ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

**Московский институт электроники и математики им. А.Н. Тихонова**

Опекунова Алина Алексеевна, группа БИВ193

**ОТЧЕТ**

**ПО ДОМАШНЕЙ РАБОТЕ**

по дисциплине «Базы данных»

Тема: «База данных публичной библиотеки»

Дата сдачи отчета 20.04.2021

Москва 2021 г.

1. Инфологическое проектирование

### **1.1. Анализ предметной области**

База данных создаётся для учета выдачи книг в библиотеке. БД должна содержать данные об авторах, читателях, издательствах и книгах.

В соответствии с предметной областью система строится с учётом следующих особенностей:

* Каждая книга может быть написана несколькими авторами (указываются все), автор может написать несколько книг.
* Каждый читатель может брать несколько книг, каждая книга может быть взята одним читателем.
* Каждая книга может быть издана в одном издательстве, каждое издательство может издавать несколько книг.
* Каждая книга может быть выдана, в списке выданных книг может находиться несколько книг.
* В списке выданных книг каждой книге соответствует единственный читатель, читатель может брать несколько книг.
* Каждая книга может быть забронирована, в списке бронированных книг может находиться несколько книг.
* Каждая книга дается на определенный срок, и в списке выданных книг отмечается дата выдачи и возврата.
* Книга бронируется на определенный срок, начиная с указанной даты, после чего она может быть либо выдана, либо нет, в обоих случаях запись о бронировании удаляется после окончания указанного срока.
* В списке забронированных книг каждой книге соответствует единственный читатель, читатель может бронировать несколько книг.
* Каждая книга относится к одной или нескольким тематикам, к каждой тематике может относиться несколько книг.
* Тематика книг: художественная литература , современная литература, классическая литература, исторический роман, любовный, мифы, сказки, фольклор, ужасы, мистика, детективы, триллеры и т.д.
* Книга может быть частью серии книг, в серии может быть несколько книг.
* Каждая книга может быть написана на одном языке и переведена на другие.
* В библиотеке может быть несколько экземпляров одной книги

### **1.2. Создание ER-диаграммы**

В процессе предварительного проектирования выделено 6 сущностей в составе БД и в скобках указаны их атрибуты, причем и**дентифицирующие** атрибуты мы выделим полужирным шрифтом, *многозначные* – курсивом, составные подчеркнем:

1. Книги (**шифр книги**, название, серия, тематика, язык оригинала, язык перевода, рейтинг)
2. Автор (**ID**, ФИО, пол, дата рождения, страна, рейтинг)
3. Читатель (**номер читательского билета**, ФИО, пол, дата рождения, телефон, адрес*,* **паспортные данные**)
4. Издательство (**код издательства**, название, *город*, *телефон*, рейтинг)
5. Список выданных книг (**номер**, книга, читатель, дата выдачи, дата возврата)
6. Список забронированных книг (**номер**, книга, читатель, дата бронирования, срок бронирования)

Отношения и зависимости между сущностями на ER–диаграмме (рис.1).

Издательства

Читатели

Книги

Авторы

издавать

писать

K

K

1

1

1

N

1

1

Выдача книг

Бронирование книг

получать

бронировать

включать

включать

осуществить

1

1

M

R

X

L

Рис. 1. ER–диаграмма ПрО «Библиотека»

### **1.3. Анализ информационных задач и круга пользователей системы**

Определим группы пользователей, их основные задачи и запросы к БД:

1. Читатель:

* Получение списка книг по тематике;
* Получение списка книг по серии;
* Получение списка книг по автору;
* Получение списка книг по рейтингу;
* Получение списка книг в наличии;

1. Сотрудник отдела комплектации:

* Изменение и добавление данных в таблице читателя;
* Изменение и добавление данных в таблице издательства;
* Изменение и добавление данных в таблице с книгами.

