|  |  |
| --- | --- |
| Library Database | |
| Databases Project Year 1 Semester 2 | |
| Date | Week 11 – Week 13 |
| Purpose | Design and implement a Database for a Library |
| Report by | Jakub Orlowski / Comp1A-Y R00267077 / |

## Objective 1:

List the entities, Attributes, Relationships and Multiplicities.

## Objective 1, Solution(1)

To begin I will establish the definitions of Entities, Attributes, Relationships and Multiplicities.

Entity: An entity is a real-world object, person etc. which you wish to store information about, such as a book in the case of this database.

Attribute: An attribute is a characteristic or a property of an entity. It describes a piece of information for example the Author of a book, where book is a entity

Relationship: A relationship describes how two or more entities are associated with each other. For example a “borrow” relationship between the member and book entities.

Multiplicity: Defines the number of instances of one entity that can be related to the number of instances of another entity

* (1:1) – One instance of entity A is related to at most one instance of a entity B
* (1:N) – One instance of entity A is related to zero, one or many instance of entity B, But one instance of entity B can be related to at most one instance of entity A
* (N:1) – Many instances of A can be related to one instance of B. (Reverse of 1:N)
* (N:N) – Many instances of A can related to many instances of B.

## Objective 1, Solution(2)

Using these definitions. I would like to identify and envision the Library as such:

**Entities:**

Entity: Book

* “BookID” – (INT, Primary Key, Unique) – Unique identifier of each book
* “Title” – (VARCHAR, Not Null) – Title of a book
* “Author” – (VARCHAR, Not null) – The author of the associated book
* “Year” – (INT) – The release year of the book
* “Section” – (VARCHAR , Not Null) – Fiction, Crime, Comedy etc..

Entity: Member

* “MemberID” – (INT, Primary Key, Unique) – Unique identifier of each member
* “FirstName” – (VARCHAR, Not Null) – First name of each member
* “LastName” – (VARCHAR, Not Null) – Last name of each member
* “Address” – (VARCHAR) – Address of a member
* “PhoneNum” – (VARCHAR) – Phone Number of a member
* “Email” – (VARCHAR) – Email of a member
* “MemberType” – (VARCHAR, Not Null) – Type of member (Child, Adult)
* “OverdueCost” – (Decimal(10,2), Not Null, Default 0.00) – Current total overdue costs of a member

Entity: Loan

* “LoanID” – (INT, Primary Key, Unique) – Unique number for a loan transaction
* “BookID” – (INT, Foreign Key referencing Book.BookID, Not Null) – The id of the book being loaned
* “MemberID” – (INT, Foreign Key referencing Member.MemberID, Not Null) – The id of the member who is loaning
* “LoanDate” – (DATE, Not Null) – The date of the loan
* “DueDate” – (DATE, Not Null) - The date of the loan due date
* “ReturnDate” – (DATE) – The date the book was returned, can be null if not yet returned

## Objective 1, Solution(3)

**Relationships and Multiplicities**

Member borrows Book

* A member can borrow zero or many Book (1:N)
* A book can be borrowed by zero or many members (1:N)
* The “Loan” entity resolves this (N:N) relationship

Book is a part of a Section

* A book belongs to only one section (1:1)
* A section can contain zero or many Books (1:N)
* This is represented by the “Section” attribute in the “Book” entity

Loan involves Book

* A loan involves only 1 book at a time (1:1)
* A book can be involved in zero or many loans (Over time) (1:N)

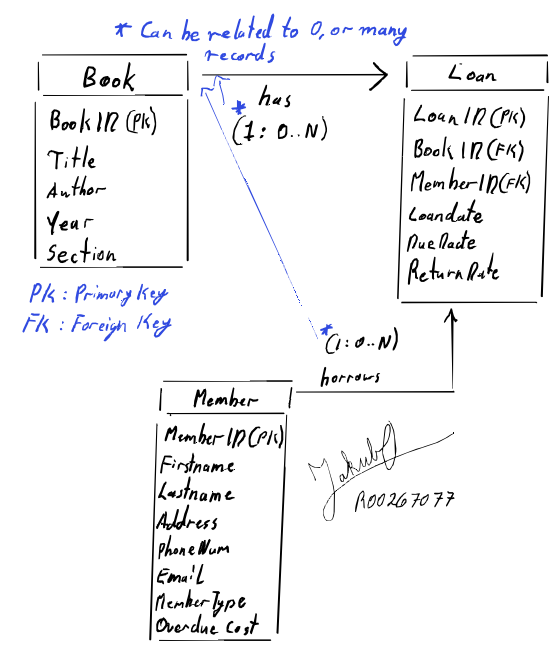
Loan is associated with Member

* A loan is associated with only one member at a time (1:1)
* A member can have zero or many loans (Current and Past) (1:N)

## Objective 2

Draw an ER diagram using the information from Objective 1.

## Objective 2 Solution:

**Explanation**

Entities in boxes  
- Each of which have their attributes listed

Lines connecting label the relationships. Has from book to loan and borrows from member to loan

Cardinality notation where the entity can be related to something zero or many times 1: 0..N

In essence

A “book” can be associated with zero or many “loan” records Each loan record is for one particular “book”

A “member” can be associated with zero or many "loan" records. Each specific ”loan” record is associated with one particular “member”

## Objective 3

Implement the design from steps 1 and 2 on your MySQL server.

## Objective 3 Solution:

Script used for making the table

**-- Creating the Member table**

**CREATE TABLE Member (**

**MemberID INT AUTO\_INCREMENT PRIMARY KEY,**

**FirstName VARCHAR(255) NOT NULL,**

**LastName VARCHAR(255) NOT NULL,**

**Address VARCHAR(255),**

**PhoneNumber VARCHAR(20),**

**Email VARCHAR(255),**

**MemberType ENUM('Adult', 'Child') NOT NULL,**

**OverdueCost DECIMAL(10, 2) NOT NULL DEFAULT 0.00,**

**-- Candidate Key: Email (assuming unique emails are desired)**

**UNIQUE KEY unique\_email (Email)**

**);**

**-- Creating the Book table**

**CREATE TABLE Book (**

**BookID INT AUTO\_INCREMENT PRIMARY KEY,**

**Title VARCHAR(255) NOT NULL,**

**Author VARCHAR(255) NOT NULL,**

**Year INT,**

**Section ENUM('Fiction', 'Crime', 'True Stories', 'History', 'Biographies', 'Romance', 'Sport', 'Non-Fiction') NOT NULL,**

**-- Candidate Key: Title, Author, Year (assuming this combination uniquely identifies a book edition)**

**UNIQUE KEY unique\_book (Title, Author, Year)**

**);**

**-- Creating the Loan table**

**CREATE TABLE Loan (**

**LoanID INT AUTO\_INCREMENT PRIMARY KEY,**

**BookID INT NOT NULL,**

**MemberID INT NOT NULL,**

**LoanDate DATE NOT NULL,**

**DueDate DATE NOT NULL,**

**ReturnDate DATE,**

**-- Foreign Key constraints to enforce relationships**

**FOREIGN KEY (BookID) REFERENCES Book(BookID) ON DELETE RESTRICT ON UPDATE CASCADE,**

**FOREIGN KEY (MemberID) REFERENCES Member(MemberID) ON DELETE RESTRICT ON UPDATE CASCADE**

**);**  
--------------------------------------------

Note that this code was done in MySQL workbench and copy pasted here

I don’t think this code needs explanation as it is very self-explanatory. However the last 2 lines which constrain foreign keys are in place so that if the foreign key is changed in the original table it’ll update across the board.

The on delete constraints don’t allow a book with a active loan to be deleted, and a member to be deleted if they have a active loan.

## Objective 4

Devise sample data, min. 4 rows for each table. Include in script form

## Objective 4 Solution:

The script below provides example data for all the tables.

