

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего
образования

«Восточно-Сибирский государственный университет технологий и управления»

Технологический колледж

Курсовой проект

Д.30.1.09.02.03.110.ПЗ

по дисциплине «Технология разработки и защиты баз данных»

Тема: Разработка базы данных «Учёт успеваемости студентов»

Выполнил: студент гр.78/1
Павлова В.А.

Руководитель: _____
к.т.н., доцент
Бильгаева Л.П.

Оценка: _____

Дата защиты: _____

Улан-Удэ

2021

АННОТАЦИЯ

В данном курсовом проекте спроектирована и реализована база данных «Учёт успеваемости студентов». В ходе курсового проекта разработана концептуальная модель предметной области, логическая модель данных, выполнена реализация базы данных в среде СУБД MS SQL Server 2012, разработано и реализовано программное приложение на языке программирования C#, реализованы SQL-запросы к базе данных.

Расчетно-пояснительная записка состоит из введения, трех основных разделов (анализ предметной области, проектирование, реализация), заключения и списка использованных источников.

Курсовой проект содержит 30 страниц, 11 таблиц, 31 рисунка и 9 источников.

СОДЕРЖАНИЕ

1	Анализ предметной области.....	5
1.1	Описание предметной области.....	5
1.2	Разработка концептуальной модели предметной области	6
2	Проектирование.....	9
2.1	Разработка логической модели данных.....	9
2.1.1	Преобразование концептуальной модели в реляционную	9
2.1.2	Нормализация таблиц.....	11
3	Реализация.....	14
3.1	Обоснование и выбор СУБД.....	14
3.2	Разработка физической модели данных	16
3.3	Описание интерфейса.....	18
3.3.1	Описание структуры главного меню	18
3.3.2	Описание форм ввода данных	22
3.3.3	Реализация запросов.....	24
	ЗАКЛЮЧЕНИЕ	29
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	30

					Д.30.1.09.02.03.110.ПЗ								
Изм.			Подпись	Дата									
Разраб.		Павлова В.А.			Разработка базы данных «Учёт успеваемости студентов»			Лит.		Лист		Листов	
Провер.		Бильгаева Л.П.								2		1	
								ВСГУТУ					
Н. Контр.													
Утв.		Михайлова С.С.											

ВВЕДЕНИЕ

В эпоху глобальной компьютеризации информационные технологии развиваются с огромной скоростью. Уже сейчас продукты информационных технологий являются неотъемлемой частью любой структуры.

Базы данных – одни из самых важных и ключевых компонентов большинства систем. Благодаря им, сбор, хранение и обработка данных вышло на новый, более быстрый и упрощённый уровень. Например, в образовательных учреждениях все данные хранятся не только в оригинальных документах, но и в электронных. Причём, это касается не только дипломных или курсовых проектов, но и простых оценок. Это позволяет быстро и чётко обработать большое количество информации, а также вывести информацию по каким-либо определенным критериям.

Целью курсового проекта является разработка базы данных «Учёт успеваемости студентов».

Для достижения поставленной цели необходимо выполнить следующие задачи:

- 1) анализ предметной области;
- 2) разработка концептуальной модели предметной области;
- 3) разработка логической модели данных;
- 4) формулирование запросов к проектируемой базе данных;
- 5) реализация базы данных;
- 6) разработка и реализация программного приложения на языке программирования C#.

Расчетно-пояснительная записка состоит из введения, трех основных разделов (анализ предметной области, проектирование, реализация), заключения и списка использованных источников.

					Д.30.1.09.02.03.110.ПЗ						
Изм.			Подпись	Дата							
Разраб.	Павлова В.А.				Разработка базы данных «Учёт успеваемости студентов»	Лит.		Лист	Листов		
Провер.	Бильгаева Л.П.							3	2		
Н. Контр.											
Утв.	Михайлова С.С.					ВСГУТУ					

В первом разделе приводится описание предметной области и разработанной на ее основе концептуальной модели предметной области.

Во втором разделе разрабатывается логическая модель данных посредством преобразования ER-модели предметной области в реляционную модель и нормализации ее таблиц, а также приводится описание сформулированных запросов к проектируемой базе данных.

В третьем разделе выполнено обоснование и выбор СУБД, описание физической модели данных в среде, выбранной СУБД, а также описание интерфейса программного приложения и реализации SQL-запросов к базе данных.

					Д.30.1.09.02.03.110.ПЗ	Лист.
						4
Изм..	Лист.	№ докум.	Подпись	Дата		

1 Анализ предметной области

1.1 Описание предметной области

Любое образовательное учреждение занимается обучением граждан. Обычно обучение занимает определенный период, который делится по годам и семестрам. В каждом семестре обучаемый изучает перечень дисциплин. Контроль знаний по изучаемым дисциплинам осуществляется в виде зачетов и экзаменов. Учет успеваемости обучаемых один из важных процессов в образовательном учреждении, который позволяет назначать стипендию, выполнять перевод обучаемых с курса на курс, получать срез знаний по тем или иным дисциплинам.

Учёт успеваемости студентов представляет собой совокупность оценок по всем дисциплинам, которые изучали обучающиеся. Такие данные обычно хранятся до окончания обучения в специальных документах, называемой учебной карточкой студента. Обычно результаты за семестр записываются в журнал.

Учёт успеваемости актуален. Если в образовательном учреждении очень много студентов, то отслеживать успехи каждого довольно трудоёмкий процесс. Поэтому целесообразно хранение результатов экзаменационной сессии хранить в базе данных. Это позволит в любой момент времени получить информацию об успеваемости любого студента, составить отчеты по требованию вышестоящих подразделений, определить претендентов на получение стипендии, а также выявить «должников».

Обычно информация об успеваемости представлена в ведомостях на дисциплину, в журнале учета и в карточке студента.

					Д.30.1.09.02.03.110.ПЗ						
Изм.			Подпись	Дата							
Разраб.	Павлова В.А.				Разработка базы данных «Учёт успеваемости студентов»	Лит.		Лист	Листов		
Провер.	Бильгаева Л.П.							5	4		
Н. Контр.											
Утв.	Михайлова С.С.					ВСГУТУ					

1.2 Разработка концептуальной модели предметной области

На основе анализа предметной области были выявлены следующие сущности: СТУДЕНТ, ДИСЦИПЛИНА, ПРЕПОДАВАТЕЛЬ.

Сущность СТУДЕНТ содержит следующую информацию: идентификационный номер студента, ФИО, дата рождения, факультет, группа, номер телефона, домашний адрес. Сущность ДИСЦИПЛИНА – название дисциплины, количество часов, тип контроля, дата сдачи, оценка. Сущность ПРЕПОДАВАТЕЛЬ – ФИО преподавателя, название дисциплины.

На основе выявленных сущностей и результату анализа предметной области определены следующие виды связей.

