|  |  |  |
| --- | --- | --- |
| 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.02.09.02.07.19ИП2 | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | Студент  \_\_\_\_\_\_\_\_\_\_\_\_\_ Кувыркин И.М.  «6» мая 2022 г | |
|  | Преподаватель  \_\_\_\_\_\_\_\_\_\_\_\_\_ Куксов Е.О.  «6» мая 2022 г. | |
|  |  | |
| 2022 | |

**Содержание**

[**Практическая работа №1.1** 3](#_Toc103157303)

[**Практическая работа №1.2** 6](#_Toc103157304)

[**Практическая работа №1.3** 9](#_Toc103157305)

[**Практическая работа №1.4** 12](#_Toc103157306)

**Практическая работа №1.1**

**Цель работы:**

По заданной схеме данных разработать базу данных, создать и сохранить скрипт с БД.

**Ход работы:**

Для выполнения работы использовалась программа Microsoft SQL Server. По выбранному варианту, была создана диаграмма, и заполнена таблицами. В каждой из таблиц были выставлены первичные ключи, и установлена у них «Спецификация идентификатора» на –Да. Также между таблицами были созданы связи, и к связям установлено правило «Каскадно».



Рисунок 1 Диаграмма

Также был создан скрипт диаграммы.

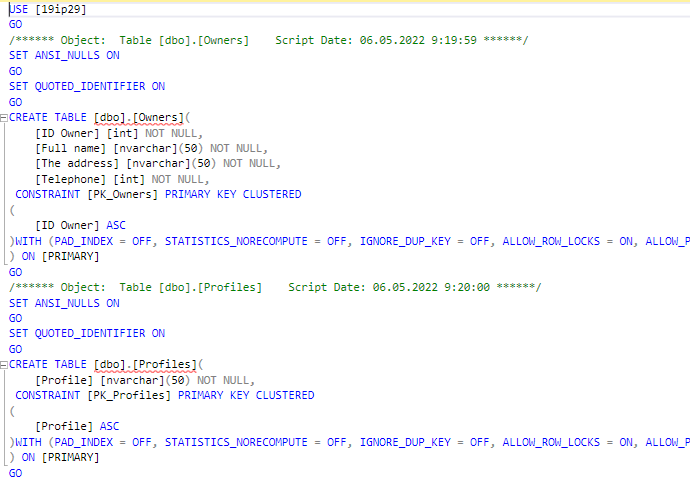


Рисунок 2 Часть скрипта

**Вывод:** в данной практической работе была разработана база данных по заданной схеме, а также был сохранен ее скрипт.

**Контрольные вопросы**

1. Что такое «база данных»?

Набор сведений, хранящихся некоторым упорядоченным способом.

1. Что такое первичный ключ (Primary key)?

Столбец, значения которого во всех строках различны.

1. Что такое внешний ключ (Foreign key)?

Это столбец (или группа столбцов), используемый в реляционной базе данных для связи данных между таблицами.

1. Какие связи существуют в реляционных базах данных?

Связь «Один к одному» - одно записи из одно таблицы соответствует другая запись из другой таблицы. Такая связь может использоваться, когда у вас имеется очень большая таблица, но часть полей будет использовать реже, чем остальная часть.

Связь «Один ко многим» - когда запись в одной таблице соответствует множеству записей в другой таблице.

Связь «Многие ко многим» - когда множество записей в одной таблице соответствуют множеству записей в другой таблице.

1. Что подразумевается под целостностью данных?

Соответствие имеющейся в [базе данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) информации её внутренней логике, структуре и всем явно заданным правилам. Каждое правило, налагающее некоторое ограничение на возможное состояние базы данных, называется [ограничением целостности](https://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%B3%D1%80%D0%B0%D0%BD%D0%B8%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D1%86%D0%B5%D0%BB%D0%BE%D1%81%D1%82%D0%BD%D0%BE%D1%81%D1%82%D0%B8&action=edit&redlink=1).

