Оглавление

1. Анализ задачи аудита информационной безопасности информационной системы

1.1 Введение

Необходимость разработки проекта «Разработка подсистемы аудита информационной системы» и соответственно создание программного решения в рамках него обусловлено тем, что повышаются современные требования, предъявляемые к определению уровня обеспечения информационной безопасности. А существенный рост рисков потерь от нарушения информационной безопасности во всех сферах обязывают разрабатывать и использовать в работе предприятий современные и обоснованные методы и средства, позволяющие количественно и качественно измерять уровень защищенности организаций и систем информационной технологий, а также оценивать экономическую эффективность затрат на информационную безопасность.

На сегодняшний день СУБД играют ключевую роль в обеспечении эффективного выполнения процессов предприятий. Вместе с тем повсеместное использование СУБД для хранения, обработки и передачи информации приводит к повышению актуальности проблем, связанных с их защитой. Именно для решения этих проблем и применяется аудит безопасности системы.

Конкуренция на данном рынке существует, но решения в основном представлены только в платных СУБД. Существует возможность использовать стандартные средства аудита таких СУБД как: Oracle, MS SQL Server и т.д. Для бесплатных же СУБД, таких, к примеру, как PostgreSQL, разработанных проектов пока нет, а стандартные средства имеют ограниченную функциональность и являются менее удобными в применении.

Цель данной работы — на основе анализа информационной безопасности информационных систем, а также на основе существующих моделей и средств для проведения аудита в системах управления базами данных, разработать решение повышающую безопасность информационной системы.

Основной задачей является разработка подсистемы аудита информационной системы, использующей СУБД PostgreSQL. Данную задачу можно решить путём создания подсистемы аудита, которая позволит обнаружить действия, нарушающие целостность основной систем, другими словами, с помощью данной подсистемы можно выявить как мошеннически введенные данные, так и несанкционированные запросы.

Объектом исследования является информационная система, использующая системы управления базами данных.

Разработка выполнена на, распространенной на рынке, бесплатной СУБД PostgreSQL, что позволяет широко использовать данный проект. Также надо отметить, что разработанная система получается менее затратной, чем аналогичные системы, базирующиеся на других технологиях и являющиеся изначально платными.

1.2 Информационная безопасность

В настоящее время организации все больше зависят от информации, которую они используют. Если подвергать риску эту информацию с точки зрения потерь или несанкционированного доступа конкурентов, то это может привести к разрушительным последствиям.

Таким образом, на данный момент одной из серьезных проблем стало управление информационной безопасностью. Управление информационной безопасностью основывается на множестве политик и внутреннем контроле, с помощью которых организация и управляет своей информационной безопасностью. Информация и системы (ИС), которые обрабатывают ее, имеют решающее значение в работе практически всех организаций. Информация становится все более уязвимой для большого количества рисков, которые могут поставить под угрозу всю организацию. Это приводит к тому, что необходимо принимать сложные решения о том, как сделать информационную безопасность эффективней [12].

Информационная безопасность — это защищенность информации и инфраструктуры, которая ее поддерживает, от преднамеренных или случайных воздействий естественного или искусственного характера, которые могут нанести ущерб владельцам или пользователям информации [14].

Важной частью общей системы управления, необходимой для достижения уставных целей и задач, является обеспечение собственной информационной безопасности на предприятиях.

Значимость деятельности направленной на обеспечение информационной безопасности становится тем более высокой, чем выше степень автоматизации процессов предприятия и чем больше в конечном продукте интеллектуальная составляющая [15].

Обеспечение информационной безопасности имеет довольно значимое значение не только для основных направлений, развитие предприятия, создание основного продукта, но и для отдельных и вспомогательных направлений деятельности и процессов, которыми могут быть, например, коммерческие переговоры и условия контрактов, ценовая политика и т.п. [15].

Так же, значимость обеспечения информационной безопасности в некоторых случаях может повышаться в связи с тем, что в общей системе предприятия есть сведений, составляющие не только коммерческую, но и *государственную тайну*, а также другие виды конфиденциальной информации (сведения, составляющие *банковскую тайну*, врачебную тайну, интеллектуальную собственность компаний-партнеров и т.п.). Федеральное законодательство регламентирует обеспечение информационной безопасности в этой сфере и, в частности, основные требования, организационные правила и процедуры непосредственно, а федеральные органы власти осуществляют контроль над выполнением этих требований.

- для сведений, составляющих *государственную тайну* — Федеральный закон РФ от 21 июля 1993 года №5485-1 "О *государственной тайне*" и связанные с ним подзаконные акты.
- для сведений, составляющих *банковскую тайну* — Федеральный закон "О банках и банковской деятельности" и связанные с ним смежные законы и подзаконные акты.
- для сведений, составляющих врачебную тайну — Основы законодательства РФ "Об охране здоровья граждан" (ст.61) и Закон РФ "О трансплантации органов и (или) тканей человека" (ст.14) [15];

Управление информационной безопасностью на уровне предприятий, как и на государственном уровне, направлено на то, чтобы нейтрализовать различные виды угроз:

- внешних, таких как неправомерные действия государственных органов (в том числе и зарубежных), противоправная деятельность преступников и преступных группировок, незаконные действия компаний-конкурентов и других хозяйствующих субъектов, недобросовестные действия компаний-партнеров, несоответствие действующей нормативно-правовой базы фактическому развитию технологий и общественных отношений, сбои и нарушения в работе глобальных информационных и телекоммуникационных систем и информационных систем компаний-партнеров и др.;
- внутренних, таких как ошибки и халатность персонала предприятия, а также намеренно допускаемые нарушения, сбои и нарушения в работе собственных информационных систем и др.[15]

Некоторые обстоятельства приводят к необходимости разработки и внедрения политики информационной безопасности, такие как:

- необходимость уменьшения стоимости страхования информационных рисков или определенных бизнес-рисков;

- необходимость внедрения международных стандартов, таких как *ISO 17799* или BS 7799.

Этапы разработки политики безопасности представлены на рисунке 1.

Рисунок 1 — Разработка политики безопасности

Для того чтобы нейтрализовать угрозы, которые имеются на нынешний момент, а также для обеспечения информационной безопасности предприятия в сфере информационной безопасности организуют систему менеджмента, в рамках этой системы проводят работу по нескольким направлениям [15]:

- формирование и практическая реализация комплексной многоуровневой политики информационной безопасности предприятия и системы внутренних требований, норм и правил;
- организация отдела информационной безопасности;
- разработка системы мер и действий на случай возникновения непредвиденных ситуаций
- проведение аудитов, комплексных проверок состояния информационной безопасности на предприятии [15].

Управление ИБ предприятия представлены на рисунке 2.

Рисунок 2 — Схема управления информационной безопасности

Рассмотрим подробней структуру управления ИБ [15].

Организационными документами, регулирующими деятельность всей организации или отдельных подразделений, а также некоторых категорий сотрудников в сфере обращения с информационными системами и информационными потоками являются политики и правила информационной безопасности

Отдел информационной безопасности является узкоспециализированным подразделением, решающим определенные вопросы защиты информации.

Разработка системы по реагированию на инциденты обеспечивает готовность всей организации (включая отдел информационной безопасности) к целенаправленным действиям в случае происшествий, связанных с информационной безопасностью.

Контроль за текущим состоянием системы мер по защите информации, в частности, независимую проверку соответствия реального положения дел установленным правилам и требованиям обеспечивает проведение внутренних аудитов информационной безопасности

Вначале необходимо провести аудит информационных процессов фирмы, выявить критически важную информацию, которую необходимо защищать. Список конфиденциальной информации предприятия, участков, где эта информация обращается, допущенных к ней лиц, а также последствий утраты или искажения этой информации — результат аудита информационных процессов. После этого становится ясно, какие объекты необходимы, защитить и от кого: ведь в большинстве инцидентов в качестве нарушителей будут выступать сами сотрудники фирмы. Различным угрозам безопасности можно присвоить вероятности их реализации. Умножив вероятность реализации угрозы

на причиняемый этой реализацией ущерб, получим риск угрозы. После чего можно приступать к разработке политики безопасности.

Под аудитом в общем случае понимается проверка соответствия некоторого объекта оценки определенным требованиям. В области защиты информации употребляется термин «аудит информационной безопасности», не определенный пока ни в одном документе РФ [33].

В данный момент нормы аудита информационной безопасности (ИБ) не получили правового закрепления. Решением этой проблемы может стать принятие стандарта аудита ИБ. Основу стандарта могут составить Федеральный закон «Об аудиторской деятельности», Федеральные правила (стандарты) аудиторской деятельности, а также «Положение по аттестации объектов информатизации по требованиям безопасности информации». Закон устанавливает ответственность аудиторской организации за качество и достоверность аудита, Правила обеспечивают аудиторов соответствующими инструментами аудита, а Положение выделяет организационную структуру аудита и порядок его проведения. Также стоит обратить внимание на развитие семейства стандартов специализированных методологий IDEF и планируемое принятие стандарта IDEF7: Information System Auditing [33].

Более подробно аудит ИБ будет рассмотрен в следующей главе.

Каждое из вышеописанных направлений деятельности в соответствии с изменением в организационной структуре, производственных процессах или внешней среде, должны постоянно совершенствоваться по мере развития организации, а конкретные задачи должны постоянно уточняться. Так, например, если предприятие начинает выпуск продукции военного назначения параллельно с выпуском гражданской продукции, то это может потребовать изменений всех основных направлений организационной работы в сфере обеспечения информационной безопасности [15]:

- корректировки стратегии и основных положений политики информационной безопасности (на всех ее уровнях);
- изменения организационной структуры и функциональных задач департамента информационной безопасности;
- совершенствования системы реагирования на инциденты;
- использование более совершенных методик проведения аудитов информационной безопасности.

1.3 Информационная безопасность информационных систем

Цели информационной безопасности, как правило, считается выполненным, если:

- Информационные системы имеются в наличии и готовы к использованию при необходимости;
- Данные и информация раскрывается только тем, кто имеет право их знать (конфиденциальность);
- Данные и информация защищена от несанкционированного изменения (целостность).

Организации должны реализовать эти цели, чтобы удостовериться, что их ценная информация защищается от возможных потерь, недоступности, изменения или неправомерного раскрытия.

Следовательно, обеспечение информационной безопасности связано с решением трех задач:

- Обеспечением доступности информации.
- Обеспечением целостности информации.
- Обеспечением конфиденциальности информации.

Именно доступность, целостность и конфиденциальность являются равнозначными составляющими информационной безопасности. Информационные системы создаются для получения определенных информационных услуг. Если по тем или иным причинам предоставить эти услуги пользователям становится невозможно, то это, очевидно, наносит ущерб всем.

Доступность — это своевременное получения требуемой информации пользователем.

При том, что в ряде случаев фактор времени является очень важным, поскольку некоторые виды информации и информационных услуг имеют смысл только в определенный промежуток времени.

Целостность информации условно можно разделить на статическую и динамическую. Статическая целостность информации предполагает неизменность информационных объектов по сравнению с их исходным состоянием. Динамическая целостность информации включает вопросы корректного выполнения сложных действий с информационными потоками. Целостность является важнейшим пунктом в информационной безопасности в тех случаях, когда информация используется для управления различными процессами. Целостность — определяет то, что информация сейчас существует в ее исходном виде, то есть при ее хранении или передаче не было произведено несанкционированных изменений.

В различных информационных системах в обязательном порядке конфиденциальными данными являются пароли для доступа к ним. Конфиденциальность — гарантия доступности конкретной информации только тому кругу лиц, для кого она предназначена.

Нарушение каждой из трех категорий приводит к нарушению информационной безопасности в целом. Так, нарушение доступности приводит к отказу в доступе к информации, нарушение целостности приводит к фальсификации информации и, наконец, нарушение конфиденциальности приводит к раскрытию информации.

Цели ИБ представлены на рисунке 3.

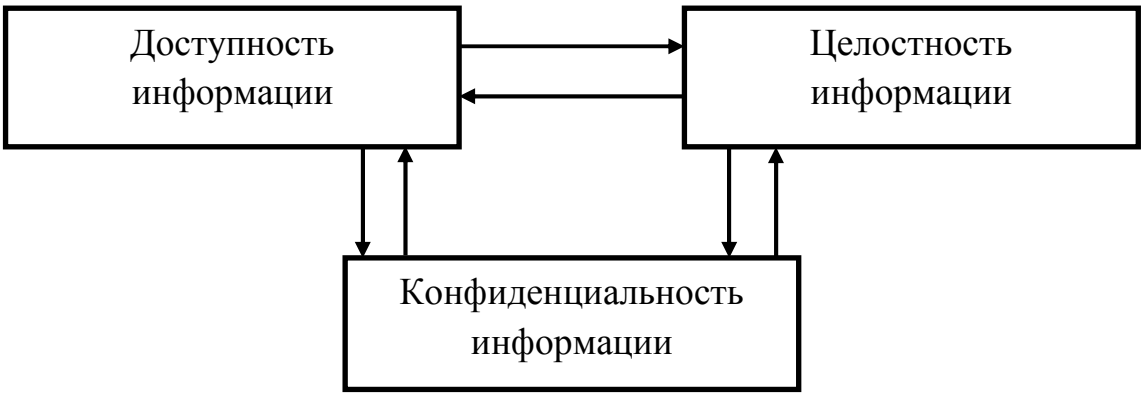

Рисунок 3 — Цели ИБ

Для того чтобы реализовать комплексный подход для обеспечения режима информационной безопасности как раз и необходимо выделение этих категорий в качестве базовых составляющих информационной безопасности. Кроме этого, нарушение одной из этих категорий может привести к нарушению или полной бесполезности двух других [12][16][34][35][36].

1.4 Механизмы обеспечения информационной безопасности информационных систем

- Идентификация и аутентификация
- Криптография и шифрование
- Методы разграничение доступа
- Регистрация и аудит

Идентификация и аутентификация

Структура идентификация и аутентификации представлена на рисунке 4.

Рисунок 4 — Структура идентификации и аутентификации

Общий алгоритм работы таких систем заключается в том, чтобы получить от субъекта (например, пользователя) информацию, удостоверяющую его личность, проверить ее подлинность и затем предоставить (или не предоставить) этому пользователю возможность работы с системой.

Наличие процедур аутентификации и/или идентификации пользователей является обязательным условием любой защищенной системы, поскольку все механизмы защиты информации рассчитаны на работу с поименованными субъектами и объектами информационных систем.

Дадим определения этих понятий.

Идентификация — присвоение субъектам и объектам доступа личного идентификатора и сравнение его с заданным.

Аутентификация (установление подлинности) — проверка принадлежности субъекту доступа предъявленного им идентификатора и подтверждение его подлинности. Другими словами, аутентификация заключается в проверке: является ли подключающийся субъект тем, за кого он себя выдает.

При построении систем идентификации и аутентификации возникает проблема выбора идентификатора, на основе которого осуществляются процедуры идентификации и аутентификации пользователя. В качестве идентификаторов обычно используют:

- набор символов (пароль, секретный ключ, персональный идентификатор и т. п.), который пользователь запоминает или для их запоминания использует специальные средства хранения (электронные ключи);
- физиологические параметры человека (отпечатки пальцев, рисунок радужной оболочки глаза и т. п.) или особенности поведения (особенности работы на клавиатуре и т. п.) [34][38][35][39][40][41][16].

Криптография и шифрование

Структура криптографии и шифрования представлена на рисунке 5.

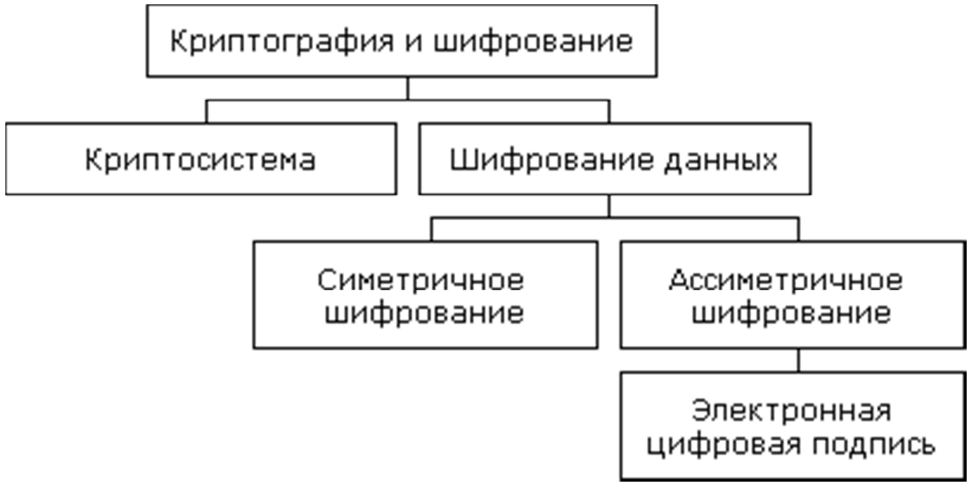

Рисунок 5 — Структура криптографии и шифрования

Структура криптосистемы представлена на рисунке 6.

Общая схема работы криптосистемы показана на рис

Рисунок 6 — Общая схема криптосистем

Криптосистемы решают такие проблемы информационной безопасности как обеспечение конфиденциальности, целостности данных, а также аутентификацию данных и их источников.

Криптографические методы защиты являются обязательным элементом безопасных информационных систем. Особое значение криптографические методы получили с развитием распределенных открытых сетей, в которых нет возможности обеспечить физическую защиту каналов связи [34][38][35][39][40][41].

Классификация систем шифрования данных

Основным классификационным признаком систем шифрования данных является способ их функционирования. По способу функционирования системы шифрования данных делят на два класса:

- системы "прозрачного" шифрования;
- системы, специально вызываемые для осуществления шифрования.

В системах "прозрачного" шифрования (шифрование "налету") криптографические преобразования осуществляются в режиме реального времени, незаметно для пользователя. Системы второго класса обычно представляют собой утилиты (программы), которые необходимо специально вызывать для выполнения шифрования.

Симметричные и асимметричные методы шифрования

Общая технология использования симметричного метода шифрования представлена на рисунке 7.

Рисунок 7 — Общая технология использования метода

Классические криптографические методы делятся на два основных типа: симметричные (шифрование секретным ключом) и асимметричные (шифрование открытым ключом).

В симметричных методах для шифрования и расшифровывания используется один и тот же секретный ключ.

Основной недостаток этого метода заключается в том, что ключ должен быть известен и отправителю, и получателю. Это существенно усложняет процедуру назначения и распределения ключей между пользователями. Указанный недостаток послужил причиной разработки методов шифрования с открытым ключом — асимметричных методов.

Асимметричные методы используют два взаимосвязанных ключа: для шифрования и расшифрования. Один ключ является закрытым и известным только получателю. Его используют для расшифрования. Второй из ключей является открытым, т. е. он может быть общедоступным по сети, и опубликован вместе с адресом пользователя. Его используют для выполнения шифрования. Схема функционирования данного типа криптосистемы показана на рисунке 8.

Рисунок 8 — Схема функциональности

Механизм электронной цифровой подписи

Для контроля целостности передаваемых по сетям данных используется электронная цифровая подпись, которая реализуется по методу шифрования с открытым ключом.

Электронная цифровая подпись представляет собой относительно небольшое количество дополнительной аутентифицирующей информации, передаваемой вместе с подписываемым текстом. Отправитель формирует цифровую подпись, используя

секретный ключ отправителя. Получатель проверяет подпись, используя открытый ключ отправителя.

Идея технологии электронной подписи состоит в следующем. Отправитель передает два экземпляра одного сообщения: открытое и расшифрованное его закрытым ключом (т. е. обратно шифрованное). Получатель шифрует с помощью открытого ключа отправителя расшифрованный экземпляр. Если он совпадет с открытым вариантом, то личность и подпись отправителя считается установленной.

При практической реализации электронной подписи также шифруется не все сообщение, а лишь специальная контрольная сумма — хэш, защищающая послание от нелегального изменения. Электронная подпись здесь гарантирует как целостность сообщения, так и удостоверяет личность отправителя.

Методы разграничение доступа

Рисунок 9 — Метода разграничения доступа

После выполнения идентификации и аутентификации подсистема защиты устанавливает полномочия (совокупность прав) субъекта для последующего контроля санкционированного использования объектов информационной системы.

Обычно полномочия субъекта представляются: списком ресурсов, доступным пользователю и правами по доступу к каждому ресурсу из списка.

Существуют следующие методы разграничения доступа:

- Разграничение доступа по спискам.
- Использование матрицы установления полномочий.
- Разграничение доступа по уровням секретности и категориям.
- Парольное разграничение доступа.

При разграничении доступа по спискам задаются соответствия: каждому пользователю — список ресурсов и прав доступа к ним или каждому ресурсу — список пользователей и их прав доступа к данному ресурсу.

Списки позволяют установить права с точностью до пользователя. Здесь нетрудно добавить права или явным образом запретить доступ. Списки используются в подсистемах безопасности операционных систем и систем управления базами данных.

Использование матрицы установления полномочий подразумевает применение матрицы доступа. В указанной матрице строками являются идентификаторы субъектов, имеющих доступ в информационную систему, а столбцами — объекты (ресурсы) информационной системы. Каждый элемент матрицы может содержать имя и размер предоставляемого ресурса, право доступа, ссылку на другую информационную структуру, уточняющую права доступа, ссылку на программу, управляющую правами доступа и др.

Недостатками матрицы являются ее возможная громоздкость и неоптимальность (большинство клеток — пустые).

Разграничение доступа по уровням секретности и категориям заключается в разделении ресурсов информационной системы по уровням секретности и категориям.

Парольное разграничение, очевидно, представляет использование методов доступа субъектов к объектам по паролю. При этом используются все методы парольной защиты [34][38][42][35].

Мандатное и дискретное управление доступом

В ГОСТе Р 50739-95 "Средства вычислительной техники. Защита от несанкционированного доступа к информации" и в документах Гостехкомиссии РФ определены два вида (принципа) разграничения доступа:

- дискретное управление доступом;
- мандатное управление доступом.

Дискретное управление доступом представляет собой разграничение доступа между поименованными субъектами и поименованными объектами. Субъект с определенным правом доступа может передать это право любому другому субъекту. Данный вид организуется на базе методов разграничения по спискам или с помощью матрицы.

Мандатное управление доступом основано на сопоставлении меток конфиденциальности информации, содержащейся в объектах (файлы, папки, рисунки) и официального разрешения (допуска) субъекта к информации соответствующего уровня конфиденциальности.

Регистрация и аудит

Схема регистрации и аудита представлена на рисунке 10.

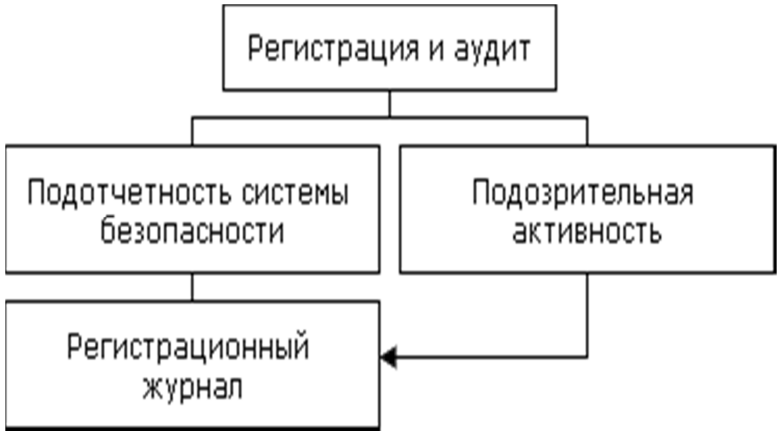

Рисунок 10 — Схема регистрации и аудита

Регистрация является еще одним механизмом обеспечения защищенности информационной системы. Этот механизм основан на подотчетности системы обеспечения безопасности, фиксирует все события, касающиеся безопасности, такие как:

- вход и выход субъектов доступа;
- запуск и завершение программ;
- выдача печатных документов;
- попытки доступа к защищаемым ресурсам;
- изменение полномочий субъектов доступа;
- изменение статуса объектов доступа и т. д.

Для сертифицируемых по безопасности информационных систем список контролируемых событий определен рабочим документом Гостехкомиссии РФ: "Положение о сертификации средств и систем вычислительной техники и связи по требованиям безопасности информации".

Эффективность системы безопасности принципиально повышается в случае дополнения механизма регистрации механизмом аудита. Это позволяет оперативно выявлять нарушения, определять слабые места в системе защиты, анализировать закономерности системы, оценивать работу пользователей и т. д.

Аудит — это анализ накопленной информации, проводимый оперативно в реальном времени или периодически (например, раз в день). Оперативный аудит с автоматическим реагированием на выявленные нештатные ситуации называется активным.

Реализация механизмов регистрации и аудита позволяет решать следующие задачи обеспечения информационной безопасности:

- обеспечение подотчетности пользователей и администраторов;
- обеспечение возможности реконструкции последовательности событий;
- обнаружение попыток нарушений информационной безопасности;
- предоставление информации для выявления и анализа проблем.

Рассматриваемые механизмы регистрации и аудита являются сильным психологическим средством, напоминающим потенциальным нарушителям о неотвратимости наказания за несанкционированные действия, а пользователям — за возможные критические ошибки.

Практическими средствами регистрации и аудита являются:

- различные системные утилиты и прикладные программы;
- регистрационный (системный или контрольный) журнал.

Организация регистрации событий, связанных с безопасностью информационной системы включает как минимум три этапа:

- сбор и хранение информации о событиях.
- защита содержимого журнала регистрации.
- анализ содержимого журнала регистрации.

На первом этапе определяются данные, подлежащие сбору и хранению, период чистки и архивации журнала, степень централизации управления, место и средства хранения журнала, возможность регистрации шифрованной информации и др.

Регистрируемые данные должны быть защищены, в первую очередь, от несанкционированной модификации и, возможно, раскрытия.

Самым важным этапом является анализ регистрационной информации. Известны несколько методов анализа информации с целью выявления несанкционированных действий.

Статистические методы основаны на накоплении среднестатистических параметров функционирования подсистем и сравнении текущих параметров с ними. Наличие определенных отклонений может сигнализировать о возможности появления некоторых угроз.

Эвристические методы используют модели сценариев несанкционированных действий, которые описываются логическими правилами или модели действий, по совокупности, приводящие к несанкционированным действиям [34][38][35][40].

1.5 Выводы

В данной главе были рассмотрены понятия информационная безопасность в общем смысле, виды угроз и насколько важно поддерживать информационную безопасность организации, именно для этого был предоставлен список необходимых организационных мер для поддержания и осуществления безопасности на предприятии. Так же были даны понятия информационная безопасность информационных систем, какие цели и задачи они преследует. А также широко рассмотрены сами механизмы обеспечения информационной безопасности информационных систем. В каждом из двух разделов было дано определение аудита, как одного из важных составляющих информационной безопасности, в следующей главе это понятие будет рассмотрено более широко.

2. Аудит информационной безопасности, этапы и подсистемы аудита

2.1 Аудит информационной безопасности

В современном мире информации необходимо использовать в работе предприятий обоснованные технические и экономические методы и средства, которые позволят количественно и качественно измерить уровень защищенности организаций и систем информационной технологий, а также оценить экономическую эффективность затрат на информационную безопасность. Необходимость разработки и использования этих методов и средств обусловлена современными требованиями бизнеса, предъявляемые к определению уровня обеспечения информационной безопасности, и существенным ростом рисков потерь (материальных, финансовых, моральных, информационных) от нарушения информационной безопасности во всех сферах жизнедеятельности общества и государства.

Аудит информационной безопасности, цель которого — установление того, насколько выполнены требования по обеспечению состояния защищенности системы информационных технологий, является одним из направлений, позволяющих оценить уровень обеспечения информационной безопасности.

Чтобы эти цели были удовлетворены и что никаких недостатков системы не существуют, стоит рассмотреть принципы аудита средств управления информационной безопасности и как они могут помочь организациям обеспечить

Во-первых, важна причина, почему аудит так важен.

Цель аудита состоит в том, чтобы оценить производительность управления. Из-за распространенного использования систем информационных технологий (ИТ), важно, чтобы средства управления существовали.

Средства управления ИТ — определенные процессы ИТ, разработанные, чтобы поддерживать бизнес-процесс. Средства управления ИТ могут быть категоризированы или как общие средства управления или как средства управления приложением.

Общие средства управления — те средства управления, которые широко распространены на все компоненты систем, процессы и данные для данной организации или системной среды. Они включают средства управления такими областями как центр обработки данных и сетевые операции, системный сбор программного обеспечения и обслуживание, система обеспечения безопасности доступа и сбор прикладной системы, разработка и обслуживание. Средства управления приложением — те средства управления, которые являются подходящими для индивидуальных подсистем учета,

такими как платежная ведомость или кредиторская задолженность. Они относятся к обработке отдельных приложений и помогают гарантировать, что транзакции произошли, авторизованы, и полностью и точно зарегистрированы, обработаны и сообщены. Это означает, что организации должны заняться расследованиями, достигают ли средства управления своих целей, выполняя аудит.

