# 2.НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

## 2.1. Постановка задачи

Множество людей, читающих бумажные книги, читают их лишь однажды, а затем они занимают место и пылятся на полках. В настоящее время покупка книг является не дешевым удовольствием, а обмен ими задаче не из простых так как сейчас не многие люди читают именно бумажные книги. С решением этим проблем может помочь приложение позволяющее обмениваться книгами с незнакомыми людьми или же продавать их по более выгодной для людей цене нежели покупать новые.

Главной задачей курсовой работы является проектирование и создание приложения с использованием базы данных для реализации возможностей обмена и продажи книг пользователями.

## 2.2. Перечень задач, подлежащих решению в процессе разработки

В процессе выполнения курсовой работы и разработки приложения требуется решить следующие задачи:

исследовать существующие аналоги приложений для обмена книгами для поиска проблем, которым можно будет избежать в проектирование приложения;

разработать базу данных для хранения информации пользователях и данных о книгах;

спроектировать и разработать приложение;

* провести тестирование приложения.

## 2.3 Анализ существующих аналогов

Приложений для обмена именно книгами не так уж и много поэтому стоит рассматривать не только их, но и случаи, когда происходит обмен любыми, вещали.

1) BEE - Book Exchange Easy

Book Exchange Easy — это онлайн-платформа приложений для обмена книгами.

Это приложение позволяет вам делиться книгами в Интернете с читателями, которых вы не обязательно знаете.

Мобильное приложение, обладающее функциями чата и оценки как книг, так и человека, с которым был произведен обмен. Данное приложение не столько платформа для обмена книгами, сколько социальная сеть для знакомства людей с одинаковым предпочтением в книгах для дальнейшего обмена. Главным недостатком этого приложения является региональная ограниченность, оно доступно лишь на территории германии.

2) BasNaBash

«Баш на Баш» — сервис для обмена машинами, личными вещами и бытовой техникой.

