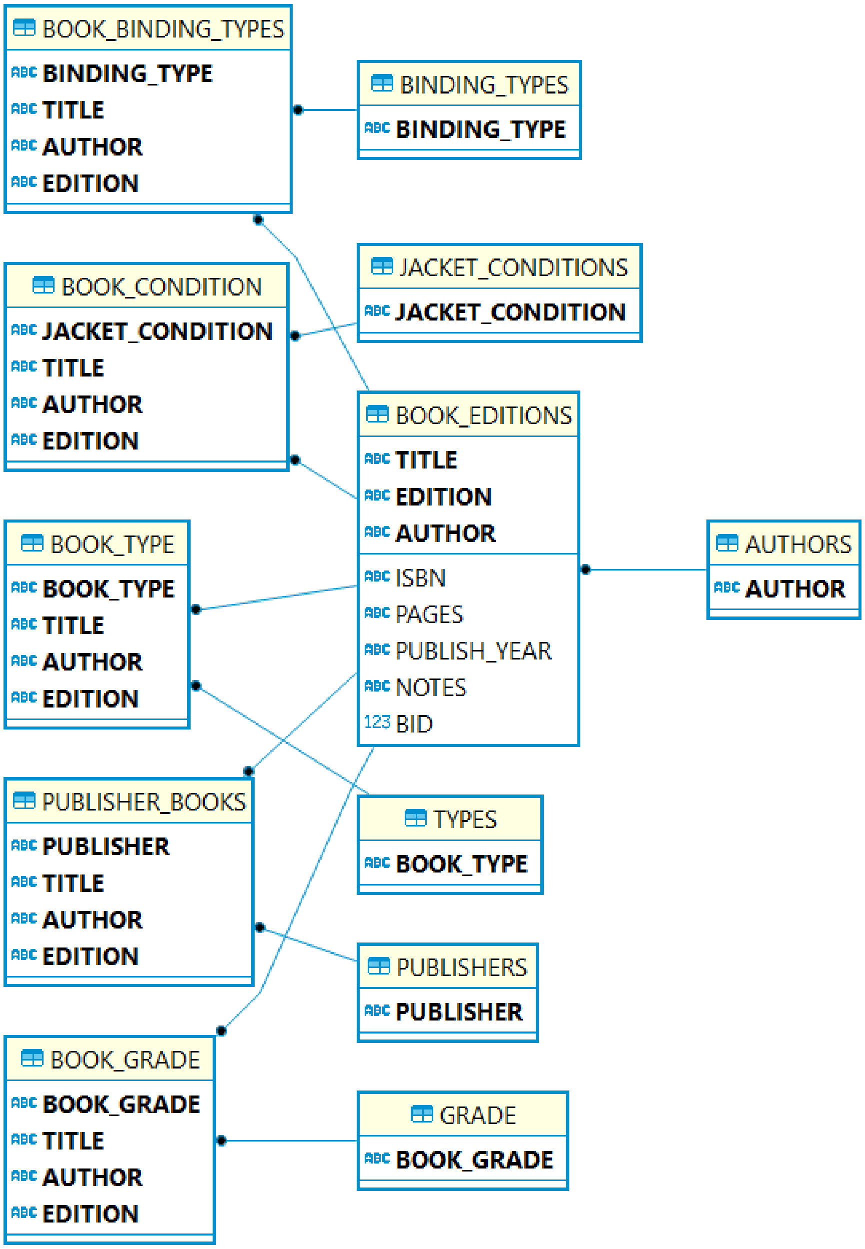
Documentation:

1. **Placing the data in BCNF form:** Initially we figured that the data coming in for any state in the database may be completely ambiguous based on the data that we received in one table where everything would be tied to an auto incremented primary key. After some examination of the data that was received by the antiquarian, we decided to find the column dependencies of one or more attributes on another which the auto incremented attribute caused many problems because every single column lowest cardinality attribute candidate key was always the auto incremented primary key so we needed to look more in depth at the semantics of the data to find a more reliable and less ambiguous primary key. We found columns in which data is repeated and tried to limit the available options via reference tables that, and this is the important part, exist regardless of the individual books being in the database. This is what allows us to assume 3NF. For example, a book(s) condition should be limited to a set of particular values because the inverse of this would mean that the antiquarian can make up a bunch of different values for any particular book that describes the same or similar condition of another. Therefore, the key identifying attributes of a book do not imply a condition via the transitive property because the values exists without the reference to the table. There are attributes we found, however, that cannot have this transitive work around such as notes. If a book has a note it is possible that the lookup table references has the same cardinality as the reference table. In which case, not only does the transitive property hold, but it is counter intuitive to it being the solution to data redundancy. Finding these simple relations helps reduce data ambiguity, relational ambiguity, and minimize constraints on the input. One of the most important things we have had to learn throughout transitioning this database into BCNF is that the normalization of a database is a function of the semantics of the data for any state not just the state the data is in now. Meaning it is important to make assumptions about the data being input into the database. Otherwise; in order to make it more efficient, the structure of the database will change based on the state of the database which is much more maintenance than constraining the data based on assumptions.
2. **Data Entry:**

ISSUES	RESOLUTION
Missing Data	Filled in unknown values with default values
Inconsistent data	Edited as much data to make it as consistent as possible, miss spelled words were not fixed
Special characters causing issues	Removed most special characters
Duplicates	Removed duplicates
Same data labeled differently	Left it in the DB, too much work to remove and not enough knowledge to distinguish between different author names, publishers, titles, etc. So we left it in the DB.

3. **Creating the tables:** The tables were created initially using a normalized version of our original tables. This version utilized an auto incrementing value as the primary key for every entry. When we found that this key was no longer suitable for our design in BCNF, alterations had to be made to every table where we deleted the auto incrementing primary key and replaced each instance by the new primary key {Title, Author, Edition}. In this process, we had to edit the foreign key constraints to accurately reflect the recent changes. These tasks were tedious, but necessary for the most optimal design of the database.


BOOK_BINDING_TYPES
BINDING_TYPE
TITLE
AUTHOR
EDITION
BINDING_TYPES
BINDING_TYPE
BOOK_CONDITION
JACKET_CONDITION
TITLE
AUTHOR
EDITION
JACKET_CONDITIONS
JACKET_CONDITION
BOOK_EDITIONS
TITLE
EDITION
AUTHOR
ISBN
PAGES
PUBLISH_YEAR
NOTES
123 BID
AUTHORS
AUTHOR
BOOK_TYPE
BOOK_TYPE
TITLE
AUTHOR
EDITION
TYPES
BOOK_TYPE
PUBLISHER_BOOKS
PUBLISHER
TITLE
AUTHOR
EDITION
PUBLISHERS
PUBLISHER
BOOK_GRADE
BOOK_GRADE
TITLE
AUTHOR
EDITION
GRADE
BOOK_GRADE