2dv513-Assignment 3

Name: Mosa Kasem Rasol
student-ID: kr222if

1. Idea

For the third assignment I chose to create a search engine for a book library management system, it's interests is in others history, literature and culture.

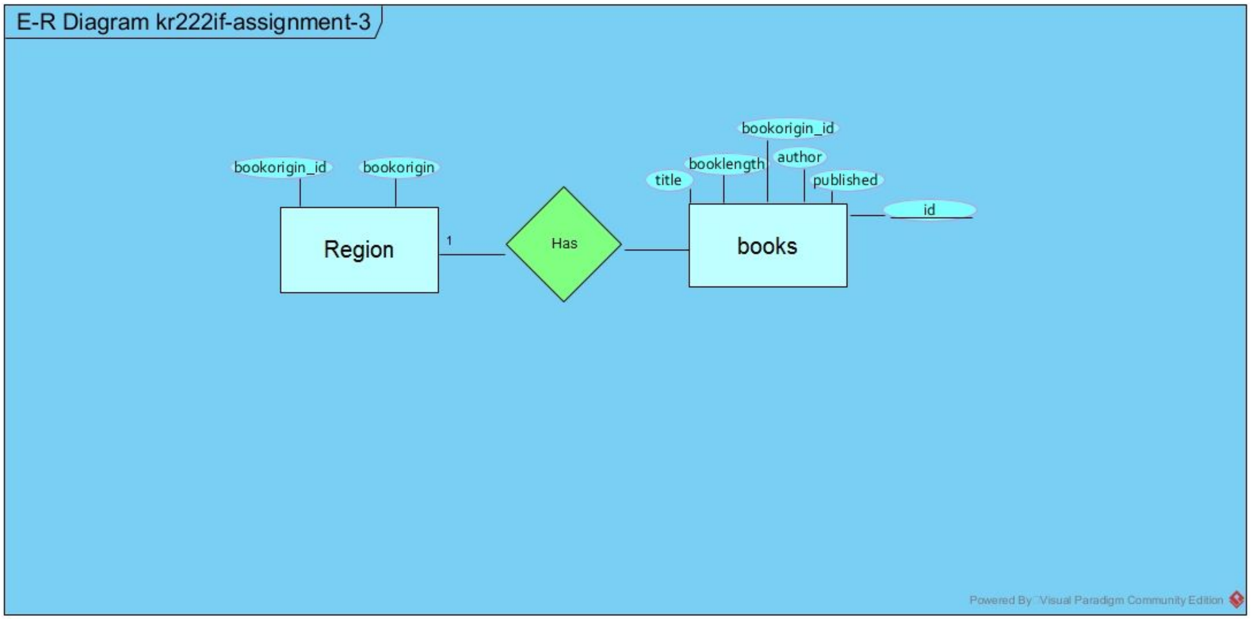
A multicultural book library, whether it's European, Asia, or Africa, you will find it here(**demonstration**)
The ability to search for books or authors.
The library will contain some data to demonstrate and it's also experimental.
The goal is to provide users with the capability to search for the book of their interest by region

OPS!: I did not include genre or description, something which can be included in the future, this is just to demonstrate the idea for the assignment

- **Admin** will be able to insert data/books
- **Users** will be able to search for data/books by
  - Longest book(length in pages)
  - All books
  - Get books by EU
  - Get books by AS
  - Get books by AF
  - (Europe, Asia, Africa)
  - Get books by Author
  - Get books by Title

The Application's **programming language** is in: **JavaScript**
The **database management** system used is: **MySql**(Xampp/phpmyadmin)
The npmjs for the server side is: **Express**

2. Logical model



The Diagram

Since the database requirements was to have inner/natural join so I opted to have region and books, it is possibly for a region to have many books.
Decided to have title of book and it's length(how many pages) aswell and it's published date (by year) and author ofcourse (I had description and genre but had to redesign and insert data manually and I had to cut short because of deadline, extension was helpful, ty M)

Every book came from some part of the world but I wasn't sure which was more interesting, where the author came from or region of the book, maybe I lived in africa for the past 34 years I know a thing or two and could probably write something as an example, that would fit to say the region of the book, author could have been an own entity, with it's own attribute like birthplace of the author, and name, and so on, but since the interest was in the books culture/background, I decided otherwise but the possibility is there still.

