Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»

**Институт компьютерных наук и технологий**

**Кафедра Компьютерные интеллектуальные технологии**

**ОТЧЁТ**

**О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ**

на тему

«Система контроля действий студента на основе технологии Oracle XMLDB»

направление подготовки: 02.03.03 «Математическое обеспечение и администрирование информационных систем»

Выполнила:

студентка группы 43506/1.

Смольянинова Софья Александровна

Подпись

Руководитель:

доцент каф. КИТ ИКНТ, к.т.н.,

Кудрявцев Дмитрий Вячеславович

Подпись

Оценка «\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

Консультант:

доцент каф. КИТ ИКНТ, к.т.н.,

Сабинин Олег Юрьевич

Подпись

Оценка «\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

Санкт-Петербург

2018

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc513238024)

[1. Анализ предметной области и постановка задачи выпускной работы 5](#_Toc513238025)

[1.1. Роль и задачи аудита действий студентов в процессе обучения 5](#_Toc513238026)

[1.2. Виды аудита 7](#_Toc513238027)

[1.2.1. Обязательный аудит (Mandatory auditing) 7](#_Toc513238028)

[1.2.2. Стандартный аудит (Standard database auditing) 7](#_Toc513238029)

[1.2.3. Audit Options 7](#_Toc513238030)

[1.2.4. Value-based auditing 8](#_Toc513238031)

[1.2.5. Fine-grained auditing (FGA пакет) 9](#_Toc513238032)

[1.2.6. DBA auditing 10](#_Toc513238033)

[1.3. Средства для аудита действий пользователей 10](#_Toc513238034)

[1.3.1. Naumen Network Manager 10](#_Toc513238035)

[1.3.2. Netwrix Auditor 11](#_Toc513238036)

[1.4. Требования к системе аудита действий студентов 12](#_Toc513238037)

[1.5. Постановка задачи и пути ее решения 13](#_Toc513238038)

[**ЗАКЛЮЧЕНИЕ** 14](#_Toc513238039)

[**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ** 15](#_Toc513238040)

# **ВВЕДЕНИЕ**

За последние несколько десятилетий сфера информационных технологий сделала большой скачек в своем развитии. Объём информации, которая поступает ежедневно, значительно вырос. В свою очередь, владение актуальной информацией играет значительную роль в современном мире. Встает вопрос, где, а главное каким образом следует хранить такое большое количество информации. Одним из способов решения данного вопроса являются базы данных, которые предоставляют возможности для хранения информации.

Базы данных – это нечто большее чем просто хранилище для информации, это взаимосвязанная система, которая также предоставляет возможности для обработки, анализа и предоставления информации. Надежная и стабильная система является важной составляющей любого бизнеса, в свою очередь любая компанию стремиться к расширению или внедрению новых продуктов, которые влияют на состояние всей системы в целом. Постоянная доступность и актуальность информации напрямую влияет на протекание бизнес-процессов компаний и даже незначительные сбои в работе или неполадки в системе могут привести к серьезным последствиям и убыткам.

Для предотвращения таких ситуаций необходимо вести контроль действий, совершаемых в системе. Отслеживание действий пользователей системы, разработчиков новых продуктов, а также администраторов баз данных помогает в короткие сроки определить причину некорректной работы системы или вовсе ее неработоспособность. Человеческий мозг не способен обрабатывать такое огромное количество информации, соответственно появляется потребность в создании системы контроля действий пользователей баз данных.

Цель данной работы – создание простой и легко интегрируемой с существующей базой данных системы контроля действий пользователей, которая будет собирать информацию о произведенных действиях и некоторую дополнительную информацию. В данной работе в качестве прототипа будет разрабатываться система контроля действий пользователей базы данных. Для удобства доступа к собранной информации необходимо реализовать интерфейс, не привязанный к какому-либо специфическому программному обеспечению.

# Анализ предметной области и постановка задачи выпускной работы

В настоящее время возрастает роль информационных технологий и самой информации в целом. Информация в свою очередь нуждается в хранении, обработке и защите. Данные функции могут быть реализованы с помощью баз данных. В большинстве случаев используются реляционные базы данных, в которых хранение информации осуществляется с помощью плоских таблиц, связанных между собой. Пользователи имеют возможность обращаться к таблицам для получения, изменения и удаления информации. В такой ситуации встает вопрос о сохранении свойств информации таких как целостности, достоверность и т.д. А также стоит рассматривать вопрос поддержании базы данных в рабочем и актуальном состоянии. Для того, чтобы более детально понять поставленную задачу необходимо разобраться с ролью и задачами контроля действий пользователей.

