|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ПРАВИТЕЛЬСТВО САНКТ-ПЕТЕРБУРГАКОМИТЕТ ПО НАУКЕ И ВЫСШЕЙ ШКОЛЕ Санкт-Петербургское государственное бюджетное профессиональное образовательное учреждение«ПЕТРОВСКИЙ КОЛЛЕДЖ» (СПб ГБПОУ «Петровский колледж») | | | | |
| Отделение информационных технологий | | | | |
|  | | | | |
| **Курсовой проект** ПМ.11 Разработка, администрирование и защита баз данных  МДК.11.01 Технология разработки и защиты баз данных | | | | |
|  | | | | |
| Проектирование и разработка базы данных для учета посещаемости и успеваемости обучающихся | | | | |
|  | Выполнил: | | | |
| Смирнов Олег Владимирович | | | |
| студент: | 52-02 | группы | |
|  |  |  | |
| Специальность: | 09.02.07 Информационные системы и программирование | | |
|  |  | | |
| Руководитель: М.М.Солодов | | | |
|  |  | | |
| Оценка |  | | |
|  | | |  |

Санкт-Петербург, 2024

**Содержание курсового проекта**

[Введение 3](#_Toc182260642)

[1 Постановка задачи на разработку информационной системы 4](#_Toc182260643)

[1.1 Описание предметной области 4](#_Toc182260644)

[1.2 Функции информационной системы 5](#_Toc182260645)

[2 Выбор средств разработки 6](#_Toc182260646)

[2.1 Выбор СУБД: 6](#_Toc182260647)

[2.2 Выбор различных средств для разработки клиентского приложения: 8](#_Toc182260648)

[2.3 Выбор языка программирования 10](#_Toc182260649)

[3 Проектирование базы данных 12](#_Toc182260650)

[3.1 Концептуальное проектирование базы данных 12](#_Toc182260651)

[3.2 Логическое проектирование базы данных 14](#_Toc182260652)

[4 Реализация приложения 18](#_Toc182260653)

[4.1 Реализация серверной части приложения 18](#_Toc182260654)

[4.2 Реализация клиентской части приложения 20](#_Toc182260655)

[ЗАКЛЮЧЕНИЕ 22](#_Toc182260656)

[Список использованных источников 23](#_Toc182260657)

[Приложение А 24](#_Toc182260658)

[Приложение Б 25](#_Toc182260659)

# Введение

В данной курсовой работе рассматривается проектирование и разработка базы данных для учета посещаемости и успеваемости обучающихся. Основное внимание уделяется автоматизации процессов сбора, хранения и анализа данных об успеваемости, а также посещаемости учащихся в образовательных учреждениях. Работа включает анализ функциональных требований и проектирование структуры базы данных.

В настоящее время многие образовательные учреждения сталкиваются с проблемами учета и анализа успеваемости и посещаемости обучающихся. Ведется работа с громоздкими таблицами, данные могут быть разрознены или не синхронизированы. Это приводит к сложностям в оценке общей картины успеваемости и посещаемости, что затрудняет управление образовательным процессом.

При разработке приложения следует сделать акцент на удобство для пользователя, сделать интерфейс как можно понятнее. Информационная система должна способствовать повышению качества и эффективности учета, а также минимизировать ручной труд за счет автоматизации.

Целями курсовой работы являются разработка базы данных и клиентского приложения, разработка функциональных требований к информационной системе, а также создание эффективного инструмента, который позволит образовательным учреждениям упрощать учет данных и анализировать динамику успеваемости.

Задачи курсовой работы:

− проектирование и разработка базы данных для учета посещаемости и успеваемости обучающихся;

− обеспечение гибкости и расширяемости структуры данных;

− создание удобного пользовательского интерфейса для работы с системой;

− автоматизация процессов сбора и анализа данных;

− повышение точности и доступности данных для анализа.

1. **Постановка задачи на разработку информационной системы**
   1. **Описание предметной области**

Для ректората ВУЗа требуется разработать информационную систему, позволяющую обрабатывать данные о студентах, обучающихся в данном ВУЗе, и их успеваемости за текущий семестр, представленные в определенном формате.

Информация о студентах представлена следующими данными: номер студенческого билета; фамилия, имя и отчество; дата рождения; пол; факультет; специальность; курс; группа; количество экзаменов; оценки, полученные на экзамене.

