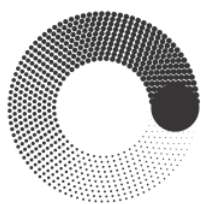


ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ

*Институт Принтмедиа и информационных технологий
Кафедра Информатики и информационных технологий*

направление подготовки
09.03.02 «Информационные системы и технологии»,

КУРСОВОЙ ПРОЕКТ

Дисциплина: Базы данных

Тема: Разработка информационно-справочной системы библиотеки

Выполнил(а): студент(ка) группы 221-373

Федосеева Е.С.

(Фамилия И.О.)

Дата, подпись _____
(Дата) (Подпись)

Проверил: _____
(Фамилия И.О., степень, звание)

Дата, подпись _____
(Дата) (Подпись)

Замечания: _____

Москва

2024

Оглавление

Введение	3
1. Проектирование	4
1.1. Описание предметной области	4
1.2. Выбор инструментов	4
2. Разработка информационной системы	8
2.1. Проектирование и физическая реализация базы данных на СУБД	8
2.2. Разработка программного продукта	11
3. Заключение	19
4. Библиографический список	20

Введение

Тема проекта – информационная система, предназначенная для учета выдачи и приема книг в библиотеке.

Проблема учета стоит перед библиотекарем каждый день, а ведение бумажной картотеки занимает большое количество рабочего времени. Поскольку перед библиотекарем стоит задача снижения материальных затрат на ведение учета, логично передать выполнение функции учета программе. Исходя из этого, возникла и была реализована идея создания приложения, которое даёт возможность просматривать наличие библиотечного фонда, персональные данные пользователей библиотеки, данные о получении и возврате книг пользователями. Это позволяет сэкономить время и силы библиотекаря, поскольку ему не нужно просматривать бумажную картотеку для получения информации о книгах и пользователях.

Цель – создание информационной системы для ведения учета книг, пользователей и истории пользования библиотекой, информации о взятых книгах.

Исходя из поставленной цели, были сформированы следующие задачи:

1. Описание предметной области;
2. Выбор и обоснование инструментов разработки;
3. Проектирование базы данных
4. Реализация физической модели данных.

1. Проектирование

1.1. Описание предметной области

Библиотека представляет собой хранилище книг. Основные ее функции - выдавать читателям книги и принимать их. Остальные функции относятся к обслуживанию процесса выдачи и приема. Экземпляр каждой книги, хранящейся в библиотеке, должен иметь индивидуальный номер. Каждый читатель должен иметь свой абонемент, который хранит как историю заказов, так и информацию о самом читателе. Каждый читатель имеет индивидуальный номер. По индивидуальному номеру работник библиотеки может определить ФИО читателя и его контактные данные. Придя в библиотеку, читатель, делает заказ на книгу. Сотрудник библиотеки может принять заказ в устной форме или заполнить бланк заказа, на котором следует указать следующие данные:

- ФИО автора;
- наименование книги;
- жанр;
- год;
- издательство.

Библиотекарь принимает заказ, проверяет наличие ассортимента и по возможности выдает необходимые книги. Информация о выданной книге заносится библиотекарем в таблицу, где указывается дата выдачи и информация о книге.

1.2. Выбор инструментов

Для реализации проекта была выбрана десктопная форма приложения. Данный формат проекта был выбран из расчета на то, что приложение будет ориентирована на библиотекарей, которые являются работниками конкретной библиотеки. Отображение информации на экране компьютера имеют больший размер и четкость, нежели на экранах карманных устройств.

Формат данного приложения подразумевает его использование на операционной системе Windows. Для того чтобы сделать это возможным, на

этапе разработки была выбрана интегрированная среда разработки Microsoft Visual Studio и язык программирования C#, который поддерживается ею.

Microsoft Visual Studio — линейка продуктов компании Microsoft, в которую входят интегрированная среда разработки программного обеспечения и множество других инструментов. Эта линейка дает возможность разрабатывать приложения разных типов, таких как консольные приложения, веб-сайты, веб-приложения, веб-службы, для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight. А также приложения с графическим интерфейсом, в том числе, с поддержкой технологии Windows Forms.

