|  |  |
| --- | --- |
| Департамент образования Ярославской области  Государственное профессиональное образовательное автономное учреждение  Ярославской области  «ЯРОСЛАВСКИЙ ПРОМЫШЛЕННО-ЭКОНОМИЧЕСКИЙ КОЛЛЕДЖ  им. Н.П. Пастухова» | |
| **ОТЧЁТ ПО УЧЕБНОЙ ПРАКТИКЕ**  **по профессиональному модулю ПМ.07**  **Соадминистрирование и автоматизация баз данных и серверов**  ОУП ПМ.07.09.02.07.18ИП2.14 | |
|  | Студент  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Д.Д. Тюкальцев  «\_\_\_\_» Декабря 2021 г. |
| Руководитель практики  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Ю.В. Маянцева  «\_\_\_\_» Декабря 2021 г. |
| 2021 | |

**Содержание**

[Практическая работа № 1 3](#_Toc89256052)

[Практическая работа № 2 9](#_Toc89256053)

[Практическая работа № 3 13](#_Toc89256054)

[Практическая работа № 4 16](#_Toc89256055)

[Практическая работа № 5 (2) 20](#_Toc89256056)

# **Практическая работа № 1**

**Тема**

Создание базы данных.

**Цель работы**

По разработанной ERD диаграмме реализовать базу данных, создать скрипт.

На основании ERD диаграммы, реализуется база данных, с использованием MS SQL SERVER. Для создания базы, сначала необходимо реализовать схему данных (Рисунок 2, 3, 4), по которой автоматически формируются таблицы. Для этого необходимо использовать поострённую ERD диаграмму на основе предметной области.

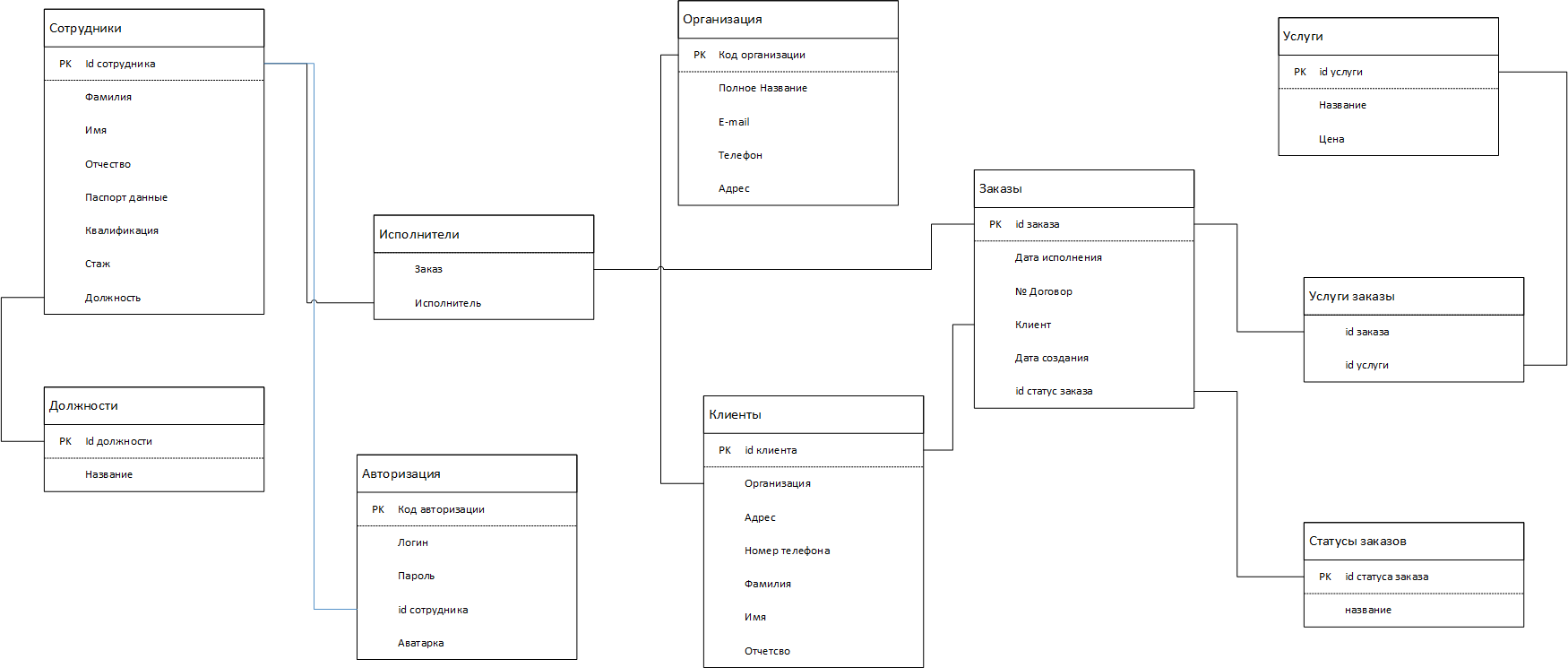
****

Рисунок 1. ERD диаграмма

Построенная база данные в MS SQL является нормализованной, находится в 3НФ (Рисунок 2, 3).

