|  |  |
| --- | --- |
| Департамент образования Ярославской области  Государственное профессиональное образовательное автономное учреждение  Ярославской области  «ЯРОСЛАВСКИЙ ПРОМЫШЛЕННО-ЭКОНОМИЧЕСКИЙ КОЛЛЕДЖ  им. Н.П. Пастухова» | |
| **Отчет по УЧЕБНОЙ практике**  **ПМ.02 Осуществление интеграции программных модулей**  УП.02.09.02.07.18ИП1.2 - 15 | |
|  | Студент  А.Э. Кязимова  «28» апреля 2021 г. |
| Руководитель практики  Ю.В. Маянцева  «28» апреля 2021 г. |
| 2021 | |

Содержание

[Практическая работа №1 3](#_Toc70504543)

[Практическая работа №2 6](#_Toc70504544)

[Практическая работа №1 8](#_Toc70504545)

[Практическая работа №2 13](#_Toc70504546)

[Практическая работа №3 17](#_Toc70504547)

[Практическая работа №4 21](#_Toc70504548)

# Практическая работа №1

Создание репозитория в хостинге IT-проектов GITHUB

**Цель работы:** получить начальный опыт создания репозитория в хостинге IT-проектов

**Ход работы:**

Для начала перейдем на официальный сайт Git hub и зайти на свой аккаунт.

Далее в правом верхнем углу рядом с фотографией профиля нашли выплывающие контекстное меню и создали новый репозиторий.

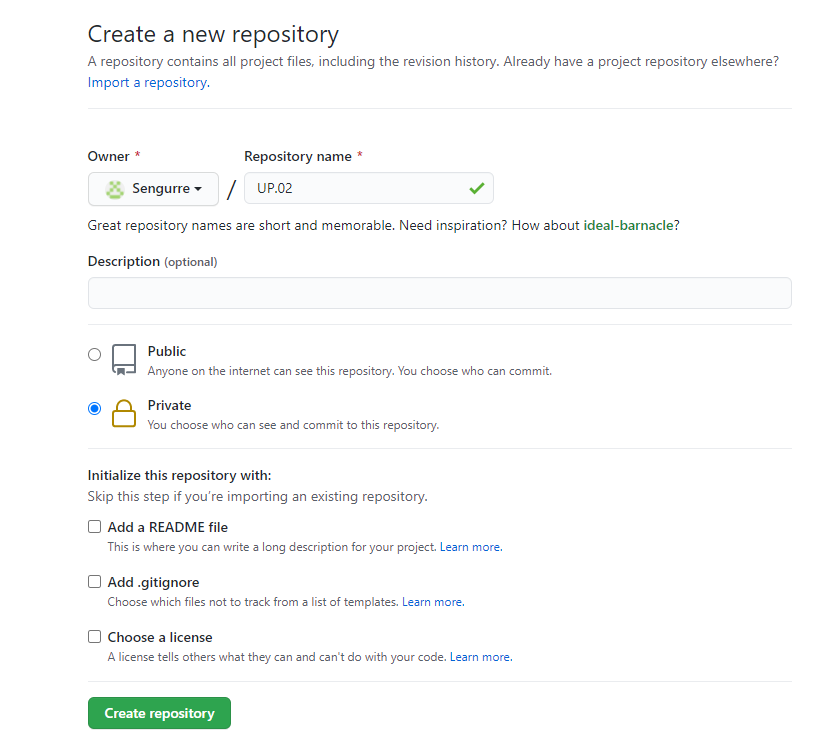


Рисунок 1.Создание репозитория UP.02

После успешного создание репозитория, нам выводится окно с параметрами, где мы приглашаем нашего преподавателя и даем ему возможность смотреть файлы на нём.

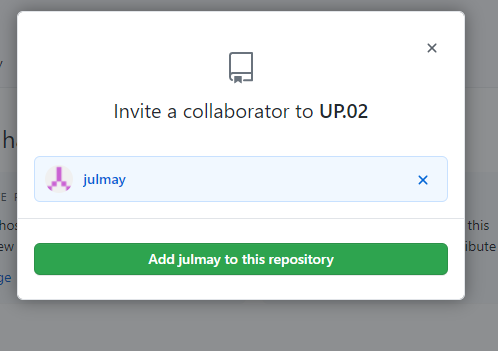


Рисунок 2. Приглашение в репозиторий

По заданной диаграмме (***согласно своему варианту***) с помощью СУБД MS SQL создадим базу данных.

Открыв программу и зайдя в свой профиль, создадим базу данных. Проще всего это сделать с помощью диаграмм БД. Правой кнопкой мыши нужно щелкнуть по папке «Диаграммы баз данных» и выберем «Создать диаграмму базы данных» (Рисунок 1).

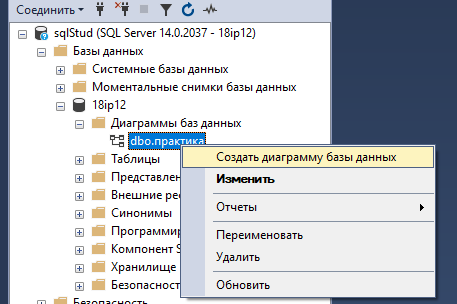


Рисунок 3 Создание диаграммы

После на щелкнем по рабочей области (белое поле) и создадим таблицу (правой кнопкой мыши вызвать меню) (Рисунок 2).

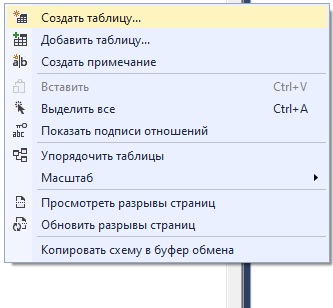


Рисунок 4 Создание таблиц

Создав таблицы, установим первичные ключи (Рисунок 3), с помощью которых соединяем таблицы (Рисунок 4).

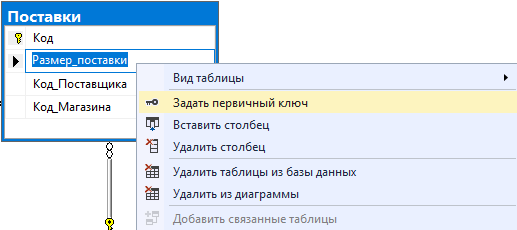


Рисунок 5 задание пе6рвичного ключа

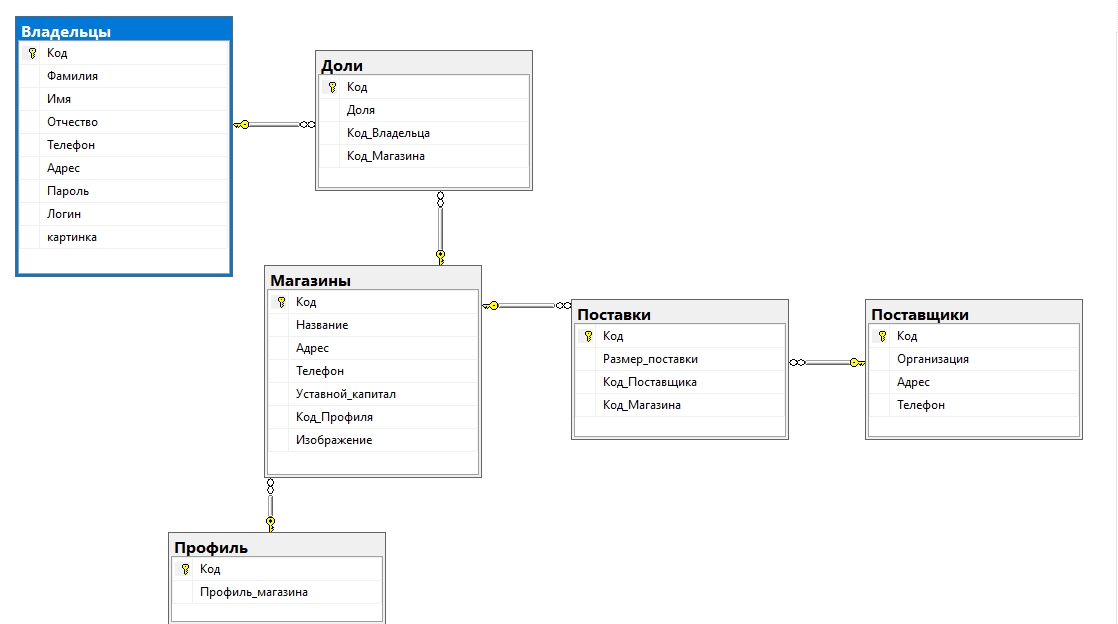


Рисунок 6 Созданная связь таблиц

После сохранения и обновления базы, заполним таблицы нужной информацией (Рисунок 7).

