

Impact Factor:

ISRA (India) = 6.317
ISI (Dubai, UAE) = 1.582
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
ПИИЦ (Russia) = 3.939
ESJI (KZ) = 8.771
SJIF (Morocco) = 7.184

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

SOI: [1.1/TAS](#) DOI: [10.15863/TAS](#)

International Scientific Journal Theoretical & Applied Science

p-ISSN: 2308-4944 (print) e-ISSN: 2409-0085 (online)

Year: 2022 Issue: 05 Volume: 109

Published: 24.05.2022 <http://T-Science.org>

Issue

Article



Liya Ildarovna Borganova

Peter the Great St. Petersburg Polytechnic University
Bachelor's Student
Institute of Computer Science and Technology

Oleg Yurievich Sabinin

Peter the Great St. Petersburg Polytechnic University
Candidate of Engineering Sciences, Docent
Institute of Computer Science and Technology

AUTOMATED SYSTEM FOR CHECKING THE CORRECTNESS OF THE IMPLEMENTATION OF FUNCTIONS IN ORACLE PL/SQL LANGUAGE USING ORACLE APPLICATION EXPRESS

Abstract: The purpose of this article is to research and develop an algorithm for checking the correctness of the implementation of functions in the Oracle PL/SQL language and creating an automated verification system based on it using the capabilities of Oracle Application Express.

Key words: Oracle PL/SQL, Oracle Application Express, automated check system, correct implementation of functions, databases.

Language: Russian

Citation: Borganova, L. I., & Sabinin, O. Y. (2022). Automated system for checking the correctness of the implementation of functions in Oracle PL/SQL language using Oracle Application Express. *ISJ Theoretical & Applied Science*, 05 (109), 735-741.

Soi: <http://s-o-i.org/1.1/TAS-05-109-68> **Doi:**  <https://dx.doi.org/10.15863/TAS.2022.05.109.68>

Scopus ASCC: 1700.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПРОВЕРКИ ПРАВИЛЬНОСТИ РЕАЛИЗАЦИИ ФУНКЦИЙ НА ЯЗЫКЕ ORACLE PL/SQL С ИСПОЛЬЗОВАНИЕМ ORACLE APPLICATION EXPRESS

Аннотация: Целью данной статьи является исследование и разработка алгоритма проверки правильности реализации функций на языке Oracle PL/SQL и создание на его основе автоматизированной системы проверки с использованием возможностей Oracle Application Express.

Ключевые слова: Oracle PL/SQL, Oracle Application Express, система автоматизированной проверки, правильность реализации функций, базы данных.

Введение

UDC 004.6

В настоящее время стремительно растёт показатель внедрения автоматизированных систем во всех областях деятельности человека. Благодаря таким технологиям процессы, которые раньше требовали больше времени на реализацию и больше рабочей силы, сейчас выполняются компьютерными системами гораздо быстрее.

Спрос на автоматизированные системы возрастает и в сфере образования. В силу введения дистанционных технологий работа преподавателей стала тесно связана с компьютерами. Электронные конспекты, видеолекции, дополнительная научная литература теперь могут размещаться на онлайн-платформах. Контрольные работы, промежуточные тесты, практические работы, загружаемые студентами, имеют возможность проверяться встроенной системой, что намного эффективнее по времени и

Impact Factor:

ISRA (India) = 6.317
ISI (Dubai, UAE) = 1.582
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
ПИИЦ (Russia) = 3.939
ESJI (KZ) = 8.771
SJIF (Morocco) = 7.184

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

по качеству, чем если бы этот процесс проходил вручную.

В частности, если рассматривать дисциплину, преподаваемую в университете и связанную с базами данных, где студенты выполняют большое количество лабораторных работ, включающих задачи по составлению на языке программирования Oracle PL/SQL различного рода программных объектов, например, таких как функции. Процесс проверки таких задач является трудоёмким в силу необходимости тестирования решений на различном наборе входных данных, а также в силу проверки отдельных компонент программного кода. К тому же вследствие загруженного графика преподавателей и высокого количества работ, особенно в период экзаменационной сессии, сам процесс может стать более продолжительным или менее качественным.