Предусмотреть возможность добавления, изменения и удаления записей в справочниках.

На основании представленных в системе данных необходимо подготовить формы определенного вида.

* 1. **Ф****ункции информационной системы**

1. По всем специальностям каждого факультета выдать список студентов, имеющих хотя бы одну двойку. Отчет составить с указанием количества двоек для каждого студента. Результат отсортировать по курсу и группе и ФИО студента. Вид выходного документа (см. Таблица 1)

Таблица 1 ̶ Список студентов с двойками

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <Факультет> | | <Специальность> | | |
| № п/п | Курс | Группа | ФИО | Количество двоек |
| … | … | … | … | … |
|  | | | | |

1. Для каждого факультета определить список студентов, представленных на стипендию (студент представляется на стипендию в том случае, если у него нет ни одной оценки меньше 4). Отдельно определить список студентов, представленных на повышенную стипендию. Вид выходного документа (см. Таблица 2)

Таблица 2 ̶ Список студентов представленных на стипендию

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <Факультет> | |  | | |
| № п/п | Курс | | Группа | ФИО |
| … | … | | … | … |
|  | | | | |

1. **Выбор средств разработки**

Перед началом создания проекта необходимо сравнить различные СУБД, интегрированные среды разработки и языки программирования. Это необходимо для того, чтобы работа была эффективной, удобной, оптимальной и результативной.

Выбор правильных инструментов помогает сэкономить время и ресурсы, обеспечит высокое качество проекта и упрощает работу. Выбор должен основываться на требованиях проекта.

* 1. **Выбор СУБД:**
     1. MySQL

MySQL – это популярная открытая система управления базами данных (СУБД), основанная на языке SQL. Она широко используется для разработки веб-приложений и предоставляет высокую производительность, надежность и гибкость. MySQL поддерживает большое количество платформ и активно используется в различных проектах благодаря своей простоте и доступности.

Плюсы:

* бесплатная и открытая СУБД;
* высокая производительность и надежность;
* широко поддерживается различными библиотеками и фреймворками;
* простой в использовании интерфейс, особенно с инструментами типа phpMyAdmin.

Минусы:

* меньше поддерживает сложные запросы по сравнению с некоторыми другими СУБД;
* ограниченные функции транзакций и поддержки некоторых типов данных по сравнению с PostgreSQL.
  + 1. PostgreSQL

PostgreSQL – это мощная открытая СУБД, известная своей расширяемостью и поддержкой сложных запросов. Она поддерживает выполнение транзакций, управление большими объемами данных и множество расширенных возможностей, таких как триггеры, хранимые процедуры и пользовательские типы данных. PostgreSQL часто выбирают для задач, требующих высокой степени надежности и согласованности данных.

Плюсы:

* очень мощная и расширяемая СУБД, поддерживающая сложные запросы и методы работы с данными;
* поддержка JSON и других нострадинских типов данных;
* Высокая степень надежности и соответствие стандартам SQL.

Минусы:

* может быть сложнее в настройке и использовании для новичков;
* может требовать больше ресурсов по сравнению с другими СУБД, например, MySQL.
  + 1. Microsoft SQL Server

Microsoft SQL Server – это реляционная СУБД, разработанная Microsoft, которая предоставляет надежные инструменты для управления данными и аналитики. SQL Server поддерживает различные стандарты безопасности, масштабируемость и интеграцию с другими продуктами Microsoft, такими как Azure, что делает его популярным выбором для предприятий.

Плюсы:

* широкие возможности аналитики и инструменты для бизнес-анализа;
* хорошая интеграция с другими продуктами Microsoft и поддержка транзакций;
* расширенные функции безопасности и резервного копирования.

Минусы:

* высокая стоимость лицензирования для коммерческого использования;
* может потребовать значительных ресурсов для настройки и обслуживания.
  + 1. Oracle Database

Oracle Database – это объектно-реляционная СУБД, разработанная компанией Oracle. Она известна своей высоким уровнем надежности и производительности, а также многочисленными инструментами для управления данными. Oracle поддерживает большой объем данных и сложные транзакции, что делает её идеальной для крупных предприятий и критически важных приложений.

Плюсы:

* высокая производительность и масштабируемость для больших предприятий;
* низкий уровень отказов и надежная поддержка транзакций;
* широкие возможности для управления данными и аналитики.