Снова слева из списков развернем папку «Таблицы», там должны были появиться созданные ранее таблицы БД (Рисунок 5).

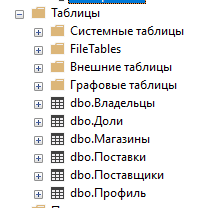


Рисунок 7 Папка "Таблицы"

ПКМ изменим их(Рисунок 6).

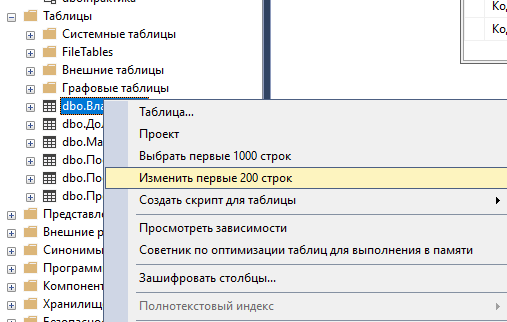


Рисунок 8 Меню для редактирования

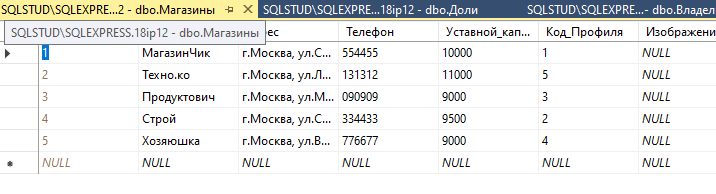


Рисунок 7 Заполненная таблица

Создав и заполнив БД, нужно подключить ее к проекту в Visual Studio.

Для этого щелкнув по проекту ПКМ создадим элемент (Рисунок 6, 7).

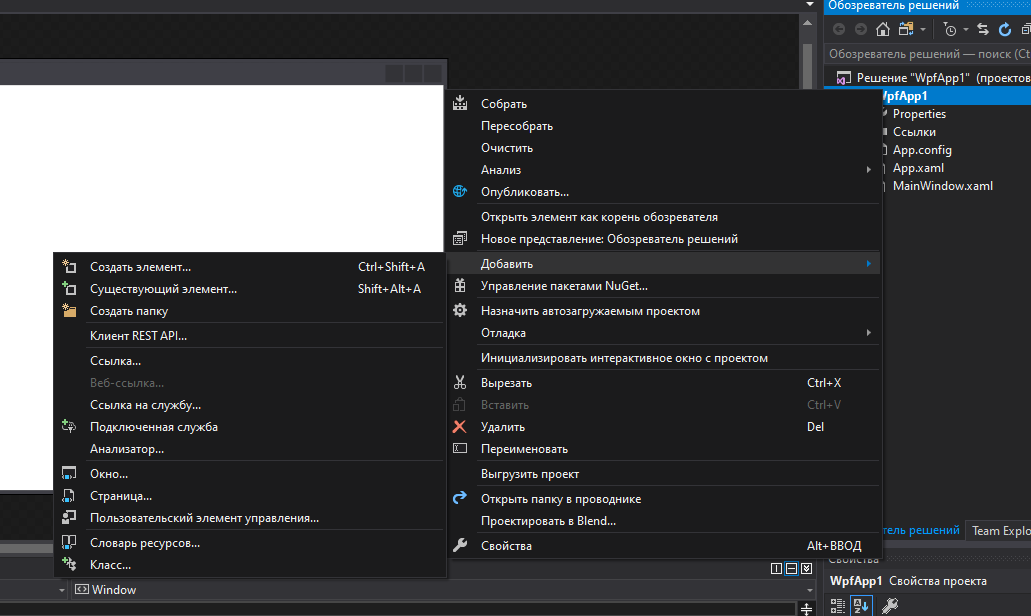


Рисунок 10 Создание элемента

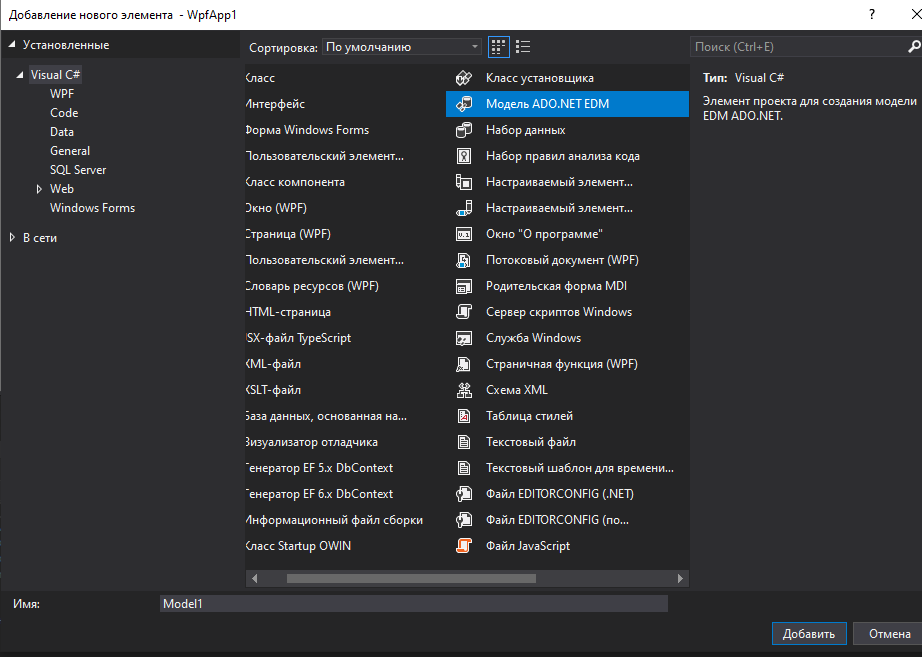


Рисунок 11 Выбор создаваемого элемента

Для подключения БД, нужно указать сервер, и если надо имя пользователя и пароль (Рисунок 10).

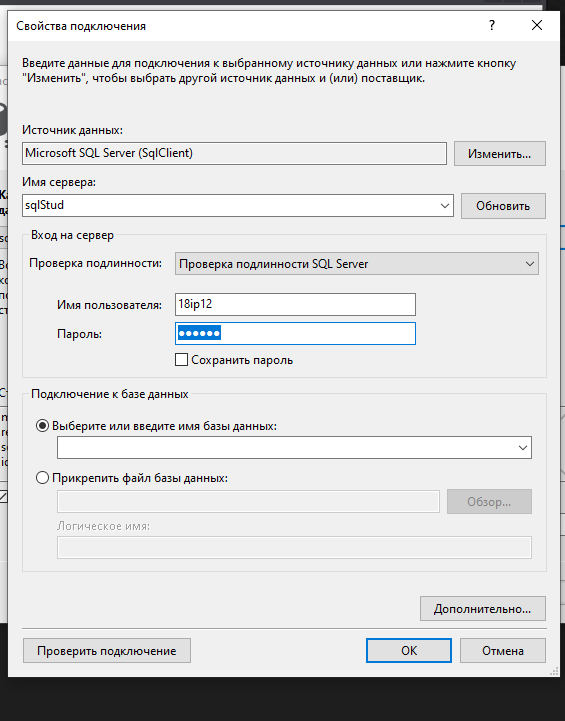


Рисунок 12 Свойство подключения

Далее нам предлагается выбрать, что добавить (Рисунок 11).