Цели и задачи исследования

В связи с этим целью данной статьи стало исследование и разработка алгоритма проверки правильности реализации функций на языке Oracle PL/SQL и создание на его основе автоматизированной системы проверки с использованием возможностей Oracle Application Express. Такая система во многом облегчит работу преподавателей, рассматриваемой дисциплины.

Для достижения поставленной цели были поставлены следующие задачи:

- провести анализ особенностей структуры создания функций на языке программирования PL/SQL.
- обосновать выбор инструмента для разработки системы.
- рассмотреть основные технологии проверки решений задач.
- разработать алгоритм проверки правильности реализации функций.
- спроектировать базу данных для хранения результатов проверки.
- создать автоматизированную систему с применением возможностей Oracle APEX.
- протестировать разработанную систему.

Особенности структуры создания функций на языке PL/SQL

PL/SQL (Procedural Language for SQL – процедурные языковые расширения для языка SQL) — это язык программирования, процедурное расширение языка SQL, разработанное корпорацией Oracle.

PL/SQL по сравнению с SQL имеет блочную структуру, возможность использования кодов на языках Java и C, возможность создания и хранения программных элементов (процедур, функций и пакетов). Он может использоваться как

встроенный язык для средства быстрой разработки прикладных систем Oracle Forms, инструмента разработки отчётов Oracle Reports и в среде разработки веб-приложений Oracle Application Express [1].

Главная его особенность в том, что с его помощью можно создавать более функциональные решения для задач пользователей.

Функция (function) — это программный модуль, который содержит одну или несколько исполняемых инструкций, с помощью списка параметров может принимать и возвращать значения и, кроме того, обязан вернуть ровно одно, возможно составное, значение с помощью оператора RETURN.

Синтаксис создания хранимой функции:
CREATE [OR REPLACE] FUNCTION
[<схема>.]<имя>

[(*<описание параметра>* [, *<описание параметра>*]...)]

RETURN *<тип данных возвращаемого значения>*

[[AUTHID {CURRENT_USER | DEFINER} | DETERMINISTIC |

PARALLEL_ENABLE...] [...n]]

[RESULT_CACHE [RELIES_ON *<имя таблицы>*] [, *имя таблицы*]...]

[[AGGREGATE | PIPELINED] USING
[<схема>.]<имя типа>

[PIPELINED]] {IS | AS} [*<оператор объявления>*]...]

BEGIN
<исполняемый оператор или команда>;

[*<исполняемый оператор или команда>*;]...
[EXCEPTION

<обработчик исключений>;
[*<обработчик исключений>*;]...]

END [*<имя>*]; [2]

Команда создания функции имеет в заголовке ключевое слово RETURN, определяющее тип данных возвращаемого функцией значения, а в исполняемом разделе — по крайней мере, один оператор RETURN, определяющий само возвращаемое значение. Этот оператор может возвращать любое значение, совместимое с типом данных, который указан в заголовке.

Назначение некоторых параметров:
<схема> — это имя схемы, которой будет принадлежать функция (по умолчанию это имя схемы текущего пользователя).

<описание параметра> — это необязательный список параметров, которые применяются для передачи данных в функцию и в некоторых случаях для возврата информации из функции в вызывающую программу.

AUTHID {CURRENT_USER | DEFINER} — это параметр, определяющий с какими

Impact Factor:

ISRA (India) = 6.317
ISI (Dubai, UAE) = 1.582
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
ПИИЦ (Russia) = 3.939
ESJI (KZ) = 8.771
SJIF (Morocco) = 7.184

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

полномочиями будет выполняться функция — создателя или текущего пользователя.

<оператор объявления> — это необязательные объявления формальных аргументов

<исполняемый оператор или команда> — это операторы или команды, выполняемые функцией при вызове. Между ключевыми словами BEGIN и END или EXCEPTION должен находиться, по крайней мере, один исполняемый оператор или команда.

<обработчик исключений> — это необязательные обработчики исключений для процедуры [2].

