**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc99786011)

[1 Общая часть 4](#_Toc99786012)

[1.1 Информационная система предприятия 4](#_Toc99786013)

[1.2 Описание предметной области 4](#_Toc99786014)

[2 Технологии администрирования базы данных 5](#_Toc99786015)

[2.1 Основные утилиты администратора баз данных 5](#_Toc99786016)

[2.2 Построение схемы базы данных 5](#_Toc99786017)

[2.3 Разработка требований к серверу базы данных 5](#_Toc99786018)

[3 Реализация 9](#_Toc99786019)

[3.1 Настройка и подключение системы контроля версий 9](#_Toc99786020)

[3.2 Администрирование базы данных 1](#_Toc99786021)

[3.2.1 Разработка объектов базы данных 1](#_Toc99786022)

[3.2.2 Реализация CRUD 1](#_Toc99786023)

[3.2.3 Фильтрация и поиск данных 1](#_Toc99786024)

[Вывод 2](#_Toc99786025)

[Приложение 3](#_Toc99786026)

[Приложение А. Программный код в системе контроля версий 3](#_Toc99786027)

# ВВЕДЕНИЕ

Производственная практика пройдена в ООО «ЖИЛФОНД-АБАКАН». Данный способ хранения информации не очень надежен, так как в случаи утраты их будет сложно или невозможно восстановить, поэтому было принято решение создать базу данных, для хранения этой информации, а также программный модуль для работы с ней.

Целью производственной практики является создание базы данных с информацией о выпускниках колледжа и разработка приложения для работы с базой данных.

Задачи на пути достижения цели:

* Оцифровать фотографии групп и выпускников;
* Изучить утилиты администратора баз данных;
* Выполнить подключение к системе контроля версий;
* Спроектировать базу данных;
* Разработать приложение для работы с БД;
* Реализовать возможности поиска и фильтрации данных;

# Общая часть

**1.1 Информационная система предприятия**

Государственное бюджетное профессиональное образовательное учреждение Республики Хакасия "Хакасский политехнический колледж", основан 25 декабря 1995 г по адресу 655012, республика Хакасия, город Абакан, улица Пушкина, дом 30. Компании был присвоен ОГРН 1021900535975 и выдан ИНН 1901027433.

Организация занимается подготовкой специалистов среднего профиля по разным направлениям: бухгалтерский учет, информационные системы и программирование, техническое обслуживание и ремонт двигателей, систем и агрегатов автомобилей, технологии деревообработки и т.д.

В организации главным является директор, у него есть заместители по разным вопросам. Так же есть различные отделения, которыми управляют заведующие, как очными, так и заочными.

Колледж работает 6 дней в неделю, в будние дни работа начинается с 8:00 утра и заканчивается в 17:00 часов вечера. Предусмотрены обеденные перерывы, так как в организации две смены обучения, то и перерывов тоже два: первый начинается с 11:10 до 11:40, второй начинается в 14:50 и заканчивается в 15:20. В субботу работа начинается с 8:00 часов утра до 14:30 дня.

**1.2 Описание предметной области**

Была выбрана предметная область Выпускники Хакасского Политехнического колледжа. С информацией работает заведующий музеем колледжа. Документы хранятся в виде фотографий, в альбомах.

# 2 Технологии администрирования базы данных

**2.1 Основные утилиты администратора баз данных**

Import и Export. Две дополнительные утилиты, поставляемые корпарацией Oracle. Они в основном применяются для копирования и восстановления данных и для перемещения данных либо в другую базу данных Oracle, либо из более старой версии в более новую.

Возможности утилит Import и Export:

* Хранение данных в файлах операционной системы для архивирования;
* Выборочное резервное копирование частей базы данных;
* Перемещение данных из одной пользовательской схемы в другую;
* Экономия пространства и повышение производительности за счёт уменьшения фрагментации;

Работа с утилитами Import и Export весьма проста. Утилита Export записывает информацию о таблицах или объектах базы данных, такую как операторы создания таблицы, операторы создания индекса, разрешения на таблицу, информация о размерах и т.д., а также данные из самих таблиц Oracle. Затем утилита Export сохраняет эту информацию в именованных файлах операционной системы. Файлы операционной системы, создаваемые утилитой Export, известны как файлы дампа. Файлы дампа, которые представлены в двоичном формате Oracle, применяются главным образом только в утилите Import. Можно назвать Файл дампа любым именем, допустимым в операционной системе. Если вы не укажите имя выходного файла для утилиты Export, то по умолчанию будет принято имя EXPDAT.DMP.

