ENUM(‘Regular’, ‘Student’, ‘Senior’)	PRIMARY KEY
borrowing_limit	INT	CHECK (borrowing_limit > 0), NOT NULL
daily_late_fee	DECIMAL(5, 2)	CHECK (daily_late_fee >= 0), NOT NULL
extra_fees	DECIMAL(5, 2)	CHECK (extra_fees >= 0), used if book/item is lost or damaged

6. **Loan:** Represents item loans in the library system

Attribute	**Data Type**	**Constraints**
loan_id	INT	PRIMARY KEY, UNIQUE
client_id	INT	FOREIGN KEY REFERENCES Client(client_id), NOT NULL
item_type	ENUM(‘Book’, ‘E-book’, ‘Audiobook’, ‘Video’, ‘Magazine’)	NOT NULL
item_id	INT	FOREIGN KEY REFERENCES Media(media_id)
book_id	INT	FOREIGN KEY REFERENCES Book(book_id)
magazine_id	INT	FOREIGN KEY REFERENCES Magazine(magazine_id)

borrow_date	ENUM(‘Active’, ‘Inactive’)	DEFAULT ‘Active’
due_date	DATE	NOT NULL
return_date	DATE	
fees_accrued	DECIMAL(5, 2)	CHECK(fees_accrued >= 0)

7. **Reservation:** Represents item reservations in the library system

Attribute	**Data Type**	**Constraints**
reservation_id	INT	PRIMARY KEY, UNIQUE
client_id	INT	FOREIGN KEY REFERENCES Client(client_id), NOT NULL
item_type	ENUM(‘Book’, ‘E-book’, ‘Audiobook’, ‘Video’, ‘Magazine’)	NOT NULL
item_id	INT	FOREIGN KEY REFERENCES Media(media_id)
book_id	INT	FOREIGN KEY REFERENCES Book(book_id)
magazine_id	INT	FOREIGN KEY REFERENCES Magazine(magazine_id)
reservation_date	DATE	NOT NULL
status	ENUM(‘Ready for pickup’, ‘In line’, ‘Processing’)	DEFAULT ‘Processing’
place_in_line	INT	NOT NULL

8. **Notification:** Represents notifications in the library system

Attribute	**Data Type**	**Constraints**
notification_id	INT	PRIMARY KEY, UNIQUE

client_id	INT	FOREIGN KEY REFERENCES Client(client_id)
message	TEXT(500)	NOT NULL

Relationships Between Entities:

Client and Membership_Type

- Relationship: A client has a membership.
- Multiplicity: One-to-many, a client can have exactly one membership type, and each membership type can be assigned to multiple clients.
- Constraints:
    - Client.membership_type references Membership_Type.type (foreign key relationship).
    - A client’s membership_type is required and constrained to one of the valid types (Regular, Student, Senior).
    - The membership_type attribute in the Client table should be a foreign key.

Client and Loan

- Relationship: A client has loans.
- Multiplicity: One-to-many, a client can have multiple loans, but each loan record is associated with exactly one client.
- Constraints:
    - Loan.client_id references Client.client_id (foreign key).
    - In a loan, the client may borrow either a book, media item, or magazine, so book_id, media_id, and magazine_id are mutually exclusive.

Book and Loan

- Relationship: A book can be loaned.
- Multiplicity: One-to-many, a book can be borrowed by many clients (loaned multiple times), but each loan record can only reference one book at a time.
- Constraints:
    - Loan.book_id references Book.book_id (foreign key).
    - The item_type in Loan should refer to the book when book_id is populated.

Media and Loan

- Relationship: A media item can be loaned.
- Multiplicity: One-to-many, a media item can be loaned multiple times, but each loan record is associated with only one media item.
- Constraints:
    - Loan.item_id references Media.media_id (foreign key).

    - The item_type in Loan should refer to the media item when item_id is populated.

Magazine and Loan

- Relationship: A magazine can be loaned.
- Multiplicity: One-to-many, a magazine can be borrowed multiple times, but each loan record is associated with only one magazine.
- Constraints:
    - Loan.magazine_id references Magazine.magazine_id (foreign key).
    - The item_type in Loan should refer to the magazine when magazine_id is populated.

Client and Reservation

- Relationship: A client has reservations
- Multiplicity: One-to-many, a client can have multiple reservations, but each reservation is linked to one specific client.
- Constraints:
    - Reservation.client_id references Client.client_id (foreign key).

Book and Reservation

- Relationship: A book can be reserved
- Multiplicity: One-to-many, a book can have multiple reservations, but each reservation is linked to one specific book.
- Constraints:
    - Reservation.book_id references Book.book_id (foreign key).
    - The item_type in Reservation should refer to the book when book_id is populated.

Media and Reservation

- Relationship: Media is reserved.
- Multiplicity: One-to-many, a media item can have multiple reservations, but each reservation is linked to one specific media item.
- Constraints:
    - Reservation.item_id references Media.media_id (foreign key).
    - The item_type in Reservation should refer to the media item when item_id is populated.

