

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра Прикладная математика и информатика

(наименование)

09.04.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Информационные системы и технологии корпоративного управления

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

на тему «Исследование и разработка информационной системы автоматизированной проверки учебных заданий при подготовке ИТ-специалистов»

Студент

А.Д. Козлов

(И.О. Фамилия)

(личная подпись)

Руководитель

д.т.н., доцент, С.В. Мкртычев

(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

Оглавление

Введение.....	4
Глава 1 Анализ исследований и разработок в области построения информационных систем автоматизированной проверки учебных заданий по составлению SQL-запросов.....	9
1.1 Методические подходы к автоматизированной проверке учебных заданий при подготовке ИТ-специалистов	10
1.2 Обзор и анализ разработок информационных систем автоматизированной проверки учебных заданий по составлению SQL-запросов.....	15
Глава 2 Анализ методологий построения информационных систем автоматизированной проверки учебных заданий по составлению SQL-запросов.....	21
2.1 Модели и алгоритмы систем автоматизированной проверки учебных заданий по составлению SQL-запросов.....	21
2.2 Методологические подходы к построению систем автоматизированной проверки учебных заданий по составлению SQL-запросов.....	31
Глава 3 Разработка информационной системы автоматизированной проверки учебных заданий на составление SQL-запросов.....	39
3.1 Алгоритмы автоматической проверки заданий на составление SQL-запросов.....	39
3.2 Логическое моделирование информационной системы автоматизированной проверки учебных заданий на составление SQL-запросов.....	43
3.3 Моделирование данных информационной системы автоматизированной проверки учебных заданий на составление SQL-запросов.....	53

Глава 4 Апробация проектных решений информационной системы автоматизированной проверки учебных заданий на составление SQL-запросов	57
Заключение	66
Список используемой литературы и используемых источников.....	68

Введение

В настоящее время в сфере подготовки ИТ-специалистов активно развиваются дистанционные технологии обучения.

Как показывает практика, одним из самых трудоемких и затратных по времени процессов является процесс проверки учебных заданий.

Так, по данным ВШЭ весной 2014-2015 учебного года, на ручную проверку заданий лабораторного практикума по дисциплине «Видеотехнологии» в группе из 27 человек у преподавателя уходило в среднем четыре часа [5].

Еще одна существенная проблема – негативное влияние человеческого фактора на результаты проверок, что проявляется в том числе на объективности оценивания учебных заданий.

Указанные проблемы приводят к снижению эффективности преподавания конкретного онлайн-курса, и, как следствие – к снижению эффективности всего процесса дистанционного обучения ИТ-специалистов.

Очевидным решением данных проблем является использование в обучающем процессе ИТ-специалистов средств автоматической проверки учебных заданий, разработанных на основе современных ИТ в том числе автоматизированных информационных систем.

Однако, следует отметить, что определенную сложность в решение данной проблемы вносит отличие методик автоматической проверки учебных заданий для различных ИТ-дисциплин. Это требует создания специализированных средств автоматической проверки заданий по каждой дисциплине.

В онлайн-курсах по подготовке ИТ-специалистов, как правило, представлены дисциплины по изучению баз данных и управлению данными.

Обязательным компонентом контента таких дисциплин являются учебные задания по составлению SQL-запросов.

Исследование и разработка информационной системы, позволяющей автоматизировать проверку таких заданий, представляет научно-

практический интерес.

Актуальность темы исследования обусловлена необходимостью исследования и разработки информационной системы автоматизированной проверки учебных заданий по составлению SQL-запросов, обеспечивающей повышение эффективности процесса подготовки ИТ-специалистов.

Объектом настоящего исследования является процесс подготовки ИТ-специалистов.

Предметом исследования является информационная система автоматизированной проверки учебных заданий при подготовке ИТ-специалистов.

Целью работы являются исследование и разработка информационной системы автоматизированной проверки учебных заданий по составлению SQL-запросов, обеспечивающей повышение эффективности процесса подготовки ИТ-специалистов.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ современного состояния исследований в области разработки и внедрения информационных систем автоматизированной проверки учебных заданий по составлению SQL-запросов;
- провести анализ методологических подходов к построению средств автоматизированной проверки учебных заданий, обосновать выбор методологии и технологии для разработки информационной системы автоматизированной проверки учебных заданий при подготовке ИТ-специалистов;
- разработать информационную систему автоматизированной проверки учебных заданий при подготовке ИТ-специалистов;
- выполнить апробацию предлагаемых проектных решений и оценить эффективность разработанной информационной системы.

Гипотеза исследования: применение разработанной в рамках

диссертационного исследования информационной системы автоматизированной проверки учебных заданий по составлению SQL-запросов обеспечит повышение эффективности процесса подготовки ИТ-специалистов.

Методы исследования. В процессе исследования использованы следующие положения и методы: системный подход, методологии и технологии проектирования информационных систем, теория баз данных, основы языка SQL.

Новизна исследования заключается в разработке моделей и алгоритмов информационной системы автоматизированной проверки учебных заданий по составлению SQL-запросов, обеспечивающей повышение эффективности процесса подготовки ИТ-специалистов.

Практическая значимость исследования заключается в возможности практического применения информационной системы автоматизированной проверки учебных заданий в процессе подготовки ИТ-специалистов.

Теоретической основой диссертационного исследования являются научные труды ученых и специалистов, занимающихся проблемами проектирования информационных систем автоматизированной проверки учебных заданий для подготовки ИТ-специалистов.

Основные этапы исследования: исследование проводилось с 2019 по 2021 год в несколько этапов:

На первом (констатирующем) этапе формулировалась тема исследования, выполнялся сбор информации по теме исследования из различных источников, проводилась формулировка гипотезы, определялись постановка цели, задач, предмета исследования, объекта исследования и выполнялось определение проблематики данного исследования.

Второй этап – поисковый. В ходе проведения данного этапа осуществлялся анализ методологий проектирования систем автоматизированной проверки учебных заданий по составлению SQL-запросов, были разработаны модели и алгоритмы информационной системы,

опубликована научная статья по теме исследования в научном сборнике.

На третьем этапе осуществлялись реализация и апробация информационной системы автоматизированной проверки учебных заданий по составлению SQL-запросов, произведена оценка ее эффективности, сформулированы выводы о полученных результатах по проведенному исследованию.

На защиту выносятся:

- модели и алгоритмы информационной системы автоматизированной проверки учебных заданий по составлению SQL-запросов;
- результаты апробации и оценки эффективности использования информационной системы автоматизированной проверки учебных заданий по составлению SQL-запросов.

По теме исследования опубликована 1 статья: Козлов А.Д. Система автоматизированной проверки учебных заданий по составлению SQL-запросов // Вестник научных конференций. 2021. №10-3(74). С. 61-62.

Диссертация состоит из введения, четырех глав, заключения и списка литературы.

Во введении обоснована актуальность темы исследования, представлены объект, предмет, цели и задачи и положения, выносимые на защиту диссертации.

В первой главе представлен анализ исследований и разработок в области построения информационных систем автоматизированной проверки учебных заданий по составлению SQL-запросов. Рассмотрены методические подходы к автоматизированной проверке учебных заданий для студентов ИТ-направлений. Даны обзор и анализ разработок информационных систем автоматизированной проверки учебных заданий по составлению SQL-запросов

Вторая глава посвящена анализу методологий построения информационных систем автоматизированной проверки учебных заданий по составлению SQL-запросов. Исследованы модели и алгоритмы систем

автоматизированной проверки учебных заданий по составлению SQL-запросов. Проанализированы методологические подходы к построению систем автоматизированной проверки учебных заданий по составлению SQL-запросов.

Третья глава посвящена разработке информационной системы автоматизированной проверки учебных заданий на составление SQL-запросов. Разработаны алгоритмы автоматической проверки заданий на составление SQL-запросов. Выполнено логическое моделирование информационной системы автоматизированной проверки учебных заданий на составление SQL-запросов.

В четвертой главе представлены результаты апробация проектных решений информационной системы автоматизированной проверки учебных заданий на составление SQL-запросов.

В заключении приводятся результаты исследования.

Работа изложена на 71 странице и включает 33 рисунка, 8 таблиц, 33 источника.

Глава 1 Анализ исследований и разработок в области построения информационных систем автоматизированной проверки учебных заданий по составлению SQL-запросов

Ручная проверка учебных заданий по составлению запросов на языке SQL трудоемка и требует много времени.

Так, в одном вопросе может быть много стилей операторов SQL, имеющих одинаковые результаты выходных данных. Более того, разные системы управления базами данных (СУБД) могут иметь разный синтаксис операторов SQL. Это может привести к ошибкам, которые приведут к ухудшению преподавания практики у студентов ИТ-специальностей.

Последнее обусловило мотивацию исследователей и специалистов к разработке автоматизированной системы оценки практических заданий по составлению SQL-запросов.

Таким образом, исследование и разработка ИС, позволяющей автоматизировать проверку учебных заданий по составлению SQL-запросов для подготовки ИТ-специалистов, представляет научно-практический интерес.

Проблематика автоматизированной проверки учебных заданий в том числе при подготовке ИТ-специалистов широко представлена в работах таких российских и зарубежных ученых и специалистов, как Андрианов И.А., Горохова-Алексеева А.В., Перязева Ю., Сеницын С.В., Poženel M., Renard L. и другие.

Активная работа по внедрению средств автоматической проверки учебных заданий ведется в ВШЭ и МИФИ [1].

Проблематике автоматизации проверки заданий по составлению SQL-запросов посвятили свои труды Говоров А.И., Калинин В.Ю., Моисеенко С.И., Талипов М., AL-Salmi A., Russel G., Vichianroj P. и другие.

Вместе с тем необходимо констатировать недостаточность работ, посвященных исследованиям в области построения систем автоматической

или автоматизированной проверки учебных заданий, используемых в процессе подготовки ИТ-специалистов.

1.1 Методические подходы к автоматизированной проверке учебных заданий при подготовке ИТ-специалистов

Рассмотрим работы, посвященные анализу методических подходов к автоматизированной проверке заданий при подготовке ИТ-специалистов.

Вопросы автоматизации контроля знаний при подготовке ИТ-специалистов в МИФИ рассмотрены в работе [14].

Отмечается в частности, что «автоматизированные средства контроля по математическому программированию разработаны и успешно используются с 1997 года. К 2006 г студентами выполнено более 20000 индивидуальных заданий.

При этом в несколько раз сокращаются затраты времени студентов и преподавателей на проведение контроля и достигается необходимый уровень его качества (рисунок 1).

Тема	Уменьшение времени в T раз	
	для студента	для преподавателя
Одномерный поиск	3.9	5.3
Нелинейное программирование	4.1	5.2
Линейное программирование	1.9	2.4

Рисунок 1 – Показатели экономии времени при автоматизированном контроле учебных заданий

В работе [2] предложена концепция междисциплинарного дистанционного практикума, в котором функция автоматической проверки решений задач является подчинённой по отношению к основной функции развития и закрепления умений и навыков, необходимых ИТ-специалисту.

Предложено архитектурное решение дистанционного практикума,

представленное на рисунке 2.

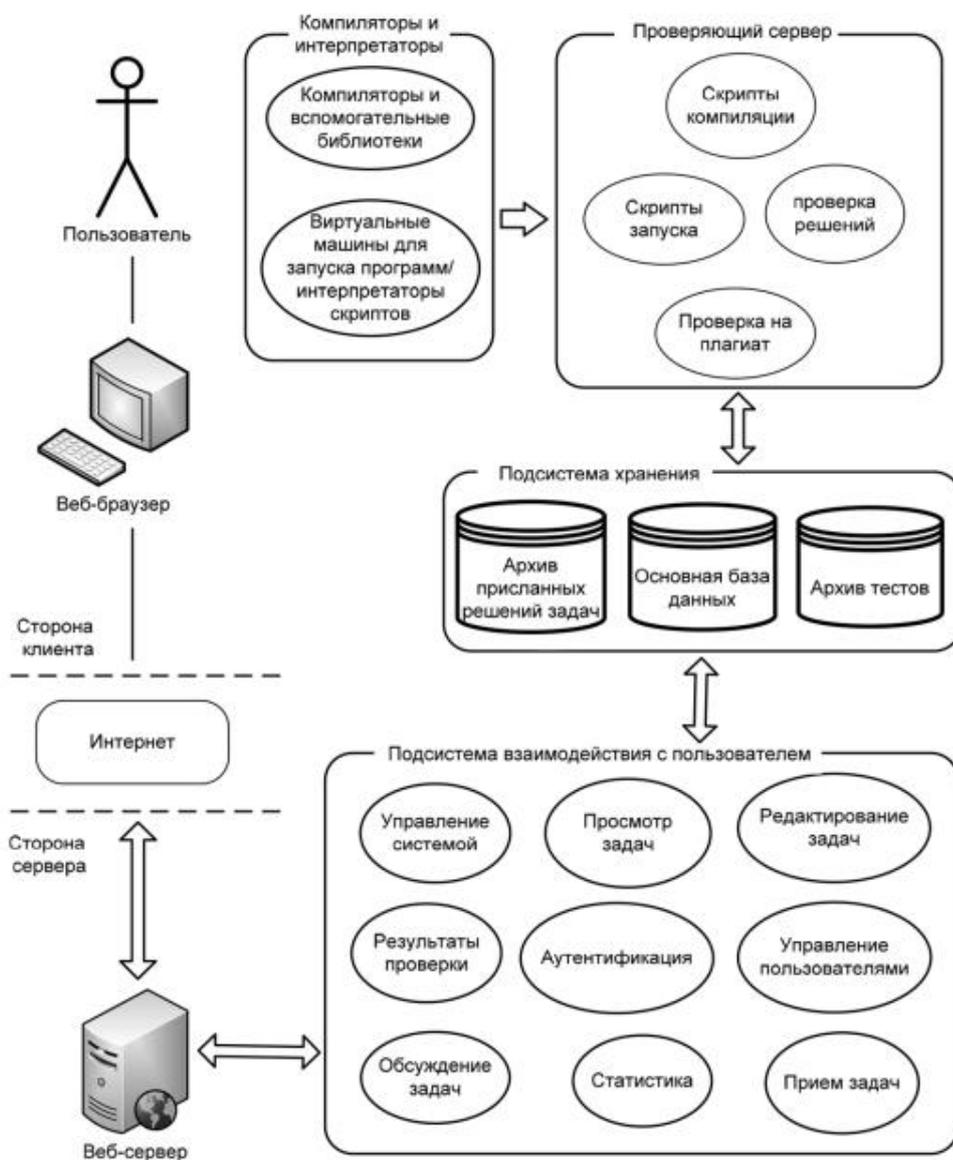


Рисунок 2 – Общая архитектура дистанционного практикума

Представлен подход к организации автоматической проверки путём тестирования программного кода, разработанного студентом в процессе решения учебной задачи, предложено расширение данного подхода для организации автоматической проверки в случаях, когда решением задачи является не программный код, а, например, схема, модель или текстовый документ, оформленные в соответствии с регламентированными правилами.

Результаты исследования показывают, что технологии автоматической

проверки решений учебных задач и программные средства на их основе являются эффективным способом развития профессиональных умений и навыков студентов ИТ-направлений.

Опросы работодателей выпускников ИТ-направлений, проведенные авторами, позволили сделать вывод о наличии явной зависимости между результатами работы студентов, зафиксированными в дистанционном практикуме в процессе обучения, и их успехами в профессиональной деятельности после окончания вуза.

Перязева Ю.В. обосновывает актуальность применения традиционных систем управления обучением (Learning Management System, LMS) в учебном процессе, которые по-прежнему доминируют в образовательных учреждениях.

«Из большого количества существующих платформ автором выделена LMS Moodle, которая в течение многих лет остается популярной.

Востребованность этой системы обусловлена, прежде всего, возможностью расширения функционала с помощью плагинов, над которыми работает сильное сообщество, сформировавшееся за много лет существования платформы.

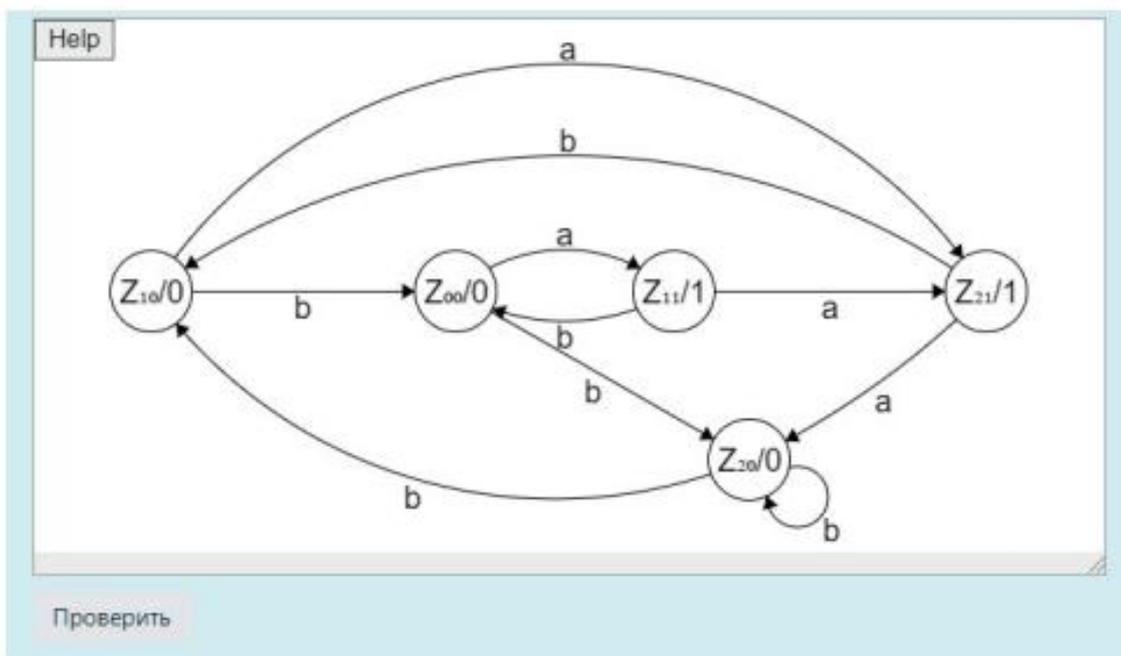
Одним из таких инструментов для организации эффективного автоматического оценивания результатов является плагин CodeRunner.

