

Содержание

Введение.....	5
1. Анализ технического задания.....	6
1.1. Описание предметной области.....	9
1.2. Аналогии.....	10
2. Модели данных.....	12
2.1. Концептуальная модель.....	12
2.2. Логическая модель.....	14
2.3. Приведение логической модели к физической.....	16
3. Разработка и реализация АИС.....	18
3.1. Создание SQL-запросов.....	18
3.2. Руководство пользователя.....	21
3.3. Руководство программиста.....	30
4. Тестирование АИС.....	35
Заключение.....	43
Список литературы.....	44
Приложение А: Модели данных.....	45
Приложение Б: Текст кода.....	46

					МИВУ 10.03.01-03ПЗ			
Изм.	Лист	№ докум	Подпись	Дата				
Разраб		Быков А.М.			Разработка АИС автошколы	Литера	Лист	Листов
Пров		Колпаков А. А.				у	4	66
Н. Контр.								
Утв		Орлов А. А.				МИВЛГУ ИБ-122		

Введение.

В современном мире автошколы сталкиваются с необходимостью эффективного управления большим объемом информации, связанной с автомобилями, инструкторами и учениками. Широкий выбор учебных автомобилей, необходимость учета данных инструкторов и студентов усложняет процесс управления записями, расписаниями и финансами.

В каждой организации, независимо от ее размера, возникает необходимость в системе управления данными, которая обеспечит оптимизацию процессов и упростит работу сотрудников. Некоторые автошколы продолжают использовать традиционные методы, такие как хранение бумажных документов в архивах и ручной поиск информации. Однако большинство организаций предпочитают внедрять компьютеризированные базы данных и программные решения, которые позволяют эффективно хранить и быстро извлекать необходимую информацию.

Учитывая большое количество данных, высокие требования к точности и необходимость постоянного обновления информации, применение баз данных для управления учебным процессом в автошколе становится жизненно важным.

Целью данной курсовой работы является создание информационной системы (далее ИС) для работы с базой данных (далее БД) автошколы, которая обеспечит удобное взаимодействие пользователей и персонала.

В рамках работы с данной базой данных каждый студент сможет получить информацию об интересующих его учебных автомобилях и инструкторах.

Для достижения этой цели я поставил перед собой ряд задач:

1. Сравнить производительность систем управления базами данных (далее СУБД) и средств программирования.
2. Разработать модели данных.
3. Разработать базу данных
4. Создать многопользовательское приложение.

1. Анализ технического задания

Современная разработка информационных систем требует внимательного подбора инструментов, которые наилучшим образом отвечают требованиям конкретного проекта. В рамках создания системы управления базой данных для автошколы я проанализирую и сопоставлю несколько технологий, чтобы определить наиболее подходящие из них.

В этом анализе будут рассмотрены языки программирования C# и C++, среды разработки Visual Studio и Eclipse, а также системы управления базами данных SQLite и MySQL. Сравнение этих технологий поможет выявить наиболее эффективные инструменты с учетом их производительности, удобства использования, гибкости, безопасности и возможности интеграции с другими системами. Это обеспечит надежность и эффективность базы данных, поддерживая высокие стандарты разработки и эксплуатации.

1. C++:

- **Производительность:** отличается высокой производительностью, так как это язык низкого уровня, позволяющий работать непосредственно с памятью и системными ресурсами.[1]
- **Гибкость:** предоставляет детальный контроль над аппаратным обеспечением.[1]
- **Область применения:** часто используется для создания системного программного обеспечения и высокопроизводительных приложений. [1]

2. C#:

- **Удобство:** является языком более высокого уровня по сравнению с C++, что делает его более доступным для изучения и использования. [3]
- **Интеграция:** тесно связан с платформой .NET, что упрощает разработку веб-приложений и бизнес-логики, что особенно актуально для онлайн-сервисов автошколы. [3]

					МИВУ 10.03.01-03ПЗ	Лист
						6
Изм.	Лист	№ докум	Подпись	Дата		

- **Безопасность:** обеспечивает автоматическое управление памятью (сборка мусора), что уменьшает риск возникновения ошибок, связанных с управлением памятью. [3]

3. Visual Studio:

- **Функциональные возможности:** предлагает широкий спектр инструментов для отладки, тестирования и разработки, что позволяет ускорить процесс создания и внедрения системы управления базой данных для автошколы.

- **Использование:** очень удобная и интуитивно понятная среда. [6]

- **Интеграция:** отлично интегрируется с экосистемой Microsoft, поддерживает множество языков программирования. [6]

4. Характеристики Eclipse:

- **Открытый исходный код:** является бесплатной платформой, доступной для использования и активно поддерживаемой сообществом разработчиков. [2]

- **Кроссплатформенность:** работает на различных операционных системах, включая Windows, macOS и Linux. [2]

- **Плагины:** обладает большим количеством доступных плагинов, что позволяет значительно расширить функциональные возможности среды. [2]

5. Характеристики SQLite:

- **Легковесность:** не требует установки или настройки отдельного сервера, что упрощает процесс использования. [5]

- **Простота:** имеет интуитивно понятный интерфейс и компактную архитектуру, что делает ее удобной в работе. [5]

- **Встроенная СУБД:** идеально подходит для приложений, которым необходимо локальное хранилище данных. [5]

6. Характеристики MySQL:

- **Производительность:** обеспечивает высокую производительность и возможность масштабирования для крупных проектов. [4]

- **Функциональность:** поддерживает сложные SQL-запросы, транзакции и интеграцию с другими системами управления базами данных. [4]

- Популярность: широко используется в веб-разработке и имеет большое сообщество пользователей, что способствует её поддержке и развитию.
- [4]

Результаты анализа характеристик могут быть представлены в виде таблиц:

1. Таблица сравнения языков программирования.
2. Таблица сравнения сред разработки.
3. Таблица сравнения систем управления базами данных.

Таблица 1 – Сравнение языков программирования

Критерий	C++	C#
Производительность	Высокая	Средняя
Простота	Сложная	Простая
Гибкость	Высокая	Средняя
Интеграция	Ограниченная	Отличная с .NET
Безопасность	Низкая (ручное управление памятью)	Высокая (автоматическая сборка мусора)

Таблица 2 – Сравнение сред программирования

Критерий	Visual Studio	Eclipse
Производительность	Высокая	Высокая
Простота	Удобная	Средняя
Гибкость	Высокая	Высокая
Интеграция	Отличная с экосистемой MS	Кроссплатформенная
Безопасность	Высокая	Высокая

Таблица 3 – Сравнение СУБД

Критерий	SQLite	MySQL
Производительность	Высокая	Высокая

Простота	Простая	Средняя
Гибкость	Низкая	Высокая
Интеграция	Локальное хранилище данных	Веб-разработка
Безопасность	Высокая	Высокая

В данном случае выбор C#, Visual Studio и SQLite является наилучшим решением. Эти технологии предлагают ряд преимуществ:

- C#: Обладает простотой в использовании, высокой производительностью, отличной совместимостью с .NET и высоким уровнем безопасности.
- Visual Studio: Это мощная и удобная интегрированная среда разработки, поддерживающая множество языков программирования и предлагающая широкий набор инструментов.
- SQLite: Эта легковесная и простая в использовании система управления базами данных идеально подходит для встроенных решений и приложений с локальным хранением информации.

Использование этих инструментов обеспечит надежную, безопасную и эффективную работу системы управления базами данных для автошколы.

1.1. Описание предметной области

В данной курсовой работе рассматривается предметная область «Автошкола». Основной задачей является проектирование и разработка приложения для автоматизации процесса оформления договора обучения.

Процесс оформления обучения в автошколе осуществляется следующим образом:

1) Поиск учеников. Встреча с потенциальными учениками для выяснения их желаемого обучения. Эту задачу выполняют инструктора.

2) Запись на курс. При записи на курс администратор регистрирует клиента в базе данных. Для записи клиенту необходимо предоставить номер телефона, а также свои паспортные данные.

3) Заключение договора. Договор обучения оформляется администратором.

В автошколе есть несколько категорий сотрудников: инструкторы и администраторы.

Основная задача курсовой работы - разработка приложения для автоматизации работы автошколы.

В Автошколе можно выделить следующие сущности:

Машины – транспортные средства:

Марка, государственный регистрационный номер.

Инструкторы:

ФИО, серия и номер паспорта, телефон.

Студенты:

ФИО, Дата рождения, серия и номер паспорта, телефона

Экзамены:

Данные студента, данные инструктора, дата, машина, отделение ГИБДД, результат.

Договоры:

Данные студента, данные инструктора, дата, машина, сумма, номер и серия паспорта студента.

База данных будет содержать информацию о студентах, машинах, инструкторах, экзаменах и договорах. Это позволит автоматизировать все основные процессы работы автошколы.

1.2. Аналогии

В настоящее время существует множество автоматизированных систем (АС) для автошкол. Одним из примеров такой системы может быть платформа, подобная avtosmartsoft.ru.

Достоинства:

- Удобный интерфейс для поиска и сравнения предложений различных автошкол.

- Возможность онлайн-записи и оплаты выбранного курса.
- Сбор и анализ данных для улучшения предложений и услуг для клиентов.

Недостатки:

- Комиссии за использование платформы, которые платят автошколы.
- Необходимость поддержания актуальной информации о курсах и ценах.
- Конкуренция с другими агрегаторами за привлечение учеников.

Другим примером может служить специализированное ПО для автошкол, такое как avtoshkolasoft.ru, предназначенное для автоматизации учета, планирования и анализа работы учебных заведений.

Достоинства:

- Полная автоматизация ключевых бизнес-процессов автошколы.
- Единая база данных учеников, инструкторов, автомобилей и финансов.
- Гибкие инструменты управления расписанием, обучением.
- Развиты аналитические возможности, отчетность и панели мониторинга.

Недостатки:

- Значительные первоначальные затраты на внедрение и обучение персонала.

- Сложность интеграции с существующими IT-системами автошколы.
- Необходимость адаптации бизнес-процессов под возможности системы.
- Зависимость от стабильной работы системы и риски простоев.

Особенности:

- Управление расписанием, записями на занятия и взаимодействием с учениками.

- Управление обучением, экзаменами и сертификацией.
- Управление финансами, бухгалтерией и кадрами.
- Веб-интерфейс для учеников и мобильные приложения для инструкторов.

2. Модели данных

Данный этап является самым важным при создании АИС. Здесь выделяются сущности, атрибуты сущностей и связи между сущностями. На основе полученной диаграммы “Сущность – связь” или логической модели строятся функциональные модели системы и диаграмма потоков данных. Для создания базы данных, нужно логическую модель представить в виде физической.

Разработка моделей данных включает 3 этапа: [7]

1) Концептуальная модель:

- определение сущностей (объектов, понятий) предметной области и их взаимосвязей;
- выявление атрибутов сущностей и описание их характеристик;
- построение диаграммы сущность-связь (ER-диаграммы) для визуального представления модели.

2) Логическая модель:

- трансформация концептуальной модели в структуру, подходящую для конкретной СУБД;
- определение типов данных, ключей, индексов, ограничений целостности;
- нормализация данных для устранения избыточности и аномалий;
- спецификация отношений между сущностями (таблицами).

3) Физическая модель:

- оптимизация логической модели для конкретной СУБД и аппаратной платформы;
- определение физического хранения данных: файлов, томов, буферов и т.д.;
- результат - физическая схема базы данных, готовая для реализации.

2.1. Концептуальная модель

Концептуальная модель данных является первым и наиболее абстрактным уровнем проектирования базы данных. На этом этапе определяются ключевые сущности предметной области, их атрибуты и взаимосвязи. [7]

Были выделены следующие базовые сущности организации и их атрибуты:

1) Студенты.

Атрибуты студента: ID, Фамилия, Имя, Отчество, Дата Рождения, Номер паспорта, Серия паспорта, Номер телефона.

2) Инструкторы.

Атрибуты инструктора: ID, Фамилия, Имя, Отчество, Номер паспорта, Серия паспорта, Номер телефона.

3) Машины.

Атрибуты машины: ID, Марка, Номер

4) Договоры.

Атрибуты договора: ID, ID_Студента, ID_Инструктора, ID_Машины, Дата, Сумма, Номер паспорта студента, Серия паспорта студента

5) Экзамены

Атрибуты экзамена: ID, ID_Студента, ID_Инструктора, ID_Машины, Дата, Отделение ГИБДД, Результат

Все выделенные сущности были объединены в общую концептуальную модель данных, которая представлена на рисунке 1.

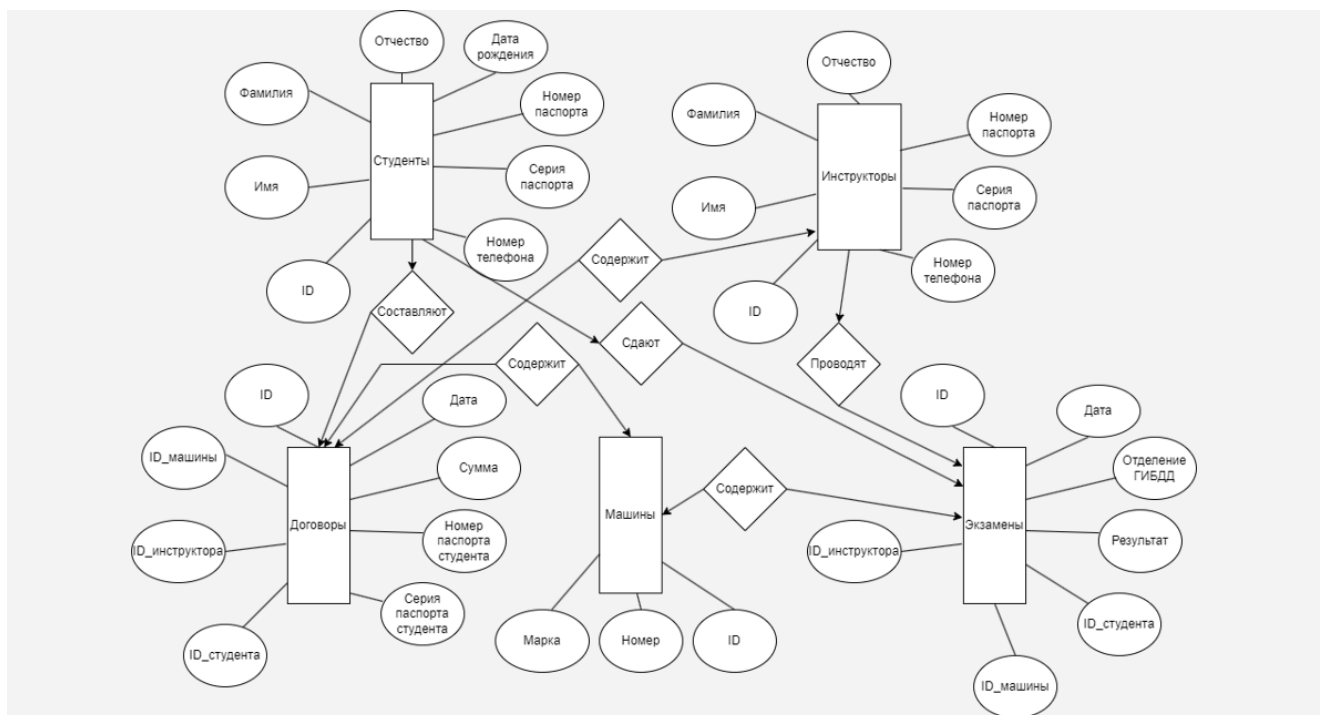


Рис. 1 – Концептуальная модель данных.

Между всеми таблицами существуют связи один ко многим.

2.2. Логическая модель

Логическая модель данных — это абстрактное представление структуры данных, которое используется для планирования и проектирования баз данных. [7]

Ее создание начинается с анализа предметной области и выделения сущностей.

Данная логическая модель данных отражает основные сущности, связанные с автошколой: студенты, договоры, машины, инструкторы, экзамены.

Основные сущности и их атрибуты:

Студенты:

- ID – Уникальный идентификатор студента, первичный ключ
- Фамилия
- Имя
- Отчество

- Дата_Рождения
- Серия_паспорта
- Номер_паспорта
- Номер_телефона

Договоры:

- ID_Студента – внешний ключ
- ID_Инструктора – внешний ключ
- ID_Машины – внешний ключ
- ID— Уникальный идентификатор студента, первичный ключ
- Серия_паспорта_студента
- Номер_паспорта_студента
- Сумма
- Дата

Машины:

- ID – уникальный идентификатор, первичный ключ
- Марка
- Номер

Инструкторы:

- ID – уникальный идентификатор, первичный ключ
- Фамилия
- Имя
- Отчество
- Номер_паспорта
- Серия_паспорта
- Номер_телефона

Экзамены:

- ID – уникальный идентификатор, первичный ключ
- ID_Студента – внешний ключ
- ID_Инструктора – внешний ключ

- ID_Машины – внешний ключ
- Дата
- Отделение_ГИБДД
- Результат

На основе этих данных была составлена логическая модель данных, представленная на рисунке 2.

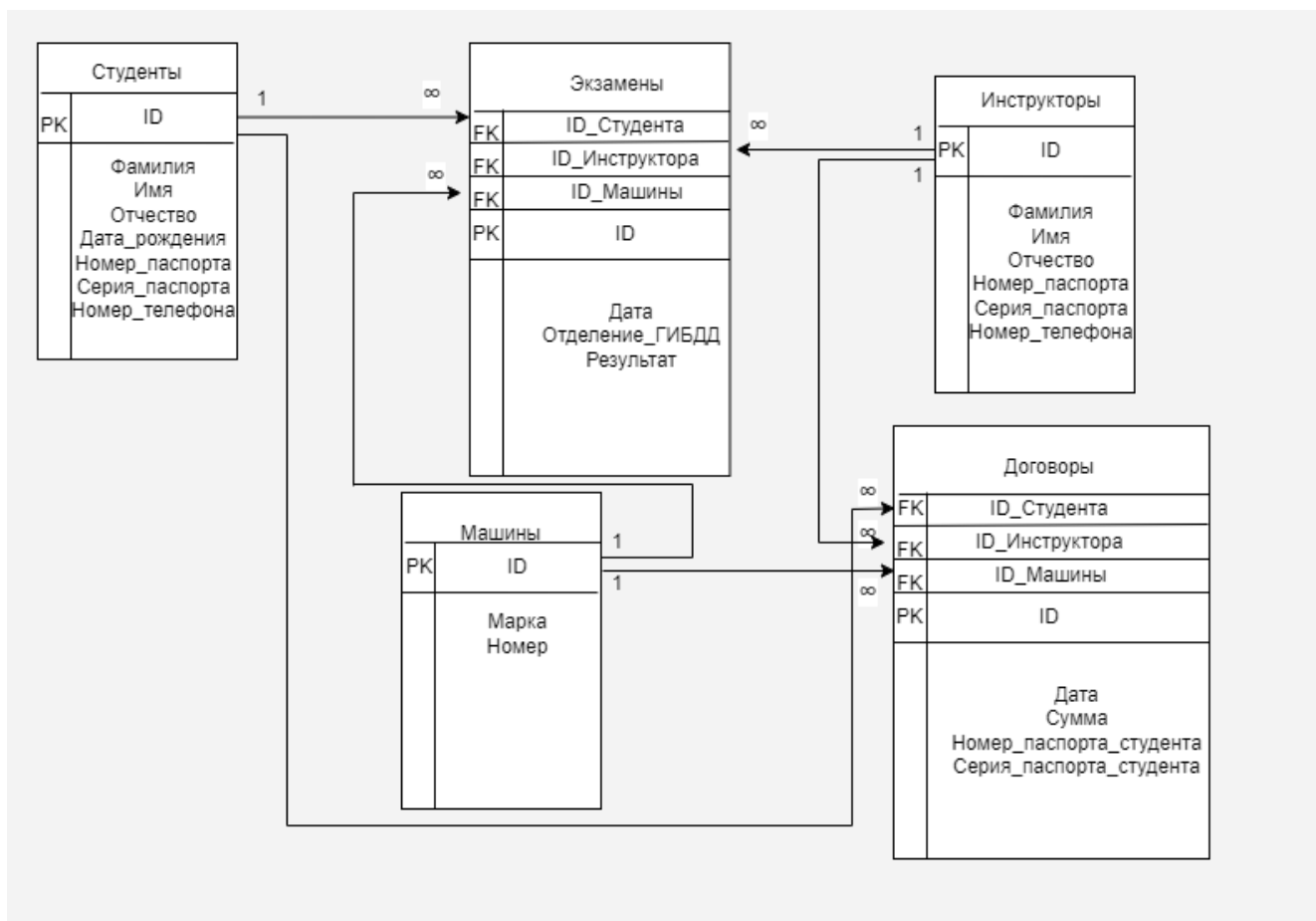


Рис. 2 – Логическая модель данных

2.3. Приведение логической модели к физической

Физическая модель данных — это заключительный шаг в процессе моделирования данных. [7] Она представляет фактические детали реализации в конкретной системе управления базами данных (СУБД).

К основным задачи физической модели данных относятся:

- 1) Определение физических характеристик таблиц и полей:

- Типы данных для каждого атрибута (integer, varchar, date и т.д.)
- Размеры полей (длина строк)
- Ограничения целостности данных (первичные/внешние ключи, уникальность, проверка значений)

2) Проектирование физической структуры базы данных:

- Определение физического хранения данных (файлы, тома, блоки)
- Настройка индексов для ускорения доступа к данным
- Решения по кластеризации таблиц
- Механизмы резервного копирования и восстановления данных

3) Оптимизация производительности:

- Анализ запросов и создание оптимальных индексов
- Денормализация данных для повышения скорости выборки
- Использование материализованных представлений

Физическая модель данных представлена на рисунке 3.

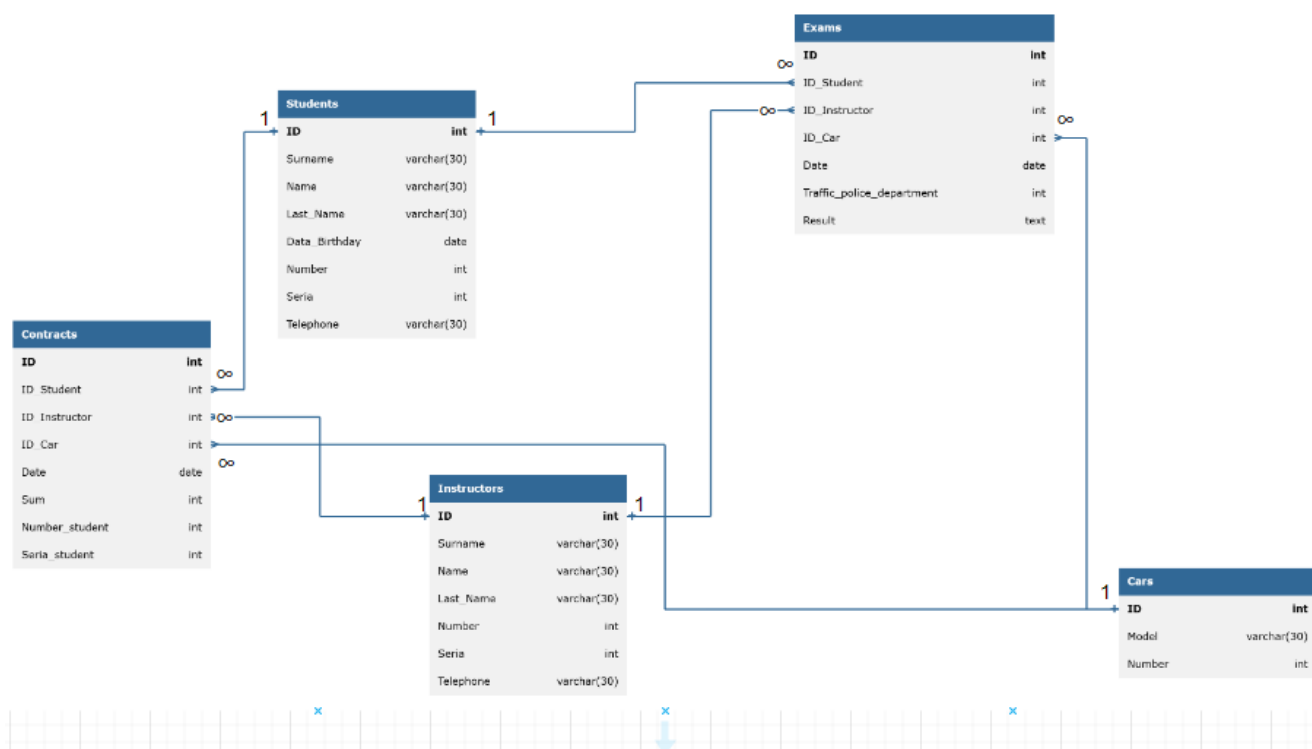


Рис. 3 – Физическая модель данных

3. Разработка и реализация АИС

3.1. Создание SQL-запросов

SQL-запросы являются основным механизмом взаимодействия с реляционными базами данных. Они позволяют эффективно управлять данными, обеспечивая их хранение, извлечение, модификацию и удаление. [7]

Результатом запроса является представление — виртуальная таблица, представляющая собой поименованный запрос, который будет подставлен как подзапрос при использовании представления. [7] В отличие от обычных таблиц реляционных баз данных, представление не является самостоятельной частью набора данных, хранящегося в базе.

1.1.1 Запрос создания таблицы (CREATE TABLE):

Запрос для создания новой таблицы Contract.

```
CREATE TABLE Договоры (  
ID INTEGER PRIMARY KEY AUTOINCREMENT,  
ID_Студента INTEGER NOT NULL  
REFERENCES Студенты (ID),  
ID_Инструктора INTEGER NOT NULL  
REFERENCES Инструкторы (ID),  
ID_Машины INTEGER NOT NULL  
REFERENCES Машины (ID),  
Дата TEXT NOT NULL,  
Сумма REAL NOT NULL,  
Номер_паспорта_студента TEXT NOT NULL,  
Серия_паспорта_студента TEXT NOT NULL  
);
```

1.1.2 Запросы для выборки данных из таблиц (SELECT)

1) Запрос для вывода данных из таблицы Студенты

```
SELECT ID,  
        Фамилия,  
        Имя,  
        Отчество,
```

Дата_рождения,
 Номер_паспорта,
 Серия_паспорта,
 Номер_телефона

FROM Студенты;

Результат выполнения показан на рисунке 4.

	ID	Фамилия	Имя	Отчество	Дата_рождения	Номер_паспорта	Серия_паспорта	Номер_телефона
1	40	Быков	Артем	Максимович	25.03.2004	1234	123456	89108917543
2	41	Иванов	Иван	Максимович	12.04.2000	4321	342987	89108903478
3	42	Сергеев	Сергей	Сергеевич	12.04.2000	9874	098721	89875583901

Рис. 4 – Результат выполнения запроса на таблице Студенты

2) Запрос для извлечения данных из таблицы Договоры и связанных с ней таблиц

```

SELECT
  Договоры.ID,
  Студенты.Фамилия || ' ' || Студенты.Имя || ' ' || Студенты.Отчество AS
  'Студент',
  Инструкторы.Фамилия || ' ' || Инструкторы.Имя || ' ' ||
  Инструкторы.Отчество AS 'Инструктор',
  Машины.Марка AS 'Автомобиль',
  Договоры.Дата,
  Договоры.Сумма,
  Договоры.Номер_паспорта_студента,
  Договоры.Серия_паспорта_студента
FROM
  Договоры
LEFT JOIN
  Студенты ON Договоры.ID_Студента = Студенты.ID
LEFT JOIN
  Инструкторы ON Договоры.ID_Инструктора = Инструкторы.ID
LEFT JOIN
  Машины ON Договоры.ID_Машины = Машины.ID"
  
```

В данном запросе помимо методов, описанных в прошлом примере, функция || ' ' || разделяет данные пробелами. LEFT JOIN – объединяет все данные в одну таблицу через внешние ключи.

Результат выполнения показан на рисунке 5.

	ID	Студент	Инструктор	Автомобиль	Дата	Сумма	Номер_паспорта_студента	Серия_паспорта_студента
1	33	Быков Артем Максимович	Малыгин Александр Александрович	Лада Калина	23.12.2024	30000	1234	123456
2	34	Иванов Иван Максимович	Поляков Алексей Александрович	Лада Калина	12.12.2024	30000	4321	342987
3	35	Сергеев Сергей Сергеевич	Игнатов Михаил Сергеевич	Лада Гранта	12.12.2022	30000	9874	098721

Рис. 5 – Результат выполнения запроса на таблице Договоры

1.1.3 Запрос для вставки в таблицу (INSERT):

Вставка в таблицу Договоры:

```
INSERT INTO Договоры (
ID,
ID_Студента,
ID_Инструктора,
ID_Машины,
Дата,
Сумма,
Номер_паспорта_студента,
Серия_паспорта_студента
)
VALUES (
'ID',
'ID_Студента',
'ID_Инструктора',
'ID_Машины',
'Дата',
'Сумма',
'Номер_паспорта_студента',
'Серия_паспорта_студента'
);
```

Запрос для обновления записей (UPDATE)

Обновление данных в таблице Экзамены:

UPDATE Экзамены

```

SET ID = 'ID',
ID_Студента = 'ID_Студента',
ID_Инструктора = 'ID_Инструктора',
Дата = 'Дата',
ID_Машины = 'ID_Машины',
Отделение_ГИБДД = 'Отделение_ГИБДД',
Результат = 'Результат'
WHERE ID = 'ID' AND
ID_Студента = 'ID_Студента' AND
ID_Инструктора = 'ID_Инструктора' AND
Дата = 'Дата' AND
ID_Машины = 'ID_Машины' AND
Отделение_ГИБДД = 'Отделение_ГИБДД' AND
Результат = 'Результат';

```

1.1.4 Запрос для удаления записей (DELETE)

Удаление данных из таблицы Инструкторы:

```

DELETE FROM Инструкторы
WHERE ID = 'ID' AND
Фамилия = 'Фамилия' AND
Имя = 'Имя' AND
Отчество = 'Отчество' AND
Номер_паспорта = 'Номер_паспорта' AND
Серия_паспорта = 'Серия_паспорта' AND
Номер_телефона = 'Номер_телефона';

```

3.2. Руководство пользователя

При запуске приложения первым шагом для пользователя служит окно авторизации, показанное на рисунке 6. Оно служит для идентификации и аутентификации пользователей.