Минусы:

* очень высокая стоимость лицензии и операций;
* сложность в управлении и настройке системы.

Конечный выбор СУБД пал на Microsoft SQL Server. Для разработки клиентского приложения будет использована программа Visual Studio, что тоже является инструментом Microsoft, а это значит, что интеграция между ними будет простой и удобной.

* 1. **Выбор различных средств для разработки клиентского приложения:**
     1. Visual Studio

Visual Studio – это интегрированная среда разработки (IDE) от Microsoft, предназначенная для создания приложений на различных языках программирования, таких как C#, VB.NET, F# и других. Visual Studio предлагает широкий набор инструментов для разработки, отладки и тестирования, что значительно ускоряет процесс разработки программного обеспечения.

Плюсы:

* поддержка множества языков программирования (C#, VB.NET, F# и др.);
* отличные инструменты отладки и профилирования;
* широкая экосистема плагинов и расширений;
* интеграция с Azure и другими сервисами Microsoft.

Минусы:

* высокая стоимость для профессиональных версий;
* требовательность к ресурсам системы;
* время на изучение интерфейса и инструментов для новичков.
  + 1. IntelliJ IDEA

IntelliJ IDEA – это мощная интегрированная среда разработки (IDE) для Java и других языков программирования, разработанная компанией JetBrains. Она предоставляет богатые возможности для автозавершения кода, рефакторинга и интеграции с системами контроля версий, что делает работу разработчиков более эффективной и продуктивной.

Плюсы:

* отличная поддержка Java и других языков, таких как Kotlin и Scala;
* интуитивно понятный интерфейс и мощные инструменты для кода;
* интеграция с системами контроля версий и сборки.

Минусы:

* платная версия может быть дорогой для небольших команд или индивидуальных разработчиков;
* некоторые функции могут требовать времени на обучение.
  + 1. Xcode

Xcode – это интегрированная среда разработки от Apple, предназначенная для создания приложений под операционные системы iOS и macOS. Xcode предоставляет инструменты для проектирования интерфейсов, написания кода, а также отладки и тестирования приложений.

Плюсы:

* идеальная среда для разработки приложений под iOS и macOS;
* отличные инструменты интерфейсного дизайна и симуляция;
* полная интеграция с экосистемой Apple.

Минусы:

* работает только на устройствах Apple;
* высокие требования к системе и иногда может быть медленной;
* ограниченные возможности для кросс-платформной разработки.

Visual Studio, из выше перечисленных, лучшее средство для разработки клиентского приложения. Программа поддерживает все этапы разработки, включая проектирование, откладку и тестирование, имеет простой интерфейс и поддерживает различные языки программирования, что позволит создать приложение на C#.

* 1. **Выбор языка программирования**
     1. Python

Python – это высокоуровневый язык программирования, известный своей простотой и читабельностью кода. Он используется в различных областях, включая веб-разработку, анализ данных, машинное обучение и автоматизацию. Python имеет большое сообщество и множество библиотек, что упрощает решение самых разных задач.

Плюсы:

* простота и читаемость: Чистый и понятный синтаксис, что делает язык идеальным для новичков;
* широкая применимость: Успешно используется в веб-разработке, научных вычислениях, машинном обучении и автоматизации;
* богатая библиотека: Множество готовых библиотек для различных задач (Django, Flask для веб-разработки).

Минусы:

* производительность: Интерпретируемый язык, может быть медленнее, чем компилируемые языки;
* поддержка мобильных приложений: Менее популярный выбор для мобильной разработки по сравнению с другими языками.
  + 1. C#

C# – это объектно-ориентированный язык программирования, разработанный Microsoft. Он широко используется для создания приложений для платформы .NET, включая веб-приложения, мобильные приложения и игры. C# отличается выразительным синтаксисом и мощной системой типов.

Плюсы:

* совместимость с Windows: Отлично подходит для разработки приложений под Windows и игр на Unity;
* совершенный инструмент: Выдающаяся интеграция с Visual Studio и мощные средства отладки;
* объектно-ориентированный подход: Высокая производительность и безопасность типов.

Минусы:

* платформенная зависимость: Хотя .NET Core делает язык кросс-платформенным, большинство библиотек изначально предназначены для Windows;
* небольшая популярность за пределами взаимодействий с Microsoft.