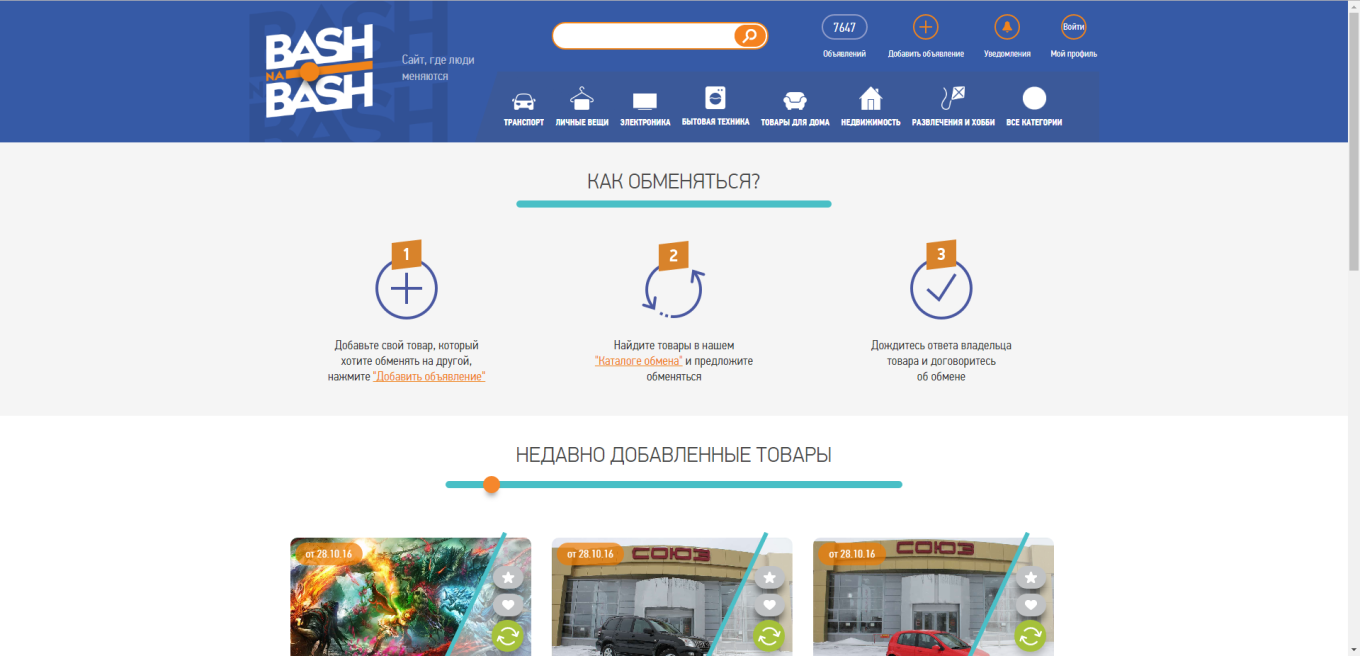


Рисунок 1 – Главная страница «Баш на Баш»

Хоть создатели сервиса и говорят, что он создан для обмена вещами, если деньги считать вещами, то это действительно так, пользователи выставляют вещи и назначают им цену именно обмена не происходит. Интересным решение является вывешивание недавно добавленных вещей на главную страницу, что позволяет заинтересовать пользователей, которые скорее всего и не думали, что подобная вещь им понадобится.

3) Swaper – сервис, позволяющий пользователям размещать вещи, которые они хотели бы обменять и указывать интересные для себя вещи, обмен на которые они готовы произвести.