Нормализация — это процесс организации данных в базе данных. Это включает создание таблиц и установление связей между этими таблицами в соответствии с правилами, предназначенными как для защиты данных, так и для того, чтобы сделать базу данных более гибкой за счет устранения избыточности и непоследовательной зависимости.

Первая нормальная форма - чтобы база данных находилась в 1 нормальной форме, необходимо чтобы ее таблицы соблюдали следующие реляционные принципы:

* В таблице не должно быть дублирующих строк
* В каждой ячейке таблицы хранится атомарное значение (одно не составное значение)
* В столбце хранятся данные одного типа
* Отсутствуют массивы и списки в любом виде

Вторая нормальная форма - Чтобы база данных находилась во второй нормальной форме (2NF), необходимо чтобы ее таблицы удовлетворяли следующим требованиям:

* Таблица должна находиться в первой нормальной форме
* Таблица должна иметь ключ
* Все неключевые столбцы таблицы должны зависеть от полного ключа (в случае если он составной)

Третья нормальная форма - требование третьей нормальной формы (3NF) заключается в том, чтобы в таблицах отсутствовала транзитивная зависимость (**Транзитивная зависимость – это когда неключевые столбцы зависят от значений других неключевых столбцов.).**

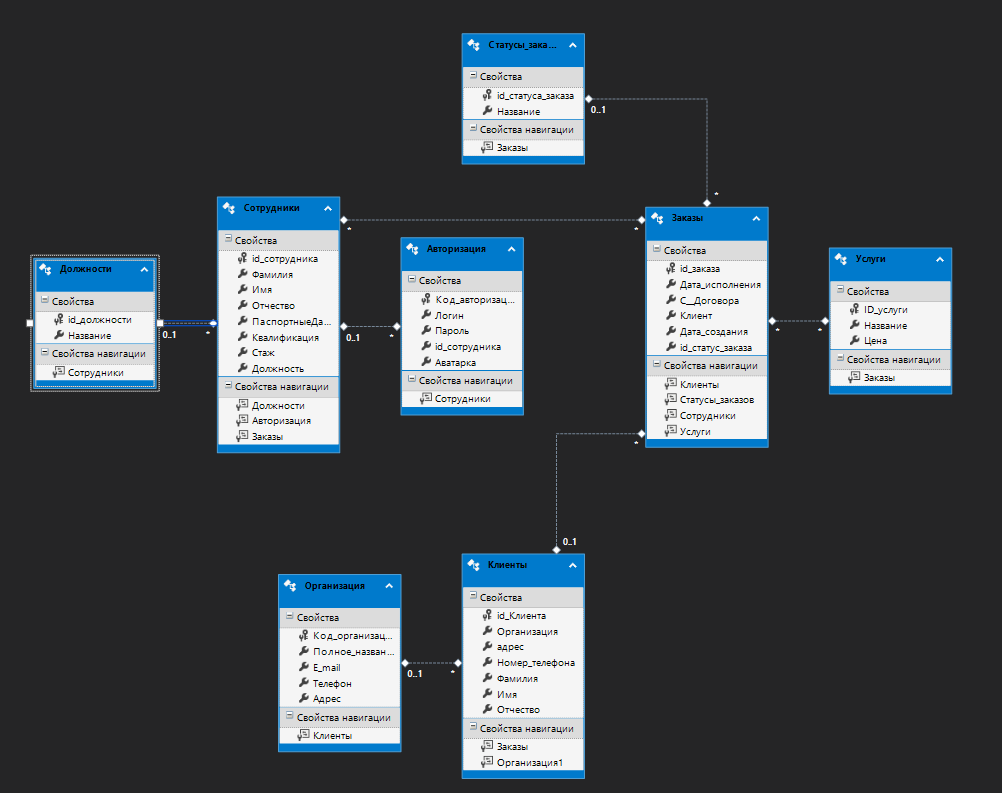


Рисунок 2. Диаграмма БД (фрагмент)

Далее необходимо сформировать скрипт развертывания этой базы. Для этого необходимо выполнить задачу «Сформировать скрипт». И выбрать необходимые объекты (Рисунок 3)

