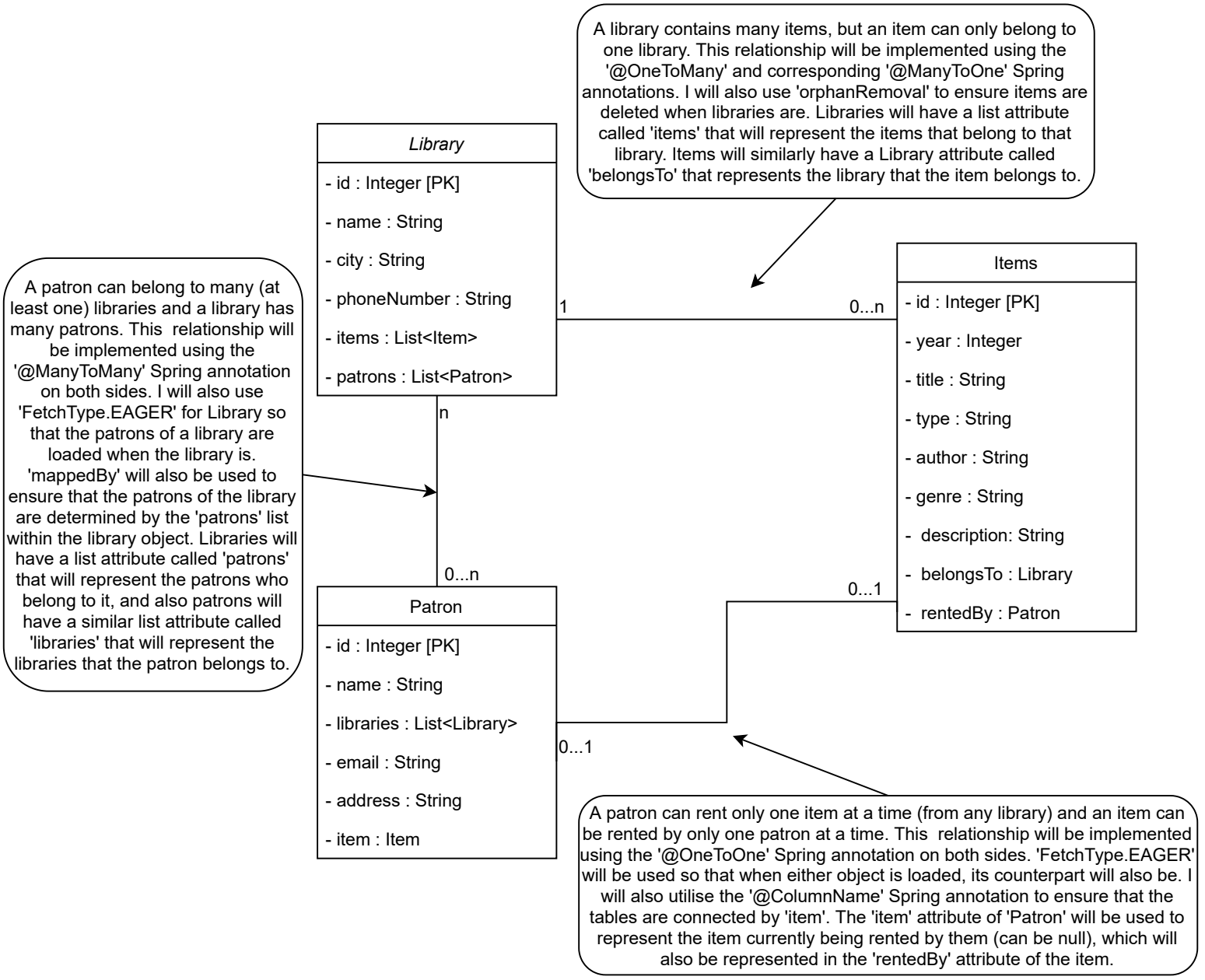
A library contains many items, but an item can only belong to one library. This relationship will be implemented using the '@OneToMany' and corresponding '@ManyToOne' Spring annotations. I will also use 'orphanRemoval' to ensure items are deleted when libraries are. Libraries will have a list attribute called 'items' that will represent the items that belong to that library. Items will similarly have a Library attribute called 'belongsTo' that represents the library that the item belongs to.
Library
- id : Integer [PK]
- name : String
- city : String
- phoneNumber : String
- items : List<Item>
- patrons : List<Patron>
1
0...n
Items
- id : Integer [PK]
- year : Integer
- title : String
- type : String
- author : String
- genre : String
- description: String
- belongsTo : Library
- rentedBy : Patron
A patron can belong to many (at least one) libraries and a library has many patrons. This relationship will be implemented using the '@ManyToMany' Spring annotation on both sides. I will also use 'FetchType.EAGER' for Library so that the patrons of a library are loaded when the library is. 'mappedBy' will also be used to ensure that the patrons of the library are determined by the 'patrons' list within the library object. Libraries will have a list attribute called 'patrons' that will represent the patrons who belong to it, and also patrons will have a similar list attribute called 'libraries' that will represent the libraries that the patron belongs to.
n
0...n
Patron
- id : Integer [PK]
- name : String
- libraries : List<Library>
- email : String
- address : String
- item : Item
0...1
0...1
A patron can rent only one item at a time (from any library) and an item can be rented by only one patron at a time. This relationship will be implemented using the '@OneToOne' Spring annotation on both sides. 'FetchType.EAGER' will be used so that when either object is loaded, its counterpart will also be. I will also utilise the '@ColumnName' Spring annotation to ensure that the tables are connected by 'item'. The 'item' attribute of 'Patron' will be used to represent the item currently being rented by them (can be null), which will also be represented in the 'rentedBy' attribute of the item.