C# предпочтителен для разработки приложения. Этот язык хорошо совместим с .Net, что позволяет пользоваться огромными возможностями библиотеки, инструментами и фреймворками. Кроме того, существует множество учебных материалов, что упростит процесс разработки и позволит найти решения на многие задачи.

1. **Проектирование базы данных**
   1. **Концептуальное проектирование базы данных**
      1. Связь Факультет – Специальность

Один факультет может иметь несколько специальностей (см. Рисунок 1)

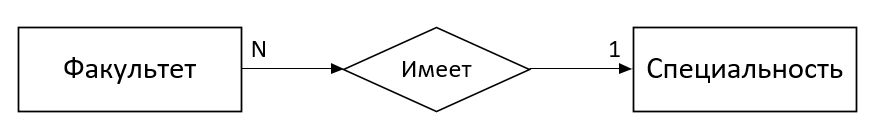


Рисунок 1 ̶ Связь Факультет – Специальность

* + 1. Связь Специальность – Группа

Несколько групп может иметь одну специальность (см. Рисунок 2)

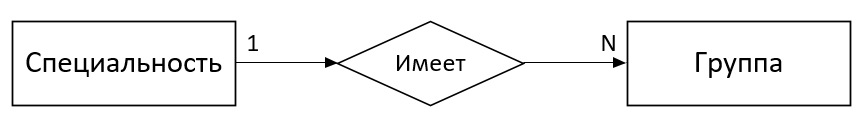


Рисунок 2 ̶ Связь Специальность – Группа

* + 1. Связь Группа – Студент

Одна группа имеет несколько студентов (см. Рисунок 3)



Рисунок 3 ̶ Связь Группа – Студент

* + 1. Связь Студент – Экзамен

Один студент может сдавать несколько экзаменов (см. Рисунок 4)

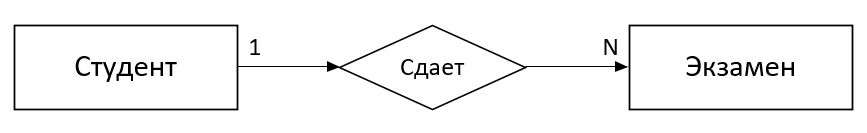


Рисунок 4 ̶ Связь Студент – Экзамен

* + 1. Связь Преподаватель – Предмет

Один преподаватель может вести несколько (см. Рисунок 5)



Рисунок 5 ̶ Связь Преподаватель – Предмет

* + 1. Связь Предмет – Экзамен

Один предмет может иметь несколько экзаменов (см. Рисунок 6)

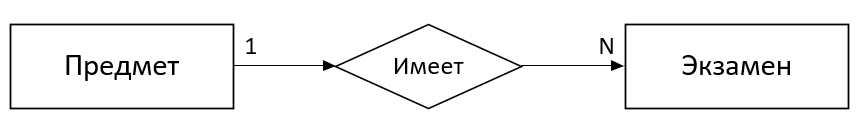


Рисунок 6 ̶ Связь Предмет – Экзамен

* + 1. Связь Экзамен – Тип экзамена

Один экзамен может иметь несколько типов (см. Рисунок 7)



Рисунок 7 ̶ Экзамен – Тип экзамена

* + 1. ER – Диаграмма (см. Рисунок 8)