## Роль и задачи аудита действий студентов в процессе обучения

**Аудит —** это мониторинг и запись произведённых действий пользователя базы данных для выбранных объектов [1]. **Также следует понимать, что аудит базы данных представляет собой** комплекс действий, направленный на обеспечение безопасности базы данных и ее контроля [2].

В процессе получения высшего образования студенты изучают ряд дисциплин, которые формируют навыки работы с базами данных. Глобальная дисциплина под названием “Базы Данных” разделена на несколько связанных между собой разделов таких как “Проектирование баз данных”,” Программирование баз данных: SQL” и “Программирование баз данных: PL/SQL”. Далее под пользователями базы данных будем понимать студентов, так как они осуществляют основную работу и активность в базе данных. Подробнее рассмотрим особенности данных дисциплин и знания, получаемые в рамках данных обучения.

“Проектирование баз данных” – это дисциплина, в рамках изучения которой рассматриваются вопросы проектирования баз данных, а именно создания концептуальной, логической и физической модели базы данных, а также нормализация этих моделей. Изучаются вопросы обеспечение возможности получения информации с помощью запросов и вопросы обеспечения целостности базы данных. Следовательно, можно заметить, что в рамках данного курса создается достаточно большое количество объектов базы данных, созданные модели наполняются данными и студенты извлекают информацию из базы данных.

“Программирование баз данных: SQL” в данном курсе изучаются основы написания SQL запросов. В качестве дополнительных заданий выступают олимпиадные задачи, которые требуют смекалки и нестандартного решения тривиальных задач с помощью языка запросов. В данном курсе в большей степени выполняются SQL запросы и создаются таблицы и представления.

“Программирование баз данных: PL/SQL” данный курс имеет большее отношение к программированию, чем предыдущие два. В процессе обучения студентами выполняются задания разной сложности, начиная от создания простых подпрограмм заканчивая созданием работ и сложных триггеров. Порой запущенные работы и некорректно написанные триггеры значительно ухудшают производительность базы данных в целом, что влияет на время выполнения операций для каждого из пользователей. В некоторых случаях, работы могут запускаться не на прямую, например, при работе программа инициирует запуск работы, которая собирает большой объём статистической информации. В таком случае производительность базы данных будет ниже обычной.

Рассматривая все вышеперечисленные дисциплины и задания, выполняемые в рамках данных дисциплин, входящие в один большой курс “Базы Данных” можно заметить, что большое количество студентов работает с базой данных, создает объекты и выполняет SQLзапросы. Поэтому необходимо производить аудит базы данных для того, чтобы при возникновении ситуаций аномальной работы базы данных можно было в достаточно короткое время детектировать причину замедления в работе или вовсе неработоспособность базы данных. Для того, чтобы точно определить, что вызвало аномалии работы базы данных необходимо проанализировать большое количество информации, которая находится в различных представлениях и файлах базы данных. В таком случае администратору базы данных нужен инструмент для проведения мониторинга состояния базы данных. В качестве учебной базы данных используется СУБД Oracle, соответственно для разработки системы контроля действий студентов необходимо рассматривать встроенные инструменты, которые реализованы в СУБД Oracle, и программные продукты которые совместимы с СУБД Oracle.

Аудит чаще всего производится для защиты и сохранения конфиденциальности информации, которая хранится в базах данных [2]. Важность и актуальность данного вопроса неуклонно растет благодаря повсеместному использованию баз данных как в бизнесе, так и в Интернете. СУБД Oracle обеспечивает большие возможности по проведению аудита, что позволяет системным администраторам внедрять расширенные средства защиты, производить раннее обнаружение подозрительных действий.

С другой стороны, аудит действий студентов можно использовать для отслеживания количества времени, проведенного за выполнением того или оного задания, количества используемых ресурсов. Также на основе некоторой информации можно сделать предположение о самостоятельности выполнения заданий.

Роль аудита заключается в автоматическом контроле действий пользователей базы данных. Это значит, что при совершении студентом какого-либо действия, система самостоятельно поместит информацию о произведенных действиях в определенной место, например, файл, таблицу или представление.

