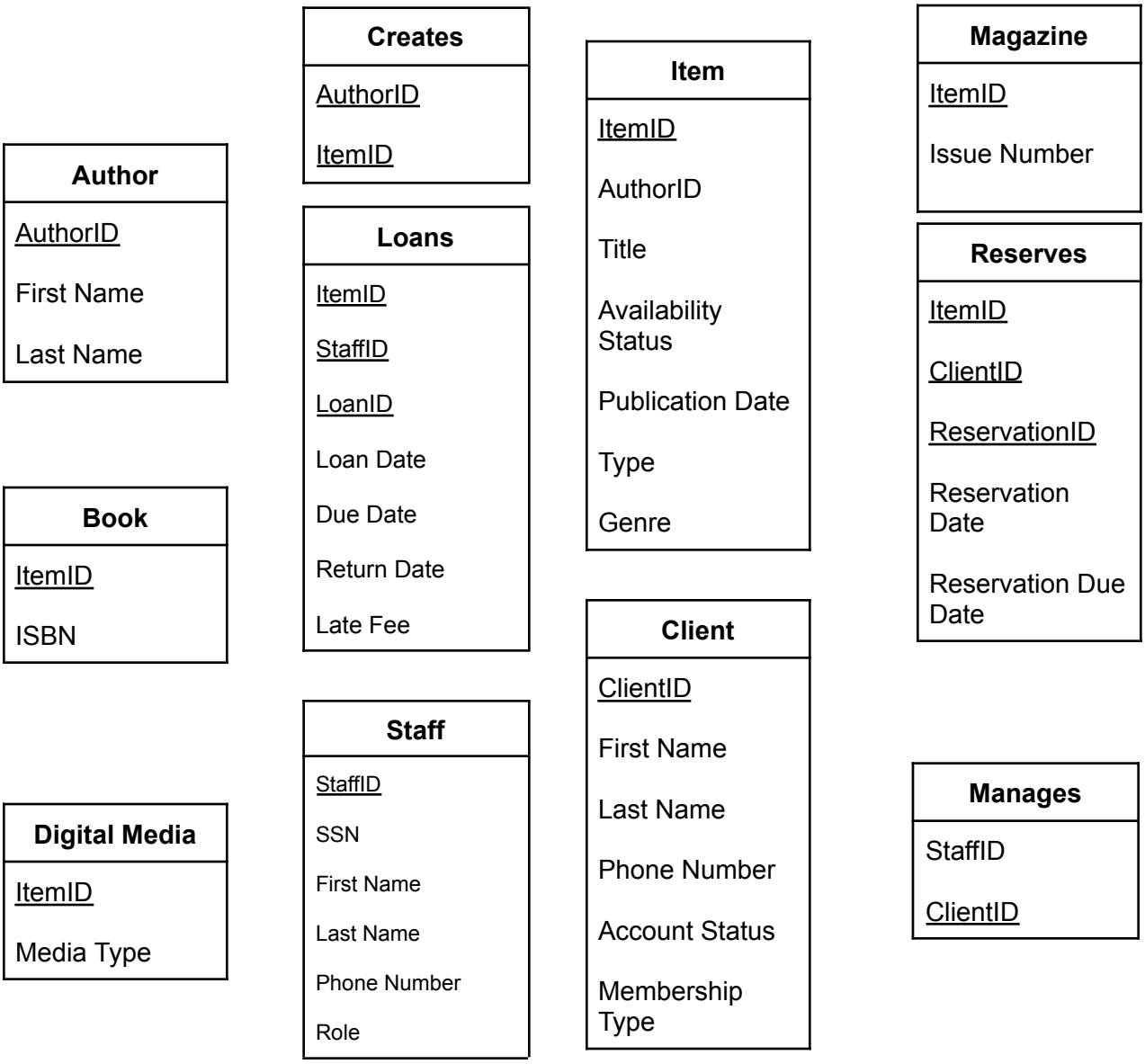
Introduction:

Project Overview: The system will manage a diverse collection of loanable items, track various types of memberships, enforce borrowing rules, and generate meaningful reports.

