

жение в области цифрового сельского хозяйства в регионе, однако необходимо продолжать исследования и развивать инфраструктуру для поддержки инноваций. С учетом отмеченного Указа Президента РБ и трендов мирового рынка белорусское сельское хозяйство может значительно продвинуться в области цифровизации и инноваций, что приведет к повышению конкурентоспособности на международной арене и обеспечению устойчивого развития аграрного сектора.

Литература

1. Герасина, Е. В. Использование искусственного интеллекта в решении экологических проблем / Е. В. Герасина, М. А. Селина // Молодой ученый. – 2023. – № 46 (493). – С. 463–465.
2. Кочеткова, Д. А. Искусственный интеллект в бизнесе: автоматизация процессов и анализ данных / Д. А. Кочеткова, М. К. Черняков // Научный лидер. – 2025. – № 14 (215).
3. О приоритетных направлениях научной, научно-технической и инновационной деятельности на 2026–2030 годы : Указ Президента Респ. Беларусь от 1 апреля 2025 г. № 135. – URL: <https://president.gov.by/ru/documents/ukaz-no-135-ot-1-aprela-2025-g> (дата обращения: 05.04.2025).

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ ПО ИЗУЧЕНИЮ SQL НА БАЗЕ РАЗЛИЧНЫХ СУБД

Е. Ю. Бондарев

*Учреждение образования «Гомельский государственный технический
университет имени П. О. Сухого», Республика Беларусь*

Научный руководитель О. Д. Асенчик

Разработан программный комплекс для автоматизации выполнения и проверки лабораторных работ при изучении языка SQL. Он поддерживает работу с различными СУБД и включает инструменты для проверки запросов, генерации индивидуальных заданий и визуализации результатов выполнения. Отмечено, что разработанный алгоритм обеспечивает унифицированные критерии оценки и минимизирует субъективные ошибки.

Ключевые слова: SQL, автоматическая проверка заданий, генерация учебных заданий, СУБД, программный комплекс.

Изучение *SQL (Structured Query Language)*, языка структурированных запросов, который используется для управления базами данных и работы с данными, хранящимися в них, является неотъемлемой частью подготовки специалистов в области информационных технологий. Эффективное изучение требует активной практической отработки навыков. Лабораторные работы – неотъемлемая часть учебного процесса, позволяющая студентам вырабатывать навыки работы с различными объектами и конструкциями языка, понимать принципы работы с различными системами управления базами данных (СУБД) и приобретать навыки решения практических задач. Традиционный подход к обучению *SQL* включает в себя выполнение лабораторных работ, которые студенты делают самостоятельно, а преподаватели проверяют вручную. Проверка большого количества работ при наличии индивидуальных вариантов заданий является очень трудоемким процессом для преподавателя. При большом количестве работ сложно обеспечить полноту проверки заданий.

Современные студенты также сталкиваются с рядом сложностей при изучении *SQL*, таких как различия в синтаксисе различных СУБД (*MySQL, PostgreSQL, Oracle*) и отсутствие удобных инструментов для анализа ошибок. Многие существующие

платформы, такие как *SQL Fiddle* и *DB-Fiddle*, позволяют выполнять *SQL*-запросы в онлайн-режиме, но не предоставляют адаптивного обучения и локального развертывания для университетских нужд.

В настоящее время существует ряд аналогичных решений, таких как онлайн-курсы, тренажеры *SQL* и различные *IDE* для работы с СУБД [1-2]. Большинство из них не предоставляет возможностей для выполнения лабораторных работ с автоматической проверкой или не поддерживает работу с несколькими СУБД. Онлайн-курсы, как правило, ограничиваются теоретическими материалами и тестами, не давая возможности студентам работать с реальными *SQL*-запросами в контексте практических заданий. Тренажеры *SQL*, такие, как *SQLZoo* или *HackerRank*, предлагают упражнения для практики, но не могут заменить полноценный учебный процесс с лабораторными работами и проверкой запросов с учетом индивидуальных заданий. *IDE*, такие как *MySQL Workbench* и *pgAdmin*, предоставляют удобный интерфейс для работы с базами данных, но не предназначены для автоматической проверки лабораторных работ.

