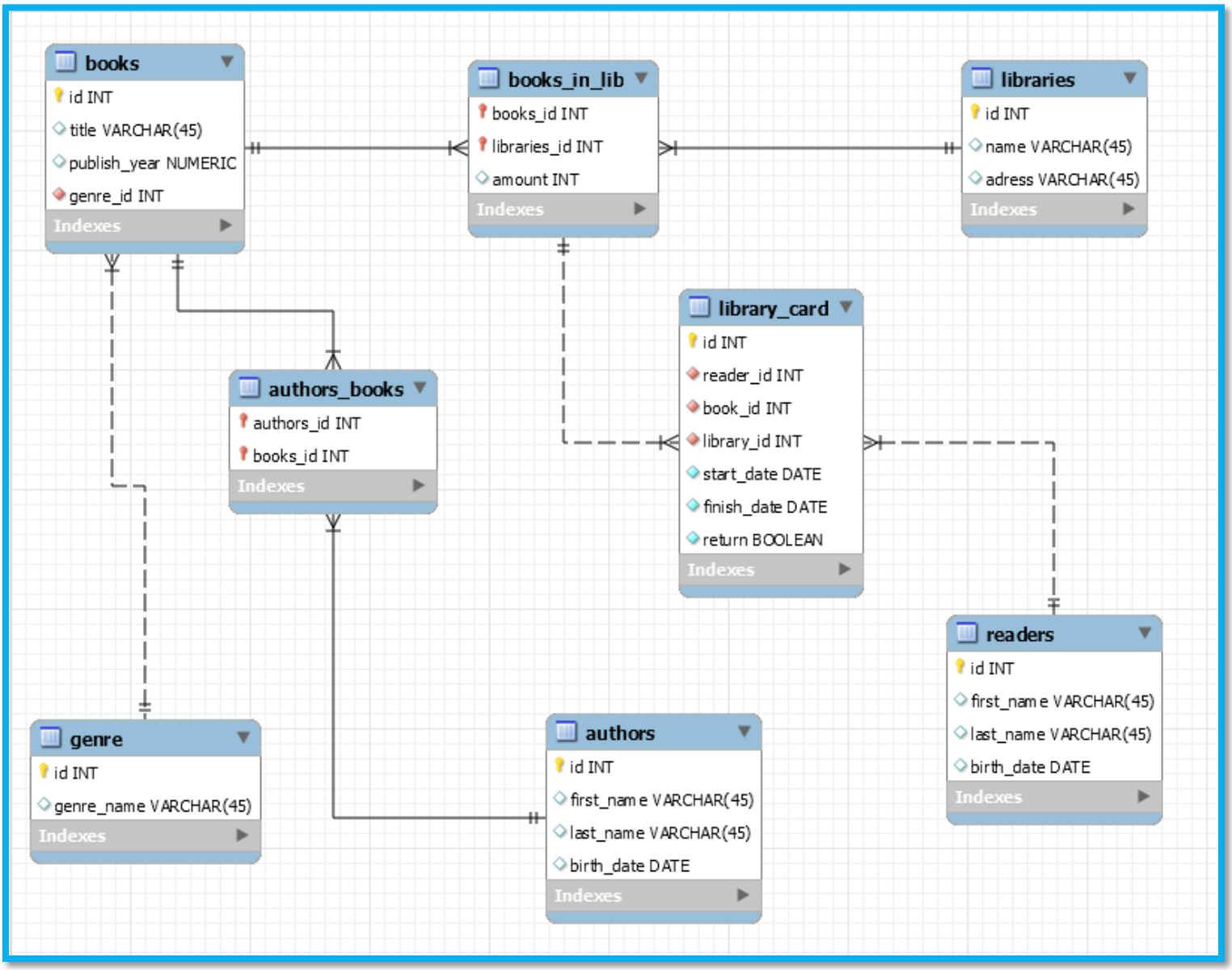
Предметная область

«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(PK);
  - Название книги;
  - Год публикации;
  - Идентификатор жанра (FK).
- Жанры:
  - Идентификатор жанра(PK);
  - Название жанра.
- Авторы:
  - Идентификатор автора (PK)
  - Имя
  - Фамилия
  - Дата рождения
- Книги и авторы (соединительная):
  - Идентификатор книги (PK, FK)
  - Идентификатор автора (PK, FK)
- Библиотеки:
  - Идентификатор библиотеки (PK)
  - Название
  - Адрес
- Книги в библиотеках (соединительная):
  - Идентификатор книги (PK, FK)
  - Идентификатор библиотеки (PK, FK)
  - Количество экземпляров данной книги в данной библиотеке
- Читатели:
  - Идентификатор читателя (PK)
  - Имя
  - Фамилия
  - Дата рождения
- Абонемент на выдачу книг:
  - Идентификатор записи про выдачу книги (PK)
  - 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 ;
```