**-- Sample data for the Member table**

**INSERT INTO Member (FirstName, LastName, Address, PhoneNumber, Email, MemberType, OverdueCost) VALUES**

**('Alice', 'Smith', '12 Main Street, Anytown', '087-1234567', 'alice.smith@email.com', 'Adult', 0.00),**

**('Bob', 'Johnson', '45 Oak Avenue, Anytown', '086-9876543', 'bob.johnson@email.com', 'Adult', 5.50),**

**('Charlie', 'Brown', '7 Pine Lane, Anytown', '083-1122334', 'charlie.brown@email.com', 'Child', 0.00),**

**('Diana', 'Miller', '10 Elm Road, Anytown', '085-4455667', 'diana.miller@email.com', 'Adult', 2.00);**

**-- Sample data for the Book table**

**INSERT INTO Book (Title, Author, Year, Section) VALUES**

**('The Great Novel', 'Jane Doe', 2020, 'Fiction'),**

**('Mystery Solved', 'John Smith', 2018, 'Crime'),**

**('True Tales of Adventure', 'Peter Jones', 2022, 'True Stories'),**

**('Ancient Civilizations', 'Laura Williams', 2015, 'History');**

**-- Sample data for the Loan table**

**INSERT INTO Loan (BookID, MemberID, LoanDate, DueDate, ReturnDate) VALUES**

**(1, 1, '2025-04-10', '2025-04-24', '2025-04-23'),**

**(2, 2, '2025-04-15', '2025-04-29', NULL),**

**(3, 3, '2025-04-01', '2025-04-15', '2025-04-14'),**

**(4, 1, '2025-04-20', '2025-05-04', NULL);**

**INSERT INTO Member (FirstName, LastName, Address, PhoneNumber, Email, MemberType, OverdueCost) VALUES**

**('Eve', 'Davis', '22 Willow Drive, Anytown', '089-7788990', 'eve.davis@email.com', 'Child', 1.00),**

**('Frank', 'Garcia', '5 Birch Court, Anytown', '087-3344556', 'frank.garcia@email.com', 'Adult', 0.00),**

**('Grace', 'Hall', '18 Maple Avenue, Anytown', '086-6677889', 'grace.hall@email.com', 'Adult', 3.75),**

**('Henry', 'Irwin', '3 Oak Street, Anytown', '083-9900112', 'henry.irwin@email.com', 'Child', 0.50);**

**INSERT INTO Book (Title, Author, Year, Section) VALUES**

**('Art of The Deal', 'Donald Trump', 2025, 'Fiction'),**

**('The American Constitution', 'Founding Fathers', 1787, 'Fiction'),**

**('Trumps Auto-Biography', 'Donald Trump', 2020, 'Crime'),**

**('Exploring the Cosmos', 'Neil deGrasse Tyson', 2019, 'Non-Fiction');**

**INSERT INTO Loan (BookID, MemberID, LoanDate, DueDate, ReturnDate) VALUES**

**(5, 4, '2025-04-05', '2025-04-19', '2025-04-18'),**

**(6, 3, '2025-04-22', '2025-05-06', NULL),**

**(7, 1, '2025-03-28', '2025-04-11', '2025-04-10'),**

**(8, 5, '2025-04-12', '2025-04-26', NULL);**

Keep in mind that this script was copy pasted from MySQL workbench

I don’t see any reason to explain this script

## Objective 5

Write 5 SQL multi-table join queries that demonstrate the functionality of the database design. All tables should be in the queries

## Objective 5 (Solution):

-- Query 1 List all loans with member names and book titles

*Select l.LoanID, m.FirstName, m.lastname, b.title, b.section, l.loandate, l.duedate, l.returndate*

*from Loan l, member m, book b*

*where l.memberID = m.memberID*

*and l.bookID = b.BookID;*

-- Query 2 List all books currently on loan, including member contact details

*select b.Title, b.section, b.Author, m.FirstName, m.LastName, m.PhoneNumber, m.Email, l.LoanDate, l.DueDate*

*from book b, member m, loan l*

*where b.bookid = l.bookid*

*and l.memberid = m.memberid*

*and l.returndate is null;*

-- Query 3: List all members and the titles of the books they have due

*select m.firstname, m.lastname, b.title, l.duedate, b.section*

*from member m, loan l, book b*

*where m.memberid = l.memberid*

*and l.bookid = b.bookid*

*and l.returndate is null*

-- Query 4 list all books, authors and the number of times theyve been loaned out

*select b.title, b.section, COUNT(l.loanid) as LoanCount*

*from book b, loan l*

*where b.bookid = l.bookid*

*group by b.bookid, b.title, b.author, b.section*

*order by LoanCount desc;*

-- Query 7: List the first name of adult members who have had a 'fiction' section book loaned, along with the title of that book and the original loan date.

*select m.firstname, b.title, b.section, l.loandate*

*from member m, loan l, book b*

*where m.memberid = l.memberid*

*and l.bookid = b.bookid*

*and m.membertype = 'Adult'*

*and b.section = 'fiction'*

## Conclusion and Statement

In conclusion I would like to propose this database format, workout for the given question and task to create a database for a library

I would like to confirm that all the work in this project is my own, that has been supplemented with lecture notes, internet access and has been permitted by the project description.

All script code has been run by me on MYSQL such that I am certain it all works

Jakub Orlowski

25/04/2025

Year 1 Semester 2

Introduction to Databases – Lecturer “Arthur Tobin”