МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМЕНИ К.Г. РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ УНИВЕРСИТЕТ)»  
(ФГБОУ ВО «МГУТУ ИМ. К.Г. РАЗУМОВСКОГО (ПКУ)»)  
УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Специальность 09.02.03 Программирование в компьютерных системах

КУРСОВОЙ ПРОЕКТ

МДК 02. Разработка и администрирование баз данных

МДК 02.02. Технология разработки и защиты баз данных

на тему: «Разработка системы автоматизации учета книг в библиотеке.»

Пояснительная записка

УКИТ.09.02.03.2019.403.18ПЗ

ПКС 305

|  |  |  |
| --- | --- | --- |
| Исполнитель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ (подпись) | Вартанян А. П. |
| Руководитель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ (подпись) | Ларионова Е. А. |
| Норм контроль | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ (подпись) | Гусева Е. Л. |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc27305809)

[1 ОБЩАЯ ЧАСТЬ 5](#_Toc27305810)

[1.1 Назначения и цели создания 5](#_Toc27305811)

[1.2 Обзор и анализ предметной области 5](#_Toc27305812)

[1.3 Жизненный цикл базы данных 7](#_Toc27305813)

[1.4 Выбор и характеристика СУБД 8](#_Toc27305814)

[1.5 Выбор и характеристика среды разработки приложения 9](#_Toc27305815)

[2 СПЕЦИАЛЬНАЯ ЧАСТЬ 11](#_Toc27305816)

[2.1 Определение основных задач, приводящих к цели 11](#_Toc27305817)

[2.2 Логическая модель базы данных 11](#_Toc27305819)

[2.3 Нормализация таблиц 11](#_Toc27305820)

[2.4 Описание таблиц 12](#_Toc27305821)

[2.5 Другие объекты базы данных 15](#_Toc27305822)

[2.6 Разработка приложения 15](#_Toc27305823)

[2.7 Инструкция пользователю 16](#_Toc27305824)

[ЗАКЛЮЧЕНИЕ 18](#_Toc27305825)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 19](#_Toc27305826)

[ПРИЛОЖЕНИЕ А](#_Toc27305827) [SQL скрипты 21](#_Toc27305828)

[ПРИЛОЖЕНИЕ Б](#_Toc27305829) [Код программы 30](#_Toc27305830)

ВВЕДЕНИЕ

Была проанализирована предметная область, исходя из этого определены объекты и их связи. На основе концепции спроектирована и реализована физическая база в конкретной СУБД и были заполнены некоторые статические, но дополняемые данные.

Выбор СУБД и среды разработки были сделаны исходя из стандартов, которые используются во всероссийском конкурсе «Ворлдскиллс», стандарты которого еще и распространяются на квалификационные работы. Таким образом, произведена подготовка к квалификационным работам.

Существует множество систем для хранения истории просмотров фильмов. Каждая из них имеет различный интерфейс, но функционально они очень схожи. Для закрепления курса по разработке и защите баз данных был повторен опыт уже существующих систем и разработана система ведения истории использования книг.

При разработке базы данных была произведена нормализация таблиц для предотвращения дублирования данных и предотвращения транзитивных зависимостей, что делает систему более гибкой и расширяемой. Для введения нового функционала новые объекты привязываются к уже существующим.

Концептуально удаление использования книг и книг не предусмотрено системой, так как система по истории использования и все попытки изменить историю использования пресекаются.

На основе базы данных было разработано компьютерное приложение. Оно позволяет добавлять книги и делать записи об их использование. Помимо этого, можно просматривать рейтинг того или иного произведения исходя из оценки прочтения книг пользователями.

К компьютерному приложению написаны инструкции пользователю с требования к техническим и программным средствам, и по установке и настройке системы. Для объяснения интерфейса продемонстрированы формы ввода и сделан отчет по разработанной системе.

1. ОБЩАЯ ЧАСТЬ
   1. Назначения и цели создания

База данных книг в библиотеке предназначена для хранение пользовательских данных и расчета оценки по книги для привлечения внимания людей к другим книгам с определенным рейтингом.

* + 1. Цель создания системы

Основной целью является получение опыта в разработке и защите систем, в основе которых лежит база данных.

