

# Реализация интерактивного сервиса поддержки обучения языку SQL

А. Э.Зинченко, email: z4444a@mail.ru

М. В.Матвеева, email: mariy.matveeva@gmail.com

Воронежский государственный университет

***Аннотация.** В данной работе рассматривается процесс проектирования интерактивного сервиса поддержки обучения языку SQL. Проведен анализ существующих приложений. На основе полученных данных выделена необходимая функциональность приложения, а также разработаны основные требования, по которым была построена модель данных и спроектирован интерфейс нового интерактивного сервиса поддержки обучения языку SQL, генерирующего задания по загружаемой пользователем схеме базы данных.*

***Ключевые слова:** интерактивный сервис поддержки обучения, база данных, модель данных, SQL.*

## **Введение**

Рынок веб-приложений каждый год пополняется огромным количеством обучающих ресурсов. Обучение языку структурированных запросов SQL остается достаточно популярным направлением, так как на данный момент его знание требуется во многих профессиональных областях, таких как: разработка и тестирование программного обеспечения, бизнес-аналитика, маркетинг.

На сегодняшний день существует множество сервисов, позволяющих овладеть и углубить знания языка запросов SQL. Данные сервисы интересны не только пользователям, желающим изучить язык SQL, но и преподавателям, которые могут использовать подобные средства для проверки уровня знаний студентов. В большинстве случаев такие сервисы включают в себя возможность самостоятельно написать операторы, которые должны вернуть или изменить данные, требуемые заданием.

У каждого из этих сервисов есть свои особенности и преимущества, но по большей мере они все схожи по функциональности и не соответствуют современным требованиям к веб-приложениям в сфере оформления визуальных интерфейсов и удобства в эксплуатации. К сожалению, ни один из сервисов на сегодняшний день не включает в

себя возможность генерации заданий по модели данных, загруженной пользователем, а также выборки задач случайным образом. Помимо вышеперечисленного, ни в одном из существующих сервисов не реализовано разграничение прав доступа и не выделены специальные функциональные возможности для преподавателей.

Таким образом, актуальность работы определяется необходимостью разработки сервиса, полностью отвечающего современным требованиям, ориентированного на программы для обучения студентов и начинающих изучать язык SQL, с которым было бы удобно работать как студенту, так и преподавателю.

Цель работы состояла в разработке и реализации алгоритма генерации индивидуальных заданий на составление SQL-команд по произвольной модели данных, а также в реализации программного модуля для проверки решений.

### 1. Анализ существующих приложений

Наиболее известные на данный момент сервисы, позволяющие строить SQL-команды: sql-ex, pgexercises, SQLBolt, sqlzoo, hackerrank.

Результаты сравнения перечисленных сервисов представлены в табл.1.

Таблица 1

*Сравнительный анализ существующих приложений*

<b>Критерии сравнения</b>	<b>Sql-ex</b>	<b>pgexercises</b>	<b>SQLBolt</b>	<b>sqlzoo</b>	<b>hackerrank</b>
Поддержка русского языка	+	-	-	-	-
Личный кабинет	-	-	-	+	+
Вход от имени преподавателя	-	-	-	-	-
Многопользовательская авторизация	+	-	-	+	+
Современный дизайн	-	+	+	-	+
Подсветка синтаксиса	-	+	+	-	+
Проверка форматирования	-	-	-	-	-
Возможность загрузки БД	-	-	-	-	-
Возможность загрузки шаблонов заданий и типовых решений	-	-	-	-	-

Случайная генерация заданий по загруженной БД	–	–	–	–	–
Возможность преподавателя назначить индивидуальные задания	–	–	–	–	–

Проанализировав пять различных сервисов, можно отметить, что все они имеют небольшой набор функциональных возможностей, а также не совсем удобный пользовательский интерфейс. Помимо этого, все, кроме одного сервиса, поддерживают только английский язык, что существенно осложняет изучение языка манипулирования данными для русскоязычных новичков.

Из представленных данных следует необходимость создания приложения, которое будет учитывать недостатки существующих сервисов и поддерживать их преимущества.

## **2. Метод генерации заданий**

Многовариантными заданиями называют особый вид задания, которые имеют несколько вариантов условий и отличаются исходными данными или деталями постановки вопроса. Многовариантные задания используются и для проверки знаний учащихся, где каждый ученик получает свой вариант условия, что затрудняет списывание, и для закрепления навыков за счет решения большого количества заданий на определенную тему.