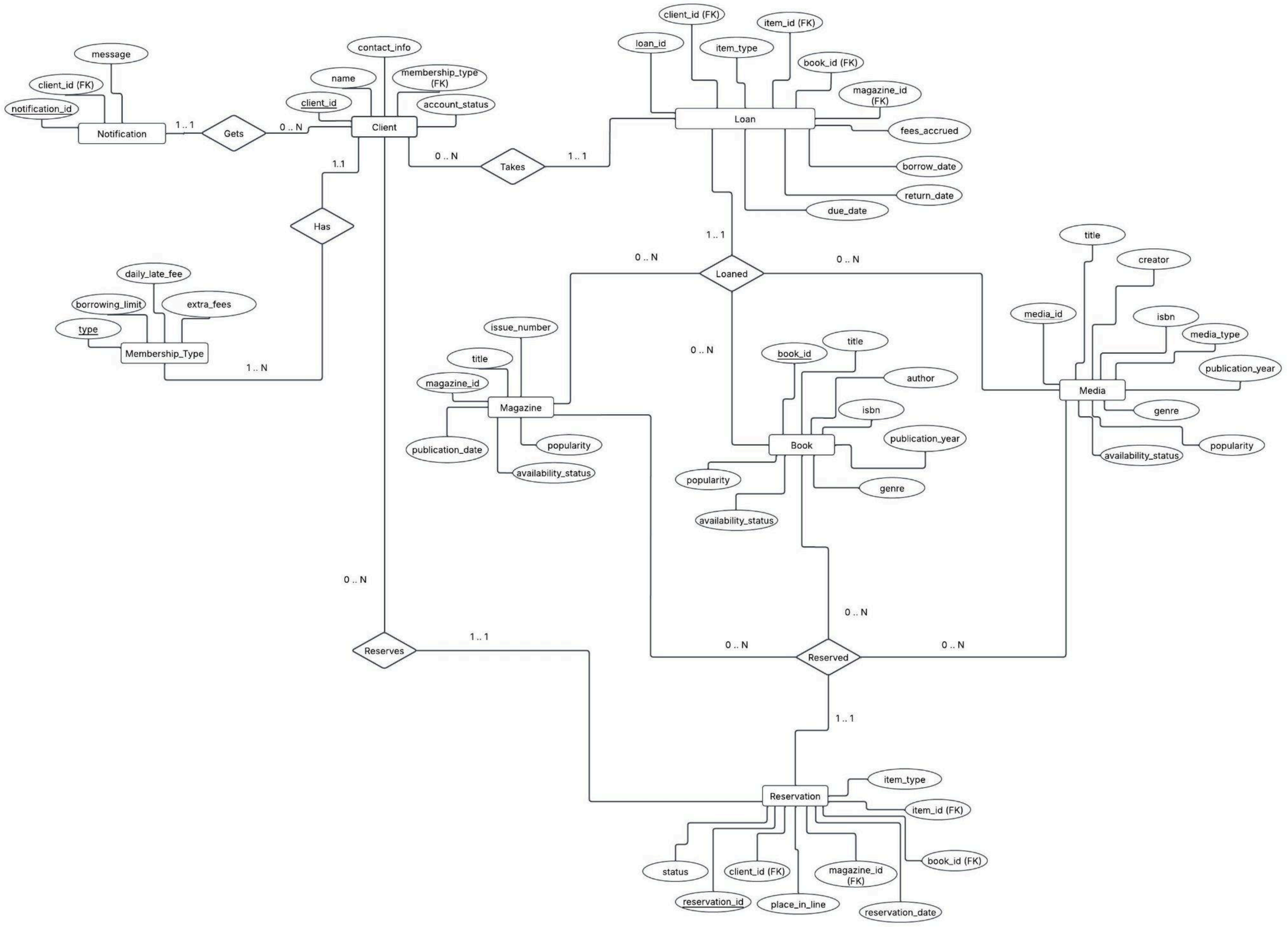
Magazine and Reservation

- Relationship: A magazine is reserved.
- Multiplicity: One-to-many, a magazine can have multiple reservations, but each reservation is linked to one specific magazine.
- Constraints:
    - Reservation.magazine_id references Magazine.magazine_id (foreign key).
    - The item_type in Reservation should refer to the magazine when magazine_id is populated.

Client and Notification

- Relationship: A client receives a notification
- Multiplicity: Many-to-one, a client can have multiple notifications, but each notification is linked to one client.
- Constraints:
    - Notification.client_id references Client.client_id (foreign key).

ER Model


message
client_id (FK)
notification_id
Notification
1 .. 1
Gets
0 .. N
contact_info
name
membership_type (FK)
client_id
account_status
Client
1.1
Has
daily_late_fee
borrowing_limit
extra_fees
type
Membership_Type
1 .. N
0 .. N
Takes
1 .. 1
client_id (FK)
item_id (FK)
loan_id
item_type
book_id (FK)
magazine_id (FK)
Loan
fees_accrued
borrow_date
return_date
due_date
1 .. 1
0 .. N
Loaned
0 .. N
title
creator
media_id
isbn
media_type
publication_year
Media
genre
popularity
availability_status
issue_number
title
magazine_id
Magazine
publication_date
popularity
availability_status
0 .. N
book_id
title
author
isbn
publication_year
Book
popularity
genre
availability_status
0 .. N
Reserves
1 .. 1
0 .. N
0 .. N
Reserved
0 .. N
1 .. 1
item_type
Reservation
item_id (FK)
book_id (FK)
status
client_id (FK)
magazine_id (FK)
reservation_id
place_in_line
reservation_date