* 1. Обзор и анализ предметной области
     1. Аннотация

Невозможно представить крупную систему, которая не использовала бы базы данных для хранения информации. Сервисы по истории представляют собой отражение базы данных для клиента в удобном виде с дополнительным клиентским функционалом. Почему не может заменить систему по хранению информации таблицы exel, какой функционал имеется в уже реализованной системе и многое другое описано в данном разделе.

* + 1. Анализ системы litres

Система litres – пример хорошей реализации базы данных хранения истории использования книг, а также выборке книг для просмотров. Основное меню состоит из обложек книг, рейтингу, возможности поставить рейтинг и сортировки книги (Рисунок 1). Данный сервис имеет большую базу книг. Для реализации подобного функционала необходимо задействовать больше времени, чем на курсовой проект. Также подобные проекты разрабатываются не в одиночку и не силами одних программистов, а используется целый ряд специалистов из различных областей

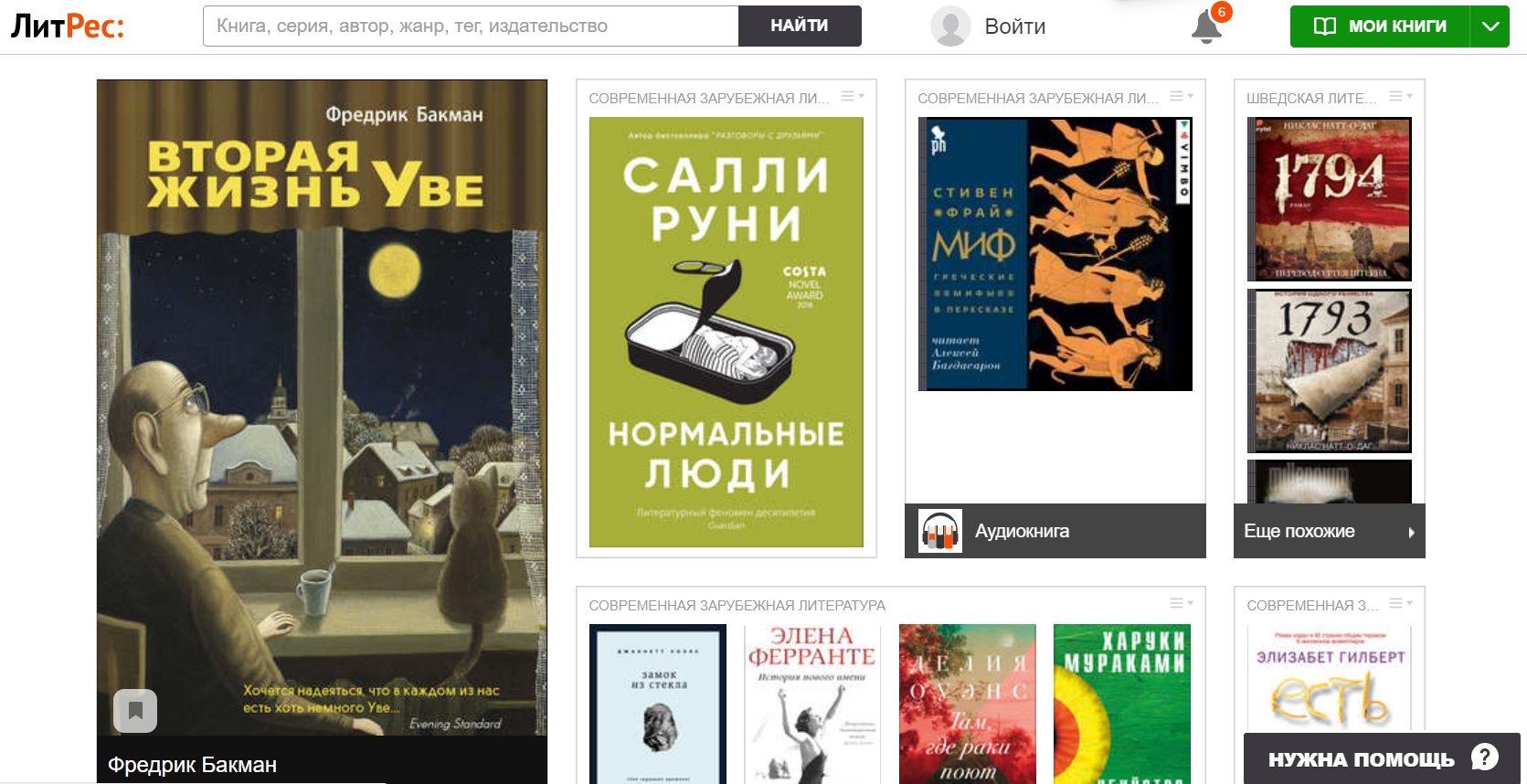


Рисунок 1 – Система litres

Данная система закрепилась на рынке и является примером системы для реализации курсового проекта.