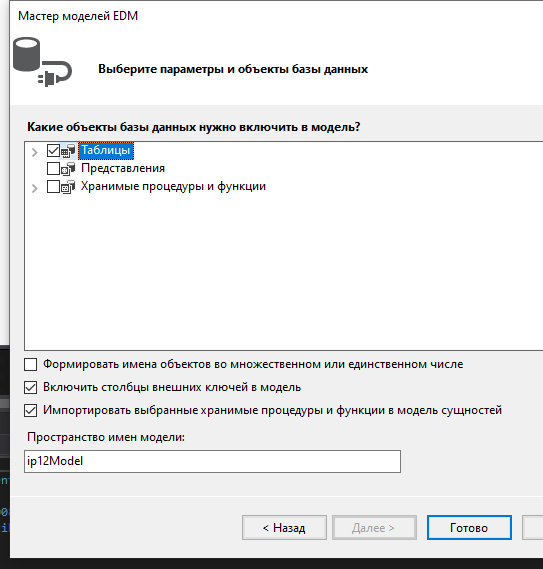


Рисунок 13 Выбор объектов БД

Нажимаем «Готово» и ждем загрузки. База данных подключена (Рисунок 12)

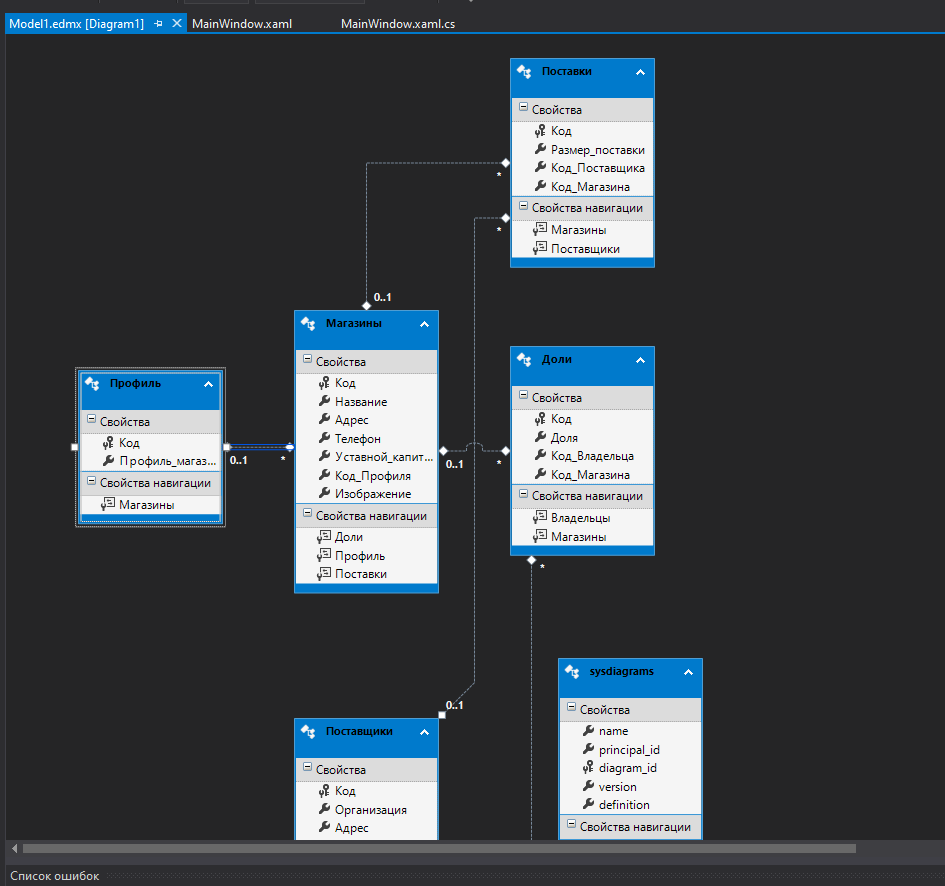


Рисунок 14 Подключенная база данных

В классе нам необходимо немного дописать код, чтобы с помощью данного класса мы могли иметь доступ к базе данных (Рисунок 13).

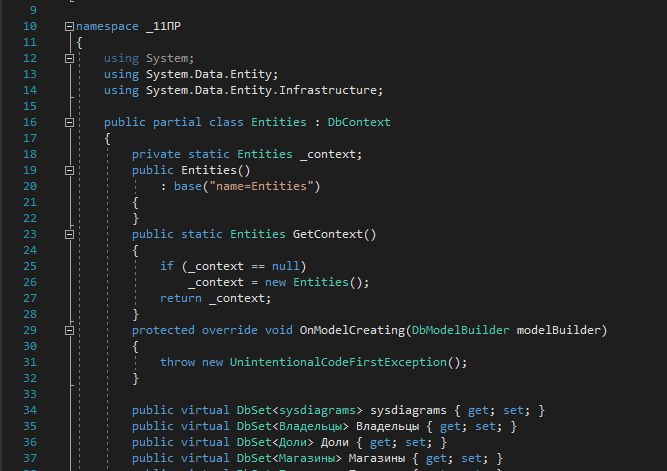


Рисунок 15 Код

Перейдем к главной форме и реализуем интерфейс для авторизации в системе (Рисунок 14).