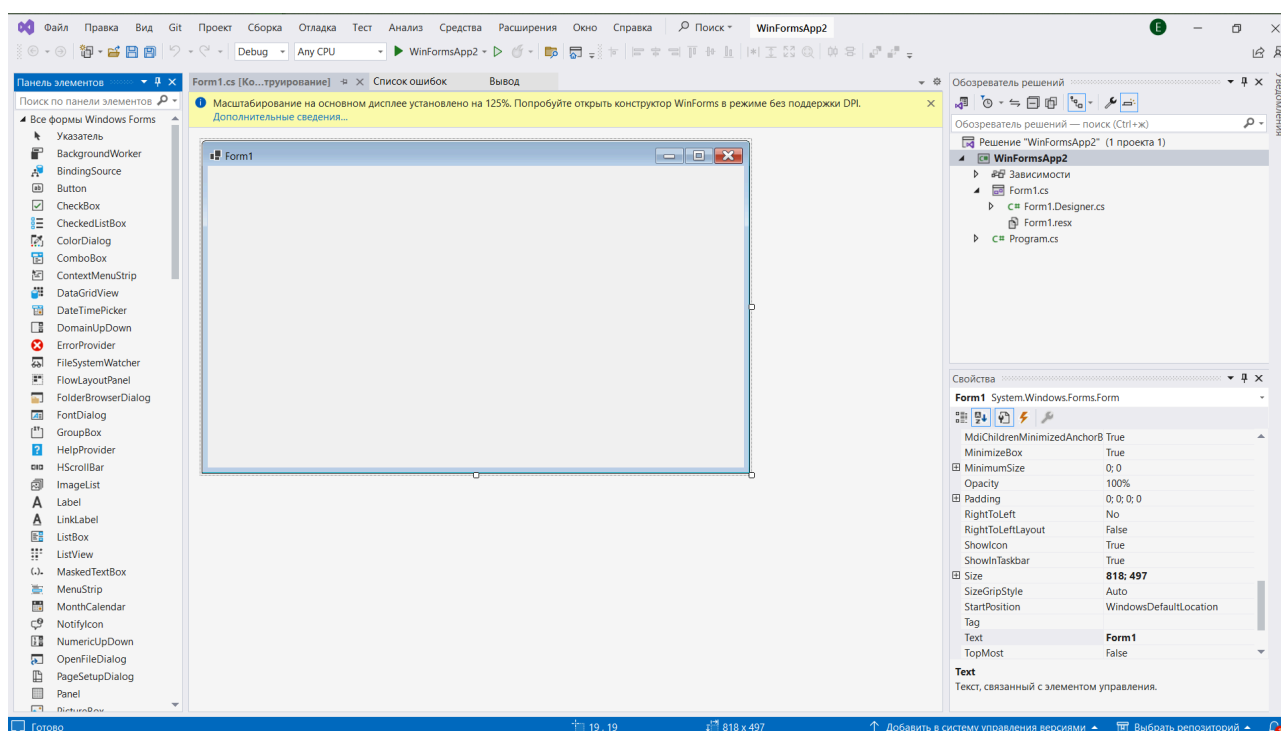


Рисунок 1.1 Оболочка Microsoft Visual Studio

C# — простой, современный объектно-ориентированный и типобезопасный язык программирования. C# относится к широко известному семейству языков C, и покажется хорошо знакомым любому, кто работал с C, C++, Java или JavaScript.

C# является объектно-ориентированным языком, но поддерживает также и компонентно-ориентированное программирование. Разработка современных приложений все больше тяготеет к созданию программных компонентов в

форме автономных и пакетов, реализующих отдельные функциональные возможности. Важная особенность таких компонентов — это модель программирования на основе свойств, методов и событий. Каждый компонент имеет атрибуты, предоставляющие декларативные сведения о компоненте, а также встроенные элементы документации. С# предоставляет языковые конструкции, непосредственно поддерживающие такую концепцию работы. Благодаря этому С# отлично подходит для создания и применения программных компонентов.

Для создания баз данных в роли СУБД в данном проекте выступает Microsoft SQL.

Microsoft SQL — инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных Microsoft SQL.

В качестве среды разработки баз данных, созданных на Microsoft SQL, предпочтение было отдано приложению SQL Server Management Studio.