1. Сотрудник по работе с читателями:

* Заполнение и изменение данных в таблице выдачи книг;
* Заполнение и изменение данных в таблице бронирования книг;
* получение списка читателей, которые взяли книги в определенный период (например, список читателей, которые должны вернуть взятые книги на текущей неделе);
* получение сведений о читателях;
* определение количества книг, взятых определенным читателем на момент запроса;
* получение числа экземпляров книги в наличии по ее названию.

2. Логическое проектирование реляционной БД

### **2.1 Преобразование ER-диаграммы в схему БД**

* Связь «Писать» между сущностями «Авторы» и «Книги» принадлежат к типу многие-ко-многим, поэтому было решено реализовать ее через отношение «Авторы книг» (книга, автор).
* Так как может быть несколько экземпляров одной книги, принято решение реализовать отношение «Экземпляры», которое будет связью между сущностями «Книги» и «Выдача книг»

Схема РБД, полученная из ER-диаграммы представлена на рис.2.

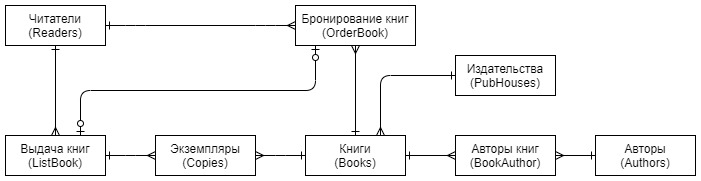


Рис. 2 Схема базы данных «Библиотека»

### **2.2 Составление реляционных отношений**

Таблица 1. Схема отношения КНИГИ (Books)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Шифр книги | B\_code | N(5) | первичный ключ |
| Название | B\_name | C(100) | обязательное поле |
| Серия | B\_genre | C(20) | необязательное поле |
| Тематика | B\_category | C(20) | обязательное поле |
| Издательство | B\_pubhouse | N(5) | Внешний ключ таблицы «PubHouses» |
| Язык оригинала | B\_Olanguage | C(20) | обязательное поле |
| Язык перевода | B\_Tlanguage | C(20) | необязательное поле |
| Рейтинг | B\_rating | N(3,2) | необязательное поле |

Таблица 2. Схема отношения АВТОРЫ КНИГ (BookAuthor)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| код книги | BA\_book | N(5) | Внешний ключ таблицы «Books» |
| ID автора | BA\_author | N(5) | Внешний ключ таблицы «Authors» |

Таблица 3. Схема отношения АВТОРЫ (Authors)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| ID | A\_ID | N(5) | первичный ключ |
| ФИО | A\_FIO | C(100) | обязательное составное поле |
| Пол | A\_sex | C(1) | обязательное поле |
| Дата рождения | A\_born | D | обязательное поле |
| Страна | A\_country | C(20) | необязательное поле |
| Рейтинг | A\_rating | N(3,2) | необязательное поле |

Таблица 4. Схема отношения ЧИТАТЕЛЬ (Readers)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Номер читательского билета | R\_number | N(5) | первичный ключ |
| ФИО | R\_FIO | C(100) | обязательное составное поле |
| Пол | R\_sex | C(1) | обязательное поле |
| Дата рождения | R\_born | D | обязательное поле |
| Телефон | R\_telephone | C(30) | обязательное поле |
| Почта | R\_mail | C(30) | необязательное поле |
| Адрес | R\_adres | C(100) | обязательное составное многозначное поле |
| Паспортные данные | R\_pasport | C(15) | Обязательное уникальное поле |

Таблица 5. Схема отношения ИЗДАТЕЛЬСТВО (PubHouses)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Код издательства | P\_code | N(5) | первичный ключ |
| Название | P\_name | C(100) | обязательное поле |
| Город | P\_addres | C(100) | обязательное составное многозначное поле |
| Телефон | P\_telephone | C(30) | обязательное многозначное поле |
| Рейтинг | P\_rating | N(3) | необязательное поле |

