

Санкт-Петербургский Государственный Университет
Математико-механический факультет
Математическое обеспечение и администрирование
информационных систем
Кафедра информационно-аналитических систем

Плужникова Анастасия Дмитриевна, гр. 19.Б08-мм

**Автоматическая генерация заданий на
основе регулярных выражений**
Выпускная квалификационная работа

Научный руководитель:
к. ф-м. н., доцент Графеева Н.Г.

Санкт-Петербург
2023

Оглавление

Введение.....	3
Цель и задачи.....	7
Выбор СУБД.....	8
Методология.....	9
Создание пользовательского интерфейса.....	14
Создание демонстрационной базы данных.....	19
Создание демонстрационного списка заданий.....	22
Заключение.....	23
Список литературы.....	24
Приложения.....	25

Введение

В последние десятилетия постепенно происходит цифровизация различных сфер жизни общества, в том числе и образования. Появляются новые образовательные платформы, ресурсы и онлайн-курсы по широкому спектру направлений. Потребителей привлекает возможность учиться в удобное время и удобном месте. Согласно исследованию Российского рынка онлайн-образования в 2021 году траты взрослого населения на дополнительное онлайн-образование превысили траты на офлайн образование (226 млрд руб. и 214 млрд руб. соответственно)[1].

Высшие учебные заведения разрабатывают собственные онлайн-курсы и включают их в учебные планы основных образовательных программ. Это повышает привлекательность программ, позволяет сэкономить временные и финансовые ресурсы. Министерство науки и высшего образования России, в свою очередь, поддерживает использование электронных ресурсов и дистанционных образовательных технологий университетами.

Одним из ключевых навыков, который получают студенты математических, экономических и технических специальностей, является навык работы с базами данных. Поэтому многие вузы, реализующие программы в таких областях, заинтересованы в разработке курсов по основам работы с базами данных.

Важнейшим аспектом процесса обучения является проверка знаний. Она позволяет оценить прогресс учащегося, указать на его сильные и слабые стороны, закрепить изученный материал. Онлайн-тестирования позволяют объективно измерить и интерпретировать результаты, а также снимают с преподавателя объёмную работу по проверке и обработке ответов студентов.

Стоит отметить, что для эффективного контроля качества освоения учебных дисциплин необходимо создать как можно больше вариантов контрольных заданий. Во-первых, это снижает возможность совместного решения заданий

учащимися, а во-вторых, при повторном прохождении теста студентом, заполнение ответа будет выполнено не по памяти, а через повторное обдумывание решения, что поможет закрепить материал. Однако в таком случае на составлении качественного теста преподавателю придётся затратить значительное количество времени. Отсюда возникает идея автоматизировать процесс составления вариантов.

Эта проблема и методы её решения были рассмотрены в ряде работ. Можно выделить два основных подхода к генерации вариантов, которые были в них описаны:

1) Генерация на основе шаблонов.

В этом методе под шаблоном подразумевается описание задачи, в котором исходные данные и/или часть задачи может меняться. В основном такой подход применяется при генерации математических задач, где вместо параметров (переменных) подставляются конкретные числа. Например, шаблон может выглядеть как «Сколько будет $[A]+[B]$?» и ответ «Ответ: $[A]+[B]= [C]$ ». В данном случае $[A]$, $[B]$ и $[C]$ – параметры. Первые два выбираются случайным образом из заранее заданного списка значений или интервала значений, а третий вычисляется на их основе. Такой подход считается довольно распространённым в силу его понятности, доступности для восприятия и описан в таких работах, как «Модели и алгоритмы генерации задач в компьютерном тестировании», «Генераторы в компьютерных учебных программах» и «Разработка компьютерных учебников и обучающих систем» [2-4].

2) Генерация на основе И/ИЛИ деревьев.

Деревья И/ИЛИ – это модель представления данных в виде дерева, каждый из узлов которого имеет тип «И» или «ИЛИ». Данный подход был рассмотрен в монографии [5]. В терминах решения задачи И-узел означает, что решение задачи разбивается на подзадачи. Решение всей задачи зависит от решения

всех подзадач. ИЛИ-узел означает, что задача может быть решена несколькими методами.

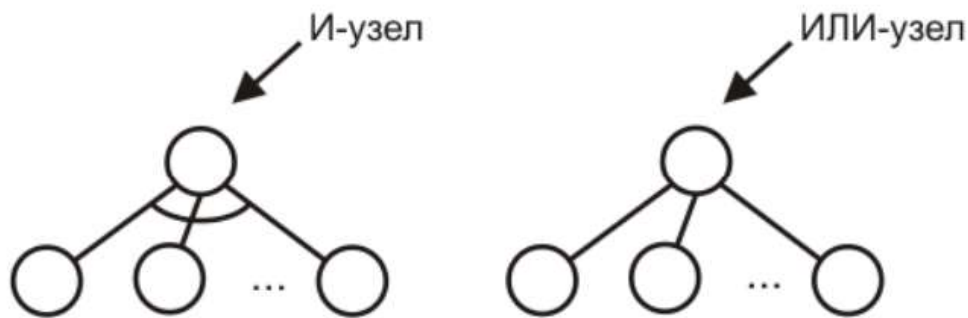


Рисунок 1. Схематическое представление узлов И/ИЛИ

Таким образом, Вариантом дерева И/ИЛИ назовем поддереву, которое получается из заданного путем отсечения выходных дуг кроме одной у всех ИЛИ-узлов.

