Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра Информационные технологии и автоматизированные системы

Курсовая работа по дисциплине

«Базы данных»

Вариант 2

Тема: «Разработка базы данных “Учет успеваемости студентов ВУЗа”»

Выполнил: студент группы РИС-20-1Б

Азмагулов А.В.

Проверил: Доцент кафедры ИТАС

к.т.н. Петренко А. А.

г. Пермь – 2022

# **Оглавление**

[Оглавление 2](#_Toc103798224)

[Реферат 3](#_Toc103798225)

[Введение 4](#_Toc103798226)

[1. Анализ предметной области 6](#_Toc103798227)

[1.1.Границы исследования базы данных. 7](#_Toc103798228)

[2. Проектирование базы данных 8](#_Toc103798229)

[2.1. Концептуальная схема Базы данных 8](#_Toc103798230)

[2.2 Логическая модель базы данных 9](#_Toc103798231)

[2.3 Выбор СУБД 10](#_Toc103798232)

[2.4 Построение физической модели базы данных 11](#_Toc103798233)

[3. Технология реализации программы 13](#_Toc103798234)

[3.1 Выбор языка программирования 13](#_Toc103798235)

[3.2 Реализация функций и интерфейса 13](#_Toc103798236)

[3.3 Разграничение прав доступа 17](#_Toc103798237)

[3.4 Тестирование 19](#_Toc103798238)

[Заключение 23](#_Toc103798239)

[Список литературы 24](#_Toc103798240)

# **Реферат**

Общий объем: 29 стр. Количество иллюстраций: 10. Количество таблиц: 2. Число использованных источников информации: 5.

Ключевые слова: Реляционная база данных, база данных, модели баз данных, Концептуальная модель, Логическая модель, Физическая модель, таблица, СУБД, Учет успеваемости.

Объект исследования: Учет успеваемости студентов высшего учебного заведения.

Предмет исследования: База данных для хранения информации об успеваемости студентов вуза.

Цель работы: Разработать базу данных для хранения информации об учебном процессе ВУЗа и интерфейс для работы пользователя с БД

Задачи:

1. Провести анализ предметной области.

2.Выбрать подходящее программное обеспечение для реализации программы

3.Разработать концептуальную, логическую и физическую модели данных

4.Разработать базу данных

5.Разработать программу для работы с базой данных

Методы исследования: Анализ, сравнение, эксперимент.

В ходе работы была реализована база данных для хранения информации об успеваемости студентов вуза, а также пользовательский интерфейс для работы с БД

# **Введение**

С начала всего своего существования человечество стремилось к сохранению и обработке информации. Ранее такую функцию выполняли наскальные рисунки и папирусы, сегодня же такую задачу выполняют базы данных и информационные системы, которые на сегодняшний день стали неотъемлемой частью общества.

Базы данных - это совокупность сведений (о реальных объектах, процессах, событиях или явлениях), относящихся к определенной теме или задаче, организованная таким образом, чтобы обеспечить удобное представление этой совокупности, как в целом, так и любой ее части.

Данная работа рассматривает процесс разработки информационной системы, в том числе этапы проектирования логической, концептуальной и физической моделей базы данных, ее реализацию и создание приложения по управлению Базы данных.

Целью курсовой работы является разработка базы данных для учета успеваемости студентов вуза, а так же интерфейс для работы пользователя с базой данных.

Постановка задачи: Создайте базу данных Учет успеваемости студентов, разработайте формы ввода, корректировки и просмотра данных.

В университете обучаются студенты разных специальностей на нескольких различных факультетах. Каждая специальность относится только к одному факультету, но на факультете несколько специальностей. На каждой специальности обучается много студентов в нескольких группах. Специальность может иметь несколько групп, а может и одну. Студенты изучают несколько дисциплин и проходят испытания (зачеты, экзамены, курсовые и контрольные работы). Результаты испытаний заносятся в базу данных.

Необходимо разработать систему для решения с использованием информации базы данных следующих задач:

1) получите списки студентов указанной группы;

2) получите списки студентов, сдававших указанные экзамены;

3) получите списки студентов, пересдававших указанный экзамен, а также количество пересдач по каждому пересдававшему;

4) получите списки назначенных на стипендию;

5) получите списки предметов, которые сдаются на указанном факультете;

6) получите списки для военкомата (юноши указанного года рождения);

7) получите списки студентов и их групп указанного факультета;

8) получите список предметов, сдававшихся указанным студентом в указанный период.

1. Анализ предметной области

База данных — это набор упорядоченной, структурированной информации, которая хранится в электронном виде в компьютерной системе.

