

- создан набор шаблонов простых Telegram-ботов;
- продемонстрирована возможность автоматизации процесса создания Telegram-ботов.

Следует отметить, что представленная версия системы является прототипом и имеет ряд ограничений. В частности, анализ пользовательских запросов реализован в упрощённом виде и не учитывает сложные формулировки. Кроме того, система не выполняет автоматическое развертывание и запуск сгенерированных ботов.

Тем не менее, полученные результаты подтверждают целесообразность выбранного подхода и демонстрируют возможность дальнейшего развития системы. В перспективе возможно использование методов машинного обучения и нейросетей для более точного анализа запросов, расширение набора поддерживаемых ботов, а также автоматизация процесса развертывания и запуска сгенерированных Telegram-ботов.

#### ЛИТЕРАТУРА

1 Telegram Bot API. Официальная документация. – URL: <https://core.telegram.org/bots/api> (дата обращения: 21.01.2026).

2 Aiogram 3.x Documentation. Асинхронный фреймворк для разработки Telegram-ботов на языке Python. – URL: <https://docs.aiogram.dev/en/dev-3.x/> (дата обращения: 20.01.2026).

УДК 004.42

Я.А. Сидорик;  
Н.Н. Пустовалова, доц., канд. техн. наук  
(БГТУ, г. Минск)

#### КОНСТРУКТОР БАЗ ДАННЫХ ДЛЯ POSTGRESQL

В настоящее время часто встречающейся проблемой является необходимость организовать информацию в структурированном виде. Это может быть актуально для менеджеров проектов, аналитиков, специалистов по качеству, начинающих разработчиков. Для решения задач учета, хранения результатов или анализа информации им нужна своя небольшая база данных. Но для создания базы требуется знание языка SQL и особенностей конкретной системы управления базами данных, например, PostgreSQL [1]. Специалист, который просто хочет быстро организовать свои данные, вынужден тратить недели на изучение основ. Это неэффективно и тормозит работу.

Конструктор баз данных способен устранить этот барьер с помощью интуитивно понятного графического интерфейса. Его задача –

превратить визуальное проектирование, похожее на рисование схемы на доске, в готовый, корректный программный код и рабочую базу данных. Пользователь должен думать о сути своей задачи, а не о синтаксисе команд.

При разработке конструктора соблюдались три ключевых принципа. Первый принцип – это визуализация. Пользователь может работать не с текстом запросов, а с графическими объектами. Таблицы выглядят как готовые блоки, связи между ними – как линии.

Второй принцип – использование шаблонов. Многие задачи являются типовыми. Поэтому в разрабатываемое приложение целесообразно встроить готовые схемы для распространенных случаев: например, структура для простого блога, для каталога товаров интернет-магазина или для учета задач в проекте.

Третий принцип – двусторонность. Приложение должно позволять не только создавать новое с нуля, но и открывать существующую базу данных, чтобы отобразить ее структуру в виде наглядной схемы, что превращает программу еще и в инструмент документирования и изучения.

Интерфейс приложения логично разделен на несколько основных модулей. В самом начале работы пользователь видит менеджер подключений. Это окно, где вводятся параметры для соединения с сервером PostgreSQL: адрес сервера, имя базы данных, логин и пароль. Удобно, что можно сохранить несколько настроек для разных проектов или серверов. После подключения открывается главное рабочее пространство. Это центральная часть программы, своего рода чистый холст. Здесь пользователь размещает таблицы, перетаскивая их мышью, может приближать или отдалять схему. Между таблицами рисуются наглядные линии, которые обозначают связи.

Рядом с холстом располагается панель инструментов с основными кнопками: «Добавить таблицу», «Добавить связь», «Выбрать шаблон», «Сгенерировать код», «Выполнить в базе данных». И, наконец, кнопка «Открыть редактор запросов» – это запасной вариант для сложных случаев.

Еще один важный элемент интерфейса – окно свойств. Оно динамически меняет свое содержимое в зависимости от того, что выбрано на холсте. Если выбрана таблица, в окне свойств можно изменить ее имя, добавить или удалить столбцы. Если выбран столбец, появляется возможность задать его тип, отметить, является ли он главным ключом, может ли содержать пустые значения. Все настройки делаются через привычные флажки и выпадающие списки.