* + 1. Анализ ведения истории просмотров в exсel

Excel представляет собой обычные таблицы, в которых может храниться и анализироваться информация. Для системы же необходимы не просто таблицы, а еще и их связи.

Весь необходимый функционал может быть реализован на макросах, а запись в таблицы и проверка данных через внешнее приложение. Но таким образом это будет реализация СУБД, что не является темой курсового проекта.

В excel нет возможности выборки данных, вставки данных с множеством проверок по типам и значения, а также по связям. Таким образом, можно сделать вывод, что для реализации курсового проекта exel не подходит.

* + 1. Access для создания систем

Access является полноценной СУБД. Также он позволяет создавать клиентский интерфейс, автоматизированные формы добавления информации по таблицам с проверками.

Данное решение является идеальным для систем, работающих в локальной сети или на отдельном компьютере. Но, когда речь идет о высоконагруженной и защищенной системе, то access занимает последнюю позицию в выборе СУБД, так как он не рассчитан для крупных и глобальных систем, что приведет к их не корректной работе.

* + 1. Выводы по анализируемым системам

Был выполнен анализ инструментов, на основе которых можно создать курсовой проект. Отличным вариантом является access, так как имеет весь необходимый функционал. Плохим же вариантом является excel, так как в нем нет необходимого для системы функционала – возможностей СУБД.

Также была рассмотрена система-сервис litres, который является полностью реализованной системой. Разумеется, не может идти речи о системе, которая хоть как-то будет конкурировать с данным сервисом, так как это не является целью курсового проекта.

* + 1. Принципы работы систем хранения информации по прочтению книг.

В подобных системах существует некоторая база книг. Данная база, как правило, заполняется автоматически по api, представленными системами хранения книг.

Также существуют сами пользователи, которые читают книгу, ставят оценку и комментарий.

Весь остальной функционал – лишь дополнения. Ядро же состоит из двух составляющих: книги и зрителей.

* 1. Жизненный цикл базы данных

-Анализ

-Проектирование

-Реализация

-Тестирование

-Эксплуатация и сопровождение

1.3.1 Выделение основных объектов

Можно выделить два основных объекта:

* пользователь,
* книга,

Данные объекты имеют в себе поля и связанные таблицы, которые определяются предметной областью. Например, для одной предметной области нет смысла расписывать подробно всех героев книги и производить связь героев и книг. Такой предметной областью является система истории прочтения книг.

* + 1. Определение связей объектов

Пользователи и книга связываются в таблице прочтения. Данная таблица является не просто реализацией связи многие ко многим, а также является полноценным объектом, так как содержит уникальные поля:

* оценка книг,
* комментарий к книге.
  1. Выбор и характеристика СУБД
     1. Аннотация

В данном разделе производится анализ различных СУБД. Для реализации базы данных отлично подходит access, но какая СУБД в итоге выбрана и почему не access описано в данном разделе.

* + 1. Выбор СУБД

В качестве СУБД отлично подходит access. В данной СУБД есть возможность реализации полной системы, используя графический интерфейс, но данная СУБД не подходит для данного курсового проекта, так как не позволит получить опыт в составлении запросов и практики создания систем с двухуровневой архитектуры.

Следовательно, выбор падает на ms sql сервер. Данная СУБД имеет больший серверный функционал, но уже отсутствует реализации создания клиентского функционала используя лишь ms sql сервер.

* + 1. Обоснование выбора СУБД

Ms sql сервер выбран по следующим причинам:

* Используется для квалификационной работы,
* Прост в установке и настройке под Microsoft windows.

В принципе, могла бы подойти любая другая СУБД, но они не удовлетворяют данным пунктам.

* 1. Выбор и характеристика среды разработки приложения
     1. Аннотация

В данном разделе производится выбор сред разработки приложения и базы данных, а также производится его обоснование.