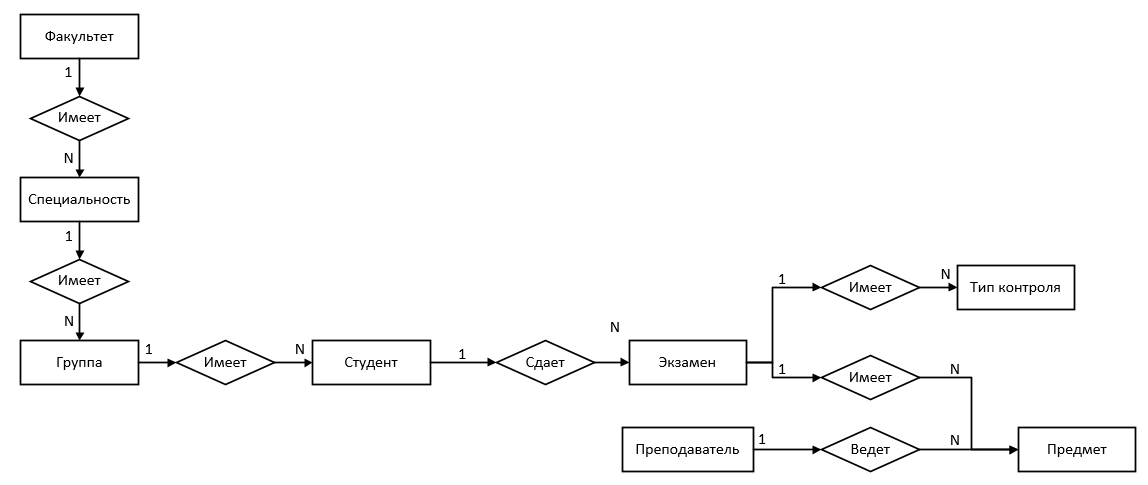


Рисунок 8 ̶ ER – Диаграмма

* 1. **Логическое проектирование базы данных**
     1. Students (Студенты)

Хранит информацию о студентах ВУЗа, таких как номер студенческого билета, фамилия, имя, отчество, пол, дата рождения, номер группы (см. Таблица 3)

Таблица 3 ̶ Таблица Студенты

|  |  |  |
| --- | --- | --- |
| **Столбец** | **Тип данных** | **Описание** |
| id\_student | INT | Идентификатор студента (первичный ключ) |
| student\_id\_number | VARCHAR(8) | Номер студенческого билета |
| full\_name | VARCHAR(255) | Фамилия, Имя, Отчество студента |
| sex | VARCHAR(3) | Пол студента |
| id\_group | INT | Идентификатор группы (внешний ключ к Groups) |
| date\_birth | DATE | Дата рождения |

* + 1. Entrance (Вход)

Хранит информацию о входах пользователей в систему, таких как логин и пароль (см. Таблица 4)

Таблица 4 ̶ Таблица с логином и паролем клиентов

|  |  |  |
| --- | --- | --- |
| **Столбец** | **Тип данных** | **Описание** |
| id\_entrance | INT | Идентификатор входа (первичный ключ) |
| id\_role | INT | Идентификатор роли (внешний ключ к Roles) |
| login | VARCHAR(50) | Логин для входа |
| password | VARCHAR(255) | Пароль для входа |
| date\_last\_login | DATE | Дата последнего входа в систему |
| id\_entity | INT | Идентификатор сущности |

* + 1. Roles (Роли)

Определяет роли, которые могут иметь пользователи клиентского приложения (студент, преподаватель, администратор) (см. Таблица 5)

Таблица 5 ̶ Таблица с ролями пользователя

|  |  |  |
| --- | --- | --- |
| **Столбец** | **Тип данных** | **Описание** |
| id\_role | INT | Идентификатор роли (первичный ключ) |
| role\_name | VARCHAR(50) | Название роли |

* + 1. Teachers (Преподаватели)

Хранит информацию о преподавателях ВУЗа, таких как имя, телефон, электронная почта (см. Таблица 6)

Таблица 6 ̶ Таблица с данными преподавателя

|  |  |  |
| --- | --- | --- |
| **Столбец** | **Тип данных** | **Описание** |
| id\_teacher | INT | Идентификатор преподавателя (первичный ключ) |
| full\_name | VARCHAR(255) | Фамилия, Имя, Отчество преподавателя |
| phone | VARCHAR(20) | Номер телефона преподавателя |
| email | VARCHAR(255) | Электронная почта преподавателя |

* + 1. Subjects (Предметы)

Хранит информацию об учебной дисциплине, такую как название предмета и преподавателя, который ведет этот предмет (см. Таблица 7)

Таблица 7 ̶ Таблица с предметами

|  |  |  |
| --- | --- | --- |
| **Столбец** | **Тип данных** | **Описание** |
| id\_subject | INT | Идентификатор предмета (первичный ключ) |
| id\_teacher | INT | Идентификатор преподавателя (внешний ключ к Teachers) |
| subject\_name | VARCHAR(255) | Название предмета |

* + 1. Faculties (Факультеты)

Хранит информацию о факультетах в ВУЗе, содержит в себе название (см. Таблица 8)

Таблица 8 ̶ Таблица с факультетами

|  |  |  |
| --- | --- | --- |
| **Столбец** | **Тип данных** | **Описание** |
| id\_faculty | INT | Идентификатор факультета (первичный ключ) |
| faculty\_name | VARCHAR(50) | Название факультета |