Для автоматизации процесса генерации заданий преподавателями достаточно широко используются генераторы заданий, размещенных на общедоступных сервисах: [kpolyakov.spb.ru](http://kpolyakov.spb.ru), [mytestbook.com](http://mytestbook.com), [generatorzadach.narod.ru](http://generatorzadach.narod.ru), [www.schoolhousetech.com](http://www.schoolhousetech.com), [www.math-aids.com](http://www.math-aids.com), [wolframalpha.com](http://wolframalpha.com). Подробное описание методов работы данных сервисов представлены в статье [1] и монографии [2].

Проанализировав подходы к генерации заданий в указанных выше сервисах можно сделать вывод, что многовариантность задач достигается путем вариации коэффициентов, числовых значений, отдельных слов и в тестовых заданиях порядка следования вариантов ответа. Для математических дисциплин данные подходы являются достаточно эффективными, однако при формировании заданий на составление SQL-команд достичь желаемого результата с помощью применения таких подход не получится, так как вариация в числовых значениях не влияет на само решение, а приводит лишь к незначительным отличиям. С целью повышения вариативности заданий на составление SQL-команд используются индивидуальные схемы данных, которые загружаются в виде SQL-скриптов, содержащих

операторы создания таблиц с ограничениями, накладываемыми на данные.

С целью повышения вариативности заданий на составление SQL-команд используются индивидуальные схемы данных, которые загружаются в виде SQL-скриптов, содержащих операторы создания таблиц с ограничениями, накладываемыми на данные.

Для реализации генераторов в общем случае требуются знания программирования и использование специальных алгоритмов. Создание редактора генератора всегда основывается на некотором методе генерации заданий.

Рассмотрев и проанализировав популярные методы генерации тестовых заданий, для осуществления задачи формирования индивидуальных заданий на построение SQL-команд был выбран метод основанный на построении шаблонов. Выбранный метод является достаточно простым в использовании и реализации, так как не требует специальных знаний для добавления новых шаблонов в систему. При применении данного метода к индивидуальным входным схемам баз данных будет возможность получить большое количество многовариантных заданий.

Таким образом система позволяет пользователю загружать схемы баз данных, представленные в виде SQL-скриптов, содержащих операторы создания таблиц с ограничениями, накладываемыми на данные. Пользователи имеют возможность задавать шаблоны заданий. Параметрами шаблона для реализации генерации заданий на составление SQL команд являются имена таблиц, имена атрибутов, числовые значения, названия функций или знаки сравнения. Для каждого типа параметров реализован алгоритм генерации.

### **3. Алгоритм генерации заданий**

Для генерации заданий используются шаблоны. Шаблон указывается в текстовом формате. В квадратных скобках в шаблоне указываются параметры. В программе в созданный массив помещаются все параметры, содержащиеся в шаблоне.

Для параметров также могут указываться дополнительные сведения. Для таблиц указывается число, для атрибутов дополнительными сведениями являются: падеж и тип данных. Программа разделяет массив из найденных параметров по их типам, так как для каждого предусмотрена различная обработка.

Для параметра «таблица» из базы данных находят подходящие таблицы, которые содержат атрибуты необходимого типа, исключая из сравнения, атрибуты, которые имеют ограничение PRIMARY KEY или FOREIGN KEY.

Если не было найдено ни одной подходящей таблицы, программа заканчивается, и задание к этому шаблону не создается. В противном случае из списка найденных таблиц, случайным образом выбирается одна и для нее также случайным образом подбираются атрибуты необходимого типа.

Параметры «число», «знак сравнения» или «функция агрегирования» выбираются случайным образом из списка возможных значений сохраненных в программе.

Для параметров «таблица» и «атрибут» находятся в базе данных эквивалентные названия. Склонение названий атрибутов и таблиц по падежам, осуществляется с помощью веб-сервиса «Морфер» 3.0. Веб-сервис «Морфер» предназначен для склонения по падежам слов на русском языке, в программу он подключается в виде библиотеки с помощью фреймворка для автоматизации сборки Gradle.

Полученные значения подставляются в шаблон, на место их соответствующих параметров. Готовые задания сохраняются в базу данных.

Дополнительно в программе осуществляется проверка на уникальность сформированного задания в базе данных.

На рисунке 1 представлена диаграмма компонентов модуля генерации заданий.