Целью аудита является:

- Обеспечение управления с достаточной гарантией того, что цели управления будут достигнуты;
- Обоснование риска, где есть существенные слабые места управления;
- Консультирование руководства по корректирующим действиям.

Общепринятая структура процесса аудита описана на рисунке 11 [12].

Рисунок 11 — Структура процесса аудита

2.2 Понятие аудита безопасности информационных систем и цели его проведения

Аудит является независимой экспертизой отдельных областей функционирования организации. Различают внешний и внутренний аудит.

Внешний аудит — это разовое мероприятие, проводимое по инициативе руководства организации или акционеров. Проведение внешнего аудита происходит регулярно, но для большинства организаций это в основном рекомендация. Обязательным требование проведения данного аудита является для многих финансовых организаций и акционерных обществ. Внутренний аудит представляет собой непрерывную деятельность, которая осуществляется на основании «Положения о внутреннем аудите» и в соответствии с планом, подготовка которого осуществляется подразделением внутреннего аудита и утверждается руководством организации. Аудит безопасности информационных систем является одной из составляющих ИТ аудита. Целями проведения аудита безопасности являются:

- анализ рисков, связанных с возможностью осуществления угроз безопасности в отношении ресурсов ИС
- оценка текущего уровня защищенности ИС;
- локализация узких мест в системе защиты ИС;
- оценка соответствия ИС существующим стандартам в области информационной безопасности;
- выработка рекомендаций по внедрению новых и повышению эффективности существующих механизмов безопасности ИС.
- Данный набор целей является полным только для внешнего аудита безопасности. Для внутреннего аудита есть ещё несколько дополнительных задач:
- разработка политик безопасности и других организационно-распорядительных документом по защите информации и участие в их внедрении в работу организации;
- постановка задач для ИТ персонала, касающихся обеспечения защиты информации;
- участие в обучении пользователей и обслуживающего персонала ИС вопросам обеспечения информационной безопасности;
- участие в разборе инцидентов, связанных с нарушением информационной безопасности;

Стоит отметить, что вышеописанные задачи, которые являются дополнительными, по сути, не являются аудитом, за исключением участия в обучении. Аудитор по определению должен осуществлять независимую экспертизу реализации механизмов безопасности в организации, что является одним из основных принципов аудиторской деятельности. Если аудитор принимает деятельное участие в реализации механизмов безопасности, то независимость аудитора утрачивается, а вместе с ней утрачивается и объективность его суждений, т. к. аудитор не может осуществлять независимый и объективных контроль своей собственной деятельности [19][50].

2.3 Виды аудита информационной безопасности

Основные виды аудита ИБ представлены на диаграмме (рисунок 12).

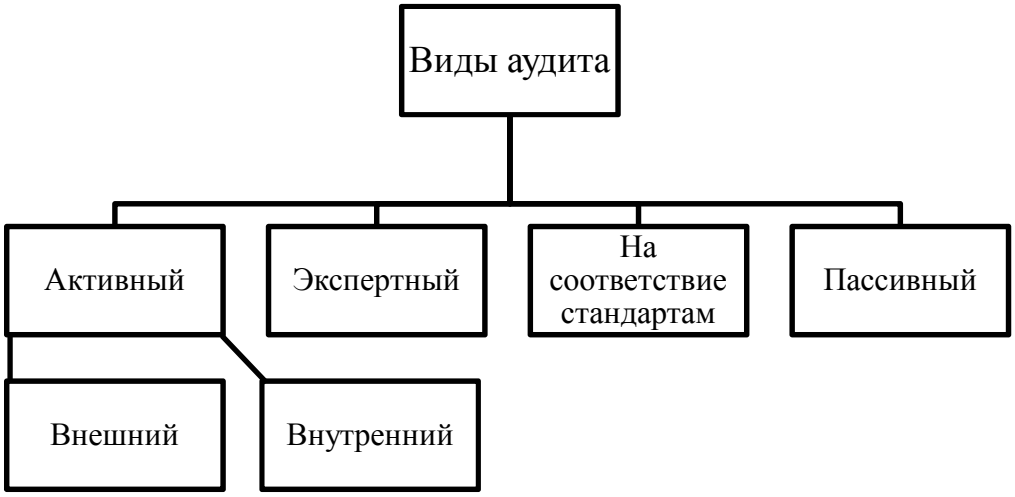

Рисунок 12 — Виды аудита

2.3.1 Активный аудит

Активный аудит — один из самых распространенных видов аудита

Данный вид аудита представляет исследование состояния защищенности информационной системы с точки зрения некоего злоумышленника, обладающего высокой квалификацией в области информационных технологий.

Компании, который проводят активный аудит, именуют его инструментальным анализом защищенности, чтобы отделить данный вид аудита от других.

Суть активного аудита состоит в том, что с помощью специального программного обеспечения (в том числе систем анализа защищенности) и специальных методов осуществляется сбор информации о состоянии системы сетевой защиты. Под состоянием

системы сетевой защиты понимаются лишь те параметры и настройки, использование которых помогает злоумышленнику проникнуть в сети и нанести урон компании.

При осуществлении данного вида аудита на систему сетевой защиты моделируется как можно большее количество сетевых атак, которые может выполнить злоумышленник. При этом аудитор искусственно ставится именно в те условия, в которых работает злоумышленник, — ему предоставляется минимум информации, только та, которую можно найти в открытых источниках.

Атаки только моделируются и не оказывают никакого разрушительного воздействия на информационную систему. Количество и разновидность этих атак зависит от используемых систем анализа защищенности и квалификации аудитора.

Результатом активного аудита являются информация обо всех уязвимостях, степени их критичности и методах устранения, а также сведения о широкодоступной информации сети заказчика, которая доступна любому потенциальному нарушителю.

После проведенного активного аудита выдаются рекомендации по модернизации системы сетевой защиты, которые позволяют устранить опасные уязвимости и тем самым повысить уровень защищенности информационной системы от действий «внешнего» злоумышленника при минимальных затратах на информационную безопасность.

Однако без проведения других видов аудита эти рекомендации могут оказаться недостаточными для создания полной системы сетевой защиты. По результатам данного вида аудита невозможно сделать вывод о корректности проекта информационной системы, с точки зрения безопасности.

Активный аудит — исследование, которое может и должно производиться периодически. Выполнение активного аудита, например раз в год, позволяет удостовериться, что уровень системы сетевой безопасности остается на прежнем уровне.

Активный аудит условно можно разделить на два вида — «внешний» и «внутренний».

При «внешнем» активном аудите специалисты моделируют действия «внешнего» злоумышленника

Процедуры, производимые при «внешнем» активном аудите представлены на рисунке 13.

Рисунок 13 — Процедуры внешнего аудита

«Внутренний» активный аудит по составу работ аналогичен «внешнему», однако при его проведении с помощью специальных программных средств моделируются действия «внутреннего» злоумышленника.

Данное деление активного аудита на «внешний» и «внутренний» актуально для заказчика в следующих случаях:

- у заказчика существуют финансовые ограничения в приобретении услуг и продуктов по защите информации;
- модель злоумышленника, которую рассматривает заказчик, не включает «внутренних» злоумышленников;
- в компании заказчика расследуется факт обхода системы сетевой защиты.

Также в ходе активного аудита производится стресс-тестирование — исследование производительности и стабильности системы. Оно направлено на определение критических точек нагрузки, при которой система вследствие атаки на отказ в обслуживании или повышенной загруженности перестает реагировать на запросы пользователей.

Стресс-тест позволит выявить слабые места в процессе формирования и передачи информации и определить те условия, при которых нормальная работа системы

невозможна. Тестирование включает в себя моделирование атак на отказ в обслуживании, пользовательских запросов к системе и общий анализ производительности.

Тест на проникновение является одной из самых эффективных составляющих активного аудита. Данный тест похож на «внешний» активный аудит, но, по сути, аудитом не является.

Основная цель данного тестирования — демонстрация результатов, которых может достигнуть злоумышленник, действующий при текущем состоянии системы сетевой защиты. Результаты данного теста более наглядны, чем результаты аудита. Однако ему свойственно множество ограничений и особенностей. Такая особенность технического характера, как: заказчик информируется только о факте уязвимости системы сетевой защиты, в то время как в результатах «внешнего» активного аудита заказчику сообщается не только факт уязвимости сети, но и сведения обо всех уязвимостях и способах их устранения [18].

2.3.2 Экспертный аудит

Экспертный аудит является сравнением идеального описания с состоянием информационной безопасности, которое базируется на следующем:

- требования, которые были предъявлены руководством в процессе проведения аудита;
- описание идеальной системы безопасности, основанное на мировом и частном опыте.

При выполнении экспертного аудита сотрудники компании-аудитора совместно с представителями заказчика проводят следующие виды работ:

- сбор исходных данных об информационной системе, об ее функциях и особенностях, используемых технологиях автоматизированной обработки и передачи данных (с учетом ближайших перспектив развития);
- сбор информации об имеющихся организационно-распорядительных документах по обеспечению информационной безопасности и их анализ;
- определение точек ответственности систем, устройств и серверов ИС;
- формирование перечня подсистем каждого подразделения компании с категорированием критичной информации и схемами информационных потоков.

Ключевой этап экспертного аудита — анализ проекта информационной системы, топологии сети и технологии обработки информации. В процессе данного анализа

выявляются такие недостатки существующей топологии сети, которые снижают уровень защищенности информационной системы.

По результатам данного этапа предлагаются изменения, если они требуются, в существующей информационной системе, а также технологии обработки информации, направленные на устранение найденных недостатков с целью достижения требуемого уровня информационной безопасности.

Следующий этап — анализ информационных потоков организации. На данном этапе определяются типы информационных потоков ИС организации, и составляется их диаграмма, где для каждого информационного потока указывается его ценность, в том числе ценность передаваемой информации, и используемые методы обеспечения безопасности, которые показывают уровень защищенности информационного потока.

На основании результатов данного этапа предлагается защита или повышение уровня защищенности тех компонентов информационной системы, которые участвуют в наиболее важных процессах передачи, хранения и обработки информации. Для менее ценной информации уровень защищенности остается прежним, что позволяет сохранить для конечного пользователя простоту работы с информационной системой.

Во время экспертного аудита проводится анализ таких документов организации как: политика безопасности, план защиты и различного рода инструкции.

Данные документы также оцениваются на предмет достаточности и непротиворечивости целям и мерам ИБ. Особое внимание на этапе анализа информационных потоков уделяется определению полномочий и ответственности конкретных лиц за обеспечение информационной безопасности различных участков/подсистем ИС. Полномочия и ответственность должны быть закреплены положениями вышеописанных документов.

Результаты экспертного аудита могут содержать предложения по построению или модернизации системы обеспечения информационной безопасности, например:

- изменения (если они требуются) в существующей топологии сети и технологии обработки информации;
- рекомендации по выбору и применению систем защиты информации и других дополнительных специальных технических средств;
- предложения по совершенствованию пакета организационно-распорядительных документов;
- рекомендации по этапам создания системы информационной безопасности;
- ориентировочные затраты на создание или совершенствование системы обеспечения информационной безопасности (СОИБ) [18].

2.3.3 Аудит на соответствие стандартам

Данный вид аудита является сравнением некоторого абстрактного описания, приводимым в стандартах с состоянием информационной безопасности.

Официальный отчет, подготовленный в результате проведения данного вида аудита, включает следующую информацию:

- степень соответствия проверяемой информационной системы выбранным стандартам;
- степень соответствия собственным внутренним требованиям компании в области информационной безопасности;
- количество и категории полученных несоответствий и замечаний;
- рекомендации по построению или модификации системы обеспечения информационной безопасности, позволяющие привести ее в соответствие с рассматриваемым стандартом;
- подробная ссылка на основные документы заказчика, включая политику безопасности, описания процедур обеспечения информационной безопасности, дополнительные обязательные и необязательные стандарты и нормы, применяемые к данной компании.

Ниже перечислены примеры стандартов, на соответствие которым проводится аудит системы информационной безопасности.

Существующие руководящие документы Гостехкомиссии:

- «Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации» (далее РД для АС);
- «Специальные требования и рекомендации по технической защите конфиденциальной информации» (СТР-К);
- «Безопасность информационных технологий. Критерии оценки безопасности информационных технологий» (ГОСТ Р ИСО/МЭК 15408-2002 или «Общие критерии»);

Зарубежные и международные стандарты:

- Международный стандарт ISO/IEC 17799 «Информационные технологии. Управление информационной безопасностью» (Information Technology — Information Security Management). На сегодняшний день является одним из самых распространенных и широко применяемым стандартом во всем мире;
- Международный стандарт WebTrust. Применим для подтверждения высокого уровня защищенности системы электронной коммерции и web-сервисов.

Причины проведения аудита на соответствие стандарту и сертификации можно условно разделить по степени того насколько это необходимо организации:

- обязательная сертификация;
- сертификация, вызванная «внешними» объективными причинами;
- сертификация, позволяющая получить выгоды в долгосрочной перспективе;
- добровольная сертификация [18].

2.3.4 Пассивный аудит

Также стоит рассмотреть технологии так называемого пассивного аудита, которые собирают информацию об уязвимостях системы другими способами, которые в отличие от активного аудита не предполагают большой нагрузки и выполнения сложных запросов.

Одним из методов подобного пассивного сканирования является проверка конфигурационных файлов сетевых устройств и средств защиты. В такой сканер загружаются конфигурационные файлы устройств, а программный комплекс проверяет соответствие их настроек принятой на предприятии политики безопасности.

Подсистема пассивного аудита безопасности выполняет следующие основные функции, которые представлены на рисунке 14.

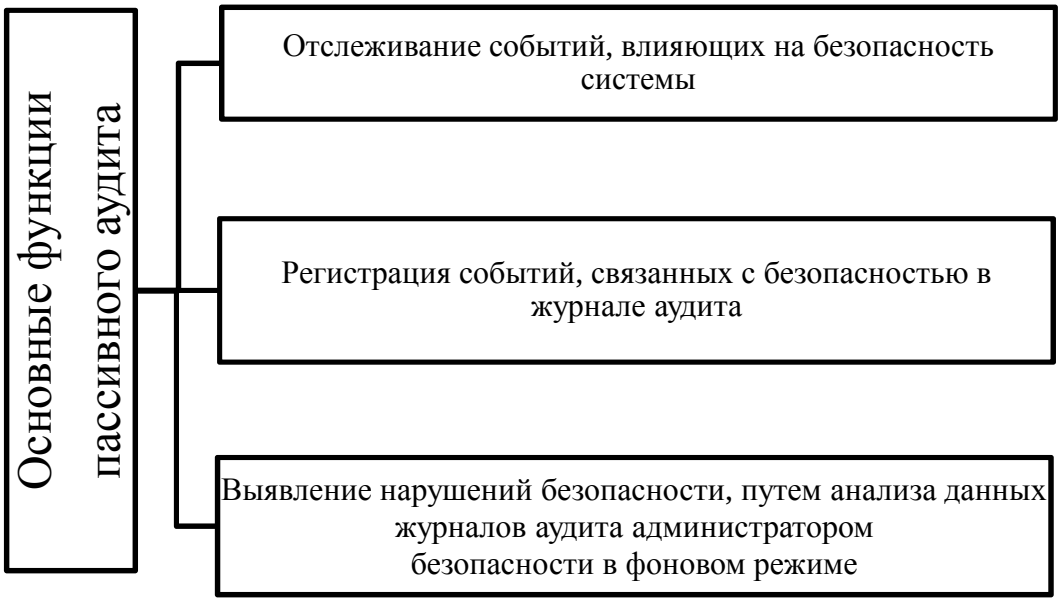

Рисунок 14 — Основные функции пассивного аудита

Эффективность функционирования системы пассивного аудита безопасности определяется следующими основными свойствами этой системы:

- наличие средств аудита, обеспечивающих возможность выборочного контроля любых происходящих в системе событий, связанных с безопасностью;
- наличие средств централизованного управления журналами аудита, политикой аудита и централизованного анализа данных аудита по всем контролируемым системам;
- непрерывность контроля над критичными компонентами ЛВС во времени.

В заключение можно отметить, что при планировании проверки состояния системы информационной безопасности важно не только точно выбрать вид аудита, исходя из потребностей и возможностей компании, но и не ошибиться с выбором исполнителя.

Как неоднократно отмечалось, результаты любого вида аудита содержат рекомендации по модернизации системы обеспечения информационной безопасности.

Если аудит проводит консалтинговая компания, которая кроме консалтинговой деятельности занимается еще и разработкой собственных систем защиты информации, то данная компания заинтересована в том, чтобы результаты аудита рекомендовали заказчику использовать ее продукты. Для того чтобы рекомендации на основе аудита были действительно объективными, необходимо, чтобы компания-аудитор была независима в выборе используемых систем защиты информации и имела большой опыт работы в области информационной безопасности [20].

2.4 Этапность работ по проведению аудита безопасности информационных систем

Работы по аудиту безопасности ИС включают в себя ряд последовательных этапов, которые в целом соответствуют этапам проведения комплексного ИТ аудита АС, который включает в себя следующее этапы представленные на рисунке 15.

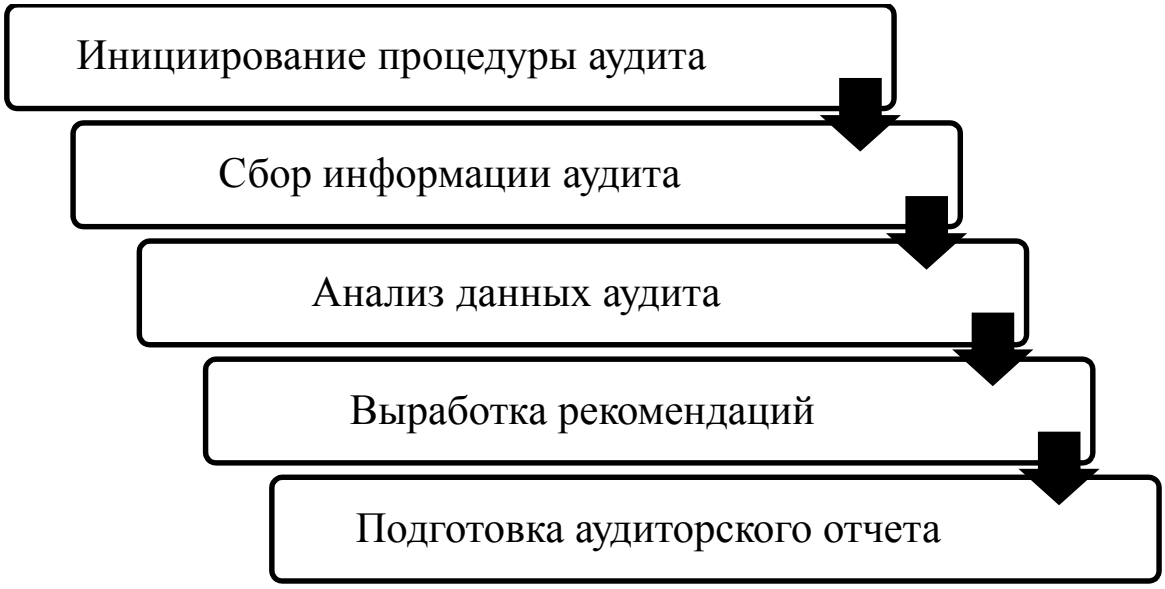

Рисунок 15 — Этапы аудита

2.4.1 Инициирование процедуры аудита

Аудит проводится не по инициативе аудитора, а по инициативе руководства компании, которое в данном вопросе является основной заинтересованной стороной. Поддержка руководства компании является необходимым условием для проведения аудита.

Аудит представляет собой комплекс мероприятий, в которых помимо самого аудитора, оказываются задействованными представители большинства структурных подразделений компании. Действия всех участников этого процесса должны быть скоординированы. Поэтому на этапе инициирования процедуры аудита должны быть решены следующие организационные вопросы:

- права и обязанности аудитора должны быть четко определены и документально закреплены в его должностных инструкциях, а также в положении о внутреннем (внешнем) аудите;
- аудитором должен быть подготовлен и согласован с руководством план проведения аудита;

- в положении о внутреннем аудите должно быть закреплено, в частности, что сотрудники компании обязаны оказывать содействие аудитору и предоставлять всю необходимую для проведения аудита информацию.

На этапе инициирования процедуры аудита должны быть определены границы проведения обследования. Одни информационные подсистемы компании не являются достаточно критичными и их можно исключить из границ проведения обследования. Другие подсистемы могут оказаться недоступными для аудита из-за соображений конфиденциальности.

Границы проведения обследования определяются в следующих терминах:

- Список обследуемых физических, программных и информационных ресурсов;
- Площадки (помещения), попадающие в границы обследования;
- Основные виды угроз безопасности, рассматриваемые при проведении аудита;
- Организационные (законодательные, административные и процедурные), физические, программно-технические и прочие аспекты обеспечения безопасности, которые необходимо учесть в ходе проведения обследования, и их приоритеты (в каком объеме они должны быть учтены).

2.4.2 Сбор информации аудита

Этап сбора информации аудита, является наиболее сложным и длительным. Это связано с отсутствием необходимой документации на информационную систему и с необходимостью плотного взаимодействия аудитора со многими должностными лицами организации.

Компетентные выводы относительно положения дел в компании с информационной безопасностью могут быть сделаны аудитором только при условии наличия всех необходимых исходных данных для анализа. Получение информации об организации, функционировании и текущем состоянии ИС осуществляется аудитором в ходе специально организованных интервью с ответственными лицами компании, путем изучения технической и организационно-распорядительной документации, а также исследования ИС с использованием специализированного программного инструментария. Рассмотрим, какая информация необходима аудитору для анализа.

Обеспечение информационной безопасности организации — это комплексный процесс, требующий четкой организации и дисциплины. Он должен начинаться с определения ролей и распределения ответственности среди должностных лиц, занимающихся информационной безопасностью. Поэтому первый пункт аудиторского обследования начинается с получения информации об организационной структуре

пользователей ИС и обслуживающих подразделений. В связи с этим аудитору требуется следующая документация:

- Схема организационной структуры пользователей;
- Схема организационной структуры обслуживающих подразделений.
- Обычно, в ходе интервью аудитор задает опрашиваемым следующие вопросы:
- Кто является владельцем информации?
- Кто является пользователем (потребителем) информации?
- Кто является провайдером услуг?

Назначение и принципы функционирования ИС во многом определяют существующие риски и требования безопасности, предъявляемые к системе. Поэтому на следующем этапе аудитора интересует информация о назначении и функционировании ИС. Аудитор задает опрашиваемым примерно следующие вопросы:

- Какие услуги и каким образом предоставляются конечным пользователям?
- Какие основные виды приложений, функционирует в ИС?
- Количество и виды пользователей, использующих эти приложения?

Ему понадобиться также следующая документация, конечно, если таковая вообще имеется в наличие (что, вообще говоря, случается нечасто):

- Функциональные схемы;
- Описание автоматизированных функций;
- Описание основных технических решений;
- Другая проектная и рабочая документация на информационную систему.

Далее, аудитору требуется более детальная информация о структуре ИС. Это позволит уяснить, каким образом осуществляется распределение механизмов безопасности по структурным элементам и уровням функционирования ИС. Типовые вопросы, которые обсуждаются в связи с этим во время интервью, включают в себя:

- Из каких компонентов (подсистем) состоит ИС?
- Функциональность отдельных компонент?
- Где проходят границы системы?
- Какие точки входа имеются?
- Как ИС взаимодействует с другими системами?
- Какие каналы связи используются для взаимодействия с другими ИС?
- Какие каналы связи используются для взаимодействия между компонентами системы?
- По каким протоколам осуществляется взаимодействие?

- Какие программно-технические платформы используются при построении системы?

На этом этапе аудитору необходимо запастись следующей документацией:

- Структурная схема ИС;
- Схема информационных потоков;
- Описание структуры комплекса технических средств информационной системы;
- Описание структуры программного обеспечения;
- Описание структуры информационного обеспечения;
- Размещение компонентов информационной системы.

Подготовка значительной части документации на ИС, обычно, осуществляется уже в процессе проведения аудита. Когда все необходимые данные по ИС, включая документацию, подготовлены, можно переходить к их анализу.

2.4.3 Анализ данных аудита

Используемые аудиторами методы анализа данных определяются выбранными подходами к проведению аудита, которые могут существенно различаться.

Первый подход, самый сложный, базируется на анализе рисков. Опираясь на методы анализа рисков, аудитор определяет для обследуемой ИС индивидуальный набор требований безопасности, в наибольшей степени учитывающий особенности данной ИС, среды ее функционирования и существующие в данной среде угрозы безопасности. Данный подход является наиболее трудоемким и требует наивысшей квалификации аудитора. На качество результатов аудита, в этом случае, сильно влияет используемая методология анализа и управления рисками и ее применимость к данному типу ИС.

Второй подход, самый практичный, опирается на использование стандартов информационной безопасности. Стандарты определяют базовый набор требований безопасности для широкого класса ИС, который формируется в результате обобщения мировой практики. Стандарты могут определять разные наборы требований безопасности, в зависимости от уровня защищенности ИС, который требуется обеспечить, ее принадлежности, а также назначения. От аудитора в данном случае требуется правильно определить набор требований стандарта, соответствие которым требуется обеспечить для данной ИС. Необходима также методика, позволяющая оценить это соответствие. Из-за своей простоты и надежности, описанный подход наиболее распространен на практике. Он позволяет при минимальных затратах ресурсов делать обоснованные выводы о состоянии ИС.

Третий подход, наиболее эффективный, предполагает комбинирование первых двух. Базовый набор требований безопасности, предъявляемых к ИС, определяется стандартом. Дополнительные требования, в максимальной степени учитывающие особенности функционирования данной ИС, формируются на основе анализа рисков. Этот подход является намного проще первого, т.к. большая часть требований безопасности уже определена стандартом, и, в то же время, он лишен недостатка второго подхода, заключающего в том, что требования стандарта могут не учитывать специфики обследуемой ИС.

Использование методов анализа рисков

Если для проведения аудита безопасности выбран подход, базирующийся на анализе рисков, то на этапе анализа данных аудита обычно выполняются следующие группы задач, представленные на рисунке 16. Перечисленный набор задач, является достаточно общим. Для их решения могут использоваться различные формальные и неформальные, количественные и качественные, ручные и автоматизированные методики анализа рисков. Суть подхода от этого не меняется

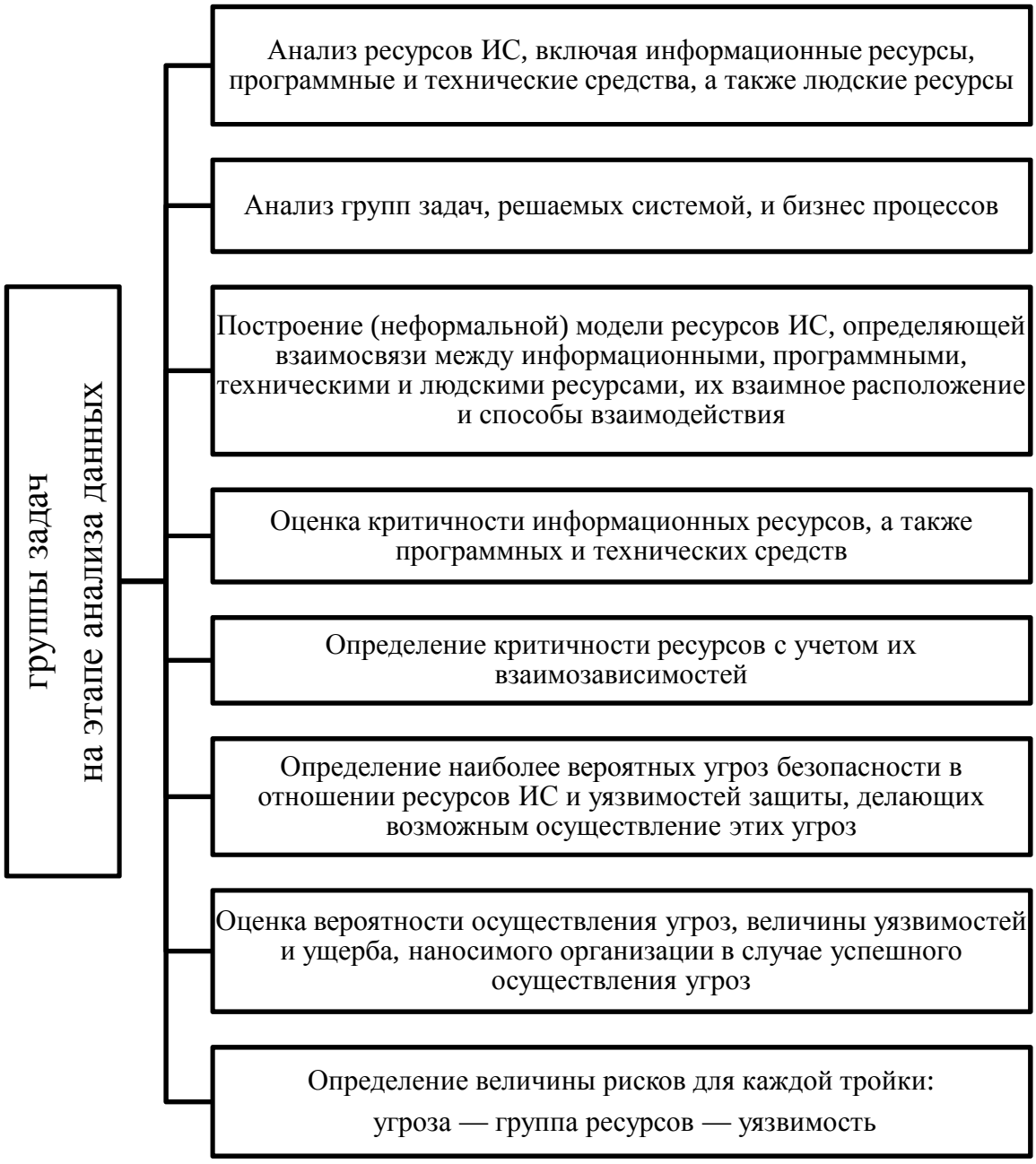

Рисунок 16 — Задачи на этапе анализа данных

Оценка рисков может даваться с использованием различных как качественных, так и количественных шкал. Необходимо, чтобы существующие риски были правильно идентифицированы в соответствии со степенью их критичности для организации. На основе такого анализа может быть разработана система первоочередных мероприятий по уменьшению величины рисков до приемлемого уровня.