* + 1. Specialties (Специальности)

Хранит информацию о специальностях в ВУЗе, содержит такую информацию как название специальности, факультет специальности (см. Таблица 9)

Таблица 9 ̶ Таблица со специальностями

|  |  |  |
| --- | --- | --- |
| **Столбец** | **Тип данных** | **Описание** |
| id\_specialty | INT | Идентификатор специальности (первичный ключ) |
| id\_faculty | INT | Идентификатор факультета (внешний ключ к Faculties) |
| specialty\_name | VARCHAR(50) | Название специальности |

* + 1. Groups (Группы)

Хранит информацию о группе, включая номер группы, специальность, курс (см. Таблица 10)

Таблица 10 ̶ Таблица с группами

|  |  |  |
| --- | --- | --- |
| **Столбец** | **Тип данных** | **Описание** |
| id\_group | INT | Идентификатор группы (первичный ключ) |
| group\_number | VARCHAR(5) | Номер группы |
| id\_specialty | INT | Идентификатор специальности (внешний ключ к Specialties) |
| course | INT | Курс, на котором обучается группа |

* + 1. Exams (Экзамены)

Хранит информацию об экзаменах студента, содержит такую информацию как дата экзамена, студент, предмет, тип контроля, оценка (см. Таблица 11)

Таблица 11 ̶ Таблица с экзаменами

|  |  |  |
| --- | --- | --- |
| **Столбец** | **Тип данных** | **Описание** |
| id\_exam | INT | Идентификатор экзамена (первичный ключ) |
| id\_student | INT | Идентификатор студента (внешний ключ к Students) |
| id\_subject | INT | Идентификатор предмета (внешний ключ к Subjects) |
| id\_control\_type | INT | Идентификатор типа контроля (внешний ключ к Control\_Types) |
| date | DATE | Дата экзамена |
| grade | DATE | Оценка за экзамен |

* + 1. Control\_Types (Типы контроля)

Определяет типы контроля, применяемых для экзамена (например, "Экзамен", "Дифференцированный зачет") (см. Таблица 12)

Таблица 12 ̶ Таблица с типами контроля

|  |  |  |
| --- | --- | --- |
| **Столбец** | **Тип данных** | **Описание** |
| id\_control\_type | INT | Идентификатор типа контроля (первичный ключ) |
| control\_type\_name | VARCHAR(50) | Название типа контроля |

* + 1. ERD – Диаграмма со связями между таблицами. (см. Рисунок 9)