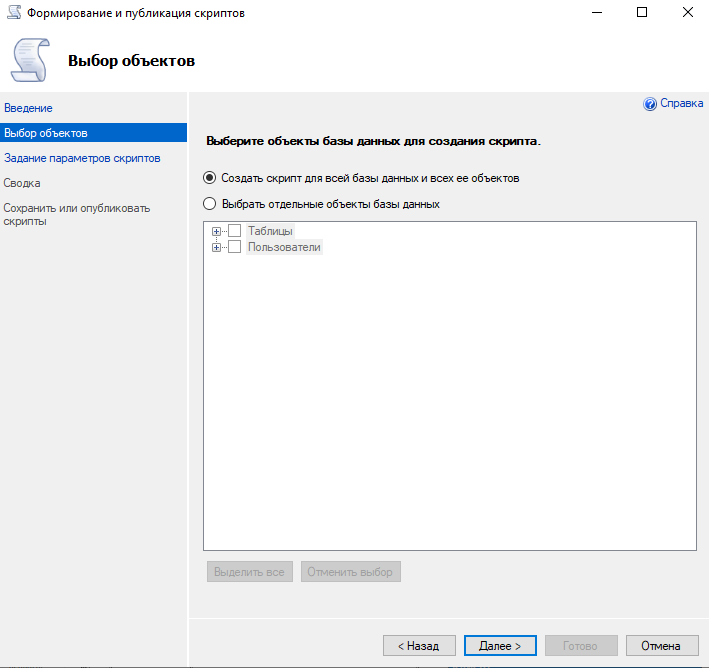


Рисунок 4. Выбор объектов для скрипта

После выбора необходимых параметров и выбора места сохранения, начнется создания скрипта (Рисунок 5)

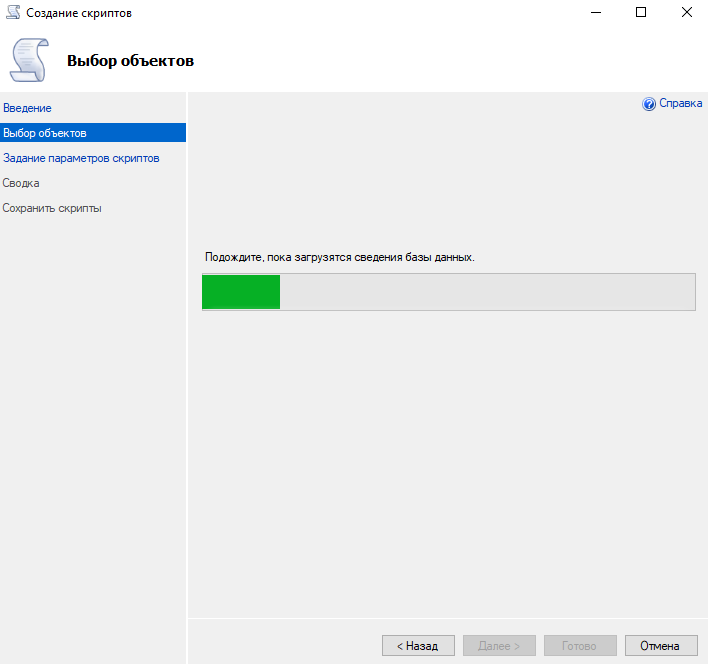


Рисунок 4. Выбор объектов для скрипта

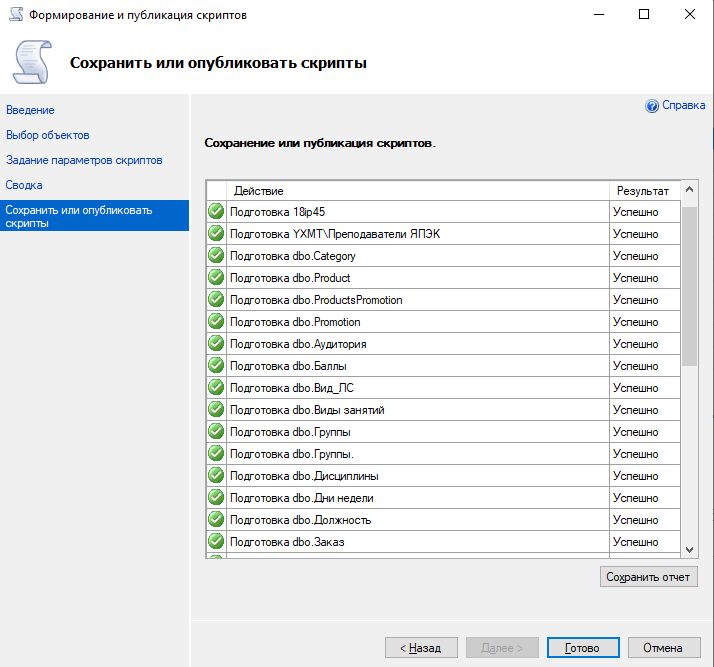


Рисунок 5. Создание скрипта



Рисунок 4 Созданный скрипт

**Вывод**

В результате выполнения первой практической работы была создана и заполнена первичными данными база данных, а также создан её скрипт, при помощи которого в будущем будет возможно развернуть данную базу.

**Контрольные вопросы**

1. **Что такое «база данных»?**

База данных - набор сведений, хранящихся некоторым упорядоченным способом. Иными словами, база данных — это хранилище данных.

1. **Из каких этапов состоит проектирование базы данных?**

Концептуальное (инфологическое) проектирование

Логическое (даталогическое) проектирование

Физическое проектирование

1. **Из каких этапов состоит проектирование базы данных?**

Таблица – набор полей

Поля – это столбцы таблицы

1. **Что такое первичный ключ (Primary кеб)?**

Столбец, значения которого во всех строках различны.

1. **Что такое внешний ключ (Foreign key)?**

Внешние ключи позволяют установить связи между таблицами.