Наиболее распространенное использование CodeRunner получило на курсах по программированию, где студентов просят написать программный код в соответствии с поставленной задачей, и этот код затем оценивается путем запуска его на серии тестов.

В работе показаны способы расширения областей применения плагина, а, следовательно, и LMS Moodle в целом, за счет создания новых типов вопросов с автоматической проверкой ответов, которые сочетают в себе различные пользовательские интерфейсы ответа, языки программирования для проверки, шаблоны, способы оценивания и вывода результата» [11, 33].

Продемонстрированы возможности применения плагина CodeRunner на

примере разработки заданий для курса компьютерной поддержки дисциплины «Прикладная теория автоматов» (рисунок 3).



	Expected	Got	
✓	$Z_{00}/0$ $Z_{10}/0$ $Z_{11}/1$ $Z_{20}/0$ $Z_{21}/1$	$Z_{00}/0$ $Z_{10}/0$ $Z_{11}/1$ $Z_{20}/0$ $Z_{21}/1$	✓
✗	<pre>['Z00/0', 'Z11/1', 'a'] ['Z00/0', 'Z20/0', 'b'] ['Z10/0', 'Z00/0', 'b'] ['Z10/0', 'Z21/1', 'a'] ['Z11/1', 'Z00/0', 'b'] ['Z11/1', 'Z21/1', 'a'] ['Z20/0', 'Z10/0', 'b'] ['Z20/0', 'Z20/0', 'a'] ['Z21/1', 'Z10/0', 'b'] ['Z21/1', 'Z20/0', 'a']</pre>	<pre>['Z00/0', 'Z11/1', 'a'] ['Z00/0', 'Z20/0', 'b'] ['Z10/0', 'Z00/0', 'b'] ['Z10/0', 'Z21/1', 'a'] ['Z11/1', 'Z00/0', 'b'] ['Z11/1', 'Z21/1', 'a'] ['Z20/0', 'Z10/0', 'b'] ['Z20/0', 'Z20/0', 'b'] ['Z21/1', 'Z10/0', 'b'] ['Z21/1', 'Z20/0', 'a']</pre>	✗

Your code must pass all tests to earn any marks. Try again.

Show differences

Рисунок 3 – Пример вывода результатов автоматической проверки

Аналогичный подход рассматривается в работе [14].

Автор описывает возможности инструмента BookWidgets по созданию интерактивных упражнений и автоматической оценки заданий в рамках

платформы Microsoft Teams for Education – единого цифрового пространства для общения, контента, заданий и приложений, позволяющее преподавателям организовать динамичную учебную среду [20].

Roženel M. и другие представили классификацию автоматизированных средств проверки заданий.

«В зависимости от степени автоматизации оценочные приложения делятся на автоматизированные и частично автоматизированные.

Обе группы используют автоматизированное тестирование программ.

Разница состоит в том, что полностью автоматизированная система тестирует отдельные компоненты и оценивает задание в соответствии с заранее определенными критериями. Обычно такая система используется для небольших заданий.

Для более сложных заданий не всегда возможно оценить все аспекты хорошего программного решения. В этом случае часто используется частично автоматизированная система.

В такой системе оценщик использует частичные результаты автоматизированного тестирования менее сложных частей программы, сосредотачивается на оценке сложных задач и принимает решения о присвоении правильной оценки.

Таким образом, итоговая оценка по-прежнему зависит от преподавателя.

Также описаны подходы к автоматизированной оценке программ.

Обычно все они основаны на двух основных принципах: динамическое тестирование программы или оценка ее кода и структуры» [27].

Оценка программы основана на предположении, что определенные измеримые атрибуты могут быть извлечены из программы и затем сравнены с моделью или решением.

Для использования в образовательных целях результаты сравнения также должны быть подкреплены педагогическими целями курса.

Тестирование программного обеспечения (ПО) можно разделить на две группы в зависимости от того, должна ли программа выполняться в процесс

оценки (динамический анализ) или должна оцениваться статически из программного кода (статический анализ).

1.2 Обзор и анализ разработок информационных систем автоматизированной проверки учебных заданий по составлению SQL-запросов

SQL – это язык структурированных запросов. Он используется во всем мире большинством крупных компаний-вендоров СУБД.

В словаре терминов компании Gartner SQL определяется как «реляционный язык данных, который предоставляет согласованный, ориентированный на ключевые слова английский набор средств для запросов, определения данных, манипулирования данными и управления данными.

Это программный интерфейс к системам управления реляционными базами данных.

SQL является точным, потому что он основан на логике предикатов, но с ним трудно иметь дело обычным пользователям, и его наиболее плодотворное положение - это протокол для взаимодействия программного обеспечения с программным обеспечением, а не для доступа человека к программному обеспечению» [19].

Следует отметить, что курс «Основы языка SQL» является обязательным компонентом программ по подготовке ИТ-специалистов, работающих в области управления данными и анализа данных.

Основой практических занятий данного курса является составление SQL-запросов для доступа, чтения, обработки и анализа данных, хранящихся в базе данных, и получения полезных сведений для управления процессом принятия обоснованных решений.

Рассмотрим работы, посвященные проблеме автоматизированной проверки заданий на составление SQL-запросов.

В работе [30] решается проблема проверки SQL-скриптов на плагиат с

помощью метода обнаружения сходства между фрагментами SQL.

Метод основан на выявлении так называемых «лексем SQL» – постоянных элементов оператора SQL, и «переменных SQL» – легко изменяемых элементов операторов SQL. Таким образом, любые операторы SQL можно заменить так называемым токеном - последовательностью лексем и переменных SQL.

Расстояние между токенами SQL можно рассчитать с помощью алгоритма определения метрики Левенштейна. Малые значения расстояния Левенштейна между токенами позволяют обнаружить операторы SQL, которые были построены путем модификации других операторов.

На рисунке 4 показаны практические результаты фактического применения метода.

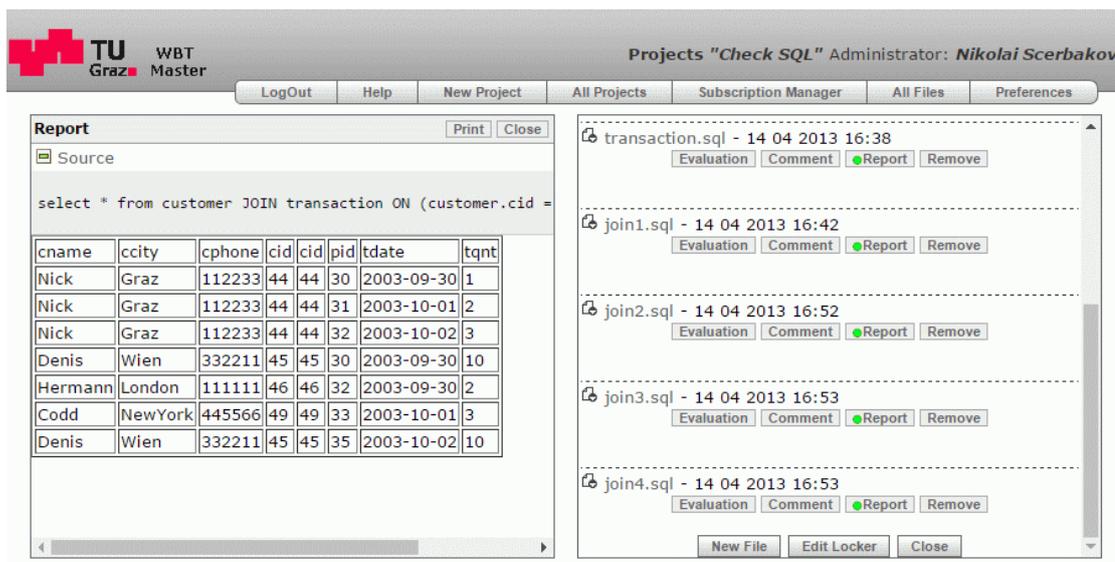


Рисунок 4 – Окно проверки заданий методом лексем SQL

В работе [29] описана управляемая среда обучения ActiveSQL, которая поддерживает практические занятия в режиме онлайн для студентов, изучающих SQL.

Среда обеспечивает автоматическую оценку SQL-запросов.

На рисунке 5 представлен пример результата проверки задания в среде

ActiveSQL.

Using db engine: oracle

Allocated db users: dbro0,dbroa0

```
select * from actor where Name like M%
```

Query Failed for some reason

ORA-00911: invalid character (DBD ERROR: error possibly near <*> indicator at char 37 in 'select * from actor where Name like M<*>% ')

Your answer gives 0 rows, but the sample solution has more than 0 rows

Accuracy Score = 0%

Weighting measures

Initial Weight = 10

Significantly more complex than sample solution (-1)

Hidden Database Check (NOT CHECKED) ([explain](#)) - done ONLY is accuracy is 100% (-3)

Final Weight = 6

Overall Mark = 0%

You have spent 20 minutes and 23 seconds on this question...

Рисунок 5 – Пример окна результата проверки задания в среде ActiveSQL

Среда ActiveSQL также содержит все материалы по оценке практических работ, которые постепенно предоставляются студентам по мере их прохождения через учебный процесс.

Задания автоматически оцениваются без вмешательства обучающего персонала, что дает обучаемым мгновенную обратную связь об их продвижении по материалу и их текущем уровне достижений.

Система разработана таким образом, чтобы свести к минимуму плагиат, одновременно поощряя продвижение по материалам и тайм-менеджменту студентов.

В работе [18] описана платформа CodeGrade, позволяющая создавать задания SQL, которые оцениваются автоматически.

По мнению автора решения, использовать SQL в CodeGrade довольно легко, а автоматизация проверки SQL-запросов интуитивно понятна.

Среди российских интернет-ресурсов можно выделить сайт Sql-ex.ru

[31].

По некоторым данным – это абсолютный лидер по посещаемости среди SQL-тренажеров, который имеет большую базу заданий и десятилетнюю историю.

В настоящее время на рейтинговых этапах разработчики сайта используют СУБД Microsoft SQL Server 2019, а на обучающем этапе - дополнительно СУБД MariaDB-10.4.

Имеющаяся на сайте справка по языку манипуляции данными SQL, выдержанная в соответствии со стандартом, содержит необходимую информацию для изучения языка и выполнения упражнений.

Пример полуавтоматической среды оценки учебных заданий на составление SQL-запросов «SQL-FE» приведен в работе [17].

Диаграмма вариантов использования среды SQL-FE показана на рисунке 6.

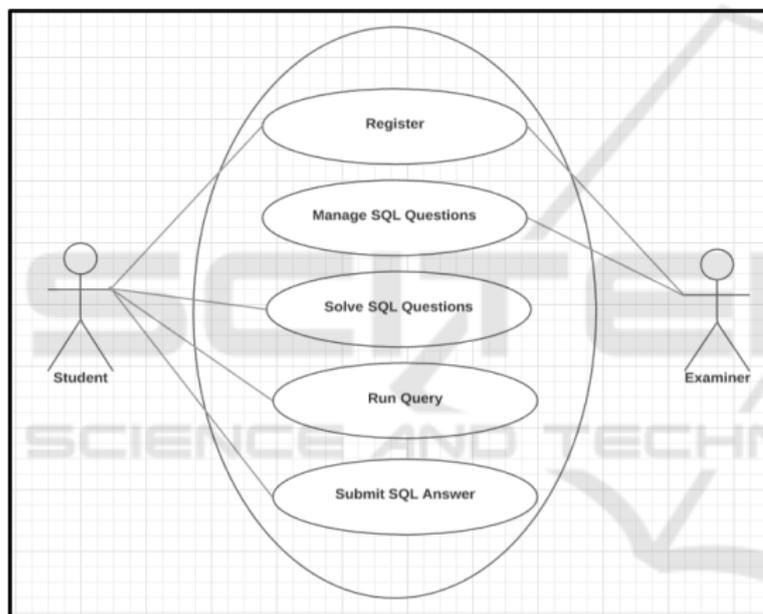


Рисунок 6 – Диаграмма вариантов использования среды SQL-FE

Как отмечается в работе, среда SQL-FE позволяет студентам формулировать операторы SQL, используя метод взаимодействия «укажи и

КЛИКНИ».

В работе описан эксперимент, в котором предлагаемый инструмент сравнивается со средой SQL Management Studio (SSMS).

По мнению автора статьи, приведенные в работе результаты, подтверждают, что использование SQL-FE может положительно сказаться на своевременной формулировке SQL-запросов, и демонстрируют значительное улучшение успеваемости студентов.

На промышленном уровне интересное решение предлагает компания Mail.ru. Так, на платформе Mail.ru Cloud Solution (MCS) появилась интеграция СУБД PostgreSQL с сервисом анализа SQL-запросов [holistic.dev](#) [16].

Сервис [holistic.dev](#) может находить логические ошибки в SQL-запросах.

Например, если выбран неверный источник данных или таблицы связываются по неправильным полям.

Такие ошибки приводят к неправильным результатам. И если отклонения небольшие, их можно не заметить. Но эти результаты могут привести к принятию неправильных бизнес-решений.

Пример использования сервиса [holistic.dev](#) для анализа SQL-запроса показан на рисунке 7.

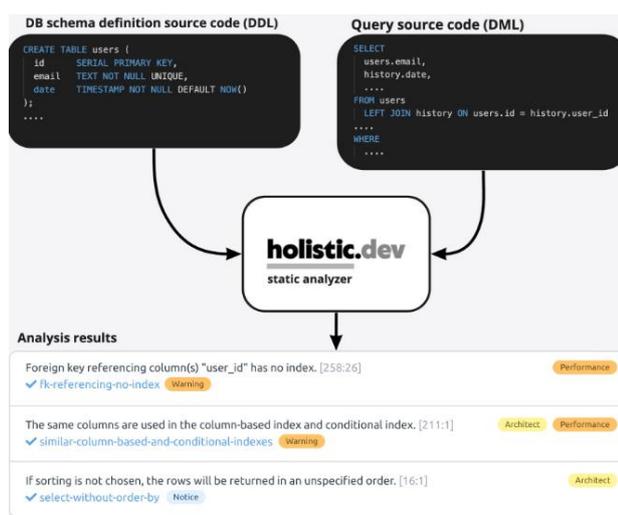


Рисунок 7 – Пример использования сервиса [holistic.dev](#) для анализа SQL-запроса

Представленное решение может быть использовано для проверки заданий по составлению SQL-запросов для СУБД PostgreSQL.

Вместе с тем необходимо констатировать недостаточность работ, посвященных проблеме разработки информационных систем для автоматизированной проверки учебных заданий по составлению SQL-запросов, что подтверждает актуальность темы настоящего исследования.

Выводы по главе 1

В главе 1 магистерской диссертации представлен анализ исследований и разработок в области построения информационных систем автоматизированной проверки учебных заданий по составлению SQL-запросов.

Результаты проделанной работы позволили сделать следующие выводы:

- ручная проверка учебных заданий по составлению запросов на SQL трудоемка и требует много времени. Кроме того, разные СУБД могут иметь разный синтаксис операторов SQL. Это может привести к ошибкам, которые снижают эффективность процесса подготовки ИТ-специалистов;
- в зависимости от степени автоматизации оценочные приложения делятся на автоматизированные и частично автоматизированные;
- результаты, представленные в работах по теме исследования, подтверждают существенное сокращение времени автоматизированного оценивания по сравнению с выставлением оценок вручную;
- необходимо констатировать недостаточность работ, посвященных проблеме разработки информационных систем для автоматизированной проверки учебных заданий по составлению SQL-запросов, что подтверждает актуальность темы настоящего исследования.

Глава 2 Анализ методологий построения информационных систем автоматизированной проверки учебных заданий по составлению SQL-запросов

2.1 Модели и алгоритмы систем автоматизированной проверки учебных заданий по составлению SQL-запросов

Как показал анализ, для автоматической проверки на правильность составления SQL-запросов используются следующие типы заданий:

- модификация данных (операторы UPDATE, DELETE);
- выборка данных (оператор SELECT).

Рассмотрим подходы, модели и алгоритмы, которые используются при построении информационных систем автоматизированной проверки SQL-запросов для каждого типа задания.

Модель обучающей системы для автоматической проверки SQL-запросов на модификацию данных представлена на рисунке 8 [10].

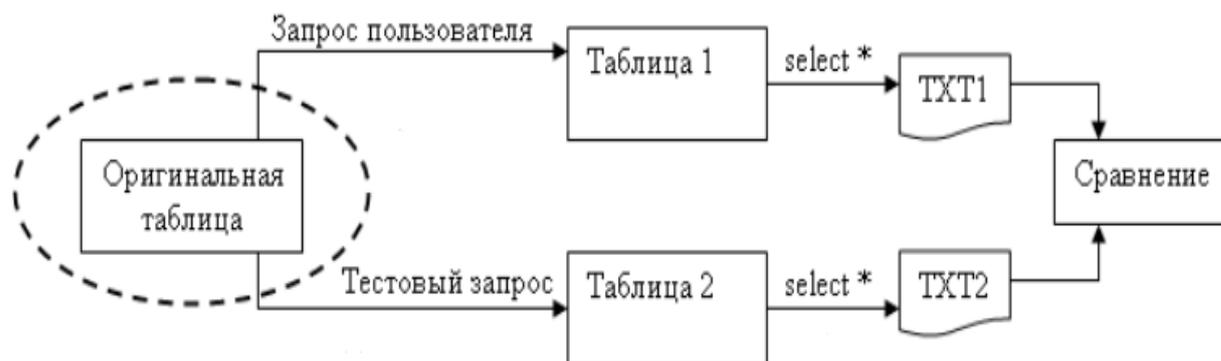


Рисунок 8 – Структурная схема обучающей системы для проверки SQL-запросов на модификацию данных

Авторы статьи рассматривают следующие методы реализации процесса проверки упражнений:

- ROLLBACK-метод: выполнение запросов пользователя на оригинальных таблицах в рамках транзакции, которая завершается

откатом после возвращения данных пользователю;

- триггер-метода: использование на оригинальных таблицах специальным образом составленные триггеры типа «INSTEAD OF»;
- метод временных таблиц: создание для пользователя временных таблицы – копий таблиц, которые подвергаются модификации. По завершении транзакции временные таблицы автоматически удаляются;
- метод постоянных таблиц: по мере необходимости для каждого пользователя создаются полноценные таблицы-копии оригинальных таблиц.

