

**Учреждение профессионального образования
«Колледж Казанского инновационного университета»**

ОТЧЕТ
по учебной практике
по ПМ.11 «Разработка, администрирование и защита баз данных»
студента__ курса, _____ группы, обучающегося по специальности
09.02.07 Информационные системы и программирование

ФИО студента

ФИО руководителя практики от
колледжа _____

Подпись руководителя практики от колледжа

Набережные Челны 20__

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
1 Проектирование информационной системы.....	4
1.1 Анализ предметной области.....	4
1.2 Инфологическое проектирование.....	8
1.3 Даталогическое проектирование.....	9
1.4 Реляционная модель.....	12
1.5 Физическая модель данных.....	13
2 Создание и заполнение таблиц.....	14
2.1 SQL. В Access.....	14
2.2 SQL-Server.....	18
3 Разработка приложения для работы с базой данных.....	25
ЗАКЛЮЧЕНИЕ.....	31
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	32
Приложение 1.....	33
Приложение 2.....	43

ВВЕДЕНИЕ

Объектом исследования является Библиотека.

Основное направление деятельности – предоставление пользователям информации на материальных носителях и справочно-библиографическое обслуживание;

Цель практики – разработка, программно-аппаратная реализация основных элементов автоматизированной системы.

Задачи практики:

- проанализировать современный уровень развития автоматизированных систем для библиотек;
- сформулировать основные требования к автоматизации движения книг;
- рассмотреть современные методы и средства разработки программных продуктов;
- разработать информационно-логическую модель автоматизированной системы;
- разработать и реализовать основные элементы автоматизированной системы.

Предметная область: “Библиотека”.

Необходимо спроектировать модель данных предметной области.

Предполагается учет следующей информации:

- учет имеющегося книжного фонда;
- ведение каталога читателей;
- учет выданных и возвращенных книг.

Перечень отчетов:

- список книг указанного автора, имеющих в библиотеке на указанную дату;
- список книг, выданных указанному читателю;

- поиск книги по фрагменту названия.

1 Проектирование информационной системы

1.1 Анализ предметной области

Система должна предусматривать режимы ведения системного каталога, отражающего перечень областей знаний, по которым имеются книги в библиотеке.

Внутри библиотеки области знаний в систематическом каталоге могут иметь уникальный внутренний номер и полное наименование.

Каждая книга может содержать сведения из нескольких областей знаний.

Каждая книга в библиотеке может присутствовать в нескольких экземплярах.

Каждая книга, хранящаяся в библиотеке, характеризуется следующими параметрами:

- уникальный шифр;
- название;
- фамилии авторов (могут отсутствовать);
- место издания (город);
- издательство;
- год издания;
- количество страниц;
- стоимость книги;
- количество экземпляров книги в библиотеке.

Книги могут иметь одинаковые названия, но они различаются по своему уникальному шифру (ISBN).

В библиотеке ведется картотека читателей.

На каждого читателя в картотеку заносятся следующие сведения:

- фамилия, имя, отчество;
- домашний адрес;

- телефон (будем считать, что у нас два телефона — рабочий и домашний);

- дата рождения.

Каждому читателю присваивается уникальный номер читательского билета.

Каждый читатель может одновременно держать на руках не более 5 книг. Читатель не должен одновременно держать более одного экземпляра книги одного названия.

Каждая книга в библиотеке может присутствовать в нескольких экземплярах. Каждый экземпляр имеет следующие характеристики:

- уникальный инвентарный номер;
- шифр книги, который совпадает с уникальным шифром из описания книг;
- место размещения в библиотеке.

В случае выдачи экземпляра книги читателю в библиотеке хранится специальный вкладыш, в котором должны быть записаны следующие сведения:

- номер билета читателя, который взял книгу;
- дата выдачи книги;
- дата возврата.

Ограничения на информацию в системе:

- Книга может не иметь ни одного автора.
- В библиотеке должны быть записаны читатели не моложе 17 лет.
- В библиотеке присутствуют книги, изданные начиная с 1960 по текущий год.

- Каждый читатель может держать на руках не более 5 книг.

- Каждый читатель при регистрации в библиотеке должен дать телефон для связи: он может быть рабочим или домашним.

- Каждая область знаний может содержать ссылки на множество книг, но каждая книга может относиться к различным областям знаний.

С данной информационной системой должны работать следующие группы пользователей:

- библиотекари;
- читатели;
- администрация библиотеки.

При работе с системой библиотекарь должен иметь возможность решать следующие задачи:

- Принимать новые книги и регистрировать их в библиотеке.
- Относить книги к одной или к нескольким областям знаний.
- Проводить каталогизацию книг, то есть назначение новых инвентарных номеров вновь принятым книгам, и, помещая их на полки библиотеки, запоминать место размещения каждого экземпляра.

- Проводить дополнительную каталогизацию, если поступило несколько экземпляров книги, которая уже есть в библиотеке, при этом информация о книге в предметный каталог не вносится, а каждому новому экземпляру присваивается новый инвентарный номер и для него определяется место на полке библиотеки.

- Проводить списание старых и не пользующихся спросом книг. Списывать можно только книги, ни один экземпляр которых не находится у читателей. Списание проводится по специальному акту списания, который утверждается администрацией библиотеки.

- Вести учет выданных книг читателям, при этом предполагается два режима работы: выдача книг читателю и прием от него возвращаемых им книг обратно в библиотеку. При выдаче книг фиксируется, когда и какой экземпляр книги был выдан данному читателю и к какому сроку читатель должен вернуть этот экземпляр книги. При выдаче книг наличие свободного экземпляра и его конкретный номер могут определяться по заданному уникальному шифру книги или инвентарный номер может быть известен заранее. Не требуется вести "историю" чтения книг, то есть требуется отражать только текущее состояние библиотеки. При приеме книги, возвращаемой читателем,

проверяется соответствие возвращаемого инвентарного номера книги выданному инвентарному номеру, и она ставится на свое старое место на полку библиотеки.

- Проводить списание утерянных читателем книг по специальному акту списания или замены, подписанному администрацией библиотеки.

- Проводить закрытие абонемента читателя, то есть уничтожение данных о нем, если читатель хочет выписаться из библиотеки и не является ее должником, то есть за ним не числится ни одной библиотечной книги.

- Читатель должен иметь возможность решать следующие задачи:

- Просматривать системный каталог, то есть перечень всех областей знаний, книги по которым есть в библиотеке.

- По выбранной области знаний получить полный перечень книг, которые числятся в библиотеке.

- Для выбранной книги получить инвентарный номер свободного экземпляра книги или сообщение о том, что свободных экземпляров книги нет. В случае отсутствия свободных экземпляров книги читатель должен иметь возможность узнать дату ближайшего предполагаемого возврата экземпляра данной книги. Читатель не может узнать данные о том, у кого в настоящий момент экземпляры данной книги находятся на руках (в целях обеспечения личной безопасности держателей требуемой книги).

- Для выбранного автора получить список книг, которые числятся в библиотеке.

Администрация библиотеки должна иметь возможность:

- получать сведения о должниках - читателях библиотеки, которые не вернули вовремя взятые книги;

- сведения о книгах, которые не являются популярными, т. е. ни один экземпляр которых не находится на руках у читателей;

- сведения о стоимости конкретной книги, для того чтобы установить возможность возмещения стоимости утерянной книги или возможность замены ее другой книгой;

- сведения о наиболее популярных книгах, то есть таких, все экземпляры которых находятся на руках у читателей.



Рисунок 1 Организационная схема Библиотеки

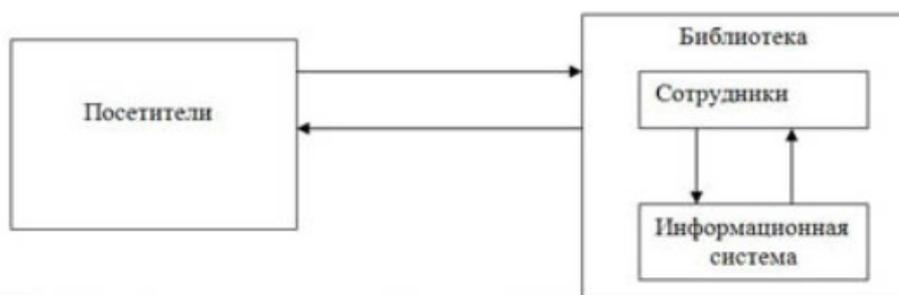


Рисунок 2 Состав и структура системы

1.2 Инфологическое проектирование

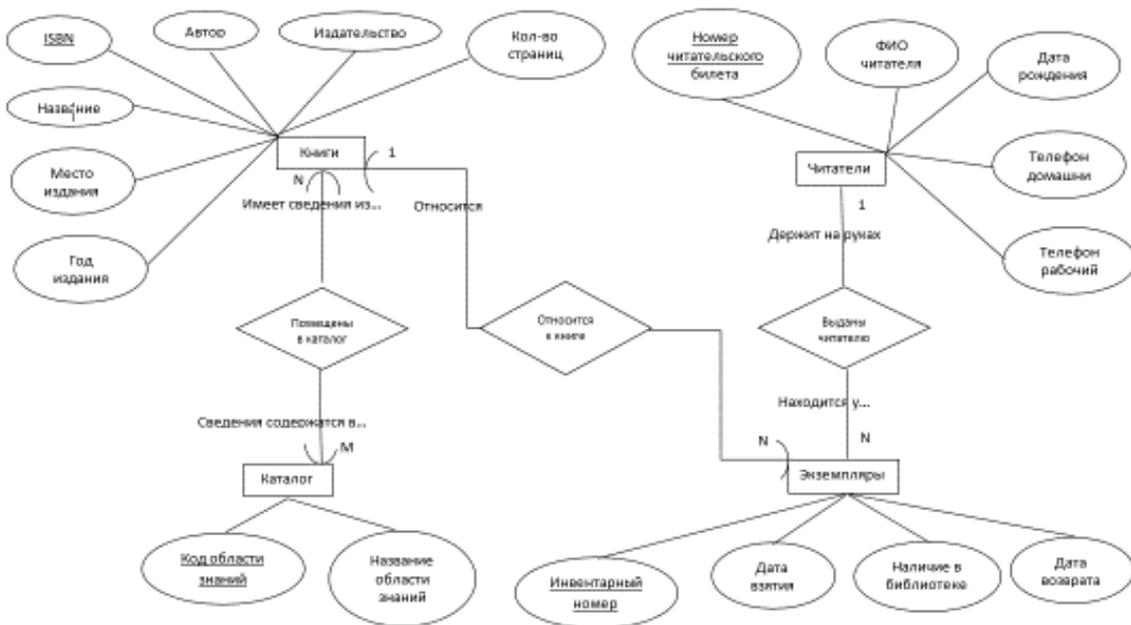


Рисунок 3 ER-модель предметной области «Библиотека» в нотации Чена

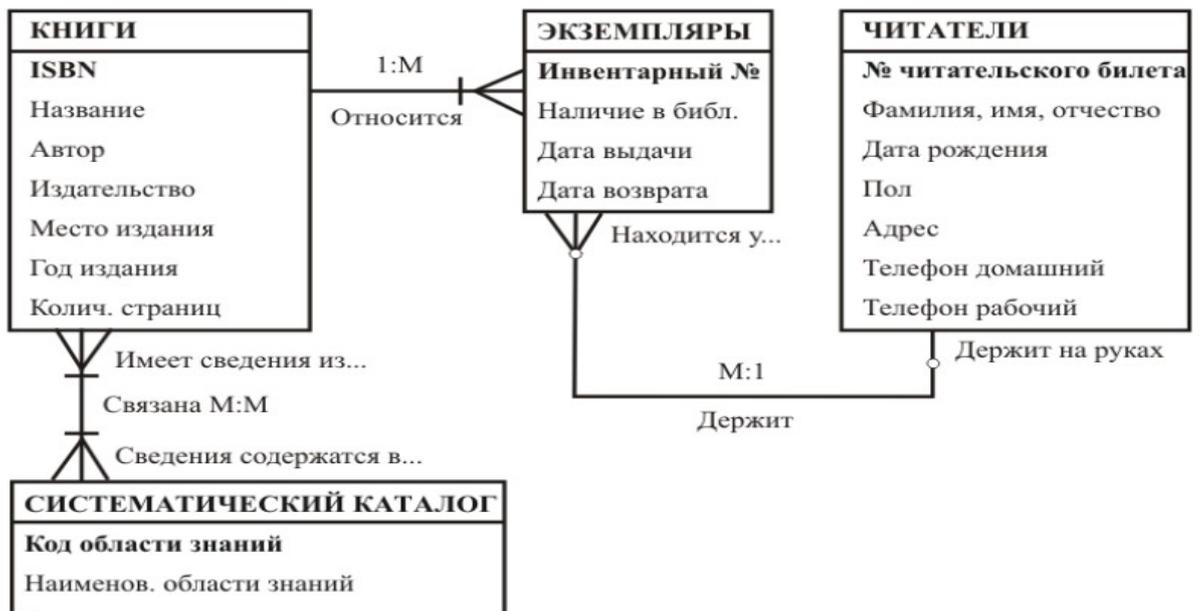
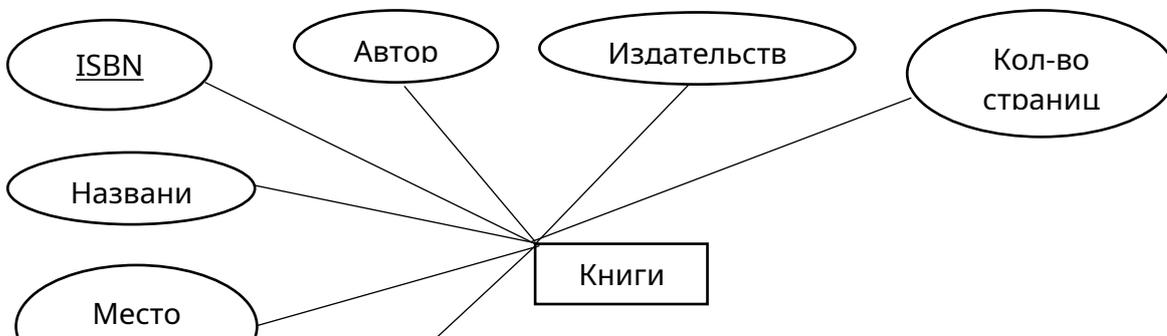