1. СТУДЕНТ – сдает – ДИСЦИПЛИНА

У одного студента множество предметов, поэтому между сущностями образуется связь типа «один-ко-многим». Класс принадлежности между сущностями СТУДЕНТ и ДИСЦИПЛИНА является обязательными, так как хотя бы один экземпляр сущности СТУДЕНТ связан с экземпляром сущности ДИСЦИПЛИНА и наоборот. ER-диаграмма представлена на рисунке 1.1.



Рисунок 1.1 – ER-диаграмма отношения СТУДЕНТ-ДИСЦИПЛИНА

2. ПРЕПОДАВАТЕЛЬ – принимает – ДИСЦИПЛИНА

Один преподаватель преподаёт одну дисциплину, поэтому между сущностями связь типа «один-к-одному». Класс принадлежности между сущностями ПРЕПОДАВАТЕЛЬ и ДИСЦИПЛИНА является обязательным, так как хотя бы один экземпляр сущности ПРЕПОДАВАТЕЛЬ связан с экземпляром сущности ДИСЦИПЛИНА и наоборот. На рисунке 1.2. представлена ER-диаграмма.

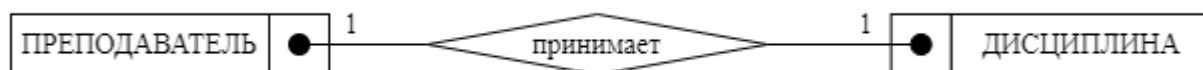


Рисунок 1.2 – ER-диаграмма отношения ПРЕПОДАВАТЕЛЬ-ДИСЦИПЛИНА

На основе анализа связей между сущностями СТУДЕНТ, ДИСЦИПЛИНА, ПРЕПОДАВАТЕЛЬ была построена ER-модель предметной области, показанная на рисунке 1.3.

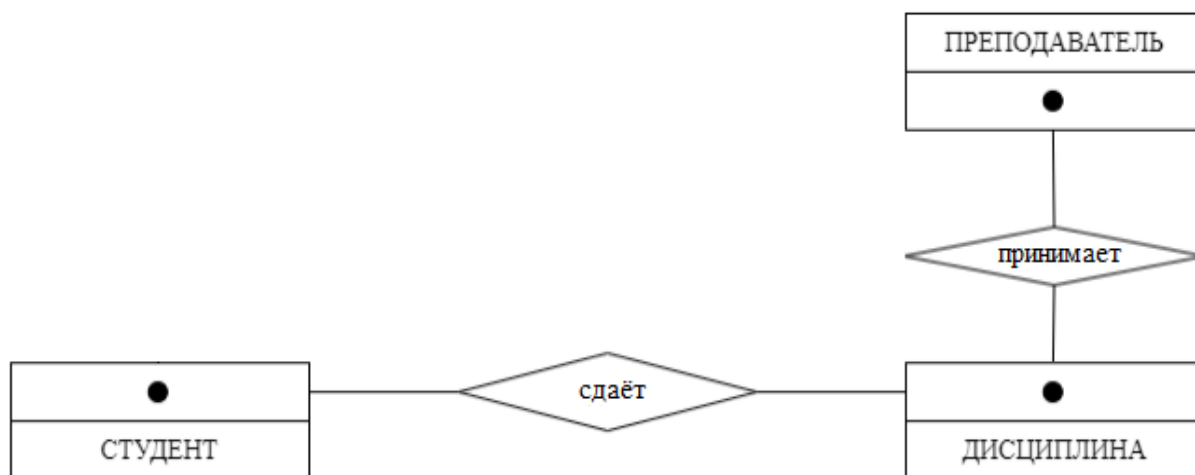


Рисунок 1.3 – ER-модель предметной области

Атрибутный состав сущностей представлен в таблицах 1.1–1.3.

Таблица 1.1 – Атрибутный состав сущности ПРЕПОДАВАТЕЛЬ

ПРЕПОДАВАТЕЛЬ	
Код преподавателя (Код_пр)	
Фамилия Имя Отчество преподавателя (ФИО_пр)	
Код дисциплины (Код_дисц)	

Таблица 1.2 –Атрибутный состав сущности СТУДЕНТ

СТУДЕНТ
Код студента (Код_ст)
Номер зачётной книжки (Н_ЗК)
Фамилия Имя Отчество студента (ФИО_ст)
Группа (Гр)
Факультет (Факульт)
Телефон (Тел)
Адрес (Адрес)

Таблица 1.3 – Атрибутный состав сущности ДИСЦИПЛИНА

ДИСЦИПЛИНА
Код дисциплины (Код_дисц)
Название (Назв_дисц)
Количество часов (Кол_во_ч)
Количество лекций (Кол_во_л)
Количество практик (Кол_во_пр)
Форма контроля (Форма_контроля)
Дата сдачи дисциплины (Дата_сдачи)
Оценка (Оценка)

Концептуальная модель предметной области представляет собой ER-модель предметной области и атрибутный состав сущностей, входящих в нее.

2 Проектирование

2.1 Разработка логической модели данных

2.1.1 Преобразование концептуальной модели в реляционную

Преобразование концептуальной модели предметной области в реляционную модель выполняется в соответствии с правилами, базирующихся на двух основных факторах – тип связи и класс принадлежности.

Преобразуем отношение СТУДЕНТ–ДИСЦИПЛИНА имеющее связь типа «один-ко-многим». Для этого воспользуемся 4 правилом, которое гласит «Если связь типа 1:М и класс принадлежности сущности на стороне М является обязательным, то необходимо построить таблицу для каждой сущности. Первичный ключ сущности должен быть первичным ключом соответствующей таблицы. Первичный ключ сущности на стороне 1 добавляется как атрибут в таблицу для сущности на стороне М».

Первичный ключ сущности на стороне 1 добавляется как атрибут в таблицу для сущности на стороне М. Результат можно увидеть на таблицах 2.1 и 2.2

Таблица 2.1

Код_ст	Н_ЗК	ФИО_ст	Гр	Факульт	Тел	Адрес
--------	------	--------	----	---------	-----	-------

Таблица 2.2

Код_дисц	Назв_дисц	Кол_во_ч	Кол_во_л	Кол_во_пр	Форма_контроля	Дата_сдачи	Оценка	Код_ст
----------	-----------	----------	----------	-----------	----------------	------------	--------	--------

Связь между таблицами показана на рисунке 2.1.

					Д.30.1.09.02.03.110.ПЗ			
Изм.			Подпись	Дата				
Разраб.	Павлова В.А.				Разработка базы данных «Учёт успеваемости студентов»	Лит.	Лист	Листов
Провер.	Бильгаева Л.П.						9	4
						ВСГУТУ		
Н. Контр.								
Утв.	Михайлова С.С.							