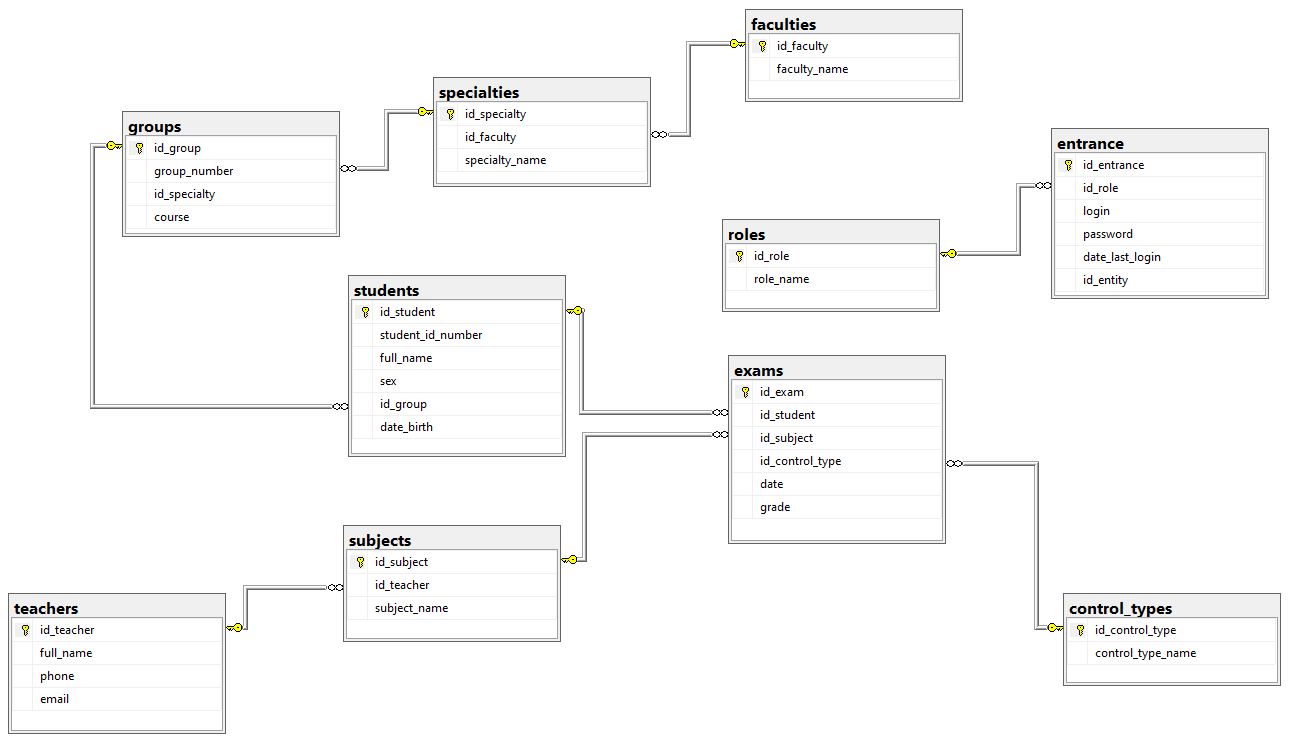
****

Рисунок 9 ̶ ERD-Диаграмма

1. **Реализация приложения**
   1. **Реализация серверной части приложения**

Скрип базы данных, созданной для клиентского приложения (см. Листинг 1, Приложение А)

Было создано 17 таблиц, 12 с данными и 5 справочников.

* + 1. Students (Студенты)

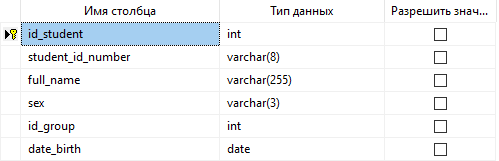


Рисунок 10 – Таблица Студенты

* + 1. Entrance (Вход)

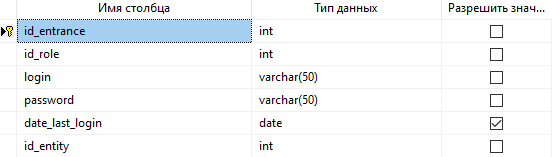


Рисунок 11 – Таблица Вход

* + 1. Roles (Роли)

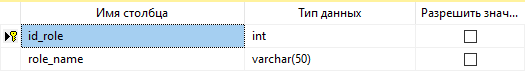


Рисунок 12 – Таблица Роли

* + 1. Teachers (Преподаватели)

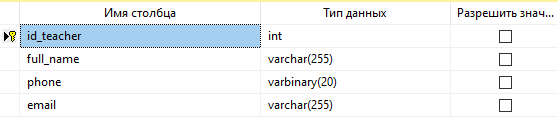


Рисунок 13 – Таблица Преподаватели

* + 1. Subjects (Предметы)

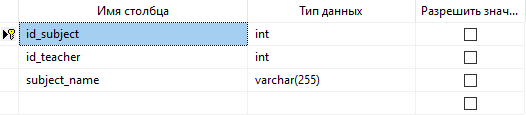


Рисунок 14 – Таблица Предметы

* + 1. Faculties (Факультеты)

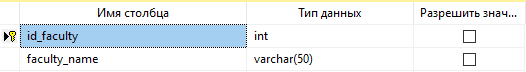


Рисунок 15 – Таблица Факультеты

* + 1. Specialties (Специальности)

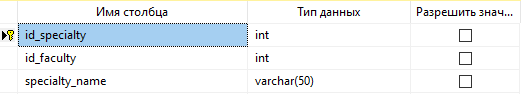


Рисунок 16 – Таблица Специальности

* + 1. Groups (Группы)

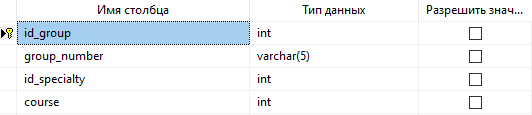


Рисунок 17 – Таблица Группы

* + 1. Exams (Экзамены)