Рисунок 4 ER-модель предметной области «Библиотека» в нотации «Вороньи лапки»

1.3 Дatalogическое проектирование



Книги	
PK	<u>ISBN</u>
	Автор Название Издательство Место издания Год издания Кол-во страниц

Рисунок 5 Таблица Книги

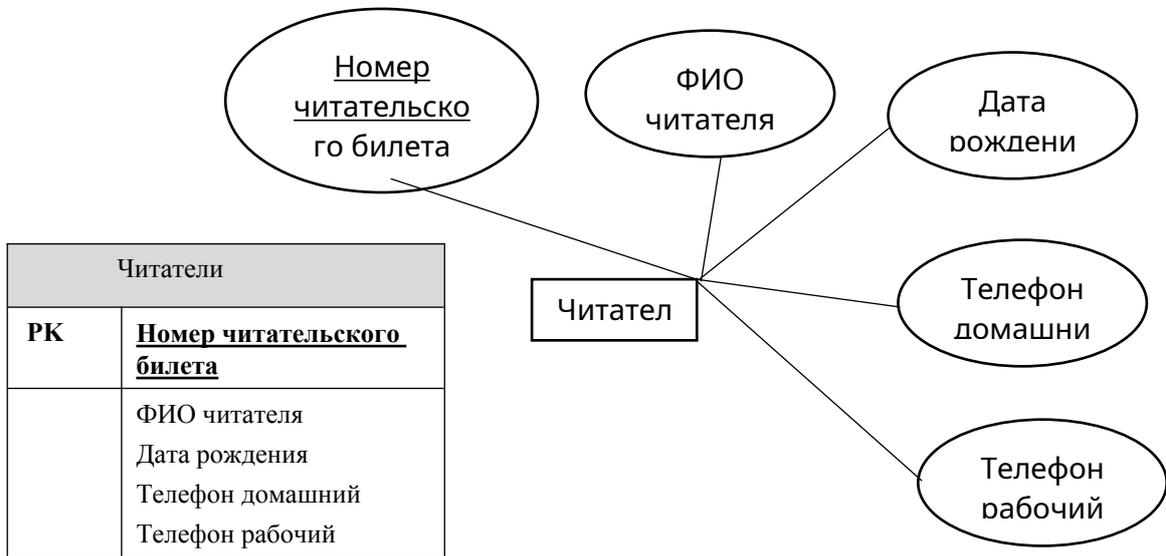


Рисунок 6 Таблица Читатели

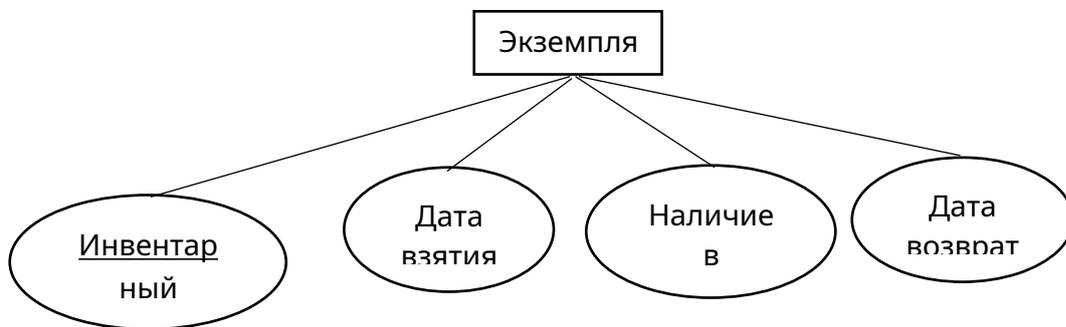


Рисунок 7 Таблица Экземпляр

Экземпляры	
PK	<u>Инвентарный номер</u>
	Дата взятия Наличие в библиотеке Дата возврата
FK	<u>Номер читательского билета</u>
FK	<u>ISBN</u>

