



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Международных образовательных программ»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии» (ИУ7)

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

Тестовая система с вариантами ответа

Студент группы ИУ7и-64Б

(Подпись, дата)

Динь Вьет Ань

(И.О. Фамилия)

Руководитель

(Подпись, дата)

Исаев А. Л.

(И.О. Фамилия)

2023 г.

РЕФЕРАТ

Расчетно-пояснительная записка 29 с., 15 рис., 2 табл..

В курсовой работе необходимо проектировать и разработать базу данных для тестовой системы с вариантами ответов. В базе данных хранятся информация о пользователях, информация о предметах, информация о вопросах и информация о тестах. Реализовать приложение, которое обеспечивает интерфейс для взаимодействия с базой данных для разных типов пользователей.

Ключевые слова: базы данных, тестовая система с вариантами ответов, СУБД, C#, PostgreSQL.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Аналитическая часть	6
1.1 Формализация задачи	6
1.2 Формализация данных	6
1.3 Типы пользователей	7
1.4 Анализ баз данных по модели данных	10
1.4.1 Дореляционные базы данных	10
1.4.2 Реляционные базы данных	10
1.4.3 Постреляционные базы данных	11
2 Конструкторская часть	12
2.1 Проектирование базы данных	12
2.1.1 Таблицы базы данных	12
2.1.2 Целостность данных БД	14
2.2 Требования к программе	16
3 Технологическая часть	17
3.1 Средства реализации	17
3.1.1 Выбор языка программирования	17
3.1.2 Выбор СУБД	17
3.2 Создание объектов БД	17
3.2.1 Создание таблиц	17
3.2.2 Создание связей	18
3.2.3 Ролевая модель на уровне БД	19
3.3 Интерфейс программы	20
4 Исследовательская часть	25
4.1 Технические характеристики	25
4.2 Описание эксперимента	25
ЗАКЛЮЧЕНИЕ	27
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	28

ВВЕДЕНИЕ

Идея образования с помощью компьютеров витала давно. Первые попытки относятся к концу 1950-х годов, и сегодня возможности использования компьютеров и информационных технологий в образовании очень разнообразны.

На современном этапе развития образования все больше внимания уделяется внедрению эффективных форм контроля знаний. Одной из таких форм является проведение тестирования. Это наиболее стандартизованный и объективный метод контроля и оценивания знаний, умений и навыков испытуемого, который лишен таких традиционных недостатков других методов контроля знаний, как неоднородность требований, субъективность экзаменаторов и т. п. Наиболее эффективным является проведение тестирования с помощью компьютерных систем тестирования.

Использование компьютера дает возможность преподавателю более совершенно управлять образовательным процессом, использовать объективные методы контроля знаний и, конечно, экономит время. Именно этим и обусловлена актуальность темы данной курсовой работы.

Цель данной работы — разработка программного обеспечения для тестовой системы с вариантами ответов.

Чтобы достигнуть поставленной цели должно решить следующие задачи:

- формализовать задачу и определить требуемый функционал;
- провести анализ существующих СУБД и выбрать наиболее подходящую;
- описать объекты, из которых состоит база данных, и связи между ними;
- спроектировать базу данных и реализовать программное обеспечение, позволяющее пользователям получать информации о тестовой системе;
- провести исследование времени выполнения запросов к БД с использованием индексов и без.

1 Аналитическая часть

1.1 Формализация задачи

Необходимо спроектировать базу данных для хранения информации об объектах в системе: пользователях, вопросах, тестах, связях между тестами и вопросами и результатах выполнения тестов. Также необходимо реализовать программное обеспечение с интерфейсом, позволяющим работать с этой базой данных для получения и изменения хранящейся в ней информации. Требуется как минимум три типа ролей – студент, преподаватель и администратор.

1.2 Формализация данных

Как сказано выше, в базе данных хранится информация о следующих объектах:

- пользователи;
- предметы;
- вопросы;
- тесты;
- результаты тестирования студентов;
- связи тестов и вопросов.

Ниже, в таблице 1.1 приведены категории и сведения о данных.

Таблица 1.1 – Категории и сведения о данных

Категория	Сведения
Пользователь	ID пользователя, логин, пароль, имя пользователя, права доступа.
Предмет	ID предмета, название предмета.
Вопрос	ID вопроса, ID предмета, текст вопроса, ответы, правильный ответ.
Тест	ID тест, ID предмета, название теста, код теста, время на выполнение.
Связь вопросов и тестов	ID связи, ID теста, ID вопроса.
Результат	ID результата, ID пользователя, ID теста, начальное время, конечное время, баллы, максимальный балл.

На рисунке 1.1 отображена ER-диаграмма системы, основанная на приведенной выше таблице.