Автоматизация проверки *SQL*-запросов позволит существенно сократить время, которое преподаватель тратит на проверку работ. Это освободит время для более детального взаимодействия со студентами и повышения качества обучения. Автоматическая проверка обеспечит объективность оценки, исключив субъективный фактор и неполноту проверки задания. Это приведет к более справедливому и прозрачному результату оценивания.

Разработка такого комплекса в виде *Web*-приложения обеспечит доступность и удобство использования: студенты смогут выполнять лабораторные работы в любое время и в любом месте.

Разрабатываемый программный комплекс представляет собой инструмент для выполнения лабораторных работ по изучению *SQL* на базе различных систем управления базами данных (СУБД). Его основная цель – создание единой платформы для автоматизированного тестирования *SQL*-запросов.

Преподаватели могут загружать задания в систему, предоставляя студентам доступ к лабораторным работам разного уровня сложности. По завершении задания формируется отчет, позволяющий оценить знания и проанализировать ошибки студентов. Отчет оформляется в виде текстового сообщения и отправляется преподавателю в личные сообщения *Moodle*.

В рамках изучения *SQL* студенты осваивают работу с базами данных в *MS SQL Server*, включая создание и управление базами данных, написание *DDL*- и *DML*-запросов, разработку пользовательских функций, а также изучение хранимых процедур, курсоров и триггеров.

Разработанный программный комплекс [3] основан на стеке технологий, включающем *ASP.NET MVC* и *Entity Framework* для серверной части, а также *Razor Pages* и *JavaScript* для веб-интерфейса и *Docker* для управления контейнерами. Такой выбор технологий обеспечивает надежность и масштабируемость системы.

Архитектура построена на слоистом принципе с применением инверсии зависимостей. Это позволяет обеспечить модульность, тестируемость и гибкость в управлении бизнес-логикой и данными. Схема взаимодействия модулей приложения представлена на рис. 1.

Пользователь, будь то студент или преподаватель, взаимодействует с системой через веб-интерфейс, который построен на *Razor Pages* и *JavaScript*. Этот интерфейс позволяет выполнять различные действия, отправлять запросы на сервер и получать соответствующие ответы.

Сервер, реализованный на *ASP.NET MVC* с использованием *Entity Framework*, является основным обработчиком логики приложения. Он принимает запросы от веб-интерфейса, передает их в соответствующие модули и взаимодействует с базой данных. Для выполнения задач сервер делегирует обработку информации отдельным модулям.

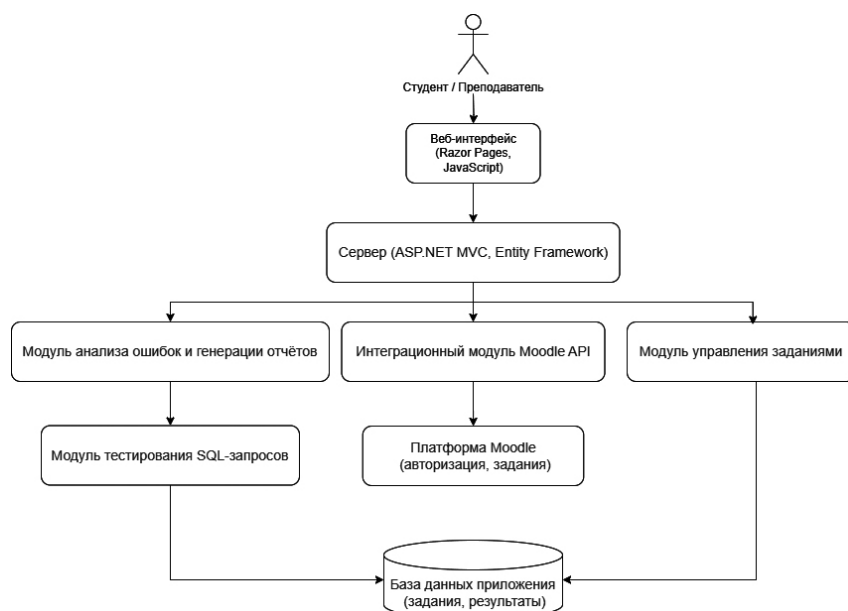


Рис. 1. Схема взаимодействия модулей приложения

Модуль анализа ошибок и генерации отчетов получает данные от сервера, обрабатывает информацию о запросах студентов и формирует отчеты по выполненным заданиям.