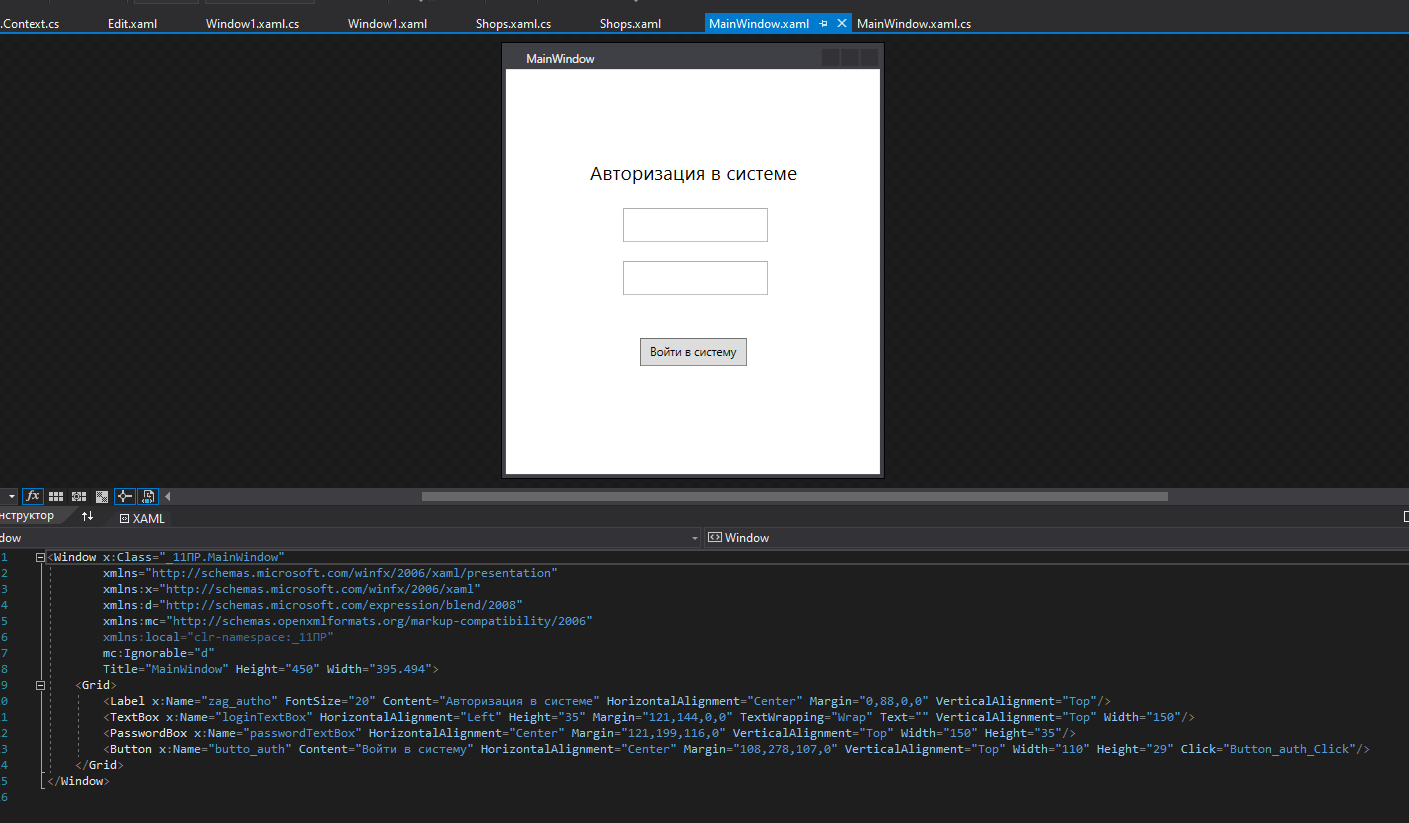


Рисунок 16 Реализованный интерфейс

**Контрольные вопросы**

1. Что такое «база данных»?

*База данных* - набор сведений, хранящихся некоторым упорядоченным способом.

1. Из каких этапов состоит проектирование базы данных?

Основные этапы проектирования баз данных

* Концептуальное (инфологическое) проектирование
* Логическое (даталогическое) проектирование
* Физическое проектировании

1. Что подразумевается под таблицей и полем в SQL?

В реляционной базе данных все данные представлены в виде простых таблиц. Каждая таблица состоит из столбцов (их называют полями или атрибутами) и строк (их называют записями или кортежами). Таблицы в реляционных базах данных обладают рядом свойств

1. Что такое первичный ключ (Primary key)?

*Первичный ключ* (сокращенно РК - primary key) - столбец, значения которого во всех строках различны. Первичные ключи могут быть логическими (естественными) и суррогатными (искусственными).

1. Что такое внешний ключ (Foreign key)?

**Внешний ключ** — это столбец или комбинация столбцов, значения которых соответствуют Первичному ключу в другой таблице. Связь между двумя таблицами задается через соответствие Первичного **ключа** в одной из таблиц **внешнему** ключу во второй.

1. Какие связи существуют в реляционных базах данных?

Один к одному;

Один ко многим;

Многие ко многим.

1. Что подразумевается под целостностью данных?

**Целостность информации** — термин, означающий, что данные не были изменены при выполнении какой-либо операции над ними, будь то передача, хранение или отображение.

1. Что такое нормализация и каковы ее преимущества?

На пересечении каждого столбца и строки может находиться только атомарное значение (одно значение, не состоящее из группы значений). Таблицы, удовлетворяющие этому условию, называют нормализованными.

1. Как создать подключение к базе данных?

Нажав по созданному проекту в VS и выбрав пункт «создать элемент» выбираем иконку для создания базы данных. Дальнейшие деймтвия описаны выше в отчете.

# Практическая работа №2

Редактирование файла readme.md. Размещение файлов в репозиторий.

**Цель работы:** получить начальный опыт размещения файлов в репозитории

**Ход работы:**

Для начала мы отредактировали файл readme.md согласно шаблону

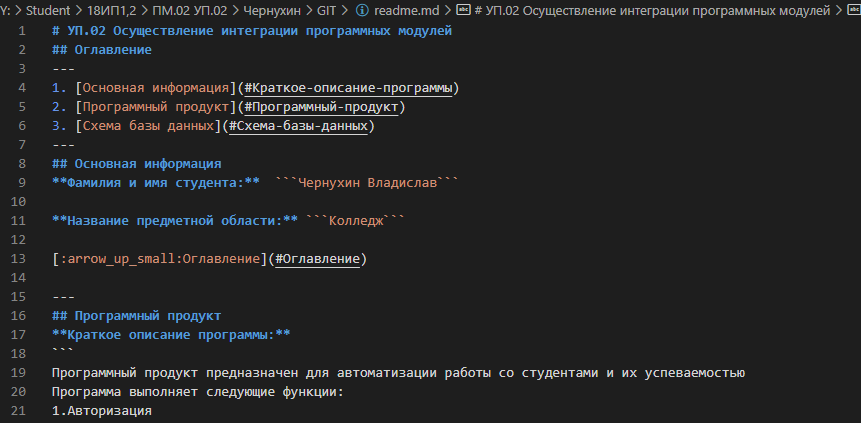


Рисунок 3. Редактирование файла readme

Далее нам нужно запустить git в режиме коносли, для этого мы нажимаем ПКМ в свободное пространство и выберем Git Bash Here

Затем мы прописываем информацию о пользователе

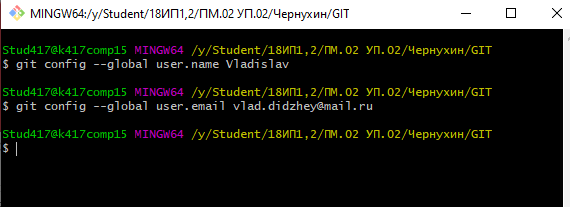


Рисунок 4. Внос информации о пользователе

Потом инициализируем локальный репозиторий через команду git init и добавляем файл командой git add

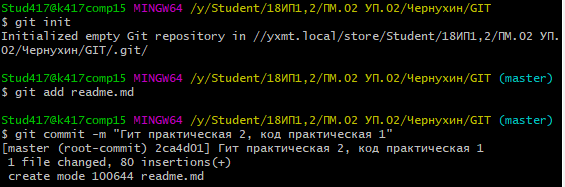


Рисунок 5.Инициализация и добавление

Отображаем изменения в локальном репозитории на удаленном и авторизируемся.

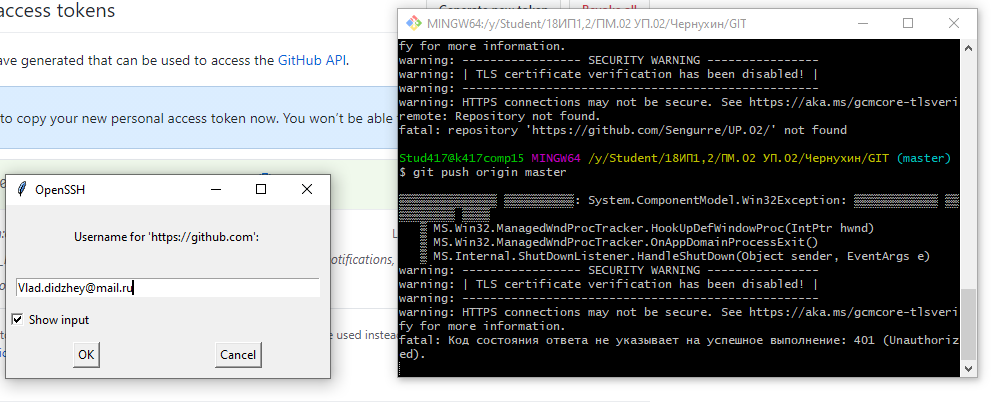


Рисунок 6.Авторизация

Контрольные вопросы

1. *Как создать новую ветвь?*

* *Благодаря команде git branch*

1. *Как просмотреть историю изменений?*

* *Благодаря команде git push*

1. *Как сохранять и синхронизировать изменения?*

* *Благодаря команде git stash*

1. *Как выполнить слияние ветвей?*

* *Благодаря команде git merge*

# Практическая работа №1

Подключение базы данных. Авторизация пользователя.

