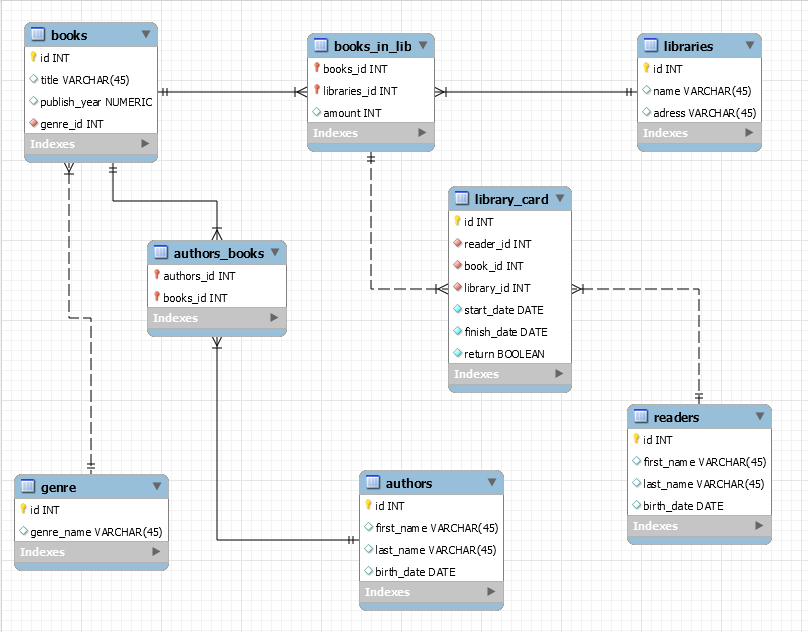
**Предметная область**

**«Библиотека »:** В системе должны поддерживаться режимы поиска книги по заданному критерию (автор , название ), заказа книги , учета клиентов и книг в книгохранилище , выдачи отчетов по запросам (местонахождение книги в архиве или ее отсутствие ), выдачи документов о должниках .

**ER - диаграмма**



**Инфологическая модель**

Данную предметную область будем описывать следующими **таблицами и их полями**:

* Книги:
  + Идентификатор книги(РК);
  + Название книги;
  + Год публикации;
  + Идентификатор жанра (FK).
* Жанры:
  + Идентификатор жанра(PK);
  + Название жанра.
* Авторы:
  + Идентификатор автора (РК)
  + Имя
  + Фамилия
  + Дата рождения
* Книги и авторы (соединительная):
  + Идентификатор книги (РК, FK)
  + Идентификатор автора (РК, FK)
* Библиотеки:
  + Идентификатор библиотеки (РК)
  + Название
  + Адрес
* Книги в библиотеках (соединительная):
  + Идентификатор книги (РК, FK)
  + Идентификатор библиотеки (РК, FK)
  + Количество экземпляров данной книги в данной библиотеке
* Читатели:
  + Идентификатор читателя (РК)
  + Имя
  + Фамилия
  + Дата рождения
* Абонемент на выдачу книг:
  + Идентификатор записи про выдачу книги (РК)
  + Идентификатор читателя (FK)
  + Идентификатор книги (FK)
  + Идентификатор библиотеки (FK)
  + Дата выдачи книги
  + Дата возврата книги
  + Пометка о возврате (да/нет)

**Связи в нашей модели:**

Один-к-одному: нет

Один-ко-многим:

**Жанры – Книги** (разные книги относятся к одному и тому же жанру)

**Читатели-Абонемент на выдачу книг**(один и тот же читатель может взять разные книги из разных библиотек и тогда на него будет больше 1 записи о выдаче книги)

**Книги в библиотеках – Абонемент на выдачу книг** (книга из библиотеки может быть взята разными людьми и тогда для пары (книга, библиотека) будет больше 1 записи о ее выдаче)

*Следующие связи реализованы, что получить связь много-ко-многим между конечными таблицами:*

**Книги-Книги и авторы**

**Авторы-Книги и авторы**

**Книги-Книги в библиотеках**

**Библиотеки – Книги в библиотеках**

Много-ко-многим**:**

**Книги-Авторы**

**Книги-Библиотеки**

**Читатели – Книги в библиотеках**

Таблицы **«Книги и авторы»** и «**Книги в библиотеках**» являются соединительными для реализации связи много-ко-многим, которая присутствует между такими сущностями:

* книги-авторы (у одной книги может быть больше 1 автора, у 1 автора может быть больше 1 книги),
* книги-библиотеки (экземпляры одной книги могут быть в разных библиотеках, в одной библиотеке есть экземпляры разных книг). Эта таблица также несет информацию о количестве экземпляров отдельной книги в отдельной библиотеке.