Обоснование выбора Oracle Application Express как инструмента для разработки системы

Oracle Application Express (Oracle APEX) — это высокоуровневая среда быстрой разработки прикладного программного обеспечения на основе СУБД Oracle Database. Он позволяет, используя только веб-браузер и не столь обширные навыки программирования, разрабатывать быстрые и практичные приложения. Oracle APEX сочетает в себе и базу данных, и высокую производительность, и гибкость, а также доступность [3].

Среда разработки Oracle Application Express выступает в качестве оптимального средства для программистов. К ее преимуществам можно отнести:

- Удобный и понятный интерфейс.
- Широкий спектр функциональности.
- Простота развертывания веб-приложения
- Необходимость знаний языков управления данными на базовом уровне.
- Необязательное изучения средств верстки веб-страниц.
- Интеграция с различными источниками данных, такими как локальная или удаленная база данных или любой веб-сервис.
- Наличие высокой безопасности системы [4; 5].

К недостаткам можно отнести то, что веб-приложения создаются с использованием собственных инструментов Oracle и могут размещаться только в базе данных Oracle. Также при командной разработке отсутствует встроенный контроль версий.

Однако плюсов использования гораздо больше и благодаря этому Oracle APEX пользуется большим спросом, является перспективным инструментом для разработки веб-приложений, активно совершенствуется и дополняется новыми функциями.

Обзор основных технологий проверки решений задач

Существующие подходы проверки задач можно разделить на две основные категории. К первому способу, который называют поиском сходств, относится проверка эквивалентности задач. По своей сути, это процесс сопоставления релевантных частей информации друг с другом. Ко второму же способу относится сравнение результатов работы задач на некотором наборе входных параметров.

Для первого подхода характерно преобразование задачи к алгебраическому представлению в виде выражения в некоторой формальной системе, которая поддерживала бы различные операции над этими выражениями. Далее полученное выражение приводится к нормальной форме, которая как раз и позволяет уже делать выводы об эквивалентности задач. Как правило для того, чтобы привести выражение к виду, возможному для дальнейшей нормализации, необходимо применить некоторый набор эквивалентных преобразований до тех пор, пока это возможно. Получение структурно похожих выражений упрощает процесс их сравнения. Обычно набор преобразований выбирается вручную [6]. При использовании данного подхода также за основу можно взять не менее известные алгоритмы: коэффициент Жаккара, алгоритм шинглов, расстояние Лефенштейна [7].

Заключительным этапом проверки для первого подхода выступает оценка схожести решений с вычислением процента схожести по определённой формуле. Затем на основе выставленных порогов определяется результат, который показывает, можно ли считать задачу верной или нет. При этом учитывается тот факт, что при проверке задачи, не столь важен её код, поскольку реализовать одну и ту же задачу можно путем применения различных подходов, вследствие чего код будет отличаться в плане написания.

В целом данный подход хорошо справляется со своей задачей. Однако, у него есть несколько недостатков. Во-первых, в рамках рассматриваемой предметной области, он не поддерживает многие отличительные особенности языка PL/SQL, такие как значение NULL, сложные арифметические выражения и другие [6]. Во-вторых, его работа требует значительных вычислительных ресурсов из-за большого количества алгебраических преобразований и операций нормализации.

Основу второго подхода составляет наличие решения, которое принимается за правильное, и с этим решением сравниваются остальные варианты реализации программного кода [8]. Определение оценки правильности осуществляется путём сравнения результатов

Impact Factor:

ISRA (India) = 6.317
ISI (Dubai, UAE) = 1.582
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
ПИИЦ (Russia) = 3.939
ESJI (KZ) = 8.771
SJIF (Morocco) = 7.184

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

работы двух задач – верно решенной и исследуемой. На основе сопоставления ответов выдается соответствующий результат.

На самом деле нет строго определенных алгоритмов реализации данной технологии. Определяющими факторами являются структура задачи, набор входных параметров для тестирования и метод оценки результатов. В рамках рассматриваемой предметной области данный подход имеет преимущество за счет своей более простой реализации, широкой функциональности, меньшей требовательности к ресурсам и самое главное за счет возможности учета особенности проверки функций, написанных на языке PL/SQL. Достигается это тем, что при проверке работы функции, учитывается процесс ее выполнения, то есть то, какие выходные данные получает пользователь и какие входные данные он должен внести.

