Create a simple Java project **“Corporate Books Reading Counter”**

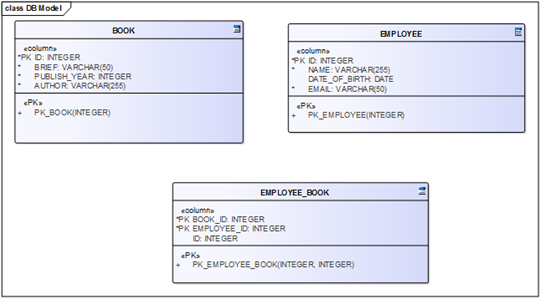
The root package is **com.epam.library**. Implement the following layers:

* Domain
* DAO
* Service
* Controller

Use the **windows console** for the application output. Use Java SE **7 or 8** API

To run the application, please, use the Main class or JUnit 4 test *(if you implement both the options, i.e. Console + JUnit, you will get additional points)*

All the data should be stored in the database (use **MySQL**)



1. Fill the Book table with the default data about books (50 books at least). Manual task.

2. Fill the Employee table with the default data (12 employees at least). Manual task.

3. Fill the intermediate table by 100-300 randomly generated values (user\_id and book\_id). There should be at least one user with no Book relations. There should be at least one user with 3+ Book relations, i.e. more than 3 books read. **Java coding task**.

4. Create/Generate 2 separate db scripts:

A. DB scheme generation script

B. default data insert script

The **Domain** layer should contain the following beans:

- Book (*title, author, brief, date of publishing*),

- Employee (*name, e-mail, date of birth*)

- Book and Employee has a ‘many-to-many’ relationship

The **DAO** layer should implement all the C.R.U.D. operations for the Book entity.

Use Java Database Connectivity (JDBC) API (DO NOT use Spring JDBC or Spring Data). In case you have issues with the JDBC implementation, you can use the Hibernate Framework with HQL. The JDBC implementation is more preferable!

The **Service** layershould implement the following operations:

1. **View report on the employees who have read more than 1 book**

The report should contain the following information:

**employee name: number of books**

The data in the report should be sorted out by the number of books upwards. The report SHOULD NOT contain the names of the employees who have read less than one book.

2. **View report on the employees who have read less than or equal to 2 books**

The report should contain the following information:

**employee name, employee date of birth: number of books**

The data in the report should be sorted out by the employee date of birth and the number of books (from the oldest to the youngest employees, and from the largest to the smallest number of books read, i.e. downward). The report SHOULD NOT contain the names of the employees who have read less than two books.

3. **Obligatory task:** **‘Rename Book’ operation**.

To rename a book, the user should enter the old book title and the new book title; the application finds the correct book by the old title and renames it with the new title.

**Additional task**:

In case the user enters the mask instead of the whole book title, for example ‘Spring in a\*’, the application should find all the books matching the mask and should rename all these books with the new title.

**Additional requirements:**

(if you follow the additional requirements, you will get extra points)

- try to use JUnit 4 framework and try to test DAO and/or Service layer

- add a logger and log some useful information about the work of your application ; and log the information about exceptions handling