Таблица 6. Схема отношения ЭКЗЕМПЛЯРЫ (Copies)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Номер | C\_num | N(5) | Первичный ключ |
| Книга | C\_book | N(5) | Внешний ключ к таблице «Books» |

Таблица 7. Схема отношения ВЫДАЧА (ListBook)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Номер выдачи | L\_num | N(5) | Первичный ключ |
| Книга | L\_book | N(5) | Внешний ключ к таблице «Copies» |
| Читатель | L\_reader | N(5) | Внешний ключ к таблице «Readers» |
| Дата выдачи | L\_startdate | D | обязательное поле |
| Дата возврата | L\_enddate | D | Возможность пустого значения, больше даты выдачи |
| Номер брони | L\_order | N(5) | Внешний ключ к таблице «OrderBook» |

Таблица 8. Схема отношения БРОНИРОВАНИЕ (OrderBook)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Номер бронирования | O\_num | N(5) | Первичный ключ |
| Книга | O\_book | N(5) | Внешний ключ к таблице «Books» |
| Читатель | O\_reader | N(5) | Внешний ключ к таблице «Readers» |
| Дата выдачи | O\_startdate | D | обязательное поле |
| Срок бронирования | O\_enddate | N(2) | обязательное поле, меньше 31 (забронировать можно только на срок меньший 31 дня) |

### **2.3 Нормализация полученных отношений**

**Таблица КНИГИ**

В отношении «Книги» вынесем в отдельные таблицы поля серия (табл.9), тематика (табл. 10). Так как одна книга может относиться к нескольким тематикам, то отношения между сущностями «Книги» и «Тематика» реализуем через таблицу «Тематика книг» (табл. 11). Создадим справочную таблицу «Языки» (табл.12)

Таблица 9. Схема отношения СЕРИЯ (Series)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Название серии | S\_name | C(20) | Первичный ключ |

Таблица 10. Схема отношения ТЕМАТИКА (Category)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Название Тематики | C\_name | C(20) | Первичный ключ |

Таблица 11. Схема отношения ТЕМАТИКА КНИГ (BookCategory)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Тематика | BC\_category | C(20) | Внешний ключ к таблице «Category» |
| Книга | BC\_book | N(5) | Внешний ключ к таблице «Books» |

Таблица 12. Схема отношения Языки (Languages)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Язык | language | C(20) | Первичный ключ |

Таблица 13. Схема отношения КНИГИ (Books) после нормализации

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Шифр книги | B\_code | N(5) | первичный ключ |
| Название | B\_name | C(100) | обязательное поле |
| Серия | B\_series | C(20) | Внешний ключ к таблице «Series» |
| Язык оригинала | B\_Olanguage | C(20) | Внешний ключ к таблице «Languages», обязательное поле |
| Язык перевода | B\_Tlanguage | C(20) | Внешний ключ к таблице «Languages», необязательное поле |
| Рейтинг | B\_rating | N(3,2) | необязательное поле |
| Издательство | B\_pubhouse | N(5) | Внешний ключ таблицы «PubHouses» |

**Таблица АВТОРЫ**

В отношении «Авторы» разделим составной атрибут ФИО на 3 атрибута Фамилия, Имя и Отчество. Создадим справочную таблицу «Страны» (табл.15).

Таблица 14. Схема отношения АВТОРЫ (Authors) после нормализации

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| ID | A\_ID | N(5) | первичный ключ |
| Фамилия | A\_surname | C(20) | обязательное поле |
| Имя | A\_name | C(10) | обязательное поле |
| Отчество | A\_fname | C(15) | необязательное поле |
| Пол | A\_sex | C(1) | обязательное поле |
| Дата рождения | A\_born | D | обязательное поле |
| Страна | A\_country | C(20) | внешний ключ к таблице «Countries» |
| Рейтинг | A\_rating | N(3,2) | необязательное поле |

Таблица 15. Схема отношения СТРАНЫ (Countries)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Название страны | Country\_name | C(20) | Первичный ключ |

**Таблица ЧИТАТЕЛИ.**

В отношении «Читатели» разделим составной атрибут ФИО на 3 атрибута Фамилия, Имя. У читателя при регистрации фиксируется обязательный номер телефона, адрес фактического проживания и одна необязательная электронная почта, поэтому можно оставить их в одной таблице со всей информацией.