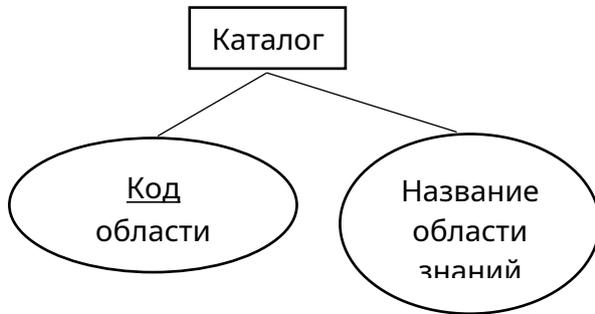


Рисунок 8 Таблица Каталог

Каталог	
PK	<u>Код области знаний</u>
	Название области знаний

СВЯЗЬ N:M

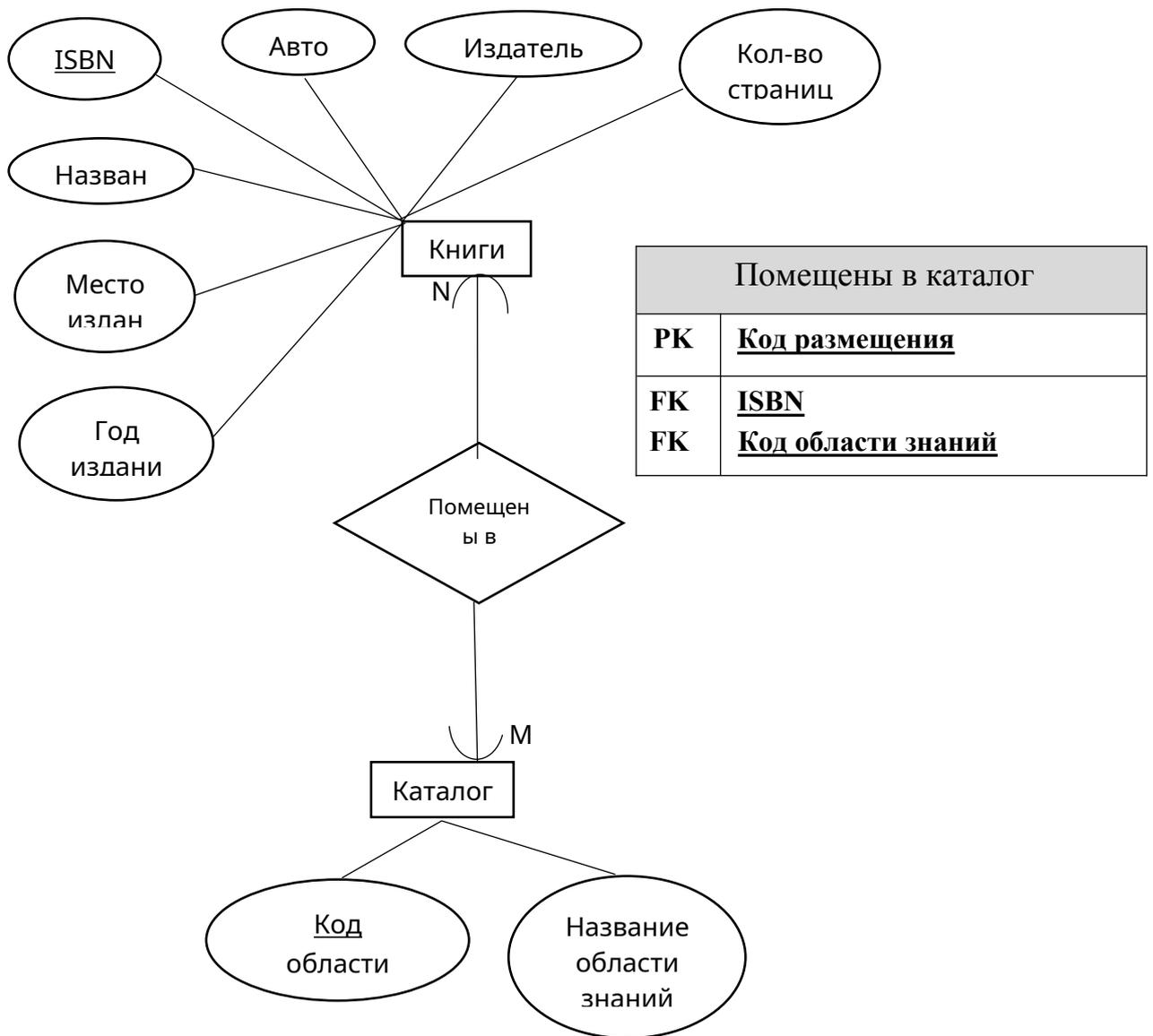


Рисунок 9 Преобразование связи M к N

1.4 Реляционная модель

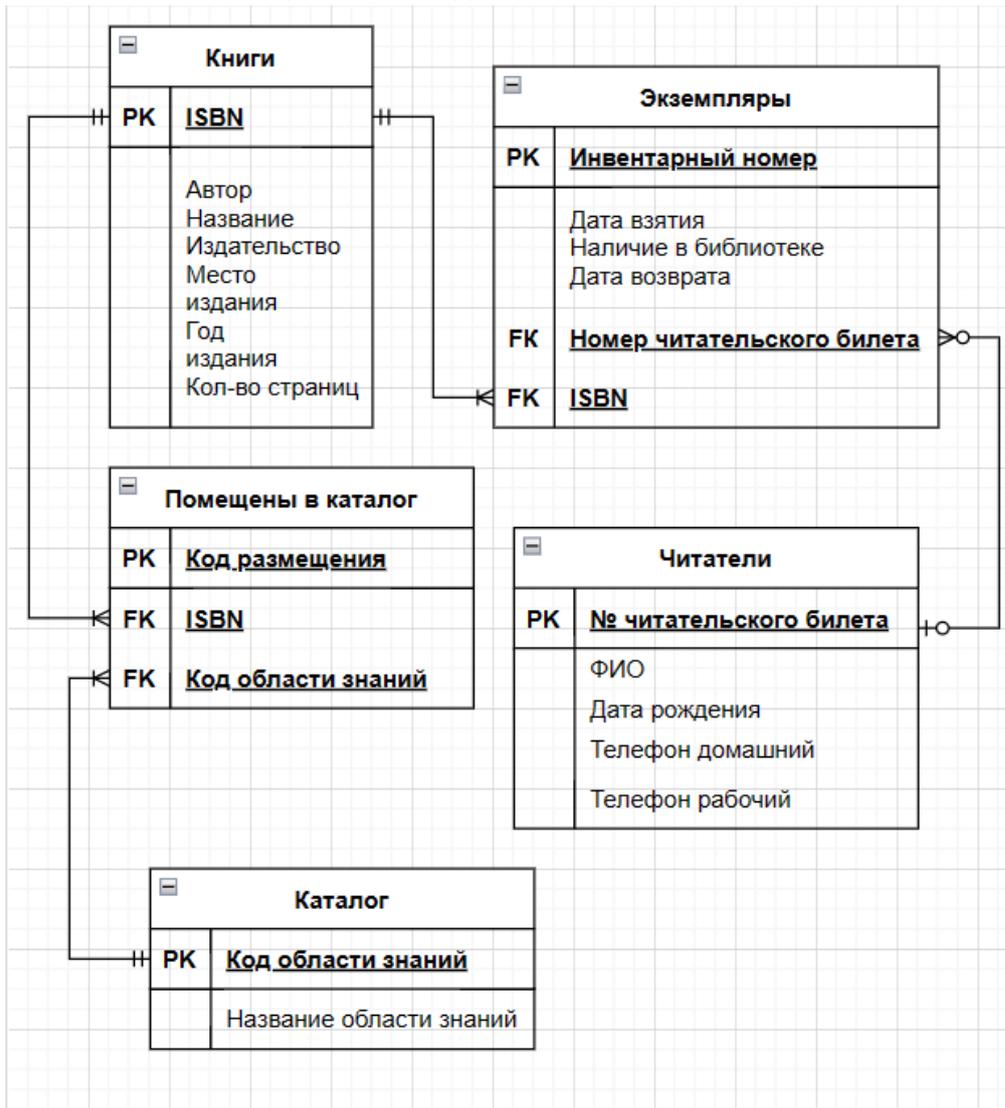


Рисунок 10 Реляционная модель

1.5 Физическая модель данных

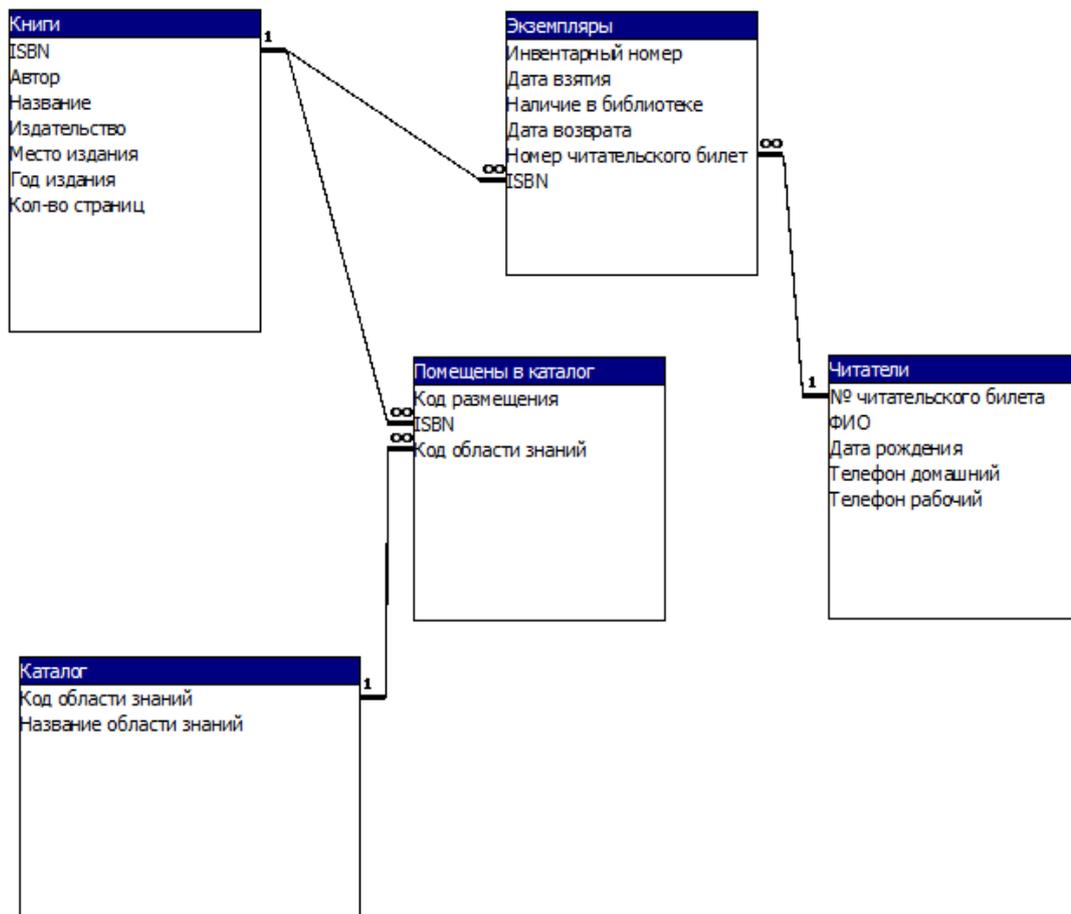


Рисунок 11 Физическая модель

2 Создание и заполнение таблиц.

2.1 SQL. В Access.

1. Создание таблицы «Книги».

Номер	Имя	Домен(тип)	NULL	Примечание
1	ISBN	INTEGER	Запрещено	Первичный ключ
2	Автор	TEXT(30)	Запрещено	
3	Название	TEXT(40)	Запрещено	Альтернативный ключ
4	Издательство	TEXT(30)	Запрещено	-
5	Место издания	TEXT(30)	Запрещено	-
6	Год издания	DATETIME	Запрещено	-
7	Кол-во страниц	SMALLINT	Запрещено	-

CREATE TABLE КНИГИ

(ISBN INTEGER NOT NULL, Автор TEXT (30) NOT NULL, Название TEXT (40) NOT NULL, Издательство TEXT (30) NOT NULL, Место_издания TEXT (30) NOT NULL, Год_издания DATETIME NOT NULL, Кол_во_страниц SMALLINT NOT NULL, CONSTRAINT ПК_КНИГИ PRIMARY KEY (ISBN), CONSTRAINT АК_КНИГИ UNIQUE (Название));

2. Создание таблицы «Каталог».

Номер	Имя	Домен(тип)	NULL	Примечание
1	Код области знаний	SMALLINT	Запрещено	Первичный ключ
2	Название области знаний	TEXT(40)	Запрещено	-

CREATE TABLE КАТАЛОГ

(Код_области_знаний SMALLINT NOT NULL, Название_области_знаний TEXT(40) NOT NULL, CONSTRAINT ПК_КАТАЛОГ PRIMARY KEY (Код_области_знаний));

3. Создание таблицы «Читатели».

Номер	Имя	Домен(тип)	NULL	Примечание
1	Номер читательского билета	INTEGER	Запрещено	Первичный ключ
2	ФИО	TEXT(40)	Запрещено	-
3	Дата рождения	DATETIME	Запрещено	-
4	Телефон домашний	TEXT(12)	-	-
5	Телефон рабочий	TEXT(12)	-	-

CREATE TABLE ЧИТАТЕЛИ

(Номер_читательского_билета INTEGER NOT NULL, ФИО TEXT(40) NOT NULL, Дата_рождения DATETIME NOT NULL, Телефон_домашний TEXT(12), Телефон_рабочий TEXT(12), CONSTRAINT ПК_ЧИТАТЕЛИ PRIMARY KEY (Номер_читательского_билета));

4. Создание таблицы «Экземпляры».

Номер	Имя	Домен(тип)	NULL	Примечание
1	Инвентарный номер	INTEGER	Запрещено	Первичный ключ
2	Дата взятия	DATETIME	-	-
3	Наличие в библиотеке	BIT	Запрещено	-
4	Дата возврата	DATETIME	-	-
5	Номер читательского билета	INTEGER	-	Внешний ключ
6	ISBN	INTEGER	Запрещено	Внешний ключ

```
CREATE TABLE ЭКЗЕМПЛЯРЫ
(Инвентарный_номер INTEGER NOT NULL,
Дата_взятия DATETIME,
Наличие_в_библиотеке BIT NOT NULL,
Дата_возврата DATETIME,
Номер_читательского_билета INTEGER,
ISBN INTEGER NOT NULL,
CONSTRAINT ПК_ЭКЗЕМПЛЯРЫ PRIMARY KEY (Инвентарный_номер),
CONSTRAINT ЭКЗЕМПЛЯРЫ_ЧИТАТЕЛИ FOREIGN KEY (Номер_читательского_билета) REFERENCES
ЧИТАТЕЛИ (Номер_читательского_билета) ON UPDATE CASCADE ON DELETE SET NULL,
CONSTRAINT ЭКЗЕМПЛЯРЫ_КНИГИ FOREIGN KEY (ISBN) REFERENCES КНИГИ (ISBN) ON UPDATE
CASCADE ON DELETE SET NULL);
```

5. Создание таблицы «Помещены в каталог».

Номер	Имя	Домен(тип)	NULL	Примечание
1	Код размещения	INTEGER	Запрещено	Первичный ключ
2	ISBN	INTEGER	Запрещено	Внешний ключ
3	Код области знаний	SMALLINT	Запрещено	Внешний ключ

```
CREATE TABLE ПОМЕЩЕНЫ_В_КАТАЛОГ
(Код_размещения INTEGER NOT NULL,
ISBN INTEGER NOT NULL,
Код_области_знаний SMALLINT NOT NULL,
CONSTRAINT ПК_Помещены_в_каталог PRIMARY KEY (Код_размещения),
CONSTRAINT ПОМЕЩЕНЫ_В_КАТАЛОГ_КНИГИ FOREIGN KEY (ISBN) REFERENCES КНИГИ (ISBN)
ON UPDATE CASCADE ON DELETE SET NULL,
CONSTRAINT ПОМЕЩЕНЫ_В_КАТАЛОГ_КАТАЛОГ FOREIGN KEY (Код_области_знаний) REFERENCES
Каталог (Код_области_знаний) ON UPDATE CASCADE ON DELETE SET NULL);
```

6. Заполнение таблицы «Книги».

```
INSERT INTO КНИГИ (ISBN, Автор, Название, Издательство, Место_издания, Год_издания, Кол_во_страниц)
VALUES (812302022, 'Моряков В. И.', 'История России', 'СЛОВО', 'Ростов н/Д', 2004, 447);
```

```
INSERT INTO КНИГИ (ISBN, Автор, Название, Издательство, Место_издания, Год_издания, Кол_во_страниц)
VALUES (812675676, 'Каштанов Ю. И.', 'Русский костюм', 'Белый город', 'Москва', 2009, 48);
```

INSERT INTO КНИГИ (ISBN, Автор, Название, Издательство, Место_издания, Год_издания, Кол_во_страниц)
VALUES (811115676, 'Аракин В. Д.', 'Словарь', 'Советская энциклопедия', 'Москва', 1969, 590);

INSERT INTO КНИГИ (ISBN, Автор, Название, Издательство, Место_издания, Год_издания, Кол_во_страниц)
VALUES (811415676, 'Hudson K.', 'Museums of influence', 'Советская энциклопедия', 'Cambridge', 1987, 190);

INSERT INTO КНИГИ (ISBN, Автор, Название, Издательство, Место_издания, Год_издания, Кол_во_страниц)
VALUES (811415076, 'Иванов, М. Н.', 'Детали машин', 'Высшая школа', 'Москва', 2008, 407);

ISBN	Автор	Название	Издательство	Место_изда	Год_издани	Кол_во_стр:
811115676	Аракин В. Д.	Словарь	Советская энциклопедия	Москва	22.05.1905	590
811415076	Иванов, М. Н.	Детали машин	Высшая школа	Москва	30.06.1905	407
811415676	Hudson K.	Museums of influence	Советская энциклопедия	Cambridge	09.06.1905	190
812302022	Моряков В. И.	История России	СЛОВО	Ростов н/Д	26.06.1905	447
812675676	Каштанов Ю.	Русский костюм	Белый город	Москва	01.07.1905	48

Рисунок 12 Таблица Книги

7. Заполнение таблицы «Каталог».

INSERT INTO КАТАЛОГ (Код_области_знаний, Название_области_знаний)
VALUES (1, 'Энциклопедии и словари');

INSERT INTO КАТАЛОГ (Код_области_знаний, Название_области_знаний)
VALUES (2, 'История и культура');

INSERT INTO КАТАЛОГ (Код_области_знаний, Название_области_знаний)
VALUES (3, 'Техника');

INSERT INTO КАТАЛОГ (Код_области_знаний, Название_области_знаний)
VALUES (4, 'Физическая культура и спорт');

INSERT INTO КАТАЛОГ (Код_области_знаний, Название_области_знаний)
VALUES (5, 'Наука');

Код_област	Название_области_знаний
1	Энциклопедии и словари
2	История и культура
3	Техника
4	Физическая культура и спорт
5	Наука

Рисунок 13 Таблица Каталог

8. Заполнение таблицы «Читатели».