* + 1. Выбор системы программирования
       1. База данных

Для создания базы данных подходит ssms. Оно позволяет в графическом режиме создавать базы данных, таблицы, настраивать связи и генерировать скрипты создания, удаления и резервного копирования.

В данной среде есть множество ошибок, но также есть и пути, следуя которым, данные ошибки не будут вызваны и произведена корректная работа.

* + - 1. Приложение

Для разработки приложения подходит .NET платформа и среда разработки visual studio. Они позволяют создавать за короткий срок работающие приложения-прототипы под windows. Также он используется в квалификационной работе.

* + 1. Сравнение с другими системами программирования
       1. База данных

Наиболее удобной является среда разработки mySql workbench, но оно подходит для mysql сервера. Эти СУБД имеют множество различий в типах данных, так что данная среда не подходит.

Существует также Erwin, но в почти свободном доступе есть его старая версия, которая прихотлива к кодировкам, выставленным в системе и может не корректно работать при смене кодировок.

* + - 1. Приложение

Очень удобным является язык программирования php и среда разработки phpStorm. Данный язык имеет неплохую оптимизацию с 7й версии и поддержку. Преимущества по php сравнению с c#:

* кроссплатформенность,
* более высокая скорость работы (не используется прослойка – виртуальная машина),
* больше возможностей в качестве web backend (c# универсалтный, а php затачивался как язык для работы backend веба)

Но все преимущества меркнут в одном основном преимуществе c# - он используется для квалификационной работы.

* + 1. Выводы по выбранной системе программирования

Выбранные средства – ssms и visual studio напрямую зависят не от удобства, а от принятых стандартов и целей создания курсового проекта. Нет сомнений, что написание курсового проекта в данных средах значительно повысят уровень подготовки для квалификационной работы.

1. СПЕЦИАЛЬНАЯ ЧАСТЬ
   1. Определение основных задач, приводящих к цели

Для закрепления навыков, полученных в ходе курса предмета «технология разработки и защиты баз данных» в рамках курсовой темы необходимо выполнить следующие задачи:

* Разработать базу данных книг в библиотеки в третьей нормальной форме,
* Настроить пользователей баз данных,
* Сделать простую защиту от sql-иъекций.

В результате выполнения этих работ, будет получен необходимый опыт в проектировании, создании и защите баз данных.

* 1. Логическая модель базы данных

Система имеет следующую логическую модель (Рисунок 3)



Рисунок 2 – Логическая модель базы данных

* 1. Нормализация таблиц
     1. Физическая модель базы данных

На физической модели базы данных (Рисунок 4) проще определить и обосновать нормализованность таблиц.