Затем можно сохранить выходные файлы, созданные Export, на диске или записать на съемный носитель для дальнейшего хранения, либо воспользоваться утилитой Import для воссоздания экспортируемых данных в целях восстановления или ведения базы данных.

SQL\*Loader — специальная утилита Oracle сервера для загрузки данных в БД из локальных файлов с данными.

Возможности утилиты:

* Загрузка данных из разных файлов в одной сессии;
* Загрузка данных в разные таблицы в одной сессии;
* Указание кодировки файла данных;
* Использование SQL функций перед загрузкой;
* Генерация уникальных ключей для колонок;
* Ограничение вставляемых данных по условию;
* Загрузка двоичных данных;
* Запись ошибочных данных в файл ошибок;

**2.2 Построение схемы базы данных**

**2.3 Разработка требований к серверу базы данных**

**Системные характеристики**

1. СХ-1: Приложение является настольным.
2. СХ-2: Приложение разрабатывается на языке программирования C#.

Существует ли некая специфика настройки интерпретатора .NET Framework для корректной работы приложения? (Должен работать соответствует спецификации Common Language Infrastructure).

Настраиваете ли вы на реализации приложения на С#? Если да, то почему (Только C#. У нас только сотрудник его знает)

Должна ли в руководстве пользователя быть описан процедура установки и настройки интерпретатора С#? (Нет).

1. СХ-3: Приложение является кроссплатформенным.

Какие ОС должны поддерживаться? (Любая, где работает .Net).

В чем вообще цель кроссплатформенности? (Лучшее решение для быстрого и эффективного запуска).

**Пользовательские требования**

1. ПТ-1: Запуск и остановка приложения.
2. ПТ-1.1: Запуска приложения производится из консоли командой «C#».

Какие параметры передаются скрипту при запуске? (Каталог с исходными файлами, каталог с конечными файлами).

Какова реакция скрипта на:

* Отсутствие параметров; (Пишет ошибку).
* Неверное количество параметров; (Пишет ошибку и поясняет, что не так).
* Неверное значение каждого из параметров (Пишет ошибку и поясняет, что не так).

1. ПТ-1.2: Остановка приложения производится выполнение команды Alt+F4.
2. ПТ-2: Конфигурирование приложения.
3. ПТ-2.1: Конфигурирование приложения сводится к указанию путей в файловой системе.
4. ПТ-2.2: Целевой кодировкой является .NET.
5. ПТ-3: Просмотр журнала работы приложения.

Предполагается ли указание целевой кодировки, или .NET используется в качестве целевой всегда? (Только .NET, других не надо).

1. ПТ-3.1: В процессе работы приложение должно выводить журнал своей работы в консоль и лог-файл, имя которого определяется правилами.

Каков формат журнала? (Дата-время)

Различаются ли форматы журнала для консоли и лог файла? (Нет).

1. ПТ-3.2: При первом запуске приложение лог-файл создается, а при последующих – дописывается.

Как приложение различает свой первый и последующие запуски? (Никак).

**Бизнес-правила**

1. БП-1: Источник и приемник файлов
2. БП-1.1: Каталоги, являющихся источником исходных (опечатка, исходных) (Да) и приемников конечных файлов, не должны совпадать.

Какова реакция приложения в случае совпадения этих каталогов? (Пишет ошибку и поясняет, что не так).

1. БП-1.2: Каталог, являющийся приемником конечных файлов, не может быть подкаталогом источника.

**Атрибуты качества**

1. АК-1: Производительность
2. АК-1.1: Приложение должно обеспечивать скорость обработки данных 5 МБ/сек.

При каких технических характеристиках системы? (i5,4GB RAM).

1. АК-2: Устойчивость к входным данным
2. АК-2.1: Приложение должно обрабатывать входные файлы размеров до 50 МБ включительно.