Управление БД производится с помощью систем управления базами данных (СУБД). СУБД — комплекс программ, позволяющих создать базу данных (БД) и манипулировать данными (вставлять, обновлять, удалять и выбирать). Такая система обеспечивает безопасность, надёжность хранения и целостность данных, а также предоставляет средства для администрирования БД.

Разработка пользовательского интерфейса (программы) потребует использование интегрированной среды разработки, а также структуру, которая станет основой интерфейса.

Для разработки базы данных необходимо использовать СУБД, которая предоставит все необходимые возможности для реализации требований к программе.

Учет успеваемости студентов – это набор данных, хранящий в себе информацию о студентах, их дисциплинах и результатов испытаний данных дисциплин. Такой учет в полной мере позволяет пользователю выяснить о успеваемости каждого студента, нагрузку на студентов, а так же получить информацию о учащихся определённых групп определенного факультета.

Как правило, учет успеваемости включает в себя информацию о студентах, дисциплинах, изучаемых студентом, а так же испытаниях, пройденных студентом по определенной дисциплине. Однако, помимо вышеперечисленных элементов, техническое задание требует добавить к успеваемости студентов информацию о факультете обучающегося, его группу, а так же возможность отслеживания пересдач студента.

## **1.1.Границы исследования базы данных.**

Границы исследования – это факторы и переменные, которые не должны быть включены в исследование. В учете успеваемости студентов данной работы границами исследования являются следующие условности:

1. Верхний предел количества хранимых объектов в одной таблице равняется тысяче.
2. Студенты, с одинаковыми именами, фамилиями, отчествами и годами рождения воспринимаются как единый человек.
3. Право пересдачи дисциплины исключает возможность ее пересдачи на более низкую оценку.
4. При неявке студента на испытание, данная информация записывается в базу данных как получение студентом неудовлетворительной оценки.
5. Право пересдачи испытания исключат ее пересдачу в тот же день, что и день испытания. Так же исключается возможность неоднократной пересдачи дисциплины в один и тот же день.
6. Проведение испытания по какой-либо дисциплине возможно лишь одной группой в один назначенный день.

**Вывод**

База данных должна предоставлять возможность хранить и обрабатывать необходимые данные. Приложение должно предоставлять возможность пользователю взаимодействовать с базой данных и данными, хранящимися в ней.

2. Проектирование базы данных

Необходимые для разработки базы данных модели, будут созданы с помощью инструмента Draw.IO, так как он является наиболее удобным среди прочих и отвечает всем требованиям, которые необходимы для моделирования БД.[5]

Средством моделирования предметной области на этапе проектирования является модель «сущность-связь». Часто ее называют ER-моделью. В ней моделирование структуры данных предметной области базируется на использовании графических средств - ER-диаграмм. В наглядном виде они представляют связи между сущностями. Основными понятиями ER-диаграммы являются сущность, атрибут, связь. Сущность представляет собой объект, информация о котором хранится в базе данных. Сущность имеет экземпляры, отличающиеся друг от друга значениями атрибутов и допускающие однозначную идентификацию. Атрибут - это свойство сущности. Атрибут, который уникальным образом идентифицирует экземпляры сущности, называется ключом.

2.1. Концептуальная схема Базы данных

Концептуальная схема – определяющая представление базы данных, единое для всех ее приложений и не зависящее от используемого в СУБД представления данных в среде хранения. Концептуальная модель базы данных построена по нотации Питера Чена, как самой наглядной из существующих.

Ход работы:

1. Создание таблиц, отражающую необходимую информацию в учете успеваемости студентов
2. Создание связей между таблицами, которые позволят пользователю корректно взаимодействовать с нужными данными.

Концептуальная модель представлена на Рисунке 1.

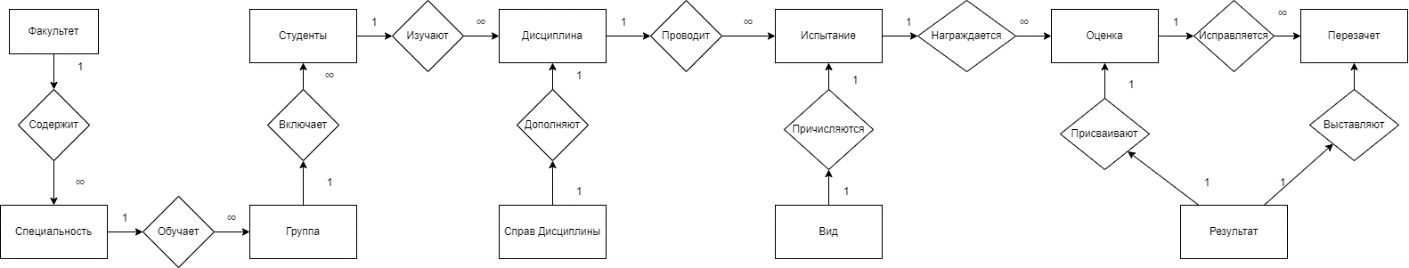


Рисунок 1 - Концептуальная модель базы данных

Концептуальная модель базы данных содержит 11 сущностей и 11 связей, корректно отображающих информацию базы данных “Учет успеваемости Студентов”.