Оценка соответствия требованиям стандарта

В случае проведения аудита безопасности на соответствие требованиям стандарта, аудитор, полагаясь на свой опыт, оценивает применимость требований стандарта к обследуемой ИС и ее соответствие этим требованиям. Данные о соответствии различных областей функционирования ИС требованиям стандарта, обычно, представляются в табличной форме. Из таблицы видно, какие требования безопасности в системе не реализованы. Исходя из этого, делаются выводы о соответствии обследуемой ИС требованиям стандарта и даются рекомендации по реализации в системе механизмов безопасности, позволяющих обеспечить такое соответствие.

2.4.4 Выработка рекомендаций

Рекомендации по результатам анализа состояния ИС, определяются используемым подходом, особенностями обследуемой ИС, состоянием дел с информационной безопасностью и степенью детализации, используемой при проведении аудита.

Рекомендации должны быть конкретными и применимыми к данной ИС, экономически обоснованными, аргументированными (подкрепленными результатами анализа) и отсортированными по степени важности. При этом мероприятия по обеспечению защиты организационного уровня практически всегда имеют приоритет над конкретными программно-техническими методами защиты.

В результате аудита не предоставляется план технического проекта подсистемы информационной безопасности, а также детальные рекомендации по внедрению конкретных программно-технических средств защиты информации. Для этого требуется более детальной проработки конкретных вопросов организации защиты.

2.4.5 Подготовка аудиторского отчета

Аудиторский отчет — основной результат проведения аудита. Его качество характеризует качество работы аудитора. Структура отчета может существенно различаться в зависимости от характера и целей проводимого аудита. Однако определенные разделы должны обязательно присутствовать в аудиторском отчете. Отчет должен содержать:

- описание целей проведения аудита
- характеристику обследуемой ИС
- указание границ проведения аудита
- указание используемых методов
- результаты анализа данных аудита

- выводы, обобщающие эти результаты и содержащие оценку уровня защищенности АС или соответствие ее требованиям стандартов
- рекомендации аудитора по устранению существующих недостатков и совершенствованию системы защиты [17-24].

2.5 Информационная безопасность в системах управления базами данных

В современных условиях любая деятельность организаций связана с обработкой больших объёмов информации, которая производится широким кругом лиц. Защита данных от несанкционированного доступа является одной из приоритетных задач при проектировании любой информационной системы. Следствием того, что в современном обществе возросло значение информации стали высоки и требования к конфиденциальности данных. Системы управления базами данных, в особенности реляционные СУБД, стали преимущественным инструментом в этой области. Обеспечение информационной безопасности СУБД приобретает решающее значение при выборе конкретного средства, которое обеспечит необходимый уровень безопасности организации в целом.

В современных СУБД поддерживается один из двух наиболее общих подходов к вопросу обеспечения безопасности данных: избирательный подход и обязательный подход. В обоих подходах единицей данных или "объектом данных", для которых должна быть создана система безопасности, может быть как вся база данных целиком, так и любой объект внутри базы данных.

Эти два подхода отличаются следующими свойствами:

- В случае избирательного управления некоторый пользователь обладает различными правами (привилегиями или полномочиями) при работе с данными объектами. Разные пользователи могут обладать разными правами доступа к одному и тому же объекту. Избирательные права характеризуются значительной гибкостью.
- В случае неизбирательного управления, наоборот, каждому объекту данных присваивается некоторый классификационный уровень, а каждый пользователь обладает некоторым уровнем допуска. При таком подходе доступом к определенному объекту данных обладают только пользователи с соответствующим уровнем допуска.

Для реализации избирательного принципа предусмотрены следующие методы. В базу данных вводится новый тип объектов БД — это пользователи. Каждому

пользователю в БД присваивается уникальный идентификатор. Для дополнительной защиты каждый пользователь кроме уникального идентификатора снабжается уникальным паролем, причем если идентификаторы пользователей в системе доступны системному администратору, то пароли пользователей хранятся чаще всего в специальном кодированном виде и известны только самим пользователям.

Пользователи могут быть объединены в специальные группы пользователей. Один пользователь может входить в несколько групп. В стандарте вводится понятие группы PUBLIC, для которой должен быть определен минимальный стандартный набор прав. По умолчанию предполагается, что каждый вновь создаваемый пользователь, если специально не указано иное, относится к группе PUBLIC.

Привилегии или полномочия пользователей или групп — это набор действий (операций), которые они могут выполнять над объектами БД.

В последних версиях ряда коммерческих СУБД появилось понятие "роли". Роль — это поименованный набор полномочий. Существует ряд стандартных ролей, которые определены в момент установки сервера баз данных. И имеется возможность создавать новые роли, группируя в них произвольные полномочия. Введение ролей позволяет упростить управление привилегиями пользователей, структурировать этот процесс. Кроме того, введение ролей не связано с конкретными пользователями, поэтому роли могут быть определены и сконфигурированы до того, как определены пользователи системы.

Пользователю может быть назначена одна или несколько ролей.

Объектами БД, которые подлежат защите, являются все объекты, хранимые в БД: таблицы, представления, хранимые процедуры и триггеры. Для каждого типа объектов есть свои действия, поэтому для каждого типа объектов могут быть определены разные права доступа.

На самом элементарном уровне концепции обеспечения безопасности баз данных исключительно просты. Необходимо поддерживать два фундаментальных принципа: проверку полномочий и проверку подлинности (аутентификацию).

Проверка полномочий основана на том, что каждому пользователю или процессу информационной системы соответствует набор действий, которые он может выполнять по отношению к определенным объектам. Проверка подлинности означает достоверное подтверждение того, что пользователь или процесс, пытающийся выполнить санкционированное действие, действительно тот, за кого он себя выдает.

Система назначения полномочий имеет в некотором роде иерархический характер. Самыми высокими правами и полномочиями обладает системный администратор или

администратор сервера БД. Традиционно только этот тип пользователей может создавать других пользователей и наделять их определенными полномочиями.

СУБД в своих системных каталогах хранит как описание самих пользователей, так и описание их привилегий по отношению ко всем объектам.

Далее схема предоставления полномочий строится по следующему принципу. Каждый объект в БД имеет владельца — пользователя, который создал данный объект. Владелец объекта обладает всеми правами-полномочиями на данный объект, в том числе он имеет право предоставлять другим пользователям полномочия по работе с данным объектом или забирать у пользователей ранее предоставленные полномочия.

В ряде СУБД вводится следующий уровень иерархии пользователей — это администратор БД. В этих СУБД один сервер может управлять множеством СУБД (например, MS SQL Server, Sybase). В СУБД Oracle применяется однобазовая архитектура, поэтому там вводится понятие подсхемы — части общей схемы БД и вводится пользователь, имеющий доступ к подсхеме.

В стандарте SQL не определена команда создания пользователя, но практически во всех коммерческих СУБД создать пользователя можно не только в интерактивном режиме, но и программно с использованием специальных хранимых процедур. Однако для выполнения этой операции пользователь должен иметь право на запуск соответствующей системной процедуры.

В стандарте SQL определены два оператора: GRANT и REVOKE соответственно предоставления и отмены привилегий.

Различают три вида привилегий:

- Объектные (Object privileges) — это разрешения на объекты схемы, такие как таблицы, представления, последовательности, пакеты. Для использования объектов схемы принадлежащих другому пользователю, необходимы привилегии на этот объект.
- Системные (System privileges) — это разрешения на операции уровня базы данных, например подключение к базе данных, создание пользователей, внесение изменений в конфигурацию базы данных.
- Ролевые (Role privileges) — это объектные и системные привилегии, которые пользователь получает как роль.

Роли — это возможность для администрирования групп или привилегий.

Ролью называется именованный набор привилегий. Объединение привилегий в роли значительно упрощает процесс назначения и снятия привилегий. Если СУБД

поддерживает управление ролями, то в SQL-операторах GRANT и REVOKE вместо имени пользователя можно указывать имя роли.

Для управления привилегиями определены следующие правила:

- объект принадлежит пользователю, его создавшему (если синтаксисом не указано создание объекта другого пользователя, конечно, при соответствующих полномочиях);
- владелец объекта, согласно стандарту, может изменять привилегии своего;
- объектная привилегия всегда соотносится с конкретным объектом, а системная - с объектами вообще [21][12][23].

2.5.1 Аудит систем управления базами данных

Также одним из направлений, позволяющих обеспечить информационную безопасность, является аудит.

В двух словах, аудит — это комплекс действий, которые позволяют определить какой пользователь и какие действия в системе выполнял и имел ли он права на эти действия

Обеспечение безопасности имеет целью, прежде всего, помешать выполнять пользователям недопустимые операции. Под термином "аудит", в рамках баз данных, часто подразумевают ведение журнала аудита. Журнал аудита — это специальный набор записей, создаваемых системой, который надежно защищён от несанкционированного доступа. Иногда эти понятия используют как синонимы [1].

Основное назначение журнала аудита — выступать в роли средства, позволяющего обнаружить действия, которые могут нарушить целостность системы. С его помощью можно выявить как мошеннически введенные данные, так и несанкционированные запросы [1].

Теоретически кандидатом на регистрацию является любое происходящее в системе событие. Особенно важны для аудита вход в систему и выход из системы, а также обновление данных. Аудит запросов несколько затруднен, если запросы производятся не через хранимые процедуры, поскольку средства аудита можно встроить в процедуры [1].

Существует возможность использовать стандартные средства аудита таких СУБД как: Oracle, MS SQL Server и т.д. Можно привести несколько примеров.

Обозначим несколько задач, которые должен решать аудит.

Аудит доступа к базе данных. Контроль доступа к БД является фундаментальной задачей для того, что бы определить кто, когда и откуда имеет доступ к информации.

Неудачные попытки, так же как и попытки входа в аномальное время в течение дня должны быть отслежены.

Аудит изменений в структуре базы данных. В базе данных никому из пользователей никогда не следует изменять структуру схемы. Администраторам баз данных следует вносить изменения в специально отведенное для этого время. Какие-либо другие изменения следует рассматривать как подозрительные. Наблюдение за структурными изменениями может включить индикаторы некорректного использования базы данных[2].

Третья задача — это аудит использования любых системных привилегий. Заключительная группа команд аудита, которая может быть задействована это организация контроля за любыми изменениями данных, при помощи самих объектов [2].

2.5.2 Аудит различных систем управления базами данных

Существует возможность использовать стандартные средства аудита таких СУБД как: Oracle, MS SQL Server и т.д. Но, как правило, данные средства есть только у платных СУБД. Для бесплатных же СУБД, таких, к примеру, как PostgreSQL, подобных решений пока не нет. Можно привести несколько примеров.

Использование средств аудита Oracle. Некоторые встроенные возможности аудита в Oracle могут быть довольно полезными. Если аудит включен на уровне экземпляров, то команда:

- AUDIT SESSION; регистрирует все подключения к базе данных и отключения от нее, как успешные, так и неудачные.
- AUDIT ALL ON LIVE.Таблица1 BY ACCESS; Эта команда заставляет Oracle регистрировать "детали" всех операций над таблицей Таблица1. Фраза BY ACCESS обеспечивает наличие элемента для каждого случая доступа, а не только одного элемента на тип доступа для данного сеанса. Oracle заносит все аудиторские действия в одну таблицу SYS.AUD\$, и, если включено много журналов аудита, эта таблица становится очень загроможденной. Но Oracle обеспечивает ряд представлений, позволяющих просмотреть содержимое AUD\$ в виде структурированных наборов [11].

Аудиторские представления позволяют запрашивать доступы к таблице Таблица1 по пользователям, по типам доступа, по дате и времени доступа. Однако мы не можем установить, что было изменено в таблице.

Задачу аудита базы данных Oracle не следует ограничивать только лишь использованием команд аудита; так же успешно могут быть применены и другие технологии. Приведем некоторые основные методы, которые могут быть использованы для аудита базы данных Oracle [2].:

- Аудит Oracle. Все привилегии, которые могут быть предоставлены пользователю или роли базы данных могут быть проконтролированы. Сюда включено доступ на чтение, запись и удаление объектов на табличном уровне [2].
- Системные триггеры. Эта возможность была представлена начиная с Oracle 8 и разрешает выполнение операций триггера, когда имеет место системное событие. Сюда включены запуск и останов базы данных, попытки входа и выхода, создание, изменение и удаление объектов схемы. С помощью автономных транзакций, можно записывать в журнал упомянутые системные события [2].
- Update, delete и insert триггеры Для того, что бы отслеживать изменения в базе на уровне столбца и строки, можно написать триггеры, которые позволят полностью сохранять данные, до или после выполненного действия. Использование этого типа контроля очень ресурсоемко, так как создается и хранится много дополнительных записей. Кроме того, что существует еще один недостаток, связанный с этим методом - доступ на чтение нельзя отследить с помощью обычных триггеров базы данных [2].
- Детализированный (Fine-grained) аудит Детализированный аудит решает проблему отслеживания доступа на чтение. Данная возможность основана на внутренних триггерах, срабатывающих, при разборе какой-нибудь части SQL-предложени я. Это очень эффективно, так как SQL-предложени е разбирается единожды для аудита и выполнения. Эта возможность использует предикаты, которые определены и проверяются каждый раз, когда происходит доступ к соответствующим объектам. Этот метод позволяет контролировать не только DML-операции на уровне строк и столбцов, но и предложения чтения [2].
- Системные журналы СУБД Oracle генерирует много журнальных файлов, и многие из них могут содержать полезную информацию для проведения аудита. Например, alert log используется для записи информации о запуске и останове базы, а также о вносимых структурных изменениях, таких как добавление файла данных в базу [2].

Рассмотрим подсистему аудита SQL Server (Database Engine) [3].

Аудит экземпляра среды Компонент SQL Server Database Engine или отдельной базы данных включает в себя отслеживание и протоколирование событий, происходящих в компоненте Компонент Database Engine. Аудит среды SQL Server позволяет проводить аудит сервера, который может включать в себя спецификации аудита сервера для событий на уровне сервера, а также спецификации аудита базы данных для событий на уровне базы данных. События аудита могут записываться в журналы событий или файлы аудита.

В SQL Server доступно несколько уровней аудита, применение которых зависит от существующих требований или стандартов установки. Подсистема аудита SQL Server предоставляет средства и процессы, необходимые для включения, хранения и просмотра аудитов на различных объектах серверов и баз данных.

Группы действий аудита сервера можно записывать для всего экземпляра, а также группы действий аудита базы данных либо действия аудита базы данных для каждой базы данных. Событие аудита будет происходить каждый раз при обнаружении действия, подлежащего аудиту.

Аудит на уровне сервера поддерживается во всех выпусках SQL Server. Аудит на уровне базы банных доступен только в выпусках Enterprise Edition, Developer Edition и Evaluation Edition.

Аудит — это сочетание в едином пакете нескольких элементов для определенной группы действий сервера или базы данных. Компоненты подсистемы аудита SQL Server совместно формируют выходные данные, называемые аудитом, аналогично тому, как определение отчета в сочетании с элементами графики и данных формирует отчет.

Подсистема аудита SQL Server использует расширенные события для создания аудита.

Объект Подсистема аудита SQL Server объединяет отдельные экземпляры действий или групп действий уровня сервера или базы данных, за которыми нужно проводить наблюдение. Аудит работает на уровне экземпляра SQL Server. В одном экземпляре SQL Server может существовать несколько аудитов.

При определении аудита задается место для вывода результатов. Оно называется назначением аудита. Аудит создается в отключенном состоянии и не выполняет автоматический аудит никаких действий. После включения аудита назначение аудита начинает получать от него данные.

Объект Спецификация аудита сервера принадлежит аудиту. На каждый аудит можно создать один объект спецификации аудита сервера, поскольку они оба создаются в области экземпляра SQL Server.

Спецификация аудита сервера собирает множество групп действий уровня сервера, вызываемых компонентом расширенных событий. В спецификацию аудита сервера можно включить группы действий аудита. Группы действий аудита — это стандартные группы действий, являющиеся атомарными событиями, происходящими в компоненте Компонент Database Engine. Эти действия передаются аудиту, который регистрирует их в целевом объекте.

Результаты аудита отправляются цели, которая может быть файлом, журналом событий безопасности Windows или журналом событий приложений Windows. Журналы необходимо периодически просматривать и архивировать, чтобы у цели оставалось достаточно места для создания дополнительных записей. Для записи в журнал событий безопасности Windows необходимо добавить в политику Создание аудитов безопасности учетную запись службы SQL Server.

Если данные аудита сохраняются в файл, то для предотвращения подмены можно ограничить доступ к файлу следующим образом:

- Учетная запись службы SQL Server должна обладать разрешением на чтение и запись.
- Администраторам аудита обычно требуется разрешение на чтение и запись. Здесь подразумевается, что администраторы аудита — это учетные записи Windows, предназначенные для администрирования файлов аудита, в том числе копирования их в другие общие папки, резервного копирования и других операций.
- Агенты чтения аудита должны иметь разрешение только для чтения файлов аудита.

Даже если запись в файл выполняется компонентом Database Engine, другие пользователи Windows могут прочитать файл аудита, если имеют нужное разрешение. Компонент Компонент Database Engine не получает монопольную блокировку, запрещающую операции чтения.

Поскольку компонент Database Engine может получать доступ к файлу, то имена входа SQL Server, имеющие разрешение CONTROL SERVER, могут использовать компонент Database Engine для доступа к файлам аудита. Чтобы зарегистрировать пользователей, читающих файлы аудита, надо определить аудит в представлении master.sys.fn_get_audit_file. В результате будут записаны имена входа с разрешением CONTROL SERVER, которые получали доступ к файлу аудита через SQL Server.

Если администратор аудита скопирует файл в другое место (в целях архивирования или по другой причине), то список управления доступом к новому месту следует сократить до следующего набора разрешений:

- администратор аудита — чтение и запись;
- агент чтения аудита — только чтение.

Можно обеспечить дополнительную защиту от несанкционированного доступа путем шифрования папки, в которой хранится файл аудита, с применением шифрования диска Windows BitLocker или шифрованной файловой системы Windows (EFS).

Для определения аудита можно использовать среду SQL Server Management Studio или Transact-SQL.

Просматривать журналы событий Windows можно с помощью программы Средство просмотра событий в Windows. Для чтения целевых файлов можно использовать Средство просмотра журнала, среду SQL Server Management Studio или функцию fn_read_audit_file.

Обычно процесс создания и использования аудита происходит следующим образом.

- Создайте аудит и определите цель.
- Создается либо спецификация аудита сервера, либо спецификация аудита базы данных, которая сопоставляет аудит. Включается спецификация аудита.
- Включите аудит.
- Считывание событий аудита можно осуществить с помощью оснастки Просмотр событий Windows, Средства просмотра журнала или функции fn_get_audit_file.

Создание аудита на Transact-SQL заключается в реализации всех аспектов аудита среды SQL Server, для этого можно использовать инструкции DDL, представления каталогов и динамические административные представления и функции; чтобы создать, изменить или удалить спецификацию аудита, можно использовать инструкции DDL.

Далее представим несколько варианта решения задачи аудита для SQL Server 2008 [4][5].

CT (Change Tracking)

Зачастую путают с CDC (Change Data Capture). Но эти инструменты различны как в назначении, так и в реализации. CT предназначен для отслеживания фактов изменений (в каких строках, какие данные были изменены (CRUD)), в то время как CDC хранит

историю изменений (все версии строк, в том числе те, которые были удалены). Что касается реализации, CDC основан на чтении журнала транзакций (асинхронен), в то время как CT работает синхронно.

Для каждой таблицы, для которой включено отслеживание изменений, создается системная таблица, в которой хранился ID измененной строки, битовая маска для идентификации измененных колонок, тип операции.

Для включения CT нужно активировать его на уровне БД и для конкретной таблицы:

```
ALTER DATABASE ChangeTracking SET change_tracking = ON
(change_retention = 10 minutes, auto_cleanup = ON)
ALTER TABLE Orders enable change_tracking WITH (track_columns_updated = ON)
```

CDC (Change Data Capture)

Средство для отслеживания измененных данных. Основными отличиями от CT являются асинхронная реализация (как писалось выше) и хранение всех версий измененных (CRUD) данных. Для хранения измененных данных CDC использует системные таблицы в схеме cdc. Для каждой таблицы, для которой активирован CDC, создается таблица.

Для активации CDC Вам нужно активировать его на уровне БД для конкретной таблицы.

С чисто практической точки зрения, значительный минус CDC это то, что невозможно зафиксировать автора изменений.

SQL Server Audit

Мощное средство, предназначенное для отслеживания всех событий и запросов и серверу (в том числе select). Область применения этого средства достаточно широка — от профилирования до вопросов, связанных с безопасностью и выявление активности пользователей в не предназначенной им части БД.

SQL Server Audit позволяет гибко настраивать фильтры отслеживаемых событий.

Для использования аудита необходимо активировать его на уровне сервера:

```
CREATE server audit ServerAudit
TO FILE (filepath = `D:`, maxsize = 1GB)
WITH (on_failture = CONTINUE)
ALTER server audit ServerAudit WITH (STATE=ON)
```

Пример создания спецификации аудита (трейса) на уровне сервера:

```
CREATE server audit specification ServerAudit_Permissions
FOR server audit ServerAudit
ADD (server_principal_change_group),
ADD (server_permission_change_group),
ADD (server_role_member_change_group);
ALTER server audit specification ServerAudit_Permissions
WITH (STATE=ON);
```

Пример создания спецификации аудита на уровне БД:

```
USE MyDb
CREATE DATABASE audit specification SA_MyDb_Orders
FOR server audit ServerAudit
ADD (SELECT, UPDATE, INSERT, DELETE ON dbo.Orders BY PUBLIC),
ADD (SELECT, UPDATE, INSERT, DELETE ON dbo.OrderDetails BY PUBLIC)
```

2.5.3 Выводы

В данной главе были рассмотрены две основных темы дипломного проекта, такие как аудит информационной безопасности и аудит информационной безопасности систем управления базами данных.

В первой теме подробно было разобраны цели, а также структура аудита информационной безопасности информационных систем. Далее в этой теме были рассмотрены основные виды аудита информационной безопасности, каждый из которых подробно разобран. После чего были представлены важнейшие этапы работ по проведению аудита в организации, каждый из этапов проанализирован.

Во второй теме рассмотрены подходы для обеспечения информационной безопасности СУБД и как они реализованы. В дальнейшем было дано подробное определение аудита СУБД и приведены несколько примером того, как аудит представлен в различных СУБД и какие инструменты в них реализованы на данный момент.

3. Разработка подсистемы аудита системы управления базой данных

3.1 Выбор системы управления базой данных

Основной задачей является разработка решения аудита информационной системы, использующей СУБД PostgreSQL. Данную задачу можно решить путём создания подсистемы аудита, которая позволит обнаружить действия, нарушающие целостность основной систем, другими словами, с помощью данной подсистемы можно выявить как мошеннически введенные данные, так и несанкционированные запросы. Разработанная подсистема осуществляет сбор, хранение и отображение данных, необходимых для аудита, но не делает обработку и анализ этих данных.

При планировании проекта была выбрана СУБД PostgreSQL.

В Таблице 1 приведен сравнительный анализ трех распространенных систем управления базами данных, конкурирующих на рынке программного обеспечения по основным показателям.

Таблица 1 — Сравнительная таблица СУБД [29]

Показатели	Microsoft SQL Server 2008	MySQL 5.1	PostgreSQL 8.4
Поддерживаемые операционные системы	1	2	2
Условии лицензирования	1	1	2
Процесс установки и поддержки	1	2	2
Наличие предустановленных драйверов в ОС семейства Windows	2	0	0
Наличие драйверов ODBC, JDBC, ADO.NET	2	2	2
Наличие View, доступных только для чтения	2	2	2
Наличие программных продуктов с открытым исходным кодом, основанных на этой СУБД	1	2	2
Использование в коммерческих проектах	1	2	1
Обновляемые View	2	1	2
Поддержка Materialized/Indexable Views	1	0	2
Возможность добавлять столбцы, изменять	2	2	2

названия, типы данных для view без их уничтожения			
Наличие графического ПО для конструирования и оптимизации запросов	2	1	2
Наличие Computed Columns	1	0	2
Поддержка функциональных индексов	1	0	0
Поддержка частичных индексов	2	0	2
Поддержка ACID-требований к транзакциям	2	1	2
Каскадное обновление/удаление внешних ключей	2	1	2
Внесение данных в несколько строк	2	2	2
Поддержка UPSERT-логики	2	2	0
Поддержка репликации	1	2	2
Возможность писать хранимые функции на разных языках программирования	1	0	2
Возможность создавать пользовательские агрегированные функции	2	1	2
Поддержка триггеров	2	2	2
Партицирование таблиц	1	2	2
Возможность создавать функции, возвращающие таблицу или набор таблиц, которые можно использовать в секции FROM запросов.	2	0	2
Поддержка создания функций	2	2	2
Поддержка хранимых процедур	2	2	2
Поддержка динамического SQL в функциях	0	0	2
Бесплатное ПО для графического управления БД	2	0	2
Наличие встроенного планировщика (не CronTab)	1	2	2
Возможность доступа к таблице из другой	2	2	2

базы данных, находящейся на том же хосте			
Чувствительность к регистру	0	0	2
Поддержка даты и времени	2	1	2
Аутентификация	2	1	2
Разграничение доступа к столбцам	2	2	2
Поддержка DISTINCT ON	0	0	2
Поддержка WITH ROLLUP	2	2	2
Поддержка WITH CUBE	2	0	0
Поддержка функций OVER..PARTITION BY	2	0	2
Поддержка рекурсивных запросов	2	0	2
Поддержка COUNT(DISTINCT), AGGREGATE(DISTINCT)	2	2	2
Поддержка OGC	2	2	2
Поддержка схем	2	0	2
Поддержка CROSS APPLY	2	0	1
Поддержка LIMIT .. OFFSET	0	2	2
Наличие Advanced Database Tuning Wizard	1	0	0
Наличие Maintenance Plan Wizard	1	0	0
Наличие Pluggable Storage Engine	0	2	0
Поддержка связанных подзапросов	2	2	2
Производительность планировщика запросов для сложных запросов	1	0	2
Наличие текстового процессора	2	2	2
Поддержка последовательностей и автоматической нумерации	2	2	2
Возможность откатить CREATE, ALTER	2	0	2

Для оценки показателей использовалась трёхбалльная система. Оценка производилась по критерию — за полную поддержку 2 балла, за частичную — 1 балл, за никакую — 0 баллов.

Результаты оценки:

- MS SQL — 81
- MySQL — 58
- PostgreSQL — 90.

По результатам сравнительного анализа (по средневзвешенной оценке: MS SQL — 1,5; MySQL — 1,07; PostgreSQL — 1,67;) СУБД PostgreSQL опережает своих конкурентов.

Так же приведем список проектов и компаний, которые используют в качестве СУБД PostgreSQL:

- Yahoo!
- MySpace
- OpenStreetMap
- Sony Online Entertainment
- BASF
- hi5.com
- Skype
- Sun xVM
- Evergreen

Далее для описания объектного моделирования в области разработки данного программного обеспечения будет использоваться язык графического описания для объектного моделирования в области разработки ПО — UML. UML является языком широкого профиля, это — открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью [31]. Все описанные далее диаграммы являются UML-моделями.

