An additional concern surrounding books was how to handle categories, subcategories and keywords. These are many-to-many relationships (books can have multiple values for each attribute, and these can be shared by different books). For each of these values, a table was created containing the corresponding book ID, and the associated value. This allows for multiple values for each. There would be some redundancy in the table itself as multiple books share values, but is a simple and efficient way of modeling the information. This doesn't include repeat information per book, which is a greater consideration.

In regards to suggestions and criticisms on Phase 2, many of the recommendations felt inappropriate or incorrect. The first note was that a professor could work at two different universities and in two different departments. This seems extremely impractical and unlikely. The chances that a professor works at two different universities alone seems exceedingly rare. Moreover, given the academic specialization that professors often have, it seems equally odd that they would work in different departments as well. Given this judgment, I have chosen to disregard that note. In the rare case that this happened, the minimal redundancy incurred by creating a second, mostly redundant record would be negligible compared to the overhead involved in designing a system to handle such an extreme edge case. In any manner, there is a likelihood that professors at different universities might have the same name. The school and department that they teach at serve as discriminating factors.

It is completely possible that different books are used for different courses. Also, it is possible that different books are used for the same course (different professors could favor a different text). This criticism has been acknowledged and incorporated into the relational schema. The current schema acknowledges that the same course can have different book depending on which course section it is, who is teaching, what year, etc.

The role of an employee (admin, superadmin, customer service representative) is captured by the role attribute in the employee entity. As stated above, there is no additional information that needs to be stored for an employee based on their role, and this attribute would be sufficient for assigning privileges.

The condition and type of the book (e.g. used, hardcover) are handled under the book entity as well; the *used* and *format* attributes correspond to these values. They may need better names, but that was the intent. Additionally, the inventory count of a book is tracked under the *quantity* attribute.

In regards to missing relationships and the lack of a weak entity set, I've covered the only missing relationship I can, which is between a book and a course. I do not believe that a weak entity set is necessary to model the information accurately and efficiently.