## **2.2 Логическая модель базы данных**

Логическая модель описывает понятия предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью. Логическая модель строится в терминах информационных единиц, но без привязки к конкретной СУБД.

Во время разработки логической модели данных определяются такие понятия, как первичный ключ - в реляционной модели данных один из потенциальных ключей отношения, выбранный в качестве основного ключа, внешний ключ - столбец или комбинация столбцов, значения которых соответствуют Первичному ключу в другой таблице

Логическая модель была построена по нотации Мартина, как наиболее наглядная и легкоусвояемая нотация из имеющихся нотаций на сегодняшний день.

Ход работы:

1. Определение первичных и внешних ключей для каждой сущности базы данных.
2. Построение связей непосредственно между соответствующими ключами сущностей.

Логическая модель изображена на Рисунке 2.

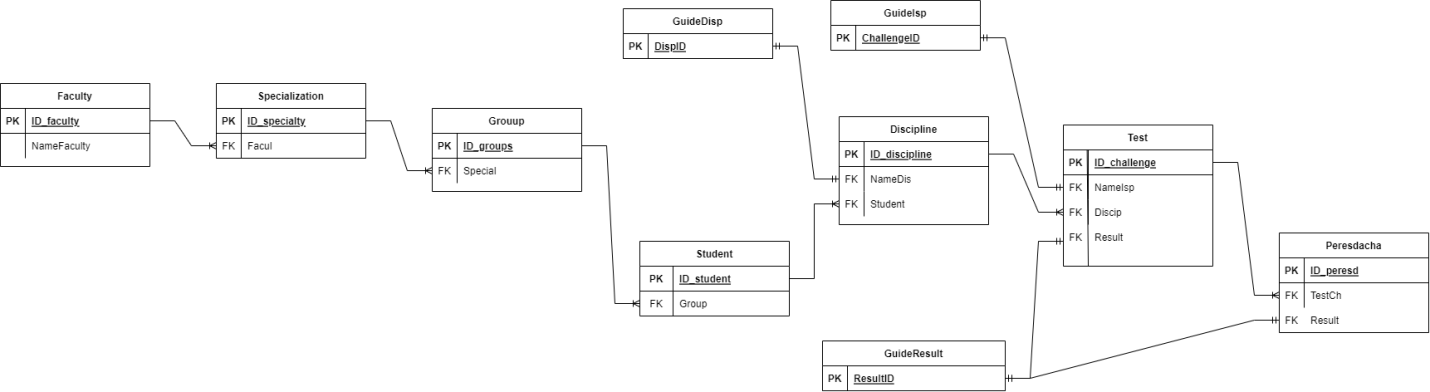


Рисунок 2 - Логическая модель базы данных

## **2.3 Выбор СУБД**

После составления логической модели, на ее основании была выбрана СУБД для перехода на дальнейшие пункты работы. Для выбора подходящего СУБД был проведен сравнительный анализ двух СУБД (Таблица 3).

Таблица 1 - Сравнение различных СУБД

|  |  |
| --- | --- |
| MS SQL Server | MySQL |
| Небольшое количество справочного материала для работы с СУБД | Большое количество справочного материала для работы с СУБД |
| Работа с БД осуществляется с помощью ПО SQL Server Management Studio, что приводит к необходимости устанавливать дополнительное ПО | Работа с БД осуществляется с помощью веб-интерфейса PhpMyAdmin, что позволяет работать без установки лишнего программного обеспечения. |
| Наличие всего необходимого в одной сборке, привычная незамысловатость установки и идеальное взаимодействие с другими продуктами Microsoft делают SQL Server системой, понятной даже неопытному программисту | Настройки MySQL и поиска наиболее подходящих расширений придется потратить время |

Исходя из вышеописанной сравнительной таблицы, принято решение по использованию MS SQL Server как основного вида СУБД в дальнейшей работе, так как данный инструмент является наиболее простым в понимании и использовании.