3.2 Диаграмма развертывания

Физическое представление программной системы не может быть полным, если отсутствует информация о том, на каких вычислительных средствах она реализована. Для представления общей конфигурации и топологии распределенной программной системы предназначены диаграммы развертывания. Диаграмма развертывания предназначена для визуализации элементов и компонентов программы. При этом представляются только компоненты-экземпляры программы, являющиеся исполняемыми файлами или динамическими библиотеками. Диаграмма развертывания содержит графические изображения процессоров, устройств, процессов и связей между ними.

При разработке диаграммы развертывания преследуются следующие цели:

- определить распределение компонентов системы по ее физическим узлам;
- показать физические связи между всеми узлами реализации системы на этапе ее исполнения;

В качестве элементов диаграммы используются: аппаратные компоненты («узлы») существуют (например, веб-сервер, сервер базы данных, сервер приложения), программные компоненты («Artifact») работают на каждом узле (например, веб-приложение, база данных), различные части этого комплекса соединяются, друг с другом изображаются отрезками линий без стрелок. Узлы представляются как прямоугольные параллелепипеды с артефактами, расположенными в них, изображенными в виде прямоугольников. Узлы могут иметь подузлы, которые представляются как вложенные прямоугольные параллелепипеды. Существует два типа узлов: узел устройства («Device»), узел среды выполнения (execution environment).

Опишем несколько представлений программной системы разрабатываемой в рамках дипломного проекта.

На первой диаграмме развертывания покажем связь и работу, которая реализована на данный момент:

На диаграмме показано, что программная система состоит из сервера базы данных (DataBaseServer) и рабочей станции, которая имеет подключение к данному серверу. Справа представлена рабочая станция, на которой инсталлировано и выполняется приложение podsystem.exe, которое является приложением, разработанным в дипломном проекте. Слева представлен узел устройства – сервер базы дынных, на котором развернуто такая среда выполнения как СУБД PostgreSQL, на которой в свою очередь может быть развернута любая база данных со своей группой таблиц (Group tables of DBMS) и двумя таблицами, таких как: таблица настроек (Table Settings), таблица записей аудита (Table Logs); которые создаются приложением podsystem.exe и инсталлируются в базу данных, к

которой приложение имеет подключение, а также с триггером (Trigger Log), генерируемым в приложении podsystem.exe и также инсталлируемым в базу данных, как и две выше описанных таблицы.

Общий смысл диаграммы в том, что на рабочей станции имеется приложение podsystem, которое подключается к базе данных, находящейся под управлением системы PostgreSQL и располагающейся на каком-то сервере базе данных, и добавляет в эту базу данных необходимые таблицы, а так же триггер со своими параметрами для аудита.

Данная диаграмма представлена на Рисунке 17.

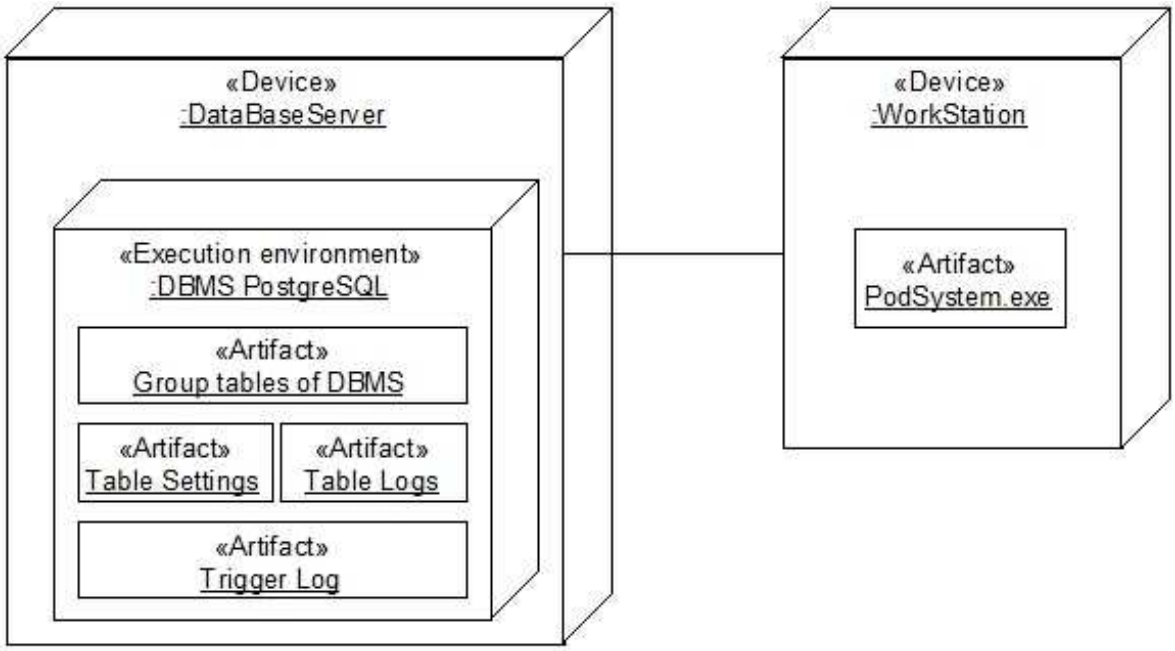

Рисунок 17 — Диаграмма развертывания

На второй диаграмме развертывания покажем связь и работу при более широком возможном использовании данного проекта:

На диаграмме представлена расширенная версия первой диаграммы. Дополнение: есть узел устройства веб-сервер (WebServer), на котором развернута некая среда выполнения приложений (Application) и есть сервер приложений (ApplicationServer), между этими двумя серверами установлено соединение. Сам сервер приложений может использовать базу данных на сервере базы данных, по также установленному между ними соединению.

Общий смысл диаграммы в том, что также имеется рабочая станция, подключение к БД, сервер БД, как и в предыдущей диаграмме. Добавляется только возможность

использования сервера баз данных неким другим сервером приложений. То есть можно предположить существование некоего веб-сервера, на котором установлены свои приложения, обслуживающие его работу, и работающие с сервером приложений, приложения которого могут использовать базу данных на сервере БД.

Данная диаграмма представлена на Рисунке 18.

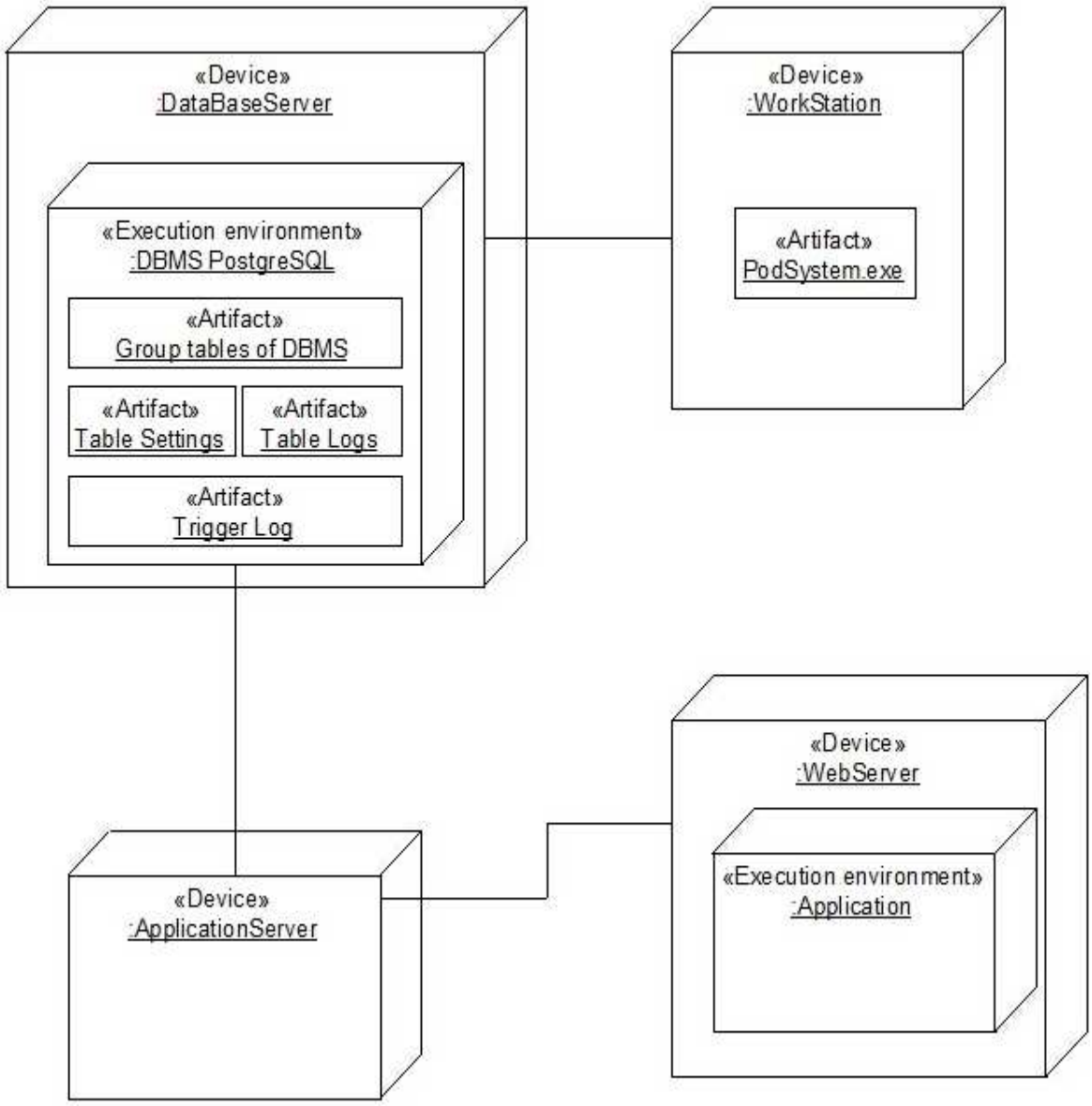

Рисунок 18 — Диаграмма развертывания

На третьей диаграмме развертывания покажем связь и работу при более широком возможном использовании проекта из второй диаграммы:

На диаграмме представлена расширенная версия второй диаграммы. Дополнение: есть ещё один узел устройства - сервер баз данных (DataBaseServer2), на котором, как и на первом сервере БД, развернута среда выполнения - СУБД PostgreSQL, на которой в свою

очередь может быть развернута любая база данных со своей группой таблиц (Group tables of DBMS2) и двумя таблицами: таблица настроек (Table Settings2), таблица записей аудита (Table Logs2); которые создаются приложением podsystem и инсталлируются в базу данных, к которой приложение имеет подключение, а также с триггером (Trigger Log2), генерируемым в приложении podsystem и также инсталлируемым в базу данных, как и две выше описанных таблицы.

Общий смысл диаграммы в том, что также имеется рабочая станция, подключение к БД, сервер БД, сервер приложений, веб-сервер, как и в предыдущей диаграмме. Добавляется только возможность подключения приложения podsystem ко второй базе данных, также под управлением системы PostgreSQL и располагающейся на другом сервере базе данных. Приложение podsystem также добавляет в эту базу данных необходимые таблицы и триггер со своими параметрами для аудита.

В данном проекте возможность работы с несколькими базами данных не реализована, но в дальнейшем может быть добавлена. Данное добавление значительно расширяет функциональность проекта.

Данная диаграмма представлена на Рисунке 19.

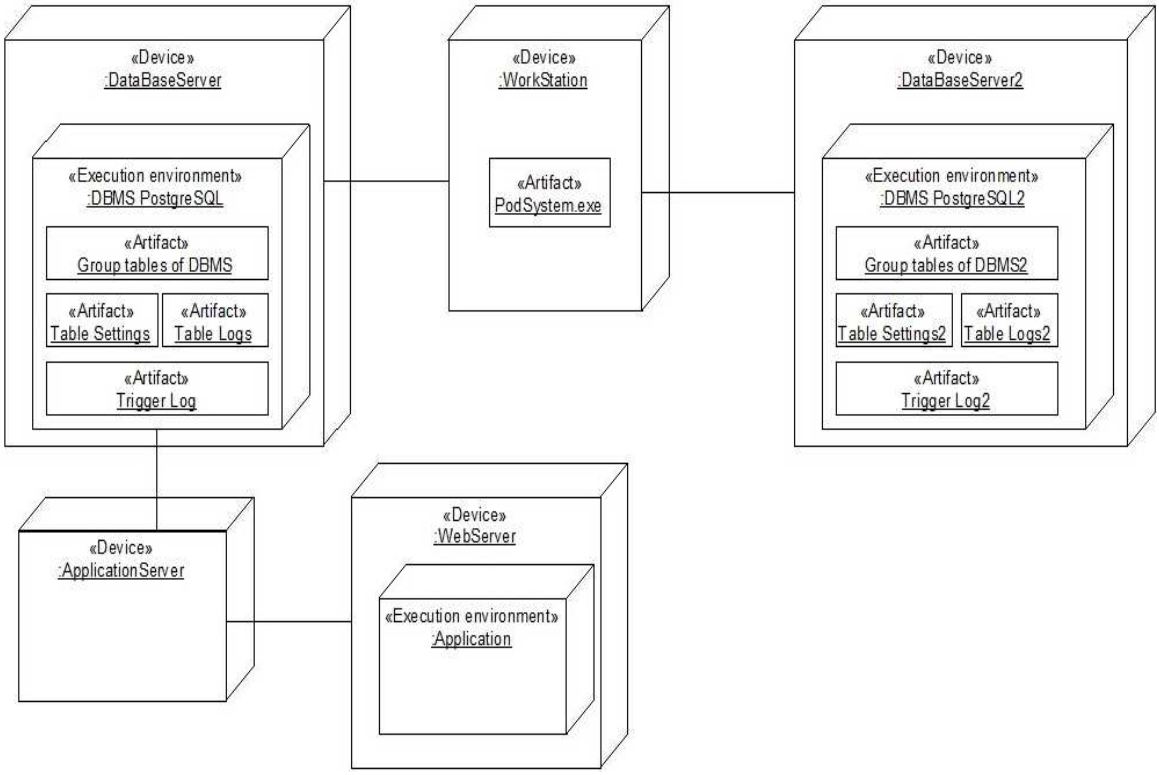

Рисунок 19 — Диаграмма развертывания

На четвертой диаграмме развертывания покажем связь и работу при самом широком возможном использовании проекта:

На диаграмме представлена расширенная версия третьей диаграммы. Дополнение: есть соединение между узлом - сервер приложений (ApplicationServer) и узлом - сервер баз данных (DataBaseServer2). Сервер приложений может использовать, по установленным между ними соединению, обе базы данных на сервере базы данных и сервере базы данных 2.

Общий смысл диаграммы в том, что также имеется рабочая станция, подключение к БД, сервер БД, сервер приложений, веб-сервер, возможность подключения приложения podsystem ко второй базе данных, также под управлением системы PostgreSQL и располагающейся на другом сервере базе данных. Добавляется возможность использования приложениями с сервера приложений второй базы данных, таким ж образом как это происходит в случае работы приложения podsystem [25][27][29].

Данная диаграмма представлена на Рисунке 20.

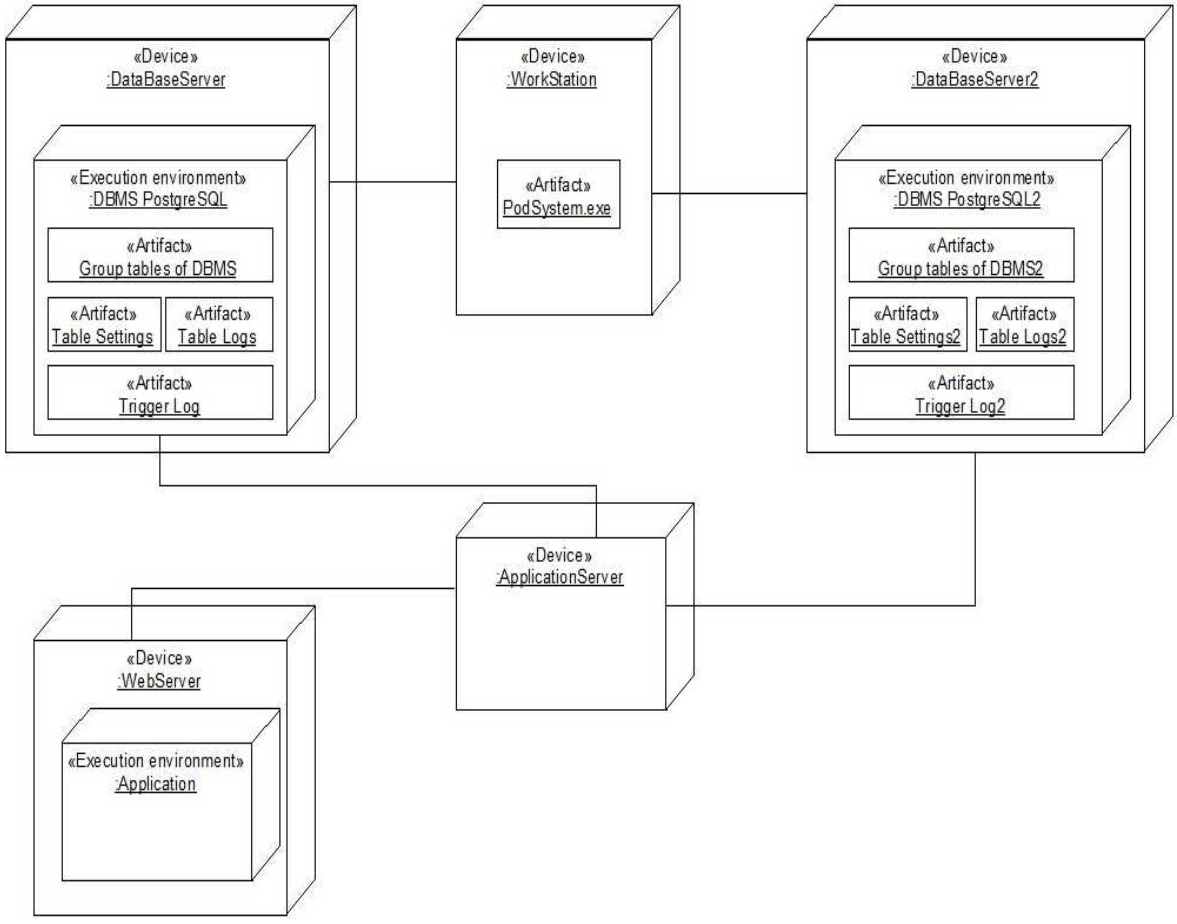

Рисунок 20 — Диаграмма развертывания

3.3 Диаграмма деятельности

Диаграммы деятельности позволяют моделировать сложный жизненный цикл объекта, с переходами из одного состояния (деятельности) в другое. Но этот вид диаграмм может быть использован и для описания динамики совокупности объектов. Они применимы и для детализации некоторой конкретной операции. Диаграммы деятельности описывают переход от одной деятельности к другой.

Действия показаны скругленными прямоугольниками; ромб - символ принятия решения с обозначениями условий возле переходов; жирная линия означает распараллеливание, а затем опять слияние воедино (синхронизацию) потоков управления [26][28].

Далее рассмотрим диаграмму деятельности для разрабатываемого проекта, которая представлена на рисунке 21.

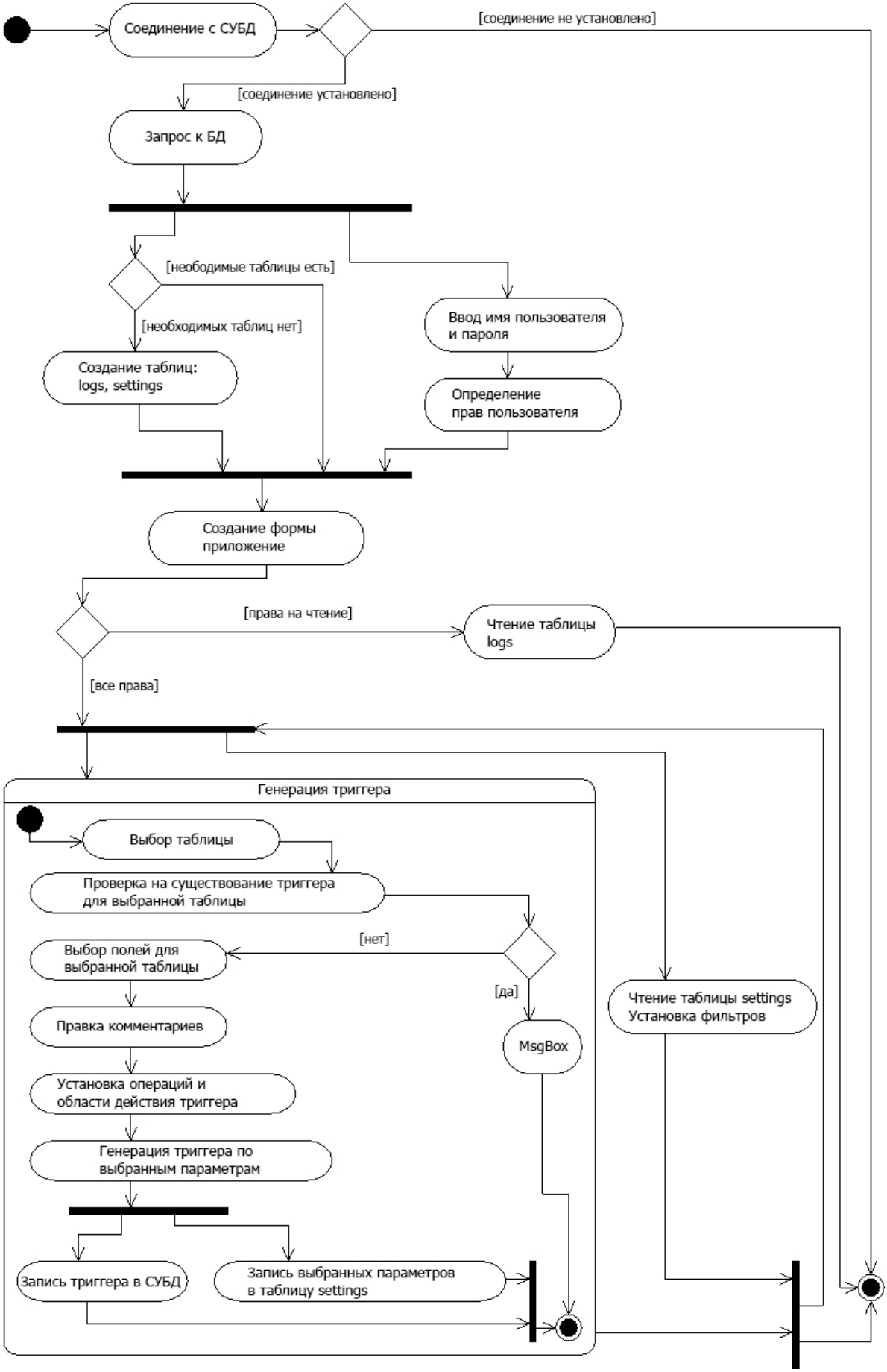

Рисунок 21 — Диаграмма деятельности

По диаграмме видно, что после запуска программы происходит операция соединение с СУБД, данное соединение происходит благодаря заранее созданному драйверу PostgreSQL. Далее следует символ принятия решения – если соединение не установлено, то дальнейшая работа приложения невозможна, и оно приходит к завершению, если же соединение установлено, то переходим к следующему этапу – запросу к БД. Операция запроса БД определяет, есть ли уже в БД необходимые для работы приложения таблицы. Далее идет распараллеливание, а именно одновременно выполняется две операции: первая операция — в зависимости от результата предыдущей операции принимается решение, если таблицы существуют, то происходит переход к следующей операции, если нет, то необходимые таблицы, а это таблицы настроек и таблица записей аудита, создаются в БД. Вторая операция — осуществляется операция ввода пароля и пользователя и дальнейшее определения прав этого пользователя. Далее происходит слияние потоков управления и создается форма приложения в которую загружается таблица записей аудита, если она была создана на предыдущем этапе, то пустая, если с ней была произведена какая-то работа, то с набором записей. В зависимости от того какие права были определены на предыдущем этапе для данного пользователя далее поток опять разветвляется. Если у данного пользователя имеются права только на чтение, то для него следует операция чтения без редактирования таблицы записей аудита и без изменения и чтения настроек, после чего возможен только выход из приложения. Если ж для пользователя определены все права, то переходим к следующей операции-генерации триггера, которая будет рассмотрена более подробно отдельно. Параллельно операции генерации, либо после нее возможна операция чтение и редактирование таблицы настроек, а также установка фильтров для просмотра таблицы записей аудита, которые находятся на форме приложения. После этих операций возможен либо выход либо их неоднократный повтор.

Операция генерации триггера: генерация триггера начинается с выбора таблицы, далее выполняется операция проверки на существование в БД триггера на выбранную таблицу, поскольку создание нескольких триггерных функций на одну таблицу невозможна; После этого происходит ветвлении на две операции: первая операция — если есть такая триггерной функция, то появляется сообщение об этом вторая; операция — если нет, то переходим к следующему этапу. Следующий этап заключается в операции выбора полей для выбранной таблицы. Потом можно добавить комментарии для выбранных полей или же оставить их пустыми. Также возможно дальнейшая операция установки области действия триггера и установки операций на которые будет срабатывать данный триггер. После всех этих операция происходит распараллеливание. Параллельно

выполняется операция записи триггера в БД и операция записи в таблицу настроек выбранных параметров (выбранные поля, выбранная таблица, имя пользователя, который совершил все операции при генерации). На этом генерация заканчивается.

3.4 Описание структуры программного продукта

Рассмотрим подробно разработанный программный продукт в данном дипломном проекте.

В качестве СУБД была использована СУБД PostgreSQL версии 8.4, в качестве языка программирования C++, а в качества инструментария для разработки ПО - Qt версии 4.7.0 [37].

Версия PostgreSQL выбрана из идеи, что возможности новых версий для нужной функциональности в проекте не столь необходимы. Использование более новых версий, по сути, никак бы не повиляло на разработку дипломного проекта, при том, что использование самого проекта предполагает версии PostgreSQL старше 8.3.

База данных, с которой ПП устанавливает соединение и производит определенные операции, была выбрана, произвольна, в ее качестве может выступать абсолютно любая БД под СУБД PostgreSQL. Программный код представлен в Приложении 1.

3.4.1 Используемые таблицы в базе данных

Для функционирования ПО, разработанного в рамках дипломного проекта, необходимо добавить в БД, с которой ПО устанавливает соединение, две таблицы: таблицу logs (для записей проводимого аудита), таблицу settings (для записи настроек триггера). Структура этих таблиц описана в таблице.

Таблица 2 Структура таблиц

Таблица logs	Таблица settings
text	table_name
added	
username	pole_name
tablename	

Рассмотрим таблицу logs:

поле text предназначено для записей проводимого аудита данной СУБД данным ПО; поле added для записи времени - когда было произведено событие аудита; поле

username для записи имени пользователя, который осуществил событие аудита; поле tablename для записи имени таблицы в которой происходит само событие.

Рассмотрим таблицу settings:

поле table_name предназначено для записи имени таблицы, на которую был создан триггер, а поле pole_name - на какие поля был создан этот триггер.

3.4.2 Используемые SQL-запросы

В ходе работы, ПО посылает СУБД некоторые необходимые запросы в формате SQL [31]. Рассмотрим наиболее важные из них.