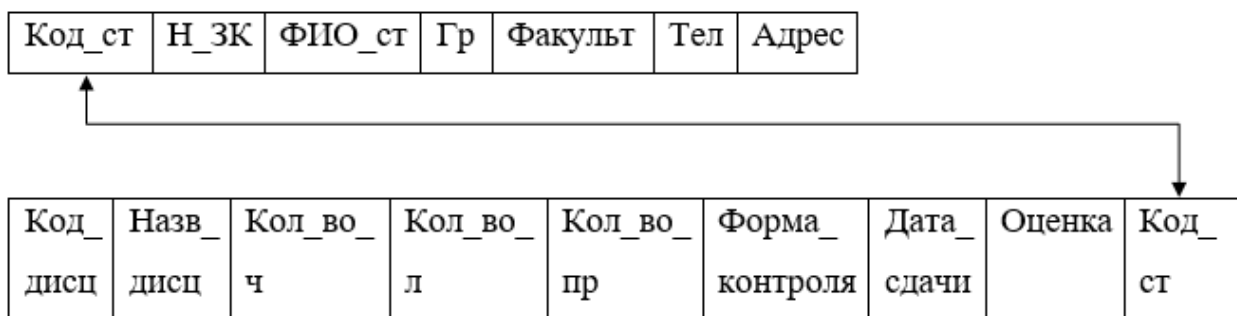


Рисунок 2.1 – Связь между таблицами СТУДЕНТ и ДИСЦИПЛИНА

Тип связи в отношении ПРЕПОДАВАТЕЛЬ–ДИСЦИПЛИНА «один-к-одному» преобразуем согласно 1 правилу, которое гласит «Если связь типа 1:1 «один-к-одному» и класс принадлежности обеих сущностей является обязательным, то необходима только одна таблица. Первичным ключом этой таблицы может быть первичный ключ любой из двух сущностей».

Первичным ключом таблицы ДИСЦИПЛИНА-ПРЕПОДАВАТЕЛЬ выбираем первичный ключ сущности ДИСЦИПЛИНА. Результат преобразования представлен на рисунке 2.2.

Код_дисц	Назв_дисц	Кол_во_ч	Кол_во_л	Кол_во_пр	Форма_контроля	Дата_сдачи	Оценка	Код_ст	Код_пр	ФИО_пр
----------	-----------	----------	----------	-----------	----------------	------------	--------	--------	--------	--------

Рисунок 2.2 – Таблица ДИСЦИПЛИНА-ПРЕПОДАВАТЕЛЬ

Реляционная модель предметной области получена путем интеграции связей выше рассмотренных таблиц и представлена на рисунке 2.3.

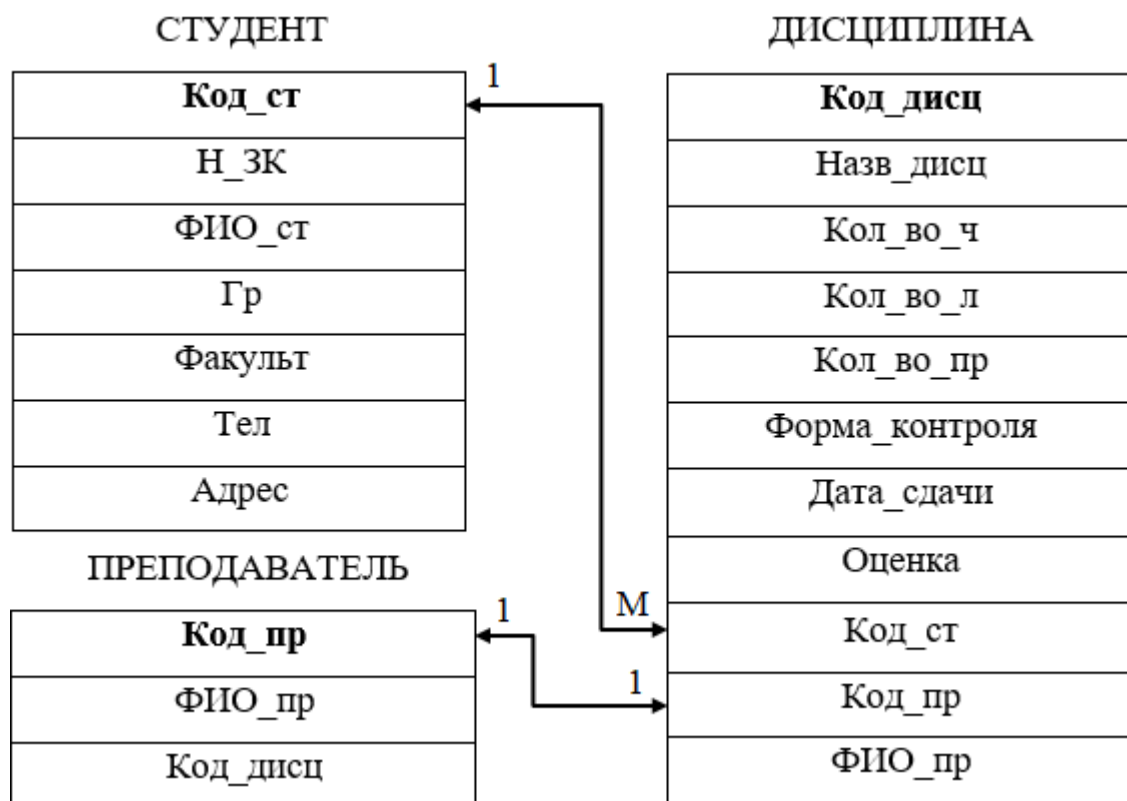


Рисунок 2.3 – Реляционная модель предметной области

2.1.2 Нормализация таблиц

Нормализация представляет собой процесс минимизации избыточных данных в таблицах.

Методика нормализации таблиц была разработана американским ученым А.Ф. Коддом в 1970 году. Суть данной методики заключается в приведении таблиц к определенной нормальной форме (далее НФ). На данный момент существует 6НФ. Однако для создания эффективной реляционной модели базы данных нужно привести все её таблицы, как минимум к 3НФ. Причем, следует отметить, что к 3НФ таблица(ы) приводится лишь при необходимости в этом.

Таблицы СТУДЕНТ и ПРЕПОДАВАТЕЛЬ находятся в нормализованном виде, а таблицу ДИСЦИПЛИНА нужно привести к 3НФ. 3НФ предполагает, что таблица находится во 2НФ и не имеет транзитивных зависимостей между не ключевыми полями.

В таблице ДИСЦИПЛИНА наблюдается транзитивная зависимость: Код_пр → ФИО_пр. Следовательно, в таблице ДИСЦИПЛИНА нарушаются правила 3НФ. Из таблицы ПРЕПОДАВАТЕЛЬ удаляются поля, участвующие в этой транзитивной зависимости – ФИО_пр. Получится таблица, характеризующая таблицу ДИСЦИПЛИНА. Она показана на рисунке 2.4.