Scope Statement: From the project overview: This project aims to provide hands-on experience designing, implementing, and managing a relational database system for a small library. The system will manage a diverse collection of loanable items, track various types of memberships, enforce borrowing rules, and generate meaningful reports.

Relational Schema Mapping:



Data Dictionary:

Author			
Attribute Name	**Data Type**	**Domain**	**Constraints**
AuthorID	INT	All positive integers	Primary Key, Auto increments
First Name	VARCHAR	String of up to 255 characters	Not Null
Last Name	VARCHAR	String of up to 255 characters	Not Null

Book			
Attribute Name	**Data Type**	**Domain**	**Constraints**
ItemID	INT	All positive integers	Primary and Foreign Key
ISBN	VARCHAR	Valid ISBN-13 Format	Not Null

Client			
Attribute Name	**Data Type**	**Domain**	**Constraints**
ClientID	INT	All positive integers	Primary Key, Auto increments
First Name	VARCHAR	String of up to 255 characters	Not Null
Last Name	VARCHAR	String of up to 255 characters	Not Null
Phone Number	CHAR(10)	String of exactly 10 numbers	Not Null
Account Status	VARCHAR	Active, Suspended, or Pending Deletion	Not Null

Membership Type	VARCHAR	Regular, Student, or Senior	Not Null

Staff			
Attribute Name	**Data Type**	**Domain**	**Constraints**
StaffID	INT	All positive integers	Primary Key, Auto increments
First Name	VARCHAR	String of up to 255 characters	Not Null
Last Name	VARCHAR	String of up to 255 characters	Not Null
Phone Number	CHAR(10)	String of exactly 10 numbers	Not Null
Role	VARCHAR	Employee, Manager, or Owner	Not Null

Digital Media			
Attribute Name	**Data Type**	**Domain**	**Constraints**
ItemID	INT	All positive integers	Primary and Foreign Key
Media Type	VARCHAR	Audiobook, Movie, Podcast, Pictures	Not Null

Item			
Attribute Name	**Data Type**	**Domain**	**Constraints**
ItemID	INT	All positive integers	Primary Key, Auto increments

Title	VARCHAR	String of up to 255 characters	Not Null
Availability Status	BOOLEAN	True or False	Not Null
Publication Date	DATE	Valid Date Format	Not Null
Type	VARCHAR	Digital or Physical	Not Null
Genre	VARCHAR	String from a predefined list of genres	Not Null
AuthorID	INT	All positive integers	Foreign Key

Magazine			
Attribute Name	**Data Type**	**Domain**	**Constraints**
ItemID	INT	All positive integers	Primary and Foreign Key
Issue Number	INT	All positive integers	Not Null

Loans			
Attribute Name	**Data Type**	**Domain**	**Constraints**
ItemID	INT	All positive integers	Primary and Foreign Key
StaffID	INT	All positive integers	Primary and Foreign Key
LoanID	INT	All positive integers	Primary Key, Auto increments
Loan Date	DATE	Valid Date Format	Not Null
Due Date	DATE	Valid Date Format	Not Null
Return Date	DATE	Valid Date Format	
Late Fee	DECIMAL	All positive decimals	

Reserves

Attribute Name	Data Type	Domain	Constraints
ItemID	INT	All positive integers	Primary and Foreign Key
ClientID	INT	All positive integers	Primary and Foreign Key
ReservationID	INT	All positive integers	Primary Key, Auto increments
Reservation Date	DATE	Valid Date Format	Not Null
Reservation Due Date	DATE	Valid Date Format	Not Null

Creates			
Attribute Name	**Data Type**	**Domain**	**Constraints**
AuthorID	INT	All positive integers	Primary and Foreign Key
ItemID	INT	All positive integers	Primary and Foreign Key

