CS6360.001 Database Design – Programming Project 1

Cliff Eddings

UT Dallas

Database Design – Programming Project 1

The following text lays out the design for a SQL programming project involving the creation of a database host application that interfaces with a backend SQL database implementing a Library Management System.

Database Schema

The database schema given in the project description lists the following entity types and the attributes for the included entities, keys are underlined:

- BOOK – Isbn, Title
- AUTHORS – Author_Id, Name
- BORROWER – Card_Id, SSN, Bname, Address, Phone, Email.
- FINES – Loan_Id, Fine_amt, Paid

The following relationships between the entity types are described as:

- BOOK_AUTHORS – Author_Id, Isbn
- BOOK_LOANS – Loan_Id, Isbn, Card_Id, Date_Out, Due_Date, Date_In

The data for the schema was given in two csv files, books and borrowers. The book data ISBN10 was used for ISBN and implemented into the database as a varchar (10) attribute. The data columns ISBN10 and Title were used to implement the BOOKS table. The author column was used to implement the AUTHORS table, but first all duplicates were removed then an auto incrementing column Author_ID was added using the int (8) data type. The two tables' relationship is defined by the BOOK_AUTHORS table, using the relationship between Author_Id and ISBN.

In the BORROWER table, from the data columns first_name and last_name was joined to create the Bname attribute. Borrower_Id is used as the attribute Card_Id and auto increments when another borrower is added. The data column ssn is used for SSN, implemented as a varchar (11) data type, and has a unique index to prevent duplicate borrowers.

The schema is defined in the EER model on page 5 of this document.

Functional Requirements

1. Graphical User Interface and Overall Design.

   The GUI interface is implemented using Microsoft Visual Studio and programmed with C#. The SQL database is MySql and the program uses the MySqlClient Connector.

2. Book Search and Availability

The form "Main Menu" handles the book search. The substring matching is handled in the search query passed to MySql. The search checks for matching substrings in the fields ISBN, Title, and Author.

Availability is displayed using a view named "book_available", which left joins the views "book_view", containing ISBN, Title, and Author, with view "aval", containing the Available field.

3. Book Loans

The form "Main Menu" handles checking out books. After clicking the "Check Out Book" button, a hidden group box is revealed and prompts the user to enter the card id of the borrower checking out the book. The due date field is handled at the database level, the column automatically updates the field using the date_add function, date out, and a 14-day interval. The method "Check_Borrower" returns a true value if the borrower has less than three books checked out and false if 3, the user is then not allowed to check out the book if a false value is returned. The method "Check_Book" checks if the book is available or not returning false if it is not and true if it is. In both cases an error message is returned describing the error.

Checking in books is implemented on the "Loans" form. The user can search for a loan using the book ISBN, borrower name, or borrower card id. Once located the book loan is selected then the "Check In" button is clicked. A hidden group box appears showing the ISBN, card id, borrower name, and the date to be checked in (the current date). The user is asked to verify the information then the second "Check In" button is clicked.

4. Borrower Management

The form Borrower handles the creation of new borrowers. The fields Name, SSN, and Address are required in the programming to be filled before entering. Card ids are auto incremented in the SQL database. The SSN is indexed as unique in the database, but the method "Check_SSN" returns true if there is no matching SSN and false if there is. If a false is returned a message displays claiming the SSN is already in use.

5. Fines

Fines are calculated in the database using the field "Days_over" multiplied by 0.25 and stored in the field "Fine_amt." Updating or refreshing of the fines are handled by clicking the “Update Fines” button on the “Fines” form. When this button is clicked, two queries are running, one to update fines for overdue books not yet turned in and one to update turned in books. Duplicate entries are prevented in the fines table using the “on duplicate key update” command in the queries.

Fines are first displayed by card id and the total fine amount displayed. When a card id or borrower is selected and the “Pay Fine” button clicked, all the fines for the borrower are displayed. The user is prompted to select one to be payed and after clicking “Execute” the fine is updated as paid. The “Show All Fines” button shows all fines paid and unpaid, since the default of the program only shows unpaid fines.

EER Diagram for Library Database

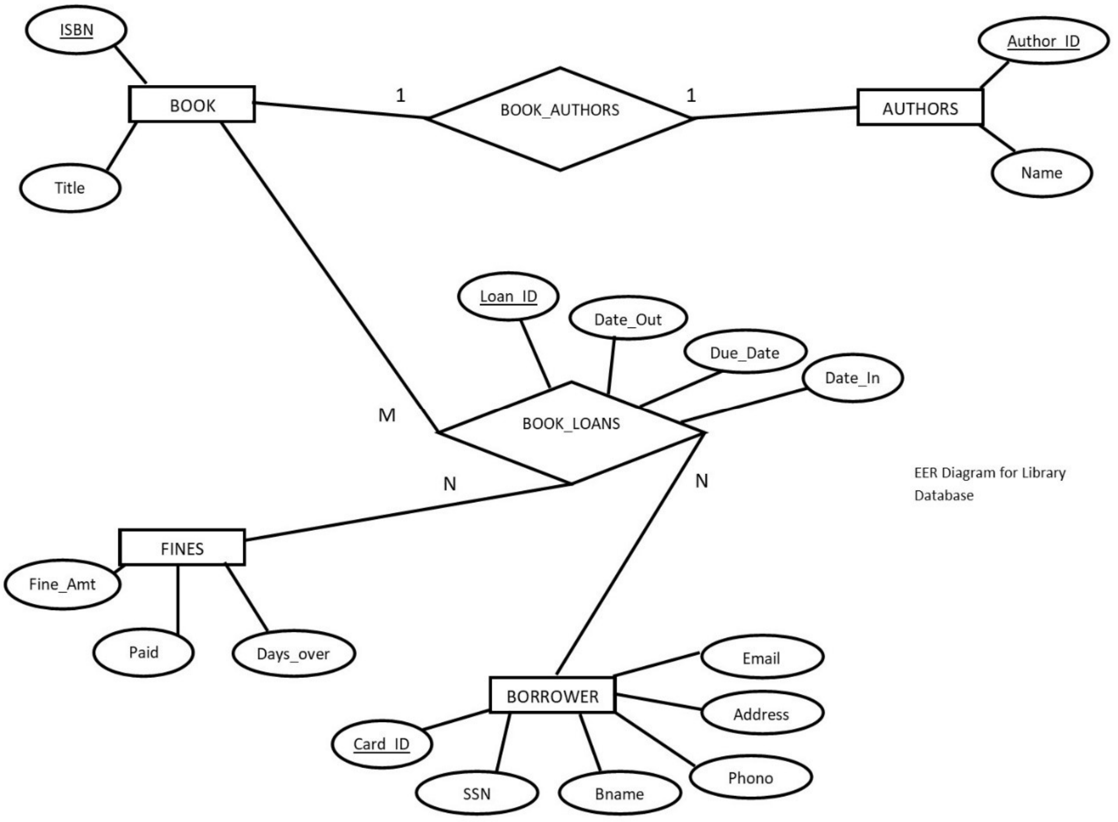

Figure 1. EER Model for the Library database showing relationships between entity sets and their attributes.