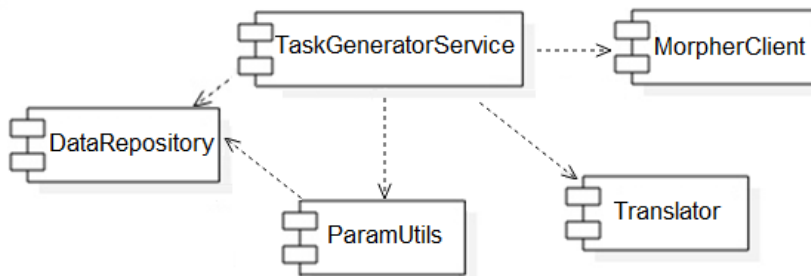


Рис. 1. Диаграмма компонентов модуля генерации заданий

Компоненты модуля генерации заданий:

- TaskGeneratorService – содержит бизнес-логику генерации заданий.
- MorpherClient – содержит классы для подключения к внешнему веб-сервису с ресурса morpher.ru.
- Translator – отвечает за поиск и подстановку эквивалентных названий.

- ParamUtils – отвечает за подбор параметров типа «число», «знак сравнения» или «функция агрегирования».
- DataRepository – включает в себя работу с базой данных.

#### 4. Метод проверки решений

Во многих приложениях, посвященных обучению языку SQL, метод проверки правильности составления SQL-команд реализован на основе сравнения результатов эталонного решения и решения пользователя. Однако при реализации данного метода проверки возможен риск выдачи положительной оценки решению, которое является ошибочным, непрофессиональным или неоптимальным с точки зрения логики, так как получено всего лишь методом подбора [3].

Из этого следует, что существует необходимость анализировать не только результаты выполнения SQL команд, но и код решения. Для реализации проверки кода решения, оптимальным вариантом является сравнения решения задания по определенной теме с типовым решением подобного класса задач. Для этого необходимо осуществить хранение типовых решений, которые удобно было бы представить так же, как и задания в виде шаблонов.

Для каждого задания необходимо указать типовое решение, примеры типовых задач с ответами представлены в табл. 2 [4]. При введении типового решения преподаватель указывает параметры в квадратных скобках. Чтобы ускорить процесс поиска всевозможных различий в ответе студента необходимо вместо параметров подставлять определенные значения.

Таблица 2

#### *Примеры типовых заданий и типовых решений*

<b>Тема</b>	<b>Типовое задание</b>	<b>Типовое решение</b>
SELECT *	Выбрать все данные о [таблица].	SELECT * FROM [таблица]
WHERE + BETWEEN	Напишите запрос, выводящий все данные о [таблица], с id входящим в множество [число].	SELECT * FROM [таблица] WHERE [атрибут] BETWEEN [число] AND [число]

CASE	Выбрать [атрибут], и если есть [атрибут], то [функция агрегирования].	SELECT [атрибут], CASE (WHEN [атрибут] THEN [атрибут] ELSE [функция агрегирования] END) FROM [таблица]
GROUP BY + HAVING	Выбрать [атрибут]. В результат включить только те [атрибут], количество которых [знак сравнения] [число].	SELECT [атрибут] FROM [таблица] GROUP BY [атрибут] HAVING COUNT ([атрибут])[знак сравнения][число].

Стоит учесть то, что если введенное решение не соответствует типу, то это не означает, что ответ неверный. Окончательную оценку правильности и оптимальности решения проводит преподаватель.

На рис. 2 представлены результаты проверки решений системой, где преподаватель может указать правильность выполнения задания.

The screenshot shows the 'SQL Expert' interface with the title 'Решения РороваР по работе "Библиотека"'. It contains three tasks:

- Задание 1:** 'Выбрать всю информацию о книгах, где количество страниц больше 150.' The student's solution is:
 

```
SELECT *
FROM books
WHERE amount_pages > 150
```

 Below the code are two buttons: 'НЕПРАВИЛЬНО' (Incorrect) and 'ПРАВИЛЬНО' (Correct).
- Задание 2:** 'Вывести всю данные о книгах, где год издания больше 1950 и меньше 2000.' The student's solution is:
 

```
SELECT *
FROM books
WHERE publisher_year > 1950 AND publisher_year < 2000
```

 Below the code are two buttons: 'НЕПРАВИЛЬНО' (Incorrect) and 'ПРАВИЛЬНО' (Correct).
- Задание 3:** 'Выбрать всю информацию о книгах, которые содержат "мир" в названии.' The student's solution is partially visible:
 

```
SELECT *
```

