**Final Exam Project:** This is part of your final exam. As such*, failure to complete this project will result in an “F” for this class, regardless of your earned score for the course.* (100 points possible)

You are maintaining a small library for you and your friends to share so everyone can add books, and check in and out books for their own use. This small library has grown lately, and you would like to create a small database to manage it better. After careful consideration, you think that the database should store the following data (which is not presented in any particular order or structure). You may have additional data not specifically listed below.

**Book author(s) – there may be more than one per book  
Book title  
ISBN – a book may have either a 10 or 13 digit ISBN (or both)  
Publisher  
Year when the book was published  
Number of pages  
Friend that donated the book to the library  
Original price paid for the book (by the friend)  
Where the book was bought  
When the book was bought  
Book category (fiction, non-fiction, reference)  
Book genre (horror, romance, adventure, etc.)**

**Book's subject matter (what the book is about; dogs, technology, cars, etc.)  
Friend that borrowed the book  
When the book was checked out  
When the book was checked in  
For how long a book was borrowed by one of your friends**

Complete the following steps:

1. Build a list of business needs, rules and assumptions based upon the scenario and information provided.

• A title may have one, or two ISBNs, an ISBN may only have one title  
• An ISBN may be 10 or 13 digits  
• A title may have one or many authors  
• A title usually has one publisher, a publisher can have one or many books  
• An author may have one or many books (authors with zero books are not relevant to  
this library)  
• A title may have one or many pages, pages may only have one title  
• A title can only belong to one category, a category can have many books  
• A book can only be donated by one friend  
• A title can only have been purchased for one price  
• A title can have one or many subject matters, and a subject matter zero or many books  
• A title/ISBN can only have one year, but a year many titles  
• A friend may have zero or many books, a book can only be checked out by one friend at  
a time  
• A book can only be borrowed (checked out, in, length of time) once at a time

1. Create a relational schema (introduced in Chapter 4) for each entity following the rules of normalization. Show your progression through each normal form from 1st through 3rd, including an explanation of how you have met each normal form.



Diagram

Description automatically generated

A picture containing application

Description automatically generated

1. Create an ER Diagram. Define the relationships between the entities, including optionality and cardinality. Use Crow's Foot notation. Refer to Figure 4.35 in Chapter 4 for an example of what your diagram should look like.

Diagram

Description automatically generated

1. For organizational purposes, create a data dictionary table (refer to Table 3.6 in Chapter 3).
2. Determine at least 5 indexes you will create and provide a justification.
3. Using the data dictionary you created in step 4, write the DDL statements that you will use to create the tables and indexes. Don't forget about the constraints.
4. Build the tables in the class RDBMS. Remember, the order you create the tables is important.
5. Using DML commands, insert data into your tables. Aim for at least 5 or more records per table.
6. Create two views of your choice. Each one should incorporate data pulled from at least three different tables. Provide an explanation of their purpose.
7. Write a detailed explanation of the process you followed to build this database At a minimum, your explanation should fill a full page, single-spaced, 12 point font. Demonstrate your understanding of database terminology and how you applied it. For example, what is a relational database, ERDs, atomicity, anomalies, entity integrity, referential integrity. This is not an all-inclusive list, but some ideas to help get you started.
8. “Forward engineer” your database to create a fully-functioning schema export. It needs to include SQL statements to create the schema; create the tables, columns, indexes/constraints; insert data; and build the views. To test your design and build, this file will be used to build a copy of your database in the instructor’s domain for testing/grading.