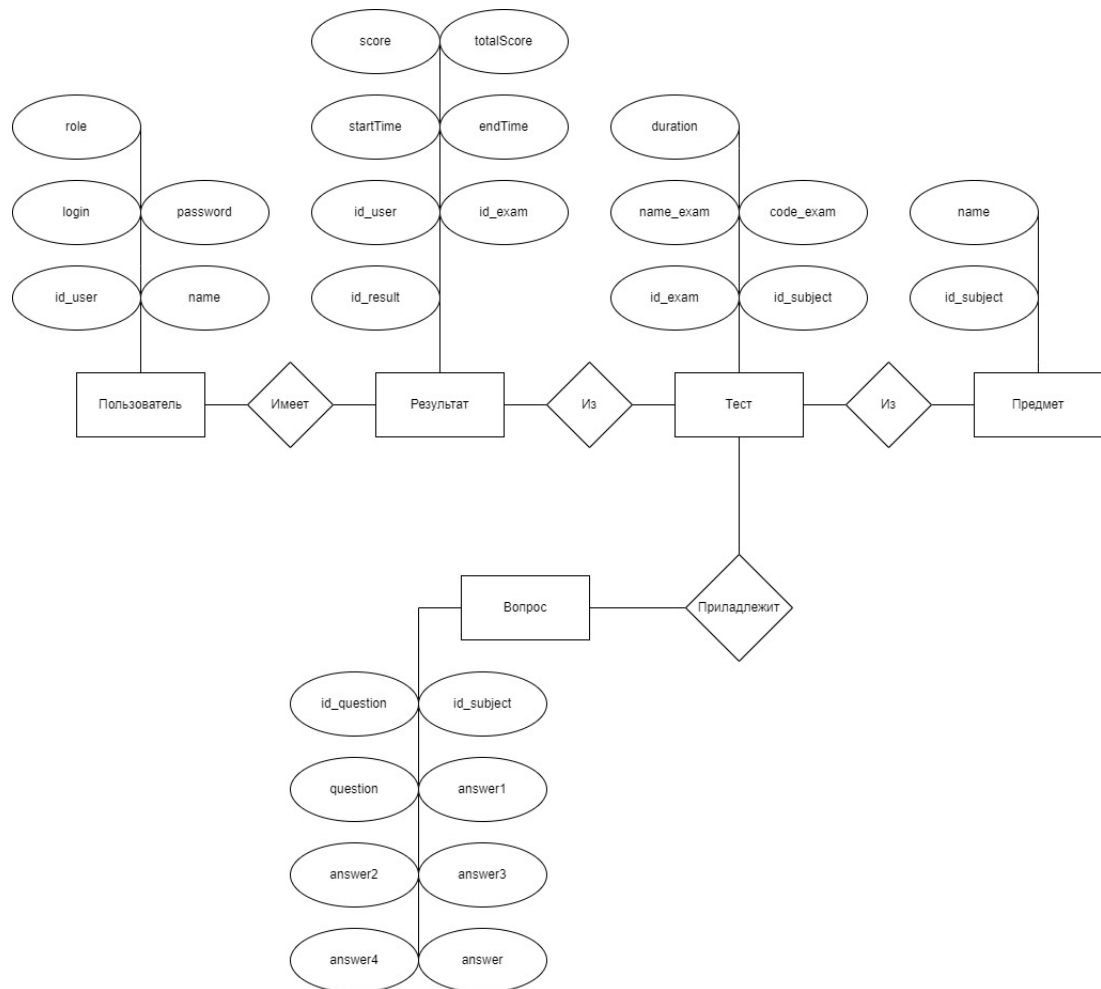


Рисунок 1.1 – ER-диаграмма

1.3 Типы пользователей

В соответствии с поставленной задачей необходима авторизация пользователей. Чтобы управлять приложением, требуется использовать модель ролей с тремя типами пользователей: студент, преподаватель и администратор. Каждый тип пользователя имеет свой набор функций

Ниже, в таблице 1.2 приведены типы пользователей и их набор возможностей.

Таблица 1.2 – Типы пользователей и их набор возможности

Тип пользователя	Возможности
Студент	Авторизация. Изменение своя пароля. Сдача теста. Посмотр своих результатов. Выход.
Преподаватель	Авторизация. Изменение своя пароля. Добавление, изменение и удаление информации о вопросах и тестах. Посмотр результатов. Выход.
Администратор	Авторизация. Изменение своя пароля. Сброс пароля аккаунтов студентов и преподавателей. Добавление, изменение и удаление информации о аккаунтах других пользователей. Добавление, изменение и удаление информации о вопросах, тестах и предметах. Посмотр результатов. Выход.

На рисунке 1.2 представлена диаграмма прецедентов для каждого типа пользователей.

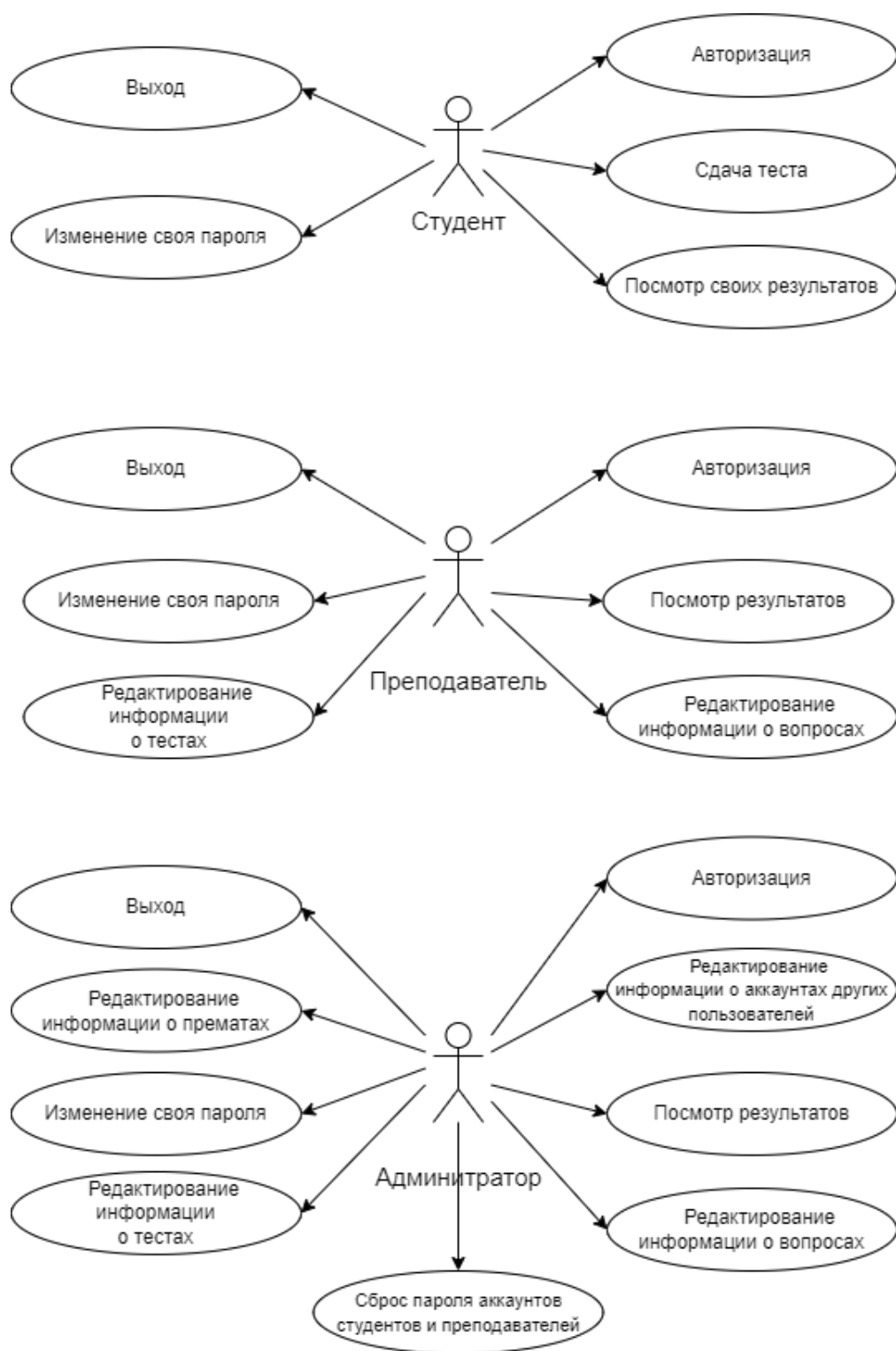


Рисунок 1.2 – Диаграмма прецедентов

1.4 Анализ баз данных по модели данных

База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе [1].

Система управления базами данных (СУБД) — это программный комплекс, обеспечивающий централизованное хранение данных и предоставляющий приложениям услуги по обработке данных [2].

Существует три группы моделей баз данных, отличающиеся структурой организации данных:

- дореляционные базы данных;
- реляционные базы данных;
- постреляционные базы данных.

1.4.1 Дореляционные базы данных

СУБД, основанные на дореляционной модели, в свою очередь, имеют три типа организации:

- система, основанная на инвертированных списках;
- иерархическая — логическая модель данных в виде структуры дерева, представляющая собой совокупность элементов, расположенных в порядке их подчинения от общего к частному [3];
- сетевая — логическая модель данных, которая является расширением иерархического подхода, строгая математическая теория, описывающая структурный аспект, аспект целостности и аспект обработки данных в сетевых базах данных [3].