Задачей аудита является сбор информации, на основе которой можно сделать некие выводы и предположения. Например, о состоянии базы данных в целом, времени выполнений каких-либо операций или о количестве израсходованных ресурсов. Рассмотрев задачу аудита в целом можно сделать вывод о том, что эта задача является крайне сложной. Решение данной задачи возможно с помощью встроенных инструментов СУБД Oracle или же с использованием внешних средств - некого программного обеспечения. Теперь рассмотрим несколько видов аудита, реализованных в СУБД Oracle, а также примеры внешних средств, которые совместимы с работой СУБД Oracle.

## Виды аудита

### Обязательный аудит (Mandatory auditing)

Некоторые действия пользователей всегда подвергаются аудиту. Например, подключение к базе данных как привилегированный пользователь или включение и выключение базы данных [5].

### Стандартный аудит (Standard database auditing)

Аудит, основанный на командах. Для начала работы со стандартным аудитом необходимо включить возможность проведения аудита и указать параметры аудита (событий входа, использование системных и объектных привилегий или использования операторов SQL) [5].  
Параметр AUDIT\_TRAIL включает и также отключает аудит базы данных.

### Audit Options

Для проведения аудита действий над объектами базы данных необходимо использовать команду CREATE AUDIT POLICY [3].

— SQL statement

Команда AUDIT TABLE позволяет проводить аудит для любых команд DDL, которые влияют на таблицу, включая команды CREATE TABLE, DROP TABLE, TRUNCATE TABLE и т.д [4]. Также можно определить действия какого пользователя будут подвергаться аудиту и успешность выполненной команды:

SQL> AUDIT TABLE BY hr WHENEVER NOT SUCCESSFUL;

Данный пример демонстрирует аудит любых команд DDL, совершаемых пользователем hr, которые потерпели неудачу.

— Аудит системных привилегий

Может использоваться для аудита выполнения любых системных привилегий (таких как DROP ANY TABLE) [3]. По умолчанию данный вид аудита использует условие BY ACCESS, это значит, что каждый раз, когда выполняется данная системная привилегия, создается запись аудита. Есть возможность группировать эти записи с используя условие BY SESSION, чтобы только одна запись генерировалась в течении сессии.

— Аудит объектных привилегий

Используется для аудита действий над таблицами, представлениями, процедурами, последовательностями и пользовательскими типами данных [3]. В отличие от аудита системных привилегий, по умолчанию

группировка строк осуществляется по условию BY SESSION. Явно указание условия BY ACCESS, приводит к генерации строк аудита для каждого действия.

### Value-based auditing

Аудит, основанный на триггерах для записи изменяемых значений при вставке, обновлении и удалении, информации в проверяемых объектах [5].

Пользователь выполняет команды INSERT, UPDATE, DELETE для информации, которая содержится в таблице с соответствующим подключенным триггером. В такой ситуации триггер работает в фоновом режиме для того, чтобы скопировать требуемые данные в таблицу, предназначенную для хранения информации аудита. Аудит, основанный на триггерах, имеет тенденцию ухудшать производительность базы данных больше, чем стандартный аудит, поскольку триггер срабатывает каждый раз, при вставке, обновлении или удалении информации. Степень ухудшения производительности зависит от эффективности триггерного кода. Аудит на основе триггеров должен использоваться только в тех ситуациях, когда информации, полученной с помощью стандартного аудита недостаточно.

### Fine-grained auditing (FGA пакет)

Аудит, основанный на пакете FGA. Аудит базы данных регистрирует тот факт, что операция произошла, но не фиксирует информацию о самой команде [5]. FGA аудит расширяет возможности контроля и мониторинга базы данных и позволяет захватывать операторы SQL, которые запрашивают или обрабатывают данные. FGA проводит аудит команд SELECT, INSERT, UPDATE, DELETE и MERGE. Параметры FGA могут быть направлены на отдельные столбцы таблицы или представления, также поддерживается более одного соответствующего столбца. Также можно добавить условие для того, чтобы информация фиксировалась только в том случае, если определенные администратором спецификации встретились. По умолчанию, если любой из определенных столбцов присутствует в операторе SQL, он проверяется. DBMS\_FGA.ALL\_COLUMNS и DBMS\_FGA.ANY\_COLUMNS предоставляют информацию, которая дает понять на основе любых или всех соответствующих столбцов будет проводиться аудит.