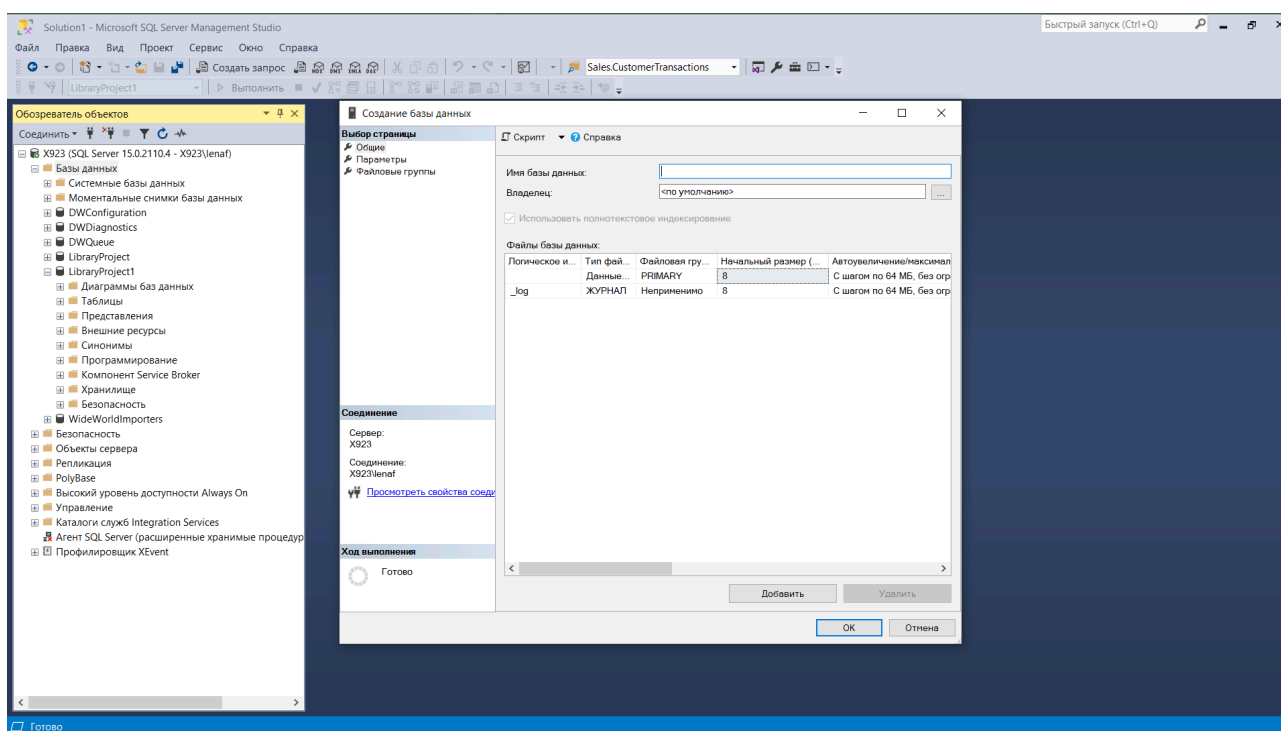


Рисунок 1.2 Оболочка SQL Server Management Studio

SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире. Данная СУБД подходит для самых различных проектов: от небольших приложений до больших высоконагруженных проектов.

SQL Server характеризуется такими особенностями как:

- Производительность. SQL Server работает очень быстро.
- Надежность и безопасность. SQL Server предоставляет шифрование данных.
- Простота. С данной СУБД относительно легко работать и вести администрирование.

2. Разработка информационной системы библиотеки

2.1. Проектирование и физическая реализация базы данных на СУБД

Для реализации данного проекта была реализована база данных, состоящая из трех таблиц. Таблица реляционной базы данных — это совокупность связанных данных, хранящихся в структурированном виде в базе данных. Структура и связи этой базы данных представлены на рисунке ниже.

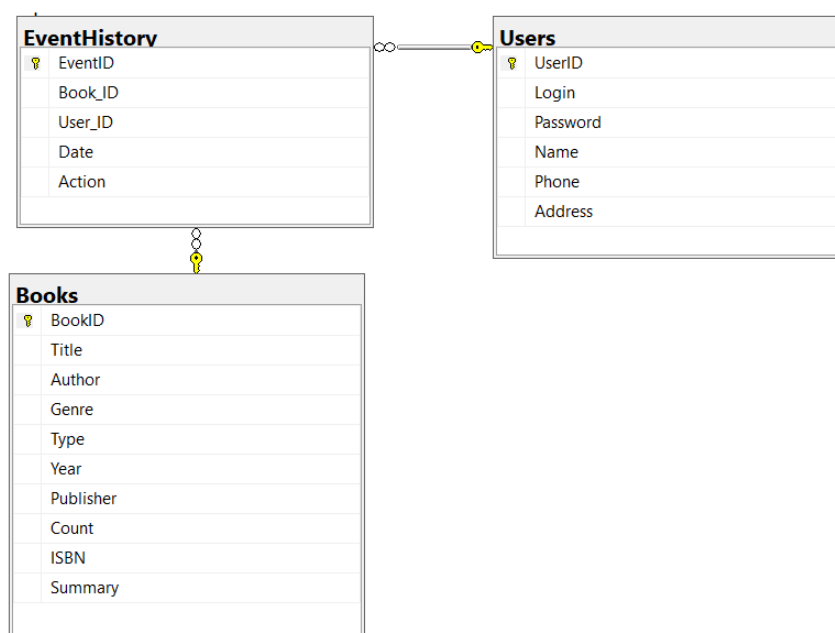


Рисунок 2.1 Структура базы данных

Далее представлены структуры всех таблиц с пояснением к каждой.

Таблица Books – книги – таблица, которой хранится информация об имеющихся книгах. Данная таблица содержит

- BookID – номер книги, является первичным ключом;
- Title – название книги;
- Author – автор книги;
- Genre – жанр книги;
- Type – литературный тип;
- Year – год издания;
- Publisher – издатель;
- Count – количество книг в библиотеке;

- ISBN – уникальный номер книжного издания;
- Summary – аннотация к книге.