Основными преимуществами дореляционных баз данных являются скорость доступа к данным и высокая производительность. А недостатками этой моделей данных являются сложность понимания и ограниченная поддержка для больших объемов данных.

1.4.2 Реляционные базы данных

Реляционная модель представляет собой совокупность данных, состоящую из набора двумерных таблиц [4]. Таблицы состоят из строк — записей с уникальным идентификатором, и столбцов — полей. На пересечении строк и столбцов находятся значения — атрибуты данных.

Преимущества реляционных баз данных:

- простота и наглядность;

- простота установки взаимосвязи между элементами данных;
- масштабируемость;
- поддержка целостности и безопасности данных.

Недостатки реляционных баз данных:

- скорость доступа к данным;
- не всегда предметную область можно представить в виде совокупности таблиц;
- необходимость размещения данных внутри таблицы до начала обработки данных.

Реляционная модель является удобной и наиболее привычной формой представления данных. Наиболее популярными реляционными СУБД являются Oracle, Microsoft SQL Server и PostgreSQL.

1.4.3 Постреляционные базы данных

Постреляционная модель базы данных - является расширенной версией реляционной модели базы данных и позволяет уменьшить ограничение на неделимости данных, хранящихся в записях таблиц. Именно поэтому хранение данных в постреляционной модели по сравнению с реляционной считается более эффективным [5].

Постреляционная модель данных допускает многозначные поля - поля, значения которых состоят из подзначений. Набор значений многозначных полей считается самостоятельной таблицей, встроенной в основную таблицу.

Достоинство постреляционной модели состоит в том, что она дает возможность формирования совокупности связанных реляционных таблиц через одну постреляционную таблицу, что обеспечивает высокую наглядность представления информации и эффективность ее обработки. Недостатком постреляционной модели является сложность решения проблемы обеспечения целостности и непротиворечивости хранимых данных [6].

Вывод

В данном разделе была проведена формализация задачи и данных, рассмотрены типы пользователей и требуемые функционалы. Также был проведен анализ существующих моделей баз данных и было решено использовать в данной работе реляционную СУБД.

2 Конструкторская часть

2.1 Проектирование базы данных

2.1.1 Таблицы базы данных

В соответствии с ER-диаграммой системы, изображенной на рисунке 1.1, база данных приложения хранит следующие таблицы:

- таблица пользователей Users;
- таблица предметов Subjects;
- таблица вопросов Questions;
- таблица тестов Exams;
- таблица результатов Results;
- таблица связей тестов и вопросов Exam_Question.

На рисунке 2.1 представлена диаграмма разрабатываемой базы данных.

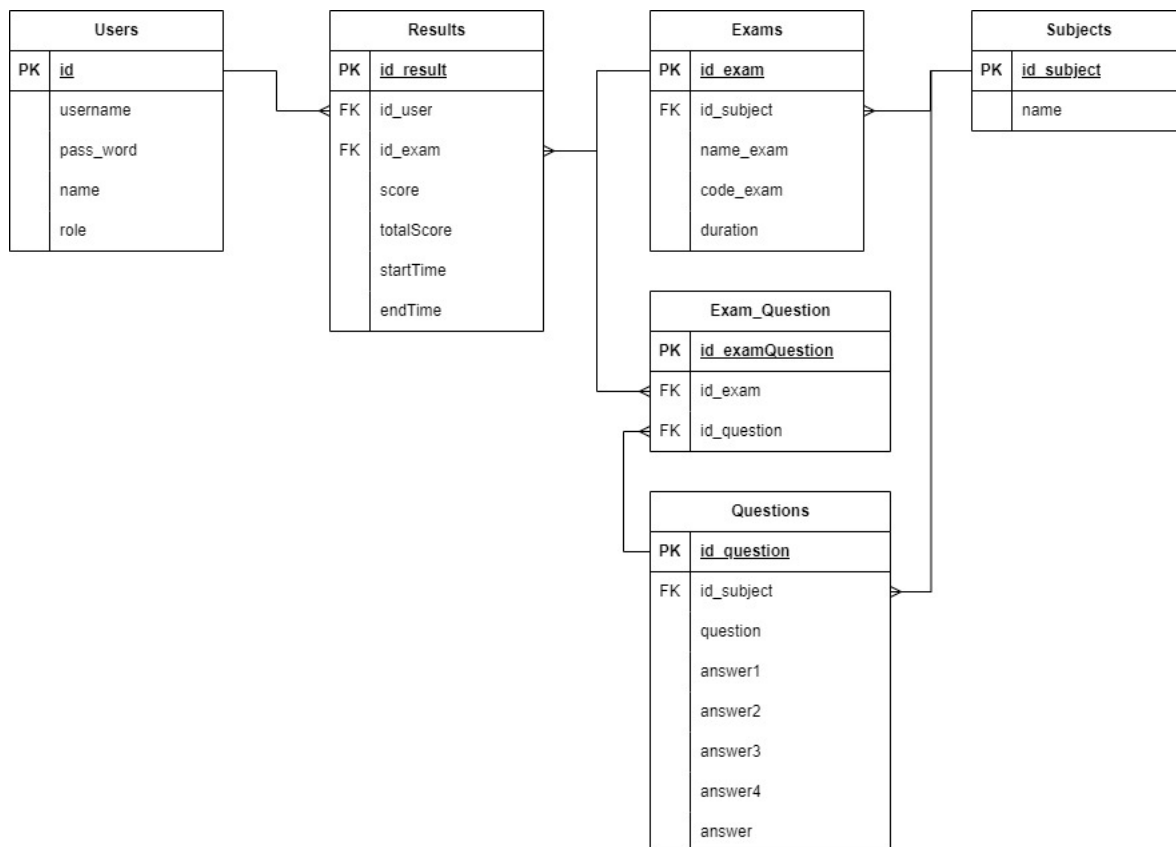


Рисунок 2.1 – Диаграмма базы данных

Таблица пользователей **Users** хранит информации о пользователях и содержит следующие поля:

- id — первичный ключ, идентификатор аккаунта, INT;
- username — логин пользователя, VARCHAR;

- pass_word — пароль пользователя, VARCHAR;
- name — имя пользователя, VARCHAR;
- role — роль (права доступа), VARCHAR.

Таблица предметов **Subjects** хранит информации о предметах и содержит следующие поля:

- id_subject — первичный ключ, идентификатор предмета, INT;
- name — название предмета, VARCHAR.

Таблица вопросов **Questions** хранит информации о вопросах и содержит следующие поля:

- id_question — первичный ключ, идентификатор вопроса, INT;
- id_subject — внешний ключ, соответствует идентификатору предмета, INT;
- question — пароль пользователя, VARCHAR;
- answer1 — первый вариант ответа, VARCHAR;
- answer2 — второй вариант ответа, VARCHAR;
- answer3 — третий вариант ответа, VARCHAR;
- answer4 — четвертый вариант ответа, VARCHAR;
- answer — правильный ответ, VARCHAR.