Как показывает практика, более востребованными являются информационные системы проверки заданий на выборку данных.

Это можно объяснить тем, что вопросы, связанные с реализацией запросов на модификацию данных, как правило, рассматриваются в рамках курсов по обучению программированию на конкретном языке высокого уровня.

Вместе с тем, приобретение профессиональных компетенций, связанных с применением языка SQL для анализа данных, очень актуально для бизнес-аналитиков.

Автоматическая проверка заданий на составление SQL-запросов на выборку данных организуется, как правило, по следующему алгоритму: система последовательно выполняет запрос обучаемого и эталонный запрос, сравнивает полученные результаты и формирует оценку на основе результатов сравнения.

При всей простоте такого решения, здесь возможны некоторые проблемы.

Так, в работе [4] отмечается, что главной проблемой построения информационной системы (ИС) проверки заданий по выбору данных из базы данных (БД), является отсутствие защиты от ввода запроса, который не соответствует изучаемой теме, но выдает правильный результат.

Рассмотрим следующий пример.

Дана модель данных предметной области «Университет», представленная на рисунке 9.

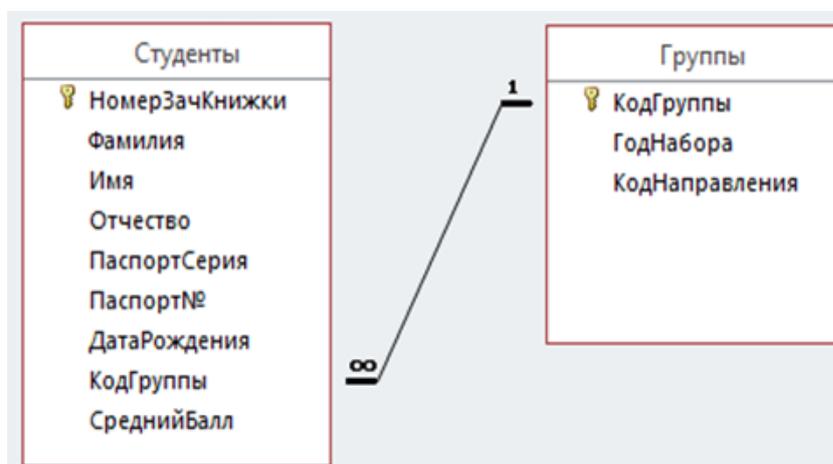


Рисунок 9 – Модель данных предметной области «Университет»

Задание 1: Создать запрос для выбора студентов по году набора.

Здесь возможны два варианта ответа, представленные соответственно на рисунке 10.

```
1) SELECT Студенты.Фамилия, Студенты.Имя, Студенты.Отчество,
Студенты.КодГруппы, Группы.ГодНабора FROM Группы INNER JOIN
Студенты ON Группы.КодГруппы = Студенты.КодГруппы WHERE
Группы.ГодНабора=2020

2) SELECT Студенты.Фамилия, Студенты.Имя, Студенты.Отчество,
Студенты.КодГруппы, Группы.ГодНабора FROM Группы, Студенты WHERE
Группы.КодГруппы = Студенты.КодГруппы AND Группы.ГодНабора=2020
```

Рисунок 10 – Варианты ответов на задание 1

Важно отметить, что оба запроса выдают одинаковый правильный

результат.

Однако, если изучаемой темой является конструкция INNER JOIN, то второй вариант считается неверным.

Чтобы избежать подобных ситуаций, необходимо ввести дополнительную проверку синтаксиса запроса. В результате неудовлетворительного результата проверки будет выведено соответствующее сообщение, и запрос, возвращающий правильный ответ, но не удовлетворяющий всем требованиям задания, будет оценен как неверный.

Предлагается алгоритм проверки заданий на выборку данных, представленный на рисунке 11.

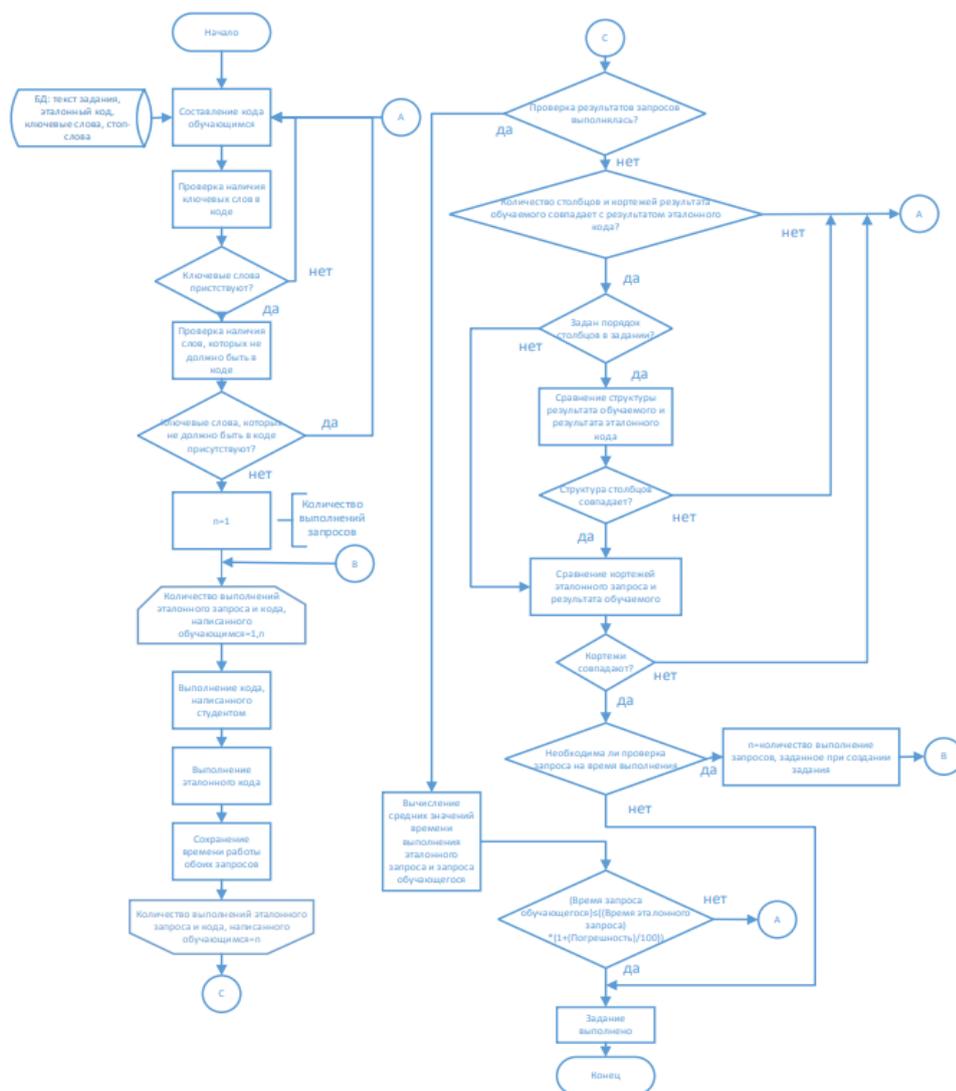


Рисунок 11 - Алгоритм проверки заданий на выборку данных

Алгоритм реализует метод решения, использующий проверку запроса на наличие стоп-слов, запрещенных к применению в тексте запроса.

Преподаватель вручную вводит стоп-слова при подготовке задания запроса к публикации.

Например, если изучается тема «Использование операции соединения типа INNER JOIN», преподаватель вводит ключевые слова «inner, join».

Семенова З.В. и другие дают описания «наиболее часто встречающихся уязвимостей систем проверки правильности составления SQL-запросов. Показано, что целесообразно в качестве основного инструмента закрытия таких уязвимостей, как «Обход решения» использовать теневую базу данных. Для закрытия такой уязвимости, как «Нечувствительность к порядку полей в команде SELECT», используется механизм курсоров в связке с созданием и сравнением словарей вида «Ключ-Значение» [13]. Архитектура системы правильности проверки составления SQL-запросов представлена на рисунке 12.

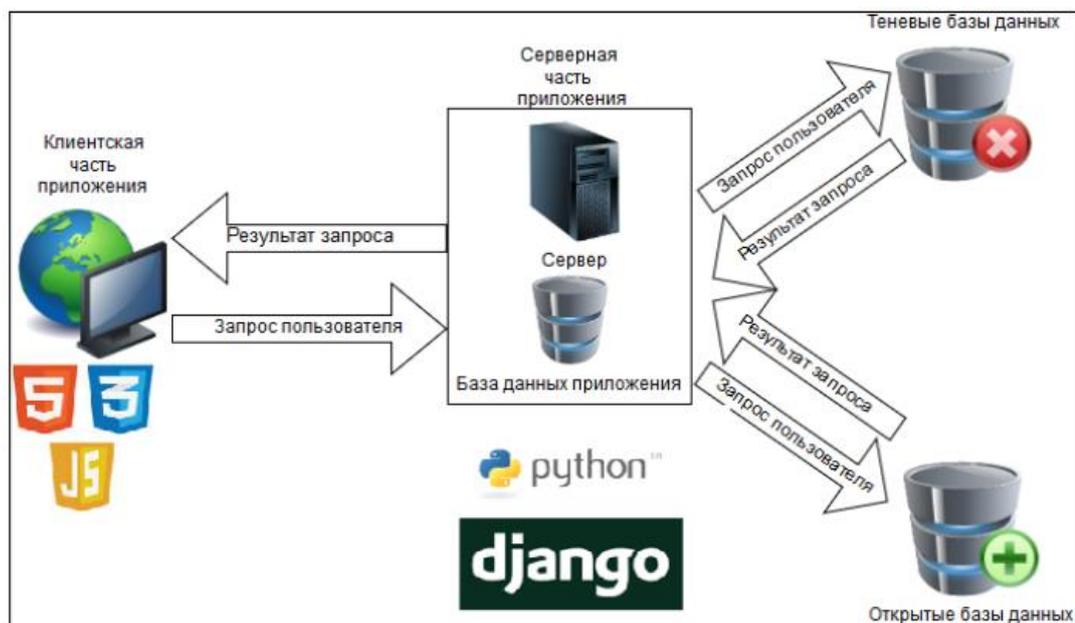


Рисунок 12 - Архитектура системы правильности проверки составления SQL-запросов

В работе [8] предлагается механизм проверки правильности студенческих заданий с помощью алгоритма сравнения с выполнением операции разности реляционной алгебры.

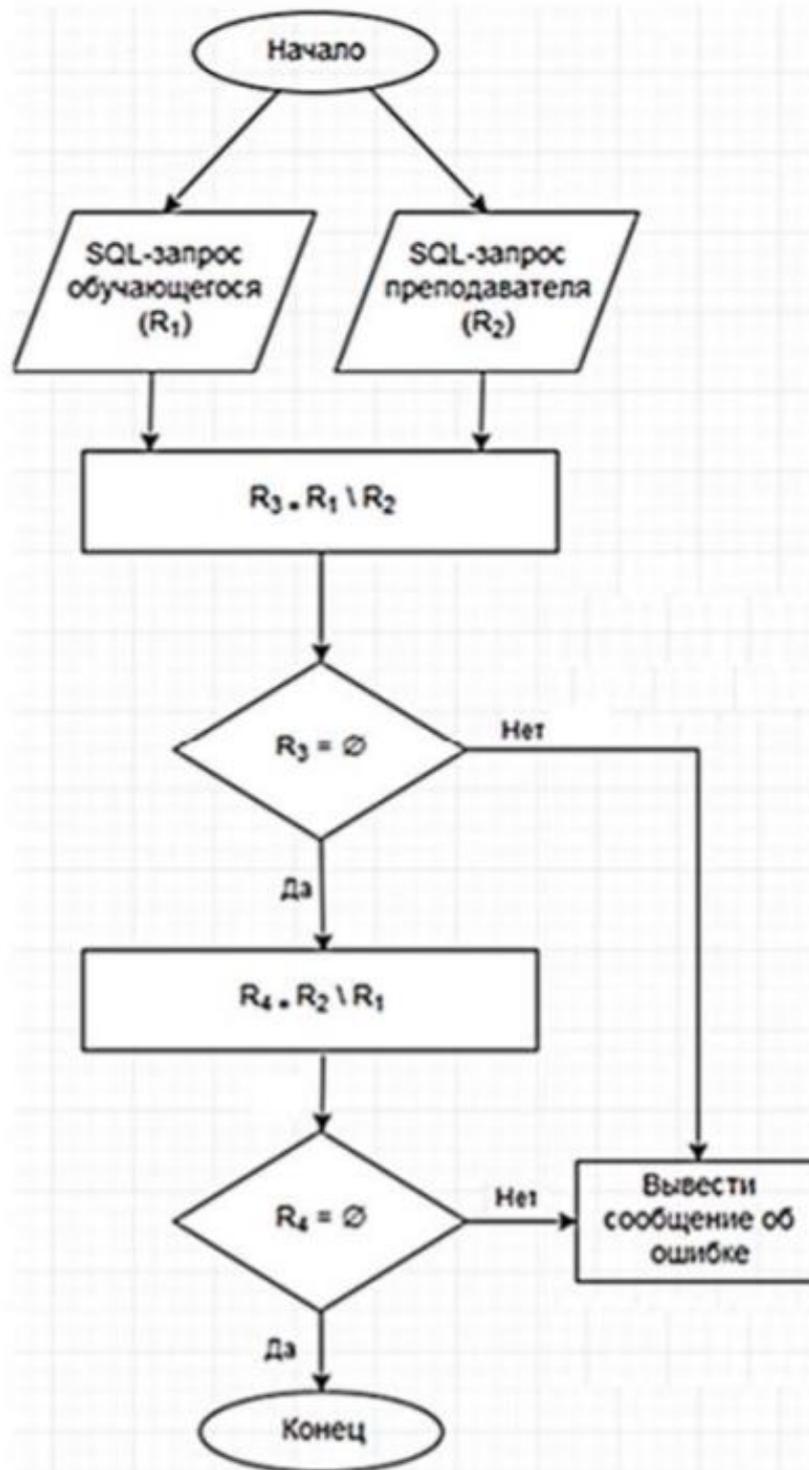


Рисунок 12 - Алгоритм сравнения с выполнением операции разности реляционной алгебры

В данном алгоритме используется операция реляционной алгебры – разность отношений R_1 и R_2 .

В работе [25] представлена разработка автоматизированной системы оценки заданий SQL с использованием метода объектно-ориентированного проектирования и фреймворка «Модель-Представление-Контроллер».

Предлагаемая система ASQLAG состоит из двух основных частей: управления заданиями и автоматического оценивания SQL (рисунок 13).

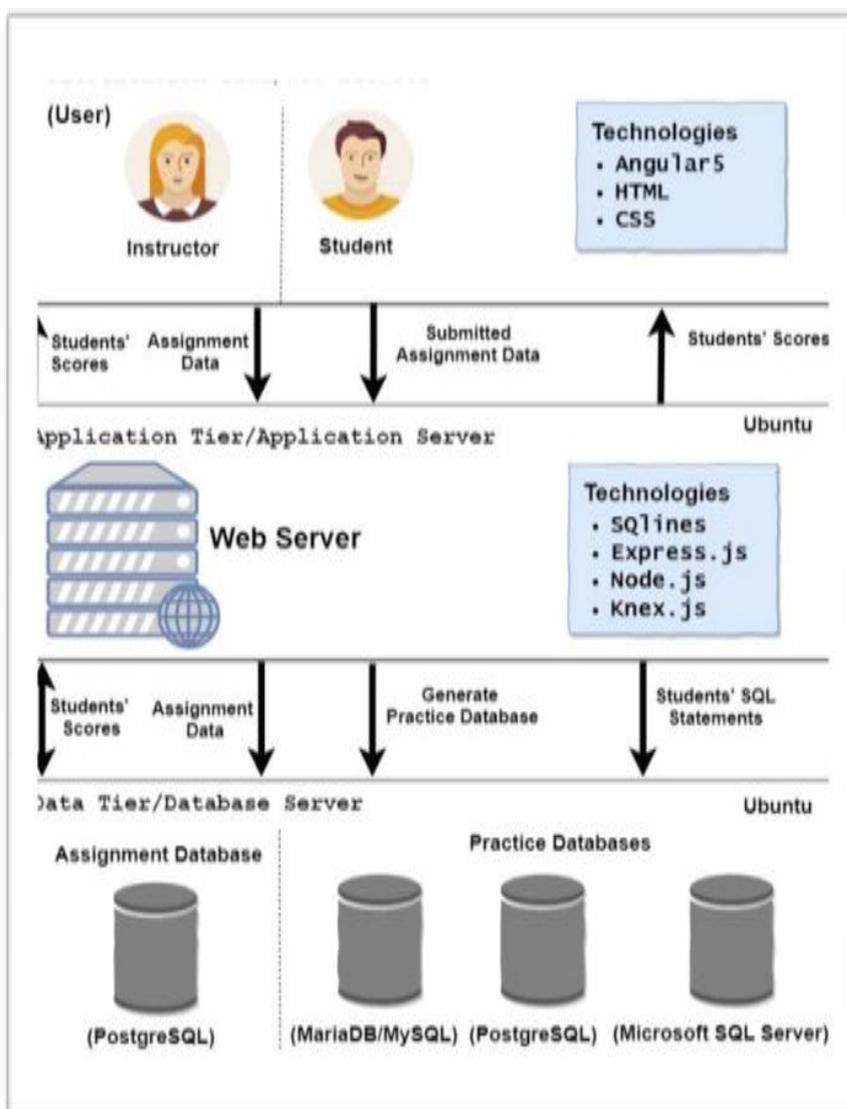


Рисунок 13 – Архитектура системы ASQLAG

Преподаватели могут просто управлять своими заданиями и информацией о студентах в любое время и в любом месте через Интернет.

Автоматизированный классификатор SQL разработан для поддержки четырех СУБД: MariaDB, MySQL, PostgreSQL и Microsoft SQL Server.

Студенты могут произвольно выбрать одну из четырех поддерживаемых СУБД, чтобы практиковать свои навыки SQL, и их задания могут быть представлены в одном из поддерживаемых синтаксисов СУБД.

В работе реализован и протестирован автоматизированный SQL-классификатор.

Результаты показывают, что время автоматизированного оценивания по сравнению с выставлением оценок вручную сократилось со многих часов до нескольких секунд на одно задание.

Баллы, полученные в результате проверки заданий, послужили хорошей мотивацией студентов и преподавателей для улучшения процесса обучения.

Исследование [26] представляет собой оригинальный инструмент для автоматической оценки SQL-операторов с использованием MySQL Sandbox.

Этот инструмент позволяет студентам создавать и выполнять запросы в режиме реального времени. По мнению автора, существующие алгоритмы автоматической оценки SQL имеют ограничения на поддерживаемые операторы и понятные уведомления об ошибках.

В статье представлен оригинальный алгоритм, который может поддерживать оценку операторов языка манипулирования данными, то есть INSERT, UPDATE и DELETE, а также команд языка определения данных: CREATE, ALTER и DROP.

Алгоритм автоматизированной проверки по составлению SQL-запросов для определения данных (DDL) представлен на рисунке 14.

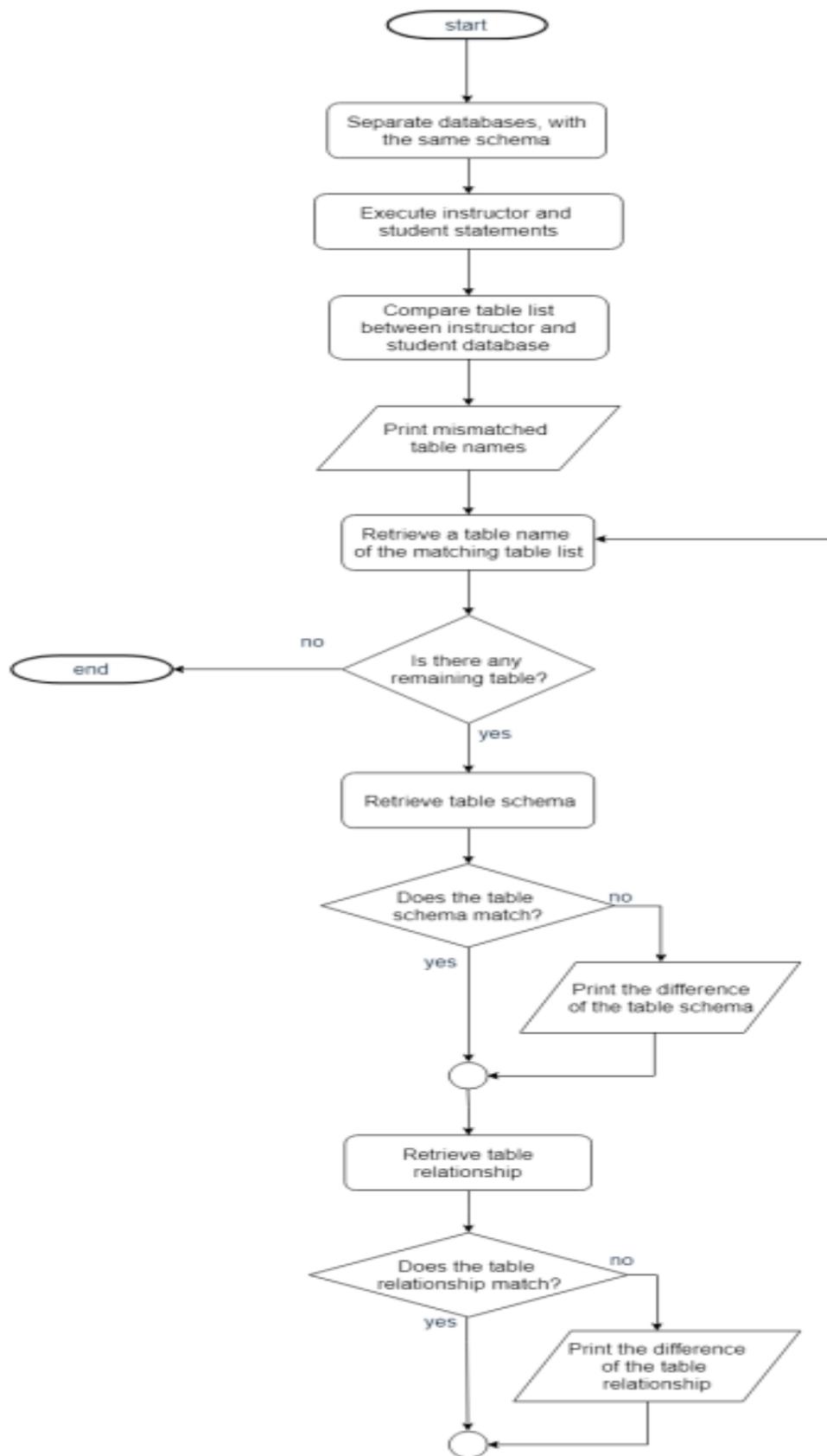


Рисунок 14 – Алгоритм автоматизированной проверки по составлению SQL-запросов для определения данных

Анализ вышеописанных моделей и алгоритмов информационных систем автоматизированной проверки учебных заданий позволил сделать следующие выводы:

- наиболее востребованной и сложной является автоматизированная проверка заданий на выборку данных, что объясняется относительной сложностью инструкции SELECT;
- базовым алгоритмом проверки запросов на выборку является алгоритм сравнения результатов выполнения проверяемого и эталонного SQL-запроса;
- одной из задач, которая решается в процессе моделирования информационной системы автоматизированной проверки учебных заданий (далее - ИСАПЗ), является устранение проблемы неоднозначности SQL-скриптов и связанных с ней уязвимостей ИСАПЗ;
- в некоторых ИСАПЗ используются алгоритмы проверки заданий на составление запросов для различных диалектов SQL.

Следует отметить, что определенная тенденция к расширению функциональных возможностей ИСАПЗ и связанное с ней усложнение их решений объясняют некоторые ограничения по применению таких систем в процессе обучения ИТ-специалистов.

Так, изучение диалектов языка SQL целесообразно при подготовке ИТ-специалистов по работе с конкретными СУБД, что не является обязательным для вузовских дисциплин.

Более дидактически обоснованным представляется изучение основ SQL, синтаксис которого мало отличается от стандартного языка структурированных запросов ANSI. К таким диалектам относится, например, язык Transact-SQL или T-SQL, используемый в СУБД MS SQL Server [32].

Проверка ответов студентов на практические задания на оригинальность целесообразна для сложных запросов с применением инструкции SELECT,

что, как правило, применяется в обучающем процессе ограниченного количества дисциплин (например, связанных с бизнес-анализом).

Существенным недостатком рассмотренных решений является сложность адаптации используемых в них моделей и алгоритмов к специфике процесса подготовки ИТ-специалистов в конкретной обучающей организации.

В этой связи представляет интерес разработка ИСАПЗ, которая может быть использована для подготовки ИТ-специалистов в различных обучающих организациях.

2.2 Методологические подходы к построению систем автоматизированной проверки учебных заданий по составлению SQL-запросов

2.2.1 Методология объектно-ориентированного проектирования

Как показал анализ, основной методологией, которая используется при построении систем автоматизированной проверки учебных заданий, является методология объектно-ориентированного анализа и проектирования, основанная на принципах объектно-ориентированного подхода (ООП) и языке визуального моделирования UML [12].

Данная методология используется на стадии логического проектирования ИСАПЗ.

Логическое проектирование определяет функции и особенности информационной системы и отношения между ее компонентами.

Логическое проектирование включает выходные данные, которые должны быть произведены информационной системой, входные данные, необходимые для системы, и процесс, который должен быть произведен системой, независимо от того, как задачи будут выполняться физически.

Логическое проектирование определяет то, что должно происходить в ИС, а не то, как это должно быть выполнено.

Другими словами, логические проекты не касаются реальных методов

реализации. При подготовке логического проекта ИС системный аналитик определяет потребности пользователя на уровне детализации, который фактически определяет поток информации в систему и из нее и необходимые источники данных. Результатом логического проектирования является логическая модель ИС.

Логическое моделирование – это обязательный процесс перед написанием программного кода.

Логическое моделирование является неотъемлемой частью крупных программных проектов, а также полезно для средних и даже небольших проектов.

Понятие логической модели ИС тесно связано с ее логической архитектурой. Следует добавить, что одной из основных задач логического моделирования является разработка логической модели данных ИС.

В процессе логического моделирования широко применяются паттерны (шаблоны) проектирования.

Одним из таких шаблонов является паттерн «Модель-Представление-Контроллер» (Model-View-Controller, MVC).

«MVC указывает, что приложение состоит из модели данных, информации представления и информации управления. Шаблон требует, чтобы каждый из них был разделен на разные объекты.

Модель (например, информация о данных) содержит только чистые данные приложения. Она не содержит логики, описывающей, как представить данные пользователю.

Представление (например, информация о презентации) представляет пользователю данные модели. Представление знает, как получить доступ к данным модели, но не знает, что означают эти данные или что пользователь может сделать, чтобы ими манипулировать.

Наконец, контроллер (например, управляющая информация) существует между представлением и моделью. Он прослушивает события, инициированные представлением (или другим внешним источником), и

выполняет соответствующую реакцию на эти события.

В большинстве случаев реакция заключается в вызове метода модели.

Поскольку представление и модель связаны через механизм уведомления, результат этого действия автоматически отражается в представлении» [22].

Большинство приложений сегодня следуют этому шаблону, многие с небольшими вариациями. Например, некоторые приложения объединяют представление и контроллер в один класс, потому что они уже очень тесно связаны.

Все варианты настоятельно рекомендуют разделение данных и их представление. Это не только упрощает структуру приложения, но и позволяет повторно использовать код.

На рисунке 15 показано, как шаблон проектирования MVC применяется к HCL Commerce. Этот шаблон используется как для веб-приложений, так и для полнофункциональных клиентов и может использовать инфраструктуру веб-сервисов.

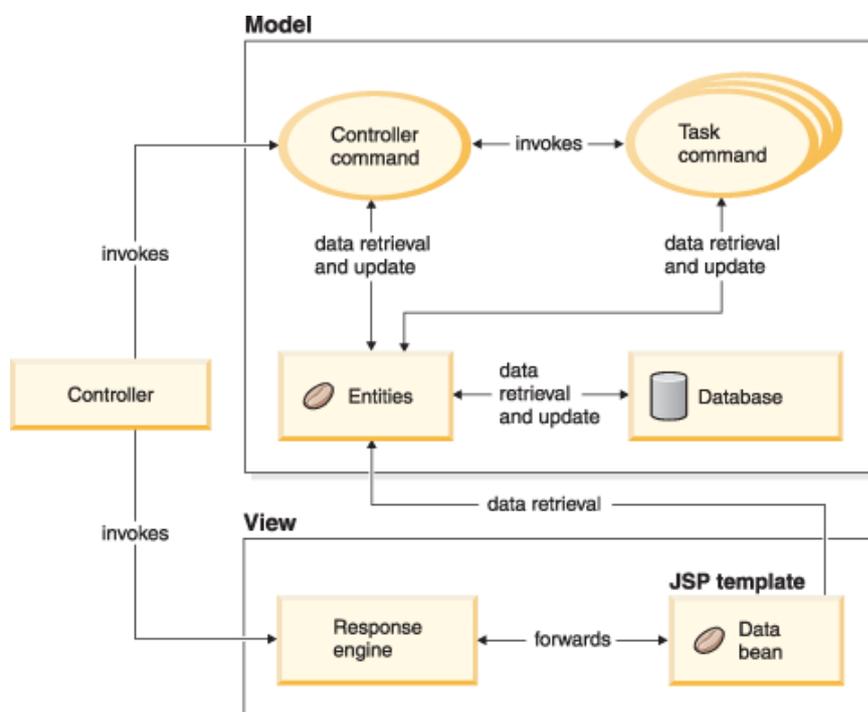


Рисунок 15 – Пример применения паттерна MVC

Пример применения MVC при разработке программного обеспечения ИСАПЗ рассмотрен в работе [25].

Здесь ООП с фреймворком MVC используется для создания диаграммы классов, содержащей классы модели. Большинство классов моделей - это классы с данными, которые должны храниться в базе данных.

Затем классы модели можно использовать в качестве входных данных на этапе проектирования базы данных.

После завершения этапа разработки программы создаются пользовательские интерфейсы, спецификации вариантов использования, диаграммы последовательности и диаграмма классов.

Диаграмма классов, полученная на этом этапе, состоит из трех типов классов: класса модели, класса представления и класса контроллера (рисунок 16).

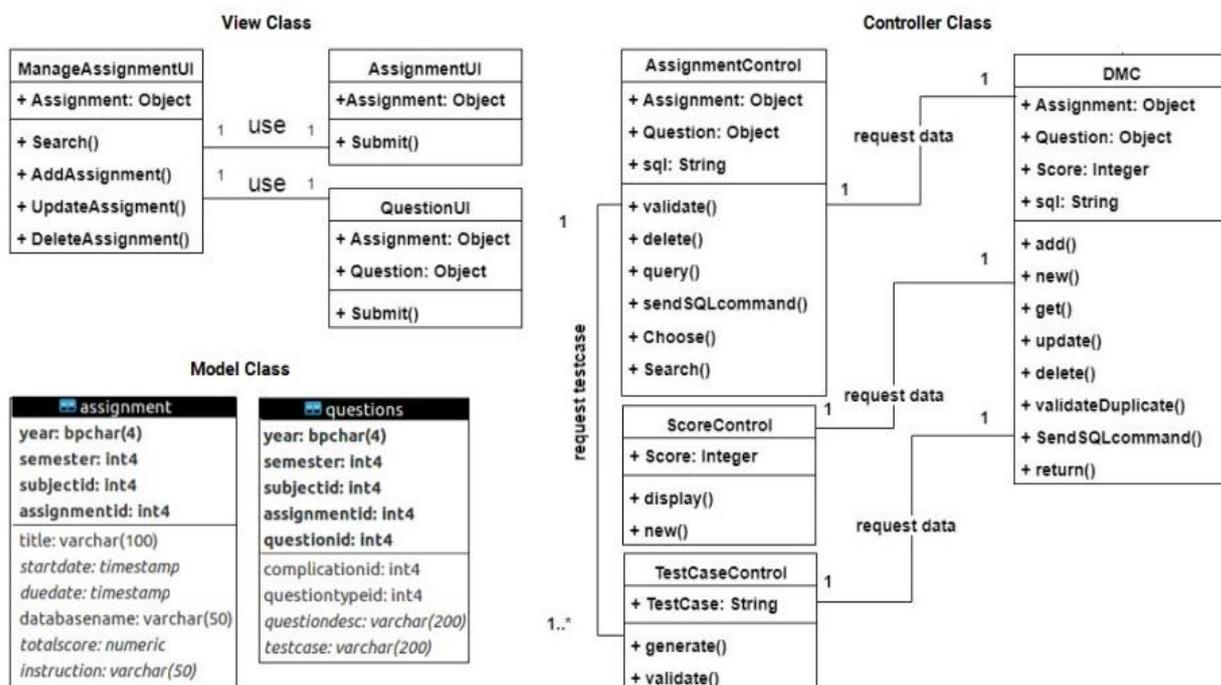


Рисунок 16 – Диаграмма классов ИСАПЗ на основе фреймворка MVC

Эти классы выполняют свои конкретные роли.

Смена классов модели не повлияет друг на друга и наоборот. Позже классы модели используются для выполнения проектирования базы данных.

Роли этих классов:

- класс модели: этот класс предназначен для управления данными;
- класс представления: этот класс предназначен для управления пользовательским интерфейсом. Он принимает данные, вводимые пользователем, и отображает данные, отправленные обратно из класса контроллера.
- класс контроллера: этот класс взаимодействует с классом представления и классом модели. Он обрабатывает вводимые пользователем данные, отправленные из класса представления, выполняет соответствующую логику приложения, отправляет запрос на получение необходимых данных модели и возвращает соответствующие ответы классу представления.

Шаблон «Модель-Представление-Контроллер» помогает создавать несколько представлений, сохраняя неизменными модель и контроллер.

2.2.2 Методология объектно-структурного моделирования

Если рассматривать системы автоматизированной проверки учебных заданий как разновидность систем управления процессом обучения, то при выборе методологического подхода к их построению следует использовать методологии, основанные на интеграции различных методологических подходов.

Одной из таких методологий является методология объектно-структурного подхода.

В работе [21] предлагается методология объектно-структурного подхода к построению информационных систем управленческого учета (ИСУУ), организованных по модели Workflow, которая состоит из нижеследующих стадий.

I. Объектно-структурное моделирование ИСУУ.

«Объектно-структурная модель (ОСМ) системы N-этапной обработки учетной информации имеет вид ориентированного графа $O(W,S,D)$, изображенного на рисунок 17.

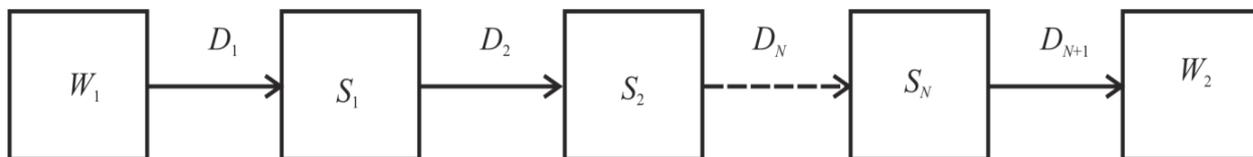


Рисунок 17 – Объектно-структурная модель системы N-этапной обработки учетной информации

На рисунке:

$W = \{W_1, W_2\}$ – узлы, обозначающие объекты класса “Склад”.

$S = \{S_1, S_2, \dots, S_N\}$ – множество узлов, обозначающие объекты класса “Этап (Склад+Агрегат+Контролер)”.

$D = \{D_1, D_2, \dots, D_{N+1}\}$ – множество дуг, нагруженных элементами информационного потока.

Физическим представлением ОСМ является бизнес-транзакция.

II. Формализация элементов ОСМ.

Заключается в представлении объектно-структурной модели ИСУУ в виде системы взаимодействующих автоматов, управляющих статусом обрабатываемого элемента материального (информационного) потока, который описывается как конечный автомат (КА) в соответствии с его жизненным циклом, представленным в таблице 1» [21].

Таблица 1 – Жизненный цикл продукта в многоэтапном производстве

Статус	Описание
1	Сырье
2	Полуфабрикат
3	Готовая продукция

«III. Разработка UML паттернов проектирования ИСУУ.

Паттерны проектирования строятся в нотации UML на основе объектных моделей конечного автомата, сгруппированных по суперклассам - классам технологической онтологии конкретной предметной области» [21].

Примеры паттернов проектирования ИСУУ представлены на рисунке 18.

WAREHOUSE	AGGREGATE	CONTROLLER
ID itemBalance	ID itemStatus	ID itemStatusControlResult
+receiptItem() +expenseItem()	+changeItemStatus()	+controlItemStatus()

Рисунок 18 – Паттерны проектирования ИСУУ

«Преимуществами предлагаемой методологии проектирования ИСУУ являются:

- универсальность объектно-структурных моделей ИСУУ;
- простота адаптации объектно-структурных моделей ИСУУ к специфике управленческого учета в конкретной организации;
- простота интеграции ИСУУ в корпоративную информационную систему (КИС) предприятия» [21].

Ввиду того, что процесс проверки учебного задания можно рассматривать как Workflow, методология объектно-структурного моделирования может быть использована в процессе логического проектирования ИСАПЗ.

Выводы по главе 2

В главе 2 магистерской диссертации представлен анализ методологий построения информационных систем автоматизированной проверки учебных заданий по составлению SQL-запросов.

Результаты проделанной работы позволили сделать следующие выводы:

- для автоматической проверки на правильность составления SQL-запросов используются задания на модификацию и выборку данных, соответственно;
- базовым алгоритмом проверки запросов на выборку является алгоритм сравнения результатов выполнения проверяемого и эталонного SQL-запросов;
- тенденция к расширению функциональных возможностей ИСАПЗ и связанное с ней усложнение их решений обуславливают некоторые ограничения по применению систем, построенных на основе рассмотренных моделей, в процессе подготовки ИТ-специалистов в различных обучающих организациях;
- в процессе логического моделирования ИСАПЗ применяется методология ООП и паттерны проектирования. Одним из таких паттернов является фреймворк «Модель-Представление-Контроллер» (Model-View-Controller, MVC).

Ввиду того, что процесс проверки учебного задания можно рассматривать как Workflow, паттерны проектирования методологии объектно-структурного моделирования могут быть также использованы в процессе логического проектирования ИСАПЗ.

Глава 3 Разработка информационной системы автоматизированной проверки учебных заданий на составление SQL-запросов

Существует два подхода автоматизированного оценивания заданий на составление SQL-запросов:

- проверка синтаксиса оператора SQL;
- проверка выходных данных после выполнения оператора SQL.

Первый метод оценки подходит для операторов SQL, имеющих только один стиль и синтаксис.

Второй метод оценки подходит для операторов SQL с небольшим количеством выходных данных.

Однако некоторые операторы SQL могут быть написаны во многих стилях.

Кроме того, как было отмечено выше, при практическом использовании различных СУБД мы не можем избежать использования нескольких стилей и синтаксисов операторов SQL.

Поэтому в разрабатываемой системе используется второй метод оценки.

3.1 Алгоритмы автоматической проверки заданий на составление SQL-запросов

3.1.1 Алгоритм проверки заданий на составление SQL-запросов модификации данных

Блок-схема алгоритма проверки SQL-запросов модификации данных, включающий операторы INSERT, UPDATE и DELETE, показан на рисунке 19.

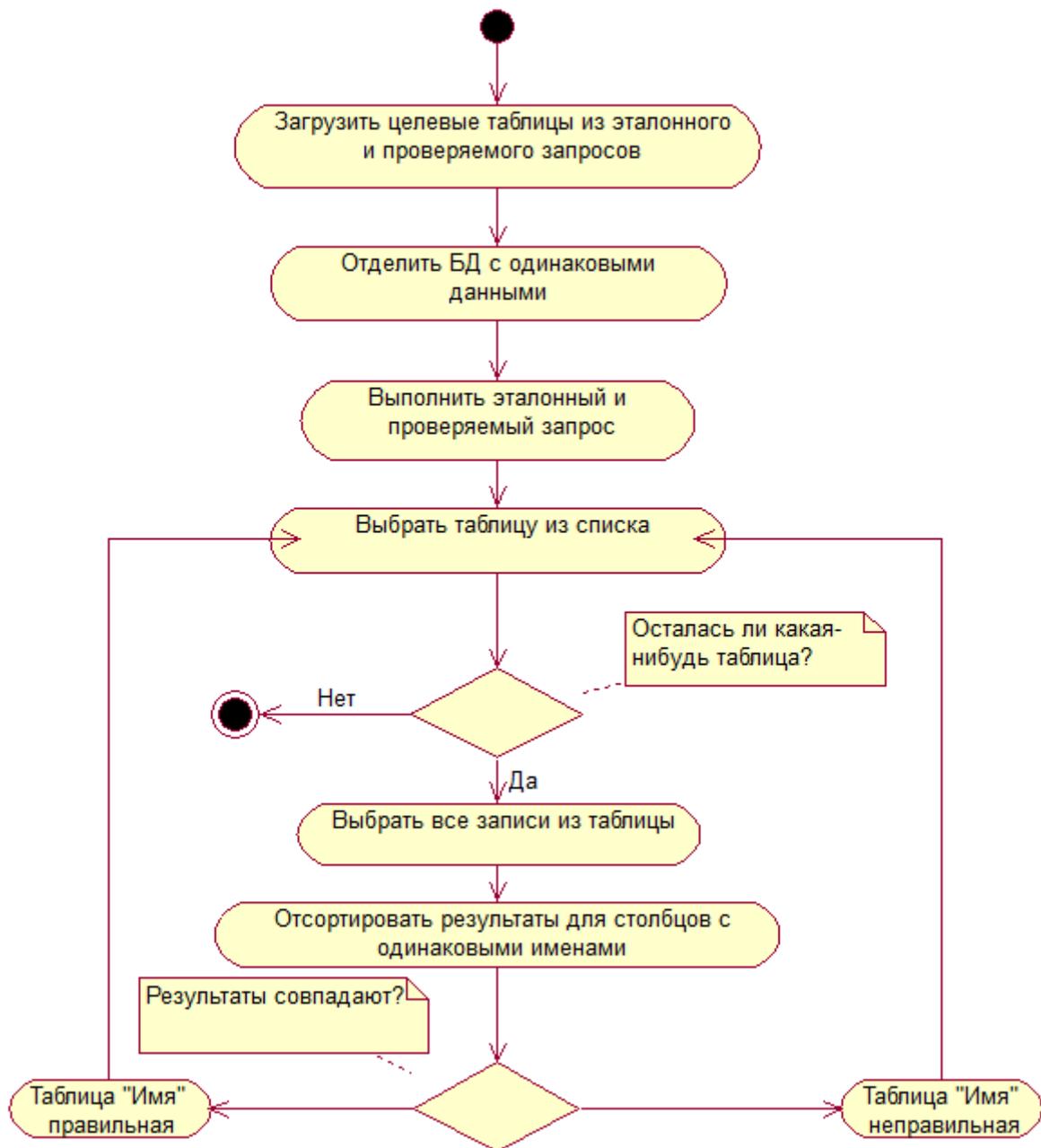


Рисунок 19 – Алгоритм проверки SQL-запросов модификации данных

Здесь эталонный запрос – это запрос, созданный преподавателем.

Проверяемый запрос – это запрос, созданный студентом.

Алгоритм начинается с создания списка целевых таблиц преподавателя и студента, которые извлекаются из утверждений преподавателя и учащегося соответственно.

Затем базы данных, в которых хранятся одни и те же наборы данных,

разделяются для выполнения инструкций студента и преподавателя.

Позже все записи извлекаются из целевых таблиц в каждой базе данных.

Наборы результатов из базы данных студента сравниваются с наборами результатов из базы данных преподавателя для проверки правильности.

Наборы результатов могут включать, вероятно, одни и те же записи, но их последовательность различается.

При оценке оператора DML алгоритм игнорирует последовательность строк результатов. Поэтому необходимо получить и сравнить отсортированные строки.

Запрос студента является правильным, если результирующие таблицы преподавателя и студента после выполнения совпадают.

В противном случае запрос студента считается неправильным.

3.1.2 Алгоритм проверки заданий на составление SQL-запросов выборки данных

Алгоритм проверки SQL-запросов выборки данных, включающий оператор SELECT, состоит из следующих шагов:

Шаг 1. Начало.

Шаг 2. Загрузить из базы данных текст задания, описание соответствующей учебной базы данных.

Шаг 3. Загрузить SQL-скрипт запроса студента.

Шаг 4. Выполнить эталонный запрос и запрос, созданный студентом, соответственно.

Шаг 5. Сравнить результаты выполнения запросов. Если наборы кортежей результата выполнения эталонного запроса и запроса студента совпадают, то выдать сообщение «Задание выполнено правильно», в противном случае – выдать сообщение «Задание выполнено неправильно».

Шаг 6. Конец.

Блок-схема алгоритма проверки SQL-запросов выборки данных показана на рисунке 20.

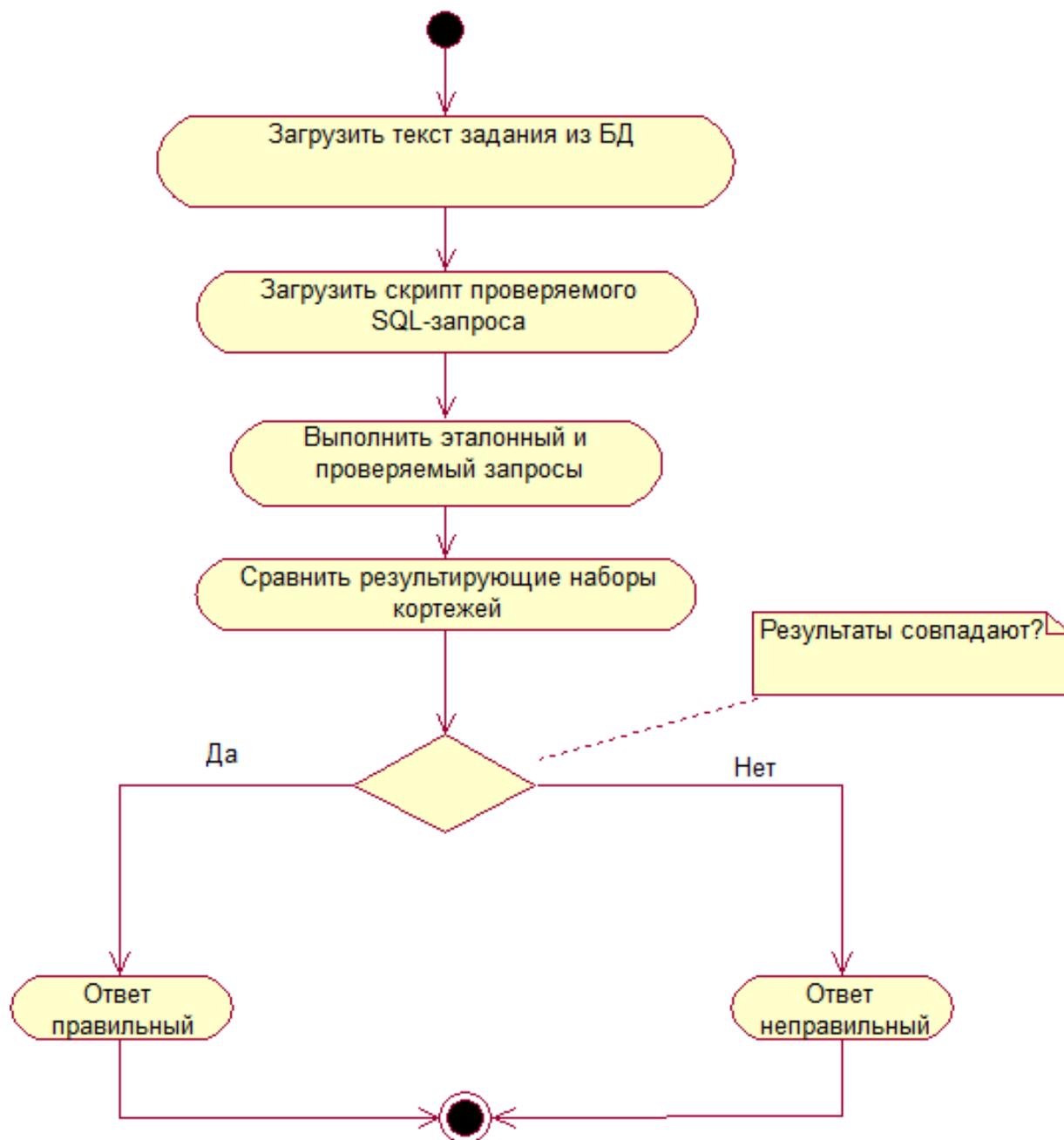


Рисунок 20 – Алгоритм проверки SQL-запросов выборки данных

Следует отметить, что в представленных алгоритмах отсутствуют проверки на оригинальность SQL-скриптов.

Это существенно упрощает реализацию алгоритмов, не снижая дидактические возможности систем автоматизированной проверки учебных заданий на составление SQL-запросов, используемых в вузах.

3.2 Логическое моделирование информационной системы автоматизированной проверки учебных заданий на составление SQL-запросов

Как было отмечено выше, при построении логической модели ИС используется методология объектно-ориентированного проектирования.

Следует констатировать, что при реализации проектов разработки программного обеспечения на основе ООП наилучшие результаты достигнуты при применении методологии RUP и CASE-средства Rational Rose.

«Методология RUP (Rational Unified Process) – это гибкая методология разработки программного обеспечения.

В методологии RUP для успешного процесса разработки необходимы три составляющие: процесс, нотация и набор утилит. Процесс описывает, что мы делаем, в каком порядке и каким образом; нотация является средством общения; набор утилит помогает автоматизировать процесс и управлять им» [9].

RUP разделяет жизненный цикл проекта разработки программного обеспечения ИС на четыре фазы: начало, уточнение, построение, внедрение.

На каждом из этапов выполняются все шесть основных дисциплин разработки: бизнес-моделирование, требования, анализ и проектирование, реализация, тестирование и развертывание.

Основная цель RUP – создание высококачественного программного обеспечения с предсказуемым бюджетом и временными рамками.

Следует отметить, что для построения логической модели ИС необходимо разработать базовые диаграммы языка UML, отражающие различные аспекты исследуемой ИС.

3.2.1 Диаграмма вариантов использования информационной системы автоматизированной проверки учебных заданий на составление SQL-запросов

Для определения функциональных требований применяется диаграмма вариантов использования UML.

Эта диаграмма позволяет описать сложные процессы относительно простыми средствами, а главное делает их понятными для всех заинтересованных лиц.

Диаграмма вариантов использования наглядно представляет взаимодействие между основными сервисами (бизнес-прецедентами), которые предоставляет исследуемый бизнес-процесс, и теми, кому эти сервисы предоставлены (бизнес-субъекты или акторы).

Как инструмент визуального моделирования и бизнес-анализа Rational Rose позволяет системному аналитику отслеживать бизнес-цели и сопоставлять их с системными требованиями, что существенно повышает эффективность процесса формирования последних.

На этапе управления требованиями RUP необходимо, чтобы все прецеденты и участники были определены, и было разработано большинство описаний прецедентов.

Акторами ИСАПЗ являются Преподаватель, Студент и Модуль управления.

Варианты использования (прецеденты) представлены в таблицах 2-7.

Таблица 2 - Описание прецедента: Регистрация/Авторизация

Прецедент: Подготовка заданий по составлению SQL-запросов
ID: 1
«Краткое описание: подготовка заданий по составлению SQL-запросов
Главный актер: Преподаватель
Второстепенный актер: нет

Продолжение таблицы 2

Прецедент: Подготовка заданий по составлению SQL-запросов
Предусловие: нет
Основной поток: Преподаватель подготавливает задания по составлению SQL-запросов
Постусловие: нет
Альтернативные потоки: нет» [9]

Таблица 3 – Описание прецедента: Загрузка тестовой БД

Прецедент: Загрузка тестовой БД
ID: 2
«Краткое описание: загрузка тестовой БД
Главный актер: Преподаватель
Второстепенный актер: нет
Предусловие: нет
Основной поток: Преподаватель загружает БД в систему
Постусловие: нет
Альтернативные потоки: нет» [9]

Таблица 4 - Описание прецедента: Просмотр результатов проверки задания

Прецедент: Просмотр результатов проверки задания
ID: 3
«Краткое описание: Просмотр результатов проверки задания
Главный актер: Преподаватель
Второстепенный актер: нет
Предусловие: нет
Основной поток: Преподаватель просматривает результаты проверки задания
Постусловие: нет
Альтернативный поток: нет» [9]

Таблица 5 - Описание прецедента: Проверка синтаксиса задания

Прецедент: Проверка синтаксиса задания
ID: 4
«Краткое описание: проверка синтаксиса задания
Главный актер: Модуль управления
Второстепенный актер: Преподаватель
Предусловие: нет
Основной поток: Модуль управления проверяет синтаксис задания, разработанного преподавателем
Постусловие: нет
Альтернативный поток: нет» [9]

Таблица 6 - Описание прецедента: Загрузка выполненного задания

Прецедент: Загрузка выполненного задания
ID: 5
«Краткое описание: загрузка выполненного задания
Главный актер: Студент
Второстепенный актер: нет
Предусловие: нет
Основной поток: Студент загружает выполненное задание
Постусловие: нет
Альтернативный поток: нет» [9]

Таблица 7 - Описание прецедента: Проверка выполненного студентом задания

Прецедент: Проверка выполненного студентом задания
ID: 6
«Краткое описание: проверка выполненного студентом задания

Продолжение таблицы 7

Прецедент: Проверка выполненного студентом задания
Главный актер: Модуль управления
Второстепенный актер: нет
Предусловие: Студент загрузил выполненное задание
Основной поток: Модуль управления проверяет выполненное студентом задание
Постусловие: нет
Альтернативный поток: нет» [9]

На рисунке 21 представлена диаграмма вариантов использования ИСАПЗ.