В таблице «**Абонемент на выдачу книг**» реализуется запись про то **кому** выдали книгу, **какую** книгу, **из какой** библиотеки.

В этой таблице сделан отдельный РК, так как, во-первых, обычно в библиотеках у абонементов есть свои шифры, а, во-вторых, чтобы один и тот же человек мог взять одну и ту же книгу в одной и той же библиотеке несколько раз (взял раз, вернул, взял снова). Потому как если мы сделаем составной ключ из 3 полей, то записи о повторной выдаче будут нарушать требование на уникальность первичного ключа.

Данная таблица также является соединительной, но она несет допольнительную информацию, потому является отдельной сущностью.

**INSERT**

-- заполняем таблицу авторов

**INSERT** **INTO** `**library**`.`authors` (`**id**`, `first\_name`, `last\_name`, `birth\_date`) **VALUES** ('4', 'Melissa', 'Nichols', '1978-12-23')

-- заполняем таблицу читателей

**INSERT** **INTO** `**library**`.`readers` (`**id**`, `first\_name`, `last\_name`, `birth\_date`) **VALUES** ('2', 'Nadiia', 'Velychko', '1975-09-22')

-- заполняем таблицу книг

**INSERT** **INTO** `**library**`.`books` (`**id**`, `title`, `publish\_year`, `genre\_id`) **VALUES** ('1', 'VOODOO ISLAND', '2010', '1');

-- заполняем соединительную таблицу авторы и их книги

**INSERT** **INTO** `**library**`.`authors\_books` (`authors\_id`, `books\_id`) **VALUES** ('1', '1');

-- заполняем таблицу библиотеки

**INSERT** **INTO** `**library**`.`libraries` (`**id**`, `name`, `adress`) **VALUES** ('1', 'Shevchenko library', 'Heroiv Dnipra 47');

-- заполняем соединительную таблицу книги в библиотеках

**INSERT** **INTO** `**library**`.`books\_in\_lib` (`books\_id`, `libraries\_id`, `amount`) **VALUES** ('1', '3', '1');

-- заполняем таблицу записей о выдаче книг

**INSERT** **INTO** `**library**`.`library\_card` (`**id**`, `reader\_id`, `book\_id`, `library\_id`, `start\_date`, `finish\_date`, `**return**`) **VALUES** ('1', '1', '1', '3', '2016-02-11', '2016-03-12', '0');

**ALTER TABLE**

-- добавляем ограничение на колонку Количество книг: не может быть < 0

**ALTER** **TABLE** books\_in\_lib

**ADD** **CONSTRAINT** chk\_amount **CHECK** (books\_in\_lib.amount>=0);

**SELECT**

-- вывести все книги определенного автора

**SELECT** books.title, books.publish\_year

**FROM** books

**JOIN** authors\_books **ON** (books.**id** = authors\_books.books\_id)

**JOIN** authors **ON** (authors\_books.authors\_id=authors.**id**)

**WHERE** authors.last\_name = 'Meyer';

-- вывести авторов опеределенной книги

**SELECT** authors.first\_name **as** Authors\_of, authors.last\_name **as** THE\_Firm

**FROM** authors

**JOIN** authors\_books **ON** (authors.**id**=authors\_books.authors\_id)

**JOIN** books **ON** (authors\_books.books\_id=books.**id**)

**WHERE** books.title="THE FIRM";

-- вывести книгу по автору и названию, если такой нет, то не выведет ничего

**SELECT** books.title, books.publish\_year, authors.first\_name, authors.last\_name

**FROM** books

**JOIN** authors\_books **ON** (books.**id** = authors\_books.books\_id)

**JOIN** authors **ON** (authors\_books.authors\_id=authors.**id**)

**WHERE** authors.last\_name = "Rudyard"

**AND** books.title="JUNGLE BOOK";

-- вывести в каких библиотеках сколько экземпляров книги осталось

**SELECT** books.**id** **as** b\_id, libraries.**id** **as** lib\_id, authors.last\_name, authors.first\_name, books.title, libraries.name, books\_in\_lib.amount -