## **2.4 Построение физической модели базы данных**

Физическая модель базы данных — это модель данных, которая определяет, каким образом представляются данные, и содержит все детали, необходимые СУБД для создания базы данных. Физическая модель базы данных строится на основе логической и содержит все детали, необходимые конкретной СУБД для создания базы: наименования таблиц и столбцов, типы данных, определения первичных и внешних ключей.

Ход работы:

1. Добавление не ключевых полей в таблицы.
2. Определение типов данных для каждого поля.
3. Определение авто-инкрементных и ненулевых полей.
4. Определение дефолтных значений и проверок CHECK в определенных сущностях

Во время разработки физической модели данных было принято решение использовать следующие типы данных:

1. INT – целочисленный тип данных.
2. VARCHAR(x) - Строковый тип, может содержать цифры, латиницу, кириллицу и спецсимволы. Длина строки <=x.
3. DATE - тип данных содержащий информацию о дате в формате “ГГГГ-ММ-ДД”

Физическая модель была построена по нотации Мартина.

Физическая модель изображена на Рисунке 3.

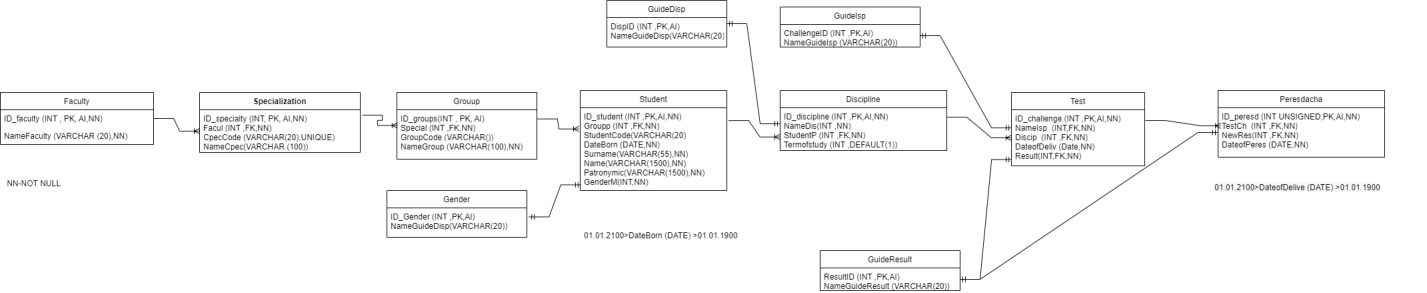


Рисунок 3 - Физическая модель базы данных

**Вывод**

База данных будет реализована с помощью СУБД MS SQL Server. Для отражения всех необходимых данных и запросов будет создано 11 сущностей и 11 связей.

# **3. Технология реализации программы**

## **3.1 Выбор языка программирования**

После составления физической модели, на ее основании был выбран язык программирования для перехода на разработку программы для работы с базами данных. Для выбора подходящего языка программирования была составлена сравнительная таблица

Таблица 2 - Сравнение различных языков программирования

|  |  |
| --- | --- |
| C# | C++ |
| Включает в себя большое количество актуальных библиотек, ускоряющих и упрощающих разработку приложения | Содержит в себе лишь базовые библиотеки, не дающие пользователю серьезного преимущества в скорости разработки по сравнению с С# |
| Легкость сборки проектов на начальных этапах создания приложений | Сложная сборка проектов на начальных этапах разработки, но более гибкая реализация в сравнении со сборкой в C# |
| Наличие готового Фреймворка, готового для непосредственной разработки приложения | Отсутствие готового Фреймворка и необходимость дополнительной установки приложений для создания интерфейса приложения |

На основании таблицы сравнения принято решение по реализации приложения на языке С# на [NET Framework.](https://docs.microsoft.com/ru-ru/dotnet/framework/get-started/overview" \t "_blank)

## **3.2 Реализация функций и интерфейса**

Для отображения таблиц используется элемент Windows Form - DataGridView, который позволяет корректно выводить данные из заданных сущностей (Рисунок 4).

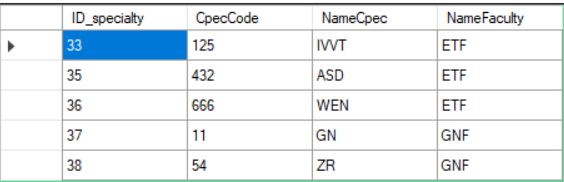


Рисунок 4 - Элемент интерфейса DataGridView

Меню для выбора отображаемой таблицы или требуемого запроса реализовано с помощью набора инструментов tabControl, для более плавного переключения между таблицами по сравнению с переключением сущностей по нажатию кнопки(Рисунок 5).