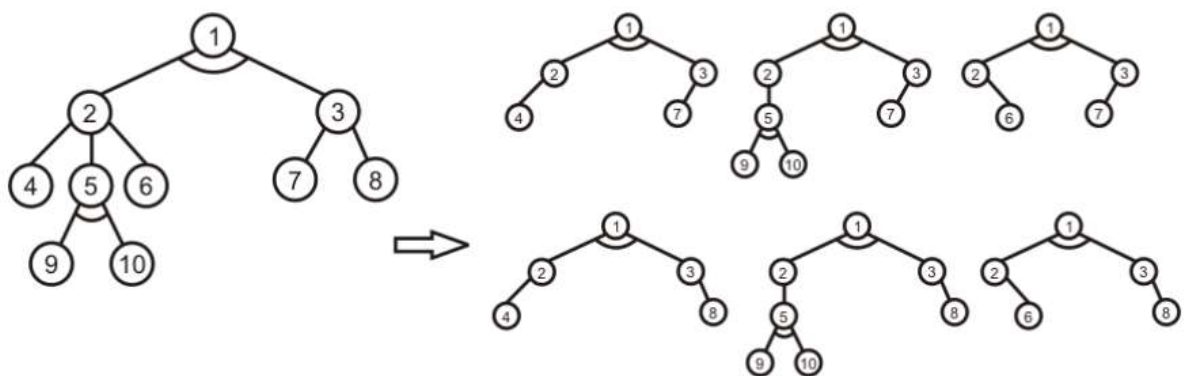


Рисунок 2. Варианты дерева И/ИЛИ

Также стоит рассмотреть системы, которые на сегодняшний день могут быть использованы для генерации тестовых заданий:

- 1) <https://www.wolframalpha.com/pro/problem-generator> - сайт, предлагающий автоматически-сгенерированные задания по различным разделам математики, таким как арифметика, линейная алгебра, статистика, теория чисел и др. К сожалению, задания представлены

только на английском языке, а в заданиях в рамках одной темы варьируются только числа.

- 2) <https://www.yaklass.ru/> - цифровой образовательный ресурс для школ, на котором ученикам предоставляется возможность решать задания по различным предметам. Портал содержит свой редактор генератор заданий, однако он только для математики
- 3) <https://abakbot.ru/online-16/191-generator-zadaniy-onlajn> – сайт предоставляет возможность генерации математических примеров. На вход в качестве параметров подаются разрядность, операнды, количество операций и использование скобок, а на выходе 20 вариантов с ответами, причём число вариантов нельзя изменить.

Таким образом, так как при составлении вариантов по курсу баз данных предполагается, что ответами к заданиям будут служить выборки из некоторой базы данных, а не результаты арифметических выражений, на основе существующих систем не получится составить многовариантные тесты по курсу баз данных.

Цель и задачи

Целью работы является создание приложения автоматической генерации вариантов по курсу баз данных.

Для достижения этой цели были поставлены следующие задачи:

- 1) Определить методологию генерации вариантов
- 2) Создать пользовательский интерфейс для генерации вариантов
- 3) Создать новую демонстрационную базу, на основе которой будет протестирована работа приложения генерации вариантов
- 4) Создать демонстрационный список шаблонов заданий, на основе которой будет протестирована работа приложения генерации вариантов

Выбор СУБД

При написании программы для заполнения базы использовалась реляционная СУБД с открытым исходным кодом PostgreSQL версии 15.0. Система обладает рядом преимуществ, такими как надёжность, расширяемость, высокая производительность и широкая функциональность. Стоит отметить, что PostgreSQL поддерживает концепцию ACID (Atomicity - атомарности, Consistency - согласованности, Isolation - изолированности, Durability - долговечности). На момент начала написания работы версия 15.0 была наиболее высокой среди представленных на официальном сайте [6].

В работе потребуется работать с регулярными выражениями. СУБД PostgreSQL предоставляет возможность работать с регулярными выражениями стандарта POSIX или стандарта SQL через операторы и строковые функции.

В работе будут использованы следующие строковые функции [7]:

- 1) `regexp_substr (string text, pattern text [, start integer [, N integer [, flags text [, subexpr integer]]]]) → text`

Возвращает подстроку N-го вхождения регулярного выражения POSIX в строке или NULL, если таких вхождений не нашлось;

Эта функция будет использоваться для поиска и выделения параметров из шаблона текста (см. Методология, этап 3).

- 2) `regexp_replace (string text, pattern text, replacement text [, start integer] [, flags text]) → text`

Заменяет подстроку первого вхождения регулярного выражения POSIX `pattern` или подстроки всех таких вхождений, если использован флаг `g`

Эта функция будет использоваться для добавления параметров в шаблоны текста и запроса (см. Методология, этап 3).

Методология

Как было указано ранее, из двух описываемых в литературе методов генерации, была выбрана генерация на основе шаблонов. Предполагается, что задания курса баз данных будут основаны на некоторой содержательной базе (объём строк должен быть достаточным для подготовки нужного числа уникальных вариантов). Для удобства разработки и первичного тестирования было решено сделать демонстрационную базу данных (см. раздел Создание демонстрационной базы данных). При этом подход, описанный в работе должен быть применим к любой базе данных.

Работу с генерацией вариантов можно разделить на 3 этапа:

- 1) Добавление/обновление информации о шаблонах заданий в таблицу. Добавляются шаблоны текстов заданий и запросов для получения ответов. В шаблонах на месте изменяемых значений должны находиться параметры определённого вида.
- 2) Добавление/обновление информации об используемых в шаблонах параметрах. Параметры из шаблонов необходимо добавить в отдельную таблицу, указав запрос для выбора конкретного значения этого параметра.
- 3) Вызов процедуры генерации заданий и ответов `pr_generate_variants (int, int)`. Процедура сгенерирует задания и ответы, добавив их в отдельную таблицу. Эта процедура может быть использована в системе проверки знаний онлайн-курсов.