Таблица тестов **Exams** хранит информации о тестах и содержит следующие поля:

- id_exam — первичный ключ, идентификатор теста, INT;
- id_subject — внешний ключ, соответствует идентификатору предмета, INT;
- name_exam — название теста, VARCHAR;
- code_exam — код теста, VARCHAR;
- duration — время для выполнения теста, INT.

Таблица результатов **Results** хранит информации о результатах и содержит следующие поля:

- id_result — первичный ключ, идентификатор результата, INT;
- id_user — внешний ключ, соответствует идентификатору пользователя, INT;
- id_exam — внешний ключ, соответствует идентификатору теста, INT;
- score — баллы, INT;
- totalScore — максимальный балл теста, INT;
- startTime — начальное время, VARCHAR;
- endTime — конечное время, VARCHAR.

Таблица связей тестов и вопросов **Exam_Question** хранит информации о связях тестов и вопросов и содержит следующие поля:

- `id_examQuestion` — первичный ключ, идентификатор связи, INT;
- `id_exam` — внешний ключ, соответствует идентификатору теста, INT;
- `id_question` — внешний ключ, соответствует идентификатору вопроса, INT.

2.1.2 Целостность данных БД

Целостность таблиц

Для обеспечения целостности таблиц необходимо, чтобы все строки в таблице имели уникальный идентификатор (первичный ключ). Как было описано выше, в разделе 2.1.1, каждая таблица имеет первичный ключ:

1. у таблицы `Users` - поле `id`;
2. у таблицы `Subjects` - поле `id_subject`;
3. у таблицы `Questions` - поле `id_question`;
4. у таблицы `Exams` - поле `id_exam`;
5. у таблицы `Results` - поле `id_result`;
6. у таблицы `Exam_question` - поле `id_examQuestion`;

Целостность ссылок

Для обеспечения ссылочной целостности необходимо создать внешние ключи, которые поддерживает согласованное состояние данных между двумя таблицами (таблицы, на которую ссылаются, и таблицы, которая ссылается на другую).

1. Между таблицами `Exams` и `Subjects` существует связь один ко многим поэтому создается один внешний ключ: `id_subject` таблицы `Exams` ссылается на первичный ключ таблицы `Subjects` (поле `id_subject`).
2. Между таблицами `Questions` и `Subjects` существует связь один ко многим поэтому создается один внешний ключ: `id_subject` таблицы `Questions` ссылается на первичный ключ таблицы `Subjects` (поле `id_subject`).
3. Таблица `Results` нужна для формализации связи один ко многим между таблицами `Users` и `Exams`, поэтому в ней создаются два внешних ключа:
 - 1) поле `id_user` ссылается на первичный ключ таблицы `Users` (поле `id_user`);
 - 2) поле `id_exam` ссылается на первичный ключ таблицы `Exams` (поле `id_exam`).
4. Таблица `Exam_Question` нужна для формализации связи один ко многим

между таблицами Exams и Questions, поэтому в ней создаются два внешних ключа:

- 1) поле id_exam ссылается на первичный ключ таблицы Exams (поле id_exam);
- 2) поле id_question ссылается на первичный ключ таблицы Questions (поле id_question).

Целостность полей

Для обеспечения целостности полей необходимо указать набор значений данных, которые являются допустимыми для поля, и определить, возможно ли использование нулевого значения.

1. Таблица Users: для полей id, username, pass_word, name и role не допустимо нулевое значение. Поле username также должно быть уникальным (т.к. авторизация пользователя осуществляется по его логину). Поле role может принимать только значения A, T и S.
2. Таблица Exams: для полей id_exam, id_subject, name_exam, code_exam и duration не допустимо нулевое значение. Поле code_exam также должно быть уникальным.
3. Таблица Questions: для всех полей не допустимо нулевое значение.
4. Таблица Subjects: для полей id_subject и name не допустимо нулевое значение.
5. Таблица Results: для всех полей не допустимо нулевое значение.
6. Таблица Exam_Question: для всех полей не допустимо нулевое значение.

2.2 Требования к программе

В соответствии с Use-Case диаграммой, представленной ранее на рисунках 1.2, программа должна предоставлять следующие возможности.

Возможности студента:

1. авторизация;
2. изменение своя пароля;
3. сдача теста;
4. просмотр своих результатов;
5. выход.

Возможности преподавателя:

1. авторизация;
2. изменение своя пароля;
3. добавление, изменение и удаление информации о вопросах;
4. добавление, изменение и удаление информации о тестах;
5. просмотр результатов;
6. выход.

Возможности админимтратора:

1. авторизация;
2. изменение своя пароля;
3. добавление, изменение и удаление информации о аккаунтах других пользователей.
4. сброс пароля аккаунтов студентов и преподавателей;
5. добавление, изменение и удаление информации о предметах;
6. добавление, изменение и удаление информации о вопросах;
7. добавление, изменение и удаление информации о тестах;
8. просмотр результатов;
9. выход.

Вывод

В данном разделе были спроектированы база данных и приложение для доступа к ней.

3 Технологическая часть

3.1 Средства реализации

3.1.1 Выбор языка программирования

В качестве языка программирования был выбран C#, так как:

1. данный язык программирования объектно-ориентирован;
2. этот язык имеет пакет для работы с выбранной СУБД PostgreSQL;
3. имеется навыки использования данного языка программирования.

В качестве среды разработки была выбрана «Visual Studio 2022», так как:

1. студенты могут использовать бесплатно;
2. она имеет множество удобств, которые облегчают процесс написания и отладки кода;
3. она обеспечивает работу с Windows Forms — интерфейсом, который упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обертки для существующего Win32 API в управляемом коде;
4. я знаком с данной средой разработки, что сократит время изучения возможностей среды.

3.1.2 Выбор СУБД

В качестве СУБД был выбран PostgreSQL. PostgreSQL[7] - это бесплатная СУБД. Эта СУБД позволяет многоверсионность данных для управления параллельными транзакциями. Также обеспечивает возможность создания пользовательских типов и программировать для них механизмы индексирования и возможность расширенный набор встроенных типов данных.

Этот выбор обусловлен тем, что я уже знаком в ранее изученном курсе и имеется опыт разработки проектов с этой СУБД, а также тем, что C# и PostgreSQL являются совместимыми технологиями.

3.2 Создание объектов БД

3.2.1 Создание таблиц

Ниже, на листинге 1 - 6, представлено создание все таблицы.

Таблица Users содержит информацию о пользователях программы.

Листинг 1 – Создание таблицы Users

```
1 create table Users
2 (
3     id serial not null primary key,
4     name varchar not null,
5     username varchar not null,
6     pass_word varchar not null,
7     role varchar
8 );
```

Таблица Subjects хранит информацию о списке предметах.