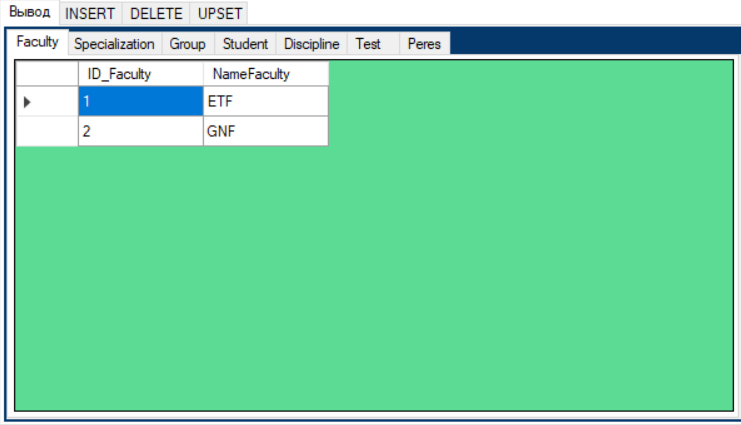


Рисунок 5 - Элемент интерфейса tabControl

Заполнение таблиц и выбор параметров для требуемого запроса реализован с помощью элементов интерфейса ComboBox и TextBox (Рисунок 6).

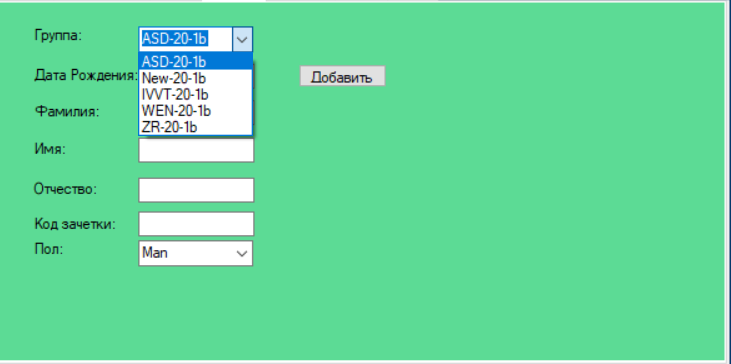


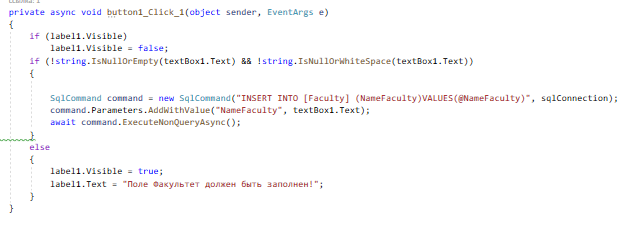
Рисунок 6 - Элементы интерфейса ComboBox и TextBox

Подключение к базе данных реализовано с помощью библиотеки

“ System.Data.SqlClient; ”, а также строки, непосредственно в которую вводятся параметры подключения(Листинг 1).

Листинг 1 - Подключение к БД 

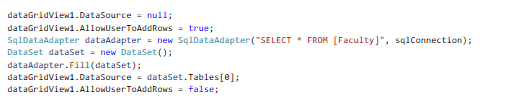
Выполнение команд, выбранных пользователем, реализованы при помощи служебного инструмента Button, задача которого вне зависимости от выбранной функции-отправление запроса. Пример функции отправки запроса к БД представлен в “Листинг 2” в виде функции button1\_Click\_1, задача которого заключается в добавлении нового факультета в таблицу Faculty.



Листинг 2 - Функция button1\_Click\_1

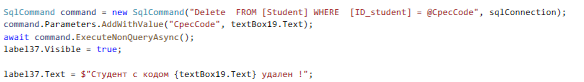
Получение данных из сущности БД выполнено с помощью запроса “SELECT”(Листинг 3).

Листинг 3 – Пример вывода элементов из таблицы



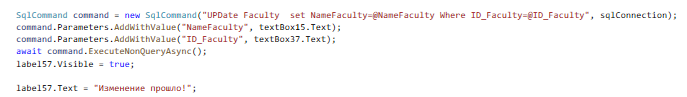
Удаление данных из сущности БД выполнено с помощью запроса   
“DELETE” (Листинг 4).