— Определение политики аудита

Для создания политики аудита в целевой таблице или представлении рекомендуется использовать PL / SQL пакет DBMS\_FGA [3]. Если строки, возвращаемые запросом, соответствуют проверяемому столбцу и указанному условию аудита, событие аудита создает запись и сохраняет ее audit\_trail.FGA автоматически фокусирует аудит на уровне команд. Важно заметить, что оператор SELECT, который возвращает тысячи строк, генерирует только одну запись аудита. Для проведения FGA аудита необходимо настроить политику аудита, которая включает в себя параметры аудита, такие как схема и имя объекта, имя политика аудита, условие и столбцы аудита, команды, подлежащие аудиту. Так же событие аудита может автоматически запускать процедуру, имя и схему, которой необходимо определить в параметре HANDLER\_MODULE и HANDLER\_SCHEMA соответственно.

### DBA auditing

Позволяет производить отслеживание действий привилегированных пользователей. Пользователи SYSDBA и SYSOPER имеют привилегии для старта и остановки базы данных (SHUTDOWN и STARTUP) [5]. Так как эти пользователи могут производить изменения при выключенной базе данных, audit trail должен храниться вне базы данных. База данных Oracle всегда фиксирует события входа в систему привилегированных пользователей посредством обязательного аудита. Также она может фиксировать и другие действия, если аудит DBA включен. Включить аудит для пользователей SYSDBA и SYSOPER можно, установив параметр инициализации:

AUDIT\_SYS\_OPERATIONS = TRUE (По умолчанию FALSE.)

Если операции SYS подвергаются контролю, параметр инициализации AUDIT\_FILE\_DEST содержит адрес хранения записей аудита.

На платформе Windows audit trail по умолчанию использует Windows event log. На платформах UNIX и Linux записи аудита хранятся в $ORACLE\_HOME/RDBMS/audit.

## Средства для аудита действий пользователей

Для проведения контроля действий студента и мониторинга производительности базы данных можно исследовать уже готовые программные продукты. Рассмотрим некоторые из них.

### Naumen Network Manager

Данный программный продукт позволяет администраторам диагностировать и устранять сбои в работе баз данных, а также обнаруживать и исправлять проблемы с производительностью [6]. Система использует драйвер JDBC/ODBC, что обеспечивает подключение ко всем популярным типам баз данных, включая Oracle, MySQL, Microsoft SQL Server, Firebird, Sybase, IBM DB2 и др.

В данном случае аудит осуществляется с помощью заданных пользователем SQL-запросов. Полученные результаты переводятся в специальный формат Naumen Network Manager и сохраняются в базе данных. Данные запросы могут генерироваться и динамически с помощью специального языка выражений Naumen Network Manager на добавление, обновление и удаление данных.

Кроме того, можно отслеживать производительность баз данных в реальном времени, например, назначив тревогу на превышение порога времени выполнения определенного запроса.

### Netwrix Auditor

Программный продукт Netwrix Auditor предоставляет подробную аналитику по событиям в базах данных: изменения объектов БД, ролей и прав доступа, данные по удачным и неудачным попыткам доступа к рабочим базам данных и структурированным данным [7]. Этот программный продукт также совместим с различными базами данных в том числе и Oracle.

Данный программный продукт может выполнять следующий действия:

— оповещать о частых изменениях файлов или большом количестве попыток доступа, совершенных за короткий промежуток времени;

— осуществлять непрерывный мониторинг базы данных, получая всю информацию о любом изменении;

— создавать детальные отчеты о действиях пользователей и администраторов в базе данных;

— осуществлять гибкую фильтрацию и поиск необходимых событий, а также возможна подписка на отчеты;

— производить работы без снижения производительности системы.

Данные программные продукты имеют достаточно большой функционал и могут быть использованы для решения разнообразных задач. Следует заметить, что это достаточно дорогостоящие программные продукты, которые преимущественно используются в больших корпорациях и данные денежные и ресурсные затраты нецелесообразны в рамках аудита базы данных, которой пользуются студенты одной кафедры. Даже совокупность работы этих программ не сможет удовлетворить всем нижеперечисленным требованиям.

## Требования к системе аудита действий студентов

Для поддержания корректной работы базы данных, а также сбора дополнительной информации о действиях студента в базе данный необходимо определить какие аспекты работы студентов необходимо подвергать контролю.