INSERT INTO ЧИТАТЕЛИ (Номер_читательского_билета, ФИО, Дата_рождения, Телефон_домашний, Телефон_рабочий)
VALUES (1, 'Иванов И.И.', '12.12.2001', '89185674534', '454545');

```
INSERT INTO ЧИТАТЕЛИ (Номер_читательского_билета, ФИО, Дата_рождения, Телефон_домашний,
Телефон_рабочий)
VALUES (2, 'Петров П.П.', '01.02.2000', '89185671111', '565656');
```

```
INSERT INTO ЧИТАТЕЛИ (Номер_читательского_билета, ФИО, Дата_рождения, Телефон_домашний,
Телефон_рабочий)
VALUES (3, 'Сидоров С.С.', '11.03.2005', '86767610011', '909090');
```

```
INSERT INTO ЧИТАТЕЛИ (Номер_читательского_билета, ФИО, Дата_рождения, Телефон_домашний,
Телефон_рабочий)
VALUES (4, 'Кисилев К.К.', '22.08.2004', '86767614444', '666666');
```

```
INSERT INTO ЧИТАТЕЛИ (Номер_читательского_билета, ФИО, Дата_рождения, Телефон_домашний,
Телефон_рабочий)
VALUES (5, 'Осипов О.О.', '12.09.2005', '86767610000', '778899');
```

Номер_чита	ФИО	Дата_рожде	Телефон_дс	Телефон_ра	Щ
1	Иванов И.И.	12.12.2001	89185674534	454545	
2	Петров П.П.	01.02.2000	89185671111	565656	
3	Сидоров С.С.	11.03.2005	86767610011	909090	
4	Кисилев К.К.	22.08.2004	86767614444	666666	
5	Осипов О.О.	12.09.2005	86767610000	778899	

Рисунок 14 Таблица Читатели

9. Заполнение таблицы «Экземпляры».

```
INSERT INTO ЭКЗЕМПЛЯРЫ(Инвентарный_номер, Дата_взятия, Наличие_в_библиотеке, Дата_возврата,
Номер_читательского_билета, ISBN)
VALUES (1, '10.02.2023', No, '18.02.2023', 1, 811115676)
```

```
INSERT INTO ЭКЗЕМПЛЯРЫ(Инвентарный_номер, Дата_взятия, Наличие_в_библиотеке, Дата_возврата,
Номер_читательского_билета, ISBN)
VALUES (2, '11.02.2023', No, '17.02.2023', 2, 811415076)
```

```
INSERT INTO ЭКЗЕМПЛЯРЫ(Инвентарный_номер, Дата_взятия, Наличие_в_библиотеке, Дата_возврата,
Номер_читательского_билета, ISBN)
VALUES (3, '12.02.2023', YES, '17.02.2023', 3, 811115676)
```

```
INSERT INTO ЭКЗЕМПЛЯРЫ(Инвентарный_номер, Дата_взятия, Наличие_в_библиотеке, Дата_возврата,
Номер_читательского_билета, ISBN)
VALUES (4, '11.02.2023', YES, '27.02.2023', 3, 811415676)
```

```
INSERT INTO ЭКЗЕМПЛЯРЫ(Инвентарный_номер, Дата_взятия, Наличие_в_библиотеке, Дата_возврата,
Номер_читательского_билета, ISBN)
VALUES (5, '11.02.2023', YES, '27.02.2023', 2, 811415676)
```

Инвентарнь	Дата_взятия	Наличие_в_	Дата_возвр.	Номер_чита	ISBN	Ц
1	10.02.2023	0	18.02.2023	1	811115676	
2	11.02.2023	0	17.02.2023	2	811415076	
3	12.02.2023	-1	17.02.2023	3	811115676	
4	11.02.2023	-1	27.02.2023	3	811415676	
5	11.02.2023	-1	27.02.2023	2	811415676	

Рисунок 15 Таблица Экземпляры

10. Заполнение таблице «Помещены в каталог».

```
INSERT INTO ПОМЕЩЕНЫ_В_КАТАЛОГ(Код_размещения, ISBN, Код_области_знаний)
VALUES (1,811115676,1)
```

```
INSERT INTO ПОМЕЩЕНЫ_В_КАТАЛОГ(Код_размещения, ISBN, Код_области_знаний)
VALUES (2,811415076,3)
```

```
INSERT INTO ПОМЕЩЕНЫ_В_КАТАЛОГ(Код_размещения, ISBN, Код_области_знаний)
VALUES (3,811415676,1)
```

```
INSERT INTO ПОМЕЩЕНЫ_В_КАТАЛОГ(Код_размещения, ISBN, Код_области_знаний)
VALUES (4,812302022,2)
```

```
INSERT INTO ПОМЕЩЕНЫ_В_КАТАЛОГ(Код_размещения, ISBN, Код_области_знаний)
VALUES (5,812675676,2)
```

Код_размеш	ISBN	Код_област	Ц
1	811115676		1
2	811415076		3
3	811415676		1
4	812302022		2
5	812675676		2

Рисунок 16 Таблица Помещены в каталог

2.2 SQL-Server.

1. Создание БД «Библиотека».

```
CREATE DATABASE Library;
```

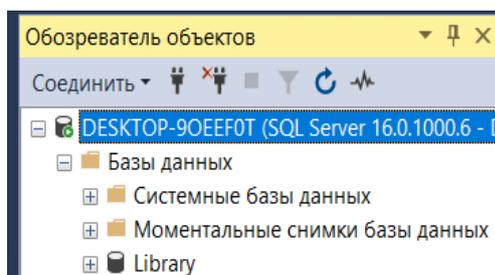


Рисунок 17 Обозреватель объектов

2. Создание таблицы «Книги».

```
use Library;  
CREATE TABLE Books  
(  
    ISBN INT NOT NULL,  
    Author NVARCHAR(30) NOT NULL,  
    Title NVARCHAR (40) NOT NULL,  
    Publishing_house NVARCHAR (30) NOT NULL,  
    City_of_publication NVARCHAR (30) NOT NULL,  
    Year INT NOT NULL,  
    Pages INT NOT NULL,  
    CONSTRAINT PK_Books PRIMARY KEY (ISBN)  
);
```

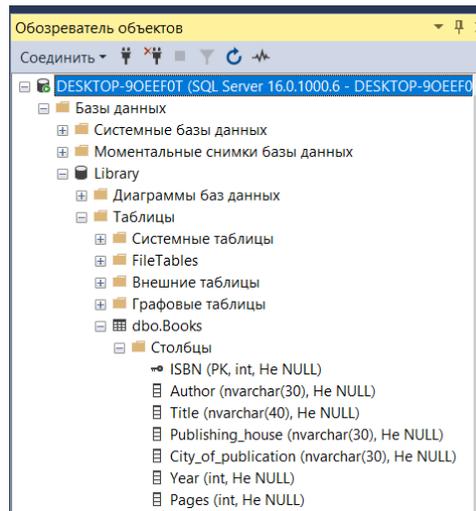


Рисунок 18 Поля таблицы Books

3. Создание таблицы «Каталог».

```
use Library;  
CREATE TABLE Catalog  
(  
    Knowledge_area_code INT NOT NULL,  
    Knowledge_area_name NVARCHAR (40) NOT NULL,  
    CONSTRAINT PK_Catalog PRIMARY KEY (Knowledge_area_code)  
);
```

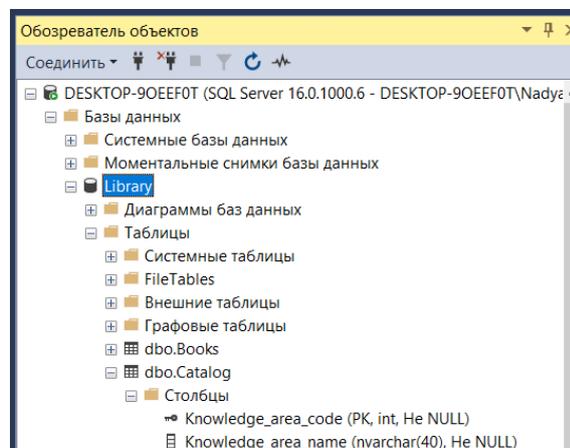


Рисунок 19 Поля таблицы Catalog

4. Создание таблицы «Читатели».

```

use Library;
CREATE TABLE Readers
(
    Library_card_number INT NOT NULL,
    Full_name NVARCHAR (40) NOT NULL,
    Date_of_Birth DATE NOT NULL,
    Home_phone NVARCHAR (12),
    Phone_working NVARCHAR (12),
    CONSTRAINT PK_Readers PRIMARY KEY (Library_card_number)
);

```

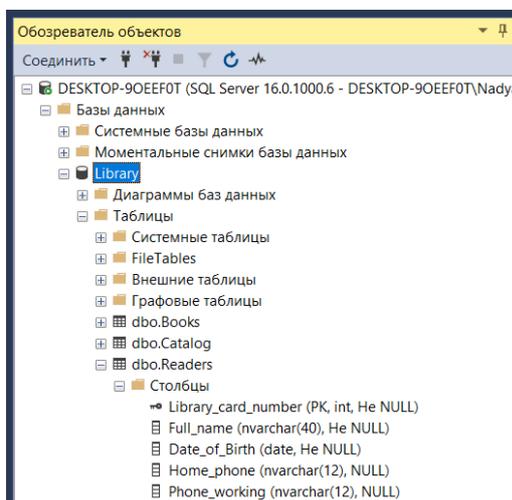


Рисунок 20 Поля таблицы Readers

5. Создание таблицы «Экземпляры».

```

use Library;
CREATE TABLE Instances
(
    Inventory_number INT NOT NULL,
    Date_of_taking DATE,
    Availability_in_the_library BIT NOT NULL,
    Return_date DATE,
    Library_card_number INT,
    ISBN INT,
    CONSTRAINT PK_Instances PRIMARY KEY (Inventory_number),
    CONSTRAINT Instances_Readers FOREIGN KEY (Library_card_number) REFERENCES Readers
(Library_card_number) ON UPDATE CASCADE ON DELETE SET NULL,
    CONSTRAINT Instances_Books FOREIGN KEY (ISBN) REFERENCES Books (ISBN) ON UPDATE CASCADE
ON DELETE SET NULL
);

```

6. Создание таблицы «Помещены в каталог».

```

use Library;
CREATE TABLE Placed_in_the_catalog
(
    Placement_code INT NOT NULL,
    ISBN INTEGER,
    Knowledge_area_code INT,
    CONSTRAINT PK_Placed_in_the_catalog PRIMARY KEY (Placement_code),
    CONSTRAINT Placed_in_the_catalog_Books FOREIGN KEY (ISBN) REFERENCES Books (ISBN) ON
UPDATE CASCADE ON DELETE SET NULL,
    CONSTRAINT Placed_in_the_catalog_Catalog FOREIGN KEY (Knowledge_area_code) REFERENCES Catalog
(Knowledge_area_code) ON UPDATE CASCADE ON DELETE SET NULL
);

```

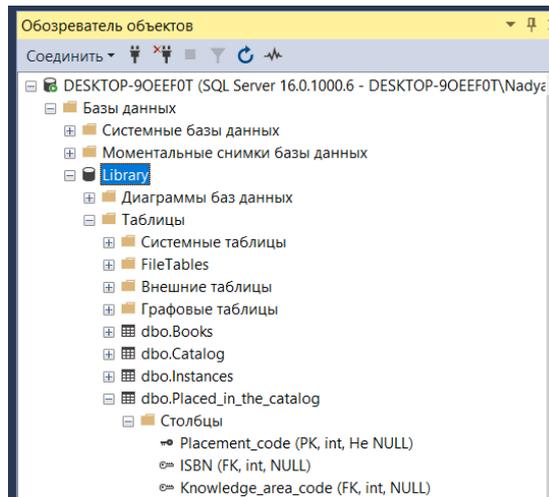


Рисунок 21 Поля таблицы Помещены в каталог

7. Создание Диаграммы БД

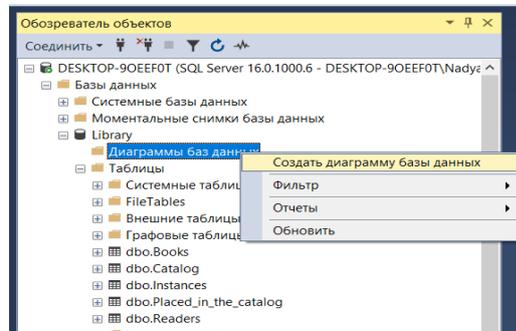


Рисунок 22 Создание диаграммы

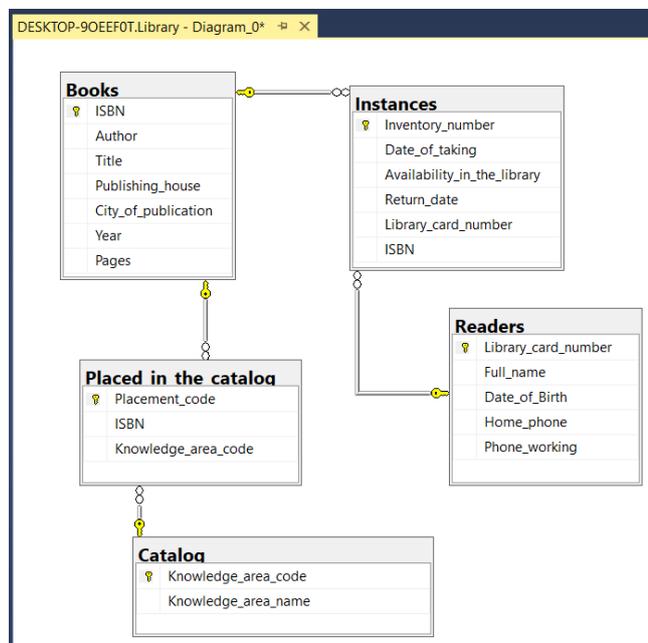


Рисунок 23 Диаграмма

8. Заполнение таблицы «Книги».