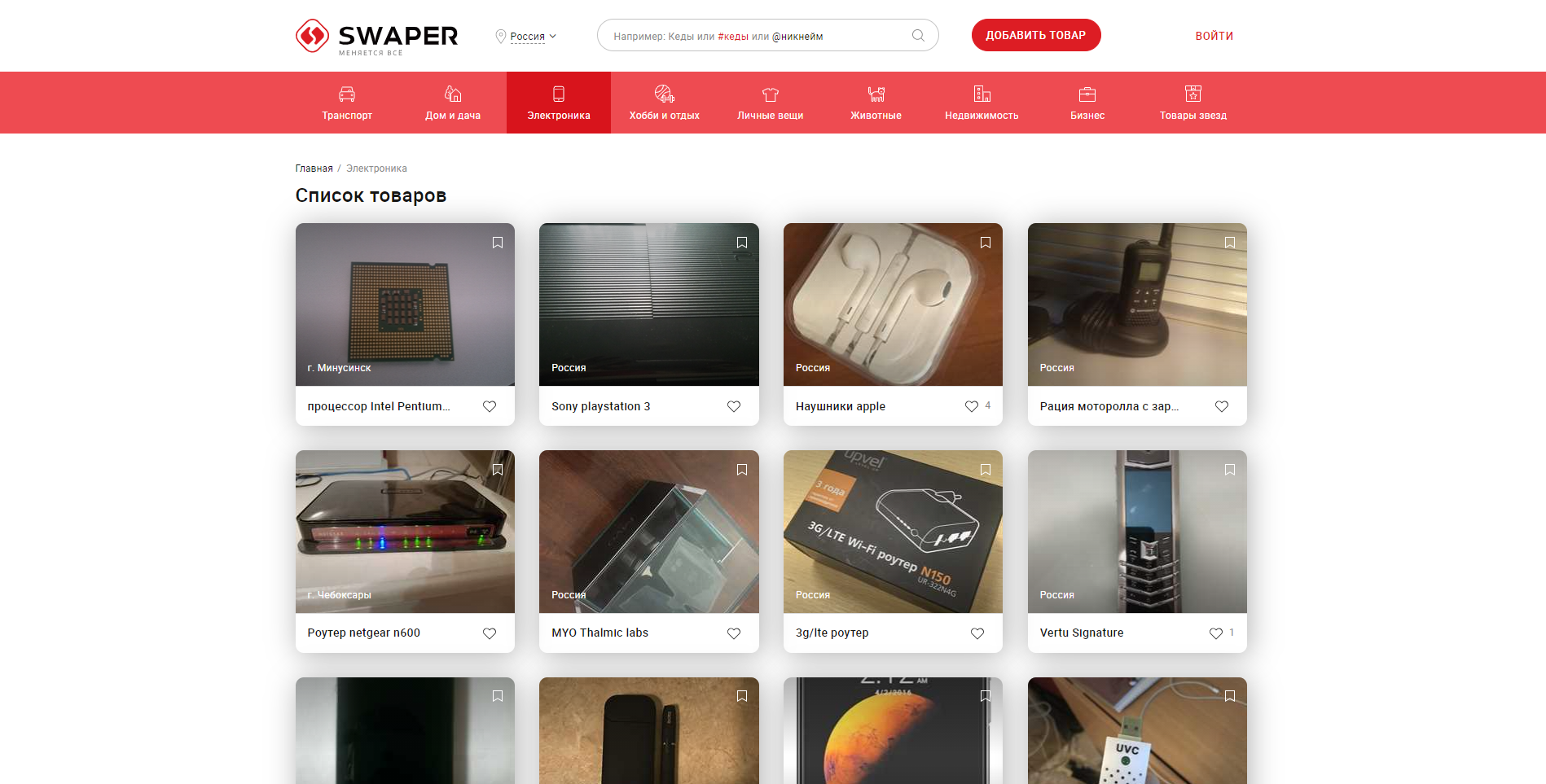
****

Рисунок 2 – Сайт «Swaper»

Сервис позволяет размещать товары, прикладывать фото и описание. Так же существует внутренняя валюта сервиса, упрощающая обмен, но непонятно почему бы не использовать просто деньги для этих целей.

## 2.4. Описание функционала

На основании рассмотренных аналогов можно предположить действие пользователей в системе и реализовать функционал для повышения удобства пользователя добавляя элементы и функции для ускорения или упрощения этих действий.

Для облегчения поиска нужно пользователю книги нужно добавить поиск по названиям, жанрам, авторам.

Реализация ветрины недавно добавленных книг поможет пользователям заинтересоваться книгами, которые до этого им возможно даже не были известны.

Так же для облегчения поиска необходимо реализовать функцию поиска по нажатию на название книги, её жанр или автора.

При выставлении книги на обмен пользователь сможет ввести ISBN книги и данные о ней заполнятся автоматически, если такой книги в базе нет пользователь сможет её добавить.

Администраторы смогут просматривать новые книги, добавленные в базу и исправлять их, с редактирование всех пользовательских книг или их удалением.

## 2.5. Обоснование выбора языка и среды разработки

### 2.5.1. Язык разработки

C# — компилируемый, статически типизированный язык программирования общего назначения.

Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C# сочетает свойства как высокоуровневых, так и низкоуровневых языков. В сравнении с его предшественником — языком C, — наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования.

C# широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр. Существует множество реализаций языка C#, как бесплатных, так и коммерческих и для различных платформ.

### 2.5.2. Среда разработки

Microsoft Visual Studio - полнофункциональная интегрированная среда разработки (IDE) с поддержкой популярных языков программирования, среди которых С, C++, VB.NET, C#, F#, JavaScript, Python.

Функциональность Visual Studio охватывает все этапы разработки программного обеспечения, предоставляя современные инструменты для написания кода, проектирования графических интерфейсов, сборки, отладки и тестирования приложений. Возможности Visual Studio могут быть дополнены путем подключения необходимых расширений.

Редактор кода Visual Studio поддерживает подсветку синтаксиса, вставку фрагментов кода, отображение структуры и связанных функций. Существенно ускорить работу помогает технология IntelliSense - автозавершение кода по мере ввода.

Visual Studio может быть использована для создания и обновления файла базы данных в SQL Server Express, что позволяет упростить работу с базой данных для создания приложения.

Встроенный отладчик Visual Studio используется для поиска и исправления ошибок в исходном коде, в том числе на низком аппаратном уровне. Инструменты диагностики позволяют оценить качество кода с точки зрения производительности и использования памяти.

Дизайнер форм Visual Studio незаменим при разработке программ с графическим интерфейсом, помогая спроектировать внешний вид будущего приложения и работу каждого элемента интерфейса.

Visual Studio предоставляет комплекс инструментов для автоматизации тестирования приложений в части проверки работы интерфейсов, модульного и нагрузочного тестирования.

## 2.6. Перечень задач, подлежащих решению в процессе разработки

Для выполнения поставленной задачи необходимо определить оптимальную архитектуру для построения системы и средства реализации, соответствующего данной архитектуре. В задачи разрабатываемой системы входят разработка базы данных для хранения информации о пользователях и книгах, разработка приложения и его последующее тестирование.