Главным фактором при проверке выступает исследование результатов, которые выдает функция на различном наборе входных параметров, как верных, удовлетворяющих условию задачи, так и неверных, обработка которых также должна дополнительно учитываться. При этом пользователь может получить либо сообщение об ошибке, что, к примеру, такое значение не поддерживается, либо результат отличный от правильного, но прописанный в критериях задачи.

Первая технология проверки решений, основывающаяся на сравнение схожести текстов, чаще всего применяется в системах проверки плагиата, потому что позволяет в процентном соотношении определить схожесть текстов, что впоследствии является хорошей основой для выставления итогового результата в зависимости от выставленных требований.

Вторая технология проверки решений, основывающаяся на сравнении результатов работы задач, хорошо подходит для текущей разработки системы по определению правильности реализации функций. В условиях поставленной задачи не требуется выявлять процент схожести кодов. В дополнении код написания функции может совпадать у нескольких студентов. И не мало важным является тот факт, что такой подход менее ресурсно- и время затратный, что играет большую роль для дальнейшего применения системы в работе преподавателей.

Таким образом, для дальнейшей работы был выбран метод, основанный на сравнении результатов работы задач, дополненный операциями, необходимыми для расширения реализации алгоритма в условиях проверки функций на PL/SQL.

Описание разработанного алгоритма

Алгоритм проверки должен учитывать особенности реализации функций, к которым относятся количество и тип данных входных и выходных параметров. В зависимости от этого различают, к примеру функции, которые не принимают входные аргументы или же принимают несколько входных параметров, которые возвращают переменную с типом данных запись или же возвращают логическое значение и многие другие [9].

На данном этапе реализации были рассмотрены варианты проверки функций, которые не принимают, принимают один или более входных аргументов и которые возвращают одно символьное или числовое значение.

Основная суть алгоритма заключается в том, чтобы в начале проверить наличие ключевых слов в синтаксисе функции студента. При их отсутствии код студента сразу считается неверным. Если же все искомые ключевые фразы были найдены, тогда на выполнение запускаются процедуры проверки. В случае необходимости в студенческую и в правильную функцию передаются входные параметры и осуществляется сравнение полученных результатов на всем наборе исходных параметров. В конце проверки выставляется результат в зависимости от количества неправильных ответов и от наличия возникших в ходе работы функции исключений.

Соответственно, если все ответы совпали и не возникли исключения, тогда функция студента считается верной. Иначе система информирует проверяющего об обнаружении ошибок или же о не совпавших ответах.

Созданные для проверки функции анализируют синтаксис кода, а именно проверяют наличие определенных ключевых слов, таких как «CREATE», «OR REPLACE», «EXCEPTION WHEN» И «WHEN OTHERS». При их обнаружении они возвращают значение с типом данных boolean равное TRUE, а при отсутствии – FALSE.

Созданные для проверки процедуры в свою очередь выполняют следующие действия:

- проверяют имя функции студента.

В условиях задачи по созданию функции может быть явно прописано, какое имя нужно ей присвоить. В таком случае, имя функции студента и правильной функции окажутся одинаковыми, а создание двух объектов с одним и тем же именем невозможно. Для решения данной проблемы необходимо изменить имя функции студента.

- проверяют функции с входными параметрами.

Здесь учитывается количество входных аргументов за счет их структуры хранения в таблице базы данных. Используются специальные символы для разделения групп входных

Impact Factor:

ISRA (India) = 6.317
ISI (Dubai, UAE) = 1.582
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
ПИИЦ (Russia) = 3.939
ESJI (KZ) = 8.771
SJIF (Morocco) = 7.184

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

параметров и самих аргументов в рамках одной группы.

- проверяют функции без входных параметров.
- формируют итоговый результат проверки.

Создание базы данных для хранения результатов

Для хранения результатов проверки функций, информации о студентах и преподавателях, правильных решениях и решениях студентов была создана собственная база данных. Ее схема представлена на рисунке 1.

Разработанная база данных состоит из шести таблиц:

1. «USER_ROLE» – это таблица, содержащая информацию о ролях, которые присваиваются каждому пользователю системы.

2. «USER_REPOSITORY» – это таблица, содержащая информацию о пользователях.