Таблица 16. Схема отношения ЧИТАТЕЛЬ (Readers) после нормализации

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Номер читательского билета | R\_number | N(5) | первичный ключ |
| Фамилия | R\_surname | C(20) | обязательное поле |
| Имя | R\_name | C(10) | обязательное поле |
| Отчество | R\_fname | C(15) | необязательное поле |
| Пол | R\_sex | C(1) | обязательное поле |
| Дата рождения | R\_born | D | обязательное поле |
| Телефон | R\_telephone | C(30) | обязательное поле |
| Почта | R\_mail | C(30) | необязательное поле |
| Адрес | R\_adres | C(100) | обязательное составное поле |
| Паспортные данные (серия и номер) | R\_pasport | C(10) | обязательное уникальное поле |

**Таблица ИЗДАТЕЛЬСТВА.**

Вынесем в отдельное отношение поле «Город» (табл.17), а так как одно издательство может иметь несколько филиалов в разных городах, то отношение между сущностью «Издательства» и «Города» реализуем через таблицу «Адрес издательства» (табл.18)

Таблица 17. Схема отношения ГОРОДА (Cities)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Код города | C\_code | N(5) | первичный ключ |
| Название города | C\_name | C(20) | обязательное поле |
| Страна | C\_country | C(20) | внешний ключ к таблице «Countries» |

Таблица 18. Схема отношения АДРЕС ИЗДАТЕЛЬСТВА (PubCity)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Город | PC\_city | N(5) | внешний ключ к таблице «Cities» |
| Издательство | PC\_pub | N(5) | внешний ключ к таблице «PubHouses» |

Также для качественной связи с издательством может понадобиться несколько номеров, поэтому необходимо вынести их вместе с ИД издательства в отдельную таблицу, чтобы все атрибуты были унарными.

Таблица 19. Схема отношения ТЕЛЕФОНЫ (Telephones)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Издательство | T\_pub | N(5) | внешний ключ к таблице «PubHouses» |
| Телефон | T\_tel | С(11) | Обязательное поле |

Таблица 20. Схема отношения ИЗДАТЕЛЬСТВО (PubHouses) после нормализации

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Код издательства | P\_code | N(5) | первичный ключ |
| Название | P\_name | C(100) | обязательное поле |
| Рейтинг | P\_rating | N(3,2) | необязательное поле |

Таблицы АВТОРЫ КНИГ (табл.2), ЭКЗЕМПЛЯРЫ (табл.6), ВЫДАЧА (табл. 7), БРОНИРОВАНИЕ (табл. 8), после нормализации остаются без изменений. Исходя из этого, получаем окончательную схему БД (рис.3).

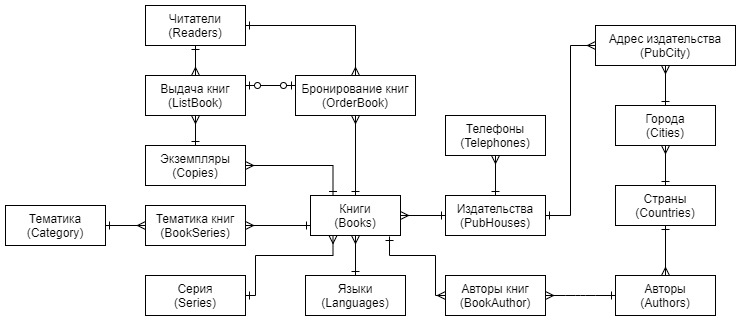


Рис. 3 Окончательная схема РБД «Библиотека»