1. **Какие связи существуют в реляционных базах данных?**

* [Многие ко многим](https://office-menu.ru/uroki-sql/41-tipy-svyazej-v-relyatsionnykh-bazakh-dannykh#manytomany);
* [Один ко многим](https://office-menu.ru/uroki-sql/41-tipy-svyazej-v-relyatsionnykh-bazakh-dannykh#onetomany);
* Один к одному.

1. **Что подразумевается под целостностью данных?**

Соответствие имеющейся в базе данных информации её внутренней логике, структуре и всем явно заданным правилам.

1. **Что такое нормализация и каковы ее преимущества?**

Это метод проектирования базы данных, который позволяет привести базу данных к минимальной избыточности.

# **Практическая работа № 2**

**Тема**

Размещение файлов в репозитории.

**Цель работы**

Разместить скрипт с базой данных, руководство по стилю, логотип и иконку в удаленный репозиторий проекта.

Для выполнения это практической работы было разработано следующие руководство по стилю (Рисунок 6).

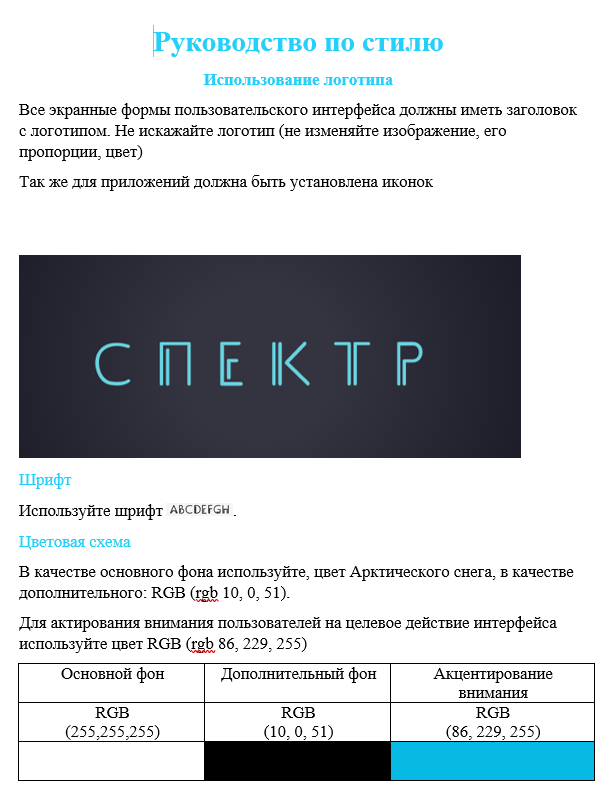


Рисунок 6. Руководство по стилю

Далее необходимо создать локальный репозиторий и выгрузить в него все данные в систему контроля версий github (Рисунок 7, 8).

Для глобального объявления пользователя используется команда “$ git config – global user.name “Имя пользователя””.

Чтобы инициировать локальный репозиторий в папке необходимо ввести команду “$ git init”.

Для добавления файлов в локальный репозиторий необходимо использовать команду “$ git add “название файла””.

Если необходимо закомментировать файл используется команда “$ git commit -m “””.

Для связывания локального репозитория с удаленным используется команда “$ git remote add origin “ссылка на репозиторий””