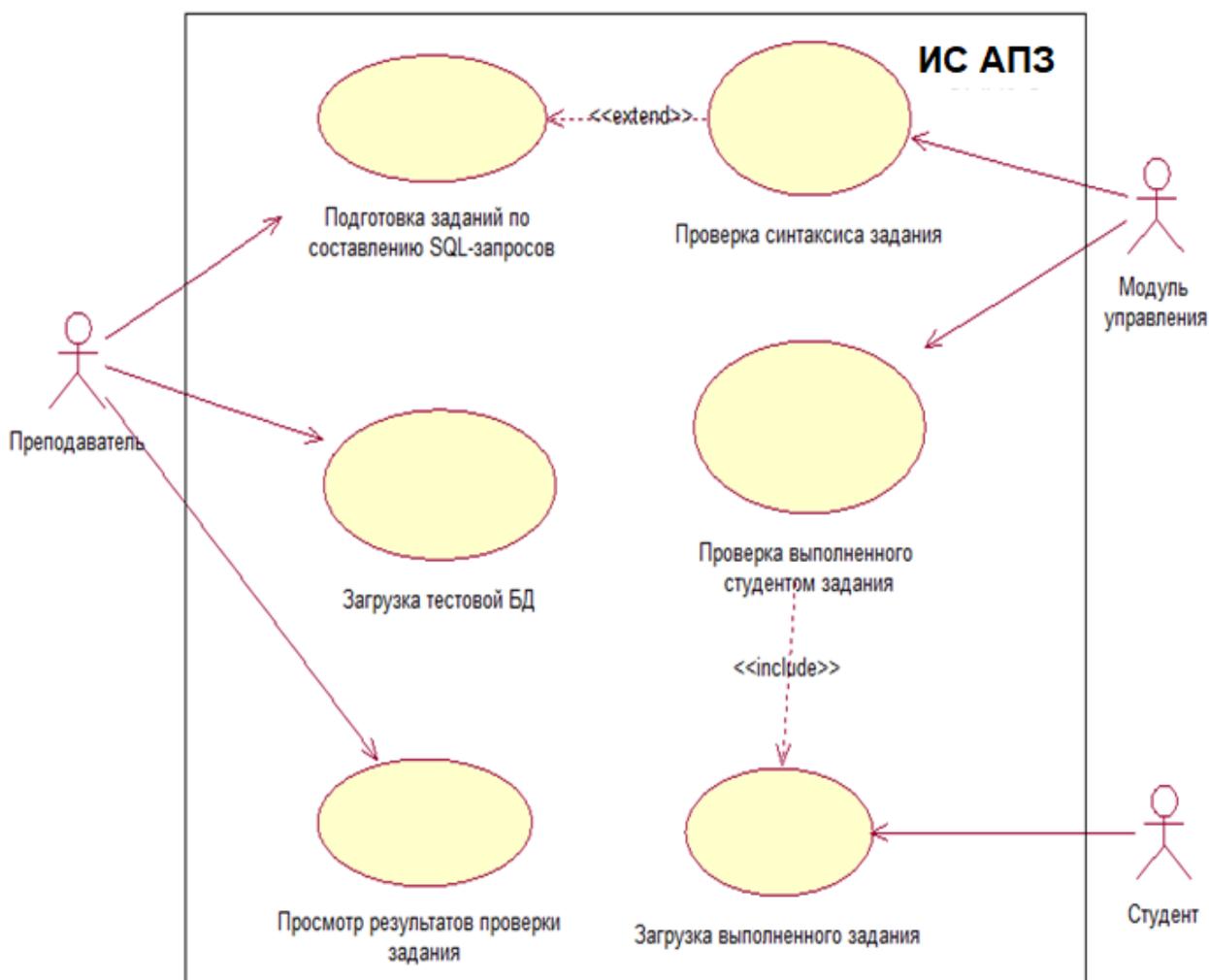


Рисунок 21 – Диаграмма вариантов использования ИСАПЗ

Представленная диаграмма вариантов использования предназначена для отражения функционального аспекта ИСАПЗ.

3.2.2 Диаграмма классов информационной системы автоматизированной проверки учебных заданий на составление SQL-запросов

Для отражения статического аспекта информационной системы используется диаграмма классов UML.

Диаграммы классов описывают внутреннюю структуру классов и отношения между ними. Сюда входят отношения классов, а также атрибуты и поведения, связанные с каждым классом. Диаграммы классов прекрасно иллюстрируют наследование и составные отношения.

На диаграмме классов UML представлены:

- классы в объектно-ориентированной системе;
- атрибуты и методы классов;
- связи между классами.

Не представлены на диаграмме классов UML:

- детали того, как классы взаимодействуют друг с другом
- алгоритмические детали, т.е. как конкретное поведение класса реализовано.

Диаграммы классов отлично подходят для:

- обнаружение связанных данных и атрибутов;
- получения быстрого представления о важных объектах в системе;
- выяснения достаточности или недостаточности классов;
- определения, являются ли отношения между объектами слишком сложными или простыми;
- выявление зависимостей между классами или объектами.

Вместе с тем диаграммы классов не позволяют решить такие вопросы, как обнаружение алгоритмического (не управляемого данными) поведения и понимание общего потока управления приложением.

На рисунке 22 изображена диаграмма классов ИСАПЗ.

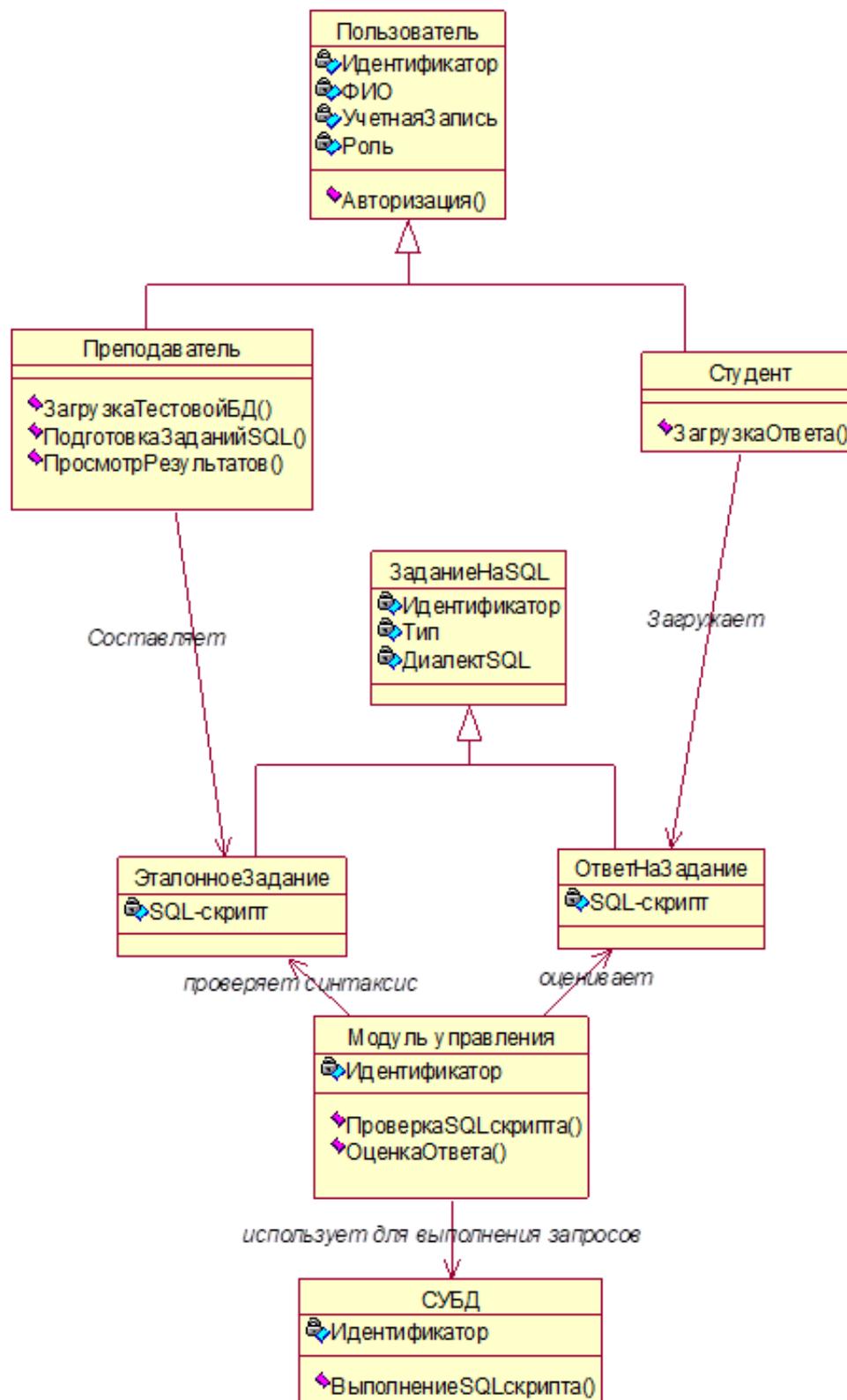


Рисунок 22 – Диаграмма классов ИСАПЗ

В таблице 8 представлена спецификация классов ИСАПЗ.

Таблица 8 - Спецификация классов ИСАПЗ

Класс	Описание
Пользователь	«Класс объектов, моделирующих на логическом уровне пользователей ИСАПЗ
Преподаватель	Класс объектов, моделирующих на логическом уровне преподавателей курса. Является наследником класса Пользователь
Студент	Класс объектов, моделирующих на логическом уровне студентов курса. Является наследником класса Пользователь
ЗаданиеНаSQL	Класс объектов, моделирующих на логическом уровне задания на составление SQL-запросов
ЭталонноеЗадание	Класс объектов, моделирующих на логическом уровне задания, разработанные преподавателем. Является наследником класса ЗаданиеНаSQL
ОтветНаЗадание	Класс объектов, моделирующих на логическом уровне ответы на задания, загруженные студентом. Является наследником класса ЗаданиеНаSQL
МодульУправления	Класс объектов, моделирующих на логическом уровне модули управления ИСАПЗ
СУБД	Класс объектов, моделирующих на логическом уровне СУБД, используемые в ИСАПЗ» [12]

Представленная диаграмма классов является основой для разработки логической модели данных ИСАПЗ.

3.2.3 Диаграмма последовательности сценариев автоматизированной проверки учебных заданий на составление SQL-запросов

Диаграмма последовательности относится к категории диаграмм взаимодействия.

Диаграммы взаимодействия моделируют динамическое поведение системы, показывая, как ее компоненты взаимодействуют для выполнения основных задач, определенных в проекте сценария использования.

Диаграмма последовательности фокусируется на временной последовательности (временном порядке) сообщений или порядке, в котором

сообщения отправляются объектам.

В этих диаграммах акцент делается на том, что происходит первым, вторым и так далее. Они представляют течение времени графически.

Поскольку диаграмма последовательности может быть разработана для каждого основного варианта использования, включенного варианта использования и расширенного варианта использования, на практике для описания работы ИС в динамике строятся несколько диаграмм последовательности.

На рисунке 23 представлена диаграмма последовательности сценария подготовки задания преподавателем.

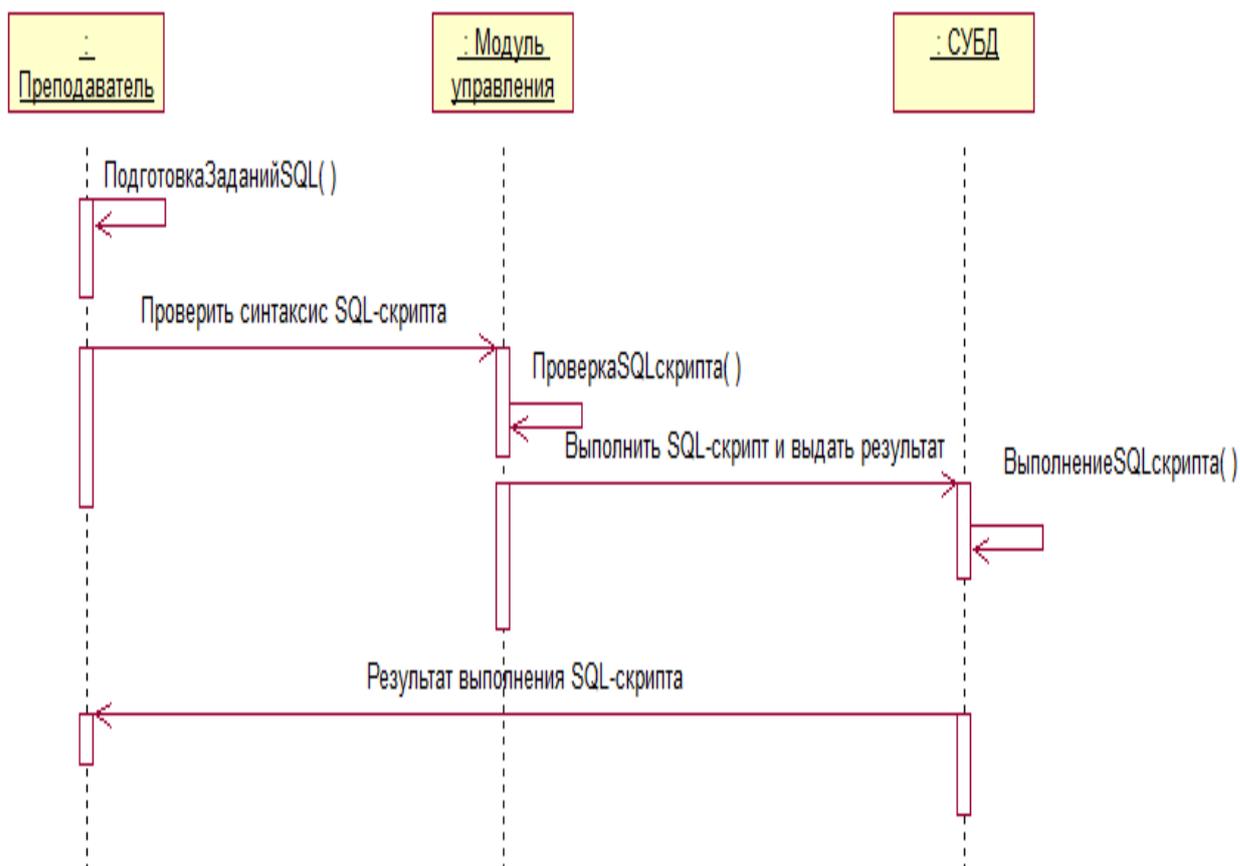


Рисунок 23 – Диаграмма последовательности сценария подготовки задания преподавателем

Сценарий подготовки задания преподавателем организован следующим

образом:

В случайный момент времени объект Преподаватель создает задание и отправляет объекту Модуль управления сообщение «Проверить синтаксис SQL-скрипта».

Объект Модуль управления получает сообщение и запускает операцию проверки SQL-скрипта. Объект Модуль управления отправляет объекту СУБД сообщение «Выполнить SQL-скрипт и выдать результат».

Объект СУБД выполняет SQL-скрипт в своей рабочей среде и отправляет результат объекту Преподаватель для корректировки задания.

Сценарий завершается.

На рисунке 24 представлена диаграмма последовательности сценария оценка ответа студента.

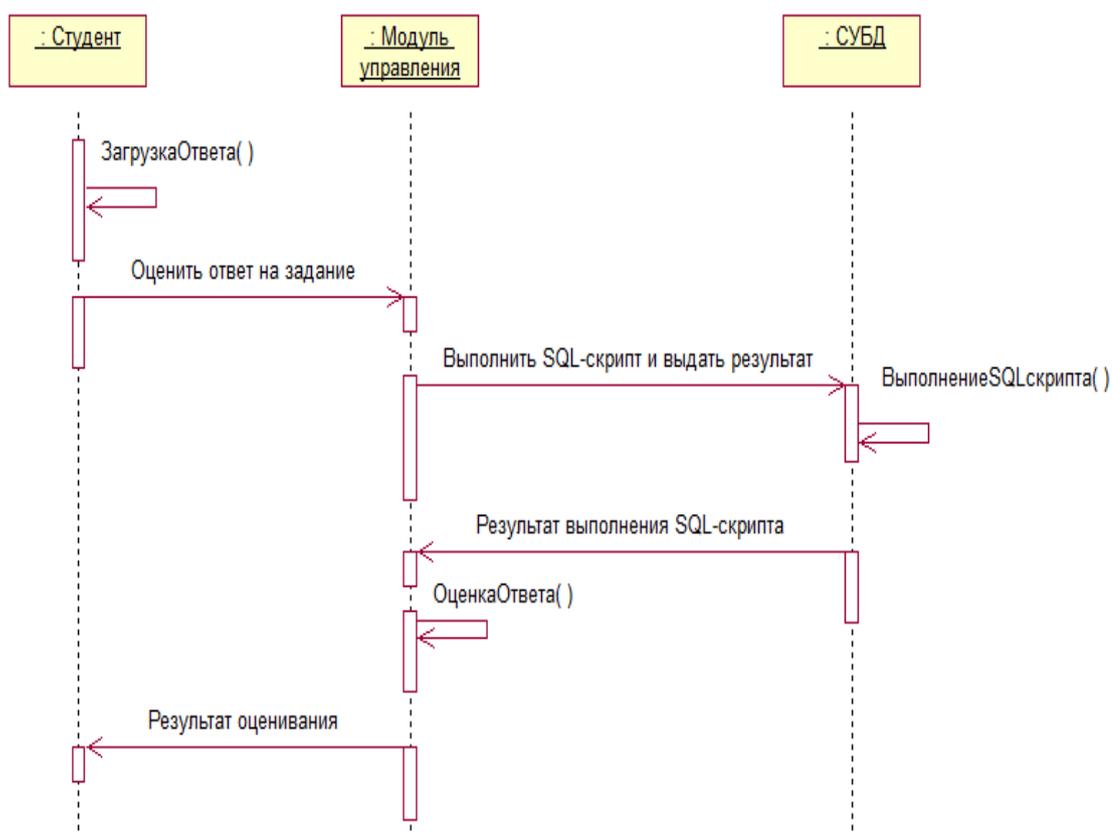


Рисунок 24 – Диаграмма последовательности сценария оценки ответа студента

Сценарий оценивания ответа студента организован следующим

образом:

В случайный момент времени объект Студент загружает задание в ИСАПЗ и отправляет объекту Модуль управления сообщение «Оценить ответ на задание».