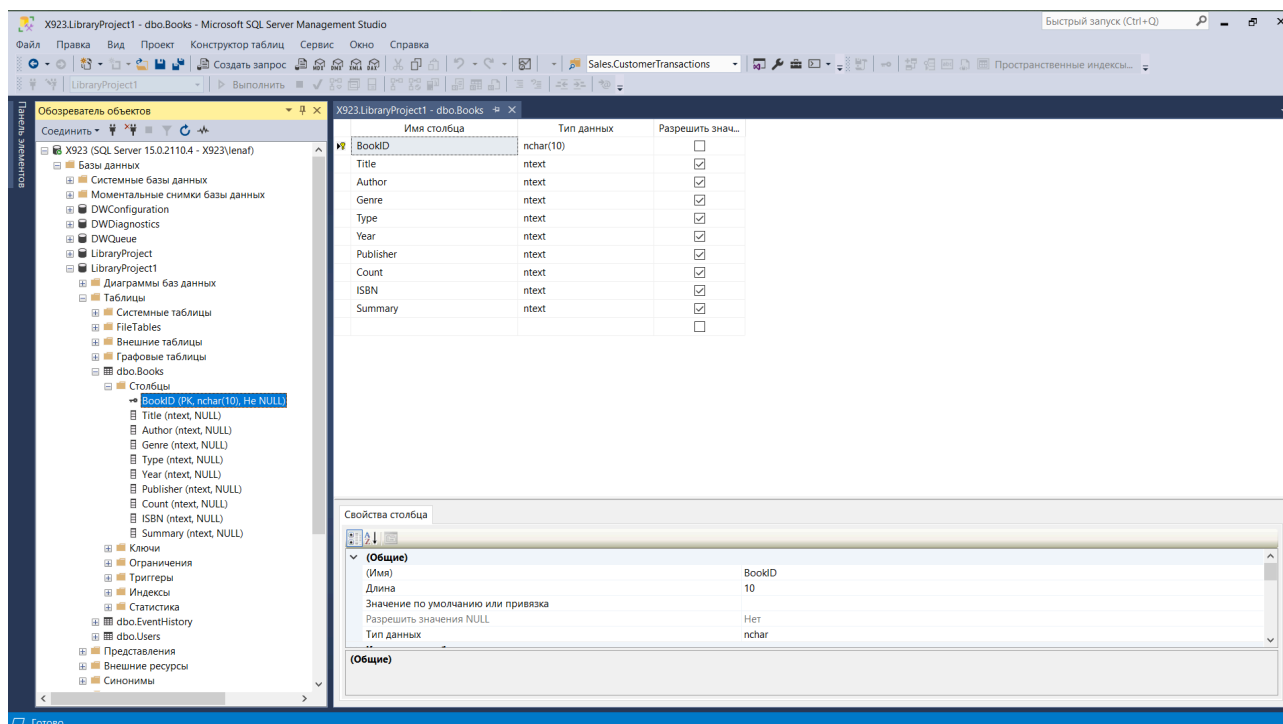


Рисунок 2.2 Таблица Books

Таблица Users – пользователи – таблица, в которой хранится информация обо всех пользователях. Данная таблица содержит:

- UserID – номер пользователя, является первичным ключом;
- Login – логин;
- Password – пароль;
- Name – ФИО пользователя;
- Phone – номер телефона;
- Address – адрес.