(

**SELECT** **COUNT**(library\_card.**id**)

**FROM** library\_card

**WHERE** library\_card.book\_id = b\_id

**AND** library\_card.library\_id = lib\_id

**AND** library\_card.**return**=0) **as** books\_in\_stock

**FROM** libraries

**JOIN** books\_in\_lib **ON** (libraries.**id** = books\_in\_lib.libraries\_id)

**JOIN** books **ON** (books\_in\_lib.books\_id=books.**id**)

**JOIN** authors\_books **ON** (books.**id** = authors\_books.books\_id)

**JOIN** authors **ON** (authors\_books.authors\_id=authors.**id**)

**WHERE** books.title="JUNGLE BOOK";

-- вывести ФИО должников и какую книгу в какую библиотеку они должны

**SELECT** **r**.last\_name, **r**.first\_name, b.title **as** Book, **l**.name **Library**

**FROM** readers **r**

**JOIN** library\_card lib\_c **ON** (**r**.**id** = lib\_c.reader\_id)

**JOIN** books b **ON** (lib\_c.book\_id = b.**id**)

**JOIN** books\_in\_lib bl **ON** (b.**id** = bl.books\_id)

**JOIN** libraries **l** **ON** (bl.libraries\_id = **l**.**id**);

**UPDATE**

-- обновить запись о выдаче книги после того, как книгу вернули, т.е. изменить метку возврата с 0 на 1 : return=0->1

**UPDATE** **library**.library\_card

**SET** library\_card.**return** = 1

**WHERE** reader\_id=

(

**SELECT** readers.**id**

**FROM** readers

**WHERE** readers.last\_name = "Yellow"

**AND** readers.first\_name = "Victor"

)

**AND**

book\_id=

(

**SELECT** books.**id**

**FROM** books

**WHERE** books.title="JUNGLE BOOK"

)

**AND**

library\_id=

(

**SELECT** libraries.**id**

**FROM** libraries

**WHERE** libraries.name="Central library"

);

**DELETE**

-- удалить запись о выдаче книги по шифру записи

**DELETE** **FROM** `**library**`.`library\_card` **WHERE** `**id**`='8';

-- удалить запись о выдаче книги по Фамилии и имени читателя и Названию книги

**DELETE** **FROM** **library**.library\_card

**WHERE** reader\_id=

(

**SELECT** readers.**id**

**FROM** readers

**WHERE** readers.last\_name = "Velychko"

**AND** readers.first\_name = "Nadiia"

)

**AND**

book\_id=

(

**SELECT** books.**id**

**FROM** books

**WHERE** books.title="THE FIRM"

);

**TRIGGER**

/\* перед каждой записью о выдаче книги мы проверяем есть ли эта книга в библиотеке, с которой мы выдаем ее - от количества экземпляров данной книги в данной библиотеке мы отнимаем число записей о выдаче данной книги с этой библиотеки с отметкой "не вернули", т.е. return = 0 \*/

delimiter $$

**create** **trigger** Check\_amount\_of\_books **before** **insert** **on** library\_card

**for** **each** **row**

**BEGIN**

**DECLARE** n **INT** **default** 0;

**SET** n = (

**SELECT** books\_in\_lib.amount - **COUNT**(library\_card.**id**)

**FROM** library\_card **JOIN** books\_in\_lib **ON** (library\_card.book\_id = books\_in\_lib.books\_id)

**WHERE** library\_card.book\_id = **NEW**.book\_id **AND** library\_card.library\_id = **NEW**.library\_id **AND** library\_card.**return**=0

);

**IF**(n <= 0) **THEN**

SIGNAL **sqlstate** '45001' **set** message\_text = "Error! number of books in store: 0";

**ELSE**

SIGNAL **SQLSTATE** '01000';

**END IF**;

**END** $$

delimiter ;