Листинг 4 – Пример Удаления элемента из таблицы



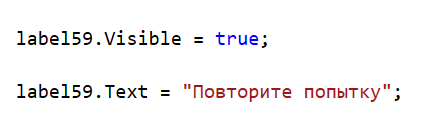
Изменение данных в сущности БД выполнено с помощью запроса “UPDATE”(Листинг 5).

Листинг 5 - Пример Обновления элемента из таблицы



Вывод информации о некорректном вводе выводится пользователю через инструмент label, высвобождающийся из скрытого состояния в нужный момент времени. Пример предупреждения пользователя расписан в Листинг 6.

Листинг 6 - Вызов сообщения о некорректности данных



Представленные функции отражают выполнение запросов только к одной из сущностей БД, к другим таблицам запросы выполняются аналогично.

## **3.3 Разграничение прав доступа**

Разграничение прав доступа разделяется на 2 пользователя: Admin и User. Для входа за конкретное лицо пользователю предлагается открытая форма с Логином и Паролем (Рисунок 7).

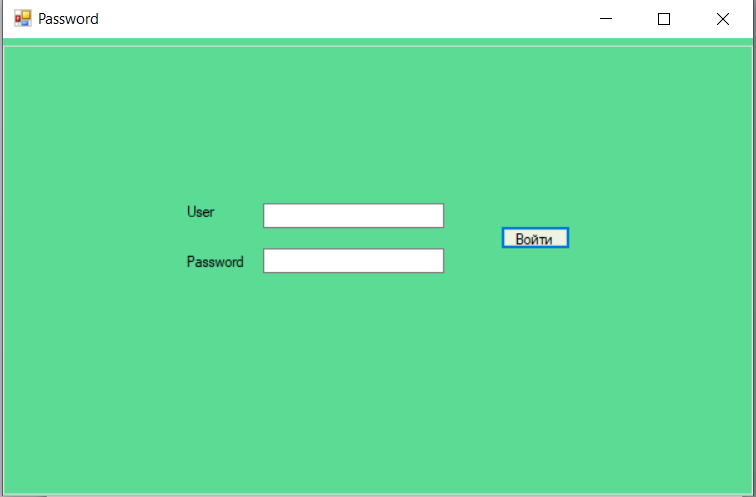


Рисунок 7 – Форма Логин-Пароль

При введенных значениях логина и пароля соответственно “User” и “123” пользователя переносит на форму Пользователя(Рисунок 8), где ему предоставляются кнопки для вывода информации по расписанным ранее запросам.

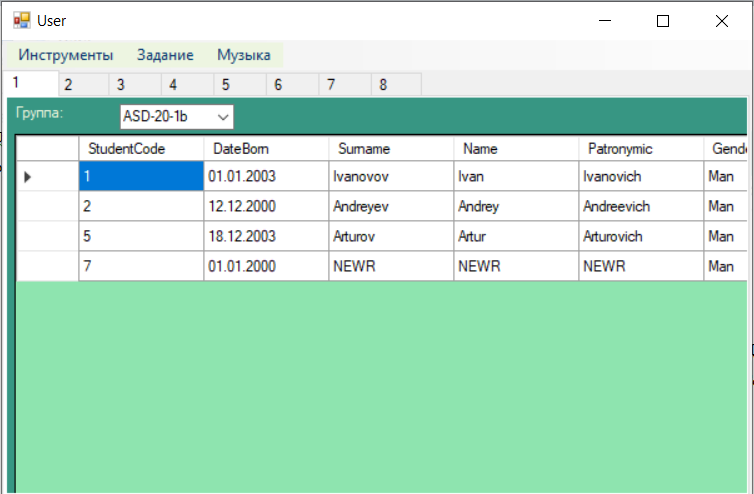


Рисунок 8 – Форма User

При введенных значениях логина и пароля соответственно “admin” и “12345” пользователя переносит на форму Администратора (Рисунок 7), где Администратор имеет права доступа для просмотра, добавления, удаления и обновления данных .

