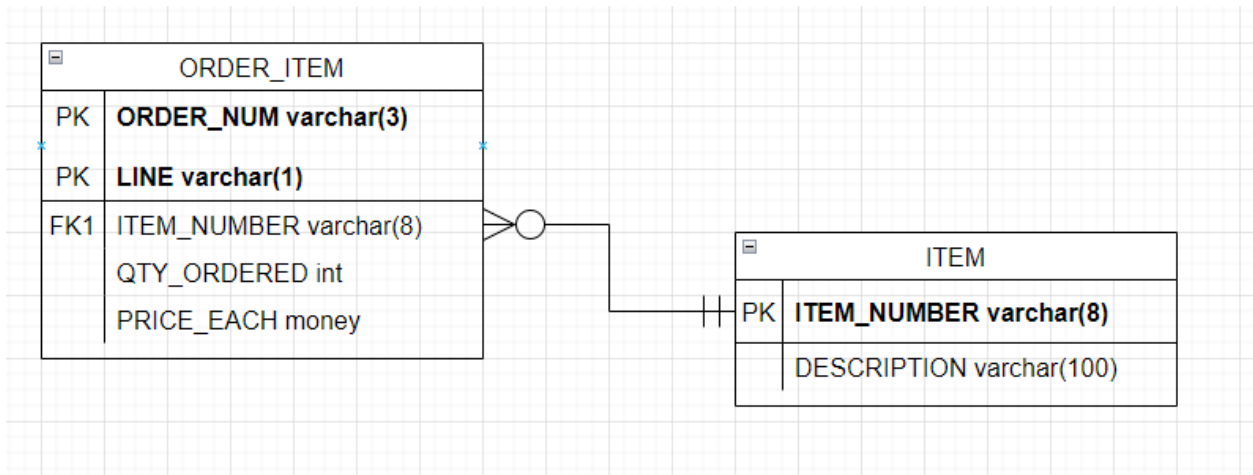


### Case Study 2:



### Case Study 3:

PublisherID	PublisherName	Click to Add
1	Pearson	
BookID	BookTitle	Authors
2	Modern Database Management	Hoffer, Kamesh, Topi
3	Data Structures and Algorithm Analysis in C++	Weiss
*	(New)	0
2	Addison Wesley	
BookID	BookTitle	Authors
1	Languages and Machines	Sudkamp
*	(New)	0
*	(New)	

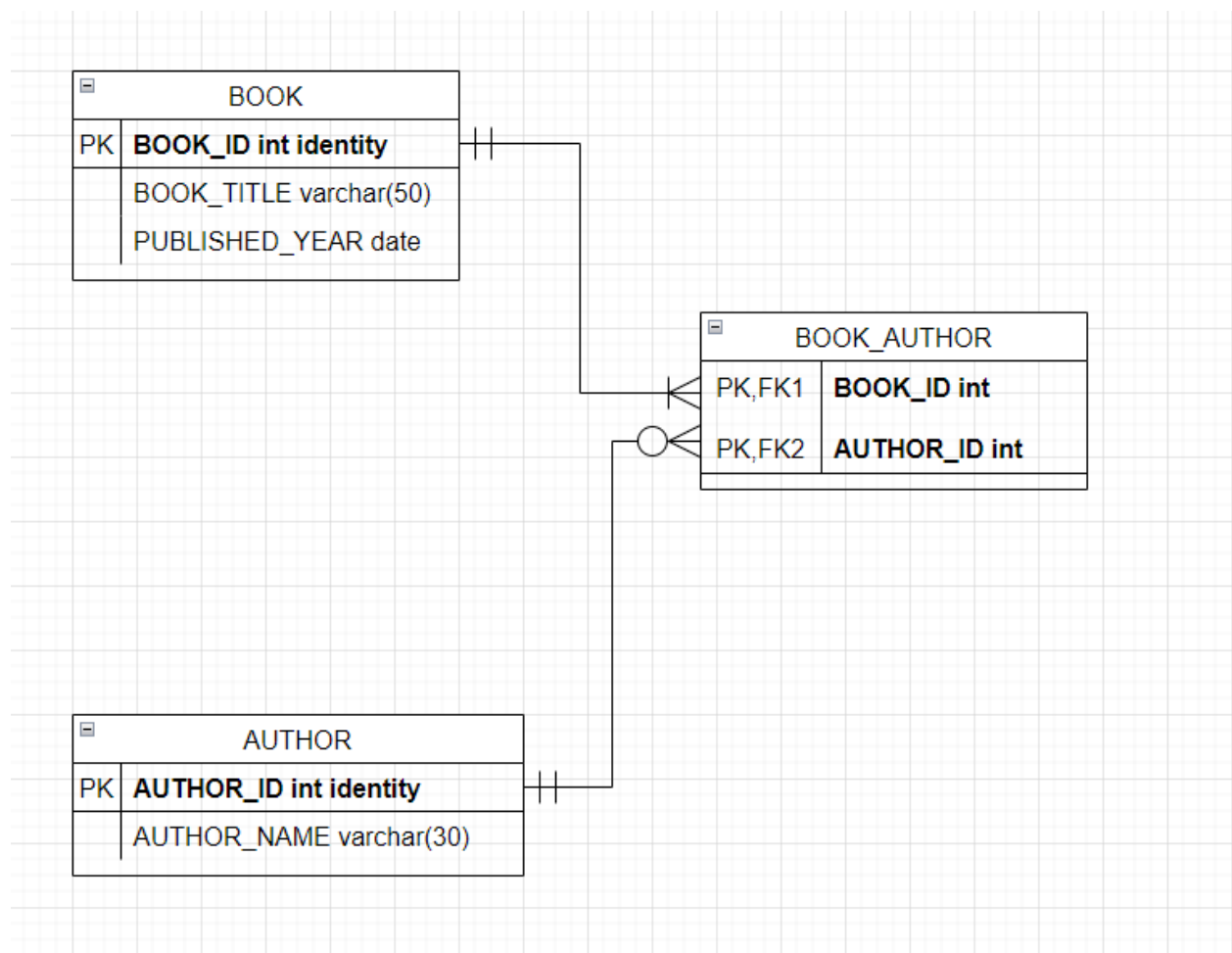
Looking at just the Book table, we have a multi-value field for the Authors of Modern Database Management. This means that we are technically only in 0NF. We need to make a separate row for each Book-Author combination. This unfortunately will give us an invalid primary key, so we instead should make this table a BookAuthor table with a composite primary key of BookID and AuthorID (we will need to add a field for AuthorID). We will also need an Author table that has Author ID as its PK and Author Name as its other attribute. This would in effect replace Authors in the Book table with AuthorID. Due to the Author table having no multi-value attributes, only one primary key,

and only one non-primary attribute, it will be in 3NF.

The BookAuthor table however is still only in 1NF, as both BookTitle and Published Year are partial functional dependencies. This means that we will need a Book table (different from our original table that has now been changed and renamed BookAuthor) that includes BookID, BookTitle, and PublishedYear. This would remove BookTitle and PublishedYear from the BookAuthor table.

What will be left in the BookAuthor table are only the two primary keys, meaning that we are in 3NF by default.

Lastly is our Book table. There are no multi-value fields, so that would get us to 1NF. After that, we would be in 2NF by default, since the Book table only has one primary key. Given that there are not any transitive functional dependencies between BookTitle and PublishedYear, this would also get us to 3NF. The final tables are shown below.



## Part 2:

