**Руководство программиста**

**информационная система для образовательного учреждения**

**«Личный кабинет студента»**

**СОДЕРЖАНИЕ**

[1 Введение 2](#_Toc181481376)

[1.1 Наименование системы 2](#_Toc181481377)

[1.2 Область применения 2](#_Toc181481378)

[2 Требования к окружению 3](#_Toc181481379)

[3 Настройка окружения разработки 4](#_Toc181481380)

[4 Установка и запуск приложения 5](#_Toc181481381)

[5 Внутреняя структура системы 6](#_Toc181481382)

[5.1 Используемый технологический стек 6](#_Toc181481383)

[5.2 Функции и их описание 6](#_Toc181481384)

[5.3 Структура данных 8](#_Toc181481385)

# Введение

## Наименование системы

Информационная система «Личный кабинет студента» (далее ИС).).

## Область применения

ИС предназначена для управления учебным процессом. Она предоставляет интерфейсы для студентов, преподавателей и администраторов, позволяя организовать взаимодействие между ними, включая выполнение заданий, управление профилем, а также создание и управление опросами и информацией.

# Требования к окружению

Минимальные требования к окружению:

1. Операционная система: Windows 7 и выше или любая ОС, поддерживающая Python;
2. Оперативная память (ОЗУ): 2 ГБ;
3. Жесткий диск: 2 ГБ;
4. Процессор: Intel Core i3 или аналогичный;
5. Разрешение экрана**:** 1024x768 пикселей и выше.
6. Сетевое подключение: стабильное соединение для работы с базой данных.
7. Python: Версия 3.8 и выше;
8. Библиотеки Python:
   * PyQt5 для создания графического интерфейса;
   * «mysql-connector-python» для работы с базой данных MySQL.
9. СУБД: MySQL версии 8.0 и выше.

# **Настройка окружения разработки**

Для настройки окружения выполните следующие действия:

1. Установите [Python](https://www.python.org/downloads/) версии 3.8 или выше;
2. Установите PyCharm или другую IDE для Python;
3. Установите необходимые библиотеки с помощью команды «pip install PyQt5 mysql-connector-python»;
4. Настройте соединение с базой данных. Параметры конфигурации представлены на рисунке 1.

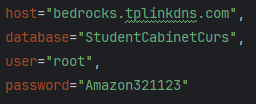


Рисунок - конфигурация БД

# Установка и запуск приложения

Для установки и запуска приложения необходимо выполнить следующие действия:

1. Склонируйте репозиторий:

* «git clone <https://github.com/AmazonGl/StudentCabinet>»

1. Перейдите в директорию:

* «cd <название\_директории>»

1. Запустите программу:

* «python StudentCabinet.py»

# Внутреняя структура системы

## Используемый технологический стек

Для разработки системы «Личный кабинет студента» использовался следующий технологический стек:

1. Язык программирования: Python — основной язык разработки системы. Он обеспечивает гибкость, читаемость кода и широкий выбор библиотек, что делает его удобным для работы с базами данных, построения интерфейсов и реализации бизнес-логики.
2. Фреймворк для пользовательского интерфейса: PyQt5 — библиотека для создания графических интерфейсов пользователя (GUI). Она предоставляет набор компонентов, таких как окна, кнопки, таблицы и формы, что позволяет реализовать удобные, современные и интуитивно понятные интерфейсы для всех пользователей системы.
3. База данных: MySQL — популярная реляционная СУБД, широко используемая для разработки приложений благодаря высокой производительности, надежности и простоте использования. MySQL предоставляет мощные инструменты для работы с большими объёмами данных, связанных с ролями пользователей, заданиями, результатами и другой информацией, что делает её идеальной для реализации системы «Личный кабинет студента».
4. Библиотека для работы с базой данных: mysql.connector — библиотека для Python, обеспечивающая простое и удобное подключение к базе данных MySQL. Она позволяет выполнять операции добавления, удаления, обновления и выборки данных с использованием SQL-запросов.

## Функции и их описание

Программа «Личный кабинет студента» предоставляет следующие функции, организованные по ролям пользователей:

Функции студента:

1. Просмотр информации и уведомлений:

* Студент может просматривать информацию, опубликованную администратором или преподавателем (новости, задания, результаты);
* Цель функции — обеспечить студентов актуальной информацией.

1. Редактирование профиля:

* Студент может редактировать свои личные данные (например, контактную информацию;
* Это позволяет поддерживать актуальность профиля.

1. Отправка выполненных заданий:

* Студент может прикреплять файлы с выполненными заданиями для оценки преподавателем;
* Функция автоматизирует процесс сдачи заданий.