Код_ дисц	Назв_ дисц	Кол_во_ ч	Кол_во_ л	Кол_во_ пр	Форма_ контроля	Дата_ сдачи	Оценка	Код_ ст	Код_ пр
--------------	---------------	--------------	--------------	---------------	--------------------	----------------	--------	------------	------------

Рисунок 2.4 – Нормализованная таблица ДИСЦИПЛИНА

Также в таблице ДИСЦИПЛИНА присутствует избыточность данных, чтобы минимизировать это, разобьём данную таблицу на две таблицы. Из таблицы ДИСЦИПЛИНА удаляются поля Кол_во_ч, Кол_во_л, Кол_во_пр, Форма_контроля и они переходят в новую таблицу УЧЕБНЫЙ ПЛАН с полями: Код_спец (Код_специализации), Спец (Специализация), Кол_во_ч, Кол_во_л, Кол_во_пр, Форма_контроля. Результат показан в таблице 2.3 и 2.4 соответственно.

Таблица 2.4 – Минимизированная таблица ДИСЦИПЛИНА

Код_ дисц	Назв_ дисц	Дата_ сдачи	Оценка	Код_ ст	Код_ пр	Код_ спец
--------------	---------------	----------------	--------	------------	------------	--------------

Таблица 2.4 – Таблица УЧЕБНЫЙ ПЛАН

Код_ спец	Спец	Кол_во_ ч	Кол_во_ л	Кол_во_ пр	Форма_ контроля
--------------	------	--------------	--------------	---------------	--------------------

Логическая модель данных предметной области, построенная в результате нормализации реляционной модели, представлена на рисунке 2.7

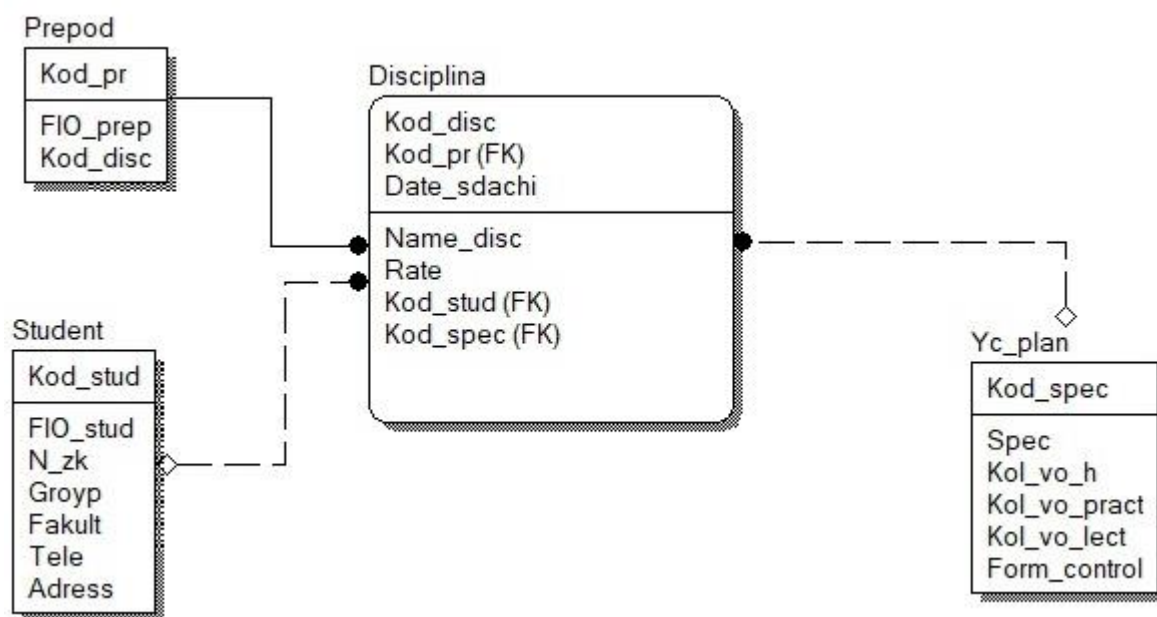


Рисунок 2.5 – Логическая модель данных

2.2 Формулирование запросов к базе данных

Построение логической модели данных позволяет сформулировать запросы к проектируемой базе данных. Запрос – это извлечение данных из базы по определенным критериям.

Запросы:

- 1) отображение студентов, сдающих выбранную дисциплину;
- 2) отображение студентов, претендующих на стипендию;
- 3) отображение студентов, сдающих дисциплину по выбранной форме контроля;
- 4) отображение студентов, сдавших выбранную дисциплину на «отлично»;
- 5) отображение студентов, у которых есть долги по дисциплинам.

3 Реализация

3.1 Обоснование и выбор СУБД

Система управления базами данных (сокр. СУБД) — комплекс программ, позволяющих создать базу данных (БД), манипулировать данными (вставлять, обновлять, удалять и выбирать) и в дальнейшем выполнять различные транзакции (добавление, сохранение, объединение, пересечение, выборка и пр.). Система обеспечивает безопасность, надёжность хранения и целостность данных, а также предоставляет средства для администрирования БД.

СУБД является посредником между логической структурой данных, необходимых разным приложениям, и физическими хранилищами данных. СУБД должны хранить логическую структуру (метаданные), предотвращая несогласованные изменения данных, нарушающие эту структуру.

Основная задача СУБД - предоставить пользователю базы данных возможность работать с ней с помощью легко понимаемого дружественного интерфейса и предоставить в его распоряжение набор операций, выражаемых в терминах языка высокого уровня (например, набор операций, которые можно выполнять с помощью языка SQL).

Таким образом, любая СУБД должна обеспечивать следующее:

- компактное хранение данных (без дублирования);
- оптимизацию доступа к данным;
- логическую целостность (согласованность) данных;
- универсальный интерфейс (язык или протокол), позволяющий задавать структуру данных, изменять и извлекать их неизвестному заранее алгоритму.

					Д.30.1.09.02.03.110.ПЗ		
Изм.			Подпись	Дата			
Разраб.	Павлова В.А.				Разработка базы данных «Меню для детского сада»	Лит.	Лист
Провер.	Бильгаева Л.П.						Листов
							14
Н. Контр.						ВСГУТУ	
Утв.	Михайлова С.С.						

СУБД Microsoft SQL Server 2012, который выбран в качестве среды для реализации базы данных, обладает как преимуществами по сравнению с файловыми системами, так и недостатками.

Преимущества:

- 1) контроль избыточности данных;
- 2) непротиворечивость данных;
- 3) больше полезной информации при том же объеме хранимых данных;
- 4) совместное использование данных;
- 5) поддержка целостности данных;
- 6) повышенная безопасность;
- 7) применение стандартов;
- 8) повышение эффективности с ростом масштабов системы;
- 9) возможность нахождения компромисса при противоречивых требованиях;
- 10) повышение доступности данных и их готовности к работе;
- 11) улучшение показателей производительности;
- 12) упрощение сопровождения системы за счет независимости от данных;
- 13) улучшенное управление параллельностью;
- 14) развитые службы резервного копирования и восстановления.