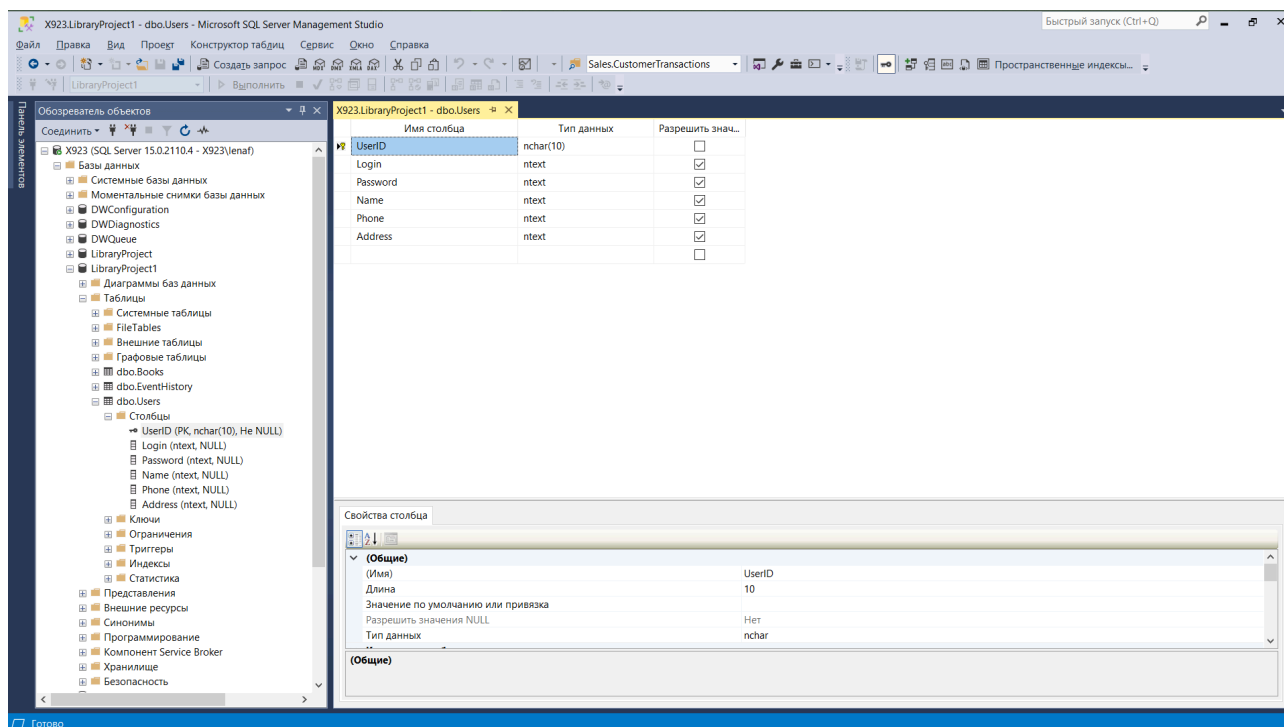


Рисунок 2.3 Таблица Users

Таблица EventHistory – история событий – таблица, в которой хранится информация о выданных и возвращенных книгах. Данная таблица содержит:

- EventID – номер события, является первичным ключом;
- Book_ID – номер книги, является внешним ключом;
- User_ID – номер пользователя, является внешним ключом;
- Date – дата события;
- Action. – какое событие произошло

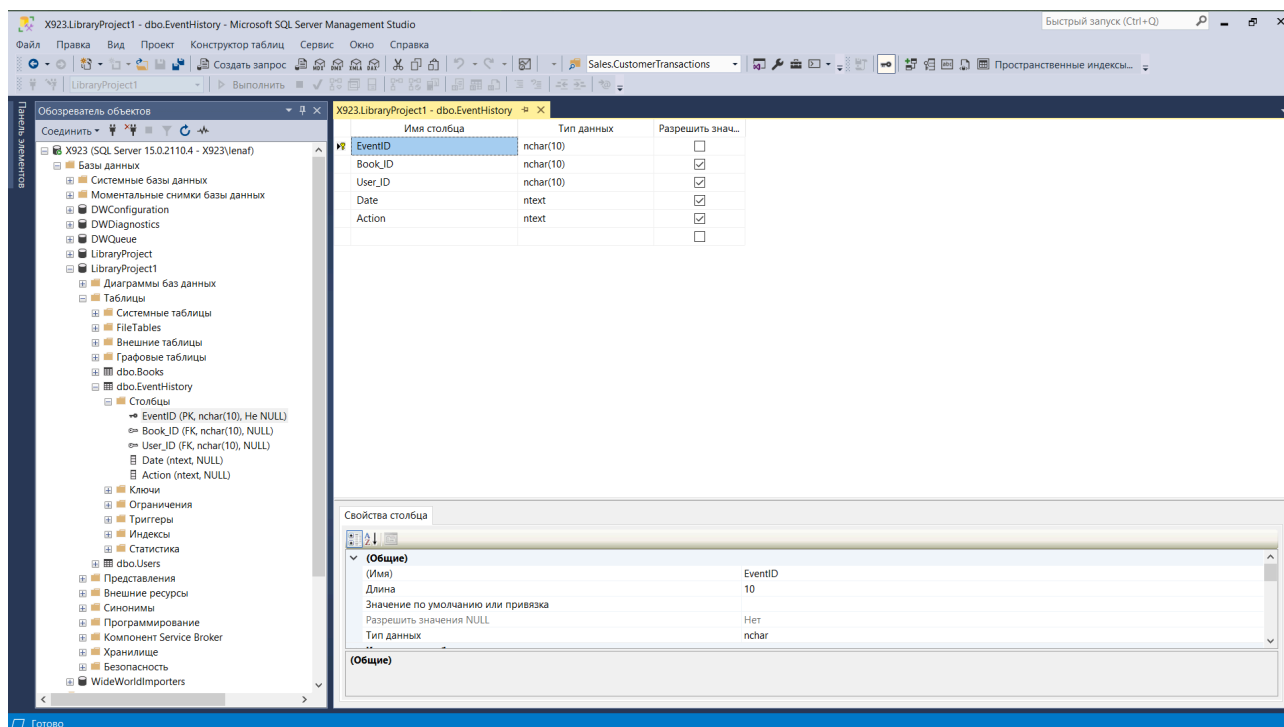


Рисунок 2.4 Таблица EventHistory

Так, с помощью приложения SQL Server Management Studio была успешно спроектирована и реализована база данных информационной системы библиотеки.

2.2. Разработка программного продукта

Первое, что видит пользователь, запуская приложение, это страница входа. На данной странице библиотекарь может выбрать таблицу, с которой будет работать.

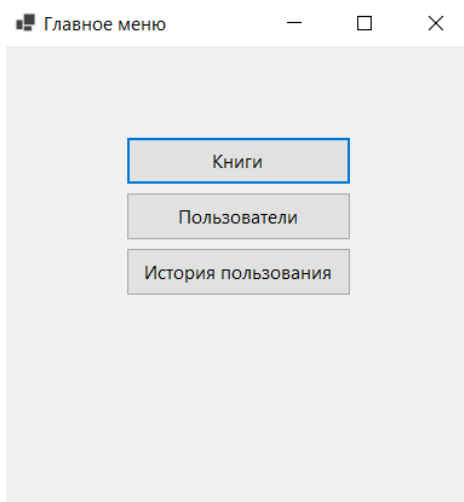


Рисунок 2.5 Окно Главное меню

При выборе кнопки “Книги” пользователь попадает в форму Books.