Подробнее о каждом из них:

Этап 1. Добавление информации о шаблонах заданий

Создаются таблицы `template` и `variant`, в которых будет содержаться информация о шаблонах и вариантах заданий, следующего вида:



Таблица	Поля	Значение
template информация о шаблонах заданий	template_id (PK) template_text template_query level	Идентификатор шаблона Шаблон текста задания Шаблон запроса Уровень сложности задания
variant информация о варианте задания	variant_id (PK) template_id (FK) parameters variant_text result	Идентификатор задания Ссылка на шаблон Параметры задания Текст задания Правильный ответ на задание

Поля `template_id` и `variant_id` заполняются через последовательности. Со скриптом создания таблиц можно ознакомиться в Приложении.

Далее необходимо добавить в таблицу `template` шаблоны заданий, запросов, а также уровень сложности задания. На месте изменяемых значений должны находиться параметры. Позже программа будет искать параметры в шаблоне и менять их на конкретные значения из базы данных, используя регулярные выражения. Было решено записывать названия параметров в виде `PT_*`, так как все параметры должны быть схожи, чтобы можно было легко найти их в остальном тексте. При этом параметры с разными именами должны отвечать за разные поля базы данных.

Например, если преподаватель планирует создать варианты по заданию

«Выведите фамилии посетителей с именем __»,

ответ на которое можно найти через выборку

«**SELECT last_name FROM visitor WHERE first_name = ' __ '»,**

где имя посетителя будет изменяемым, ему необходимо выбрать, каким параметром будут параметризоваться имена (например, PT_FIRST_NAME), какой будет уровень сложности у задания и добавить в таблицу template следующую запись:

template_id	template_text	template_query	level
1	Выведите фамилии посетителей с именем PT_FIRST_NAME	SELECT last_name FROM visitor WHERE first_name = PT_FIRST_NAME	1

Если в задании используются несколько значений из одной таблицы, пользователь может параметризовать их одним именем, например

template_id	template_text	template_query	level
2	Выведите фамилии посетителей с именем PT_FIRST_NAME или PT_FIRST_NAME	SELECT last_name FROM visitor WHERE first_name = PT_FIRST_NAME OR first_name = PT_FIRST_NAME	2

Этап 2. Добавление информации об используемых параметрах.

Далее необходимо создать таблицу parameter следующего вида:



The diagram shows a table titled 'parameter' with three columns: 'parameter_id' (marked as a primary key), 'parameter_name', and 'parameter_query'. Each column is associated with the data type 'int'.

parameter		
 parameter_id		int
parameter_name		int
parameter_query		int

Таблица	Поля	Значение
parameter информация о параметрах	parameter_id (PK) parameter_name parameter_query	Идентификатор параметра Имя параметра Запрос, по которому будет происходить выбор значения на место параметра

Поле `parameter_id` заполняется через последовательность. Со скриптом создания таблиц можно ознакомиться в Приложении.

После этого можно добавить в таблицу `parameter` информацию о параметрах, которые используются в шаблонах заданий.

SQL-запросы (поле `parameter_query`) должны выбирать из таблиц или списков значений одно случайное значение и сохранять его в переменную `var_value` (эта переменная используется в работе процедуры генерации вариантов).

Например, пользователь добавил задания из примеров второго этапа. Соответственно, ему теперь нужно описать параметр **PT_FIRST_NAME**. Если у пользователя есть таблица `visitor` и он хочет параметризовать её поле `first_name` параметром **PT_FIRST_NAME**, он мог бы добавить в таблицу `parameter` следующую запись:

parameter_id	parameter_name	parameter_query
1	PT_FIRST_NAME	SELECT first_name FROM visitor ORDER BY random() LIMIT 1

Этап 3. Вызов процедуры генерации заданий и ответов `pr_generate_variants` (int, int). Она принимает на вход идентификатор шаблона и количество вариантов, которые необходимо сгенерировать.

Программа с помощью регулярных выражений будет последовательно перебирать параметры в шаблонах текста и запроса и менять их на конкретные значения, которые будет находить через запросы из таблицы `parameter`. Ответы

на задания она будет искать, выполняя преобразованный запрос из таблицы template.

Например, в результате вызова pr_generate_variants (2, 2), в таблицу variant могут добавиться следующие записи:

variant_id	template_id	parameters	variant_text	result
1	2	{Милан,Софья}	Выведите фамилии посетителей с именем Милан или Софья	{ Мацнова, Машадиева, Мелешкова,Мелик-Еганов, Мерибанов,... }
2	2	{Анна,Павел}	Выведите фамилии посетителей с именем Анна или Павел	{ Мейлихова, Менежетдинова, Мещанинов, Мильченкова, Милюсева, ... }

Процедура должна будет генерировать любое заданное пользователем число вариантов, при этом, если появляются дубликаты заданий, процедуре необходимо указать на них. Также процедура должна будет обрабатывать возможные ошибки пользователя, например, использование параметра, не описанного в таблице parameter.

Создание пользовательского интерфейса

Для удобства пользователей было решено создать интерфейс для работы с параметрами, шаблонами и вариантами, которое позволит генерировать варианты заданий по базам данных на основе существующей базы.

Интерфейс написан на языке Python с использованием кроссплатформенной библиотеки для разработки графического интерфейса Tkinter, его расширениями Pmw и customTkinter.

Также в работе используется модуль для работы с PostgreSQL базами данных - Psycopg2.