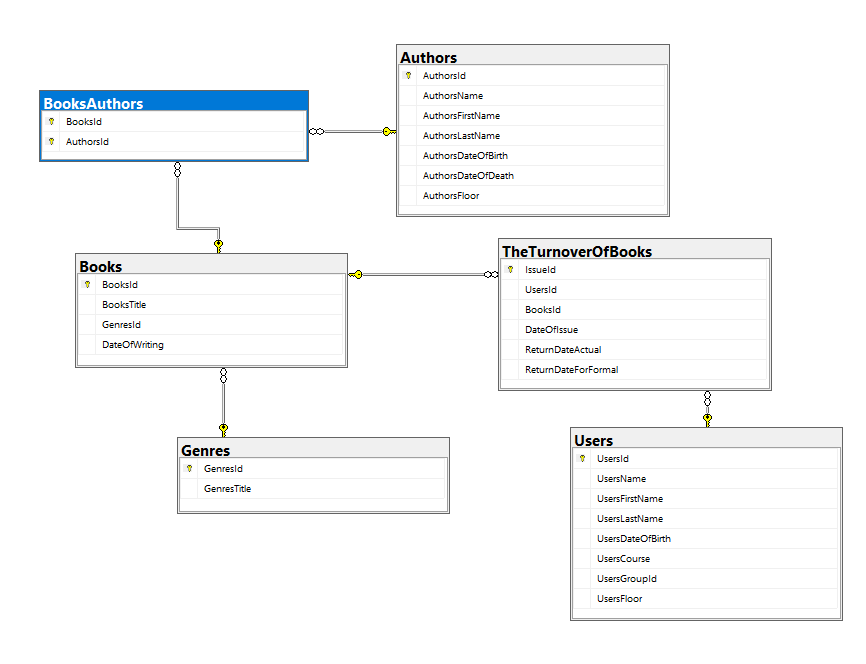


Рисунок 3– Физическая модель базы данных

* 1. Описание таблиц

Таблица1 – «Author»(Автор)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| Authors\_Id | Номер автора | Int | PK |
| Authors\_Name | Имя автора | Nvarchar(30) |  |
| Authors\_First\_Name | Фамилия автора | Nvarchar(20) |  |
| Authors\_Date\_Of\_Birth | Дата рождения автора | Date |  |
| Authors\_Date\_Of\_Death | Дата смерти автора | Date |  |
| Authors\_Floor | Пол автора | Nvarchar(20) |  |

Таблица «Book» (Книг)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| Books\_Id | Номер книги | Int | PK |
| Books\_Title | Название книги | Nvarchar(30) |  |
| Date\_Of\_Writing | Дата написания | Date |  |
| GenresId | Номер жанра | Int | FK |

Таблица 2 – «BookAuthor»(Автор Книг)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| Authors\_Id | Номер автора | Int | PK |
| Books\_Id | Номер книги | Int | PK |

Таблица 3 – «Users»(пользователи)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| Users\_Id | Номер пользователя | Int | PK |
| Users\_Name | Имя пользователя | Nvarchar(30) |  |
| Users\_First\_Name | Отчество пользователя | Nvarchar(20) |  |
| Users\_Last\_Name | Фамилия пользователя | Nvarchar(20) |  |
| Users\_Date\_Of\_Birth | Дата рождения | Date |  |
| Users\_Floor | Пол пользователя | Nvarchar(5) |  |

Таблица 4 – «TheTurnoverOfBooks»(Оборот книг)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| Issu\_Id | Номер выпуска | Int | PK |
| Books\_Id | Номер книги | Int |  |
| Users\_Id | Номер пользователя | Int |  |
| Date -Of\_Issue | Дата выпуска | Date |  |
| Return\_Date\_Actual | Фактическая дата | Date |  |
| Return\_Date\_For\_Formal | Формальная дата | Date |  |

Таблица 5 – «Genres»(Жанры)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| Genres\_Id | Номер жанра | Int | PK |
| Genres\_Title | Название жанра | Nvarchar(30) |  |

* 1. Другие объекты базы данных
     1. Описание представлений

Для реализации курсового проекта были созданы представления для получения книг и рейтинга по книгам.

* 1. Разработка приложения
     1. Структурная схема приложения

Структурная схема приложения отражена в пункте 2.2.

* + 1. Функции приложения

Приложение имеет следующие функции:

* Вести учет книг в библиотеке колледжа
* Производить регистрацию новых изданий.
* Осуществлять поиск литературы:
* по тематике;
* по автору;
* по дате выпуска;
* Осуществлять списание книг из библиотеки.
* Выдавать статистику о состоянии фонда учебной литературы в виде:
* общее количество книг в библиотеке:
* жанры книг,
* количество книг по каждому жанру.
* Просматривать личную карточку конкретного читателя.
* Регистрировать книги читателя в личной карточке.
* Осуществлять поиск задолжников.
* Просматривать список задолжников.
  + 1. Решение главной задачи проекта

Главной задачей проекта является обеспечение возможности людям вести историю прочитанных книг. Так как приложение является переносимым и может не иметь выхода в сеть, то была добавлена возможность создавать книги. Для обеспечения возможности людям вести историю прочитанных книг была создана соответствующая функция, которая выполняется после выбора и ввода нужных компонентов и нажатия на кнопку «прочитано».

* + 1. Описание процесса отладки и тестирования приложения

Процесс отладки происходил в реальном времени, без использования специальных автоматических сценариев тестирования.

* + 1. Защита информационной системы

Для обеспечения корректного ввода в поля в системе присутствует валидация данных. Она обеспечивает защиту от некорректного ввода. В больших системах валидация реализована в виде пользовательских типов данных, которые вызывают исключения при вводе неправильных данных.

Для защиты базы данных используется защита от sql-инъекций посредством обычной проверки. Данная защита является достаточной в рамках курсового проекта.