№	Название	Автор	Жанр	Тип
3	Дон Кихот	Мигель де Серва...		
4	Обломов	Иван Гончаров		
5	Отцы и дети	Иван Тургенев		
6	Евгений Онегин	Александр Пушк...		
7	Война и мир	Лев Толстой		
8	Тихий Дон	Михаил Шолохов		
9	Бедная Лиза	Николай Карамз...	Повесть	

Рисунок 2.6 Работа с таблицей книг

Для добавления в приложение списка книг необходимо было подключить созданную ранее базу данных, содержащую весь необходимый материал. Ниже представлена программная реализация заполнения списка книг из базы данных.

Ссылка: 6

```
public void updateBookListView()
{
    using (sql_command = new SqlCommand("select * from Books", string_con))
    {
        using (reader = sql_command.ExecuteReader())
        {
            ListViewItem tmp;
            listView1.Items.Clear();
            while (reader.HasRows)
            {
                while (reader.Read())
                {
                    tmp = listView1.Items.Add(reader[0].ToString());
                    for (int i = 1; i < 10; i++)
                    {
                        tmp.SubItems.Add(reader[i].ToString());
                    }
                }
                reader.NextResult();
            }
        }
        listView1.View = View.Details;
    }
}
```

Листинг 2.1 Код функции заполнения списка книг из базы данных

Для осуществления в приложении функционала поиска по одному или нескольким параметрам книги из списка необходимо реализовать метод чтения и вывода результата выполнения SQL запроса:

Ссылка: 6

```
public void readQueryResult()
{
    using (reader = sql_command.ExecuteReader())
    {
        ListViewItem tmp;
        listView1.Items.Clear();
        while (reader.HasRows)
        {
            while (reader.Read())
            {
                tmp = listView1.Items.Add(reader[0].ToString());
                for (int i = 1; i < 10; i++)
                {
                    tmp.SubItems.Add(reader[i].ToString());
                }
            }
            reader.NextResult();
        }
    }
}
```

Листинг 2.2 Код вывода результата запроса в список

```
if (textBox1.Text.Length > 0)
{
    using (sql_command = new SqlCommand($"select * from Books where cast (Title as nvarchar(100)) = '{textBox1.Text}'", string_con))
    {
        readQueryResult();
    }
}
```

Листинг 2.3 (Пример) Код, выполняющийся при поиске по названию

Для элементов списка добавлено контекстное меню, с помощью которого мы можем добавлять, изменять и удалять информацию о книгах.

