

# Введение

В современном образовательном процессе автоматизация работы с данными студентов приобретает всё большее значение. Одним из эффективных инструментов для хранения и обработки информации является реляционная база данных SQLite, а язык программирования Python позволяет создавать удобные приложения для её управления.

В ходе практики была поставлена задача — разработать приложение для управления студенческими группами с использованием Python и библиотеки Tkinter для интерфейса, а также встроенной базы данных SQLite. Для повышения эффективности работы с текстом, формированием отчётов и документацией активно применялись современные нейросетевые языковые модели (например, GPT-4), что позволило ускорить процесс подготовки текстовых материалов и повысить их качество.

Использование нейросетей для генерации текстов и SQL-запросов открывает новые возможности для оптимизации работы, однако требует грамотного составления запросов (prompts) и учёта особенностей работы искусственного интеллекта.

# Глава 1. Теоретические основы работы с SQL и нейросетями в автоматизации обработки студенческих данных

# 1.1. Преимущества использования SQL и Python

- SQLite — это лёгкая, встроенная реляционная база данных, не требующая отдельного сервера. Она идеально подходит для небольших и средних проектов, в том числе для учёта студентов и групп. Python, благодаря своей простоте и большому количеству библиотек, позволяет быстро создавать приложения с графическим интерфейсом (например, на Tkinter) и реализовывать разнообразные функции: просмотр, изменение, фильтрацию и экспорт данных.

## 1.2. Роль нейросетей в текстовой работе и генерации кода

- Современные языковые модели (нейросети, такие как GPT-4) активно применяются для автоматизации рутинных задач: написание отчётов, генерация SQL-запросов, создание инструкций и документации. Работа с нейросетями строится на использовании prompts — текстовых запросов, которые формулируются пользователем для получения нужного результата.

# Преимущества текстовой работы с нейросетью:

- - Скорость: Быстрое создание отчётов, инструкций, примеров кода.
- - Гибкость: Возможность адаптировать стиль и структуру текста под требования.
- - Поддержка творчества: Генерация идей и нестандартных решений.

# Недостатки:

- - Необходимость проверки: Возможны ошибки и неточности в сгенерированном тексте или коде.
- - Зависимость от качества запроса: Чем точнее и понятнее prompt, тем лучше результат.
- - Ограниченность понимания: Нейросеть не всегда понимает контекст задачи так глубоко, как человек.

## 1.3. Что такое prompt

Prompt — это текстовое задание или вопрос, который пользователь задаёт нейросети для получения ответа. Например:

«Сгенерируй SQL-запрос для выбора всех студентов из группы "ИС-21".»

Чем более конкретно и понятно сформулирован prompt, тем выше вероятность получить релевантный и полезный результат.

# Технические детали работы

- - Язык отчёта: русский
- - Язык программирования: Python 3
- - База данных: SQLite
- - Графический интерфейс: Tkinter
- - Нейросеть: GPT-4 (OpenAI, Perplexity)
- - Среда разработки: VS Code / PyCharm
- - Оформление отчёта: MS Word / Google Docs
- - Скриншоты: встроенные средства ОС