Недостатки:

- 1) сложность;
- 2) размер программного обеспечения;
- 3) стоимость СУБД;
- 4) дополнительные затраты на аппаратное обеспечение;
- 5) затраты на преобразование приложений;
- 6) производительность;
- 7) более серьезные последствия при выходе системы из строя.

Для сравнения, ниже представлены преимущества СУБД от компании Oracle с соответствующим названием:

- 1) существует несколько версий продукта, благодаря которым можно подобрать более подходящую СУБД в зависимости от цели пользователя;
- 2) СУБД может быть размещена на нескольких серверах, что открывает возможность для хранения очень большого количества записей;
- 3) в СУБД встроена функция, с помощью которой физическое управление данными не будет влиять на логическую структуру;
- 4) постоянные обновления с новейшим функционалом;
- 5) крайне надёжная СУБД;

Недостатки:

- 1) большая стоимость;
- 2) требовательная к ресурсам система;

3.2 Разработка физической модели данных

В ходе проектирования базы данных предметной области «Учёт успеваемости студентов» нами были выделены следующие таблицы: ПРЕПОДАВАТЕЛЬ, СТУДЕНТ, ДИСЦИПЛИНА и УЧЕБНЫЙ ПЛАН. Для каждой таблицы определим тип поля и свойства поля (таблицы 3.1 – 3.4).

Таблица 3.1 – Таблица ПРЕПОДАВАТЕЛЬ

Поле	Тип поля	Свойство поля
Код_пр	Int	Ключевое поле, не Null
ФИО_пр	Nvarchar(20)	He Null
Код_дисц	Int	He Null

Таблица 3.2 – Таблица СТУДЕНТ

Поле	Тип поля	Свойство поля
Код_ст	Int	Ключевое поле, не Null
ФИО_ст	Nvarchar(20)	He Null
Н_ЗК	Int	He Null
Гр	Int	He Null

Факульт	Nvarchar(20)	He Null
Тел	Int	He Null
Адрес	Nvarchar(20)	He Null

Таблица 3.3 – Таблица ДИСЦИПЛИНА

Поле	Тип поля	Свойство поля
Код_дисц	Int	Ключевое поле, не Null
Назв_дисц	Nvarchar(20)	He Null
Дата_сдачи	Date	He Null
Оценка	Int	He Null
Код_ст	Int	He Null
Код_пр	Int	He Null
Код_спец	Int	He Null

Таблица 3.4 – Таблица УЧЕБНЫЙ ПЛАН

Поле	Тип поля	Свойство поля
Код_спец	Int	Ключевое поле, не Null
Спец	Nvarchar(20)	He Null
Кол_во_ч	Int	He Null
Кол_во_л	Int	He Null
Кол_во_пр	Int	He Null
Форма_контроля	Nvarchar(20)	He Null

Физическая модель данных предметной области «Учёт успеваемости студентов» представлена на рисунке 3.1

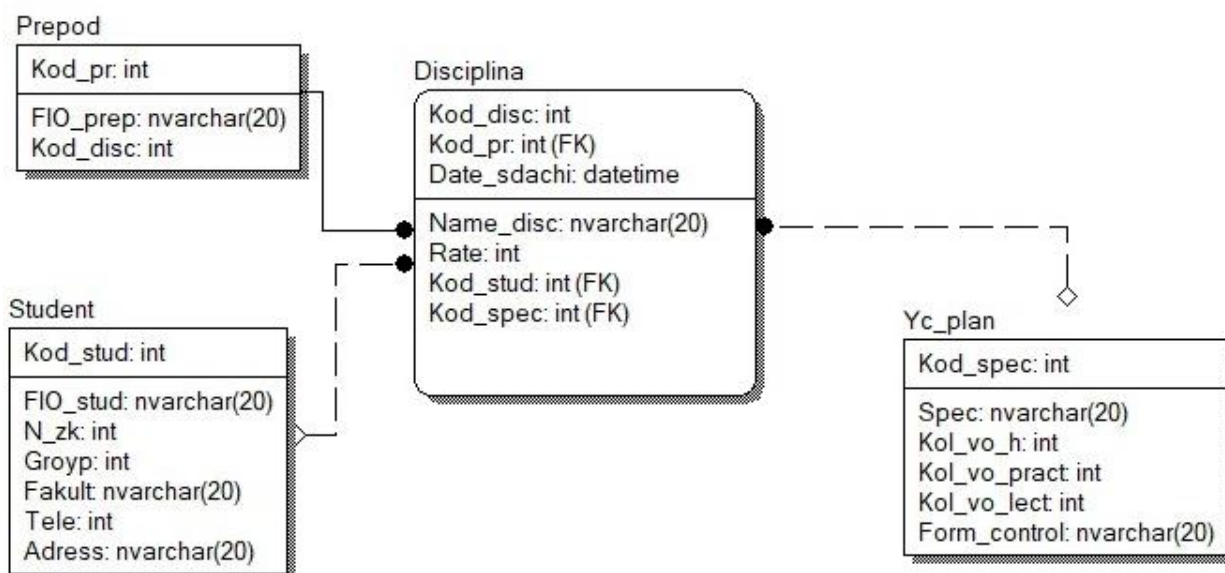


Рисунок 3.1 – Физическая модель данных предметной области

3.3 Описание интерфейса

3.3.1 Описание структуры главного меню

Главное меню представляет систему вкладок, представленной на рисунке 3.2.

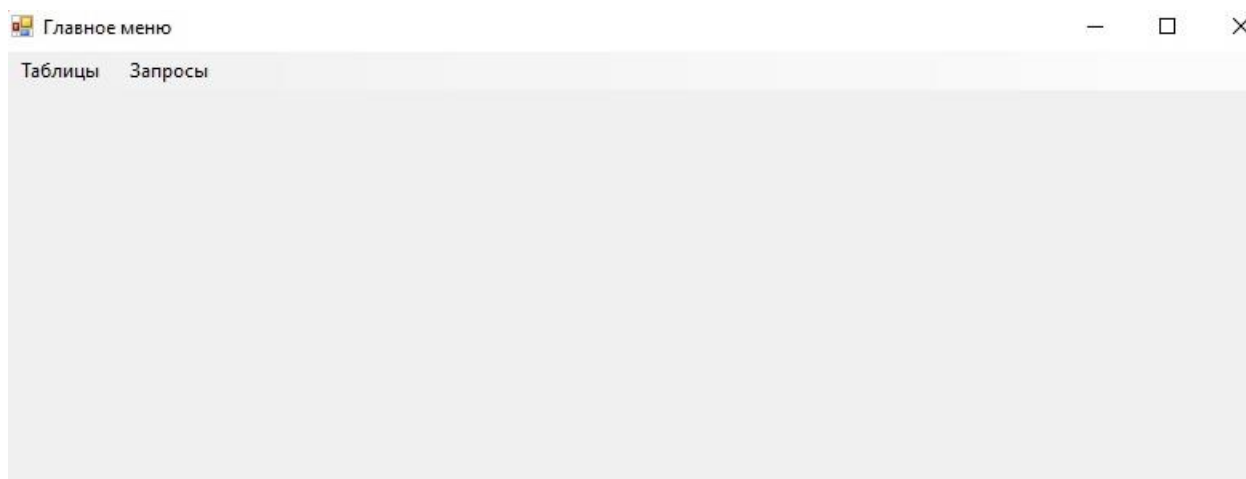


Рисунок 3.2 – Главное меню приложения

Для формирования главного меню были использованы такие элементы управления, как MenuStrip. Было создано 2 основных пункта меню: «Таблицы» и «Запросы». Они разделяются на другие пункты, при нажатии на которые будут открываться новые формы. Для пункта «Таблицы» это:

«Преподаватель», «Студент», «Дисциплина», «Учебный план», как показано на рисунке 3.3.