**2.4 Определение дополнительных ограничений целостности**

1. Дата возврата книги всегда больше даты выдачи (данное ограничение целостности можно реализовать при создании таблицы «Выдача книг»)
2. В случае если забронированная книга будет выдаваться, дата выдачи больше даты бронирования и входит в интервал срока бронирования (это ограничение целостности можно реализовать в виде триггера для добавления данных в таблицу «Выдача книг», то есть пользователь должен получать сообщение об ошибке, если при вводе данных дата выдачи не входит в указанный диапазон (дата бронирования + срок бронирования))

**2.5 Описание групп пользователей и прав доступа**

Используются следующие сокращения: добавление данных (I – insert), чтение данных (S – select), модификация данных (U – update), удаление данных (D – delete).

Опишем для каждой группы пользователей права доступа к каждой таблице (Таблица 19, 20). Права доступа должны быть распределены так, чтобы для каждого объекта БД был хотя бы один пользователь, который имеет право добавлять и удалять данные из объекта.

Таблица 19. Права доступа к таблицам для групп пользователей

|  |  |  |  |
| --- | --- | --- | --- |
| Таблицы | Группы пользователей (роли) | | |
| Читатель | Сотрудник отдела комплектации | Сотрудник отдела по работе с читателями |
| Тематика | S | ISUD | S |
| Тематика книг | S | ISUD | S |
| Серия | S | ISUD | S |
| Книги | S | ISUD | S |
| Языки |  | ISUD |  |
| Экземпляры |  | ISUD | S |
| Читатели | S – только свои | S | ISUD |
| Выдача книг | S – только свои |  | ISUD |
| Бронирование книг | S – только свои |  | ISUD |
| Авторы книг | S | ISUD | S |
| Авторы | S | ISUD | S |
| Издательства | S | ISUD | S |
| Адрес издательства |  | ISUD |  |
| Город |  | ISUD |  |
| Страны | S | ISUD | S |
| Телефон |  | ISUD |  |

Права доступа читателя к отношению «Читатели» будут назначены через представление, так как он может просматривать только свои данные, а не данные всех пользователей библиотеки. То же самое касается и отношений «Выдача книг» и «Бронирование книг» – читатель может просматривать только свои собственные заказы или список выданных ему книг.

Таблица 20. Права доступа к представлениям для групп пользователей

|  |  |  |  |
| --- | --- | --- | --- |
| Представления | Группы пользователей (роли) | | |
| Читатель | Сотрудник отдела комплектации | Сотрудник отдела по работе с читателями |
| Профиль читателя  (my\_infomation) | S |  |  |
| Заказ книг (my\_order) | S |  |  |
| Список выданных книг (my\_list) | S |  |  |
| Список книг определенной тематики (bcategory) | S |  |  |
| Список книг определенной серии (bseries) | S |  |  |
| Список книг определенного автора (bauthor) | S |  |  |
| Список книг в наличии (curr\_books ) | S |  | S |
| Список читателей, которые взяли книги в определенный период и еще не вернули их (curr\_debtors) |  |  | S |
| Определение количества книг, взятых определенным читателем и еще не возвращенных на момент запроса (curr\_num\_book) |  |  | S |
| Получение числа экземпляров книг с одинаковым названием (сopies\_books) |  |  | S |

3. Реализация проекта базы данных

Для реализации БД была выбрана СУБД PostgreSQL, которая позволяет задавать ограничения целостности.

**3.1 Создание таблиц**

1. Отношение Readers (читатели):

CREATE TABLE Readers (

R\_number NUMERIC(5) PRIMARY KEY,

R\_surname VARCHAR(20) NOT NULL,

R\_name VARCHAR(10) NOT NULL,

R\_fname VARCHAR(15),

R\_sex CHAR(1) NOT NULL CHECK(R\_sex IN('м','ж')),

R\_born DATE NOT NULL,

R\_telephone CHAR(11) NOT NULL,

R\_mail VARCHAR(20),

R\_address VARCHAR(100) NOT NULL,

R\_pasport CHAR(10) NOT NULL UNIQUE);