Листинг 2 – Создание таблицы Subjects

```
1 create table Subjects
2 (
3     id_subject serial not null primary key,
4     name varchar
5 );
```

Таблица Exams содержит информацию о тестах.

Листинг 3 – Создание таблицы Exams

```
1 create table Exams
2 (
3     id_exam serial not null primary key,
4     id_subject int null,
5     name_exam varchar not null,
6     code_exam varchar not null,
7     duration integer not null
8 );
```

Таблица Questions содержит информацию о вопросах.

Листинг 4 – Создание таблицы Questions

```
1 create table Questions
2 (
3     id_question serial not null primary key,
4     id_subject int null,
5     question varchar not null,
6     answer1 varchar,
7     answer2 varchar,
8     answer3 varchar,
9     answer4 varchar,
10    answer varchar
11 );
```

Таблица Exam_Question хранит информацию о связях между тестами и вопросами.

Листинг 5 – Создание таблицы Exam_Question

```
1 create table if not exists Exam_Question(
2     id_examQuestion serial not null primary key,
3     id_exam int not null,
4     id_question int not null
5 );
```

Таблица Results хранит информацию о результатах.

Листинг 6 – Создание таблицы Results

```
1 create table Results(
2     id_result serial not null primary key,
3     id_exam int not null,
4     id_user int not null,
5     startTime varchar null,
6     endTime varchar null,
7     score int null,
8     totalScore int null
9 );
```

3.2.2 Создание связей

Ниже, на листингах 7 - 10, представлено создание внешних ключей для обеспечения целостности связей между таблицами.

В таблице Exams на поле id_subject создается внешний ключ, который ссылается на поле id_subject (первичный ключ) таблицы Subjects.

Листинг 7 – Создание внешнего ключа в таблице Exams

```
1 FOREIGN KEY (id_subject) REFERENCES Subjects(id_subject)
```

В таблице Questions на поле id_subject создается внешний ключ, который ссылается на поле id_subject (первичный ключ) таблицы Subjects.

Листинг 8 – Создание внешнего ключа в таблице Questions

```
1 FOREIGN KEY (id_subject) REFERENCES Subjects(id_subject)
```

В таблице Results создаются 2 внешних ключа: id_user, который ссылается на поле id (первичный ключ) таблицы Users, и id_exam, который ссылается на поле id_exam (первичный ключ) таблицы Exams.

Листинг 9 – Создание внешнего ключа в таблице Results

```
1 FOREIGN KEY (id_user) REFERENCES Users(id),  
2 FOREIGN KEY (id_exam) REFERENCES Exams(id_exam)
```

В таблице Exam_Question создаются 2 внешних ключа: id_exam, который ссылается на поле id_exam (первичный ключ) таблицы Exams, и id_question, который ссылается на поле id_question (первичный ключ) таблицы Questions.

Листинг 10 – Создание внешнего ключа в таблице Exam_Question

```
1 FOREIGN KEY (id_exam) REFERENCES Exams(id_exam),  
2 FOREIGN KEY (id_question) REFERENCES Questions(id_question)
```

3.2.3 Ролевая модель на уровне БД

В соответствии с ролевой моделью, были созданы следующие роли.

1. studentuser — роль, связанная с студентом. Студент обладает правой на доступ к таблице Exams, чтобы увидеть список тестов.
2. teacheruser — роль, связанная с преподавателем. Преподаватель обладает правами на доступ к таблице Exams, Questions, Results и Exam_Question, чтобы управлять вопросами и тестами и увидеть все результаты.
3. adminuser — роль, связанная с администратором. Он обладает доступом к всем таблицам.

На листинге 11 представлен код выдачи прав по ролям.

Листинг 11 – Выдача прав ролям

```
1 CREATE ROLE studentuser WITH
2     LOGIN
3     NOSUPERUSER
4     NOCREATEDB
5     NOREPLICATION
6     CONNECTION LIMIT -1;
7 GRANT SELECT ON public.exams TO studentuser;
8
9 CREATE ROLE teacheruser WITH
10     LOGIN
11     NOSUPERUSER
12     NOCREATEDB
13     NOREPLICATION
14     PASSWORD 'Teacher@123 '
15     CONNECTION LIMIT -1;
16 GRANT SELECT ON public.exams TO teacheruser;
17 GRANT SELECT ON public.results TO teacheruser;
18 GRANT SELECT ON public.questions TO teacheruser;
19 GRANT SELECT ON public.exam_question TO teacheruser;
20
21 GRANT INSERT ON public.exams TO teacheruser;
22 GRANT INSERT ON public.questions TO teacheruser;
23 GRANT INSERT ON public.exam_question TO teacheruser;
24
25 GRANT DELETE ON public.exams TO teacheruser;
26 GRANT DELETE ON public.questions TO teacheruser;
27 GRANT DELETE ON public.exam_question TO teacheruser;
28
29 CREATE ROLE adminuser WITH
30     LOGIN
31     NOSUPERUSER
32     NOCREATEDB
33     NOREPLICATION
34     PASSWORD 'Admin@123 '
35     CONNECTION LIMIT -1;
36 GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA PUBLIC TO adminuser;
```

3.3 Интерфейс программы

Интерфейс программы разбит на страницы. Каждая страница предоставляет разный функционал. Ниже, на рисунках 3.1 - 3.9, продемонстрирован интерфейс разработанного приложения.

На рисунке 3.1 представлена страница авторизации пользователя.

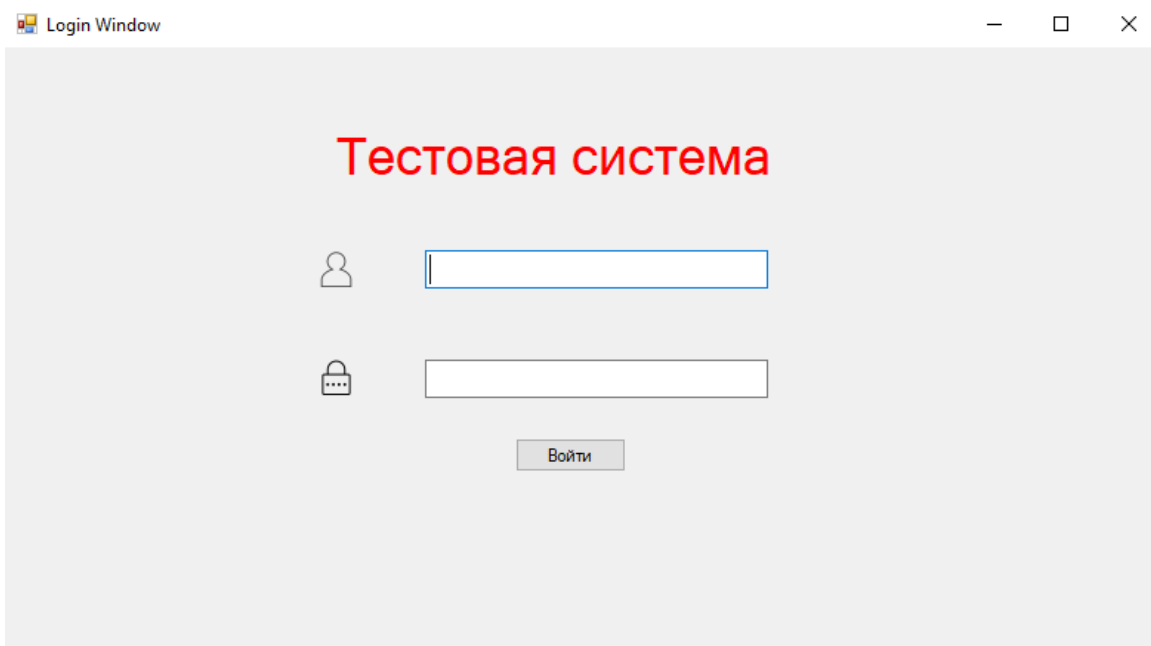


Рисунок 3.1 – Страница авторизации.