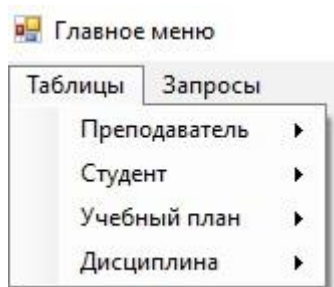


Рисунок 3.3 – Подпункты «Таблицы»

Так же у этих подпунктов есть ещё один подпункт «Добавить». Пример такого подпункта представлен на рисунке 3.4.

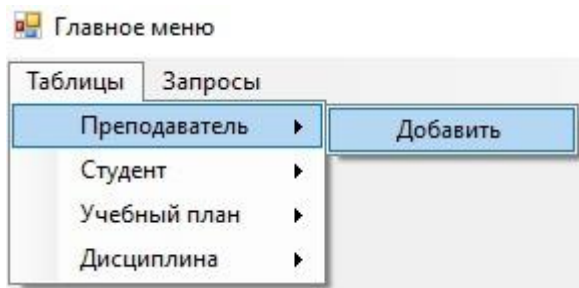


Рисунок 3.4 – Пример подпункта «Добавить»

Для пункта «Запросы» это: «Запрос1», «Запрос2», «Запрос3», «Запрос4», «Запрос5». В пунктах «Таблицы» и «Запросы» размещена компонента dataGridView для отображения таблиц нашей базы данных и запросов. В пунктах Запрос 1, Запрос 3 и Запрос 4 добавлены 1 Label, 1 combobox и 1 Button, для возможности выбора из определённого списка, а в пунктах Запрос 2 и Запрос 5 добавлены по 1 Label и по 1 Button, так как запросы статические. На рисунках 3.5-3.9 показаны вкладки запросов.

Запросы

Выберите название дисциплины, чтобы отобразить всех студентов, которые сдают эту дисциплину

ТРЗБД

Выполнить запрос

Рисунок 3.5 – Вкладка «Запрос 1»

Запросы

Отображение всех учеников с хорошими оценками

Выполнить запрос

Рисунок 3.6 – Вкладка «Запрос 2»

Запросы

Выберите форму контроля, чтобы отобразить всех студентов с этой формой

Курсовой проект

Выполнить запрос

Рисунок 3.7 – «Запрос 3»

Запросы

Выберите название дисциплины, чтобы отобразить студентов, которые получили по этой дисциплине 5

ТРЗБД

Выполнить запрос

Рисунок 3.8 – Вкладка «Запрос 4»



Рисунок 3.9 – Вкладка «Запрос 5»

В пунктах «Таблицы» также расположена компонента ContextMenuStrip добавляющая возможность удаления и редактирования записей (рис.3.10).

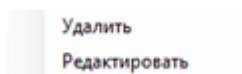


Рисунок 3.10 – Элементы ContextMenuStrip

3.3.2 Описание форм ввода данных

В компоненты Label заносятся названия полей таблиц. В TextBox пользователь пишет новые данные или же, если он выбрал в контекстном меню функцию редактирования, исправляет те данные. При нажатии на кнопку, данные, которые были записаны в TextBox, приводятся к нужным типам данных и создаётся новый объект класса, а затем добавляется в базу данных. Если же это редактирование, то сначала выполняется поиск нужной записи из базы данных по первичному ключу, а затем полям по очереди присваиваются новые значения, также приведённые к нужным типам данных.

Форма ввода и редактирования данных таблицы ПРЕПОДАВАТЕЛЬ содержит 3 компоненты Label и 3 TextBox и 1 Button.

Рисунок 3.11 – Форма добавления и редактирования данных в таблицу
ПРЕПОДАВАТЕЛЬ

Рисунок 3.12 – Форма добавления и редактирования данных в таблицу
СТУДЕНТ

Форма ввода и редактирования данных таблицы СТУДЕНТ содержит 7 Label и 7 TextBox и 1 Button.

Рисунок 3.13 – Форма добавления и редактирования данных в таблицу
ДИСЦИПЛИНА

Форма ввода и редактирования данных таблицы ДИСЦИПЛИНА содержит 7 Label, 7 TextBox и 1 Button.

Рисунок 3.13 – Форма добавления и редактирования данных в таблицу
УЧЕБНЫЙ ПЛАН

Форма ввода и редактирования данных таблицы УЧЕБНЫЙ ПЛАН содержит 6 Label и 6 TextBox и 1 Button.

3.3.3 Реализация запросов

Запрос 1: Вывод всех студентов, сдающих выбранную дисциплину.

					Д.30.1.09.02.03.110.ПЗ	Лист.
Изм..	Лист.	№ докум.	Подпись	Дата		24

На рисунке 3.14 показан код запроса, а на рисунке 3.15 результат его выполнения.

```

List<Disciplina> disc = db.Disciplina.ToList();
List<Student> stud = db.Student.ToList();
dataGridView1.DataSource = disc.Join(stud, x => x.Kod_stud, y => y.Kod_stud,
    (x, y) => new { x.Kod_disc, x.Name_disc, y.FIO_stud }).Where(x => x.Name_disc == comboBox1.Text).ToList();
List<string> str = new List<string>() { "Код дисциплины", "Название дисциплины", "ФИО студента" };
List<string> str1 = new List<string>() { "Kod_disc", "Name_disc", "FIO_stud" };
columns(str, str1);

```

Рисунок 3.14 – Код 1 запроса к базе данных

Отображение студентов, сдающих выбранную дисциплину

	Код дисциплины	Название дисциплины	ФИО студента	Дата сдачи
▶	1	ТРЗБД	Воркунов А.И.	21.12.2021
	1	ТРЗБД	Ереванова А.Д.	21.12.2021
	1	ТРЗБД	Жуков Д.А.	21.12.2021
	1	ТРЗБД	Степанова К.С.	21.12.2021
	1	ТРЗБД	Тушева Е.М.	21.12.2021
	1	ТРЗБД	Акопян К.С.	28.12.2021
	1	ТРЗБД	Дамбаева В.В.	28.12.2021
	1	ТРЗБД	Эрдынеева О.Б.	28.12.2021
	1	ТРЗБД	Борисов П.П.	28.12.2021

ТРЗБД ▼

Выполнить запрос

Рисунок 3.15 – Результат выполнения 1 запроса

Запрос 2: Вывод всех учеников с положительными оценками.

