**Assignment 5**

Attached you'll find a database of books for a library. The data is in a single sheet. You are to create a normalized database to practical business standard levels.

**1. Provide what level you are normalizing to.**

The database LibraryBooks is normalised to the Third Normal Form (3NF). 3NF ensures that there are no repeating groups, no attribute is dependent on only part of the primary key, and that all transient dependencies are eliminated. 3NF is suitable for most business applications.

**2. Provide the text description of the business rules for how the entities you create relate**

* BookHeader Table
  + Each BookID is system-generated, unique, and associated with one Title and one AuthorName
  + Each Title and AuthorName is dependent on one BookID, and only the BookID
* BookDetail Table
  + Each BookID in the BookHeader table has a One-to-Many relationship with the BookDetail Table using the BookID as the foreign key
  + Each BookID can have multiple PublisherIDs (Publishers at a specific address), PublishDates, MSRPs, ISBN10, ISBN13, and Lengths
  + A PublisherID is a system-generated, unique ID to represent each Publisher at a specific address (ex. PublisherName “At Lacus Foundation” at Street “6369 Vulputate Street”)
  + The PublishDate, MSRP, ISBN10, ISBN13, and Length are all dependent on both the BookID and PublisherID together
  + In this data sample, ISBN13 is not unique so it could not be used as a key. In the real world, ISBN13 is unique for each publishing of a book and ISBN10 can be converted to ISBN13.
* PostalZipHeader Table
  + Each PostalZip has a one-to-many relationship with the PublisherHeader Table via the PostalZip foreign key. Each City and Region is related to a unique PostalZip. Each PostalZip can be associated with multiple PublisherIDs
* PublisherHeader Table
  + Each PublisherID in the BookDetail Table has a one-to-one relationship with the PublisherID in the PublisherHeader table using the PostalZip as the Foreign Key
  + Each PublisherID is associated with one Publisher, one PostalZip, and one Street
* CategoryDetail Table
  + Each BookID shares a one-to-many relationship with the CategoryDetail table
  + Each BookID can have many Categories and together, BookID and Category make a unique entity
  + The BookID is the Foreign Key to BookHeader(BookID)
* InventoryDetail Table
  + Each BookID has a one-to-many relationship with the InventoryDetail table using the BookID as the Foreign Key
  + There is a one-to-one relationship between the BookDetail and InventoryDetail using the BookID and PublisherID as the Foreign Keys.
  + Both the InventoryDetail and BookDetail tables share the same Primary Keys, BookID and PublisherID, so this could be merged into the same table. However, this data is not related to each other. CopiesAvailable, CopiesOnLoan, and DeweyDecimal relate specifically to the in-house library inventory system, whereas PublishDate, MRSP, ISBN10, ISBN13, and Length relate to publicly-available information. Therefore, it is the best course of action to separate these two tables in case the library chooses to expand their database to include customer information and historical book check-outs
  + The CopiesAvailable, CopiesOnLoan, and DeweyDecimal are dependent on both the BookID and the PublisherID

**3. Provide an ERD for the database showing the relationships. The type of ERD doesn't matter, as long as I can clearly see the Primary Keys, Foreign Keys, and the Many side of the relationships. If you use SQL Servers diagramming tool then I suggest using a print screen and pasting that into a document.**