use Library;

INSERT INTO Books (ISBN, Author, Title, Publishing_house, City_of_publication, Year, Pages)

```
VALUES (812302022, 'Моряков В. И.', 'История России', 'СЛОВО', 'Ростов н/Д', 2004, 447);
```

```
INSERT INTO Books (ISBN, Author, Title, Publishing_house, City_of_publication, Year, Pages)  
VALUES (812675676, 'Каштанов Ю. И.', 'Русский костюм', 'Белый город', 'Москва', 2009, 48);
```

```
INSERT INTO Books (ISBN, Author, Title, Publishing_house, City_of_publication, Year, Pages)  
VALUES (811115676, 'Аракин В. Д.', 'Словарь', 'Советская энциклопедия', 'Москва', 1969, 590);
```

```
INSERT INTO Books (ISBN, Author, Title, Publishing_house, City_of_publication, Year, Pages)  
VALUES (811415676, 'Hudson K.', 'Museums of influence', 'Советская энциклопедия', 'Cambridge', 1987, 190);
```

```
INSERT INTO Books (ISBN, Author, Title, Publishing_house, City_of_publication, Year, Pages)  
VALUES (811415076, 'Иванов, М. Н.', 'Детали машин', 'Высшая школа', 'Москва', 2008, 407);
```

ISBN	Author	Title	Publishing_...	City_of_pu...	Year	Pages
811115676	Аракин В. Д.	Словарь	Советская э...	Москва	1969	590
811415076	Иванов, М. ...	Детали ма...	Высшая шк...	Москва	2008	407
811415676	Hudson K.	Museums of...	Советская э...	Cambridge	1987	190
812302022	Моряков В...	История Ро...	СЛОВО	Ростов н/Д	2004	447
812675676	Каштанов ...	Русский кос...	Белый город	Москва	2009	48
* NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 24 Заполнение таблицы Книги

9. Заполнение таблицы «Каталог».

```
use Library;
```

```
INSERT INTO Catalog (Knowledge_area_code, Knowledge_area_name) VALUES (1, 'Энциклопедии и словари');
```

```
INSERT INTO Catalog (Knowledge_area_code, Knowledge_area_name) VALUES (2, 'История и культура');
```

```
INSERT INTO Catalog (Knowledge_area_code, Knowledge_area_name) VALUES (3, 'Техника');
```

```
INSERT INTO Catalog (Knowledge_area_code, Knowledge_area_name) VALUES (4, 'Физическая культура и спорт');
```

```
INSERT INTO Catalog (Knowledge_area_code, Knowledge_area_name) VALUES (5, 'Наука');
```

Knowledge...	Knowledge...
1	Энциклопе...
2	История и ...
3	Техника
4	Физическа...
5	Наука
* NULL	NULL

Рисунок 25 Заполнение таблицы Каталог

10. Заполнение таблицы «Читатели».

```
use Library;
```

```
INSERT INTO Readers (Library_card_number, Full_name, Date_of_Birth, Home_phone, Phone_working) VALUES (1, 'Иванов И.И.', '12.12.2001', '89185674534', '454545');
```

```
INSERT INTO Readers (Library_card_number, Full_name, Date_of_Birth, Home_phone, Phone_working) VALUES (2, 'Петров П.П.', '01.02.2000', '89185671111', '565656');
```

```
INSERT INTO Readers (Library_card_number, Full_name, Date_of_Birth, Home_phone, Phone_working) VALUES (3, 'Сидоров С.С.', '11.03.2005', '86767610011', '909090');
```

```
INSERT INTO Readers (Library_card_number, Full_name, Date_of_Birth, Home_phone, Phone_working) VALUES (4, 'Кисилев К.К.', '22.08.2004', '86767614444', '666666');
```

```
INSERT INTO Readers (Library_card_number, Full_name, Date_of_Birth, Home_phone, Phone_working) VALUES (5, 'Осипов О.О.', '12.09.2005', '86767610000', '778899');
```

	Library_car...	Full_name	Date_of_Bir...	Home_pho...	Phone_wo
▶	1	Иванов И.И.	2001-12-12	89185674534	454545
	2	Петров П.П.	2000-02-01	89185671111	565656
	3	Сидоров С.С.	2005-03-11	86767610011	909090
	4	Кисилев К.К.	2004-08-22	86767614444	666666
	5	Осипов О.О.	2005-09-12	86767610000	778899
*	NULL	NULL	NULL	NULL	NULL

Рисунок 26 Заполнение таблицы Читатели

11. Заполнение таблицы «Экземпляры».

```
use Library;
INSERT INTO Instances (Inventory_number, Date_of_taking, Availability_in_the_library, Return_date, Library_card_number, ISBN)
VALUES (1, '10.02.2023', 0, '18.02.2023', 1, 811115676)
```

```
INSERT INTO Instances (Inventory_number, Date_of_taking, Availability_in_the_library, Return_date, Library_card_number, ISBN)
VALUES (2, '11.02.2023', 0, '17.02.2023', 2, 811415076)
```

```
INSERT INTO Instances (Inventory_number, Date_of_taking, Availability_in_the_library, Return_date, Library_card_number, ISBN)
VALUES (3, '12.02.2023', 1, '17.02.2023', 3, 811115676)
```

```
INSERT INTO Instances (Inventory_number, Date_of_taking, Availability_in_the_library, Return_date, Library_card_number, ISBN)
VALUES (4, '11.02.2023', 1, '27.02.2023', 3, 811415676)
```

```
INSERT INTO Instances (Inventory_number, Date_of_taking, Availability_in_the_library, Return_date, Library_card_number, ISBN)
VALUES (5, '11.02.2023', 1, '27.02.2023', 2, 811415676)
```

	Inventory_n...	Date_of_ta...	Availability...	Return_date	Library_car...	ISBN
▶	1	2023-02-10	False	2023-02-18	1	811115676
	2	2023-02-11	False	2023-02-17	2	811415076
	3	2023-02-12	True	2023-02-17	3	811115676
	4	2023-02-11	True	2023-02-27	3	811415676
	5	2023-02-11	True	2023-02-27	2	811415676
*	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 27 Заполнение таблицы Экземпляры

12. Заполнение таблице «Помещены в каталог».

```
use Library;
INSERT INTO Placed_in_the_catalog (Placement_code, ISBN, Knowledge_area_code)
VALUES (1, 811115676, 1)
```

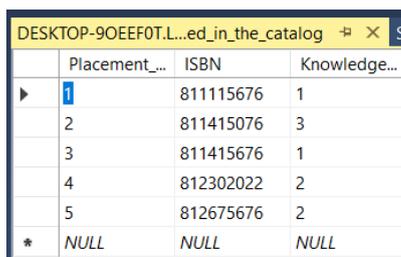
```
INSERT INTO Placed_in_the_catalog (Placement_code, ISBN, Knowledge_area_code)
VALUES (2, 811415076, 3)
```

```
INSERT INTO Placed_in_the_catalog (Placement_code, ISBN, Knowledge_area_code)
VALUES (3, 811415676, 1)
```

```
INSERT INTO Placed_in_the_catalog (Placement_code, ISBN, Knowledge_area_code)
VALUES (4, 812302022, 2)
```

```
INSERT INTO Placed_in_the_catalog (Placement_code, ISBN, Knowledge_area_code)
```

VALUES (5,812675676,2)



	Placement_...	ISBN	Knowledge...
▶ 1		811115676	1
2		811415076	3
3		811415676	1
4		812302022	2
5		812675676	2
*	NULL	NULL	NULL

Рисунок 28 Заполнение таблицы Помещены в каталог

13. Создание запросов

Вывести список книг указанного автора, имеющих в библиотеке на указанную дату (на сегодня).

```
use Library;  
SELECT * FROM Books INNER JOIN Instances  
ON Books.ISBN=Instances.ISBN  
WHERE Author = 'Аракин В. Д.' AND Availability_in_the_library = 1
```

Вывести список книг, выданных указанному читателю;

```
use Library;  
SELECT * FROM Readers INNER JOIN Instances  
ON Readers.Library_card_number=Instances.Library_card_number  
WHERE Full_name = 'Сидоров С.С.'
```

Поиск книги по фрагменту названия.

```
use Library;  
SELECT * FROM Books  
WHERE Title Like '%лов%'
```

3 Разработка приложения для работы с базой данных

Для работы с базой данных в Access было создано приложение на C#. Структуру форм и код программы можно увидеть в Приложении 1.

Также было создано приложение на C# для работы с базой данных SQL-Server. Структуру форм и код программы можно увидеть в Приложении 2.

Для организации связи между C# и SQL-Server используется технология Change Tracking.

Обычно приложение имеет постоянное соединение с базой данных, но не всегда это возможно. Примерами таких ситуаций могут быть ненадежное соединение, периодически подключаемые пользователи или мобильные устройства. Обычно в такой ситуации приложение имеет легкий, но функциональный локальный кэш и периодическую одно или двухстороннюю синхронизацию с основной базой данных. Одной из самых главных проблем для разработчиков до выхода SQL Server 2008 была необходимость определить, какие данные были изменены в базе данных, чтобы произвести синхронизацию. Требовалось проектировать и реализовывать свое индивидуальное решение, что обычно означало использование триггеров на отслеживаемых таблицах, которые записывали бы данные об изменениях. Необходимо было написать код, который на основании этих данных отображал изменения, а также не забыть о периодической очистке старых ненужных данных. Спроектировать такое решение простым и в то же время не оказывающим тяжелого влияния на текущую нагрузку в базе данных это довольно тяжелая задача.

С появлением новой технологии Change Tracking решать такие задачи становится гораздо легче. Как только мы включаем Change Tracking на таблице он начинает отслеживать все DML операции и первичные ключи строк, которые они затронули. Мы можем определить, какая операция была произведена над строкой: вставка, изменение или удаление, а используя соединение с исходной таблицей по первичному ключу – получить актуальные данные в остальных столбцах.

Алгоритм для создания приложения с применением технологии Change Tracking:

1. Запускаем Visual Studio и создаем новый проект Windows Forms.
2. В меню Project -> Add New Item выбираем Local Database Cache и добавляем его в текущий проект.
3. В проект будет добавлен файл .sync и откроется окно Configure Data Synchronization.
4. Для установки Server connection нажимаем кнопку New и создаем подключение с нашим локальным SQL Server.

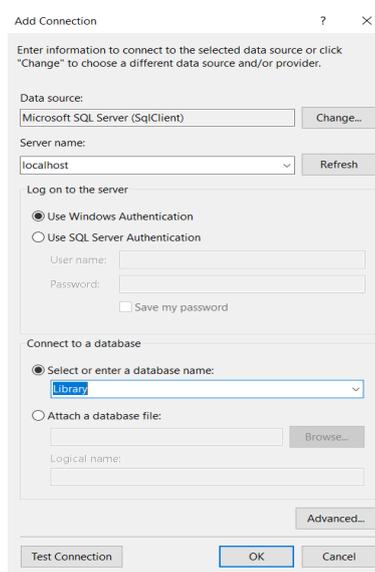


Рисунок 29 Создание подключения

5. Нажимаем Ok.
6. Опцию Use SQL Server change tracking оставляем включенной по умолчанию.
7. В раздел Cached Tables с помощью кнопки Add добавляем таблицы, для которых мы хотим настроить локальное кэширование. Появляется окно Configure Tables for Offline Use. Значение параметра Data to download оставляем по умолчанию New and incremental changes after first synchronization. Наше приложение будет извлекать из сервера записи, которые были изменены с момента последней синхронизации. Во время первой синхронизации будет загружена вся таблица. Если выбрать значение Entire table each time, то при

синхронизации локальная копия таблицы будет заменяться ее версией с сервера баз данных.

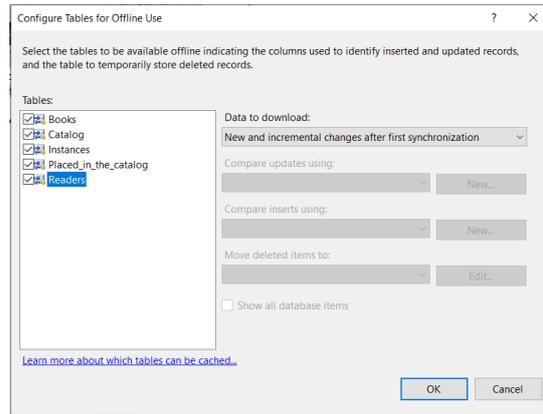


Рисунок 30 Выбор таблиц

8. Нажимаем Ок.