1. Отношение OrderBook (Бронирование):

CREATE TABLE OrderBook (

O\_num NUMERIC(5) PRIMARY KEY,

O\_book NUMERIC(5) NOT NULL,

O\_reader NUMERIC(5) NOT NULL,

O\_startdate DATE NOT NULL,

O\_enddate DATE NOT NULL,

FOREIGN KEY (O\_book) REFERENCES Books(B\_code),

FOREIGN KEY (O\_reader) REFERENCES Readers(R\_number));

1. Отношение ListBook (Выдача):

CREATE TABLE ListBook (

L\_num NUMERIC(5) PRIMARY KEY,

L\_book NUMERIC(5) NOT NULL,

L\_reader NUMERIC(5) NOT NULL,

L\_startdate DATE NOT NULL,

L\_enddate DATE CHECK(L\_enddate>L\_startdate),

L\_order NUMERIC(5),

FOREIGN KEY (L\_book) REFERENCES Copies(C\_num),

FOREIGN KEY (L\_reader) REFERENCES Readers(R\_number),

FOREIGN KEY (L\_order) REFERENCES OrderBook(O\_num));

1. Отношение Books (книги):

CREATE TABLE Books (

B\_code NUMERIC(5) PRIMARY KEY,

B\_name VARCHAR(100) NOT NULL,

B\_series VARCHAR(20),

B\_Olanguage VARCHAR(20) NOT NULL,

B\_Tlanguage VARCHAR(20),

B\_year NUMERIC (4) NOT NULL,

B\_rating NUMERIC(3,2),

B\_Pubhouse NUMERIC(5) NOT NULL,

FOREIGN KEY (B\_series) REFERENCES Series(S\_name),

FOREIGN KEY (B\_Olanguage) REFERENCES Languages(language\_name),

FOREIGN KEY (B\_Tlanguage) REFERENCES Languages(language\_name),

FOREIGN KEY (B\_Pubhouse) REFERENCES Pubhouses(P\_code));

1. Отношение Category (Тематика):

CREATE TABLE Category (

C\_name VARCHAR(50) PRIMARY KEY);

1. Отношение BookCategory (Тематика книг):

CREATE TABLE BookCategory (

BC\_category VARCHAR(20) NOT NULL,

BC\_book NUMERIC(5) NOT NULL,

FOREIGN KEY (BC\_category) REFERENCES Category(C\_name),

FOREIGN KEY (BC\_book) REFERENCES Books(B\_code));

1. Отношение Series (Серии):

CREATE TABLE Series (

S\_name VARCHAR(20) PRIMARY KEY);

1. Отношение Languages (Языки):

CREATE TABLE Languages (

Language\_name VARCHAR(20) PRIMARY KEY);

1. Отношение PubHouses (Издательства):

CREATE TABLE PubHouses (

P\_code NUMERIC(5) PRIMARY KEY,

P\_name VARCHAR(100) NOT NULL,

P\_rating NUMERIC(3,2));

1. Отношение Telephones (Телефоны):

CREATE TABLE Telephones (

T\_pub NUMERIC(5) NOT NULL,

T\_tel NUMERIC(11) NOT NULL,

FOREIGN KEY (T\_pub) REFERENCES Pubhouses(P\_code));

1. Отношение PubCity (Адрес издательства):

CREATE TABLE PubCity (

PC\_city NUMERIC(5) NOT NULL,

PC\_pub NUMERIC(5) NOT NULL,

FOREIGN KEY (PC\_pub) REFERENCES Pubhouses(P\_code),

FOREIGN KEY (PC\_city) REFERENCES Cities(C\_code));

1. Отношение Cities (Города):

CREATE TABLE Cities (

C\_code NUMERIC(5) PRIMARY KEY,

C\_name VARCHAR(20) NOT NULL,

C\_country VARCHAR(20) NOT NULL,

FOREIGN KEY (C\_country) REFERENCES Countries(Country\_name));