На рисунке 3.2 представлена главная страница студента. С этой страницы можно перейти к посмотру своих результатов, выбору теста для выполнения или изменению своего пароли.

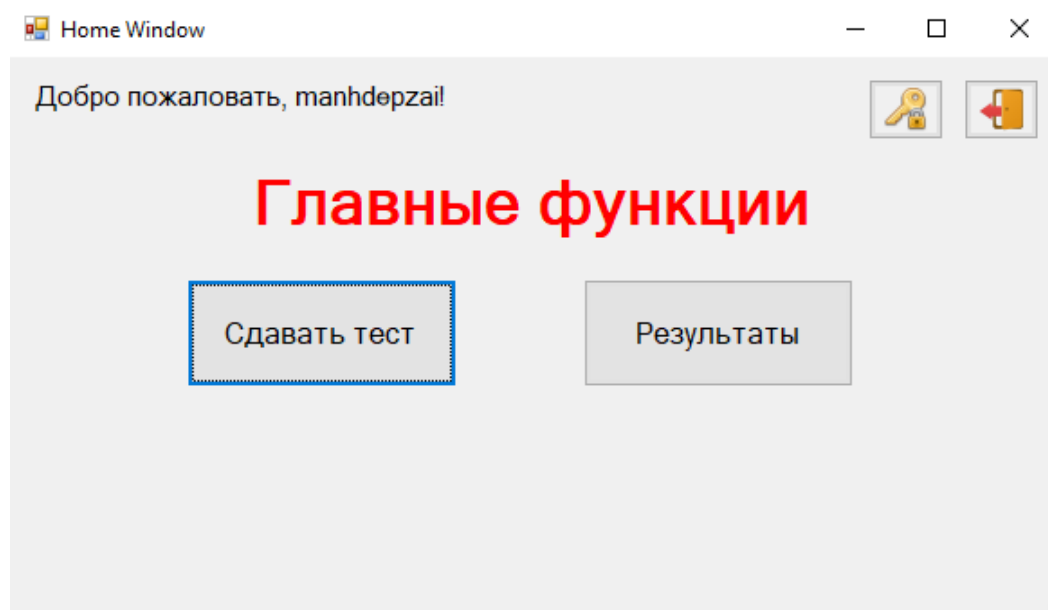


Рисунок 3.2 – Главная страница студента.

На рисунке 3.3 представлена страница выбора теста. Нужно выбрать тест, который студент хочет сдавать, и нажать на кнопку Старт.

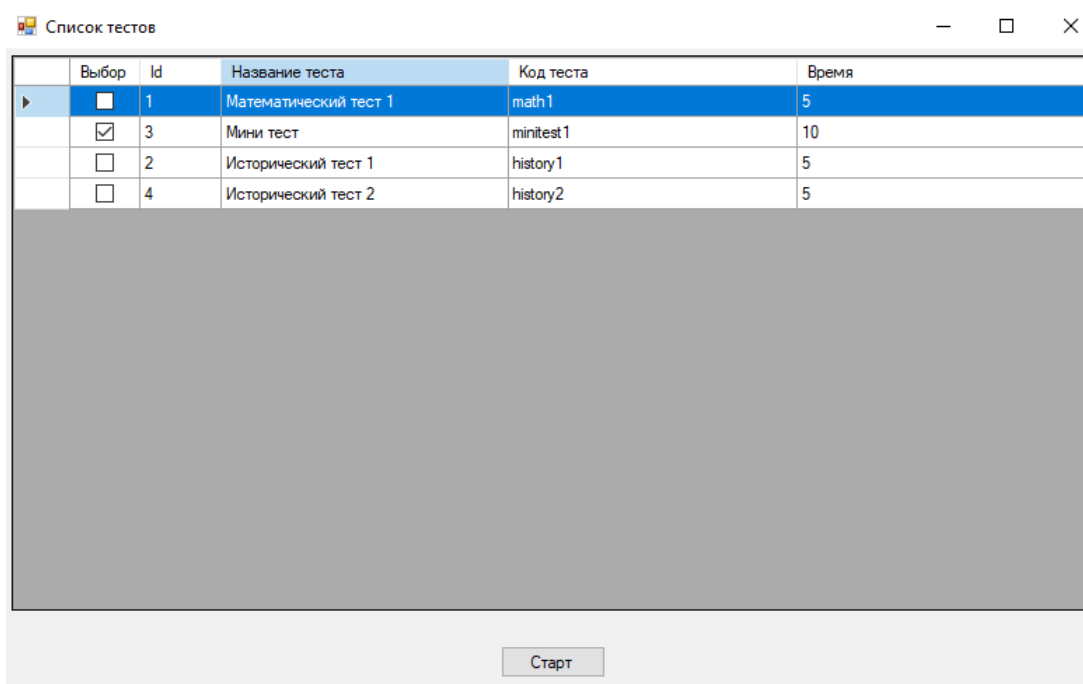


Рисунок 3.3 – Страница выбора теста.

На рисунке 3.4 представлена страница выполнения теста. Студент выбирает вопрос и ответ для этого вопроса. Если студент завершил тест, можно нажать на кнопку Завершить, чтобы закончить тест. Если время теста истекло, тест завершится автоматически.