The relation

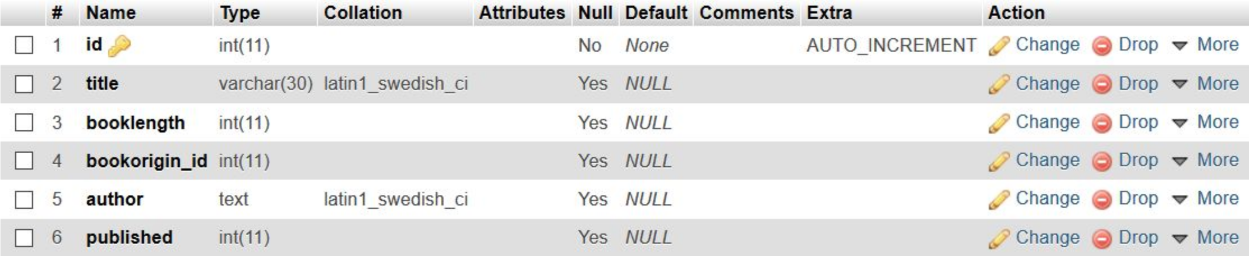
Regarding the relation, so books or "book" I meant (a typoo).
A single book can have 1 region.
A single region can have many books.
Book cannot belong to different regions.

3. SQL-Design

The E/R diagram is translated in the following:

books(

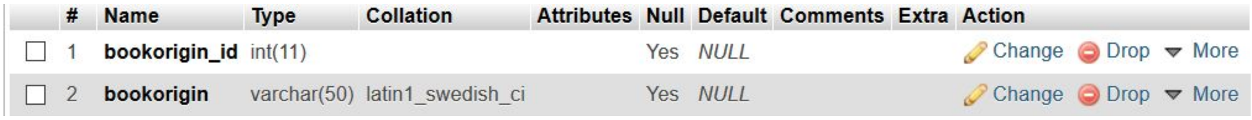
id,
title: string,
booklength: int,
bookorigin_id: int,
author: string,
published: int
)



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop ▼ More
□	2	**title**	varchar(30)	latin1_swedish_ci		Yes	*NULL*			Change Drop ▼ More
□	3	**booklength**	int(11)			Yes	*NULL*			Change Drop ▼ More
□	4	**bookorigin_id**	int(11)			Yes	*NULL*			Change Drop ▼ More
□	5	**author**	text	latin1_swedish_ci		Yes	*NULL*			Change Drop ▼ More
□	6	**published**	int(11)			Yes	*NULL*			Change Drop ▼ More

region(

bookorigin_id: int,
bookorigin: string
)



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**bookorigin_id**	int(11)			Yes	*NULL*			Change Drop ▼ More
□	2	**bookorigin**	varchar(50)	latin1_swedish_ci		Yes	*NULL*			Change Drop ▼ More

4. Queries

1.The first query is pretty basic

```
'SELECT * FROM books'
```

Select everything from books

2.Second query is trying to find the longest book available

```
'SELECT title, booklength FROM books where booklength = (SELECT
MAX(booklength) FROM books)'
```

The query selects the longest book in books and retrieves the title and it’s length.

3,4,5.Third query is to find all books available that are from europe so a bookorigin with ID 1 is for europe ID 2 is for Asia and 3 for Africa (more bookorigins_id can be added for more region), and this query is repeated 3 times for all three ID’s

3.Europe:

```
`SELECT * FROM books
    INNER JOIN
    region ON region.bookorigin_id = books.bookorigin_id
    WHERE books.bookorigin_id = 1`
```

4.Asia:

```
`SELECT * FROM books
    INNER JOIN
    region ON region.bookorigin_id = books.bookorigin_id
    WHERE books.bookorigin_id = 2`
```

5.Africa:

```
`SELECT * FROM books
    INNER JOIN
    region ON region.bookorigin_id = books.bookorigin_id
    WHERE books.bookorigin_id = 3`
```

The query inner joins books bookorigin_id with region bookorigin_id and fetches all books that matches with given ID.

6. Updating is done also (but removed as I tried to make it possible with a friendly userinterface but sadly as said, had some troubles with handlebars, every book has a unique ID and it is possible to link those ID with an A tag. The query goes like this.

```
UPDATE books
SET title='Alfred Schmidt Castle', author=’Johny’, bookorigin_id= 2, published= 2019,
booklength= 452
```

```
WHERE ID =1;
```

5. Implementation

The program is running on node, with library known as express which works as server side and also is connected to phpmyadmin is running on an apache server.

The files are written in javascript programming language and MySql library was used to execute and connect to mysql server.

The mysql server is running on a different port.

If you wish to run the application:
You need a mysql server

- Clone the repo
- Navigate to the cloned repo in a terminal
    1. Write **npm install** in the terminal
    2. Start up **MySql**
    3. /bookclub-node-mysql/model/**dbConnect.js** configure the settings to your DB
    4. Write **npm start** in the terminal
    5. Visit localhost:3000

6. **The Video**
Sorry about the quality, i’ve been very sick