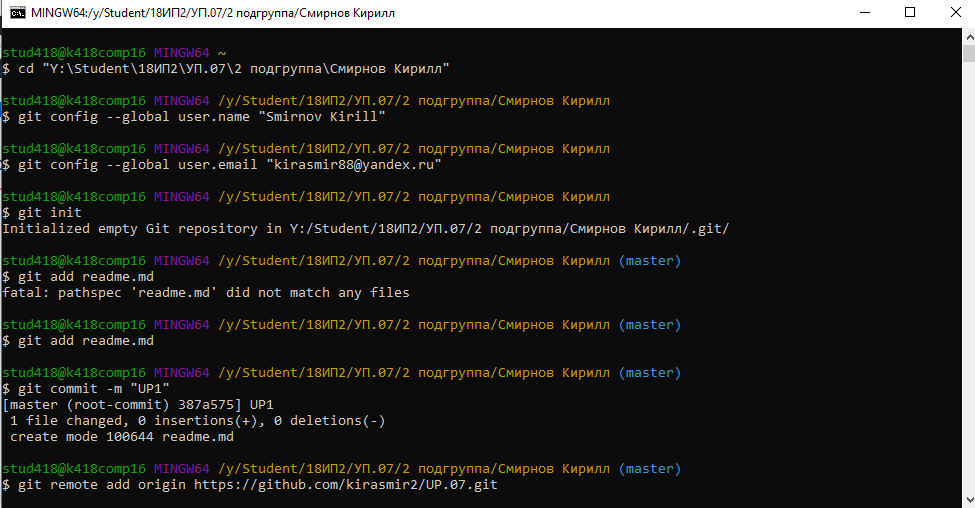


Рисунок 7. Создание локального репозитория и выгрузка в него данных

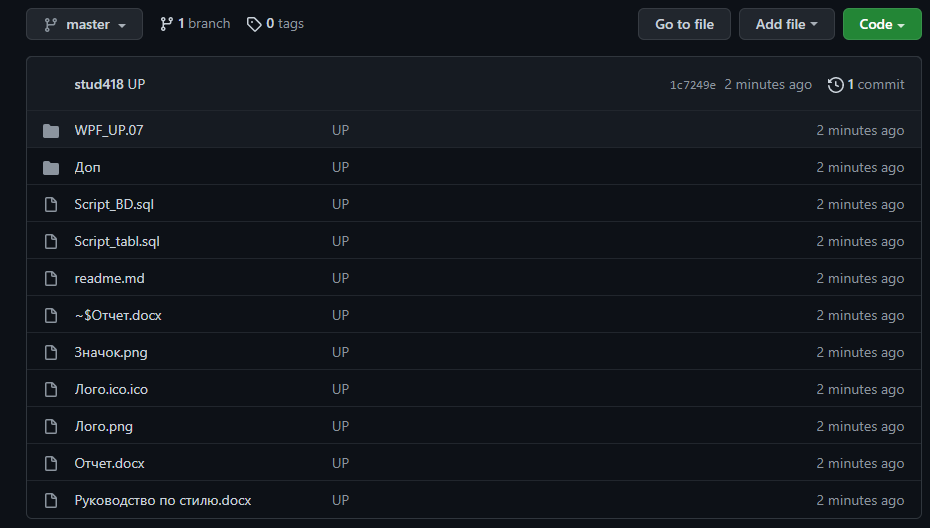


Рисунок 8. Данные загруженные в удаленный репозиторий проекта

**Вывод**

В результате выполнения практической работы был размещен скрипт с базой данных, создано руководство по стилю и выгружен логотип и иконка в удаленный репозиторий проекта.

**Контрольные вопросы**

1. **В чем заключается экономия времени при использовании системы контроля версий?**

Система Git очень экономична и не требует рассылки большого количества файлов. Отслеживаются и пересылаются изменения в файлах и ссылки на эти изменения. То есть основная рассылка — это рассылка разницы в ваших редактированиях.

1. **В чем преимущества использования системы контроля версий?**

Позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

1. **Что такое Git?**

Git — одна из распределенных систем контроля версий.

1. **Как начать использовать Git?**

Скачать систему git и завести репозиторий.

1. **Как начать использовать GitHub?**

Зайти на сайт и зарегистрироваться.

1. **Основные (наиболее часто используемые) команды Git.**

git pull, git commit, git checkout, git push

1. **Какие сервисы существуют для Git?**

GitHub, Codebase, SourceForge, SourceHut, Gitea, Bitbucket, GitLab

1. **Как работать с локальным репозиторием?**

Работа с локальным репозиторием. Клонирование репозитория, чтобы начать работать с репозиторием, следует создать копию проекта со всей его историей локально.

1. **Как работать с распределенным репозиторием?**

Для работы с распределенным репозиторием необходимо файлы можно загружать сразу в удаленный репозиторий, для этого требуется просто выбрать необходимые файлы и сохранить их.

# **Практическая работа № 3**

**Тема**

Создание проекта в Visual Studio. Настройка проекта. Создание словаря стилей. Создание Git репозитория с помощью Visual Studio.

**Цель работы**

Создать и настроить проект в Visual Studio, создать репозиторий, используя средства Visual Studio.

Для начала работы необходимо создать папку Resources и в ней ещё 3 папки: Icons, Image, Theme. После чего в папку Icons необходимо добавить иконку в формате .ico. А в папку Image эмблему (Рисунок 9).

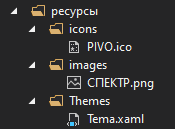


Рисунок 9. Созданные папки

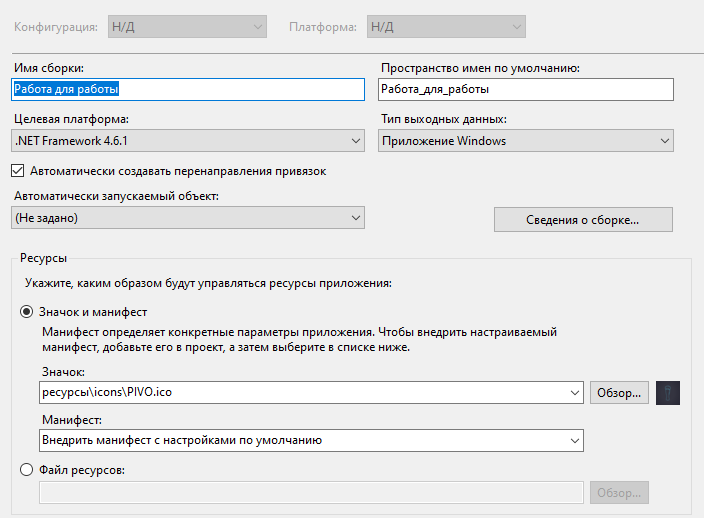


Рисунок 10. Добавление иконки

После этого в папку Themes добавляем словарь стилей (Рисунок 11, 12).