1. Прохождение опросов:

* Студент может участвовать в опросах, созданных администратором или преподавателем;
* Это помогает собирать обратную связь.

Функции преподавателя:

Преподаватель имеет доступ ко всем функциям студента и следующими возможностями:

1. Создание и редактирование заданий:

* Преподаватель может добавлять новые задания для студентов, задавать сроки выполнения и прикреплять инструкции;
* Это упрощает процесс постановки задач.

1. Просмотр выполненных заданий:

* Преподаватель может просматривать работы студентов, оставлять комментарии и выставлять оценки;
* Функция обеспечивает прозрачность процесса проверки.

1. Создание опросов:

* Преподаватель может создавать опросы для студентов (например, для сбора мнений);
* Это помогает взаимодействовать с аудиторией.

Функции администратора:

Администратор обладает всеми функциями других пользователей, а также следующими возможностями:

1. Управление пользователями:

* Добавление, удаление и редактирование профилей студентов, преподавателей и других администраторов;
* Цель функции — управление доступом к системе.

1. Управление данными:

* Администратор может добавлять, редактировать или удалять любые записи в базе данных, связанные с заданиями, категориями данных и уведомлениями.

1. Публикация информации и уведомлений:

* Возможность создавать новости, объявления или другую важную информацию для всех пользователей системы;
* Функция обеспечивает централизованное управление информацией.

1. Создание опросов:

* Администратор может создавать опросы для любых категорий пользователей (например, для сбора данных);
* Это помогает принимать решения на основе собранной информации.

1. Управление категориями данных:

* Возможность добавлять, редактировать или удалять категории данных (например, учебные материалы);
* Это позволяет структурировать данные для удобства пользователей.

## Структура данных

Для информационной системы «Личный кабинет студента» данные берутся из базы данных, содержащей несколько таблиц. Описание таблиц и их полей приведено ниже:

Таблица «Пользователи»:

* id\_пользователя – уникальный код пользователя, автоматически генерируется при создании;
* фио – фамилия, имя и отчество пользователя;
* пароль – пароль пользователя;
* роль – код роли, берется из таблицы «Роли».

Таблица «Роли»:

* id\_роли – уникальный код роли, автоматически генерируется;
* id\_названия – идентификатор названия роли.

Таблица «Студенты»:

* id\_студента – уникальный код студента, автоматически генерируется;
* задания – задания, связанные с данным студентом (ссылка на таблицу «Задания»);
* опросы – опросы, доступные студенту;
* информация – важная информация для студента;
* правила – перечень правил.

Таблица «Преподаватели»:

* id\_преподаватель – уникальный код преподавателя, автоматически генерируется;
* задания – задания, назначенные преподавателем;
* информация – информация, доступная для преподавателей.

Таблица «Администраторы»:

* id\_администратора – уникальный код администратора, автоматически генерируется;
* информация – общая информация, связанная с администратором;
* данные – данные, управляемые администратором (ссылка на таблицу «Данные»).

Таблица «Данные»:

* id\_данных – уникальный идентификатор записи;
* записи – текстовые записи, относящиеся к данным;
* добавка – дополнительная информация;
* сведения – уточненные сведения.

Таблица «Задания»:

* id\_задания – уникальный код задания, автоматически генерируется;
* название – название задания;
* описание – описание задания;
* **оценка** – оценка за выполнение задания.

Таблица «Задания»:

* id\_студента – код студента, выполнявшего задание (ссылка на таблицу «Студенты»);
* id\_задания – код задания (ссылка на таблицу «Задания»).

Логическая ER диаграмма показана на рисунке 2.