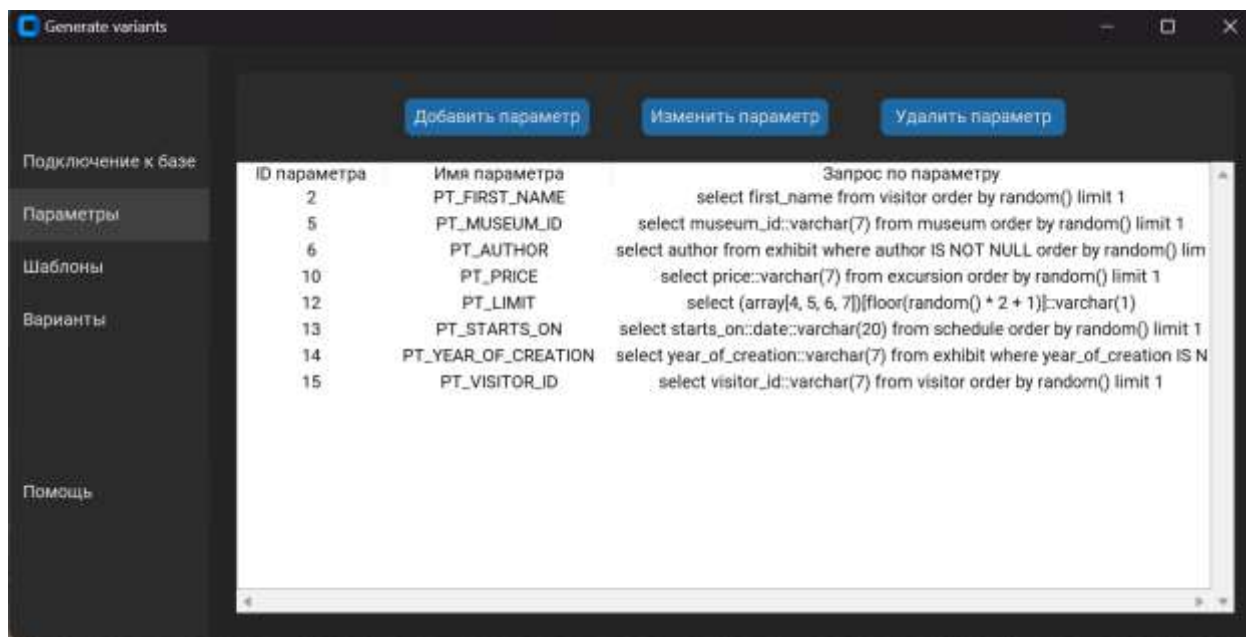
В приложении представлено 5 разделов со следующей функциональностью:

Подключение к базе

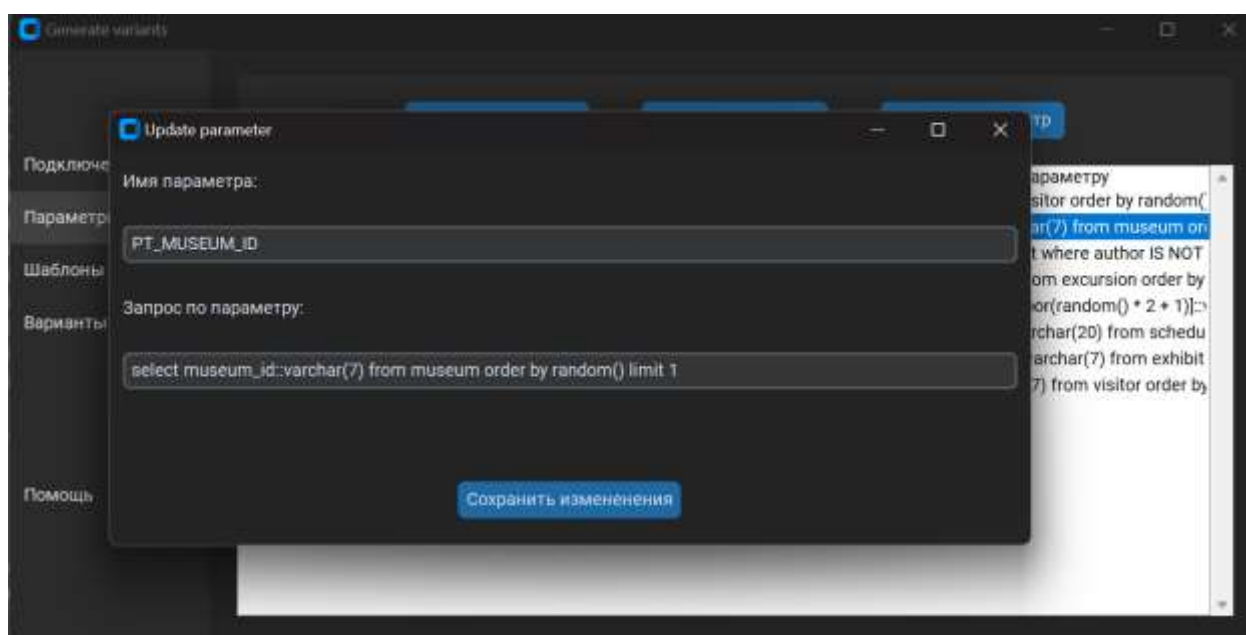
Перед началом работы с приложением необходимо подключиться к базе данных, на основе которой будут генерироваться задания.

Для этого необходимо ввести Имя базы данных, Имя пользователя (по умолчанию, postgres), Пароль и Хост (по умолчанию, localhost). Далее нажать Подключиться. После успешного подключения появится надпись Подключено.