## 2.7. Вывод

На основании анализа аналогов, был выявлен функционал, предстоящий к решению. Было принято решение использовать следующие инструменты: в качестве языка программирования был выбран С#. Для реализации сервиса была выбрана интегрированная среда разработки Microsoft Visual Studio.

# 3.ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ

## 3.1. Общие сведения

Данная курсовая работа представляет собой приложение «BookSharing» для обмена и продажи книг.

## 3.2. Структура приложения

Приложение «BookSharing», представляющее собой оконное приложение с возможностью выставить собственные книги для обмена или продажи, а также найти книги для других пользователей использую поиск или просматривая последние добавленные в приложения книги.

## 3.3. Структура данных

### 3.3.1. Концептуальная модель данных

Для облегчения разработки приложения требуется спланировать структуру данных для базы данных приложения.

При проектирование концептуальной модели данных были выделены сущности: Книги, Жанры, Авторы, Пользовательские книги, Пользователи и Отчеты.

Сущность Книги служит для хранения книг служащих основой для заполнения пользовательских книг.

Сущности Жанр и Автор является справочной для заполнения книг в базе. Эти сущности будут использоваться для заполнения полей для поиска по жанру и автору.

Сущность Пользователи необходима для хранения данных о пользователе.

Сущность Пользовательские книги нужна для создания книг пользователей, которые они выставляют на обмен. Эти книги будут отображаться на странице книг самого пользователя, а также появляться на главной странице и отображается в результатах поиска.

Сущность Отчеты нужна для хранения отчетов о создании или изменении книг. Отчет создается автоматически, когда создается или изменяется пользовательская книга. Созданные отчеты будут отображается на странице администратора для проверки и изменения или удаления книг, по которым создан этот отчет.

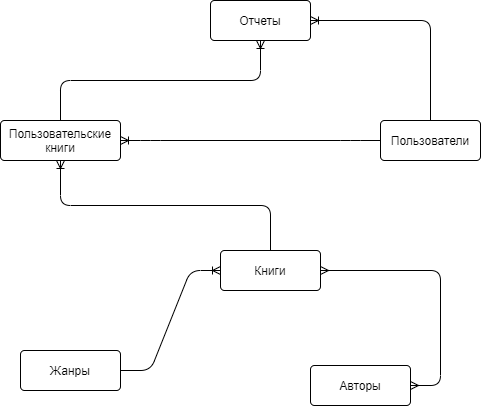


Рисунок 3 – Концептуальная модель данных

### 3.3.2. Логическая модель данных

На этапе проектирования логической схемы данных были определены атрибуты сущностей, а также добавлена сущность АвторКнига.

Атрибуты сущности Пользователи логин и пароль необходимы для хранения логина и пароля пользователя чтобы они могли входить в систему. Атрибут email для отображения электронной почты на страницах книг других пользователей чтобы связаться с ними. Атрибут права необходимо для определения простой это пользователь или администратор и в зависимости от этого открывать определенные страницы приложения.

Сущность Книги содержит атрибуты ISBN, название и жанр для хранения соответствующих данных. Данная сущность связывается с сущностью АвторКнига для определения авторов, написавших эту книгу.

Сущность АвторКнига содержит атрибуты книга и автор для соотнесения книг и написавших их авторов, а также атрибут код, единично идентифицирующий эту связь.

Сущности Жанр является справочной и у нее лишь один атрибут, название.

Сущность Автор, как и сущность Книги связывается с сущность АвторКнига и имеет единственный атрибут имя.

