ТАШКЕНТСКИЙ
УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

БЕЗОПАСНОСТЬ БАЗЫ ДАННЫХ

С.К.ГАНИЕВ

А.Я.ИРГАШЕВА

К.А.ТАШЕВ

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ УЗБЕКИСТАН

ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

С.К. ГАНИЕВ,

Д.Я. ИРГАШЕВА,

К.А. ТАШЕВ.

БЕЗОПАСНОСТЬ БАЗЫ ДАННЫХ

Учебник

(под редакцией профессора Ганиева С.К.)

Рекомендовано Министерством высшего и среднего специального образования Республики Узбекистан в качестве учебника для студентов направления 5330500 – "Компьютерный инжиниринг" (*"Компьютерный инжиниринг", "ИТ-сервис", "Информационная безопасность", "Мультимедийные технологии"*) высших учебных заведений.

Ташкент – 2017

UDK: 004.65-056.5
ВБК: 32.973.202-018.2
Г 19

С.К. Ганиев, Д.Я. Иргашева, К.А. Ташев. **Безопасность Базы Данных. Т.: « Aloqachi», 2017,224 бет.**

Учебник «Безопасность базы данных» подготовлен преподавателями кафедры "Информационная безопасность" базового высшего учебного заведения Ташкентского университета информационных технологий и включает в себя такие вопросы как: основные характеристики методов, средств и механизмов обеспечения безопасности базы данных; виды управления базами данных; технологические аспекты информационной безопасности базы данных; модели и методы разграничения доступа в базы данных; информационная безопасность распределенных систем базы данных; аудит безопасности и резервное копирование базы данных; принципы защиты систем управления базами данных, также приведены нормативные документы обеспечения безопасности базы данных.

Учебник предназначен для студентов направления 5330500 – "Компьютерный инжиниринг" (*"Компьютерный инжиниринг", "ИТ-сервис", "Информационная безопасность", "Мультимедийные технологии"*) высших учебных заведений и может быть использован лицами, занимающимися в сфере безопасности информационной технологии и компьютерных систем.

"Ma'lumotlar bazasi xavfsizligi" fani bo'yicha darslik tayanch oliy o'quv yurti Toshkent axborot texnologiyalari universitetining "Axborot xavfsizligi" kafedrasi professor-o'qituvchilari tomonidan tayyorlangan bo'lib, unda ma'lumotlar bazasi xavfsizligini ta'minlovchi usullar, vositalar va mexanizmlarining asosiy xarakteristikalari; ma'lumotlar bazasini boshqarish turlari; ma'lumotlar bazasi xavfsizligining texnologik jihatlari; ma'lumotlar bazasidan foydalanishni cheklashning modellari va usullari; ma'lumotlar bazasining taqsimlangan tizimida axborot xavfsizligi; xavfsizlik auditi va ma'lumotlar bazasini rezervli nusxalash masalalari hamda ma'lumotlar bazasini boshqarish tizimlarining himoya prinsiplari muhokama etilib, ma'lumotlar bazasi xavfsizligini ta'minlashdagi me'yoriy xujjatlar keltirilgan.

Darslik oliy o'quv yurtining 5330500- "Kompyuter injiniringi" *("Kompyuter injiniringi", "AT-servis", "Axborot xavfsizligi", "Multimedia texnologiyalari")* ta'lim yo'nalishi talabalari uchun mo'ljallangan bo'lib, undan axborot texnologi-yalari, kompyuter tizimlari xavfsizligi sohasida faoliyat ko'rsatuvchilar foydala-nishlari mumkin.

"Data Base Security" textbook is prepared by professor-teachers of department of "Information Security" of base top educational institution, Tashkent University of information technology and it includes features of methods, tools and mechanisms of data base security; methods of data base management; technical aspects of data base security; access control methods and models in database; information security in distributive database systems; security audit and database backup; security principles of database management systems and regulations in database security.

This textbook is designed for students of 5330500 – "Computer engineering" ("Computer engineering", "IT-service", "Information Security", "Multimedia technologies") school of educational institutions and people that deal in information technologies and computer system security fields can use.

UDK: 004.65-056.5
ВБК: 32.973.202-018.2

Рецензенты:

Сагатов М.В. – ТГТУ, заведующий кафедрой «Информационные системы», д.т.н., профессор.

Мирзаев О.Н.–первый заместитель директора центра обеспечения информационной безопасности при Министерстве по развитию информационных технологий и коммуникаций Республики Узбекистан.

Chul Soo LEE - проректор-консультант по ИКТ Ташкентского университета информационных технологий.

ISBN 978-9943-326-87-3

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ

MUNDARIJA

CONTENT

ВВЕДЕНИЕ

Использование автоматизированных информационных систем (АИС) порождает общие информационные ресурсы в виде базы или совокупности баз данных, состояние и функционирование которых может критически влиять на жизнедеятельность предприятия, организации. В результате требуется отдельный специфический контроль состояния, получивший в процессе внедрения автоматизированных информационных систем в практику информационного обеспечения деятельности предприятий и организаций специальный термин – «защита базы данных». Взгляды на функции и содержание задач, решаемых в процессе защиты базы данных, формировались вместе со становлением индустрии автоматизированных информационных систем, менялись с изменениями и усложнениями программно-технических аспектов их реализации, но со временем постепенно формировался некоторый их базовый перечень, и защита баз данных вошло неотъемлемым ключевым компонентом в теорию и практику автоматизированных информационных систем.

Термин «защита базы данных» означает предупреждение несанкционированного доступа к данным, их изменения или разрушения со стороны пользователей; предупреждение изменения или разрушения данных при сбоях аппаратных и программных средств и ошибках в работе сотрудников группы эксплуатации.

Предлагаемый вниманию читателей учебник «Защита базы данных» состоит из пяти глав.

Первая глава учебника посвящена базовым положениям. Приводятся основные характеристики методов, средств и механизмов обеспечения безопасности базы данных, виды систем управления базами данных (СУБД), а также технологические аспекты информационной безопасности базы данных.

Во второй главе рассматриваются модели и методы разграничения доступа в базы данных. Подробно анализируются вопросы организации разграничения доступа в базы данных на основе дискреционной, мандатной и ролевой моделей. Особое внимание удалено сравнительной оценке вышеуказанных моделей с точки зрения эффективности обеспечения защиты базы данных от заданного множества угроз безопасности.

Третья глава отведена принципам информационной безопасности распределенных систем базы данных. Приводится концепция информационной безопасности в распределенных системах базы данных. Подробно освещены технологии «клиент-сервер», технологии реплицирования, технологии объектного связывания как самостоятельные направления в технологии распределенных систем.

Четвертая глава учебника посвящена аудиту безопасности и резервному копированию базы данных. Излагаются особенности проведения аудита безопасности в системах управления базами данных. Подробно освещены вопросы проведения и реализации аудита безопасности баз данных на примере *СУБД Oracle*. Рассматриваются вопросы резервного копирования, как периодическое дублирование (реплицирование) или создание запасных копий критически важных данных с целью их восстановления в случае потери оригинала. Далее освещены технологии реплицирования, а также процесс синхронизации репликации в современных системах управления базами данных.

Содержание пятой главы – это описание стандартов спецификации по обеспечению безопасности базы данных. Обсуждаются архитектура и принцип функционирования подсистемы безопасности базы данных, профили защиты систем управления базами данных. Приведены нормативные документы в области обеспечения безопасности базы данных.

В приложениях приведены список сокращений и толковый словарь терминов на русском, узбекском, английском языках.

Глава 1. МЕТОДЫ, СРЕДСТВА И МЕХАНИЗМЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ БАЗЫ ДАННЫХ

1.1. Основные характеристики методов, средств и механизмов обеспечения безопасности базы данных

Исследования по проблемам защиты компьютерной информации, проведенные в конце 70-х—начале 80-х годов, развитые впоследствии в различных приложениях и закрепленные в соответствующих стандартах, определяют в качестве составных элементов понятия *безопасности информации* три компонента:

- *конфиденциальность* (защита от несанкционированного доступа);

- *целостность* (защита от несанкционированного изменения информации);

- *доступность* (защита от несанкционированного удержания информации и ресурсов, защита от разрушения, защита работоспособности).

Составляющим безопасности информации противостоят соответствующие *угрозы.* Под *угрозой безопасности информации* понимается *осуществляемое или потенциально осуществимое воздействие на компьютерную систему, которое прямо или косвенно может нанести ущерб безопасности информации.* Угрозы реализуют или пытаются реализовать *нарушители* информационной безопасности.

Формализованное описание или представление *комплекса возможностей нарушителя по реализации тех или иных угроз безопасности информации* называют *моделью нарушителя (злоумышленника).*

Качественное описание комплекса организационно-технологических и программно-технических мер по обеспечению защищенности информации в компьютерной системе (КС) называют *политикой безопасности.* Формальное (математическое, алгоритмическое, схемотехническое)

выражение и формулирование политики безопасности называют *моделью безопасности.*

Некоторые термины относящийся к обеспечению безопасности баз данных (БД) приведены ниже:

- доступ к информации (access to information) - ознакомление с информацией, ее обработка (в частности, копирование), модификация, уничтожение;

- субъект доступа (access subject) - лицо или процесс, действия которого регламентируются правилами разграничения доступа;

- объект доступа (access object) - единица информации автоматизированной системы, доступ к которой регламентируется правилами разграничения доступа;

- правила разграничения доступа (security policy) - совокупность правил, регламентирующих права субъектов доступа к объектам доступа;

- санкционированный доступ (authorized access to information) - доступ к информации, который не нарушает правил разграничения доступа;

- несанкционированный доступ (unauthorized access to information) - доступ к информации, который нарушает правила разграничения доступа с использованием штатных средств, предоставляемых средствами вычислительной техники или автоматизированными системами;

- уровень полномочий субъекта доступа (subject privilege) - совокупность прав доступа субъекта доступа (для краткости в дальнейшем мы будем использовать термин «привилегия»);

- нарушитель правил разграничения доступа (security policy violator) - субъект доступа, который осуществляет несанкционированный доступ к информации;

- модель нарушителя правил разграничения доступа (security policy violator model) - абстрактное (формализованное или неформализованное) описание нарушителя правил разграничения доступа;

- целостность информации (information integrity) - способность средства вычислительной техники (в рассматриваемом случае - информационной системы в целом) обеспечить неизменность информации в условиях случайного и (или) преднамеренного искажения (разрушения);

- метка конфиденциальности (sensitivity label) - элемент информации, характеризующий конфиденциальность объекта;

- многоуровневая защита (multilevel secure) - защита, обеспечивающая разграничение доступа субъектов с различными правами доступа к объектам различных уровней конфиденциальности.

В структуре программного обеспечения компьютера, за организацию, размещение и оперирование данными во внешней (долговременной) памяти отвечает операционная система компьютера, соответствующий компонент которой чаще всего называется «файловой системой». Данные во внешней памяти компьютера представлены именованными совокупностями, называемыми файлами. В большинстве случаев операционная (файловая) система не «знает» внутренней смысловой логики организации данных в файлах и оперирует с ними как с однородной совокупностью байтов или строк символов.

С точки зрения смысла и назначения КС файлы данных имеют структуру, отражающую информационно-логическую схему предметной области КС. Эта структура данных в файлах должна обязательно учитываться в операциях обработки. Вместе с тем, в силу невозможности в большинстве случаев размещения файлов баз данных сразу целиком в оперативной памяти компьютера, структуру данных в файлах баз данных приходится учитывать при организации операций обращения к файлам во внешней памяти.

Отсюда вытекает основная особенность СУБД как вида программного обеспечения. Будучи по природе *прикладным программным обеспечением,* т.е. предназначенным для решения конкретных прикладных задач, СУБД изначально выполняли и *системные функции* — расширяли возможности

файловых систем *системного программного обеспечения.* В общем плане можно выделить следующие *функции,* реализуемые *СУБД:*

- организация и поддержание логической структуры данных (схемы базы данных);
- организация и поддержание физической структуры данных во внешней памяти;
- организация доступа к данным и их обработка в оперативной и внешней памяти.

Организация и поддержание логической структуры данных (схемы базы данных) обеспечивается средствами *модели организации данных.* В обиходе просто «модель данных».

Модель данных определяется способом организации данных, ограничениями целостности и множеством операций, допустимых над объектами организации данных. Соответственно модель данных разделяют на три составляющие — *структурную, целостную* и *манипуляционную.*

Известны три основные модели организации данных:

- иерархическая;
- сетевая;
- реляционная.

Модель организации данных, по сути, определяет *внутренний информационный язык* автоматизированного банка данных, реализующего автоматизированную информационную систему.

Модели данных, поддерживаемые СУБД, довольно часто используются в качестве критерия для классификации СУБД. Исходя из этого, различают *иерархические СУБД, сетевые СУБД* и *реляционные СУБД.*

Другой важной функцией СУБД является *организация и поддержание физической структуры данных во внешней памяти.* Эта функция включает организацию и поддержание внутренней структуры файлов базы данных, иногда называемой *форматом файлов базы данных,* а также создание и поддержание специальных структур (индексы, страницы) для эффективного

и упорядоченного доступа к данным. В этом плане эта функция тесно связана с третьей функцией СУБД — организацией доступа к данным.

Организация и поддержание физической структуры данных во внешней памяти может производиться как на основе штатных средств файловых систем, так и на уровне непосредственного управления СУБД устройствами внешней памяти.

Организация доступа к данным и их обработка в оперативной и внешней памяти осуществляется через реализацию процессов, получивших название транзакций. *Транзакцией называют последовательную совокупность операций, имеющую отдельное смысловое значение по отношению к текущему состоянию базы данных*. Так, например, транзакция по удалению отдельной записи в базе данных последовательно включает определение страницы файла данных, содержащей указанную запись, считывание и пересылку соответствующей страницы в буфер оперативной памяти, собственно удаление записи в буфере ОЗУ, проверку ограничений целостности по связям и другим параметрам после удаления и, наконец, «выталкивание» и фиксацию в файле базы данных нового состояния соответствующей страницы данных.

Транзакции принято разделять на две разновидности — изменяющие состояние базы данных после завершения транзакции и изменяющие состояние БД лишь временно, с восстановлением исходного состояния данных после завершения транзакции. Совокупность функций СУБД по организации и управлению транзакциями называют *монитором транзакций*.

Транзакции в теории и практике СУБД по отношению к базе данных выступают внешними процессами, отождествляемыми с действиями пользователей банка данных. При этом источником, инициатором транзакций может быть как один пользователь, так и несколько пользователей сразу. По этому критерию СУБД классифицируются на *однопользовательские* и *многопользовательские* СУБД. Как правило, в однопользовательских СУБД монитор транзакций в виде отдельного

функционального элемента СУБД не реализуется. Соответственно в многопользовательских СУБД главной функцией монитора транзакций является обеспечение эффективного совместного выполнения транзакций над общими данными сразу от нескольких пользователей.

Непосредственная обработка и доступ к данным в большинстве СУБД осуществляется через организацию в оперативной памяти штатными средствами операционной системы или собственными средствами системы *буферов оперативной памяти,* куда на время обработки и доступа помещаются отдельные компоненты файла базы данных (страницы). Поэтому другой составной частью функций СУБД по организации доступа и обработки данных является *управление буферами оперативной памяти.*

Еще одной важной функцией СУБД с точки зрения организации доступа и обработки данных является так называемая журнализация всех текущих изменений базы данных. *Журнализация* представляет собой основное средство обеспечения сохранности данных при всевозможных сбоях и разрушениях данных. Во многих СУБД для нейтрализации подобных угроз создается журнал изменений базы данных с особым режимом хранения и размещения.

Резервная копия БД и журнал изменений, как правило, размещаются на отдельных от основного файла БД носителях.

Схематично взаимодействие компонент СУБД представлено на рис. 1.1.

Ядром СУБД является *процессор описания и поддержания структуры базы данных.* Он реализует модель организации данных, средствами которой проектировщик строит *логическую структуру (схему) базы данных,* соответствующую инфологической схеме предметной области КС, и обеспечивает построение и поддержание *внутренней схемы базы данных.*

Процессором описания и поддержания структуры данных в терминах используемой модели данных (иерархическая, сетевая, реляционная) обеспечиваются установки заданной логической структуры базы данных, а

также трансляция (перевод) структуры базы данных во внутреннюю схему базы данных (в физические структуры данных). В КС на базе реляционных СУБД процессор описания и поддержания структуры базы данных реализуется на основе *языка базы данных,* являющегося составной частью *языка структурированных запросов SQL.*

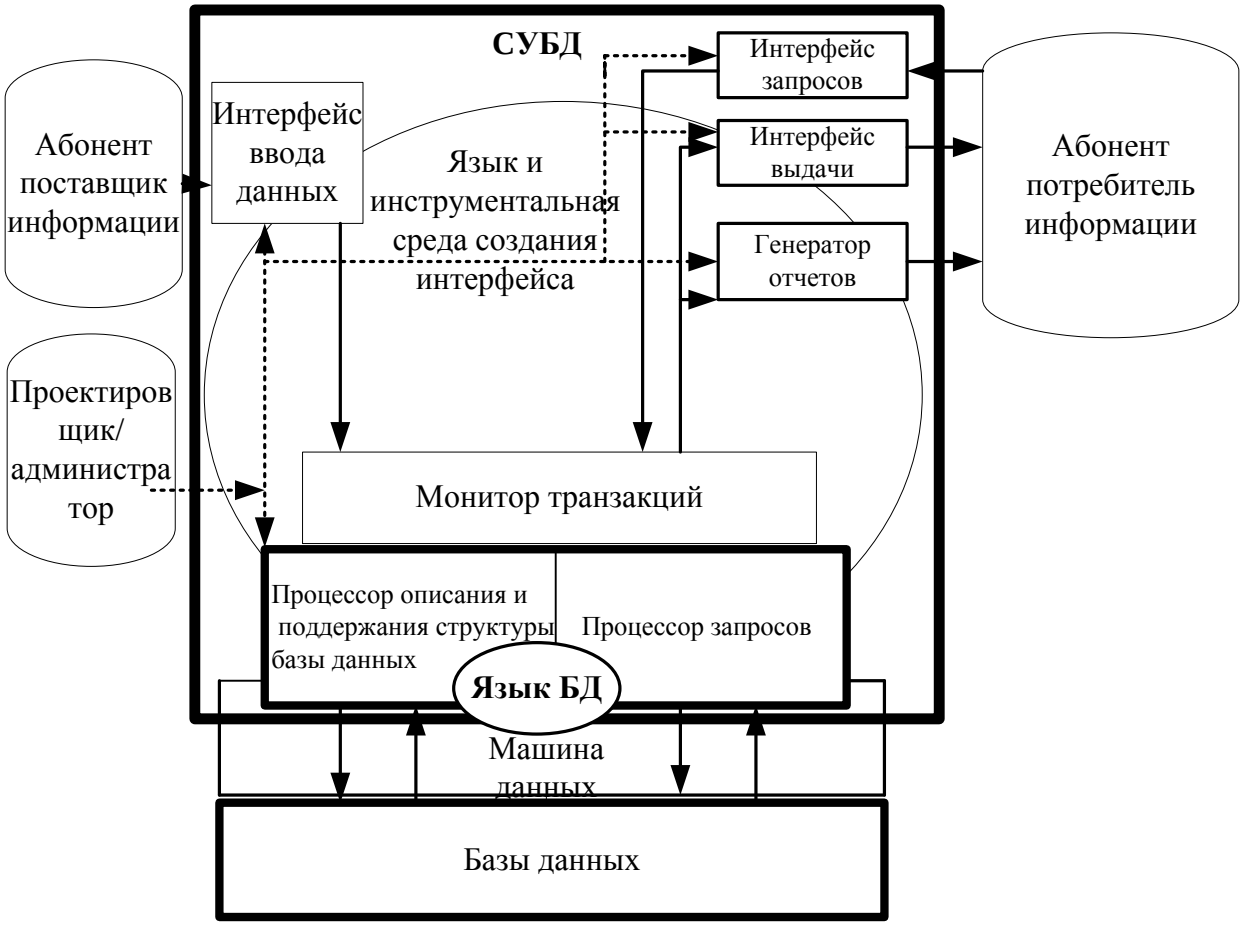


Рис.1.1. Структура и взаимодействие компонент СУБД

Интерфейс ввода данных СУБД реализует *входной информационный язык банка данных,* обеспечивая абонентам-поставщикам информации средства описания и ввода данных в информационную систему. Одной из современных тенденций развития СУБД является стремление приблизить входные информационные языки и интерфейс ввода к естественному языку общения с пользователем в целях упрощения эксплуатации информационных систем так называемых «неподготовленными» пользователями. Данная

проблема решается через применение диалоговых методов организации интерфейса и использование *входных форм.* Входные формы, по сути, представляют собой электронные аналоги различного рода анкет, стандартизованных бланков и таблиц, широко используемых в делопроизводстве и интуитивно понятных большинству людей (неподготовленных пользователей). Интерфейс ввода при этом обеспечивает средства создания, хранения входных форм и их интерпретацию в терминах описания логической структуры базы данных для передачи вводимых через формы сведений процессору описания и поддержания структуры базы данных.

Интерфейс запросов совместно с процессором запросов обеспечивает концептуальную модель использования информационной системы в части стандартных типовых запросов, отражающих информационные потребности пользователей-абонентов системы. Интерфейс запросов предоставляет пользователю средства выражения своих информационных потребностей. Современной тенденцией развития СУБД является использование диалогово-наглядных средств в виде специальных «конструкторов» или пошаговых «мастеров» формирования запросов.

Процессор запросов интерпретирует сформированные запросы в терминах *языка манипулирования данными* и совместно с процессором описания и поддержания структуры базы данных собственно и исполняет запросы. В реляционных СУБД основу процессора запросов составляет язык манипулирования данными, являющийся основной частью языка SQ*L*. Тем самым на базе процессора запросов и процессора описания и поддержания структуры базы данных образуется низший уровень оперирования данными в СУБД, который иногда называют *машиной данных.* Стандартные функции и возможности машины данных используют компоненты СУБД более высокого порядка, что позволяет разделить и стандартизировать компоненты СУБД и банка данных на три уровня —логический уровень, машина данных и собственно сами данные.

Функции *монитора транзакции,* как уже отмечалось, заключаются в организации совместного выполнения транзакций от нескольких пользователей над общими данными. При этом дополнительной функцией, неразрывно связанной, в том числе и с основной функцией, является обеспечение целостности данных и ограничений над данными, определяемыми правилами предметной области КС.

Интерфейс выдачи СУБД получает от процессора запросов результаты исполнения запросов (обращений к базе данных) и переводит эти результаты в форму, удобную для восприятия и выдачи пользователю-абоненту информационной системы. Для отображения результатов исполнения запросов в современных СУБД используются различные приемы, позволяющие «визуализировать» данные в привычной и интуитивно понятной неподготовленному пользователю форме. Обычно для этого применяются табличные способы представления структурированных данных, а также специальные *формы выдачи* данных.

Формы выдачи лежат также и в основе формирования так называемых *«отчетов»,* выдающих результаты поиска и отбора информации из БД в письменной форме для формализованного создания соответствующих текстовых документов, т. е. для документирования выводимых данных. Для подобных целей в состав современных СУБД включаются *генераторы отчетов.*

Современные программные средства, реализующие те или иные СУБД, представляют собой совокупность *инструментальной среды создания и использования базы данных* в рамках определенной модели данных (реляционной, сетевой, иерархической или смешанной) и *языка СУБД* (язык описания данных, язык манипулирования данными, язык и средства создания интерфейса).

Пользователей СУБД можно разделить на три группы:

1. Прикладные программисты - отвечают за создание программ, использующих базу данных. В смысле защиты данных программист может

быть как пользователем, имеющим привилегии создания объектов данных и манипулирования ими, так и пользователем, имеющим привилегии только манипулирования данными.

2. Конечные пользователи базы данных - работают с БД непосредственно через терминал или рабочую станцию. Как правило, конечные пользователи имеют строго ограниченный набор привилегий манипулирования данными. Этот набор может определяться при конфигурировании интерфейса конечного пользователя и не изменяться. Политику безопасности в данном случае определяет администратор безопасности или администратор базы данных (если это одно и то же должностное лицо).

3. Администраторы базы данных - образуют особую категорию пользователей СУБД. Они создают сами базы данных, осуществляют технический контроль функционирования СУБД, обеспечивают необходимое быстродействие системы. В обязанности администратора, кроме того, входит обеспечение пользователям доступа к необходимым им данным, а также написание необходимых пользователю внешних представлений данных. Администратор определяет правила безопасности и целостности данных.

Модели безопасности данных. Модель безопасности включает:

• модель компьютерной (информационной) системы;

• критерии, принципы, ограничения и целевые функции защищенности информации от угроз;

• формализованные правила, ограничения, алгоритмы, схемы и механизмы безопасного функционирования системы.

В основе большинства моделей безопасности лежит субъектно-объектная модель компьютерных систем, в том числе и баз данных как ядра автоматизированных информационных систем. База данных КС разделяется на субъекты базы данных (активные сущности), объекты базы данных (пассивные сущности) и порождаемые действиями субъектов процессы над объектами (рис.1.2).

Определяются два основополагающих принципа безопасности функционирования информационных систем:

- персонализация (идентификация) и аутентификация (подтверждение подлинности) всех субъектов и их процессов по отношению к объектам;
- разграничение полномочий субъектов по отношению к объектам, и обязательная проверка полномочий любых процессов над данными.

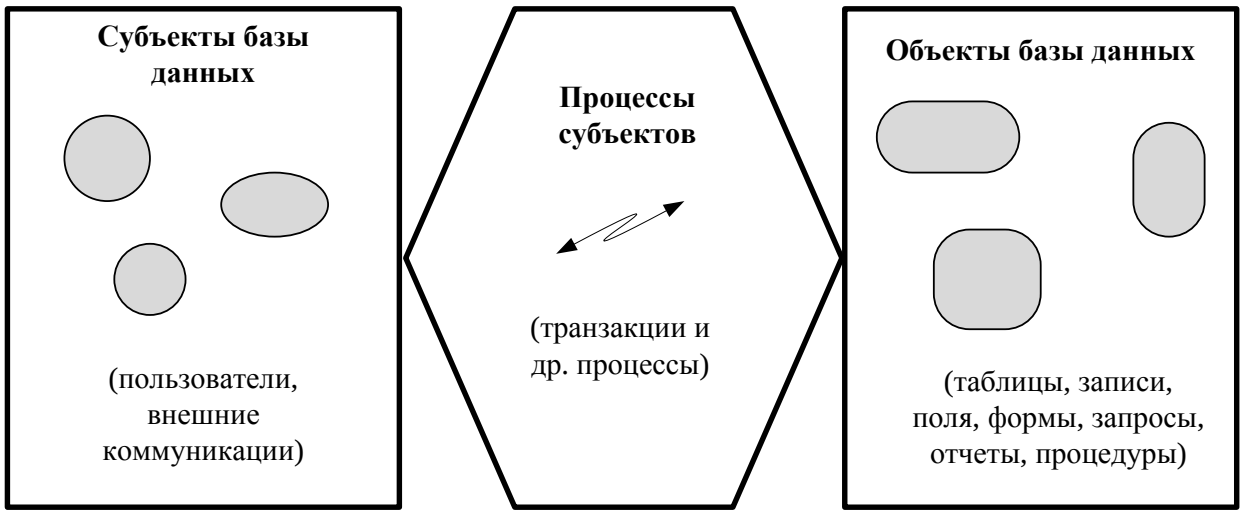


Рис.1.2. База данных КС в моделях безопасности данных

Соответственно в структуре ядра СУБД выделяется дополнительный компонент, называемый монитором (сервером, менеджером, ядром) безопасности (Trusted Computing Base - TCB), который реализует определенную политику безопасности во всех процессах обработки данных. Если в схемотехническом аспекте компьютерную систему представить как совокупность ядра, включающего компоненты представления данных и доступа (манипулирования) к данным, а также надстройки, которая реализует интерфейсные и прикладные функции, то роль и место монитора безопасности можно проиллюстрировать схемой, приведенной на рис.1.3.

В узком смысле политика безопасности, реализуемая монитором безопасности компьютерной системы, собственно и определяет модель безопасности (вторая и третья компоненты).

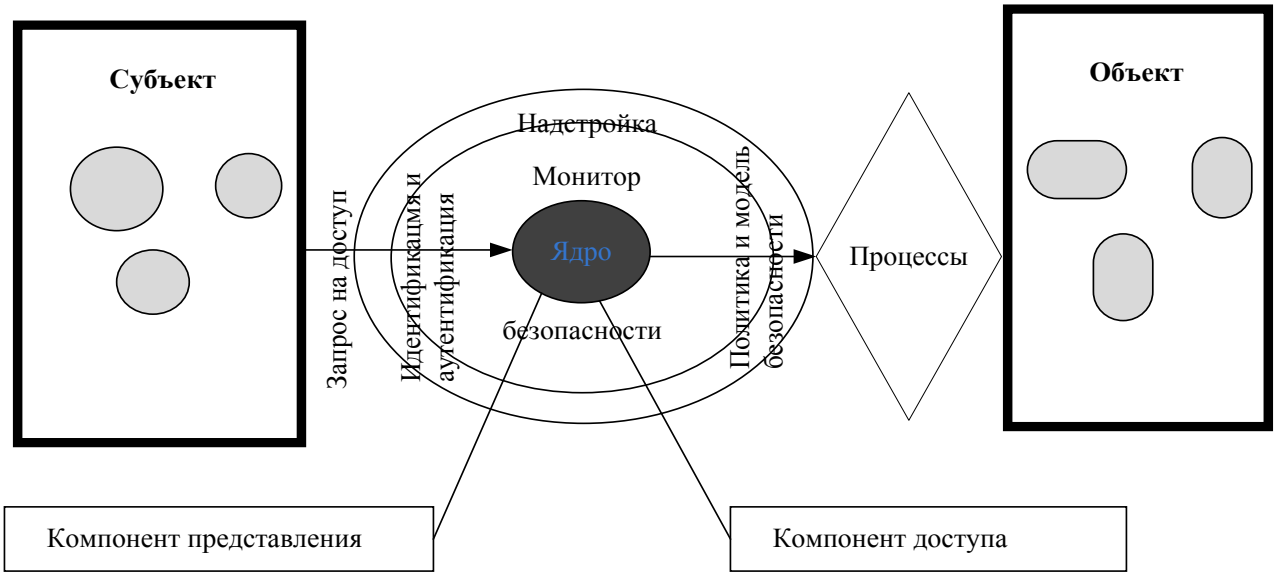


Рис. 1.3. Схематический аспект защиты информации в компьютерных системах

Простейшая (одноуровневая) модель безопасности данных строится на основе дискреционного (избирательного) принципа разграничения доступа, при котором доступ к объектам осуществляется на основе множества разрешенных отношений доступа в виде троек — «субъект доступа – тип доступа – объект доступа». Наглядным и распространенным способом формализованного представления дискреционного доступа является матрица доступа, устанавливающая перечень пользователей (субъектов) и перечень разрешенных операций (процессов) по отношению к каждому объекту базы данных (таблицы, запросы, формы, отчеты). На рис. 1.4 приведен пример, иллюстрирующий матрицу доступа.

Важным аспектом моделей безопасности является управление доступом. Существует два подхода:

• добровольное управление доступом;

• принудительное управление доступом.

При добровольном управлении доступом вводится так называемое владение объектами. Добровольное управление доступом заключается в том, что права на доступ к объектам определяют их владельцы. Иначе говоря, соответствующие ячейки матрицы доступа заполняются теми субъектами

(пользователями), которым принадлежат права владения над соответствующими объектами базы данных. В большинстве систем права владения объектами могут передаваться. В результате при добровольном управлении доступом реализуется полностью децентрализованный принцип организации и управления процессом разграничения доступа.

		ТАБЛИЦЫ				
		Сотрудник Установленные данные	Сотрудник Конфиденциальные данные	Операции	Командировки	Задания
Пользователи	Каримов	Ч, М				
	Салимов	Ч	Ч	Ч, С, М	Ч, С, М	Ч, С, М
	Аълоев	Ч, М, С, У	Ч, М, С, У			
	Рузиев	Ч, М, С, У	Ч, М, С, У	Ч, М, С, У	Ч, М, С, У	Ч, М, С, У

Обозначения:

Ч – чтение;

М – модификация;

С – создание;

У – удаление (записей)

Рис. 1.4. Модель безопасности на основе матрицы доступа (дискреционный принцип разграничения доступа)

Такой подход обеспечивает гибкость настраивания системы разграничения доступа в базе данных на конкретную совокупность пользователей и ресурсов, но затрудняет общий контроль и аудит состояния безопасности данных в системе.

Принудительный подход к управлению доступом предусматривает введение единого централизованного администрирования доступом. В базе данных выделяется специальный доверенный субъект (администратор), который (и только он), собственно, и определяет разрешения на доступ всех остальных субъектов к объектам базы данных. Иначе говоря, заполнять и изменять ячейки матрицы доступа может только администратор системы.

Принудительный способ обеспечивает более жесткое централизованное управление доступом. Вместе с тем он является менее гибким и менее точным в плане настройки системы разграничения доступа на потребности и полномочия пользователей, так как наиболее полное представление о содержимом и конфиденциальности объектов (ресурсов) имеют, соответственно, их владельцы.

На практике может применяться комбинированный способ управления доступом, когда определенная часть полномочий на доступ к объектам устанавливается администратором, а другая часть владельцами объектов.

Исследования различных подходов к обеспечению информационной безопасности в традиционных (некомпьютерных) сферах и технологиях показали, что одноуровневой модели безопасности данных недостаточно для адекватного отражения реальных производственных и организационных схем. В частности, традиционные подходы используют категорирование информационных ресурсов по уровню конфиденциальности (совершенно секретно - СС, секретно - С, конфиденциально - К, и т. п.). Соответственно субъекты доступа к ним (сотрудники) также категорируются по соответствующим уровням доверия, получая так называемого допуска (допуск степени 1, допуск степени 2 и т. д.). Понятие допуска определяет мандатный (полномочный) принцип разграничения доступа к информации. В соответствии с мандатным принципом работник, обладающий допуском степени «1», имеет право работать с любой информацией уровня «СС», «С» и «К». Работник с допуском «2» соответственно имеет право работы с любой информацией уровня «С» и «К». Работник с допуском «3» имеет право работать с любой информацией только уровня «К».

Мандатный принцип построения системы разграничения доступа в СУБД реализует многоуровневую модель безопасности данных, называемую еще моделью Белл - ЛаПадула (по имени ее авторов - американских специалистов Д. Белл и Л. ЛаПадула), которая иллюстрируется схемой, приведенной на рис. 1.5.

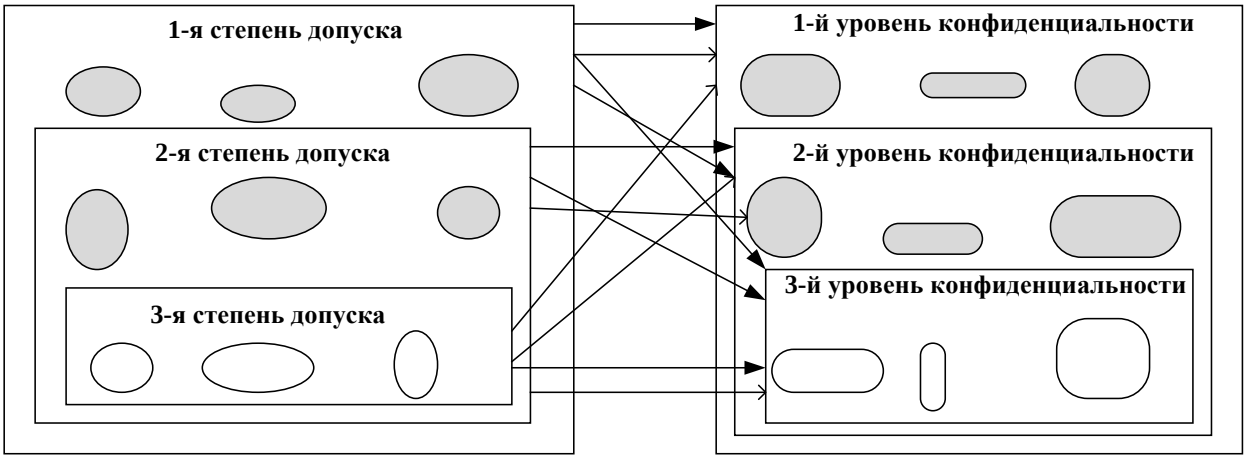


Рис. 1.5. Модель безопасности данных Белл - ЛаПадула (мандатный принцип разграничения доступа)

В модели Белл - ЛаПадула объекты и субъекты категорируются по иерархическому мандатному принципу доступа. Субъект, имеющий допуск 1-й (высшей) степени, получает доступ к объектам 1-го (высшего) уровня конфиденциальности и автоматически ко всем объектам более низких уровней конфиденциальности (т. е. к объектам 2-го и 3-го уровней). Соответственно, субъект со 2-й степенью допуска имеет доступ ко всем объектам 2-го и 3-го уровней конфиденциальности, и т. д.

В модели Белл - ЛаПадула устанавливаются и поддерживаются два основных ограничения политики безопасности:

• запрет чтения вверх (no read up - NRU);

• запрет записи вниз (no write down - NWD).

Ограничение NRU является логическим следствием мандатного принципа разграничения доступа, запрещая субъектам читать данные из объектов более высокой степени конфиденциальности, чем позволяет их допуск.

Ограничение NWD предотвращает перенос (утечку) конфиденциальной информации путем ее копирования из объектов с высоким уровнем конфиденциальности в не конфиденциальные объекты или в объекты с меньшим уровнем конфиденциальности.

На практике в реальных политиках мониторов безопасности баз данных чаще всего применяется дискреционный принцип с принудительным управлением доступом, «усиливаемый» элементами мандатного принципа в сочетании с добровольным управлением доступом (допуска субъектов устанавливает и изменяет только администратор, уровень конфиденциальности объектов устанавливают и изменяют только владельцы).

Контрольные вопросы

1. Перечислите составные элементы понятия «информационная безопасность».
2. Какие функции реализует система управления базами данных?
3. Какие модели организации данных вы знаете?
4. Объясните понятие «транзакция».
5. Объясните структуру и взаимодействие компонент системы управления базами данных.
6. Что включает модель безопасности данных?
7. Объясните принципы дискреционного и мандатного разграничения доступа в системе управления базами данных.

1.2. Виды систем управления базами данных

По языкам общения СУБД делятся на открытые, замкнутые и смешанные. Открытые системы - это системы, в которых для обращения к базам данных используются универсальные языки программирования. Замкнутые системы имеют собственные языки общения с пользователями БД.

По числу уровней в архитектуре различают одноуровневые, двухуровневые, трехуровневые системы. В принципе возможно выделение и большего числа уровней. Под архитектурным уровнем СУБД понимают функциональный компонент, механизмы которого служат для поддержки

некоторого уровня абстракции данных (логический и физический уровень, а также «взгляд» пользователя - внешний уровень) (рис.1.6).

По выполняемым функциям СУБД делятся на информационные и операционные. Информационные СУБД позволяют организовать хранение информации и доступ к ней. Для выполнения более сложной обработки необходимо писать специальные программы. Операционные СУБД выполняют достаточно сложную обработку, например, автоматически позволяют получать агрегированные показатели, не хранящиеся непосредственно в базе данных, могут изменять алгоритмы обработки и т.д.

По сфере возможного применения различают универсальные и специализированные, обычно проблемно-ориентированные СУБД.

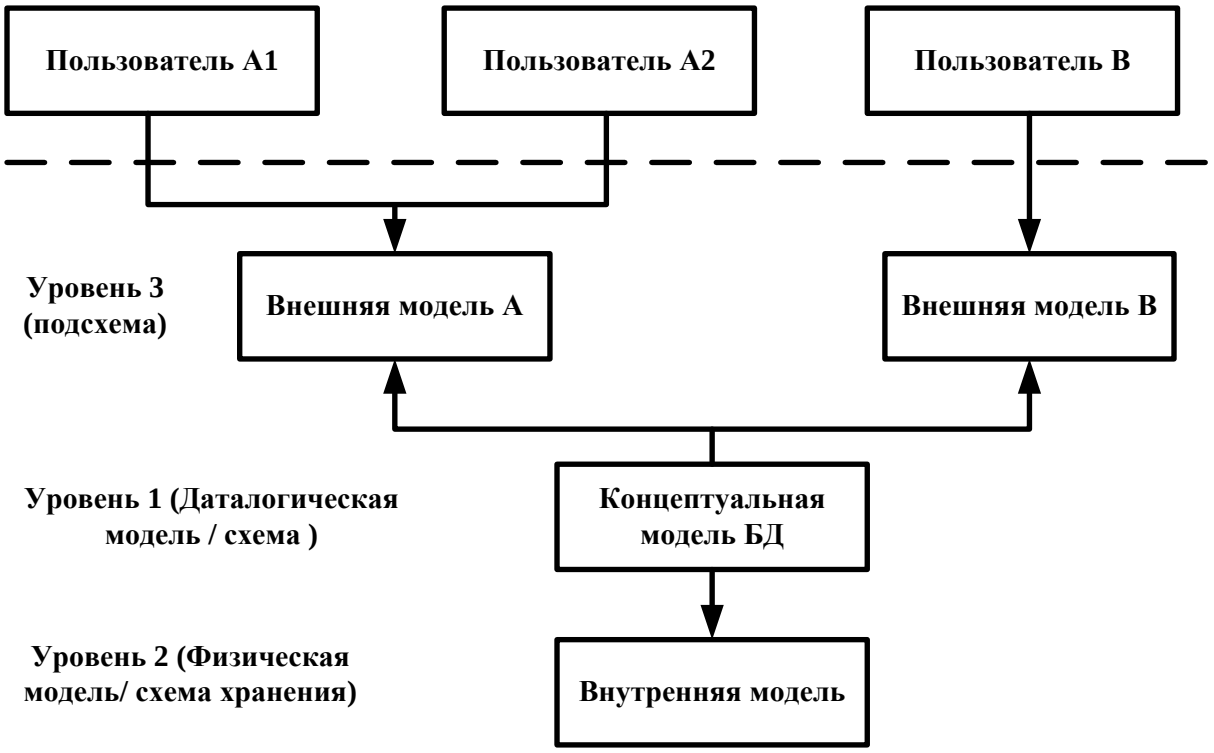


Рис. 1.6. Классификация СУБД по числу уровней в архитектуре (пример трехуровневой архитектуры)

Системы управления базами данных поддерживают разные типы данных. Набор типов данных, допустимых в разных СУБД, различен. Кроме того, ряд СУБД позволяет разработчику добавлять новые типы данных и

новые операции над этими данными. Такие системы называются расширяемыми системами баз данных (РСБД).

Дальнейшим развитием концепции РСБД являются системы объектно-ориентированных баз данных, обладающие достаточно мощными выразительными возможностями, чтобы непосредственно моделировать сложные объекты.

По мощности СУБД делятся на настольные и корпоративные. Характерными чертами настольных СУБД являются сравнительно невысокие требования к техническим средствам, ориентация на конечного пользователя, низкая стоимость.

Корпоративные СУБД обеспечивают работу в распределенной среде, высокую производительность, поддержку коллективной работы при проектировании систем, имеют развитые средства администрирования и более широкие возможности поддержания целостности. Эти системы сложны, дороги, требуют значительных вычислительных ресурсов.

Сравнительные характеристики настольных и корпоративных СУБД приведены в табл. 1.2.

Системы обоих классов интенсивно развиваются, причем некоторые тенденции развития присущи каждому из этих классов. Прежде всего, это использование высокоуровневых средств разработки приложений (что раньше было присуще, в основном, настольным системам), рост производительности и функциональных возможностей, работа в локальных и глобальных сетях и др.

Наиболее известными из корпоративных СУБД являются Oracle, DB2, Sybase, MS SQL Server, Progress и некоторые другие.

По ориентации на преобладающую категорию пользователей можно выделить СУБД для разработчиков и для конечных пользователей. Системы, относящиеся к первому классу, должны иметь качественные компиляторы и позволять создавать «отчуждаемые» программные продукты, обладать развитыми средствами отладки, включать средства документирования

проекта и другие возможности, позволяющие строить эффективные сложные системы. Основными требованиями, предъявляемыми к системам, ориентированным на конечного пользователя, являются: удобство интерфейса, высокий уровень языковых средств, наличие интеллектуальных модулей подсказок, повышенная защита от непреднамеренных ошибок и т.п.

Таблица 1.2

Критерий	Настольные	Корпоративные
Простота использования	+	
Стоимость программного обеспечения	+	
Стоимость эксплуатации	+	
Функциональные возможности: администрирование, работа с Интернет/интранет и др.		+
Надежность функционирования		+
Поддерживаемые объемы данных		+
Быстродействие		+
Возможности масштабирования		+
Работа в гетерогенной среде		+

Существует разделение СУБД по поколениям. К первому поколению СУБД относят системы, основанные на иерархической и сетевой моделях (60-70-е гг. XX в.), ко второму поколению - реляционные системы. СУБД третьего поколения должны поддерживать сложные структуры данных и более развитые средства обеспечения целостности данных, отвечать требованиям, предъявляемым к открытым системам.

Контрольные вопросы

1. Какие виды СУБД различают по языкам общения?
2. Какие виды СУБД различают по числу уровней в архитектуре?
3. Какие виды СУБД различают по мощности?
4. Сравнительные характеристики настольных и корпоративных СУБД.

1.3. Технологические аспекты информационной безопасности базы данных

Практическая реализация политик и моделей безопасности приводящийся выше, а также аксиоматических принципов построения и функционирования защищенных информационных систем обусловливает необходимость решения ряда программно-технологических задач, которые можно сгруппировать по следующим направлениям:

- технологии идентификации и аутентификации;
- языки безопасности баз данных;
- технологии обеспечения безопасности повторного использования объектов;
- технологии надежного проектирования и администрирования.

1.3.1. Технологии идентификации и аутентификации

Технологии идентификации и аутентификации являются обязательным элементом защищенных систем, так как обеспечивают аксиоматический принцип персонализации субъектов и, тем самым, реализуют первый (исходный) программно-технический рубеж защиты информации в компьютерных системах.

Под идентификацией понимается различение субъектов, объектов, процессов по их образам, выражаемым именами.

Под аутентификацией понимается проверка и подтверждение подлинности образа идентифицированного субъекта, объекта, процесса.

В системотехническом плане структуру систем идентификации/аутентификации можно проиллюстрировать схемой, приведенной на рис. 1.7.

При регистрации объекта идентификации/аутентификации в системе монитором безопасности формируется его образ, информация по которому

подвергается необратимому без знания алгоритма и шифра-ключа, т. е. криптографическому преобразованию и сохраняется в виде ресурса, доступного в системе исключительно монитору безопасности. Таким образом формируется информационный массив внутренних образов объектов идентификации/аутентификации.