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.

Преимущества нормализации

Нормализация имеет целый ряд преимуществ:

* лучшая общая организация базы данных;
* сокращение числа ненужных повторений данных;
* согласованность данных внутри базы данных;
* более гибкая структура базы данных;
* эффективные возможности обеспечения безопасности и надежности базы данных.

**Практическая работа №1.2**

**Цель работы:**

По заданной схеме данных разработать базу данных, создать и сохранить скрипт с БД, реализовать программное приложение с возможностью авторизации в системе.

**Ход работы:**

Для работы откроем среду разработки Visual Studio и создадим новый проект «WPF». В качестве названия проекта укажите наименование вашей предметной области + «IS». 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». Зададим данному элементу название. Теперь нужно реализовать возможность работать с базой данных непосредственно из кода.

Для этого создадим новый класс, например, с названием DataBase.

В классе нам необходимо немного дописать код, чтобы с помощью данного класса мы могли иметь доступ к базе данных. Для этого:

1. Необходимо объявить статичную приватную переменную, которая будет являться объектом класса «Entities» (по умолчанию, если вы не меняли название при подключении базы данных).
2. Написать публичный статичный метод, который будет возвращать нам данный объект. В случае, если этот объект не был создан, необходимо создать его.

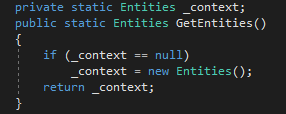


Рисунок 3

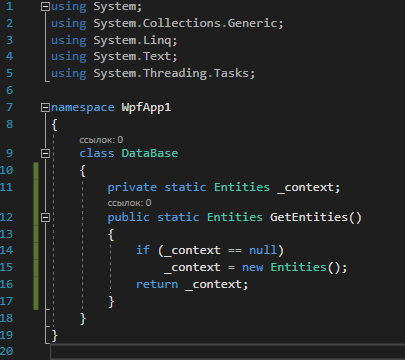


Рисунок 4

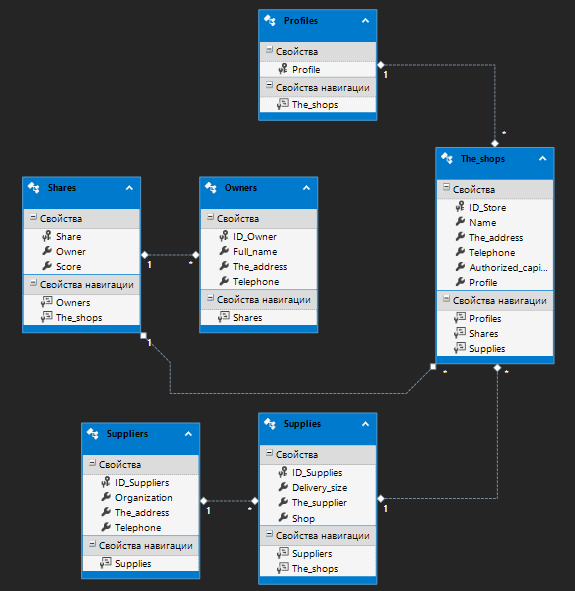


Рисунок 5

**Вывод:**

**Контрольные вопросы**

1. Как создать подключение к базе данных?

Для того чтобы создать подключение базы данных нужно в обозревателе решений создать новый элемент «Модель ADO.NET EDM». Задать название, после чего настроить соединение где мы указываем сервер, пароль. После чего выбираем таблицы для выгрузки и нажимаем готово.