* 1. Инструкция пользователю
     1. Общее сведение об информационной системе

Информационная система позволяет вести историю прочитанных книг для зарегистрированных в ней пользователей.

* + 1. Требования к программным средствам

Для работы приложения необходимо следующее программное обеспечение:

* установленная платформа «.Net»,
* установленный ms sql сервер.
  + 1. Формы ввода

Вкладка авторизации позволяет создать аккаунт, зарегистрироваться, сменить пароль аккаунта.

Во вкладка истории пользователь может просмотреть историю прочитанных книг, произвести поиск по своей библиотеке и добавить запись о том, что прочитал конкретную книгу еще раз.

* + 1. Отчеты

Данная система генераций отчетов не имеет.

ЗАКЛЮЧЕНИЕ

В процессе выполнения курсового проекта:

* проанализирована предметная область,
* произведен и обоснован подбор программных средств,
* составлена концептуальная модель системы,
* создана логическая модель базы данных,
* создана физическая модель базы данных и реализована в ms sql сервере,
* обоснована третья форма одной из таблиц,
* расписаны таблицы базы данных с русским обозначением,
* произведена настройка ms sql сервера,
* разработано приложение,
* написана инструкция пользователю.

Цели, поставленные в курсовой работе, были полностью достигнуты. Параллельно с курсовой работой была произведена подготовка к квалификационной работе.

Реализованная система не может являться конкурентом уже существующим системам, но она является отличным опытом в разработке систем, основанных на клиент-серверной архитектуре.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. База знаний о взаимодействии с ms sql посредством c#. (2016). Web: https://o7planning.org/ru/10515/working-with-sql-server-database-using-csharp (дата обращения 15.11.2018).
2. Rob Miles. C# Programming Yellow Book. – М.: DCSUH. 2015.
3. База знаний о создании баз данных ms sql server. (2017). Web: https://docs.microsoft.com/ru-ru/sql/relational-databases/databases/create-a database?-View-sql-server-2016 (дата обращения 18.11.2018).
4. Артеменко Ю.Н. C# 7.0. Карманный справочник. – М.: «Альфа‑Книга». 2017.
5. База знаний работы с планировщиком задач в Windows. Web: https://public-pc.com/rabota-s-planirovshhikom-zadaniy-v-windows-10/ (дата обращения 20.11.2018).
6. База знаний о создании диаграмм в ms sql server. (2018). Web: https://remotehelper.ru/subd/ms-sql-server/ms-sql-server-kak-sozdat-diagrammu-bd-so-svyazyami (дата обращения 21.11.2018).
7. База знаний о ролях ms sql server. (2015). Web: https://professorweb.ru/my/sql-server/2012/level3/3\_12.php (дата обращения 23.11.2018).
8. База знаний об ограничении целостности данных ms sql server. (2016). Web: http://www.intuit.ru/studies/courses/5/5/lecture/138?page=2 (дата обращения 09.12.2018).
9. Э. Таненбаум, Д. Уэзеролл – «Компьютерные сети» 5е издание 2017 год.
10. В. Олифер, Н. Олифер – «Компьютерные сети. Принципы, технологии, протоколы. Учебник» 2016 год.
11. Алена, Игоревна Знароченкова Межпоколенноое наследование паттернов супружеского поведения / Алена Игоревна Знароченкова. - М.: LAP Lambert Academic Publishing, 2011.
12. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, и др.. - М.: СПб: Питер, 2001.
13. Гриндер, Джон Паттерны гипнотических техник Милтона Эриксона / Джон Гриндер, Ричард Бэндлер , Джудит Делозье. - М.: Прайм-Еврознак, 2007.
14. Гэри, Маклин Холл Адаптивный код. Гибкое кодирование с помощью паттернов проектирования и принципов SOLID. Руководство / Гэри Маклин Холл. - М.: Альфа-книга, 2017.
15. Демарко, Т. Балдеющие от адреналина и зомбированные шаблонами. Паттерны поведения проектных команд / Т. Демарко. - М.: Символ-плюс, 2016.
16. Пит браун. «Silverlight практическое руководство» - М: Прайм-Еврознак, 20012

**Приложение А**

USE [master]

GO

/\*\*\*\*\*\* Object: Database [libbrary] Script Date: 03.02.2021 18:15:21 \*\*\*\*\*\*/