On the right side, there is a 'Типовое решение' (Typical solution) section with a template:
 

```
SELECT *
FROM [p1.Table]
WHERE [p2.Attribute] BETWEEN [p3.Number] AND [p4.Number]
```

Рис. 2. Страница, содержащая результаты проверки решений студента

## 5. Функциональные возможности и основные процессы системы поддержки обучения

Проанализировав все необходимые элементы интерактивного сервиса, возможности которого удовлетворяли бы студентов и преподавателей, была составлена следующая структура интерактивного сервиса (рис. 3):



Рис. 3. Общая структура интерактивного сервиса

Серверная часть приложения состоит из следующих модулей, каждый из которых отвечает за определенную функциональность:

- модуль создания объектной модели базы данных;
- модуль генерации заданий;
- модуль проверки ответа студента с типовым решением.

На входе в систему поступает готовая база данных в виде SQL скрипта, который загружает преподаватель. В модуле создание объектной модели загруженный скрипт преобразуется и сохраняется в базу данных в виде отдельных названий таблиц, столбцов и их ограничений. На основе созданной объектной модели в модуле генерации заданий приложение с помощью хранимых тем и шаблонов, которые также загружает преподаватель, генерирует набор индивидуальных заданий по схеме. Сгенерированные задания поступают студенту в текстовом формате.

Студент в текстовом формате отправляет решённые задания с сайта на сервер, где они проверяются в системе на соответствие с типовым решением, которые хранятся в базе данных.

На основании результата проверки задания системой, преподаватель принимает решение о правильности решения студента.

В результате работы приложения отображаются и суммируются правильно решённые задания студента, которые потом также хранятся на сервере.



## 6. Реализация программного комплекса

При реализации проекта использовались следующие программные средства:

- интерактивная среда разработки IntelliJ IDEA Community Edition 2018.1.3;
- язык разработки Java;
- реляционная система управления базами данных PostgreSQL;
- фреймворк для автоматизации сборки Gradle;
- фреймворк для Java-платформы Spring;
- фреймворк React JS.

Приложение имеет REST-архитектуру, которая позволяет разделить написание серверной части и клиентской части. Серверная часть подразделяется на 3 слоя: слой для работы с базой данных, слой, отвечающий за бизнес-логику, и слой, обрабатывающий запросы к серверу и отправляющий сформированные ответы.

Слой для работы с базой данных представлен следующими типами данных:

- сущности, которые отражают структуру базы данных в виде Java-классов, которые ставятся в соответствие таблицам с помощью ORM-фреймворка Hibernate;
- классы-репозитории, которые предоставляют операции чтения, создания, обновления и удаления данных из базы посредством соответствующего манипулирования сущностями.

Слой, отвечающий за бизнес-логику приложения, содержит 3 основных модуля:

- модуль ScriptParser;
- модуль TaskGenerator;
- модуль CheckAnswer.

Сервисы, относящиеся к модулю ScriptParser, отвечают за анализ SQL-скриптов и преобразование их в объектную модель. Модуль TaskGenerator осуществляет генерацию заданий на составление SQL-команд по указанной базе данных и выбранным шаблонам. Модуль CheckAnswer реализует проверку решений заданий.

Классы-контроллеры относятся к слою, обрабатывающем HTTP-запросы клиента и формирующему ответы по соответствующим URL.

### Заключение

Результатом работы является реализация интерактивного сервиса поддержки обучения языку SQL с возможностью генерации

индивидуальных задач по схемам баз данных и проверки введенных решений.

### **Список литературы**

1. Коновалов Я. Ю. Генератор контрольных заданий по высшей математике: опыт создания и применения / Я. Ю. Коновалов, С. К. Соболев // Инженерный вестник. – 2015. – № 04. – С. 1046–1055.

2. Посов И. А. Обзор генераторов и методов генерации учебных заданий / И. А. Посов // Образовательные технологии и общество. – 2014. – Т. 17, № 4. – С. 593–609.

3. Семенова З. В. Сравнительная характеристика средств автоматизированной проверки правильности составления SQL-запросов / З. В. Семенова, С. А. Любич, А. Г. Кузнецов, П. А. Мальцев // Вестник СибАДИ. – 2017. – № 3(55). – С. 152–160.

4. Матвеева М. В. Язык манипулирования данными. Практикум / М. В. Матвеева. – Воронеж : Издательский дом ВГУ, 2019. – 44 с.