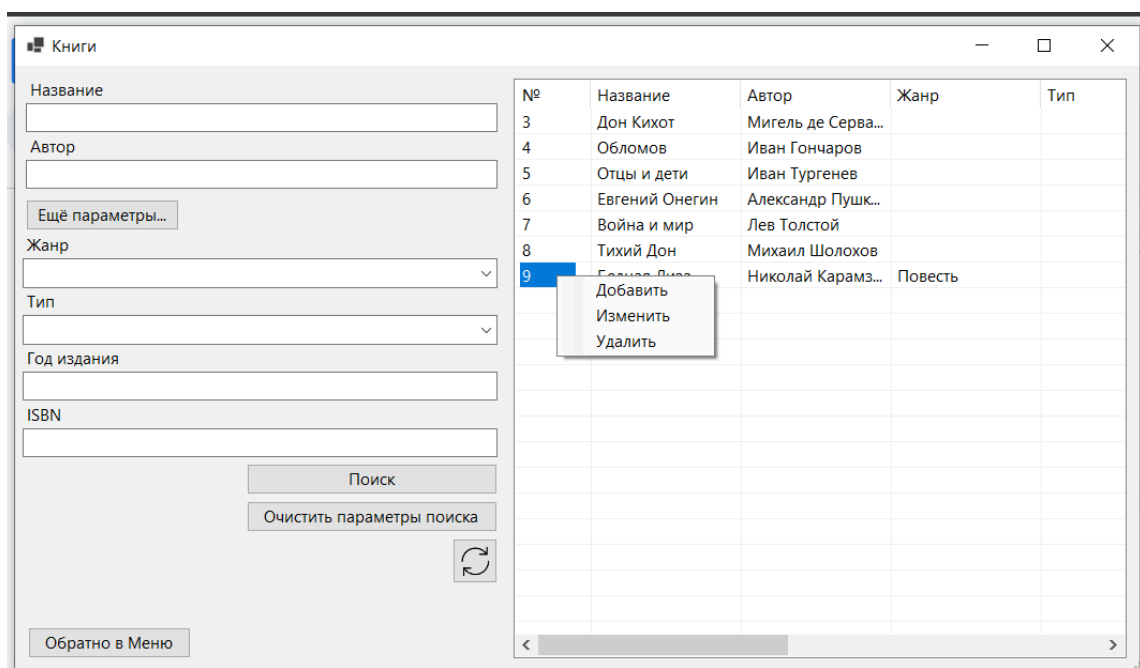


Рисунок 2.7 Контекстное меню таблицы

На листинге 2.3 представлена функция добавления книги в базу данных. Данная функция осуществляет добавление информации о книге в базу данных после нажатия кнопки “Сохранить” при заполнении формы AddBook.

```
int BookID;
string sql;
try
{
    using (string_con = new SqlConnection("Server=X923;Database=LibraryProject1;Trusted_Connection=True;"))
    {
        string_con.Open();
        using (sql_command = new SqlCommand("select top (1) [BookID] FROM [LibraryProject1].[dbo].[Books] order by [BookID] desc", string_con))
        {
            using (reader = sql_command.ExecuteReader())
            {
                while (reader.Read())
                {
                    BookID = Int32.Parse(reader.GetString(0)) + 1;
                    strBookID = BookID.ToString();
                }
            }
            sql = "INSERT INTO Books " +
                "(BookID,Title,Author,Genre,Type,Count,Year,Publisher,ISBN,Summary) " +
                "VALUES ('" + strBookID + "','" +
                textBox1.Text + "','" + textBox2.Text +
                "','" + comboBox1.Text + "','" + comboBox2.Text +
                "','" + textBox5.Text + "','" + textBox6.Text +
                "','" + textBox7.Text + "','" + textBox8.Text +
                "','" + richTextBox1.Text + "')";
        }
        using (sql_command1 = new SqlCommand(sql, string_con))
        {
            sql_command1.ExecuteNonQuery();
        }
        string_con.Close();
    }
}
catch (Exception ex) //catch exeption
{
    MessageBox.Show(ex.Message);
}
this.Close();
```

Листинг 2.3 Код функции добавления книги в базу данных

Рисунок 2.8 Форма для добавления информации о книгах

Изменение записей таблицы Books базы данных осуществлено с помощью двух функций. Сначала необходимо загрузить заполненные поля записи в поля формы, а после реализовать функцию сохранения измененных данных в таблицу. Код этих двух функций представлен в листингах 2.4 и 2.5 соответственно.

```
index = dbIndex;
object tmp;
using (string_con = new SqlConnection("Server=X923;Database=LibraryProject1;Trusted_Connection=True;"))
{
    string_con.Open();
    using (sql_command = new SqlCommand($"select Title,Author,Genre,Type,Year,Publisher,Count,ISBN,Summary from Books where BookID={dbIndex}", string_con))
    {
        using (reader = sql_command.ExecuteReader())
        {
            while (reader.Read())
            {
                if ((tmp = reader.GetValue(0)) != null)
                {
                    textBox1.Text = tmp.ToString();
                }
                if ((tmp = reader.GetValue(1)) != null)
                {
                    textBox2.Text = tmp.ToString();
                }
                if ((tmp = reader.GetValue(2)) != null)
                {

```

Листинг 2.4 Код загрузки информации из базы данных в форму
Форма изменения книги выглядит аналогично форме добавления.

Ссылка: 1

```
private void button1_Click(object sender, EventArgs e)
{
    sql = "UPDATE Books " +
        "SET Title = '" + textBox1.Text + "', Author = '" + textBox2.Text + "', Genre = '" +
        comboBox1.Text + "', Type = '" + comboBox2.Text + "', Count = '" + textBox5.Text +
        "', Year = '" + textBox6.Text + "', Publisher = '" + textBox7.Text + "', ISBN = '" +
        textBox8.Text + "', Summary = '" + richTextBox1.Text + "' WHERE BookID=" + index + ";";
    using (string_con = new SqlConnection("Server=X923;Database=LibraryProject1;Trusted_Connection=True;"))
    {
        string_con.Open();
        using (sql_command = new SqlCommand(sql, string_con))
        {
            sql_command.ExecuteNonQuery();
        }
        string_con.Close();
    }
    this.Close();
}
```

Листинг 2.5 Код сохранения измененной информации в базу данных

Удаление записей из базы данных реализовано через функцию, выполняющую запросы SQL, которые удаляют данные о книге из таблиц Books и

```

using (string_con = new SqlConnection("Server=X923;Database=LibraryProject1;Trusted_Connection=True;"))
{
    if (listView1.SelectedItems.Count > 0)
    {
        ListViewItem selectedItem = listView1.SelectedItems[0];
        using (sql_command = new SqlCommand($"select BookID from Books where BookID={selectedItem.Text}", string_con))
        {
            string_con.Open();
            using (reader = sql_command.ExecuteReader())
            {
                while (reader.Read())
                {
                    temp = reader.GetString(0);
                }
            }
            string_con.Close();
        }
        using (sql_command = new SqlCommand($"delete from EventHistory where Book_ID={temp}", string_con))
        {
            string_con.Open();
            sql_command.ExecuteNonQuery();
            string_con.Close();
        }
        using (sql_command = new SqlCommand($"delete from Books where BookID={temp}", string_con))
        {
            string_con.Open();
            sql_command.ExecuteNonQuery();
            string_con.Close();
        }
    }
    string_con.Open();
    updateBookListView();
    string_con.Close();
}

```

EventHistory по номеру книги BookID.