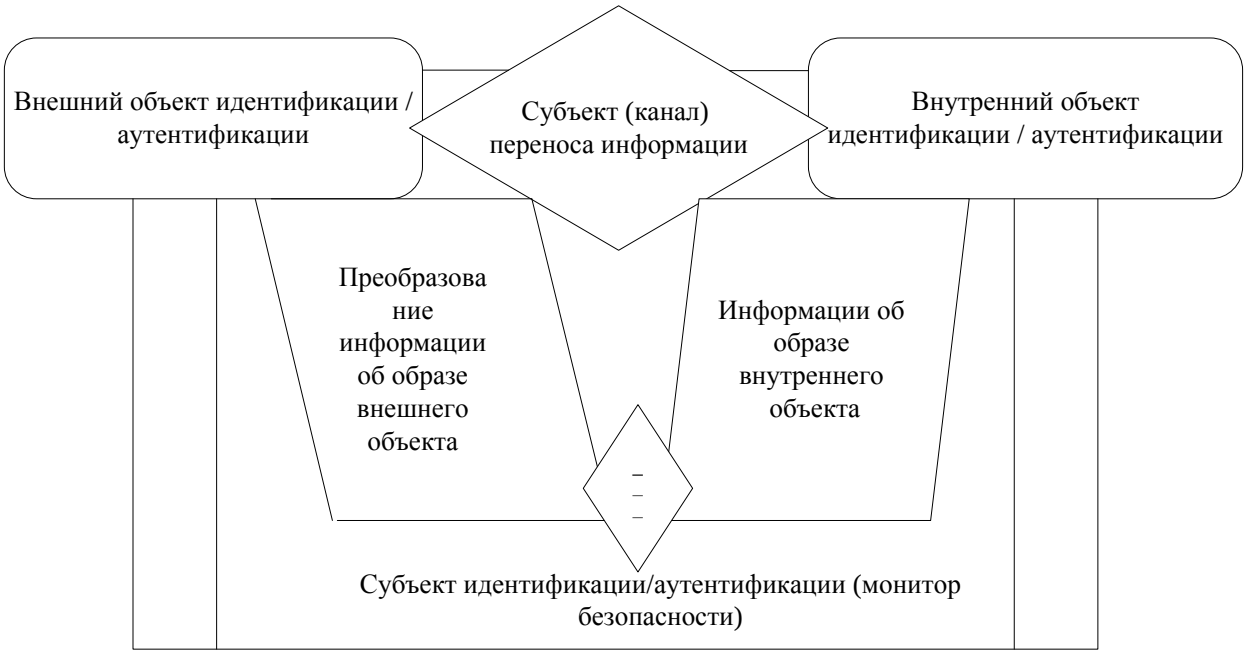


Рис.1.7. Системотехнический аспект идентификации/аутентификации

Впоследствии при идентификации/аутентификации (очередной вход в систему пользователя, запрос процесса на доступ к объекту, проверка подлинности объекта системы при выполнении над ним действий и т. д.) объект через канал переноса информации передает монитору безопасности информацию о своем образе, которая подвергается соответствующему преобразованию. Результат этого преобразования сравнивается с соответствующим зарегистрированным внутренним образом, и при их совпадении принимается решение о распознавании (идентификации) и подлинности (аутентификации) объекта.

Информационный массив внутренних образов объектов идентификации/аутентификации является критическим ресурсом системы,

несанкционированный доступ к которому дискредитирует всю систему безопасности. Поэтому помимо всевозможных мер по исключению угроз несанкционированного доступа к нему сама информация о внутренних образах объектов идентификации/аутентификации находится в зашифрованном виде.

В общем плане для идентификации/аутентификации пользователей субъектов в компьютерных системах могут использоваться их биометрические параметры (отпечатки пальцев, рисунок радужной оболочки глаз, голос, почерк и т. д.), либо специальные замково-ключевые устройства (смарт-карты, магнитные карты и т. п.). Однако при доступе непосредственно в КС (в базы данных), чаще всего используются парольные системы идентификации/аутентификации.

Парольные системы основаны на предъявлении пользователем в момент аутентификации специального секретного (известного только подлинному пользователю) слова или набора символов - пароля. Пароль вводится пользователем с клавиатуры, подвергается криптопреобразованию и сравнивается со своей зашифрованной соответствующим образом учетной копией в системе. При совпадении внешнего и внутреннего парольного аутентификатора осуществляется распознавание и подтверждение подлинности соответствующего субъекта.

Парольные системы являются простыми, но при условии правильной организации подбора и использования паролей, в частности, безусловного сохранения пользователями своих паролей втайне, достаточно надежным средством аутентификации, и, в силу данного обстоятельства, широко распространены.

Основной недостаток систем парольной аутентификации заключается в принципиальной оторванности, отделимости аутентификатора от субъекта-носителя. В результате пароль может быть получен тем или иным способом от законного пользователя или просто подобран, подсмотрен по набору на

клавиатуре, перехвачен тем или иным способом в канале ввода в систему и предъявлен системе злоумышленником.

Поэтому в некоторых случаях парольные аутентификаторы могут усиливаться диалогово-вопросными системами «коллективного вхождения». В системах коллективного вхождения парольную аутентификацию должны одновременно пройти сразу все зарегистрированные для работы в системе пользователи. Иначе говоря, поодиночке пользователи работать в системе не могут. Вероятность подбора, перехвата и т. д. злоумышленником (злоумышленниками) сразу всех паролей, как правило, существенно меньше, и, тем самым, надежность подобных систем аутентификации выше.

Аутентификации в распределенных информационных системах в принципе должны подвергаться и объекты (ресурсы, устройства), а также процессы (запросы, пакеты и т. д.). Аутентифицированный (подлинный) пользователь, обращаясь к объектам системы и порождая соответствующие процессы, должен, в свою очередь, убедиться в их подлинности, например, отправляя распечатать сформированный в базе данных конфиденциальный отчет на сетевой принтер, специально предназначенный для распечатки соответствующих конфиденциальных документов.

Для аутентификации процессов широкое распространение нашли технологии меток (дескрипторов) доступа.

Технология меток или дескрипторов доступа отражает сочетание одноуровневой и многоуровневой моделей безопасности данных и основывается на присвоении администратором системы всем объектам и субъектам базы данных специальных дескрипторов доступа, содержащих набор параметров уровня конфиденциальности, допустимых операций, допустимых имен объектов или субъектов доступа и других особых условий доступа. Субъект доступа, инициируя в соответствии со своим дескриптором (меткой) разрешенный процесс, передает ему свою метку доступа (помечает своей меткой). Ядро безопасности СУБД проверяет подлинность метки процесса, сравнивая ее с меткой доступа пользователя-субъекта, от имени

которого выступает процесс. При положительном результате метка доступа процесса сравнивается с меткой доступа объекта, операцию с которым намеревается осуществлять процесс. Если дескрипторы доступа процесса и объекта совпадают, монитор безопасности разрешает соответствующий доступ, т. е. разрешает осуществление процесса (операции).

Проверка подлинности метки процесса предотвращает возможные угрозы нарушения безопасности данных путем формирования субъектом для инициируемого им процесса такой метки, которая не соответствует его полномочиям.

Для проверки подлинности меток в системе формируется специальный файл (массив) учетных записей. При регистрации нового пользователя в системе для него создается учетная запись, содержащая его идентификационный номер (идентификатор), парольный аутентификатор и набор дескрипторов доступа к объектам базы данных (метка доступа). При инициировании пользователем (субъектом) какого-либо процесса в базе данных и передаче ему своей метки доступа ядро безопасности СУБД подвергает метку процесса криптопреобразованию, сравнивает ее с зашифрованной меткой соответствующего субъекта (пользователя) в массиве учетных записей и выносит решение о подлинности метки.

Массив учетных записей, в свою очередь, является объектом высшей степени конфиденциальности в системе, и доступен только администратору. Ввиду исключительной важности массива учетных записей для безопасности всей системы, помимо шифрования его содержимого, принимается ряд дополнительных мер к его защите, в том числе специальный режим его размещения, проверка его целостности. Таким образом, на сегодняшний день наработан и используется развитый набор технологий идентификации/аутентификации в защищенных компьютерных системах. Вместе с тем, основные бреши безопасности чаще всего находятся злоумышленниками именно на этом пути.

1.3.2. Языки безопасности базы данных

Для определения конкретных назначений или установления правил и ограничений доступа при проектировании банков данных КС, а также в целях управления системой разграничения доступа и в более широком смысле системой коллективной обработки данных администратору системы необходим специальный инструментарий. Такой инструментарий должен основываться на определенном языке, позволяющем описывать и устанавливать те или иные назначения доступа и другие необходимые установки политики безопасности в конкретной КС.

Как следует из рассмотрения внутренней схемы баз данных, одной из основных функций систем управления базами данных является создание и поддержание собственной системы размещения и обмена данными между внешней (дисковой) и оперативной памятью. От эффективности реализации в каждой конкретной СУБД данной функции (формат файлов данных, индексирование, хэширование и буферизация) во многом зависит и эффективность функционирования СУБД в целом. Поэтому основные усилия создателей первых СУБД в конце 60-х — начале 70-х годов были сосредоточены именно в этом направлении. В результате для реализации любой функции по вводу, обработке или выводу данных требовались квалифицированные программисты для написания специальных программ на алгоритмических языках высокого уровня (в 70-х годах ФОРТРАН, КОБОЛ и др.), «знающих» особенности структуры и способы размещения данных во внешней и оперативной памяти. В итоге работа с базами данных осуществлялась через посредника в виде квалифицированного программиста, «переводящего» информационные потребности пользователя в машинный код, что схематично иллюстрируется на рис. 1.8.

Такое положение дел приводило к большим накладным расходам при создании и эксплуатации автоматизированных информационных систем и в определенной степени сдерживало распространение вычислительной

техники в процессах информационного обеспечения деятельности предприятий и организаций.

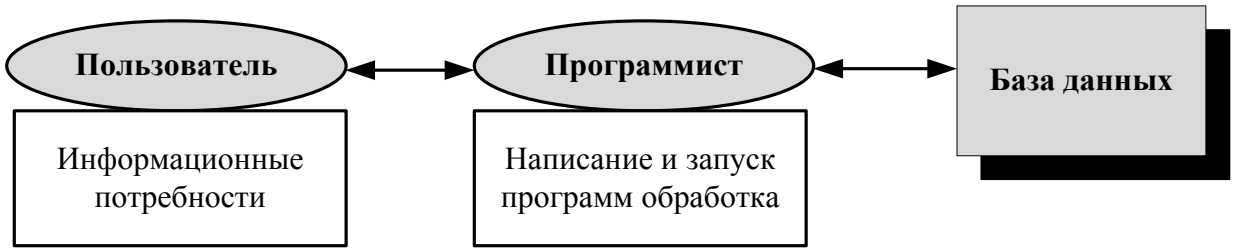


Рис.1.8. Схема взаимодействия пользователя с базой данных в ранних СУБД.

Основателем теории реляционных СУБД Е. Коддом было выдвинуто предложение о создании специального языка для общения (взаимодействия) пользователя-непрограммиста с базами данных. Идея такого языка сводилась к набору из нескольких фраз-примитивов английского языка («выбрать», «обновить», «вставить», «удалить»), через которые пользователь-непрограммист ставил бы «вопросы» к СУБД по своим информационным потребностям. В этом случае дополнительной функцией СУБД должна быть интерпретация этих «вопросов» на низкоуровневый язык машинных кодов для непосредственной обработки данных и предоставление результатов пользователю. Так родилась уже упоминавшаяся по структуре СУБД «машина данных». Иначе говоря, машина данных «понимает» язык базы данных и в результате разделяет собственно данные и задачи по их обработке. В таком подходе взаимодействие пользователя с базой данных можно проиллюстрировать схемой, приведенной на рис. 1.9.

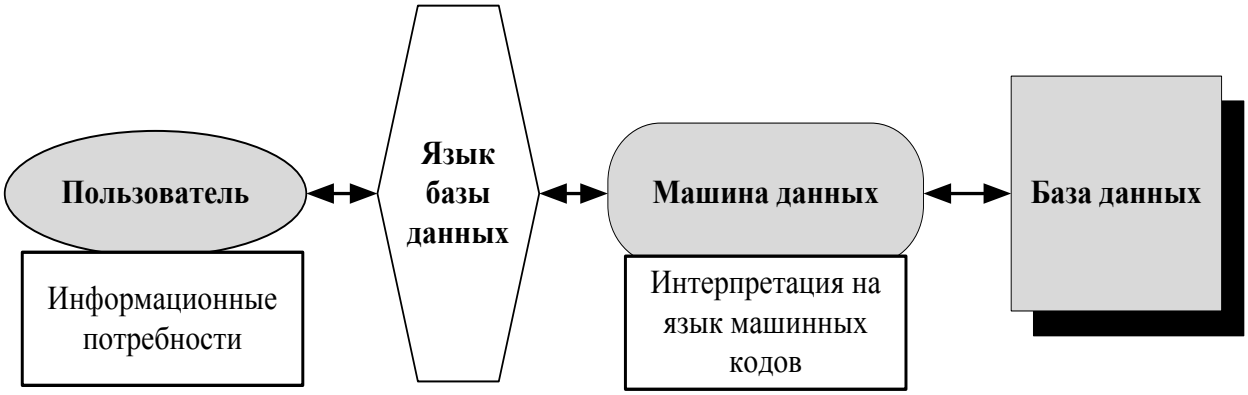


Рис. 1.9. Схема взаимодействия пользователя с базой данных через язык баз данных.

В практику эти идеи впервые претворились в ходе реализации проекта System R (1975-1979 гг.) с участием еще одного известного специалиста по базам данных Криса Дейта. В ходе проекта System R был создан язык SEQUEL, трансформировавшийся впоследствии в язык структурированных запросов SQL (Structured Query Language). При этом дополнительно к возможностям формирования «вопросов» к базе данных пользователю также решено было предоставить и возможность описания самой структуры данных, ввода данных и их изменения. Примерно в то же время в компании IBM был создан еще один реляционный язык-QBE (Query-By-Example), т. е. язык запросов по образцу, применявшийся впоследствии во многих коммерческих системах обработки табличных данных и послуживший идеологической основой для создания визуальных «конструкторов» запросов в современных СУБД.

Быстрое и массовое распространение языка SQL в реляционных СУБД к середине 80-х годов привело фактически к принятию его в качестве стандарта по организации и обработке данных. В 1986 г. Американским национальным институтом стандартов (ANSI) и Международной организацией по стандартизации (ISO) язык был признан стандартным языком описания и обработки данных в реляционных СУБД. В 1989 г. ANSI/ISO была принята усовершенствованная версия SQL — SQL2, а в 1992 г. третья версия — SQL3.

Язык SQL относится к так называемым декларативным (непроцедурным) языкам программирования. В отличие от процедурных языков (С, Паскаль, Фортран, Кобол, Бейсик) на нем формулируются предложения (инструкции) о том, «что сделать», но не «как сделать, как получить». Машина данных в СУБД исполняет роль интерпретатора и как раз строит машинный код, реализующий способ получения результата, задаваемого SQL-инструкциями.

Язык SQL состоит из двух частей:

- языка описания (определения) данных — DDL (Data Definition Language);

- языка манипулирования данными — DML (Data Manipulation Language).

Синтаксис SQL-инструкций включает:

- название инструкции (команду);

- предложения, определяющие источники, условия операции;

- предикаты, определяющие способы и режимы отбора записей, задаваемых предложениями;

- выражения, значения которых задают свойства и параметры выполнения инструкции и предложения.

Структуру SQL-инструкций можно разделить на две основные части, схематично представленные на рис. 1.10.

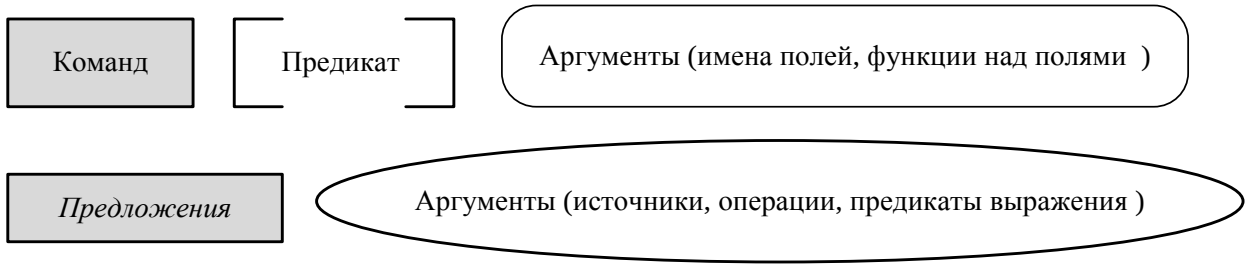


Рис. 1.10. Структура SQL-инструкций.

Первая часть включает название (команду) SQL-инструкции, предикат (необязательный элемент) и аргументы инструкции, которыми являются перечисляемые через запятую имена полей одной или нескольких таблиц.

Вторая часть состоит из одного или нескольких предложений, аргументы которых могут задавать источники данных (имена таблиц, операции над таблицами), способы, условия и режимы выполнения команды (предикаты сравнения, логические и математические выражения по значениям полей таблиц). Перечень SQL-инструкций разделяется по частям языка SQL.

В состав языка DDL входят несколько базовых инструкций, обеспечивающих основной набор функций при создании реляционных таблиц и связей между ними.

CREATETABLE... — создать таблицу;

CREATEINDEX... — создать индекс;

ALTERTABLE... — изменить структуру ранее созданной таблицы;

DROP... — удалить существующую таблицу и базы данных.

В структуре инструкций CREATETABLE и ALTERTABLE важную роль играет предложение CONSTRAINT (создать ограничения на значения данных) со следующими установками — NOT NULL (не допускаются нулевые значения по соответствующему полю), AUTOINC (поле с инкрементальным, т. е. последовательно возрастающим с каждой новой записью, характером значений) и PRIMARY KEY (определение для поля уникального, т. е. без повторов индекса, что в результате задает режим заполнения данного поля с уникальными неповторяющимися по различным строкам значениями).

В состав языка DML также входят несколько базовых инструкций, охватывающих тем не менее основные операции по вводу, обработке и выводу данных.

SELECT... — выбрать данные из базы данных;

INSERT... — добавить данные в базу данных;

UPDATE... — обновить данные в базе данных;

DELETE... — удалить данные;

GRANT... — предоставить привилегии пользователю;

REVOKE... — отменить привилегии пользователю;

COMMIT... — зафиксировать текущую транзакцию;

ROLLBACK... — прервать текущую транзакцию.

Важное значение имеют разновидности инструкции SELECT— SELECT... INTO ... (выбрать из одной или нескольких таблиц набор записей, из которого создать новую таблицу) и UNION SELECT, которая в

дополнении с исходной инструкцией SELECT (SELECT... UNION SELECT...) реализует операцию объединения таблиц.

Помимо предложения CONSTRAINT в SQL-инструкциях используются следующие предложения:

FROM... — указывает таблицы или запросы, которые содержат поля, перечисленные в инструкции SELECT;

WHERE... — определяет, какие записи из таблиц, перечисленных в предложении FROM, следует включить в результат выполнения инструкции SELECT, UPDATE или DELETE;

GROUP BY... — объединяет записи с одинаковыми значениями в указанном списке полей в одну запись;

HAVING... — определяет, какие сгруппированные записи отображаются при использовании инструкции SELECT с предложением GROUP BY;

IN... — определяет таблицы в любой внешней базе данных, с которой ядро СУБД может установить связь;

ORDERBY... — сортирует записи, полученные в результате запроса, в порядке возрастания или убывания на основе значений указанного поля или полей.

В качестве источника данных по предложению FROM, помимо таблиц и запросов, могут использоваться также результаты операций соединения таблиц в трех разновидностях—INNER JOIN... ON..., LEFT JOIN. ..ON... и RIGHT JOIN...ON... (внутреннее соединение, левое и правое внешнее соединение, соответственно).

Предикаты используются для задания способов и режимов использования записей, отбираемых на основе условий в инструкции SQL. Такими предикатами являются:

ALL... — отбирает все записи, соответствующие условиям, заданным в инструкции SQL;

DISTINCT... — исключает записи, которые содержат повторяющиеся значения в выбранных полях;

DISTINCTROW... — опускает данные, основанные на целиком повторяющихся записях;

TOP... — возвращает записей, находящихся в начале или в конце диапазона, описанного с помощью предложения ORDER BY;

Выражениями в инструкциях SQL являются любые комбинации операторов, констант, значений текстовых констант, функций, имен полей, построенные по правилам математических выражений и результатом которых является конкретное, в том числе и логическое значение.

Базовых инструкций языка SQL представлены инструкции GRANT и REVOKE, предоставляющие или отменяющие привилегии пользователям. Структура инструкции GRANT выглядит следующим образом:

GRANTсписок_привилегий_через_запятую ON ИмяОбъекта

ТОИменаПользователей_через_запятую

[WITHGRANTOPTION];

где:

- список привилегий составляют разрешенные инструкции (операции) над объектом (таблицей) - SELECT, INSERT, UPDATE, DELETE;
- список пользователей представляется их именами-идентификаторами или может быть заменен ключевым словом PUBLIC, которое идентифицирует всех пользователей, зарегистрированных в системе;
- директива WITH GRANT OPTION наделяет перечисленных пользователей дополнительными особыми полномочиями по предоставлению указанных в списке привилегий-полномочий другим пользователям.

В большинстве случаев право подачи команд GRAND и REVOKE по конкретному объекту автоматически имеют пользователи, создавшие данный объект, т. е. их владельцы. В других подходах этим правом наделяются доверенные субъекты, т. е. администраторы.

Хотя в явном виде такой подход не предусматривает создание матрицы доступа, тем не менее, реализуется классический принцип дискреционного разграничения доступа с сочетанием как добровольного, так и принудительного управления доступом.

На самом деле в большинстве СУБД привилегии и установки доступа, как и структура базы данных, «прописываются» в системных таблицах БД, т.е. в системном каталоге БД, который можно рассматривать, в том числе и в качестве матрицы доступа.

Как уже отмечалось, дискреционный принцип обладает большой гибкостью по настройке системы разграничения доступа на особенности предметной области базы данных и потребности пользователей, но не обеспечивает эффективной управляемости и затрудняет проведение какой либо целенаправленной политики безопасности в системе. Преодоление этого недостатка достигается двумя путями использованием техники «представлений» и специальными расширениями языка SQL.

«Представлением» называется глобальный авторизованный запрос на выборку данных, формирующий для пользователя «свое» представление определенного объекта (объектов), совокупность которых формирует некую виртуальную базу данных, со своей схемой (объектами) и данными (отобранными или специально преобразованными). При входе пользователя в систему в процессе его идентификации и аутентификации ядро безопасности отыскивает для пользователя соответствующие представления-запросы и передает запрос основному ядру СУБД для выполнения. В результате выполнения запроса пользователь «видит» и имеет доступ только к тем объектам, которые соответствуют его полномочиям и функциям.

В целом создание системы разграничения доступа через технику представлений является более простым способом, чем непосредственное использование инструкций GRANT, и осуществляется в два этапа:

1. Для всех зарегистрированных пользователей в системе с помощью конструкций CREATE VIEW создаются свои представления объектов базы данных.

2. С помощью инструкций «GRANT SELECT ONИмяПредставления ТО ИмяПользователя» созданные представления авторизуются со своими пользователями.

Вместе с тем такой подход является более грубым по сравнению с применением инструкции GRANT непосредственно к объектам базы данных, т. к. не обеспечивает расщепления установок доступа к объектам на уровне отдельных операций (SELECT, INSERT, UPDATE, DELETE).

Поэтому другим подходом являются специальные расширения языка SQL, основанные на событийно-процедурной идеологии с введением специальных правил (RULE) безопасности:

CREATESECURITYRULEИмяПравила

GRANTсписок_привилегий_через_запятую ON ИмяОбъекта

WHERE условия

ТО Имена Пользователей_через_запятую

Введение правил безопасности обеспечивает более широкие и гибкие возможности реализации различных политик безопасности с большей степенью контроля и управляемости, но, как, впрочем, и техника представлений и непосредственное использование инструкций GRANT, не позволяет строить системы с мандатным принципом разграничения доступа.

Для решения этой задачи могут предлагаться более кардинальные расширения языка SQL с введением возможностей создания объектов базы данных с метками конфиденциальности. Следует, однако, заметить, что подобные примеры в коммерческих и сертифицированных по требованиям безопасности СУБД чрезвычайно редки.

По языкам безопасности баз отметим, что в современных СУБД для реализации установок, правил и ограничений доступа разрабатывается и используется специальный диалогово-наглядный интерфейс, автоматически

формирующий соответствующие конструкции языка SQL и позволяющий в большинстве случаев обходиться без непосредственного программирования.

1.3.3. Технологии обеспечения безопасности повторного использования объектов

Компьютерная система в целом, устройства оперативной и внешней (дисковой) памяти в частности, являются классическим примером среды многократного повторного информационного использования. Технологии обеспечения безопасности повторного использования объектов направлены на предотвращение угроз безопасности от случайного или преднамеренного извлечения интересующей злоумышленника информации по следам предшествующей деятельности или из технологического «мусора».

Часть этих технологий реализуются на уровне операционных систем, а часть являются специфическими функциями, осуществляемыми в автоматизированных информационных системах СУБД. Данные технологии условно можно разделить на три группы:

- изоляция процессов;
- очистка памяти после завершения процессов;
- перекрытие косвенных каналов утечки информации.

Изоляция процессов является стандартным принципом и приемом обеспечения надежности многопользовательских (многопроцессных) систем и предусматривает выделение каждому процессу своих непересекающихся с другими вычислительных ресурсов, прежде всего областей оперативной памяти. В СУБД данные задачи решаются мониторами транзакций.

Очистка памяти после завершения процессов направлена непосредственно на предотвращение несанкционированного доступа к конфиденциальной информации после завершения работы процессов с конфиденциальными данными уполномоченными пользователями. Так же как и изоляция процессов, чаще всего данная функция выполняется

операционными системами. Кроме того, следует отметить, что очистке подлежат не только собственно участки оперативной памяти, где во время выполнения процессов размещались конфиденциальные данные, но и участки дисковой памяти, используемые в системах виртуальной памяти.

Как уже отмечалось, при реализации систем разграничения доступа возможны косвенные каналы утечки информации. Источниками косвенных каналов утечки информации являются еще и ряд технологические аспекты, связанные с характеристиками процессов обработки данных. В этом плане различают косвенные каналы - «временные» и «по памяти».

В первом случае некоторые элементы конфиденциальной информации неуполномоченный пользователь получает на основе анализа времени выполнения отдельных процессов уполномоченными пользователями (скажем, по неправдоподобно большому времени для очевидно простой или какой-либо типовой операции).

Во втором случае соответственно «подозрение» вызывает занимаемый объем некоторых объектов (файлов, таблиц и т. п.), объем содержимого которых, с точки зрения того, что «видит» пользователь, явно не соответствует их фактическому объему.

Иначе говоря, все операции обработки данных должны выполняться строго над той выборкой данных, которая соответствует «представлению» пользователя.

В наиболее критичных с точки зрения безопасности информации случаях применяют еще один, наиболее жесткий вариант - технологию разрешенных процедур. В этом случае пользователям системы разрешается работать с базой данных исключительно через запуск разрешенных процедур.

Данный подход основывается также на уже рассмотренной технике хранимых (stored) процедур. Администратором системы для каждого пользователя формируется набор процедур обработки данных в соответствии

с полномочиями и функциями пользователя. Для безопасности хранимые процедуры в файле данных шифруются.

Ядро СУБД после идентификации и аутентификации пользователя при его входе в систему предъявляет ему разрешенный набор процедур. Непосредственного доступа к самим данным пользователь не имеет и работает только с результатами их обработки по соответствующим процедурам.

Данные КС размещаются в файлах данных, структура которых может быть известна нарушителю. Поэтому еще одной угрозой безопасности данных КС является возможность несанкционированного доступа к файлам данных вне программного обеспечения КС (СУБД) средствами операционной системы или дисковых редакторов. Для нейтрализации этой угрозы применяются методы и средства криптозащиты. В большинстве случаев криптографические средства защиты встраиваются непосредственно в программное обеспечение СУБД. Файл (файлы) базы данных при размещении на устройствах дисковой памяти шифруется. Соответственно, криптографическая подсистема при открытии файла данных его дешифрует. Специфической особенностью СУБД является особый порядок работы с файлами базы данных через организацию специальной буферизации страниц в оперативной памяти.

В развитых СУБД имеется возможность выборочного шифрования объектов базы данных исходя из уровня их конфиденциальности, вплоть до шифрования отдельных полей записей.

1.3.4. Технология надежного проектирования и администрирования

Часть угроз безопасности информации возникает из-за непреднамеренных (или преднамеренных) ошибок на этапах жизненного цикла КС - при разработке программного обеспечения СУБД; при проектировании и создании на базе СУБД конкретной КС, и, в том числе, при

проектировании системы разграничения доступа; при администрировании и сопровождении системы и, в том числе, при реагировании и действиях пользователей во внештатных ситуациях; при технологических операциях по резервированию, архивированию и восстановлению информации после сбоев; при выводе КС из эксплуатации. С целью нейтрализации или снижения вероятности данных угроз применяются ряд организационно-технологических и технических средств, решений, объединяемых в общую группу технологий надежного проектирования и администрирования. Их также условно можно разделить на следующие подгруппы:

- технологии надежной разработки программного обеспечения;
- технологии надежного проектирования и создания КС;
- технические средства и специальный инструментарий администрирования КС;
- протоколирование и аудит процессов безопасности.

Технологии надежной разработки программного обеспечения включают общие подходы к снижению ошибок при разработке программного кода и ряд более специфических аспектов, основанных на изначальном учете в концепции и структуре ядра системы (подсистема представления данных и подсистема доступа к данным) той или иной модели и технологий безопасности данных. Как показывает анализ выявленных уязвимостей в системах безопасности компьютерных систем, вероятность наличия и нахождения злоумышленниками брешей существенно выше в тех случаях, когда системы защиты реализуются в виде надстройки или внешней оболочки над ядром и интерфейсом исходных незащищенных систем.

Технологии надежного проектирования и создания на базе программного обеспечения СУБД конкретных АИС направлены на предотвращение логических ошибок в информационной инфраструктуре систем и в подсистемах разграничения доступа, строящихся на основе поддерживаемой СУБД модели и технологий безопасности данных. В этом

отношении основным и широко распространенным является структурно-функциональный подход.

При наличии большого количества пользователей (субъектов) и объектов информационных систем (баз данных) схема разграничения доступа может быть очень сложной и запутанной, что создает трудности для администрирования и порождает предпосылки для логических ошибок. Для преодоления этой угрозы в рамках структурно-функционального подхода применяют технику рабочих групп.

Рабочая группа объединяет пользователей, имеющих какое-либо общее технологическое отношение к базе данных (выполняющих похожие операции) и близкие параметры конфиденциальности по отношению к общим данным.

Администратор системы может создавать рабочие группы, рассматривая их как коллективных пользователей, с определенной идентификацией и набором полномочий. Каждый пользователь обязательно должен являться членом какой-либо рабочей группы. Полномочия, определенные для рабочей группы, автоматически распространяются на всех пользователей — членов группы, что является отражением некоторых элементов зонально-функционального принципа разграничения доступа. Дополнительно для каждого пользователя в его личной учетной записи могут быть уточнены и конкретизированы его полномочия.

Такой подход позволяет в большинстве случаев существенно уменьшить количество субъектов доступа в системе, сделать схему разграничения доступа более простой, «прозрачной» и управляемой, и тем самым снизить вероятность таких логических ошибок как неправильное предоставление доступа конкретного пользователя к конкретному объекту, превышение полномочий конкретного пользователя по доступу к ряду объектов, предоставление избыточных прав доступа и т.п.

Процессы в базе данных в технологиях рабочих групп помечаются как меткой пользователя, так и меткой рабочей группы, и, соответственно ядро безопасности СУБД проверяет подлинность обеих меток.

Проектирование системы доступа на основе технологии рабочих групп может проводиться «сверху» (дедуктивно) и «снизу» (индуктивно).

В первом способе сначала на основе анализа функциональной структуры и организационной иерархии пользователей (субъектов) формируются рабочие группы и осуществляются групповые назначения доступа. Далее каждый пользователь при его регистрации в системе включается в состав одной или нескольких групп, отвечающих его функциям. И, наконец, в заключение для каждого пользователя анализируются особенности его функциональных потребностей и доверительных характеристик и при необходимости осуществляются индивидуальные дополнительные назначения доступа. Формирование групп, групповые и индивидуальные установки доступа при этом осуществляются администратором системы, что соответствует принудительному способу управления доступом.

Такой подход позволяет снизить вероятность ошибочных назначений доступа и обеспечивает жесткую централизованную управляемость системой доступа, но может порождать, в свою очередь, дублирование групповых и индивидуальных полномочий доступа субъектов к объектам (проблема дублирования), а также избыточность доступа субъекта к одним и тем же объектам через участие в разных группах (проблема пересечения групп или в более широком смысле проблема оптимизации групп).

При втором (индуктивном) способе проектирования рабочих групп первоначально осуществляются индивидуальные назначения доступа субъектов (пользователей) к объектам. Назначения производятся на основе опроса и анализа функциональных потребностей и доверительных характеристик пользователей, и могут осуществляться администратором системы (принудительный способ управления доступом) или через

индивидуальные запрашивания субъектами доступа владельцев объектов (принцип добровольного управления доступом). Далее, уже администратором системы, производится анализ общих или схожих установок доступа у различных субъектов, на основе которого они объединяются в рабочие группы. Выделенные общие установки доступа используются в качестве групповых назначений доступа. При этом анализ схожести доступа при большом количестве субъектов и объектов представляет непростую задачу и решается администратором системы в значительной степени эвристически.

Дополнительным организационным способом повышения надежности и безопасности в процессе администрирования и сопровождения системы является разделение общего администрирования и администрирования безопасности. Общий администратор строит, поддерживает и управляет информационной инфраструктурой системы — информационно-логическая схема, категорирование конфиденциальности объектов (ресурсов и устройств), интерфейсные и диалоговые элементы, формы, библиотеки запросов, словарно-классификационная база, резервирование и архивирование данных. Администратор безопасности организует и управляет системой разграничения доступа — доверительные характеристики (допуска) пользователей, конкретные назначения доступа, регистрация и формирование меток доступа пользователей.

Доступ к массиву учетных записей пользователей имеет только администратор безопасности. Совмещение функций общего администрирования и администрирования безопасности одновременно одним пользователем не допускается, что объективно повышает надежность системы.

Протоколирование и аудит событий безопасности являются важным средством обеспечения управляемости состоянием и процессами безопасности, создают условия для расследования фактов нарушения

информационной безопасности, анализа и исключения их причин, снижения отрицательных последствий и ущерба от них.

Документированию подлежат все события, критичные с точки зрения безопасности в системе:

- вход/выход пользователей;

- регистрация новых пользователей, смена привилегий и назначений доступа (все обращения к массивам учетных записей);

- все операции с файлами (создание, удаление, переименование, копирование, открытие, закрытие);

- обращения к/из удаленной системе(ы).

При этом по каждому такому событию устанавливается минимально необходимый перечень регистрируемых параметров, среди которых:

- дата и время события;

- идентификатор пользователя-инициатора;

- тип события;

- источник запроса (для распределенных систем — сетевое имя терминала, рабочей станции и т. п.);

- имена затронутых объектов;

- изменения, внесенные в учеты в системы, в том числе в массивы учетных записей;

- метки доступа субъектов и объектов.

В СУБД такой подход хорошо вписывается в событийно-процедурную технологию с использованием техники журнализации. При этом доступ к журналу событий имеет только администратор безопасности, который при обнаружении фактов или признаков нарушений безопасности имеет возможность восстановления хода событий, анализа и устранения источников и причин нарушения безопасности системы.

В этом отношении журнал событий безопасности является необходимым средством аудита безопасности. Аудит безопасности заключается в контроле и отслеживании событий в системе с целью

выявления, своевременного обнаружения проблем или нарушений безопасности и сигнализации об этом администратору безопасности. Ввиду того что процессы доступа, различных процедур, операций, информационных потоков в компьютерных системах являются многоаспектными, не строго детерминированными, т. е. частично или полностью стохастическими, разработка аналитических, алгоритмических или иным образом аналитических автоматизированных процедур обнаружения фактов и признаков нарушений информационной безопасности является чрезвычайно сложной и неопределенной задачей. Поэтому в настоящее время разрабатывается ряд эвристических и нейросетевых технологий, которые в некоторых случаях с успехом воплощаются в специальном программном инструментарии администратора безопасности, обеспечивая автоматизированный аудит безопасности системы.

В простейшем случае журнализация изменений заключается в последовательной записи во внешнюю память всех изменений, выполняемых в базе данных. Записывается следующая информация:

- порядковый номер, тип и время изменения;
- идентификатор транзакции;
- объект, подвергшийся изменению (номер хранимого файла и номер блока данных в нём, номер строки внутри блока);
- предыдущее состояние объекта и новое состояние объекта.

Формируемая таким образом информация называется журнал изменений базы данных. Журнал содержит отметки начала и завершения транзакции, и отметки принятия контрольной точки.

В СУБД с отложенной записью блоки данных внешней памяти снабжаются отметкой порядкового номера последнего изменения, которое было выполнено над этим блоком данных. В случае сбоя системы эта отметка позволяет узнать какая версия блока данных успела достичь внешней памяти.

СУБД с отложенной записью периодически выполняет контрольные точки. Во время выполнения этого процесса все незаписанные данные переносятся на внешнюю память, а в журнал пишется отметка принятия контрольной точки. После этого содержимое журнала, записанное до контрольной точки может быть удалено.

Журнал изменений может не записываться непосредственно во внешнюю память, а аккумулироваться в оперативной. В случае подтверждения транзакции СУБД дожидается записи оставшейся части журнала на внешнюю память. Таким образом, гарантируется, что все данные, внесённые после сигнала подтверждения, будут перенесены во внешнюю память, не дожидаясь переписи всех измененных блоков из дискового кэша. СУБД дожидается записи оставшейся части журнала так же при выполнении контрольной точки.

В случае логического отказа или сигнала отката одной транзакции журнал сканируется в обратном направлении, и все записи отменяемой транзакции извлекаются из журнала вплоть до отметки начала транзакции. Согласно извлеченной информации выполняются действия, отменяющие действия транзакции, а в журнал записываются компенсирующие записи. Этот процесс называется откат (rollback).

В случае физического отказа, если ни журнал, ни сама база данных не повреждена, то выполняется процесс прогонки (rollforward). Журнал сканируется в прямом направлении, начиная от предыдущей контрольной точки. Все записи извлекаются из журнала вплоть до конца журнала. Извлеченная из журнала информация вносится в блоки данных внешней памяти, у которых отметка номера изменений меньше, чем записанная в журнале. Если в процессе прогонки снова возникает сбой, то сканирование журнала вновь начнется сначала, но фактически восстановление продолжиться с той точки, откуда оно прервалось.

Для увеличения отказоустойчивости СУБД может записывать одновременно несколько идентичных копий журнала изменений. Если в

случае отказа одна из копий журнала окажется недоступной, СУБД восстановит базу данных, используя любую из доступных копий. Такая стратегия называется мультиплексированием журнала изменений.

Контрольные вопросы

1. Какие программно-технологические задачи решаются при обеспечении безопасности базы данных?
2. Место идентификации/аутентификации в защищенной системе.
3. Какие средства идентификации/аутентификации вы знаете?
4. Какой вид аутентификации, в основном, используется в системах управления базами данных?
5. Перечислите основные недостатки парольной аутентификации.
6. Перечислите основные и дополнительные функции системы управления базами данных.
7. Объясните схему взаимодействия пользователя с базой данных в ранних СУБД.
8. Объясните структура SQL-инструкции.
9. Что такое «представление»?
10. Технологии обеспечения безопасности повторного использования объектов.
11. Технологии надежного проектирования и администрирования.
12. Объясните технологии журнализации.
13. Какая информация пишется в журналы изменений?
14. Что такое мультиплексирование журналов изменений?

Глава 2. МОДЕЛИ И МЕТОДЫ РАЗГРАНИЧЕНИЯ ДОСТУПА В БАЗЫ ДАННЫХ

2.1. Модели безопасности базы данных

Одним из основных направлений информационной безопасности является создание формальных моделей информационной безопасности, называемых также моделями разграничения доступа. Под моделью информационной безопасности понимают формально описанную политику безопасности - совокупность норм и правил, обеспечивающих эффективную защиту системы обработки информации от заданного множества угроз безопасности.

Модели безопасности играют важную роль в процессах разработки и исследования защищенных компьютерных систем, так как обеспечивают системотехнический подход, включающий решение следующих важнейших задач:

- выбор и обоснование базовых принципов архитектуры защищенных компьютерных систем, определяющих механизмы реализации средств и методов защиты информации;
- подтверждение свойств (защищенности) разрабатываемых систем путем формального доказательства соблюдения политики безопасности (требований, условий, критериев);
- составление формальной спецификации политики безопасности как важнейшей составной части организационного и документационного обеспечения разрабатываемых защищенных компьютерных систем.

По сути, модели безопасности являются исходным связующим элементом в триаде «Заказчик (Потребитель) - Разработчик (Производитель)-Эксперт (Аудитор)». На основе моделей безопасности заказчики могут формулировать те требования к защищенным КС, которые соответствуют политике безопасности, технологическим процессам обработки информации,

принятым в своих организациях и предприятиях. Разработчики на основе моделей безопасности формируют технико-технологические требования и программно-технические решения по разрабатываемым системам. Эксперты, основываясь на моделях безопасности, строят методики и спецификации оценки защищенности конкретных систем, осуществляют сертификацию разработанных систем по требованиям защиты информации.

Модель управления доступом – это структура, которая определяет порядок доступа субъектов к объектам (рис.2.1).

Рис. 2.1. Структура модели управления доступом

Для реализации правил и целей этой модели используются технологии управления доступом и механизмы безопасности. Среди множества моделей безопасности можно выделить основные типы моделей: дискреционные модели, мандатные модели, модели с ролевым разграничением доступа. Каждая модель использует различные методы для управления доступом субъектов к объектам, каждая имеет свои преимущества и ограничения. Надо отметить, что эти методы не обязательно применяются отдельно друг от друга, а могут комбинироваться для удовлетворения различных требований к безопасности системы рис.2.2.

Рис.2.2. Системы контроля доступа

Контрольные вопросы

1. Объясните понятие «политика безопасности» и «модель безопасности».
2. Место модели безопасности при создании и исследовании защищенных компьютерных систем.
3. Структура модели управления доступом.
4. Объясните схему системы контроля доступом.

2.2. Организация разграничения доступа в базы данных на основе дискреционной модели

Дискреционные модели безопасности - модели, основанные на дискреционном управлении доступом (Discretionary Access Control). Дискреционное разграничение доступа – разграничение доступа между поименованными субъектами и поименованными объектами. Реализация дискреционного разграничения доступа осуществляется с учетом следующих положений:

а) каждый пользователь должен быть аутентифицирован прежде чем он получит доступ к данным;

б) в соответствии с аутентифицирующей информацией (идентификатор и пароль) пользователю назначается система его полномочий. Полномочие

пользователя заключается в том, что он имеет право доступа только к фиксированному набору данных и процедур их обработки. При этом никакой запрос пользователя не должен допускать вовлечение в процесс обработки недоступных ему данных. Совокупность аутентифицирующей информации пользователя и полномочий его доступа к сегментам данным и функциям их обработки образует маркер доступа, определяющий уровень допуска.

Для задания и проверки полномочий удобно субъект-объектные отношения представлять в виде матрицы безопасности, которая изображена на рис. 2.3. В матрице безопасности перечисляются по строкам все пользователи, а по столбцам все фрагменты данных. На пересечениях отмечаются допустимые операции над данными.

Пользователь	Таблица 1				…	Таблица M			
	Поле 1	Поле 2	…	Поле N	…	Поле 1	Поле 2	…	Поле N
User 1	RWCD	RWCD		RW		RW	RWD		R
User 2	RWC	RWCD		RWC		R	RW		RWCD
…									
User X	R	RWD		RWCD		RWD	R		RWC

Рис. 2.3. Пример матрицы безопасности

Проверка полномочий, основанная на матрице безопасности, не гарантирует защищенность системы, так как не предоставляет средств проверки подлинности пользователя (процесса), запрашивающего данные, что может привести к несанкционированному доступу (рис. 2.4).

Поэтому защищенная СУБД должна предусматривать наличие в подсистеме разграничения доступа (ПРД) проверку подлинности, которая обеспечивает, подтверждение заявленных пользователем или процессом, идентификаторов.

Однако, применение маркеров доступа в ПРД не позволяет организовать разграничение доступа к данным с разной степенью конфиденциальности. Действительно, если пользователь заявил санкционированные полномочия доступа к данным фрагмента *L*, а в нем содержится информация различной степени конфиденциальности, то

автоматически данный пользователь получает возможность работать со всеми данными фрагмента *L*.

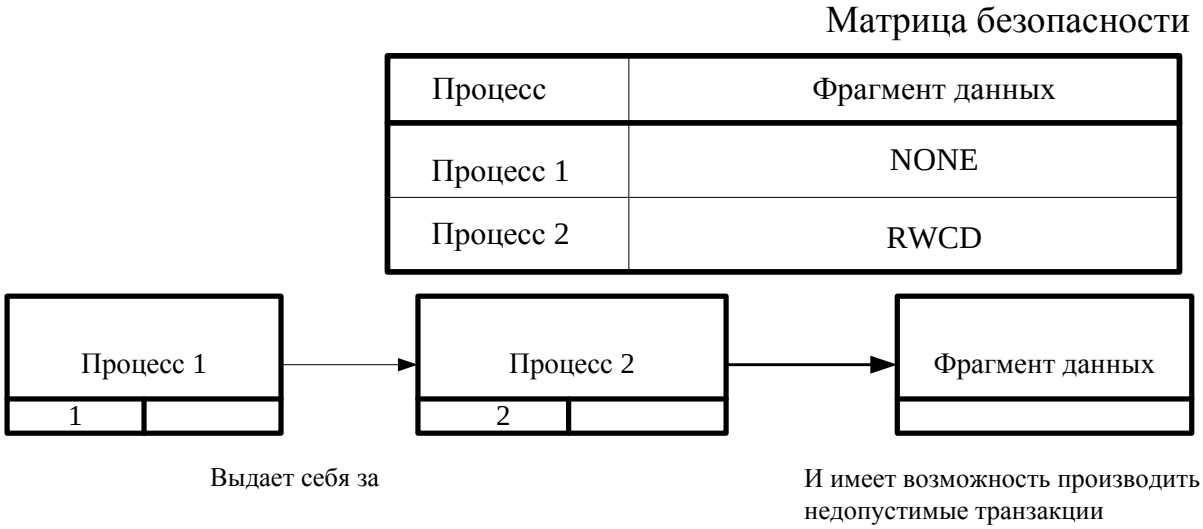


Рис. 2.4. Пример несанкционированного доступа к данным

К достоинствам дискреционного разграничения доступа можно отнести хорошую детализацию защиты и относительно простую реализацию. Но этот вид обладает и рядом недостатков. Доступ ограничивается только к именованным объектам, а не собственно к хранящимся данным. Так например в случае применения реляционной СУБД объектом будет являться таблица. В этом случае нельзя в полном объеме ограничить доступ только к части информации, хранящейся в таблице. Кроме этого, существует проблема троянских программ (троянских коней). Когда пользователь вызывает какую-либо программу на компьютере, в системе инициируется некоторая последовательность операций, зачастую скрытых от пользователя. Эти операции обычно управляются операционной системой.

Применение средств дискреционного разграничения доступа не позволяет решить задачу контроля за передачей информации. Это обусловлено тем, что указанные средства не могут помешать авторизованному пользователю законным образом получить конфиденциальную информацию и затем сделать ее доступной для других, неавторизованных, пользователей. Это становится возможным потому, что

привилегии существуют отдельно от данных (в случае реляционных СУБД - отдельно от строк реляционных таблиц), в результате чего данные оказываются «обезличенными» и ничто не мешает передать их кому угодно даже средствами самой СУБД, получив доступ к таблице или представлению.

Одна из первых моделей безопасности была модель дискреционного доступа, модель АДЕПТ-50. В модели представлено четыре типа объектов, относящихся к безопасности: пользователи (u), задания (j), терминалы (t) и файлы (f), причем каждый объект описывается четырехмерным кортежем.

Контрольные вопросы

1. Что понимается под управлением доступа?
2. Объясните матрицу безопасности в дискреционном управлении доступом.
3. Преимущества дискреционного разграничения доступа.
4. Недостатки дискреционного разграничения доступа.

2.3. Организация разграничения доступа в базы данных на основе мандатной модели

Мандатные модели основаны на мандатном разграничении доступа (Mandatory Access Control), представляющем собой совокупность правил предоставления доступа, определенных на множестве атрибутов безопасности субъектов и объектов.

Мандатное разграничение доступа – это разграничение доступа субъектов к объектам данных, основанное на характеризуемой меткой конфиденциальности информации, которая содержится в объектах, и на официальном разрешении (допуске) субъектов обращаться к информации такого уровня конфиденциальности. В отличии от дискреционного мандатный доступ накладывает ограничения на передачу информации от одного пользователя к другому. Это позволяет разрешить проблему троянских коней. Примерами мандатных моделей: модель Белл-ЛаПадула,

модель MMS. Как известно, в мандатных моделях, в частности, в классической модели Белл-ЛаПадулы, в качестве классифицирующего множества используется линейная решетка уровней конфиденциальности. Отображение (классификация) субъектов доступа (пользователей) на решетку уровней конфиденциальности отражает парадигму градуированного доверия (уровни допуска), а отображение объектов доступа - парадигму рангового измерения конфиденциальности (грифы секретности), т.е. степени ущерба от неконтролируемого распространения соответствующей информации.

Метки конфиденциальности главным образом характеризует данные: их принадлежность (группу принадлежности), важность, представительность, уровни конфиденциальности и ценности данных объекта (таблицы, столбца, строки или поля) и пр. Метки конфиденциальности неизменны на всем протяжении существования объекта защиты (они уничтожаются только вместе с ним) и располагаются вместе с защищаемыми данными.

Реализация такого вида разграничения в ПРД СУБД не дает проигнорировать метки конфиденциальности при получении доступа к информации. Такие реализации ПРД, как правило, представляют собой комплекс средств как на машине-сервере, так и на машине-клиенте, при этом возможно использование специальной защищенной версии операционной системы.

Существует также реализация некоторой модификации мандатного разграничения доступа посредством применения сложного набора хранимых процедур. При этом метки добавляются в таблицу в качестве дополнительного атрибута, доступ к таблицам запрещается вообще и ни одно приложение не может выполнить интерактивный SQL-запрос, а только хранимую процедуру. Реализации политики разграничение доступа в этом случае достаточно сложна и предполагает определенный уровень доверия к администратору безопасности, так как он имеет право изменять структуру базы данных, а значит, и хранимые процедуры, представления. Физически же

администратор безопасности в данном случае не изолирован от управления конфиденциальными (секретными) данными.

Установлено, что разграничение доступа пользователей различной степени благонадежности к фрагментам мультиконфиденциальных данных возможно реализовать посредством построения многоуровневой политики разграничение доступа, в которой управление и контроль доступа осуществляется в соответствии со степенями конфиденциальности хранимой в БД информации.

Многоуровневая политика разграничения доступа строится на основе модели Белл-ЛаПадула которая предназначена для управления субъектами, то есть активными процессами, запрашивающими доступ к данным, и объектами, то есть файлами, таблицами, представлениями, записями, полями. Сущность моделирования по Белл-ЛаПадула заключается в классификации объектов по степени конфиденциальности, а субъектов – по уровню благонадежности. После чего формируется правило, описывающее набор полномочий уровня допуска относительно класса конфиденциальности.

Механизм защиты базы данных, основанный на модели Белл-ЛаПадула, строится на основе «обратного наследования» категорий информации классами конфиденциальности данных. Смысл обратного наследования иллюстрируется рис. 2.5 и заключается в следующем. Задается некоторая иерархия классов конфиденциальности данных. Кроме того, по признаку значимости (важности) формируется совокупность групп данных в рамках классов, представляющая множество категорий. С повышением степени важности информации наследуется соответствующая категория информации. Это означает, что пользователь, допущенный к соответствующей категории информации, обладает правами чтения данных с меньшей важностью.

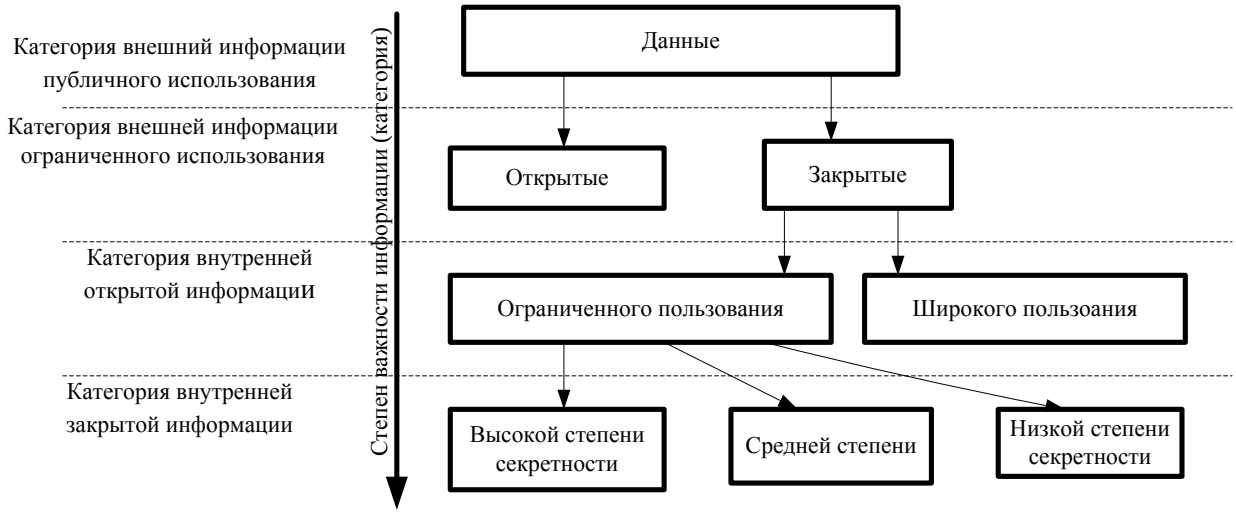


Рис. 2.5. Иерархия классов конфиденциальности данных

Доступ к данным соответствующей категории обеспечивается так называемым фильтром доступа, который выбирается в соответствии с уровнем допуска пользователя. Модель ПРД с мандатным разграничением доступа представлена на рис. 2.6.

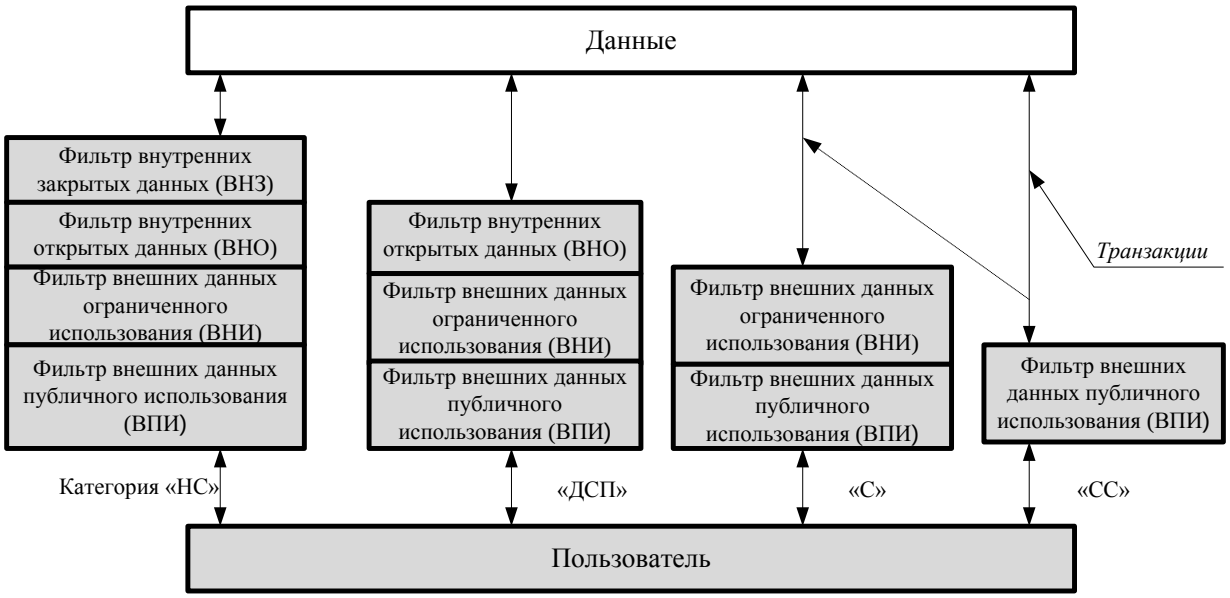


Рис. 2.6. Модель ПРД с мандатным разграничением доступа

На более высоких уровнях модели безопасности данных определяется система полномочий пользователя по обработке информации (рис. 2.7).

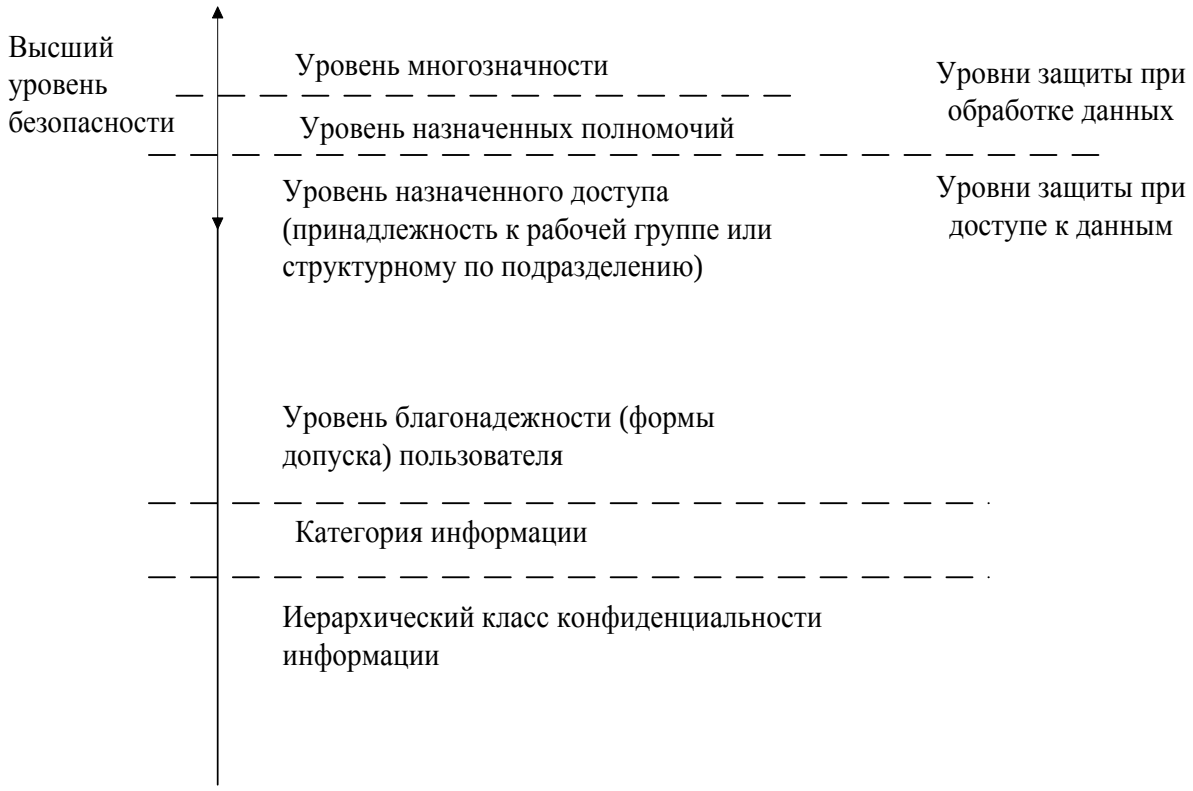


Рис. 2.7. Классификационные признаки объектов и субъектов при мандатном разграничении доступа

Свойство системы защиты, которое позволяет субъекту иметь право на запись в объект только в том случае, если класс субъекта такой же или ниже, чем записываемого объекта, будем называть σ-свойством (сигма). Пример реализации σ-свойства изображен на рис.2.8. Данные в таблице отсортированы в порядке убывания степени конфиденциальности. Процесс с допуском «С» может записать данные только в кортежи с соответствующим или более высоким классом конфиденциальности, а читать данные – только из кортежей с соответствующим или более низким классом конфиденциальности.

Реализация принципов Белл-ЛаПадула связана с требованием поддержки для одной и той же таблицы нескольких уровней защиты, что приводит к некоторому снижению устойчивости, надежности и управляемости СУБД.

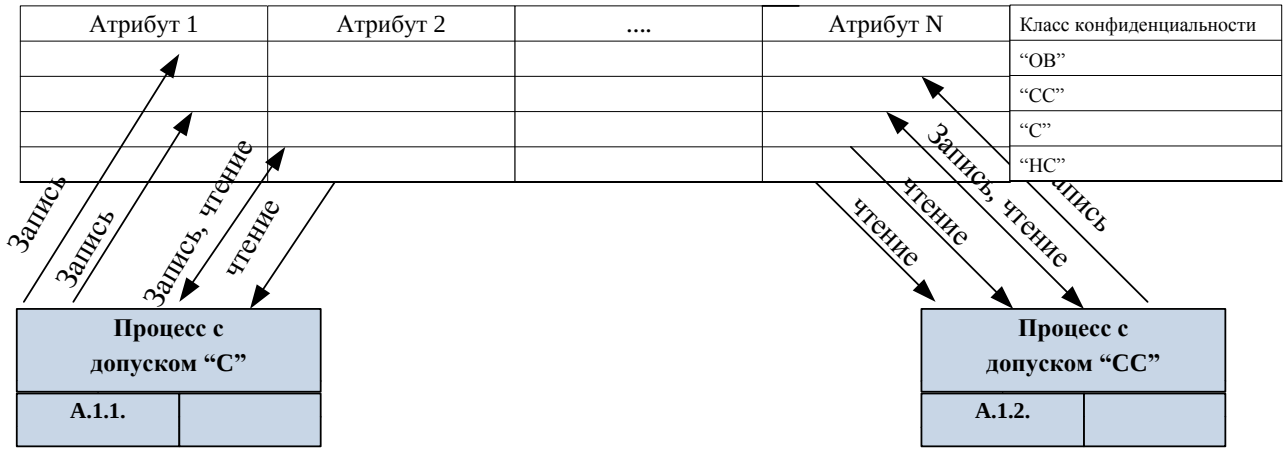


Рис. 2.8. Пример применения правил Белл-ЛаПадула

Известны также работы по расширению мандатных моделей введением более сложных классифицирующих множеств, учитывающих, в том числе, и тематический аспект, основывающийся на использовании тематических классификаторов-рубрикаторов (дескрипторных, иерархических, фасетных) – т.н. MLS-решеток. Однако модели с MLS-решетками не изменяют сущности исходных мандатных моделей, имеющих чрезвычайно абстрактный по отношению к реальным СУБД, в частности, не регламентируют в большинстве случаев процессы присвоения и изменения классификаций сущностей (субъектов и объектов) доступа, а в более широком плане не регламентируют процессы инициализации субъектов доступа, а также множественные доступы (один субъект - одновременно к нескольким объектам, несколько субъектов - одновременно к одному объекту). Кроме того, мандатные модели ни каким образом не учитывают структуру системы объектов в СУБД, рассматривая их как множество объектов, охваченное линейным порядком по отношению конфиденциальности (секретности). В результате применения мандатных моделей в СУБД не обеспечивает единого механизма организации данных и разграничения доступа к ним.

Контрольные вопросы

1. Объясните суть мандатного разграничения доступа?
2. Иерархия классов конфиденциальности данных.

3. Классификационные признаки объектов и субъектов при мандатном разграничении доступа.
4. Объясните пример применения правил Белл-ЛаПадула в системах управления базами данных.

2.4. Организация разграничения доступа в базы данных на основе ролевой модели

Модели с ролевым разграничением доступа (Role-Based Access Control) представляют собой развитие политики дискреционного разграничения доступа. Права доступа субъектов системы к объектам группируются с учетом специфики их применения, образуя роли. При этом правила данной модели являются более гибкими, чем правила мандатной модели, построенные на основе жестко определенной решетки ценности информации. В ролевой модели классическое понятие «субъект» заменяется понятиями «пользователь» и «роль». Пользователь - это человек, работающий с системой и выполняющий определенные служебные обязанности. Роль - это активно действующая в системе абстрактная сущность, с которой связан ограниченный, логически связанный набор полномочий, необходимый для осуществления определенной деятельности. При использовании ролевой политики управление доступом осуществляется в две стадии: во-первых, для каждой роли указывается набор полномочий, представляющих набор прав доступа к объектам, и, во-вторых, каждому пользователю назначается список доступных ему ролей.

В ПРД с ролевым разграничением доступа (РРД) пользователи не могут передавать права на доступ к информации другим пользователям, что является фундаментальным отличием РРД от дискреционного и мандатного видов разграничения доступа.

Определение членства и распределение полномочий роли при РРД (в отличие от дискреционного доступа) зависит не от администратора

безопасности, а от политики безопасности, принятой в организации и конкретно СУБД. Роль можно понимать как множество действий, которые пользователь или группа пользователей может исполнять в контексте организации. Понятие роли включает описание обязанностей, ответственности и квалификации. Функции распределяются по ролям администратором безопасности СУБД. Доступ пользователя к роли также определяется этим администратором.

Ролевой вид разграничения доступа подразумевает контроль доступа на уровне абстракции и описании сущностей, используемых в организации. Если ПРД построена на основе РРД, то добавление и удаление ролей становится несложным процессом. Таким образом, РРД позволяет администратору безопасности оперировать с абстракциями более высокого уровня, чем традиционные списки контроля доступа, используемые дискреционным доступом.

Политики безопасности, основанные на РРД, описываются в терминах пользователей, ролей, операций и защищаемых объектов (рис. 2.9).

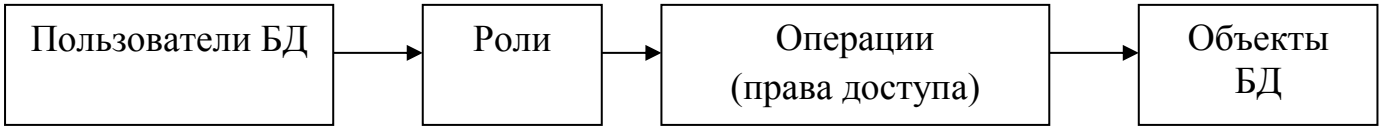


Рис. 2.9. Механизм доступа пользователей к информации, хранимой в базе данных, при РРД

Для того, чтобы применить некоторую операцию к защищаемому РРД объекту, пользователь должен выполнять некоторую роль. До того, как пользователь может выполнять данную роль, он должен быть авторизован для данной роли администратором безопасности СУБД. Таким образом, РРД наделяет администратора способностью устанавливать ограничения на авторизацию в роли, активацию роли, выполнение операций.

Задание ролей позволяет определить более четкие и понятные для пользователей компьютерной системы правила разграничения доступа. При

этом РРД наиболее эффективно используется в компьютерных системах, для пользователей которых четко определен круг их должностных полномочий и обязанностей.

Роль является совокупностью прав доступа на объекты компьютерной системы. Однако РРД не является частным случаем дискреционного разграничения доступа, так как правила РРД определяют порядок предоставления прав доступа субъектам компьютерной системы в зависимости от сессии его работы и от имеющихся или отсутствующих у него ролей в каждый момент времени, что является характерным для систем мандатного разграничения доступа. В то же время правила РРД являются более гибкими, чем правила мандатного разграничения доступа, построенные на основе жестко определенной решетки (шкалы) ценности информации.

Основными элементами базовой модели РРД являются:

U - множество пользователей;

R - множество ролей;

P - множество прав доступа на объекты компьютерной системы;

S - множество сессий пользователей;

$PA: \quad R \to 2^R$ - функция, определяющая для каждой роли множество прав доступа; при этом для каждого $p \in P$ существует $r \in R$ такая, что $p \in PA(r)$;

$UA: \quad U \to 2^R$ - функция, определяющая для каждого пользователя множество ролей, на которые он может быть авторизован;

$user: \quad S \to U$ - функция, определяющая для каждой сессии пользователя, от имени которого она активизирована;

$roles: \quad S \to 2^R$ - функция, определяющая для пользователя множество ролей, на которые он авторизован в данной сессии; при этом в каждый момент времени для каждого $s \in S$ выполняется условие $roles(s) \subseteq UA(user(s))$.

Множество ролей, на которые авторизуется пользователь в течение одной сессии, модифицируется самим пользователем. В базовой модели РРД отсутствуют механизмы, позволяющие одной сессии активизировать другую сессию. Все сессии активизируются пользователем. Важным механизмом базовой модели РРД являются ограничения, накладываемые на множества ролей, на которые может быть авторизован пользователь или на которые он авторизуется в течение одной сессии. Данный механизм также необходим для широкого использования РРД, так как обеспечивает большее соответствие, используемым в компьютерных системах, технологиям обработки информации.

В базовой модели РРД предполагается, что множества U, R, P и функции PA, UA не изменяются с течением времени или существует единственная роль - «администратор безопасности», которая предоставляет возможность изменять эти множества и функции. В реальных компьютерных системах, в которых одновременно могут работать сотни и тысячи пользователей, а структура ролей и прав доступа может быть очень сложной, проблема администрирования является чрезвычайно важной задачей. Для решения этой задачи рассматривается построенная на основе базовой модели РРД модель администрирования РРД.

Общая структура элементов базовой модели РРД имеет вид, представленный на рис.2.10.

В дополнение к используемым элементам базовой модели РРД в модели администрирования РРД рассматриваются следующие элементы:

AR - множество административных ролей;

AP - множество административных прав доступа;

$\mathrm{APA}: \mathrm{AR} \rightarrow 2^{\mathrm{AP}}$ – функция, определяющая для каждой административной роли множество административных прав доступа;

$AUA: U \rightarrow 2^{AR}$ - функция, определяющая для каждого пользователя множество административных ролей, на которые он может быть авторизован.

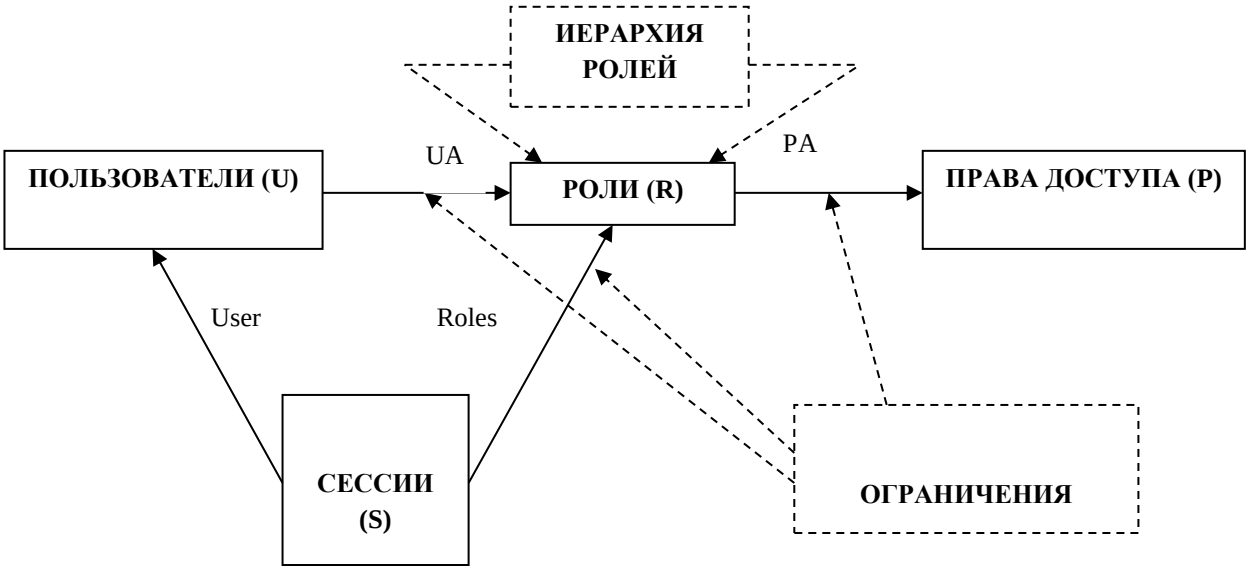


Рис.2.10. Структура базовой модели РРД

Общая структура элементов модели администрирования РРД имеет вид, представленный на рис. 2.11.

При администрировании множеств авторизованных ролей пользователей изменяются значения функции *AUA.* Для изменения значений функции *UA()* определяются специальные административные роли из множества *AR.*

Для администрирования множеств авторизованных ролей пользователей необходимо определять:

- для каждой административной роли множество ролей, множества авторизованных пользователей, которых она позволяет изменять;

- для каждой роли предварительное условие, которому должны соответствовать пользователи, прежде чем они будут включены в множество ее авторизованных пользователей. Типовая структура иерархии ролей и иерархии административных ролей представлена на рис. 2.12. *а* и *б.*

Минимальная роль в иерархии - служащий *(E).* Иерархия ролей разработчиков проектов имеет максимальную роль - директор *(DIR),* минимальную роль - инженер *(ED).* В управлении выполняются работы по двум проектам. В каждом проекте определены максимальная роль - руководитель проекта *(PL1, PL2* соответственно), минимальная роль -

инженер проекта *(E1, E2* соответственно) и не сравнимые между собой роли - инженер по производству *(PE1, PE2* соответственно) и инженер по контролю *(QE1, QE2* соответственно). Иерархия административных ролей состоит из четырех ролей с максимальной ролью – администратор безопасности *(SS)*.

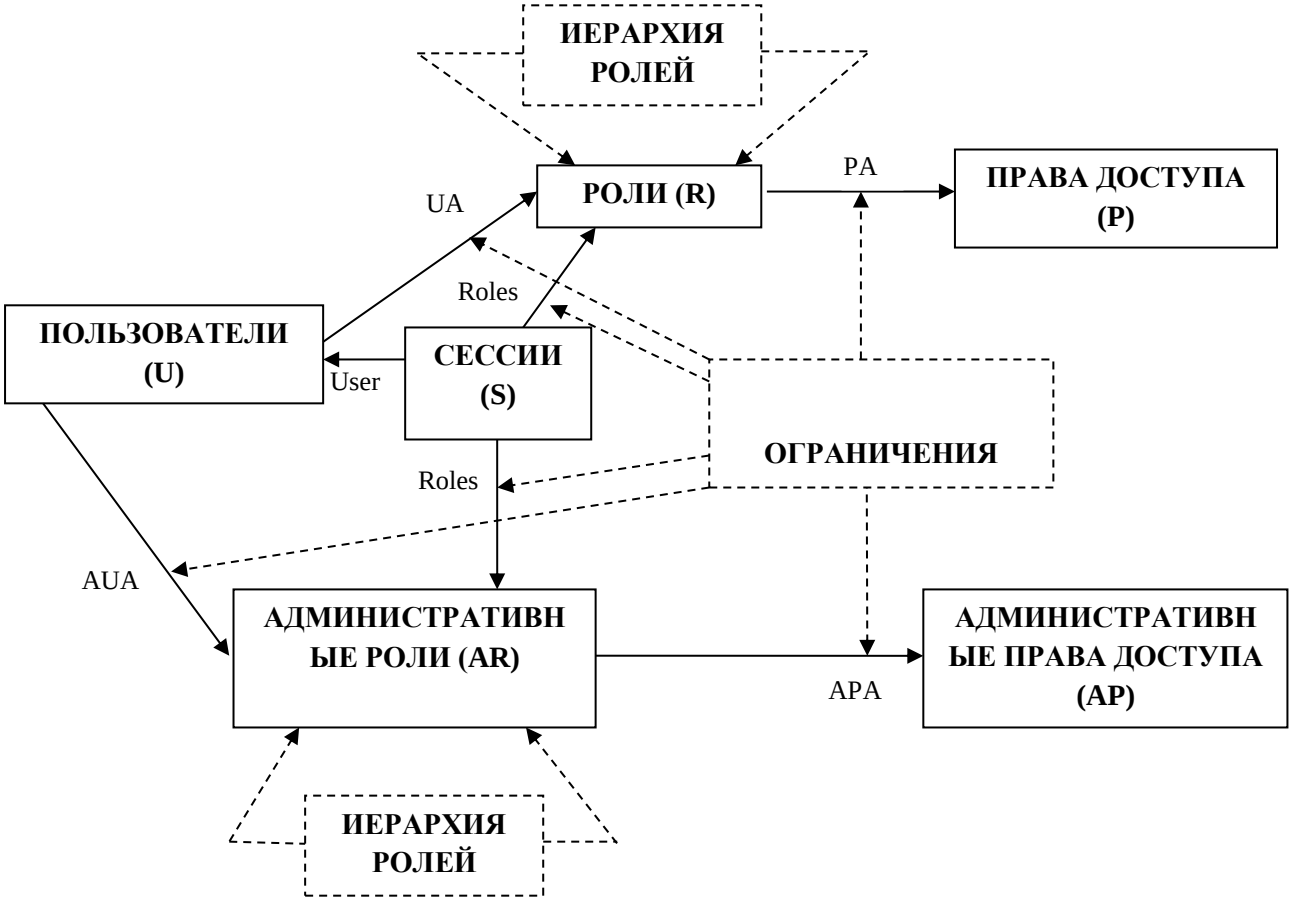
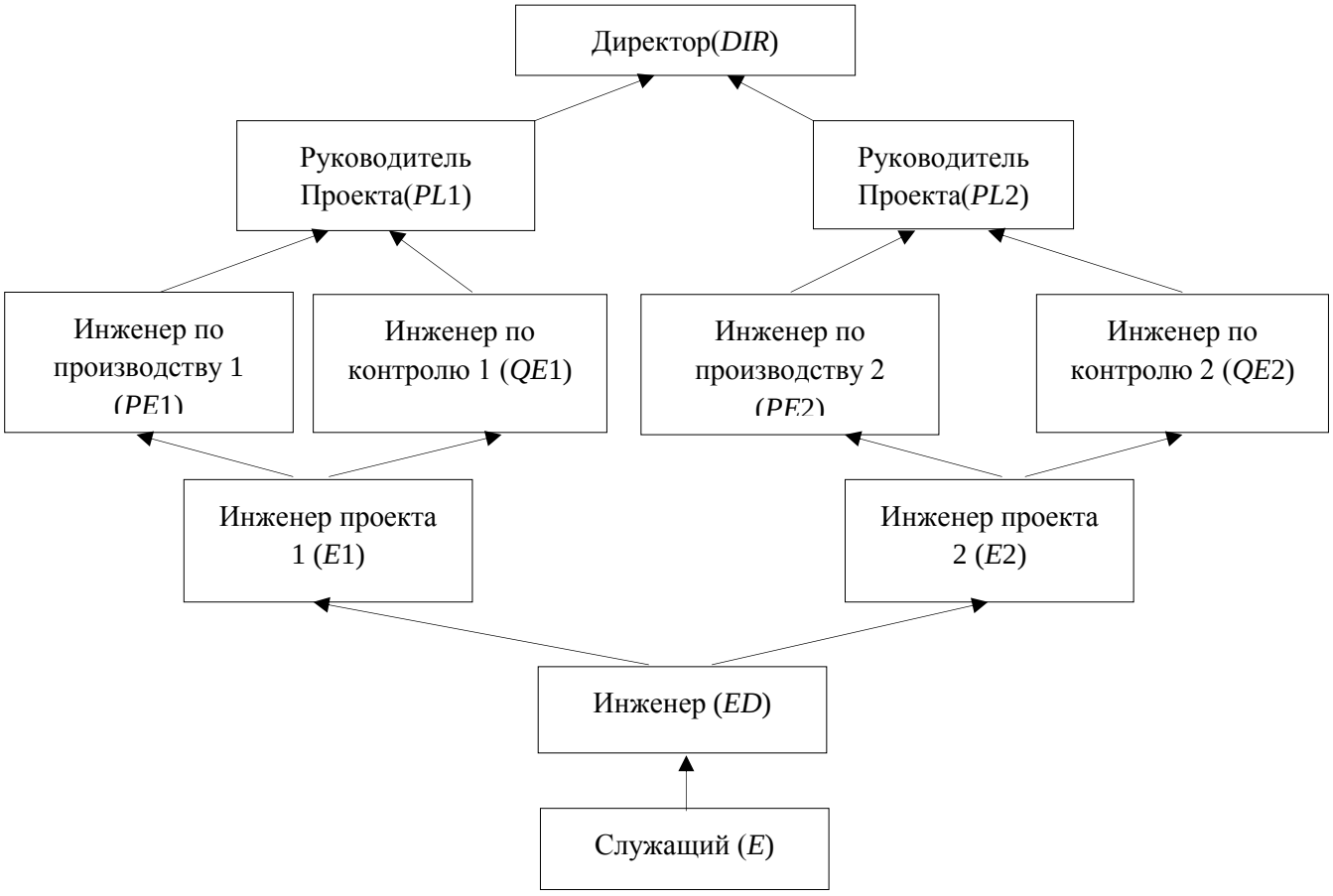
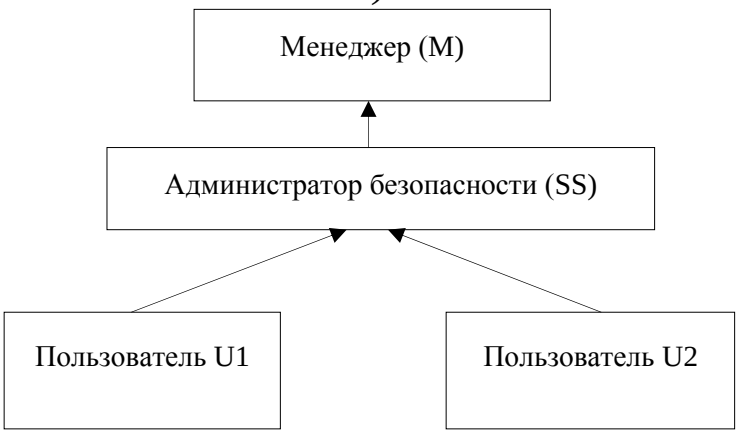


Рис.2.11. Структура модели администрирования РРД

Определение правил администрирования, позволяющих изменять иерархию ролей, является самой сложной задачей, в рассматриваемой модели администрирования РРД. Для решения данной задачи используются подходы, реализованные при определении правил администрирования множеств авторизованных ролей пользователей и прав доступа ролей. Задаются три иерархии, элементами которых являются:



а)



б)

Рис.2.12.Иерархия административных ролей

- возможности - множества прав доступа и других возможностей;
- группы - множества пользователей и других групп;
- объединения - множества пользователей, прав доступа, групп, возможностей и других объединений.

Иерархия объединений является наиболее общей и может включать в себя иерархии возможностей и групп. Определение возможностей и групп требуется для обеспечения соответствия правил администрирования ролей в модели и используемых на практике технологий обработки информации и создания административных структур организаций. Например, для выполнения своих функций пользователю может быть необходим некоторый набор прав доступа, причем отсутствие в этом наборе некоторого права доступа может сделать бессмысленным обладание имеющимися правами.

На основе иерархий возможностей, групп и объединений задается иерархия ролей, элементами которой являются роли-возможности, роли-группы, роли-объединения:

Роли-возможности - роли, которые обладают только определенными в соответствующей возможности правами доступа.

Роли-группы - роли, на которые могут быть авторизованы одновременно только все пользователи соответствующей группы.

Роли-объединения - роли, которые обладают возможностями, правами доступа и на которые могут быть авторизованы группы пользователей и отдельные пользователи.

Контрольные вопросы

1. Объясните суть модели с ролевым разграничения доступа.
2. Раличие модели с ролевым разграничением доступа от дискреционной и мандатной моделей.
3. Объясните механизм доступа пользователей к информации в модели с ролевым разграничением доступа.
4. Структура базовой модели с ролевым разграничением доступа.
5. Структура модели администрирования ролевого разграничения доступа.
6. Объясните иерархию административных ролей.

Глава 3. ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ РАСПРЕДЕЛЕННЫХ СИСТЕМ БАЗЫ ДАННЫХ

3.1. Концепция информационной безопасности в распределенных системах базы данных

Системы распределенных вычислений появляются, прежде всего, по той причине, что в крупных автоматизированных информационных системах, построенных на основе корпоративных сетей, не всегда удается организовать централизованное размещение всех базы данных и СУБД на одном узле сети. Поэтому системы распределенных вычислений тесно связаны с системами управления распределенными базами данных.

Распределенная база данных - это совокупность логически взаимосвязанных баз данных, распределенных в компьютерной сети.

Система управления распределенной базой данных - это программная система, которая обеспечивает управление распределенной базой данных и прозрачность ее распределенности для пользователей.

Распределенная база данных может объединять базы данных, поддерживающие любые модели (иерархические, сетевые, реляционные и объектно-ориентированные базы данных) в рамках единой глобальной схемы. Подобная конфигурация должна обеспечивать для всех приложений прозрачный доступ к любым данным независимо от их местоположения и формата.

Основные принципы создания и функционирования распределенных баз данных:

- прозрачность расположения данных для пользователя (иначе говоря, для пользователя распределенная база данных должна представляться и выглядеть точно так же, как и нераспределенная);

- изолированность пользователей друг от друга (пользователь должен «не чувствовать», «не видеть» работу других пользователей в тот момент, когда он изменяет, обновляет, удаляет данные);

- синхронизация и согласованность (непротиворечивость) состояния данных в любой момент времени.

Из основных вытекает ряд дополнительных принципов:

- локальная автономия (ни одна вычислительная установка для своего успешного функционирования не должна зависеть от любой другой установки);

- отсутствие центральной установки (следствие предыдущею пункта);

- независимость от местоположения (пользователю все равно, где физически находятся данные, он работает так, как будто они находятся на его локальной установке);

- непрерывность функционирования (отсутствие плановых отключений системы в целом, например, для подключения новой установки или обновления версии СУБД);

- независимость от фрагментации данных (как от горизонтальной фрагментации, когда различные группы записей одной таблицы размещены на различных установках или в различных локальных базах, так и от вертикальной фрагментации, когда различные поля-столбцы одной таблицы размещены на разных установках);

- независимость от реплицирования (дублирования) данных (когда какая-либо таблица базы данных (или ее часть) физически может быть представлена несколькими копиями, расположенными на различных установках);

- распределенная обработка запросов (оптимизация запросов должна носить распределенный характер - сначала глобальная оптимизация, а далее локальная оптимизация на каждой из задействованных установок);

- распределенное управление транзакциями (в распределенной системе отдельная транзакция может требовать выполнения действий на разных

установках, транзакция считается завершенной, если она успешно завершена на всех вовлеченных установках);

- независимость от аппаратуры (желательно, чтобы система могла функционировать на установках, включающих компьютеры разных типов);

- независимость от типа операционной системы (система должна функционировать вне зависимости от возможного различия ОС на различных вычислительных установках);

- независимость от коммуникационной сети (возможность функционирования в разных коммуникационных средах);

- независимость от СУБД (на разных установках могут функционировать СУБД различного типа, на практике ограничиваемые кругом СУБД, поддерживающих SQL).

В обиходе СУБД, на основе которых создаются распределенные информационные системы, также характеризуют термином «распределенные СУБД», и, соответственно, используют термин «распределенные базы данных».

Практическая реализация распределенных вычислений осуществляется через отступление от некоторых рассмотренных выше принципов создания и функционирования распределенных систем. В зависимости от того, какой принцип приносится в «жертву» (отсутствие центральной установки, непрерывность функционирования, согласованного состояния данных и др.) выделились несколько самостоятельных направлений в технологиях распределенных систем - технологии «Клиент-сервер», технологии реплицирования, технологии объектного связывания.

Реальные распределенные информационные системы, как правило, построены на основе сочетания всех трех технологий, но в методическом плане их целесообразно рассмотреть отдельно.

В *технологиях «Клиент-сервер»* отступают от одного из главных принципов создания и функционирования распределенных систем -

отсутствия центральной установки. Поэтому можно выделить две основные идеи, лежащие в основе клиент-серверных технологий:

- общие для всех пользователей данные на одном или нескольких серверах;

- много пользователей (клиентов), на различных вычислительных установках, совместно (параллельно и одновременно) обрабатывающих общие данные.

Иначе говоря, системы, основанные на технологиях «Клиент-сервер», распределены только в отношении пользователей, поэтому часто их не относят к «настоящим» распределенным системам, а считают отдельным классом многопользовательских систем.

Важное значение в технологиях «Клиент-сервер» имеют понятия сервера и клиента.

Под сервером в широком смысле понимается любая система, процесс, компьютер, владеющие каким-либо вычислительным ресурсом (памятью, временем, производительностью процессора и т. д.).

Клиентом называется также любая система, процесс, компьютер, пользователь, запрашивающие у сервера какой-либо ресурс, пользующиеся каким-либо ресурсом или обслуживаемые сервером иным способом.

В своем развитии системы «Клиент-сервер» прошли несколько этапов, в ходе которых сформировались различные модели систем «Клиент-сервер». Их реализация и, следовательно, правильное понимание основаны на разделении структуры СУБД на три компонента:

- компонент представления, реализующий функции ввода и отображения данных, называемый иногда еще просто как интерфейс пользователя;

- прикладной компонент, включающий набор запросов, событий, правил, процедур и других вычислительных функций, реализующий предназначение автоматизированной информационной системы в конкретной предметной области;

- компонент доступа к данным, реализующий функции хранения-извлечения, физического обновления и изменения данных.

Исходя из особенностей реализации и распределения в системе этих трех компонентов различают четыре модели технологий «Клиент-сервер»:

- модель файлового сервера (File Server - FS);
- модель удаленного доступа к данным (Remote Data Access - RDA);
- модель сервера базы данных (DataBase Server - DBS);
- модель сервера приложений (Application Server - AS).

Модель файлового сервера является наиболее простой и характеризует не столько способ образования информационной системы, сколько общий способ взаимодействия компьютеров в локальной сети. Один из компьютеров сети выделяется и определяется файловым сервером, т. е. общим хранилищем любых данных. Суть FS- модели иллюстрируется схемой, приведенной на рис. 3.1.

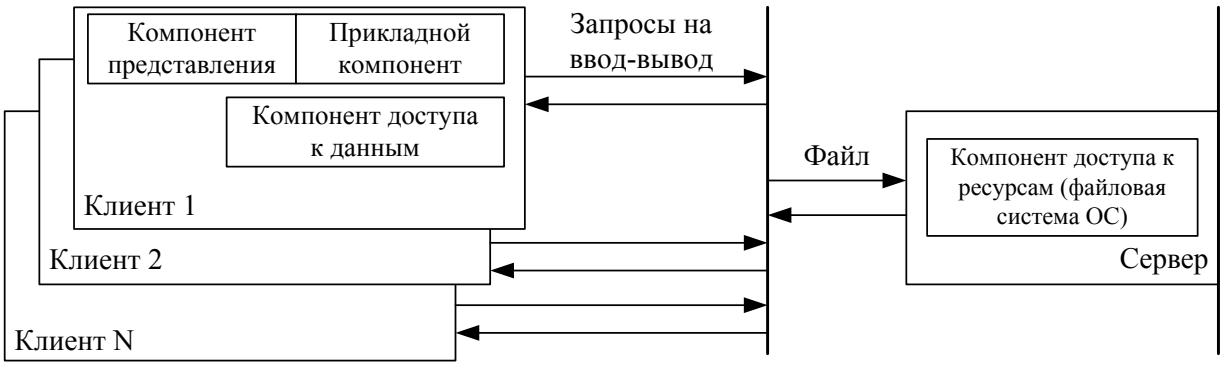


Рис. 3.1. Модель файлового сервера

В FS-модели все основные компоненты размещаются на клиентской установке. При обращении к данным ядро СУБД, в свою очередь, обращается с запросами на ввод-вывод данных за сервисом к файловой системе. С помощью функций операционной системы в оперативную память клиентской установки полностью или частично на время сеанса работы копируется файл базы данных. Таким образом, сервер в данном случае выполняет чисто пассивную функцию.

Достоинством данной модели являются ее простота, отсутствие высоких требований к производительности сервера (главное, требуемый объем дискового пространства). Следует также отметить, что программные компоненты СУБД в данном случае не распределены, т.е. никакая часть СУБД на сервере не инсталлируется и не размещается.

Недостатки данной модели - высокий сетевой трафик, достигающий пиковых значений особенно в момент массового вхождения в систему пользователей, например в начале рабочего дня. Однако более существенным недостатком, с точки зрения работы с общей базой данных, является отсутствие специальных механизмов безопасности файла (файлов) базы данных со стороны СУБД. Иначе говоря, разделение данных между пользователями (параллельная работа с одним файлом данных) осуществляется только средствами файловой системы ОС для одновременной работы нескольких прикладных программ с одним файлом.

Несмотря на очевидные недостатки, модель файлового сервера является естественным средством расширения возможностей персональных (настольных) СУБД в направлении поддержки многопользовательского режима и, очевидно, в этом плане еще будет сохранять свое значение.

Модель удаленного доступа к данным основана на учете специфики размещения и физического манипулирования данных во внешней памяти для реляционных СУБД. В RDA-модели компонент доступа к данным в СУБД полностью отделен от двух других компонентов (компонента представления и прикладного компонента) и размещается на сервере системы.

Компонент доступа к данным реализуется в виде самостоятельной программной части СУБД, называемой SQL-сервером, и инсталлируется на вычислительной установке сервера системы. Функции SQL-сервера ограничиваются низкоуровневыми операциями по организации, размещению, хранению и манипулированию данными в дисковой памяти сервера. Иначе говоря, SQL-сервер играет роль машины данных. Схема RDA-модели приведена на рис. 3.2.

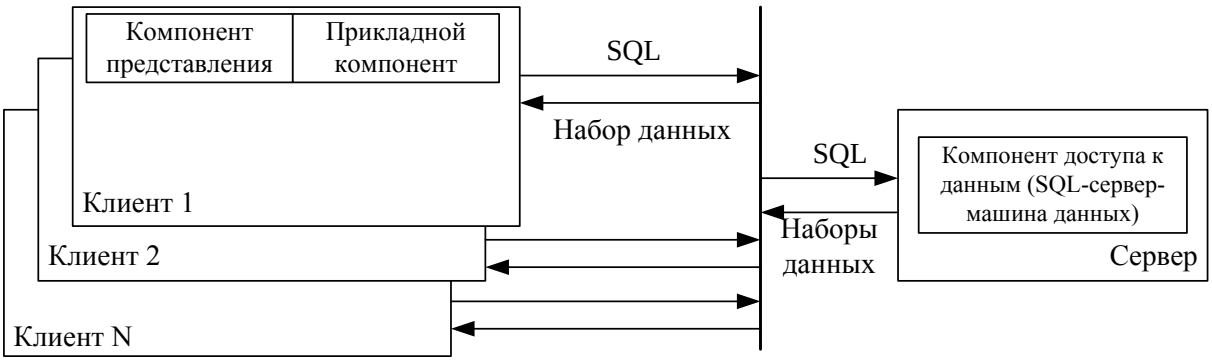


Рис 3.2. Модель удаленного доступа к данным (RDA-модель)

В файле (файлах) базы данных, размещаемом на сервере системы, находится также и системный каталог базы данных, в который помещаются в том числе и сведения о зарегистрированных клиентах, их полномочиях и т. п.

На клиентских установках инсталлируются программные части СУБД, реализующие интерфейсные и прикладные функции. Пользователь, входя в клиентскую часть системы, регистрируется через нее на сервере системы и начинает обработку данных.

Прикладной компонент системы (библиотеки запросов, процедуры обработки данных) полностью размещается и выполняется на клиентской установке. При реализации своих функций прикладной компонент формирует необходимые SQL-инструкции, направляемые SQL-серверу. SQL-сервер, представляющий специальный программный компонент, ориентированный на интерпретацию SQL-инструкций и высокоскоростное выполнение низкоуровневых операций с данными, принимает и координирует SQL-инструкции от различных клиентов, выполняет их, проверяет и обеспечивает выполнение ограничений целостности данных и направляет клиентам результаты обработки SQL-инструкций, представляющие, как известно, наборы (таблицы) данных.

Таким образом, общение клиента с сервером происходит через SQL-инструкции, а с сервера на клиентские установки передаются только результаты обработки, т. е. наборы данных, которые могут быть существенно меньше по объему всей базы данных. В результате резко уменьшается

загрузка сети, а сервер приобретает активную центральную функцию. Кроме того, ядро СУБД в виде SQL-сервера обеспечивает также традиционные и важные функции по обеспечению ограничений целостности и безопасности данных при совместной работе нескольких пользователей.

Другим, может быть неявным, достоинством RDA-модели является унификация интерфейса взаимодействия прикладных компонентов информационных систем с общими данными. Такое взаимодействие стандартизовано в рамках языка SQL специальным протоколом ODBC (Open Database Connectivity - открытый доступ к базам данных), играющим важную роль в обеспечении интероперабельности (многопротокольности), т.е. независимости от типа СУБД на клиентских установках в распределенных системах.

Интероперабельность СУБД - способность СУБД обслуживать прикладные программы, первоначально ориентированные на разные типы СУБД. Иначе говоря, специальный компонент ядра СУБД на сервере (так называемый драйвер ODBC) способен воспринимать, обрабатывать запросы и направлять результаты их обработки на клиентские установки, функционирующие под управлением реляционных СУБД других, не «родных» типов.

Такая возможность существенно повышает гибкость в создании распределенных информационных систем на базе интеграции уже существующих в какой-либо организации локальных баз данных под управлением настольных или другого типа реляционных СУБД.