Объект Модуль управления получает сообщение и запускает операцию проверки SQL-скрипта. Объект Модуль управления отправляет объекту СУБД сообщение «Выполнить SQL-скрипт и выдать результат».

Объект СУБД выполняет SQL-скрипт в своей рабочей среде и отправляет результат объекту Модуль управления.

Объект Модуль управления выполняет операцию оценки ответа и отправляет результат объекту Студент.

Сценарий завершается.

Разработанные диаграммы последовательности отражают динамический аспект ИСАПЗ.

3.3 Моделирование данных информационной системы автоматизированной проверки учебных заданий на составление SQL-запросов

Проектирование базы данных - это набор шагов, которые помогают создавать, внедрять и поддерживать системы управления данными предприятия.

Основная цель проектирования БД - создание логической и физической моделей проектов для ИСАПЗ [24].

В процессе проектирования БД задействованы следующие этапы моделирования данных:

- логическое моделирование. На этом этапе полный дизайн изложен на бумаге без учета каких-либо конкретных требований к системе управления базами данных (СУБД) или ее физической реализации;
- физическое моделирование. Этот этап следует за логической моделью

и, следовательно, включает физическую реализацию логической модели. Здесь принимается во внимание СУБД и другие факторы физической реализации.

Этап реализации жизненного цикла разработки БД включает в себя:

- преобразование и загрузку данных, включая импорт и преобразование данных из старой системы в новую БД;
- тестирование БД. На этом этапе выявляются ошибки данных в новой системе и проверяются все спецификации требований к БД.

В процессе проектирования БД используются различные программные средства.

Для моделирования данных ИСАПЗ используем CASE-средство MySQL Workbench [23].

MySQL Workbench – это графический инструмент для работы с серверами и базами данных.

Функциональность инструмента поддерживает все задачи моделирования данных: позволяет создавать модели схемы пользовательской БД графически, осуществлять обратный и прямой инжиниринг между схемой и действующей БД, а также редактировать все аспекты БД с помощью всеобъемлющего редактора таблиц.

Редактор таблиц предоставляет простые в использовании средства для редактирования таблиц, столбцов, индексов, триггеров, секционирования, параметров, вставок и привилегий, подпрограмм и представлений.

БД ИСАПЗ предназначена для хранения данных, связанных с данными о заданиях и оценках студентов.

Логическая модель данных ИСАПЗ построена на основе ее диаграммы классов.

Классы объектной модели ИСАПЗ, полученные на этапе разработки диаграммы классов, преобразуются в сущности. При этом анализируются отношения между ними.

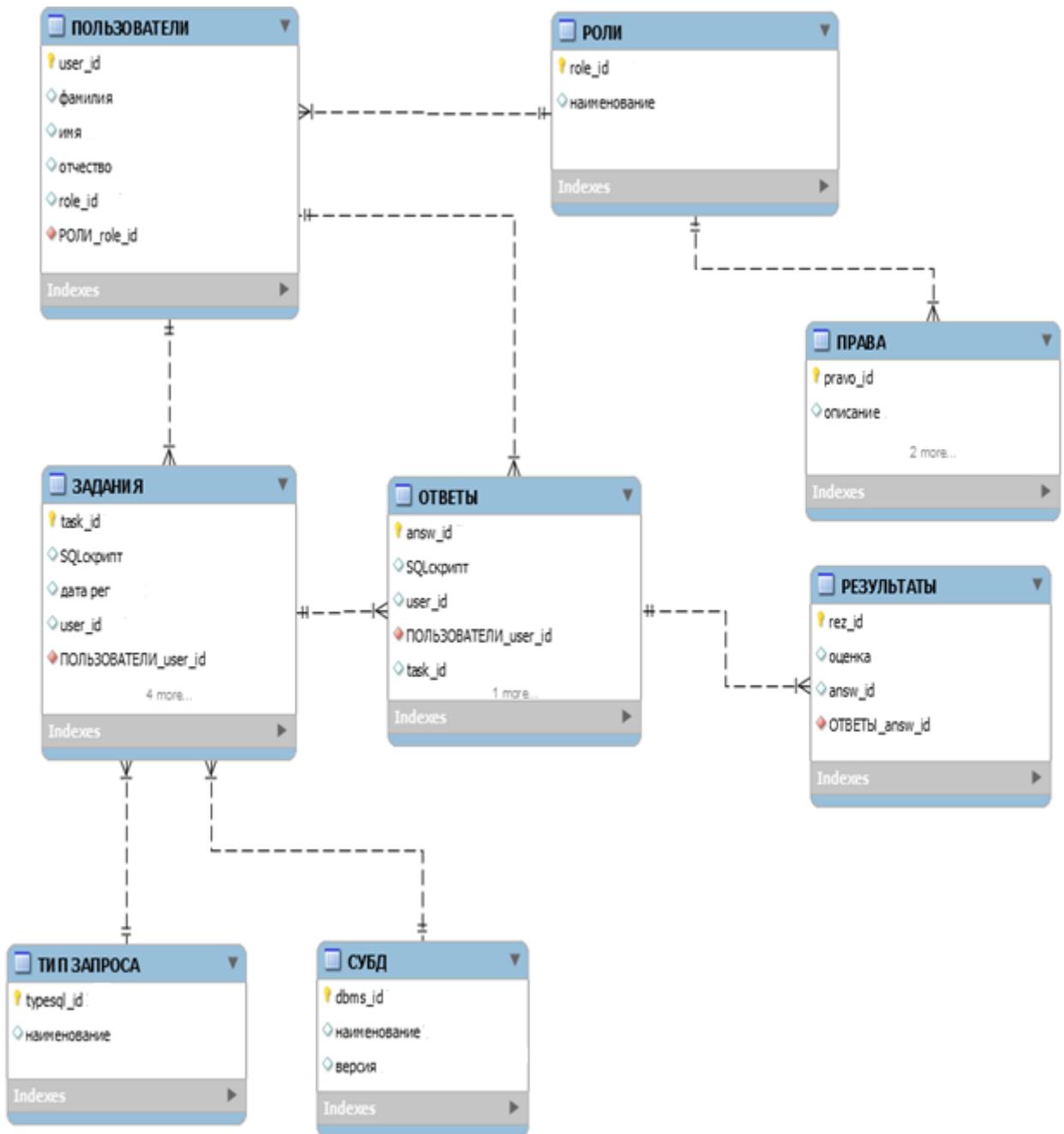


Рисунок 25 – Логическая модель данных ИСАПЗ

В процессе моделирования были выделены следующие сущности:

- Пользователи;
- Задания;
- Ответы;
- Роли;
- Права;

- СУБД;
- Тип запроса;
- Результаты.

Между сущностями установлены следующие типы связи:

- Пользователи-Роли – М:1;
- Пользователи-Задания – 1:М;
- Пользователи-Ответы – 1:М;
- Роли-Права – 1:М;
- Задания-СУБД – М:1;
- Задания-Тип запроса – М:1;
- Задания-Ответы - 1:М;
- Ответы-Результаты - 1:М.

Отношения приведены к третьей нормальной форме.

Все связи между сущностями неидентифицирующие.

Выводы по главе 3

В главе 3 магистерской диссертации описан процесс разработки ИСАПЗ на составление SQL-запросов.

Результаты проделанной работы позволили сделать следующие выводы:

- в ИСАПЗ используется подход, основанный на проверке выходных данных после выполнения оператора SQL;
- в представленных алгоритмах проверки заданий отсутствуют проверки на оригинальность SQL-скриптов. Это существенно упрощает реализацию алгоритмов, не снижая дидактические возможности систем автоматизированной проверки учебных заданий на составление SQL-запросов, используемых в вузах.

Для построения логической модели ИС необходимо разработать базовые диаграммы языка UML, отражающие различные аспекты исследуемой ИС.

Глава 4 Аprobация проектных решений информационной системы автоматизированной проверки учебных заданий на составление SQL-запросов

Для апробации проектных решений разработан прототип ИСАПЗ на составление SQL-запросов, диаграмма компонентов которого представлена на рисунке 26.

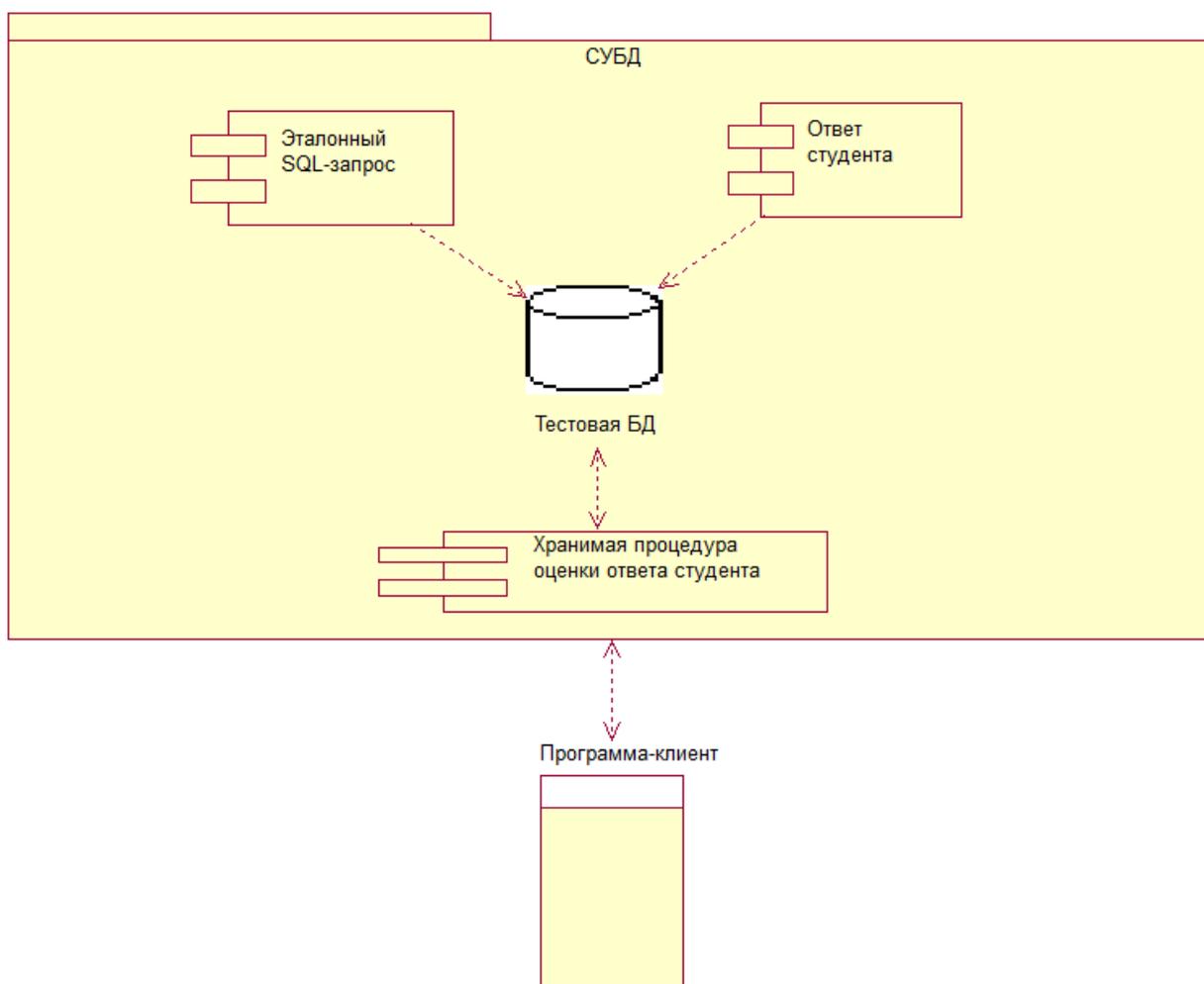


Рисунок 26 – Диаграмма компонентов прототипа ИСАПЗ

В качестве СУБД выбрана СУБД MS SQL Server 2014 Express [15].

В качестве программы клиент используется программа SQL Server Management Studio.

SQL Server Management Studio (SSMS)— это компонент рабочей станции \ клиент, который будет установлен, если выбрать компонент рабочей станции на этапах установки.

Это позволит подключаться к СУБД SQL Server и управлять ее объектами с правами администратора системы.

На рисунке 27 показана диаграмма развертывания ИСАПЗ.

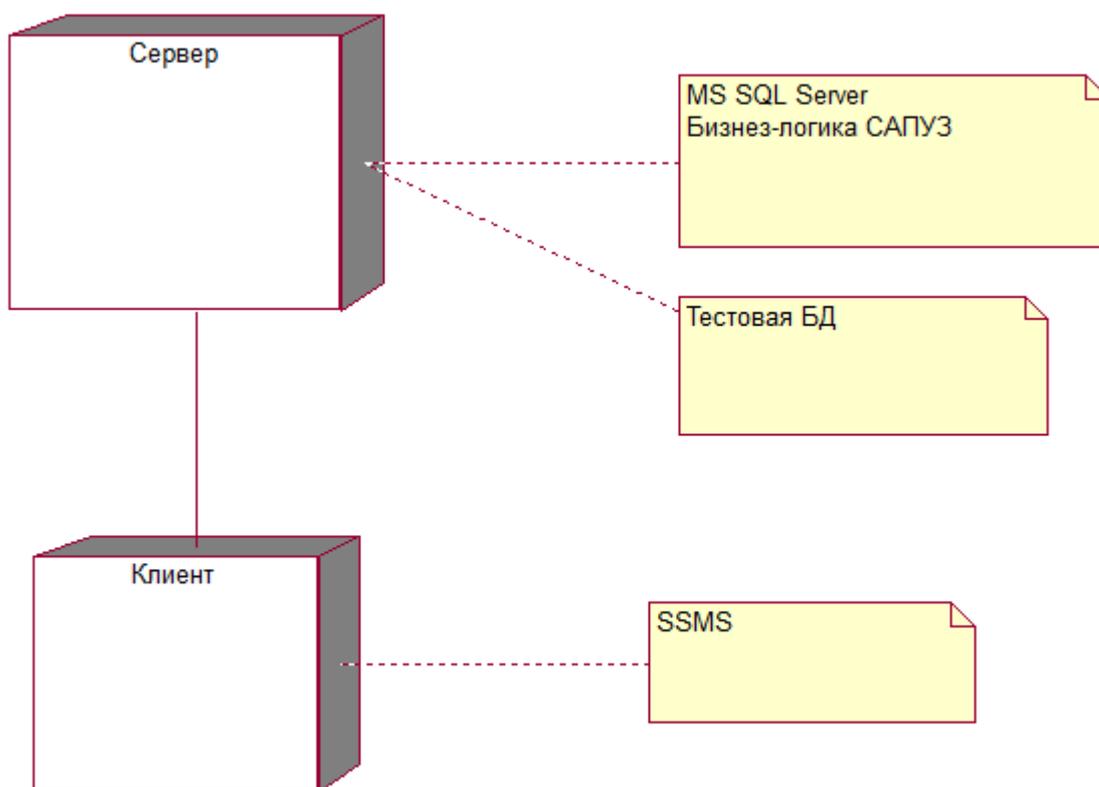


Рисунок 27 – Диаграмма развертывания ИСАПЗ

Рассмотрим работу ИСАПЗ на следующем примере.

Задание: на языке T-SQL разработать SQL-запрос выборки из таблицы «Заказы» ФИО клиентов-женщин, у которых значение поля «Сумма» не превышает 3000 руб.

Структура таблицы «Заказы» представлена на рисунке 28.

Column Name	Data Type
id	int
фио	nchar(20)
пол_М_Ж	nchar(1)
сумма	money

Рисунок 28 – Структура таблицы «Заказы»

Исходные данные таблицы «Заказы», развернутой на тестовой базе, показаны на рисунке 29.

id	фио	пол	сумма
1	Андреев А.	М	1000,0000
2	Борисов Б.	М	2000,0000
3	Григорьева Г.	Ж	1500,0000
4	Васильева В.	Ж	3000,0000
5	Данилова Д.	Ж	4000,0000

Рисунок 29 – Исходные данные таблицы «Заказы»

Эталонный запрос имеет следующий вид:

```
SELECT фио FROM ЗАКАЗЫ WHERE пол='Ж' AND сумма <= 3000
```

Результаты выполнения запроса показан на рисунке 30.

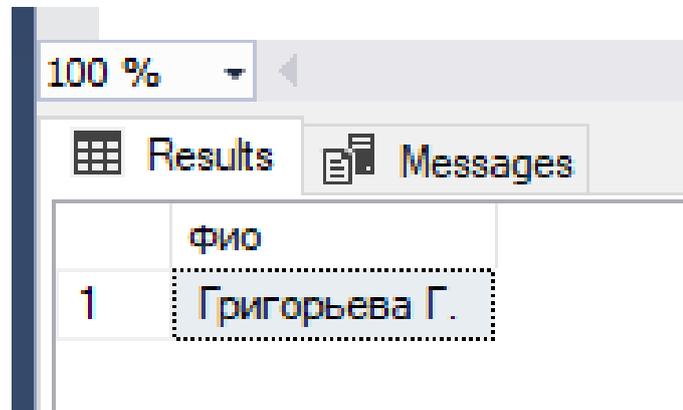
фио
1 Григорьева Г.
2 Васильева В.