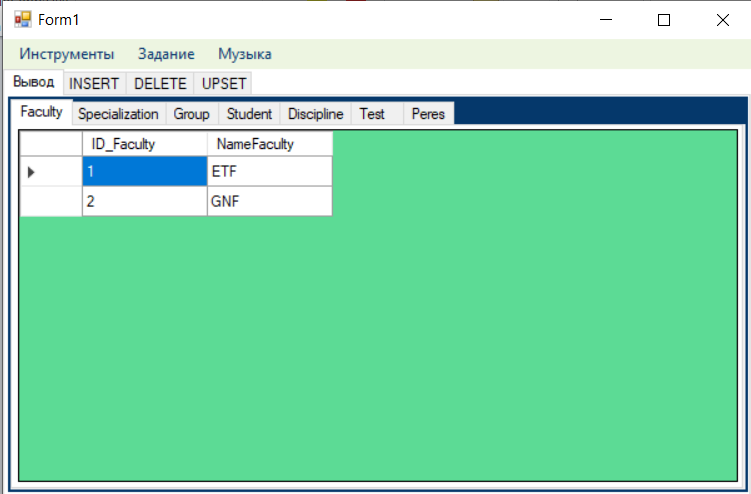


Рисунок 9 – Форма Admin

## **3.4 Тестирование**

Для проверки работы реализованной программы необходимо провести тестирование ключевых функций и нескольких запросов к БД

Тест 1: Вывод информации из таблиц. (Рисунок 10).

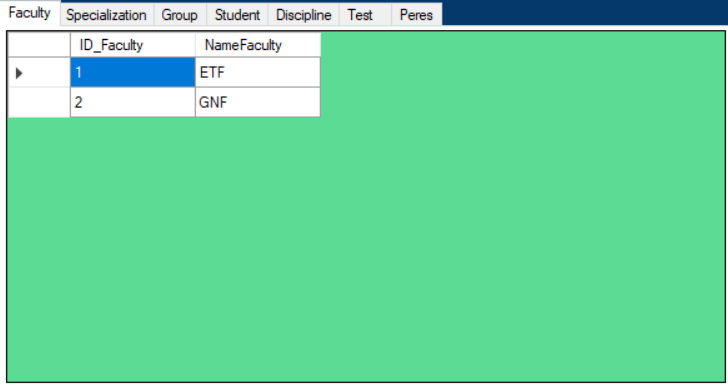


Рисунок 10 - Результаты первого теста

Данные таблицы “Дисциплина” успешно получены.

Тест 2: Добавление элемента в таблицу. (Рисунок 11).

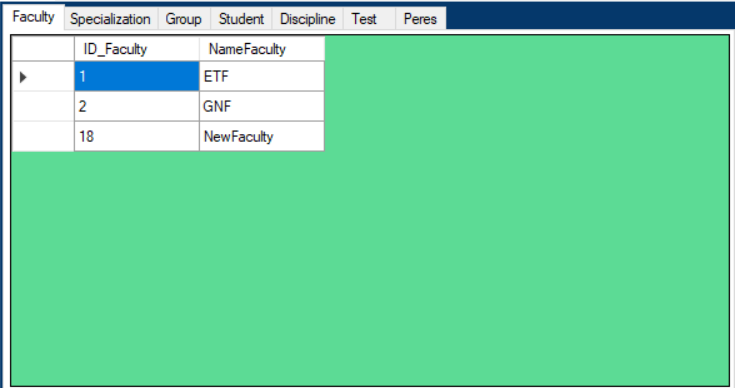


Рисунок 11 - Результаты второго теста

Факультет “NewFaculty” успешно добавлен.

Тест 3: Редактирование элемента таблицы (Рисунок 12).

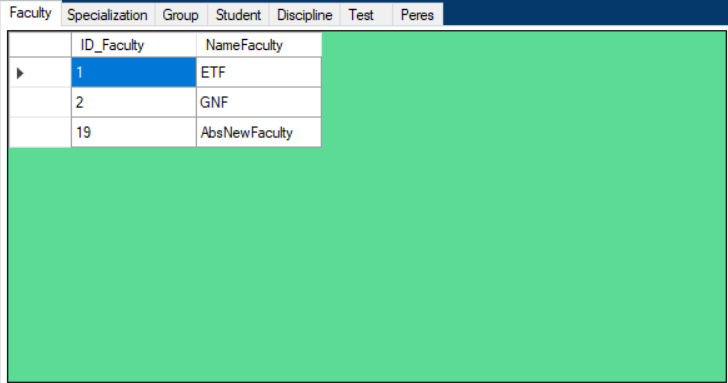


Рисунок 12 - Результаты четвертого теста

Название факультета “NewFaculty” успешно изменено на

“AbsNewFaculty ”.

Тест 4: Удаление элемента таблицы. (Рисунок 13).