**Практическая работа №1.3**

**Цель работы:**

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

**Ход работы:**

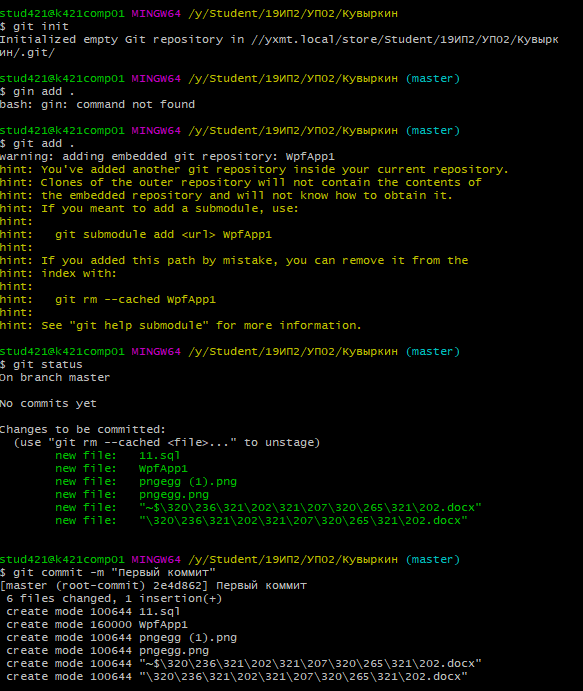


Рисунок 6

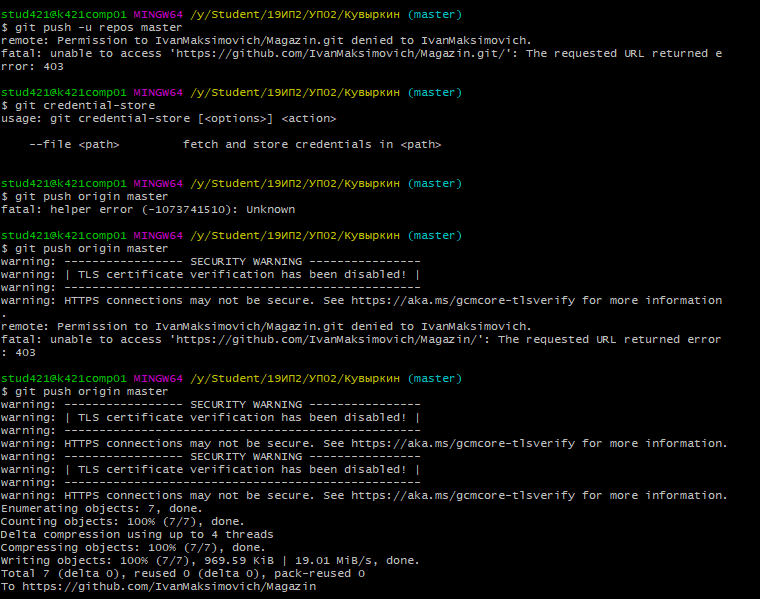


Рисунок 7

**Контрольные вопросы**

1. В чем заключается экономия времени при использовании системы контроля версий?
2. В чем преимущества использования системы контроля версий?
3. Что такое Git?

Одна из распределенных систем контроля версий.

1. Как начать использовать Git?
2. Как начать использовать GitHub?
3. Основные (наиболее часто используемые) команды Git.
4. Какие сервисы существуют для Git?
5. Как работать с локальным репозиторием?
6. Как работать с распределенным репозиторием?