Создание таблиц

1) Создание таблицы logs (для записи результатов аудита)

```
CREATE TABLE logs (
  "text" text,
  added timestamp without time zone,
  username text,
  tablename text)
```

2) Создание таблицы settings (для хранения информации об установленных триггерах и полей, на которые срабатывает данный триггер)

```
CREATE TABLE settings (
  table_name text,
  pole_name text)
```

3) Запрос полей в виде списка заданной таблицы, для дальнейшего выбора для создания триггера.

```
SELECT attname FROM pg_attribute, pg_type
WHERE typname = 'jilci'
AND attrelid = typrelid
AND attname NOT IN ('cmin', 'cmax', 'ctid', 'oid', 'tableoid', 'xmin', 'xmax');
```

4) Запрос существующих таблиц в БД, при создании триггера, исключая служебные таблицы

```
SELECT tablename FROM pg_tables
WHERE tablename NOT LIKE 'pg\\_%'
```

```
AND tablename NOT LIKE 'sql\\_%';
AND tablename NOT LIKE 'logs' AND tablename NOT LIKE 'settings'
```

5) Запрос комментариев по заданной таблице и заданному индексу нужного поля.

```
select description from pg_description
join pg_class on pg_description.objoid = pg_class.oid
where relname = 'jilci' and objsubid = 5
```

6) Установка комментариев выбранного поля в выбранной таблице

```
COMMENT ON COLUMN jilci.num IS 'Номер квартиры';
```

7) Удаление существующего триггера для выбранной таблицы.п
Применяется каскадное удаление, поскольку триггер связан с триггерной функцией.

```
DROP FUNCTION add_to_log_jilci () CASCADE
```

8) Создание триггера на определенную триггерную функцию.

```
CREATE TRIGGER log
AFTER INSERT OR UPDATE OR DELETE ON jilci FOR EACH ROW EXECUTE PROCEDURE add_to_log_jilci ();
```

9) Запись настроек триггера

```
INSERT INTO settings (table_name, pole_name) VALUES ('table_name, list_pole')
```

10) Удаление настроек триггера

```
DELETE FROM settings WHERE table_name = 'table_name'
```

11) Запрос поля содержащее время, с использованием функции date для нужного форматирования [32]

```
SELECT DISTINCT date(added) FROM logs
```

12) Генерация и создание триггерной функцию (пример на таблицу jilci при выборе полей fio и num)

```
CREATE OR REPLACE FUNCTION add_to_log_jilci()
  RETURNS trigger AS
$BODY$ DECLARE
mstr varchar(30);
astr_0 varchar(100);
astr_o_0 varchar(100);
astr_1 varchar(100);
astr_o_1 varchar(100);
tmp varchar(30);
retstr varchar(254);

BEGIN  IF TG_OP = 'INSERT' THEN
astr_0= NEW.num;
astr_1= NEW.fio;
mstr := ' Добавление нового: ';
retstr := ' ';

if NEW.num is not null then
retstr := retstr || mstr;
tmp=(select description from pg_description join pg_class on pg_description.objoid = pg_class.oid where relname = 'jilci' and objsubid = 5);

if tmp is not null then
retstr := retstr || tmp;
retstr := retstr || ' ';
end if;
```

```
retstr := retstr || astr_0;
end if;

if NEW.fio is not null then
retstr := retstr || mstr;
tmp=(select description from pg_description join pg_class on pg_description.objoid = pg_class.oid where relname = 'jilci' and objsubid = 2);

if tmp is not null then
retstr := retstr || tmp;
retstr := retstr || ' ';
end if;

retstr := retstr || astr_1;
end if;

INSERT INTO logs(text,added) values (retstr,NOW());
RETURN NEW;

ELSIF TG_OP = 'UPDATE' THEN
astr_0= NEW.num;
astr_1= NEW.fio;
astr_o_0= OLD.num;
astr_o_1= OLD.fio;
mstr := ' Изменение: ';
retstr := ' ';

if (astr_0<>astr_o_0) then
retstr := retstr || mstr;
tmp=(select description from pg_description join pg_class on pg_description.objoid = pg_class.oid where relname = 'jilci' and objsubid = 5);
if tmp is not null then
retstr := retstr || tmp;
retstr := retstr || ' ';
end if;
```

```
retstr := retstr || 'с ';
retstr := retstr || astr_o_0;
retstr := retstr || ' на ';
retstr := retstr || astr_0;
end if;

if (astr_1<>astr_o_1) then
retstr := retstr || mstr;
tmp=(select description from pg_description join pg_class on pg_description.objoid = pg_class.oid where relname = 'jilci' and objsubid = 2);
if tmp is not null then
retstr := retstr || tmp;
retstr := retstr || ' ';
end if;

retstr := retstr || 'с ';
retstr := retstr || astr_o_1;
retstr := retstr || ' на ';
retstr := retstr || astr_1;
end if;

INSERT INTO logs(text,added) values (retstr,NOW());
RETURN NEW;

ELSIF TG_OP = 'DELETE' THEN
astr_o_0= OLD.num;
astr_o_1= OLD.fio;
mstr := ' Удаление: ';
retstr := ' ';

if OLD.num is not null then
retstr := retstr || mstr;
tmp=(select description from pg_description join pg_class on pg_description.objoid = pg_class.oid where relname = 'jilci' and objsubid = 5);
```

```
if tmp is not null then
retstr := retstr || tmp;
retstr := retstr || ' ';
end if;

retstr := retstr || astr_o_0;
end if;

if OLD.fio is not null then
retstr := retstr || mstr;
tmp=(select description from pg_description join pg_class on pg_description.objoid = pg_class.oid where relname = 'jilci' and objsubid = 2);
if tmp is not null then
retstr := retstr || tmp;
retstr := retstr || ' ';
end if;
retstr := retstr || astr_o_1;
end if;
INSERT INTO logs(text,added) values (retstr,NOW());
RETURN OLD;
END IF;
END;
$BODY$
  LANGUAGE 'plpgsql' VOLATILE
  COST 100;
ALTER FUNCTION add_to_log_jilci() OWNER TO postgres;
```

3.4.3 Снимки экранов интерфейса с описанием

При запуске ПП, появляется окно (рис.22) с запросом имени пользователя и пароля.

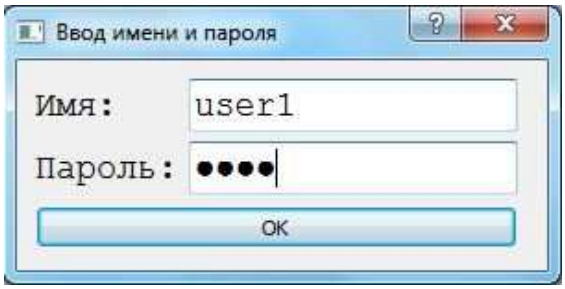

Рисунок 22 — Запрос имени и пользователя

После ввода имени и пользователя открывается главное окно приложения (рис.23).

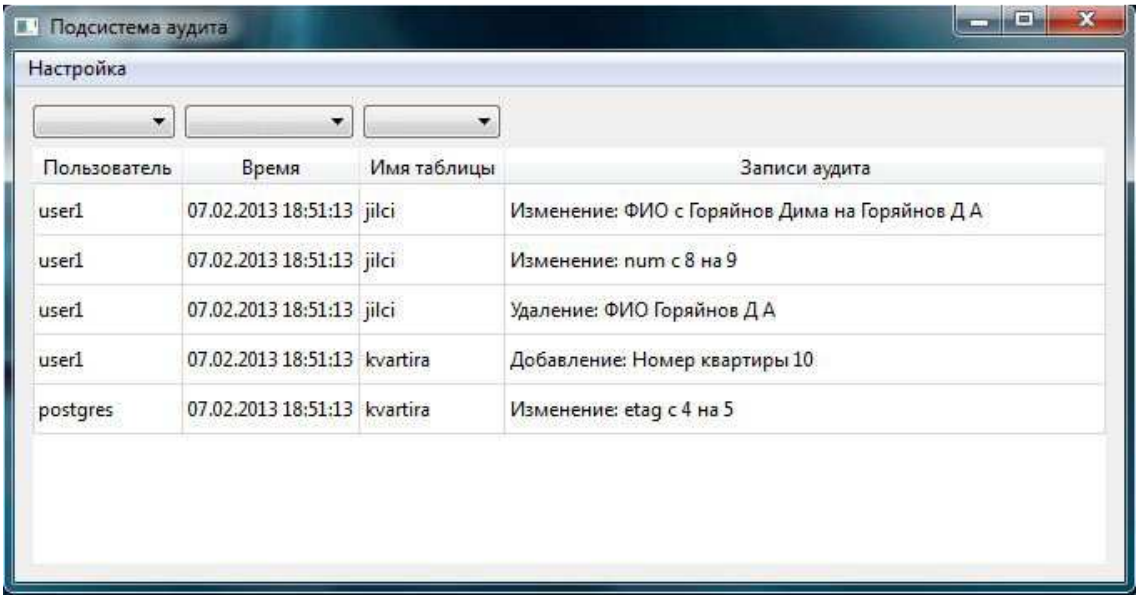

Рисунок 23 — Главное окно

Все последующие действия выполняются именно из этого окна.

Главное окно имеет меню, фильтры таблицы и саму таблицу, которая сама является представлением таблицы logs, которая имеется в БД.

При выборе фильтров пользователя (рис.24), времени или таблицы (рис.25) происходит обновление таблицы согласно выбранным фильтрам, как это видно на рис.26.

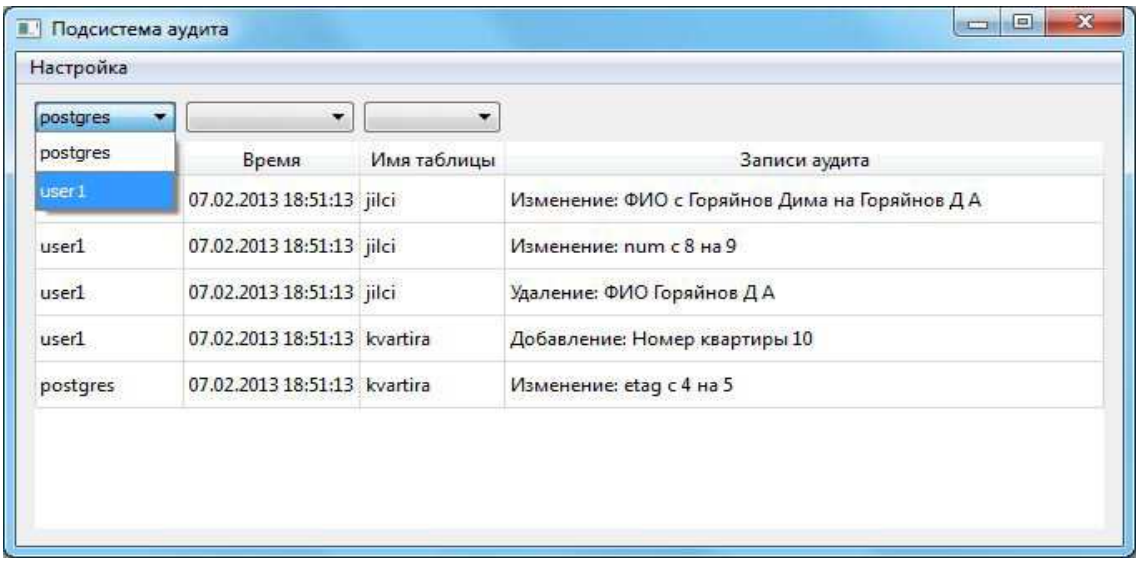

Рисунок 24 — Фильтры главного окна

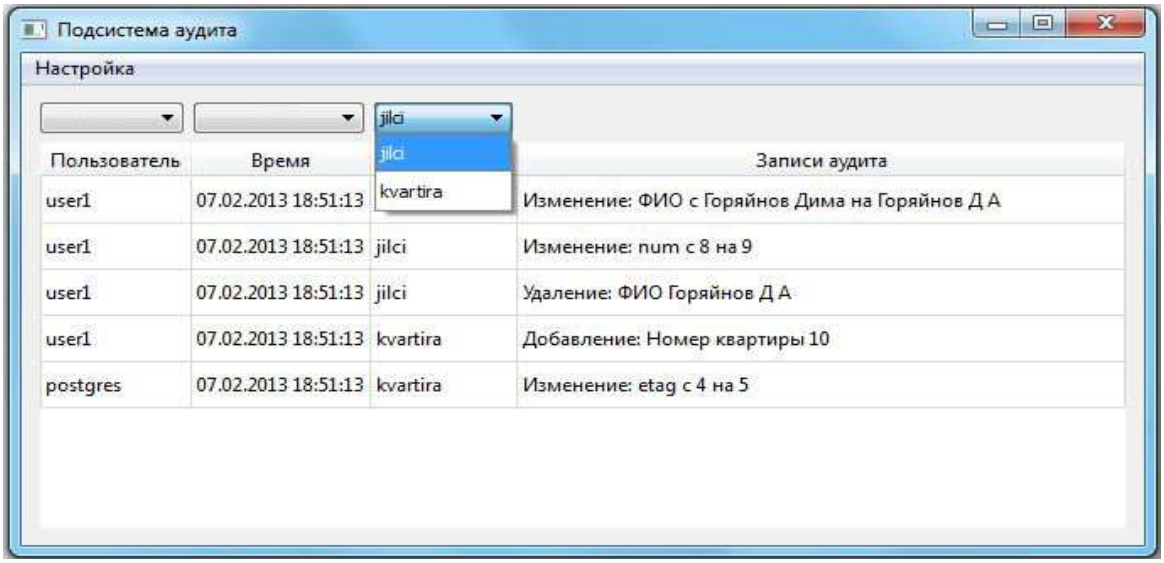

Рисунок 25 — Фильтры главное окна

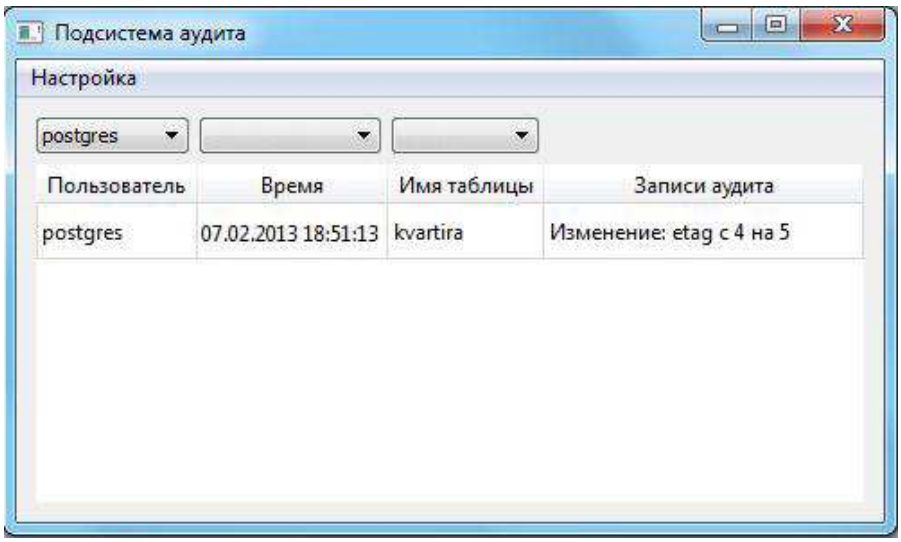

Рисунок 26 — Использование фильтров

Рассмотрим остальную функциональность приложения.

В приложение имеется меню, которое состоит из 2 пунктов: "создание триггера" и "просмотр таблицы настроек" (рис.27).

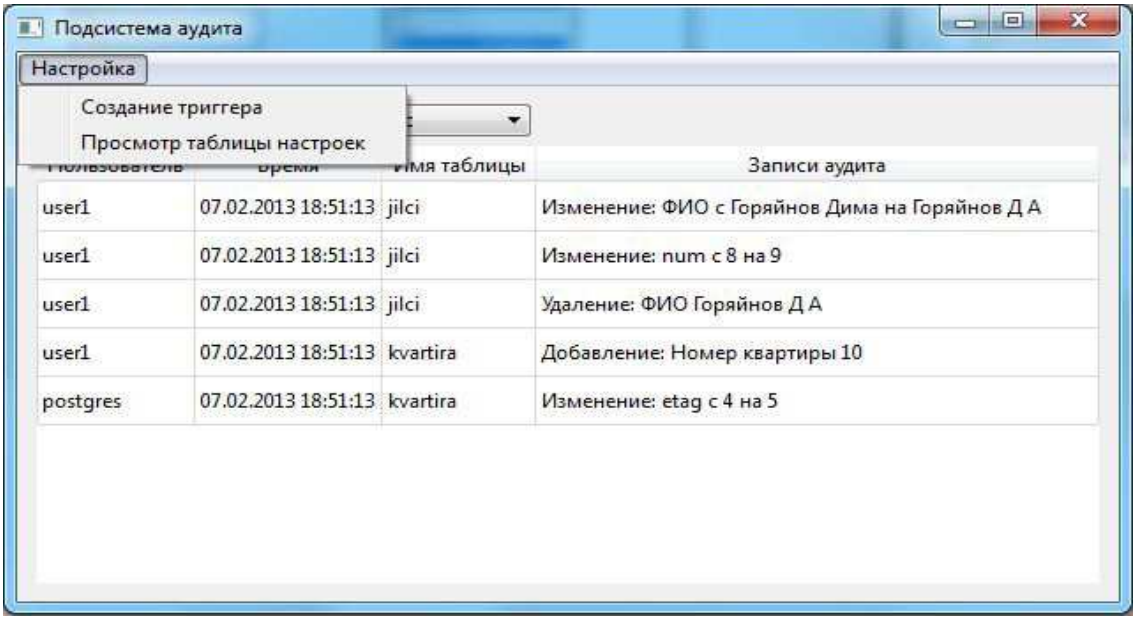

Рисунок 27 — Меню главного окна

При выборе первого пункта появляется окно генерации триггера (рис.28). При генерации триггера можно выбрать операции на который он будет срабатывать, а также для чего выполнение триггера предназначено. Также в этом окне можно выбрать из списка таблиц нужную таблицу, на которую и создается триггер, и поля этой таблицы, аудит которых будет производится. В процессе выбора полей можно добавить комментарии, с которыми и будут записывать в таблицу logs отслеживаемые операции

(рис.29). После нажатия на кнопку "Создать триггер" происходит генерации триггера по выбранным параметрам.

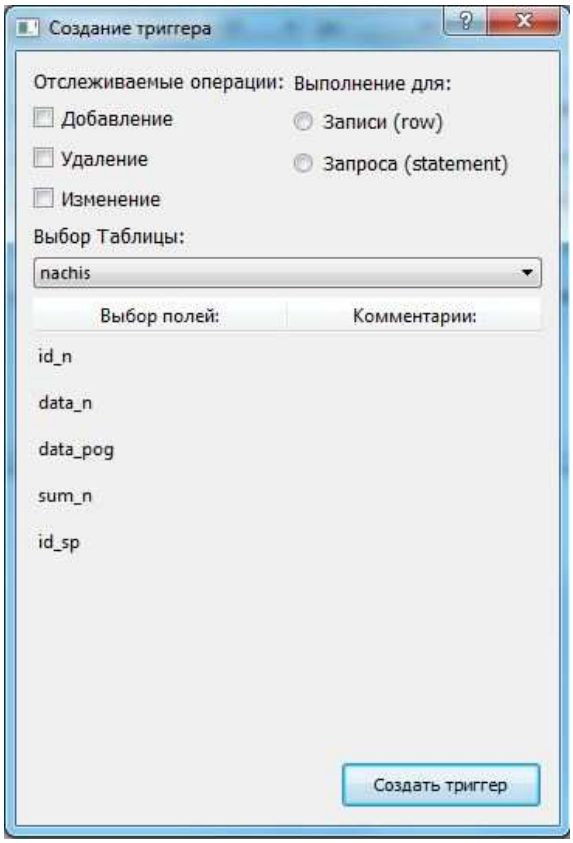

Рисунок 28 — Создание триггера

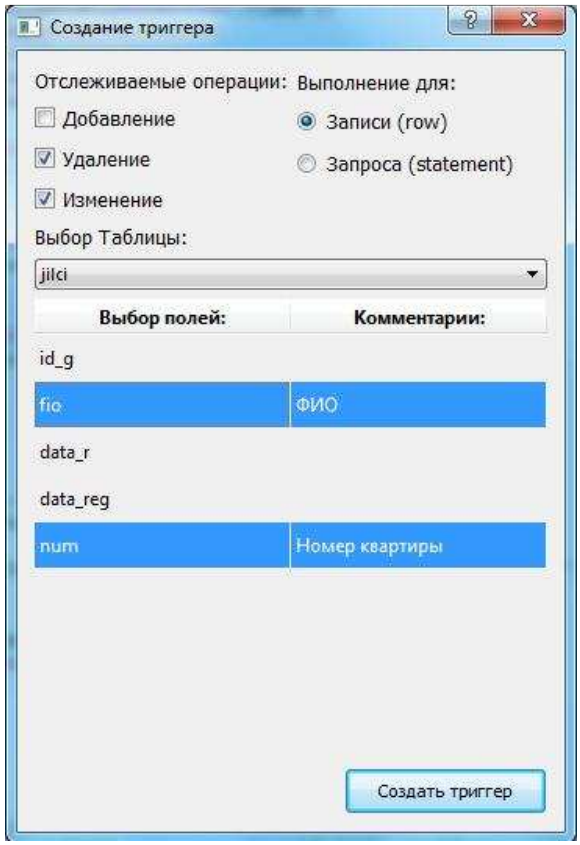

Рисунок 29 — Выбор полей

Предусмотрена защита от добавления триггера на таблицу, где триггер уже существует, при попытке это сделать появляется окно (рис. 30)

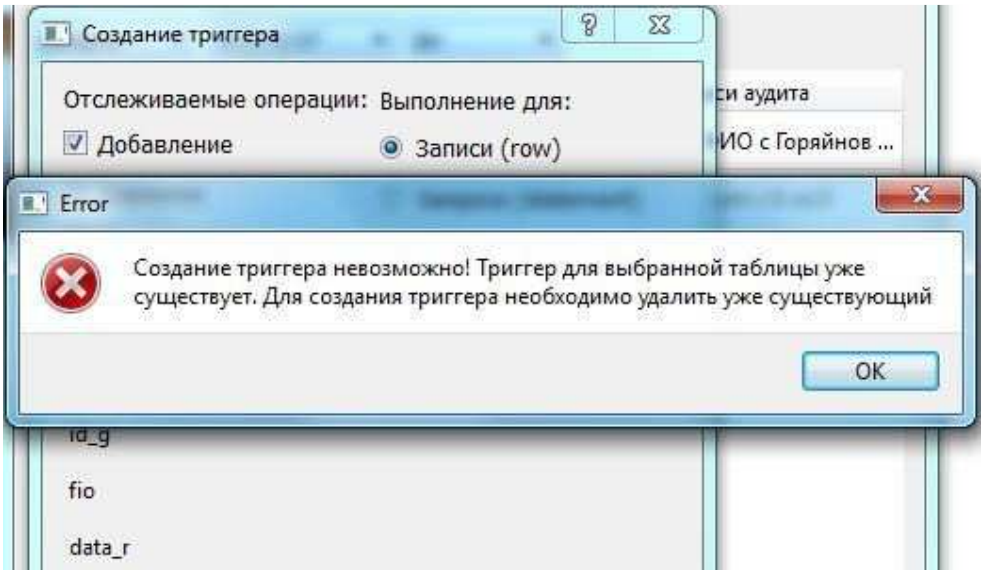

Рисунок 30 — Защита от повторения

При выборе в меню главного окна второго пункта появится окно (рис. 31), в котором можно увидеть какие триггеры на какие таблицы и поля были созданы. Также есть возможность удалить его нажатием кнопки 'delete', при этом триггер и связанная с ним триггерная функция, а также записи в таблице настроек удаляться.

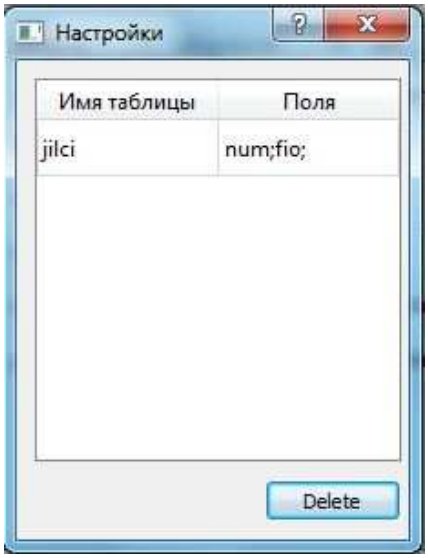

Рисунок 31 — Настройки

3.5 Выводы

В рамках данной главы была рассмотрена практическая составляющая дипломного проекта — составление модели и разработка подсистемы аудита СУБД, а также рассмотрены некоторые ее возможности и будущие дополнения.

4. Экономическая оценка

4.1 Концепция экономического обоснования

Необходимость разработки проекта «Разработка подсистемы аудита информационной системы» и создание программного решения в рамках него обусловлено тем, что требованиями бизнеса, предъявляемые к определению уровня обеспечения информационной безопасности, и существенный рост рисков от нарушения информационной безопасности во всех сферах жизнедеятельности общества и государства, диктуют настоятельную необходимость использовать в своей работе обоснованные методы и средства, позволяющие количественно и качественно измерять уровень защищенности организаций и систем информационной технологий, а также оценивать экономическую эффективность затрат на информационную безопасность. Одним из направлений, позволяющих оценить уровень обеспечения информационной безопасности, является аудит информационной безопасности.

На сегодняшний день СУБД играют ключевую роль в обеспечении эффективного выполнения процессов предприятий. Именно для решения проблем защита и применяется аудит безопасности системы.

Для контроля и оценки состояния защищенности информационной системы необходимо постоянно отслеживать и анализировать данные, которые имеют неподходящий для понимания формат, поэтому в проекте предусмотрен формат записи отслеживаемых данных, что в свою очередь повышает эффективность анализа.

Повышение защищенности информации происходит благодаря разработке системы, которая обеспечивает эффективное обнаружения мошеннически введенных данных и несанкционированных запросов, их фиксированию и представлению результатов в доступной форме.

Данная продукция находится на начальной стадии разработки и не предполагает на данном этапе ее дальнейшую продажу.

В связи с вышеизложенным производится экономическая оценка.

4.2 Трудоемкость выполнения НИР

В основе определения стоимости разработки системы лежит перечень выполненных работ и их трудоемкость. Перечень работ по разработке системы и трудоемкость их выполнения приведены в таблице 3

Таблица 3 – Трудоемкость выполнения НИР

Наименование работ	Трудоемкость, чел./дни		Машинное время, час
	Старший научный сотрудник	Инженер	
Разработка и утверждение технического задания	1	4	4
Изучение современного состояния в сфере аудита событий безопасности	-	7	33
Составление обзора существующих средств и методов аудита событий безопасности СУБД	-	10	50
Анализ собранной информации	-	3	21
Разработка новой подсистемы аудита событий безопасности, основанной на бесплатной СУБД PostreSQL	1	10	120
Разработка интерфейса настройки и визуализации подситсемы	1	10	
Реализация разработанной подсистемы аудиты, основанной на бесплатной СУБД PostreSQL	1	23	99
Тестирование разработанной подсистемы аудита	1	5	22
Анализ проделанной работы	1	5	20
Оформление пояснительной записки	-	7	59
Сдача проекта	1	1	-
ИТОГО:	7	85	428

На основе трудоемкости выполнения работ по разработке системы рассчитываются издержки на оплату труда ее исполнителей, являющиеся одной из основных статей калькуляции себестоимости разработки.

4.3 Смета затрат на проведение НИР

В этом разделе мы оценим затраты, необходимые на проведение разработки научно-технического продукта.

4.3.1 Статья «Материалы»

В данную статью включаются затраты на материалы, необходимые для выполнения НИР с учетом транспортно-заготовительных расходов. Транспортно-заготовительные расходы составляют 15% от расходов на материалы.

Затраты на статью «Материалы» представлены в таблице 4.

Таблица 4 – Расходы, относящие к статье «Материалы»

Наименование материалов	Количество, шт.	Цена за единицу, руб.	Стоимость, руб.
Бумага, пачка	1	150,0	150,0
Картридж для принтера черный	1	1489,0	1489,0
Картридж для принтера цветной	1	2190,0	2190,0
CD-диск	1	50,0	50,0
ИТОГО:			**3879,0**
Транспортно-заготовительные расходы (15%)			**581,85**
ИТОГО:			**4460,85**

4.3.2 Статья «Спецоборудование»

Расходы на специальное оборудование не предусмотрены.

4.3.3 Статья «Расходы на оплату труда»

Основная заработная плата исполнителей рассчитывается по формуле:

$$C_{зо} = \frac{\sum T_t \bullet C_{зо_{мес}}}{23}(1+\frac{H_н}{100}),$$

где T_1 - трудоемкость выполнения работ руководителя и инженера (см. табл. 4.1)

$T_{рук}$ = 13 чел/дней, $T_{инж}$ = 98 чел/дней, соответственно:

$C_{зо_{мес}}$ - месячные оклады исполнителей: старшего научного сотрудника - 40000 руб, инженера 25000 руб.;

$H_н$ - норматив начислений, 12%;

Т - среднее количество рабочих дней в месяц (23 дня).

$$C_{зо} = (\frac{7 \bullet 40000}{23} + \frac{85 \bullet 25000}{23}) \bullet 1{,}12 = (12174 + 92391.3) \bullet 1{,}12 = 117113$$

руб.

Основная заработная плата исполнителей равна 117113 руб.

4.3.4 Статья «Страховые взносы в государственные внебюджетные фонды»

Данная статья рассчитывается пропорционально заработной плате разработчиков в размере 30%, в том числе:

- фонд социального страхования - 2,9 %;
- пенсионный фонд - 22%;
- федеральный фонд обязательного медицинского страхования – 2,1%;
- территориальный фонд обязательного медицинского страхования - 3%;
- страхование от несчастных случаев - 0,5%.

$$C_{СН} = {C_{ЗО} \bullet H_{СН}}/{100},$$

$$C_{СН} = 117113 \bullet 0{,}3 = 35134 \ руб.$$

Страховые взносы в государственные внебюджетные фонды: 39818,4 руб.

4.3.5 Статья «Затраты по работам, выполняемым сторонними организциями»

В качестве расходов на оплату услуг сторонних организаций при выполнении разработки условно выступает стоимость машинного времени.

Стоимость машинного времени рассчитывается по формуле:

$$C_{MB} = t_{MB} \bullet P_{MB}$$

где t_{MB} - время использования ПЭВМ (428 час), P_{MB} - стоимости часа машинного времени (30 руб./час).

$$C_{MB} = 428 \bullet 30 = 12840 \text{ руб.}$$

Стоимость машинного времени равна 12840 руб.