Авторизация

Логин

Пароль

Рис. 6 – Окно авторизации

В окне авторизации пользователь вводит логин и пароль от своей учетной записи, что показано на рисунках 7 и 8. Если логин и пароль правильные, то он попадает в рабочее окно, что показано рисунках 9-11.

Авторизация

Логин

Пароль

Рис. 7 – Ввод данных от учетной записи студента

Авторизация

Логин

Пароль

Рис. 8 – Ввод данных от учетной записи администратора

■ Студент

Регистрация Инструкторы

Введите ваши данные

Фамилия	Имя	Отчество	Дата рождения	Номер паспорта	Серия паспорта	Номер телефона
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Рис. 9 – Рабочее окно студента

■ Студент

Регистрация Инструкторы

	Фамилия	Имя	Отчество	Номер_паспорта	Серия_паспорта	Номер_телефон
	Малыгин	Александр	Александрович	1111	222222	89107843578
	Поляков	Алексей	Александрович	2543	333333	89100984389
	Игнатов	Михаил	Сергеевич	8974	444444	89107320978
▶*						

Студент

Быков Артем Маи
▼

Инструктор

Малыгин Алексат
▼

Машина

Рено Logan
▼

Выбрать параметр

Фамилия
▼

Ввести значение параметра

Рис.10 – Рабочее окно студента

					МИВУ 10.03.01-03ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		23

■ Администратор

Студенты Инструкторы Договоры Машины Экзамены Архив

	Фамилия	Имя	Отчество	Дата_рождения	Номер_паспорта	Серия_паспорта	Номер_телефон
	Быков	Артём	Максимович	25.03.2004	123456	6543	89108917543
	Иванов	Иван	Максимович	12.04.2000	432178	3429	89108903478
▶	Сергеев	Сергей	Сергеевич	12.04.2000	987489	0987	89875583901
*							

Фамилия Имя Отчество Дата рождения Номер паспорта Серия паспорта Номер телефона

Добавить Изменить Удалить Поиск

Выбрать параметр Ввести значение параметра

Фамилия

Рис. 11 – Рабочее окно администратора

Рабочее окно студента, показанное на рисунке 9-10 имеет следующие функции:

- 1) Регистрация в системе.
- 2) Текстовые поля для ввода данных студента
- 3) Выбор инструктора

Форма администратора, представленная на рисунке 11 состоит из нескольких страниц для каждой из таблиц в базе. Она имеет следующие функции:

- 1) Возможность просмотра необходимых данных, они показаны на рисунках 12 – 17.

	Фамилия	Имя	Отчество	Дата_рождения	Номер_паспорта	Серия_паспорта	Номер_телефон
	Быков	Артём	Максимович	25.03.2004	123456	6543	89108917543
	Иванов	Иван	Максимович	12.04.2000	432178	3429	89108903478
▶	Сергеев	Сергей	Сергеевич	12.04.2000	987489	0987	89875583901
*							

Фамилия
 Имя
 Отчество
 Дата рождения
 Номер паспорта
 Серия паспорта
 Номер телефона

Добавить Изменить Удалить Поиск

 Выбрать параметр
 Ввести значение параметра

Рис. 12 – Данные из таблицы Студенты

	Фамилия	Имя	Отчество	Номер_паспорта	Серия_паспорта	Номер_телефон
▶	Малыгин	Александр	Александрович	111187	2222	89107843578
	Поляков	Алексей	Александрович	254329	3333	89100984389
	Игнатов	Михаил	Сергеевич	897465	4444	89107320978
*						

Фамилия
 Имя
 Отчество
 Номер паспорта
 Серия паспорта
 Номер телефона

Добавить Изменить Удалить Поиск

 Выбрать параметр
 Ввести значение параметра

Рис. 13 – Данные из таблицы Инструкторы

	Студент	Инструктор	Автомобиль	Дата	Сумма	Номер_паспорта	Серия_паспорта
►	Быков Артем ...	Малыгин Алек...	Лада Калина	23.12.2024	30000	123456	1234
	Иванов Иван ...	Поляков Алек...	Лада Калина	12.12.2024	30000	432143	3429
	Сергеев Серге...	Игнатов Миха...	Рено Logan	10.08.2024	30000	987489	0987
*							

Студент Инструктор Машина Сумма Дата Серия паспорта студента

Быков Артем Май ▼ Малыгин Алекса ▼ Рено Logan ▼ 30000 10.08.2024 0987

Выбрать параметр Ввести значение параметра Номер паспорта студента

Добавить Изменить Удалить Поиск Студент ▼ 987489

Рис. 14 – Данные из таблицы Договоры

	Марка	Номер
►	Рено Logan	A507CH132
	Лада Калина	B402HM74
	Лада Гранта	A503GI52
*		

Выбрать параметр Ввести значение параметра

Поиск Марка ▼

Рис. 15 – Данные из таблицы Машины

Администратор

Студенты

Инструкторы

Договоры

Машины

Экзамены

Архив

	Студент	Инструктор	Автомобиль	Дата	Отделение_ГИБД	Результат
▶	Быков Артем ...	Малыгин Алек...	Лада Калина	25.12.2024	3039	Сдал
	Иванов Иван ...	Поляков Алек...	Лада Калина	12.12.2024	3039	Не сдал
	Сергеев Серге...	Игнатов Миха...	Лада Гранта	12.12.2022	3039	Сдал
*						

ID Студента

ID Инструктора

ID Машины

Дата

Отделение ГИБДД

Результат

Быков Артем Маи

Малыгин Алекса

Рено Logan

Выбрать параметр

Ввести значение параметра

Добавить

Изменить

Удалить

Поиск

Студент

Рис. 16 – Данные из таблицы Экзамены

Администратор

Студенты

Инструкторы

Договоры

Машины

Экзамены

Архив

	Студент	Инструктор	Автомобиль	Дата	Сумма	Номер_паспорта	Серия_паспорта
*							

Рис. 17 – Архив

2) Каждая из страниц имеет текстовые поля для ввода данных, а также кнопки для их добавления, обновления или удаления. Для добавления данных в таблицу необходимо заполнить все поля, как показано на рисунке 18 и нажать на кнопку. Результат показан на рисунке 19.

Фамилия	Имя	Отчество	Дата рождения	Номер паспорта	Серия паспорта	Номер телефона
Алексеев	Иван	Петрович	25.01.2000	894437	2214	89107950732

Добавить
Изменить
Удалить
Поиск

Выбрать параметр
 Фамилия ▼

Ввести значение параметра

Рис. 18 – Заполнение полей

	Фамилия	Имя	Отчество	Дата_рождения	Номер_паспорта	Серия_паспорта	Номер_телефон
►	Быков	Артем	Максимович	25.03.2004	123456	6543	89108917543
	Иванов	Иван	Максимович	12.04.2000	432178	3429	89108903478
	Сергеев	Сергей	Сергеевич	12.04.2000	987489	0987	89875583901
	Алексеев	Иван	Петрович	25.01.2000	894437	2214	89107950732
*							

Рис. 19 – Запись добавлена в таблицу

3) Для изменения данных необходимо выбрать запись, изменить данные и нажать на кнопку. Результат показан на рисунках 20-21.

■ Администратор
— □ ×

Студенты
Инструкторы
Договоры
Машины
Экзамены
Архив

	Фамилия	Имя	Отчество	Дата_рождения	Номер_паспорта	Серия_паспорта	Номер_телефон
	Быков	Артем	Максимович	25.03.2004	123456	6543	89108917543
	Иванов	Иван	Максимович	12.04.2000	432178	3429	89108903478
	Сергеев	Сергей	Сергеевич	12.04.2000	987489	0987	89875583901
►	Алексеев	Иван	Петрович	25.01.2000	894437	2214	89107950732
*							

Фамилия	Имя	Отчество	Дата рождения	Номер паспорта	Серия паспорта	Номер телефона
Алексеев	Иван	Петрович	25.01.2000	894437	2218	89107950732

Добавить
Изменить
Удалить
Поиск

Выбрать параметр
 Фамилия ▼

Ввести значение параметра

Рис. 20 – Изменение данных

	Фамилия	Имя	Отчество	Дата_рождения	Номер_паспорта	Серия_паспорта	Номер_телефон
	Быков	Артем	Максимович	25.03.2004	123456	6543	89108917543
	Иванов	Иван	Максимович	12.04.2000	432178	3429	89108903478
	Сергеев	Сергей	Сергеевич	12.04.2000	987489	0987	89875583901
	Алексеев	Иван	Петрович	25.01.2000	894437	2218	89107950732
▶*							

Рис. 21 – Запись обновляется

4) Для удаления данных необходимо также выбрать запись и нажать на кнопку удаления. Результат представлен на рисунках 22-23.

■ Администратор

Студенты Инструкторы Договоры Машины Экзамены Архив

	Фамилия	Имя	Отчество	Дата_рождения	Номер_паспорта	Серия_паспорта	Номер_телефон
	Быков	Артем	Максимович	25.03.2004	123456	6543	89108917543
	Иванов	Иван	Максимович	12.04.2000	432178	3429	89108903478
	Сергеев	Сергей	Сергеевич	12.04.2000	987489	0987	89875583901
▶	Алексеев	Иван	Петрович	25.01.2000	894437	2218	89107950732
*							

Подтверждение удаления

Вы уверены, что хотите удалить эту запись?

Да Нет

Фамилия: Алексей Иван Отчество: Петрович Дата рождения: 25.01.2000 Номер паспорта: 894437 Серия паспорта: 2218 Номер телефона: 89107950732

Добавить Изменить **Удалить** Поиск

Выбрать параметр: Фамилия Ввести значение параметра:

Рис. 22 – Выбор записи

	Фамилия	Имя	Отчество	Дата_рождения	Номер_паспорта	Серия_паспорта	Номер_телефон
	Быков	Артем	Максимович	25.03.2004	123456	6543	89108917543
	Иванов	Иван	Максимович	12.04.2000	432178	3429	89108903478
▶	Сергеев	Сергей	Сергеевич	12.04.2000	987489	0987	89875583901
*							

Рис. 23 - Запись удалена

5) Также каждая из страниц обладает возможностью поиска данных по параметрам. Для этого необходимо выбрать параметр из выпадающего списка, ввести нужное значение в текстовое поле и нажать кнопку поиска. Результаты показаны на рисунках 24-25.