Рисунок 11. Фрагмент Словаря стилей для таблиц

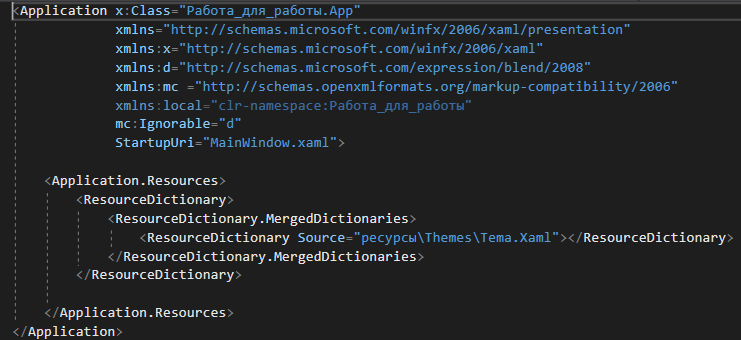


Рисунок 12. Код для использования словаря стилей

Далее, при размещении кнопок или таблицы, они автоматические будут оформлены в заданном стиле (Рисунок 13)

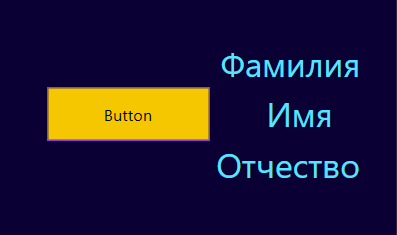


Рисунок 13. Форма WPF

Далее необходимо сохранить и выгрузить локальный репозиторий (Рисунок 14, 15).

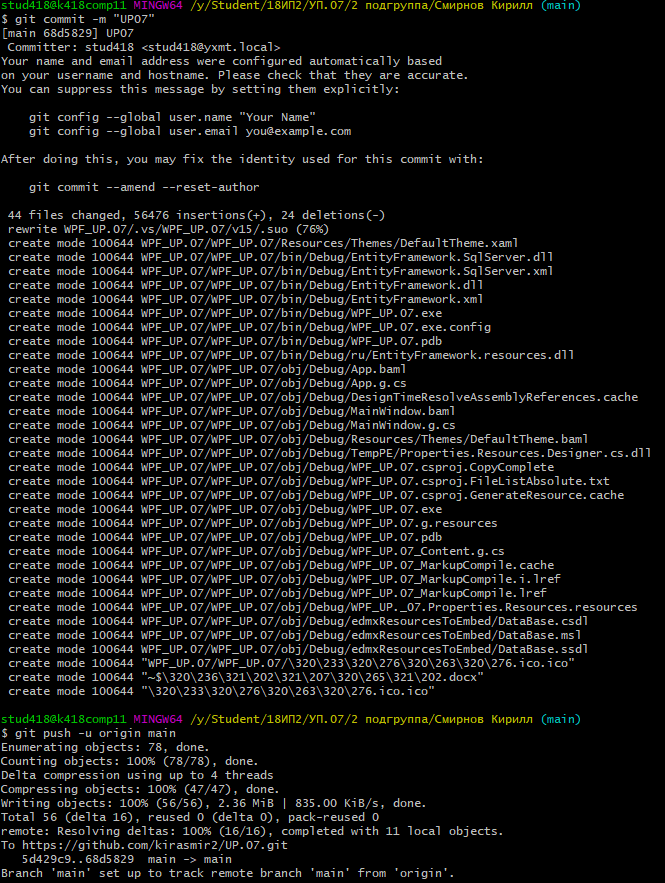


Рисунок 14. Выгрузка локального репозитория

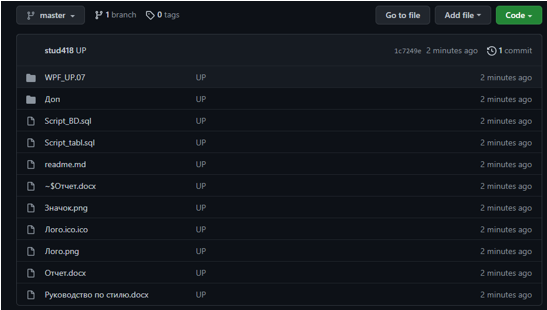


Рисунок 15 Удаленный репозиторий

После этого используя словарь стилей, требуется воссоздать легендарную авторизацию (Рисунок 16)

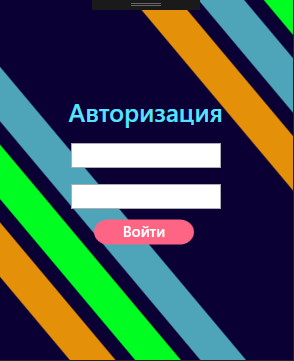


Рисунок 16. Форма авторизации

**Вывод**

В результате выполнения практической работы было сформировано WPF приложение Visual studio, был создан словарь ресурсов, а также изменена иконка приложения, затем в данный словарь были внесены данные для элементов: “Button”, “Window”, “DataGrid”. После изменений в словаре ресурсов приложение было загружено на GitHub.

**Контрольные вопросы**

1. **Для чего предназначен словарь ресурсов в проекте WPF?**

Словарь ресурсов представляет собой просто XAML-документ, который хранит необходимые ресурсы.