Проанализировав работу базы данных за длительный промежуток времени, было выявлено, что для корректной работы базы данных и контроля действий студентов необходимо отслеживать и записывать информацию о:

— времени подключения и отключения студентов, а также простоя со время сессии;

— количестве израсходованных ресурсов, например, памяти и дисковом месте;

— общее время работы и простоя на заданный интервал времени у одного или нескольких студентов;

— созданных и измененных пользователем объектах, таких как таблицы, представления, синонимы, триггеры и пакеты;

— запущенных триггерах, работах, заданиях как за весь период, так и в заданный интервал времени;

— состояние заданий и триггеров на текущей момент времени;

— созданных пользователях, выданных и изъятых правах, как за весь период, так и в заданный интервал времени;

— выполненных студентами командах SELECT.

Необходимо создать возможность получения данной информации как для определенного студента, так и для всех студентов единовременно. Также должна быть возможность просматривать информацию за определенный промежуток времени или же за весь период работы базы данных.

## Постановка задачи и пути ее решения

Для решения задачи поддержания базы данных в рабочем состоянии, сохранения высокой скорости выполнения операций, а также анализа работы студентов необходимо отслеживать действия студентов. Большая часть из вышеперечисленных требований может быть реализована при помощи корректно спроектированной системы аудита, которая предоставит необходимую информацию в доступной форме. Рассмотрев уже готовые программные продукты и требования, выдвинутые к системе контроля действий студента можно сделать выводы о том, что функционал этих программ слишком обширен и их использование нецелесообразно как с точки зрения итоговой производительности базы данных, так и с точки зрения материальных затрат на приобретение лицензии. Данная система должна использовать комбинацию из нескользких стандартных средств аудита, представленный в базе данных Oracle, для эффективной работы. Также разработанная система должна не сильно снижать производительность всей системы в целом. Для доступа к информации о проведенном аудите необходимо обеспечить доступ к базе данных извне, информация должна быть доступна для просмотра, без помощи программного обеспечения для работы с базой данных. Одним из вариантов решения поставленной задачи является реализация системы с интерфейсом, в котором информацию можно просматривать с помощью веб-браузера.

# **ЗАКЛЮЧЕНИЕ**

В данной работы была проведен анализ предметной области – системы контроля действий пользователей баз данных, где были рассмотрены основные аспекты, на которые стоит обратить при разработке подобной системы. На основе анализа предметной области были определены требования, которым должна подчиняться разрабатываемая система.

Для выбора инструментов разработки был проведен сравнительный анализ возможностей аудита СУБД Oracle и анализ возможностей сторонних программный обеспечений.

Таким образом в результате данной работы были сформированы критерии, которым должна соответствовать разрабатываемая система и определены инструменты для ее разработки.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Oracle Database Online Documentation 12c Release 1 (12.1) / Database Administration / Database Security Guide / Auditing Database Activity : [Электронный ресурс] // Oracle Help Center [сайт]. [2017] URL: <https://docs.oracle.com/database/121/TDPSG/GUID-BF747771-01D1-4BFB-8489-08988E1181F6.htm#TDPSG50000>
2. Oracle Database Online Documentation 12c Release 1 (12.1) / Database Administration/ Introduction to Auditing: [Электронный ресурс] // Oracle Help Center [сайт]. [2017] URL: <https://docs.oracle.com/database/121/DBSEG/auditing.htm#DBSEG60612>
3. Oracle Database Online Documentation 12c Release 1 (12.1) / Database Administration/Configuring Audit Policies: [Электронный ресурс] // Oracle Help Center [сайт]. [2017] URL: <https://docs.oracle.com/database/121/DBSEG/audit_config.htm#DBSEG1025>
4. Oracle Database Online Documentation 12c Release 1 (12.1) / Database Administration/ AUDIT (Unified Auditing) : [Электронный ресурс] // Oracle Help Center [сайт]. [2017] URL: https://docs.oracle.com/database/121/SQLRF/statements\_4008.htm#SQLRF5611
5. Ben-Natan, Ron. How to secure and audit Oracle 10g and 11g // Boca Raton, Fla. : Auerbach ; London : aylor & Francis, 2009. — C.187-253.
6. Функциональные возможности Naumen Network Manager / Мониторинг баз данных — naumen.ru:[Электронный ресурс] URL: <https://www.naumen.ru/products/network_manager/functions/monitoring-baz-dannykh/>. Дата обращения: 01.05.2018.
7. Программное обеспечение для аудита Oracle Database— netwrix.ru:[Электронный ресурс] URL: <https://www.netwrix.ru/oracle_database_monitoring_and_auditing.html> . Дата обращения: 01.05.2018.