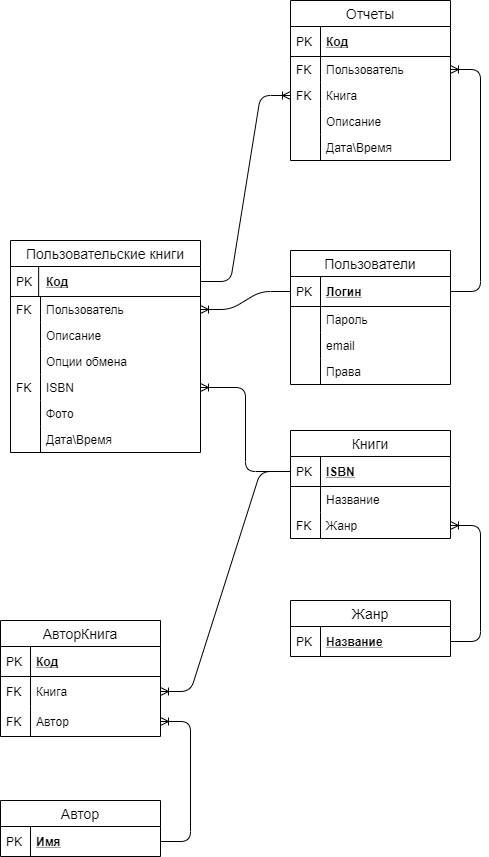


Рисунок 4 – Логическая модель данных

Атрибуты сущности пользовательские книги пользователь и ISBN соответственно связываются с сущностями Пользователь и Книги. Атрибут описание хранит описание книги, написанное пользователем и отображаемое как на общих страницах, так и при просматривании полной информации о книгах. Атрибут опции обмена содержит информацию, написанную пользователем при создании пользовательской книги, и содержит информации о том, на что он хотел бы обменять выставленную книги, которая отображается при просмотре полной информации о книге. Атрибут фото нужен для хранения пути к изображению книги, загруженному пользователем. Атрибут дата/время необходим для хранения даты и времени в которые книга была добавлена в базу.

Сущность Отчеты имеет атрибуты книга и пользователь связанные соответственно с сущностями пользователь и пользовательское книги, а также атрибут описание для хранения описания отчета и атрибут дата/время ля хранения даты и времени в которые отчет был сформирован.

### 3.3.3. Физическая модель данных

На основании разработанной логической модели реализуем физическую модель данных.

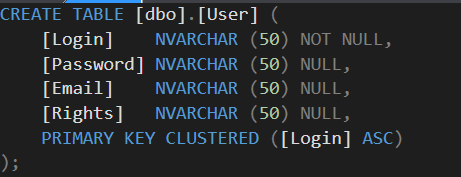


Рисунок 5 – Код формирования таблицы User

Таблица User созданная на основании сущности Пользователи имеет четыре колонки: Login, Password, Email, Rights типа NVARCHAR (50) для хранения текстовой информации. Столбец Login не может быть пустым так как является первичным ключом служащим для связи данной таблицы с другими и сохранения уникальности строк данной таблицы.

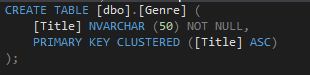


Рисунок 6 – Код формирования таблицы Genre

Таблица Genre созданная на основании сущности Жанр имеет единственный столбец Title тип NVARCHAR (50) служащий для хранения наименования жанра и являющийся первичным ключом служащим для связи данной таблицы с другими и единичной идентификации строк данной таблицы.

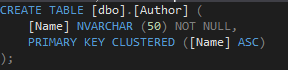


Рисунок 7 – Код формирования таблицы Author

Таблица Author созданная на основании сущности Автор имеет единственный столбец Name тип NVARCHAR (50) служащий для хранения имени автора и являющийся первичным ключом служащим для связи данной таблицы с другими и сохранения уникальности строк данной таблицы.

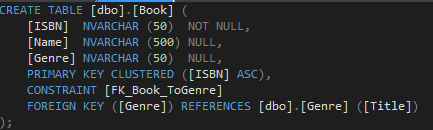


Рисунок 8 – Код формирования таблицы Book

Таблица Book созданная на основании сущности Книги имеет три колонки: ISBN, Genre типа NVARCHAR (50) для хранения текстовой информации и Name типа NVARCHAR (500) для хранения названий книг, которые могут быть достаточно длинными. Столбец ISBN не может быть пустым так как является первичным ключом служащим для связи данной таблицы с другими и сохранения уникальности строк данной таблицы. Вторичным ключом для связи этой таблицы с таблицей Genre является столбец Genre, данные в нем должны соответствовать данным из столбца Title для сохранения целостности.