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

**Ход работы:**

Для создания таблицы, переходим в решения базы данных, для этого мы нажмём на раскрытие ветки нашей созданной базы данных и правой кнопкой мыши по папке «Таблицы» в диалоговом окне, выбираем «Создать таблицу» (Рис.3)

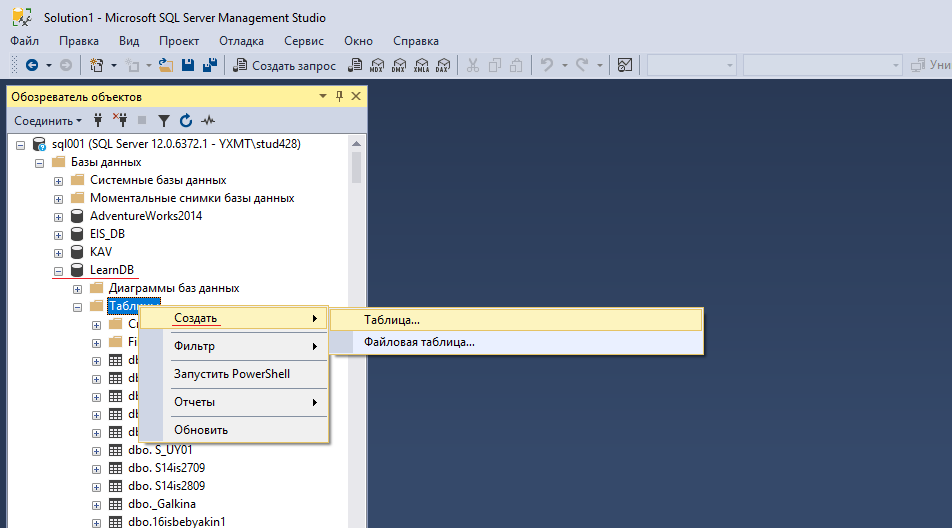


Рисунок 7.Создание таблицы

Создав и заполнив нужное количество таблиц, мы создаем диаграмму и выставляем связи между таблицами.

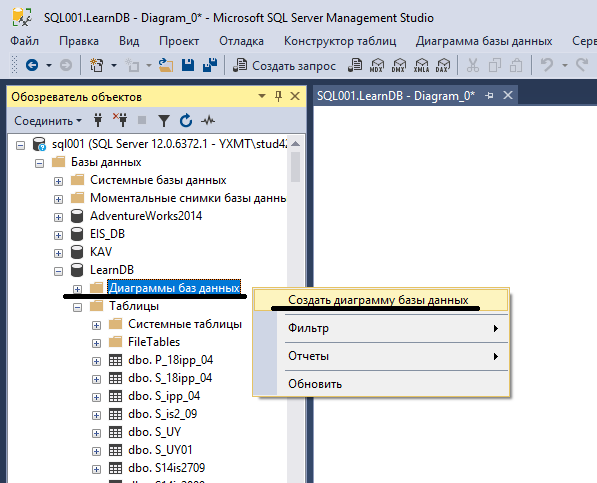


Рисунок 8. Cоздание диаграммы

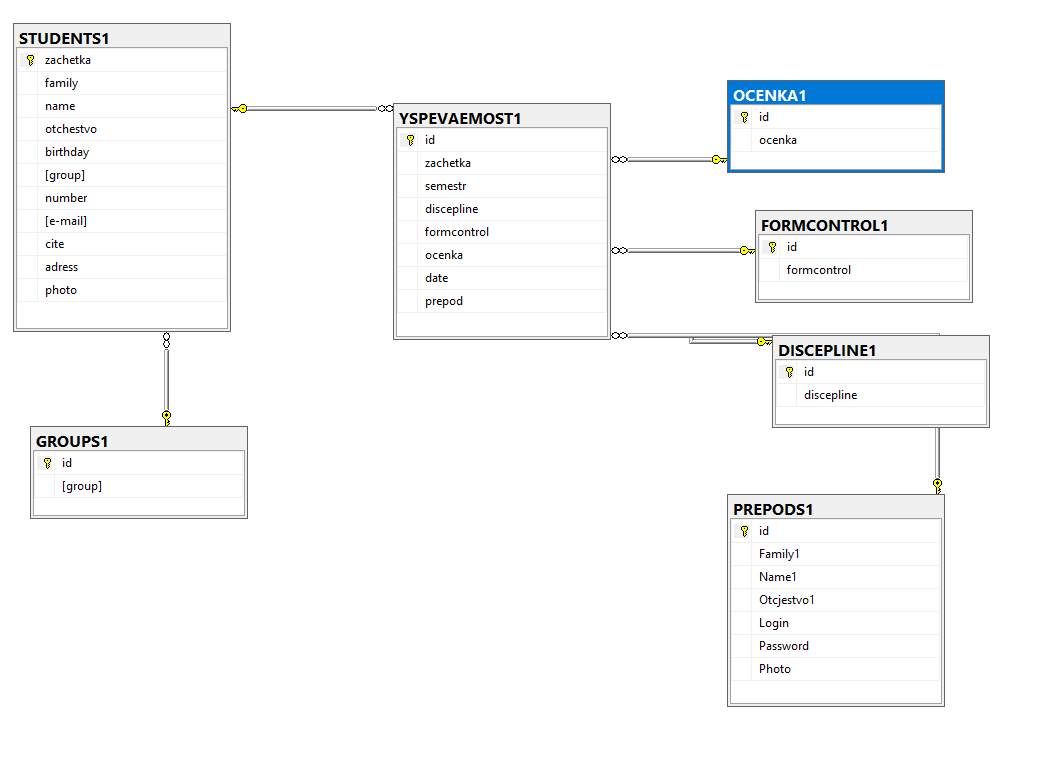


Рисунок 9. Готовая база

Далее нам нужно подключить базу данных и сделать окно авторизации. Для этого откроем среду разработки Visual Studio и создадим новый проект «WPF». Следующим шагом будет подключение базы данных к проекту. Для этого на панели проекта нажмем правой кнопкой мыши по нашему проекту и выберем пункт «Добавить» -> «Создать новый элемент». Из списка элементов выберем элемент «Модель ADO.NET EDM». Укажем не обходимые нам таблицы и создадим подключение.