Параметры

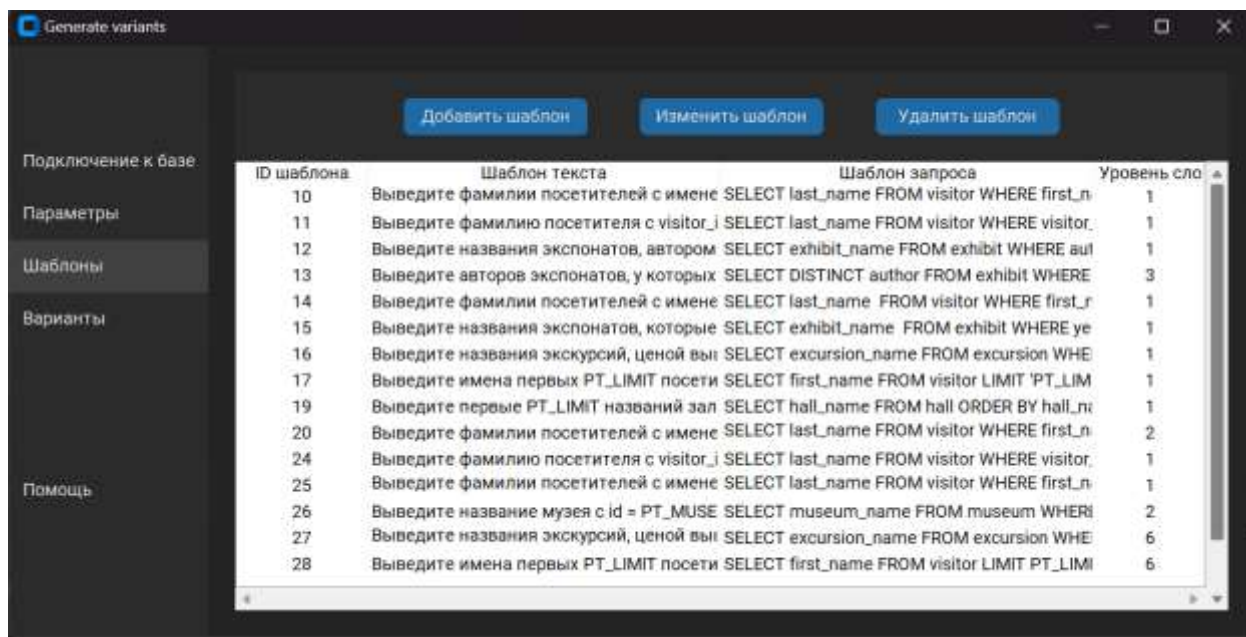


Если в базе ещё нет таблицы параметров, будет предложено её создать. Форма Параметры позволяет просмотреть информацию о параметрах, а также добавлять, изменять или удалять их. При добавлении или изменении параметра пользователь может указать имя параметра и запрос по параметру. При сохранении нового параметра или внесении изменений проверяется, что имя параметра уникально, а SQL-запрос возвращает одно значение.

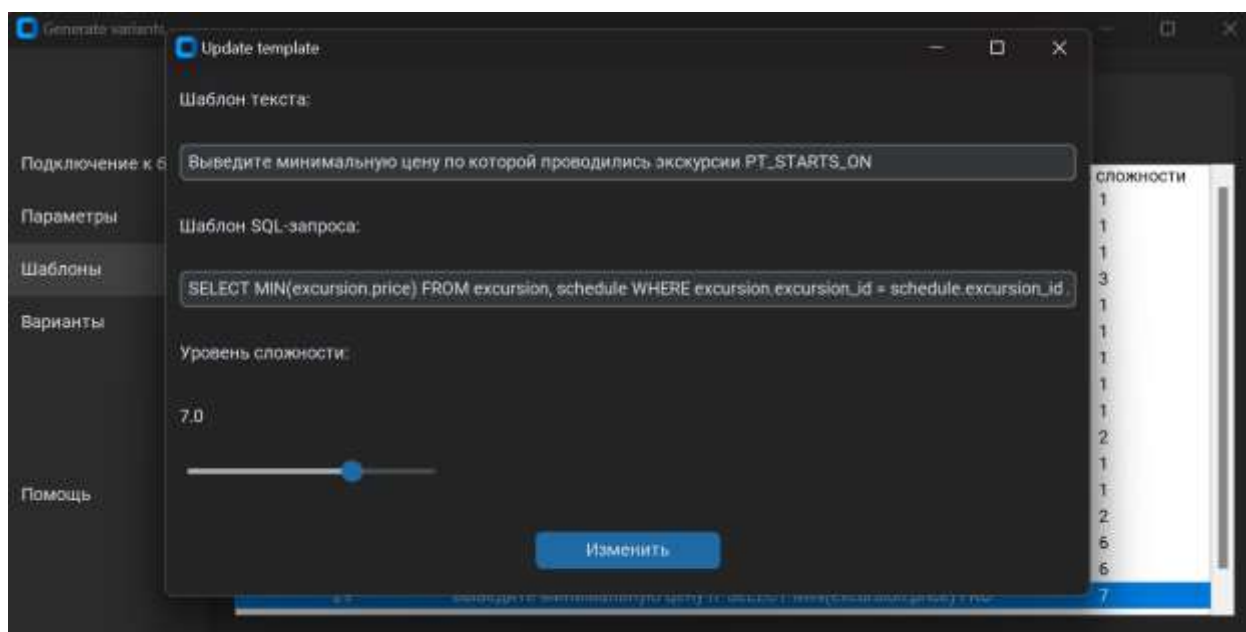


При изменении и удалении параметра нужный параметр выделяется ПКМ. Выделение снимается ЛКМ. Все изменения сразу же отображаются на форме.

Шаблоны

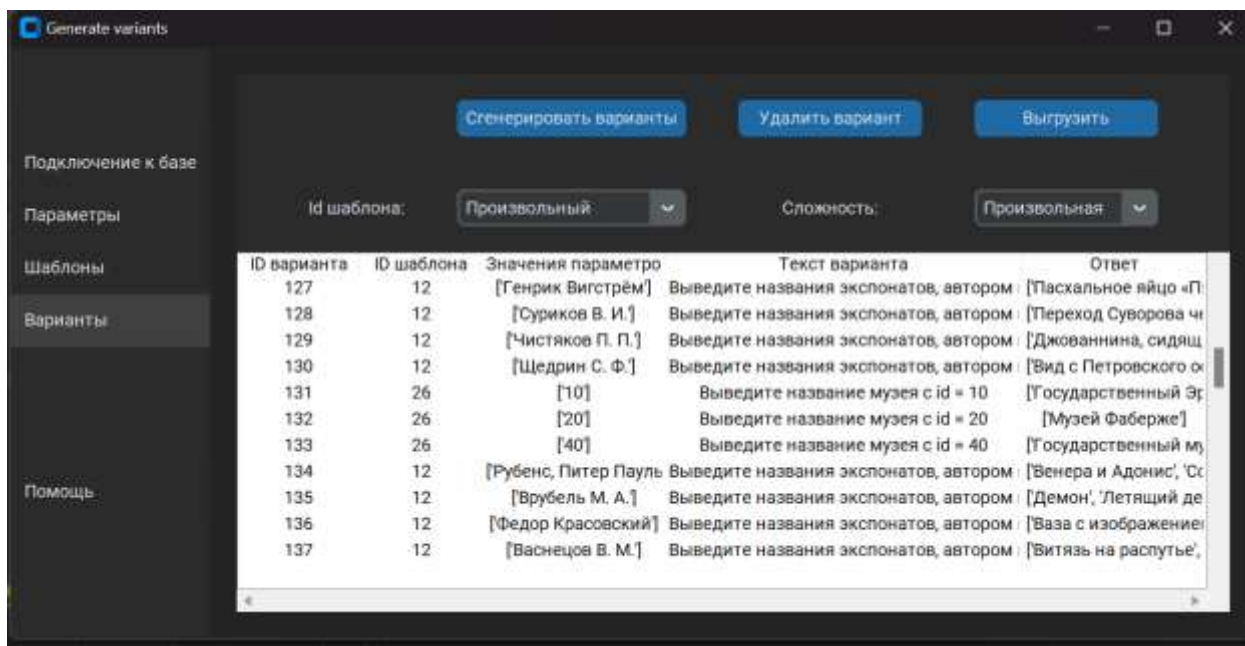


Если в базе ещё нет таблицы шаблонов, будет предложено её создать. Форма Шаблоны позволяет просмотреть информацию о шаблонах, а также добавлять, изменять или удалять их. При добавлении или изменении шаблона пользователь может указать шаблон текста, шаблон запроса и сложность (от 1 до 10).



При изменении и удалении шаблона нужный шаблон выделяется ПКМ. Выделение снимается ЛКМ. Все изменения сразу же отображаются на форме.

Варианты



Если в базе ещё нет этой таблицы вариантов, будет предложено её создать. Также, если в базе ещё нет процедуры генерации, она добавится.

Форма Варианты позволяет просмотреть информацию о вариантах, а также генерировать, выгружать или удалять их. Также можно отсортировать варианты по id шаблона и сложности заданий.

Для генерации вариантов пользователю необходимо нажать на кнопку Сгенерировать варианты, в появившемся окне выбрать шаблон из таблицы шаблонов, ввести количество вариантов (есть автоматическая проверка вводимых символов – можно вводить только числа) и нажать кнопку Сгенерировать (или клавишу Enter). Если генерация прошла успешно, появится соответствующее сообщение. Иногда в процессе генерации могут возникать дубликаты, информация о которых выведется в сообщении.

Также у пользователя есть возможность выгрузки вариантов в файл. Будут выгружены только те задания, которые отображены в текущем списке. То есть, если стоит в фильтре указаны произвольные сложность и id шаблона, будут выгружены все варианты в порядке увеличения id шаблона. Если указана сложность – будут выгружены только задания определённой сложности. Если указан id шаблона – выгрузятся варианты только одного задания.

Для выгрузки необходимо нажать кнопку Выгрузить и в появившемся окне выбрать директорию и указать имя файла. Создастся txt-файл со списком заданий.

Пример выгруженного задания:

Задание № 1 (Сложность 1)

Выведите названия экспонатов, автором которых является PT_AUTHOR

PARAMETERS={['Строцци, Бернардо']['Генрик Вигстрём']['Чистяков П. П.']['Щедрин С. Ф.']['Рубенс, Питер Пауль (Пьетро Пауло)']['Врубель М. А.']['Федор Красовский']['Васнецов В. М.']['Суриков В. И.']}

RESULT={['Исцеление Товита']['Пасхальное яйцо «Пятнадцатилетие царствования»', 'Бонбоньерка в форме кресла', 'Флакон парфюмерный с крышкой в виде женской головки', 'Бюст императора Александра III', 'Настольные часы с глобусом']['Джованнина, сидящая на подоконнике']['Вид с Петровского острова в Петербурге', 'Вид Неаполя. Набережная Санта Лючия', 'Новый Рим. Замок Святого Ангела']['Венера и Адонис', 'Союз Земли и Воды (Шельда и Антверпен)', 'Вакх', 'Отцелюбие римлянки (Кимон и Перо)', 'Уход Агари из дома Авраама', 'Пир у Симона Фарисея']['Демон', 'Летающий демон', 'Венеция']['Ваза с изображением цветов']['Витязь на распутье', 'Бой скифов со славянами', 'Богоматерь с младенцем']['Переход Суворова через Альпы в 1799 году', 'Покорение Сибири Ермаком', 'Взятие снежного городка']}

Помощь

Раздел содержит справочную информацию. Инструкция разбита на этапы работы.

Также для удобства пользователей перед входом в приложение появляется приветственное окно с краткой инструкцией, а при наведении курсора на некоторые поля ввода - всплывающие подсказки.

С кодом пользовательского интерфейса можно ознакомиться в Приложении 1.

С пользовательской инструкцией можно ознакомиться в Приложении 2.

Создание демонстрационной базы данных

В качестве основы для заданий было решено создать демонстрационную базу данных MUSEUMS.

В базе хранится информация о четырёх художественных музеях Санкт-Петербурга: Государственном Эрмитаже, Музее Фаберже, Государственном Русском музее и Государственном музее «Царскосельская коллекция» [8, 9, 10, 11]. Выбор именно этих музеев обусловлен достаточной информацией о залах, выставленных экспонатах и экскурсиях на их официальных сайтах. Общее число добавленных в базу залов - 65. Общее число добавленных экспонатов - 192. Общее число добавленных экскурсий - 25.

Также была добавлена информация о посетителях музеев на основе открытой базы имён и фамилий [12]. Объединение имён и фамилий в таблицу посетителей происходило с помощью отдельной процедуры. Суммарно было добавлено 146132 посетителя.

Заполнение графика проведения экскурсий также происходило через специальные процедуры для ежедневных и еженедельных экскурсий. Всего было добавлено 20932 сеанса во временном интервале от 02.01.2023 до 31.03.2024.

Количество экскурсий, на которые записался каждый посетитель, выбиралось случайным образом и не превышало 5. При помощи процедур в таблицу visitor_schedule, связывающую посетителей и сеансы экскурсий, было добавлено 438652 строки. Таким образом, в среднем в экскурсионную группу записано около 20 человек.

Итого, общий объём базы – более 600 тысяч строк.

ER-диаграмма базы:

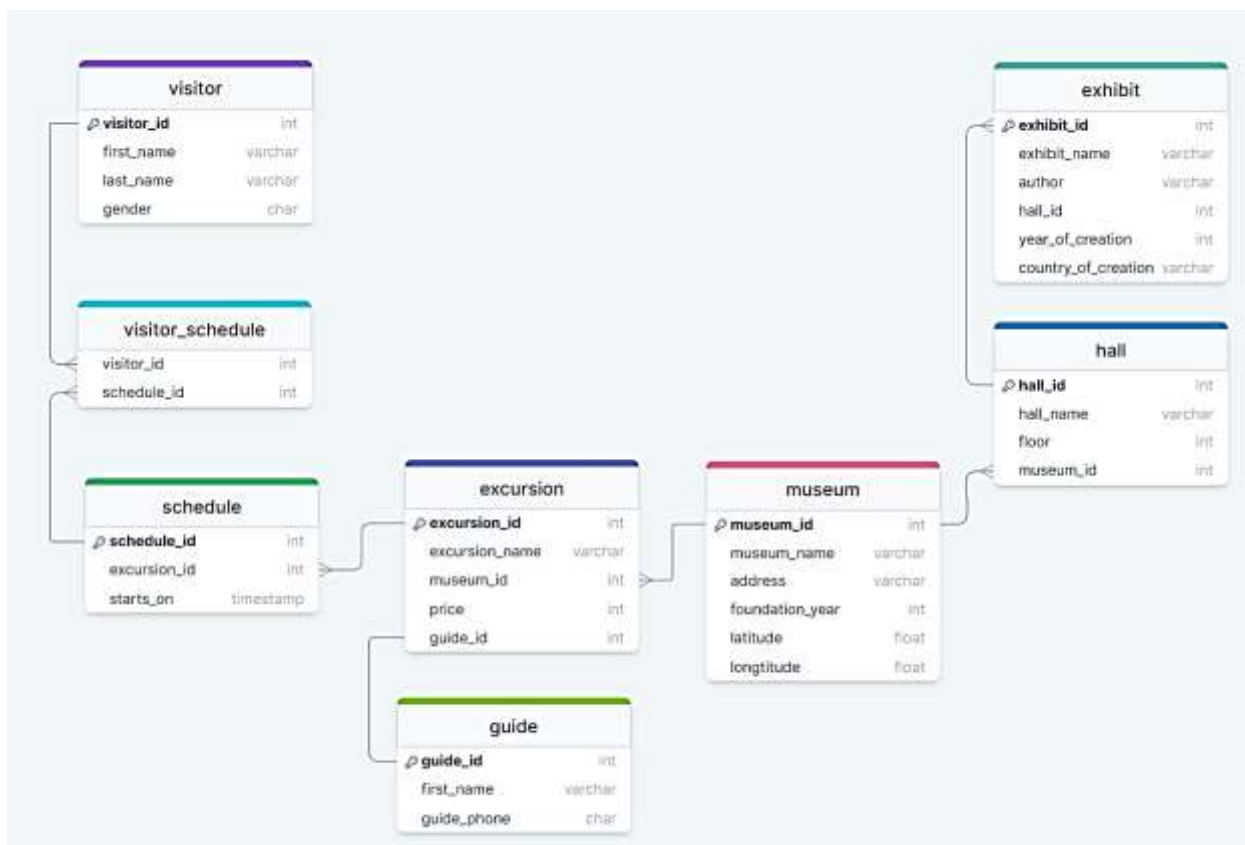


Таблица	Поля	Значение
museum информация о музеях	museum_id (PK) museum_name address foundation_year latitude longitude	Идентификатор музея Название музея Адрес Год основания Широта Долгота
hall информация о залах	hall_id (PK) hall_name floor museum_id (FK)	Идентификатор зала Название зала Этаж Ссылка на музей
exhibit информация об экспонатах	exhibit_id (PK) exhibit_name author hall_id (FK) year_of_creation country_of_creation	Идентификатор экспоната Наименование экспоната Автор Ссылка на зал Год создания Страна создания
guide информация о гидах	guide_id (PK) first_name phone	Идентификатор гида Имя гида Телефон гида