Выбрать параметр Ввести значение параметра

Фамилия Быков

Рис. 24 – Выбор параметра

	Фамилия	Имя	Отчество	Дата_рождения	Номер_паспорта	Серия_паспорта	Номер_телефон
	Быков	Артём	Максимович	25.03.2004	123456	6543	89108917543
»							

Рис. 25 – Результат поиска

3.3. Руководство программиста

3.3.1 Введение

Приложение разработано на языке программирования C# в среде Visual Studio с использованием СУБД SQLite. Программа включает в себя несколько форм: форма авторизации, рабочая форма администратора, рабочая форма студента.

3.3.2 Установка и настройка

Необходимые компоненты:

- Visual Studio 2019 или новее
- .NET Framework 4.7.2 или новее
- СУБД SQLite

Установка:

- 1) Откройте Visual Studio
- 2) Запустите проект

3) Установите пакет SQLite через пункт управление NuGet пакетами

3.3.3 Структура приложения

Проект состоит из следующих файлов:

- Form1.cs – Форма авторизации
- StudentForm.cs – Форма студента
- AdminForm.cs – Форма администратора

3.3.4 Описание форм

1) Form1.cs

- Конструктор Form1: инициализирует форму.
- Метод btnLogin_Click: обрабатывает нажатие кнопки входа, проверяет введенные в текстовые поля логин и пароль и открывает соответствующую форму.

2) StudentForm.cs

- Конструктор StudentForm: инициализирует форму, подключается к базе данных и выполняет запросы, заполняет ComboBox2 данными об инструкторах, ComboBox4 – данными о студентах, ComboBox3 данными о машинах, ComboBox1 параметрами для поиска.
 - Метод LoadInstructors: выполняет SQL-запрос и загружает данные об инструкторах и выводит их в dataGridView1.
 - Метод button7_Click: обрабатывает нажатие кнопки выбора инструктора и машины, выполняет SQL-запрос, заносит запись в таблицу Договоры.
 - Метод button1_Click: обрабатывает нажатие кнопки подтверждения регистрации, считывает данные из textbox1,2,4,5,6,16,20, проверяет их на ошибки и заполняет ими таблицу Студенты.
 - Метод button15_Click: обрабатывает нажатие кнопки поиска, считывает данные из textbox23 и combobox9, производит поиск, делая запрос к таблице Инструкторы.

3) AdminForm.cs

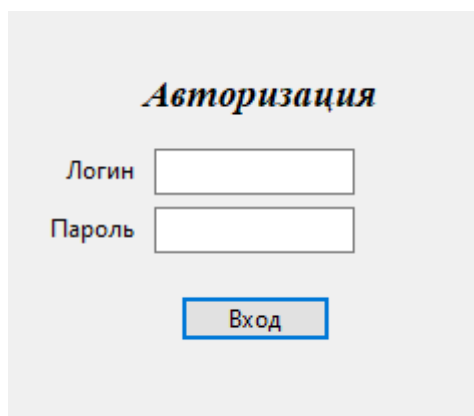
- Конструктор AdminForm.cs: инициализирует форму, устанавливает соединение с базой данных, вызывает методы обращения к базе данных и заполняет dataGridView1-5 данными из таблиц, заполняет ComboBox 1-11 необходимыми данными.
- Метод LoadStudents: выполняет SQL-запрос и заполняет dataGridView1 данными о студентах.
- Метод LoadInstructors: выполняет SQL-запрос и заполняет dataGridView2 данными об инструкторах.
- Метод LoadContracts: выполняет SQL-запрос и заполняет dataGridView3 данными о договорах.
- Метод LoadCars: выполняет SQL-запрос и заполняет dataGridView4 данными о машинах.
- Метод LoadExams: выполняет SQL-запрос и заполняет dataGridView5 данными об экзаменах.
- Метод button3_Click: обрабатывает нажатие кнопки удаления записей из dataGridView1, выполняет SQL-запрос и удаляет запись.
- Метод button1_Click: обрабатывает нажатие кнопки добавления записей в dataGridView1, выполняет SQL-запрос и добавляет запись в зависимости от введенных данных.
- Метод LoadStudentData: делает SQL-запрос и обновляет данные в dataGridView1.
- Метод button4_Click: обрабатывает нажатие кнопки добавления записей в dataGridView2, выполняет SQL-запрос и добавляет запись в зависимости от введенных данных.
- Метод LoadInstructorData: делает SQL-запрос и обновляет данные в dataGridView2.

- Метод button7_Click: обрабатывает нажатие кнопки обновления записей в dataGridView3, выполняет SQL-запрос и обновляет запись в зависимости от введенных данных.
- Метод LoadContractsData: делает SQL-запрос и обновляет данные в datagridview3.
- Метод button8_Click: обрабатывает нажатие кнопки добавления записей в dataGridView5, выполняет SQL-запрос и добавляет запись в зависимости от введенных данных.
- Метод LoadExamsData: делает SQL-запрос и обновляет данные в datagridview5.
- Метод button6_Click: обрабатывает нажатие кнопки удаления записей из dataGridView2, выполняет SQL-запрос и удаляет запись.
- Метод button11_Click: обрабатывает нажатие кнопки удаления записей из dataGridView3, выполняет SQL-запрос и удаляет запись, после заносит запись в архив(datagridview6).
- Метод button10_Click: обрабатывает нажатие кнопки удаления записей из dataGridView5, выполняет SQL-запрос и удаляет запись.
- Метод button9_Click: обрабатывает нажатие кнопки обновления записей в dataGridView5, выполняет SQL-запрос и обновляет запись в зависимости от введенных данных.
- Метод button2_Click: обрабатывает нажатие кнопки обновления записей в dataGridView1, выполняет SQL-запрос и обновляет запись в зависимости от введенных данных.
- Метод button5_Click: обрабатывает нажатие кнопки обновления записей в dataGridView2, выполняет SQL-запрос и обновляет запись в зависимости от введенных данных.
- Метод button12_Click: обрабатывает нажатие кнопки обновления записей в dataGridView3, выполняет SQL-запрос и обновляет запись в зависимости от введенных данных.

- Метод InitializeSearchComboBox: инициализирует и заполняет данными combobox7.
- Метод button13_Click: обрабатывает нажатие кнопки поиска записей из dataGridView1, выполняет SQL-запрос выводит запись.
- Метод InitializeSearchComboBox1: инициализирует и заполняет данными combobox8
- Метод button14_Click: обрабатывает нажатие кнопки поиска записей из dataGridView2, выполняет SQL-запрос выводит запись.
- Метод InitializeSearchComboBox2: инициализирует и заполняет данными combobox9.
- Метод button15_Click: обрабатывает нажатие кнопки поиска записей из dataGridView3, выполняет SQL-запрос выводит запись.
- Метод InitializeSearchComboBox3: инициализирует и заполняет данными combobox10.
- Метод button16_Click: обрабатывает нажатие кнопки поиска записей из dataGridView4, выполняет SQL-запрос выводит запись.
- Метод InitializeSearchComboBox4: инициализирует и заполняет данными combobox11.
- Метод button17_Click: обрабатывает нажатие кнопки поиска записей из dataGridView5, выполняет SQL-запрос выводит запись.
- Метод InitializeDataGridView: инициализирует datagridview6.

4. Тестирование АИС

Тестирование автоматизированной информационной системы является неотъемлемой частью процесса разработки, направленной на обеспечение ее корректной работы, надежности и безопасности. При открытии приложения первым шагом для пользователя является окно авторизации, показанное на рисунке 35, которое служит для идентификации и аутентификации пользователей. Это окно играет важную роль в защите данных и управлении доступом к функционалу системы, что делает его критически важным элементом для тестирования.



The image shows a login window titled "Авторизация" (Authorization). It features two text input fields: the first is labeled "Логин" (Login) and the second is labeled "Пароль" (Password). Below these fields is a button labeled "Вход" (Login/Enter). The entire form is centered on a light gray background.

Рис. 26 – Окно авторизации

В окне авторизации пользователь вводит логин и пароль от своей учетной записи, если логин и пароль правильные, то пользователь попадает в рабочее окно, а если логин или пароль, который был введен не корректен, то приложение выдаст ошибку, что логин и пароль не правильный, это показано на рисунке 27.

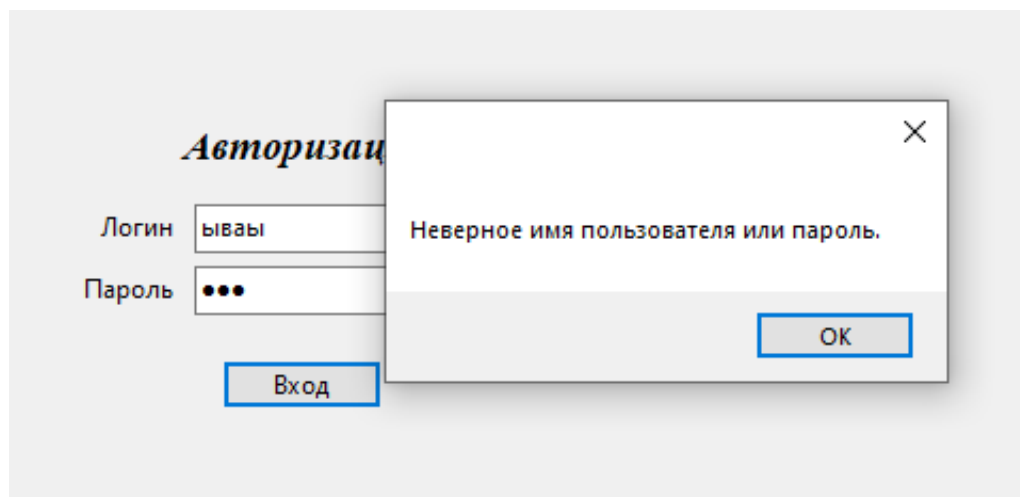


Рис. 27 – Ошибка входа при неправильном логине или пароле

Если логин и пароль правильные, то пользователь попадает в одно из рабочих окон. Начнем с формы студента. Здесь пользователь может выбрать инструктора и машину для обучения, а также выполнить поиск по имеющимся инструкторам.

Если запись не нашлась, то будет выведено сообщение. Результат представлен на рисунках 28-29.