1. **Как определить стиль для элемента управления в WPF?**

Чтобы определить, для какого элемента хотим задать стиль, прописывается тег <Style TargetType=“x:Type название\_элемента\_управления”/>

1. **За что отвечает атрибут x:Key у стиля?**

Атрибут x:Key отвечает за изменение одного конкретного элемента с индивидуальным названием.

1. **Как применить стиль, у которого указан атрибут x:Key, к элементу на форме?**

Для того, чтобы задать стиль для элемента, нужно использовать атрибут

Style = “{StaticResource название\_указанное\_в\_атрибуте\_x:Key}”

# **Практическая работа № 4**

**Тема**

Подключение базы данных. Авторизация в системе.

**Цель работы**

По заданной схеме данных разработать базу данных, создать программное приложение с возможностью авторизации в системе.

Для начала работы необходимо создать подключение к БД (Рисунок 17)

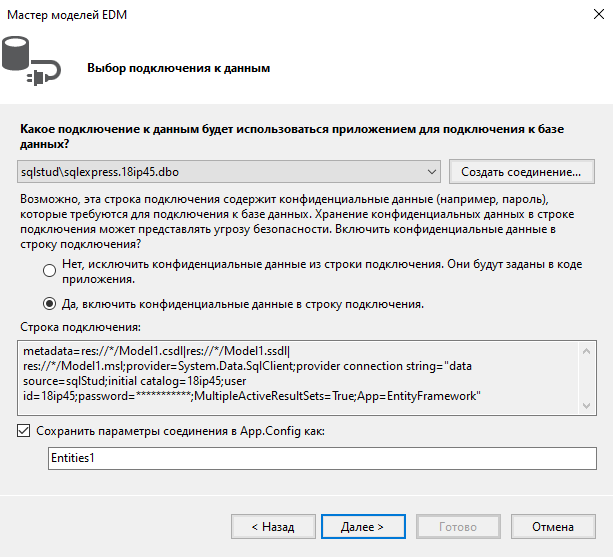


Рисунок 17. подключение к БД

После чего выбрать таблицы, которые будут использоваться в программе (Рисунок 18)

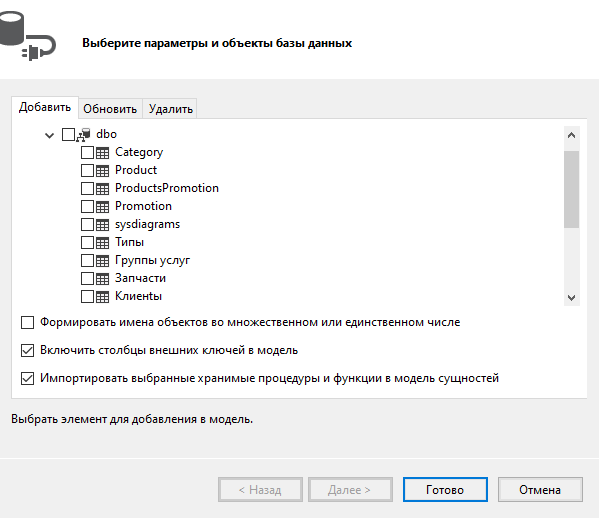


Рисунок 18 Выбор таблиц

Для работы с БД необходимо создать класс (Рисунок 19)

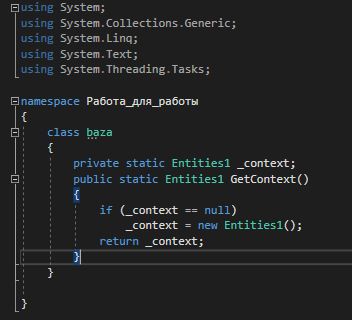


Рисунок 19. Создание класса

Далее была реализована форма “Авторизация”, которая использует БД для авторизации пользователя.

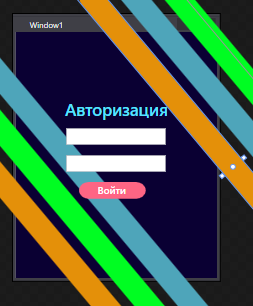


Рисунок 20. Форма авторизации

**Вывод**

В результате выполнения практической работы к созданному ранее приложению была подключена база данных, создан класс для работы с ней, также при помощи элементов из словаря данных было сформировано окно авторизации, после чего был разработан его функционал.

**Контрольные вопросы**

1. **Что такое «Entity Framework»?**

Entity Framework — это решение для работы с базами данных, которое используется в программировании на языках семейства.NET. Оно позволяет взаимодействовать с СУБД с помощью сущностей (entity), а не таблиц.

1. **В чем плюсы и в чем минусы «Entity Framework»?**

Entity Framework позволяет значительно сократить код работы с базами данных.

Однако стоит учитывать, что Entity Framework выступает прослойкой между приложением и базой данных, поэтому может ухудшаться производительность. Для небольших проектов это допустимо, но если программа должна работать под большой нагрузкой, то лучше использовать чистый ADO.NET.