К недостаткам RDA-модели можно отнести высокие требования к клиентским вычислительным установкам, так как прикладные программы обработки данных, определяемые спецификой предметной области информационной системы, выполняются на них.

Другим недостатком является все же существенный трафик сети, обусловленный тем, что с сервера базы данных клиентам направляются

наборы (таблицы) данных, которые в определенных случаях могут занимать достаточно существенный объем.

Модель сервера базы данных. Развитием PDA-модели стала модель сервера базы данных. Ее сердцевиной является механизм хранимых процедур. В отличие от PDA-модели, определенные для конкретной предметной области информационной системы события, правила и процедуры, описанные средствами языка SQL, хранятся вместе с данными на сервере системы и на нем же выполняются. Иначе говоря, прикладной компонент полностью размещается и выполняется на сервере системы. Схематично DBS-модель приведена на рис. 3.3.

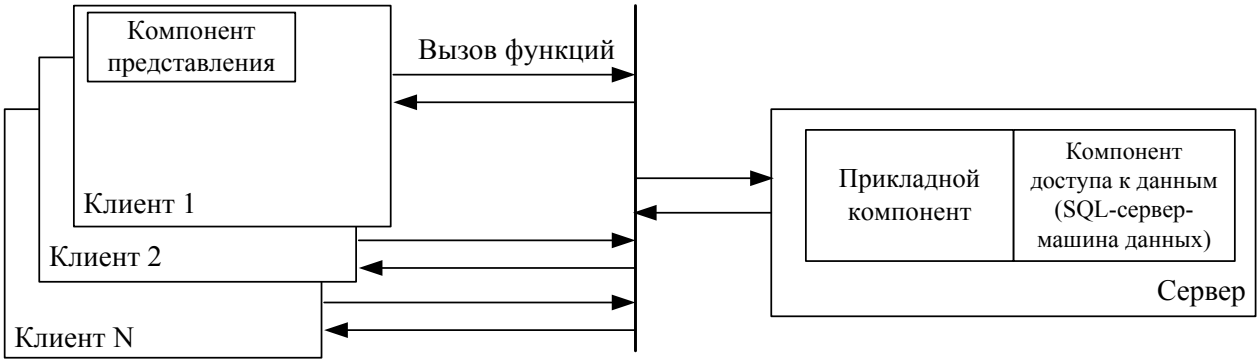


Рис. 3.3. Модель сервера базы данных

На клиентских установках в DBS-модели размещается только интерфейсный компонент (компонент представления), что существенно снижает требования к вычислительной установке клиента. Пользователь через интерфейс системы на клиентской установке направляет на сервер базы данных только лишь вызовы необходимых процедур, запросов и других функций по обработке данных. Все затратные операции по доступу и обработке данных выполняются на сервере и клиенту направляются лишь результаты обработки, а не наборы данных, как в RDA-модели. Этим обеспечивается существенное снижение трафика сети в DBS-модели по сравнению с RDA -моделью.

Следует заметить, что на сервере системы выполняются процедуры прикладных задач одновременно всех пользователей системы. В результате резко возрастают требования к вычислительной установке сервера, причем как к объему дискового пространства и оперативной памяти, так и к быстродействию. Это основной недостаток DBS-модели.

К достоинствам же DBS-модели, помимо разгрузки сети, относится и более активная роль сервера сети, размещение, хранение и выполнение на нем механизма событий, правил и процедур, возможность более адекватно и эффективно «настраивать» распределенную информационную систему на все нюансы предметной области.

Также более надежно обеспечивается согласованность состояния и изменения данных и, вследствие этого, повышается надежность хранения и обработки данных, эффективно координируется коллективная работа пользователей с общими данными.

Чтобы разнести требования к вычислительным ресурсам сервера в отношении быстродействия и памяти по разным вычислительным установкам, используется *модель сервера приложений*.

Суть AS-модели заключается в переносе прикладного компонента информационной системы на специализированный, в отношении повышенных ресурсов по быстродействию, дополнительный сервер системы. Схема AS-модели приведена на рис. 3.4.

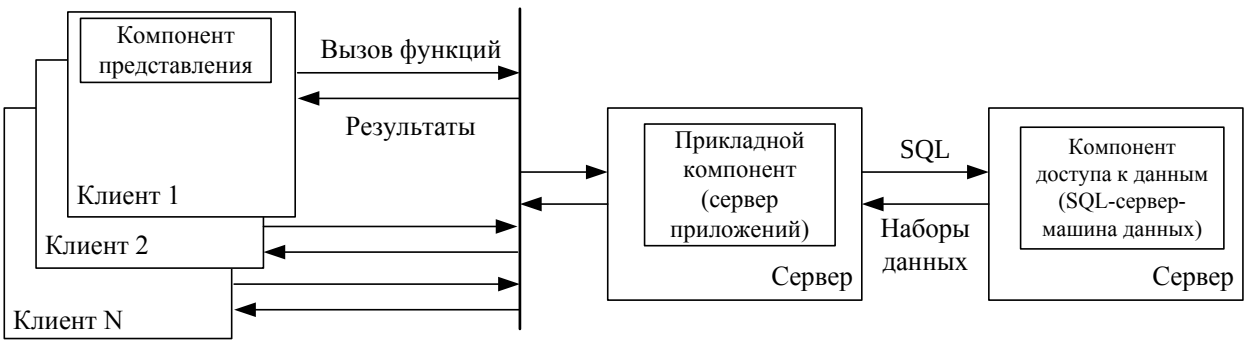


Рис. 3.4. Модель сервера приложений (AS-модель)

Как и в DBS-модели, на клиентских установках располагается только интерфейсная часть системы, т. е. компонент представления. Однако вызовы функций обработки данных направляются на сервер приложений, где эти функции совместно выполняются для всех пользователей системы. За выполнением низкоуровневых операций по доступу и изменению данных сервер приложений, как в RDA-модели, обращается к SQL-серверу, направляя ему вызовы SQL-процедур, и получая, соответственно, от него наборы данных.

Как известно, последовательная совокупность операций над данными (SQL-инструкций), имеющая отдельное смысловое значение, называется транзакцией.

В этом отношении сервер приложений управляет формированием транзакций, которые выполняет SQL-сервер. Поэтому программный компонент СУБД, инсталлируемый на сервере приложений, еще называют монитором обработки транзакций (Transaction Processing Monitors - TRM), или просто монитором транзакций.

AS-модель, сохраняя сильные стороны DBS-модели, позволяет оптимально построить вычислительную схему информационной системы, однако, как и в случае RDA-модели, повышает трафик сети.

В практических случаях используются смешанные модели, когда простейшие прикладные функции и обеспечение ограничений целостности данных поддерживаются хранимыми на сервере процедурами (DBS-модель), а более сложные функции предметной области (так называемые правила бизнеса) реализуются прикладными программами на клиентских установках (RDA-модель) или на сервере приложений (AS-модель).

Контрольные вопросы

1. Основные принципы создания и функционирования распределенных баз данных.
2. Перечислите модели технологии «Клиент-сервер».
3. Модель файлового сервера.

4. Модель удаленного доступа к данным.
5. Модель сервера базы данных.
6. Модель сервера приложений.

3.2. Безопасность базы данных в централизованных многопользовательских информационных системах

Существует ряд проблем обеспечения целостности распределенной БД, помимо тех аспектов, которые присущи любым БД:

- возможность одновременного доступа нескольких пользователей к одной и той же информации (особенно если эти обращения к БД - корректирующие);

- физический разброс отдельных частей БД по разным компьютерам;

- разнотипность источников информации.

Первая проблема имеет место в любых распределенных БД, вторая - если база данных является распределенной, третья - если система является гетерогенной.

Первая группа проблем в области обеспечения целостности в распределенных БД обусловлена в основном возникновением опасности искажения данных при их одновременной корректировке разными пользователями.

Возможны разные схемы обеспечения целостности данных при выполнении корректирующих обращений в многопользовательском режиме:

- запрещение корректировки информации, если ее корректирует другой пользователь (блокировка);

- корректировка разных копий информационных единиц и последующее устранение возникающих коллизий.

Если СУБД предоставляет возможность выбора способа обеспечения целостности при многопользовательских обращениях, то на результат этого выбора будут влиять многие факторы, в том числе:

- степень конкуренции при выполнении корректирующих обращений - насколько часто возникает ситуация одновременной корректировки одной и той же информационной единицы;
- ограничения на время реакции системы,
- требования к актуальности и непротиворечивости данных в каждый момент времени;
- характеристика технических средств.

Вторая группа проблем обеспечения целостности в распределенных системах вызвана распределением данных и, как следствие, распределением процедур их обработки, т.е. это проблемы, обусловленные именно разнесением данных на разные узлы системы.

Как известно, существует два подхода к обеспечению целостности в распределенных информационных системах - строгая целостность (tight consistency) и нестрогая целостность (loose consistency). Первый вариант гарантирует целостность данных в любой момент времени, например, с помощью двухфазного протокола фиксаций (2PC). Обеспечение строгой целостности требует высокого качества коммуникаций, поскольку все узлы должны быть постоянно доступны. Второй подход допускает наличие временной задержки между внесением изменений в публикуемую базу и их отражением на узлах подписчиков.

Протокол двухфазной фиксации транзакции состоит в последовательном прохождении базы данных в процессе выполнения транзакции через два этапа. Первый этап (первая фаза) - выполнение синхронизированного захвата всех объектов данных, к которым имело место обращение от имени транзакции. Объекты данных захватываются на всех серверах. На втором этапе (во второй фазе) либо происходят все изменения на всех серверах, либо, в случае хотя бы одной ошибки, происходит откат к состоянию, в котором находилась база данных до выполнения первого этапа.

Механизм двухфазной фиксации транзакции имеет ряд недостатков:

- захват всех необходимых данных на всех серверах может надолго заблокировать доступ к данным;
- велика вероятность отказа от обновления из-за какой-нибудь, пусть единичной, ошибки;
- если какой-либо сервер или выход в глобальную сеть окажется недоступным, произойдет потеря транзакции;
- использование в структуре сети координирующего узла связано с дополнительной опасностью, поскольку выход его из строя приведет к блокировке данных, затронутых транзакцией, до тех пор, пока он не будет восстановлен;
- сложность обработки транзакции, при использовании этого протокола сама служит источником дополнительного трафика, что увеличивает время реакции системы.

Кроме того, недостаточная пропускная способность сети и малая скорость передачи данных могут увеличить время реакции до недопустимого уровня.

Разные СУБД поддерживают разные технологии обеспечения целостности.

Способы защиты данных. При работе в многопользовательском режиме особую актуальность приобретает защита данных от несанкционированного доступа. Существуют различные приемы управления доступом к базе данных. Эти приемы обеспечивают разный уровень безопасности. Некоторые из них присущи любым информационным системам, другие - конкретным СУБД.

Шифрование. Шифрование базы данных - это простейший способ защиты, при котором ее файл видоизменяется и становится недоступным для чтения с помощью стандартных служебных программ или текстовых редакторов. Шифрование обычно применяется при электронной передаче базы данных или сохранении ее на внешний носитель.

Дешифрование базы данных - это операция, обратная шифрованию.

Скрытие объектов. Другим способом защиты объектов в базе данных от посторонних пользователей является скрытие всей базы данных при просмотре каталогов средствами операционной системы или скрытие отдельных объектов БД при работе с базой данных средствами конкретной СУБД. Этот способ защиты не является достаточно надежным, поскольку скрытые объекты относительно просто можно отобразить.

Использование параметров запуска. Эти параметры позволяют задать стартовую форму, которая автоматически открывается при открытии базы данных. При определении формы можно скрыть окно базы данных, предоставляемое СУБД, и установить собственную кнопочную форму. При этом пользователь может выполнять с базой данных только те действия, которые допускает интерфейс, предоставляемый данным приложением.

Использование пароля. Другим простейшим способом защиты является установка пароля для открытия базы данных. После установки пароля при каждом открытии базы данных будет появляться диалоговое окно, в которое требуется ввести пароль. Только те пользователи, которые введут правильный пароль, смогут открыть базу данных.

В принципе может быть установлен единый пароль для всех пользователей БД. Но наиболее гибким и распространенным способом защиты базы данных является защита на уровне пользователя, при котором каждому пользователю присваивается пароль. При запуске СУБД каждый пользователь должен быть идентифицирован; система проверяет соответствие идентификатора пользователя и его пароля.

Для каждого пользователя могут быть определены не только уникальный код, но и уровень доступа, и объекты, доступ к которым получает пользователь.

Многие СУБД позволяют кроме единичных пользователей создавать еще и их группы.

Запрещение репликации базы данных. Как указывалось выше, репликация позволяет пользователям создавать копию общей базы данных.

Это может быть в принципе использовано и для дальнейшего нелегального распространения реплицированных данных. Поэтому в некоторых случаях может потребоваться запретить репликацию базы данных.

Запрещение установки паролей и настройки параметров запуска пользователями. При многопользовательском доступе к данным важно обеспечить не только сохранность и конфиденциальность данных, но и возможность доступа к данным всем тем пользователям данных, для которых это необходимо. Поэтому СУБД должна иметь механизмы, не позволяющие любым пользователям устанавливать пароль на БД.

Также желательно иметь механизм установки запрета на изменение параметров запуска, которые определяют такие свойства, как настраиваемые меню, настраиваемые панели инструментов и стартовую форму.

Создание и удаление пользователей. При работе в многопользовательской среде большое значение приобретает понятие пользователь базы данных - владелец определенного набора объектов базы данных.

Пользователи системы могут быть разделены на классы. В системе любого размера всегда имеются некоторые типы суперпользователей - пользователей, которые автоматически имеют большинство (или все) привилегий и могут передать свой статус суперпользователя кому-нибудь с помощью привилегии или группы привилегий. Администратор базы данных *(DBA)* является термином, наиболее часто используемым для такого суперпользователя и для привилегий, которыми он обладает.

Других пользователей создают администраторы баз данных; они же дают им начальные привилегии. Создавать пользователей могут только администраторы. Давать права пользователям и отбирать их могут не только администраторы, но и другие пользователи, обладающие соответствующими правами.

Пользователи могут объединяться в группы. Группа пользователей - это пользователи, наделенные одинаковым набором привилегий. Один и тот

же пользователь в принципе может входить в разные группы. Каждый пользователь имеет специальное идентификационное имя или номер (Authorization ID).

Поскольку большинство промышленно эксплуатируемых корпоративных СУБД являются SQL-серверами, рассмотрим вопросы управления пользователями на примере SQL-систем.

Конкретные формы процесса управления пользователями в различных СУБД могут значительно отличаться друг от друга. Процесс управления пользователями в большой мере зависит от используемой операционной системы, архитектуры БД. В связи с этим в соответствующей части SQL наблюдается большая зависимость от платформы и производителя.

Процесс управления пользователями можно разбить на три главных этапа. Сначала необходимо создать учетную запись пользователя в базе данных. Далее пользователя необходимо наделить привилегиями сообразно тем задачам, которые пользователь предположительно будет решать в рамках базы данных. Наконец, после того, как доступ к данным пользователю будет уже не нужен, необходимо либо удалить из базы данных его учетную запись, либо отменить ранее предоставленные ему привилегии.

Перед началом работы с БД пользователь должен быть идентифицирован с помощью процедуры входа, обычно включающей запрос имени и пароля пользователя. После входа запускается сеанс (sessions) работы с СУБД.

Определение и отмена привилегий. Распределенные БД предполагают работу с базой данных многих пользователей. Однако не всем пользователям следует разрешать выполнять любые действия с базой данных. Поэтому пользователям предоставляются привилегии.

Привилегии в базе данных делятся на две категории: системные привилегии (system privileges) и объектные привилегии (object privileges). Системные привилегии контролируют общий доступ к базе данных. К ним

относятся право создавать таблицы и другие объекты, а также право администрировать базу данных.

Объектные привилегии связаны с конкретным объектом базы данных. Объектная привилегия состоит из трех частей:

- объекта, к которому применяется привилегия;
- операции, которые она разрешает;
- пользователя, которому даются эти привилегии.

Одна из первых привилегий, которая должна быть определена, - это привилегия создателей таблиц. Если все пользователи будут иметь возможность создавать в системе базовые таблицы, это может привести к избыточности данных, их несогласованности и, как следствие, к неэффективности системы.

Пользователь, создавший таблицу, является ее владельцем. Это означает, что пользователь имеет все привилегии в созданной им таблице и может передавать привилегии другим пользователям.

Каждый пользователь в среде SQL имеет специальное идентификационное имя (или номер).

Привилегии даются оператором GRANT (ПРЕДОСТАВИТЬ) и отменяются оператором REVOKE (ОТМЕНИТЬ).

Оператор GRANT имеет следующий синтаксис:

GRANT привилегия.,..ON имя объекта

TO {пользователь, которому предоставляется привилегия.,..}|PUBLIK

[WITH GRANT OPTION];

привилегия:=

{ALL PRIVILEGES}

| {SELECT

| DELETE

| {INSERT [(имя столбца.,..)]}

| {UPDATE [(имя столбца.,..)]}

| {REFERENCES [(имя столбца.,..)]}

| USAGE}.

Когда SQL получает оператор GRANT, он проверяет привилегии пользователя, подавшего эту команду, чтобы определить, допустим ли оператор GRANT

Для пользователя таблицы могут быть назначены следующие типы привилегий:

- SELECT - разрешение выполнять запросы в таблице.
- INSERT - разрешение выполнять оператор INSERT (вставка новой строки) в таблице.
- UPDATE - разрешение выполнять оператор UPDATE (обновле¬ние значений полей) в таблице. Можно ограничить эту привилегию для определенных столбцов таблицы.
- DELETE - разрешение выполнять оператор DELETE (удаление записей) в таблице.
- REFERENCES - разрешение определить внешний ключ.

В одном операторе GRANT можно назначить несколько привилегий, перечислив их через запятую, или использовать аргумент ALL, означающий, что пользователю передаются все привилегии для данной таблицы.

В одном операторе GRANT можно назначить привилегии нескольким пользователям одновременно, перечислив их через запятую, или использовать аргумент PUBLIC, означающий, что привилегии передаются все пользователям. Однако последней возможностью нужно пользоваться с осторожностью, так как PUBLIC означает не только текущих пользователей, но и всех пользователей, которые могут быть введены в систему в дальнейшем.

Предположим, что пользователь Mansurov является владельцем таблицы «Sotrudnik» и хочет позволить пользователю Karimov выполнять запросы к ней. В этом случае пользователь Mansurov должен ввести команду

GRANT SELECT ON Sotrudnik TO Karimov;

Предложение WITH GRANT OPTION позволяет передать пользователю возможность назначать привилегии для этой таблицы. Если, например, команда выглядит следующим образом:

GRANT SELECT ON Sotrudnik

TO Karimov WITH GRANT OPTION;

то пользователь Karimov получает возможность, в свою очередь, передавать право назначать привилегии другим пользователям, т. е. пользователь Karimov может задать следующую команду:

GRANT SELECT ON Mansurov.Sotrudnik

TO Salimov WITH GRANT OPTION;

Обратите внимание на то, что когда на таблицу ссылается пользователь, не являющийся владельцем схемы, то перед именем таблицы указывается имя схемы.

Большинство привилегий объекта использует один и тот же синтаксис. Из перечисленных выше привилегий исключение составляют UPDATE и REFERENCES.

При задании привилегии UPDATE можно использовать тот же синтаксис, который применялся выше. Это будет означать, что пользователю дается право обновлять содержимое всех столбцов таблицы. Можно также после названия привилегии в скобках указать имена столбцов (если их несколько, то имена указываются в любой последовательности через запятую), на которые распространяется данная привилегия. Например, привилегия UPDATE может выглядеть следующим образом:

GRANT UPDATE (dolgnost, oklad) ON Sotrudnik TO Karimov;

При задании привилегии REFERENCES также задаются имена столбцов.

Чтобы ограничить возможность просмотра таблицы только отдельными столбцами, следует воспользоваться механизмом создания представлений и назначать привилегии не для реальной таблицы, а для представления. Представления могут использоваться также и для

обеспечения возможности ограничить просмотр только определенными строками.

Отмена привилегий осуществляется с помощью оператора REVOKE. Эта команда имеет синтаксис, схожий с синтаксисом оператора GRANT. Например, отмена привилегии на просмотр таблицы «Sotrudnik» для пользователя Karimov будет выглядеть следующим образом:

REVOKE SELECT ON Sotrudnik TO Karimov;

В конкретных СУБД могут поддерживаться привилегии, отличающиеся от привилегий, приведенных выше. Так, в некоторых СУБД имеется возможность задавать привилегию INDEX, которая позволяет пользователям создавать индексы. Но объект INDEX в стандарте SQL не определен, и синтаксис команды задания этой привилегии может отличаться от системы к системе.

Стандартом SQL не определено, кто имеет право отменять привилегии. Однако обычно действует подход, при котором привилегии отменяются тем пользователем, который их предоставил.

Контрольные вопросы

1. Проблемы обеспечения целостности распределенной базы данных.
2. Объясните протокол двухфазной фиксации транзакции.
3. Способы шифрования и дешифрования базы данных.
4. Как происходит отображение и скрытие объектов?
5. Объясните механизм использования параметров запуска.
6. Использование паролей при защите распределенной базы данных.
7. Как назначаются и отменяются привилегии?

3.3.Технология объектного связывания данных

Унификация взаимодействия прикладных компонентов с ядром информационных систем в виде SQL-серверов, наработанная для клиент-серверных систем, позволила выработать аналогичные решения и для

интеграции разрозненных локальных баз данных под управлением настольных СУБД в сложные децентрализованные гетерогенные распределенные системы. Такой подход получил название *объектного связывания данных.*

С узкой точки зрения, технология объектного связывания данных решает *задачу обеспечения доступа из одной локальной базы*, открытой одним пользователем, *к данным в другой локальной базе*, возможно находящейся на другой вычислительной установке, открытой и эксплуатируемой другим пользователем.

Решение этой задачи основывается на поддержке современными «настольными» СУБД (MS Access, MS FoxPro, dBase и др.) технологии *«объектов доступа к данным» — DAO*. При этом следует отметить, что под объектом понимается интеграция данных и методов их обработки в одно целое (объект), на чем, как известно, основываются объектно-ориентированное программирование и современные объектно-ориентированные операционные среды. Другими словами, СУБД, поддерживающие DAO, получают возможность внедрять и оперировать в локальных базах объектами доступа к данным, физически находящимся в других файлах, возможно на других вычислительных установках и под управлением других СУБД.

Технически технология DAO основана на уже упоминавшийся *протоколе ODBC*, который принят за *стандарт доступа* не только *к данным* на SQL-серверах клиент-серверных систем, но и в качестве стандарта доступа к любым данным под управлением реляционных СУБД. Непосредственно для доступа к данным на основе протокола ODBC используются инициализируемые на тех установках, где находятся данные, специальные программные компоненты, называемые *драйверами ODBC*, или инициализируемые ядра тех СУБД, под управлением которых были созданы и эксплуатируются внешние базы данных. Схематично принцип и

особенности доступа к внешним базам данных на основе объектного связывания иллюстрируются на рис. 3.5.

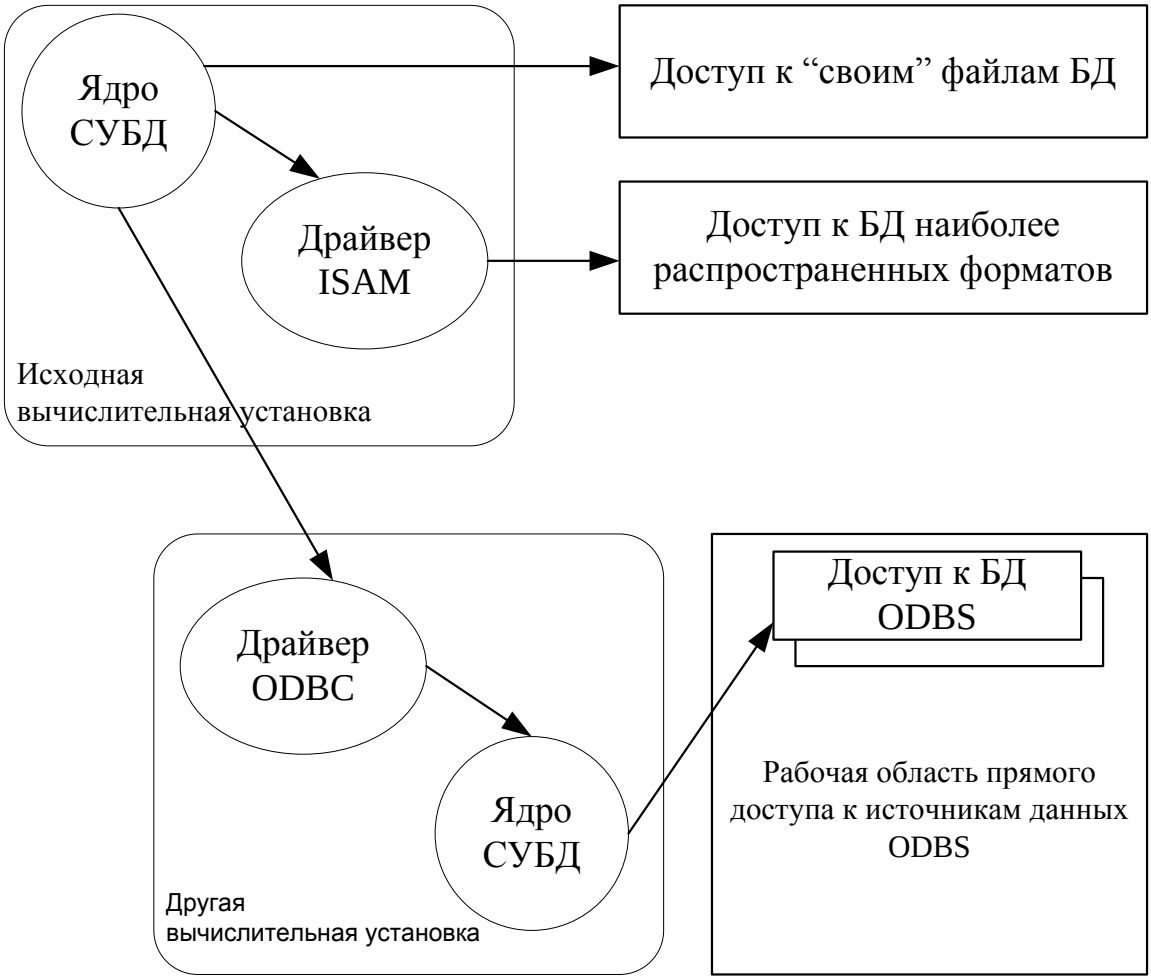


Рис. 3.5. Принцип доступа к внешним данным на основе протокола ODBC

Прежде всего, современные настольные СУБД обеспечивают возможность *прямого доступа к объектам* (таблицам, запросам, формам) *внешних баз данных «своих» форматов.* Иначе говоря, в открытую в текущем сеансе работы базу данных пользователь имеет возможность вставить специальные ссылки-объекты и оперировать с данными из другой (внешней, т. е. не открываемой специально в данном сеансе) базы данных. *Объекты из внешней базы данных, вставленные в текущую базу данных,* называются *связанными,* и, как правило, имеют специальные обозначения для отличия от внутренних объектов. При этом следует подчеркнуть, что *сами данные физически* в файл (файлы) текущей базы данных не помещаются, а

остаются в файлах «своих» баз данных. В системный каталог текущей базы данных помещаются все необходимые для доступа сведения о связанных объектах — внутреннее имя и внешнее, т. е. истинное имя объекта во внешней базе данных, полный путь к файлу внешней базы и т. п.

Связанные объекты для пользователя ничем не отличаются от *внутренних объектов.* Пользователь может также открывать связанные во внешних базах таблицы данных, осуществлять поиск, изменение, удаление и добавление данных, строить запросы по таким таблицам и т. д. Связанные объекты можно интегрировать в схему внутренней базы данных, т. е. устанавливать *связи между внутренними и связанными таблицами.*

Технически оперирование связанными объектами из внешних баз данных «своего» формата мало отличается от оперирования сданными из текущей базы данных. *Ядро СУБД* при обращении к данным связанного объекта по системному каталогу текущей базы данных находит сведения о месте нахождения и других параметрах соответствующего файла (файлов) внешней базы данных и прозрачно, т. е. невидимо для *пользователя открывает* этот *файл* (файлы), а далее обычным порядком организует в оперативной памяти *буферизацию страниц внешнего файла данных для непосредственно доступа и манипулирования данными.* Следует также заметить, что на основе возможностей многопользовательского режима работы с файлами данных современных операционных систем, с файлом внешней базы данных, если он находится на другой вычислительной установке, может *в тот же момент времени работать и другой пользователь,* что и обеспечивает коллективную обработку общих распределенных данных.

На рис. 3.6 приведен пример схемы локальных баз данных, использующих совместные данные по линии информационного обеспечения производства и сбыта продукции. Стрелками на рисунке показаны связи типа «один-ко-многим» (Острие стрелки соответствует стороне «многое»).

Нетрудно заключить, что подобный принцип построения распределенных систем при больших объемах данных в связанных таблицах приведет к *существенному увеличению трафика сети,* так как по сети постоянно передаются, даже не наборы данных, а страницы файлов баз данных, что может приводить к пиковым перегрузкам сети. Поэтому представленные схемы локальных баз данных со взаимными связанными объектами нуждаются в дальнейшей тщательной проработке с точки зрения интенсивности, направленности потоков данных в сети между локальными базами исходя из информационных технологий, обусловленных производственно-технологическими и организационными процессами.

Не менее существенной проблемой является *отсутствие надежных механизмов безопасности данных* и обеспечения *ограничений целостности.* Так же как и в модели файлового сервера, совместная работа нескольких пользователей с одними и теми же данными обеспечивается только функциями операционной системы по одновременному доступу к файлу нескольких приложений.

Аналогичным образом обеспечивается доступ к данным, находящимся в *базах данных наиболее распространенных форматов* других СУБД, таких, например, как базы данных СУБД FoxPro, dBASE, а так же к табличным данным.

При этом доступ может обеспечиваться как *непосредственно ядром СУБД,* так и специальными дополнительными драйверами ISAM (Indexed Sequential Access Method), входящими, как правило, в состав комплекта СУБД. Такой подход реализует *интероперабельность* построенных подобным образом распределенных гетерогенных систем, т.е. «разномастность» типов СУБД, поддерживающих локальные базы данных. При этом, однако, объектное связывание ограничивается только непосредственно таблицами данных, исключая другие объекты базы данных (запросы, формы, отчеты), реализация и поддержка которых зависят от специфики конкретной СУБД.

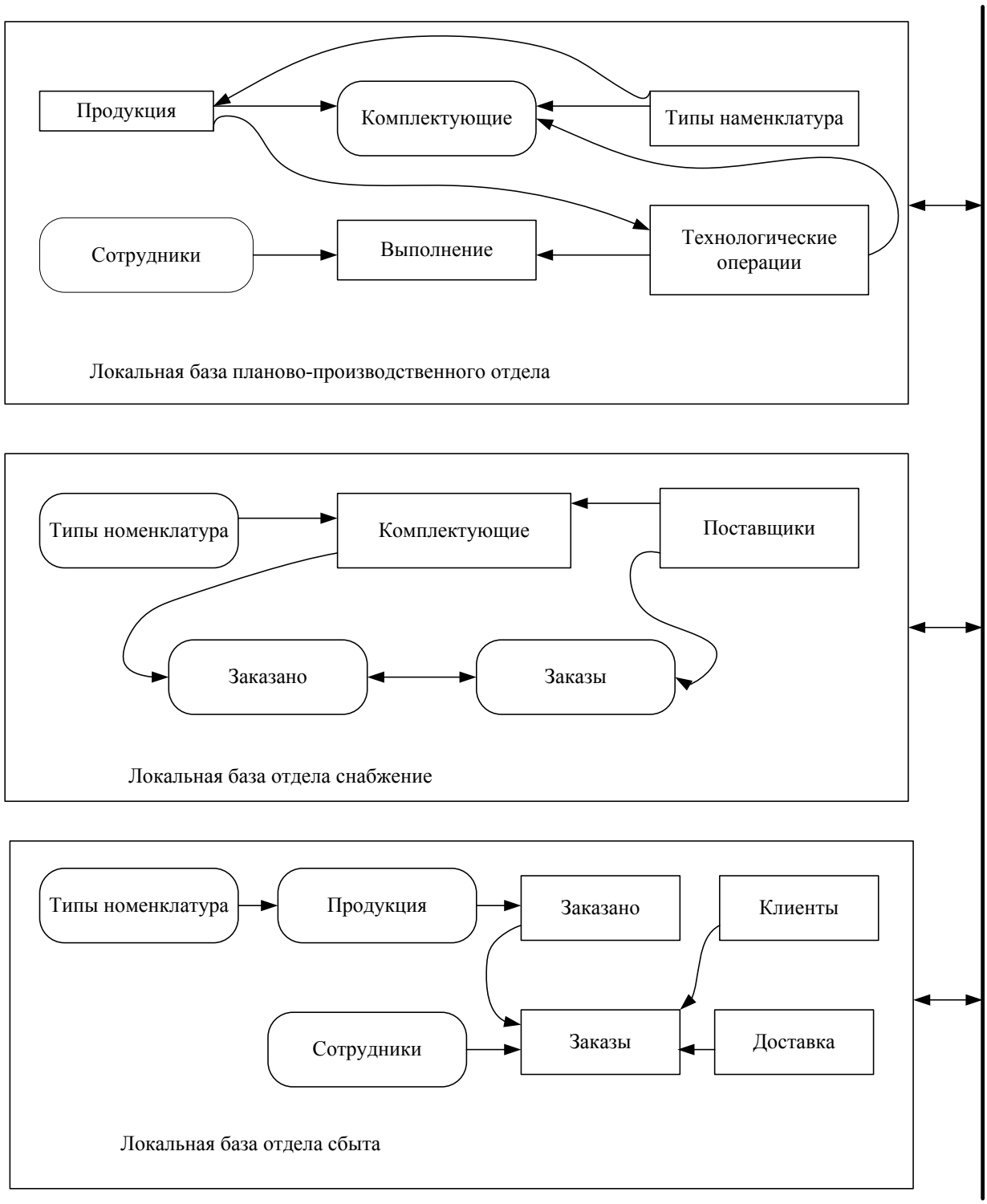


Рис. 3.6. Пример схем локальных баз данных со связанными объектами

Доступ к базам данных других СУБД (рис. 3.5) реализуется через технику *драйверов ODBC*, которые *инсталлируются и выполняются на тех вычислительных установках, где находятся удаленные данные.* «Идеология» в данном случае такова. В составе настольной СУБД, поддерживающей

локальную базу данных, можно инсталлировать дополнительный программный компонент, называемый драйвером ODBC. Инсталлируемый драйвер ODBC «регистрируется» в специальном подкаталоге системного каталога операционной системы. Так образуется *рабочая область прямого доступа к источникам данных ODBC.*

Для непосредственного доступа к источникам данных ODBC ядро СУБД по системному каталогу внутренней локальной базы данных определяет *местонахождение источника,* по *протоколу взаимодействия приложений* (API) осуществляет *вызов (запуск)* на вычислительной установке удаленных данных драйвера ODBC и направляет ему по протоколу ODBC *SQL-инструкции* на доступ и обработку данных. При этом *режим* такого доступа регулируется рядом *параметров* (интервал вызова процедур, максимальное время обработки запроса, количество однократно пересылаемых по сети записей из набора данных, формируемых по запросам, время блокировок записей и т. д.). Данные параметры записываются в специальный реестр операционной системы при инсталляции и регистрации соответствующего драйвера ODBC.

При таком подходе каждая локальная СУБД на своей вычислительной установке выполняет роль SQL-сервера, т. е. машины данных, в случае обращения на доступ извне (из других вычислительных установок) к данным из «ее» файлов данных. Так как непосредственную обработку данных в данном случае выполняет «родная» СУБД, знающая все особенности логической и физической структуры «своих» файлов данных, то обеспечивается, как правило, более эффективная обработка, а самое главное, проверяются и выполняются ограничения целостности данных по логике предметной области источников данных.

Определенной проблемой технологий объектного связывания является появление «брешей» в системах защиты данных и разграничения доступа. Вызовы драйверов ODBC для осуществления процедур доступа к данным помимо пути, имени файлов и требуемых объектов (таблиц), если

соответствующие базы защищены, содержат в открытом виде пароли доступа, в результате чего может быть проанализирована и раскрыта система разграничения доступа и защиты данных.

Контрольные вопросы

1. Объясните суть технологии объектного связывания данных.
2. Принципы доступа к внешним данным на основе протокола ODBC.
3. Объясните схемы локальных баз данных со связанными объектами.
4. Какой подход реализует интерперабельность?

Глава 4. АУДИТ БЕЗОПАСНОСТИ И РЕЗЕРВНОЕ КОПИРОВАНИЕ БАЗЫ ДАННЫХ

4.1. Особенности проведения аудита безопасности в системах управления базами данных

Под аудитом информационной системы (ИС) или информационной технологии (ИТ) понимается системный процесс получения и оценки объективных данных о текущем состоянии системы, технологии, действиях и событиях, происходящих в ней, устанавливающий уровень их соответствия определенному критерию и предоставляющий результаты заказчику.

Проведение аудита позволяет оценить текущую безопасность функционирования ИТ, оценить риски и управлять ими, прогнозировать их влияние на бизнес-процессы организации, корректно и обоснованно подходить к вопросу обеспечения безопасности информационных активов организации, основными из которых являются:

- идеи;
- знания;
- проекты;
- результаты внутренних обследований.

Общее понятие аудита. Датой рождения аудита принято считать 1844г., когда в Англии приняли закон об акционерных компаниях, согласно которому их правления должны были ежегодно отчитываться перед аукционерами, причем отчет должен был быть проверен и подтвержден специальным человеком - независимым аудитором.

В настоящее время аудит, пройдя несколько этапов своего развития стал частью хозяйственной жизни стран. От проверки бухгалтерских счетов акционерных компании отдельными профессиональными аудиторами аудит развивался до комплексного понятия, включающего в себя ряд услуг (проверку бухгалтерской отчетности, финансовый анализ, консультирование), оказываемых профессиональными аудиторами и

аудиторскими фирмами. Среди таких фирм есть и небольшие, включающие в себя десяток сотрудников, и гиганты с численностью до нескольких тысяч человек.

Виды аудита. Аудит безопасности информационных систем обычно подразделяют на внешний и внутренний.

Внешний аудит проводится в основном вне организации и, как правило, специализированными организациями, занимающимися аудитом информационной безопасности. В этом случае, анализируются меры риска от внешних атак и атак со стороны (даже если организация защищена межсетевыми экранами). При проведении внешнего аудита осуществляется сканирование портов, поиск уязвимостей в сетевом и прикладном программном обеспечении,

Осуществляются попытки взаимодействия с Web-серверами, почтовым и файловыми серверами, попытки вхождения в локальные сети организации. По желанию руководства организации, может проводиться специальный вид внешнего аудита - Ethical Hacking. В этом случае специальная организация (в мире это является широко распространенной практикой, такие подразделении имеют специальное название Tiger Team) осуществляет избранные атаки на серверы, сайты и хосты организации. Такие атаки могут продемонстрировать уязвимости ИС организации.

Внутренний аудит, как правило, проводился специальной командой из числа персонала организации. Его задачей является оценка риска существующей технологии применения ИС. Этот вид аудита выполняется с привлечением средств автоматизации аудита, реализующих какой-либо стандарт. Внутренний аудит проводится внутри сетевого пространства, ограниченного межсетевым экраном организации. Он также включает в себя сканирование портов и уязвимостей внутренних хостов организации. Кроме того, анализируются организация и выполнение установленной политики безопасности, контроль и управление доступом к ресурсам, парольная политика персонала организации и ее выполнение, Данный вид аудита

дополняет стандартные методики проведения аудита более исчерпывающим рассмотрением сетевых уязвимостей.

Проведения аудита безопасности баз данных на примере СУБД Oracle. СУБД Oracle - функционально развитый продукт, и в нем существует несколько возможностей проведения аудита.

Аудит Oracle может помочь в определении неавторизованного доступа или внутреннего злоупотребления по отношению к информации содержащейся в базе данных.

Аудит в Oracle разделен на три части:

- аудит таких выражений как CREATE TABLE или CREATE SESSION,
- аудит привилегий ALTER USER и
- аудит на объект на объектном уровне SELECT TABLE.

Основная конфигурация. Записи аудита могут помещаться либо в аудиторскую таблицу базы данных, либо в аудиторский журнал операционной системы. Запись аудита в журнал операционной системы в некоторых случаях более защищена, но эта возможность доступна не для всех платформ. Аудит включается для записи в базу данных добавлением следующей строки в файле init.ora.

audit_trail = db

Примеры. Пример включения аудита для попыток доступа к базе данных:

SQL> audit create session;

Audit succeeded.

SQL>

Приведенная команда будет отслеживать доступ всех пользователей, независимо от того успешен он или нет.

Формат всех команд аудита по документации Oracle выглядит следующим образом:

audit {statement_option/privilege_option}

[by user] [by

{session/access}] [whenever {successful/unsuccessful}]

Обязательными являются только лишь statement_option и privilege_option части выражения. Другие части являются опционными и их использование позволяет сделать аудит более специфичным.

Чтобы пользователь мог задать команду аудита, необходимым условием для него является наличие привилегии «AUDIT SYSTEM». Найти пользователей, которые имеют эту привилегию, можно выполнив следующее:

```
SQL> select *
  2  from dba_sys_privs
  3  where privilege like '%AUDIT%';
GRANTEE                        PRIVILEGE                      ADM
------------------------       ----------------------
CTXSYS                         AUDIT ANY                      NO
CTXSYS                         AUDIT SYSTEM                   NO
DBA                            AUDIT ANY                      YES
DB                             AUDIT SYSTEM                   YES
IMP_FULL_DATABASE              AUDIT ANY                      NO
MDSYS                          AUDIT ANY                      YES
MDSYS                          AUDIT SYSTEM                   YES
WKSYS                          AUDIT ANY                      NO
WKSYS                          AUDIT SYSTEM                   NO

9 rows selected.
SQL>
```

Выше приведенные результаты принадлежат базе данных Oracle 9i. Пользователи по умолчанию MDSYS, CTXSYS и WKSYS были бы неплохой мишенью для атакующего, так как любые действия аудита могут быть

выключены любым из этих пользователей, что бы скрыть любые предпринятые действия.

Теперь аудит будет отслеживать все попытки доступа, необходимо подождать, когда какие-нибудь пользователи войдут в систему что бы выполнить свою работу.

Пример включения аудита для контроля изменений в базе данных.

Для краткости, не все изменения объектов схемы будем отслеживать, в этом примере. Хотя в принципе, можно отслеживать изменения любых объектов БД: таблиц, индексов, кластеров, представлений, последовательностей, процедур, триггеров, библиотек и т.д. В этом примере аудит будет включен на выборочной группе объектов. Настройка аудита может быть выполнена за два этапа, создание команд аудита и запуск на исполнение, как показано ниже:

```
set head off
set feed off
set pages 0
spool aud.lis
select 'audit '//name//';'
from system_privilege_map
where (name like 'CREATE%TABLE%'
or name like 'CREATE%INDEX%'
or name like 'CREATE%CLUSTER%'
or name like 'CREATE%SEQUENCE%'
or name like 'CREATE%PROCEDURE%'
or name like 'CREATE%TRIGGER%'
or name like 'CREATE%LIBRARY%')
union
select 'audit '//name//';'
from system_privilege_map
where (name like 'ALTER%TABLE%'
```

```
or name like 'ALTER%INDEX%'
or name like 'ALTER%CLUSTER%'
or name like 'ALTER%SEQUENCE%'
or name like 'ALTER%PROCEDURE%'
or name like 'ALTER%TRIGGER%'
or name like 'ALTER%LIBRARY%')
union
select 'audit '//name//';'
from system_privilege_map
where (name like 'DROP%TABLE%'
or name like 'DROP%INDEX%'
or name like 'DROP%CLUSTER%'
or name like 'DROP%SEQUENCE%'
or name like 'DROP%PROCEDURE%'
or name like 'DROP%TRIGGER%'
or name like 'DROP%LIBRARY%')
union
select 'audit '//name//';'
from system_privilege_map
where (name like 'EXECUTE%INDEX%'
or name like 'EXECUTE%PROCEDURE%'
or name like 'EXECUTE%LIBRARY%')
/
spool off
@@aud.lis
```

Данный скрипт выведет набор команд аудита в спул файл, который затем запустится для выполнения команд аудита.

После этого базу данных необходимо перезапустить. Простая проверка покажет, что аудит действительно включен.

```
SQL> select name,value from v$parameter
```

```
  2  where name like 'audit%';

NAME                         VALUE
---------------------------- ----------
audit_trail                  DB
audit_file_dest              ?/rdbms/audit

SQL>
```

Но контролируемые действия не отслеживаются до тех пор, пока эти действия не заданы явно; это верно, кроме случаев привилегированного доступа к базе данных, запуска и останова базы данных, структурных изменений, таких как добавление файла данных. Эти действия отслеживаются в файле операционной системы в $ORACLE_HOME/rdbms/audit до тех пор пока audit_file_dest не переопределено в файле init.ora . В Windows эти события появляются в Event Viewer.

Для того, что бы проверить наличие того, что какие-нибудь привилегии или выражения уже используются для аудита следующее:

```
SQL> select * from dba_stmt_audit_opts
  2  union
  3  select * from dba_priv_audit_opts;
no rows selected
SQL>
```

Что бы найти какие объекты уже контролируются аудитом, необходимо запросить представление dba_obj_audit_opts.

Контрольные вопросы:

1. Что понимается под аудитом информационной системы?
2. Какие активы организации рассматривается при проведении аудита информационной безопасности?
3. Перечислите виды аудита информационной безопасности.

4. На какие части разделен аудит в *Oracle*?

4.2.Восстановление базы данных

Для того чтобы избежать потери важной информации, необходимо проводить регулярное резервное копирование или бэкап (от англ. backup – защита, поддержка) данных. *Резервное копирование* – это периодическое дублирование или создание запасных копий критически важных данных с целью их восстановления в случае потери оригинала. Можно сказать, что резервное копирование – это страхование от потери информации в случае поломки оборудования или случайного удаления файлов пользователем. Существует два основных метода резервного копирования – это копирование файловой системы компьютера и копирование образа жесткого диска.

Копирование образа жесткого диска – это создание точной копии всего жесткого диска, что позволяет восстанавливать не только данные пользователя, но Windows и всю информацию о состоянии операционной системы, такую как данные системного реестра, драйверы, профили пользователей, системные настройки, программы и приложения.

Файловое копирование – это копирование файловой системы компьютера, то есть папок и файлов, хранящихся на компьютере. Такое копирование поможет восстановить папки и файлы пользователя, но не сможет вернуть систему в рабочее состояние. Что касается конкретных способов реализации этих двух типов копирования, то тут также можно выделить несколько основных видов: полное копирование, дифференциальное копирование и инкрементальное копирование. Полное копирование – это копирование всех указанных данных целиком и полностью, будь то образ диска или файловая система, без учета изменений, произошедших в промежутках между копированиями.

Под дифференциальным резервным копированием понимается копирование изменившейся информации со времени последнего полного

бэкапа. То есть каждое последующее копирование включает в себя все файлы, которые изменились со времени первого бэкапа. Таким образом, чтобы сделать восстановление резервной копии нужно взять первый полный и последний бэкап. Инкрементальный бэкап копирует только новые и изменившиеся файлы со времени последнего копирования, а не первого. Поэтому он занимает меньше места на носителе, чем дифференциальный. Но инкрементальный бэкап сложнее восстанавливать, так как приходится учитывать не только первый и последний бэкап-файлы, но и все промежуточные. Существует еще один способ копирования: «зеркальное копирование». Этот способ предполагает, что как только на диске появляется новый файл, он тут же появляется и в копии (в режиме реального времени). Некоторые специалисты называют это способ копирования двухсторонней синхронизацией.

Требования к системе резервного копирования:

- надёжность хранения информации. Обеспечивается применением отказоустойчивого оборудования систем хранения, дублированием информации и заменой утерянной копии другой в случае уничтожения одной из копий;

- простота в эксплуатации - автоматизация (по возможности минимизировать участие человека: как пользователя, так и администратора);

- быстрое внедрение (простая установка и настройка программ, быстрое обучение пользователей).