■ Студент

Регистрация Инструкторы

	Фамилия	Имя	Отчество	Номер_паспорта	Серия_паспорта	Номер_телефон
	Малыгин	Александр	Александрович	111187	2222	89107843578
»						

Студент: Быков Артем Май ▾

Машина: Рено Logan ▾

Инструктор: Малыгин Алексат ▾

Выбрать параметр: Ввести значение параметра

Выбрать: Поиск

Фамилия: ▾ Малыгин

Рис. 28 – Корректный поиск

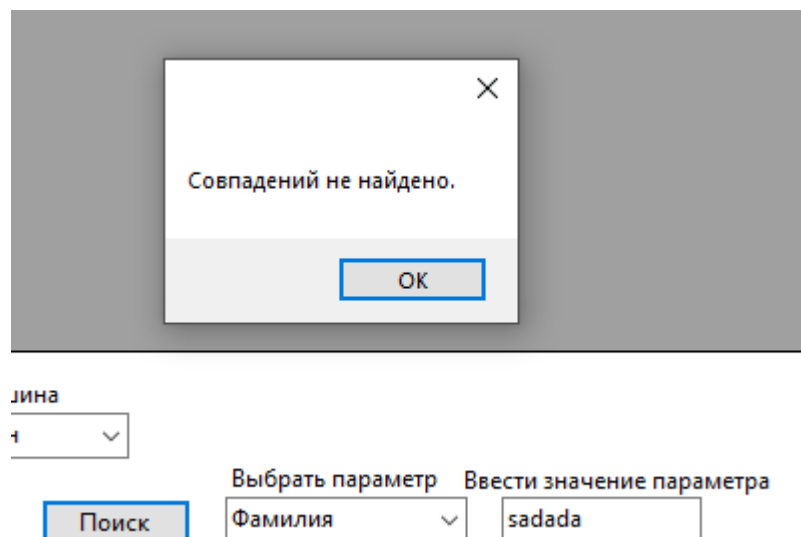


Рис. 29 – Записи не найдены

Перед выбором инструктора и машины пользователю нужно зарегистрироваться в системе.

Пользователь заполняет данные, при неправильном заполнении будут выходить ошибки, как представлено на рисунке 30.

Рис. 30 – Сообщение при неправильном заполнении полей

Если все данные корректны, то при нажатии на кнопку будет выведено сообщение об успешной регистрации, представленное на рисунке 31.

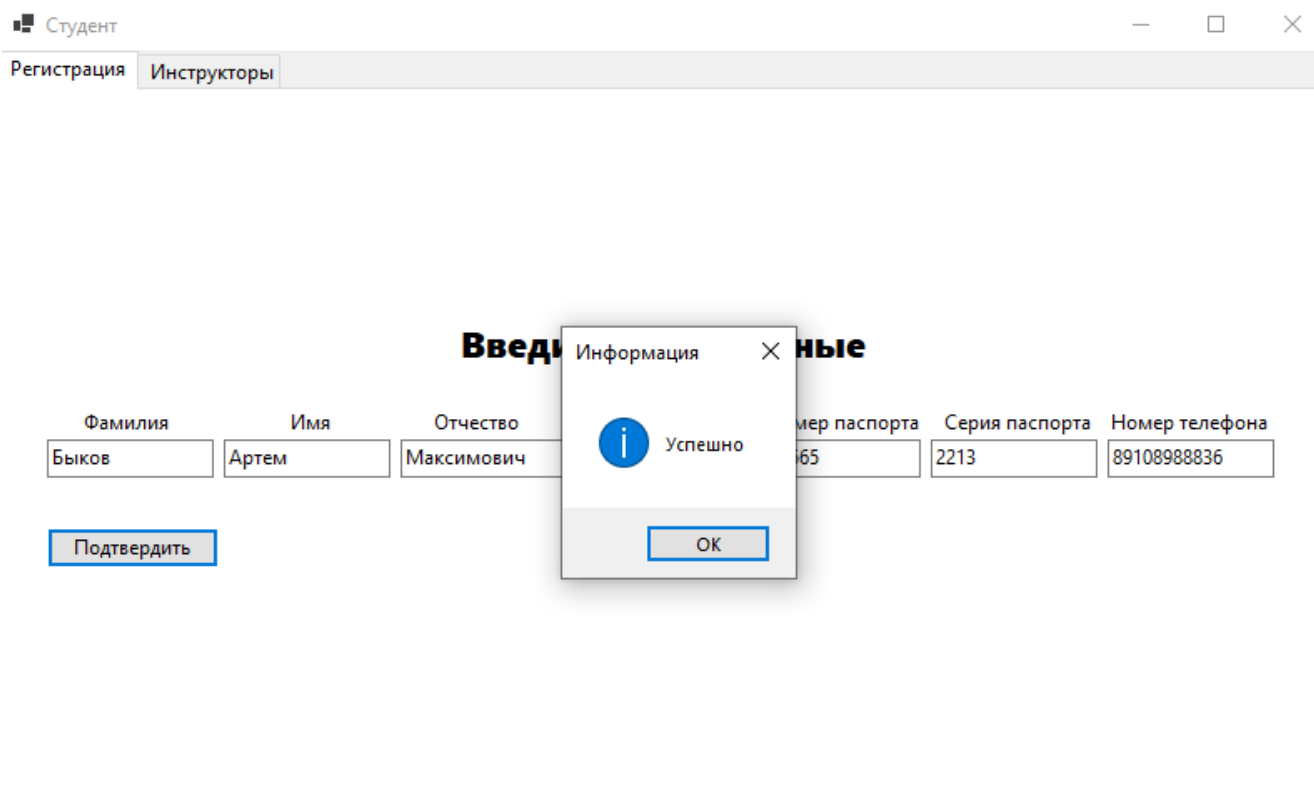


Рис. 31 – Сообщение об успешной регистрации

Теперь перейдем к форме администратора. Он может найти нужные ему данные с помощью поиска. Если запись не нашлась, то будет выведено сообщение. Результат представлен на рисунках 32-33.