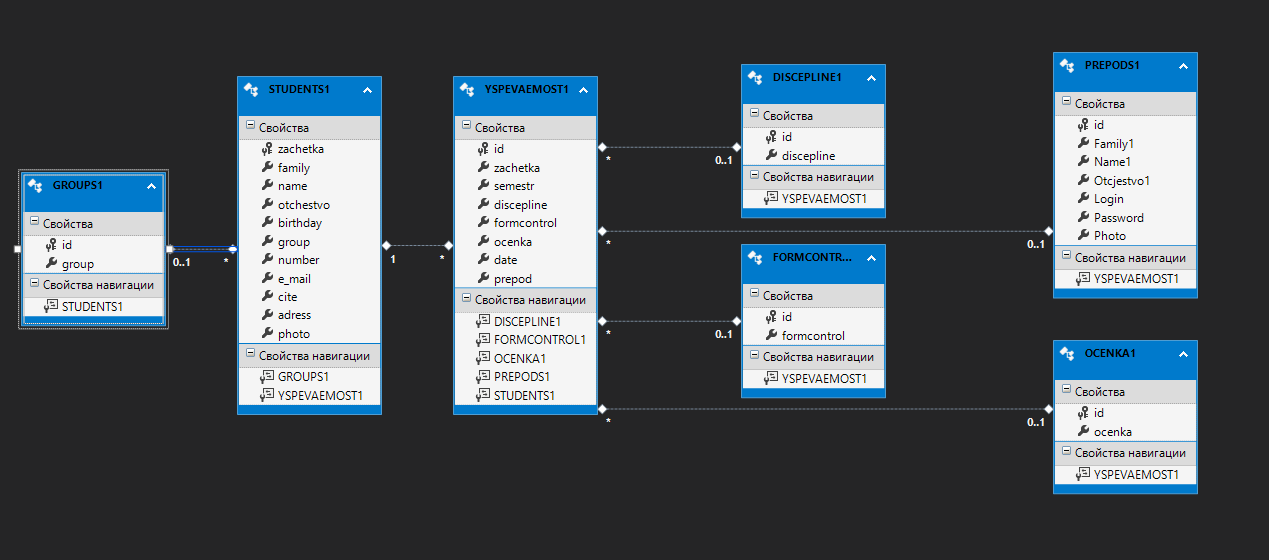


Рисунок 10. Подключенная база

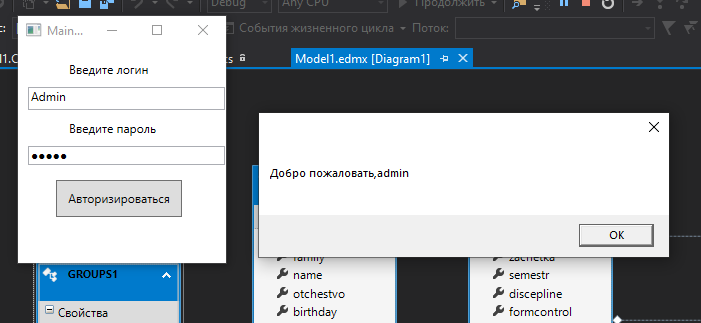


Рисунок 11.Работа окна авторизации

**Контрольные вопросы**

1. Что такое «база данных»?

*База данных* - набор сведений, хранящихся некоторым упорядоченным способом. Можно сравнить базу данных со шкафом, в котором хранятся документы. Иными словами, база данных — это хранилище данных. Сами по себе базы данных не представляли бы интереса, если бы не было систем управления базами данных (СУБД).

1. Из каких этапов состоит проектирование базы данных?

Основные этапы проектирования баз данных

* Концептуальное (инфологическое) проектирование
* Логическое (даталогическое) проектирование
* Физическое проектирование

1. Что подразумевается под таблицей и полем в SQL?

* В таблице не может быть двух одинаковых строк. В математике таблицы, обладающие таким свойством, называют отношениями - по-английски relation, отсюда и название - реляционные.
* Столбцы располагаются в определенном порядке, который создается при создании таблицы. В таблице может не быть ни одной строки, но обязательно должен быть хотя бы один столбец.
* У каждого столбца есть уникальное имя (в пределах таблицы), и все значения в одном столбце имеют один тип (число, текст, дата...).
* На пересечении каждого столбца и строки может находиться только атомарное значение (одно значение, не состоящее из группы значений). Таблицы, удовлетворяющие этому условию, называют нормализованными.

1. Что такое первичный ключ (Primary key)?

*Первичный ключ* (сокращенно РК - primary key) - столбец, значения которого во всех строках различны. Первичные ключи могут быть логическими (естественными) и суррогатными (искусственными). Так, для нашей таблицы Пользователи первичным ключом может стать столбец e-mail (ведь теоретически не может быть двух пользователей с одинаковым e-mail). На практике лучше использовать суррогатные ключи, т.к. их применение позволяет абстрагировать ключи от реальных данных.

1. Что такое внешний ключ (Foreign key)?

Внешние ключи позволяют установить связи между таблицами. Внешний ключ устанавливается для столбцов из зависимой, подчиненной таблицы, и указывает на один из столбцов из главной таблицы. Как правило, внешний ключ указывает на первичный ключ из связанной главной таблицы.

1. Какие связи существуют в реляционных базах данных?

Всего существует 3 типа связей:

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

1. Что подразумевается под целостностью данных?

Под целостностью базы данных понимают соответствие имеющейся в базе данных информации её внутренней логике, структуре и всем явно заданным правилам. Каждое правило, налагающее некоторое ограничение на возможное состояние базы данных, называется ограничением целостности.

Откроется мастер создания моделей EDM. Выбираем «Конструктор EF из базы данных» и жмем кнопку «Далее»

1. Как создать подключение к базе данных?

* Для этого на панели проекта нажмем правой кнопкой мыши по нашему проекту и выберем пункт «Добавить» -> «Создать новый элемент»
* Из списка элементов выберем элемент «Модель ADO.NET EDM». Зададим данному элементу название.
* В следующем окне нажимаем на кнопку «Создать соединение»
* В окне «Свойства подключения» укажите актуальные для вас данные подключения
* Выбор версии оставьте без изменений
* В следующем окне укажите необходимые таблицы
* После завершения работы с мастером, система покажет вам схема данных вашей базы данных. Обратите внимание, как здесь отражены связи между сущностями (например, в связях «многие ко многим» отсутствует промежуточная таблица, которая связывала две сущности)

# Практическая работа №2

Отображение данных на форме.

**Цель работы:** реализовать отображение данных базы данных на форме.

**Ход работы:** Для отображения данных нам нужно будет создать новое оконо, правым кликом по проекту в окне добавления элементов добавляем нужные нам интсрументы.

Добавим картинку для вывода изображение пользователя, так же настроим вывод его данных и добавим элемент DataGrid для вывода данных из

Дополним текущую форму новым элементом, который будет отображать данные об аренде книг в табличном виде – DataGrid (рисунок8). Так как данный элемент имеет более сложную структуру, чем остальные элементы управления, давайте более детально рассмотрим его.

В разметке XAML внутри данного элемента находится DataGrid.Columns – элемент, который хранит информацию о столбцах таблицы. Создадим типизированный список объектов данного класса и добавим в него пару объектов. Теперь перейдем к получению данных из базы данных. Один из способов формирования запроса к базе данных является использование языка LINQ – язык, встроенный в NET., по синтаксису схожий с языком запросов SQL. Осталось только в качестве источника данных указать результаты нашего запроса. Используем свойство ItemSource у DataGrid и присваиваем ему результат запроса в виде списка (используем метод ToList).