1. **Какие возможности предоставляет «Entity Framework»?**

Entity Framework позволяет использовать:

* foreign keys;
* связи one-to-one, one-to-many и many-to-many;
* параметризацию запросов;
* хранимые процедуры.

1. **Как создать подключение к базе данных?**

Для создания подключения к БД необходимо в обозревателе решений создать элемент “Модель ADO.NET ADM”. После чего откроется мастер создания моделей EDM. Дальше необходимо выбрать “Конструктор EF из базы данных” и нажать кнопку “Далее”. В следующем окне нажать на кнопку “Создать соединение”. В окне “Свойства подключения” указать актуальные данные подключения. После необходимо поменять “Проверка подлинности Windows” на “Проверка подлинности SQL”. В следующем окне необходимо казать необходимые таблицы.

# **Практическая работа № 5 (1)**

**Тема**

Отображение набора данных, навигация.

**Цель работы**

Реализовать отображение данных базы данных на форме. Сделать подсветку строк по условию.

В начале работы необходимо дополнить текущую форму новым элементом, который будет отображать данные об учебном расписании в табличном виде – DataGrid (Рисунок 21). Далее необходимо создать фотографию пользователя и имя, фамилию, которая будет выводиться при входе в аккаунт. (рисунок 21)

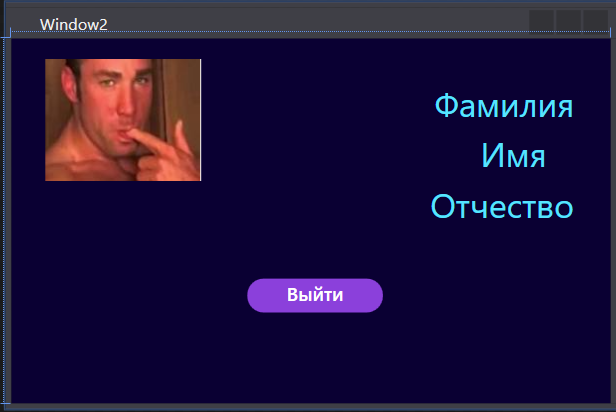


Рисунок 21 Форма WPF

Далее для реализации подсветки необходимо ввести код в окне. Для подсветки одного блока необходимо поменять пармаетр Trigger value на текст из таблицы БД. Параметр Setter value необходим для изменения цвета Условием в данной работе является название аптеки и товара (Рисунок 23)

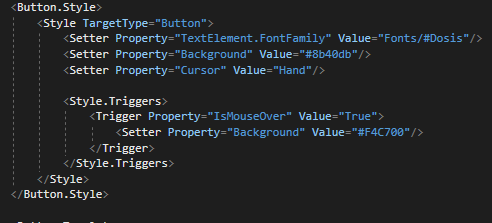


Рисунок 22. Подсветка кнопки

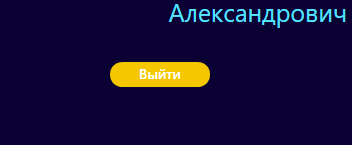


Рисунок 23. Форма WPF

**Вывод**

В результате выполнения практической работы было реализовано отображение данных базы данных на форме. Сделана подсветка строк по условию.

**Контрольные вопросы**

1. Что такое Windows Presentation Foundation? В чем преимущество перед WinForms?

Windows Presentation Foundation (WPF) — аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

Преимущества WPF:

* Использование традиционных языков .NET-платформы - C# и VB.NET для создания логики приложения;
* Возможность декларативного определения графического интерфейса с помощью специального языка разметки XAML
* Независимость от разрешения экрана
* Новые возможности, которых сложно было достичь в WinForms, например, создание трехмерных моделей, привязка данных, использование таких элементов, как стили, шаблоны, темы и др.;
* Хорошее взаимодействие с WinForms
* Богатые возможности по созданию различных приложений: это и мультимедиа, и двухмерная и трехмерная графика, и богатый набор встроенных элементов управления
* Аппаратное ускорение
* Создание приложений под множество ОС семейства Windows - от Windows XP до Windows 10;

1. Что такое XAML? Для чего он используется в WPF?

XAML (eXtensible Application Markup Language) - язык разметки, используемый для инициализации объектов в технологиях на платформе .NET. Применительно к WPF (а также к Silverlight) данный язык используется прежде всего для создания пользовательского интерфейса декларативным путем.

1. Что такое LINQ? Как работает технология LINQ?

Аббревиатура LINQ обозначает целый набор технологий, создающих и использующих возможности интеграции запросов непосредственно в язык C#. Традиционно запросы к данным выражаются в виде простых строк без проверки типов при компиляции или поддержки IntelliSense. Кроме того, разработчику приходится изучать различные языки запросов для каждого типа источников данных: баз данных SQL, XML-документов, различных веб-служб и т. д. Технологии LINQ превращают запросы в удобную языковую конструкцию, которая применяется аналогично классам, методам и событиям.

# **Практическая работа № 5 (2)**

**Тема**

Создание представлений и работа с ними.

**Цель работы**

Реализовать представление, на основе которого реализовать отображение данных базы данных на форме.