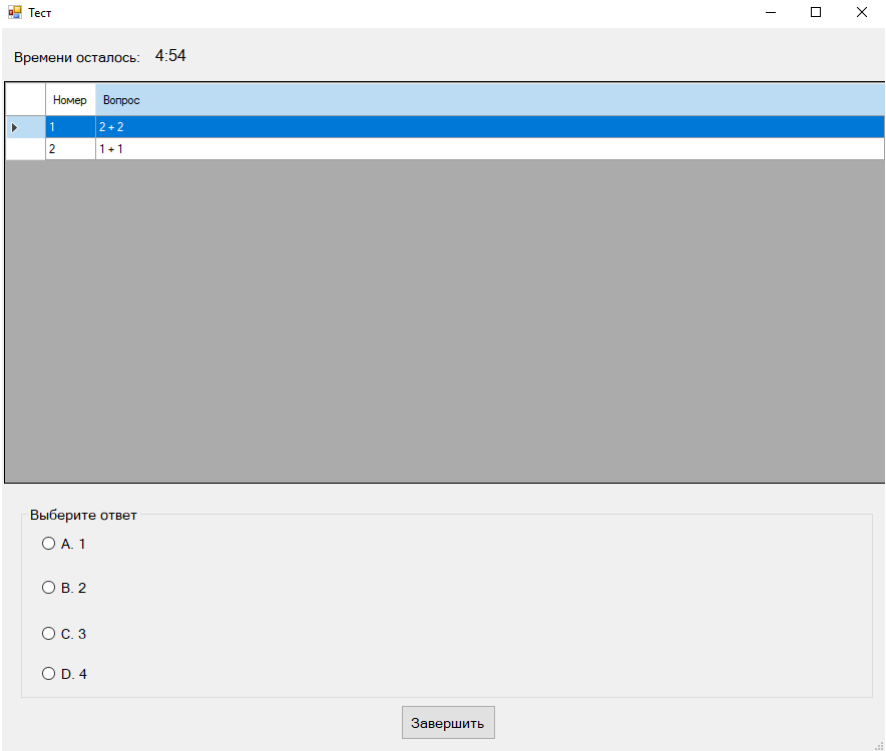


Рисунок 3.4 – Страница выполнения теста.

На рисунке 3.5 представлена страница просмотра результатов студента. На этой странице показаны все результаты студента.

Результаты						
Тест	Студент	Баллы	Макс. балл	Начальное время	Конечное время	
Исторический тест 1	Дао Тунг Куан	1	2	5/10/2023 2:34:37 AM	5/10/2023 2:34:47 AM	
Математический тест 1	Дао Тунг Куан	0	2	5/11/2023 2:30:25 PM	5/11/2023 2:30:34 PM	
Мини тест	Дао Тунг Куан	1	2	5/11/2023 2:39:50 PM	5/11/2023 2:40:07 PM	

Рисунок 3.5 – Страница просмотра результатов студента.

На рисунке 3.6 представлена главная страница преподавателя. С этой страницы можно перейти к посмотру всех результатов, изменению информации о вопросах и тестах или изменению своего пароли.

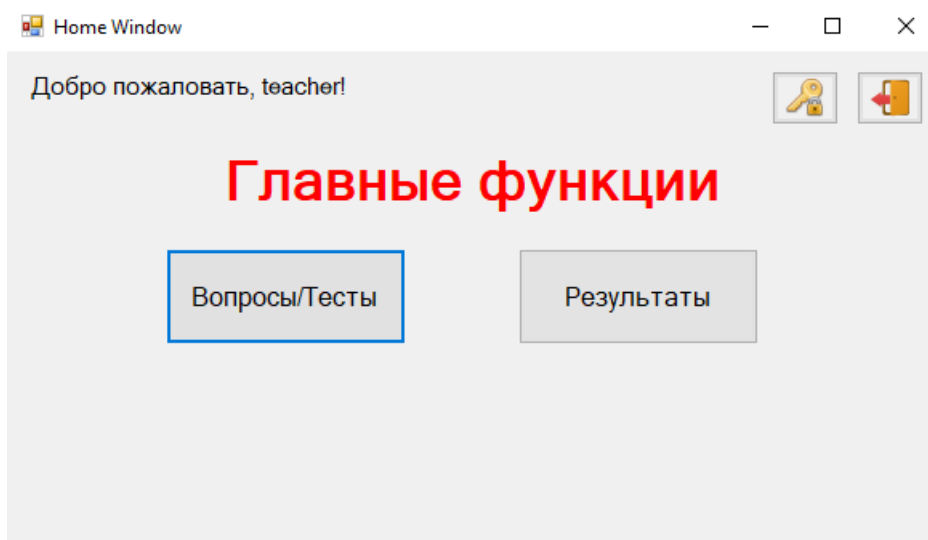


Рисунок 3.6 – Главная страница преподавателя.

На рисунке 3.7 представлена главная страница администратора. На этой странице можно перейти к управлению информацией о пользователях, вопросах, тестах и предметах.

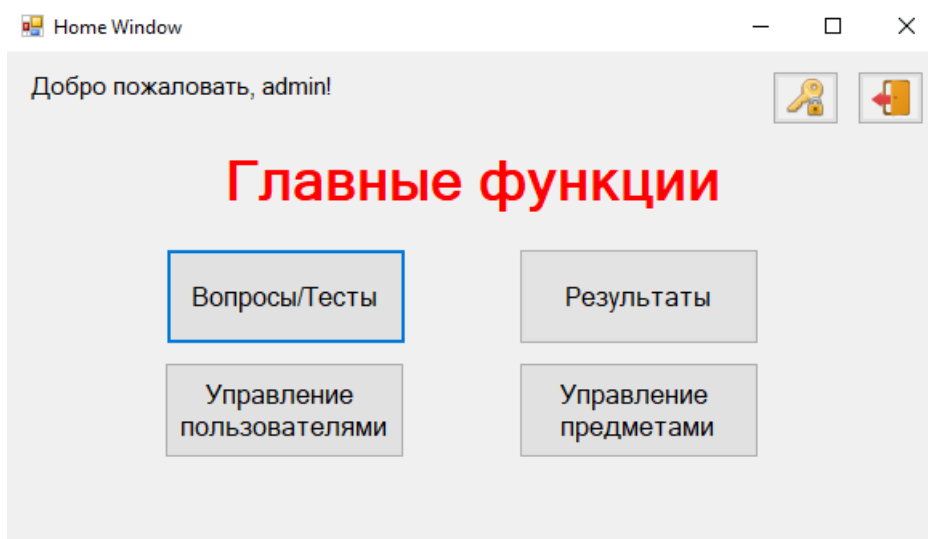


Рисунок 3.7 – Главная страница администратора.

На рисунке 3.8 представлена страница управления пользователях. На этой странице администраторы могут управлять информацией о пользователях.