Какова реакция приложения на файлы, размер которых превышает 50 МБ? (Не трогает).

1. АК-2.2: Если входной файл не является текстовым, приложение должно произвести обработку.

Обработку чего должно произвести приложение? (Этого файла. Не важно, что станет с файлом, лишь бы скрипт не умер).

**Ограничения**

1. О-1: Приложение разрабатывается на языке программирования C#, использование которого обусловлено возможностью заказчика осуществляется поддержку приложения силами собственного IT-отдела.
2. О-2: Процедура установки и настройки интерпретатора выходят за рамки данного проекта и не описываются в документации.
3. О-3: Кроссплатформенные возможности приложения сводятся к способности работать по OC семейства Windows и Linux, поддерживающих работу интерпретатора версии.
4. О-4: Целевая кодировка .NET является жестко заданной, и ее изменение в процессе эксплуатации приложения не предусмотрено.

**Детальные спецификации**

1. ДС-1: Интерпретатор
2. ДС-1.1: Минимальная версия – 4.8.
3. ДС-2: Параметры комодной строки.
4. ДС-2.1: При указании излишнего количества параметров комодной строки приложение должно игнорировать все параметры комодной строки.
5. ДС-2.2: При указании неверно значения любого из параметров комодной строки приложение должно завершить работу, выдать сообщение об использовании, а также сообщив имя неверного указанного параметра, его значение и суть ошибки.
6. ДС-3: Сообщения.
7. ДС-3.1: Сообщения об ошибках:

* Directory not exists or inaccessible.
* Destination dir may not reside within source dir tree.
* Wring file name or inaccessible path.

1. ДС-4: Форматы и размеры файлов.
2. ДС-4.1: Приложение должно обрабатывать текстовые файлы на русском и английском языках в следующих исходных кодировках: WIN1251, CP866, KOI8R.

Обрабатываемы файлы могут быть представлены в следующих форматах, определяемых расширениями файлов:

* Plain Text(TXT);
* Hyper Text Markup Language Document (HTML);
* Mark Down Document (MD);
* Joint Photographic Experts Group (JPEG);
* Bitmap Picture (BMP).

1. ДС-4.2: Приложение должно обрабатывать файлы размеров до 50 МБ (включительно), игнорируя любой файл, размер которого превышает 50 МБ.
2. ДС-4.3: Если файл с расширение из ДС-4.1 содержит внутри себя данные, не соответствующие формату файла, допускается повреждение таких данных.

# 3 Реализация

**3.1 Настройка и подключение системы контроля версий**

Для подключения проекта к системе контроля версий GitHub необходимо в верхнем левом углу visual studio нажать на меню Git, затем выбрать создать репозиторий. Меню Git в visual studio представлено на рисунке 5.

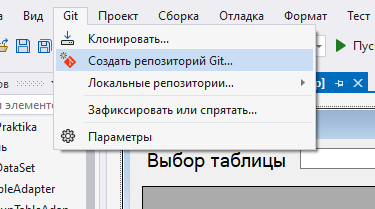


Рисунок 5 – Меню Git

После этого необходимо создать репозиторий Git в открывшемся окне. Окно создания репозитория представлено на рисунке 6.