Manages			
Attribute Name	**Data Type**	**Domain**	**Constraints**
StaffID	INT	All positive integers	Foreign Key
ClientID	INT	All positive integers	Primary and Foreign Key

Data Definition Language:

```
CREATE TABLE Author (
        AuthorID INT PRIMARY KEY AUTO_INCREMENT,
        FirstName VARCHAR(255) NOT NULL,
```

```
        LastName VARCHAR(255) NOT NULL,
);

CREATE TABLE Item (
        ItemID INT PRIMARY KEY AUTO_INCREMENT,
        Title VARCHAR(255) NOT NULL,
        AvailabilityStatus BOOLEAN NOT NULL,
        PublicationDate DATE,
        AuthorID INT,
        Type VARCHAR(10) NOT NULL CHECK (Type IN ('Physical', 'Digital')),
        Genre VARCHAR,
        FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID)
);

CREATE TABLE Book (
        ItemID INT Primary Key,
        ISBN VARCHAR(255) NOT NULL,
        FOREIGN KEY (ItemID) REFERENCES Item(ItemID)
);

CREATE TABLE Magazine (
        ItemID INT Primary Key,
        IssueNumber INT NOT NULL,
        FOREIGN KEY (ItemID) REFERENCES Item(ItemID)
);

CREATE TABLE DigitalMedia (
        ItemID INT Primary Key,
        MediaType VARCHAR(20) NOT NULL CHECK (MediaType IN (‘Audiobook’, ‘Movie’,
‘Podcast’, ‘Pictures’)),
        FOREIGN KEY (ItemID) REFERENCES Item(ItemID)
);

CREATE TABLE Creates (
   AuthorID INT,
   ItemID INT,
   PRIMARY KEY (AuthorID, ItemID),
   FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID),
   FOREIGN KEY (ItemID) REFERENCES Item(ItemID)
);

CREATE TABLE Client(
        ClientID INT PRIMARY KEY AUTO_INCREMENT,
        FirstName VARCHAR(255) NOT NULL,
```

```
        LastName VARCHAR(255) NOT NULL,
        PhoneNumber CHAR (10) NOT NULL,
        AccountStatus VARCHAR(20) NOT NULL CHECK (AccountStatus IN ('Active',
'Suspended', 'Pending Deletion')),
        MembershipType VARCHAR(10) NOT NULL CHECK (MembershipType IN ('Regular',
'Student',' Senior')),
);

CREATE TABLE Loans(
        LoanID INT PRIMARY KEY AUTO_INCREMENT,
        StaffID INT,
        ItemID INT,
        LoanDate DATE NOT NULL,
        DueDate DATE NOT NULL,
        ReturnDate DATE,
        LateFee DECIMAL,
        FOREIGN KEY (StaffID) REFERENCES Staff(StaffID),
        FOREIGN KEY (ItemID) REFERENCES Item(ItemID)
);

CREATE TABLE Reserves(
        ReservationID INT PRIMARY KEY AUTO_INCREMENT,
        ClientID INT,
        ItemID INT,
        ReservationDate DATE NOT NULL,
        ReservationDueDate DATE NOT NULL,
        FOREIGN KEY (ClientID) REFERENCES Client(ClientID),
        FOREIGN KEY (ItemID) REFERENCES Item(ItemID)
);

CREATE TABLE Manages(
        ClientID INT PRIMARY KEY,
        StaffID INT,
        FOREIGN KEY (StaffID) REFERENCES Staff(StaffID),
```

```
        FOREIGN KEY (ClientID) REFERENCES Client(ClientID)
);

CREATE TABLE Staff (
        StaffID INT PRIMARY KEY AUTO_INCREMENT,
        FirstName VARCHAR(100) NOT NULL,
        LastName VARCHAR(100) NOT NULL,
        PhoneNumber CHAR(10) NOT NULL,
        Role VARCHAR(10) NOT NULL CHECK (Role IN ('Employee', 'Manager', 'Owner')
);
```