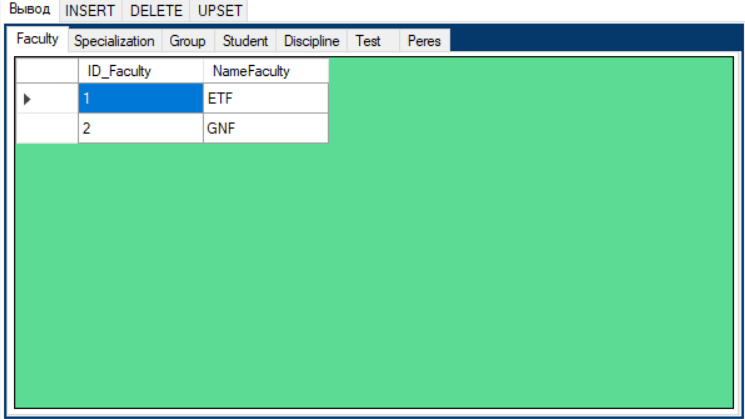


Рисунок 13 – Результаты третьего теста

Элемент “AbsNewFaculty” успешно удален.

Тест 5: Выполнение запроса “Получение студентов указанной группы” (Рисунок 14).

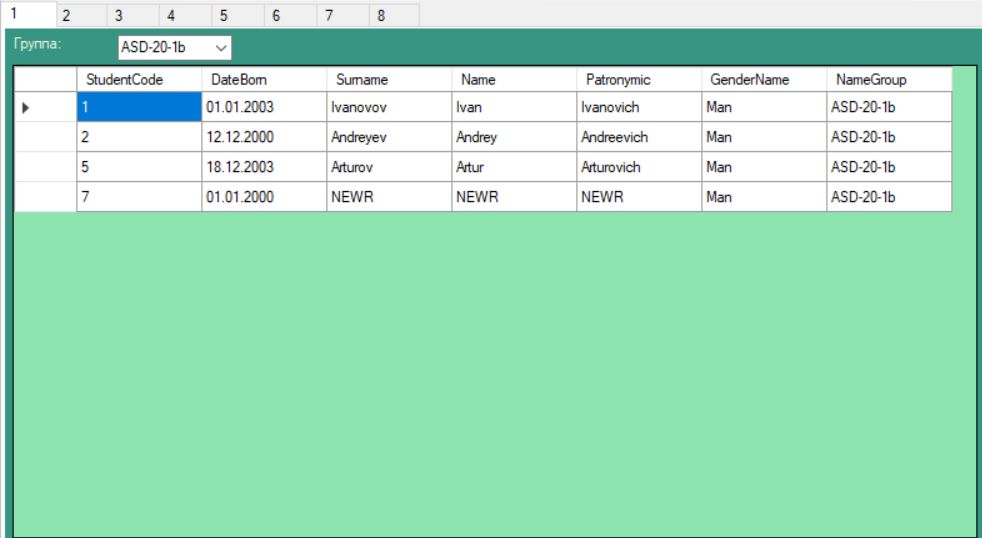


Рисунок 14 - Результаты пятого теста

Список и количество учеников успешно выведены.

После тестирования основных функций программы, ошибки не были выявлены.

**Выводы**

Программа реализована и соответствует требованиям. Программа корректно выполняется все запросы пользователя.

# **Заключение**

В ходе выполнения курсовой работы была создана базы данных для учета успеваемости студентов. Была также разработана программа для удобного пользования элементами СУБД пользователя. Представленная база данных и программа вполне могут быть использованы Высшими Учебными Заведениями для составления учебного плана, так как данная программа позволяет корректно добавлять, удалять, обновлять просматривать данные при работе с базой данных.

Дальнейшей перспективой работы можно считать улучшение интерфейса, добавление новых сущностей и связь между пользователем и базой данных без вмешательства СУБД.

# **Список литературы**

1.   Балалаев М.А. Проектирование баз данных: Методические указания по выполнению курсового проекта по дисциплине «Базы данных» / М.А. Балалаев; ДВГУПС. .     Каф. «Системы автоматизированного проектирования». Хабаровск: Изд - во ДВГУПС, 2007. - 30 с. .     Гурвиц Г.А. Мюгозой Ассезз 2007. Разработка приложений на реальном примере/ Г.А. Гурвиц. - СПб.: БХВ-Петербург, 2007. - 672 с .

2.   Диго С.М. Базы данных: проектирование и использование. Учебник. - М: Финансы и статистика, 2009. .

3  Прайс Джейсон Oracle Database 11g: SQL. Операторы SQL и программы PL/SQL; Лори - , 2012. - 660 c. .

4. Астахова И. Ф., Мельников В. М., Толстобров А. П., Фертиков В. В. СУБД. Язык SQL в примерах и задачах; ФИЗМАТЛИТ - Москва, 2012. - 168 c.  
5 Инструмент для онлайн-моделирования Draw.io [Электронный ресурс] - <https://app.diagrams.net/> - (Дата обращения 01.04.2022)