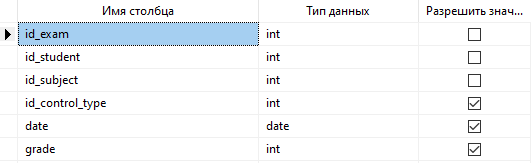


Рисунок 18 – Таблица Экзамены

* + 1. Control\_Types (Типы контроля)

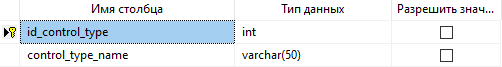


Рисунок 19 – Таблица Типы контроля

* 1. **Реализация клиентской части приложения**

В клиентской части приложения должно быть:

* авторизация под разными ролями: студент (возможность просматривать данные об экзаменах в таблице), преподаватель (возможность просматривать и добавлять информацию о всех экзаменах, также просматривать информацию о студентах), админ (возможность редактировать, добавлять и удалять).
* возможность добавления, изменения и удаления записей в таблицах;
* выполненные запросы, описанные в задании (см. пункт 1.2);
* возможность выгрузки отчетов по выполнению запросов;
  + 1. **Авторизация под разными ролями: студент (возможность просматривать данные об экзаменах в таблице), преподаватель (возможность просматривать и добавлять информацию об экзаменах, также просматривать информацию о студентах), админ (возможность редактировать, добавлять и удалять)**

Нужно создать три разные формы. Первая форма – это меню, где написана минимальная информация о салоне, вторая нужна для входа в аккаунт, а третья для создания аккаунта, если он отсутствует.

* + 1. **Возможность добавления, изменения и удаления записей в таблице;**

Для пользователя с ролью админ добавить возможность добавления, изменения и удаления записей. Удаление сделать на одной форме, а для изменения и добавления создать отдельную (см. Рисунок 34, см. Рисунок 35) (см. Приложение А, см. Листинг А.5)

* + 1. **Выполненные запросы, описанные в задании (см. пункт 1.2);**

Добавим на 5 форму кнопку, которая будет открывать 7 форму для вывода отчета (см. Рисунок 36) (см. Рисунок 37) (см. Листинг 3)

* + 1. **Возможность выгрузки отчетов по выполнению запросов;**

На форму с отчетами нужно добавить кнопки для выгрузки отчетов (см. Рисунок 41) (см. Приложение А, см. Листинг А.8) (см. Приложение Б, см. Рисунки Б.10-12)

# ЗАКЛЮЧЕНИЕ

# Список использованных источников

1. "Базы данных: Создание, проектирование и реализация" / К. Дейт. - 9-е изд. - Москва: "Вильямс", 2020. - 1280 с.
2. "Базы данных: учебник по SQL"/ Khan Academy, 2024. - https://www.khanacademy.org/computing/computer-programming/sql
3. "Документация PostgreSQL"/ PostgreSQL, 2024. - https://www.postgresql.org/docs/
4. "Разработка клиентских приложений для работы с базами данных"/ М. В. Сидоров. - Журнал "Программирование", 2021, № 4, с. 55-62.
5. "SQL. Самоучитель"/ Д. Гроссман. - СПб.: Питер, 2022. - 304 с.
6. "Современные подходы к проектированию баз данных"/ А. С. Иванов, С. В. Петров. - Научно-технический журнал "Информационные технологии", 2023, № 1, с. 25-32.
7. "Введение в SQL"/ W3Schools, 2024. - https://www.w3schools.com/sql/

# Приложение А

**Листинг программы**

**А.1 Листинг скрипта базы данных и таблиц**

Листинг 1 – Создание базы данных и таблиц

USE [102204677]

**А.2 Листинг входа в аккаунт**

Листинг А.2 – Листинг входа в аккаунт

public partial class AutoLogin : Form

**А.3 Листинг регистрации**

Листинг А.3 – Листинг регистрации

public partial class Registration : Form

**А.4 Листинг для вывода данных**

Листинг А.4 – Листинг вывода данных

string connectionString = @"Data Source=LAPTOP-

**А.5 Листинг удаления записей**

Листинг А.5 – Листинг удаления записей

public partial class foradmin : Form

**А.6 Листинг изменения и добавления данных**

Листинг А.6 – Листинг изменения и добавления данных

public partial class AddEdit : Form

{

**А.7 Листинг**

Листинг А.7 – Листинг

private SqlConnection connection;

# Приложение Б

**проверки программы**