На рисунке 3.16 показан код запроса, а на рисунке 3.17 результат его выполнения.

```

List<Disciplina> disciplinas = db.Disciplina.ToList();
List<Student> stud = db.Student.ToList();
dataGridView1.DataSource = disciplinas.Join(stud, x => x.Kod_stud, y => y.Kod_stud,
    (x, y) => new { x.Name_disc, x.Prepod.FIO_prep, y.FIO_stud, y.N_zk, x.Rate, x.Yc_plan.Form_control })
    .Where(x => x.Rate == 5 || x.Rate == 4).ToList();
List<string> str = new List<string>() { "Название дисциплины", "ФИО преподавателя", "ФИО студента", "Номер зачётной книжки", "Оценка", "Форма контроля" };
List<string> str1 = new List<string>() { "Name_disc", "FIO_prep", "FIO_stud", "N_zk", "Rate", "Form_control" };
columns(str, str1);

```

Рисунок 3.16 – Код 2 запроса к базе данных

Запросы

Отображение студентов, претендующих на стипендию

	Название дисциплины	ФИО преподавателя	ФИО студента	Номер зачётной книжки	Оценка	Форма контроля
▶	ТРЗБД	Бильгаева Л.П.	Ереванова А.Д.	1010102	4	Зачёт
	ТРЗБД	Бильгаева Л.П.	Акопян К.С.	1878101	5	Курсовой проект
	ТРЗБД	Бильгаева Л.П.	Дамбаева В.В.	1878105	4	Курсовой проект
	ТРЗБД	Бильгаева Л.П.	Эрдынеева О.Б.	1878119	4	Курсовой проект
	ТРЗБД	Бильгаева Л.П.	Борисов П.П.	1878102	4	Курсовой проект
	ТРПО	Будаев Е.С.	Акопян К.С.	1878101	5	Курсовой проект
	ТРПО	Будаев Е.С.	Борисов П.П.	1878102	4	Курсовой проект
	ТРПО	Будаев Е.С.	Дамбаева В.В.	1878105	4	Курсовой проект
	ТРПО	Будаев Е.С.	Павлова В.А.	1878110	5	Курсовой проект
	ТРПО	Росников А.В.	Воркунов А.И.	1010101	5	Зачёт
	ТРПО	Росников А.В.	Ереванова А.Д.	1010102	4	Зачёт

Выполнить запрос

Рисунок 3.17 – Результат выполнения 2 запроса

Запрос 3: Вывод данных о всех студентах, которые сдают дисциплины по выбранной форме контроля.

На рисунке 3.18 показан код запроса, а на рисунке 3.19 результат его выполнения.

```

List<Disciplina> disciplinas = db.Disciplina.ToList();
List<Student> stud = db.Student.ToList();
dataGridView1.DataSource = disciplinas.Join(stud, x => x.Kod_stud, y => y.Kod_stud,
(x, y) => new { x.Name_disc, x.Prepod.FIO_prep, y.FIO_stud, x.Yc_plan.Form_control })
.Where(x => x.Form_control == comboBox1.Text).ToList();
List<string> str = new List<string>() { "Название дисциплины", "ФИО преподавателя", "ФИО студента", "Форма контроля" };
List<string> str1 = new List<string>() { "Name_disc", "FIO_prep", "FIO_stud", "Form_control" };
columns(str, str1);

```

Рисунок 3.18 – Код 3 запроса к базе данных

Запросы

Отображение студентов, сдающих дисциплину по выбранной форме контроля

	Название дисциплины	ФИО преподавателя	ФИО студента	Форма контроля
▶	ТРЗБД	Бильгаева Л.П.	Акопян К.С.	Курсовой проект
	ТРЗБД	Бильгаева Л.П.	Дамбаева В.В.	Курсовой проект
	ТРЗБД	Бильгаева Л.П.	Эрдынеева О.Б.	Курсовой проект
	ТРЗБД	Бильгаева Л.П.	Борисов П.П.	Курсовой проект
	ТРПО	Будаев Е.С.	Акопян К.С.	Курсовой проект
	ТРПО	Будаев Е.С.	Борисов П.П.	Курсовой проект
	ТРПО	Будаев Е.С.	Дамбаева В.В.	Курсовой проект
	ТРПО	Будаев Е.С.	Павлова В.А.	Курсовой проект
	ЭВМ	Жилкина М.А.	Акопян К.С.	Курсовой проект
	ЭВМ	Жилкина М.А.	Борисов П.П.	Курсовой проект
	ЭВМ	Жилкина М.А.	Дамбаева В.В.	Курсовой проект
	ЭВМ	Жилкина М.А.	Эрдынеева О.Б.	Курсовой проект

Курсовой проект ▼

Выполнить запрос

Рисунок 3.19 – Результат выполнения 3 запроса

Запрос 4: Вывод всех студентов, которые сдали выбранную дисциплину на «5».

На рисунке 3.20 показан код запроса, а на рисунке 3.21 результат его выполнения.

```
List<Disciplina> disciplinas = db.Disciplina.ToList();
List<Student> stud = db.Student.ToList();
dataGridView1.DataSource = disciplinas.Join(stud, x => x.Kod_stud, y => y.Kod_stud,
(x, y) => new { x.Name_disc, x.Prepod.FIO_prep, y.FIO_stud, y.N_zk, x.Rate, x.Yc_plan.Form_control })
.Where(x => x.Name_disc == comboBox1.Text && x.Rate == 5).ToList();
List<string> str = new List<string>() { "Название дисциплины", "ФИО преподавателя",
"ФИО студента", "Номер зачётной книжки", "Оценка", "Форма контроля" };
List<string> str1 = new List<string>() { "Name_disc", "FIO_prep", "FIO_stud",
"N_zk", "Rate", "Form_control" };
columns(str, str1);
```

Рисунок 3.20 – Код 4 запроса к базе данных

Отображение студентов, сдавших выбранную дисциплину на отлично

	Название дисциплины	ФИО преподавателя	ФИО студента	Номер зачётной книжки	Оценка	Форма контроля
▶	ТРПО	Будаев Е.С.	Акопян К.С.	1878101	5	Курсовой проект
	ТРПО	Будаев Е.С.	Павлова В.А.	1878110	5	Курсовой проект
	ТРПО	Росников А.В.	Воркунов А.И.	1010101	5	Зачёт

ТРПО ▼

Выполнить запрос

Рисунок 3.21 – Результат выполнения 4 запроса

Запрос 5: отобразить студентов, у которых имеются долги по дисциплинам.