excursion информация об экскурсиях	excursion_id (PK) excursion_name museum_id (FK) price guide_id (FK)	Идентификатор экскурсии Название экскурсии Ссылка на музей Цена экскурсии Ссылка на гида
schedule информация о графике проведения экскурсий	schedule_id (PK) excursion_id (FK) starts_on	Идентификатор сеанса Ссылка на экскурсию Дата и время начала экскурсии
visitor информация о посетителях музея	visitor_id (PK) first_name last_name gender	Идентификатор посетителя Имя посетителя Фамилия посетителя Пол посетителя
visitor_schedule таблица связи посетителей и расписания	visitor_id (FK) schedule_id (FK)	Ссылка на посетителя Ссылка на сеанс

Со скриптами создания и заполнения демонстрационной базы данных можно ознакомиться в Приложении 3.

Создание демонстрационного списка заданий

Был составлен демонстрационный список заданий с постепенным увеличением сложности. Были охвачены основные темы для составления выборок, такие как базовый синтаксис запросов, операторы WHERE, ORDER BY, GROUP BY, LIMIT, ALL, ANY, LIKE агрегатные функции, многотабличные запросы. Всего было составлено 33 задания.

Пример шаблона простого задания (уровень сложности 1-3):

Выведите фамилии посетителей с именем PT_FIRST_NAME или PT_LAST_NAME.

Пример шаблона задания средней сложности (уровень сложности 4-6):

Выведите фамилии посетителей мужского пола, имена которых начинаются на PT_FIRST_NAME.

Пример шаблона задания повышенной сложности (уровень сложности 7-10):

Выведите среднюю цену по которой проводились экскурсии PT_STARTS_ON. Округлите до двух знаков после запятой.

Также к каждому шаблону задания был составлен шаблон запроса с аналогичными параметрами.

Был составлен демонстрационный список параметров для этих заданий.

С полным перечнем заданий можно ознакомиться в Приложении 4.

С полным перечнем параметров можно ознакомиться в Приложении 5.

Заключение

В соответствии с поставленной целью - создание приложения автоматической генерации вариантов по курсу баз данных – была определена методология генерации: согласно ней работа с генерацией разбивается на три этапа, а ключевым инструментом генерации являются регулярные выражения; для удобства пользователей был создан графический интерфейс; было проведено тестирование интерфейса - разработана новая демонстрационная база данных объёмом более 600 тысяч строк, по которой созданы демонстрационные шаблоны заданий по разным темам курса баз данных разных уровней сложности. В приложении предусмотрена возможность выгрузки заданий для дальнейшего использования в системе проверки онлайн-курсов.

Приложение позволяет преподавателям не прописывать различные варианты вручную, а следовательно затрачивается меньше времени и усилий при разработке онлайн-курсов.

В дальнейшем приложение может быть расширено на работу с другими СУБД. Также можно доработать используемые функции, сделав их работу более быстрой и интерфейс, сделав его более удобным для пользователей.

Список литературы

1. «Исследование российского рынка онлайн-образования»
https://netology.ru/edtech_research_2022
2. Кручинин В.В., Морозова Ю.В. Модели и алгоритмы генерации задач в компьютерном тестировании // Известия Томского политехнического университета, 2004, № 5. – С.127-131.
3. Кручинин В. В. Генераторы в компьютерных учебных программах. Томск: издво Том. Ун-та, 2003. — 200 с
4. Башмаков А. И., Башмаков И. А. Разработка компьютерных учебников и обучающих систем. М.: Издательский дом «Филинь», 2003. – С. 616.
5. Кручинин В. В. Методы построения алгоритмов генерации и нумерации комбинаторных объектов на основе деревьев И/ИЛИ. Томск: Изд-во «В-Спектр», 2007. – С. 200.
6. PostgreSQL: The World's Most Advanced Open Source Relational Database [Электронный ресурс] – Режим доступа: <https://www.postgresql.org/>
7. PostgreSQL Documentation: 15: 9.4. String Functions and Operators [Электронный ресурс] – Режим доступа: <https://www.postgresql.org/docs/current/functions-string.html#FUNCTIONS-STRING-FORMAT>
8. Эрмитаж – главная страница [Электронный ресурс] – Режим доступа: <https://hermitagemuseum.org/wps/portal/hermitage/?lng=ru>
9. Сайт музея Фаберже в Санкт-Петербурге [Электронный ресурс] – Режим доступа: <https://fabergemuseum.ru/>
10. Русский музей [Электронный ресурс] – Режим доступа: <https://rusmuseum.ru/>
11. ГМЦК [Электронный ресурс] – Режим доступа: <https://museumfa.ru/>
12. База данных имён и фамилий [Электронный ресурс] – Режим доступа: https://mydata.biz/ru/catalog/databases/names_db

Приложения

1. [Код пользовательского интерфейса](#)
2. [Пользовательская инструкция](#)
3. [Скрипты создания и заполнения демонстрационной базы данных](#)
4. [Демонстрационный список заданий](#)
5. [Демонстрационный список параметров](#)