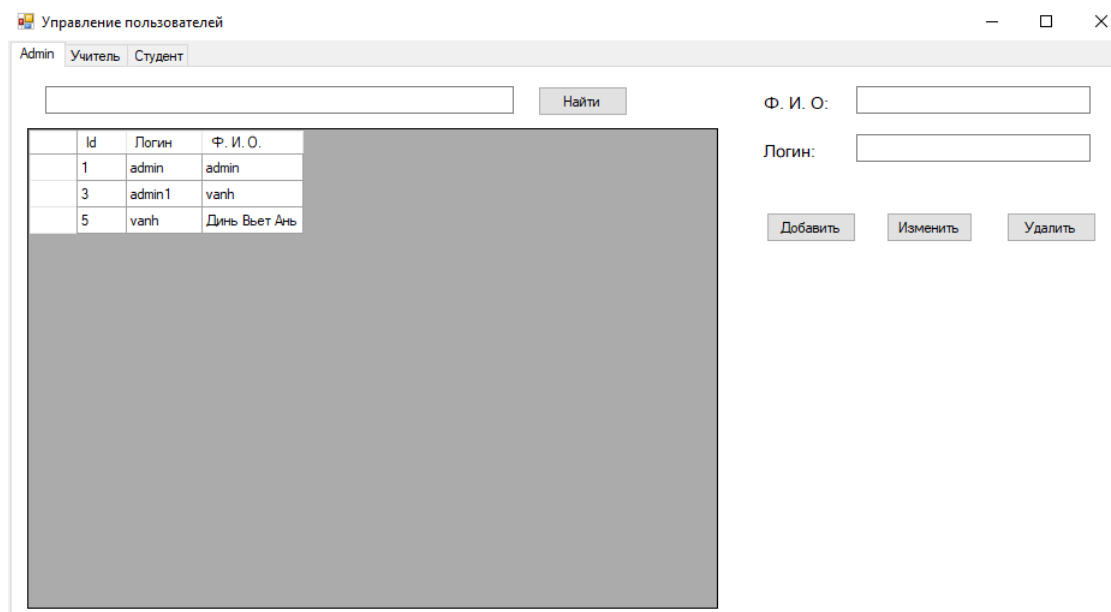


Рисунок 3.8 – Страница управления пользователях.

На рисунке 3.9 представлена главная страница изменения пароля. На этой странице пользователь может изменить свой пароль.

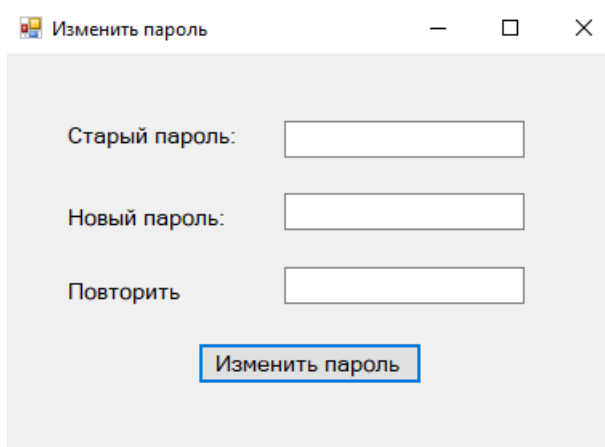


Рисунок 3.9 – Страница изменения пароля.

Вывод

В данном разделе были выбраны средства реализации: язык программирования C# и среда разработки VisualStudio и была выбрана СУБД PostgreSQL. Также были приведены детали реализации и был продемонстрирован полученный результат.

4 Исследовательская часть

4.1 Технические характеристики

Технические характеристики устройства, на котором выполнялось тестирование:

- операционная система: Window 10 Home Single Language;
- память: 8 Гб;
- процессор: 11th Gen Intel(R) Core(TM) i5-1135G7 2.42 ГГц;
- количество ядер процессора: 8.

Тестирование проводилось на ноутбуке, включенном в сеть электропитания. Во время тестирования ноутбук был нагружен только встроенными приложениями окружения, а также непосредственно системой тестирования. Сторонние приложения запущены не были.

4.2 Описание эксперимента

Цель эксперимента заключается в выявлении зависимости времени обработки запросов к таблицам базы данных от использования индексирования. Эксперимент проводился на таблице Questions, содержащих информацию о вопросах. В таблице Questions хранятся 2000 записей.

Время выполнения поиска по первичному ключу в таблице Questions показано на рисунке 4.1.

```
db_cw=# explain (analyze) select * from Questions where id_question = 612;
              QUERY PLAN
-----
Index Scan using questions_pkey on questions  (cost=0.28..8.29 rows=1 width=201) (actual time=0.278..0.280 rows=1 loops=1)
  Index Cond: (id_question = 612)
Planning Time: 1.942 ms
Execution Time: 0.655 ms
```

Рисунок 4.1 – Результат выполнения запроса без использования индексирования.

Результаты выполнения запроса после создания индекса question_id_index для первичного ключа таблицы Questions, представленного на рисунке 4.2, отображены на рисунке 4.3.

```
db_cw=# create index question_id_index on Questions using btree(id_question);
CREATE INDEX
```

Рисунок 4.2 – Создание индекса.

```
db_cw=# explain (analyze) select * from Questions where id_question = 612;
               QUERY PLAN
-----
Index Scan using question_id_index on questions  (cost=0.28..8.29 rows=1 width=201) (actual time=0.034..0.035 rows=1 loops=1)
  Index Cond: (id_question = 612)
Planning Time: 1.494 ms
Execution Time: 0.049 ms
```

Рисунок 4.3 – Результат выполнения запроса с использованием индексирования.

Можно сделать вывод о том, что создание индекса уменьшило время выполнения поиска в 13,4 раз.

Вывод

В результате эксперимента видно, что использование индексирования значительно ускоряет выполнение SELECT запросов. Поэтому использование индексирования в данной работе будет способствовать повышению производительности системы.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы были разработаны база данных для тестовой системы с вариантами ответов и приложение, которое обеспечивает интерфейс для взаимодействия с базой данных.

Была спроектирована предметная база данных, выделена ролевая модель как на уровне БД, так и на уровне приложения, были изучены возможности языка C# и получен опыт работы с СУБД PostgreSQL.

Также в рамках выполнения курсовой работы были выполнены все задачи:

- формализовать задачу и определить требуемый функционал;
- провести анализ существующих СУБД и выбрать наиболее подходящую;
- описать объекты, из которых состоит база данных, и связи между ними;
- спроектировать базу данных и реализовать программное обеспечение, позволяющее пользователям получать информации о тестовой системе;
- провести исследование времени выполнения запросов к БД с использованием индексов и без.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Что такое база данных | Oracle Россия и СНГ [Электронный ресурс]. Режим доступа: <https://www.oracle.com/ru/database/what-is-database/> (дата обращения: 25.05.2023).
2. Новиков Б. А. Горшкова Е. А. Графеева Н. Г. Основы технологий баз данных: учебное пособие. — М.: ДМК Пресс, 2020. с. 582.
3. Сатиндер Б. Г. Introduction Database Management System [Электронный ресурс]. Режим доступа: https://books.google.ru/books?id=NGlSs8YFs3EC&pg=PA77&redir_esc=y#v=onepage&q&f=false, (дата обращения 25.05.2023).
4. Бородина А. И. Реляционная модель данных [Электронный ресурс]. Режим доступа: http://www.bseu.by/it/tohod/lekcii2_3.htm, (дата обращения 25.05.2023).
5. Постреляционная модель базы данных [Электронный ресурс]. Режим доступа: http://db4.sbras.ru/elbib/data/show_page.phtml?20+1588, (дата обращения 25.05.2023).
6. Постреляционная модель: понятие, достоинства и недостатки. [Электронный ресурс]. Режим доступа: <https://studfile.net/preview/5407090/page:4/>, (дата обращения 25.05.2023).
7. Документация по PostgreSQL [Электронный ресурс]. Режим доступа: <https://postgrespro.ru/docs/postgresql/> (дата обращения: 12.04.2023).

ПРИЛОЖЕНИЕ А