4.3.6 Статья «Командировочные расходы»

Затраты на служебные командировки работников не предусмотрены.

4.3.7 Статья «Прочие прямые расходы»

Затраты по данной статье составляют затраты по средствам связи и коммуникаций (интернет), по получению НТИ и другое, поэтому:

$$C_{ППР} = C_{КОМ} \bullet k = C_{ЧАС} \bullet t_{ИНТ} \bullet k$$

где $t_{ИНТ}$ - время использования интернета (336 час), $C_{ЧАС}$ - стоимости часа интернета (2 руб./час), $k = 1.2$ - коэффициент, учитывающий остальные виды прочих расходов.

$$C_{ППР} = 336 \bullet 2 \bullet 1.2 = 806 \text{ руб.}$$

Прочие прямые расходы составляют 806 руб.

4.3.8 Статья «Накладные расходы»

В эту статью включаются расходы на управление и хозяйственное обслуживание НТПр. Величина накладных расходов определяется на основании норматива, установленного в СПбГЭТУ и берется равной 33%.

$$C_{HP} = C_{ЗО} \bullet H_{HP} \Big/ 100$$

$$C_{HP} = 117113 \bullet 33 \Big/ 100 = 38647 \text{ руб.}$$

Накладные расходы оставляют 38647 руб.

4.3.9 Статья «Себестоимость НТПр»

Себестоимость рассчитывается по формуле:

$$C = C_M + C_{ЗО} + C_{СН} + C_{МВ} + C_{ППР} + C_{HP}$$

$$C = 213685 \text{ руб.}$$

Себестоимость НТПр составляет 213685 руб.

На основании полученных данных в таблице 5 приведена калькуляция себестоимости НТПр.

Таблица 5 – Расходы, относящие к статье «Себестоимость НТПр»

Наименование статьи расходов	Сумма, руб.
Материалы	4461
Спецоборудование	Отсутствует
Расходы на оплату труда	117113
Страховые взносы в государственные внебюджетные фонды	35134
Затраты по работам, выполняемым сторонними организциями	12840
Командировочные расходы	Отсутствуют
Прочие прямые расходы	806
Накладные расходы	38647
ИТОГО себестоимость:	**209001**

4.4 Экономическая оценка НИР

Существовала проблема защиты системы хранения данных, с помощью аудита этой системы, и сокращения расходов на саму систему путем ее создания на бесплатной основе.

В процессе выполнения разработки разработан подход, обеспечивающий низкую стоимость и высокую эффективность, но требующий дальнейшей проработки.

В связи с этим, на данном этапе выполнения НИР отсутствует информации о применении разработки, а так же возможности ее продажи.

Поэтому производится экономическая оценка эффективности НИР, которая включает качественную оценку эффекта от разработки, в том числе научно-технический уровень качества разработки.

Оценка уровня качества осуществляется по следующим параметрам

1) Надежность системы
2) Эффективное обнаружение мошеннически введенных данных и несанкционированных запросов
3) Фиксирование отслеживаемых действий в системе
4) Оценка и анализ аудита неправомерных действий и дальнейшее использования полученной информации для улучшения информационной безопасности системы

Оценка уровня качества НТПр представлена в таблице 6.

Таблица 6 – Оценка уровня качества НТПр

№ п\п	Показатель	Бальная оценка		Коэффициент значимости, W_i
		Традиционная система, $Б_{КОН_i}$	Новая система, $Б_i$	
1	Эффективное обнаружение мошеннически введенных данных и несанкционированных запросов	9	9	0,2
2	Фиксирование отслеживаемых действий в системе	4	6	0,3
3	Представление результатов аудита в доступной форме	3	7	0,2
4	Настройка доступа к системе	6	9	0,2
5	Представление информации, повышающее эффективность работы системы	6	8	0,1

Уровень качества определяется по формуле:

$$K_{КАЧ} = \sum_{i=1}^{n} W_i \frac{Б_i}{Б_{КОН_i}}$$

$$K_{КАЧ} = 0{,}2 \cdot \frac{9}{9} + 0{,}3 \cdot \frac{6}{4} + 0{,}2 \cdot \frac{7}{3} + 0{,}2 \cdot \frac{9}{6} + 0{,}1 \cdot \frac{8}{6} = 1{,}55$$

Уровень качества равен 1,55.

Повышение уровня качества оправдывает разработку, т.к. приводит к сокращению потерь пользователя более высокому сокращению, чем при использовании традиционной системы, а также положительным социальным последствиям.

4.5 Выводы

1) Трудоемкость НИР – 92 чел-дни
2) Себестоимость НИР составляет 213685 руб.
3) Уровень качества равен 1,55
4) Эффективное выявление причин, способствующих повышению защищенности информации НТПр обеспечивается за счет:

а) Эффективного обнаружение мошеннически введенных данных и несанкционированных запросов

б) Представление результатов аудита, повышающих эффективность работы системы

в) Представление данных результатов в доступной форме

Все вышеизложенное делает работу экономически целесообразной.

5. Защита интеллектуальной собственности

5.1 Введение

Мною Горяйновым Дмитрием Андреевичем в процессе выполнения дипломного проекта разработана программа для ЭВМ «Подсистема аудита ИС».

Этот результат научно-технической деятельности входит в перечень охраняемых объектов интеллектуальной собственности Гражданского Кодекса РФ. Программа разработана по личной инициативе. Правообладателем программы является Горяйнов Дмитрий Андреевич.

5.2 Программы для ЭВМ и базы данных как объекты интеллектуальной собственности. Описание и определение

5.2.1 Интеллектуальная собственность

Под **интеллектуальной собственностью** понимают особый вид гражданских прав (исключительное право) в отношении результатов интеллектуальной деятельности, таких как изобретения, промышленные образцы (дизайн), компьютерные программы, другие произведения науки, произведения литературы, искусства, которые принято называть **объектами интеллектуальной собственности,** а также различных средств индивидуализации производителя товаров и услуг, таких как товарные знаки, знаки обслуживания, фирменные наименования и др. [2, ст. 1225]. Основным содержанием таких прав является монополия их владельца на использование этих объектов, включая право запретить или разрешить их использование другим, а также право переуступить другому лицу эти правомочия или отказаться от них вовсе.

Согласно определению интеллектуальной собственности, принятому в российском законодательстве, а также на основании определения Стокгольмской конференции от 14 июля 1967 г., программы для ЭВМ (компьютерные программы) и базы данных относятся к объектам интеллектуальной собственности. Программам для ЭВМ и базам данных предоставляется охрана нормами авторского права как литературным произведениям в соответствии с Бернской конвенцией, причем программы для ЭВМ охраняются как литературные произведения, а базы данных - как сборники.

В Российской Федерации вопросы предоставления правовой охраны программам для ЭВМ и базам данных регулируются Гражданским кодексом РФ, Часть 4 (ГК РФ Ч.4).

5.2.2 Программа для ЭВМ

Под **программой для ЭВМ** понимается *"... представленная в объективной форме совокупность данных и команд, предназначенных для функционирования ЭВМ и других*

компьютерных устройств в целях получения определенного результата". Кроме того, в понятие программы для ЭВМ входят *"...подготовительные материалы, полученные в ходе разработки программы для ЭВМ, и порождаемые ею аудиовизуальные отображения"* [2, ст. 1261].

С точки зрения программистов и пользователей программа для ЭВМ представляет собой детализацию алгоритма решения какой-либо задачи и выражена в форме определенной последовательности предписаний, обеспечивающих выполнение компьютером преобразования исходных данных в искомый результат.

Можно выделить следующие объективные формы представления программы для ЭВМ:

- **исходная программа** (или исходный текст) - последовательность предписаний на алгоритмическом (понятном человеку) языке высокого уровня, предназначенных для автоматизированного перевода этих предписаний в последовательность команд в объектном коде;
- **рабочая программа** (или объектный код) - последовательность машинных команд, т. е. команд, представленных на языке, понятном ЭВМ;
- **программа, временно введенная в память ЭВМ,** - совокупность физических состояний элементов памяти запоминающего устройства ЭВМ (ОЗУ), сохраняющихся до прекращения подачи электропитания к ЭВМ;
- **программа, постоянно хранимая в памяти ЭВМ,** - представленная на языке машины команда (или серия команд), выполненная в виде физических особенностей участка интегральной схемы, сохраняющихся независимо от подачи электропитания [4].

Исходная и рабочая программы, как правило, представляются в виде записи на том или ином языке, выполненной на бумаге или машиночитаемом носителе данных: магнитном или оптическом диске, магнитной ленте и т. п.

Предоставляемая законодательством правовая охрана распространяется *"... на все виды программ для ЭВМ (в том числе на операционные системы и программные комплексы), которые могут быть выражены на любом языке и в любой форме, включая исходный текст и объектный код …"* [2, ст.1261]. Так как преобразование исходного текста программы для ЭВМ в объектный (машинный) код с помощью специальных программ-трансляторов не меняет сущности данной программы как произведения, то если охраняется исходный текст программы, значит, охране подлежит и соответствующий ей объектный код. Обратное тоже справедливо.

Правовая охрана программ для ЭВМ распространяется только в отношении формы их выражения и *«… не распространяется на идеи, концепции, принципы, методы, процессы, системы, способы, решения технических, организационных или иных задач, открытия, факты, языки программирования»* [2, ст.1259, п. 5].

5.2.3 База данных

Под **базой данных** понимается *"... представленная в объективной форме совокупность самостоятельных материалов (статей, расчетов, нормативных актов, судебных решений и иных подобранных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ)"* [2, ст. 1260, п. 2].

Любая база данных включает, как правило, три составные части: содержимое, т. е. хранимую в памяти информацию, программное обеспечение, необходимое для функционирования базы данных, а также другие электронные вспомогательные материалы (тезаурус, указатели, систему запросов).

Содержимое базы данных может быть представлено материалами - данными, как являющимися объектами авторского права, так и не являющимися таковыми, а также теми и другими. Такие данные могут использоваться независимо от самой базы данных, и любые другие лица могут использовать их для подбора и организации иных баз данных. Однако, если эти данные охраняются авторским правом, то необходимо получить согласие автора или иного правообладателя на включение этих данных во вновь создаваемую базу.

С точки зрения организации структуры, база данных - это совокупность средств и методов описания, хранения и манипулирования данными, позволяющих производить сбор, накопление и обработку информационных массивов. Организация различных баз данных отличается видом объектов данных (числовые, текстовые данные, графические изображения и т. п.) и отношений между ними. База данных может строиться посредством указаний между объектами данных в виде совокупности записей, каждая из которых может включать ссылки как на несколько предыдущих записей, так и на несколько последующих (подчиненных) записей. Записи могут представлять собой как объекты данных, так и связи между объектами данных.

Таким образом, база данных состоит из содержания и процесса упорядочения этого содержания. Правовая охрана распространяется только на оригинальный творческий подбор и упорядочение информации (структуру базы данных), что позволяет говорить об охране базы как сборника произведений, а также на вспомогательные материалы,

необходимые для функционирования базы данных, которые не затрагивают содержимого этой базы.

5.2.4 Авторское право на программу для ЭВМ и базу данных

Предпосылкой охраноспособности программы для ЭВМ и базы данных является их творческий характер, т. е. они должны быть продуктом личного творчества автора. Творческий характер деятельности автора предполагается до тех пор, пока не доказано обратное [2, ст. 1257].

Момент возникновения авторского права является важнейшим юридическим фактом, который устанавливается в силу создания произведения (программы для ЭВМ или базы данных). *"Для возникновения, осуществления и защиты авторских прав не требуется регистрация произведения или соблюдение каких-либо иных формальностей"* [2, ст.1259, п.4].

Часто возникает вопрос: насколько необходимо для возникновения прав на программу для ЭВМ или базу данных их обнародование? Закон устанавливает, что это не является обязательным условием: *"Авторские права распространяется как на обнародованные, так и на необнародованные произведения, выраженные в какой-либо объективной форме …"* [2, ст. 1259, п. 3].

Таким образом, только сам факт создания программы или базы данных, зафиксированных в объективной форме, является основанием возникновения авторского права на эти объекты. С этого момента права автора или иного правообладателя защищаются законом.

Права в отношении программ для ЭВМ и баз данных подразделяются на **личные неимущественные** и **исключительные права.**

Личные права включают **право авторства**, **право на имя** и **право на неприкосновенность** (целостность), **право на обнародование** программы для ЭВМ или базы данных [2, ст. 1265-1268]. Они связаны непосредственно с автором программы для ЭВМ или базы данных: принадлежат лицу, чьим творческим трудом созданы программа для ЭВМ или база данных - автору, являются неотчуждаемыми, т. е. не могут быть переуступлены другому лицу, и не ограничены каким-либо сроком [2, ст. 1228].

Исключительные права непосредственно связаны с понятием **"использования"** программ для ЭВМ и баз данных: *"Автору произведения или иному правообладателю принадлежит исключительное право использовать произведение … в любой форме и любым не противоречащим закону способом …"* [2, ст. 1270, п.1]. При этом под использованием понимается осуществление определенных действий с программами для

ЭВМ или базами данных, а именно: опубликование (выпуск в свет); воспроизведение (полное или частичное) в любой форме, любыми способами; распространение; модификацию и иное использование [2, ст. 1270, п.2]. Они могут принадлежать автору или иному правообладателю (гражданину или юридическому лицу). Правообладатель может распоряжаться исключительным правом на произведение [2, ст. 1270, п.1], в том числе по своему усмотрению разрешать или запрещать другим лицам использование [2, ст. 1229, п.1]. Распоряжение принадлежащим правообладателю исключительным правом может осуществляться любым, не противоречащим закону и существу такого исключительного права способом, в том числе путем его отчуждения по договору другому лицу или предоставления другому лицу права использования [2, ст. 1233, п.1]. Срок действия исключительного права ограничен

Каждая составляющая понятия использования программы для ЭВМ или базы данных имеет конкретное содержание, которое также определено законом:

- **воспроизведение** - *"... изготовление одного или более экземпляров произведения или его части в любой материальной форме, … в том числе запись в память ЭВМ"* [2, ст. 1270, п. 2, п.п.1];
- **распространение** – предоставление доступа к произведению *"... путем продажи или иного отчуждения его оригинала или экземпляров"* [2, ст. 1270, п. 2, п.п.2].
- **публичный показ (выпуск в свет) -** *"... любая демонстрация оригинала или экземпляров произведения непосредственно … … либо с помощью технических средств в месте, открытом для свободного посещения, или в месте, где присутствует значительное число лиц … "* [2, ст. 1270, п. 2, п.п.3];

Обращает на себя внимание то, что понятие "использование" не связано с функционированием программы (или ее выполнением) с целью получения результата. Последнее лучше относить к понятию "потребление" или "пользование" программы. Поэтому каждый раз, когда пользователь запускает программу для того, чтобы произвести расчеты, построить графики или таблицы и т. п., он не "использует" (поскольку не создается новый экземпляр), а "потребляет" программу, не нарушая при этом ничьих прав.

В целях **оповещения о своих правах** правообладатель *"... вправе использовать знак охраны авторского права, который помещается на каждом экземпляре произведения и состоит из следующих элементов: латинской буквы С в окружности; имени или наименования правообладателя; года первого опубликования произведения"* [2, ст. 1271]. Знак охраны авторского права может проставляться на упаковке, на самом программном продукте, а также на всех сопроводительных и дополнительных материалах,

однако это не является обязательным. Следует иметь в виду, что сам знак ничего не защищает (защищает закон!), и что наличие или отсутствие знака охраны никак не связано с возникновением авторского права. Знак охраны авторского права - это цивилизованная форма предупреждения желающих использовать данный объект, что права на него охраняются законом, с одновременным сообщением о том, кому эти права принадлежат.

Исключительные права на программу или базу данных переходят по наследству в установленном законом порядке, и их можно реализовать в течение срока действия авторского права.

Права на программу для ЭВМ или базу данных не связаны с **правом собственности на материальный носитель**, на котором они зафиксированы. Передача прав на материальный носитель не влечет за собой передачи каких-либо прав на программу для ЭВМ или базу данных [2, ст.1227]. Иными словами, передача носителя информации (например, дискеты) с зафиксированной на нем программой третьему лицу не означает передачи каких-либо прав на эту программу.

5.2.3 Правообладание

Если человек (или группа людей) самостоятельно, по личной инициативе создал программу для ЭВМ или базу данных, то он является одновременно и автором, и правообладателем созданного произведения, что позволяет ему по собственному усмотрению использовать эту программу (или базу данных) в личных целях, продавать, раздавать бесплатно, разрешать тиражировать и распространять или иным образом распоряжаться своими исключительными правами.

Если права на программу для ЭВМ или базу данных принадлежат одновременно нескольким авторам, то их действия по использованию произведения и распоряжению исключительными правами на произведение лучше всего урегулировать в специальном договоре между всеми авторами.

Если программа для ЭВМ или база данных разрабатываются по заказу, то заказчик заключает с разработчиком договор на создание данного объекта. В этом случае, *"… когда программа для ЭВМ или база данных создана по договору, предметом которого было ее создание (по заказу), исключительное право на такую программу или такую базу данных принадлежит заказчику, если договором между подрядчиком (исполнителем) и заказчиком не предусмотрено иное"* [2, ст.1296, п.1]. Если программа или база данных создана по договорам, *"…которые прямо не предусматривали ее создание, исключительное право на такую программу или базу данных принадлежит подрядчику (исполнителю), если договором между ним и заказчиком не предусмотрено иное"*. В этом случае заказчик сохраняет за собой право использования программы или базы данных, но

возможность распоряжения исключительными правами на такие программы или базы должна быть закреплена в договоре между заказчиком и исполнителем[2, ст.1297, п.1]. В обоих случаях, когда автору созданных по заказу (или иному договору) программы для ЭВМ или базы данных не принадлежит исключительное право, он имеет право на вознаграждение [2, ст. 1296, п.4, ст.1297, п.3].

Если автор (работник) связан с каким-либо предприятием (работодателем) трудовыми отношениями и создал программу или базу данных в пределах установленных для него трудовых обязанностей, то в этом случае исключительное право на созданную программу для ЭВМ или базу данных принадлежит работодателю, если трудовым или иным договором между работодателем и автором не предусмотрено иное, а такая программа или база данных называется служебным произведением [2, ст. 1295, п. 1, 2]. Работодатель в данном случае правомочен распоряжаться программой для ЭВМ или базой данных по своему усмотрению. За автором сохраняются только личные права, и автор не имеет права распоряжаться исключительными правами и предоставлять для использования программу для ЭВМ или базу данных другим физическим или юридическим лицам. Однако, если предварительно между автором и работодателем был заключен договор в отношении использования созданной программы для ЭВМ или базы данных и распоряжения исключительными правами на них, то руководствуются условиями договора.

Кроме личных (неимущественных) прав автору “служебной” программы для ЭВМ (базы данных) принадлежит право на вознаграждение при условии использования работодателем созданных произведений или передачи исключительного права другому лицу. Размер и порядок выплаты этого определяется договором [2, ст. 1295, п. 2,]. К сожалению, такие договоры заключаются редко, и напрасно, так как только при заключении договора автор наиболее эффективно может реализовать свое право на вознаграждение за каждый вид использования созданной им программы для ЭВМ или базы данных. Для работодателя такой договор тоже целесообразен, так как наличие в нем условия о выплате вознаграждения автору позволяет дополнительно стимулировать разработчика к созданию конкурентоспособных программы или базы.

5.2.6 Передача исключительных прав на программу для ЭВМ и БД

Исключительные права на программу для ЭВМ и базу данных могут быть переданы в полном объеме (отчуждение исключительных прав) другим физическим или юридическим лицам. Передача исключительных прав должна быть оформлена

заключением договора в письменной форме, который называется **“Договор об отчуждении исключительного права”** [2, ст.1234].

При полной уступке исключительных прав автором или иным правообладателем правоприобретатель (физическое или юридическое лицо) получает возможность осуществлять все эти права в полном объеме: использовать программу для ЭВМ или базу данных самостоятельно, например, путем изготовления и распространения экземпляров программ или баз данных, разрешать и запрещать третьим лицам их использование, передать все права или отказаться от них вовсе, - в то время как прежний правообладатель лишается этой возможности.

Обладатель исключительного права на программу для ЭВМ или базу данных может предоставить другому физическому или юридическому лицу право использования соответствующих объектов по **лицензионному договору** [2, ст.1235, п.1]. В лицензионном договоре должны быть определены следующие основные условия: способы использования объекта, порядок выплаты вознаграждения и срок действия договора, а также территория, на которой используется данная программа для ЭВМ или база данных [2, ст. 1235, п.3-6]. Законом предусмотрены лицензионные договоры двух видов: **исключительная** лицензия и **неисключительная** (простая) лицензия. В случае исключительной лицензии передаются права на монопольное использование объекта лицензии одному лицу без сохранения за лицензиаром права выдачи лицензий другим лицам. В случае неисключительной лицензии передаются права на использование объекта лицензии в пределах, обусловленных договором, при этом правообладатель оставляет такие же права за собой и может выдавать неисключительные лицензии на тот же объект на тех же или иных условиях другим лицам [2, ст.1236].

5.2.7 Нарушение прав на программу для ЭВМ и базу данных

Специфика программ для ЭВМ и баз данных такова, что они очень уязвимы в смысле их незаконного использования (прежде всего, путем копирования и распространения копий). Незаконно изготовленные (скопированные) или используемые экземпляры программы для ЭВМ или базы данных называются **контрафактными**, а несанкционированное использование чужих программ или баз данных путем опубликования (выпуска в свет), воспроизведения (полного или частичного), распространения, иного использования считается нарушением исключительных прав на программы для ЭВМ или базы данных, т. е. **нарушением авторского права**.

5.3 Официальная регистрация программ для ЭВМ и баз данных

5.3.1 Право на официальную регистрацию

В ст. 1262 ГК РФ Ч.4 [2] закреплено право автора или иного правообладателя на государственную регистрацию программы для ЭВМ или базы данных: *"Правообладатель в течение срока действия исключительного права на программу для ЭВМ или базу данных может по своему желанию зарегистрировать такую программу для ЭВМ или такую базу данных в федеральном органе исполнительной власти по интеллектуальной собственности "*. Исключение составляют программы для ЭВМ и базы данных, в которых содержатся сведения, составляющие государственную тайну.

Предусмотренная регистрация не является правообразующей и носит факультативный характер, т. е. с ней не связано возникновение прав на программу для ЭВМ или базу данных, однако такая процедура представляется полезной по следующим соображениям.

1. Она является **официальным уведомлением общественности** о наличии у правообладателей прав в отношении рассматриваемых объектов. Подобное уведомление обеспечивается публикацией соответствующих сведений в Официальном бюллетене федерального органа исполнительной власти по интеллектуальной собственности.

2. Государственная регистрация **содействует защите прав** в случае возникновения конфликтных ситуаций при нарушении прав или установлении приоритета. При этом депонированные в федеральном органе материалы могут рассматриваться судом при сборе доказательств в качестве свидетельства наличия соответствующих прав у лица, подавшего заявку на регистрацию сведений.

3. Как свидетельствует практика, факт регистрации программы для ЭВМ или базы данных является для соответствующих органов достаточным основанием для разрешения вывоза данных объектов с территории России. Кроме того, налоговые органы и некоторые страховые компании, занимающиеся страхованием сделок, связанных с объектами интеллектуальной собственности, требуют от своих клиентов предварительной регистрации программ для ЭВМ и баз данных. Официальные документы о регистрации обеспечивают также документальное подтверждение наличия и использования таких объектов для бухгалтерского учета при продаже-покупке лицензий, выплате авторских вознаграждений.

5.3.2 Процедура официальной регистрации

Процедура официальной регистрации программ для ЭВМ и баз данных в целом определена ст. 1262 ГК РФ Ч.4 и включает подачу заявки в федеральный орган

исполнительной власти по интеллектуальной собственности (Роспатент), проверку поданных документов и собственно регистрацию. После поступления заявки на регистрацию в Роспатент проверяется наличие необходимых документов и их соответствие установленным требованиям. При положительном результате проверки сведения о программе для ЭВМ или базе данных вносятся, соответственно, в Реестр программ для ЭВМ или Реестр баз данных под уникальным регистрационным номером и выдается заявителю (здесь заявителем называют правообладателя, подавшего заявку на регистрацию программы или базы данных в Роспатент) **свидетельство** о государственной регистрации установленной формы, в котором указаны регистрационный номер объекта по Реестру, название программы или базы данных, имя или наименование правообладателя, фамилии авторов и дата регистрации. Сведения о зарегистрированных программах для ЭВМ и базах данных публикуются в официальном бюллетене Роспатента.

5.3.3 Заявка на официальную регистрацию

Состав заявки на официальную регистрацию программы для ЭВМ или базы данных (далее - Заявка) определен п. 2 ст. 1262 ГК РФ Ч.1, а также в **Правилах составления, подачи и рассмотрения заявок на официальную регистрацию программ для электронных вычислительных машин и баз данных** (далее - Правила).

Заявка должна относиться к одной программе или одной базе данных. При этом *"Программа для ЭВМ, состоящая из нескольких программ для ЭВМ (программный комплекс), которые не могут быть использованы самостоятельно, регистрируется в целом (без регистрации каждой входящей в нее (него) программы для ЭВМ)"* [3, п. 5]. Заявка должна содержать следующие **документы**:

- заявление о государственной регистрации;
- депонируемые материалы, идентифицирующие программу для ЭВМ или базу данных, включая реферат;
- документ, подтверждающий уплату государственной пошлины в установленном размере или основание для освобождения от уплаты государственной пошлины или уменьшения его размера.

В Правилах подробно описаны **требования**, предъявляемые к документам заявки.

Заявление на официальную регистрацию представляется отпечатанным на типографском бланке или в виде компьютерной распечатки согласно образцам, приведенным в приложениях к Правилам (формы РП и РП/ДОП).

Согласно п. 11 Правил, заполненное заявление (форма РП) должно содержать все предусмотренные в нем сведения, касающиеся регистрируемой программы для ЭВМ или

базы данных, в соответствии с требованиями подстрочника бланка. Если какие-то сведения отсутствуют, то в соответствующей графе заявления должно быть указано, что их "нет" "отсутствуют" или "не имеется".

При заполнении отдельных граф заявления необходимо руководствоваться положениями а) – п) п. 13 Правил, а также следующими дополнительными **рекомендациями**.

- Графа 2 "Основание возникновения прав на данное произведение" не заполняется, если заявка подается от имени автора (физического лица).

- В графе 3 "Название произведения" приводится полное название регистрируемого объекта и после него в круглых скобках - сокращенное название, если таковое имеется.

- При заполнении графы 7 "Дата и место первого выпуска в свет произведения" необходимо руководствоваться определенным в ст. 1270, п. 2, п.п.ГК РФ Ч.4 понятием "публичного показа (опубликования)". Опубликованием, например, может считаться демонстрация программы для ЭВМ или базы данных на конференции, выставке, распространение демонстрационных дискет и рабочих экземпляров программы или базы данных и документации на них и т. п.

- В графе 8 "Сведения о всех произведениях, являющихся объектами авторского права" рекомендуется сообщать о произведениях-аналогах, использованных в качестве прототипа или составной части при разработке заявляемого объекта, а также о программных средствах, библиотеках программ, файлах и т. п., использованных в качестве инструментария при создании регистрируемого объекта, и на использование которых должно быть законное право (например, лицензия, номер которой указывается).

- В разделе "Краткое описание авторского вклада в данное произведение" графы 9А рекомендуется указывать сведения о личном творческом вкладе автора в разработку заявляемого на регистрацию объекта (например, программирование части программы, программного модуля или блока, разработка структуры записи файла или файлов, входящих в состав базы данных, и т. п.). Не может считаться творческим вкладом в создание программы для ЭВМ или базы данных "разработка алгоритма".

- При заполнении раздела "документ о регистрационном сборе..." графы 10 следует принимать во внимание пункт 1 п.п.14) ст. 333.35 НК РФ, предусматривающий освобождение от уплаты государственной пошлины в случае, если *"физическое лицо – гражданин РФ, являющийся единственным автором*

программы для ЭВМ, базы данных.. и правообладателем на нее, испрашивающим свидетельство о регистрации на свое имя, … является … .учащимся (воспитанником) образовательных учреждений (независимо от их форм собственности) – за совершение действий, предусмотренных пунктами 4-7 статьи 333.30 …". Согласно пункту 2 ст.333.35 Налогового кодекса РФ указанная льгота предоставляется по ходатайству автора. Основанием для предоставления льготы является копия документа, выданного образовательным учреждением.

- В графе 11 "Сведения об адресате" указываются имя и адрес лица, которому следует отправлять корреспонденцию по данной заявке. Это может быть один из авторов, руководитель подразделения, в том числе специального, доверенное лицо.
- В последней графе приводится подпись заявителя. Если заявителем является физическое лицо (автор или его правопреемник), то указываются фамилия, имя и отчество заявителя без указания должности. При этом подпись не требуется заверять у нотариуса. Это справедливо и для случая, когда заявителями являются два и более физических лиц.
- При заполнении заявления все адреса следует указывать с индексами. Рекомендуется также указывать сведения без сокращений (например, "Санкт-Петербург", а не "СПб", "Российская Федерация", а не "РФ" и др.).