Модуль тестирования *SQL*-запросов предназначен для проверки корректности *SQL*-запросов, выполняемых студентами. Он подключается к базе данных приложения, выполняет тестовые запросы и передает результаты в модуль анализа ошибок.

Интеграционный модуль *Moodle API* отвечает за связь с платформой *Moodle*. Он отправляет и получает данные о заданиях и пользователях, обеспечивая синхронизацию информации между *Moodle* и серверной частью системы. Платформа *Moodle* используется для авторизации студентов и преподавателей, а также для хранения информации о заданиях.

Модуль управления заданиями отвечает за создание, редактирование и администрирование заданий. Он сохраняет задания в базе данных приложения и взаимодействует как с сервером.

База данных приложения выступает централизованным хранилищем данных. В ней содержится информация о заданиях. Она обслуживает запросы серверной части, модуля тестирования и интеграционного модуля *Moodle API*, обеспечивая корректное хранение и доступ к данным.

Алгоритм сравнения *SQL*-запросов начинается с выполнения двух запросов: студенческого и эталонного. Оба запроса выполняются в идентичных условиях, например, с использованием *Entity Framework*, чтобы обеспечить сопоставимость результатов. Затем выполняется сравнение данных, полученных из обоих запросов. В первую очередь, проверяется количество строк в результатах. Если оно отличается, это уже

указывает на ошибку. Далее анализируется структура данных, в частности, количество и порядок столбцов. Если структура данных не совпадает с эталонным результатом, это также считается несоответствием. Важным аспектом является проверка типов данных в каждом столбце. Несоответствие типов может указывать на ошибку, которая, возможно, неочевидна, на первый взгляд, но может привести к проблемам при дальнейшей обработке данных. Следующий шаг – построчное сравнение значений. Каждая строка из результата студенческого запроса сопоставляется с соответствующей строкой из эталонного результата. Если строки идентичны, то сравнение продолжается, но, если возникают различия, это фиксируется как ошибка.

Поддержка работы с *MySQL*, *PostgreSQL*, *SQLite*, *Oracle* и другими СУБД делает данный инструмент универсальным и применимым в различных образовательных учреждениях. Система тестирования *SQL*-запросов будет работать в среде, приближенной к реальной, что позволит студентам лучше подготовиться к работе с реальными базами данных.

Таким образом, программный комплекс дает возможность значительно сократить трудозатраты преподавателя на проверку. Автоматизация проверки также способствует повышению объективности оценки, исключая субъективный фактор и обеспечивая единообразие критериев оценки для всех студентов и вариантов заданий, создает основу для индивидуализации обучения.

Л и т е р а т у р а

1. SQL Academy: Интерактивные тренажеры SQL. – URL: <https://sql-academy.org> (дата обращения: 10.01.2025).
2. Подборка SQL тренажеров. – URL: https://pikabu.ru/story/podborka_sql_trenazhyorov_10605001 (дата обращения: 10.01.2025).
3. SqlStudio: GitHub. – URL: <https://github.com/EvgeniBondarev/SqlStudio> (дата обращения: 10.01.2025).

ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЕ ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ И УЧЕТА ДАННЫХ АВТОТРАНСПОРТА С ОЦЕНКОЙ ЭФФЕКТИВНОСТИ РЕСУРСОВ В РУП «ГОМЕЛЬЭНЕРГО»

Е. И. Маслак

*Учреждение образования «Гомельский государственный технический
университет имени П. О. Сухого», Республика Беларусь*

Научный руководитель В. В. Комраков

Рассмотрен процесс разработки информационно-аналитического веб-приложения для управления и учета данных автотранспорта в РУП «Гомельэнерго». Описаны технологии разработки, функциональные возможности системы и ее влияние на эффективность управления транспортными ресурсами. Особое внимание уделено вопросам автоматизации учета, анализа эксплуатационных данных и оптимизации затрат.

Ключевые слова: веб-приложение, учет автотранспорта, анализ данных, эффективность ресурсов, автоматизация, информационные технологии, мониторинг, управление транспортом.

Современные предприятия, такие как РУП «Гомельэнерго», активно используют автотранспорт для выполнения различных производственных задач, включая доставку оборудования, транспортировку персонала и выполнение технического об-