По способу передачи резервной копии на устройство хранения можно выделить следующие методы резервного копирования данных: через локальную вычислительную сеть, через сеть хранения данных без участия сервера резервного копирования, через сеть хранения данных с использованием механизма «мгновенных копий». Преимущества «классического» способа резервного копирования через КС на базе TCP/IP – простота реализации и внесения изменений в инфраструктуру резервного копирования: достаточно установить программное обеспечение сетевого

агента резервного копирования на сервере-клиенте и настроить взаимодействие по сети с сервером резервного копирования. Резервное копирование происходит по следующей схеме: сервер резервного копирования отправляет по КС команду агенту резервного копирования на сервере-клиенте, который выполняет операции по подготовке и отправке данных на сервер резервного копирования. Сервер резервного копирования принимает и записывает данные на устройство хранения. При каких-либо изменениях в сетевой инфраструктуре сетевой агент может быть достаточно быстро перенастроен, и система резервного копирования продолжит работу. Кроме того, технология дешева в реализации и не требует прямого подключения серверов-клиентов к сети хранения данных, т. е. может быть использована для защиты данных серверов, не подключенных к SAN (Storage Area Network – сеть хранения данных), или серверов с виртуализированным подключением к сети хранения данных. К недостаткам метода относятся загрузка КС трафиком резервного копирования, нагрузка на сервер-клиент, зависимость от пропускной способности КС и интерфейсов сервера резервного копирования, необходимость выполнять резервное копирование в течение заданного «временного окна», ограниченные возможности по масштабированию. Конечно, можно организовать выделенную КС для передачи трафика резервного копирования, установить дополнительные управляемые серверы резервного копирования, агрегировать сетевые интерфейсы сервера резервного копирования в единый «толстый» интерфейс, однако эти меры ведут к усложнению инфраструктуры защиты данных и ухудшению ее масштабируемости. Тем не менее, указанная технология вполне подходит для резервного копирования редко изменяемых данных, таких как системные данные ОС и приложений.

В базе данных можно создавать файлы данных двух типов.

Первичный файл данных (primary data file) обязательным. В нём хранится загрузочная информация каталога базы данных и указатели на другие файлы базы данных. Первичный файл данных может также содержать

объекты и пользовательские данные. Для имени первичного файла рекомендуется расширение mdf.

Вторичный файлы данных (Secondary data file) не являются обязательными и определяются пользователем. В них содержатся объекты и пользовательские данные. Для повышения производительности вторичные файлы рекомендуется размещать на разных дисках. В базе данных может быть не более 32 766 вторичных файлов данных. Для имени вторичного файла данных рекомендуется расширение ndf.

Модель восстановления (recovery model) – это параметр конфигурации базы данных, который управляет регистрацией транзакций, созданием резервных копий журнала транзакций и параметрами восстановления базы данных. Выбор модели восстановления оказывает существенное влияния как на восстановление данных, так и на производительность в зависимость от того, выполняет модель восстановления регистрацию транзакций или нет.

Модель полного восстановления означает, что ядро базы данных регистрирует в журнале транзакций все операции и никогда не выполняет усечение журнала. Это модель позволяет восстановить базу данных до ее состояния на момент сбоя.

Простая модель восстановления регистрирует минимум данных о большинстве транзакций и выполняет усечение журнала транзакций после каждой контрольной точки. Это модель восстановления не поддерживает резервное копирование и восстановление журнала транзакций. Более того, не позволяет восстанавливать отдельные страницы данных.

Модель с неполным протоколированием означает, что ядро базы данных ведет минимальную регистрацию массовых операций, таких как SELECT INTO и BULKINSERT. Если в резервной копии журнала содержатся какие-либо массовые операции, базу данных можно восстановить до состояния, соответствующего концу резервной копии журнала транзакций, а не до определенного момента времени. Это модель восстановления используются только для больших массовых операций.

Основные вида резервного копирования. Выделяют два основных вида резервного копирования:

1. Непротиворечивое (холодное) резервное копирование, когда копии создаются, в случае закрытой для пользователей БД (close). Копия базы данных, созданной в автономном режиме, содержит: все файлы данных, журналы повторов и управляющие файлы. После остановки БД, все файлы базы копируются на один из backup дисков. По окончании копирования осуществляется перезагрузка базы данных.

2. Резервное (горячее) копирование в оперативном режиме, к примеру, когда БД всё время находится в оперативном режиме и доступна пользователям.

Резервные копии журнала транзакций можно создавать только для баз данных, в которых установлена модель полного восстановления или модель восстановления с неполным протоколированием. Также резервное копирование журнала транзакций возможно только после выполнения резервного копирования. Резервная копия журнала транзакций содержит только часть данных, поэтому для восстановления базы данных требуется также ее полная копия.

Из двух видов резервирования наибольший выигрыш надежности достигается при резервировании замещением. Однако для реализации этого вида резервирования требуется автомат контроля состояния системы и коммутации при отказе работающей системы. База данных может быть «разнесена» на множество файловых групп, каждая из которых может включать множество дополнительных файлов данных и журналов транзакций (рис. 4.1). Наибольший эффект от разнесения базы данных на файловые группы достигается применением RAID массивов. Применение RAID массивов повышает производительность файловой подсистемы, уменьшается время отклика системы, ее доступность и надёжность.

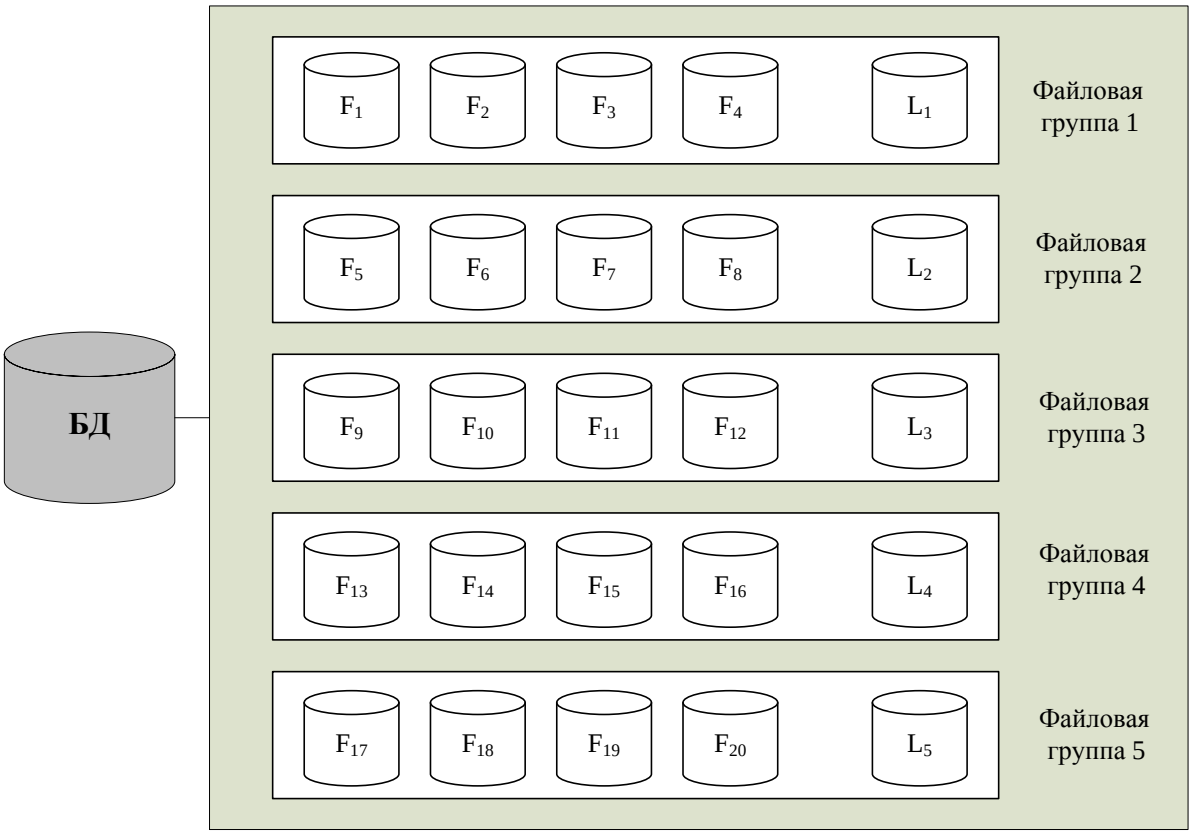


Рис. 4.1. Разбиения файла базы данных на файловые группы.

При наличии автомата контроля и коммутации структурная схема резервированной системы с кратностью m=1 будет выглядеть так, как показано на рис.4.2:

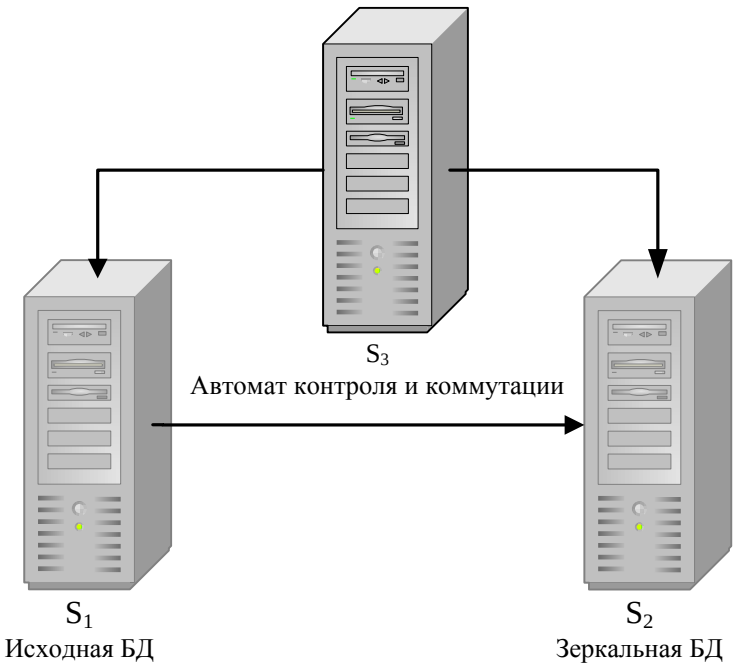


Рис.4.2. Структурная схема модуля резервного копирования с автоматом контроля и коммутации

На рис.4.2 приняты следующие обозначения:

S1, S2 – основная и резервная системы защиты информации;

S3 – автомат контроля правильности функционирования системы защиты и коммутации при обнаружении отказа основной системы;

Контрольные вопросы

1. Что такое резервное копирование базы данных?
2. Основные методы резервного копирования.
3. Требования к системе резервного копирования.
4. Какие файлы создаются в базе данных?
5. Основные виды резервного копирования базы данных.
6. Структурная схема модуля резервного копирования с автоматом контроля и коммутации.

4.3. Процесс синхронизаций репликации в современных системах управления базами данных

Во многих случаях узким местом распределенных систем, построенных на основе технологий «Клиент-сервер» или объектного связывания данных, является недостаточно высокая производительность из-за необходимости передачи по сети большого количества данных. Определенную альтернативу построения быстродействующих распределенных систем предоставляют технологии реплицирования данных.

Репликой называют *особую копию базы данных для размещения на другом компьютере сети с целью автономной работы пользователей с одинаковыми (согласованными) данными общего пользования.*

Основная идея реплицирования заключается в том, что пользователи работают автономно с одинаковыми (общими) данными, растиражированными по локальным базам данных, обеспечивая с учетом отсутствия необходимости передачи и обмена данными по сети

максимальную для своих вычислительных установок производительность. Программное обеспечение СУБД для реализации такого подхода соответственно дополняется функциями тиражирования (реплицирования) баз данных, включая тиражирование как самих данных и их структуры, так и системного каталога с информацией о размещении реплик.

При этом, однако, возникают две проблемы обеспечения одного из основополагающих принципов построения и функционирования распределенных систем, а именно — *непрерывности согласованного состояния данных*:

- обеспечение согласованного состояния во всех репликах количества и значений общих данных;

- обеспечение согласованного состояния во всех репликах структуры данных.

Обеспечение согласованного состояния общих данных, в свою очередь, основывается на реализации одного из *двух принципов*:

- принципа непрерывного размножения обновлений (любое обновление данных в любой реплике должно быть немедленно размножено);

- принципа отложенных обновлений (обновления реплик могут быть отложены до специальной команды или ситуации).

Принцип непрерывного размножения обновлений является основополагающим при построении так называемых «*систем реального времени»,* таких, например, как системы управления воздушным движением, системы бронирования билетов пассажирского транспорта и т. п., где требуется непрерывное и точное соответствие реплик или других растиражированных данных во всех узлах и компонентах подобных распределенных систем.

Реализация принципа *непрерывного размножения обновлений* заключается в том, что *любая транзакция считается успешно завершенной, если она успешно завершена на всех репликах системы.* На практике реализация этого принципа встречает существенные затруднения, связанные

с *тупиками*. Предположим, что на одной вычислительной установке пользователь обновляет данные в своей реплике. На время осуществления транзакции (транзакций) соответствующие записи в базе данных этой реплики ядром локальной СУБД заблокированы от изменения другими пользователями. Вместе с тем транзакция может быть зафиксирована и, следовательно, разблокированы соответствующие данные только тогда, когда данная транзакция послана и также завершена на других репликах системы. Предположим также, что в другой реплике системы, находящейся на другом компьютере сети, в это же время другой пользователь проводит свои обновления (транзакции) с теми же записями, которые, естественно, в этот момент также заблокированы от изменений для других пользователей. Так образуется тупик. Одна транзакция не может быть зафиксирована в своей реплике, потому что заблокированы соответствующие записи в другой реплике. А разблокировка этих записей в другой реплике также невозможна до тех пор, пока не разблокируются соответствующие записи в первой реплике, т. е. когда завершится транзакция в первой реплике. Создается тупиковая ситуация.

Для обнаружения тупиков в реплицированных системах применяются такие же алгоритмы, которые были разработаны в мониторах транзакций централизованных систем «Клиент-сервер».

В целом ряде предметных областей распределенных информационных систем режим реального времени с точки зрения непрерывности согласования данных не требуется. Такие системы автоматизируют те организационно-технологические структуры, в которых информационные процессы не столь динамичны. Если взять, к примеру, автоматизированную информационную систему документооборота, то традиционная «скорость» перемещения и движения служебных документов соответствует рабочему дню или в лучшем случае рабочим часам. В этом случае обновление реплик распределенной информационной системы, если она будет построена на

технологии реплицирования, требуется, скажем, только лишь один раз за каждый рабочий час, или за каждый рабочий день.

Такого рода информационные системы можно строить на основе *принципа отложенных обновлений.* Накопленные в какой-либо реплике изменения данных *специальной командой* пользователя направляются для обновления всех остальных реплик систем. Такая операция называется синхронизацией реплик. Возможность конфликтов и тупиков в этом случае при синхронизации реплик существенно снижается, а немногочисленные подобные конфликтные ситуации легко разрешить организационными мерами.

Решение второй проблемы согласованности данных, а именно — согласованности структуры данных, осуществляется через частичное отступление, как и в системах «Клиент-сервер», от принципа отсутствия центральной установки и основывается на технике «главной» реплики.

Суть этой техники заключается в том, что одна из реплик базы данных системы объявляется главной. При этом изменять структуру базы данных можно только в главной реплике. Эти изменения структуры данных тиражируются на основе принципа отложенных обновлений, т. е. через специальную синхронизацию реплик. Частичность отступления от принципа отсутствия центральной установки заключается в том, что в отличие от чисто централизованных систем, выход из строя главной реплики не влечет сразу гибель всей распределенной системы, так как остальные реплики продолжают функционировать автономно. Более того, на практике СУБД, поддерживающие технологию реплицирования, позволяют пользователю с определенными полномочиями (администратору системы) преобразовать любую реплику в главную и тем самым полностью восстановить работоспособность всей системы.

Процесс синхронизации реплик в современных СУБД. Этот процесс включает обмен только теми данными, которые были изменены или добавлены в разных репликах. С этой целью в системном каталоге базы

данных создаются специальные таблицы текущих изменений и организуется система глобальной идентификации (именования) всех объектов распределенной системы, включая раздельное по именование одинаковых объектов в разных репликах. Такой подход несколько увеличивает объем базы данных, но позволяет существенно ограничить транспортные расходы на синхронизацию реплик.

Важным, с точки зрения гибкости и эффективности функционирования распределенных информационных систем, построенных на технологиях реплицирования, является возможность создания так называемых частичных реплик и включения в реплики как *реплицируемых,* так и *нереплицируемых объектов.* Частичной репликой называется *база данных, содержащая ограниченное подмножество записей полной реплики.* Распространенным способом создания частичных реплик является использование фильтров, устанавливаемых для конкретных таблиц полной (главной) реплики. Частичные реплики позволяют решить некоторые проблемы, связанные с разграничением доступа к данным и повышают производительность обработки данных. Так, к примеру, в реплику базы данных для определенного подразделения целесообразно реплицировать только те записи таблицы, которые относятся к данному подразделению, исключив тем самым доступ к другим записям. Техника частичных реплик также снижает затраты на синхронизацию реплик, так как ограничивает количество передаваемых по сети изменений данных.

Возможность включения в реплики объектов базы данных, которые не подлежат репликации, позволяет более гибко и адекватно настроить схему и прочие объекты БД (запросы, формы и отчеты) на специфику предметной области, особенности ввода данных и решаемые информационные задачи по конкретному элементу распределенной системы.

На рис. 4.3 иллюстрируется подход к организации общей схемы распределенной информационной системы по делопроизводству некоторой организационной структуры на основе технологий репликации данных.

Технологии репликации данных в тех случаях, когда не требуется обеспечивать большие потоки и интенсивность обновляемых в информационной сети данных, являются экономичным решением проблемы создания распределенных информационных систем с элементами централизации по сравнению с использованием дорогостоящих «тяжелых» клиент-серверных систем.

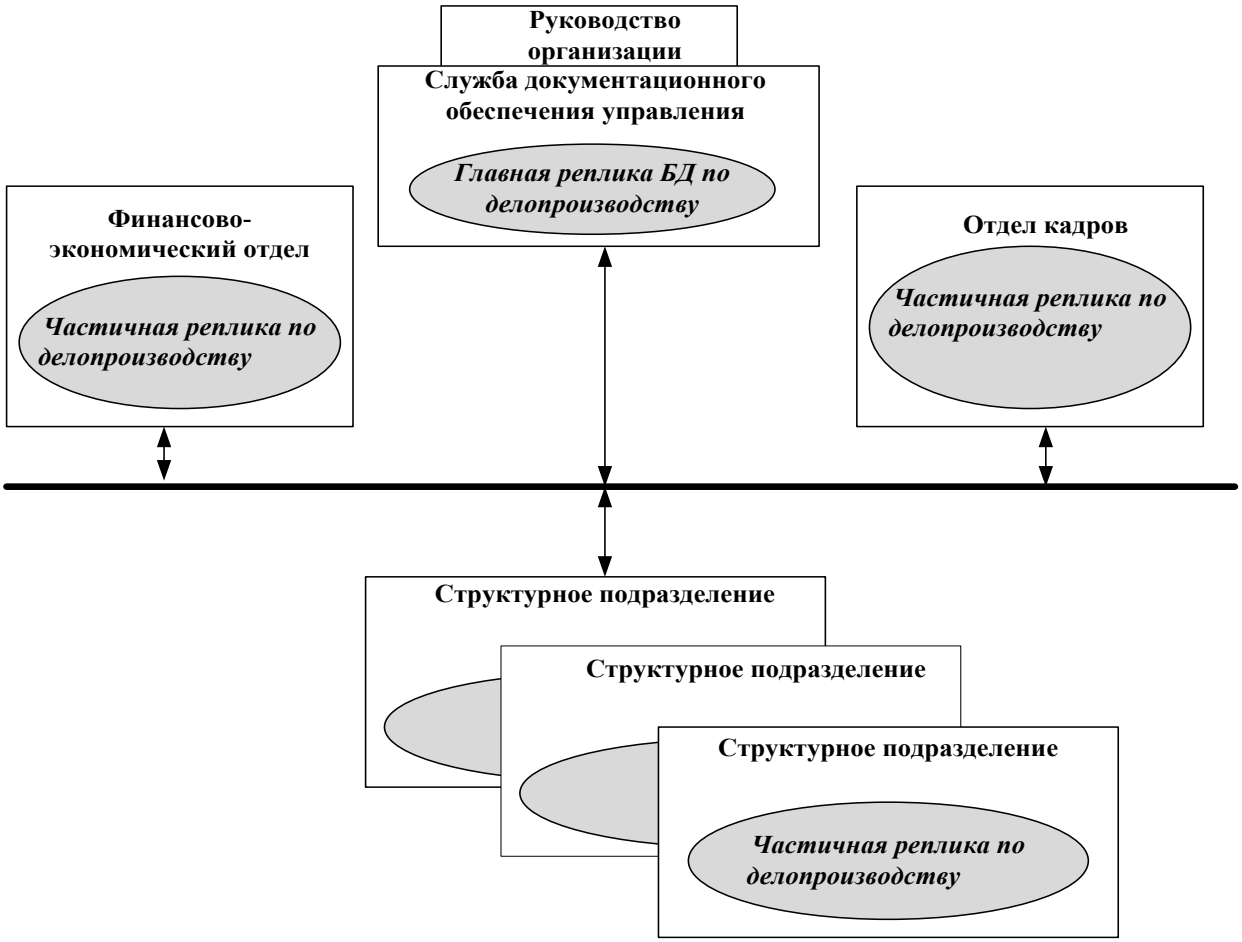


Рис. 4.3. Пример подхода к организации схемы распределенной информационной системы по делопроизводству на основе технологии реплицирования.

На практике для совместной коллективной обработки данных применяются смешанные технологии, включающие элементы объектного связывания данных, репликации и клиент-серверных решений. При этом дополнительно к проблеме логического проектирования, т. е. проектирования логической схемы организации данных (таблицы, поля, ключи, связи, ограничения целостности), добавляется не менее сложная проблема транспортно-технологического проектирования информационных

потоков, разграничения доступа и т.д. К сожалению, пока не проработаны теоретико-методологические и инструментальные подходы для автоматизации проектирования распределенных информационных систем с учетом факторов как логики, так и информационно-технологической инфраструктуры предметной области. Тем не менее развитие и все более широкое распространение распределенных информационных систем, определяемое самой распределенной природой информационных потоков и технологий, является основной перспективой развития автоматизированных информационных систем.

Контрольные вопросы

1. Объясните процесс репликации данных.
2. Объясните процесс создания частичных реплик.
3. Объясните случаи возникновения «тупиковых» ситуаций.
4. Объясните процесс синхронизации реплик.

Глава 5. СТАНДАРТЫ И СПЕЦИФИКАЦИИ ПО ОБЕСПЕЧЕНИЮ БЕЗОПАСНОСТИ БАЗЫ ДАННЫХ

5.1. Архитектура и принцип функционирования подсистемы безопасности базы данных

Системы управления базами данных, в особенности реляционные СУБД, стали доминирующим инструментом хранения больших массивов информации. Сколько-нибудь развитые информационные приложения полагаются не на файловые структуры операционных систем, а на многопользовательские СУБД, выполненные в технологии клиент/сервер. В связи с этим, обеспечение информационной безопасности СУБД, и в первую очередь их серверных компонентов, приобретает решающее значение для безопасности организации в целом. Как мы уже отметили, для СУБД важны все три основных аспекта информационной безопасности - конфиденциальность, целостность и доступность. Общая идея защиты баз данных состоит в следовании рекомендациям, сформулированным для класса безопасности С2 в «Критериях оценки надежных компьютерных систем». В принципе некоторые СУБД предлагают дополнения, характерные для класса В1, однако практическое применение подобных дополнений имеет смысл, только если все компоненты информационной структуры организации соответствуют категории безопасности В. Достичь этого непросто и с технической, и с финансовой точек зрения. Следует, кроме того, учитывать два обстоятельства. Во-первых, для подавляющего большинства коммерческих организаций класс безопасности С2 достаточен. Во-вторых, более защищенные версии отстают по содержательным возможностям от обычных «собратьев», так что поборники секретности по сути обречены на использование морально устаревших (хотя и тщательно проверенных) продуктов со всеми вытекающими последствиями в плане сопровождения.

Идентификация и проверка подлинности пользователей.

Обычно в СУБД для идентификации и проверки подлинности пользователей применяются либо соответствующие механизмы операционной системы, либо SQL-оператор CONNECT. Например, в случае СУБД Oracle оператор CONNECT имеет следующий вид:

CONNECT пользователь[/пароль] [@база_данных].

Так или иначе, в момент начала сеанса работы с сервером баз данных, пользователь идентифицируется своим именем, а средством аутентификации служит пароль. Детали этого процесса определяются реализацией клиентской части приложения.

Некоторые операционные системы, такие как UNIX, позволяют во время запуска программы менять действующий идентификатор пользователя. Приложение, работающее с базой данных, как правило, имеет привилегии, значительно превосходящие привилегии обычных пользователей. Естественно, что при этом приложение предоставляет тщательно продуманный, строго фиксированный набор возможностей. Если пользователь сумеет тем или иным способом завершить приложение, но сохранить подключение к серверу баз данных, ему станут доступны по существу любые действия с данными.

Управление доступом. Для иллюстрации вопросов, связанных с управлением доступом, будет использоваться СУБД INGRES.

Обычно в СУБД применяется произвольное управление доступом, когда владелец объекта передает права доступа к нему (чаще говорят - привилегии) по своему усмотрению. Привилегии могут передаваться субъектам (отдельным пользователям), группам, ролям или всем пользователям.

Привилегии роли имеют приоритет над привилегиями пользователей и групп. Иными словами, пользователю как субъекту не обязательно иметь права доступа к объектам, обрабатываемым приложениям с определенной ролью. Отметим, что в СУБД Oracle под ролью понимается набор

привилегий. Такие роли служат средством структуризации привилегий и облегчают их модификацию.

Совокупность всех пользователей именуется как PUBLIC. Придание привилегий PUBLIC - удобный способ задать подразумеваемые права доступа. Поручать администрирование различных баз данных разным людям имеет смысл только тогда, когда эти базы независимы и по отношению к ним не придется проводить согласованную политику выделения привилегий или резервного копирования. В таком случае каждый из администраторов будет знать ровно столько, сколько необходимо.

Можно провести аналогию между пользователем INGRES и администраторами баз данных с одной стороны, и суперпользователем операционной системы (root в случае ОС UNIX) и служебными пользователями (в ОС UNIX это могут быть bin, lp, uucp и т.д.) с другой стороны. Введение служебных пользователей позволяет администрировать функциональные подсистемы, не получая привилегий суперпользователя. Точно так же информацию, хранящуюся на сервере баз данных, можно разделить на отсеки, так что компрометация администратора одного отсека не означает обязательной компрометации другого.

Виды привилегий. Привилегии в СУБД можно подразделить на две категории: привилегии безопасности и привилегии доступа.

Привилегии безопасности всегда выделяются конкретному пользователю во время его создания (оператором CREATE USER) или изменения характеристик (оператором ALTER USER). Таких привилегий пять:

- security - право управлять безопасностью СУБД и отслеживать действия пользователей. Пользователь с этой привилегией может подключаться к любой базе данных, создавать, удалять и изменять характеристики пользователей, групп и ролей, передавать права на доступ к базам данным другим пользователям, управлять записью регистрационной информации, отслеживать запросы других пользователей и, наконец,

запускать INGRES-команды от имени других пользователей. Привилегия security необходима администратору сервера баз данных, а также лицу, персонально отвечающему за информационную безопасность. Передача этой привилегии другим пользователям (например, администраторам баз данных) увеличивает число потенциально слабых мест в защите сервера баз данных.

- createdb - право на создание и удаление баз данных. Этой привилегией, помимо администратора сервера, должны обладать пользователи, которым отводится роль администраторов отдельных баз данных.

- operator - право на выполнение действий, которые традиционно относят к компетенции оператора. Имеются в виду запуск и остановка сервера, сохранение и восстановление информации. Помимо администраторов сервера и баз данных этой привилегией целесообразно наделить также администратора операционной системы.

- maintain_locations - право на управление расположением баз администратора сервера баз данных и операционной системы.

- trace - право на изменение состояния флагов отладочной трассировки. Данная привилегия полезна администратору сервера баз данных и другим знающим пользователям при анализе сложных, непонятных ситуаций.

Привилегии безопасности позволяют выполнять административные действия.

Привилегии доступа, в соответствии с названием, определяют права доступа субъектов к определенным объектам. Привилегии доступа выделяются пользователям, группам, ролям или всем посредством оператора GRANT и изымаются с помощью оператора REVOKE. Эти привилегии, как правило, присваивает владелец соответствующих объектов (он же - администратор базы данных) или обладатель привилегии security (обычно администратор сервера баз данных).

Прежде чем присваивать привилегии группам и ролям, их (группы и роли) необходимо создать с помощью операторов CREATE GROUP и CREATE ROLE.

Для изменения состава группы служит оператор ALTER GROUP.

Оператор DROP GROUP позволяет удалять группы, правда, только после того, как опустошен список членов группы.

Оператор ALTER ROLE служит для изменения паролей ролей, а DROP ROLE - для удаления ролей.

Напомним, что создавать и удалять именованные носители привилегий, а также изменять их характеристики может лишь пользователь с привилегией security. При совершении подобных действий необходимо иметь подключение к базе данных iidbdb, в которой хранятся сведения о субъектах и их привилегиях.

Привилегии доступа можно подразделить в соответствии с видами объектов, к которым они относятся. В СУБД INGRES таких видов пять:

- таблицы и представления
- процедуры
- базы данных
- сервер баз данных
- события

Присваивание привилегий доступа производится с помощью оператора GRANT. В самом общем виде оператор GRANT имеет следующий формат:

GRANT привилегии

ON объекты

TO кому;

Применительно к таблицам и представлениям можно управлять следующими правами доступа:

SELECT - право на выборку данных;

INSERT - право на добавление данных;

DELETE - право на удаление данных;

UPDATE - право на обновление данных (можно указать определенные столбцы, разрешенные для обновления)

REFERENCES- право на использование внешних ключей, ссылающихся на данную таблицу (можно указать определенные столбцы)

По умолчанию пользователь не имеет никаких прав доступа к таблицам и представлениям - их необходимо передать с помощью операторов GRANT.

По отношению к процедуре можно предоставить право на выполнение. При этом не нужно заботиться о выделении прав доступа к объектам, обрабатываемым процедурой - их наличие не обязательно. Таким образом, процедуры баз данных являются удобным средством предоставления контролируемого доступа для выполнения строго определенных действий над данными.

Права доступа к базе данных как к единому целому может предоставлять ее администратор или пользователь с привилегией security. Эти «права» на самом деле устанавливают ряд ограничений на использование базы данных, то есть по сути являются запретительными. Имеется в виду ограничение на число операций ввода/вывода или число строк, возвращаемых одним запросом, ограничение права создания таблиц и процедур и т.п. По умолчанию пользователь не стесняется количественными лимитами и получает право на создание объектов в базе.

Отметим, что при создании базы данных указывается ее статус - общая или личная. Это влияет на подразумеваемые права доступа к базе. По умолчанию право на подключение к общей базе предоставляется всем. Право на подключение к личной базе нужно передавать явным образом. Право на подключение необходимо для выполнения всех прочих операций с базой и содержащимися в ней объектами.

Привилегии (которые в данном случае точнее было бы назвать ограничениями) QUERY_IO_LIMIT и QUERY_ROW_LIMIT проверяются на основании оценок, выданных оптимизатором запросов. Если оптимизатор предсказывает, что запрос превысит отведенный лимит числа операций ввода

вывода или возвращаемых строк, он (запрос) отвергается. Наложение подобных количественных ограничений препятствует монополизации сервера одним клиентом и может использоваться как один из инструментов поддержания высокой готовности.

Для отмены привилегий, выданных ранее (как разрешительных, так и запретительных), служит оператор REVOKE.

Использование представлений для управления доступом. СУБД предоставляют специфическое средство управления доступом - представления. Представления позволяют сделать видимыми для субъектов определенные столбцы базовых таблиц (реализовать проекцию) или отобрать определенные строки (реализовать селекцию). Не предоставляя субъектам прав доступа к базовым таблицам и сконструировав подходящие представления, администратор базы данных защитит таблицы от несанкционированного доступа и снабдит каждого пользователя своим видением базы данных, когда недоступные объекты как бы не существуют.

Приведем пример создания представления, содержащего два столбца исходной таблицы и включающего в себя только строки с определенным значением одного из столбцов:

```
CREATE VIEW empview AS
SELECT name, dept
FROM employee
WHERE dept = 'shoe';
```

Предоставим всем право на выборку из этого представления:

```
GRANT SELECT
ON empview
TO PUBLIC;
```

Субъекты, осуществляющие доступ к представлению empview, могут пытаться запросить сведения об отделах, отличных от shoe, например:

```
SELECT *
FROM empview
```

```
WHERE dept = 'toy';
```

но в ответ просто получат результат из нуля строк, а не код ответа, свидетельствующий о нарушении прав доступа. Это принципиально важно, так как лишает злоумышленника возможности получить список отделов косвенным образом, анализируя коды ответов, возвращаемые после обработки SQL-запросов.

Иерархия прав доступа. Оператор GRANT и другие средства управления доступом СУБД позволяют реализовать следующие виды ограничения доступа:

- операционные ограничения (за счет прав доступа SELECT, INSERT, UPDATE, DELETE, применимых ко всем или только некоторым столбцам таблицы);
- ограничения по значениям (за счет механизма представлений);
- ограничения на ресурсы (за счет привилегий доступа к базам данных).

При обработке запроса СУБД сначала проверяет права доступа к объектам. Если операционные ограничения оказываются нарушенными, запрос отвергается с выдачей соответствующей диагностики. Нарушение ограничений на значения влияет только на количество результирующих строк; никакой диагностики при этом не выдается. Наконец, после учета двух предыдущих ограничений, запрос поступает на обработку оптимизатору. Если тот обнаружит превышение ограничений на ресурсы, запрос будет отвергнут с выдачей соответствующей диагностики.

На иерархию привилегий можно посмотреть и с другой точки зрения. Каждый пользователь, помимо, собственных, имеет привилегии PUBLIC. Кроме этого, он может входить в различные группы и запускать приложения с определенными ролями. Как соотносятся между собой права, предоставленные различным именованным носителям привилегий?

Иерархия авторизации выглядит для СУБД INGRES следующим образом:

- роль (высший приоритет)
- пользователь
- группа
- PUBLIC (низший приоритет)

Для каждого объекта, к которому осуществляется доступ, INGRES пытается отыскать в иерархии привилегию, относящуюся к запрашиваемому виду доступа (SELECT, EXECUTE и т.п.). Например, при попытке доступа к таблице с целью обновления, INGRES проверяет привилегии роли, пользователя, группы и всех пользователей. Если хотя бы на одном уровне иерархии привилегия UPDATE имеется, запрос передается для дальнейшей обработки. В противном случае используется подразумеваемое право доступа, которое предписывает отвергнуть запрос.

Рассмотрим подробнее трактовку ограничений на ресурсы. Пусть, например, на всех четырех уровнях иерархии специфицированы свои ограничения на число результирующих строк запроса (привилегия QUERY_ROW_LIMIT):

роль	1700
пользователь	1500
группа	2000
PUBLIC	1000

Если пользователь в момент начала сеанса работы с СУБД задал и роль, и группу, будет использовано ограничение, накладываемое ролью 1700. Если бы привилегия QUERY_ROW_LIMIT для роли отсутствовала, или пользователь не задал роль в начале сеанса работы, пользователь смог бы получать результаты не более чем из 1500 строк и т.п. Если бы привилегия QUERY_ROW_LIMIT вообще не была специфицирована ни на одном уровне иерархии, СУБД воспользовалась бы подразумеваемым значением, которое в данном случае означает отсутствие ограничений на число результирующих строк.

Обычно используемая роль и группа задаются, соответственно, как аргументы опций -R и -G в командной строке запуска приложения. Пример:

QBF -Gaccounting company_db

Если опция -G отсутствует, применяется подразумеваемая группа пользователя, если таковая имеется.

Наконец, если в командной строке sql задана опция

-u пользователь

то в число проверяемых входят также привилегии указанного пользователя.

Метки безопасности и принудительный контроль доступа. Выше были описаны средства произвольного управления доступом, характерные для уровня безопасности С. Как уже указывалось, они в принципе достаточны для подавляющего большинства коммерческих приложений. Тем не менее, они не решают одной весьма важной задачи - задачи слежения за передачей информации. Средства произвольного управления доступом не могут помешать авторизованному пользователю законным образом получить секретную информацию и затем сделать ее доступной для других, неавторизованных пользователей. Нетрудно понять, почему это так. При произвольном управлении доступом привилегии существуют отдельно от данных (в случае реляционных СУБД - отдельно от строк реляционных таблиц). В результате данные оказываются «обезличенными», и ничто не мешает передать их кому угодно даже средствами самой СУБД.

В «Критериях оценки надежных компьютерных систем», применительно к системам уровня безопасности В, описан механизм меток безопасности, реализованный в версии INGRES/Enhanced Security (INGRES с повышенной безопасностью). Применять эту версию на практике имеет смысл только в сочетании с операционной системой и другими программными компонентами того же уровня безопасности. Тем не менее, рассмотрение реализации меточной безопасности в СУБД INGRES интересно с познавательной точки зрения, а сам подход, основанный на разделении

данных по уровням секретности и категориям доступа, может оказаться полезным при проектировании системы привилегий многочисленных пользователей по отношению к большим массивам данных.

В СУБД INGRES/Enhanced Security к каждой реляционной таблице неявно добавляется столбец, содержащий метки безопасности строк таблицы. Метка безопасности состоит из трех компонентов:

- уровень секретности. Смысл этого компонента зависит от приложения. В частности, возможен традиционный спектр уровней от «совершенно секретно» до «несекретно»;
- категории. Понятие категории позволяет разделить данные на «отсеки» и тем самым повысить надежность системы безопасности. В коммерческих приложениях категориями могут служить «финансы», «кадры», «материальные ценности» и т.п.;
- области. Является дополнительным средством деления информации на отсеки. На практике компонент «область» может действительно иметь географический смысл, обозначая, например, страну, к которой относятся данные.

Каждый пользователь СУБД INGRES/Enhanced Security характеризуется степенью благонадежности, которая также определяется меткой безопасности, присвоенной данному пользователю. Пользователь может получить доступ к данным, если степень его благонадежности удовлетворяет требованиям соответствующей метки безопасности. Более точно:

- уровень секретности пользователя должен быть не ниже уровня секретности данных;
- набор категорий, заданных в метке безопасности данных, должен целиком содержаться в метке безопасности пользователя;
- набор областей, заданных в метке безопасности пользователя, должен целиком содержаться в метке безопасности данных.

Специальная привилегия DOWNGRADE, позволяет изменять метки безопасности, ассоциированные с данными. Подобная возможность необходима, например, для коррекции меток, по тем или иным причинам оказавшихся неправильными.

Представляется естественным, что СУБД INGRES/Enhanced Security допускает не только скрытое, но и явное включение меток безопасности в реляционные таблицы. Появился новый тип данных, security label, поддерживающий соответствующие операции сравнения.

INGRES/Enhanced Security - первая СУБД, получившая сертификат, эквивалентный аттестации на класс безопасности B1. Вероятно, метки безопасности постепенно войдут в стандартный репертуар систем управления базами данных.

Поддержание целостности данных в СУБД. Для коммерческих организаций обеспечение целостности данных по крайней мере не менее важно, чем обеспечение конфиденциальности. Конечно, неприятно, когда кто-то подглядывает за суммами на счетах клиентов, но гораздо хуже, когда в процессе перевода денег со счета на счет часть суммы исчезает в неизвестном направлении.

Известно, что главными врагами баз данных являются не внешние злоумышленники, а ошибки оборудования, администраторов, прикладных программ и пользователей.

С точки зрения пользователя СУБД, основными средствами поддержания целостности данных являются ограничения и правила.

Ограничения. Ограничения могут относиться к таблицам или отдельным столбцам. Ограничения на столбцы задаются при создании таблицы, в операторах CREATE TABLE

Табличные ограничения относятся к группе столбцов и могут задаваться как при создании таблицы, так и позже, посредством оператора ALTER TABLE.

Следующий пример содержит именованное ограничение, связывающее значения в двух столбцах:

```
CREATE TABLE dept (
dname char(10),
budget money,
expenses money,
CONSTRAINT check_amount CHECK (budget > 0 and expenses <= budget)
);
```

{Бюджет должен быть положительным, а расходы не должны выходить за рамки бюджета}

Ссылочные ограничения отвечают за целостность связей между таблицами. Подобное ограничение требует, чтобы каждому значению в столбце или группе столбцов одной таблицы соответствовало ровно одно значение в другой таблице. Название ограничения объясняется тем, что такие значения играют роль ссылок между таблицами в реляционной модели.

Приведем пример ссылочного ограничения:

```
CREATE TABLE emp (
ename char(10),
edept char(10) references dept(dname)
);
```

{Ни один работник не должен числиться в неизвестном отделе}

Ограничения всех видов накладываются владельцем таблицы и влияют на исход последующих операций с данными. Перед завершением выполнения SQL-оператора производится проверка имеющихся ограничений. При обнаружении нарушений СУБД сигнализирует о ненормальном завершении и аннулирует внесенные оператором изменения.

Отметим, что для наложения ссылочного ограничения необходимо обладать привилегией REFERENCES по отношению к таблице, на которую делается ссылка (dept в примере выше).

Ограничения можно не только накладывать, но и отменять. При этом между ограничениями могут существовать зависимости, и отмена одного из них может потребовать ликвидации других (ссылочных) ограничений, зависящих от первоначального. Рассмотрим следующий пример:

```
CREATE TABLE dept (
name char(10) NOT NULL,
location char(20),
CONSTRAINT dept_unique UNIQUE(name)
);
CREATE TABLE emp (
name char(10),
salary decimal(10,2),
edept char(10) CONSTRAINT empref REFERENCES dept(name)
);
```

Если требуется удалить ограничение dept_unique, можно воспользоваться следующим оператором:

```
ALTER TABLE dept
DROP CONSTRAINT dept_unique cascade;
```

Слово cascade означает, что следует удалить также все ограничения, прямо или косвенно зависящие от dept_unique. В данном случае будет изъято ограничение empref. Если вместо cascade указать restrict, то есть сделать попытку удалить только ограничение dept_unique, СУБД зафиксирует ошибку. Тем самым обеспечивается целостность системы ограничений.

В СУБД INGRES делается попытка примирить контроль ограничений и эффективность функционирования. При массовом копировании данных контроль ограничений отключается. Это значит, что необходимо дополнять копирование запуском процедуры глобальной проверки целостности.

Правила. Правила позволяют вызывать выполнение заданных действий при определенных изменениях базы данных. Обычно действие - это вызов

процедуры. Правила ассоциируются с таблицами и срабатывают при изменении этих таблиц.

В отличие от ограничений, которые являются лишь средством контроля относительно простых условий, правила позволяют проверять и поддерживать сколь угодно сложные соотношения между элементами данных в базе. Как и в случае ограничений, проверка правил отключается при массовых операциях копирования. Администратор базы данных может также явным образом отменить проверку правил, воспользовавшись оператором

SET NORULES;

Оператор

SET RULES;

позволит затем восстановить работу механизма правил. По умолчанию этот механизм включен.

Для удаления правил служит оператор

DROP RULE правило;

СУБД обеспечивает автоматическое удаление правил в тех случаях, когда удаляется соответствующая таблица. Тем самым поддерживается целостность системы таблиц и правил.

В контексте информационной безопасности важно отметить, что создать правило, ассоциируемое с таблицей, может владелец этой таблицы, имеющий право на выполнение соответствующей процедуры. Пользователь, действия которого вызывают срабатывание правила, должен обладать лишь необходимыми правами доступа к таблице. Тем самым правила неявно расширяют привилегии пользователей. Подобные расширения нуждаются в строгом административном контроле, поскольку даже незначительное изменение правила или ассоциированной процедуры может кардинально повлиять на защищенность данных. Ошибка же в сложной системе правил вообще чревата непредсказуемыми последствиями.

Средства поддержания высокой готовности. В коммерческих приложениях высокая готовность аппаратно-программных комплексов является важнейшим фактором. Применительно к СУБД средства поддержания высокой готовности должны обеспечивать нейтрализацию аппаратных отказов, особенно касающихся дисков, а также восстановление после ошибок обслуживающего персонала или прикладных программ.

Подобные средства должны с самого начала закладываться в архитектуру комплекса. Например, необходимо использовать тот или иной вид избыточных дисковых массивов. Конечно, это сделает аппаратно-программное решение более дорогим, но зато убережет от возможных убытков во время эксплуатации.

Кластерная организация сервера баз данных. Обычно кластер содержит также несколько дисковых подсистем, совместно используемых узлами-компьютерами, и избыточные связи между компонентами. С внешней точки зрения кластер выглядит как единое целое, а наличие нескольких узлов способствует повышению производительности и устойчивости к отказам.

Тиражирование данных. В контексте информационной безопасности тиражирование можно рассматривать как средство повышения доступности данных. Стала легендой история про бакалейщика из Сан-Франциско, который после разрушительного землетрясения восстановил свою базу данных за 16 минут, перекачав из другого города предварительно протиражированную информацию.

Развитые возможности тиражирования предоставляет СУБД INGRES. В Informix OnLine-DS 7.1 поддерживается модель тиражирования, состоящая в полном отображении данных с основного сервера на вторичные.

В конфигурации серверов Informix OnLine-DS с тиражированием выделяется один основной и ряд вторичных серверов. На основном сервере выполняется и чтение, и обновление данных, а все изменения передаются на вторичные серверы, доступные только на чтение (рис. 5.1). В случае отказа основного сервера вторичный автоматически или вручную переводится в

режим доступа на чтение и запись (рис. 5.2). Прозрачное перенаправление клиентов при отказе основного сервера не поддерживается, но оно может быть реализовано в рамках приложений.

После восстановления основного сервера возможен сценарий, при котором этот сервер становится вторичным, а бывшему вторичному, который уже функционирует в режиме чтения-записи, придается статус основного; клиенты, которые подключены к нему, продолжают работу. Таким образом, обеспечивается непрерывная доступность данных.

Тиражирование осуществляется путем передачи информации из журнала транзакций (логического журнала) в буфер тиражирования основного сервера, откуда она пересылается в буфер тиражирования вторичного сервера. Такая пересылка может происходить либо в синхронном, либо в асинхронном режиме. Синхронный режим гарантирует полную согласованность баз данных - ни одна транзакция, зафиксированная на основном сервере, не останется незафиксированной на вторичном, даже в случае сбоя основного сервера. Асинхронный режим не обеспечивает абсолютной согласованности, но улучшает рабочие характеристики системы.

Побочный положительный эффект тиражирования - возможность вынести преимущественно на вторичный сервер ресурсоемкие приложения поддержки принятия решений. В этом случае они могут выполняться с максимальным использованием средств параллельной обработки, не подавляя приложений оперативной обработки транзакций, сосредоточенных на основном сервере. Это также можно рассматривать как фактор повышения доступности данных.

Защита коммуникаций между сервером и клиентами. Проблема защиты коммуникаций между сервером и клиентами не является специфичной для СУБД, она присуща всем распределенным системам. Вполне естественно, что и решения здесь ищутся общие, такие, например, как в распределенной вычислительной среде (Distributed Computing Environment, DCE) концерна OSF. Разработчикам СУБД остается

«погрузить» свои программные продукты в эту среду, что и сделала компания Informix, реализовав Informix- DCE/Net.