На рисунке 3.22 показан код запроса, а на рисунке 3.23 результат его выполнения.

```
List<Disciplina> disciplinas = db.Disciplina.ToList();
List<Student> stud = db.Student.ToList();
dataGridView1.DataSource = disciplinas.Join(stud, x => x.Kod_stud, y => y.Kod_stud,
(x, y) => new { x.Name_disc, x.Prepod.FIO_prep, y.FIO_stud, y.N_zk, x.Rate, x.Yc_plan.Form_control })
.Where(x => x.Rate == 2).ToList();
List<string> str = new List<string>() { "Название дисциплины", "ФИО преподавателя", "ФИО студента",
"Номер зачётной книжки", "Оценка", "Форма контроля" };
List<string> str1 = new List<string>() { "Name_disc", "FIO_prep", "FIO_stud", "N_zk",
"Rate", "Form_control" };
columns(str, str1);
```

Рисунок 3.22 – Код 5 запроса к базе данных

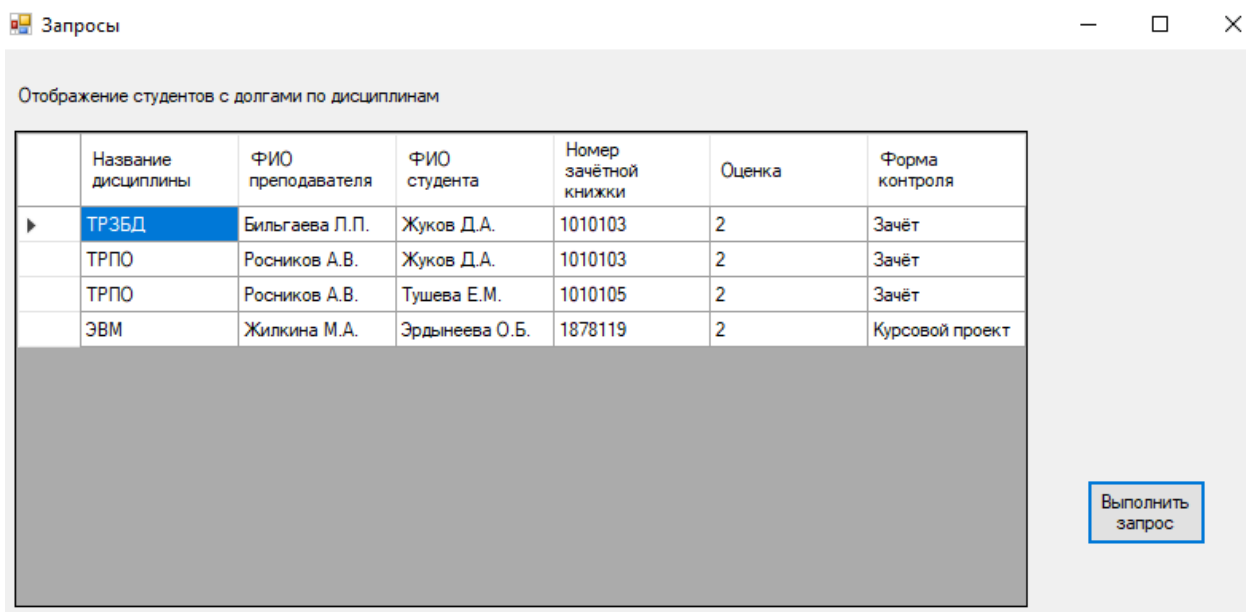


Рисунок 3.23 – Результат выполнения 5 запроса

ЗАКЛЮЧЕНИЕ

В данном курсовом проекте была спроектирована и реализована база данных «Учёт успеваемости студентов» и разработано программное приложение.

В ходе выполнения данного курсового проекта был проведён анализ предметной области, была разработана концептуальная модель предметной области, на основе которой разработана логическая модель данных путем преобразования концептуальной модели в реляционную. Далее была произведена нормализация таблиц и формулирование запросов к проектируемой базе данных. Выполнена реализация базы данных в среде СУБД Microsoft SQL Server-2019. Для работы с базой данных разработано и реализовано программное приложение, которое имеет удобный и дружественный интерфейс, позволяющий пользователю эффективно работать с программой. Также программа позволяет выполнять необходимые запросы к базам данных.

Таким образом, все поставленные задачи выполнены, цель достигнута.

					Д.30.1.09.02.03.110.ПЗ						
Изм.			Подпись	Дата							
Разраб.		Павлова В.А.			Разработка базы данных «Учёт успеваемости студентов»			Лит.	Лист	Листов	
Провер.		Бильгаева Л.П.								29	2
Н. Контр.								ВСГУТУ			
Утв.		Михайлова С.С.									

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Шилдт, Герберт Полный справочник по С# / Герберт Шилдт: Пер.с англ. – М.: Издательский дом «Вильямс», 2004. – 752с.: ил. – Парал, тит. Англ.
2. Павловская, Т. А. С#.Программирование на языке высокого уровня. Учебник для вузов / Т.А. Павловская. – СПб: Питер, 2009. – 432с.: ил.
3. Осипов, Н. А. Разработка приложений ASP.NET с применением Entity Framework / Н.А. Осипов. – СПб: Университет ИТМО, 2016. – 80с.
4. Осетрова, И. С. Администрирование MS SQL Server 2014 / И.С. Осетрова. – СПб: Университет ИТМО, 2016. – 90с.
5. Кара-Ушанов, В. Ю. SQL – язык реляционных баз данных: учебное пособие / В.Ю. Кара-Ушанов. – Екатеринбург: Изд-во Урал. ун-та, 2016. – 156с.
6. Горбаченко, В. И. Проектирование информационных систем с СА Erwin Modeling Suite 7.3: учебное пособие / В.И. Горбаченко, Г.Ф. Убиенных, Г.В. Бобрышева – Пенза: Изд-во ПГУ, 2012. – 154с.
7. Халимон, В. И. Базы данных: учебное пособие / В.И. Халимон, Г.А. Мамаева, А.Ю. Рогов, В.Н. Чепикова – СПб: СПбГТИ(ТУ), 2017. – 118с.
8. Стружкин, Н. П. Базы данных: проектирование: учебник для СПО / Н.П. Стружкин, В.В. Годин. – М.: Изд-во Юрайт, 2019. – 477с. – (Серия: Профессиональное образование).
9. Попова-Коварцева, Д. А. Основы проектирования баз данных: учебное пособие / Д.А. Попова-Коварцева, Е.В. Сопченко, – Самара: Изд-во Самарского университета, 2019. – 112 с.: ил.

					Д.30.1.09.02.03.110.ПЗ			
Изм.			Подпись	Дата				
Разраб.	Павлова В.А.				Разработка базы данных «Учёт успеваемости студентов»	Лит.	Лист	Листов
Провер.	Бильгаева Л.П.						30	2
						ВСГУТУ		
Н. Контр.								
Утв.	Михайлова С.С.							