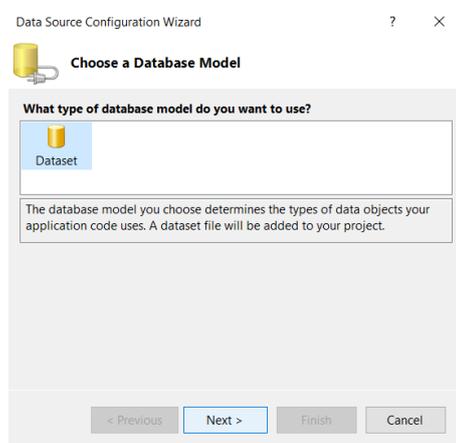


Рисунок 31 Настройка подключения БД

9. Нажимаем Next. Ставим галочки на названиях нужных таблиц. Запоминаем имя для DataSet. Нажимаем Finish.

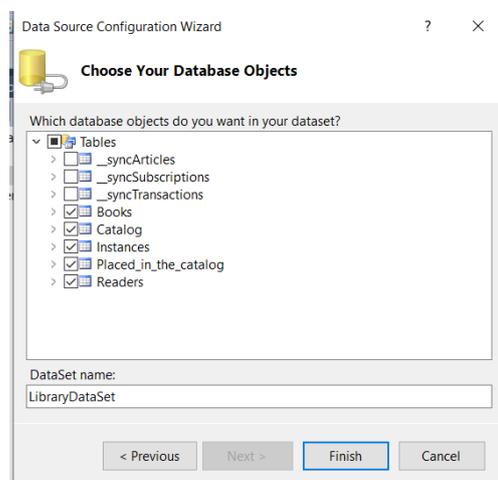


Рисунок 32 Выбор нужных таблиц

10. Запускаем Visual Studio. Создаем приложение Windows Form в C#.
11. Размещаем на форме компонент DataGridView. В его контекстном меню выбираем Other Data Sources.

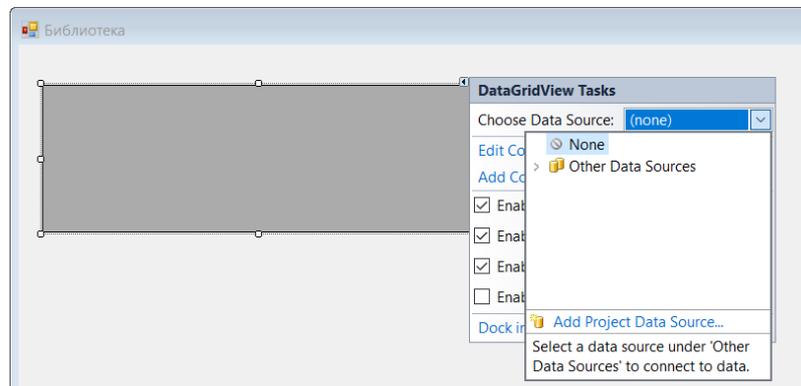


Рисунок 33 Подключение таблицы

12. Выбираем название той таблицы, которая будет отображаться.
13. Запускаем проект и видим, что таблица отобразилась в окне DataGridView.

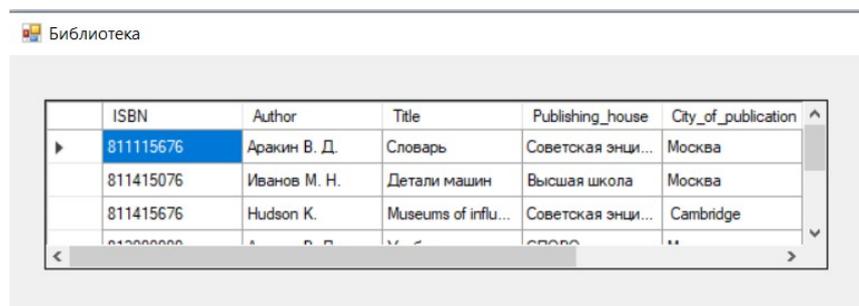


Рисунок 34 Отображение таблицы

14. Аналогичным образом поступаем с другими таблицами.
15. Разместим на форме 3 кнопки: Sync, Submit Changes и Refresh Data.
16. Кнопка Refresh Data будет обновлять наши элементы DataGridView записями из локального кэша. В обработчик на нажатие этой кнопки добавим следующий код.

```
private void button3_Click(object sender, EventArgs e)
{
    try
    {
        this.booksTableAdapter.Fill(this.libraryDataSet.Books);
        this.catalogTableAdapter.Fill(this.libraryDataSet.Catalog);
        this.readersTableAdapter.Fill(this.libraryDataSet.Readers);
        this.instancesTableAdapter.Fill(this.libraryDataSet.Instances);
        this.placed_in_the_catalogTableAdapter.Fill(this.libraryDataSet.Placed_in_the_catalog);
    }
    catch (System.Exception ex)
    {
    }
}
```

```

        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
}

```

17. Задача кнопки Submit Changes – передавать и сохранять изменения в элементах DataGridView в наш локальный кэш приложения. На ней соответственно будет исполняться следующий код.

```

private void button2_Click(object sender, EventArgs e)
{
    try
    {
        this.Validate();

        this.booksBindingSource.EndEdit();
        this.catalogBindingSource.EndEdit();
        this.readersBindingSource.EndEdit();
        this.instancesBindingSource.EndEdit();
        this.placedinthecatalogBindingSource.EndEdit();
        this.booksTableAdapter.Update(this.libraryDataSet.Books);
        this.catalogTableAdapter.Update(this.libraryDataSet.Catalog);
        this.readersTableAdapter.Update(this.libraryDataSet.Readers);
        this.instancesTableAdapter.Update(this.libraryDataSet.Instances);
        this.placed_in_the_catalogTableAdapter.Update(this.libraryDataSet.Placed_in_the_catalog);

        MessageBox.Show("Update successful");
    }
    catch (System.Exception ex)
    {
        MessageBox.Show("Update failed: " + ex.Message);
    }
}

```

18. С помощью кнопки Sync будет выполняться синхронизация нашего локального кэша с основной базой данных.

```

private void button1_Click(object sender, EventArgs e)
{
    try
    {
        LocalDataCache1SyncAgent syncAgent = new LocalDataCache1SyncAgent();
        Microsoft.Synchronization.Data.SyncStatistics syncStats = syncAgent.Synchronize();

        MessageBox.Show("Changes downloaded: " + syncStats.TotalChangesDownloaded.ToString() +
            Environment.NewLine + "Changes uploaded: " + syncStats.TotalChangesUploaded.ToString());
    }
    catch (System.Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
}

```

19. Также для того, чтобы обеспечить именно двухстороннюю синхронизацию, в окне Solution Explorer на MyLocalDataCache.sync жмем правой клавишей мыши, выбираем View Code и задаем такой код:

```

namespace WindowsFormsApplication31 {

```

```

public partial class LocalDataCache1SyncAgent {

    partial void OnInitialized(){
        this._booksSyncTable.SyncDirection = Microsoft.Synchronization.Data.SyncDirection.Bidirectional;
        this._catalogSyncTable.SyncDirection = Microsoft.Synchronization.Data.SyncDirection.Bidirectional;
        this._readersSyncTable.SyncDirection = Microsoft.Synchronization.Data.SyncDirection.Bidirectional;
        this._instancesSyncTable.SyncDirection = Microsoft.Synchronization.Data.SyncDirection.Bidirectional;
        this._placed_in_the_catalogSyncTable.SyncDirection = Microsoft.Synchronization.Data.SyncDirection.Bidirectional;
    }
}
}

```

20. Получилось готовое приложение.

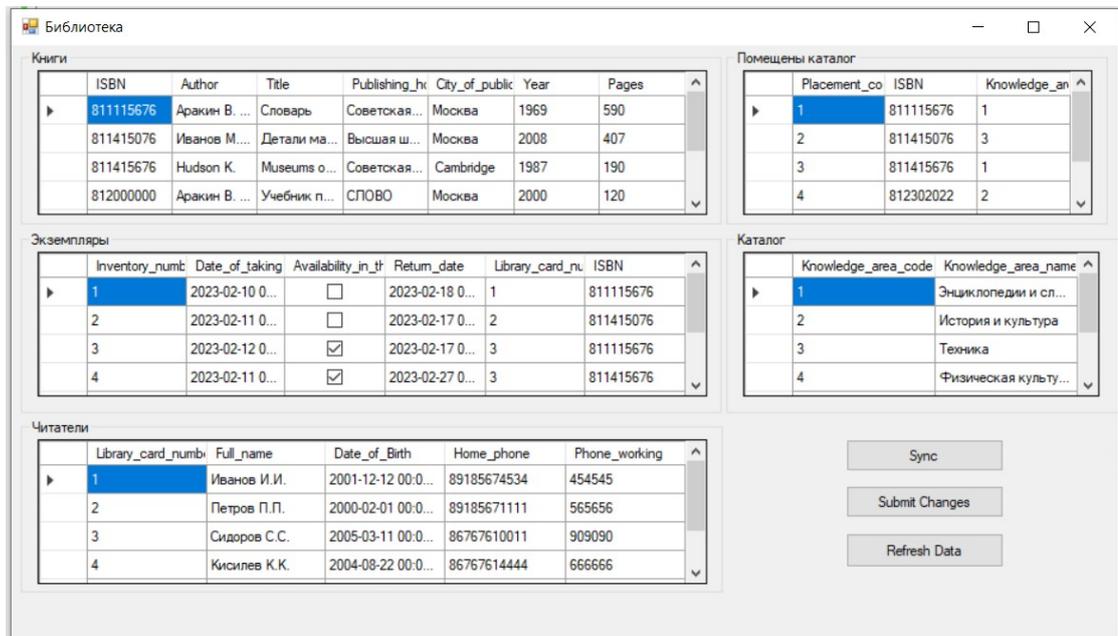


Рисунок 35 Приложение с применением технологии Change Tracking

ЗАКЛЮЧЕНИЕ

По итогам выполнения программы учебной практики, рассчитанной на 72 часа по профессиональному модулю «Разработка и администрирование баз данных» были приобретены следующие знания и умения:

- знание основных положений теории баз данных, хранилищ данных, баз знаний;
- знание основных принципов построения концептуальной, логической и физической модели данных;
- знание современных инструментальных средств разработки схемы базы данных;
- знание методов описания схем баз данных в современных СУБД;
- знание структуры данных СУБД, общего подхода к организации представлений, таблиц, индексов и кластеров;
- знание методов организации целостности данных;
- умение осуществлять сбор, обработку и анализ информации для проектирования баз данных;
- умение спроектировать базу данных на основе анализа предметной области;
- умение разработать объекты базы данных в соответствии с результатами анализа предметной области;
- умение реализовывать базу данных в конкретной системе управления базами данных;
- навык разработки прикладных программ;
- навык применения стандартных методов для защиты объектов базы данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. . Логачев, М. С. Информационные системы и программирование. Администратор баз данных. Выпускная квалификационная работа [Электронный ресурс]: учебник / М.С. Логачёв. — Москва: ИНФРА-М, 2021. — 439 с. — Режим доступа: <https://znanium.com/catalog/product/1141783>
2. . Голицына, О. Л. Основы проектирования баз данных [Электронный ресурс]: учеб. пособие / О. Л. Голицына, Т. Л. Партыка, И. И Попов. – 2-е изд., перераб. и доп. – М.: Форум, НИЦ ИНФРА-М, 2021. – 416 с. – Режим доступа: <https://znanium.com/catalog/product/1190668>
3. . Кумскова, И.А. Базы данных [Электронный ресурс]: учебник / Кумскова И.А. — Москва: КноРус, 2020. — 400 с. — Режим доступа: <https://book.ru/book/932493>
4. . Стасышин, В. М. Базы данных: технологии доступа [Электронный ресурс]: учебное пособие / В. М. Стасышин, Т. Л. Стасышина. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2020. — 164 с. — Режим доступа: <https://urait.ru/bcode/455863>
5. . Чулюков, В.А. Проектирование баз данных. Практический курс [Электронный ресурс]: учебное пособие / Чулюков В.А., Астахова И.Ф., Башарина С.О., Сидорова О.А. — Москва: Русайнс, 2020. — 163 с. — Режим доступа: <https://book.ru/book/938011>
6. .Полищук, Ю. В. Базы данных и их безопасность [Электронный ресурс]: учебное пособие / Ю. В. Полищук, А. С. Боровский. - Москва : ИНФРА-М, 2021. - 210 с. – Режим доступа: <https://znanium.com/catalog/product/1084368>
7. .Мартишин, С. А. Базы данных. Практическое применение СУБД SQL- и NoSQL-типа для проектирования информационных систем [Электронный ресурс]: учебное пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. — Москва: ФОРУМ: ИНФРА-М, 2021. — 368 с. – Режим доступа: <https://znanium.com/catalog/product/1243192>

8.

