Database Project

By: Ammar Amin 20104536

Hassan Mikawi 20104451

Omar Mohamed Anwar 20104420

To: Dr. Ashraf Tammam and TA Dalia Refaat

Library Management System

**Project Scenario**

The purpose of the project is to make a convenient library management system the both the library’s customers and staff can access to fulfil their needs from the library. For example, let’s say a staff member wants to add a new book to the library’s database or a customer wants to borrow a book. These two examples are the main functions of the database. To achieve this, we start with the entities we need, which are:

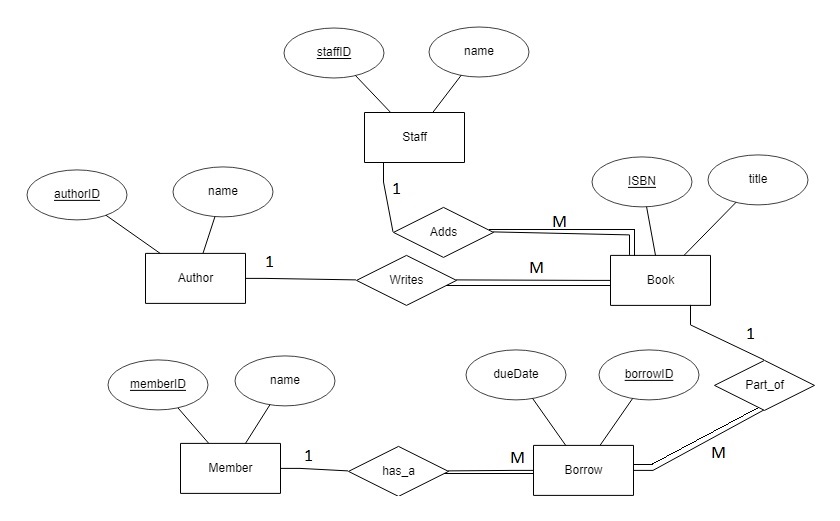
* Author: **authorID**, name
* Book: **ISBN**, title
* Staff: **staffID**, name
* Member: **memberID**, name
* Borrow: **borrowID**, dueDate

Note that underlined attributes in bold are primary keys.

Now that we know what entities we need, we will design an ER diagram to illustrate the different relationships between our used entities.

For instance, a Book is written by its Author and a Member borrows a book.

The ER diagram below shows these relationships and more:

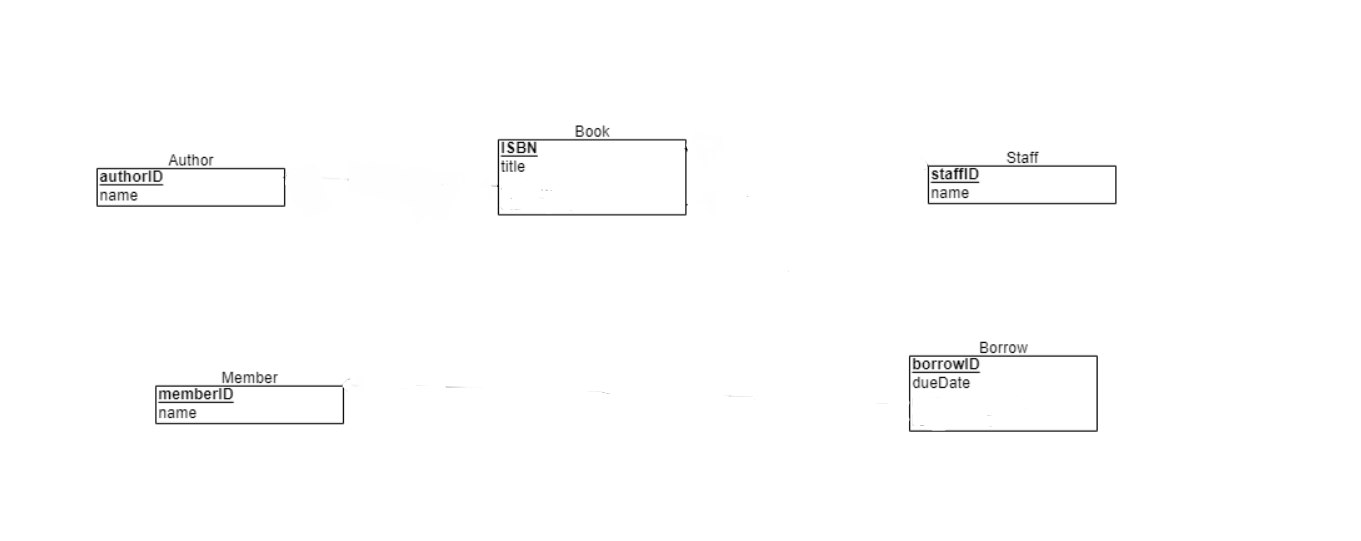


The relationships can be easily explained:

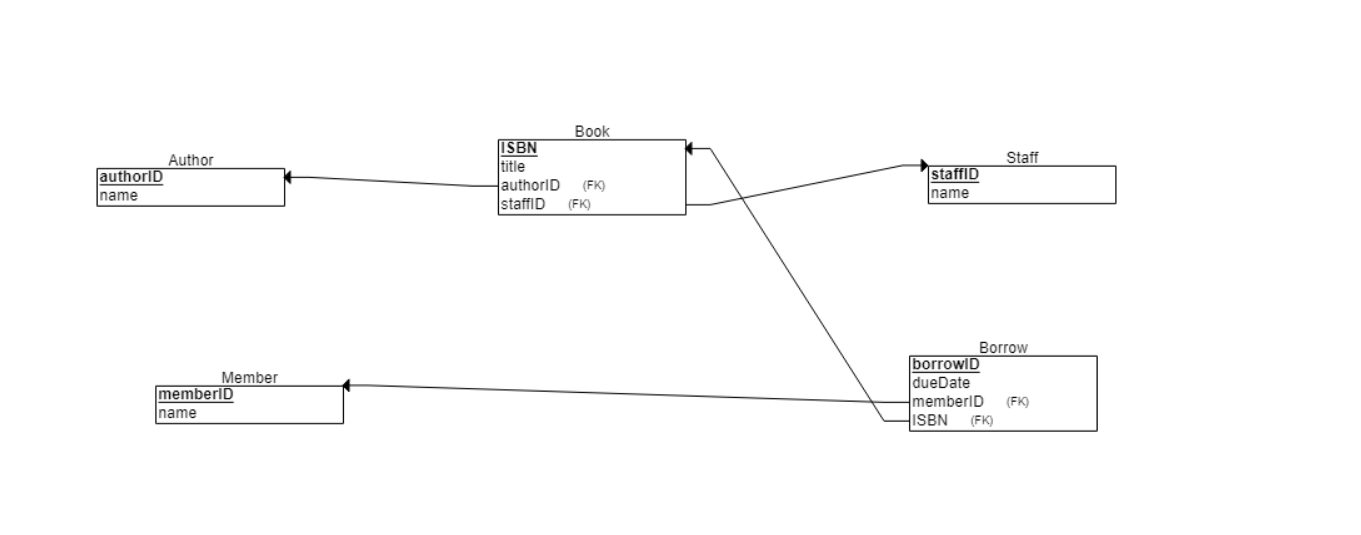
* Adds (one to many): a staff member can add many books. He doesn’t have to, though. However, is a book is added, it must be added by a staff member, not anyone else, which explains the total participation from one side.
* Writes (one to many): very similar to the above relationship. An author writes many books, and a book must be written by an author.
* Has\_a (one to many): A member (or customer) can have many borrows and of course, a borrow must include a member in it.
* Part\_of (one to many): a book can be part of many borrows, and a borrow must have a book in it.

After designing the ERD, we must create the relational schema

Before normalization, we add each entity with its attributes like this:



We then come back to the ERD diagram to see the relationships. In our case, all we have is one to many, so for all tables to be in boyce-codd, the primary key of the one side will be a foreign key in the many side like this:



Now that everything is ready, it is time to code each entity using SQL:

create database libManagement

use libManagement

create table Author(

authorID varchar(15),

authorName varchar(30),

primary key(authorID)

)

create table \_Member(

memberID varchar(15),

memberName varchar(30),

primary key(memberID)

)

create table Staff(

staffID varchar(15),

staffName varchar(30),

primary key(staffID)

)

create table Book(

ISBN varchar(15),

title varchar(30),

authorID varchar(15),

staffID varchar(15),

primary key(ISBN),

foreign key(authorID) references Author,

foreign key(staffID) references Staff

)

create table Borrow(

borrowID varchar(15),

duedate date,

memberID varchar(15) not null,

ISBN varchar(15) not null,

primary key(borrowID),

foreign key(memberID) references \_Member,

foreign key(ISBN) references Book

)

alter table Book alter column authorID varchar(15) not null

alter table Book alter column staffID varchar(15) not null

insert into Author(authorID,authorName) values('1','Shakespere')

insert into Author(authorID,authorName) values('2','JK Rowling')

insert into Author(authorID,authorName) values('3','Aghatha Christie')

insert into Author(authorID,authorName) values('4','Barbara Cartland')

insert into Author(authorID,authorName) values('5','Danielle Steel')

insert into Author(authorID,authorName) values('6','Harold Robbins')

insert into Author(authorID,authorName) values('7','Georges Simenon')

insert into Author(authorID,authorName) values('8','Enid Blyton')

insert into Author(authorID,authorName) values('9','Sidney Sheldon')

insert into Author(authorID,authorName) values('10','Eiichiro Oda')

insert into \_Member(memberID,memberName) values('ammar123','Ammar Amin')

insert into \_Member(memberID,memberName) values('mostafa73@','Mostafa Ahmed')

insert into \_Member(memberID,memberName) values('3','Omar Ahmed')

insert into \_Member(memberID,memberName) values('4','Youssef Sameh')

insert into \_Member(memberID,memberName) values('5','Fekry Nabil')