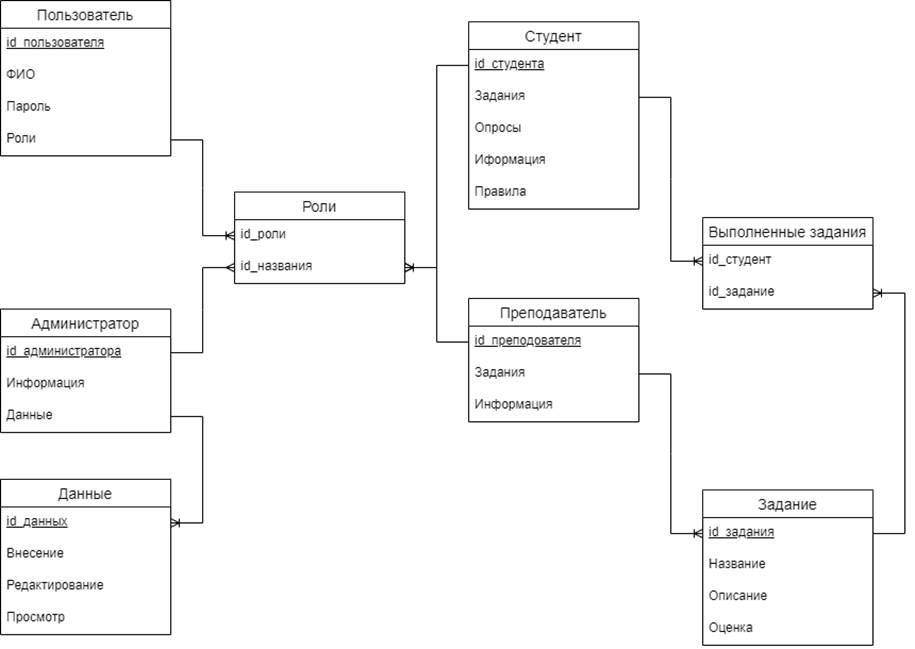


Рисунок 2 - Логическая ER диаграмма

Данная модель базы данных информационной системы «Личный кабинет студента» включает в себя следующие сущности: Пользователи, Роли, Студенты, Преподаватели, Администраторы, Данные, Задания, и Выполненные задания.

В базе данных были определены следующие связи:

1. «роль» в таблице «Пользователи» связана с «id\_роли» в таблице «Роли» – один ко многим, Один пользователь может иметь только одну роль, но одной роли может быть присвоено множество пользователей;
2. «id\_роли» в таблице «Роли» связана с сущностями «Студенты», «Преподаватели», и «Администраторы» – один ко многим, у одной роли может быть несколько студентов, преподавателей или администраторов, но каждая сущность относится только к одной роли.
3. «id\_задания» в таблице «Выполненные задания» связана с «id\_задания» в таблице «Задания» – один ко многим, Одно задание может быть выполнено несколькими студентами, но каждое выполненное задание связано с конкретной записью;
4. «id\_студента» в таблице «Выполненные задания» связана с «id\_студента» в таблице «Студенты» – один ко многим, Один студент может выполнить множество заданий, но каждая запись выполненного задания привязана к одному студенту;
5. «id\_данных» в таблице «Данные» связана с «id\_администратора» в таблице «Администраторы» – один ко многим, Один администратор может управлять несколькими наборами данных, но каждая запись данных привязана к одному администратору;

Физическая ER диаграмма представлена на рисунке 3.

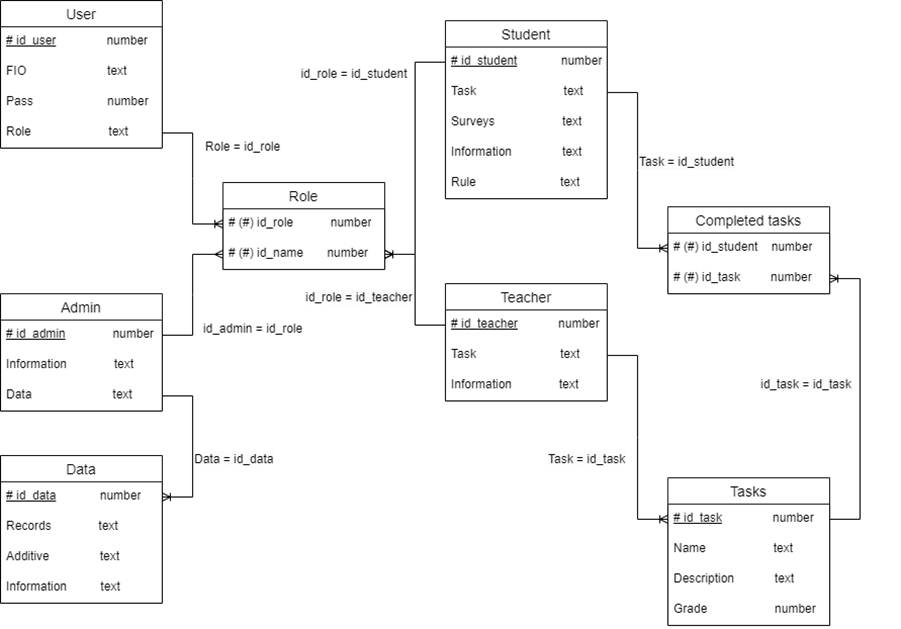


Рисунок 3 - Физическая ER диаграмм