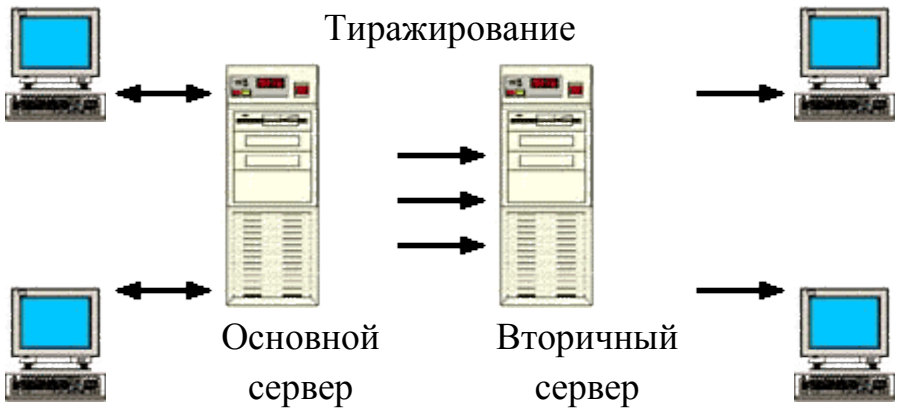


Рис. 5.1. Тиражирование. Основной сервер доступен на чтение и запись, вторичный - только на чтение

Рис. 5.2. Когда основной сервер выходит из строя, вторичный переводится в режим доступа и на чтение, и на запись

Informix-DCE/Net открывает доступ к сервисам DCE для всех инструментальных средств Informix, а также любых приложений или инструментальных комплексов от независимых поставщиков, которые используют интерфейс ODBC (рис. 5.3).

Ключевым компонентом в реализации взаимодействий клиент-сервер в среде DCE является сервис безопасности. Основные функции, предоставляемые этим сервисом, - аутентификация, реализуемая средствами Kerberos, авторизация (проверка полномочий) и шифрование.

Informix-DCE/Net использует все средства обеспечения безопасности, имеющиеся в DCE. Например, для каждого приложения клиент-сервер администратор может задать один из пяти уровней защиты:

- защита пересылаемых данных только при установлении соединения клиента с сервером;

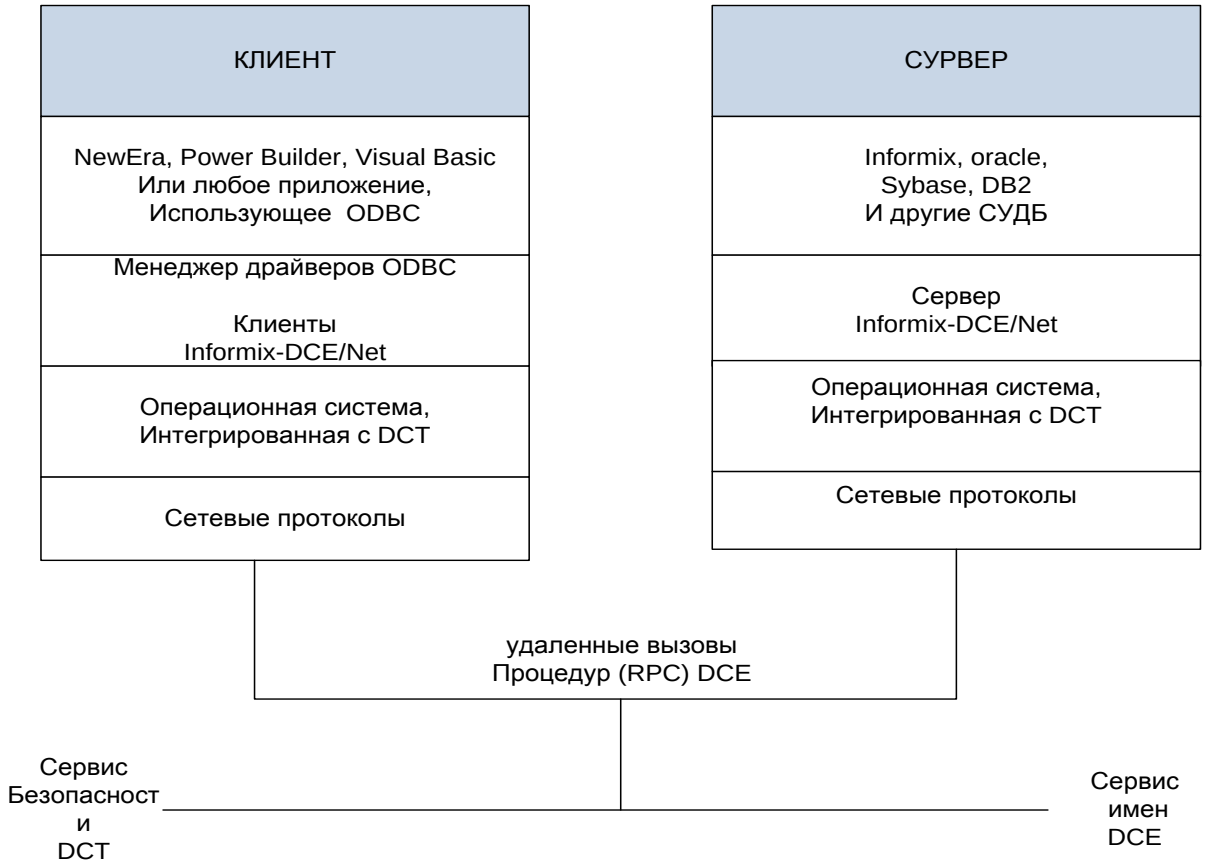


Рис. 5.3. Конфигурация прикладной или инструментальной среды клиент-сервер, использующей Informix-DCE/Net

- защита данных только на начальном этапе выполнения удаленного вызова процедуры, когда сервер впервые получает запрос;
- подтверждение подлинности источника данных. Проверяется, что все поступающие на сервер данные получены от определенного клиента;
- подтверждение подлинности источника и целостности данных. Проверяется, что отправленные данные не были изменены;

- подтверждение подлинности источника, целостности и конфиденциальности данных. Выполняются проверки, предусмотренные на предыдущем уровне и осуществляется шифрование всех пересылаемых данных.

Сервис аутентификации DCE, поддерживаемый Informix-DCE/Net, существенно улучшает характеристики безопасности распределенной среды, упрощая, в то же время, деятельность как пользователей, так и администраторов. Достаточно иметь единое входное имя и пароль для DCE, чтобы обращаться к любой погруженной в эту среду базе данных. При запуске приложения Informix-DCE/Net запрашивает аутентификационную информацию пользователя у DCE, и подключает его к требуемой базе.

Наличие единой точки администрирования входных имен и прав доступа к базам данных и приложениям способствует упорядочению общей ситуации с безопасностью. Например, если уничтожается входное имя DCE, то администратор может быть уверен, что данный пользователь уже не сможет получить доступ ни к одному из системных ресурсов.

Контрольные вопросы

1. Механизмы архитектуры подсистемы безопасности базы данных.
2. Виды привилегий в СУБД.
3. Объясните использование представлений при управлении доступом.
4. Иерархия прав доступа.
5. Метка безопасности и принудительный контроль доступа.
6. Из каких компонентов состоит метка безопасности?
7. Объясните суть кластерной организации сервера баз данных.
8. Что понимается под тиражированием данных?
9. Каким образом происходит защита коммуникации между сервером и клиентами?

5.2. Профили защиты систем управления базами данных

СУБД, как и операционные системы, содержат комбинацию сервисов безопасности, однако, в отличие от ОС, не являются самодостаточными. СУБД используют механизмы и функции ОС. Такая двухуровневость ведет к появлению специфических угроз и требует привлечения соответствующих средств противодействия. Например, базы данных располагаются в файлах или на дисках, управляемых ОС, следовательно, к объектам БД можно обратиться как штатными средствами СУБД, так и с помощью механизмов ОС, получив доступ к файлу или устройству. Подобные возможности должны учитываться в профиле защиты (ПЗ) для СУБД (его прототип соответствует классу безопасности С2 «Оранжевой книги»). Здесь вводится понятие аутентификационного пакета, который предоставляет для СУБД механизм подтверждения подлинности заявляемого идентификатора пользователя. Еще одно проявление упомянутой выше двухуровневости - предположение безопасности базовой конфигурации, состоящее в том, что базовая система (операционная система, и/или сетевые сервисы безопасности, и/или специальное программное обеспечение) установлены, сконфигурированы и управляются безопасным образом. Аналогичную направленность имеют цели безопасности для среды, предусматривающие, что базовая система должна обеспечить механизмы управления доступом, которые позволят защитить от несанкционированного доступа все связанные с СУБД файлы; кроме того, ОС предоставит средства для изоляции функций безопасности и защиты процессов СУБД. Можно отметить, что в распределенной среде управление доступом и изоляция могут поддерживаться не только средствами базовой ОС, но и архитектурно, путем разнесения компонентов СУБД по узлам сети и использования межсетевых экранов.

Переходя к функциональным требованиям безопасности, можно указать на важность требований согласованности данных между функциями

безопасности (FPT_TDC), а также согласованности данных функций безопасности при дублировании в пределах распределенного объекта оценки (FPT_TRC). Согласованность достигается с помощью некоторой формы обработки распределенных транзакций или путем обновления дублируемых данных с применением какого-либо протокола синхронизации.

Для защиты от атак на доступность в профиле защиты предусмотрены реализация квот, выделяемых пользователям (FRU_RSA.1), а также базовые ограничения на параллельные сеансы (FTA_MCS.1).

Необходимо учесть специфику современных СУБД, в частности, требования обеспечения динамической целостности данных, реализуемые механизмом транзакций. Требования безопасного восстановления носят слишком общий характер. Защита от стандартных угроз, существующих в сетевой среде, целиком переложена на базовую систему.

Документ профиль защиты системы управления базами данных (Database Management System Protection Profile) определяет требования безопасности для систем управления базой данных в организациях, где имеются требования для защиты конфиденциальности, целостности и доступности информации, хранимой в базе данных. Несанкционированное раскрытие, модификация или отказ в обслуживании такой информации может оказать неблагоприятное воздействие на функционирование организации. Данный ПЗ определяет:

- совокупность *основных требований*, которым все совместимые с ПЗ базы данных должны удовлетворять;

- совокупность *аутентификационных пакетов* (один или более таких пакетов должны быть обеспечены в совместимой с ПЗ базе данных).

Администраторы этих систем имеют возможность:

- управлять и постоянно контролировать действия конечных пользователей, чтобы способствовать предотвращению нарушения их прав в пределах системы;

- управлять потреблением ресурса индивидуальными пользователями, и учитывать действия пользователей.

Аутентификационные пакеты предоставляют средства для подтверждения подлинности пользователя:

- аутентификация в ОС (пользователь аутентифицирован ОС хоста и идентифицирован в базе данных);

- аутентификация в базе данных (пользователь идентифицирован и аутентифицирован СУБД).

Подход разделения основных требований и аутентификационных пакетов был принят для того, чтобы упростить сопровождение этого профиля защиты. Это предполагает, что в будущих изданиях этого профиля защиты список предлагаемых аутентификационных пакетов может расширяться, например, чтобы включить каталог, основанный на аутентификации.

Для того чтобы заявить соответствие этому профилю защиты в задании по безопасности, должно быть установлено аутентификационный пакет. Заявления соответствия ПЗ должны устанавливать либо «СУБД в режиме аутентификации средствами ОС», либо «СУБД в режиме аутентификации средствами базы данных», либо «СУБД в режиме аутентификации средствами ОС и средствами базы данных».

Профили защиты СУБД определены для некой обобщенной среды со средним уровнем риска для активов. Требования доверия и минимальная стойкость функций выбраны в соответствии с этим уровнем риска.

Обычно СУБД используется для того, чтобы предоставить многим пользователям одновременный доступ к базе данных.

СУБД может быть сконфигурирована многими способами:

- *автономная система* с одиночным пользователем базы данных (например, приложение, основанное на работе отдельного пользователя на персональном компьютере);

- много пользователей базы данных, работающих за *терминалами, которые связаны с центральной машиной* (например, традиционный вариант терминал – среда мэйнфрейма);

- *сеть интеллектуальных рабочих станций, поддерживающих связь с центральным сервером* (архитектура «клиент-сервер»);

- *сеть интеллектуальных клиентских рабочих станций, поддерживающих связь с приложением сервера*, которое в свою очередь связано с СУБД (например, Web-браузер, поддерживающий связь с Web-сервером, который формирует динамические страницы с помощью СУБД).

В любой из вышеупомянутых конфигураций сами данные могут постоянно находиться на одном сервере или могут быть распределены среди многих независимых серверов.

СУБД представляет собой приложение, использующее функции нижележащей базовой системы (операционной системы хоста и/или сетевых сервисов, и/или специально заказанного программного обеспечения), и является ИТ-компонентом конкретной системы.

Приложение СУБД может состоять из одного или нескольких выполняемых загрузочных модулей и одного или нескольких файлов данных. Они будут подчинены администрированию основных системных прав, как и любые другие основные системные процессы и файлы.

СУБД может расширять функциональные возможности средств обеспечения безопасности базовой системы. Например, база данных может реализовать гораздо лучший разветвленный механизм привилегий, чем операционная система хоста.

Аутентификационные пакеты. Аутентификационный пакет предоставляет для базы данных механизм подтверждения подлинности заявляемого идентификатора пользователя. В пределах данного профиля защиты это может быть обеспечено следующими двумя механизмами.

Внешним – с помощью операционной системы хоста (аутентификация средствами ОС). В этой схеме аутентификации при идентификации и

аутентификации пользователя база данных полагается на операционную систему хоста, которая в этом случае обеспечивает подтверждение подлинности пользователя в базе данных. База данных использует предоставленный операционной системой идентификатор для установления идентификатора базы данных.

Непосредственно в пределах базы данных (аутентификация средствами базы данных). В этой схеме аутентификации база данных верифицирует заявляемый идентификатор пользователя, используя свой собственный механизм аутентификации.

По крайней мере, один из вышеупомянутых сервисов аутентификации должен быть предоставлен соответствующей базой данных. Активы ИТ, требующие защиты, состоят из информации, хранимой в пределах СУБД, конфиденциальность, целостность или доступность которой может быть скомпрометирована. Активами ИТ являются: объекты базы данных и данные, содержащиеся в пределах этих объектов базы данных.

Объектами БД могут быть объединения частей данных, содержащихся в других объектах базы данных. Данные управления базой данных используются СУБД для того, чтобы организовать и защитить объекты базы данных. Данные аудита базы данных генерируются СУБД в процессе ее функционирования.

Принятые угрозы безопасности СУБД, наряду с нарушителями, которые могли бы провоцировать эти угрозы, определены ниже.

Этим угрозам будут противостоять:

а) технические меры безопасности, предоставленные СУБД, вместе с

б) техническими мерами безопасности, предоставленными базовой системой;

в) не технические операционные меры безопасности в среде (процедурные, физические меры и относящиеся к персоналу).

Нарушителями могут быть:

- лица, которые не являются уполномоченными пользователями базовой системы (операционной системы и/или сетевых сервисов, и/или специального программного обеспечения);

- лица, которые являются уполномоченными пользователями СУБД;

- лица, которые являются уполномоченными пользователями базовой системы.

Пользователями этой системы могут быть:

а) лица, которые не являются пользователями базы данных;

б) лица, которые являются пользователями базы данных.

Угрозы, предотвращаемые СУБД. Нарушители могут инициировать следующие типы угроз СУБД (или СУБД должна противостоять следующим угрозам):

T.ACCESS - *несанкционированный доступ к базе данных*. Посторонний или пользователь системы, который в настоящее время не является уполномоченным пользователем базы данных, обращается к СУБД.

T.DATA - *несанкционированный доступ к информации*. Уполномоченный пользователь базы данных обращается к информации, содержащейся в пределах СУБД, без разрешения пользователя базы данных, который является собственником данных или который отвечает за защиту данных. Эта угроза включает несанкционированный доступ к информации СУБД, остаточной информации, хранящейся в памяти или в ресурсах хранения, управляемых СУБД, или к данным управления БД.

T.RESOURCE - *чрезмерное использование ресурсов*. Аутентифицированный пользователь базы данных использует глобальные ресурсы базы данных путем, который ставит под угрозу возможность других пользователей базы данных получить доступ к СУБД. Эта угроза относится к доступности информации в пределах СУБД. Например, пользователь базы данных мог выполнять действия, связанные с использованием чрезмерных ресурсов, периодически препятствуя законному доступу других пользователей базы данных к данным, ресурсам и сервисам. Такие

нападения могут быть злонамеренными, происходить в результате невнимательности или небрежности, или в случае, когда пользователь базы данных может просто не сознавать потенциальные последствия своих действий. Воздействие таких нападений на готовность и надежность системы может быть усилено многими пользователями, действующими одновременно.

T. ATTACK - *необнаруженное нападение*. Необнаруженная компрометация СУБД происходит в результате действий нарушителя (уполномоченного или неуполномоченного пользователя базы данных), пытающегося выполнить действия, которые он не уполномочен выполнять. Эта угроза включена, потому что независимо от обеспечения контрмер, адресованным другим угрозам, все же имеется еще остаточная угроза нарушения политики безопасности нарушителями, пытающимися противостоять этим контрмерам.

T.ABUSE.USER - *неправильное использование привилегий*. Необнаруженная компрометация СУБД происходит в результате действий пользователя базы данных (преднамеренных или нет), Например, пользователь базы данных может предоставить доступ к объекту БД, ответственным за который он является, другому пользователю базы данных, способному использовать эту информацию для мошеннических целей. Отметим, что эта угроза не распространяется на пользователей базы данных с высоким уровнем доверия.

Угрозы, предотвращаемые средой:

T.OPERATE - *опасная операция*. Компрометация базы данных может произойти из-за неправильной конфигурации, администрирования и/или функционирования сложной системы.

T.CRASH - *внезапные прерывания*. Внезапные прерывания функционирования СУБД могут приводить к потере или разрушению данных, связанных с безопасностью, таких как данные управления БД и данные аудита. Такие прерывания могут являться результатом ошибки

оператора или сбоев программного обеспечения, аппаратных средств, источников питания или носителей данных.

T.PHYSICAL - *физическое нападение*. Критичные к безопасности части СУБД или базовой операционной системы и/или сетевых сервисов могут быть подвергнуты физическому нападению, которое может нарушить безопасность.

P.ACCESS - Доступ к объектам БД определяется:

- владельцем объекта БД;

- идентификатором субъекта базы данных, пытающегося получить доступ;

- привилегиями доступа к объекту БД, которыми владеет субъект базы данных;

- административными привилегиями субъекта базы данных;

- ресурсами, выделенными субъекту.

Заметим, что эта политика включает следующее:

- *владение* – владельцы объектов БД ответственны за свои объекты;

- *дискреционное управление доступом* – владельцы объектов БД могут предоставлять другим пользователям базы данных доступ или управление своими объектами БД на основе дискреционного управления доступом;

- *ресурсы* - пользователи базы данных уполномочены использовать только те ресурсы, которые распределены им.

P.ACCOUNT - Пользователи базы данных ответственны за:

- операции на объектах, которые определены владельцем объекта;

- действия, определенные администраторами базы данных.

СУБД зависит как от технических аспектов ИТ, так и от функциональных аспектов ее среды.

Предположения по СУБД:

A.TOE.CONFIG - СУБД инсталлирован, сконфигурирован и управляется в соответствии со своей оцененной конфигурацией.

Основные системные предположения.

Физические предположения:

A.PHYSICAL - ресурсы функционирования СУБД и базовой системы расположены в пределах управления средствами доступа, которые предотвращают несанкционированный физический доступ посторонних, пользователей системы и пользователей базы данных.

Предположения конфигурации:

A.SYS.CONFIG - базовая система (операционная система и/или сервисы безопасности сети, и/или специальное программное обеспечение) инсталлированы, сконфигурированы и управляются в соответствии со своей безопасной конфигурацией.

A.ACCESS - базовая система конфигурирована так, что только санкционированная группа лиц может получить доступ к системе.

A.MANAGE - будут назначены одно или более компетентных доверенных лиц для того, чтобы управлять СУБД, базовой системой и безопасностью информации.

Предположения связности:

A.PEER - предполагается, что любые другие компоненты ИТ, с которыми взаимодействует СУБД, будут под тем же самым управлением и функционируют под той же самой политикой безопасности.

A.NETWORK - предполагается, что в распределенной среде базовые сервисы сети будут основаны на безопасных протоколах взаимодействия, которые обеспечат аутентичность пользователей.

В таблице 5.1 представлено отношение целей безопасности СУБД к каждой из угроз и политик безопасности и показано, что всякой угрозе соответствует, по крайней мере, одна цель безопасности ИТ, и что всякая политика безопасности удовлетворена, по крайней мере, одной целью безопасности ИТ. В таблице слово «ДА» указывает, что указанная цель безопасности ИТ уместна для определенной угрозы или политики безопасности.

O.ACCESS - СУБД должен обеспечить конечных пользователей и администраторов возможностью управления доступом к их собственным данным или ресурсам или к тем, за которые они отвечают в соответствии с политикой безопасности P.ACCESS. Для этого СУБД имеет следующие более конкретные цели:

Таблица 5.1

Взаимосвязь угроз и политик с целями безопасности СУБД

Политики / Угрозы	O.I&A.TOE	O.ACCESS	O.AUDIT	O.RESOURCE	O.ADMIN.TOE
T.ACCESS	ДА	ДА		ДА	ДА
T.DATA	ДА	ДА			ДА
T.RESOURCE	ДА	ДА		ДА	ДА
T.ATTACK	ДА	ДА	ДА		ДА
T.ABUSE.USER	ДА	ДА	ДА		ДА
P.ACCESS		ДА		ДА	
P.ACCOUNT		ДА	ДА		

O.ACCESS.OBJECTS - СУБД должен предотвратить несанкционированное или непредусмотренное раскрытие, ввод, модификацию или уничтожение данных и объектов базы данных, а также просмотр базы данных, управление данными и аудит данных базы данных.

O.ACCESS.CONTROL - СУБД должен предоставить возможность пользователям базы данных, которые являются собственниками или ответственными за данные, управлять доступом к этим данным других уполномоченных пользователей базы данных.

O.ACCESS.RESIDUAL - СУБД должен предотвратить несанкционированный доступ к остаточным данным, остающимся в объектах и ресурсах после использования этих объектов и ресурсов.

O.RESOURCE - СУБД должен предоставить средства управления использованием ресурсов базы данных уполномоченными пользователями СУБД.

O.I&A.TOE - СУБД с поддержкой или без поддержки базовой системы должен предоставить средства идентификации и аутентификации пользователей СУБД.

O.AUDIT - СУБД должен предоставить средства подробной регистрации значимых для безопасности событий для того, чтобы в достаточной мере помочь администратору СУБД:

- обнаруживать предпринятые нарушения безопасности или потенциальную ошибку в конфигурации средств безопасности СУБД, которые оставили бы базу данных незащищенной от компрометации;
- обязать индивидуальных пользователей базы данных быть ответственными за любые выполняемые ими действия, которые являются значимыми для безопасности базы данных в соответствии с политикой P.ACCOUNT.

O.ADMIN.TOE - там, где необходимо, СУБД вместе с базовой системой должен предоставить функции, позволяющие уполномоченному администратору эффективно управлять СУБД и его функциями безопасности, обеспечивая, чтобы только уполномоченные администраторы могли получать доступ к такой функциональности.

Цели безопасности для среды. Следующие цели безопасности ИТ должны быть удовлетворены средой, в которой СУБД используется:

O.ADMIN.ENV - там где необходимо, СУБД вместе с базовой системой должен предоставить функциональные возможности, позволяющие уполномоченному администратору эффективно управлять СУБД и его функциями безопасности, обеспечивая, чтобы только уполномоченные администраторы могли получать доступ к такой функциональности.

O.FILES - базовая система должна обеспечить механизмы управления доступом, которые позволят защитить от несанкционированного доступа все связанные с СУБД файлы и каталоги (включая выполняемые программы, библиотеки рабочих программ, файлы базы данных, экспортируемые файлы,

файлы повторной регистрации, управляемые файлы, файлы с трассировкой и файлы с дампом).

O.I&A.ENV - базовая операционная система должна предоставить средства идентификации и аутентификации пользователей, когда требуется с помощью СУБД надежно подтвердить подлинность пользователей.

O.SEP - базовая операционная система должна предоставить средства для изоляции функций безопасности СУБД и уверенность в том, что ее компоненты не будут искажаться. Составляющими, реализующие функции безопасности СУБД, являются: 1) файлы, используемые СУБД для того, чтобы хранить базу данных и 2) процессы СУБД, управляющие базой данных.

Следующие, не связанные с ИТ, цели безопасности должны быть удовлетворены процедурными и другими мерами, предпринятыми в пределах среды ОО.

O.INSTALL - Ответственные за СУБД должны обеспечить, чтобы:

- ОО был поставлен, инсталлирован, управлялся и использовался в соответствии с эксплуатационной документацией СУБД;

- базовая система была инсталлирована и использовалась в соответствии с ее эксплуатационной документацией. Если элементы системы сертифицированы, то они должны быть инсталлированы и использоваться в соответствии с необходимой документацией сертификации.

O.PHYSICAL - ответственный за СУБД должен обеспечить, чтобы те части СУБД, которые являются критичными к политике безопасности, были защищены от физического нападения.

O.AUDITLOG - администраторы базы данных должны обеспечить, чтобы средства аудита использовались и управлялись эффективно. Эти процедуры должны применяться в журнале аудита базы данных и/или журнале аудита для базовой операционной системы, и/или для сетевых сервисов безопасности. В особенности:

- должны быть предприняты необходимые действия для того, чтобы обеспечить продолжительное функционирование аудита, например, для того, чтобы обеспечить достаточную свободную память, необходимую для регулярной архивации журнала аудита;

- журналы регистрации событий аудита должны регулярно просматриваться и необходимо определить действия или события, которые могут привести к нарушению безопасности в будущем;

- системные часы должны быть защищены от несанкционированной модификации (так, чтобы целостность меток времени аудита не была скомпрометирована).

O.RECOVERY - ответственный за СУБД должен предоставить возможность для процедур и/или механизмов восстановления функционирования на месте после системного сбоя или другого прерывания.

O.QUOTA - администраторы базы данных должны обеспечивать, чтобы каждый пользователь СУБД имел необходимые квоты, которые:

- достаточны для выполнения операций, к которым пользователь имеет доступ;

- достаточно ограничены, чтобы пользователь не мог нарушить режим эксплуатации, доступ к ресурсам и монополизировать ресурсы.

O.TRUST - ответственный за СУБД должен обеспечить, чтобы только у пользователей с высоким уровнем доверия была привилегия, которая позволяет им:

- устанавливать или изменять конфигурацию журнала аудита для базы данных;

- изменять или удалять любую запись аудита в журнале аудита базы данных;

- создавать любые учетные данные пользователя или изменять любые атрибуты безопасности пользователя;

- предоставлять полномочия на использование административных привилегий.

O.AUTHDATA - ответственный за СУБД должен обеспечить, чтобы данные аутентификации для любых учетных данных пользователя СУБД, так же как и для базовой системы, надежно поддерживалась и не раскрывались лицам, которые не уполномочены использовать эту учетную запись. В особенности:

- носители, на которых хранятся данные аутентификации для базовой операционной системы и/или сетевых сервисов безопасности, не должны быть физически устранимы из базовой платформы несанкционированными пользователями;
- пользователи не должны раскрывать свои пароли другим лицам;

в) пароли, сгенерированные администратором системы, должны быть распределены безопасным способом.

O.MEDIA - ответственный за СУБД должен обеспечить, чтобы конфиденциальность, целостность и доступность данных, хранимых на носителях данных, были адекватно защищены. В особенности:

- сетевые и автономные устройства хранения данных, на которых располагается база данных и данные, связанные с безопасностью (такие как, резервные копии операционной системы, резервные копии базы данных, журналы транзакций и журналы аудита), физически не могли быть несанкционированно устранены из базовой платформы пользователями.

- сетевые и автономные устройства хранения данных должны подходящим образом сохраняться и поддерживаться, а также регулярно проверяться для того, чтобы обеспечить целостность и доступность связанных с безопасностью данных.

- носители, на которых хранятся связанные с базой данных файлы (включая файлы базы данных, экспортные файлы, файлы повторной регистрации, файлы управления, файлы трассировки и файлы дампов), будут очищены до того, как они будут повторно использоваться в целях, не связанных с базой данных.

Таблица 5.2 иллюстрирует, как каждая из вышеупомянутых целей противопоставлена угрозе, поддерживает цель безопасности СУБД, поддерживает политику или отображена в предположениях о безопасном использовании.

Таблица 5.2

Отображение целей безопасности среды на угрозы, цели безопасности СУБД, политику и предположения о безопасном использовании

Цели безопасности среды	**Противостоящая угроза**	**Поддерживаемая цель СУБД**	**Поддерживаемая политика**	**Отображение в предположении о безопасном использовании**
O.INSTALL	T.OPERATE			A.TOE.CONFIG, A.SYS.CONFIG, A.MANAGE
O.PHYSICAL	T.PHYSICAL			A.ACCESS, A.PEER, A.PHYSICAL
O.AUDITLOG		O.AUDIT	P.ACCOUNT	A.MANAGE
O.RECOVERY	T.CRASH			A.MANAGE
O.QUOTA		O.RESOURCE		A.MANAGE
O.TRUST			P.ACCESS	A.MANAGE
O.AUTHDATA		O.I&A.TOE	P.ACCESS	A.MANAGE, A.PEER, A.NETWORK
O.MEDIA	T.CRASH			A.MANAGE
O.ADMIN.ENV		O.ADMIN.TOE		A.MANAGE
O.FILES	T.ACCESS		P.ACCESS	A.MANAGE
O.I&A.ENV	T.ACCESS	O.I&A.TOE	P.ACCESS	A.MANAGE
O.SEP	T.ACCESS		P.ACCESS	A.MANAGE

Любой заявленный для соответствия этому ПЗ объект оценки должен, как минимум, обеспечивать выполнение всех функциональных требований безопасности, как определено в основных требованиях.

Дополнительно, любой соответствующий СУБД должен идентифицировать и обеспечить выполнение, по крайней мере, одного из указанных пакетов аутентификации. Для каждого заявленного пакета аутентификации СУБД должен обеспечивать выполнение всех соответствующих функциональных требований безопасности.

Контрольные вопросы

1. Что такое профили защиты систем управления базами данных?
2. Что такое аутентификационные пакеты?
3. Перечислите способы конфигурирования системы управления базами данных.
4. Перечислите угрозы на безопасность объекта оценки и меры, противостоящие этим угрозам.
5. Перечислите нарушителей профили защиты в системе управления базами данных.
6. Какие задачи приведены в документе «Профиль защиты системы управления базами данных»?

5.3. Нормативные документы в области обеспечения безопасности базы данных

Специалистам в области информационной безопасности (ИБ) сегодня почти невозможно обойтись без знаний соответствующих стандартов и спецификаций. На то имеется причина в том, что необходимость исследования некоторым стандартам закреплена законодательно. Однако наиболее убедительны содержательные причины. Во-первых, стандарты и спецификации - одна из форм накопления знаний, прежде всего о процедурном и программно-техническом уровнях ИБ. В них зафиксированы

апробированные, высококачественные решения и методологии, разработанные наиболее квалифицированными специалистами. Во-вторых, и те, и другие являются основным средством обеспечения взаимной совместимости аппаратно-программных систем и их компонентов, причем в Internet-сообществе это средство действительно работает, и весьма эффективно.

Основные понятия стандарта и спецификации:

- стандарт - документ, в котором в целях добровольного многократного использования устанавливаются характеристики продукции, правила осуществления и характеристики процессов производства, эксплуатации, хранения, перевозки, реализации и утилизации, выполнения работ или оказания услуг. Стандарт также может содержать требования к терминологии, символике, упаковке, маркировке или этикеткам и правилам их нанесения;
- стандартизация - деятельность по установлению правил и характеристик в целях их добровольного многократного использования, направленная на достижение упорядоченности в сферах производства и обращения продукции и повышение конкурентоспособности продукции, работ или услуг.

С практической точки зрения, количество стандартов и спецификаций (международных, национальных, отраслевых и т.п.) в области информационной безопасности бесконечно. Чтобы охватить различные аспекты информационной безопасности, разные виды и конфигурации информационно-коммуникационных систем (ИКС), предоставить полезные сведения для самых разнообразных групп целевой аудитории, можно их выделить на следующие уровни.

На верхнем уровне можно выделить две существенно отличающиеся друг от друга группы стандартов и спецификаций:

- оценочные стандарты, предназначенные для оценки и классификации информационных систем и средств защиты по требованиям безопасности;
- спецификации, регламентирующие различные аспекты реализации и использования средств и методов защиты.

Эти группы, разумеется, не конфликтуют, а дополняют друг друга. Оценочные стандарты описывают важнейшие, с точки зрения информационной безопасности, понятия и аспекты ИС, играя роль организационных и архитектурных спецификаций. Другие спецификации определяют, как именно строить ИКС предписанной архитектуры и выполнять организационные требования.

Технические спецификации, применимые к современным распределенным ИКС, создаются, главным образом, «Тематической группой по технологии Internet» (Internet Engineering Task Force, IETF) и ее подразделением - рабочей группой по безопасности. Ядром рассматриваемых технических спецификаций служат документы по безопасности на IP-уровне (IPsec). Кроме этого, анализируется защита на транспортном уровне (Transport Layer Security, TLS), а также на уровне приложений (спецификации GSS-API, Kerberos). Необходимо отметить, что Internet-сообщество уделяет должное внимание административному и процедурному уровням безопасности («Руководство по информационной безопасности предприятия», «Как выбирать поставщика Интернет-услуг», «Как реагировать на нарушения информационной безопасности»).

В вопросах сетевой безопасности невозможно разобраться без освоения спецификаций X.800 «Архитектура безопасности для взаимодействия открытых систем», X.500 «Служба директорий: обзор концепций, моделей и сервисов» и X.509 «Служба директорий: каркасы сертификатов открытых ключей и атрибутов».

Британский стандарт BS 7799 «Управление информационной безопасностью. Практические правила», полезно для руководителей

организаций и лиц, отвечающих за информационную безопасность, без сколько-нибудь существенных изменений воспроизведено в международном стандарте O’z DSt ISO/IEC 27000-2008.

Первым оценочным стандартом, получившим международное признание и оказавшим исключительно сильное влияние на последующие разработки в области информационной безопасности, стал стандарт Министерства обороны США «Критерии оценки доверенных компьютерных систем» (Department of Defense Trusted Computer System Evaliation Criteria, TCSEC,), более известный (по цвету обложки) под названием «Оранжевая книга».

Без преувеличения можно утверждать, что в «Оранжевой книге» заложен понятийный базис ИБ. Достаточно лишь перечислить содержащиеся в нем понятия: безопасная и доверенная системы, политика безопасности, уровень гарантированности, подотчетность, доверенная вычислительная база, монитор обращений, ядро и периметр безопасности. Исключительно важно и выделение таких аспектов политики безопасности, как добровольное (дискреционное) и принудительное (мандатное) управление доступом, безопасность повторного использования объектов. Последним по порядку, но отнюдь не по значению следует назвать принципы классификации по требованиям безопасности на основе параллельного ужесточения требований к политике безопасности и уровню гарантированности.

После «Оранжевой книги» была выпущена целая «Радужная серия». С концептуальной точки зрения, наиболее значимый документ в ней - «Интерпретация «Оранжевой книги» для сетевых конфигураций» (Trusted Network Interpretation). Он состоит из двух частей. Первая содержит собственно интерпретацию, во второй описываются сервисы безопасности, специфичные или особенно важные для сетевых конфигураций.

Важнейшее понятие, введенное в первой части, - сетевая доверенная вычислительная база. Другой принципиальный аспект - учет динамичности сетевых конфигураций. Среди защитных механизмов выделена

криптография, помогающая поддерживать как конфиденциальность, так и целостность.

Новым для своего времени стал систематический подход к вопросам доступности, формирование архитектурных принципов ее обеспечения.

Упомянем также достаточное условие корректности фрагментирования монитора обращений, являющееся теоретической основой декомпозиции распределенной ИС в объектно-ориентированном стиле в сочетании с криптографической защитой коммуникаций.

Переходя к знакомству с «Гармонизированными критериями Европейских стран», отметим отсутствие в них априорных требований к условиям, в которых должна работать информационная система. Предполагается, что сначала формулируется цель оценки, затем орган сертификации определяет, насколько полно она достигается, т. е. в какой мере корректны и эффективны архитектура и реализация механизмов безопасности в конкретной ситуации. Чтобы облегчить формулировку цели оценки, стандарт содержит описание десяти примерных классов функциональности, типичных для правительственных и коммерческих систем.

В «Гармонизированных критериях» подчеркивается различие между системами и продуктами информационных технологий, но для унификации требований вводится единое понятие - объект оценки.

Важно указание и на различие между функциями (сервисами) безопасности и реализующими их механизмами, а также выделение двух аспектов гарантированности - эффективности и корректности средств безопасности.

Первое примечательное отклонение от этого курса произошло в 1997 году, когда был принят РД по отдельному сервису безопасности - межсетевым экранам (МЭ). Его основная идея - классифицировать МЭ на основании осуществляющих фильтрацию потоков данных уровней эталонной

семиуровневой модели - получила международное признание и продолжает оставаться актуальной.

Среди технических спецификаций на первое место, безусловно, следует поставить документ X.800 «Архитектура безопасности для взаимодействия открытых систем». Здесь выделены важнейшие сетевые сервисы безопасности: аутентификация, управление доступом, обеспечение конфиденциальности и/или целостности данных, а также невозможность отказаться от совершенных действий. Для реализации сервисов предусмотрены следующие сетевые механизмы безопасности и их комбинации: шифрование, электронная цифровая подпись (ЭЦП), управление доступом, контроль целостности данных, аутентификация, дополнение трафика, управление маршрутизацией, нотаризация. Выбраны уровни эталонной семиуровневой модели, на которых могут быть реализованы сервисы и механизмы безопасности. Наконец, детально рассмотрены вопросы администрирования средств безопасности для распределенных конфигураций.

Спецификация Internet-сообщества RFC 1510 «Сетевой сервис аутентификации Kerberos (V5)» относится к более частной, но весьма важной и актуальной проблеме - аутентификации в разнородной распределенной среде с поддержкой концепции единого входа в сеть. Сервер аутентификации Kerberos представляет собой доверенную третью сторону, владеющую секретными ключами обслуживаемых субъектов и помогающую им в попарной проверке подлинности. О весомости данной спецификации свидетельствует тот факт, что клиентские компоненты Kerberos присутствуют в большинстве современных операционных систем.

Краткие аннотации подробно рассматриваемых в курсе стандартов и спецификаций «Гармонизированные критерии Европейских стран» стали весьма передовым документом для своего времени, они подготовили появление международного стандарта O’z DST ISO/IEC 15408:2008 «Критерии оценки безопасности информационных технологий» (Evaluation

criteria for IT security), в русскоязычной литературе обычно (но не совсем верно) именуемого «Общими критериями» (ОК).

На сегодняшний день «Общие критерии» - самый полный и современный оценочный стандарт. На самом деле, это метастандарт, определяющий инструменты оценки безопасности ИС и порядок их использования; он не содержит предопределенных классов безопасности. Такие классы можно строить, опираясь на заданные требования.

ОК содержат два основных вида требований безопасности:

- функциональные, соответствующие активному аспекту защиты, предъявляемые к функциям (сервисам) безопасности и реализующим их механизмам;
- требования доверия, соответствующие пассивному аспекту; они предъявляются к технологии и процессу разработки и эксплуатации.

Требования безопасности формулируются, и их выполнение проверяется для определенного объекта оценки - аппаратно-программного продукта или информационной системы.

Подчеркнем, что безопасность в ОК рассматривается не статично, а в соответствии с жизненным циклом объекта оценки. Кроме того, последний предстает в контексте среды безопасности, характеризующейся определенными условиями и угрозами.

«Общие критерии» способствуют формированию двух базовых видов используемых на практике нормативных документов - это профиль защиты и задание по безопасности.

Профиль защиты представляет собой типовой набор требований, которым должны удовлетворять продукты и/или системы определенного класса. Задание по безопасности содержит совокупность требований к конкретной разработке, их выполнение позволит решить поставленные задачи по обеспечению безопасности.

В последующей части курса будут детально рассмотрены как сами «Общие критерии», так и разработанные на их основе профили защиты и проекты профилей.

Криптография - область специфическая, но общее представление о ее месте в архитектуре безопасности и о требованиях к криптографическим компонентам иметь необходимо. Для этого целесообразно ознакомиться с Федеральным стандартом США FIPS 140-2 «Требования безопасности для криптографических модулей» (Security Requirements for Cryptographic Modules). Он выполняет организующую функцию, описывая внешний интерфейс криптографического модуля, общие требования к подобным модулям и их окружению. Наличие такого стандарта упрощает разработку сервисов безопасности и профилей защиты для них.

Криптография как средство реализации сервисов безопасности имеет две стороны: алгоритмическую и интерфейсную. Нас будет интересовать исключительно интерфейсный аспект, поэтому, наряду со стандартом FIPS 140-2, мы рассмотрим предложенную в рамках Internet-сообщества техническую спецификацию «Обобщенный прикладной программный интерфейс службы безопасности» (Generic Security Service Application Program Interface, GSS-API).

Интерфейс безопасности GSS-API предназначен для защиты коммуникаций между компонентами программных систем, построенных в архитектуре клиент/сервер. Он создает условия для взаимной аутентификации общающихся партнеров, контролирует целостность пересылаемых сообщений и служит гарантией их конфиденциальности. Пользователями интерфейса безопасности GSS-API являются коммуникационные протоколы (обычно прикладного уровня) или другие программные системы, самостоятельно выполняющие пересылку данных.

Технические спецификации IPsec [IPsec] имеют, без преувеличения, фундаментальное значение, описывая полный набор средств обеспечения конфиденциальности и целостности на сетевом уровне. Для доминирующего

в настоящее время протокола IP версии 4 они носят факультативный характер; в перспективной версии IPv6 их реализация обязательна. На основе IPsec строятся защитные механизмы протоколов более высокого уровня, вплоть до прикладного, а также законченные средства безопасности, в том числе виртуальные частные сети. Разумеется, IPsec существенным образом опирается на криптографические механизмы и ключевую инфраструктуру.

Точно так же характеризуются и средства безопасности транспортного уровня (Transport Layer Security, TLS). Спецификация TLS развивает и уточняет популярный протокол Secure Socket Layer (SSL), используемый в большом числе программных продуктов самого разного назначения.

В упомянутом выше инфраструктурном плане очень важны рекомендации X.500 «Служба директорий: обзор концепций, моделей и сервисов» (The Directory: Overview of concepts, models and services) и X.509 «Служба директорий: каркасы сертификатов открытых ключей и атрибутов» (The Directory: Public-key and attribute certificate frameworks). В рекомендациях X.509 описан формат сертификатов открытых ключей и атрибутов - базовых элементов инфраструктур открытых ключей и управления привилегиями.

Как известно, обеспечение информационной безопасности - проблема комплексная, требующая согласованного принятия мер на законодательном, административном, процедурном и программно-техническом уровнях. При разработке и реализации базового документа административного уровня - политики безопасности организации - отличным подспорьем может стать рекомендация Internet-сообщества «Руководство по информационной безопасности предприятия» (Site Security Handbook). В нем освещаются практические аспекты формирования политики и процедур безопасности, поясняются основные понятия административного и процедурного уровней, содержится мотивировка рекомендуемых действий, затрагиваются темы анализа рисков, реакции на нарушения ИБ и действий после ликвидации нарушения. Более подробно последние вопросы рассмотрены в

рекомендации «Как реагировать на нарушения информационной безопасности» (Expectations for Computer Security Incident Response). В этом документе можно найти и ссылки на полезные информационные ресурсы, и практические советы процедурного уровня.

При развитии и реорганизации корпоративных информационных систем, несомненно, окажется полезной рекомендация «Как выбирать поставщика Internet-услуг» (Site Security Handbook Addendum for ISPs). В первую очередь ее положений необходимо придерживаться в ходе формирования организационной и архитектурной безопасности, на которой базируются прочие меры процедурного и программно-технического уровней.

Для практического создания и поддержания режима информационной безопасности с помощью регуляторов административного и процедурного уровней пригодится знакомство с британским стандартом BS 7799 «Управление информационной безопасностью. Практические правила» (Code of practice for information security management) и его второй частью BS 7799-2:2002 «Системы управления информационной безопасностью - спецификация с руководством по использованию» (Information security management systems - Specification with guidance for use). В нем разъясняются такие понятия и процедуры, как политика безопасности, общие принципы организации защиты, классификация ресурсов и управление ими, безопасность персонала, физическая безопасность, принципы администрирования систем и сетей, управление доступом, разработка и сопровождение ИС, планирование бесперебойной работы организации.

В настоящее время в нашей стране ведутся большие исследования по этой работе и ниже приведены существующие и действующие несколько стандартов и спецификаций:

1. O’z DSt 1092:2009 - Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи.

2. O’z DSt 1105:2009 - Информационная технология. Криптографическая защита информации. Алгоритм шифрования данных.

3. O’z DSt 1106:2009 - Информационная технология. Криптографическая защита информации. Функция хэширования.

4. O’z DSt 1108:2011 - Информационная технология. Взаимосвязь открытых систем. Структура сертификата открытого ключа ЭЦП и сертификата атрибута.

5. O’z DSt 1135:2007 - Информационная технология. Требования к базам данных и обмену информацией между органами государственного управления и государственной власти на местах.

6. O’z DSt 1204:2009 - Информационная технология. Криптографическая защита информации. Требования безопасности к криптографическим модулям.

7. O’z DSt 1270:2009 - Электронный документооборот. Взаимодействие систем электронного документооборота.

8. O’z DSt 2295:2011 - Электронный документ. Требования к формированию, применению и хранению.

9. O’z DSt 2590:2012 - Информационная технология. Требования к интеграции и взаимодействию информационных систем государственных органов, используемых в рамках формирования. Национальной информационной системы.

10. O’z DSt 2826:2014 - Информационная технология. Криптографическая защита информации. Процессы формирования и поверки электронной цифровой подписи на базе эллиптических кривых.

11. O’z DSt 2875:2014 - Информационная технология. Требования датацентрам. Инфраструктура и обеспечение информационной безопасности.

12. O’z DSt 2927:2015 - Информационные технологии. Информационная безопасность. Термины и определения.

13. O’z DSt ISO 7498-2:2011 (ISO 7498-2:1989, MOD) - Информационная технология. Взаимосвязь открытых систем, базовая эталонная модель. Часть 2. Архитектура безопасности.

14. O’z DSt ISO/IEC 15945:2015 - Информационная технология. Методы обеспечение безопасности. Спецификация служб ДСТ для поддержки применения электронных подписей.

15. O’z DSt ISO/IEC 18045:2013 - Информационная технология. Методы обеспечение безопасности методология оценки безопасности информационных технологии.

16. O’z DSt 1986:2010 - Информационная технология. Информационные системы. Стадии создания.

17. O’z DSt 2863:2014 - Информационная технология. Интерактивные государственные услуги. Классификация и основные требования к формированию.

18. O’z DSt 2814:2014 - Информационная технология. Автоматизированные системы. Классификация по уровню защищенности от несанкционированного доступа к информации.

19. O’z DSt 2815:2014 - Информационная технология. Межсетевые экраны. Классификация по уровню защищенности от несанкционированного доступа к информации.

20. O’z DSt 2816:2014 - Информационная технология. Классификация программного обеспечения средств защиты информации по уровню контроля отсутствия недекларированных возможностей.

21. O’z DSt 2817:2014 - Информационная технология. Средства вычислительной техники. Классификация по уровню защищенности от несанкционированного доступа к информации.

Главной задачей стандартов безопасности является создание основы для взаимодействия между производителями, потребителями и экспертами по квалификации продуктов информационных технологий.

Производителям стандарты необходимы в качестве средства сравнения возможностей их информационных продуктов. Кроме того, стандарты необходимы для процедуры сертификации, которая является механизмом объективной оценки свойств информационных продуктов.

Потребители заинтересованы в методике позволяющей обоснованно выбрать информационный продукт, отвечающий их потребностям. Для этого им необходима шкала оценки безопасности.

Эксперты по квалификации продуктов информационных технологий рассматривают стандарты как, инструмент, позволяющий им оценить уровень безопасности, обеспечиваемый продуктами информационных технологий.

Контрольные вопросы

1. Основные понятия стандарта и спецификации.
2. Техническая спецификация X.800.
3. Перечислите виды требований безопасности в общих критериях.
4. Стандарты, действующие в Республике Узбекистан, в области обеспечения информационной безопасности.