**Вывод:**

**Практическая работа №1.4**

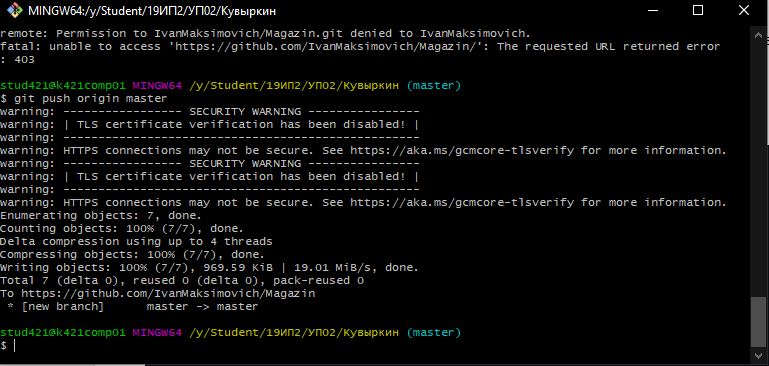


Рисунок 8

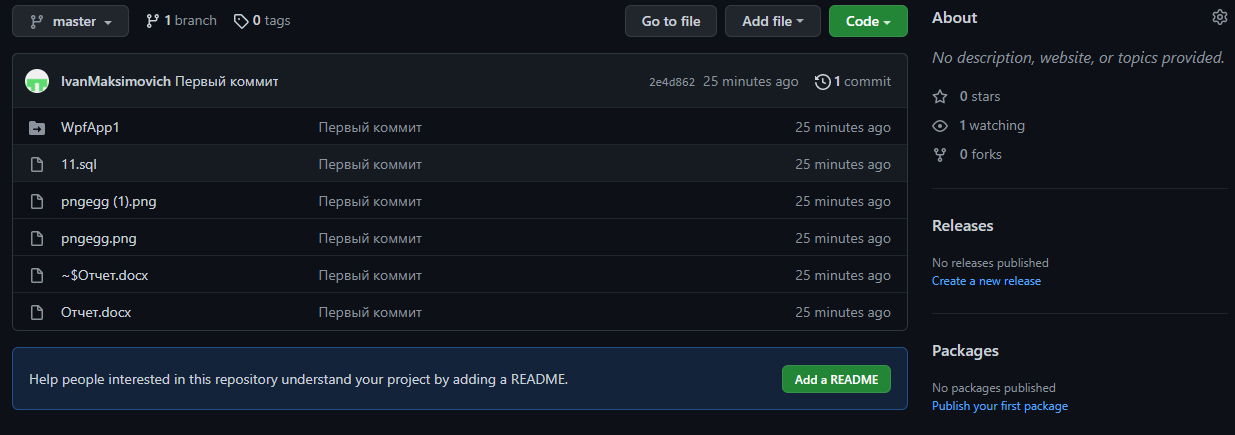


Рисунок 9

**Практическая работа №1.5**

**Цель работы:**

Реализовать авторизацию в системе.\

**Ход работы:**

**Вывод: ва**

**Практическая работа №2**

**Цель работы:**

Реализовать тестовый набор для авторизации пользователя в системе. Разработать тестовый сценарий и провести тестирование.

**Ход работы:**

**Аннотация теста**

|  |  |
| --- | --- |
| **Название проекта** | Sneakers shop |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Кувыркин Иван |
| **Дата(ы) теста** | 14.05.2022 |

**Тестовый пример #1:**

|  |  |
| --- | --- |
| **Тестовый пример #** | Test #1 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Регистрация |
| **Краткое изложение теста** | Ввод неверных данных |
| **Этапы теста** | 1. Пользователь заходит на страницу входа 2. Пользователь вводит логин и пароль 3. Если пароль и логин верны, то выходит окно приветствование. |
| **Тестовые данные** | Неверный логин, верный пароль. |
| **Ожидаемый результат** | Ошибка, выводиться сообщение неверный логин или пароль. |
| **Фактический результат** | Ошибка, выводиться сообщение неверный логин или пароль. |
| **Статус** | Зачет |
| **Предварительное условие** | - |
| **Постусловие** | - |
| **Примечания/комментарии** | - |

**Тестовый пример #2:**

|  |  |
| --- | --- |
| **Тестовый пример #** | Test #2 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Регистрация |
| **Краткое изложение теста** | Ввод верных данных |
| **Этапы теста** | 1. Пользователь заходит на страницу входа 2. Пользователь вводит логин и пароль 3. Если пароль и логин верны, то выходит окно приветствование. |
| **Тестовые данные** | Верный логин, верный пароль. |
| **Ожидаемый результат** | Сообщения приветствия, с именем учетной записи. |
| **Фактический результат** | Сообщения приветствия, с именем учетной записи. |
| **Статус** | Зачет |
| **Предварительное условие** | - |
| **Постусловие** | - |
| **Примечания/комментарии** | - |

