1 Practice 14.1. Databases with Doctrine

> This practice is the next of the series of practices, consisting of a Library application. Each practice will be based on the previous one. Before starting it, finish the previous one, make the corrections you consider, if needed, and start coding.
>
> Before start, create a new git branch named `library-u12` and switch to it:

```
git branch library-u12
git checkout library-u12
```

> Do all the practice in the `library-u12` branch. Once finished, do the submission via GitHub Classroom as detailed at the end of the practice.

1.1 Overview

In this practice we're going to do the models and the databases access classes for our library app using Doctrine and the tools provided by Symfony. Some of the database operations, such inserts and updates will be coded "manually" and in the next unit we will learn how to do these operations with an API.
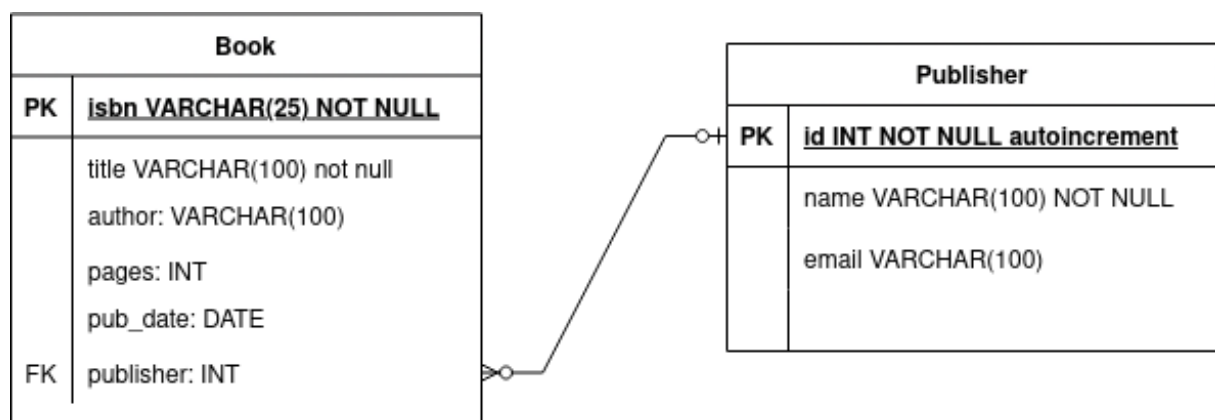

Figure 1: Database schema

1.2 Exercise 1. Database connection

Create a `.env.local` file with the database connection string:

```
DATABASE_URL="mysql://user:password@127.0.0.1:3306/librarysymfony?serverVersion=
```

or, if you are using MariaDb:

```
DATABASE_URL="mysql://user:password@127.0.0.1:3306/librarysymfony?serverVersion=
MariaDB&charset=utf8mb4"
```

The database name must be **librarysymfony**. Use the username and password you want.

1.3 Exercise 2. Creating the models

Create, using Symfony, the models **Book** and **Publisher** according to the schema. Make sure that it accomplishes the relation: each book can have zero or one publisher and each publisher can have zero or many books.

The description of the tables must be:

```
mysql> describe book;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| isbn         | varchar(25)  | NO   | PRI | NULL    |       |
| publisher_id | int(11)      | YES  | MUL | NULL    |       |
| title        | varchar(100) | NO   |     | NULL    |       |
| author       | varchar(100) | YES  |     | NULL    |       |
| pages        | int          | YES  |     | NULL    |       |
| pub_date     | date         | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+

mysql> describe publisher;
+-------+--------------+------+-----+---------+----------------+
| Field | Type         | Null | Key | Default | Extra          |
+-------+--------------+------+-----+---------+----------------+
| id    | int          | NO   | PRI | NULL    | auto_increment |
| name  | varchar(100) | NO   |     | NULL    |                |
| email | varchar(100) | YES  |     | NULL    |                |
+-------+--------------+------+-----+---------+----------------+
```

In addition, you can use the next data if you want some sample data to check the queries:

```
INSERT INTO `publisher` (`id`, `name`, `email`) VALUES
  (NULL, 'Clarion Books', 'info@clarion.com'),
  (NULL, 'Ecco', 'ecco_info@ecco.com'),
  (NULL, 'Scribner', 'scribner@scr.com');

INSERT INTO `book` (`isbn`, `publisher_id`, `title`, `author`, `pages`,
↪  `pub_date`) VALUES
  ('A111B3', '1', 'The Lord of the Rings', 'J.R.R. Tolkien', '1536',
↪   '2020-11-03'),
  ('C221B6', '3', 'Factotum', 'Charles Bukowski', '208', '2002-03-31'),
  ('A546783', '1', 'A Wizard of Earthsea', 'Ursula K. Le Guin', '210',
↪   '2012-09-11'),
```

```
 ('F666764', '2', 'The Lathe Of Heaven', 'Ursula K. Le Guin', '192',
↪  '2008-04-15'),
 ('66788745', '2', 'Foundation', 'Isaac Asimov', '816', '2022-07-07'),
 ('A343445', NULL, 'El Lazarillo de Tormes', NULL, '103', NULL);
```

1.4 Exercise 3. Select queries

Do the next queries:

Query	Template	Route
Select a publisher by its id	single publisher	`/publisher/{id<\d+>}`
Select all the publishers	publisher list	`/publisher_list`
Search a publisher by its name	publisher list	`/publisher/search/{name}`
Select a book by its isbn	single book	`/book/{isbn}`
Select all the books	book list	`/book_list`
Search a book by its title AND author	book list	`/book/search/{title}/{author}`

As an example for the last query, if we are using the data set of the previous point and the query string is `/search/the` the result will be *The Lord of the Rings* and *The Late of Heaven*. If the query is `/search/the/ursula` the result will be only *The Late of Heaven*.

1.5 Exercise 4. Insert queries

Do the queries to insert a new publisher and a new book. These queries must be done manually in the code with sample data for each one.

The routes must be:

- `/book/new`
- `/publisher/new`

Create a new template for each one.

1.6 Exercise 5. Update queries

Do the queries to update a single book and a single publisher. These queries must be done manually in the code with sample data for each one.

The routes must be:

- `/book/edit/{isbn}`
- `/publisher/edit/{id<\d+>}`

You can reuse the same templates of the previous point but with a different message or you can create new templates.

1.7 Exercise 6. Delete queries

Do the queries to delete a single book and a single publisher. These queries must be done manually in the code with sample data for each one.

The routes must be:

- `/book/delete/{isbn}`
- `/publisher/delete/{id<\d+>}`

You can reuse the same templates of the previous point but with a different message or you can create new templates.

1.8 How to submit to GitHub Classroom

Once you finish the task, make a commit with the comment "PRACTICE 12.1 SUBMISSION COMMIT" and push it to GitHub.

1. This task must be submitted to the same repository of the previous one. Remember to make a new branch before starting the practice.
2. Once you finish the task, make a commit with the comment "PRACTICE 12.1 SUBMISSION COM-MIT", merge the branch into the main branch, and push it to GitHub. For example:

```
git commit -m "PRACTICE 12.1 SUBMISSION COMMIT"
git checkout main
git merge library-u12
git push
```

3. It's recommended to tag your commit with the tag "Practice_12.1".

4. Before that, you can do the commits and push you want. If you change your code after your submission commit, make another commit and push with the same text in the message adding the corrections you've done.

If you have any doubt in your task, you can push your code and ask me by email what's your problem. It will make it easier for both the solutions of code issues.