Список литератур

1. Борисов М.А., Романов О.А. Основы организационно-правовой защиты информации. Изд.4-е-М: Ленанд, 2015г.
2. Тармоқ стандарти. Алоқа ва ахборотлаштириш соҳасида ахборот хавфсизлиги. Атамалар ва таърифлар.TSt 45-010:2010.
3. Шаньгин В.Ф. Информационная безопасность. М:ДМК Пресс, 2014г.
4. Платонов В.В. Программно-аппаратные средства защиты информации: учебник для студ. Учреждений выс.образования/.-М.: Издательский центр «Академия», 2014г.
5. Мельников Д.А. Информационная безопасность открытых систем: учебник/ -М.:Флинта: Наука, 2013г.
6. Stamp, Mark. Information security: principles and practice/ Mark Stamp/ - 2nd ed. ISBN 978-0-4-470-62639-9(hardback)/ QA76.9.A25S69, USA, 2011.
7. Зрюмов Е.А., Зрюмова А.Г. Базы данных для инженеров: Учебное пособие. - Барнаул: Изд-во АлтГТУ, 2010. – 131 с.
8. Н. А. Гайдамакин. Автоматизированные информационные системы, базы и банки данных. Вводный курс. Гелиос АРВ. 2002. – 368 с.
9. В.Г. Олифер, Н.А. Олифер. Компьютерные сети. Приниципы, технологии, протоколы 4 издание –Питер. 2010. 944с.
10. Joel Scambray, Vincent Liu. Hacking exposed. Web Applications 3. Caleb Sima. 2010y.
11. P.Y.A. Ryan, S.A. Schneider, M.H. Goldsmith, G. Lowe and A.W. Roscoe. The Modelling and Analysis of Security Protocols: the CSP Approach. First published 2000. The original version is in print December 2010 with Pearson Education.
12. Н. А. Гайдамакин. Разграничение доступа к информации в компьютерных системах. Монография. Издательство Уральского университета. 2003. – 328 с.

13. Michael Lee, Gentry Bieker. MASTERING Microsoft SQL Server 2008. Wiley Publishing, Inc. 2009, 723 p.
14. Гайдамакин Н.А. Теоретические основы компьютерной безопасности, Учебное пособие, Екатеринбург – 2008. 212 с.
15. Ғаниев С. К., Каримов М. М., Ташев К. А. Ахборот хавфсизлиги. Ахборот-коммуникацион тизимлар хавфсизлиги. Олий ўқув юрт талабалари учун мўлжалланган. "Алоқачи", 2008.
16. Аткинсон, Леон. MySQL библиотека профессионала. -М.: Издательство «Вильяме», 2002.-624 стр.
17. Ю.В. Романец, П.А. Тимофеев, В.Ф. Шаньгин. Защита информации в компьютерных системах и сетях: 2-е изд., перераб. и доп. — М.: Радио и связь, 2001.
18. С. Д. Кузнецов. Основы баз данных, 2-е издание. Бином. Лаборатория знаний, Интернет-университет информационных технологий. 2007 г.
19. А. В. Кузин, С. В. Левонисова. Базы данных. Издательство: «Академия». 2012 г.
20. Itzik Ben-Gan. Training Kit (Exam 70-461): Querying Microsoft SQL Server 2. Издательство: «Microsoft Press». 2012 г.
21. Карпова Т. С. Базы данных: модели, разработка, реализация. Национальный Открытый Университет «ИНТУИТ». 2016 год. 241 стр.
22. Бессарабов Н. В. Модели и смыслы данных в Cache и Oracle. Национальный Открытый Университет «ИНТУИТ». 2016 год. 617 стр.
23. Фейерштейн С., Прибыл Б. Oracle PL/SQL. Для профессионалов. 6-е изд. Издательство: Питер. 2015 г. 1024 стр.
24. Alfred Basta, Melissa Zgola. Database Security. Paperback – 2014.
25. Ron Ben Natan. Implementing Database Security and Auditing. Digital Press; 1 edition (May 2, 2005). 432 pages.
26. Maria Grazia Fugini, Silvana Castano, Giancarlo Martella. Database Security (Acm Press Books) 1st Edition. Pearson Education Ltd; 1st edition. 456 pages.

27. Josh Shaul, Aaron Ingram. Practical Oracle Security: Your Unauthorized Guide to Relational Database Security 1st Edition. Syngress; 1 edition (November 26, 2007). 288 pages.
28. С.А. Нестеров Название: Базы данных. Издательство: СПбГПУ. Год: 2013.
29. В.П. Агальцов. Распределенные и удаленные базы данных: Учебник. - М.: ид. форум, НИЦ инфра-м, 2013.
30. Крис Фиайли. SQL. Руководство по изучению языка. Серия: Quick Start Издательство: ДМК Пресс Год: 2013 Страниц: 456.

Список сокращенных слов

БД – база данных.

СУБД – система упарвления базамы данных.

SQL – (structured query language) язык структурированных запросов.

TCB – (Trusted Computing Base) доверенная вычислительная база.

NRU – (no read up - NRU) запрещается читать в верх.

NWD – (no write down - NWD) запрещается писать в вниз.

ANSI – Национальный институт стандартизации Америки .

ISO – Международная организация стандартизации.

DDL – язык описания данных DDL (Data Definition Language).

DML – язык манипуляции данных DML (Data Manipulation Language).

DAC – модель основанная на дискреционном управлении доступом (Discretionary Acces Control).

MAC – модель основанная на мандатном разграничении доступа (Mandatory Access Control).

RBAC – ролевая модель разграничения доступа (Role – Based Access Control).

FS – модель файлового сервера (File Server).

RDA – модель дистанционного доступа к данным (Remote Data Access).

DBS – модель сервера базы данных (Data Base Server).

AS – модель сервера приложений (Application Server).

ODBC – Open Database Connectivity – открытое использование базы данных.

Authorization ID – специальный идентификационное название или номер каждого пользователя.

SAN – Storage Area Network – сеть хранения данных.

DCE – распределенная вычислительная среда концерна OSF.

TLS – Transport Layer Security – безопасность транспортного уровня.

SSL – Security Socket Layer – безопасный уровень сокета.

Глоссарий

Автоматизированная информационная система - совокупность программных и аппаратных средств, предназначенных для создания, передачи, обработки, распространения, хранения и/или управления данными и информацией и производства вычислений.

Автоматлаштирилган ахборот тизими – маълумотларни ва ахборотни яратиш, узатиш, ишлаш, тарқатиш, сақлаш ва/ёки бошқаришга ва ҳисоблашларни амалга оширишга мўлжалланган дастурий ва аппарат воситалар.

Automated Information System - a set of software and hardware designed for the creation, transmission, processing, distribution, storage and/or data and information management and production calculations.

Авторизация- представление пользователю определенных прав доступа на основе положительного результата его аутентификации в системе.

Авторизация –тизимда фойдаланувчига, унинг ижобий аутенфикациясига асосан, маълум фойдаланиш хукукларини такдим этиш.

Authorization -View user specific access rights on the basis of a positive result in its authentication system.

Авторизация данных - определение и установление степени приватности данных в базе данных.

Маълумотлар авторизацияси– маълумотларнинг маълумотлар базасига тегишли даражасини аниқлаш ва белгилаш.

Data authorization- identification and degree of privacy data in the database.

Авторское право - совокупность правовых норм (раздел гражданского права), которые регулируют отношения, возникающие в связи с созданием и использованием произведений науки, литературы и искусства.

Муаллифлик ҳуқуқи – фан, адабиёт ва санъат асарларини яратиш, фойдаланиш ва ҳуқуқий ҳимоялашда вужудга келадиган муносабатларни тартибга солувчи ҳуқуқий нормалар мажмуи.

Copyright - the body of law (Civil Law Section), which regulate the relations arising in connection with the creation and use of scientific, literary and artistic works (copyright.

Администратор базы данных - специальное должностное лицо (группа лиц), имеющий(ие) полное представление о базе данных и отвечающее за ее ведение, использование и развитие. Входит в состав администрации банка данных.

Маълумотлар базаси маъмури – маълумотлар базаси хусусида тўлиқ тасаввурга эга ва ундан фойдаланиш ва ривожлантириш учун жавобгар махсус лавозимли шахс (шахслар гурухи). Маълумотлар банки маъмурияти таркибига киради.

Database Administrator - special officer (group of persons) having (s) a complete picture of the database and is responsible for its maintenance, use and development. Included in the administration of the bank data.

Администратор доступа - одно из должностных лиц в составе администрации банка данных, отвечающее за организацию доступа пользователей к базам данных.

Фойдаланиш маъмури – маълумотлар базасидан фойдаланишни ташкил этишга жавобгар, маълумотлар банки маъмурияти таркибидаги лавозимли шахслардан бири.

Administrator access - one of the officials in the administration part of the data bank, the organization responsible for user access to databases.

Администратор защиты- **субъект доступа, ответственный за защиту автоматизированной системы от несанкционированного доступа к информации.**

Ҳимоя маъмури – автоматлаштирилган тизимни ахборотдан рухсатсиз фойдаланишдан ҳимоялашга жавобгар фойдаланиш субъекти.

Security administrator-access entity responsible for the protection of the automated system from unauthorized access to information.

Администратор системы - лицо, отвечающее за эксплуатацию системы и поддержание ее в работоспособном состоянии.

Тизим маъмури – тизимни эксплуатациясига ва унинг ишга лаёкатлигини таъминлашга жавобгар шахс.

The system administrator - the person responsible for the operation of the system and maintaining it in working condition.

Администратор службы безопасности -человек (или группа людей), имеющий(ие) полное представление об одной или нескольких системах обеспечения безопасности и контролирующий(ие) проектирование и их использование.

Хавфсизлик хизмати маъмури– хавфсизликни таъминлашнинг бир ёки бир неча тизими ҳамда лойиҳалашни назоратлаш ва улардан фойдаланиш хусусида тўлиқ тасаввурга эга шахс (ёки шахслар гурухи).

Administrator security-person (or group of people) having (s) complete understanding of one or more security systems and controls (s) design and use.

Администрация банка данных - группа лиц (подразделение), отвечающих за эксплуатацию банка данных: ведение баз данных, организацию коллективного доступа к ним пользователей и развитие системы.

Маълумотлар банки маъмурияти – маълумотлар банкининг эксплуатациясига жавобгар шахслар гурухи (бўлинма): маълумотлар

базасини юритиш, ундан ташқари коллектив фойдаланишни ташкил этиш ва тизимни ривожлантириш.

Administration Data Bank - a group of persons (unit), responsible for the operation of a data bank: maintenance of databases, the collective access to users and system development.

Администрирование базы данных - выполнение функций определения, организации, управления и защиты данных в базе

Маълумотлар базасини маъмурлаш – базадаги маълумотларни аниқлаш, ташкил этиш, бошқариш ва ҳимоялаш вазифаларини бажариш.

Database Administration - acting as determining the organization, management and protection of data in the database.

Алгоритм - упорядоченный конечный набор четко определенных правил для решения задач за конечное количество шагов.

Алгоритм – амалларнинг чекланган сони ёрдамида масала ечимини белгиловчи буйруқларнинг чекланган тўплами.

Algorithm - an ordered finite set of clearly defined rules for solving a finite number of steps.

Аппаратные средства защиты - механические, электромеханические, электронные, оптические, лазерные, радио, радиотехнические, радиолокационные и другие устройства, системы и сооружения, предназначенные для защиты информации от несанкционированного доступа, копирования, кражи или модификации.

Ҳимоянинг аппарат воситалари – ахборотни рухсатсиз фойдаланишдан модификацияланишидан, нусхалашдан, ўғирланишидан ҳимоялашга мўлжалланган механик, электромеханик, электрон, оптик, лазер, радио, радиотехник, радиолокацион ва бошқа қурилмалар, тизимлар ва иншоатлар.

Hardware protection - mechanical, electromechanical, electronic, optical, laser, radio, radio, radar and other devices, systems and structures designed to protect the information from unauthorized access, copying, modification or theft.

Аппаратура технической разведки - совокупность технических устройств обнаружения, приема, регистрации, измерения и анализа, предназначенная для получения разведывательной информации.

Техник разведка аппаратураси – разведка ахборотини олишга мўлжалланган аниқлаш, қабул қилиш, қайдлаш, ўлчаш ва таҳлиллаш техник қурилмалари мажмуи.

Equipment and technical intelligence - a set of technical detection devices, receiving, recording, measurement and analysis, designed for intelligence.

Атака «противник в середине» - атака на протокол криптографический, в которой противник С выполняет этот протокол как с участником А, так и с участником В. Противник С выполняет сеанс с участником А от имени В, а с участником В от имени А. В процессе выполнения противник пересылает сообщения от А к В и обратно, возможно, подменяя их. В частности, в случае протокола аутентификации абонента успешное осуществление атаки «противник в середине» позволяет противнику аутентифицировать себя для В под именем А. Для осуществления атаки «противник в середине» необходимо обеспечивать синхронизацию двух сеансов протокола.

«Душман ўртада» хужуми – крипттографик протоколига хужум булиб, бунда душман С ушбу протоколни иштирокчи А ва иштирокчи В билан бажаради. Душман С иштирокчи А билан сеансни иштирокчи В номидан, иштирокчи В билан эса иштирокчи А номидан бажаради. Бажариш жараёнида душман иштирокчи А дан иштирокчи В га ва аксинча хабарни узгартириб узатади. Хусусан, абонентни аутентификациялаш протоколи холида «душман ўртада» хужумининг мувафакиятли амалга оширилишини душманга иштирокчи В учун ўзини иштирокчи А номидан

аутентификациялашга имкон беради. «Душман ўртада» хужумини амалга ошириш учун протоколнинг иккита сеансининг синхронланишини таьминлаш лозим.

Man-in-the-middle attack - attack on a cryptographic protocol in which the enemy With this protocol performs as a party A and party B. with C. Enemy performs session with party A on behalf of B, and a participant on behalf of A. During Runtime opponent forwards messages from A to B and back, possibly replacing them attacks. In particular, in the case of an authentication protocol is connected to the success of the attack "in the middle of the enemy" allows authenticate itself to the enemy in the name of A. To carry out the attack "in the middle of the enemy" is necessary to ensure the synchronization of the two sessions of the protocol.

Атака на отказ в обслуживании - атака с целью вызвать отказ системы, то есть создать такие условия, при которых легитимные пользователи не смогут получить доступ к предоставляемым системой ресурсам, либо этот доступ будет значительно затруднён.

Хизмат килишдан воз кечишга ундайдиган хужум – тизим бузилишига сабаб булувчи хужум, яьни шундай шароитлар тугдирадики, конуний фойдаланувчи тизим такдим этган ресурслардан фойдалана олмайди ёки фойдаланиши анчагина қийинлашади.

Denial-of-Service attack (DoS attack)- attack to cause failure of the system, that is to create the conditions under which legitimate users cannot get access to the resources provided by the system, or that access will be significantly hampered.

Атака со словарем паролей — атака на криптосистему, основанная на переборе значений пароля.

Пароллар луғатига асосланган хужум – пароль қийматларини саралашга асосланган хужум.

Attack with a dictionary of passwords - an attack on a cryptosystem based on iterating values password.

Атака со словарем — атака на криптосистему, использующая словарь элементов текста, открытого.

Луғатга асосланган хужум – криптотизимга очиқ матн элементлари луғатидан фойдаланишга асосланган хужум.

With a dictionary Attack - attack on the cryptosystem that uses a dictionary of text elements open.

Аттестация объекта защиты - официальное подтверждение органом по сертификации или другим специально уполномоченным органом наличия на объекте защиты необходимых и достаточных условий, обеспечивающих выполнение установленных требований и норм эффективности защиты информации.

Химоя обьектининг аттестацияси – химоя обьектида ахборотни химоялашда белгиланган талаблар ва самарадорлик меьёрларининг бажарилишини таьминловчи зарурий ва етарли шароитлар борлиги хусусида сертификация берувчи органнинг ёки бошка махсус ваколатли органнинг расмий тасдиғи.

Attestation object protection - official confirmation by the certification body or other specially authorized presence in the facility protection necessary and sufficient conditions for fulfilling specified requirements and performance standards of information security.

Аудит (безопасности) — ведение контроля защищенности путем регистрации (фиксации в файле аудита) заранее определенного множества событий, характеризующих потенциально опасные действия в системе компьютерной, влияющие на ее безопасность

Хавфсизлик аудити – компьютер тизими хавфсизлигига таъсир этувчи бўлиши мумкин бўлган хавфли ҳаракатларни характерловчи, олдиндан аниқланган ходисалар тўпламини рўйхатга олиш(аудит файлида қайдлаш) йули билан ҳимояланишни назоратлаш.

Security audit – maintain security control by registering (fixation in the audit file) a predetermined set of events that characterize the potentially dangerous actions in the computer affecting its safety.

Аутентификатор - средство аутентификации, представляющее отличительный признак пользователя. Средствами аутентификации пользователя могут быть дополнительные кодовые слова, биометрические данные и другие отличительные признаки пользователя.

Аутентификатор – фойдаланувчининг фарқли аломатини ифодаловчи аутентификация воситаси. Қўшимча код сўзлари, биометрик маълумотлар ва фойдаланувчининг бошқа фарқли аломатлари аутенфикация воситалари бўлиши мумкин.

Authenticator - authentication means representing the hallmark of the user. Means of user.

Аутентификационные данные — информация, используемая для верификации предъявленного идентификатора пользователя.

Аутентификация маълумотлари - тақдим етилган фойдаланувчи идентификаторини тасдиқлаш учун ишлатиладиган ахборот.

Authentication data - information used to verify the presented user ID.

Аутентификация - проверка идентификации пользователя (проверка подлинности), устройства или другого компонента в системе, обычно для принятия решения о разрешении доступа к ресурсам системы; проверка целостности хранящихся или передающихся данных для обнаружения их несанкционированной модификации.

Аутентификация – одатда тизим ресурсларидан фойдаланишга рухсат этиш хусусида қарор қабул қилиш учун фойдаланувчининг (хақиқийлигини), қурилманинг ёки тизимнинг бошқа ташкил этувчисининг идентификациясини текшириш; сақланувчи ва узатилувчи маълумотларнинг рухсатсиз модификацияланганлигини аниқлаш учун текшириш.

Authentication - checking user authentication (authentication), device, or other component in the system, usually to make a decision about granting access to system resources; checking the integrity of stored or transmitted data to detect unauthorized modification.

Аутентификация биометрическая — способ аутентификации абонента (пользователя), основанный на проверке его биометрических характеристик (отпечатков пальцев, геометрии руки, лица, голоса, рисунка сетчатки глаза и т. п.). К преимуществам данного метода относится неотделимость биометрических характеристик от пользователя: их нельзя забыть, потерять или передать другому пользователю.

Биометрик аутентификация – абонентни (фойдаланувчини) унинг биометрик характеристикаси (бармок излари, панжа геометрияси, юзи, овози, кўз пардасининг тўри ва х.) асосидаги аутентификациялаш усули. Ушбу усулнинг афзаллиги – биометрик характеристикаларни фойдаланувчидан ажратиб бўлмаслиги. Уларни эсдан чиқаришнинг, йуқотишнинг ёки бошқа фойдаланувчига беришнинг иложи йуқ.

Biometric Authentication - authentication method subscriber (user), based on its verification of biometrics (fingerprints, hand geometry, face, voice, retina pattern, etc.). The advantages of this method is the inseparability of the biometric characteristics of the user: they cannot be forgotten, lost or transferred to another user.

Аутентификация данных (цифровая подпись) - процесс подтверждения подлинности (отсутствия фальсификации или искажения) произвольных

данных, предъявленных в электронной форме. Данные могут представлять собой: сообщения, файл, элемент базы данных (программы), идентификатор (аутентификатор) пользователя, адрес сетевого абонента и т.п.

Маълумотлар аутентификацияси (рақамли имзо) – электрон шаклда тақдим этилган ихтиёрий маьлумотларнинг ҳақиқийлигини (сохталаштиришнинг ёки бузилишнинг йўқлигини) тасдиқлаш жараёни. Маьлумотлар, хабарлар, файл, маьлумотлар базаси (дастур) элементлари, фойдаланувчининг идентификатори (аутентификатори), тармоқ абоненти адреси ва х. кўринишида бўлиши мумкин.

Authentication data (digital signature) - authentication process (lack of falsification or distortion) of arbitrary data, presented in electronic form. These may be: the message file, the database item (program), an identifier (an authenticator) the user is connected to a network address, etc.

Аутентификация двухфакторная — аутентификация пользователей на основе двух разнородных факторов, как правило, на основе того, что знает пользователь, и того, чем он владеет (например, на основе пароля и физического идентификатора).

Икки факторли аутентификация – фойдаланувчиларни иккита турли факторлар асосида аутентификациялаш, одатда, фойдаланувчи биладиган нарса ва эгалик қиладиган нарса (масалан, пароль ва физик идентификатори) асосида.

Two-factor authentication- user authentication based on two different factors are usually based on what the user knows, and what he owns (ex. password-based and physical identifier).

Аутентификация источника данных - подтверждение подлинности источника полученных данных.

Маълумотлар манбаининг аутентификацияси– олинган маълумотлар манбаининг ҳақиқийлигини тасдиқлаш.

Data origin authentication - confirmation of the authenticity of the source of the data obtained.

Аутентификация многофакторная — реализация контроля доступа, представляющая собой идентификацию пользователя на основе нескольких независимых факторов. Представляет гибкий подход, позволяющий организации реализовать устойчивую систему управления доступом на основе использования различных методов аутентификации. Технологии а.м. включают: одноразовые пароли, аутентификацию на основе сертификатов, аутентификацию на основе контекста и др.

Кўп факторли аутентификация - бир неча мустақил факторлар асосида фойдаланиш назоратини амалга ошириш. Аутентификациянинг турли усуллари асосида фойдаланишни бошқаришнинг барқарор тизимини амалга оширишни ташкил этиш имкониятига эга мосланувчан ёндашиш ҳисобланади. Кўп факторли аутентификация технологияси таркибига бир мартали пароллар, сертификатлар асосидаги аутентификация, контекст асосидаги аутентификация ва х. киради.

Multifactor Authentication - implementing access control, which is a user identification based on several independent factors. A flexible approach allows organizations to implement robust access control system based on the use of different authentication methods.

Аутентификация односторонняя — аутентификация сторон, при которой одна из сторон проверяет, что взаимодействующая с ней сторона - именно та, за которую себя выдает. А. о. реализуется протоколом идентификации с двумя участниками: доказывающим и проверяющим.

Бир тарафлама аутентификация – тарафларнинг аутентификацияси бўлиб, тарафларнинг бири у билан ўзаро ҳаракатдаги тарафнинг ҳақиқатан ҳам ўзи эканлигини текширади. Бир тарафлама аутентификация иккита иштирокчи:

исботловчи ва текширувчи билан аутентификациялаш протоколи орқали амалга оширилади.

O**ne-way authentication –** authentication of the parties, in which one of the parties to check that it interacts with the side - namely that for which he is. Identification protocol is implemented with two participants: proving and inspection.

Аутентификация пользователя - подтверждение подлинности пользователя с помощью предъявляемого им аутентификатора. Еще - проверка соответствия пользователя предъявляемому им идентификатору.

Фойдаланувчининг аутентификацияси – фойдаланувчи тақдим этган идентификатори ёрдамида унинг ҳақиқийлигини тасдиқлаш. Яна – фойдаланувчининг у тақдим этган идентификаторга мослигини таққослаш.

User Authentication - User authentication using against them authenticator. Also, to check compliance against them user ID.

Аутентичность — 1. Подлинность. 2. Свойство гарантирующее, что субъект или ресурс идентичны заявленным. Аутентичность применяется к таким субъектам, как пользователи, процессы, системы и информация.

Аслига тўғрилик - 1. Хақиқийлик 2. Субъект ёки ресурснинг сўралганига мувофиқлигини кафолатланувчи хусусият.

Authenticity - 1. Authenticity. 2. Feature ensures that the subject or resource identical stated. Authenticity applies to entities such as people, processes, systems and information.

База данных - совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимо от прикладных программ. Является информационной моделью предметной области. БД, как правило,

представляются тремя уровнями абстракции: внешним, концептуальным и внутренним.

Маълумотлар базаси - татбиқий дастурларга боғлиқ бўлмаган холда маълумотларни тавсифлаш, сақлаш ва манипуляциялашнинг умумий принципларини кўзда тутувчи, маълум қоидалар бўйича ташкил этилган маълумотлар мажмуи. Предмет сохасининг информацион модели ҳисобланади. Маълумотлар базаси одатда абстракциянинг ташқи, концептуал ва ички сатхлари орқали ифодаланади.

Database - a set of data organized according to certain rules, general principles providing descriptions, storing and manipulating data, regardless of the application. Information is the domain model. Database, usually represented by three levels of abstraction: external, conceptual and internal.

Банк данных - автоматизированная информационная система централизованного хранения и коллективного использования данных. В состав банка данных входят одна или несколько баз данных, справочник баз данных, система управления базами данных, а также библиотеки запросов и прикладных программ. еще - система, предоставляющая услуги по хранению и поиску данных определенной группе пользователей по определенной тематике.

Маълумотлар банки- маълумотларни марказлашган сақлаш ва коллектив фойдаланишнинг автоматлаштирилган информацион тизими. Банк таркибига бир ёки бир неча базаси, маълумотлар базасининг бошқариш тизими хамда сўровлар ва татбиқий дастурлар библиотекаси киради. Яна фойдланувчиларнинг маълум гурухига, маълум тематика бўйича маълумотларни сақлаш ва қидириш хизматларини тақдим этиш.

Databank - automated information system for centralized storage and sharing of data. The structure of the data bank includes one or more databases, reference databases, database management system, as well as libraries of queries and

applications. More - a system that provides services for data storage and retrieval specific group of users on a particular topic.

Безопасная операционная система - операционная система, эффективно управляющая аппаратными и программными средствами с целью обеспечения уровня защиты, соответствующего содержанию данных и ресурсов, контролируемых этой системой.

Хавфсиз операцион тизим – маълумотлар ва ресурслар мазмунига мос ҳимоялаш даражасини таъминлаш мақсадида аппарат ва дастурий воситаларни самарали бошқарувчи операцион тизим.

Secure operating system - an operating system that effectively manages the hardware and software to provide the level of protection corresponding to the content data and resources controlled by the system.

Безопасность - свойство системы противостоять внешним или внутренним дестабилизирующим факторам, следствием воздействия которых могут быть нежелательные ее состояния или поведение. еще - состояние, в котором файлы данных и программы не могут быть использованы, просмотрены и модифицированы неавторизованными лицами (включая персонал системы), компьютерами или программами.

Хавфсизлик - таъсири натижасида номақбул холатларга олиб келувчи атайин ёки тасодифан, ички ва ташқи беқарорловчи факторларга қарши тизимнинг тура олиш хусусияти. Яна маълумотлар файлларининг ва дастурларнинг ишлатилиши, кўриб чиқилиши ва авторизацияланмаган шахслар (жумладан тизим ходими), компъютерлар ёки дастурлар томонидан модификацияланиши мумкин бўлмаган холат.

Security - property system to withstand external or internal factors destabilizing effect of which may be undesirable its state or behavior. Also, a state in which data files and programs may not be used, viewed and modified by unauthorized persons (including staff system) computers or programs.

Безопасность данных - защита данных от несанкционированной (случайной или намеренной) модификации, разрушения или раскрытия. еще - свойство компьютерной системы противостоять попыткам несанкционированного доступа к обрабатываемой и хранимой информации.

Маълумотлар хавфсизлиги – маълумотларни рухсатсиз (атайин ёки тасодифан) модификацияланишидан, бузилишидан, фош этилишидан ҳимоялаш. Яна - компъютер тизимининг ишланадиган ва сақланадиган ахборотдан рухсатсиз фойдаланишга қарши тура олиши хусусияти.

Data security - protection of data against unauthorized (accidental or intentional) modification, destruction or disclosure. Also, property of a computer system to resist the attempts of unauthorized access to information stored and processed.

Безопасность данных персональных — достигается путем исключения доступа несанкционированного, в том числе случайного, к данным персональным, результатом которого может стать уничтожение, изменение, блокирование, копирование, распространение данных персональных, а также иных несанкционированных действий.

Шахсий маълумотлар хавфсизлиги - бунга натижаси шахсий маълумотларни йўқ қилиш, ўзгартириш, блокировка қилиш, нусхалаш, тарқатиш бўлиши мумкин бўлган рухсатсиз, хусусан тасодифий, фойдаланишни ҳамда бошқа рухсатсиз харакатларни истисно қилиш йўли билан эришилади.

Personal Data Security - is achieved by eliminating unauthorized access, including random, personal data, which may result in destruction, alteration, blocking, copying, distribution of personal data, as well as other illegal actions.

Безопасность информации - состояние информации, при котором исключаются случайные или преднамеренные несанкционированные воздействия на информацию или несанкционированное ее получение, еще -

состояние уровня защищенности информации при ее обработке техническими средствами, обеспечивающее сохранение таких ее качественных характеристик (свойств) как секретность /конфиденциальность/, целостность и доступность.

Ахборот хавфсизлиги - ахборот ҳолати бўлиб, унга биноан ахборотга тасодифан ёки атайин рухсатсиз таъсир этишга ёки унинг олинишига йўл қўйилмайди. Яна - ахборотни техник воситалар ёрдамида ишланишида унинг махфийлик (конфиденциаллик), яхлитлик ва фойдаланувчанлик каби характеристикаларини (хусусиятларини) сақланишини таъминловчи ахборотнинг ҳимояланиш сатҳи холати.

Information security - state information , which prevents accidental or intentional tampering or unauthorized information to receive it, also - state -level data protection during processing technologies to support the preservation of its qualitative characteristics (properties) as privacy / confidentiality / integrity and availability.

Безопасность информационная - способность системы противостоять случайным или преднамеренным, внутренним или внешним информационным воздействиям, следствием которых могут быть ее нежелательное состояние или поведение.

Ахборот хавфсизлиги – таъсири натижасида номақбул ҳолатларга олиб келувчи атайин ёки тасодифан, ички ва ташқи информацион таъсирларга қарши тизимнинг тура олиш хусусияти.

Safety information - the system's ability to resist accidental or intentional, internal or external information influences, that could result in an undesirable state or her behavior.

Безопасность информационной системы - свойство информационной системы противостоять попыткам несанкционированного доступа. Совокупность элементов, необходимых для обеспечения адекватной защиты

компьютерной системы; включает аппаратные и/или программные функции, характеристики и средства; операционные и учетные процедуры, средства управления доступом на центральном компьютере, удаленных компьютерах и телекоммуникационных средствах; административные мероприятия, физические конструкции и устройства; управление персоналом и коммуникациями.

Ахборот тизими хавфсизлиги - ахборот тизимининг рухсатсиз фойдаланишига уринишга қарши тура олиши хусусияти. Компъютер тизимининг адекват ҳимояланишини таъминлашга зарурий элементлар мажмуи: аппарат ва/ёки дастурий функциялар, характеристикалар ва воситалар, амалий ва қайдлаш муолажалари; марказий компъютердан, масофадаги компъютерлардан ва телекоммуникация воситаларидан фойдаланишни бошқариш воситалари; маъмурий тадбирлар, физик конструкциялар ва қурилмалар; ходимларни ва коммуникацияларни бошқариш.

Information system security – property information system to resist attempts of unauthorized access. Assembly of components necessary to ensure adequate protection of the computer system; comprises hardware and / or software functions, characteristics, and means; operating and accounting procedures, controls access to the central computer, remote computers and telecommunication facilities; administrative measures, physical structures and devices; personnel management and communications.

Безопасность информационно - коммуникационных технологий (безопасность ИТТ) — все аспекты, связанные с определением, достижением и поддержанием конфиденциальности, целостности, доступности, неотказуемости, подотчетности, аутентичности и достоверности информационно-телекоммуникационных технологий.

Ахборот - коммуникация технологиялар хавфсизлиги (АТТ хавсизлиги) – ахборот - телекоммуникация технологияларининг конфиденциаллигини, яхлитлигини, фойдаланувчанлигини, бош тортмаслигини, ҳисдорлигини, аслига тўғрилигини ва ишончлигини аниқлаш, уларга эришиш ва уларни мададлаш билан боғлиқ барча жихатлар.

ICT security – communication technology (ICT security) - All aspects related to the definition, achieving and maintaining confidentiality, integrity, availability, non-repudiation, accountability, authenticity and reliability of information and telecommunication technologies.

Безопасность компьютерных систем – свойство компьютерных систем противостоять попыткам несанкционированного доступа к обрабатываемой и хранимой информации, вводу информации, приводящей к деструктивным действиям, и навязыванию ложной информации.

Компьютер тизим хавфсизлиги – компъютер тизимининг деструктив ҳаракатларга ва ёлғон ахборотни зўрлаб қабул қилинишига олиб келувчи ишланувчи ва сақланувчи ахбортдан рухсатсиз фойдаланишга уринишларга қарши тура олиш ҳусусияти.

Security of computer systems - property computer systems to resist attempts of unauthorized access to information processed and stored, the input of information, leading to destructive actions, and the imposition of false information.

Безопасность сетевая — меры, предохраняющие сеть информационную от доступа несанкционированного, случайного или преднамеренного вмешательства в нормальные действия или попыток разрушения ее компонентов. Включает защиту оборудования, программного обеспечения, данных.

Тармоқ хавфсизлиги - ахборот тармоғини рухсатсиз фойдаланишдан, меъёрий ишлашига тасодифан ёки атайин аралашишдан ёки тармоқ компонентларини бузишга уринишдан эхтиёт қилувчи чоралар. Асбоб-

ускуналарни, дастурий - таъминотни, маълумотларни ҳимоялашни ўз ичига олади.

Network Security - measures that protect the network information from unauthorized access, accidental or intentional interference with normal activities or attempts to destroy its components. Includes the protection of hardware, software, data.

Безопасность системы — защищенность системы от несанкционированного использования ее ресурсов и функциональных возможностей, а также от возможных нарушений ее функционирования, вызванных различными предсказуемыми и непредсказуемыми обстоятельствами.

Тизим хавфсизлиги - тизимнинг унинг ресурсларидан ва функциональ имкониятларидан рухсатсиз фойдаланишдан, ҳамда тизим ишлашида турли башорат қилинадиган ёки қилинмайдиган холатлар сабаб бўлувчи бўлиши мумкин бўлган бузилишлардан ҳимояланиши.

System Security - the security of the system from unauthorized use of its resources and capabilities, as well as possible violations of its functioning caused by various predictable and unpredictable circumstances.

Безотказность - способность системы выполнять возложенные на нее функции в требуемый момент времени в задаваемых условиях.

Бузилмаслик – тизимнинг унга юклатилган вазифаларни берилган шароитда исталган вақт онида бажариш қобилияти.

Reliability - the ability of the system to fulfill its function in the desired time in the given conditions.

Брандмауэр - метод защиты сети от угроз безопасности, исходящих от других систем и сетей, с помощью централизации доступа к сети и контроля за ним аппаратно-программными средствами. еще - является защитным барьером, состоящим из нескольких компонентов (например,

маршрутизатора или шлюза, на котором работает программное обеспечение брандмауэра).

Брандмауэр – аппарат-дастурий воситалар ёрдамида тармоқдан фойдаланишни марказлаштириш ва уни назоратлаш йўли билан тармоқни бошқа тизимлардан ва тармоқлардан келадиган хавфсизликка тахдидлардан ҳимоялаш усули. Яна бир неча компонентлардан (масалан, брандмауэр дастурий таъминоти ишлайдиган маршрутизатор ёки шлюздан) ташкил топган ҳимоя тўсиғи ҳисобланади.

Firewall - a method of protecting the network from security threats from other systems and networks by centralizing network access and control of hardware and software. Also, is a protective barrier, consisting of several components (such as a router or gateway that is running firewall software).

Ведение базы данных - деятельность, направленная на обновление и восстановление базы данных, а также на перестройку ее структуры.

Маълумотлар базасини юритиш – маълумотлар базасини янгилаш ва тиклаш ҳамда унинг структурасини қайта қуришга йўналтирилган фаолият.

Maintaining a database - activities aimed at updating and restoring the database, as well as the restructuring of its structure.

Верификация - процесс сравнения двух уровней спецификации средств вычислительной техники или их комплексов на надлежащее соответствие. Еще - в программировании доказательство правильности программ. Различают два подхода к верификации: статические и конструктивные методы.

Верификация – ҳисоблаш воситалари ёки уларнинг комплекси спецификациясининг икки сатҳини тегишли мосликка таққослаш жараёни. Яна - дастурлашда – дастур тўғрилигининг тасдиғи. Верификацияга иккита ёндашиш фарқланади: статик ва конструктив усуллар.

Verification - the process of comparing two levels of specification of computer equipment or systems for proper alignment. Also - programming proof of the correctness of programs. There are two approaches to verification: static and constructive methods.

Взламывание пароля — техника (способ) тайно получать доступ к системе (сети) информационной, в которой нападающая сторона с помощью вскрывателя паролей пробует угадать (подобрать) или украсть пароли.

Паролни бузиб очиш - ахборот тизимидан (тармоғидан) яширинча фойдаланиш техникаси (усули) бўлиб, ҳужум қилувчи тараф паролларни фош қилувчи ёрдамида паролларни аниқлашга (танлашга) ёки ўғирлашга уриниб кўради.

Cracking password - tech (method) secretly to access the system (network) information, in which the attacker-using opener tries to guess passwords (pick) or steal passwords.

Виды механизмов защиты — некоторыми видами механизмов защиты являются: шифрование, аспекты административного управления ключами, механизмы цифровой подписи, механизмы управления доступом, механизмы целостности данных, механизмы обмена информацией аутентификации, механизмы заполнения трафика, механизм управления маршрутизацией, механизм нотаризации, физическая или персональная защита, надежное аппаратное/программное обеспечение.

Ҳимоя механизмлари турлари - ҳимоя механизмларининг баъзи турлари - шифрлаш, калитларни маъмурий бошқариш жиҳатлари, рақамли имзо механизмлари, фойдаланишни бошқариш механизмлари, маълумотлар яхлитлиги механизмлари, аутентификация ахборотини алмашиш механизмлари, трафикни тўлдириш механизмлари, маршрутлашни бошқариш механизми, нотаризация механизми, физик ёки шахсий ҳимоя, ишончли аппарат дастурий таъминот.

Types of protection mechanisms- some kinds of protection mechanisms are: encryption, key management aspects of administrative, digital signature mechanisms, access control mechanisms, mechanisms for data integrity, information exchange mechanisms authentication mechanisms fill traffic routing control mechanism, the mechanism of notarization, physical or personal protection, reliable hardware / software.

Виды угроз — угрозы могут классифицироваться на случайные и преднамеренные и могут быть активными и пассивными.

Таҳдид турлари - таҳдидларни тасодифанларига ва атайинларига, активларига ва пассивларига таснифлаш мумкин.

Types of Threats - threats can be classified into random and deliberate and can be active or passive.

Внешняя схема базы данных - формальное описание базы данных на внешнем уровне в соответствии с конкретной моделью данных.

Маълумотлар базасининг ташқи схемаси – маълумотларнинг муайян моделига мувофиқ маълумотлар базасининг ташқи сатҳдаги расмий тавфсифи.

External scheme database - the formal description of the database at the external level, in accordance with a specific data model.

Внутренняя схема базы данных - формальное описание базы данных на внутреннем уровне в соответствии с конкретной моделью данных.

Маълумотлар базасининг ички схемаси – маълумотларнинг муайян моделига мувофиқ маълумотлар базасининг ички сатҳидаги расмий тавсифи.

Internal schema database - formal description of the database at the domestic level in accordance with a specific data model.

Восстанавливаемость - свойство загружаемого модуля, состоящее в возможности защиты его в процессе выполнения от модификации как им самим, так и любым другим модулем. Программа восстановления может заменить такой модуль новым экземпляром, не повлияв при этом ни на порядок обработки, ни на конечный результат.

Тикланувчанлик – юкланувчи модулнинг бажарилиши жараёнида модификацияланишидан ўзи ёки ихтиёрий бошқа модуль томонидан ҳимоялаш мумкинлиги ҳусусияти. Тиклаш дастури бундай модулни, ишлаш тартибига, якуний натижага таъсир этмасдан, янги нусха билан алмаштириши мумкин.

Recoverability (refreshable) – loadable module property of being able to protect it during the execution of the modification of both themselves and any other module. The recovery program can replace a module with a new instance, without affecting neither an order processing or the end result.

Восстановление - 1) Возврат к исходному значению или к нормальному функционированию. 2) Процесс, с помощью которого станция передачи данных разрешает конфликт или исправляет ошибки, возникающие при/передаче данных.

Тикланиш - 1) Дастлабки қийматига ёки меъёрий ишлашига қайтиш. 2) Жараён бўлиб, унинг ёрдамида маълумотларни узатиш станцияси ихтилофни ҳал этади ёки маълумотларни узатишда пайдо бўлувчи хатоликни тузатади.

Recovery (regeneration) - 1) Return to the initial value or to normal functioning. 2) The process by which data transmission station resolves the conflict or corrects errors. Arising in / data transmission.

Восстановление базы данных - 1) Полная или частичная повторная загрузка базы данных. 2) Воссоздание содержимого базы данных по резервной копии,

выполняемое в случае машинных сбоев или программных ошибок для поддержания целостности данных.

Маълумотлар базасини тиклаш – 1) Маълумотлар базасини тўлиқ ёки қисман қайта юклаш. 2) Машина янглишиши ёки дастурий хатолик холида маълумотлар яхлитлигини мададлаш мақсадида маълумотлар базаси таркибини резерв нусха бўйича аслига келтириш.

D**atabase recovery -** 1) Complete or partial reload database. 2) Restoration of the database contents from a backup performed in the case of machine failures or software errors to maintain data integrity.

Восстановление данных - процесс копирования данных с носителя, содержащего защитную копию данных, на носитель оригинал в случае нарушения на нем целостности данных.

Маълумотларни тиклаш – элтувчининг асл нусхасида маълумотлар яхлитлиги бузилганида унга маълумотларнинг ҳимоя нусхаси бўлган элтувчидан нусҳалаш жараёни.

Data recovery- the process of copying data from one media containing protecting your data on original carrier in case of violation of the integrity of the data on it.

Вскрыватель паролей — программа компьютерная, которая осуществляет подбор или похищение паролей.

Паролларни фош қилувчи - паролларни танлашни ёки ўғрилашни амалга оширувчи компьютер дастури.

Password cracker - computer program that carries out the selection or stealing passwords.

Гарантия защиты - наличие сертификата соответствия для технического средства обработки информации или аттестата на объект информатики, подтверждающих, что безопасность обрабатываемой информации соответствует требованиям стандартов и других нормативных документов.

Ҳимоянинг кафиллиги – ишланадиган ахборот хавфсизлигининг стандартлар ва бошқа меъёрий хужжатлар талабларига мослигини тасдиқловчи, ахборотни ишловчи техник воситаларга мослик сертификатининг ёки информатика объектига аттестатнинг мавжудлиги.

Security accredidation - a certificate of conformity to the technical means of information processing or certificate for Informatics to confirming that the security of information processed complies with the standards and other normative documents.

Государственная тайна - сведения, охраняемые государством, разглашение которых может оказать отрицательное воздействие на качественное состояние военно-экономического потенциала страны или повлечь другие тяжкие последствия для ее обороноспособности, государственной безопасности, экономических и политических интересов. К государственной тайне относится секретная информация с грифами «особой важности» и «совершенно секретно».

Давлат сири - давлат томонидан муҳофаза қилинувчи, фош қилиниши давлатнинг ҳарбий-иқтисодий потенциалининг сифатли ҳолатига салбий таъсир этувчи ёки унинг мудофаа имконияти, давлат хавфсизлиги, иқтисодий ва сиёсий манфаатлари учун бошқа оғир оқибатларга олиб келиши мумкин бўлган маълумотлар. Давлат сири "жуда муҳим" ва "мутлақо махфий" грифли ахборот тааллуқли.

State secret - information protected by the state, the disclosure of which could have a negative impact on the qualitative state of military-economic potential of the country or cause other serious consequences for its defense, national security, economic and political interests. To state secret is secret information classified "special importance" and "top secret".

Готовность системы - мера способности системы выполнять свои функции при нахождении в рабочем состоянии. Количественно готовность можно оценивать с помощью коэффициента готовности.

Тизимнинг тайёрлиги – тизимнинг ишлаш холатида ўз вазифаларини бажариш қобилиятининг ўлчови. Миқдоран, тайёргарликни тайёрлик коэффициенти ёрдамида баҳолаш мумкин.

System availability - measure the system's ability to perform its functions when in working condition. Readiness can be assessed quantitatively by the coefficient of readiness.

Данные - информация, представленная в формализованном виде, пригодном для передачи, интерпретации или обработки с участием человека либо автоматическими средствами.

Маълумотлар – одам иштироки билан ёки автоматик тарзда узатишга, изоҳлашга ёки ишлашга яроқли, формаллашган кўринишда ифодаланган ахборот.

Data - information presented in a formalized manner suitable for communication, interpretation or processing involving human or automated means.

Данные идентификационные — совокупность уникальных идентификационных данных, соответствующая конкретному участнику, позволяющая осуществить однозначную его идентификацию в системе.

Идентификация маълумотлари - тизимда бир маъноли идентификацияланишига имкон берувчи, муайян қатнашчига тегишли ноёб идентификация маълумотлари мажмуи.

Data identification - a set of unique identification data corresponding to a specific party, it allows an unambiguous identification of the system.

Данные персональные — любая информация, относящаяся к определенному или определяемому на основании такой информации

физическому лицу (субъекту персональных данных), в том числе его фамилия, имя, отчество, год, месяц, дата и место рождения, адрес, семейное, социальное, имущественное положение, образование, профессия, доходы, другая информация.

Шахсий маълумотлар - бундай ахборот асосида аниқ ёки аниқланувчи физик шахсга (шахсий маълумотлар субъектига) тегишли ҳар қандай ахборот, жумладан, унинг исми-шарифи, туғилган йили, ойи, куни ва манзили, оилавий, ижтимоий, мулкий ҳолати, маълумоти, касби, даромади, бошқа ахборот.

Personal data - any information relating to an identified or identifiable on the basis of such information to an individual (the subject of personal data), including its name, first name, year, month, date and place of birth, address, marital, social, property status, education, occupation, income, other information.

Дезинформация - сознательное искажение передаваемых сведений с целью ложного представления у лиц, использующих эти сведения; передача ложной информации.

Дезинформация – фойдаланувчи шахсларга ёлғон тасаввурни шакллантириш мақсадида уларга узатилувчи хабарни атайин бузиб кўрсатиш; ёлғон ахборотни узатиш.

Misinformation - deliberate distortion of transmitted data with the purpose of the false representations in individuals using this information; transmission of false information.

Доверие — основа для уверенности в том, что продукт или система технологий информационных отвечают целям безопасности.

Ишонч - ахборот технологиялари маҳсулотининг ёки тизимининг хавфсизлик мақсадларига жавоб беришига ишониш учун асос.

Assurance - basis for confidence that the product or system information technology meet the security objectives.

Доверительность - свойство соответствия безопасности некоторым критериям.

Ишончлилик – хавфсизликнинг қандайдир мезонларга мослик хусусияти.

Trusted functionality - compliance with safety properties of some criteria.

Доктрина информационной безопасности — совокупность официальных взглядов на цели, задачи, принципы и основные направления обеспечения информационной безопасности.

Ахборот хавфсизлиги доктринаси - ахборот хавфсизлигини таъминлаш мақсадларига, масалаларига, принципларига ва асосий йўналишларига расмий қарашлар мажмуи.

Information Security Doctrine -totality of official views on the goals, objectives, principles and guidelines ensuring the information security.

Документ - форма существования информации в виде тестовых и графических материалов, выполненных любыми способами, а также в виде перфорированных и магнитных носителей, фото - и кинопленок. Текстовые и графические материалы могут быть написаны от руки, нарисованы, выгравированы, начерчены, напечатаны на машинке или исполнены типографским способом.

Ҳужжат - ахборотнинг ихтиёрий усулларда амалга оширилган матнли ва график материаллар кўринишида ҳамда перфорацияланган ва магнит элтувчиларда, фото ва киноплёнка кўринишида мавжудлик шакли. Матнли ва график материаллар қўлда ёзилиши, тасвирланиши, чизилиши, машинкада ёзилиши ёки типографик учул орқали бажарилиши мумкин.