Рисунок 30 – Результат выполнения эталонного запроса

Ответ студента:

```
SELECT фио FROM ЗАКАЗЫ WHERE пол='Ж' AND сумма < 3000
```

Результат выполнения запроса студента показан на рисунке 31.



	фио
1	Григорьева Г.

Рисунок 31 – Результат выполнения запроса студента

Хранимая процедура оценки ответа студента представлена в листинге 1.

Листинг 1. Хранимая процедура оценки ответа студента

```
USE [testDB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[sp_SQLAnswerAssement]
AS
BEGIN
IF OBJECT_ID('Клиенты1','U') IS NOT NULL
BEGIN
DROP TABLE Клиенты1
```

END

--Выполнение эталонного запроса

```
SELECT фид INTO Клиенты1 FROM ЗАКАЗЫ WHERE пол='Ж' AND сумма  
<= 3000
```

```
IF OBJECT_ID('Клиенты2','U') IS NOT NULL
```

```
BEGIN
```

```
DROP TABLE Клиенты2
```

```
END
```

--Выполнение запроса из ответа студента

```
SELECT фид INTO Клиенты2 FROM ЗАКАЗЫ WHERE пол='Ж' AND сумма  
< 3000
```

--Сравнение результатов

```
SELECT * FROM Клиенты1
```

```
EXCEPT
```

```
SELECT * FROM Клиенты2
```

```
IF @@ROWCOUNT > 0
```

```
BEGIN
```

```
PRINT 'ОШИБКА!'
```

```
END
```

```
ELSE
```

```
BEGIN
```

```
PRINT 'ПРАВИЛЬНО!'
```

```
END
```

```
END
```

Результаты выполнения хранимой процедуры представлены на рисунке

32.

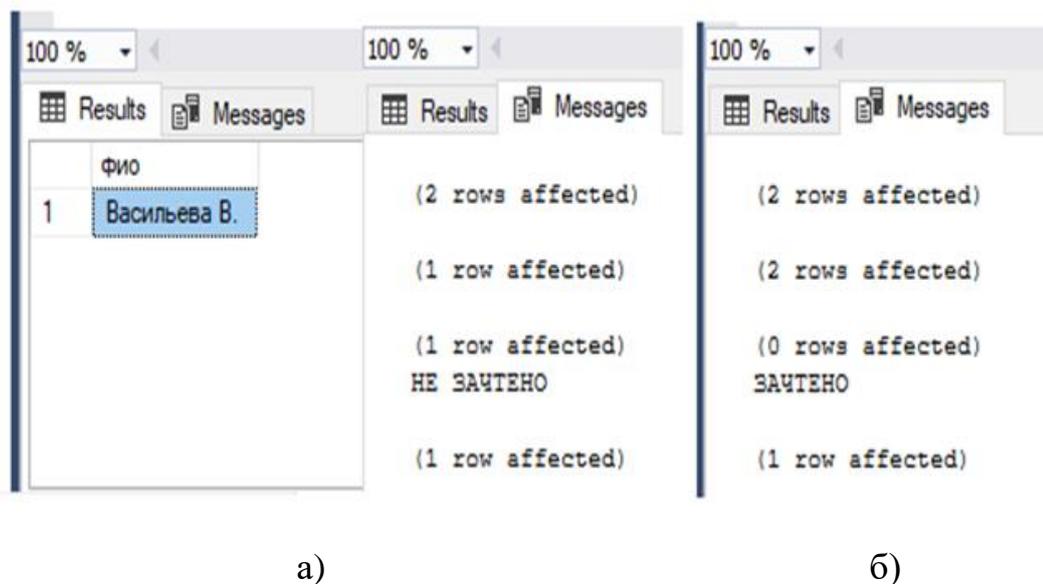


Рисунок 32 - Результаты оценки ответов студентов (а- неправильный ответ, б-правильный ответ)

Представленная процедура может реализована с параметрами, в качестве которых используются эталонный запрос и ответ студента, соответственно.

Имеется возможность проверки стиля SQL-запроса.

В листинге 2 приведен код хранимой процедуры, который проверяет вхождение подстроки 'INNER JOIN' в строку запроса студента.

Листинг 2

```
USE [testDB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[sp_SqlCompare1]
AS
BEGIN
DECLARE @REZ INT
```

```

SELECT @REZ=CHARINDEX('JOIN','SELECT Студенты.Фамилия,
Студенты.Имя, Студенты.Отчество, Студенты.КодГруппы,
Группы.ГодНабора
FROM Группы INNER JOIN Студенты ON Группы.КодГруппы =
Студенты.КодГруппы WHERE Группы.ГодНабора=2020')
IF @REZ>0
PRINT '+'
ELSE
PRINT '-'
SELECT @REZ=CHARINDEX('JOIN','SELECT Студенты.Фамилия,
Студенты.Имя, Студенты.Отчество, Студенты.КодГруппы,
Группы.ГодНабора
FROM Группы,Студенты WHERE Группы.КодГруппы =
Студенты.КодГруппы AND Группы.ГодНабора=2020')
IF @REZ>0
PRINT '+'
ELSE
PRINT '-'
END

```

Результат выполнения хранимой процедуры представлен на рисунке 33.

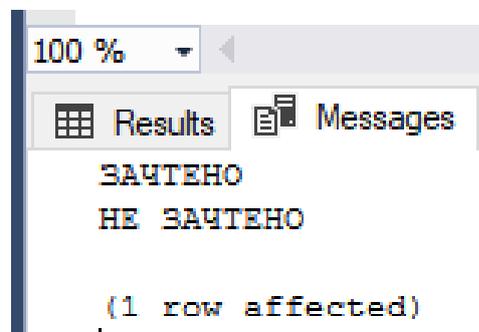


Рисунок 33 – Результат проверки стиля SQL-запроса

Представленные результаты тестирования бизнес-логики прототипа ИСАПЗ подтверждают его работоспособность.

Для оценки эффективности ИСАПЗ используем показатель эффективности управления ИС.

Эффективность управления – это многоплановое понятие, которое включает следующие компоненты:

- целевую эффективность управления;
- экономическую эффективность управления;
- функциональную эффективность управления.

«Целевая эффективность управления характеризует степень достижения цели действий управляемого объекта при фактическом уровне реализации функций управления.

Под экономической эффективностью управления понимается степень полезной отдачи от выделенных средств на разработку, эксплуатацию системы и осуществление управления.

Под функциональной эффективностью управления понимается степень реализации органом управления возлагаемых на него функций.

Величина показателя функциональной эффективности управления $K_{\text{фн}}$ может вычисляться с помощью следующей формулы:

$$K_{\text{фн}} = \frac{\sum_{i=1}^n P_{\text{фн}i}}{n}, \quad (1)$$

где n - количество функций управления, реализуемых ИС;

$P_{\text{фн}i}$ - вероятность выработки ИС эффективного управляющего воздействия при реализации i -й функции управления» [3].

Для ИС с высокой эффективностью управления величина данного показателя должна превышать значение 0.5.

Оценим эффективность ИСАПЗ на основе показателя функциональной

эффективности управления.

В рассматриваемом случае ИСАПЗ предназначена для управления процессом обучения ИТ-специалистов основам языка SQL и поддерживает следующие функции управления:

- подготовка заданий на составление SQL-запросов;
- оценка ответов студентов.

В каждой сессии используется только одна из вышеперечисленных функций, на выполнение которой может негативно повлиять человеческий фактор.

Пусть вероятность выработки эффективного управляющего воздействия для данной функции равна 0,5.

Тогда значение показателя функциональной эффективности управления будет равно:

$$K_{эу} = 1,5/2 = 0,75 \quad (2)$$

Таким образом, коэффициент эффективности управления ИСАПЗ $K_{эу} > 0,5$, что свидетельствует о высокой функциональной эффективности управления системы.

Выводы по главе 4

В главе 3 магистерской диссертации описана апробация ИСАПЗ.

Результаты проделанной работы позволили сделать нижеследующие выводы.

Апробация прототипа ИСАПЗ подтвердила его работоспособность.

Расчет величины коэффициента функциональной эффективности управления ИСАПЗ подтвердил высокую функциональную эффективность управления системы.

Заключение

В настоящее время в сфере подготовки ИТ-специалистов активно развиваются дистанционные технологии обучения. Обязательным компонентом контента таких курсов являются учебные задания по составлению SQL-запросов.

Как показывает практика, одним из самых трудоемких и затратных по времени процессов является проверка учебных заданий.

Очевидным решением данной проблемы является использование при подготовке ИТ-специалистов средств автоматической проверки учебных заданий, разработанных на основе современных ИТ в том числе автоматизированных информационных систем.

Магистерская диссертация посвящена актуальной проблеме исследования и разработки информационной системы автоматизированной проверки учебных заданий по составлению SQL-запросов, обеспечивающей повышение эффективности процесса подготовки ИТ-специалистов.

Выполненные в работе научные исследования представлены следующими основными результатами:

- проанализировано современное состояние исследований в области разработки и внедрения информационных систем автоматизированной проверки учебных заданий по составлению SQL-запросов. Как показал анализ, ручная проверка учебных заданий по составлению запросов на SQL трудоемка и требует много времени. Кроме того, разные СУБД могут иметь разный синтаксис операторов SQL. Это может привести к ошибкам, которые приведут к ухудшению преподавания практики у студентов ИТ-специальностей. Необходимо констатировать недостаточность работ, посвященных проблеме разработки информационных систем для автоматизированной проверки учебных заданий по составлению SQL-запросов, что подтверждает актуальность темы настоящего исследования;
- проведен анализ методологических подходов к построению средств

автоматизированной проверки учебных заданий. Как показал анализ, для автоматической проверки на правильность составления SQL-запросов используются задания на модификацию и выборку данных, соответственно. Базовым алгоритмом проверки запросов на выборку является алгоритм сравнения результатов выполнения проверяемого и эталонного SQL-запроса. Ввиду того, что процесс проверки учебного задания можно рассматривать как Workflow, методология объектно-структурного моделирования может быть использована в процессе логического проектирования ИСАПЗ;

- разработан прототип информационной системы автоматизированной проверки учебных заданий при подготовке ИТ-специалистов.
- выполнена апробация прототипа, подтвердившая его работоспособность. Произведен расчет величины коэффициента функциональной эффективности управления ИСАПЗ, который подтвердил высокую функциональную эффективность управления системы.

Таким образом, в работе решена актуальная научно-практическая проблема исследования и разработки информационной системы автоматизированной проверки учебных заданий по составлению SQL-запросов, обеспечивающей повышение эффективности процесса подготовки ИТ-специалистов.

Гипотеза исследования подтверждена.

Значение диссертационной работы определяется тем, что в ее рамках исследованы возможности повышения эффективности процесса подготовки ИТ-специалистов за счет разработки и внедрения информационной системы автоматизированной проверки учебных заданий по составлению SQL-запросов.

Список используемой литературы и используемых источников

1. Автоматическая проверка заданий в онлайн-курсе (командный проект) [Электронный ресурс]. URL: [http://wiki.cs.hse.ru/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B2%D0%B5%D1%80%D0%BA%D0%B0_%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D0%B9_%D0%B2_%D0%BE%D0%BD%D0%BB%D0%B0%D0%B9%D0%BD-%D0%BA%D1%83%D1%80%D1%81%D0%B5_\(%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82\)](http://wiki.cs.hse.ru/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B2%D0%B5%D1%80%D0%BA%D0%B0_%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D0%B9_%D0%B2_%D0%BE%D0%BD%D0%BB%D0%B0%D0%B9%D0%BD-%D0%BA%D1%83%D1%80%D1%81%D0%B5_(%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82)) (дата обращения: 18.09.2021).
2. Андрианов И.А., Ржеуцкая С.Ю., Харина М.В. Междисциплинарный дистанционный практикум для студентов ИТ-направлений // Открытое образование. Т. 25. № 2. 2021. С. 41-50.
3. Вдовин В. М., Суркова Л. Е., Шурупов А.А. Предметно-ориентированные экономические информационные системы [Электронный ресурс]: учебное пособие. М. : Дашков и К, 2016. 386 с. URL: <https://znanium.com/catalog/product/415090> (дата обращения: 28.10.2021).
4. Говоров А.И. Методы и алгоритмы автоматизированной тренировки и оценки профессиональных умений по составлению SQL-запросов. Диссертация на соискание ученой степени к.т.н. Санкт-Петербург: ИТМО, 2018. 159 с.
5. Горохова-Алексеева А.В. Автоматическая проверка заданий с нечетким алгоритмом // Новые информационные технологии в автоматизированных системах. 2016. №19. С. 225-228.
6. ГОСТ 24.103-84. Единая система стандартов автоматизированных систем управления. Автоматизированные системы управления. Основные положения.
7. ГОСТ 24.702-85. Единая система стандартов автоматизированных систем управления. Эффективность автоматизированных систем управления.

Основные положения.

8. Карпова Т.С., Малышева С.Ю. Расширение систем электронного тестирования на примере тестирования SQL-запросов // Интеллектуальные технологии на транспорте. 2019. №1 (17). С. 33-40.

9. Леоненков А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose [Электронный ресурс] : учебное пособие. Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. 317 с. URL: <https://www.iprbookshop.ru/97554.html> (дата обращения: 06.09.2021).

10. Моисеенко С.И., Калинин В.Ю.. Упражнения по SQL. Модификация данных // Educational Technology & Society, No. 8(1), 2005.

11. Перязева Ю. В. Возможности автоматической проверки заданий в LMS Moodle // Современные информационные технологии и ИТ-образование. 2019. Т. 15, № 4. С. 876- 885.

12. Самуйлов С. В. Объектно-ориентированное моделирование на основе UML [Электронный ресурс] : учебное пособие. Саратов: Вузовское образование, 2016. 37 с. URL: <http://www.iprbookshop.ru/47277.html> (дата обращения: 03.09.2021).

13. Семенова З.В., Любич С.А., Кузнецов А.Г., Мальцев П.А. Система автоматизированной проверки правильности составления SQL-запросов: защита от уязвимостей // Динамика систем, механизмов и машин. 2017. Том 5, № 4. С. 90-94.

14. Синицын С. В., Налютин Н. Ю., Петухова Е. А., Садчиков С. М. Вопросы автоматизации контроля знаний при подготовке современных ИТ-специалистов // Открытое образование. 2006. №4. С.70-77.

15. СУБД MS SQL Server [Электронный ресурс]. URL: <https://www.microsoft.com/en-ie/sql-server/sql-server-downloads> (дата обращения: 21.09.2021).

16. Талипов М. На платформе Mail.ru Cloud Solutions появился сервис для автоматического анализа SQL-запросов // Журнал Mail.ru Cloud Solutions

[Электронный ресурс]. URL: <https://mcs.mail.ru/blog/v-mcs-poyavilsya-servis-dlya-avtomaticheskogo-analiza-sql-zaprosov> (дата обращения: 18.09.2021).

17. AL-Salmi, A. A Web-based Semi-Automatic Assessment Tool for Formulating Basic SQL Statements: Point-and-Click Interaction Method. In Proceedings of the 10th International Conference on Computer Supported Education (CSEDU 2018), pp. 191-198.

18. Creating an automatically graded SQL assignment in CodeGrade [Электронный ресурс]. URL: <https://www.codegrade.com/blog/creating-an-automatically-graded-sql-assignment-in-codegrade> (дата обращения: 18.09.2021).

19. Gartner Glossary [Электронный ресурс]. URL: <https://www.gartner.com/en/information-technology/glossary/sql-structured-query-language> (дата обращения: 18.09.2021).

20. Microsoft Teams for Education [Электронный ресурс]. URL: <https://www.microsoft.com/en-in/education/products/teams> (дата обращения: 18.09.2021).

21. Mkrtychev S 2018 Methodology to design management accounting information systems CEUR Workshop Proceedings 2258 21-28.

22. Model-View-Controller design pattern [Электронный ресурс]. URL: <https://help.hcltechsw.com/commerce/9.1.0/developer/concepts/csdmvcdespat.html> (дата обращения: 03.09.2021).

23. MySQL Workbench [Электронный ресурс]. URL: <http://www.mysql.com/products/workbench/features.html> (дата обращения: 14.09.2021).

24. Naeem T. All You Need to Know About Database Design [Электронный ресурс]. URL: <https://www.astera.com/type/blog/all-you-need-to-know-about-database-design/> (дата обращения: 21.09.2021).

25. Pakroom Vichianroj, Areerat Trongratsameethong, “ASQLAG - Automated SQL Assignment Grading System for Multiple DBMSs”, Journal of Technology and Innovation in Tertiary Education, 2018, Vol.1, No.1, pp. 41- 59.

26. Permpool T. et al. Interactive SQL Learning Tool with Automated

Grading using MySQL Sandbox, Conference: 2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA).

27. Požnel M., Fürst L. and Mahnič V., "Introduction of the automated assessment of homework assignments in a university-level programming course," 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015, pp. 761-766.

28. Renard L. How to create automatically graded assignments in Microsoft Teams for Education [Электронный ресурс]. URL: <https://www.bookwidgets.com/blog/2020/08/how-to-create-automatically-graded-assignments-in-microsoft-teams-for-education> (дата обращения: 18.09.2021).

29. Russell G., Andrew Cumming A. Automatic Checking of SQL: Computerised Grading, Conference: Twelfth International Conference on Learning, At: Granada, Spain, 2005, Volume: 12.

30. Scerbakov N., Schukin A., Sabinin O. Plagiarism detection in SQL student assignments, 20th International Conference on Interactive Collaborative Learning At: Budapest, Hungary, 2017.

31. SQL - тренажёр [Электронный ресурс]. URL: <https://sql-ex.ru/> (дата обращения: 18.09.2021).

32. Transact-SQL Reference (Database Engine) [Электронный ресурс]. URL: <https://docs.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-ver15> (дата обращения: 18.09.2021).

33. Welcome to Moodle CodeRunner [Электронный ресурс]. URL: <https://coderunner.org.nz/> (дата обращения: 18.09.2021).