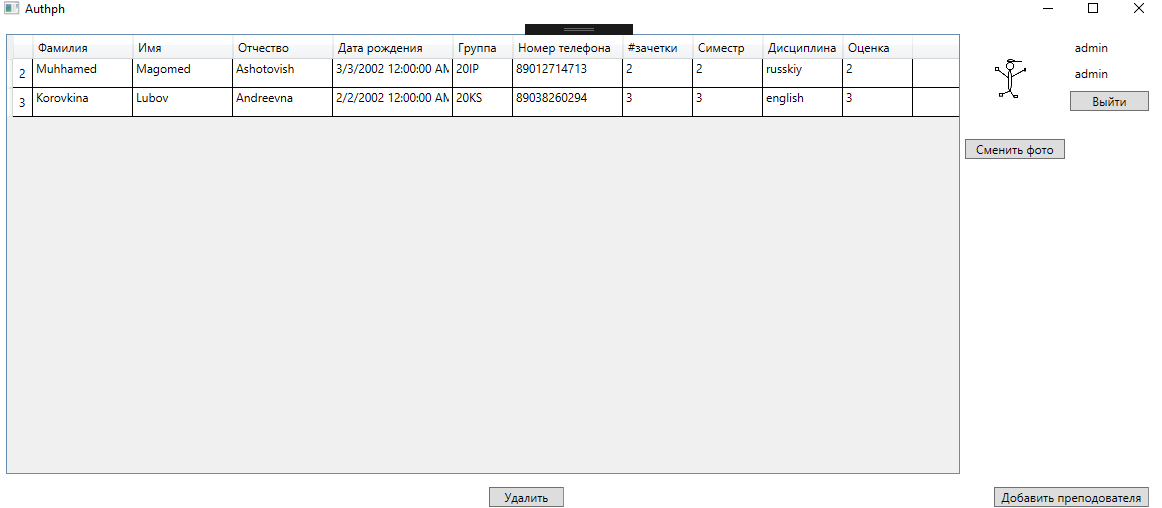


Рисунок 12. Окно вывода данных

Следующим шагом будет создание окна «Информация о студенте», реализуем мы это следующим образом, нажимая 2 раза на ID из DataGrid, у нас открывается новое окно, где мы увидим подробную информацию о студенте. Двойной клик мы реализуем через событие MouceDouble\_click в коде XAML и так же будем передавать из таблицы в данное окно ID выбранного студента, для упрощения вывода данных.

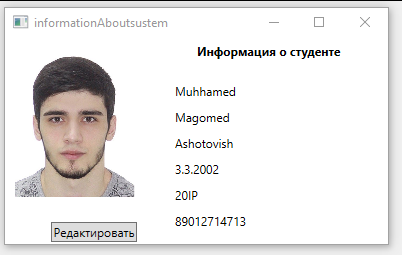


Рисунок 13. Окно информации

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, основанном на xml и представляющем альтернативу программному созданию графики и элементов управления, а также возможность комбинировать XAML и C#/VB.NET;
* Независимость от разрешения экрана: поскольку в WPF все элементы измеряются в независимых от устройства единицах, приложения на WPF легко масштабируются под разные экраны с разным разрешением;
* 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, благодаря чему, например, в приложениях WPF можно использовать традиционные элементы управления из WinForms;
* Богатые возможности по созданию различных приложений: это и мультимедиа, и двухмерная и трехмерная графика, и богатый набор встроенных элементов управления, а также возможность самим создавать новые элементы, создание анимаций, привязка данных, стили, шаблоны, темы и многое другое;
* Аппаратное ускорение графики - вне зависимости от того, работаете ли вы с 2D или 3D, графикой или текстом, все компоненты приложения транслируются в объекты, понятные Direct3D, и затем визуализируются с помощью процессора на видеокарте, что повышает производительность, делает графику более плавной;
* Создание приложений под множество ОС семейства Windows - от Windows XP до Windows 10;

Также стоит учитывать, что по сравнению с приложениями на Windows Forms объем программ на WPF и потребление ими памяти в процессе работы в среднем несколько выше. Но это с лихвой компенсируется более широкими графическими возможностями и провышенной производительностью при отрисовке графики.

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. Хотя функциональность XAML только графическими интерфейсами не ограничивается: данный язык также используется в технологиях WCF и WF, где он никак не связан с графическим интерфейсом. То есть его область шире. Применительно к WPF мы будем говорить о нем чаще всего именно как о языке разметки, который позволяет создавать декларативным путем интерфейс, наподобие HTML в веб-программировании. Однако опять же повторюсь, сводить XAML к одному интерфейсу было бы неправильно, и далее на примерах мы это увидим.

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.

Для разработчика, который создает запросы, наиболее очевидной частью LINQ является интегрированное выражение запроса. Выражения запроса используют декларативный синтаксис запроса. С помощью синтаксиса запроса можно выполнять фильтрацию, упорядочение и группирование данных из источника данных, обходясь минимальным объемом программного кода. Одни и те же базовые выражения запроса позволяют одинаково легко получать и преобразовывать данные из баз данных SQL, наборов данных ADO .NET, XML-документов, XML-потоков и коллекций .NET.

# Практическая работа №3

Редактирование данных в БД с помощью приложения.

**Цель работы:** реализовать редактирование данных из базы данных с помощью программного приложения.

**Ход работы:** создадим новую форму для редактирования данных студента, в форму мы будем переходить из формы, где выводили информацию о студенте. При инициализации мы получим объект класса аренды и назначим его в качестве контекста этой формы. Поэтому, когда мы будем указывать свойство, к которому будет привязан элемент управления, привязки будет ассоциироваться с данным объектом. Первым делом, обновим модель нашей базы данных. Для этого, с помощью свойства, которое отслеживает изменения в базе данных (ChangeTracker) получаем все объекты, в которые были внесены изменения c помощью метода Entries, приводим их к списку с помощью метода ToList() и проходимся по каждому элементу из данного списка, используя метод ForEach, и вызываем метод обновления объекта Reload().

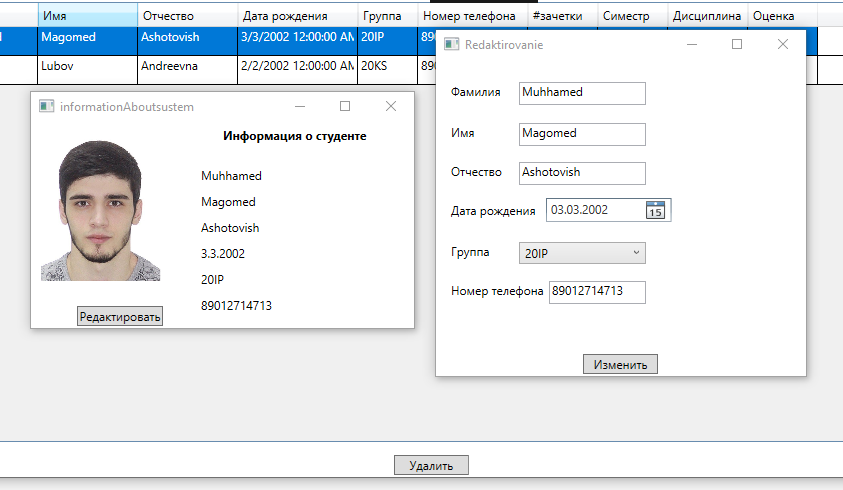


Рисунок 14. Окно редактирования

Для упрощения передачи данных в базу для элемента TextBox мы создадим Binding и назовем полями из нашей базы:

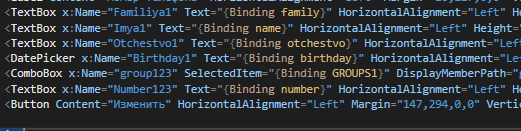


Рисунок 15. Создание связи

На форму с нашей таблицей добавим кнопку «Сменить фото» рядом с изображением профиля.

Используя экземпляр диалогового окна, к которому мы присваиваем фильтр только на картинки, благодаря этому мы сможет указать путь к картинке.