CREATE DATABASE [libbrary]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'libbrarydata', FILENAME = N'c:\public\ libbrarydata.mdf' , SIZE = 8192KB , MAXSIZE = 102400KB , FILEGROWTH = 10240KB )

LOG ON

( NAME = N'libbrarylog', FILENAME = N'c:\public\libbrarylog.ldf' , SIZE = 3072KB , MAXSIZE = 102400KB , FILEGROWTH = 10240KB )

GO

ALTER DATABASE [libbrary] SET COMPATIBILITY\_LEVEL = 140

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [libbrary].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [libbrary] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [libbrary] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [libbrary] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [libbrary] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [libbrary] SET ARITHABORT OFF

GO

ALTER DATABASE [libbrary] SET AUTO\_CLOSE ON

GO

ALTER DATABASE [libbrary] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [libbrary] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [libbrary] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [libbrary] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [libbrary] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [libbrary] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [libbrary] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [libbrary] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [libbrary] SET ENABLE\_BROKER

GO

ALTER DATABASE [libbrary] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [libbrary] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [libbrary] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [libbrary] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [libbrary] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [libbrary] SET READ\_COMMITTED\_SNAPSHOT OFF

GO

ALTER DATABASE [libbrary] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [libbrary] SET RECOVERY SIMPLE

GO

ALTER DATABASE [libbrary] SET MULTI\_USER

GO

ALTER DATABASE [libbrary] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [libbrary] SET DB\_CHAINING OFF

GO

ALTER DATABASE [libbrary] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [libbrary] SET TARGET\_RECOVERY\_TIME = 60 SECONDS

GO

ALTER DATABASE [libbrary] SET DELAYED\_DURABILITY = DISABLED

GO

ALTER DATABASE [libbrary] SET QUERY\_STORE = OFF

GO

USE [libbrary]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Authors] Script Date: 03.02.2021 18:15:21 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Authors](

[AuthorsId] [int] NOT NULL,

[AuthorsName] [nvarchar](30) NOT NULL,

[AuthorsFirstName] [nvarchar](20) NULL,

[AuthorsLastName] [nvarchar](20) NOT NULL,

[AuthorsDateOfBirth] [date] NOT NULL,

[AuthorsDateOfDeath] [date] NULL,

[AuthorsFloor] [nvarchar](20) NOT NULL,

PRIMARY KEY CLUSTERED

(

[AuthorsId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Books] Script Date: 03.02.2021 18:15:21 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Books](

[BooksId] [int] NOT NULL,

[BooksTitle] [nvarchar](30) NOT NULL,

[GenresId] [int] NOT NULL,

[DateOfWriting] [date] NOT NULL,

PRIMARY KEY CLUSTERED

(

[BooksId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[BooksAuthors] Script Date: 03.02.2021 18:15:21 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[BooksAuthors](

[BooksId] [int] NOT NULL,

[AuthorsId] [int] NOT NULL,

PRIMARY KEY CLUSTERED

(

[BooksId] ASC,

[AuthorsId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Genres] Script Date: 03.02.2021 18:15:21 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Genres](

[GenresId] [int] NOT NULL,

[GenresTitle] [nvarchar](30) NOT NULL,

PRIMARY KEY CLUSTERED

(

[GenresId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[TheTurnoverOfBooks] Script Date: 03.02.2021 18:15:21 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TheTurnoverOfBooks](

[IssueId] [int] NOT NULL,

[UsersId] [int] NOT NULL,

[BooksId] [int] NOT NULL,

[DateOfIssue] [date] NOT NULL,

[ReturnDate] [date] NULL,

PRIMARY KEY CLUSTERED

(

[IssueId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Users] Script Date: 03.02.2021 18:15:21 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Users](

[UsersId] [int] NOT NULL,

[UsersName] [nvarchar](30) NOT NULL,

[UsersFirstName] [nvarchar](20) NULL,

[UsersLastName] [nvarchar](20) NOT NULL,

[UsersDateOfBirth] [date] NOT NULL,

[UsersCourse] [nvarchar](5) NOT NULL,

[UsersGroupId] [nvarchar](15) NOT NULL,

[UsersFloor] [nvarchar](20) NOT NULL,

PRIMARY KEY CLUSTERED

(

[UsersId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Books] WITH CHECK ADD CONSTRAINT [FkGenresId] FOREIGN KEY([GenresId])