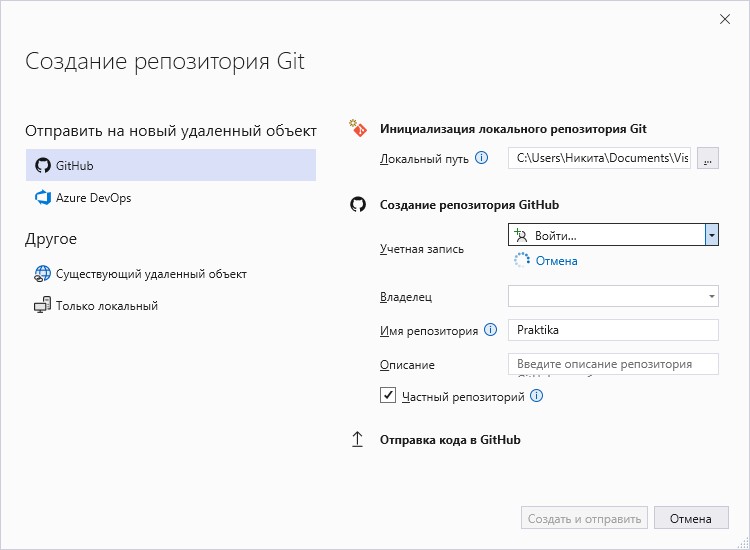


Рисунок 6 – Окно авторизации Git

После выбора учетной записи откроется окно для авторизации на сайте. После авторизации необходимо нажать на кнопку “Создать и отправить”. Чтобы убедиться, в том, что репозиторий создан, необходимо перейти на сайт Github и после авторизации перейти во вкладку Repositories. Добавленный репозиторий показан на рисунке 7.

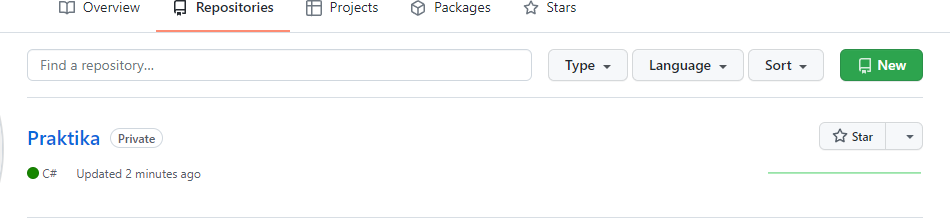


Рисунок 7 – Репозиторий на сайте GitHub

Для отправки изменений в Git необходимо нажать на счетчик изменений в правом нижнем углу. Откроется окно с изменениями проекта, которое показано на рисунке 8. В данном окне будут показаны измененные файлы. При необходимости можно посмотреть, что именно было изменено в файле.

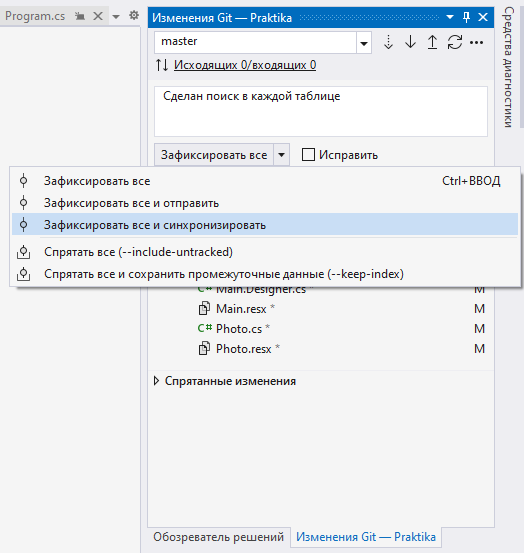


Рисунок 8 – Окно изменений Git

Чтобы отправить новую версию программы необходимо ввести описание изменений в специальное окно, затем нажать «Зафиксировать все и синхронизировать». После этого изменения появятся на сайте. Сохранение изменений на сайте показан на рисунке 9.

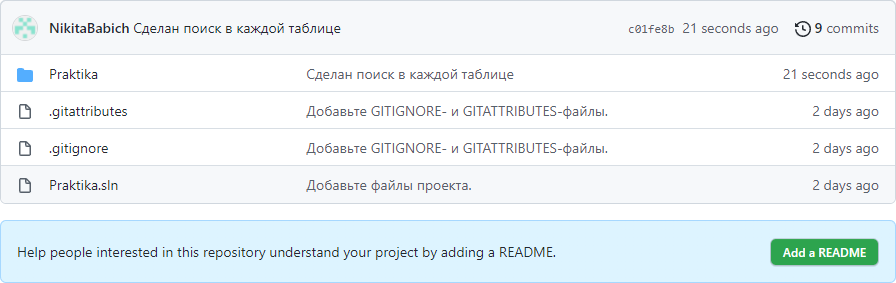


Рисунок 9 – Сохранение изменений на сайте GiHub

**3.2 Администрирование базы данных**

**3.2.1 Разработка объектов базы данных**

Для разработки базы данных была выбрана СУБД MSSQL Management Studio, так как она идеально подходит для реализации небольшой базы данных. Была спроектирована база данных, которая содержит в себе 9 таблиц: Таблица с учителями, таблица со студентами, таблица с группами, и таблица, в которой содержится информация о студенте и группе, в которой он обучался. Диаграмма базы и связи между таблицами показаны на рисунке 10.