Рассмотрим сценарии использования приложения. Первый сценарий – создание таблицы. Пользователь нажимает кнопку «Добавить таблицу» на панели инструментов. На холсте появляется новый прямоугольник с заголовком, например, «Таблица1». Пользователь сразу может переименовать ее во что-то осмысленное, например, «Товары». Затем в окне свойств он добавляет столбцы. Для каждого столбца вводит название, допустим, «Наименование», «Цена», «Наличие».

Далее нужно выбрать тип данных, которые имеют понятные названия: «Текст», «Число», «Флажок», «Дата». Программа сама сопоставит их с правильными типами PostgreSQL. Если пользователь отмечает столбец как «Главный ключ», программа автоматически позаботится о том, чтобы значения в нем были уникальными.

Второй, еще более важный сценарий, – создание связей между таблицами. Допустим, у нас уже есть таблица «Категории» с ключом «Код», и мы хотим связать с ней таблицу «Товары». Пользователь выбирает на панели инструментов режим «Создать связь». Затем он кликает сначала на столбец «Код» в таблице «Категории», а потом на таблицу «Товары». Программа делает всю сложную работу. Она анализирует типы данных и автоматически создает в таблице «Товары» новый столбец с подходящим именем, например, «КодКатегории». На холсте между таблицами рисуется линия связи. Кроме того, программа может спросить у пользователя, что делать с товарами, если категория будет удалена – удалять их вместе с ней или запрещать удаление. Все эти сложные правила реализуются автоматически.

Система шаблонов призвана ускорить начало работы. Часто пользователь не знает, с чего начать. Приложение предлагает ему выбрать готовую схему. Например, шаблон «Мини-блог» создаст на холсте три таблицы: «Авторы», «Статьи» и «Комментарии», уже связанные между собой правильным образом. Пользователю останется только переименовать их под свои нужды и добавить специфичные столбцы. Это не только экономит время, но и служит отличным обучающим примером. Когда схема готова, наступает самый ответственный момент – перенос ее в реальную базу данных. Здесь у пользователя есть два пути. Первый и самый простой – нажать кнопку «Выполнить в базе данных». Приложение само, в правильном порядке, сформирует и отправит на сервер все необходимые команды: создаст таблицы, пропишет связи, настроит ключи. Весь процесс будет отображен в журнале операций. Для дополнительного контроля всегда доступна вторая кнопка – «Сгенерировать код». Она открывает окно, в котором отображается весь текст программного кода, который будет выполнен.

Пользователь может его изучить, сохранить в файл или даже скопировать и выполнить в другой программе.

Важно, чтобы работа с базой данных была безопасной. Поэтому перед выполнением различных операций приложение сделает резервную копию структуры или, как минимум, выдаст четкое предупреждение о необратимости действий.

Несмотря на всю автоматизацию, предусматривается возможность для ручного управления. В любой момент пользователь может открыть встроенный редактор запросов, где можно выполнить любой специальный запрос, исправить мелкую ошибку или просто поэкспериментировать, т. е. приложение не ограничивает опытных пользователей, а дополняет их инструментарий.

Таким образом, приложение позволяет сделать мощные технологии баз данных доступными для широкого круга специалистов. В дальнейшем предполагается использовать для работы программные роботы [2], что сделает еще более удобным использование данного программного средства.

#### ЛИТЕРАТУРА

1. The PostgreSQL Reference Manual. Volume 1: SQL Language Reference // Network Theory Ltd., 2007. – 716 p.

2. Пустовалова Н.Н., Блинова Е.А. Программные роботы и базы данных // Информационные технологии. Материалы докладов 88-ой науч.-техн. конф. проф.-преподав. состава, научных сотрудников и аспирантов. – Минск: БГТУ, 2024. – С. 16-18.

УДК 004.853

Н.В. Ржеутская, ст. преп. (БГТУ, г. Минск)

### **АДАПТИВНАЯ СИСТЕМА АВТОМАТИЗИРОВАННОГО ОЦЕНИВАНИЯ ЗНАНИЙ НА ОСНОВЕ СЕМАНТИЧЕСКОГО АНАЛИЗА ТЕКСТОВЫХ ОТВЕТОВ С ИСПОЛЬЗОВАНИЕМ NLP**

Современные образовательные и корпоративные информационные системы все чаще ориентируются на формирование прикладных компетенций, требующих не только воспроизведения фактов, но и глубокого понимания изучаемого материала. В этих условиях традиционные системы тестирования, основанные на закрытых вопросах, демонстрируют ограниченную эффективность. Они не позволяют адекватно оценить способность обучающегося формулировать рас-