3. «STUDENT_CODE» – это таблица, содержащая информацию о решениях студентов, то есть их варианты реализации функций.

4. «CORRECT_CODE» – это таблица, содержащая информацию о правильных решениях реализации функции.

5. «STUDENT_MARK» – это таблица, содержащая информацию об успехах студентов.

6. «PAGE_ACCESS» – это таблица, содержащая информацию о ролях и о номерах страниц, на которые им соответственно предоставляется доступ.

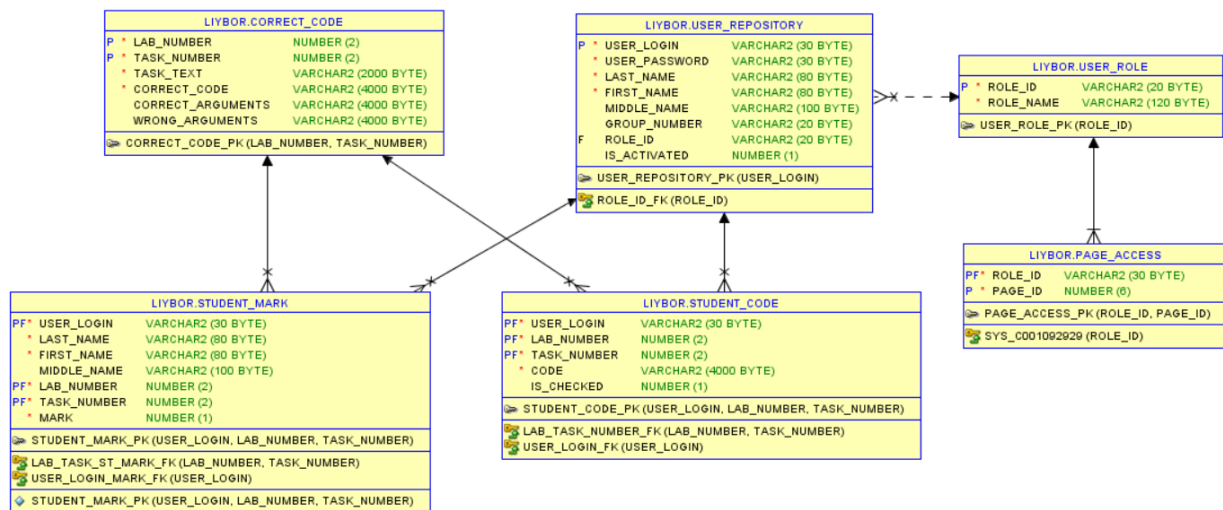


Рисунок 1 – ER-диаграмма разработанной базы данных.

Создание автоматизированной системы с применением возможностей Oracle APEX

Заключительным этапом стало создание веб-приложения на основе Oracle Application Express с 3 страницами по умолчанию: Global Page – страница с глобальными настройками веб-приложения, Home Page – домашняя страница, Login Page – страница входа и несколькими дополнительными страницами [10].

В начала работы для каждого пользователя открывается страница входа в приложение, где он вводит логин и пароль, либо выбирает вариант регистрации. При регистрации открывается специальная форма, на поле с логином установлена проверка на уникальность и в случае ошибки, выходит сообщению пользователю с просьбой ввести новый логин.

Была реализована собственная схема аутентификации и авторизации пользователей.

А также в зависимости от роли пользователю предоставлялись определенные привилегии на доступ к самим страницам или к принадлежащим им объектам [11].

Тестирование разработанной системы

Для проведения тестирования для начала были созданы пользователи – преподаватель и студент. От имени преподавателя в таблицу с правильными функциями были внесены новые записи. Затем от имени студента были загружены функции для проверки, как правильные, так и неправильные.

Алгоритм тестирования заключается в проверке работы системы на различном наборе функций. В данном случае проверка происходила от имени преподавателя. В начале была проверена функция с правильной реализацией. Результат работы системы представлен на рисунке 2.