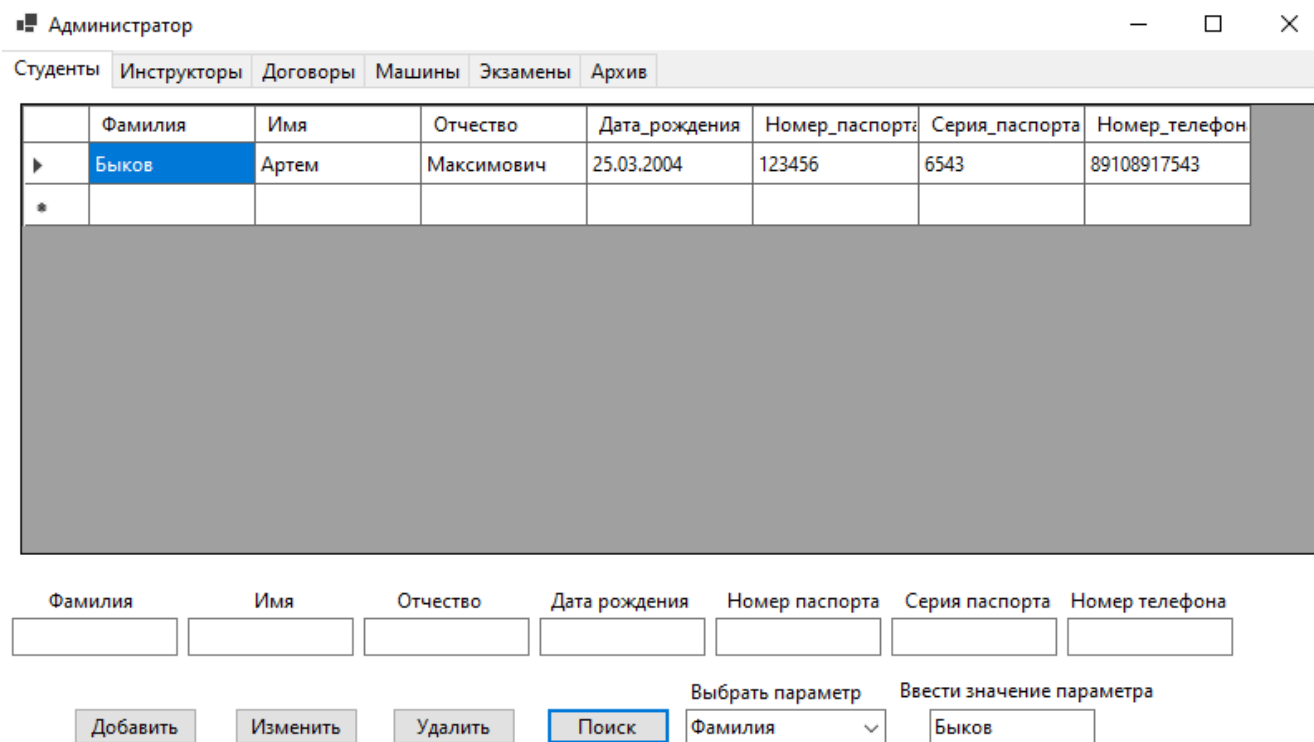


Рис. 32 – Пример корректного поиска

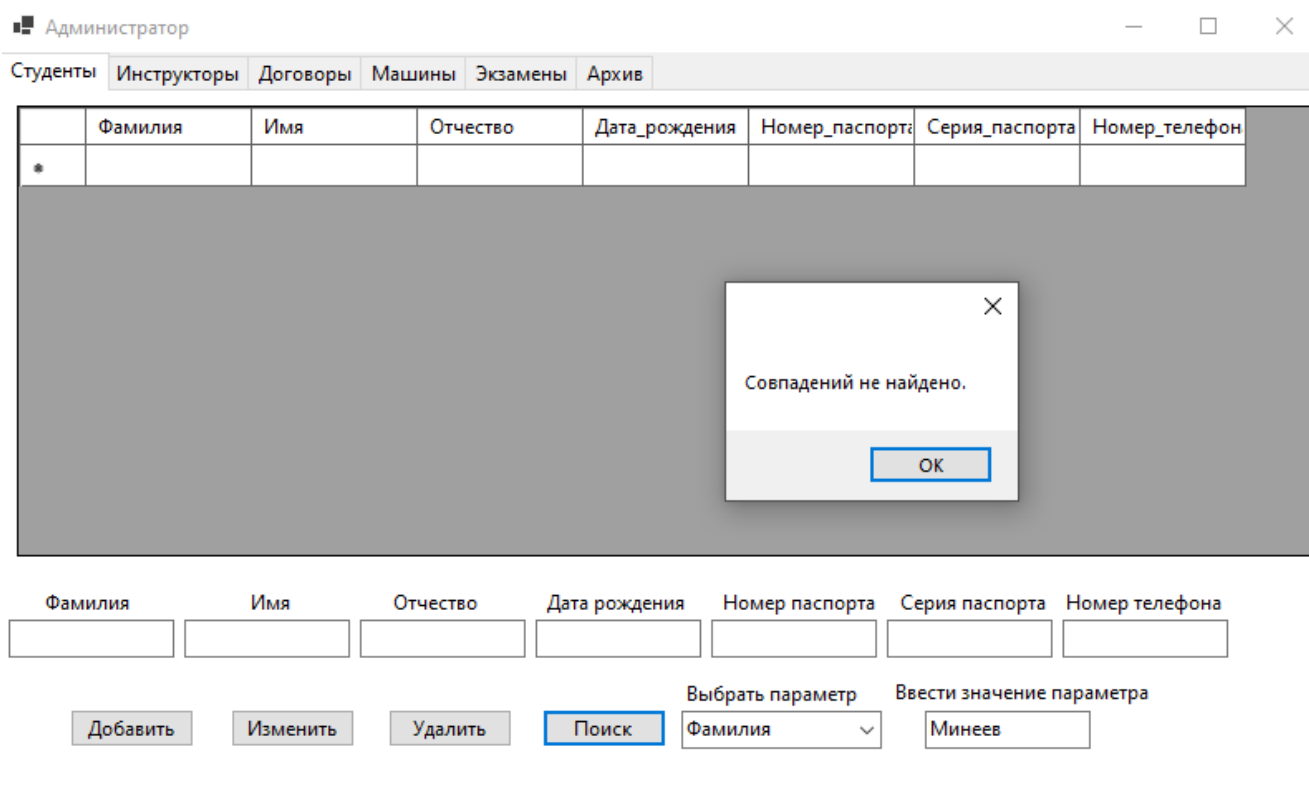


Рис. 33 – Записи не найдены

В данном случае показан пример поиска на таблице Студенты. Для всех остальных таблиц он работает аналогично.

Также администратор может редактировать или удалять записи. Для этого ему нужно выбрать запись. При нажатии на кнопку они будут изменены или удалены. Результат показан на рисунках 34-36.

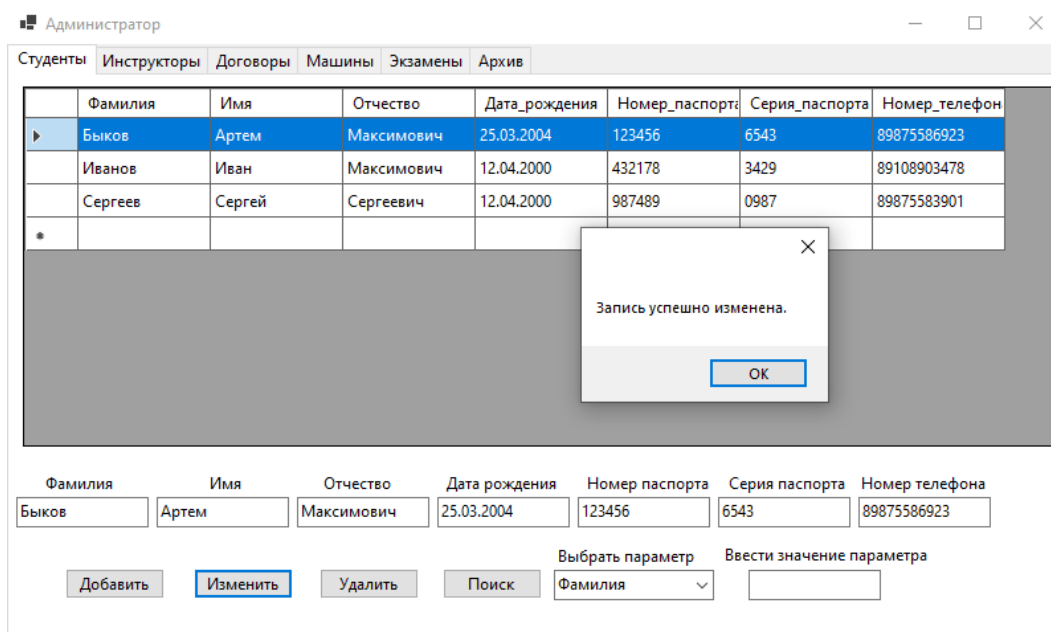


Рис. 34 – Обновление записи

Администратор

Студенты | Инструкторы | Договоры | Машины | Экзамены | Архив

	Фамилия	Имя	Отчество	Дата_рождения	Номер_паспорта	Серия_паспорта	Номер_телефон
	Быков	Артём	Максимович	25.03.2004	123456	6543	89875586923
	Иванов	Иван	Максимович	12.04.2000	432178	3429	89108903478
▶	Сергеев	Алексей	Алексеевич	21.02.2004	123456	2213	89108987534
*							

Запись успешно удалена.

OK

Фамилия: Сергеев
 Имя: Алексей
 Отчество: Алексеевич
 Дата рождения: 21.02.2004
 Номер паспорта: 123456
 Серия паспорта: 2213
 Номер телефона: 89108987534

Добавить
 Изменить
 Удалить
 Поиск
 Выбрать параметр: Фамилия
 Ввести значение параметра:

Рис. 35 – Удаление записи

	Фамилия	Имя	Отчество	Дата_рождения	Номер_паспорта	Серия_паспорта	Номер_телефон
	Быков	Артём	Максимович	25.03.2004	123456	6543	89875586923
▶	Иванов	Иван	Максимович	12.04.2000	432178	3429	89108903478
*							

Рис. 36 – Удаление записи

В данном случае показан пример поиска на таблице Студенты. Для всех остальных таблиц он работает аналогично.

Также администратор может добавить данные в любую таблицу, как показано на рисунке 37.

■ Администратор

Студенты Инструкторы Договоры Машины Экзамены **Архив**

	Фамилия	Имя	Отчество	Дата_рождения	Номер_паспорта	Серия_паспорта	Номер_телефон
	Быков	Артём	Максимович	25.03.2004	123456	6543	89875586923
	Иванов	Иван	Максимович	12.04.2000	432178	3429	89108903478
▶	Сергеев	Алексей	Алексеевич	21.02.2004	123456	2213	89108987534
*							

Фамилия	Имя	Отчество	Дата рождения	Номер паспорта	Серия паспорта	Номер телефона
Сергеев	Алексей	Алексеевич	21.02.2004	123456	2213	89108987534

Добавить
Изменить
Удалить
Поиск

Выбрать параметр
 Фамилия ▼

Ввести значение параметра

Рис. 37 – Добавление данных

Также у администратора есть доступ к архиву. В архив заносятся все удаленные администратором договора, это представлено на рисунке 38.

■ Администратор

Студенты Инструкторы Договоры Машины Экзамены **Архив**

	Студент	Инструктор	Автомобиль	Дата	Сумма	Номер_паспорта	Серия_паспорта
▶	Иванов Иван ...	Поляков Алек...	Лада Калина	12.12.2024	30000	432143	3429
*							

Рис. 38 – Занесение в архив


При занесении данных администратором возможны ошибки в данных, как представлено на рисунке 39.

Администратор

Студенты Инструкторы Договоры Машины Экзамены Архив

	Фамилия	Имя	Отчество	Номер_паспорта	Серия_паспорта	Номер_телефон
▶	Малыгин	Александр	Александрович	111187	2222	89107843578
	Поляков	Алексей	Александрович	254329	3333	89100984389
	Игнатов	Михаил	Сергеевич	897465	4444	89107320978
*						

Ошибка ввода

 Номер паспорта должен состоять ровно из 6 символов.

OK

Фамилия: sada Имя: adad Отчество: dada Номер паспорта: 222222 Серия паспорта: 222 Номер телефона: 89108987531

Выбрать параметр: Фамилия Ввести значение параметра:

Рис. 39 – Ошибка при добавлении данных

Заключение

В ходе данной курсовой работы были решены поставленные задачи, направленные на создание Автоматизированной Информационной Системы (АИС) для автошколы. Основное внимание уделялось разработке базы данных, обеспечивающей эффективное управление данными. Использование такой системы позволяет автоматизировать процессы оформления договоров.

Разработанная АИС также значительно улучшила взаимодействие пользователей и персонала автошколы.

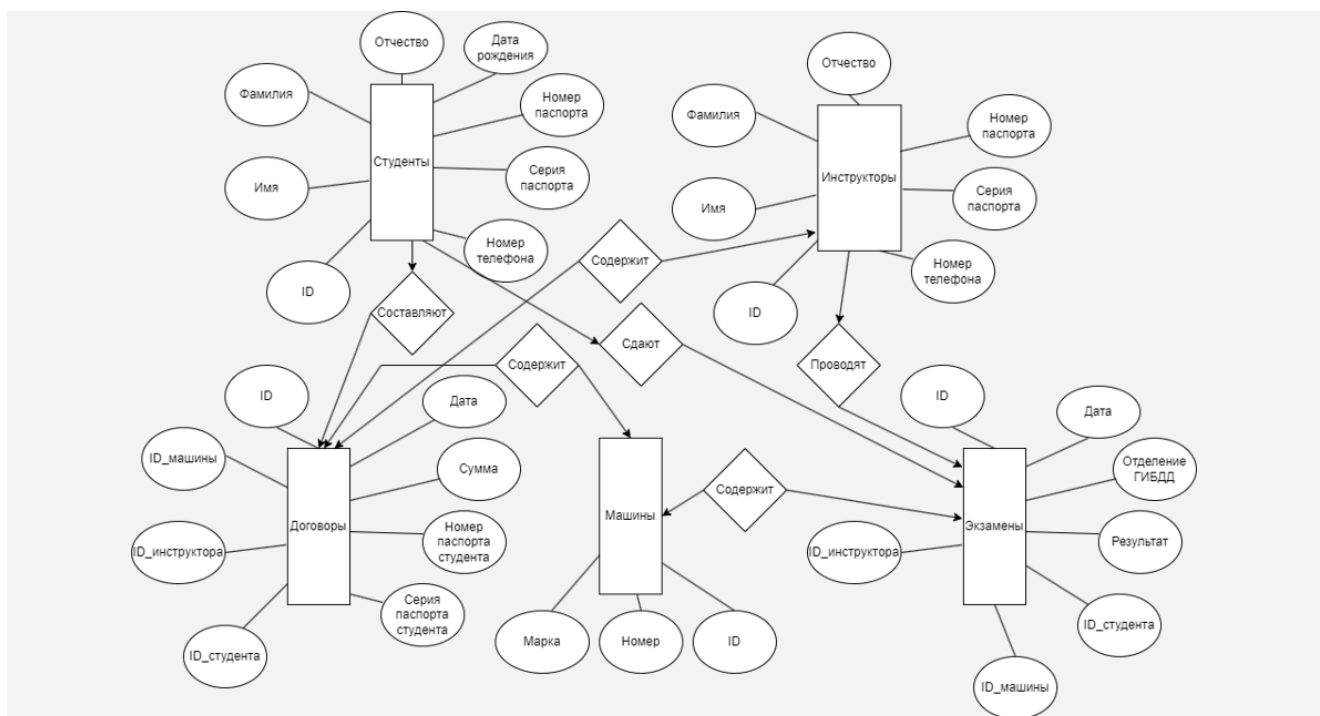
На основании выполненного сравнительного анализа производительности различных СУБД и средств программирования были выбраны оптимальные решения для разработки данной базы данных и многопользовательского приложения. Это позволило создать надежную, производительную и масштабируемую систему, готовую к дальнейшему расширению.