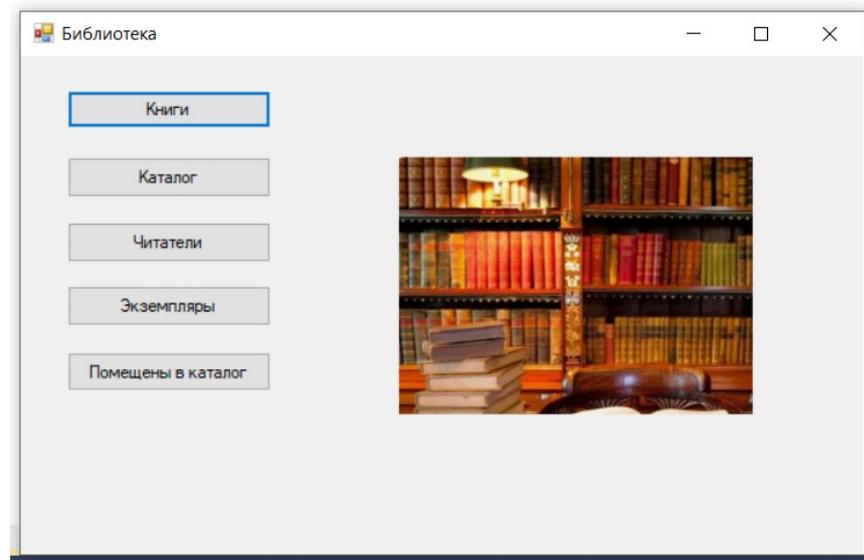


Рисунок 36 Главное меню в приложении

```
using System;
using System.Data;
using System.Windows.Forms;

namespace WindowsFormsApplication21
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form2 newForm = new Form2();
            newForm.Show();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Form3 newForm = new Form3();
            newForm.Show();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            Form4 newForm = new Form4();
            newForm.Show();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            Form5 newForm = new Form5();
            newForm.Show();
        }

        private void button5_Click(object sender, EventArgs e)
        {

```

```

Form6 newForm = new Form6();
newForm.Show();
}
}
}
}

```

ISBN	Автор	Название	Издательство	Место_издания	Год_издания	Кол_во_страниц
800000000	народ	Курочка Ряба	Белый город	Москва	1981	9
811415076	Иванов, М. Н	Детали машин	Высшая школа	Москва	2008	407
811415676	Hudson K.	Museums of influe...	Советская энци...	Cambridge	1987	190
812302022	Моряков В. И.	История России	СЛОВО	Ростов н/Д	2004	447
812343456	народ	Колобок	МОСКВА	Москва	1960	5
812675676	Каштанов Ю. И.	Русский костюм	Белый город	Москва	2009	48

ISBN:
 Автор:
 Название:
 Издательство:
 Место издания:
 Год издания:
 Кол-во страниц:

Рисунок 37 Форма для работы с таблицей Книги

```

using System;
using System.Data;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WindowsFormsApplication21
{
    public partial class Form2 : Form
    {
        DataSet НаборДанных;//ds
        OleDbDataAdapter Адаптер;//da
        OleDbConnection Подключение;//con
        OleDbCommand Команда;//cmd
        void GetKnigi()
        {
            Подключение = new OleDbConnection("Data Source='D:\\SQL1.mdb';User " +
            "ID=Admin;Provider='Microsoft.Jet.OLEDB.4.0';");
            Адаптер = new OleDbDataAdapter("Select * From [Книги]", Подключение);
            НаборДанных = new DataSet();
            Подключение.Open();
            Адаптер.Fill(НаборДанных, "Книги");
            dataGridView1.DataSource = НаборДанных.Tables["Книги"];
            Подключение.Close();
        }
        public Form2()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}

```

```

private void Form2_Load(object sender, EventArgs e)
{
    GetKnigi();
}

private void button2_Click(object sender, EventArgs e)//удаление записи
{
    string query = "Delete From Книги Where ISBN=@ISBN";
    Команда = new OleDbCommand(query, Подключение);
    Команда.Parameters.AddWithValue("@ISBN", dataGridView1.CurrentRow.Cells[0].Value);
    Подключение.Open();
    Команда.ExecuteNonQuery();
    Подключение.Close();
    GetKnigi();
}

private void button3_Click(object sender, EventArgs e)//добавление записи
{
    string query = "Insert into Книги
(ISBN,Автор,Название,Издательство,Место_издания,Год_издания,Кол_во_страниц) values
(@ISBN,@avtor,@nazv,@izdat,@Mizd,@god,@str)";
    Команда = new OleDbCommand(query, Подключение);
    Команда.Parameters.AddWithValue("@ISBN", Convert.ToInt64(textBox1.Text));
    Команда.Parameters.AddWithValue("@avtor", textBox2.Text);
    Команда.Parameters.AddWithValue("@nazv", textBox3.Text);
    Команда.Parameters.AddWithValue("@izdat", textBox4.Text);
    Команда.Parameters.AddWithValue("@Mizd", textBox5.Text);
    Команда.Parameters.AddWithValue("@god", Convert.ToInt32(textBox6.Text));
    Команда.Parameters.AddWithValue("@str", Convert.ToInt32(textBox7.Text));
    Подключение.Open();
    Команда.ExecuteNonQuery();
    Подключение.Close();
    GetKnigi();
}

private void button4_Click(object sender, EventArgs e)
{
    string query = "Update Книги Set
ISBN=@ISBN,Автор=@avtor,Название=@nazv,Издательство=@izdat,Место_издания=@Mizd,Год_издания=@god,Кол_во_страниц=@str Where ISBN=@ISBN";
    Команда = new OleDbCommand(query, Подключение);
    Команда.Parameters.AddWithValue("@ISBN", Convert.ToInt64(textBox1.Text));
    Команда.Parameters.AddWithValue("@avtor", textBox2.Text);
    Команда.Parameters.AddWithValue("@nazv", textBox3.Text);
    Команда.Parameters.AddWithValue("@izdat", textBox4.Text);
    Команда.Parameters.AddWithValue("@Mizd", textBox5.Text);
    Команда.Parameters.AddWithValue("@god", Convert.ToInt32(textBox6.Text));
    Команда.Parameters.AddWithValue("@str", Convert.ToInt32(textBox7.Text));
    Подключение.Open();
    Команда.ExecuteNonQuery();
    Подключение.Close();
    GetKnigi();
}

private void dataGridView1_CellEnter_1(object sender, DataGridViewCellEventArgs e)
{
    textBox1.Text = dataGridView1.CurrentRow.Cells[0].Value.ToString();
    textBox2.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
    textBox3.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
    textBox4.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
    textBox5.Text = dataGridView1.CurrentRow.Cells[4].Value.ToString();
    textBox6.Text = dataGridView1.CurrentRow.Cells[5].Value.ToString();
    textBox7.Text = dataGridView1.CurrentRow.Cells[6].Value.ToString();
}

```

```

}
}

```

Код_области_знаний	Название_области_знаний
1	Физика
2	Химия
3	Биология
4	История
5	География
6	Спорт
7	Музыка

Код области знаний:

Название области знаний:

Вставить
Удалить
Обновить
Закреть

Рисунок 38 Форма для работы с таблицей Каталог

```

using System;
using System.Data;
using System.Windows.Forms;
using System.Data.OleDb;
namespace WindowsFormsApplication21
{
    public partial class Form3 : Form
    {
        DataSet НаборДанных;//ds
        OleDbDataAdapter Адаптер;//da
        OleDbConnection Подключение;//con
        OleDbCommand Команда;//cmd
        void GetKatalog()
        {
            Подключение = new OleDbConnection("Data Source='D:\\SQL1.mdb';User " +
            "ID=Admin;Provider='Microsoft.Jet.OLEDB.4.0';");
            Адаптер = new OleDbDataAdapter("Select * From [Каталог]", Подключение);
            НаборДанных = new DataSet();
            Подключение.Open();
            Адаптер.Fill(НаборДанных, "Каталог");
            dataGridView1.DataSource = НаборДанных.Tables["Каталог"];
            Подключение.Close();
        }
        public Form3()
        {
            InitializeComponent();
        }

        private void Form3_Load(object sender, EventArgs e)
        {
            GetKatalog();
        }

        private void button2_Click(object sender, EventArgs e)//удаление записи
        {
            string query = "Delete From Каталог Where Код_области_знаний=@Код_области_знаний";
            Команда = new OleDbCommand(query, Подключение);
            Команда.Parameters.AddWithValue("@Код_области_знаний", dataGridView1.CurrentRow.Cells[0].Value);
            Подключение.Open();
            Команда.ExecuteNonQuery();
            Подключение.Close();
        }
    }
}

```



```

using System.Data;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WindowsFormsApplication21
{
    public partial class Form4 : Form
    {
        DataSet НаборДанных;//ds
        OleDbDataAdapter Адаптер;//da
        OleDbConnection Подключение;//con
        OleDbCommand Команда;//cmd
        void GetChit()
        {
            Подключение = new OleDbConnection("Data Source='D:\\SQL1.mdb';User " +
"ID=Admin;Provider='Microsoft.Jet.OLEDB.4.0';");
            Адаптер = new OleDbDataAdapter("Select * From [Читатели]", Подключение);
            НаборДанных = new DataSet();
            Подключение.Open();
            Адаптер.Fill(НаборДанных, "Читатели");
            dataGridView1.DataSource = НаборДанных.Tables["Читатели"];
            Подключение.Close();
        }
        public Form4()
        {
            InitializeComponent();
        }

        private void Form4_Load(object sender, EventArgs e)
        {
            GetChit();
        }

        private void button1_Click(object sender, EventArgs e)//закрыть
        {
            this.Close();
        }

        private void button2_Click(object sender, EventArgs e)//удалить
        {
            string query = "Delete From Читатели Where Номер_читательского_билета=@Номер_читательского_билета";
            Команда = new OleDbCommand(query, Подключение);
            Команда.Parameters.AddWithValue("@Номер_читательского_билета",
dataGridView1.CurrentRow.Cells[0].Value);
            Подключение.Open();
            Команда.ExecuteNonQuery();
            Подключение.Close();
            GetChit();
        }

        private void button3_Click(object sender, EventArgs e)//вставка
        {
            string query = "Insert into Читатели
(Номер_читательского_билета,ФИО,Дата_рождения,Телефон_домашний,Телефон_рабочий) values
(@Nchit,@FIO,@DRozhd,@TDom,@TRab)";
            Команда = new OleDbCommand(query, Подключение);
            Команда.Parameters.AddWithValue("@Nchit", Convert.ToInt32(textBox1.Text));
            Команда.Parameters.AddWithValue("@FIO", textBox2.Text);
            Команда.Parameters.AddWithValue("@DRozhd", textBox3.Text);
            Команда.Parameters.AddWithValue("@TDom", textBox4.Text);
            Команда.Parameters.AddWithValue("@TRab", textBox5.Text);
            Подключение.Open();
            Команда.ExecuteNonQuery();
        }
    }
}

```



```

        Подключение = new OleDbConnection("Data Source=\\D:\\SQL1.mdb";User " +
"ID=Admin;Provider=Microsoft.Jet.OLEDB.4.0";");
        Адаптер = new OleDbDataAdapter("Select * From [Экземпляры]", Подключение);
        НаборДанных = new DataSet();
        Подключение.Open();
        Адаптер.Fill(НаборДанных, "Экземпляры");
        dataGridView1.DataSource = НаборДанных.Tables["Экземпляры"];
        Подключение.Close();
    }
    public Form5()
    {
        InitializeComponent();
    }

    private void Form5_Load(object sender, EventArgs e)
    {
        GetEx();
    }

    private void button1_Click_1(object sender, EventArgs e)
    {
        this.Close();
    }

    private void button2_Click(object sender, EventArgs e)//удаление
    {
        string query = "Delete From Экземпляры Where Инвентарный_номер=@Инвентарный_номер";
        Команда = new OleDbCommand(query, Подключение);
        Команда.Parameters.AddWithValue("@Инвентарный_номер", dataGridView1.CurrentRow.Cells[0].Value);
        Подключение.Open();
        Команда.ExecuteNonQuery();
        Подключение.Close();
        GetEx();
    }

    private void button3_Click(object sender, EventArgs e)//вставка
    {
        string query = "Insert into Экземпляры
(Инвентарный_номер,Дата_взятия,Наличие_в_библиотеке,Дата_возврата,Номер_читательского_билета,ISBN) values
(@Inv,@DVzat,@Nal,@TDvozv,@NChit,@ISBN)";
        Команда = new OleDbCommand(query, Подключение);
        Команда.Parameters.AddWithValue("@Inv", Convert.ToInt32(textBox1.Text));
        Команда.Parameters.AddWithValue("@DVzat", textBox2.Text);
        Команда.Parameters.AddWithValue("@Nal", Convert.ToBoolean(textBox3.Text));
        Команда.Parameters.AddWithValue("@TDvozv", textBox4.Text);
        Команда.Parameters.AddWithValue("@NChit", Convert.ToInt32(textBox5.Text));
        Команда.Parameters.AddWithValue("@ISBN", Convert.ToInt32(textBox6.Text));
        Подключение.Open();
        Команда.ExecuteNonQuery();
        Подключение.Close();
        GetEx();
    }

    private void dataGridView1_CellEnter(object sender, DataGridViewCellEventArgs e)
    {
        textBox1.Text = dataGridView1.CurrentRow.Cells[0].Value.ToString();
        textBox2.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
        textBox3.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
        textBox4.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
        textBox5.Text = dataGridView1.CurrentRow.Cells[4].Value.ToString();
        textBox6.Text = dataGridView1.CurrentRow.Cells[5].Value.ToString();
    }

    private void button4_Click(object sender, EventArgs e)//обновление
    {

