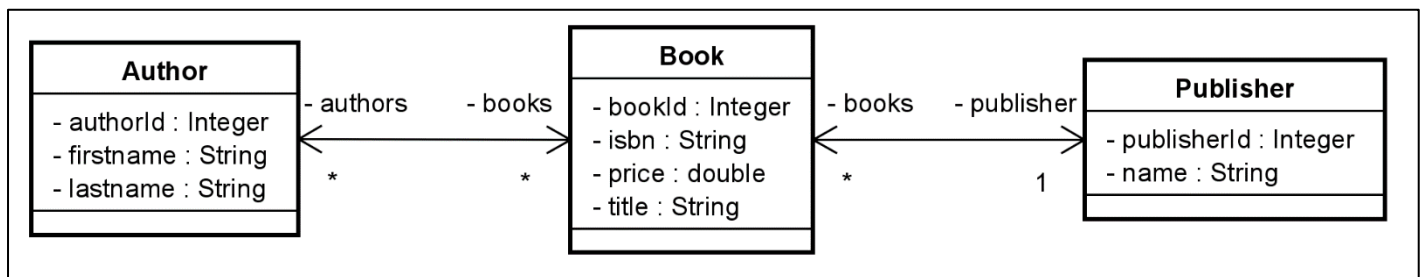
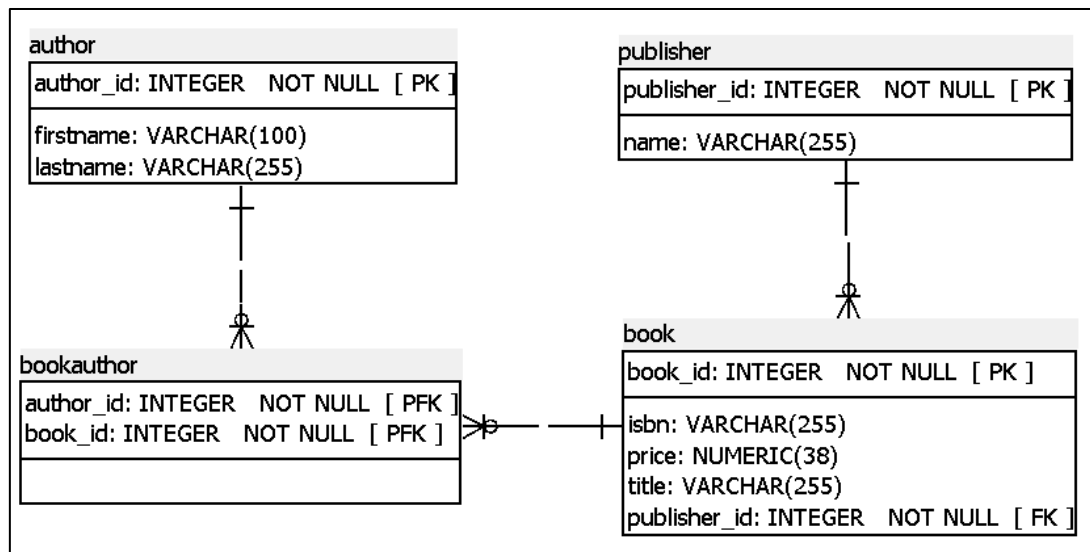


Books-DB

Erstelle eine Spring-Applikation, auf Basis JPA, zur Verwaltung von Büchern. Für die postgres-Datenbank 'booksdb' sind alle Tabellen ausschließlich über das JPA-Mapping zu erzeugen.



Erstelle das Mapping für die Entity-Klassen auf Basis des ER- und Klassendiagramms. **Alle Namen und Datentypen sind entsprechend dem ERD und KD einzuhalten!**

Erzeuge alle notwendigen Methoden und Konstruktoren der Pojo-Klassen mit den Lombok-Annotationen.

Um die Daten aus der JSON-Datei `booksdb.json` einzulesen sind alle notwendigen JSON-Annotationen in die Pojo-Klassen einzufügen.

Die Klasse `pojos.Publisher.java`:

- Der Primärschlüsselwert wird in der Datenbank vergeben
- JPA-Annotationen
- Lombok-Annotationen
- Json-Annotationen

Die Klasse `pojos.Book.java`:

- Der Primärschlüsselwert wird in der Datenbank vergeben
- Join-Table Definition
- JPA-Annotationen
- Lombok-Annotationen
- JSON-Annotationen




Die Klasse `pojos.Author.java`:

Die Assoziationen zu den Klassen `Publisher` und `Author` sind bidirektional.

- Der Primärschlüsselwert wird in der Datenbank vergeben
- JPA-Annotationen
- Lombok-Annotationen

Die Interfaces `database.PublisherRepository.java`
`database.BookRepository.java`
`database.AuthorRepository.java`






Erstelle folgende JPQL-Queries:

-  `getAllAuthorsOfPublisher(String publisherName)`
Finde eine Liste aller Autoren, die Bücher für einen Publisher (bestimmt durch einen Parameter) geschrieben haben. Die Autoren werden aufsteigend nach dem Nachnamen sortiert.
-  `geAuthorsWithAtLeastNumberBooks(Integer number)`
Finde eine Liste aller Autoren die mehr als eine, durch einen Parameter gegebene Anzahl an Büchern geschrieben haben (z.B die mehr als 10 Bücher geschrieben haben)
-  `getBooksWithPriceOverMaxPriceOfPublisher(String publisherName)`
Finde eine Liste aller Bücher deren Preis höher ist als der maximale Preis aller Bücher eines Verlags, gegeben durch den Namen des Verlags.

Die Klasse `database.InitDatabase.java`

Spring-Komponente

`initDatabaseFromJson()`: Methode wird bei Programstart automatisch aufgerufen

-  Der Inhalt der Datei `booksdb.json` wird eingelesen
-  Alle bidirektionalen Assoziationen zwischen den Pojo-Objekten werden hergestellt
-  Es ist darauf zu achten, dass von keinem Autor doppelte Objekt existieren.
-  Alle Pojo-Objekte werden in der Datenbank persistiert
-  Ausgabe der Anzahl der persisierten Objekte auf der Spring Konsole:
`INFO a.k.booksdb.database.InitDatabase : publisher saved: 193`
`INFO a.k.booksdb.database.InitDatabase : books saved: 1056`
`INFO a.k.booksdb.database.InitDatabase : authors saved: 1119`

`runQueries()`: Aufruf der drei Queries und Ausgabe der Werte auf der Konsole. Verwende für den `publisherName=Harper` und für `number=10`