1. Отношение Countries (Страны):

CREATE TABLE Countries (

Country\_name VARCHAR(20) PRIMARY KEY);

1. Отношение BookAuthor (Авторы книг):

CREATE TABLE BookAuthor (

BA\_book NUMERIC(5) NOT NULL,

BA\_author NUMERIC(5) NOT NULL,

FOREIGN KEY (BA\_book) REFERENCES Books(B\_code),

FOREIGN KEY (BA\_author) REFERENCES Authors(A\_ID));

1. Отношение Authors (Авторы):

CREATE TABLE Authors (

A\_ID NUMERIC(5) PRIMARY KEY,

A\_surname VARCHAR(20) NOT NULL,

A\_name VARCHAR(10) NOT NULL,

A\_fname VARCHAR(15),

A\_sex CHAR(1) NOT NULL CHECK(A\_sex IN('м','ж')),

A\_born DATE NOT NULL,

A\_country VARCHAR(20),

A\_rating NUMERIC(3,2),

FOREIGN KEY (A\_country) REFERENCES Countries(Country\_name));

1. Отношение Copies (Экземпляры):

CREATE TABLE Copies (

C\_num NUMERIC(5) PRIMARY KEY,

C\_book NUMERIC(5) NOT NULL,

FOREIGN KEY (C\_book) REFERENCES Books(B\_code));

**3.2 Создание представлений**

**--Представления для читателя (только его личные данных, информация о забронированных и взятых им книгах)**

CREATE OR REPLACE VIEW my\_infomation AS

SELECT \* FROM Readers

WHERE R\_number=’номер чит.билета’;

CREATE OR REPLACE VIEW my\_orders AS

SELECT \* FROM OrderBook

WHERE O\_reader=’номер чит.билета’;

CREATE OR REPLACE VIEW my\_list AS

SELECT \* FROM ListBook

WHERE L\_reader=’номер чит.билета’;

**-- Список книг определенной тематики**

CREATE OR REPLACE VIEW bcategory AS

SELECT bc.bc\_category, b.\*

FROM BookCategory bc, Books b

WHERE bc.BC\_book=B\_code;

**-- Список книг определенной по серии**

CREATE OR REPLACE VIEW bseries AS

SELECT b.\*

FROM Books b, Series s

WHERE s\_name=B\_series;

**-- Список книг определенного автора**

CREATE OR REPLACE VIEW bauthor AS

SELECT books.\*

FROM Books, Authors, BookAuthor

WHERE ba\_book=b\_code AND ba\_author=a\_ID;

**-- Список книг в наличии на текущий момент**

CREATE OR REPLACE VIEW curr\_books AS

SELECT \*

FROM Books

WHERE b\_code NOT IN(SELECT l\_book FROM ListBook WHERE L\_enddate IS NULL);

Комментарий: когда книга выдана, дата возврата не указана, она указывается в момент возврата

**--Список читателей, которые взяли книги в определенный период (например, за текущий месяц) и еще не вернули их**

CREATE OR REPLACE VIEW curr\_debtors AS

SELECT r.\*

FROM Readers r,ListBook l

WHERE r.r\_number=l.l\_reader AND (CURRENT\_DATE-l\_startdate)<31 AND (L\_enddate IS NULL)

**--Определение количества книг, взятых определенным читателем и еще не возвращенных на момент запроса**

CREATE OR REPLACE VIEW curr\_num\_books AS

SELECT COUNT(l\_book) AS "количество взятых книг", r\_number, r\_surname, r\_name, r\_fname, r\_sex, r\_born, r\_telephone, r\_mail, r\_address, r\_pasport

FROM Readers r,ListBook l

WHERE r.r\_number=l.l\_reader AND (L\_enddate IS NULL)

GROUP BY r\_number, r\_surname, r\_name, r\_fname, r\_sex, r\_born, r\_telephone, r\_mail, r\_address, r\_pasport

**--Получение числа доступных экземпляров книг**

CREATE OR REPLACE VIEW сopies\_books AS

SELECT booklist.num AS "количество доступных экземпляров",b1.\*

FROM books b1,

(SELECT c\_book, COUNT(c\_num) AS num FROM Copies c

WHERE c\_num NOT IN(SELECT l\_book FROM ListBook WHERE L\_enddate IS NULL)

GROUP BY c\_book) as booklist

WHERE b1.b\_code=booklist.c\_book;

**3.3 Назначение прав доступа**

**Пользователи:**

* Читатель – reader
* Сотрудник отдела комплектации – staff
* Сотрудник по работе с читателями – librarian

**Таблицы:**

grant select on Readers to staff

grant select, insert, update on Readers to librarian

grant select, insert, update on ListBook to librarian

grant select, insert, update on OrderBook to librarian

grant select on Category to reader

grant select, insert, update on Category to staff

grant select on Category to librarian

grant select on BookCategory to reader

grant select, insert, update on BookCategory to staff

grant select on BookCategory to librarian

grant select on Series to reader

grant select, insert, update on Series to staff

grant select on Series to librarian

grant select on Books to reader

grant select, insert, update on Books to staff

grant select on Books to librarian

grant select, insert, update on Languages to staff

grant select, insert, update on Copies to reader

grant select on Copies to librarian

grant select on Telephones

grant select on PubHouses to reader

grant select, insert, update on PubHouses to staff

grant select on PubHouses to librarian

grant select on BookAuthor to reader

grant select, insert, update on BookAuthor to staff

grant select on BookAuthor to librarian

grant select, insert, update on PubCity to staff

grant select on Countries to reader

grant select, insert, update on Countries to staff

grant select on Countries to librarian

grant select, insert, update on Cities to staff

grant select on Authors to reader

grant select, insert, update on Authors to staff

grant select on Authors to librarian

**Представления:**

grant select on my\_infomation to reader

grant select on my\_orders to reader

grant select on my\_list to reader

grant select on bcategory to reader

grant select on bseries to reader

grant select on bauthor to reader

grant select on curr\_books to reader

grant select on curr\_books to librarian

grant select on curr\_debtors to librarian

grant select on curr\_num\_book to librarian

grant select on copies\_books to librarian

### **Создание индексов**

Анализ готовых запросов показывает, что для повышения эффективности работы с данными необходимо создать индексы для всех внешних ключей.

create index ind\_book\_order on OrderBook(O\_book)

create index ind\_reader\_order on OrderBook(O\_reader)

create index ind\_book\_copy on Copies(C\_book)

create index ind\_book\_list on ListBook(L\_book)

create index ind\_reader\_list on ListBook (L\_reader)

create index ind\_order\_list on ListBook (L\_order)

create index ind\_series\_books on Books(B\_series)

create index ind\_olan\_books on Books(B\_Olanguage)

create index ind\_tlan\_books on Books(B\_Tlanguage)

create index ind\_pub\_books on Books(B\_Pubhouse)

create index ind\_category on BookCategory (BC\_category)

create index ind\_book on BookCategory (BC\_book)

create index ind\_tel\_pub on Telephones (T\_pub)

create index ind\_pub on PubCity (PC\_pub)

create index ind\_city on PubCity (PC\_city)

create index ind\_country on Cities (C\_country)

create index ind\_book\_ba on BookAuthor (BA\_book)

create index ind\_author\_ba on BookAuthor (BA\_author)

create index ind\_country\_author on Authors (A\_country)

### **Разработка стратегии резервного копирования**

Интенсивность обновления разработанной базы данных низкая, поэтому для обеспечения сохранности вполне достаточно проводить полное резервное копирование БД раз в день (перед окончанием рабочего дня). Для разработанной БД нет необходимости держать сервер включенным круглосуточно, поэтому можно создать соответствующее задание операционной системы, которое будет автоматически запускаться перед выключением сервера.