password is related to created

* I think this is probably more info overall than you'll want to track for your project here. If you want to, then by all means, go for it: but it's definitely a more complex schema than I'd typically see and we'll be writing code around it later in the term, so you may want to prune a bit. The areas that seem most likely to me are reducing complexity of the book schema (not sure you need to track chapters, paragraphs, or sentences, and if you do I can suggest some practices for doing so that don't require building them into the db schema itself) and the contributor section (author/narrator could probably be merged into contributor and the role a contributor has on a given project could be defined in a lookup table).
* Speaking of lookup tables, there aren't many in here. It looks like you've defined some relationships as one to one that I'd suggest are one to many, like book->ebook. In other places, like author->publisher, you have many to many line endings but no lookup table in between.
* In some of your one to many relationships you have fields defining foreign IDs at both ends of the relationships (e.g. book.language\_id and language.book\_id). For a relationship to be one to many you would need to only have one end of the relationship defined or the other (e.g., if many books can be in a language, then you would have a book.language\_id field but not a language.book\_id field; if you then want to say a book can be in multiple languages, you would use a lookup table between them to make the relationship fully many to many)
* The way you've included addresses is interesting here: It is a good normalization procedure to identify that they represent redundant information across the author, narrator, and publisher tables. Often a database designer might keep that info all inside their respective parents rather than creating a separate address table, just for the sake of simplicity, but pulling it out is correct. If you're going to do that, though, your author, narrator, and publisher tables should all have an address\_id but your address table doesn't need author\_id, publisher\_id, or narrator\_id fields. The reason for that mostly comes down to what you're going to put into those fields when they aren't relevant, e.g. what goes in the author\_id field when the current address is for a publisher.

I'd suggest trimming some of the extra tables, checking on the way relationships are structured, and going forward with a simplified version of the schema.



[**Mike**](https://app.slack.com/team/U019YEK61T6)  [3:20 PM](https://dig540-2020.slack.com/archives/D019SF4EATD/p1601839243003600)

[@John Bell](https://dig540-2020.slack.com/team/U019P0F58E9) Thank you for the feedback, I realized after last week what the object of the exercise was -- to establish the basis of the semester long project.  Totally agree with your comments regarding the complexity I established and no I would not want to keep it for the semester project.  I will redo and submit by Wednesday.  On a separate issue, when I have gone to build and save a table -- I am getting an error message regarding the primary key not being the appropriate type -- though I am using integer as you demonstrated in you demo.