Document - existence form information in the form of test and graphics made by any means, as well as perforated and magnetic carriers, the photo - and films. Text

and graphics can be written by hand, painted, engraved, drawn, typed or filled in hard copy.

Документ конфиденциальный — документ ограниченного доступа на любом носителе, содержащий информацию конфиденциальную.
Махфий ҳужжат - махфий ахборотли ҳар қандай элтувчидан фойдаланиш чекланган ҳужжат.
Confidential document - document restricted in any medium, containing confidential information.

Документированная информация — зафиксированная на материальном носителе информация с реквизитами, позволяющими ее идентифицировать.
Ҳужжатланган ахборот - реквизитлари идентификацияланишига имкон берувчи, материал элтувчида қайдланган ахборот
Documented information - fixed in a tangible medium with requisites allowing its identification.

Домен безопасности - ограниченная группа объектов и субъектов безопасности, к которым применяется одна методика безопасности со стороны одного и того же администратора безопасности.
Хавфсизлик домени – хавфсизликнинг битта маъмури томонидан хавфсизликнинг бир хил усули қўлланиладиган хавфсизлик субъектлари ва объектларининг чекланган гурухи.
Security domain - limited group of objects and subjects of security, to which the one method of security from the same security administrator.

Достоверность - свойство информации быть правильно воспринятой; вероятность отсутствия ошибок.
Ишончлилик – ахборотниниг тўғри ўзлаштирилиш хусусияти; хатолик йўқлигининг эҳтимоллиги.

Validity, adequacy - property information to be correctly perceived; the probability of no errors.

Доступ - предоставление данных системе обработки данных или получение их из нее путем выполнения операций поиска, чтения и (или) записи данных.

Фойдаланиш - маълумотларни ишлаш тизимига маълумотларни тақдим этиш ёки ундан қидириш, ўқиш ва/ёки ёзиш амалларини бажариш йўли билан маълумотларни олиш.

Access - providing data processing system or getting them out of it by doing a search, read and (or) data record.

Доступ к информации - процесс ознакомления с информацией, ее документирование, модификация или уничтожение, осуществляемые с использованием штатных технических средств.

Ахборотдан фойдаланиш – штатга оид техник воситалардан фойдаланиб ахборот билан танишиш, уни хужжатлаш, нусхалаш, модификациялаш ёки ахборотни йўқ қилиш жараёни.

Access to information - the process of reviewing the information, documenting, modification or destruction, implemented by the staff of technical means. still - familiar with the information, information processing, in particular, copying, modification or destruction of information.

Доступ к конфиденциальной информации — санкционированное полномочным должностным лицом ознакомление конкретного лица с информацией, содержащей сведения конфиденциального характера.

Конфиденциаль ахборотдан фойдаланиш - муайян шахсга таркибида конфиденциаль характерли маълумот бўлган ахборот билан танишишга ваколатли мансабдор шахснинг рухсати.

Access to confidential information - authorized authorized official introduction of a particular person with the information containing confidential information.

Доступ несанкционированный к информации — получение защищаемой информации заинтересованным субъектом с нарушением установленных правовыми документами или собственником, владельцем информации прав или правил доступа к защищаемой информации.

Ахборотдан рухсатсиз фойдаланиш - манфаатдор субъект томонидан ўрнатилган ҳуқуқий ҳужжатларни ёки мулкдор, ахборот эгаси томонидан ҳимояланувчи ахборотдан фойдаланиш ҳуқуқлари ёки қоидаларини бузиб ҳимояланувчи ахборотга эга бўлиш.

Unauthorized access to information - preparation of protected information interested entity in violation of the legal instruments or by the owner, the owner of the information or rights of access to protected information.

Доступ ограниченный — доступ к ресурсу информационному, разрешаемый установленными для данного ресурса правилами доступа только определенному кругу лиц, обладающих соответствующими полномочиями.

Чекланган фойдаланиш - ахборот ресурсидан, ушбу ресурсга фақат мос ваколатларга эга шахсларнинг маълум доирасига ўрнатилган фойдаланиш қоидалари бўйича рухсатли фойдаланиш.

Restricted access - access to the resources of the information allowed by the established rules for the resource access only certain persons with appropriate authority.

Доступность — свойство объекта находиться в состоянии готовности и используемости по запросу авторизованного логического объекта.

Фойдалувчанлик - авторизацияланган фойдаланувчи сўрови бўйича мантиқий объектнинг тайёрлик ва фойдаланувчанлик ҳолатида бўлиши хусусияти

Availability, accessibitity - property of an object in a state of readiness and usage upon request authorized entity.

Живучесть - свойство системы оставаться работоспособной в условиях внешних воздействий.

Яшовчанлик – тизимнинг ташқи таъсирлар шароитида ишга лаёқатли қолиши хусусияти.

Viability - property of the system to remain operational under external influences.

Журнал - в вычислительной технике набор данных (файл), используемый операционной или иной системой для сбора и учета статистической информации, различных сообщений и других данных.

Журнал –статистик ахборотни, турли хабарларни ва бошқа маълумотларни йиғиш ва ҳисобга олиш учун операцион ёки бошқа тизим фойдаланадиган ҳисоблаш техникасидаги маълумотлар набори (файл).

Journal, log - in computing the data set (file) used by the operating system or another for collection and recording of statistical information, different messages and other data.

Журнал восстановления - журнал, обеспечивающий возможность восстановления базы данных или файла. Содержит информацию о всех изменениях в БД (файле) с того момента, когда было установлено, что данные достоверны и была сделана последняя резервная копия.

Тиклаш журнали – маълумотлар базаси ёки файлни тиклаш имкониятини таъминловчи журнал. Унда маълумотлар базасидаги (файлдаги) маълумотларнинг хақиқийлиги аниқланган ва охирги резерв нусха олинган ондан бошлаб, барча ўзгаришлар хусусида ахборот мавжуд.

Recovery log - magazine, providing the ability to restore a database or file. Contains information about all the changes in DB (file) from the moment when it was found that the data is reliable and has been made the last backup.

Журнал ошибок - файл, в который система записывает информацию о сбоях.

Хатоликлар журнали – тизим томонидан адашишлар хусусидаги ахборот ёзиладиган файл.

Error Log - file in which the system records information about failures.

Журнализация - процесс записи в системный журнал информации о сообщениях, запросах, выполнявшихся программах, использованных наборах данных и других сведений.

Журналлаштириш – тизимли журналга хабарлар, сўровлар, бажарилган дастурлар, ишлатилган маълумотлар набори ва бошқа маълумотлар хусусида ахборотни ёзиш жараёни.

Journalizing - process the system log information about messages, queries, execute programs, use a set of data and other information.

Запрос идентификации (опознания) - запрос, заданный ведущей станцией ведомой станции для ее идентификации или определения ее состояния.

Идентификация сўрови – бошқарувчи станциянинг бошқарилувчи станцияга уни идентификациялаш ёки холатини аниқлаш учун берган сўрови.

Request identification - query specified slave master station to identify it or determine its status.

Зарегистрированный пользователь - пользователь, имеющий приоритетный номер в данной системе коллективного пользования.

Руйхатга олинган фойдаланувчи – берилган коллектив фойдаланувчи тизимда устувор номерли фойдаланувчи.

Authorized user - a user with a priority number in the system of collective use.

Защита - средство для ограничения доступа или использования всей или части вычислительной системы; юридические, организационные и

технические, в том числе программные, меры предотвращения несанкционированного доступа к аппаратуре, программам и данным.

Ҳимоялаш - ҳисоблаш тизимидан ёки унинг қисмидан фойдаланишни чеклаш воситаси; аппаратурадан, дастурдан ва маълумотлардан рухсатсиз фойдаланишни бартараф этувчи ташкилий ва техник, жумладан, дастурий чоралар.

Protection, security, lock out - means for restriction of access or use of all or part of the computing system; legal, organizational and technical, including program, measures of prevention of unauthorized access to the equipment, programs and data.

Защита данных - охрана данных от несанкционированного, умышленного или случайного их раскрытия, модификации или уничтожения.

Маълумотларни ҳимоялаш – маълумотларни рухсатсиз, атайин ёки тасодифан очилишидан, модификацияланишидан ёки йўқ қилинишидан қўриқлаш.

Data protection - protection of data from unauthorized, deliberate or their casual disclosure, modification or destruction.

Защита информации - включает в себя комплекс мероприятий, направленных на обеспечение информационной безопасности. На практике под этим понимается поддержание целостности, доступности и, если нужно, конфиденциальности информации и ресурсов, используемых для ввода, хранения, обработки и передачи данных.

Ахборотни ҳимоялаш –ахборот хавфсизлигини таъминлашга йўналтирилган тадбирлар комплекси. Амалда ахборотни ҳимоялаш деганда маълумотларни киритиш, сақлаш, ишлаш ва узатишда унинг яхлитлиги, фойдаланувчанлигини ва, агар керак бўлса, ахборот ва ресурсларнинг конфиденциаллигини мададлаш тушунилади.

Information protection - includes a complex of the actions aimed at providing information security. In practice it is understood as maintenance of integrity, availability and if it is necessary, confidentiality of information and the resources used for input, storage, and processing and data transmission.

Защита от несанкционированного доступа - предотвращение или существенное затруднение несанкционированного доступа к программам и данным путем использования аппаратных, программных и криптографических методов и средств защиты, а также проведение организационных мероприятий. Наиболее распространенным программным методом защиты является система паролей.

Рухсатсиз фойдаланишдан ҳимоялаш – аппарат-дастурий ва криптографик усуллар ва воситалар ёрдамида, ҳамда ташкилий тадбирларни ўтказиб дастурлардан ва маълумотлардан рухсатсиз фойдаланишни бартараф этиш ёки жиддий қийинлаштириш. Ҳимоялашнинг энг кенг тарқалган дастурий усули пароллар тизими ҳисобланади.

Protection from unauthorized access - prevention or essential difficulty of unauthorized access to programs and this way of use of hardware, program and cryptographic methods and means of protection, and also carrying out organizational actions. The most widespread program method of protection is the system of passwords.

Защищенная программа- программа, защищенная от копирования.

Ҳимояланган дастур – нусхалашдан ҳимояланган дастур.

Copy protected software - the program protected from copying.

Защищенная система- система, вход в которую требует ввода пароля.

Ҳимояланган тизиим – кириш учун пароль талаб қилинадиган тизим.

Protected system- system, an entrance in which demands password input.

Защищенность - в вычислительной технике способность системы противостоять несанкционированному доступу к программам и данным (безопасность, секретность), а также их случайному искажению или разрушению (целостность).

Ҳимояланганлик – ҳисоблаш техникасида тизимнинг дастурлардан ва маълумотлардан рухсатсиз фойдаланишига (хавсизлик, махфийлик) ҳамда уларни тасодифан бузилишига (яхлитлик) ёки ўзгартирилишига қарши тура олиш хусусияти.

Security, immunity - in computer facilities ability of system to resist to unauthorized access to programs and data (safety, privacy), and also to their casual distortion or destruction (integrity).

Защищенный файл - файл, для доступа к записям которого необходимо ввести пароль.

Ҳимояланган файл – ёзувларидан фойдаланиш учун парол талаб қилинадиган файл.

Protected file - the file, for access to which records it is necessary to enter the password.

Злоумышленник - лицо или организация, заинтересованные в получении несанкционированного доступа к программам или данным, предпринимающие попытку такого доступа или совершившие его.

Нияти бузуқ – дастурлардан ёки маълумотлардан рухсатсиз фойдаланишдан манфаатдор, бундай фойдаланишга уринган ёки амалга оширган шахс ёки ташкилот.

Intruder - the person or the organization interested in receiving unauthorized access to programs or data, making an attempt of such access or made it.

Идентификатор - средство идентификации доступа, представляющее собой отличительный признак субъекта или объекта доступа. Основным средством идентификации доступа для пользователей является пароль.

Идентификатор – субъект ёки объектнинг фарқланувчи аломатидан иборат фойдаланишнинг идентификация воситаси. Фойдаланувчилар учун асосий идентификация воситаси пароль ҳисобланади.

Identifier - means of identification of the access, representing a distinctive sign of the subject or object of access. The main means of identification of access for users is the password.

Идентификатор доступа- уникальный признак субъекта или объекта доступа.

Фойдаланиш идентификатори – фойдаланувчи субъект ёки объектнинг ноёб аломати.

Access identifier - unique sign of the subject or object of access.

Идентификатор пользователя– символическое имя, присваиваемое отдельному лицу или группе лиц и разрешающее использование ресурсов вычислительной системы.

Фойдаланувчи идентификатори – ҳисоблаш тизими ресурсларидан фойдаланиш учун алоҳида шахсга ёки шахслар гурухига бериладиган рамзий исм.

User identifier, userid – symbol the check name appropriated to the individual or a group of persons and allowing use of resources of the computing system.

Идентификация - присвоение субъектам и объектам доступа идентификатора и/или сравнение предъявляемого идентификатора с перечнем присвоенных идентификаторов.

Идентификация – фойдаланиш субъектлари ва обектларига идентификатор бериш ва/ёки тақдим этилган идентификаторни берилганлари рўйҳати билан таққослаш.

Identification -assignment to subjects and objects of access of the identifier and/or comparison of the shown identifier with the list of the appropriated identifiers.

Идентификация и аутентификация — общий термин для обозначения процесса представления своего имени и подтверждения подлинности (идентичности) пользователя системы, который выполняется им для получения права доступа к ресурсам.

Идентификация и аутентификация — тизим фойдаланувчисининг ресурслардан фойдаланиш хуқуқини қўлга киритиш учун бажариладиган ўз номини таништириш ва хақиқийлигини (айнан ўзи эканлигини) тасдиқлаш жараёнини белгиловчи умумий атама.

Identification and Authentication (I&A) — the general term for designation of process of representation of the name and confirmation of authenticity (identity) of the user of system who is carried out by it for receiving right of access to resources.

Иерархическая модель данных- модель данных для представления данных иерархической структуры.

Маълумотларнинг шажара модели – шажара структураси маълумотларини ифодаловчи маълумотлар модели.

Hierarchical model - model given for data presentation of hierarchical structure.

Иерархическая структура данных- структура данных, представляющая собой множество, частично упорядоченное таким образом, что существует только один элемент этого множества, не имеющий предыдущего, а все другие элементы имеют только один предыдущий.

Маълумотларнин шажара структураси – қисман тартибга солинган тўпламдан иборат маълумотлар структураси. Бунда ушбу тўпламнинг олдинги элементларига эга бўлмаган фақат битта элементи мавжуд, бошқа барча элементлари эса фақат битта олдинги элементга эга.

Hierarchical data structure - the structure of data representing a set, partially ordered in such a way that exists only one element of this set which doesn't have previous, and all other elements have only one previous.

Избирательное управление доступом- метод управления доступом субъектов системы к. объектам, основанный на идентификации и опознавании пользователя, процесса и/или группы, к которой он принадлежит. Управление является избирательным в том смысле, что субъект с определенными правами может осуществлять передачу прав любому объекту независимо от установленных ограничений.

Фойдаланишни танлаб бошқариш – фойдаланувчини, жараённи ва/ёки у тегишли гуруҳни идентификациялашга ва танишга асосланган тизим субъектларининг объектлардан фойдаланишни бошқариш усули. Бунда маълум хуқуқли субъект хуқуқларни ҳар қандай объектга, ўрнатилган чеклашларга боғлиқ бўлмаган ҳолда, узатишни амалга ошириши мумкин.

Discretionary access control - method of control over access of subjects of system to. To the objects, based on identification and an identification of the user, process and/or group to which it belongs. Management is selective in the sense that the subject with certain rights can carry out transfer of rights to any object irrespective of the set restrictions (access can be provided and not directly).

Избыточность- введение в систему дополнительных компонентов сверх минимально необходимого их числа с целью повышения надежности системы. Различают избыточность аппаратную, информационную, алгоритмическую.

Ортиқчалик – тизим ишончлилигини ошириш мақсадида унга керагидан ортиқ қўшимча компонентларнинг киритилиши. Аппарат, алгоритм, информацион ортиқчаликлар фарқланади.

Redudancy - introducing additional components into the system in excess of the minimum required number of them in order to increase system reliability. There are redundant hardware, information, algorithmic.

Инженерия социальная — обход системы безопасности системы информационной с помощью информации, получаемой из контактов с обслуживающим персоналом и пользователями на основе введения их в заблуждение за счет различных уловок, обмана и пр.

Ижтимоий инженерия – хизматчи ходимлар ва фойдаланувчилар билан, турли ҳийла-найранг, алдов орқали чалғитиш асосидаги мулоқотдан олинадиган ахборот ёрдамида ахборот тизимининг хавфсизлик тизимини четлаб ўтиш.

Social engineering — round of system of safety of system information by means of information received from contacts with the service personnel and users on the basis of their introduction in delusion at the expense of various tricks, deception and so forth.

Инсайдер — член группы людей, имеющей доступ к закрытой информации, принадлежащей этой группе. Как правило, является ключевым персонажем в инциденте, связанным с утечкой информации. С этой точки зрения различают следующие типы инсайдеров: халатные, манипулируемые, обиженные, нелояльные, подрабатывающие, внедренные и т.п.

Инсайдер – гуруҳга тегишли яширин ахборотдан фойдаланиш ҳуқуқига эга гурух аъзоси. Одатда, ахборот сирқиб чиқиш билан боғлиқ можорода муҳим шахс ҳисобланади. Шу нуқтаи назардан, инсайдерларнинг қуйидаги хиллари фарқланади: бепарволар; манипуляцияланувчилар, ранжиганлар, қўшимча пул ишловчилар ва х.

Insider — the member of group of the people having access to the classified information, belonging this group. As a rule, is the key character in the incident, connected with information leakage. From this point of view distinguish the following types of insiders: negligent, manipulated, offended, disloyal, earning additionally, introduced, etc.

Информационная надежность – 1.Способность алгоритма или программы правильно выполнять свои функции при различных ошибках в исходных данных. 2.Способность информационной системы обеспечивать целостность хранящихся вней данных.

Ахборот ишончлилиги – 1. Дастлабки маълумотлардаги турли хатоликларда алгоритм ёки дастурнинг ўз вазифасини тўғри бажариш қобилияти. 2. Ахборот тизимининг унда сақланаётган маълумотлар яхлитлигини таъминлаш қобилияти.

Information reliability –1. Ability of algorithm or the program it is correct to carry out the functions at various mistakes in basic data. 2. Ability of information system to provide integrity of the data which were stored in it.

Информационная система - организационно упорядоченная совокупность документов (массивов документов) и информационных технологий, в том числе с использованием средств вычислительной техники и связи, реализующих информационные процессы.

Ахборот тизими – хужжатларнинг (хужжатлар массивининг) ва ахборот технологияларининг, хусусан ахборот жараёнларини амалга оширувчи хисоблаш техникаси ва алоқа воситаларидан фойдаланиб, ташкилий тартибга солинган мажмуи.

Information system - organizationally ordered set of documents (document files) and information technologies, including with use of computer aids and the communications, realizing information processes.

Информационная технология - система технических средств и способов обработки информации.

Ахборот технологияси – ахборотни ишлаш усуллари ва техник воситалари тизими.

Information technology - system of technical means and ways of information processing.

Информация аутентификации - информация, используемая для установления подлинности личности, за которую выдает себя пользователь.

Аутентификация ахбороти–фойдаланувчининг ҳақиқатдан айнан ўзи эканлигига ишонч хосил қилишда фойдаланиладиган ахборот.

Authenfication information - information used for establishment of authenticity of the personality for which the user gives out himself.

Инцидент - зафиксированный случай попытки получения несанкционированного доступа или проведения атаки на компьютерную систему.

Можоро – рухсатсиз фойдаланиш хуқуқига эга бўлишга ёки компъютер тизимига хужум ўтказишга уринишнинг қайд этилган холи.

Incident— the recorded case of attempt of receiving unauthorized access or carrying out attack to computer system.

Искажение - отклонение значений параметров сигнала данных от установленных требований. Еще - изменение содержимого сообщения, передаваемого по линии связи.

Бузилиш – маълумотлар сигнали параметрлари қийматларининг ўрнатилган талаблардан четланиши. Яна -алоқа линияси бўйича узатилувчи хабар таркибининг ўзгариши.

Distortion - deviation of values of parameters of a signal of data from the established requirements. Still - change of contents of the message transferred on the communication lines.

Канал передачи данных — физическая среда, по которой передается информация из одного устройства в другое.

Маълумотларни узатувчи канал - физик муҳит, у орқали ахборот бир қурилмадан иккинчисига узатилади.

Data transmission channel — the physical environment on which information from one device is transferred to another.

Канал проникновения — физический путь от злоумышленника к источнику конфиденциальной информации, посредством которого возможен несанкционированный доступ к охраняемым сведениям.

Кириб олиш канали - нияти бузуқдан то конфиденциаль ахборот манбаигача бўлган йўл. У орқали ҳимояланувчи маълумотлардан рухсатсиз фойдаланиши мумкин.

Insecurity channel — actual path from the malefactor to a source of confidential information by means of which unauthorized access to protected data is possible.

Киберпреступность — действия отдельных лиц или групп, направленные на взлом систем компьютерной защиты, на хищение или разрушение информации в корыстных или хулиганских целях.

Кибержиноятилик - ғаразли ёки хулиганлик мақсадларида ҳимоялашнинг компьютер тизимларини бузиб очишга, ахборотни ўғирлашга ёки бузишга йўналтирилган алоҳида шахсларнинг ёки гуруҳнинг ҳаракатлари.

Cybercrime — actions of individuals or the groups, directed on breaking of systems of computer protection, on plunder or information destruction in the mercenary or hooligan purposes.

Ключ базы данных - ключ, присвоенный системой управления базами данных и однозначно идентифицирующий запись базы данных.

Маълумотлар базаси калити –маълумотлар базасини бошқариш тизими томонидан берилган ва маълумотлар базасидаги ёзувни бир маънода идентификацияловчи калит.

Database key - the key which is appropriated by a database management system and unambiguously identifying record of a database.

Ключ управления доступом - значение, предъявляемое процессом системе управления базами данных и сравниваемое ею с соответствующим замком с целью предотвращения несанкционированного доступа к данным.

Фойдаланишни бошқариш калити –жараён томонидан маълумотлар базасини бошқариш тизимига берилувчи ва маълумотлардан рухсатсиз фойдаланишни бартараф этиш мақсадида мос қулф билан таққосланувчи қиймат.

Access control key - the value shown by process to a database management system and compared by it to the corresponding lock for the purpose of prevention of unauthorized access to data.

Код – 1. Представление символа двоичным кодом. 2. Криптографический прием, в котором используется произвольная таблица или кодировочная книга для преобразования текста в закодированную форму.

Код - 1. Символни иккилик код орқали ифодалаш. 2. Матнни кодланган шаклга ўзгартиришда ихтиёрий жадвалдан ёки кодлаш китобидан фойдаланувчи криптографик усул.

Code -1.Symbol representation by a binary code. 2. Cryptographic reception in which any table or the quoted book for transformation of the text to the coded form is used.

Код аутентификации — вид алгоритма кодирования имитозащищающего информации. Как правило, к. а. сопоставляет сообщению его код аутентичности сообщения. Алгоритм принятия решения о подлинности информации основан на проверке значения кода аутентичности сообщения.

Аутентификация коди – ахборотни имитоҳимояловчи кодлаш алгоритмининг тури. Одатда, аутентификация коди хабарни унинг ҳақиқийлиги коди билан таққослайди. Ахборотнинг ҳақиқийлиги хусусида қараб қабул қилиш алгоритми хабар ҳақиқийлиги коди қийматини текширишга асосланган.

Authentication code — type of coding algorithm simulation protected information. As a rule, it is post code authenticity of the message. Decision algorithm authentication information is based on checking the authenticity of the message code value.

Компрометация - утеря критичной информации либо получение ее неавторизованными для этого субъектами (лицами, программами, процессами и т.д.)

Обрўсизлантириш –жиддий ахборотни йўқотиш ёки уни авторизацияланмаган субъектлар (шахслар, дастурлар жараёнлар ва х.к.) томонидан ўзлаштирилиши.

Compromising - loss of critical information or receiving it the subjects not authorized for this purpose (persons, programs, processes, etc.)

Контроль доступа - определение и ограничение доступа пользователей, программ или процессов к устройствам, программам и данным вычислительной системы.

Фойдаланиш назорати –фойдаланувчиларнинг, дастурларнинг ёки жараёнларнинг ҳисоблаш тизимлари қурилмаларидан, дастурларидан ва маълумотларидан фойдаланишларини аниқлаш ва чеклаш.

Access control - definition and restriction of access of users, programs or processes to devices, programs and data of the computing system.

Концептуальная модель - формальное представление проблемной области на понятийном уровне.

Концептуал модель – муаммоли соҳанинг тушунча сатхидаги расман ифодаси.

Conceptual model- formal representation of problem area at conceptual level.

Концепция защиты информации - система взглядов и общих технических требований по защите информации.

Ахборотни ҳимоялаш концепцияси –ахборотни ҳимоялаш бўйича қарашлар ва умумий техник талаблар тизими.

The concept of information security - frame of reference and the general technical requirements on information security.

Лицензия - разрешение на право продажи или предоставления услуг.

Лицензия – сотиш ёки хизмат кўрсатиш ҳуқуқига рухсатнома.

License - permission to the right of sale or service.

Лицензия в области защиты информации - разрешение на право проведения тех или иных работ в области защиты информации, оформленное лицензионным соглашением (договором).

Ахборот ҳимояси соҳасидаги лицензия –ахборот хавфсизлиги соҳасида у ёки бу ишларни бажариш ҳуқуқига лицензион битим (шартнома) билан расмийлаштирилган рухсатнома.

License information security - permission to the right of carrying out these or those works in the field of the information security, issued by the license agreement/contract/.

Ложная информация - информация, ошибочно отражающая характеристики и признаки, а также информация о не существующем реально объекте.

Ёлғон ахборот – характеристикаларни ва аломатларни нотўғри акслантирувчи ҳамда реал мавжуд бўлмаган объект ҳусусидаги ахборот.

False information - information which is mistakenly reflecting characteristics and signs, and also information on object not existing really.

Мандат - разновидность указателя, определяющего путь доступа к объекту и разрешенные над ним операции.

Мандат – объектдан фойдаланиш ва унинг устида рухсат этилган амалларни бажариш йўлини аниқловчи кўрсаткич тури.

Capability, Mandate - kind of the index defining a way of access to object and operations allowed over it.

Мандатное управление доступом - концепция (модель) доступа субъектов к информационным ресурсам по грифу секретности разрешенной к пользованию информации, определяемому меткой секретности (конфиденциальности).

Фойдаланишни мандатли бошқариш –- махфийлик (конфиденциаллик) белгиси орқали аниқланувчи махфийлик грифи буйича ахборотдан фойдаланишга рухсат этилган субъектларнинг ахборот ресурсларидан фойдаланиш концепцияси (модели).

Mandate management access - the concept (model) of access of subjects to information resources on the security classification of information allowed for using determined by a tag of privacy/confidentiality/.

Матрица доступа - таблица, отображающая правила доступа субъектов к информационным ресурсам, данные о которых храняться в диспетчере доступа. Еще- таблица, отображающая правила разграничения доступа.

Фойдаланиш матрицаси – хусусидаги маълумотлар фойдаланиш диспетчерида сақланувчи ахборот ресурсларидан субъектларнинг фойдаланиш қоидаларини акс эттирувчи жадваллар; яна - фойдаланишни чеклаш қоидаларини акс эттирувчи жадвал.

Access matrix – the table displaying rules of access of subjects to information resources, given about which are stored in the dispatcher of access. Also, the table displaying rules of differentiation of access.

Матрица полномочий - таблица, элементы которой определяют права (полномочия, привилегии) определенного объекта относительно защищаемых данных.

Ваколатлар матрицаси – элементлари муайян объектнинг ҳимояланувчи маълумотларга нисбатан ҳуқуқларини (ваколатларини, имтиёзларини) белгиловчи жадвал.

Privilege matrix - table, elements which determine the right (powers and privileges) with respect to a certain object protected data.

Менеджмент риска — полный процесс идентификации, контроля, устранения или уменьшения последствий опасных событий, которые могут оказать влияние на ресурсы информационно-телекоммуникационных технологий.

Хавф-хатар менеджменти — ахборот-телекоммуникация технология ресурсларига таъсир этиши мумкин булган хавфли ходисалар окибатларини идентификациялашнинг, назоратлашнинг, бартараф этишнинг ёки камайтиришнинг тулиқ жараёни.

Risk management — full process of identification, control, elimination or reduction of consequences of dangerous events which can have impact on resources of information and telecommunication technologies.

Надежность - характеристика способности функционального узла, устройства, системы выполнять при определенных условиях требуемые функции в течение определенного периода времени.

Ишончлилик – маълум шароитларда берилган вақт оралиғида функциональ узелнинг, қурилманинг, тизимнинг ўзига топширилган вазифаларни бажариш қобилиятининг характеристикаси.

Reliability - the ability of the functional characteristics of node devices, the system under certain circumstances to carry out the desired function during a certain period of time.

Нападающий — субъект, действия которого нарушают безопасность информации в рассматриваемой компьютерной системе.

Хужумчи - ҳаракати кўрилаётган компьютер тизимида ахборот хавфсизлигини бузадиган субъект.

Attacker - a subject whose actions violate the information security in a under consideration computer system.

Нарушение полномочий - попытка пользователя или программы выполнить неразрешенную операцию.

Ваколатларнинг бузилиши –фойдаланувчининг ёки дастурнинг рухсат этилмаган амални бажаришга уриниши.

Privilege violation - user or program attempts to perform an unauthorized operation.

Нарушение системы безопасности — успешное поражение средства управления безопасностью, которое завершается проникновением в систему.

Хавфсизлик тизимининг бузилиши - тизимга суқилиб кириш билан тугалланадиган хавфсизликни бошқариш воситаларининг шикастланиши.

Security system violation - the successful defeat security controls, which concludes with penetration into the system.

Нарушение целостности - искажение содержимого записей файла или базы данных. Происходит вследствие машинных сбоев, программных ошибок, а также ошибочных действий пользователей.

Яхлитликнинг бузилиши - файл ёки маълумотлар базаси ичидаги ёзувларнинг бузилиши. Машинанинг янглишиши, дастурий хатоликлар ҳамда фойдаланувчиларнинг нотўғри ҳаракатлари натижасида рўй беради.

Integrity violation - the distortion of the contents of the recorded files or database. This is due to machine failures, software errors and erroneous actions of users.

Нарушение целостности информации - утрата информации, при ее обработке техническими средствами, свойства целостности в результате ее несанкционированной модификации или несанкционированного уничтожения.

Ахборот яхлитлигининг бузилиши –ахборотнинг, уни техник воситалари ёрдамида ишланишида, йўқотилиши рухсатсиз модификацияланиши ёки йўқ қилиниши натижасида, яхлитлик хусусиятининг йўқолиши.

Information integrity violation - the loss of information when it is processed by technical means, the integrity of the property as a result of its unauthorized modification or unauthorized destruction.

Нарушитель - субъект, действия которого нарушают безопасность информации в рассматриваемой компьютерной системе.

Бузғунчи – ҳаракатлари кўрилаётган компьютер тизимида ахборот хавфсизлигини бузадиган субъект.

Attacker - a subject whose actions violate the information security in a computer system under consideration.

Нарушитель внешний — рекомендуется использовать термин противник.

Ташқи бузғунчи - душман атамасидан фойдаланиш тавсия этилади.

External violator -it is recommended to use the term enemy.

Обработка данных - систематическое выполнение операций над данными.

Маълумотларни ишлаш – маълумотлар устида амалларнинг мунтазам бажарилиши.

Data processing - manipulation of data by a computer.

Ошибка в данных - ошибочное представление одного или нескольких исходных данных может стать причиной аварийного завершения программы либо оказаться необнаруженной, но результаты нормально завершившейся программы будут при этом неверными.

Маълумотлардаги хатолик - бир ёки бир неча дастлабки маълумотларнинг хато ифодаланиши дастурнинг аварияли тугалланишига сабаб бўлиши мумкин ёки хатолик аниқланмаслиги мумкин, аммо тугалланган дастур натижаси нотўғри бўлади.

Data error - a condition in which data on a digital medium has been altered erroneously. The error can manifest as several incorrect bits or even a single bit that is 0 when it should be 1 or vice versa.

Пароль — уникальная последовательность символов, которую необходимо ввести, но запросу компьютера, чтобы получить доступ к системе, программе или данным.

Пароль – тизимдан, дастурдан ёки маълумотлардан фойдаланишга рухсат олиш учун компьютер сўрови бўйича киритиладиган символларнинг ноёб кетма-кетлиги.

Password - a password is an unspaced sequence of characters used to determine that a computer user requesting access to a computer system is really that particular user.

Подделка информации - умышленная несанкционированная модификация информации при ее обработке техническими средствами с целью получения определенных выгод (преимуществ) перед конкурентом или нанесения ему ущерба.

Ахборотни соҳталаш – ахборотнинг техник воситаларда ишланишида рақибнинг олдида муайян фойда (афзаллик) олиш мақсадида ахборотни атайин рухсатсиз модификациялаш.

Fake information (Forgery) - intentional unauthorized modification of data when it is processed by technical means to obtain certain benefits (benefits) to a competitor or suffering damage.

Подотчетность — возможность проверки; имеет две стороны: во-первых, любое состояние системы можно вернуть в исходное для выяснения того, как система в нем оказалась; во-вторых, имеющийся порядок проведения аудита безопасности позволяет гарантировать, что система удовлетворяет всем заявленным требованиям.

Ҳисобдорлик - текшириш имконияти. Иккита жиҳатга эга. Биринчидан, тизимнинг ҳар қандай ҳолатини, ушбу ҳолатга қай тарзда тушиб қолганини аниқлаш учун, дастлабки ҳолатига қайтариш. Иккинчидан, хавфсизлик аудитини ўтказишнинг мавжуд тартиби тизимнинг барча билдирилган талабларни қониқтиришини кафолатлашга имкон беради.

Auditability - ability to test; has two aspects: firstly, any state of the system can be reset to determine how the system was in it; Second, the existing procedures for auditing the security helps ensure that your system meets all the stated requirements.

Политика безопасности (информации в организации) — совокупность документированных правил, процедур, практических приемов или

руководящих принципов в области безопасности информации, которыми руководствуется организация в своей деятельности.

Хавфсизлик сиёсати (ташкилотдаги ахборот хавфсизлиги сиёсати) - ташкилот ўз фаолиятида риоя қиладиган ахборот хавфсизлиги соҳасидаги ҳужжатланган қоидалар, муолажалар, амалий усуллар ёки амал қилинадиган принциплар мажмуи.

Security policy - set of documented policies, procedures, practical methods or guidelines in the field of information security used by the organization in its activities.

Полномочия - право пользователя (терминала, программы, системы) осуществлять те или иные процедуры над защищенными данными.

Ваколатлар –ҳимояланган маълумотлар устида у ёки бу муолажани бажариши бўйича фойдаланувчининг (терминалнинг, дастурнинг, тизимнинг) ҳуқуқи.

Privileges - the right of the user (terminal program, system) to implement certain procedures over the protected data.

Полномочное управление доступом - разграничение доступа субъектов к объектам, основанное на характеризуемой меткой конфиденциальности информации, содержащейся в объектах, и официальном разрешении субъектов обращаться к информации такого уровня конфиденциальности.

Фойдаланишни ваколатли бошқариш - объектлар таркибидаги ахборотнинг конфиденциаллигини характерловчи белгига ва субъектларнинг бундай конфиденциаллик даражасига эга информацияга мурожаат этишларига расмий рухсатга асосланган субъектларнинг объектлардан фойдаланишларини чеклаш.

Plenipotentiary access control - access control subjects to objects based on the characterized Tagged confidentiality of the information contained in the objects,

and the authorization of subjects to access information of such a level of confidentiality.

Предоставление права на доступ - выдача разрешения (санкции) на использование определенных программ и данных.

Фойдаланиш ҳуқуқини тақдим этиш – муайян дастурлар ва маълумотлардан фойдаланишга рухсат (санкция) бериш.

Authorization - authorization (approval) to use certain programs and data.

Профиль защиты - документ, описывающий задачи обеспечения защиты информации в терминах функциональных требований и требований гарантированности.

Ҳимоя профили – ахборотни ҳимоялаш масалаларини функциональ талаблар ва кафолатланиш талаблари атамаларида тавсифловчи хужжат.

Protection Profile - document describing the task of ensuring the protection of information in terms of the functional requirements and the requirements of the warranty.

Разграничение доступа - совокупность методов, средств и мероприятий, обеспечивающих защиту данных от несанкционированного доступа пользователей.

Фойдаланишни чеклаш - маълумотларни, фойдаланувчиларнинг рухсатсиз фойдаланишларидан ҳимоялашни таъминловчи усуллар, воситалар ва тадбирлар мажмуи.

Access control - a set of methods, tools and measures to ensure the protection of data from unauthorized users.

Разделение привилегий - принцип открытия механизма защиты данных, при котором для доступа к ним необходимо указать не один, а два пароля (например, двумя лицами).

Имтиёзларнинг бўлиниши - маълумотлардан фойдаланиш учун битта эмас, балки иккита паролни кўрсатиш (масалан, иккита шахс паролини) лозим бўлган маълумотларни ҳимоялаш механизмини очиш принципи.

Privilege sharing - the principle of the opening mechanism of protection of data in which to access them you must specify not one, but two passwords (for example, two persons).

Резидентный - постоянно присутствующий в оперативной памяти.

Резидент - асосий хотирада доимо мавжуд.

Resident - constantly present in memory.

Сервер-посредник - брандмауэр, в котором для преобразования IP-адресов всех авторизованных клиентов в IP-адреса, ассоциированные с брандмауэром, используется процесс, называемый трансляцией адресов (address translation).

Сервер-воситачи - брандмауэр бўлиб, унда барча авторизацияланган мижозларнинг IP-адресларини брандмауэр билан боғлиқ IP-адресларга ўзгартириш учун адресларни трансляциялаш (adress translation) деб аталувчи жараёндан фойдаланилади.

Proxy server - firewall, in which to convert the IP-addresses of all authorized clients in IP-addresses associated with a firewall, use a process called NAT (address translation).

Стратегия защиты - формальное определение критериев, особенно оперативных, которыми следует руководствоваться при обеспечении защиты системы от известных угроз.

Ҳимоялаш стратегияси - тизимнинг маълум тахдидлардан ҳимоялашни таъминлашда амал қилиниши лозим бўлган мезонларни, айниқса, оператив мезонларни расмий аниқлаш.

Security strategy - a formal definition of the criteria, particularly operational, to be followed while protecting the system against known threats.

Тип доступа - сущность права доступа к определенному устройству, программе, файлу и т.д. (обычно read, write, execute, append, modify, delete).

Фойдаланиш тури - маълум қурилмадан, дастурдан, файлдан ва ҳ. фойдаланиш ҳуқуқининг маъноси (одатда read, write, execute, append, modify, delete).

Access type - essence of the right of access to a particular device, programs, files, etc. (usually read, write, execute, append, modify, delete).

Угроза (безопасности информации) — совокупность условий и факторов, создающих потенциальную или реально существующую опасность нарушения безопасности информации.

Тахдид (ахборот хавфсизлигига тахдид) - ахборот хавфсизлигини бузувчи потенциаль ёки реал мавжуд хавфни туғдирувчи шароитлар ва омиллар мажмуи.

Threat - set of conditions and factors that create potential or actual violations of the existing danger of information security.

Управление доступом - определение и ограничение доступа пользователей, программ и процессов к данным, программам и устройствам вычислительной системы.

Фойдаланишни бошқариш - фойдаланувчиларнинг, дастурларнинг ва жараёнларнинг маълумотлардан, ҳисоблаш техникаси дастурлари ва қурилмаларидан фойдаланишларини белгилаш ва чеклаш.

Access control - definition and limitation of access users, programs, and processes the data, programs, and devices of a computer system.

Фальсификация - использование различных технологий для обхода систем управления доступом на основе IP-адресов с помощью маскирования под другую систему, используя ее IP-адрес.

Сохталаштириш - бошқа тизим IP-адресидан фойдаланиб, унга ўхшаб ниқобланиш ёрдамида IP-адреслар асосида фойдаланишни бошқариш тизимини четлаб ўтиш учун турли технологиялардан фойдаланиш.

Spoofing - the use of different technologies to bypass access control systems, IP-based addresses using masking under another system using its IP-address.

Фишинг — технология интернет-мошенничества, заключающаяся в краже личных конфиденциальных данных, таких как пароли доступа, данные банковских и идентификационных карт и т.д.

Фишинг - фойдаланиш пароли, банк ва идентификация карталари маълумотлари ва ҳ. каби шахсий конфиденциаль маълумотларни ўғрилашдан иборат интернет-фирибгарлик технологияси.

Phishing - Internet-fraud technique, is used for stealing personal confidential data such as passwords, bank and identification cards, etc.

Фрод - обман; мошенничество, жульничество; подделка. Вид интернет-мошенничества, при котором мошенник самыми разными способами незаконно получает какую-то часть денег или услуг, относящихся к какому-либо сервису.

Фрод - инглизча fraud - алдаш, фирибгарлик, ғирромлик, қалбакилик Интернет-фирибгарлик тури бўлиб, фирибгар турли усуллар ёрдамида пулнинг ёки қандайдир серверга тегишли хизмат қисмига ноқонуний эга бўлади.

Fraud - deception; fraud scam; fake. Kind of Internet fraud in which the scammer in many ways unlawfully obtains some of the money or services relating to any service.

Хакер - пользователь, который пытается вносить изменения в системное программное обеспечение, зачастую не имея на это право. Хакером можно назвать программиста, который создает более или менее полезные вспомогательные программы, обычно плохо документированные и иногда вызывающие нежелательные побочные результаты.

Хакер - тизимли дастурий таъминотга, кўпинча ноқонуний ўзгартиришлар киритишга уринувчи фойдаланувчи. Одатда ёмон ҳужжатланган ва баъзида ножоиз қўшимча натижалар туғдирувчи озми-кўпми фойдали ёрдамчи дастурлар яратувчи дастурчини хакер деб аташ мумкин.

Hacker - a user who is trying to make changes to system software, often without that right. Can be called a hacker programmer who creates a more or less useful utility programs are usually poorly documented and sometimes cause unwanted side effects.

Целостность информации - способность средства вычислительной техники или системы автоматизированной обеспечивать неизменность информации в условиях случайного и/или преднамеренного искажения (разрушения).

Ахборот яхлитлиги - тасодифан ва/ёки атайин бузилиш ҳолларида ҳисоблаш техникаси воситаларининг ёки автоматлаштирилган тизимнинг ахборотни ўзгартирмаслигини таъминловчи хусусияти.

Information Integrity - the ability of computers and automated systems to provide consistent information in a casual and / or intentional distortion (destruction).

Ценность информации - свойство информации, определяемое ее пригодностью к практическому использованию в различных областях целенаправленной деятельности человека.

Ахборот қиммати –ахборотнинг инсоннинг мақсадли фаолиятининг турли соҳаларида амалий фойдаланишга яроқлиги орқали аниқланувчи хусусияти.

Information value - property information, determine its applicability to practical use in various fields of purposeful human activity.

Шлюз прикладного уровня - исключает прямое взаимодействие между авторизованным клиентом и внешним хостом. Шлюз фильтрует все входящие и исходящие пакеты на прикладном уровне модели OSI. Связанные с приложениями программы-посредники перенаправляют через шлюз информацию, генерируемую конкретными сервисами TCP/IP.

Илова сатҳи шлюзи - авторизациядан ўтган мижоз ва ташқи хост ўртасидаги тўғридан-тўғри ўзаро алоқа амалга ошишига йўл қўймайди. Шлюз OSI моделининг илова сатҳида кирувчи ва чиқувчи тармоқ пакетларининг барчасини фильтрлайди. Иловалар билан боғлиқ дастур-воситачилар TCP/IPнинг аниқ хизматлари генерациялайдиган ахборотни шлюз орқали узатилишини таъминлайди.

Application-level gateway - eliminates the direct interaction between an authorized client and the external host. Gateway filters all incoming and outgoing packets at the application layer model OSI. Application-related program intermediary redirect gateway information generated by a particular service TCP/IP.

Шлюз сеансового уровня - исключает прямое взаимодействие между авторизованным клиентом и внешним хостом. Он принимает запрос доверенного клиента на определенные услуги и, после проверки допустимости запрошенного сеанса, устанавливает соединение с внешним

хостом. После этого шлюз просто копирует пакеты в обоих направлениях, не осуществляя их фильтрации.

Сеанс сатҳи шлюзи - авторизациядан ўтган мижоз ва ташқи хост ўртасидаги тўғридан-тўғри ўзар оалоқа амалга ошишига йўл қўймайди. Шлюз ишончли мижоздан сўров қабул қилади ва сўралган сеансга рухсат этилганлигини текширувидан сўнг ташқи хост билан алоқани ўрнатади. Шундан сўнг иккала йўналишда тармоқ пакетларини фильтрламасдан нусҳа олади.

Circuit-level gateway - eliminates the direct interaction between an authorized client and the external host. It takes a trusted client request for certain services and, after validation of the requested session, establishes the connection with the external host. After this, the gateway simply copies the packets in both directions, not realizing their filtration.

Шпионское программное обеспечение — вид вредоносного программного обеспечения, осуществляющего деятельность по сбору информации о конфигурации компьютера, деятельности пользователя и любой другой конфиденциальной информации без согласия самого пользователя.

Хуфия дастурий таъминот – фойдаланувчиларни рухсатисиз компьютер конфигурациялари, фойдаланувчилар фаолияти ва ҳар қандай бошқа конфиденциаль ахборотни йиғиш бўйича фаолият олиб борадиган зарарли дастурий таъминот тури.

Spyware - type of malicious software, carrying out activities to collect information about your computer configuration, user activity, and any other confidential information without the consent of the user.

Экспертиза системы защиты информации - оценка соответствия представленных проектных материалов по защите информации (на объекте) поставленной цели, требованиям стандартов и других нормативных документов.

Ахборотни ҳимоялаш тизимининг экспертизаси - ахборотни ҳимоялаш бўйича тақдим этилган лойиҳа материалларининг қўйилган мақсади, стандартлар талабларига ва бошқа меъерий хужжатларга мослини баҳолаш.

Expert operation of the system of protection to information - conformity assessment submitted project materials for the protection of information (on-site) goal, the standards and other regulatory documents.

Эффективность - свойство объекта удовлетворять требованиям к услуге с заданными количественными характеристиками**.**

Самарадорлик – берилган миқдордаги характеристикалари билан хизмат кўрсатишга бўлган талабларни қондирувчи объектнинг хусусияти.

Efficiency - object property to satisfy the requirements of the service with the given quantitative characteristics.

Язык Java - новый язык программирования, разработанный Sun Microsystems, Inc. Он может использоваться как обычный язык программирования для разработки сетевых приложений. Кроме того, он используется для написания небольших приложений, называемых апплетами.

Java тили - Sun Microsystems, Inc. томонидан ишлаб чиқилган янги дастурлаш тили. У оддий тил сифатида тармоқ иловаларини ишлаб чиқиш учун қўлланилиши мумкин. Ундан ташқари у апплет деб аталувчи катта бўлмаган иловаларни ёзишда қўлланилади.

Language Java - a new programming language developed by Sun Microsystems, Inc. It can be used as a conventional programming language for development of network applications. In addition, it is used for writing small applications called applets.

Язык администрирования базы данных – искусственный язык для описания действий, связанных с администрированием базы данных.

Маълумотлар базасини маъмурлаш тили – маълумотлар базасини маъмурлаш билан боғлиқ ҳаракатларни тавсифлаш учун қўлланиладиган сунъий тил.

Database administration language - artificial language to describe actions related to database administration.

Язык базы данных - искусственный язык для описания процессов создания, ведения и использования баз данных.

Маълумотлар базаси тили – маълумотлар базасини яратиш, юритиш ва қўллаш жараёнларини тавсифлаш учун қўлланиладиган сунъий тил.

Database language - artificial language to describe the creation, maintenance and use of databases.

Язык запросов - искусственный язык для описания запросов, поиска данных в базах данных и действий над запросами.

Сўровлар тили – маълумотлар базасида маълумотларни қидириш сўровлари ва улар устида амаллар бажаришни тавсифлашда қўлланиладиган сунъий тил.

Query language - artificial language to describe the query, retrieval of data in databases and actions on requests.