REFERENCES [dbo].[Genres] ([GenresId])

GO

ALTER TABLE [dbo].[Books] CHECK CONSTRAINT [FkGenresId]

GO

ALTER TABLE [dbo].[BooksAuthors] WITH CHECK ADD CONSTRAINT [FkAuthorsId] FOREIGN KEY([AuthorsId])

REFERENCES [dbo].[Authors] ([AuthorsId])

GO

ALTER TABLE [dbo].[BooksAuthors] CHECK CONSTRAINT [FkAuthorsId]

GO

ALTER TABLE [dbo].[BooksAuthors] WITH CHECK ADD CONSTRAINT [FkBooksId] FOREIGN KEY([BooksId])

REFERENCES [dbo].[Books] ([BooksId])

GO

ALTER TABLE [dbo].[BooksAuthors] CHECK CONSTRAINT [FkBooksId]

GO

ALTER TABLE [dbo].[TheTurnoverOfBooks] WITH CHECK ADD CONSTRAINT [FkBooksId1] FOREIGN KEY([BooksId])

REFERENCES [dbo].[Books] ([BooksId])

GO

ALTER TABLE [dbo].[TheTurnoverOfBooks] CHECK CONSTRAINT [FkBooksId1]

GO

ALTER TABLE [dbo].[TheTurnoverOfBooks] WITH CHECK ADD CONSTRAINT [FkUsersId] FOREIGN KEY([UsersId])

REFERENCES [dbo].[Users] ([UsersId])

GO

ALTER TABLE [dbo].[TheTurnoverOfBooks] CHECK CONSTRAINT [FkUsersId]

GO

USE [master]

GO

ALTER DATABASE [libbrary] SET READ\_WRITE

GO

**Приложение В**

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Anna

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

string connectionString = @"Data Source=LAPTOP-OV1RVUFK\SQLEXPRESS02;Initial Catalog=TA;Integrated Security=True";

public MainWindow()

{

InitializeComponent();

// Создание подключения

SqlConnection connection = new SqlConnection(connectionString);

try

{

// Открываем подключение

connection.Open();

}

catch (SqlException ex)

{

}

finally

{

// закрываем подключение

connection.Close();

//tb.Text = "Подключение закрыто...";

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

MainData.Text = "";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string sqlExpression = "Select BooksTitle FROM Books ";

SqlCommand command = new SqlCommand(sqlExpression, connection);

using (SqlDataReader reader = command.ExecuteReader())

{

if (reader.HasRows) // если есть данные

{

while (reader.Read()) // построчно считываем данные

{

MainData.Text += reader.GetValue(0) + "\n";

}

}

}

}

}

private void st\_Click(object sender, RoutedEventArgs e)

{

MainData.Text = "";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string sqlExpression = "Select ReturnDateActual,ReturnDateForFormal FROM TheTurnoverOfBooks";

SqlCommand command = new SqlCommand(sqlExpression, connection);

using (SqlDataReader reader = command.ExecuteReader())

{

if (reader.HasRows) // если есть данные

{

while (reader.Read()) // построчно считываем данные

{

MainData.Text += reader.GetValue(0) + "\n";

}

}

}

}

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

MainData.Text = "";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string sqlExpression = "Select AuthorsName,AuthorsFirstName,AuthorsLastName FROM Authors";

SqlCommand command = new SqlCommand(sqlExpression, connection);

using (SqlDataReader reader = command.ExecuteReader())

{

if (reader.HasRows) // если есть данные

{

while (reader.Read()) // построчно считываем данные

{

MainData.Text += reader.GetValue(0) + "\n";

}

}

}

}

}

private void delcust\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string sqlExpression = $"DELETE FROM Books WHERE BooksId = {delcustnum.Text}";

using (SqlCommand command = new SqlCommand(sqlExpression, connection))

{

command.ExecuteReader();

}

}

}

private void delcount\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string sqlExpression = $"DELETE FROM TheTurnoverOfBooks WHERE IssueId = {delcountnum.Text}";

using (SqlCommand command = new SqlCommand(sqlExpression, connection))

{

command.ExecuteReader();

}

}

}

}

}