Примеры заполнения заявлений на официальную регистрацию программы для ЭВМ от имени автора и базы данных от имени юридического лица приведены в прил. 2 и 6 соответственно.

Депонируемые материалы, представляемые на регистрацию, должны обеспечивать однозначную идентификацию регистрируемой программы или базы данных, включая реферат.

При оформлении депонируемых материалов следует руководствоваться п.п. 14 - 23 Правил.

Материалы, идентифицирующие программу для ЭВМ, представляются в одном экземпляре, как правило, в виде распечатки исходного текста (полного или фрагментов) в объеме до 70 страниц. Представление депонируемых материалов в иной форме допускается при наличии обоснования заявителя о том, что данная форма в большей степени обеспечивает идентификацию регистрируемой программы для ЭВМ. Допускается включать в состав этих материалов подготовительные материалы, полученные в ходе ее разработки, а также порождаемые ею аудиовизуальные отображения в любой визуально воспринимаемой форме (распечатки, фотографии, рисунки и т. п.). Например, если регистрируемая программа для ЭВМ включает охраноспособные по нормам авторского

права изображения на экране дисплея или музыкальные произведения, правообладателем которых является лицо, обладающее правом на саму программу, эти материалы могут быть задепонированы вместе с листингом порождающей их программы.

В целях идентификации регистрируемой базы данных следует представлять материалы (в одном экземпляре), отражающие объективную форму представления и организации совокупности содержащихся в ней данных и принципы их систематизации, позволяющие нахождение и обработку этих данных с помощью ЭВМ в объеме до 50 страниц. Материалы могут включать перечень файлов, составляющих базу данных, описание связей между этими файлами, а также описание структур записей файлов с указанием наименования, типа, размера и содержания полей, представленное в виде таблиц.

Если объем депонируемых материалов позволяет, то в них рекомендуется (хотя и не обязательно) включать описание **состава** регистрируемого объекта:

- для программы - список программных модулей и файлов, составляющих программу, с указанием их назначения (это может быть один программный модуль или файл). Пример описания состава программного комплекса приведен в прил. 4;
- для базы данных - список файлов, составляющих базу данных, с указанием их содержания (это может быть один файл). Пример описания состава базы данных приведен в прил. 8.

Примечание: *При выполнении учебного задания включение состава регистрируемого объекта (программы для ЭВМ или базы данных)* ***обязательно****, поскольку в этом случае указанный документ, как правило, является заменяющим исходный текст программы либо описание структуры базы данных.*

В объем депонируемых материалов включается **титульный лист**, при оформлении которого следует руководствоваться п. 18 Правил, а также ГОСТ 19103-77 и ГОСТ 19104-78. На титульном листе должна быть представлена следующая информация:

- полное имя или официальное фирменное наименование правообладателя (заявителя) или правообладателей (заявителей);
- фамилии, имена, отчества авторов;
- название программы для ЭВМ или базы данных, так как оно указано в Заявлении;
- сведения о полноте представления идентифицирующих материалов (полный исходный текст, фрагменты исходного текста, фрагменты исходного текста с

исключением конфиденциальных частей, описание структуры базы данных в полном объеме, фрагменты описания структуры базы данных и т. п.);

- объем документа (количество листов, включая титульный);
- знак охраны авторского права, состоящий из упомянутых ранее трех элементов [2, ст. 1271], если программа для ЭВМ или база данных была выпущена в свет к моменту подачи заявки на официальную регистрацию;
- год подачи заявки на официальную регистрацию в Роспатент.

Примеры оформления титульных листов к депонируемым материалам, представляемым на регистрацию программы для ЭВМ от имени автора и базы данных от имени юридического лица, приведены в прил. 3 и 7 соответственно.

В состав депонируемых материалов входит также **реферат**, который представляется в двух экземплярах отдельно от листинга программы для ЭВМ или описания структуры базы данных и не входит в их объем. Реферат должен содержать информацию, определенную в п.п. 18а) - 18и), п.21 и п.23 Правил, в полном объеме. При этом:

- аннотация реферата должна содержать сведения, определенные п. 18г) Правил;
- объем памяти указывается в Кбайтах или Мбайтах и определяется для программ как объем памяти, занимаемый исходным текстом программы (листингом), а для баз данных - как объем памяти, необходимый для хранения информации об одном объекте базы данных, если невозможно указать ее полный объем.

Следует обратить внимание на соблюдение объема текста аннотации- до 700 печатных знаков. Примеры рефератов к регистрируемым программе для ЭВМ и базе данных приведены в прил. 5 и 9 соответственно.

Документ, подтверждающий уплату государственной пошлины в установленном размере, а также основание для уменьшения его размера, представляется согласно ст.333.30 Налогового кодекса РФ.

Сведения, указываемые во всех документах заявки на официальную регистрацию, должны быть терминологически единообразными и не должны допускать разночтений.

5.4 Особенности коммерческой реализации программ для ЭВМ и баз данных

5.4.1 Программный продукт и формы его продажи

В настоящее время на рынке интеллектуального товара стремительно растет количество разнообразного программного обеспечения. Компьютерные программы стали товаром, приносящим немалую прибыль. Чтобы успешно продвигать товар на рынок,

фирмы-производители затрачивают значительные средства на рекламу и разработку сервисных услуг, сопровождающих эксплуатацию их продукции и заключающихся в бесплатном техническом обслуживании, поставке по льготным ценам новых версий и т. п. Увеличение массовых продаж способствовало появлению специфической формы предложения программ для ЭВМ и баз данных как товара, получившей название **программного продукта**.

Программный продукт - персонифицированная программа для ЭВМ или база данных, которая предназначена для самостоятельного использования конкретным пользователем в личных целях. Программный продукт включает не только исполняемый модуль и набор файлов, обеспечивающих функционирование программы или базы данных, но и содержит целый ряд вспомогательных средств, которые фирма-производитель предоставляет пользователю для самостоятельной установки и обслуживания программного продукта на своей ЭВМ с максимальной адаптацией к конфигурации системы и файловой структуре (программы-инсталляторы и т. п.), а также самостоятельного освоения программного продукта (руководство пользователя в автоматизированном виде и на бумажном носителе и т. п.). Программный продукт оформляется как запечатанный в упаковку комплект дискет или оптических дисков, на которых записаны основные и вспомогательные программные средства и файлы, вместе с необходимой сопроводительной документацией.

Коммерческая реализация (продажа) программного продукта связана с понятием использования программы для ЭВМ или базы данных третьими лицами (пользователями) и осуществляется на основании лицензионного договора с правообладателем. Договор заключается в письменном виде и может определять следующие условия: способы использования, порядок выплаты вознаграждения и срок действия договора, а также территорию, на которой используется данный продукт [2, ст. 1235, 1236].

Одним из типов лицензионного договора на программу для ЭВМ или базу данных является традиционный двухсторонний договор правообладателя – **лицензиара**, с покупателем (пользователем) - **лицензиатом**, в котором определяется способы, сроки, территория использования программы или базы данных. Такие договоры составляются, как правило, при единичных продажах программного продукта, предназначенного для решения достаточно узких прикладных задач (научных, отраслевых и т. п.), при продажах программного продукта, требующего регулярного обновления и дополнения (некоторые базы данных), а также при передаче прав на тиражирование и распространение программ для ЭВМ или баз данных.

Однако для массовых продаж программных продуктов, предназначенных для использования в личных целях (редакторов текстов, электронных таблиц, компьютерных игр и др.), такой тип лицензионного договора неудобен. В этом случае применяется особый тип договора - одностороннее оформление договора на продажу и предоставление массовым пользователям доступа к программному продукту. Такой тип договора известен под названием **этикеточной (или оберточной) лицензии**, или договора на использование программы для ЭВМ (базы данных) с конечным пользователем. Осуществляется такая продажа путем изложения типовых условий лицензионного договора на продаваемых экземплярах программного продукта. Пользователь знакомится с условиями договора, которые обычно помещаются на упаковке, вкладыше или дискете, входящих в комплект программного продукта, и в случае согласия с ними приобретает экземпляр программы для ЭВМ или базы данных. Факт вскрытия упаковки или начало эксплуатации программы свидетельствует о том, что пользователь согласился с условиями такого договора, и приравнивается к его подписанию со стороны пользователя.

Часто вместе с этикеточной лицензией в комплект программного продукта вкладывается регистрационный талон (уведомление), который пользователь может отослать на фирму-производитель, сообщив сведения о себе. В этом случае он будет зарегистрирован на фирме как официальный пользователь и может рассчитывать на получение гарантий и услуг, указанных в этикеточной лицензии, в полном объеме. Регистрационный номер пользователя будет рассматриваться как номер лицензионного соглашения.

Особенностью этого типа договора по сравнению с традиционными двухсторонними договорами является то, что в этом случае пользователь приобретает право использования программы для ЭВМ или базы данных только **в личных целях**. Иные способы использования не предусматриваются.

Возможность применения такого способа распоряжения исключительным правом на программу для ЭВМ или базу данных массового применения закреплена в ГК РФ Ч.4 ст. 1300 «Информация об авторском праве»

5.4.2 Договор на использование программы для ЭВМ и базы данных

Текст договора должен содержать определенную исчерпывающую формулировку лицензионного соглашения между владельцем прав на программу для ЭВМ или базу данных (далее - объект договора) и покупателем (приобретателем прав на использование объекта договора). При составлении договора целесообразно придерживаться

определенной структуры, чтобы можно было проверить, содержит ли лицензионный договор все существенные условия.

Структура договора предусматривает следующие разделы:

- Стороны договора.
- Преамбула.
- Термины и их определения.
- Предмет договора.
- Техническая документация.
- Техническая помощь.
- Усовершенствования и улучшения.
- Обязательства и ответственность сторон.
- Платежи.
- Информация и отчетность.
- Обеспечение конфиденциальности.
- Защита передаваемых прав.
- Реклама.
- Разрешение споров.
- Срок действия договора.
- Иные условия.
- Заключительные положения.
- Адреса сторон.

При составлении конкретного договора стороны могут, ориентируясь на условия договора и особенности объекта договора, объединять или разделять разделы, исключать их, изменять названия разделов и т. п.

Стороны договора. В этом разделе должны быть определены договаривающиеся стороны. Для юридических лиц указываются их полные фирменные наименования, а для физических лиц (граждан) - их фамилии, имена, отчества, адреса. При этом владелец прав на объект договора именуется "Лицензиар", а приобретатель лицензии на использование объекта договора - "Лицензиат".

Преамбула. В преамбуле кратко излагаются правовое положение договаривающихся сторон и их намерения в отношении объекта договора:

- Лицензиар сообщает о своих правах на объект договора, при этом указываются номера официальных документов (если они имеются), подтверждающих эти правомочия;

- Лицензиат сообщает о своих намерениях в отношении использования объекта договора (например, применение, тиражирование, распространение или продажа, иное введение в хозяйственный оборот), а также указывает специальную область, в которой будет осуществляться использование объекта договора.

Термины и их определения. Для удобства изложения условий договора рекомендуется использовать специальные термины, которые являются краткими выражениями понятий, используемых в тексте договора. Перечень терминов с определением их значений приводится в этом разделе.

Предмет договора. В этом разделе определяется предмет договора - предоставление лицензии, включая указание:

- вида лицензии (исключительная или неисключительная);
- срока, на который предоставляется лицензия;
- способа возмещения стоимости лицензии Лицензиатом (безвозмездно, за вознаграждение, в обмен на другой объект или услуги и т. п.);
- способа использования (применение, тиражирование, распространение и иное введение в хозяйственный оборот) и объема использования (например, количество экземпляров объекта договора, разрешенных к использованию, количество компьютеров, на которые будет установлена программа, количество экземпляров программы, изготавливаемых для последующего распространения Лицензиатом, и т. п.);
- ограничения лицензии (это может быть ограничение территории использования, установление контроля за использованием со стороны Лицензиара и др.);
- возможности передачи прав третьим лицам;
- предоставления льгот.

В этом же разделе могут быть оговорены условия, также относящиеся к предмету договора, но часто выносимые в отдельный раздел “Обеспечение договора”:

- форма передачи объекта договора (например, передача дискет или непосредственная запись в память ЭВМ);
- передача сопровождающей или технической документации к объекту договора (например, инструкции пользователя по применению программы для ЭВМ или базы данных, методические описания и другая документация, касающаяся содержания, функционирования и применения объекта договора);
- условия передачи (в том числе сроки);

- оказание необходимой технической помощи - любой помощи, оказываемой Лицензиаром по сопровождению объектов договора (например, установка программы для ЭВМ или базы данных на компьютеры пользователя, настройка их на конкретные условия, обучение пользователя, консультации по применению, устранение сбоев, ошибок и др.).

При необходимости подробного урегулирования какого-либо из этих вопросов он может быть вынесен в отдельный раздел.

Техническая документация. Если Лицензиар предоставляет Лицензиату техническую и другую сопровождающую документацию, относящуюся к объекту договора, то в этом разделе приводится перечень предоставляемых материалов, оговариваются условия передачи (срок и процедура оформления факта передачи), обязанности Лицензиара по устранению возможных недостатков в документации и права Лицензиата при пользовании документацией (возможность копирования и тиражирования, предоставления третьим лицам и др.).

Техническая помощь. Лицензиар может принять на себя обязательства по оказанию технической и консультационной помощи при эксплуатации объекта договора. При этом, как правило, Лицензиат несет все расходы, связанные с пребыванием специалистов Лицензиара на территории Лицензиата при оказании оговоренной технической помощи (командировочные расходы, вознаграждение за оказанные услуги и т. п.).

Усовершенствования и улучшения. Стороны договариваются о взаимных обязанностях в отношении возможных усовершенствований и улучшений, связанных с функционированием или применением объекта договора (программы для ЭВМ или базы данных) и появившихся в течение срока действия договора.

Обязательства и ответственность сторон. Лицензиар заверяет Лицензиата в том, что на момент подписания договора ему ничего не известно о правах третьих лиц, которые могут быть нарушены предоставлением данной лицензии. Лицензиар также гарантирует техническую работоспособность объекта договора, комплектность технической и сопровождающей документации и т. п.

За невыполнение этих условий, несоблюдение сроков и условий, предусмотренных в других разделах договора, устанавливается ответственность сторон согласно действующему законодательству. В пунктах этого раздела стороны по обоюдному согласию могут назначить и конкретные взыскания за нарушение отдельных условий договора, например наложение договорного штрафа за несоблюдение срока передачи документации.

Здесь же оговариваются условия, при которых стороны освобождаются от ответственности (например, из-за возникновения обстоятельств непреодолимой силы - форс-мажор).

Платежи. Этот раздел включается в договор в случае возмездного предоставления лицензии на объект договора. В нем определяются размер вознаграждения и условия его выплаты.

Вознаграждение может выплачиваться единовременно или поэтапно заранее оговоренными суммами. В зависимости от цели и способов использования объекта договора кроме единовременных или поэтапных платежей могут быть предусмотрены и текущие (периодические) платежи в течение срока действия договора (например, при приобретении прав на тиражирование и распространение программы для ЭВМ или базы данных), выплачиваемые Лицензиару как часть выручки Лицензиата от тиражирования и продажи экземпляров таких программ или дохода от иной хозяйственной деятельности, связанной с использованием объекта лицензии.

Информация и отчетность. Этот раздел появляется чаще всего в том случае, если в договоре предусмотрены текущие платежи в течение срока действия договора. Тогда стороны могут договориться, что Лицензиар вправе потребовать от Лицензиата регулярного предоставления бухгалтерских данных, отчетов и другой информации с целью контроля за правильностью исполнения текущих платежей. Оговариваются сроки предоставления отчетов, а также могут быть конкретизированы те сведения, которые подлежат проверке со стороны Лицензиара.

Обеспечение конфиденциальности. Под конфиденциальностью принято понимать соблюдение мер по предотвращению случайного или преднамеренного разглашения определенных сведений, касающихся объекта, а иногда и предмета договора, третьим лицам.

Лицензиар вправе обязать Лицензиата сохранять конфиденциальность полученных от Лицензиара документации и других сведений (в том числе устных), относящихся к объекту договора и его эксплуатации, и предпринимать все меры по обеспечению конфиденциальности, в частности, ограничивая соответствующими обязательствами лиц, работающих непосредственно с объектом договора. При нарушении обязательств по обеспечению конфиденциальности Лицензиат возмещает Лицензиару понесенные в связи с этим убытки.

Защита передаваемых прав. В этом разделе Лицензиар в целях защиты своих авторских прав на объект договора обязует Лицензиата корректно распоряжаться переданными по договору правами, а именно: признавать действительность прав

Лицензиара на объект договора в соответствии с Законом о ПрЭВМ и БД и Законом об АП; предотвращать несанкционированное использование объекта договора третьими лицами и незамедлительно уведомлять Лицензиара, если такие случаи стали известны Лицензиату; не выдвигать претензий против условий договора и не распоряжаться переданными правами каким-либо иным образом без специального письменного согласия Лицензиара.

Реклама. Вопрос о рекламе в качестве обязательства по договору следует по возможности решать конкретно. Лицензиар может возложить на Лицензиата определенные обязанности по рекламированию программ для ЭВМ или баз данных тем или иным образом, который зависит от области применения и способа использования объекта договора. Например, это может быть в случае приобретения права тиражировать и распространять экземпляры программы.

Разрешение споров. В этом разделе стороны оговаривают порядок разрешения споров по вопросам договора. Как правило, стороны предполагают решать возникшие споры путем переговоров. Для случая неэффективных переговоров может быть оговорен особый порядок разрешения спора, например, путем обращения в третейский суд, но этот путь не должен противоречить действующему законодательству. Особенно важен данный раздел в случае, когда договор заключают представители разных стран. В этом случае оговаривается, законодательство какой страны будет применяться при разрешении споров.

Срок действия договора. В разделе определяются дата (или условие) вступления в силу договора, срок его действия, условия его досрочного расторжения, а также права сторон в отношении объекта договора по истечении срока договора.

Иные условия. Данный раздел содержит условия, не предусмотренные в других разделах договора.

Заключительные положения. В данном разделе, как правило, устанавливают порядок внесения в договор изменений и дополнений, другие пояснения, касающиеся его правового статуса. Здесь же фиксируются место и дата совершения договора.

Адреса сторон. Указываются адреса сторон: для юридических лиц - место нахождения, для физических лиц - место жительства. Дополнительно могут быть указаны банковские реквизиты.

Договор скрепляется подписями сторон. От имени юридического лица договор подписывает его руководитель или лицо, которому предоставлено такое право по уставу или доверенности.

Двухсторонний лицензионный договор всегда должен соответствовать особенностям конкретной сделки, и чем подробнее в его условиях рассмотрены возможные случаи, способные привести в будущем к спорам между сторонами, тем реальнее избежать подобных споров и возможных судебных разбирательств по их поводу.

5.5 Титульный лист

ПРОГРАММА ДЛЯ ЭВМ

Подсистема аудита информационной системы

Исходный текст программы

Всего 10 листов

Правообладатель: Горяйнов Дмитрий Андреевич

Автор: _Горяйнов_Дмитрий_Андревич
(ф., и., о.)

Санкт-Петербург
2013

5.6 Состав регистрируемой программы

qsqlpsql4, qsqlpsqld4, libqsqlpsql4, libqsqlpsqld4	-	драйверы для подключения программы к СУБД;
podsystem	-	Основная часть, выполняющая основные функции программы;
enter	-	Проверка пользователя и пароля;
tab	-	Работа с таблицей настроек для проводимого аудита;
pole	-	Генерация основных инструментов для аудита системы;

5.7 Реферат

Автор: Горяйнов Дмитрий Андреевич

Правообладатель: Горяйнов Дмитрий Андреевич

Программа для ЭВМ: Подсистема аудита информационной системы

Аннотация: Подсистема предназначена для обнаружения действий, нарушающих целостность СУБД и для выявления мошеннически введенных данных и несанкционированных запросов. Это достигается путем создания подсистемы, которая подключается к основной СУБД и записывает отслеживаемые данные в СУБД, к которой осуществлено подключение.

Тип ЭВМ: IBM PC/AT и совместимые с ней

ОС: Windows XP/7

Язык программирования: C++

Объем: 150 Кбайт

5.8 Лицензионный договор

ЛИЦЕНЗИОННЫЙ ДОГОВОР

НА ИСПОЛЬЗОВАНИЕ ПРОГРАММЫ ДЛЯ ЭВМ

Стороны в Договоре:

Гражданин **Горяйнов Дмитрий Андреевич, проживающий по адресу: 188304, г. Гатчина, ул. Колхозная, д. 6, ч/д,** именуемый в дальнейшем "ЛИЦЕНЗИАР", с одной стороны, и

Государственное образовательное учреждение высшего профессионального образования "Санкт-Петербургский государственный электротехнический университет "ЛЭТИ" им. В.И.Ульянова (Ленина)", именуемый в дальнейшем "ЛИЦЕНЗИАТ", в лице **проректора по научной работе Шестопалова М.Ю.**, действующего на основании Доверенности, с другой стороны,

принимая во внимание:

1) что Лицензиар является автором и правообладателем программы для ЭВМ **"Подсистема аудита для информационной системы";**

2) Лицензиат желает получить на условиях настоящего Договора лицензию на использование упомянутой программы для ЭВМ с целью проведения научных исследований в области медицинского приборостроения;

3) Лицензиар готов предоставить Лицензиату такую лицензию,

договорились о следующем.

1. Термины и их определения

1.1. "ПРОГРАММА ДЛЯ ЭВМ (ПрЭВМ)" - программное обеспечение **"Подсистема аудита для информационной системы"**.

1.2. "ДОКУМЕНТАЦИЯ" - комплект документов, передаваемых Лицензиаром Лицензиату, включающий руководство пользователя по применению и обслуживанию программы для ЭВМ.

1.3. "ПРОИЗВОДСТВЕННАЯ ПЛОЩАДКА" - **научные лаборатории и кафедры** Лицензиата.

1.4. "РАБОЧЕЕ МЕСТО" - конкретная ЭВМ, на которой используется Программа для ЭВМ.

2. Предмет Договора

2.1. Лицензиар предоставляет Лицензиату на срок действия настоящего Договора и за вознаграждение, уплачиваемое Лицензиатом, неисключительную лицензию на

использование ПрЭВМ. При этом Лицензиату предоставляется право на установку ПрЭВМ не более чем на **10 (десяти**) Рабочих местах.

2.2. Лицензиар передает Лицензиату Документацию к ПрЭВМ.

2.3. Предоставленное Лицензиату в рамках настоящего Договора право ограничено Производственной площадкой.

2.4. Лицензиар осуществляет авторский контроль за соблюдением объемов использования ПрЭВМ по настоящему Договору, при этом Лицензиат обеспечивает возможность такого контроля.

2.5. Лицензиар сохраняет за собой право самому использовать ПрЭВМ и предоставлять неисключительные лицензии на право ее использования третьим лицам.

3. Обеспечение Договора

3.1. Лицензиар передает Лицензиату ПрЭВМ в объеме и виде, достаточном для ее использования, и Документацию в течение 15 (пятнадцати) дней со дня подписания настоящего Договора. ПрЭВМ передается Лицензиату в виде в количестве 5 (пяти) штук, содержащих ПрЭВМ. По факту передачи ПрЭВМ и Документации составляется акт сдачи-приемки с перечнем переданных материалов, подписываемый обеими Сторонами.

3.2. Если Лицензиат установит неполноту или неправильность полученных ПрЭВМ или Документации, то Лицензиар в течение 15 (пятнадцати) дней после сообщения ему об этом Лицензиатом обязан передать недостающие материалы или устранить недостатки ранее переданных ПрЭВМ и Документации.

3.3. Для оказания помощи в освоении ПрЭВМ Лицензиар по просьбе Лицензиата оказывает консультации пользователям ПрЭВМ.

3.4. Для целей использования ПрЭВМ в объеме, предусмотренном п. 2.1 настоящего Договора, Лицензиат может изготавливать в необходимом ему количестве копии ПрЭВМ и копии Документации.

4. Усовершенствования

4.1. Лицензиар обязуется незамедлительно информировать Лицензиата о всех произведенных им усовершенствованиях ПрЭВМ и, при желании Лицензиата, передать ему в согласованные сроки новые варианты ПрЭВМ. В отношении новых вариантов ПрЭВМ, переданных Лицензиаром Лицензиату, распространяются все условия настоящего Договора.

4.2. Лицензиат обязуется предоставлять Лицензиару информацию об использовании ПрЭВМ, которая могла бы быть полезной для усовершенствования ПрЭВМ.

5. Платежи

5.1. За предоставление прав, предусмотренных настоящим Договором, Лицензиат выплачивает Лицензиару единовременное вознаграждение в размере 20000(двадцать тысяч) рублей.

5.2. Вознаграждение, предусмотренное п. 5.1 настоящего Договора, выплачивается Лицензиатом в течение 30 (тридцати) дней, следующих после подписания акта приемки-сдачи.

6. Реклама

6.1. Лицензиат обязуется при опубликовании результатов исследований, полученных с использованием ПрЭВМ, сообщать в рекламных целях, что исследования производились с использованием ПрЭВМ Лицензиара с указанием авторского права Лицензиара.

7. Защита передаваемых прав

7.1. Лицензиат обязуется не вносить самовольно каких-либо изменений в ПрЭВМ и Документацию и не дополнять их какими-либо комментариями. Подобные изменения или дополнения возможны только с согласия Лицензиара.

7.2. Лицензиат обязуется предпринимать все необходимые меры для предотвращения несанкционированного копирования ПрЭВМ и Документации третьими лицами, а также несанкционированной передачи ПрЭВМ и Документации работниками Лицензиата третьим лицам.

7.3. Если Лицензиату станет известно о противоправном использовании ПрЭВМ третьими лицами, то он незамедлительно сообщит об этом Лицензиару.

8. Ответственность Сторон и разрешение споров

8.1. За невыполнение или ненадлежащее выполнение обязательств по настоящему Договору Стороны несут имущественную ответственность в соответствии с действующим законодательством.

8.2. Стороны освобождаются от ответственности за неисполнение или ненадлежащее исполнение обязательств, принятых по настоящему Договору, если неисполнение явилось следствием обстоятельств непреодолимой силы (форс-мажор).

8.3. Сторона, нарушившая свои обязательства по настоящему Договору, освобождается от ответственности за неисполнение или ненадлежащее исполнение этих обязательств, если это нарушение было вызвано причинами, за которые отвечает другая Сторона.

8.4. В случае возникновения споров между Лицензиаром и Лицензиатом по вопросам, предусмотренным настоящим Договором, Стороны примут все меры к разрешению их путем переговоров между собой. В случае невозможности разрешения указанных споров путем переговоров они будут разрешаться в порядке, предусмотренном действующим законодательством.

9. Срок действия Договора и условия его расторжения

9.1. Настоящий Договор заключен на срок 2 года и вступает в силу с даты его подписания обеими Сторонами.

9.2. По истечении срока действия настоящего Договора Лицензиат вправе использовать ПрЭВМ, включая усовершенствованные варианты, на Производственной площадке на любом количестве Рабочих мест. При этом обязательства Лицензиата, предусмотренные пп. 7.1 и 7.2 настоящего Договора, сохраняются бессрочно.

9.3. Действие настоящего Договора по обоюдному согласию Сторон может быть досрочно прекращено, но не ранее чем через три месяца после предложения об этом одной из Сторон. При этом Лицензиат не освобождается от обязательств по платежам, возникшим до расторжения настоящего Договора.

9.4. Настоящий Договор может быть досрочно расторгнут в одностороннем порядке со стороны Лицензиара из-за невыполнения Лицензиатом своих обязательств по пп. 7.1 или 7.2. В этом случае Лицензиат лишается права дальнейшего использования ПрЭВМ в любой форме и обязан вернуть ее Лицензиару.

9.5. Если Лицензиат откажется от дальнейшего использования ПрЭВМ, то он уничтожит все имеющиеся у него копии ПрЭВМ.

10. Заключительные положения

10.1. Все изменения и дополнения к настоящему Договору действительны только в тех случаях, если они совершены в письменной форме и подписаны обеими Сторонами.

10.2. Стороны не имеют права передавать свои права и обязательства по настоящему Договору третьим лицам без письменного согласия на то другой Стороны.

10.3. Во всем остальном, что не предусмотрено условиями настоящего Договора, будут применяться нормы законодательства Российской Федерации.

11. Адреса Сторон

11.1. ЛИЦЕНЗИАР: **Горяйнов Дмитрий Андреевич, адрес: 188304, г. Гатчина, ул. Колхозная, д. 6, ч/д.**

11.2. ЛИЦЕНЗИАТ: **СПбГЭТУ, адрес: 197376, Санкт-Петербург, ул. Проф. Попова, д. 5.**

Настоящий Договор составлен в двух экземплярах для каждой из Сторон и подписан "___"_____ 2013 г. в г. Санкт-Петербурге.