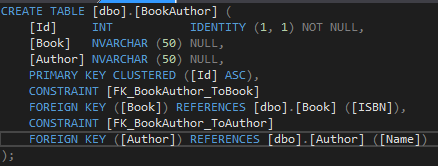


Рисунок 8 – Код формирования таблицы BookAuthor

Таблица BookAuthor созданная на основании сущности АвторКнига имеет три колонки: Book, Author типа NVARCHAR (50) для хранения текстовой информации и Id типа INT являющийся первичным ключом, служащим для связи данной таблицы с другими и сохранения уникальности строк данной таблицы, а так же имеющим свойство идентификатора автоматически присваивающим данном столбцу значения про создании новой строки. Колонки Book и Author являются вторичными ключами соответственно связывающимися с столбцами ISBN из таблицы Book и Name из таблицы Author. Данные, хранящиеся в данных столбцах, должны соответствовать данным из связанных таблиц для сохранения целостности.

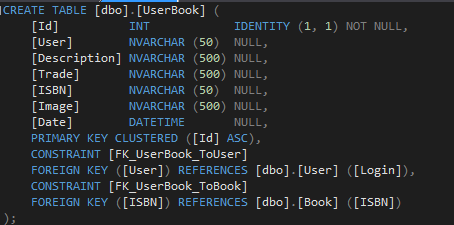


Рисунок 9 – Код формирования таблицы UserBook

Таблица UserBook созданная на основании сущности Пользовательские книги имеет семь колонок: User и ISBN типа NVARCHAR (50) для хранения текстовой информации, Description и Trade типа NVARCHAR (500) для хранения объемной текстовой информации, Image типа NVARCHAR (500) для хранения пути к изображению, Date типа DATETIME для хранения даты и времени и Id типа INT являющийся первичным ключом, служащим для связи данной таблицы с другими и сохранения уникальности строк данной таблицы, а так же имеющим свойство идентификатора автоматически присваивающим данном столбцу значения про создании новой строки. Колонки ISBN и User являются вторичными ключами соответственно связывающимися с столбцами ISBN из таблицы Book и Login из таблицы User. Данные, хранящиеся в данных столбцах, должны соответствовать данным из связанных таблиц для сохранения целостности.

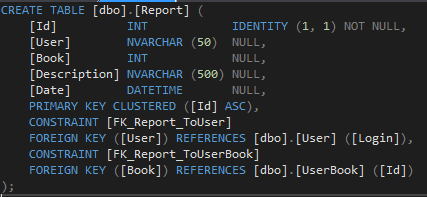


Рисунок 10 – Код формирования таблицы Report

Таблица Report созданная на основании сущности Отчеты имеет пять колонок: User типа NVARCHAR (50) для хранения текстовой информации, Description NVARCHAR (500) для хранения объемной текстовой информации, Date типа DATETIME для хранения даты и времени, Book типа INT для хранения id пользовательской книги и Id типа INT являющийся первичным ключом, служащим для связи данной таблицы с другими и сохранения уникальности строк данной таблицы, а так же имеющим свойство идентификатора автоматически присваивающим данном столбцу значения про создании новой строки. Колонки Book и User являются вторичными ключами соответственно связывающимися с столбцами Id из таблицы UserBook и Login из таблицы User. Данные, хранящиеся в данных столбцах, должны соответствовать данным из связанных таблиц для сохранения целостности.

## 3.4 Вывод

В ходе конструкторской части курсовой работы была определена структура приложения, которое будет выполнено в виде оконного приложения с возможностью выставить собственные книги для обмена или продажи, а также найти книги для других пользователей использую поиск или просматривая последние добавленные в приложения книги.

Так же в ходе работы были разработаны концептуальная и логическая модели данных и на их основании была реализована физическая модель данных.

# 4.Проектно-технологическая часть

## 4.1 Требования к аппаратной платформе.

Исходя из характеристик разработанного приложения и ресурсов, требуемых для его исправной работы необходимо, чтобы компьютер обладал следующими минимальными характеристиками:

* процессор: 1 ГГц и быстрее с поддержкой PAE, NX и SSE2;
* RAM: 1 Гбайт (32 бит) или 2 Гбайт (64 бит);
* HDD: 16 Гбайт (32 бит) или 20 Гбайт (64 бит);
* видеокарта: поддержка Microsoft DirectX 9 с драйвером WDDM;
* монитор;
* клавиатура;
* мышь.