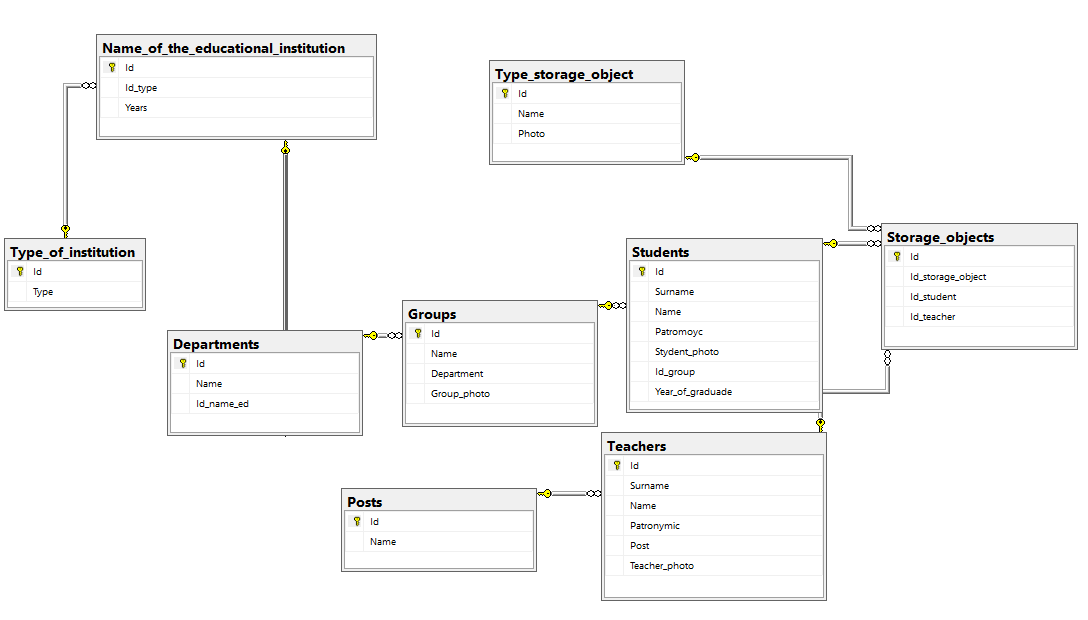


Рисунок 10 – Диаграмма БД

**3.2.2 Реализация CRUD**

**3.2.3 Фильтрация и поиск данных**

# Вывод

В ходе прохождения производственной практики была реализована база данных и приложение для хранения и обработки информации о студентах Хакасского политехнического колледжа. Также в ходе прохождения практики были оцифрованы фотографии выпускников колледжа, для добавления фотографий в базу данных. База данных была реализована при помощи СУБД Microsoft SQL Server и Microsoft SQL Server Management Studio. Сама программа была разработана в среде программирования Visual Studio с использованьем языка программирования C#. В ходе разработки применялась технология LINQ to SQL. В данном приложении было реализовано пять форм: «Главная форма», «Студенты», «Учителя», «Группы» и «Студенты-группы». На главной форме приложения пользователь может просматривать таблицы с данными и выбирать с какой таблицей работать. Также на главной форме пользователь может искать данные при помощи текста. В данном приложении были реализованы возможности добавления, редактирования и удаления данных о студентах, учителях и группах, которые выполняют свои функции на определенных формах.

Данное приложение можно использовать в музее колледжа, чтобы желающие могли узнать кто обучался в колледже, и в каком году его закончила та или иная группа. Так же данное приложение можно переделать под сайт, и внедрить его на сайт колледжа, чтобы информацию о выпускниках могли просматривать люди, которые не могут посетить музей в колледже.

# Приложение

**Приложение А. Программный код в системе контроля версий**