ЛИЦЕНЗИАР:	**От ЛИЦЕНЗИАТА:**
	Проректор по научной работе СПбГЭТУ
_____ Д. А. Горяйнов	_____ В.М.Кутузов

5.9 Заявление на государственную регистрацию

6. Выводы

Целью работы являлось проанализировать информационную безопасность информационных систем, аудит информационной системы, а также модели и средства аудита, создав подсистему.

В данном дипломном проекте стояла задача разработки подсистемы аудита. Данная задача была реализована путем создания подсистемы аудита, которая позволит обнаружить действия, нарушающие целостность основной систем, другими словами, с помощью данной подсистемы можно выявить как мошеннически введенные данные, так и несанкционированные запросы.

В итоге в дипломном проекте решены задачи: анализ информационной безопасности систем, аудита информационной безопасности и разработка программного решения аудита информационной безопасности информационной системы, использующей СУБД PostgreSQL.

Решение задачи было осуществлено при помощи системы управления базами данных PostgreSQL, с помощью SQL запросов, которые были описаны, и инструментария QT. Проведение аудита при помощи дипломного проекта осуществляется должным образом, что в свою очередь поможет повысить информационную безопасность любой рассматриваемой системы и базы данных в этой системе.

В соответствующих главах диплома была проведена экономическая оценка, а также приведена важная информация, касающаяся охраны интеллектуальной собственности.

7. Список литературы

1) Дейв Энсор, Йен Стивенсон «Oracle. Проектирование баз данных.», Издательство «BHV», 1999
2) Пит Финниган
   http://www.securityfocus.com/infocus/1689,
   Oracle Magazine RE ,Январь/Февраль 2004
3) Библиотека MSDN. Подсистема аудита SQL Server (Database Engine). SQL Server 2012.
   http://msdn.microsoft.com/ru-ru/library/cc280386.aspx
   http://msdn.microsoft.com/ru-ru/library/cc280424.aspx
4) Библиотека MSDN. Отслеживание изменений в SQL Server 2008.
   http://habrahabr.ru/post/111207/
   http://blogs.msdn.com/b/alexejs/archive/2009/08/09/change-tracking.aspx
   http://blogs.msdn.com/b/alexejs/archive/2009/08/07/cdc.aspx
5) Статья all-oracle.ru. Пользователи в Oracle: Управление привилегиями.
   http://all-oracle.ru/content/view/?part=1\&id=83
6) Т.С. Карпова «Базы данных: модели, разработка, реализация.»
   http://www.intuit.ru/department/database/dbmdi/13/
7) Т.С. Карпова «Базы данных: модели, разработка, реализация.»
   http://www.intuit.ru/department/database/dbmdi/11
8) И.Ю. Баженова «SQL и процедурно-ориентированные языки.»
   http://www.intuit.ru/department/database/cdba/8
9) Л.Н. Полякова «Основы SQL.»
   http://www.uchi-it.ru/11/4/17.html/
10) Л.Н. Полякова «Основы SQL.»
    http://www.uchi-it.ru/11/4/16.html/
11) Рик Гринвальд, Роберт Стаковьяк, Гэри Додж, Дэвид Кляйн, Бен Шапиро, Кристофер Дж. Челья «Программирование баз данных Oracle для профессионалов»; Вильямс, Диалектика; 2007
12) NATASHA DEYSEL, «A MODEL FOR INFORMATION SECURITY CONTROL AUDIT FOR SMALL TO MEDIUM-SIZED ORGANISATIONS», 2009
13) Проект «КОНЦЕПЦИЯ АУДИТА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ СИСТЕМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ОРГАНИЗАЦИЙ »

www.admin-smolensk.ru/information_security/_licen/016.doc

14) Статья «Информационная безопасность» http://ru.wikipedia.org/wiki/Информационная_безопасность
15) А.А. Анисимов. Менеджмент в сфере информационной безопасности – М: Интернет-Университет Информационных Технологий – ИНТУИТ. http://www.intuit.ru/department/itmngt/manofis/7/
16) Информационная безопасность. Курс лекций. http://gendocs.ru/v2674/%D0%BB%D0%B5%D0%BA%D1%86%D0%B8%D0%B8_-_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D1%8C?page=3
17) ISACA Russia Chapter. CISA Exam Preparation Course. April-May 2001.
18) CRAMM v.4.0 User’s Guide.
19) What is CRAMM? http://www.gammassl.co.uk/topics/hot5.html
20) SANS/GIAC Site Certification Program, http://www.sans.org/SCORE
21) SysTrust Services, http://www.aicpa.org/assurance/systrust/index.htm
22) Ernst&Young (CIS) Limited, Independent Accountant's Report,https://processcertify.ey.com/vimpelcom2/vimpelcom_opinion.html
23) BSI/IT Baseline Protection Manual, http://www.bsi.bund.de/gshb/english/menue.htm
24) Александр Астахов. Анализ защищенности автоматизированных систем, GLOBALTRUST.RU, 2002 г.,http://www.globaltrust.ru/security/Pubs/Pub1_AAM_SecEval.htm
25) Александр Астахов, аудит безопасности ИС, CISA, 2002 http://www.iso27000.ru/chitalnyi-zai/audit-informacionnoi-bezopasnosti/vidy-audita-informacionnoi-bezopasnosti
26) Статья «Аудит информационной безопасности» http://ru.wikipedia.org/wiki/Аудит_информационной_безопасности
27) Виталий Иванод, Аудит безопасности сетевых устройств http://www.compress.ru/article.aspx?id=22568&iid=1036, КомпьютерПресс 11'2011
28) Лилия Козленко, ИБ в современных СУБД http://www.compress.ru/article.aspx?id=10099&iid=419#02, КомпьютерПресс 11'2011
29) Статья «Сравнение СУБД» http://blog.groovytel.ru/2009/11/20/%D1%81%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5-%D1%81%D1%83%D0%B1%D0%B4-

%D0%B4%D0%BB%D1%8F-%D0%B2%D0%B5%D0%B1-%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%BE%D0%B2/

30) Т.С. Карпова. Защита информации в БД – М: Интернет-Университет Информационных Технологий – ИНТУИТ. http://www.intuit.ru/department/database/dbmdi/13/

31) Статья «UML» http://ru.wikipedia.org/wiki/UML

32) Статься «Диаграмма развертывания» http://www.info-system.ru/designing/methodology/uml/theory/deployment_diagram_theory.html

33) Статья «Диаграмма деятельности» http://www.info-system.ru/designing/methodology/uml/theory/activity_diagramm_theory.html

34) А.В. Леонков. Язык UML в анализе и проектировании программных сисстем и бизнес-процессов – М: Интернет-Университет Информационных Технологий – ИНТУИТ. http://www.intuit.ru/department/se/uml2/8/

35) А.В. Бабич. Введение в UML – М: Интернет-Университет Информационных Технологий – ИНТУИТ. http://www.intuit.ru/department/se/intuml/4/1.html

36) Статья «Диаграмма развертывания» http://ru.wikipedia.org/wiki/Диаграмма развёртывания

37) C++ GUI Programming with Qt 4 By Jasmin Blanchette, Mark Summerfield Publisher: Prentice Hall, Pub Date: June 21, 2006

38) Справочник SQL, http://dimonchik.com/insert.html

39) Статья «SQL. Функции даты и времени», http://www.site-do.ru/db/sql13.php

40) Б.П. Пальчун, А.С. Старостин Концептуальные вопросы построения интеллектуальных и адаптивных систем информационной безопасности, Россия, г. Москва, ФГУП «Концерн "Системпром"», 2012

41) Галатенко В. А. Основы информационной безопасности. – М: Интернет-Университет Информационных Технологий – ИНТУИТ. РУ, 2003.

42) Щербаков А. Ю. Введение в теорию и практику компьютерной безопасности. – М.: Издательство Молгачева С. В., 2001.

43) Галатенко В. А. Стандарты информационной безопасности. – М: Интернет-Университет Информационных Технологий – ИНТУИТ. РУ, 2004.

44) Теория и практика обеспечения информационной безопасности / Под ред. П. Д. Зегжды. – М: Яхтсмен, 1996.

45) Грязнов Е., Панасенко С. Безопасность локальных сетей – Электрон. журнал "Мир и безопасность" № 2, 2003. – Режим доступа к журн.: www.daily.sec.ru.
46) В. Г. Олифер, Н. А. Олифер. Компьютерные сети. Принципы, технологии, протоколы. – СПб: Питер, 2000.
47) Карпов Е. А., Котенко И. В., Котухов М. М., Марков А. С., Парр Г. А., Рунеев А. Ю. Законодательно-правовое и организационно-техническое обеспечение информационной безопасности автоматизированных систем и информационно-вычислительных сетей / Под редакцией И. В.Котенко. – СПб.: ВУС, 2000.
48) Спортак Марк, Паппас Френк. Компьютерные сети и сетевые технологии. – М.: ТИД "ДС", 2002.
49) Котухов М. М., Марков А. С. Законодательно-правовое и организационно-техническое обеспечение информационной безопасности автоматизированных систем. – 1998.
50) Статья «Разработка рекомендаций по организации и проведению аудита безопасности информационной системы хозяйствующего субъекта» http://www.audit-ib.ru/
51) В.В. Домарев, Безопасность ИТ: Системный подход, ТИД "ДС", 2004

Приложение 1

Код файла main.cpp

```
#include <QtDebug>
#include <QtGui>
#include <QtSql>
#include <QSqlDatabase>
#include <QSqlQueryModel>
#include <QSqlError>
#include <QTableView>
#include <QSqlDriver>
#include "enter.h"
#include "mainwindow.h"
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QString  st1;
    QString  st2;
    Enter dialog;
    dialog.show();
    while (dialog.exec()){}
    st1=dialog.st_e1;
    st2=dialog.st_e2;
    QSqlDatabase db;
    db = QSqlDatabase::addDatabase("QPSQL");
    db.setHostName("localhost");
    db.setDatabaseName("postgres");
    db.setUserName(st1);
    db.setPort(5432);
    db.setPassword(st2);
    MainWindow main;
    if (!db.open()) {
             qDebug() << QObject::trUtf8("Ne smogla ya otkrit bazu!") <<
db.lastError().text();
            qDebug() << QApplication::libraryPaths();
            return -1;
      } else {
        main.show();
        }
    return app.exec();
}
```

Код файла mainwindow.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QtGui>
#include <QtSql>
#include "tab.h"
#include "pole.h"
#include "comment.h"
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    model = new QSqlTableModel();
    model->setTable("logs");
    model->select();
    QTextCodec::setCodecForTr(QTextCodec::codecForName("CP1251"));
    connect(ui->action, SIGNAL(triggered()), this, SLOT(tab()));
    connect(ui->action_2,SIGNAL(triggered()), this, SLOT(pole()));
    QTableWidgetItem *item_h_1 = new QTableWidgetItem;
    item_h_1->setText(tr("Пользователь"));
    ui->tableWidget->setHorizontalHeaderItem(0, item_h_1);
    QTableWidgetItem *item_h_2 = new QTableWidgetItem;
    item_h_2->setText(tr("Время"));
    ui->tableWidget->setHorizontalHeaderItem(1, item_h_2);
    QTableWidgetItem *item_h_3 = new QTableWidgetItem;
    item_h_3->setText(tr("Имя таблицы"));
    ui->tableWidget->setHorizontalHeaderItem(2, item_h_3);
    QTableWidgetItem *item_h_4 = new QTableWidgetItem;
    item_h_4->setText(tr("Записи аудита"));
    ui->tableWidget->setHorizontalHeaderItem(3, item_h_4);
    QString str;
    MainWindow::on_comboBox_currentIndexChanged(str);
    ui->comboBox->setItemDelegate(new MyItemDelegate(this));
    ui->comboBox_2->setItemDelegate(new MyItemDelegate(this));
    ui->comboBox_3->setItemDelegate(new MyItemDelegate(this));
    QSqlQueryModel *model1 = new QSqlQueryModel();
    int sum;
    QSqlRecord record;
    ui->comboBox  ->addItem("");
    ui->comboBox_2->addItem("");
    ui->comboBox_3->addItem("");
```

```
            model1->setQuery("SELECT DISTINCT username FROM logs");
            sum=model1->rowCount();
            for(int i = 0; i < sum;i++){
            record = model1->record(i);
            ui->comboBox->addItem(record.value("username").toString());;
            }
            model1->setQuery("SELECT DISTINCT date(added) FROM logs");
            sum=model1->rowCount();
            for(int i = 0; i < sum;i++){
            record = model1->record(i);
            ui->comboBox_2->addItem(record.value("date").toString());
            }
            model1->setQuery("SELECT DISTINCT tablename FROM logs");
            sum=model1->rowCount();
            for(int i = 0; i < sum;i++){
            record = model1->record(i);
            ui->comboBox_3->addItem(record.value("tablename").toString());;
            }
      }
      void MainWindow::tab()
      {
         QSqlQueryModel *model = new QSqlQueryModel();
         QTextCodec::setCodecForTr(QTextCodec::codecForName("CP1251"));
         model->setQuery(tr("SELECT  table_name AS \"Имя таблицы\", pole_name
AS \"Поля\" FROM settings "));
         L_TAB *dialog = new L_TAB;
         dialog->tableView->setModel(model);
         dialog->tableView->horizontalHeader()->setResizeMode(
QHeaderView::Stretch );
         dialog->tableView-
>setSelectionBehavior(QAbstractItemView::SelectRows);
         dialog->tableView->verticalHeader()->hide();
         dialog->show();
      }
      void MainWindow::pole()
      {
         QSqlQueryModel *model = new QSqlQueryModel();
         model->setQuery("SELECT tablename FROM pg_tables WHERE tablename NOT
LIKE 'pg\\_%' AND tablename NOT LIKE 'sql\\_%' AND tablename NOT LIKE 'logs'
AND tablename NOT LIKE 'settings'");
         Pole *dialog = new Pole;
         dialog->comboBox->setModel(model);
         dialog->comboBox->setItemDelegate(new MyItemDelegate(this));
```

```
         dialog->tableWidget->setItemDelegate(new MyItemDelegate(this));
         dialog->tableWidget->horizontalHeader()-
>setResizeMode(QHeaderView::Stretch );
         dialog->show();
      }
      MainWindow::~MainWindow()
      {
          delete ui;
      }
      void MainWindow::on_comboBox_currentIndexChanged(QString )
      {
          QSqlQueryModel *model_ch = new QSqlQueryModel();
          //---------------filter------------------
          if      (ui->comboBox->currentText()!=""      &      ui->comboBox_2-
>currentText()=="" & ui->comboBox_3->currentText()==""){
              model_ch->setQuery("SELECT * FROM logs where username='"
                                 + ui->comboBox->currentText()
                                 + "'");
          }
          if      (ui->comboBox->currentText()==""      &      ui->comboBox_2-
>currentText()!="" & ui->comboBox_3->currentText()==""){
              model_ch->setQuery("SELECT * FROM logs where date(added)='"
                                 + ui->comboBox_2->currentText()
                                 + "'");
          }
          if      (ui->comboBox->currentText()==""      &      ui->comboBox_2-
>currentText()==="" & ui->comboBox_3->currentText()!=""){
              model_ch->setQuery("SELECT * FROM logs where tablename='"
                                 + ui->comboBox_3->currentText()
                                 + "'");
          }
          //-------------<1-1--------------------
          if      (ui->comboBox->currentText()!=""      &      ui->comboBox_2-
>currentText()!="" & ui->comboBox_3->currentText()==""){
              model_ch->setQuery("SELECT * FROM logs where username='"
                                 + ui->comboBox->currentText()
                                 + "' and date(added)='"
                                 + ui->comboBox_2->currentText()
                                  + "'");
          }
          if      (ui->comboBox->currentText()!=""      &      ui->comboBox_2-
>currentText()==="" & ui->comboBox_3->currentText()!=""){
              model_ch->setQuery("SELECT * FROM logs where username='"
```

```
                                + ui->comboBox->currentText()
                                + "' and tablename='"
                                + ui->comboBox_3->currentText()
                                 + "'");
            }
            if      (ui->comboBox->currentText()==""      &      ui->comboBox_2-
>currentText()!="" & ui->comboBox_3->currentText()!=""){
                model_ch->setQuery("SELECT * FROM logs where date(added)='"
                                + ui->comboBox_2->currentText()
                                + "' and tablename='"
                                + ui->comboBox_3->currentText()
                                 + "'");
            }
            //------------------<2-2------------------
            if      (ui->comboBox->currentText()!=""      &      ui->comboBox_2-
>currentText()!="" & ui->comboBox_3->currentText()!=""){
                model_ch->setQuery("SELECT * FROM logs where username='"
                                + ui->comboBox->currentText()
                                + "' and date(added)='"
                                + ui->comboBox_2->currentText()
                                + "' and tablename='"
                                + ui->comboBox_3->currentText()
                                + "'");
            }
            if      (ui->comboBox->currentText()==""      &      ui->comboBox_2-
>currentText()==""" & ui->comboBox_3->currentText()=="")
                model_ch->setQuery("SELECT * FROM logs");
            //---------------filter------------------
                ui->tableWidget->setRowCount(model_ch->rowCount());
                for(int i = 0;i < model_ch->rowCount();i++){
                QTableWidgetItem *item_0 = new QTableWidgetItem;
                QTableWidgetItem *item_1 = new QTableWidgetItem;
                QTableWidgetItem *item_2 = new QTableWidgetItem;
                QTableWidgetItem *item_3 = new QTableWidgetItem;
                QString name = model_ch->record(i).value("username").toString();
                item_0->setText(name);
                ui->tableWidget->setItem(i,0,item_0);
                QDateTime              name1              =              model_ch-
>record(i).value("added").toDateTime();
                item_1->setText(name1.toString(Qt::SystemLocaleShortDate));
                ui->tableWidget->setItem(i,1,item_1);
                QString              name2              =              model_ch-
>record(i).value("tablename").toString();
```

```
            item_2->setText(name2);
            ui->tableWidget->setItem(i,2,item_2);
            QString name3 = model_ch->record(i).value("text").toString();
            item_3->setText(name3);
            ui->tableWidget->setItem(i,3,item_3);
        }
         ui->tableWidget->horizontalHeader()->setStretchLastSection(0);
         if (model_ch->rowCount()!=0){
         ui->tableWidget->resizeColumnsToContents();
         ui->comboBox   ->setFixedWidth(ui->tableWidget->horizontalHeader()-
>sectionSize(0)-5);
         ui->comboBox_2->setFixedWidth(ui->tableWidget->horizontalHeader()-
>sectionSize(1)-5);
         ui->comboBox_3->setFixedWidth(ui->tableWidget->horizontalHeader()-
>sectionSize(2)-5);
         }
         ui->tableWidget->horizontalHeader()->setStretchLastSection(1);
    }
    void MainWindow::on_comboBox_2_currentIndexChanged(QString )
    {
        QString str;
    MainWindow::on_comboBox_currentIndexChanged(str);
    }
    void MainWindow::on_comboBox_3_currentIndexChanged(QString )
    {
        QString str;
    MainWindow::on_comboBox_currentIndexChanged(str);
    }
```

Код файла enter.cpp

```
    #include <QtGui>
    #include "enter.h"
    #include "mainwindow.h"
    Enter::Enter(/*MainWindow *main,*/ QWidget *parent)
        : QDialog(parent)//,_main(main)
    {
        setupUi(this);
        connect(okButton, SIGNAL(clicked()), this, SLOT(enter_bd()));
    }
    void Enter::enter_bd()
    {
        st_e1 = lineEdit->text();
```

```
    st_e2 = lineEdit_2->text();
   // Enter::close();
    this->close();
}
Enter::~Enter()
{}
```

Код файла tab.cpp

```
#include <QtGui>
#include "tab.h"
#include "mainwindow.h"
#include "ui_mainwindow.h"
L_TAB::L_TAB(QWidget *parent)
    : QDialog(parent)
{
    setupUi(this);
}
void L_TAB::on_pushButton_clicked()
{
    QSqlQueryModel *model = new QSqlQueryModel();
    model->setQuery("SELECT * FROM settings");
    tableView->setFocus();
    QModelIndex index = tableView->currentIndex();
    QSqlRecord record;
    record = model->record(index.row());
    model->setQuery("DELETE FROM settings WHERE table_name = '"
                    +record.value("table_name").toString()
                    +"'");
    model->setQuery("DROP FUNCTION add_to_log_"
                    +record.value("table_name").toString()
                    +"() CASCADE");
    model->setQuery("SELECT * FROM settings");
    tableView->setModel(model);
}
```

Код файла pole.cpp

```
#include <QtGui>
#include "QList"
#include "pole.h"
#include "mainwindow.h"
#include "ui_mainwindow.h"
Pole::Pole(QWidget *parent)
```

```
          : QDialog(parent)
      {
          setupUi(this);
          //pre_size=0;
      }
      void Pole::on_pushButton_clicked()
      {
          QStringList list, list_com;
        //        QModelIndexList    mlist    =    listView->selectionModel()-
>selectedIndexes();
          QString list_pole,list_pole_com;
          QString tmp_n, tmp_o,tmp_d,tmp_str,tmp_str_n,
                  tmp_str_o,tmp_str_u,tmp_if,tmp_if_1,tmp_if_2,
sel_com,str_com;
          QSqlQueryModel *model = new QSqlQueryModel();
          QTextCodec::setCodecForTr(QTextCodec::codecForName("CP1251"));
          // установка кириллицы
      QModelIndexList mlist1=tableWidget->selectionModel()->selectedIndexes();
      for(int i = 0;i < mlist1.count();i+=2){
      //Получаем отображаемое имя
      list.append(mlist1.at(i).data(Qt::DisplayRole).toString());
      }
      for(int i = 1;i < mlist1.count();i+=2){
      //Получаем комментарии
      list_com.append(mlist1.at(i).data(Qt::DisplayRole).toString());
      }
      for(int i = 0; i < list.count();i++){
      //Записываем отображаемое имя
      list_pole += list.at(i) + ";";
      }
      for(int i = 0; i < list_com.count();i++){
      //Записываем комментарии
      list_pole_com += list_com.at(i) + ";";
      }
      int sum=0;
      // проверка на существование --->>
          model->setQuery ("SELECT table_name FROM settings");
          sum=model->rowCount();
          QSqlRecord record;
          flag = 0;
          for(int i = 0; i < sum;i++){
          record = model->record(i);
```

```
          if     (record.value("table_name").toString()     ==     comboBox-
>currentText())
          flag = 1;
      }
      // проверка на существование ---<<
          if (flag==0){
          model->setQuery  ("INSERT  INTO  settings  (table_name,  pole_name)
VALUES ('"
                            +comboBox->currentText()
                            +"','"
                            +list_pole
                            +"')");
      //comment
          for(int i = 0; i < list.count();i++){
          model->setQuery  ("COMMENT ON COLUMN "
                            +comboBox->currentText() // table
                            +"."
                            +list.at(i) // name of pole
                            +" IS '"
                            +list_com.at(i) // com for pole
                            +"';");
      }
          //construct strings for trig fun--->>
          tmp_str = "retstr := retstr || mstr; ";
          sel_com = "tmp=(select description from pg_description "
                    "join pg_class on pg_description.objoid = pg_class.oid "
                    "where relname = '"
                    +comboBox->currentText()
                    +"' and objsubid = ";
          str_com = "retstr := retstr || tmp; retstr := retstr || ' ';";
          if (list.count()>0){
              for(int i = 0;i < list.count();i++){
              int ind=i*2;
              QString ii=QString::number(i);
              tmp_d = tmp_d
                      +" astr_"
                      +ii
                      +" varchar(100);"
                      +" astr_o_"
                      +ii
                      +" varchar(100);";
              tmp_n = tmp_n
                      +" astr_"
```

```
                        +ii
                        +"= NEW."+list.at(i)+";";
                tmp_o = tmp_o
                        +" astr_o_"
                        +ii
                        +"= OLD."+list.at(i)+";";
                tmp_if_1 = " if NEW."
                           +list.at(i)+
                           +" is not null then ";
                tmp_str_n  =  tmp_str_n  +  tmp_if_1+  tmp_str  +  sel_com  +
QString::number((mlist1.at(ind).row()+1))
                            + ");  if tmp is not null then "
                            + str_com
                            + " end if; retstr := retstr || astr_"
                            + ii
                            + "; end if;";
                tmp_if_2 = " if OLD."
                           +list.at(i)+
                           +" is not null then ";
                tmp_str_o  =  tmp_str_o  +  tmp_if_2+  tmp_str  +  sel_com  +
QString::number((mlist1.at(ind).row()+1))
                            + ");  if tmp is not null then "
                            + str_com
                            + " end if; retstr := retstr || astr_o_"
                            + ii
                            + "; end if;";
                tmp_if = "if (astr_"
                         +ii
                         +"<>"
                         +"astr_o_"
                         +ii
                         +") then ";
                tmp_str_u  =  tmp_str_u  +  tmp_if  +  tmp_str  +  sel_com  +
QString::number((mlist1.at(ind).row()+1))
                            + ");  if tmp is not null then "
                            + str_com
                            + " end if; retstr := retstr || '"+tr("C ")+"';"
                            + " retstr := retstr || astr_o_"
                            + ii
                            + ";"
                            + " retstr := retstr || '"+tr(" Ha ")+"';"
                            + " retstr := retstr || astr_"
                            + ii
```

```
                + ";"
                + "end if;";
    }
 }
 //construct strings for trig fun---<<
 //generete trig fun & trig--->>
model->setQuery
    //   qDebug() <<
         (
         "CREATE OR REPLACE FUNCTION add_to_log_"
         //+QString::number(sum+1)
         +comboBox->currentText()
         +"() RETURNS TRIGGER AS $$ "
         "DECLARE "
         "mstr varchar(30);"
         +tmp_d
         +"tmp varchar(30);"
         "retstr varchar(254);"
         " BEGIN "
         " IF TG_OP = 'INSERT' THEN "
         +tmp_n
         +"mstr := '"+tr(" Добавление нового: ")+"';"
         "retstr := ' ';"
         +tmp_str_n
         +"INSERT INTO logs(text,added) values (retstr,NOW());"
         "RETURN NEW;"
         "ELSIF TG_OP = 'UPDATE' THEN "
         +tmp_n
         +tmp_o
         +"mstr := '"+tr(" Изменение: ")+"';"
         "retstr := ' ';"
         +tmp_str_u
         +"INSERT INTO logs(text,added) values (retstr,NOW());"
         " RETURN NEW;"
         "ELSIF TG_OP = 'DELETE' THEN "
         +tmp_o
         +"mstr := '"+tr(" Удаление: ")+"';"
         "retstr := ' ';"
         +tmp_str_o
         +"INSERT INTO logs(text,added) values (retstr,NOW());"
         "RETURN OLD;"
         "END IF;"
         "END;"
```

```
                "$$ LANGUAGE plpgsql;");
        model->setQuery (
                "CREATE TRIGGER log "
                "AFTER INSERT OR UPDATE OR DELETE ON "
                +comboBox->currentText()
                +" FOR EACH ROW EXECUTE PROCEDURE add_to_log_"
                +comboBox->currentText()
                +"();");
        //generete trig fun & trig---<<
    }   else   QMessageBox::critical(0,   "Error",   tr("Создание   триггера
невозможно! Триггер для выбранной таблицы уже существует. Для создания
триггера необходимо удалить уже существующий"));
    }
    void Pole::on_comboBox_currentIndexChanged(QString )
    {
        QSqlQueryModel *model = new QSqlQueryModel();
        model->setQuery("SELECT  attname  FROM  pg_attribute,  pg_type  WHERE
typname = '"
                        +comboBox->currentText()
                        +"'  AND  attrelid  =  typrelid  AND  attname  NOT  IN
('cmin', 'cmax', 'ctid', 'oid', 'tableoid', 'xmin', 'xmax')");
        tableWidget->setSelectionMode(QAbstractItemView::MultiSelection);
        tableWidget->setSelectionBehavior(QAbstractItemView::SelectRows);
        QTextCodec::setCodecForTr(QTextCodec::codecForName("CP1251"));
        QTableWidgetItem *item_h_1 = new QTableWidgetItem;
        item_h_1->setText(tr("Выбор полей:"));
        tableWidget->setHorizontalHeaderItem(0, item_h_1);
        QTableWidgetItem *item_h_2 = new QTableWidgetItem;
        item_h_2->setText(tr("Комментарии:"));
        tableWidget->setHorizontalHeaderItem(1, item_h_2);
        tableWidget->setRowCount(model->rowCount());
        for(int i = 0;i < model->rowCount();i++){
            QTableWidgetItem *item = new QTableWidgetItem;
            QString name = model->record(i).value("attname").toString();
            item->setText(name);
            item->setFlags(Qt::ItemIsSelectable | Qt::ItemIsEnabled);
            tableWidget->setItem(i,0,item);
        }
        tableWidget->verticalHeader()->hide();
    }
```