```

```

string query = "Update Экземпляры Set
Инвентарный_номер=@Inv,Дата_взятия=@DVzat,Наличие_в_библиотеке=@Nal,Дата_возврата=@TDvozv,Номер_чи
тательского_билета=@NChit, ISBN=@ISBN Where Инвентарный_номер=@Inv";
Команда = new OleDbCommand(query, Подключение);
Команда.Parameters.AddWithValue("@Inv", Convert.ToInt32(textBox1.Text));
Команда.Parameters.AddWithValue("@DVzat", textBox2.Text);
Команда.Parameters.AddWithValue("@Nal", Convert.ToBoolean(textBox3.Text));
Команда.Parameters.AddWithValue("@TDvozv", textBox4.Text);
Команда.Parameters.AddWithValue("@NChit", Convert.ToInt32(textBox5.Text));
Команда.Parameters.AddWithValue("@ISBN", Convert.ToInt32(textBox6.Text));
Подключение.Open();
Команда.ExecuteNonQuery();
Подключение.Close();
GetEx();
}
}
}

```

Код размещения	ISBN	Код области знаний
1	800000000	3
2	811415676	2
4	812302022	1
5	812675676	4
6	812675676	7

Код размещения:

ISBN:

Код области знаний:

Вставить
Удалить
Обновить
Заккрыть

Рисунок 41 Форма для работы с таблицей Помещены в каталог

```

using System;
using System.Data;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WindowsFormsApplication21
{
    public partial class Form6 : Form
    {
        DataSet НаборДанных;//ds
        OleDbDataAdapter Адаптер;//da
        OleDbConnection Подключение;//con
        OleDbCommand Команда;//cmd
        void GetPom()
        {
            Подключение = new OleDbConnection("Data Source='D:\\SQL1.mdb';User " +
            "ID=Admin;Provider='Microsoft.Jet.OLEDB.4.0';");
            Адаптер = new OleDbDataAdapter("Select * From [Помещены_в_каталог]", Подключение);
            НаборДанных = new DataSet();
            Подключение.Open();
            Адаптер.Fill(НаборДанных, "Помещены_в_каталог");
            dataGridView1.DataSource = НаборДанных.Tables["Помещены_в_каталог"];
            Подключение.Close();
        }
        public Form6()
        {

```

```

        InitializeComponent();
    }

    private void Form6_Load(object sender, EventArgs e)
    {
        GetPom();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        this.Close();
    }
    private void button3_Click(object sender, EventArgs e)//вставка
    {
        string query = "Insert into Помещены_в_каталог (Код_размещения,ISBN,Код_области_знаний) values
(@KRaz,@ISBN,@KObl)";
        Команда = new OleDbCommand(query, Подключение);
        Команда.Parameters.AddWithValue("@KRaz", Convert.ToInt32(textBox1.Text));
        Команда.Parameters.AddWithValue("@ISBN", Convert.ToInt32(textBox2.Text));
        Команда.Parameters.AddWithValue("@KObl", Convert.ToInt32(textBox3.Text));
        Подключение.Open();
        Команда.ExecuteNonQuery();
        Подключение.Close();
        GetPom();
    }
    private void button2_Click(object sender, EventArgs e)//удаление
    {
        string query = "Delete From Помещены_в_каталог Where Код_размещения=@Код_размещения";
        Команда = new OleDbCommand(query, Подключение);
        Команда.Parameters.AddWithValue("@Код_размещения", dataGridView1.CurrentRow.Cells[0].Value);
        Подключение.Open();
        Команда.ExecuteNonQuery();
        Подключение.Close();
        GetPom();
    }

    private void dataGridView1_CellEnter(object sender, DataGridViewCellEventArgs e)
    {
        textBox1.Text = dataGridView1.CurrentRow.Cells[0].Value.ToString();
        textBox2.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
        textBox3.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
    }

    private void button4_Click(object sender, EventArgs e)//обновление
    {
        string query = "Update Помещены_в_каталог Set
Код_размещения=@KRaz,ISBN=@ISBN,Код_области_знаний=@KObl Where Код_размещения=@KRaz";
        Команда = new OleDbCommand(query, Подключение);
        Команда.Parameters.AddWithValue("@KRaz", Convert.ToInt32(textBox1.Text));
        Команда.Parameters.AddWithValue("@ISBN", Convert.ToInt32(textBox2.Text));
        Команда.Parameters.AddWithValue("@KObl", Convert.ToInt32(textBox3.Text));
        Подключение.Open();
        Команда.ExecuteNonQuery();
        Подключение.Close();
        GetPom();
    }

    private void button5_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}
}

```

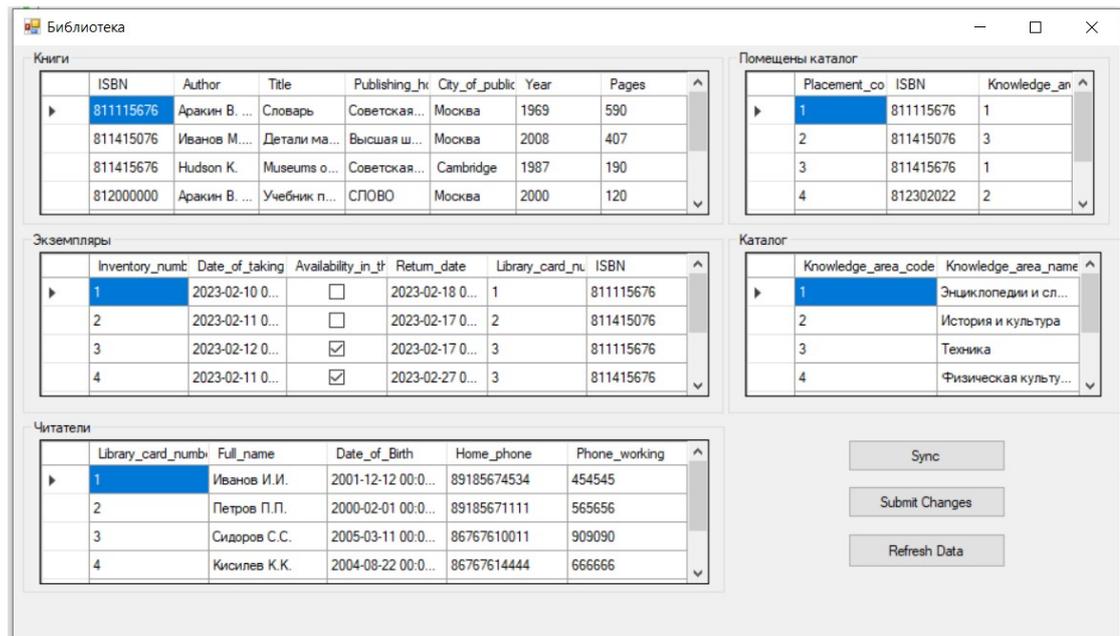


Рисунок 42 Приложение с технологией Change Tracking.

Код для Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication31
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'libraryDataSet.Catalog' table. You can move, or remove it, as needed.
            this.catalogTableAdapter.Fill(this.libraryDataSet.Catalog);
            // TODO: This line of code loads data into the 'libraryDataSet.Readers' table. You can move, or remove it, as needed.
            this.readersTableAdapter.Fill(this.libraryDataSet.Readers);
            // TODO: This line of code loads data into the 'libraryDataSet.Instances' table. You can move, or remove it, as needed.
            this.instancesTableAdapter.Fill(this.libraryDataSet.Instances);
            // TODO: This line of code loads data into the 'libraryDataSet.Placed_in_the_catalog' table. You can move, or remove
            it, as needed.
            this.placed_in_the_catalogTableAdapter.Fill(this.libraryDataSet.Placed_in_the_catalog);
            // TODO: This line of code loads data into the 'libraryDataSet.Books' table. You can move, or remove it, as needed.
            this.booksTableAdapter.Fill(this.libraryDataSet.Books);
            // TODO: This line of code loads data into the 'libraryDataSet.Readers' table. You can move, or remove it, as needed.
            this.readersTableAdapter.Fill(this.libraryDataSet.Readers);
        }
    }
}
```

```

        // TODO: This line of code loads data into the 'libraryDataSet.Placed_in_the_catalog' table. You can move, or remove
it, as needed.
        this.placed_in_the_catalogTableAdapter.Fill(this.libraryDataSet.Placed_in_the_catalog);
        // TODO: This line of code loads data into the 'libraryDataSet.Instances' table. You can move, or remove it, as needed.
        this.instancesTableAdapter.Fill(this.libraryDataSet.Instances);
        // TODO: This line of code loads data into the 'libraryDataSet.Catalog' table. You can move, or remove it, as needed.
        this.catalogTableAdapter.Fill(this.libraryDataSet.Catalog);
        // TODO: This line of code loads data into the 'libraryDataSet.Books' table. You can move, or remove it, as needed.
        this.booksTableAdapter.Fill(this.libraryDataSet.Books);

    }

    private void button3_Click(object sender, EventArgs e)
    {
        try
        {
            this.booksTableAdapter.Fill(this.libraryDataSet.Books);
            this.catalogTableAdapter.Fill(this.libraryDataSet.Catalog);
            this.readersTableAdapter.Fill(this.libraryDataSet.Readers);
            this.instancesTableAdapter.Fill(this.libraryDataSet.Instances);
            this.placed_in_the_catalogTableAdapter.Fill(this.libraryDataSet.Placed_in_the_catalog);
        }
        catch (System.Exception ex)
        {
            System.Windows.Forms.MessageBox.Show(ex.Message);
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        try
        {
            this.Validate();

            this.booksBindingSource.EndEdit();
            this.catalogBindingSource.EndEdit();
            this.readersBindingSource.EndEdit();
            this.instancesBindingSource.EndEdit();
            this.placedinthecatalogBindingSource.EndEdit();
            this.booksTableAdapter.Update(this.libraryDataSet.Books);
            this.catalogTableAdapter.Update(this.libraryDataSet.Catalog);
            this.readersTableAdapter.Update(this.libraryDataSet.Readers);
            this.instancesTableAdapter.Update(this.libraryDataSet.Instances);
            this.placed_in_the_catalogTableAdapter.Update(this.libraryDataSet.Placed_in_the_catalog);

            MessageBox.Show("Update successful");
        }
        catch (System.Exception ex)
        {
            MessageBox.Show("Update failed: " + ex.Message);
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            LocalDataCache1SyncAgent syncAgent = new LocalDataCache1SyncAgent();
            Microsoft.Synchronization.Data.SyncStatistics syncStats = syncAgent.Synchronize();

            MessageBox.Show("Changes downloaded: " + syncStats.TotalChangesDownloaded.ToString() +
Environment.NewLine + "Changes uploaded: " + syncStats.TotalChangesUploaded.ToString());
        }
        catch (System.Exception ex)
        {

```

```
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
}
}
```

Код для LocalDataCache1.cs

```
namespace WindowsFormsApplication31 {

    public partial class LocalDataCache1SyncAgent {

        partial void OnInitialized(){
            this._booksSyncTable.SyncDirection = Microsoft.Synchronization.Data.SyncDirection.Bidirectional;
            this._catalogSyncTable.SyncDirection = Microsoft.Synchronization.Data.SyncDirection.Bidirectional;
            this._readersSyncTable.SyncDirection = Microsoft.Synchronization.Data.SyncDirection.Bidirectional;
            this._instancesSyncTable.SyncDirection = Microsoft.Synchronization.Data.SyncDirection.Bidirectional;
            this._placed_in_the_catalogSyncTable.SyncDirection = Microsoft.Synchronization.Data.SyncDirection.Bidirectional;
        }
    }
}
```