Листинг 2.6 Код функции удаления книги из базы данных

Работа с формами “Пользователи” и “История пользования” (таблицами Users и EventHistory) организована аналогично: можно осуществить поиск по одному или нескольким параметрам записей таблицы, у элементов таблицы есть контекстное меню, из которого можно получить доступ к добавлению, изменению и удалению записей.

При выборе кнопки “Пользователи” пользователь попадает в форму Users.

№	Логин	ФИО	Телефон
1	12234	Федосеева Елена Сергее...	6832496
3	abs	1345	
4	123	2334	

Рисунок 2.9 Работа с таблицей пользователей

В данном разделе мы можем осуществлять поиск по имени пользователя, его логину и номеру телефона.

При выборе кнопки “История пользования” пользователь попадает в форму EventHistory.

№	Название	Автор	ФИО	Логин
1	Дон Кихот	Мигель де ...	Федосеева Елен...	12234
2	Война и м...	Лев Толстой	Федосеева Елен...	12234
3	Евгений О...	Александр ...	Федосеева Елен...	12234

Рисунок 2.10 Работа с таблицей истории пользования

В данном разделе мы можем добавлять, изменять и удалять информацию о событиях, осуществлять поиск по имени пользователя, его логину и номеру телефона и по названию книги, её автору, жанру, типу, году издания и ISBN.

Форма добавления записей в таблицу “История пользования” была реализована иначе. Для заполнения формы AddEvent необходимо осуществить поиск по пользователю и поиск по книге, данные о которых будут отображаться в таблице при нажатии кнопки “Поиск”. При выборе пункта списка из таблицы данные записи будут занесены в поля формы. В листинге 2.7 представлен код функции сохранения записи события в таблицу.

Рисунок 2.11 Форма добавления события

```

string sql;
int EventID;
try
{
    using (string_con = new SqlConnection("Server=X923;Database=LibraryProject1;Trusted_Connection=True;"))
    {
        string_con.Open();
        using (sql_command = new SqlCommand("select top (1) [EventID] FROM [LibraryProject1].[dbo].[EventHistory] order by [EventID] desc", string_con))
        {
            using (reader = sql_command.ExecuteReader())
            {
                while (reader.Read())
                {
                    EventID = Int32.Parse(reader.GetString(0)) + 1;
                    strEventID = EventID.ToString();
                }
            }
            sql = "INSERT INTO EventHistory " + "(EventID,Book_ID,User_ID,Date,Action) " +
                "VALUES(' " + strEventID + "',' " + strBookID + "',' " + strUserID +
                "',' " + dateTimePicker1.Text + "',' " + comboBox1.Text + "',')";
            using (sql_command1 = new SqlCommand(sql, string_con))
            {
                sql_command1.ExecuteNonQuery();
            }
            string_con.Close();
        }
    }
}
catch (Exception ex) //catch exeption
{
    MessageBox.Show(ex.Message);
}
this.Close();

```

Листинг 2.7 Код сохранения записи в таблицу EventHistory

Таким образом, программа, разработанная в процессе создания данного проекта, действительно является удобной в использовании для библиотекаря.

3. Заключение

По итогу цель данного проекта была достигнута.

Информационная система была реализована с использованием языка запросов SQL и при помощи системы управления базами данных Microsoft SQL Server. За счет чего были получены навыки работы с данными инструментами.

В ходе проекта были получены навыки:

- проектирования баз данных,
- создания десктопного приложения,
- подключения созданной базы данных к приложению.

В результате осуществления данного проекта была создана информационная система для библиотеки, которая позволила упростить процесс учета выдачи книг.

4. Библиографический список

1. С# и Windows Forms [Электронный ресурс]. - <https://metanit.com/sharp/windowsforms/1.1.php>
2. Попов Д.И., Попова Е.Д. Информационные технологии. Базы данных [Текст]: учеб. пособие для студ. вузов /Д.И. Попов. – М: гос. ун-т, печати, 2009. – 117 с.
3. Обзор средств SQL [Электронный ресурс]. - <https://learn.microsoft.com/ru-ru/sql/tools/overview-sql-tools?view=sql-server-ver16>
4. Карпова И.П. Базы данных [Текст]: учеб. пособие для студ. вузов /И.П. Беляев. – М: Московский государственный институт электроники и математики, 2009. – 130 с
5. Справочник по Transact-SQL (ядро СУБД) [Электронный ресурс]. - <https://learn.microsoft.com/ru-ru/sql/t-sql/language-reference?view=sql-server-ver16>
6. Введение в MS SQL Server и T-SQL [Электронный ресурс]. – URL: <https://metanit.com/sql/sqlserver/1.1.php>