insert into \_Member(memberID,memberName) values('6','Amr Mohamed')

insert into \_Member(memberID,memberName) values('7','George Mostafa')

insert into \_Member(memberID,memberName) values('8','Reda Moussa')

insert into \_Member(memberID,memberName) values('9','Osama Ahmed')

insert into \_Member(memberID,memberName) values('10','Salah Kamel')

insert into Staff(staffID,staffName) values('123','Demi McIntyre')

insert into Staff(staffID,staffName) values('4892r','Eliseo Rojas')

insert into Staff(staffID,staffName) values('d43j4','Adaline Hendricks')

insert into Staff(staffID,staffName) values('rft5','Dash Leon')

insert into Staff(staffID,staffName) values('eof','Amora Barnett')

insert into Staff(staffID,staffName) values('rim8','Stephen Sofo')

insert into Staff(staffID,staffName) values('rmf4','Brynlee Wiley')

insert into Staff(staffID,staffName) values('jfu5','Mathew Christensen')

insert into Staff(staffID,staffName) values('fj8','Larmen Mata')

insert into Staff(staffID,staffName) values('fp0','Ray Faulkner')

insert into Book (ISBN,title,authorID,staffID) values('9r8f','Moby Dick','6','123')

insert into Book (ISBN,title,authorID,staffID) values('gtkm',' Anna Karenina','8','123')

insert into Book (ISBN,title,authorID,staffID) values('oogto',' Madame Bovary','6','123')

insert into Book (ISBN,title,authorID,staffID) values('pdfgr','War and Peace','6','rft5')

insert into Book (ISBN,title,authorID,staffID) values('4545','The Great Gatsby','6','123')

insert into Book (ISBN,title,authorID,staffID) values('4433','Lolita','6','123')

insert into Book (ISBN,title,authorID,staffID) values('2233','Middlemarch','4','eof')

insert into Book (ISBN,title,authorID,staffID) values('1122','Naruto','6','123')

insert into Book (ISBN,title,authorID,staffID) values('0405','Monster','6','123')

insert into Book (ISBN,title,authorID,staffID) values('3549','Berserk','6','123')

select \* from Book

Our database is now ready. All that remains is providing it with an easy to use interface for the end user, which will be done using python.

**Library Management System GUI**

This is a simple GUI application for a library management system, built with Python using the tkinter, customtkinter, and pypyodbc libraries.

**Features**

* Login system for staff and members.
* Staff can add books to the library.
* Members can borrow books from the library.

**Dependencies**

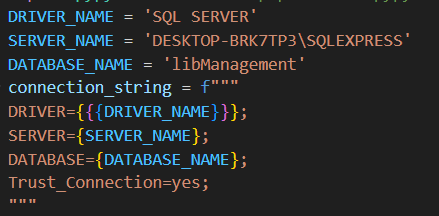
* tkinter
* customtkinter
* pypyodbc

You can install the dependencies with pip:

pip install tkinter customtkinter pypyodbc

**Database Setup**

The application connects to a SQL Server database. You need to set up the database and update the connection string in the script with your server name and database name.

DRIVER\_NAME = 'SQL SERVER'

SERVER\_NAME = 'your\_server\_name'

DATABASE\_NAME = 'your\_database\_name'

connection\_string = f"""

DRIVER={{{DRIVER\_NAME}}};

SERVER={SERVER\_NAME};

DATABASE={DATABASE\_NAME};

Trust\_Connection=yes;

"""

**To Connect:**

db = odbc.connect(connection\_string)

**Queries Needed in the GUI**

The script executes several SQL queries to fetch data from the Staff, Author, \_Member, Book, and Borrow tables in the database. The results of these queries are stored in lists.

A computer screen with text on it

Description automatically generated

**Login Screen**

**A screenshot of a computer login

Description automatically generated**

A screen shot of a computer program

Description automatically generatedThe login function is called when the "Login" button is clicked. It checks the entered username and password against the list of staff and members fetched from the database. If the credentials match a staff member, the staff\_window function is called. If they match a member, the member\_window function is called.

**A screen shot of a computer program

Description automatically generated**

**Staff Window**

The staff\_window function creates a new window with fields for a book title, ISBN, and author, and an "Add Book" button. When the "Add Book" button is clicked, the add\_book function is called, which inserts a new book into the Book table in the database. Test staff window with Login

Username: Dash Leon

Password: rft5

A screenshot of a book

Description automatically generated

A computer screen shot of text

Description automatically generated

**A computer screen with text

Description automatically generated**

**Member Window**

The member\_window function creates a new window with fields for a BorrowID, due date, and book title which shows available books to rented based on books and previous rented books, and a "Borrow Book" button. When the "Borrow Book" button is clicked, the borrow\_book function is called, which inserts a new record into the Borrow table in the database.

Test member window with Login

Username: Omar Ahmed

Password: 3

A screenshot of a book

Description automatically generated

A screen shot of a computer program

Description automatically generated

**Usage**

To run the application, simply execute the script with a Python interpreter:

python gui\_lb.py