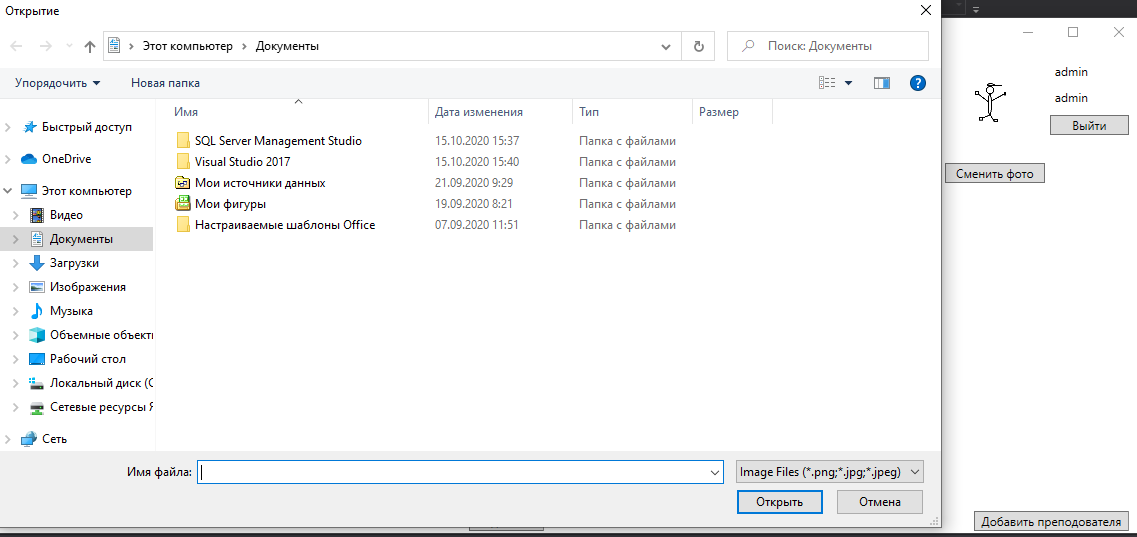


Рисунок 16.Смена фото

Контрольные вопросы

1. Что такое «реляционная алгебра»? Для чего она используется?

Реляционная алгебра базируется на теории множеств и является основой логики работы баз данных.

Реляционная алгебра состоит из операций над отношениями и их составляющими (атрибутами и кортежами). Результат любой операции реляционной алгебры - новое отношение. Такие системы операций называются замкнутыми. Рассмотрим семь основных операций реляционной алгебры.

1. Перечислите основные операции реляционной алгебры. Расскажите подробно об одной из них.

В первую входят операции, совершаемые над любыми множествами:

* объединение,
* пересечение,
* разность
* декартово произведение.

Во вторую группу входят операции, применимые только к отношениям:

* выборка,
* проекция,
* соединение.
* **Объединение** - операция над двумя отношениями, в результате которой получается новое отношение, состоящее из всех кортежей исходных отношений. Общие для исходных отношений кортежи в новом отношении встречются только по одному разу.
* *Пример объединения*
* Нужно объединить два отношения *Физ\_лица* и *Юр\_лица*.

|  |  |  |
| --- | --- | --- |
| Отношение *Физ\_лица* | | |
| **ФИО** | **Адр\_регистрации** | **Факт\_адр** |
| Иванов Ю.М. | Москва, Тверская 2 | С.-Петербург,Садовая ул. 12 |
| Сергеев И.А. | С.-Петербург, Седова 23 | С.-Петербург, Гороховая ул. 34 |
| ..... | ..... | ..... |



|  |  |  |
| --- | --- | --- |
| Отношение *Юр\_лица* | | |
| **Наим** | **Адр\_регистрации** | **Адр\_офиса** |
| Альфа | Новгород, Садовая ул. 2 | С.-Петербург,Садовая ул. 42 |
| Бета. | С.-Петербург, Московский пр. 23 | Гатчина, Лесная ул. 34 |
| ..... | ..... | ..... |

* Реальный запрос на языке SQL состоит из двух объединённых операторов SELECT. Первым SELECT выбирается вся таблица (реальная, а не теоретическое отношение) Физ\_лица, вторым - Юр\_лица. Результаты обеих выборок выводятся в общую таблицу. Оба оператора объединяются в один запрос предложением UNION:

|  |  |  |
| --- | --- | --- |
| **ИМЯ** | **Адр\_официальный** | **Фактический\_адр** |
| Иванов Ю.М. | Москва, Тверская 2 | С.-Петербург,Садовая ул. 12 |
| Сергеев И.А. | С.-Петербург, Седова 23 | С.-Петербург, Гороховая ул. 34 |
| Альфа | Новгород, Садовая ул. 2 | С.-Петербург,Садовая ул. 42 |
| Бета. | С.-Петербург, Московский пр. 23 | Гатчина, Лесная ул. 34 |
| ..... | ..... | ..... |

1. Какую операцию реляционной алгебры вы использовали в данной работе и зачем?

**Пересечение** - операция над двумя отношениями, в результате которой получается новое отношение, состоящее из кортежей, принадлежащих обоим исходным отношениям.

# Практическая работа №4

Добавление и удаление данных из БД с помощью приложения.

**Цель работы** реализовать добавление и удаление данных из базы данных с помощью программного приложения.

**Ход работы:** Создайте новую форму для добавления новой записи в таблицу.

Рассмотрим новый способ отображения ошибок, когда в одном окне будем выводить несколько ошибок через класс String builder

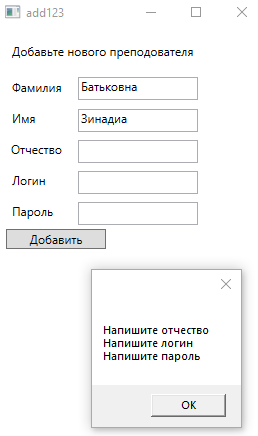


Рисунок 17.Вывод ошибок

И проверим работу добавление преподавателя

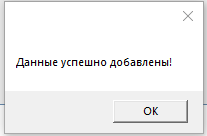


Рисунок 18.Добавление данных

Добавим на форму с арендой кнопку «Удалить».

Нам нужно, чтобы кнопка была доступна только тогда, когда выделена любая строка в DataGrid. Для этого, подпишем dataGrid на событие SelectionLineChanged, и для этого события реализуем следующий код.

Далее, нам необходимо проверить, какое решение принял пользователь (на какую кнопку нажал). Поэтому мы должны сравнить наш результат со значением из перечисления MessageBoxResult (значение Yes).

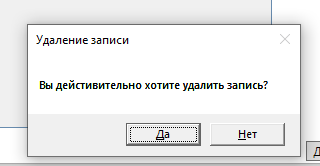


Рисунок 19.Удаление записи

**Контрольные вопросы**

1. Опишите структуру команды "INSERT INTO"

Команда "INSERT INTO" (можно перевести с английского как "ВСТАВИТЬ В") вставляет данные в таблицу, название которой идёт после неё. В нашем случае это "USERS". Затем идёт слово "SET" (переводится как "ЗАДАТЬ"), после которого через запятую перечисляются названия столбцов в таблице и их значения, который надо вставить.

В нашем примере первый столбец ID принимает максимальное уникальное значение, поэтому нет большого смысла передавать его при добавлении данных. Для разнообразия во втором добавлении передано не только значение NAME, но и значение DATE, хотя это необязательно, потому что при создании столбца DATE было сказано, что он принимает значение равное текущей дате и времени, если не передать ему что-то другое.

Базы данных могут быть настроены по-разному. Формат даты может отличаться, из-за чего значение DATETIME может несохраниться, если не угадать с форматом. Поэтому вместо строки, содержащей дату в формате "ГГГГ-ММ-ДД ЧЧ:ММ:СС" можно написать "NOW()" (переводится "сейчас"), тогда в таблицу будет вставлено значение текущей даты в нужном формате.

1. Опишите структуру команды "DELETE"

DELETE — в языках, подобных MDl, SQL-операция удаления записей из таблицы. Критерий отбора записей для удаления определяется выражением *where*. В случае, если критерий отбора не определён, выполняется удаление всех записей.