Impact Factor:

ISRA (India)	= 6.317	SIS (USA)	= 0.912	ICV (Poland)	= 6.630
ISI (Dubai, UAE)	= 1.582	ПИИЦ (Russia)	= 3.939	PIF (India)	= 1.940
GIF (Australia)	= 0.564	ESJI (KZ)	= 8.771	IBI (India)	= 4.260
JIF	= 1.500	SJJI (Morocco)	= 7.184	OAJI (USA)	= 0.350

Результат проверки	
Номер лабораторной	1
Номер задачи	1
Код студента	Правильный код
<pre>CREATE OR REPLACE FUNCTION GET_JOB (P_JOB_ID IN JOBS.JOB_ID%TYPE) RETURN JOBS.JOB_TITLE%TYPE AS P_JOB_TITLE JOBS.JOB_TITLE%TYPE; BEGIN SELECT JOB_TITLE INTO P_JOB_TITLE FROM JOBS WHERE JOB_ID = P_JOB_ID; RETURN P_JOB_TITLE; EXCEPTION WHEN NO_DATA_FOUND THEN RETURN NULL; WHEN OTHERS THEN RETURN NULL; END GET_JOB;</pre>	<pre>CREATE OR REPLACE FUNCTION GET_JOB (P_JOB_ID IN JOBS.JOB_ID%TYPE) RETURN JOBS.JOB_TITLE%TYPE AS P_JOB_TITLE JOBS.JOB_TITLE%TYPE; BEGIN SELECT JOB_TITLE INTO P_JOB_TITLE FROM JOBS WHERE JOB_ID = P_JOB_ID; RETURN P_JOB_TITLE; EXCEPTION WHEN NO_DATA_FOUND THEN RETURN NULL; WHEN OTHERS THEN RETURN NULL; END GET_JOB;</pre>
Результат проверки	Функция правильная!

Рисунок 2 – Результат проверки правильной функции.

Затем были проверены неправильные функции и в зависимости от причины неверной реализации преподавателю отображались соответствующие сообщения.

Результат проверки функции с отсутствием ключевого слова CREATE представлен на рисунке 3.

Результат проверки	
Номер лабораторной	1
Номер задачи	4
Код студента	Правильный код
<pre>FUNCTION LFNAME RETURN VARCHAR2 AS VAL VARCHAR2(100); BEGIN SELECT EMPLOYEE_ID INTO VAL FROM EMPLOYEES WHERE LAST_NAME = 'King' AND FIRST_NAME = 'Steven'; RETURN VAL; EXCEPTION WHEN OTHERS THEN RETURN NULL; END LFNAME;</pre>	<pre>CREATE OR REPLACE FUNCTION LFNAME RETURN VARCHAR2 AS VAL VARCHAR2(100); BEGIN SELECT EMPLOYEE_ID INTO VAL FROM EMPLOYEES WHERE LAST_NAME = 'King' AND FIRST_NAME = 'Steven'; RETURN VAL; EXCEPTION WHEN OTHERS THEN RETURN NULL; END LFNAME;</pre>
Результат проверки	Сообщение об ошибке
Функция неправильная!	Отсутствует ключевое слово CREATE!
Неправильные ответы	

Рисунок 3 – Результат проверки неправильной функции с отсутствием ключевого слова CREATE.

Результат проверки функции с ошибкой в вычислениях представлен на рисунке 4. В данном случае отображаются ответы,

которые не совпали с результатом правильной функции, и соответственно по этой причине функция и оказалась неправильной.

Impact Factor:

ISRA (India) = 6.317
ISI (Dubai, UAE) = 1.582
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
ПИИЦ (Russia) = 3.939
ESJI (KZ) = 8.771
SJIF (Morocco) = 7.184

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

Результат проверки	
Номер лабораторной	2
Номер задачи	5
Код студента	Правильный код
<pre>CREATE OR REPLACE FUNCTION GET_ANNUAL_COMP (P_EMP_SAL IN EMPLOYEES.SALARY%TYPE, P_EMP_COM IN EMPLOYEES.COMMISSION_PCT%TYPE) RETURN EMPLOYEES.SALARY%TYPE AS BEGIN IF (P_EMP_SAL IS NULL) THEN RETURN 0; ELSIF (P_EMP_COM IS NULL) THEN RETURN (P_EMP_SAL*10); ELSE RETURN (P_EMP_SAL*10) + (P_EMP_COM*P_EMP_SAL*10); END IF; EXCEPTION WHEN OTHERS THEN RETURN 0; END GET_ANNUAL_COMP;</pre>	<pre>CREATE OR REPLACE FUNCTION GET_ANNUAL_COMP (P_EMP_SAL IN EMPLOYEES.SALARY%TYPE, P_EMP_COM IN EMPLOYEES.COMMISSION_PCT%TYPE) RETURN EMPLOYEES.SALARY%TYPE AS BEGIN IF (P_EMP_SAL IS NULL) THEN RETURN 0; ELSIF (P_EMP_COM IS NULL) THEN RETURN (P_EMP_SAL*12); ELSE RETURN (P_EMP_SAL*12) + (P_EMP_COM*P_EMP_SAL*12); END IF; EXCEPTION WHEN OTHERS THEN RETURN 0; END GET_ANNUAL_COMP;</pre>
Результат проверки	Сообщение об ошибке
Функция неправильная!	
Неправильные ответы	
Ошибки при разных аргументах: SELECT GET_ANNUAL_COMP(20,40) FROM DUAL -> 8200 <> SELECT GET_ANNUAL_COMP(20,40) FROM DUAL -> 9840 SELECT GET_ANNUAL_COMP(30,;) FROM DUAL -> 300 <> SELECT GET_ANNUAL_COMP(30,;) FROM DUAL -> 360	

Рисунок 4 – Результат проверки неправильной функции с ошибкой в вычислениях.

Таким образом, в ходе проведения тестирования была проверена работоспособность системы. Процесс тестирования других функций, внесенных для проверки, проводился аналогичным образом.

Выводы

Результаты проверки показали, что были достигнуты все поставленные для реализации

задачи. Автоматизированная система проверяет функции как правильные, так и неправильные и в последнем случае выводит соответствующее сообщение об ошибке и список неправильных ответов.

References:

1. Zadvor'ev, I.S. (2017). *Yazyk PL/SQL: uchebno-metodicheskoe posobie*. (p.188). Moscow.
2. Andreeva, N. V., Kozhevnikov, V. A., & Sabinin, O. YU. (2019). *Programmirovaniye baz dannyh: osnovy PL/SQL: uchebnik*. (p.183). SPb.: Izd-vo Politekh. un-ta.
3. Shigina, N. A. (2016). *Prikladnaya informatika. Oblachnye instrumenty razrabotki programmogo obespecheniya v uchebnom processe vuza*, Tom 11, № 2 (62), pp. 41–56.
4. (n.d.). *Oracle APEX Architecture*. Retrieved 17.04.2022 from: <https://apex.oracle.com/en/platform/architecture/>
5. (n.d.). *Oracle Application Express – 2021*. Retrieved 10.05.2021 from https://ru.xcv.wiki/wiki/Oracle_Application_Express
6. (n.d.). A Symbolic Approach to Proving Query Equivalence Under Bag Semantics /Q. Zhou [и др.]. — 2021.— arXiv: Retrieved from <https://arxiv.org/abs/2004.00481>
7. (n.d.). *Semanticheskij poisk: ot prostogo skhodstva ZHakkara k slozhnomu SBERT*. Retrieved from <https://habr.com/ru/company/skillfactory/blog/566414/> (visited on 17.04.2022).
8. Karpova, T. S., & Malysheva, S. YU. (2019). *Intellektual'nye tekhnologii na transporte. Rasshirenie sistem elektronmogo testirovaniya na primere testirovaniya SQL-zaprosov*. (pp.33–40).
9. (n.d.). *Funkcii (FUNCTION) PL/SQL: programmirovaniye na primerah*. Retrieved 10.05.2022 from <https://oracle-patches.com/db/sql/>
10. (n.d.). *Rukovodstvo po Oracle Application Express. Obzor IDE*. — Retrieved 17.05.2022 from <https://habr.com/ru>
11. (n.d.). *Kak nastroit' postranichnuyu avtorizaciyu v Oracle APEX*. Retrieved 10.05.2022 from <https://rtportal.ru/index.php?id=109:oracle-apex-page-auth>