**Тестовый пример #3:**

|  |  |
| --- | --- |
| **Тестовый пример #** | Test #3 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Регистрация |
| **Краткое изложение теста** | Отсутствие данных для входа |
| **Этапы теста** | 1. Пользователь заходит на страницу входа 2. Пользователь вводит логин и пароль 3. Если пароль и логин верны, то выходит окно приветствование. |
| **Тестовые данные** | Оставить пустые строки, логина и пароля. |
| **Ожидаемый результат** | Ошибка, выводиться сообщение неверный логин или пароль. |
| **Фактический результат** | Ошибка, выводиться сообщение неверный логин или пароль. |
| **Статус** | Зачет |
| **Предварительное условие** | - |
| **Постусловие** | - |
| **Примечания/комментарии** | - |

**Контрольные вопросы:**

1. **Что такое тест-кейс**Тест-кейс — это профессиональная документация тестировщика, последовательность действий, направленная на проверку какого-либо функционала, описывающая как придти к фактическому результату.
2. **Перечислите общие требования тест-кейсов  
   Требования к тест-кейсу**• Для измерения покрытия требований, требования к продукту должны быть проанализированы и впоследствии разбиты на пункты. Если тест-кейсы покрывают все требования, то может быть дан положительный или отрицательный ответ о реализации данного требования в продукте.  
   • Тест является хорошим в случае, когда он может обеспечить высокую вероятность обнаружения ошибки. Показать, что в программе полностью отсутствуют ошибки невозможно, поэтому процесс тестирования должен быть направлен на выявление прежде не найденных ошибок.  
   • Четкие, однозначные формулировки шагов. Описание шагов для прохождения тест-кейса должно содержать всю необходимую информацию, но при этом тест-кейс не должен быть слишком детализирован. Например, если тест-кейс содержит такие шаги, как авторизация, в описании необходимо указывать логин и пароль, но не нужно указывать в каком углу экрана находится окно авторизации.  
   • Отсутствие зависимостей тест-кейсов. Если тесты связанны между собой, становится проблематичным изменение, дополнение или удаление конкретного тест-кейса, появляется необходимость изменять связанные с ним тесты. Более того, взаимосвязанные тесты обладают конкретный сценарием от перехода одного теста к другому. Это приведет к тому что не все сценарии перехода от одного теста к другому будут протестированы и появляется вероятность пропустить баг.  
   • Ожидаемый результат необходимо прогнозировать заранее и прописывать его в тест-кейсе. Если ожидаемый результат не определить заранее, может возникнуть ситуация, когда тестировщик видит то, что он хочет увидеть. При заранее определенном результате тестировщику необходимо только сравнить ожидаемый результат с фактическим.  
   • Необходимо уделять внимание не только тестам, которые проверяют правильные данные, но и тем тестам, которые проверяют работу программы при неправильных, непредусмотренных данных. Большое количество ошибок связано именно с теми действиями пользователя, которые не предусмотрены программой.  
   • Также необходимо проверять, не делает ли программа то, чего не должна. Нужно производить проверку на нежелательные побочные эффекты.

**Вывод:** в данной практической работе был реализован тестовый набор для авторизации пользователя в системе. Разработаны сценарии и проведено тестирование.

**Практическая работа №3**

**Цель работы:**

Создать главное окно с возможностью навигации по страницам.

**Ход работы:**

**Практическая работа №4**

**Цель работы:**

Реализовать отображение данных базы данных на форме.

**Ход работы:**

**Контрольные вопросы**

1. Что такое Windows Presentation Foundation? В чем преимущество перед WinForms?
2. Что такое XAML? Для чего он используется в WPF?
3. Что такое LINQ? Как работает технология LINQ?