Список литературы

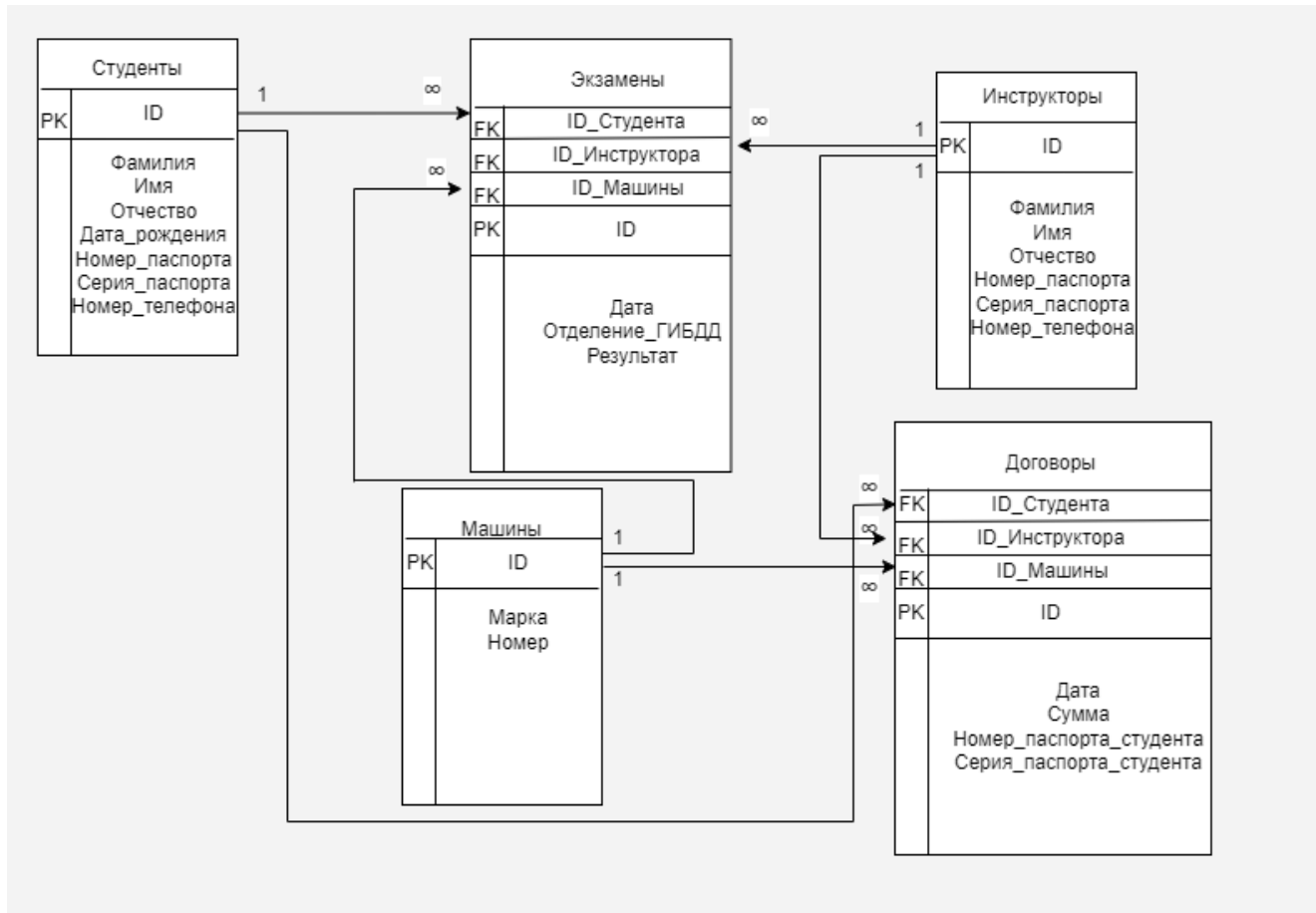
- 1) Бьерн Страуструп, Язык программирования C++. Специальное издание. Пер. с англ. – М.: Издательство Бином, 2011 г. – 1136 с.: ил.
- 2) Казарин С.А., Клишин А.П. К 143 Среда разработки Java-приложений Eclipse: (ПО для объектно-ориентированного программирования и разработки приложений на языке Java): Учебное пособие. Москва 2008. — 77 с.
- 3) Медведев, М. А. М42 Программирование на СИ# : учеб. пособие / М. А. Медведев, А. Н. Медведев. — Екатеринбург : Изд-во Урал. ун-та, 2015. — 64 с.
- 4) Мотев А. А. М85 Уроки MySQL. Самоучитель. — СПб.: БХВ-Петербург, 2006. — 208 с.: ил.
- 5) Рындина С. В.. – Пенза : Изд-во ПГУ, 2023. – 82 с. , Цифровые технологии управления получением, хранением, передачей и обработкой больших данных: SQLite : учеб.-метод. пособие.
- 6) Сапаров А.Ю., Разработка Windows Forms приложений на языке программирования C#: учебно-методическое пособие / Сост.: А.Ю. Сапаров, Ижевск, 2020. 61 с.
- 7) Сергеева Т.И. Базы данных: модели данных, проектирование, язык SQL: учеб. пособие / Т.И. Сергеева, М.Ю. Сергеев. Воронеж: ФГБОУ ВПО «Воронежский государственный технический университет», 2012. 233 с.

Приложение А: Модели данных

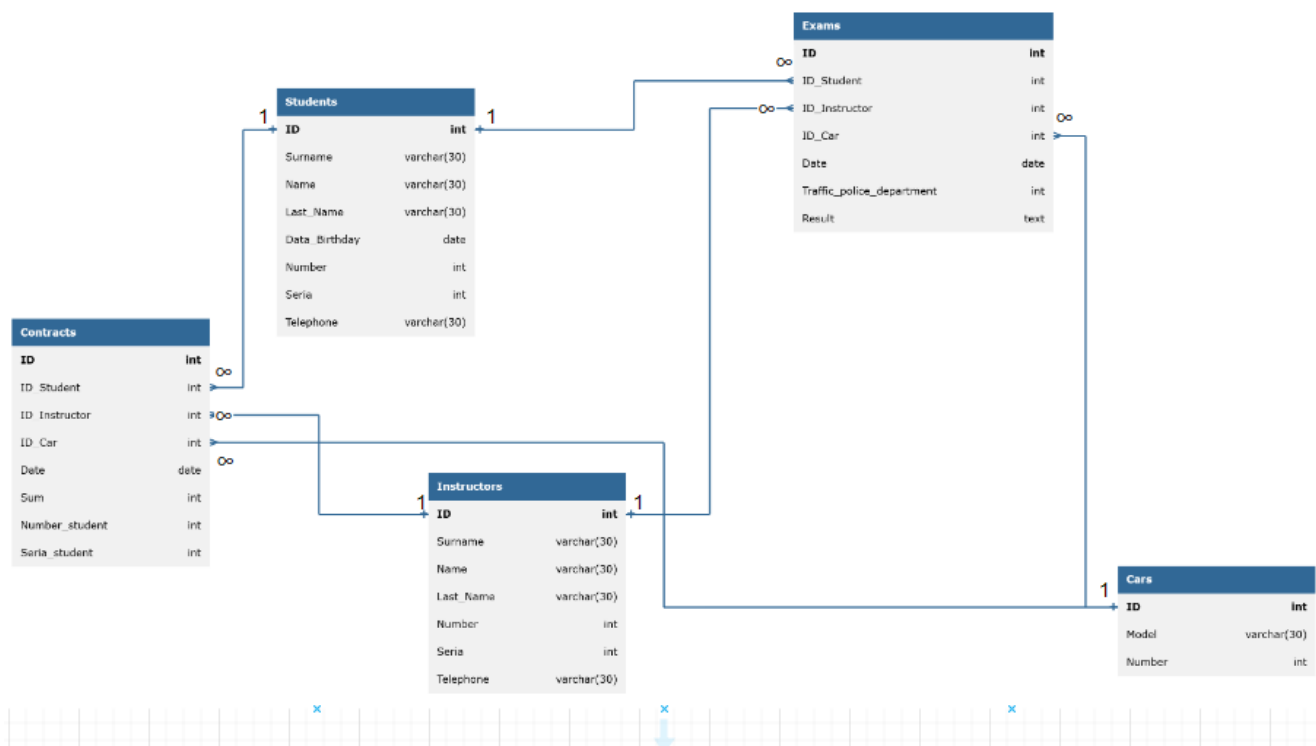
Концептуальная модель.



Логическая модель.



Физическая модель.



Приложение Б: Текст кода

Весь исходный код представлен по ссылке:

<https://github.com/artiombykov/databasekursovaya>